



รายงานวิจัยฉบับสมบูรณ์

โครงการ การศึกษาวิธีการการควบคุมความแออัดของข้อมูลแบบ ABR ใน ATM เน็ตเวอร์ค

โดย

ผศ.ดร.ลัญจนกร วุฒิสัทติกุลกิจ

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

จพ
วศ 15
010358

มิถุนายน 2543



รายงานวิจัยฉบับสมบูรณ์

โครงการ การศึกษาวิธีการการควบคุมความแออัดของข้อมูลแบบ ABR ใน ATM เน็ตเวิร์ค

โดย

ผศ.ดร.ลัญจกร วุฒิสัทติกุลกิจ

จุฬาลงกรณ์มหาวิทยาลัย

มิถุนายน 2543



รายงานวิจัยฉบับสมบูรณ์

โครงการ การศึกษาวิธีการการควบคุมความแออัดของข้อมูลแบบ ABR ใน ATM เน็ตเวอร์ค

คณะผู้วิจัย

1. ผศ.ดร.ลัญจกร วุฒิสัทติกุลกิจ

สังกัด

ภาควิชาวิศวกรรมไฟฟ้า
จุฬาลงกรณ์มหาวิทยาลัย

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

สถาบันวิทยบริการ
ชุดโครงการ.....

จุฬาลงกรณ์มหาวิทยาลัย

เลขหมู่ กฟ ๑๙ 15
เลขทะเบียน ๐10358
วัน,เดือน,ปี ๒๕๖๓.๓.๔๓

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณท่าน ศ.ดร.ประสิทธิ์ ประพัฒน์มงคลการ หัวหน้าห้องปฏิบัติการวิจัย Telecommunication System Research Laboratory (TSRL) ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เป็นอย่างสูงที่ให้การสนับสนุนด้านอุปกรณ์วิจัยและสถานที่ พร้อมให้คำปรึกษาทางด้านวิชาการและการถ่ายทอดประสบการณ์การทำงานวิจัยที่เป็นประโยชน์อย่างยิ่งต่อโครงการวิจัยนี้ ผู้วิจัยขอขอบคุณ นายศุภเกียรติ เล่าหะเมทธิ ซึ่งเป็นบุคคลที่มีส่วนสำคัญต่อความสำเร็จของโครงการนี้ และท้ายสุดนี้ผู้วิจัยขอขอบคุณสำนักงานกองทุนสนับสนุนการวิจัยที่ให้ความสนับสนุนโครงการวิจัยนี้เป็นอย่างดี



บทคัดย่อ

รหัสโครงการ	PDF/11/2541
ชื่อโครงการ	การศึกษาวิธีการการควบคุมความแออัดของข้อมูลแบบ ABR ใน ATM เน็ตเวอร์ค
คณะผู้วิจัย	ผศ.ดร.ลัญจกร วุฒิสัทติกุลกิจ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
E-mail :	lunch@ee.eng.chula.ac.th
ระยะเวลาที่ทำการวิจัย	1 กรกฎาคม 2541 – 30 มิถุนายน 2543
วัตถุประสงค์	ศึกษาพัฒนาและออกแบบวิธีการควบคุมความแออัดของข้อมูล ABR แบบหนึ่งจุดไปหลายจุดในโครงข่าย ATM
ระเบียบวิธีการวิจัย	งานวิจัยนี้มุ่งเน้นการพัฒนาวิธีการแบบใหม่สำหรับใช้ควบคุมความแออัดของข้อมูล ABR ในโครงข่าย ATM โดยเน้นปัญหาการรวบรวมข่าวสารความคับคั่งในแต่ละสวิตชิงโหนด ในการทดสอบเพื่อเปรียบเทียบสมรรถนะของแต่ละวิธีอาศัยการจำลองระบบด้วยโปรแกรมคอมพิวเตอร์ โดยประสิทธิภาพการทำงานของแต่ละวิธีพิจารณาจากความถูกต้องและความเร็วของการป้อนกลับข่าวสารเพื่อใช้ในการควบคุมอัตราการส่งของแหล่งข้อมูล ABR เพื่อให้ได้ระบบที่มีเสถียรภาพและมีวิสัยสามารถสูงสุด
ผลการวิจัย	งานวิจัยนี้ได้ค้นพบวิธีการใหม่ที่สามารถแก้ไขปัญหาคับคั่งที่ติดอยู่ได้อย่างสมบูรณ์ โดยโครงข่ายจะสามารถป้อนกลับข่าวสารการควบคุมได้อย่างรวดเร็วและถูกต้องสมบูรณ์ทำให้โครงข่ายสามารถใช้ประโยชน์จากทรัพยากรที่มีอยู่จำกัดได้อย่างมีประสิทธิภาพและไม่มีปัญหาของความไม่เสถียรภาพ
สรุปและอภิปรายผล	ผลจากการวิจัยได้พัฒนาวิธีการรวบรวมข่าวสารความแออัดคับคั่งของข้อมูล ABR สำหรับการต่อเชื่อมแบบหนึ่งจุดไปหลายจุดในโครงข่าย ATM แบบใหม่ 2 วิธี วิธีแรกให้สมรรถนะที่ดีที่สุดและให้ค่าตอบที่ถูกต้องที่สุดกว่าวิธีการที่ได้มีผู้เสนอมาก่อนทั้งหมด ส่วนวิธีที่สองให้ผลที่ใกล้เคียงกับวิธีแรกมากแต่มีความซับซ้อนน้อยกว่ามาก
ข้อเสนอแนะ	วิธีการที่พัฒนาขึ้นมีความสมบูรณ์ถูกต้องสามารถใช้งานได้มีประสิทธิภาพดีแล้ว ดังนั้นปัญหาที่น่าจะทำการวิจัยเพิ่มเติมจึงเป็นส่วนของการทำงานร่วมกันระหว่างวิธีการควบคุมความคับแบบต่าง ๆ กัน
คำสำคัญ	โครงข่าย ATM บริการ ABR วิธีการควบคุมความแออัด วิธีการรวบรวมข่าวสารความคับคั่ง

Abstract

- Project Code :** PDF/11/2541
- Project Title :** A Study of Congestion Control Algorithms for Available Bit Rate (ABR) Services in ATM Networks
- Investigators :** Assistant Professor Dr. Lunchakorn Wuttisittikulij
Department of Electrical Engineering
Faculty of Engineering Chulalongkorn University
- E-mail :** lunch@ee.eng.chula.ac.th
- Project Period :** 1 กรกฎาคม 2541 – 30 มิถุนายน 2543
- Objectives :** To study develop and design a congestion control technique for point-to-multipoint ABR traffic in ATM networks
- Methodology :** This research aims to develop a new technique for congestion control of ABR traffic in ATM networks with an emphasis on the problem of feedback signaling consolidation algorithms in each switching node. Computer simulations are used to evaluate the performance of these algorithms. The efficiency of each algorithm is measured from the feedback speed and correctness for controlling transmission rate of ABR sources so as to achieve the maximum throughput system without the stability problem.
- Results :** This research has resulted in new algorithms that can completely solve all the existing problems. The network is able to send the feedback control information at the maximum speed with accurate information, allowing an effective network resource utilization and causing no instability problem.
- Discussion Conclusion :** In this research, two new congestion information consolidation algorithms for ABR traffic in point-to-multipoint environment have been developed. The first algorithm offers optimal solution to this problem and hence it is superior to all previously known techniques. Whereas the second algorithm can result in near-optimum but with much less complexity.
- Suggestions :** The developed algorithms is complete and highly effective. Therefore, the next interesting issue is the interoperability between different congestion control algorithms.
- Keywords :** ATM networks, ABR services, Congestion control and Consolidation algorithm

Executive Summary

In this report, an extensive study of congestion control for Available Bit Rate (ABR) services in ATM Networks is presented. At present, the point-to-multipoint ABR (Available Bit Rate) service in ATM networks is considered important for supporting many emerging applications, such as data broadcasting over subscribed members and Internet multicast networking. The main problem of point-to-multipoint ABR service in ATM network is that there exists a number of feedback information from all the branch points sent back to the source through BRM cells. This causes a serious problem of how to handle such a large amount of feedback information. A common technique to this problem is that each branch point in the multicast tree consolidates the information in BRM cells received from downstream branch points or destinations and forwards the selected one to its upstream switch. This significantly reduces the amount of feedback information in the system. The main challenge of this problem is how to consolidate BRM cells in such a way that the transient response of each traffic source can be fast while avoiding undesired consolidation noise. In previous studies, no known consolidation algorithm is able to achieve both requirements simultaneously. Some algorithms provide very fast transient response, but they suffer seriously from the problem of consolidation noise. Whereas the others can eliminate consolidation noise, but under certain traffic conditions they react rather slowly to traffic changes. In this investigation, we present a complete solution to this problem. The key of this new algorithm lies in the way in which feedback information is stored at each switch, *i.e.* information is kept on a per-branch basis for each VC. The results show that this proposed consolidation algorithm completely remove consolidation noise and provides very fast transient response in both overload and underload conditions. However, the proposed algorithm adds more complexity to the switches, due to extra variables being needed. Consequently, an alternative algorithm with much less complexity is developed. It is shown that this new algorithm offers similar performance to the first while the number of variables needed at each switch is kept to minimum.

สารบัญ

บทที่ 1 บทนำ.....	1
1.1 บทนำ.....	1
1.2 องค์ประกอบของโครงข่าย ATM.....	1
1.3 ขั้นตอนการติดต่อขอเริ่มการสื่อสาร.....	2
1.4 การแบ่งชั้น โปรโตคอล.....	3
1.5 ชั้นฟิสิคอลล.....	4
1.6 ATM Layer.....	5
1.7 ประเภทของการให้บริการ.....	6
1.8 พื้นฐานการควบคุมการรับส่งข้อมูล.....	8
1.9 กลไกการป้อนกลับสัญญาณควบคุม.....	9
1.10 โครงสร้างของเซลล์ RM.....	9
1.11 คุณลักษณะการให้บริการแบบ ABR.....	11
1.12 การทำงานของแหล่งกำเนิดข้อมูล.....	12
1.13 การผลิตและค่าเริ่มต้นของเซลล์ RM.....	13
1.14 การทำงานของสวิตช์.....	13
1.15 ขอบเขตของงานวิจัย.....	15
1.16 ประโยชน์ที่ได้รับ.....	16
บทที่ 2 การควบคุมทราฟฟิกในโครงข่ายเอทีเอ็ม.....	17
2.1 การควบคุมทราฟฟิกในโครงข่าย ATM.....	17
2.2 การควบคุมการส่งข้อมูลบนการให้บริการแบบ ABR ในโครงข่าย ATM.....	18
2.3 คุณลักษณะของแหล่งกำเนิดข้อมูล.....	20
2.4 วิธีควบคุมความคับคั่ง.....	23
2.4.1 วิธีการป้อนกลับบิตเดียว (Binary Feedback Schemes).....	24
2.4.2 วิธีการป้อนกลับค่าอัตราส่งแน่นอน (Explicit Rate Feedback Schemes).....	24
2.5 วิธี ERICA (Explicit Rate Indicate for Congestion Avoidance).....	25
2.6 วิธี FMMRA (Fast Max-Min Rate Allocation).....	29

บทที่ 3 การส่งข้อมูลประเภท ABR จากหนึ่งจุดไปสู่หลายจุดในโครงข่ายเอทีเอ็ม	32
3.1 กล่าวนำ	32
3.2 การควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุดสำหรับการให้บริการแบบ ABR	33
3.3 วิธีรวบรวมข่าวสารความคับคั่ง	35
3.3.1 Robert algorithm	36
3.3.2 TS algorithm	37
3.3.3 RSS algorithm	40
3.3.4 Wait-for-all algorithm	42
3.3.5 Immediate Rate Calculation algorithm	44
3.4 ปัญหาการทำงานของวิธีควบคุมความคับคั่งร่วมกับวิธีรวบรวมข่าวสารความคับคั่ง	47
3.5 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 1	48
3.6 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 2	50
บทที่ 4 ผลการจำลองแบบ	52
4.1 การจำลองสำหรับ One hop configuration	52
4.1.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ	52
4.1.2 สมรรถนะของ วิธี ERICA	52
4.1.3 สมรรถนะของวิธี FMMRA	55
4.1.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA	58
4.1.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ	58
4.2 ผลการจำลองสำหรับ Chain configuration	59
4.2.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ	59
4.2.2 สมรรถนะของ วิธี ERICA	59
4.2.3 สมรรถนะของวิธี FMMRA	65
4.2.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA	68
4.2.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ	68
4.3 ผลการจำลองสำหรับ Chain configuration with low ICR	69
4.3.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ	69
4.3.2 สมรรถนะของ วิธี ERICA	69

4.3.3	สมรรถนะของวิธี FMMRA.....	72
4.3.4	เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA.....	75
4.3.5	เปรียบเทียบประสิทธิภาพระหว่าง วิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ.....	67
4.4	ผลการจำลองสำหรับ Jiang configuration.....	76
4.4.1	รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ.....	76
4.4.2	สมรรถนะของ วิธี ERICA.....	76
4.4.3	สมรรถนะของวิธี FMMRA.....	80
4.4.4	เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA.....	81
4.4.5	เปรียบเทียบประสิทธิภาพระหว่าง วิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ.....	81
4.5	ผลการจำลองสำหรับ VBR Chain configuration.....	82
4.3.1	รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ.....	82
4.3.2	สมรรถนะของ วิธี ERICA.....	83
4.3.3	สมรรถนะของวิธี FMMRA.....	86
4.3.4	เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA.....	90
4.3.5	เปรียบเทียบประสิทธิภาพระหว่าง วิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ.....	90
บทที่ 5 บทสรุปและข้อเสนอแนะ.....		91
5.1	เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA.....	91
5.1.1	Max-Min fairness.....	91
5.1.2	ผลตอบสนองชั่วคราว.....	91
5.1.3	Consolidation noise.....	91
5.1.4	ความซับซ้อนในการพัฒนา.....	92
5.1.5	ความเหมาะสมในการทำงานกับ วิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ.....	92
5.2	เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่ง.....	93
5.2.1	Consolidation noise.....	93
5.2.2	ผลตอบสนองชั่วคราวในสภาวะโหลดเกิน.....	94
5.2.3	ผลตอบสนองชั่วคราวในสภาวะโหลดต่ำ.....	94
5.2.4	ความซับซ้อนในการพัฒนา.....	95

5.2.5	RM cell Overhead.....	96
5.2.6	Change to non responsive branch.....	97
5.2.7	Change to newly joined branch.....	97
5.3	ข้อเสนอแนะ.....	98
เอกสารอ้างอิง.....		98
ภาคผนวก.....		101



จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ



1.1 บทนำ

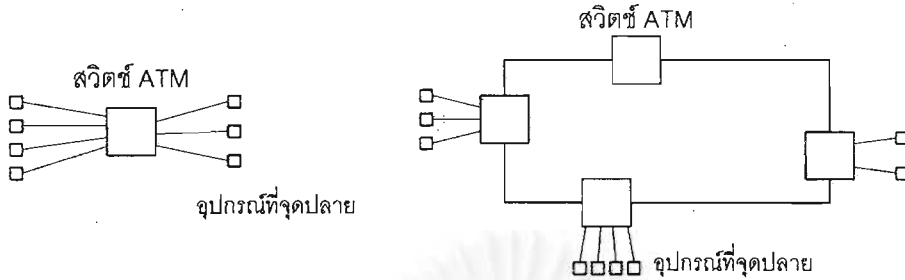
Asynchronous Transfer Mode (ATM) เป็นเทคโนโลยีการรับและส่งข้อมูลแบบใหม่ที่ได้รับ การออกแบบมาเพื่อให้สามารถรองรับการสื่อสารข้อมูลชนิดต่าง ๆ เช่น เสียง (voice) ภาพ (image) วิดีโอ (video) ข้อมูลประเภทคำ (data) และ มัลติมีเดีย (multimedia) ภายในโครงข่ายเดียวกันได้ หลักการของระบบ ATM อาศัยการแบ่งข้อมูลที่จะส่งออกเป็นเซลล์เล็ก ๆ ขนาด 53 ไบต์ โดยมีลักษณะการส่งคล้ายกับโครงข่ายแพ็กเก็ตสวิตซิง (packet switched networks) การที่กำหนดให้เซลล์มีขนาดเล็กเพื่อ ที่ให้การส่งข้อมูลมีเวลาประวิง (delay) น้อย ซึ่งหมายความว่าโครงข่ายสามารถใช้รองรับการส่งข้อมูล ประเภทเรียลไทม์ (real time) เช่น สัญญาณเสียงได้ และการที่กำหนดให้เซลล์มีขนาดคงที่ก็เพื่อที่จะทำ ให้การออกแบบสวิตซ์ง่ายขึ้น ซึ่งช่วยให้การสร้างระบบโครงข่ายสามารถใช้ฮาร์ดแวร์ (hardware) เป็น องค์ประกอบหลักได้ ปัจจัยเหล่านี้ทำให้โครงข่าย ATM มีขีดความสามารถในการรับส่งข้อมูลที่มีความเร็วสูงมาก ๆ ได้โดยที่ยังสามารถควบคุมขนาดของดีเลย์ให้อยู่ในระดับที่เหมาะสม

ในปีค.ศ. 1989 ระบบ ATM ได้รับเลือกให้เป็นเทคโนโลยีพื้นฐานที่เหมาะสมกับการนำไปใช้ ในการพัฒนาระบบสื่อสาร B-ISDN (Broadband-Integrated Services Digital Networks) สำหรับอนาคต จากองค์การระหว่างประเทศ ITU-T และต่อมาในปีค.ศ. 1991 กลุ่มบริษัทผู้ผลิตอุปกรณ์ทางด้านสื่อสาร โทรคมนาคมได้รวมตัวกันขึ้นเพื่อพัฒนาและกำหนดทิศทางของเทคโนโลยี ATM เพื่อให้อุปกรณ์ที่แต่ละบริษัทผลิตขึ้นสามารถทำงานร่วมกันได้ภายใต้มาตรฐานเดียวกัน กลุ่มดังกล่าวมีชื่อเรียกว่า The ATM Forum ในช่วงหลายปีที่ผ่านมากลุ่มนี้มีจำนวนสมาชิกเพิ่มมากขึ้นอย่างรวดเร็ว (จากสมาชิกเพียง 4 ในช่วงเริ่มแรก จนมีสมาชิกมากกว่า 900 ในช่วงเวลา เพียง 5 ปี ในปัจจุบันกลุ่มนี้ก็ยังคงมีบทบาทสำคัญ อย่างยิ่งต่อการร่างข้อกำหนดต่าง ๆ ของเทคโนโลยี ATM อย่างต่อเนื่อง หมายเหตุ The ATM forum ไม่ใช่ องค์การระหว่างประเทศที่ทำหน้าที่กำหนดมาตรฐาน ดังนั้นข้อกำหนดที่ร่างขึ้นจึงมิใช่มาตรฐานสากลที่แท้จริง หากแต่ข้อกำหนดเหล่านี้ได้รับการยอมรับอย่างแพร่หลายจากองค์การหรือบริษัทที่เกี่ยวข้อง ด้วยเหตุนี้ข้อกำหนดที่ออกโดย The ATM forum จึงอาจจะกล่าวว่าเป็นมาตรฐานระหว่างประเทศ โดยนัย

1.2 องค์ประกอบของโครงข่าย ATM

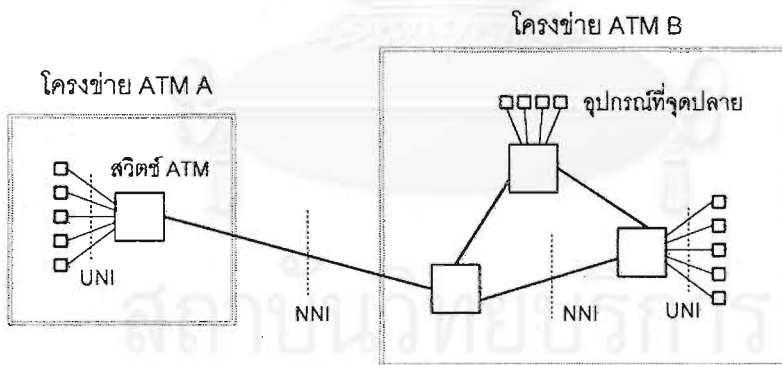
โครงข่าย ATM ประกอบด้วย 2 ส่วนหลักคือ ส่วนของสวิตซ์ (switch) และ อุปกรณ์ที่จุดปลาย (endpoint) สวิตซ์นั้นทำหน้าที่เชื่อมต่ออุปกรณ์ที่จุดปลายหลาย ๆ จุดเข้าด้วยกัน ซึ่งสำหรับในระบบ โครงข่ายขนาดเล็ก ๆ แล้ว อาจจะใช้สวิตซ์เพียงตัวเดียวในการต่อเชื่อมอุปกรณ์ที่จุดปลายทั้งหมดได้ แต่ ถ้าหากเป็นโครงข่ายที่มีขนาดใหญ่ขึ้นจำเป็นจะต้องมีการใช้สวิตซ์หลาย ๆ ตัวมากขึ้น โดยที่สวิตซ์เหล่านี้ก็จะต่อเชื่อมถึงกันเป็น โครงสร้างรูปแบบ ๆ หนึ่ง ดูตัวอย่างของโครงข่าย ATM ในรูปที่ 1.1 ประกอบ

ส่วนตัวอย่างของอุปกรณ์ที่จุดปลายที่ต่อเชื่อมอยู่กับสวิตช์ ATM ก็มีอยู่มากมายหลายชนิด เช่น เครื่องคอมพิวเตอร์ประเภทต่าง ๆ PCs workstations และ supercomputers กล้องวิดีโอ โมเด็ม xDSL เป็นต้น



รูปที่ 1.1 ตัวอย่างของโครงข่าย ATM

มาตรฐานการเชื่อมต่อของอุปกรณ์สื่อสารในระบบ ATM สามารถแบ่งออกได้เป็น 2 กลุ่ม คือ user-network interface (UNI) และ network-node interface (NNI) จากรูปที่ 1.2 ประกอบ UNI เป็นส่วนของการเชื่อมต่อระหว่างอุปกรณ์จุดปลายกับสวิตช์ ATM หรือก็คือจุดต่อระหว่างผู้ใช้กับโครงข่าย ATM นั้นเอง ส่วน NNI ก็คือการเชื่อมต่อระหว่างโครงข่าย ATM 2 โครงข่าย หรือระหว่างอุปกรณ์สวิตช์ ATM 2 ชุด จุดเชื่อมต่อทั้งสองแบบมีหน้าที่แตกต่างกัน ส่วนของ UNI มีหน้าที่ติดต่อกับผู้ใช้บริการในช่วงการขอเริ่มการติดต่อสื่อสาร ช่วงที่มีการรับส่งข้อมูล และการสิ้นสุดการติดต่อสื่อสาร ส่วนจุดเชื่อมต่อ NNI ทำหน้าที่รับส่งข้อมูลทั้งในส่วนของผู้ใช้และสัญญาณควบคุม ระหว่างสวิตช์ที่ติดกันหรือระหว่างโครงข่าย

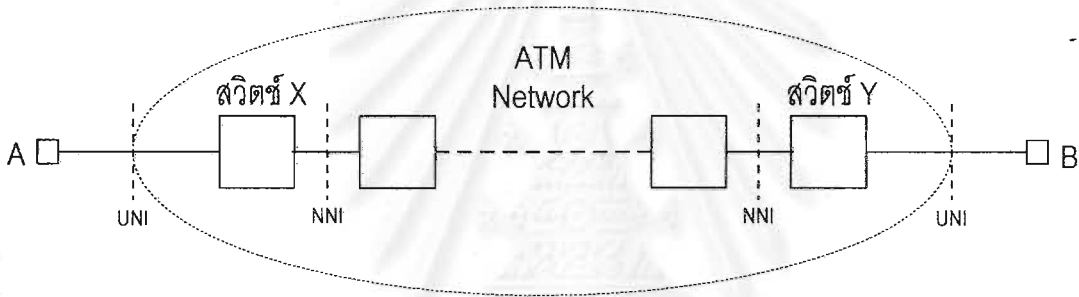


รูปที่ 1.2 มาตรฐานการเชื่อมต่อ UNI และ NNI ภายในโครงข่าย ATM

1.3 ขั้นตอนการติดต่อขอเริ่มการสื่อสาร (call establishment)

สมมติว่าอุปกรณ์สื่อสารที่จุดปลาย A ต้องการขอเริ่มการติดต่อกับอุปกรณ์สื่อสาร B เช่น ต้องการขอประชุมทางวิดีโอโดยผ่านโครงข่าย ATM (ดูรูปที่ 1.3 ประกอบ) ทั้งสองฝ่ายจะต้องมีการตกลงกันในรายละเอียดต่าง ๆ ถึงความพร้อมของแต่ละฝ่ายก่อนที่จะมีการส่งสัญญาณวิดีโอระหว่างกันได้ ซึ่งมีขั้นตอนในการทำดังต่อไปนี้ เริ่มแรกจะต้องมีการกำหนดและสร้าง VC (virtual connection) ระหว่างอุปกรณ์สื่อสาร A และ B ขึ้นก่อน โดย A จะแจ้งความจำนงขอสร้างช่องสัญญาณเชื่อมต่อกับ B

ไปที่โครงข่าย ATM ซึ่งส่วนของโครงข่ายก็จะทำการจัดสรรแบนด์วิดท์ส่วนหนึ่งในโครงข่ายที่มีขนาดใหญ่เพียงพอที่จะรองรับการประชุมทางวิดีโอได้ตามคุณภาพของบริการที่ A ต้องการ พิจารณารูปที่ 1.3 A จะติดต่อขอเชื่อมต่อกับ B โดยการส่งสัญญาณซิกเนลถึงไปที่สวิตช์ X โดยอาศัยโปรโตคอล UNI สวิตช์ X ก็จะทำการติดต่อกับสวิตช์ตัวอื่น ๆ เพื่อพิจารณาว่าภายในโครงข่ายยังมีแบนด์วิดท์เหลือพอที่จะเชื่อมต่อระหว่าง A กับ B หรือไม่ ในการติดต่อส่วนนี้ก็จะอาศัยโปรโตคอล NNI ถ้าหากมีแบนด์วิดท์เพียงพอก็จะต่อเชื่อมระหว่างสวิตช์ X และ Y เข้าด้วยกัน แล้วสวิตช์ก็ทำการส่งสัญญาณซิกเนลถึงไปที่ B เพื่อบอกถึงการขอเชื่อมต่อของ A กับ B ถ้าหาก B มีความประสงค์จะติดต่อกับ A และอยู่ในสถานะที่จะทำการติดต่อได้ B ก็จะส่งสัญญาณซิกเนลถึงตอบรับกลับไปไปที่สวิตช์ Y ซึ่งสัญญาณนี้จะถูกส่งต่อกลับไปไปที่ A โดยผ่านสวิตช์ X ตามโปรโตคอลของ ATM นั้นได้กำหนดไว้ว่าเซลล์ทุกเซลล์ของการติดต่อระหว่างอุปกรณ์คู่ใดคู่หนึ่งจะต้องถูกส่งผ่านบนเส้นทางเดียวกันเสมอทั้งขาไปและขากลับ นอกจากนี้การส่งของแต่ละเซลล์จะต้องเรียงตามลำดับที่ถูกต้องเสมอ จากที่กล่าวมาทั้งหมดนี้จะเห็นว่ารูปร่างการสื่อสารในโครงข่าย ATM เป็นการสื่อสารประเภท Connection-Oriented



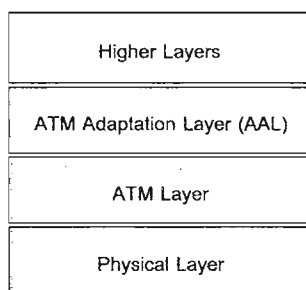
รูปที่ 1.3 การขอเริ่มการติดต่อระหว่าง A กับ B ผ่านโครงข่าย ATM

1.4 การแบ่งชั้นโปรโตคอล

โปรโตคอลในระบบ ATM สามารถแบ่งออกได้เป็น 4 ชั้น (layer) คือ

1. physical layer
2. ATM adaptation layer (AAL)
3. ATM layer
4. higher layer protocols

แต่ละชั้นโปรโตคอลจะมีหน้าที่แตกต่างกันอย่างชัดเจน เช่น physical layer กำหนดรูปแบบและเทคโนโลยีที่ใช้ในการรับส่งข้อมูล ตัวอย่างโปรโตคอลที่สามารถนำมาใช้ได้คือ SONET หรือ SDH ชั้นของ AAL มีหน้าที่ในการแปลงข้อมูลของผู้ใช้บริการแต่ละประเภทให้อยู่ในรูปแบบที่เหมาะสมกับการส่งผ่านโครงข่าย ATM ส่วนชั้น higher layer protocols จะเป็นส่วนของโปรโตคอลรูปแบบอื่น ๆ ที่อาจจะนำมาส่งผ่านโครงข่าย ATM หรือจะเป็นโปรโตคอลของ ATM เองในชั้นที่สูงขึ้น



รูปที่ 1.4 การแบ่งชั้นโปรโตคอลในระบบ ATM

1.5 ชั้นฟิสิกอล (Physical Layer)

ในชั้นฟิสิกอลประกอบด้วยชั้นย่อย 2 ส่วนคือ

1. lower physical media dependent (PMD) sublayer
2. higher transmission convergence (TC) sublayer

ในชั้นย่อย PMD เป็นส่วนที่เกี่ยวข้องกับการกำหนดชนิดของสื่อกลางที่ใช้ในการส่งบิตข้อมูล เช่น อาจจะเป็น UTP, coaxial cable, fiber หรือ wireless และในแต่ละชนิดของสื่อกลางที่ใช้ในชั้นย่อย PMD ก็ยังระบุถึงรายละเอียดของวิธีการส่งบิตข้อมูลที่เหมาะสมกับสื่อกลางชนิดนั้น ๆ ด้วย เช่น bit transfer และ bit alignment นอกจากนี้ยังจะอาจรวมไปถึงวิธีที่ใช้สำหรับทำ line coding, opto-electronic conversion, modulation และ demodulation จะเห็นได้ว่าชั้น PMD เป็นส่วนที่เกี่ยวข้องโดยตรงกับสื่อกลางที่ใช้โดยจะกำหนดรายละเอียดของการรับส่งบิตข้อมูลที่แตกต่างกันสำหรับสื่อกลางแต่ละชนิด

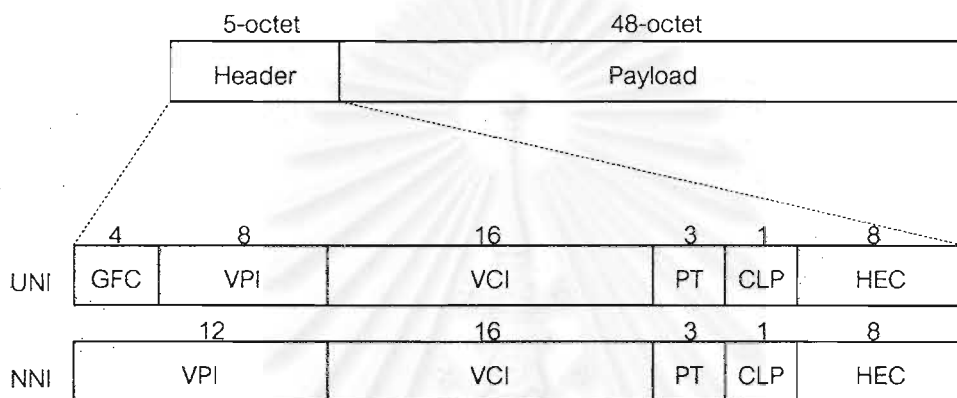
ส่วนของชั้นย่อย TC จะไม่สนใจกับลักษณะการส่งหรือชนิดของสื่อกลาง แต่จะทำหน้าที่จัดการกับเซลล์ที่มาจาก ATM layer ให้สามารถส่งผ่านสื่อกลางนั้น ๆ ได้ ดังนั้นจะเห็นว่าชั้นย่อยนี้เองที่ทำให้การทำงานของ ATM layer ไม่ได้ขึ้นอยู่กับชนิดของสื่อกลางที่ใช้เลย ตัวอย่างของชั้นย่อย TC คือ SONET

ตารางที่ 1.1 ตัวอย่างมาตรฐานการติดต่อ UNI ใน Physical layer ที่กำหนดโดย ATM forum

Private Network				
Bit Rate	PMD: Physical Media	Distance	TC	Comments
155.52 Mbps	UTP-Category 5 (data grade)	100 m	STS-3c	NRZ
155.52 Mbps	Single mode fiber	2 km	STS-3c	
	Multimode fiber	2 km		
	Coaxial pair	900 ft.		
622.08 Mbps	Single mode fiber	2 km	STS-12	
Public Network				
Bit Rate	PMD: Physical Media	Distance	TC	Comments
155.52 Mbps	Single mode fiber	15 km	STS-3c	
622.08 Mbps	Single mode fiber	15 km	STS-12	

1.6 ATM Layer

โครงสร้างของเซลล์ ATM ประกอบด้วย 2 ส่วนคือ ส่วนหัวหรือเฮดเดอร์ (header) และส่วนของข้อมูล (payload) ดังที่แสดงในรูปที่ 1.5 ในส่วนหัวนั้นมีความยาวเท่ากับ 5 octet และส่วนของข้อมูลมีความยาวเท่ากับ 48 octet สังเกตว่ารูปแบบของเซลล์มีอยู่ 2 ประเภททั้งนี้ขึ้นอยู่กับลักษณะและจุดหรือบริเวณของการใช้งานว่าเป็น UNI หรือ NNI โดยจะแตกต่างกันก็เฉพาะตรงส่วนของ 4 บิตแรก ในกรณีของ UNI จะใช้ทำหน้าที่เป็น GFC (Generic Flow Control) ซึ่งยังมีได้มีการกำหนดเป็นมาตรฐานการใช้งานที่แน่นอน ส่วนในกรณีของ NNI จะกำหนดให้เป็นส่วนหนึ่งของ VPI (Virtual Path Identifier)



รูปที่ 1.5 โครงสร้างของเซลล์ ATM

แต่ละส่วนของเฮดเดอร์ในเซลล์มีหน้าที่ดังต่อไปนี้

1. Virtual Connection Identifier (VCI) เป็นค่าที่กำหนดให้สำหรับการเชื่อมต่อสื่อสารของผู้ใช้ โดยแต่ละการเชื่อมต่อก็จะได้ค่า VCI ที่แตกต่างกันไป
2. Virtual Path Identifier (VPI) เป็นค่าที่ใช้งานในลักษณะที่คล้ายกันกับ VCI หากแต่ VPI ค่า ๆ หนึ่งใช้ในการระบุถึงกลุ่มของ VCI จำนวนหนึ่ง
3. The payload type (PT) เป็นตัวบ่งบอกว่าเซลล์นี้เป็นเซลล์ประเภทใด คือเป็นเซลล์ข้อมูลของผู้ใช้หรือเป็นเซลล์ที่บรรจุสัญญาณควบคุมเพื่อใช้ในการจัดการภายในโครงข่าย
4. The cell loss priority (CLP) มีความยาวเพียง 1 บิต ถ้าบิตนี้ถูกเซตเป็น 1 เซลล์เหล่านี้จะถือว่าเป็นเซลล์ที่มีความสำคัญไม่มากนัก ส่วนของโครงข่ายสามารถที่จะทิ้งเซลล์เหล่านี้หากเกิดความคับคั่งขึ้นในระบบ แต่ถ้าค่า CLP = 0 เซลล์เหล่านี้ถือว่ามีความสำคัญและโครงข่ายจะต้องพยายามส่งผ่านเซลล์เหล่านี้ให้สำเร็จ
5. The header error control (HEC) มีหน้าที่ในการตรวจจับว่ามีความผิดพลาดของบิตเฉพาะในส่วนของเฮดเดอร์เกิดขึ้นหรือไม่ ส่วนของข้อมูล (payload) มิได้มีการตรวจสอบเลย ค่าของ HEC ทั้ง 8 บิตได้จากการนำ 4 octet แรกของเฮดเดอร์ไปหารด้วยโพลิโนเมียล

$x^8 + x^2 + x + 1$ เศษที่ได้นำมา EXOR กับ 01010101 แล้วนำผลที่ได้ไปบรรจุลงใน HEC ที่ภาครับก็จะนำข้อมูลที่รับได้มาผ่านกระบวนการที่กลับกันกับที่ภาคส่ง

1.7 ประเภทของการให้บริการ

ในการออกแบบโครงข่ายที่สามารถรองรับข้อมูลหลากหลายประเภทพร้อม ๆ กันได้นั้นเป็นเรื่องที่ค่อนข้างยุ่งยากและซับซ้อนมาก เพราะการสื่อสารข้อมูลแต่ละชนิดมีความต้องการคุณภาพในการส่งที่แตกต่างกัน ยกตัวอย่างเช่น ในการรับส่งสัญญาณเสียง ถ้าหากมีการดีเลย์ของเซลล์ในระหว่างการส่งของข้อมูลมากเกินไปค่าค่าหนึ่ง เช่น 500 ms เซลล์เหล่านี้เมื่อถึงที่ภาครับจะไม่มีประโยชน์เลย แต่ถ้าเป็นการส่งข้อมูลประเภทค่าการดีเลย์ของเซลล์ไม่ถือเป็นเรื่องสำคัญนัก ขอเพียงแต่เซลล์ทุกเซลล์เดินทางถึงจุดหมายโดยไม่เกิดการสูญเสียระหว่างทาง จากความหลากหลายของระดับความต้องการของการสื่อสารที่แตกต่างกันไปในแต่ละประเภทของทราฟฟิกนี้เอง ทำให้การที่จะออกแบบระบบที่สามารถจัดการกับความต้องการที่แตกต่างกันของทราฟฟิกแต่ละประเภทพร้อม ๆ กันเป็นเรื่องที่กระทำได้ยากมากโดยเฉพาะอย่างยิ่งในสถานะปัจจุบันที่มีการให้บริการสื่อสารในรูปแบบใหม่ ๆ เกิดขึ้นมากมาย จากปัญหาดังกล่าวนี้เอง กลุ่มผู้ที่เกี่ยวข้องกับการพัฒนาและวิจัยโครงข่าย ATM จึงได้พยายามจัดแบ่งประเภทของการให้บริการออกเป็น 5 ประเภทใหญ่ โดยที่การให้บริการของแต่ละประเภทจะมีคุณภาพที่แตกต่างกันไป เพื่อการสื่อสารข้อมูลแต่ละชนิดสามารถเลือกใช้ประเภทของการให้บริการที่เหมาะสมกับความต้องการได้

ในปี ค.ศ. 1996 กลุ่มของผู้ที่มีบทบาทสำคัญต่อการพัฒนาและกำหนดทิศทางของเทคโนโลยี ATM ที่มีชื่อเรียกว่า The ATM Forum ได้ตกลงกันเพื่อออกมาตรฐานกำหนดประเภทของการให้บริการออกเป็น 5 ประเภท ตามคุณภาพของการให้บริการ (Quality of Service หรือเรียกย่อ ๆ ว่า QoS) ดังนี้คือ

- CBR Constant Bit Rate
- rt-VBR Real-Time Variable Bit Rate
- nrt-VBR Non-Real-Time Variable Bit Rate
- UBR Unspecified Bit Rate
- ABR Available Bit Rate

โดยคุณภาพของการให้บริการ (QoS) จะวัดจากค่าพารามิเตอร์ต่าง ๆ ดังนี้คือ

- Peak-to-peak Cell Delay Variation (peak-to-peak CDV)
- Maximum Cell Transfer Dealy (maxCTD)
- Cell Loss Ratio (CLR)

Constant Bit Rate (CBR) การให้บริการประเภทนี้ผู้ใช้จะตกลงกับส่วนของโครงข่ายถึงค่า PCR (Peak Cell Rate) ที่ต้องการในช่วงเริ่มต้นก่อนการรับส่งข้อมูล ในระหว่างการรับส่งข้อมูลผู้ใช้จะ

ได้รับอนุญาตให้ส่งข้อมูลในอัตราสูงสุดเท่ากับ PCR ตลอดช่วงเวลาการให้บริการ ส่วนทางด้านของโครงข่ายจะต้องจัดสรรแบนด์วิธที่เพียงพอกับความต้องการอย่างต่อเนื่อง และต้องประกันว่าเซลล์ข้อมูลที่ได้รับจะมีดีเลย์ที่ไม่เกินค่า Maximum Cell Transfer Delay (maxCTD) เสมอ การให้บริการแบบนี้จึงเหมาะสมกับการส่งข้อมูลที่มีอัตราการส่งคงที่ เช่น เสียงหรือวิดีโอที่ยังไม่ได้ผ่านขบวนการบีบอัดสัญญาณ (compression) สังเกตว่าข้อมูลที่อาศัยการรับส่งข้อมูลประเภท CBR จะได้รับการรับประกันว่าจะไม่มีการสูญหายของเซลล์ข้อมูล และมีค่าการดีเลย์เชิงเวลาในระดับที่ผู้ใช้อยอมรับได้

Real-Time Variable Bit Rate (rt-VBR) ออกแบบไว้สำหรับใช้ในการรับส่งข้อมูลที่มีอัตราการส่งไม่คงที่ และต้องการการรับส่งสัญญาณที่ส่งได้อย่างรวดเร็วทันเวลาและต่อเนื่อง (real-time) ก่อนที่จะเริ่มการให้บริการผู้ใช้จะต้องตกลงและกำหนดค่าพารามิเตอร์ 3 ค่า คือ Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) และ Maximum Burst Size (MBS) กับส่วนของโครงข่ายไว้ก่อน ในส่วนของโครงข่ายก็จะต้องเตรียมแบนด์วิธที่เพียงพอที่ให้แก่ใจว่าเซลล์ข้อมูลจะไม่เกิดการดีเลย์เกินค่า maxCTD ทั้งนี้ทางโครงข่ายอาจจะไม่ต้องจัดสรรแบนด์วิธที่มีขนาดเท่ากับ PCR ก็ได้เนื่องจากการให้บริการประเภทนี้สามารถใช้ประโยชน์จากการทำมัลติเพล็กซ์ทางสถิติ (Statistical Multiplexing) เพื่อเพิ่มประสิทธิภาพการใช้ช่องสัญญาณ

Non-Real-Time (nrt-VBR) มีไว้สำหรับรับส่งข้อมูลที่มีคุณลักษณะเป็นเบิสต์ (bursty traffic) และการดีเลย์ของเซลล์ข้อมูลไม่ส่งผลกระทบต่อคุณภาพของการทำงาน ก่อนการเริ่มการรับส่งข้อมูล ผู้ใช้จะต้องตกลงกับโครงข่ายถึงค่าพารามิเตอร์ที่สำคัญ 3 ค่า คือ PCR, SCR และ MBS เช่นเดียวกับกรณีของ rt-VBR การให้บริการประเภทนี้อาจจะได้ประโยชน์จากการทำ Statistical Multiplexing ทั้งนี้ผู้ใช้ออมให้มีเซลล์สูญเสียได้บางส่วน แต่ต้องมีค่าที่ค่อนข้างต่ำมาก

Unspecified Bit Rate (UBR) คือ การส่งข้อมูลที่ไม่มีการกำหนดอัตราการส่งข้อมูล และอาจมีการสูญเสียของเซลล์ได้ ใ้รรองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องทราบอัตราการส่งข้อมูล เช่น การสื่อสารข้อมูลคอมพิวเตอร์ เนื่องจากการที่ข้อมูลที่อยู่บนการเชื่อมต่อประเภท UBR จะไม่ได้รับการรับประกันการสูญหายของข้อมูล ดังนั้นเพื่อให้การรับส่งมีความถูกต้องและแน่นอนการควบคุมข้อผิดพลาดในการส่งจึงต้องกระทำโดยชั้น โปรโตคอลที่เหนือกว่าชั้น ATM ขึ้นไป เช่น ใช้ โปรโตคอล TCP

Available Bit Rate (ABR) เป็นการให้บริการที่เหมาะสมกับการส่งข้อมูลที่ต้องไม่มีการสูญเสียของเซลล์ข้อมูล แต่ไม่ได้มีการประกันว่าเซลล์ข้อมูลจะได้รับการส่งอย่างรวดเร็ว การส่งแบบ ABR มีไว้สำหรับรองรับแหล่งกำเนิดข้อมูลที่ไม่ทราบอัตราส่งข้อมูลที่แน่นอน เช่น การสื่อสารข้อมูลคอมพิวเตอร์ ในส่วนของการควบคุมความคับคั่งของข้อมูล จะกระทำโดยการป้อนกลับข้อมูลการติดขัดของระบบโดยผ่านเซลล์พิเศษที่เรียกว่า Resource Management (RM) มายังผู้ส่งเพื่อใช้ในการคำนวณอัตราการส่งข้อมูลที่เหมาะสม กลไกการทำงานที่กล่าวมานี้ช่วยทำให้ระบบสามารถใช้แบนด์วิธที่เหลือจากการบริการแบบอื่นได้อย่างมีประสิทธิภาพ การให้บริการแบบนี้ผู้ใช้จะกำหนดค่าพารามิเตอร์ 2 ค่า คือ Peak Cell Rate (PCR) และ Minimum Cell Rate (MCR) ในระหว่างการส่งแหล่งกำเนิด

ข้อมูลจะต้องควบคุมอัตราการส่งมิให้เกินค่า PCR และสามารถที่จะคาดหวังจากทางโครงข่ายว่าแหล่งกำเนิดข้อมูลจะสามารถส่งข้อมูลได้ด้วยอัตราอย่างน้อยเท่ากับ MCR

เราสามารถเลือกวิธีการส่งแบบใดแบบหนึ่งที่เหมาะสมในการส่งข้อมูลที่เราต้องการได้ ตัวอย่างเช่น การส่งแบบ CBR จะมีความเหมาะสมกับการสื่อสารทางเสียงเพราะปริมาณข้อมูลที่จะส่งมีอัตราที่คงที่ ขณะที่การส่งแบบ VBR มีความเหมาะสมในการส่งสัญญาณวิดีโอซึ่งมีความต้องการในการใช้แบนด์วิดท์ (bandwidth) ที่ไม่แน่นอน และการส่งแบบ ABR ก็ใช้ได้ดีในการส่งข้อมูลดาต้าซึ่งต้องการให้การส่งแน่นอนและไม่มีการสูญเสียของเซลล์เลย

โดยปกติแล้วระบบโครงข่าย ATM จะพยายามส่งข้อมูลประเภท CBR และ VBR ก่อนเพราะต้องการให้เกิดความแน่นอนว่าข้อมูลเหล่านี้จะถูกส่งได้ทันเวลาตามค่า QoS ที่ตั้งไว้ และเมื่อใดที่โครงข่ายมีแบนด์วิดท์เหลือใช้จึงจะส่งข้อมูลแบบ ABR จากรูปที่ 1.6 ซึ่งแสดงลักษณะตัวอย่างลักษณะการใช้แบนด์วิดท์ของข้อมูลประเภท CBR และ VBR จะเห็นว่าในบางช่วงเวลากการใช้แบนด์วิดท์ของ CBR และ VBR มีปริมาณน้อย ซึ่งแบนด์วิดท์ของช่องสัญญาณส่วนนี้น่าจะสามารถนำมาใช้ในการส่งข้อมูล ABR ได้ในปริมาณมาก เพื่อให้การใช้ช่องสัญญาณเกิดประโยชน์สูงสุด ในทางกลับกันถ้ามีการใช้แบนด์วิดท์ของข้อมูล CBR และ VBR มาก ปริมาณของการส่งข้อมูลของ ABR จะต้องลดลงด้วย เพื่อมิให้ส่งผลกระทบต่อ QoS ของการส่งข้อมูล CBR และ VBR



รูปที่ 1.6 ตัวอย่างปริมาณการใช้แบนด์วิดท์ของช่องสัญญาณข้อมูล CBR และ VBR

1.8 พื้นฐานการควบคุมการรับส่งข้อมูล ABR

ดังที่กล่าวไว้แล้วว่า อัตราการส่งข้อมูลแบบ ABR จากแหล่งกำเนิดข้อมูลแต่ละแห่งจะขึ้นอยู่กับสถานะและปริมาณแบนด์วิดท์ของช่องสัญญาณที่เหลืออยู่ภายในระบบโครงข่าย ดังนั้นทางโครงข่ายจะต้องมีวิธีในการบอกสถานะของแบนด์วิดท์ไปยังแหล่งกำเนิดข้อมูลเพื่อใช้ในการควบคุมอัตราการส่งให้เหมาะสม เราสามารถแบ่งวิธีการบอกสถานะของโครงข่ายออกได้เป็น 4 วิธีคือ

1. No information ไม่มีการส่งข้อมูลใด ๆ จากส่วนของโครงข่ายไปที่แหล่งกำเนิดข้อมูลเลย
2. A binary congestion indication ใช้ข้อมูลเพียง 1 บิตเพื่อบอกว่ามีความแออัดคั่งของช่องสัญญาณในการส่งผ่านข้อมูลภายในโครงข่ายหรือไม่

3. Detailed congestion indication บอกข้อมูลสถานะความแออัดบังคับของโครงข่ายโดยละเอียด
4. Explicit bandwidth (or rate) information บอกข้อมูลเกี่ยวกับปริมาณแบนด์วิดท์หรืออัตราการส่งที่แหล่งกำเนิดข้อมูลสามารถส่งได้โดยตรงเลย

สำหรับวิธีแรกแหล่งกำเนิดข้อมูลจะต้องตรวจสอบสถานะของโครงข่ายเองทั้งหมด เช่นอาจจะส่งเซลล์พิเศษออกไปและจับเวลาว่าเซลล์นั้นใช้เวลานานเท่าใดในการส่งผ่านโครงข่ายและเดินทางกลับมาอีกครั้ง ถ้าเวลาที่ใช้ในการส่งเซลล์นี้นานก็แสดงว่าช่องสัญญาณภายในโครงข่ายในขณะนั้นมีความแออัดสูง ข้อมูลนี้จะช่วยให้แหล่งกำเนิดข้อมูลสามารถปรับอัตราการส่งข้อมูลให้เหมาะสมกับสภาพของโครงข่ายได้ สังเกตว่าวิธีนี้ความรับผิดชอบในการควบคุมการใช้แบนด์วิดท์จะอยู่ที่แหล่งกำเนิดข้อมูลแทบทั้งหมด ส่วนของโครงข่ายแทบไม่ต้องทำอะไรเลย

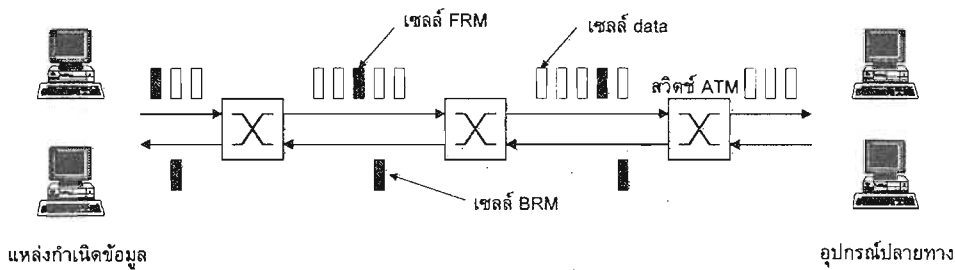
วิธีที่สองแหล่งกำเนิดข้อมูลได้รับข้อมูลเกี่ยวกับสถานะของโครงข่ายมากขึ้น โดยได้รับข้อมูลโดยตรงจากโครงข่าย แต่ข้อมูลที่รับมาก็ค่อนข้างจำกัดเพราะใช้เพียงแค่อัตราเดียว ก็จะบอกเพียงว่าโครงข่ายมีความแออัดหรือไม่ แต่จะไม่รู้ว่าความแออัดที่เกิดขึ้นอยู่ในส่วนใดของโครงข่าย วิธีที่สามแก้ปัญหาของวิธีที่สองโดยเพิ่มข้อมูลที่เกี่ยวข้องกับการแออัดมากขึ้น เช่นบอกเลยว่าส่วนใดของโครงข่ายที่เกิดการแออัด แต่นั่นหมายถึงว่าจะต้องมีการใช้เซลล์พิเศษในการส่งข้อมูลเหล่านี้ ซึ่งจะทำให้เกิดการสูญเสียของแบนด์วิดท์บางส่วนในโครงข่ายเพื่อใช้ในการส่งรายละเอียดของความแออัด

วิธีสุดท้ายเป็นวิธีที่ส่วนของโครงข่ายนั้นมีความรับผิดชอบในการควบคุมการใช้แบนด์วิดท์ของโครงข่ายมากที่สุด โดยสวิทช์ภายในโครงข่ายแต่ละตัวจะต้องทำการคำนวณเพื่อหาค่าขนาดของแบนด์วิดท์ที่เหมาะสมสำหรับให้แหล่งกำเนิดข้อมูล ABR แต่ละแห่งได้ใช้งาน โดยค่าที่คำนวณได้นี้จะถูกส่งผ่านช่องสัญญาณควบคุมไปบอกแหล่งกำเนิดสัญญาณทุก ๆ แห่ง เพื่อให้ได้รับทราบถึงขนาดแบนด์วิดท์ที่มีเหลืออยู่สำหรับแหล่งกำเนิดข้อมูล ABR แต่ละแห่ง และทำการปรับอัตราการส่งข้อมูลให้เหมาะสมต่อไป วิธีนี้เป็นวิธีที่มีประสิทธิภาพที่สุด แต่ก็มีความซับซ้อนมากที่สุดด้วย

1.9 กลไกการป้อนกลับสัญญาณควบคุม

การควบคุมการรับส่งข้อมูลแบบ ABR จะอาศัยเซลล์ชนิดพิเศษที่เรียกว่าเซลล์ RM (resource management cell) ในการป้อนกลับสัญญาณควบคุมระหว่างอุปกรณ์สวิทช์และแหล่งกำเนิดข้อมูล โดยปกติเซลล์ RM จะถูกสร้างขึ้นจากแหล่งกำเนิดข้อมูล แล้วแทรกไปกับเซลล์ข้อมูลปกติ (data cell) ทุก ๆ ครั้งที่ได้ส่งเซลล์ข้อมูลธรรมดาออกไปเป็นจำนวน $N_{rm}-1$ เซลล์ (N_{rm} ที่ใช้กันทั่วไปมีค่าเท่ากับ 32) เซลล์ RM ที่ส่งออกจากแหล่งกำเนิดข้อมูล (source) มักจะเรียกว่าเป็นเซลล์ Forward RM (FRM) เมื่อเซลล์ดังกล่าวเดินทางไปถึงที่ปลายทาง (destination) เซลล์นั้นก็จะถูกส่งกลับไปที่ต้นทางผ่านเส้นทางเดิม สำหรับเซลล์ RM ที่ส่งไปในทิศทางจากกลับนี้จะเรียกว่าเซลล์ Backward RM (BRM) สังเกตว่าเซลล์ RM จะถูกส่งผ่านสวิทช์ส่วนต่าง ๆ ของโครงข่ายที่เกี่ยวข้องทั้งหมด ทั้งในทิศทางไปและในทิศทางย้อนกลับ ดังนั้นสวิทช์แต่ละแห่งก็สามารถจะบรรจุข้อมูลที่เกี่ยวข้องกับการควบคุมความคับคั่งที่เหมาะสมไปให้แหล่งกำเนิดข้อมูลแต่ละแห่งเพื่อปรับอัตราการส่งให้เหมาะสมได้ รูปที่ 1.7 แสดงตัวอย่างการรับ

ส่งข้อมูลเพียงทิศทางเดียวบนโครงข่าย ATM พร้อมการป้อนกลับของสัญญาณควบคุม สำหรับการส่งข้อมูลในทิศทางตรงข้ามก็จะมีลักษณะเดียวกัน



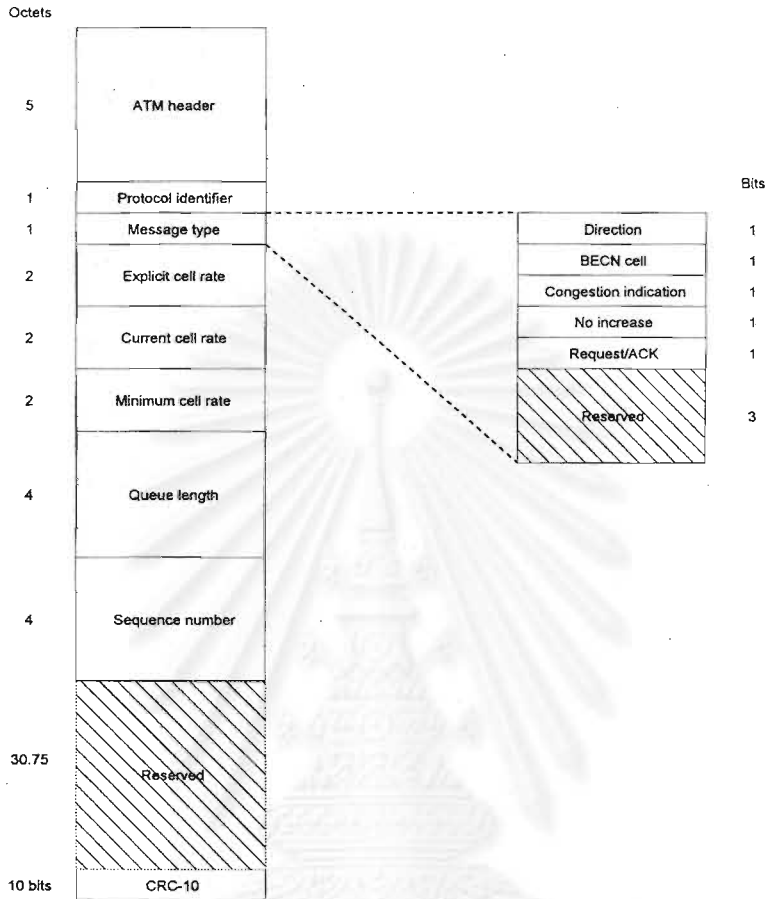
รูปที่ 1.7 แสดงการส่งผ่านเซลล์ข้อมูลและเซลล์ RM ในการเชื่อมต่อแบบ ABR

1.10 โครงสร้างของเซลล์ RM

สำหรับโครงสร้างของเซลล์ RM ที่ใช้ในการส่งสัญญาณควบคุมป้อนกลับมีลักษณะดังที่แสดงในรูปที่ 1.8 ซึ่งมีส่วนประกอบดังนี้

1. Header (5 octets) : ในส่วนเฮดเดอร์ของเซลล์ ATM กำหนดให้ค่า PT=110 เพื่อบ่งบอกว่าเป็นเซลล์ RM เซลล์เหล่านี้ใช้สำหรับควบคุมอัตราการส่งข้อมูลใน VC (virtual channel) หนึ่ง ๆ โดยที่ค่า VCI และ VPI จะมีค่าเหมือนกับเซลล์ข้อมูลธรรมดา ในทำนองเดียวกัน การควบคุมอัตราการส่งใน VP (virtual path) คือจะมี ค่า VPI เหมือนกัน และใช้ค่า VCI = 6
2. Protocol identifier (1 octet) : เป็นตัวบอกประเภทการบริการของเซลล์ RM ตัวนี้ เช่น สำหรับการบริการแบบ ABR ใช้ ID = 1
3. Message type (1 octet) : ประกอบด้วยตัวชี้บ่งขนาด 1 บิต ดังต่อไปนี้
 - 3.1 Direction (DIR) : บอกทิศทาง คือ เซลล์ FRM (DIR = 0) และ เซลล์ BRM (DIR = 1)
 - 3.2 BECN cell (BN) : เป็นตัวบอกว่าเซลล์ RM ตัวนี้สร้างมาจากที่ไหน BN = 0 เมื่อเซลล์ถูกสร้างมาจากแหล่งกำเนิดข้อมูล และ BN = 1 เมื่อเซลล์นั้นสร้างมาจาก ส่วนของสวิตช์หรืออุปกรณ์ปลายทาง
 - 3.3 Congestion indication (CI) : CI = 1 เพื่อแสดงว่าโครงข่ายมีความคับคั่งเกิดขึ้น
 - 3.4 No increase (NI) : NI = 1 เป็นการแสดงถึงการไม่อนุญาตให้แหล่งกำเนิดข้อมูลเพิ่มอัตราการส่ง
- 1.5 Request/acknowledge (RA) : ยังไม่มีการใช้งานตามข้อกำหนดของ ATM Forum
4. Explicit cell rate (2 octets) : ใช้ในการจำกัดค่าอัตราการส่ง ACR ของแหล่งกำเนิดข้อมูล
5. Current cell rate (2 octets) : เป็นค่า ACR ปัจจุบัน ซึ่งค่านี้เป็นค่าที่อุปกรณ์สวิตช์ใช้ในการหาค่า ER
6. Minimum cell rate (2 octets) : ค่านี้ถูกกำหนดสำหรับแหล่งกำเนิดข้อมูล โดยค่านี้นำมาคำนวณเพื่อแบ่งความจุในโครงข่าย
7. Queue length (4 octets) : ยังไม่มีการใช้งานตามข้อกำหนดของ ATM Forum
8. Sequence number (4 octets) : ยังไม่มีการใช้งานตามข้อกำหนดของ ATM Forum

9. CRC-10 (10 bits) : เป็นรหัสที่ใช้สำหรับตรวจจับความผิดพลาดของ RM payload (ข้อมูลทั้งหมดยกเว้นส่วนของเฮดเดอร์)



รูปที่ 1.8 โครงสร้างของเซลล์ RM

1.11 คุณลักษณะการให้บริการแบบ ABR

โดยสรุปแล้วลักษณะของการให้บริการแบบ ABR มีคุณสมบัติดังต่อไปนี้

1. การเชื่อมต่อแบบ ABR จะใช้ความจุที่เหลือจากการเชื่อมต่อแบบ CBR และ VBR ดังนั้น ABR จึงเป็นตัวเพิ่มค่า utilization ของโครงข่ายโดยไม่มีผลกระทบต่อคุณภาพของการให้บริการหรือ QoS ของการเชื่อมต่อแบบ CBR และ VBR เลย
2. การเชื่อมต่อหนึ่ง ๆ ของ ABR จะมีลักษณะเป็นพลวัต (dynamic) คืออาจมีการเปลี่ยนแปลงค่าอัตราการส่งของข้อมูลในช่วงระหว่างค่า MCR และ PCR โดยค่า MCR อาจมีค่าเท่ากับ 0 ก็ได้ แต่ถ้า MCR มีค่าไม่เท่ากับ 0 แล้ว แสดงว่าโครงข่ายต้องสามารถรับประกันว่าแหล่งกำเนิดข้อมูลนั้นสามารถส่งข้อมูลได้ด้วยอัตราการส่งอย่างต่ำเท่ากับค่า MCR แต่กระนั้นในบางช่วงเวลา แหล่งกำเนิดข้อมูลอาจจะส่งข้อมูลที่อัตราการส่งที่น้อยกว่าค่า MCR ได้
3. การที่โครงข่ายมีกระบวนการการป้อนกลับไปยังแหล่งกำเนิดข้อมูล ABR เพื่อให้ ABR มีการใช้ช่องสัญญาณในปริมาณที่เหมาะสม มีผลทำให้ในบางช่วงเวลาแหล่งกำเนิดข้อมูลมีข้อมูลใน

ปริมาณที่มากกว่าที่ช่องสัญญาณจะรับได้ ดังนั้นข้อมูลบางส่วนอาจจะต้องถูกจัดเก็บไว้ในบัฟเฟอร์เพื่อรอให้ช่องสัญญาณสื่อสารว่างลง คุณลักษณะดังกล่าวทำให้การบริการแบบ ABR นี้เหมาะกับการให้บริการที่สามารถทนต่อการเปลี่ยนแปลงอัตราการส่ง และมีค่าการหน่วงเวลาที่ไม่สามารถคาดเดาได้

4. สำหรับแหล่งกำเนิดข้อมูลแบบ ABR มีการปรับแต่งอัตราการส่งข้อมูลได้เพื่อรองรับกระบวนการป้อนกลับ และเป็นการรับประกันอัตราส่วนของการสูญหายของข้อมูลที่มีค่าน้อย ๆ ซึ่งเป็นจุดเด่นของการบริการแบบ ABR

1.12 การทำงานของแหล่งกำเนิดข้อมูล

ในการรับส่งข้อมูลของแหล่งกำเนิดข้อมูลบนการเชื่อมต่อแบบ ABR มีตัวแปรที่เกี่ยวข้องกับอัตราการส่งข้อมูลทั้งหมด 4 ตัว ที่กำหนดคุณลักษณะของการส่งข้อมูล ตัวแปรทั้งสี่ตัวสามารถอธิบายได้ดังต่อไปนี้

1. Allowed cell rate (ACR) : อัตราการส่งข้อมูลที่อนุญาตให้แหล่งกำเนิดข้อมูลส่งได้ในปัจจุบัน โดยแหล่งกำเนิดข้อมูลอาจส่งข้อมูลในอัตราเท่าใดก็ได้ระหว่าง 0 ถึง ACR
2. Minimum cell rate (MCR) : เป็นค่าน้อยที่สุดของ ACR นั่นคือโครงข่ายจะอนุญาตให้แหล่งกำเนิดข้อมูลส่งข้อมูลได้อย่างน้อยเท่ากับ MCR เสมอ อย่างไรก็ตามค่า MCR อาจถูกตั้งให้เท่ากับ 0 ก็ได้
3. Peak cell rate (PCR) : เป็นค่ามากที่สุดของ ACR
4. Initial cell rate (ICR) : เป็นที่ ACR ค่าตอนเริ่มต้น

แหล่งกำเนิดข้อมูลจะเริ่มต้นโดยการกำหนดให้ค่า $ACR = ICR$ หลังจากนั้นค่า ACR ก็จะมีการเปลี่ยนแปลงไปเรื่อย ๆ ขึ้นอยู่กับขบวนการการป้อนกลับของโครงข่าย โดยการป้อนกลับในที่นี้จะอยู่ในรูปของการส่งเซลล์ RM กลับมายังแหล่งกำเนิดข้อมูลเป็นช่วง ๆ ในแต่ละเซลล์ RM จะประกอบด้วย 3 ส่วนหลักที่ใช้ในขบวนการป้อนกลับ ดังต่อไปนี้

1. Congestion indication (CI) bit
2. No increase (NI) bit
3. Explicit cell rate (ER)

โดยแหล่งกำเนิดข้อมูลจะตอบสนองกับข้อมูลในเซลล์ RM ตามกฎต่อไปนี้

แหล่งกำเนิดข้อมูลจะทำการตรวจสอบบิต 2 บิต ที่ได้มาจากขบวนการป้อนกลับก่อน ถ้าหากมีการอนุญาตให้การเพิ่มอัตราการส่งข้อมูลได้ แหล่งกำเนิดข้อมูลก็จะเพิ่มอัตราการส่งขึ้นเป็นค่าคงที่เท่ากับ $RIF \times PCR$ โดยที่ค่า RIF (rate increase factor) เป็นค่าสัดส่วนของการเพิ่มอัตราการส่งข้อมูล แต่ถ้าระบบมีความคับคั่งเกิดขึ้นแหล่งกำเนิดข้อมูลก็ทำการการลดอัตราการส่งข้อมูลลง การลดนี้จะเป็นการลดลงแบบเอ็กโพเนนเชียล (exponential) ซึ่งมีค่าเท่ากับ $RDF \times ACR$ โดยค่า RCF (rate decrease factor) คือค่าคงที่สัดส่วนของการลดอัตราการส่งข้อมูล ในขั้นตอนท้ายสุดแหล่งกำเนิดข้อมูลก็จะตรวจสอบค่า

ACR และ ER ถ้า ER มีค่าน้อยกว่า ACR แหล่งกำเนิดข้อมูลจะลดค่า ACR ให้เท่ากับค่า ER อย่างไรก็ตามการปรับค่าของ ACR นี้จะมีค่าอยู่ระหว่างค่า PCR และ MCR เสมอ จากที่กล่าวมาทั้งหมดสามารถสรุปหลักการการทำงานได้ดังตารางต่อไปนี้

ตารางที่ 1.2 การตอบสนองของแหล่งกำเนิดข้อมูลต่อค่าที่อ่านจากเซลล์ RM

NI	CI	Action
0	0	$ACR = \max[MCR, \min[ER, PCR, ACR + RIF \times PCR]]$
0	1	$ACR = \max[MCR, \min[ER, ACR(1 - RIF)]]$
1	0	$ACR = \max[MCR, \min[ER, ACR]]$
1	1	$ACR = \max[MCR, \min[ER, ACR(1 - RIF)]]$

1.13 การผลิตและค่าเริ่มต้นของเซลล์ RM

โดยปกติจะเป็นหน้าที่ของแหล่งกำเนิดข้อมูลที่จะต้องผลิตเซลล์ RM ขึ้นมาแล้วแทรกไปกับเซลล์ข้อมูลธรรมดาเป็นช่วง ๆ ค่าเริ่มต้นของเซลล์ RM จะถูกกำหนดให้เป็นไปตามที่แสดงไว้ในตารางที่ 1.3 เช่น บิต CI จะถูกเซตให้เป็น 0 เสมอ ในบางสถานการณ์อาจจะเป็นไปได้ว่าสวิตช์หรือแม้แต่อุปกรณ์ปลายทางเองก็สามารถที่จะผลิตเซลล์ RM ขึ้นมาเองได้ เช่นเกิดความคับคั่งอย่างรุนแรงทำให้เกิดความจำเป็นที่จะต้องรีบส่งสัญญาณควบคุมไปบอกแหล่งกำเนิดข้อมูลให้ลดอัตราการส่งลงทันที

ตารางที่ 1.3 การกำหนดค่าตั้งต้นของเซลล์ RM

ส่วนของบิต	สร้างโดยแหล่งกำเนิดข้อมูล	สร้างโดยสวิตช์ หรืออุปกรณ์ปลายทาง
Direction (DIR)	0	1
BECN Cell (BN)	0	1
Congestion Indication (CI)	0	0 หรือ 1
No increase (NI)	0 หรือ 1	0 หรือ 1
Explicit Cell Rate (ECR)	$\leq PCR$	ค่าอัตราการส่งข้อมูลใดๆ
Current Cell Rate (CCR)	ACR	0
Minimum Cell Rate (MCR)	MCR	0

1.14 การทำงานของสวิตช์

เพื่อรองรับการให้บริการแบบ ABR อุปกรณ์สวิตช์ ATM จะมีหน้าที่ 2 อย่างคือ

1. Congestion control : เพราะการบริการแบบ ABR เป็นการบริการที่ต้องการการสูญเสียเซลล์ที่น้อยที่สุด ดังนั้นขบวนการควบคุมอัตราการส่งข้อมูลของ ABR จึงจำเป็นต้องมีในสวิตช์ เพื่อควบคุมให้แหล่งกำเนิดข้อมูลส่งข้อมูลในอัตราการส่งข้อมูลที่โครงข่ายสามารถรองรับได้ สำหรับการทำงานแบบนี้สวิตช์จะต้องตรวจสอบความยาวของคิว (queue) และต้องส่งสัญญาณเพื่อบอกให้ลดอัตราการส่งข้อมูลเมื่อคิวของบัพเฟอร์เริ่มเต็ม

2. Fair capacity allocation : อุปกรณ์สวิตช์ ATM แต่ละตัวควรแบ่งความจุให้ทุก ๆ การเชื่อมต่อที่ผ่านสวิตช์อย่างเท่าเทียมกัน (fair share) และเมื่อเกิดการคับคั่งขึ้น สวิตช์จะต้องลดอัตราการส่งข้อมูลของการเชื่อมต่อที่มีการใช้ความจุมากกว่าที่ควรจะเป็น (คือค่า fair share นั้นเอง)

การทำงานของสวิตช์สำหรับรองรับการควบคุมความคับคั่ง และการแบ่งความจุอย่างเท่าเทียมกันนี้ สามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ binary feedback schemes ซึ่งใช้เฉพาะบิต EFCI, CI และ NI เท่านั้น และ explicit rate feedback schemes ที่มีการใช้ฟิลด์ ER ด้วย ตัวแปรต่าง ๆ ที่เกี่ยวข้องกับการทำงานของการรับส่งประเภท ABR ได้นำมาสรุปไว้ในตารางที่ 1.4

ตารางที่ 1.4 ตัวแปรต่าง ๆ ของ ABR

Label	ชื่อ	คำอธิบาย	Default
PCR	Peak cell rate	อัตราการส่งข้อมูลสูงสุดที่กำหนดไว้ที่แหล่งกำเนิดข้อมูล	-
MCR	Minimum cell rate	อัตราการส่งข้อมูลต่ำสุดของแหล่งกำเนิดข้อมูลที่โครงข่ายรับประกัน	0
ICR	Initial cell rate	ค่าเริ่มต้นของ ACR ; อัตราการส่งข้อมูลที่แหล่งกำเนิดข้อมูลควรส่งคอนเริ่มต้น หรือหลังจากมีการหยุดส่ง	PCR
RIF	Rate increase factor	ค่าสัดส่วนการเพิ่มอัตราการส่งข้อมูล	1/16
Nrm		จำนวนเซลล์มากที่สุดที่แหล่งกำเนิดข้อมูลจะส่งระหว่างเซลล์ FRM ที่ติดกัน	32
Mrm		ควบคุมการแบ่งแบนด์วิดท์ระหว่างเซลล์ FRM, BRM และ data	2
RDF	Rate decrease factor	ค่าอัตราส่วนที่ใช้ควบคุมการลดลงของอัตราการส่งข้อมูล	1/16
ACR	Allowed cell rate	ค่าขอบเขตบนของอัตราการส่งข้อมูลของแหล่งกำเนิดข้อมูลซึ่งค่านี้จะปรับตามขบวนการป้อนกลับมีค่าในช่วง MCR และ PCR	-
CRM		จำนวนของเซลล์ FRM ที่สามารถส่งได้โดยไม่มีการรับเซลล์ BRM	$2^{19}-1$
ADTF	ACR decrease time factor	เวลาระหว่างการส่งเซลล์ RM ก่อนอัตราการส่งจะลดลงเป็น ICR	0.5 ms
Tm		ค่าขอบเขตบนของเวลาระหว่างเซลล์ FRM ที่ติดกันของ active source	100 ms
FRTT	Fixed round-trip time	ผลรวมของเวลาที่ใช้ในการเดินทาง ไปกลับระหว่างแหล่งกำเนิดข้อมูลกับปลายทาง	-
TBE	Transient buffer exposure	จำนวนเซลล์ที่แหล่งกำเนิดข้อมูลสามารถส่งได้ก่อนได้รับเซลล์ RM ตัวแรก	$2^{24}-1$
CDF	Cutoff decrease factor	ควบคุมการลดค่า ACR เมื่อทำงานร่วมกับ CRM	1/16
TCR	Tagged cell rate	ค่าขอบเขตบนของอัตราการส่ง out-of-rate FRM cell	10 cells

1.15 ขอบเขตของงานวิจัย

จากที่ได้กล่าวมาข้างต้นเราสามารถสรุปเป้าหมายของการควบคุมความคับคั่งแบบต่าง ๆ ได้ดังนี้

1. สามารถป้องกันการเกิดความคับคั่งในโครงข่ายได้อย่างมีประสิทธิภาพ
2. สามารถใช้ประโยชน์จากทรัพยากรที่เหลือจากการให้บริการแบบอื่นที่มีลำดับความสำคัญมากกว่าบริการแบบ ABR ได้มากที่สุด
3. แหล่งกำเนิดแบบ ABR ที่มีลำดับความสำคัญเท่ากันกันควรได้รับความยุติธรรมในการส่งข้อมูล คือ ควรได้รับการอนุญาตให้ส่งข้อมูลที่อัตราส่งใกล้เคียงกัน

วิธีการควบคุมความคับคั่งในปัจจุบันมีการพัฒนาขึ้นมามากมาย โดยสามารถแบ่งประเภทตามข่าวสารที่ใช้ในการป้อนกลับได้เป็น 2 แบบสำคัญด้วยกัน ได้แก่ การป้อนกลับแบบบิตเดียว และการป้อนกลับค่าแน่นอน การป้อนกลับแบบบิตเดียวมีข้อดีคือ รูปแบบง่าย แต่ก็มีข้อเสียหากว่าโครงข่ายนั้นมีการเปลี่ยนแปลงอย่างรวดเร็วจะทำให้วิธีการดังกล่าวไม่สามารถปรับตัวตามสภาวะโครงข่ายที่เปลี่ยนแปลงไปได้ และทำให้เกิดความคับคั่งในโครงข่ายในที่สุด นอกจากนี้ยังไม่มีความยุติธรรมระหว่างการเชื่อมต่ออีกด้วย ส่วนการป้อนกลับค่าแน่นอนมีประสิทธิภาพดีกว่า ด้วยความสามารถในการปรับอัตราส่งที่แหล่งกำเนิดได้อย่างรวดเร็ว และสามารถจัดสรรอัตราส่งให้การเชื่อมต่อได้อย่างยุติธรรมที่สุด วิธี ERICA และ FMMRA เป็นตัวอย่างวิธีการควบคุมความคับคั่งแบบป้อนกลับค่าแน่นอนที่มีประสิทธิภาพ และได้รับความสนใจอย่างมากในงานวิจัยโดยทั่วไป

การพัฒนาวิธีการควบคุมความคับคั่งที่ได้กล่าวมาเป็นการพัฒนาเพื่อการสื่อสารแบบจุดต่อจุด (point-to-point) เพียงอย่างเดียว หากแต่งานวิจัยในช่วงหลังเริ่มให้ความสำคัญกับการประยุกต์การควบคุมความคับคั่งให้สามารถใช้งานในการส่งแบบหนึ่งจุดไปสู่หลายจุด (point-to-multipoint) ได้อีกด้วย ความต้องการการสื่อสารแบบหนึ่งจุดไปสู่หลายจุดทำให้การควบคุมความคับคั่งในการให้บริการแบบ ABR มีความซับซ้อนมากขึ้น เนื่องจากปริมาณข่าวสารความคับคั่งที่ใช้ในป้อนกลับจากปลายทางหลายจุดนั้นอาจก่อให้เกิดปัญหาในการเลือกใช้ข่าวสารต่าง ๆ ดังกล่าว เพื่อให้การป้อนกลับข่าวสารความคับคั่งในการส่งจากหนึ่งจุดไปสู่หลายจุดมีความถูกต้อง จึงจำเป็นต้องมีการพัฒนาวิธีการรวบรวมข่าวสารความคับคั่ง (Consolidation Algorithm) ที่มีประสิทธิภาพขึ้น ที่ผ่านมามีผู้เสนอวิธีการรวบรวมข่าวสารความคับคั่งขึ้นหลายวิธีบ้างแล้ว โดยทั่วไปสมรรถนะของวิธีการรวบรวมข้อมูลป้อนกลับพิจารณาจาก

1. แบบเน้นความเร็วในการป้อนกลับ วิธีการรวบรวมข่าวสารความคับคั่งประเภทนี้จะเน้นความเร็วในการป้อนกลับ โดยจะพยายามป้อนกลับข่าวสารความคับคั่งให้เร็วที่สุดโดยไม่สนใจความถูกต้องของข่าวสารที่ป้อนกลับไป วิธีนี้อาจทำให้เกิดความผิดพลาดในการส่งข้อมูลได้

2. แบบเน้นความถูกต้องในการป้อนกลับ วิธีการแบบนี้จะรวบรวมข่าวสารความถูกต้องให้ครบโดยการรอข่าวสารให้ป้อนกลับมาจากทุก ๆ เส้นทาง ดังนั้นข่าวสารที่ได้จะถูกต้อง แต่ความเร็วในการป้อนกลับจะน้อย และอาจไม่ทันต่อสถานะของโครงข่ายที่เป็นจริง

รายงานวิจัยฉบับนี้นำเสนอการนำประยุกต์วิธีการควบคุมความคับคั่งแบบ ERICA และ FMMRA มาใช้ในบนการส่งจากหนึ่งจุดไปสู่หลายจุด โดยเปรียบเทียบวิธีการรวบรวมข่าวสารความคับคั่งจำนวน 5 แบบที่รู้จักทั่วไปกับวิธีการรวบรวมความคับคั่งแบบใหม่ที่พัฒนาขึ้นในโครงการวิจัยนี้อีก 2 วิธี โดยทำการเปรียบเทียบในแง่ความสามารถของวิธีการควบคุมความคับคั่ง และในแง่ของวิธีการรวบรวมข่าวสารความคับคั่ง ประสิทธิภาพที่สนใจศึกษาเปรียบเทียบได้แก่ ความสามารถในการปรับอัตราส่งที่แหล่งกำเนิด ขนาดหน่วยความจำ เวลาประวิง สัญญาณรบกวนจากการรวบรวมข้อมูลป้อนกลับ ความซับซ้อนในการทำงาน ความสามารถในการปรับแต่งมาใช้กับวิธีการรวบรวมข้อมูลการป้อนกลับ การเปลี่ยนแปลงในกรณีที่มี branch ใหม่เข้ามาเพิ่ม และ การเปลี่ยนแปลงในกรณีที่มี branch หนึ่งไม่ตอบสนอง

1.16 ประโยชน์ที่ได้รับ

1. ทำให้ทราบถึงวิธีการควบคุมความคับคั่งที่เหมาะสมที่สุดสำหรับการส่งประเภท ABR จากหนึ่งจุดไปสู่หลายจุดในโครงข่าย ATM
2. พัฒนานิสิตในระดับปริญญาโทให้มีความสามารถในการทำงานวิจัยในเชิงลึกได้
3. พัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถจำลองการทำงานของโครงข่าย ATM ที่มีทราฟฟิกแบบ ABR ได้
4. ได้พัฒนาวิธีการรวบรวมข่าวสารความคับคั่งบนการส่งประเภท ABR จากหนึ่งจุดไปสู่หลายจุดในโครงข่าย ATM เพื่อแก้ปัญหาต่าง ๆ ที่เกิดขึ้นทั้งหมดได้
5. เนื้อหาบางส่วนในรายงานวิจัยฉบับนี้ ได้รับการตีพิมพ์ในบทความทางวิชาการจำนวนทั้งสิ้น 5 ฉบับ ดังรายละเอียดที่แนบมากับรายงานฉบับนี้ด้วย

บทที่ 2

การควบคุมทราฟฟิกในโครงข่ายเอทีเอ็ม

2.1 การควบคุมทราฟฟิกในโครงข่าย ATM

ATM Forum Traffic Management 4.0 [1] ได้แบ่งระดับการให้บริการของโครงข่าย ATM ตามประเภทของแหล่งกำเนิดข้อมูล เพื่อให้สามารถรองรับการใช้งานแบบต่าง ๆ ได้ ระดับการให้บริการของโครงข่าย ATM จะเป็นดังนี้

1. CBR (Constant Bit Rate) ใช้รองรับแหล่งกำเนิดข้อมูลที่ต้องการอัตราการส่งข้อมูลคงที่ที่ Peak Cell Rate (PCR) โดยไม่ต้องการให้เกิดการสูญเสียข้อมูลและเกิดการหน่วงเวลาขึ้นในระหว่างการส่ง เช่น โทรศัพท์ และ สัญญาณภาพวิดีโอในลักษณะที่ยังไม่ได้รับการบีบอัด

2. rt-VBR (Real-Time Variable Bit Rate) ใช้รองรับแหล่งกำเนิดข้อมูลที่มีอัตราการส่งข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ และต้องการใช้งานในการส่งแบบ Real-time เช่น สัญญาณภาพวิดีโอที่ได้รับการบีบอัดแล้ว เซลล์ข้อมูลที่ส่งแบบนี้จะได้รับประกันโดยโครงข่ายว่าจะไม่มีการสูญหายของเซลล์ข้อมูล และ มีการหน่วงเวลาอยู่ในระดับต่ำตามที่ต้องการ การต่อเชื่อมประเภทนี้จะควบคุมการส่งโดยมีตัวแปรได้แก่ PCR, Sustainable Cell Rate (SCR) และ Maximum Burst Size (MBS)

3. nrt-VBR (Non-Real-Time Variable Bit Rate) ใช้รองรับแหล่งกำเนิดข้อมูลที่มีอัตราการส่งข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ และไม่ต้องมีการใช้งานแบบเวลาจริง (Non-Real-time) เช่น การสื่อสารข้อมูลแบบ Bursty traffic ข้อมูลที่อยู่บนการเชื่อมต่อประเภท nrt-VBR จะได้รับการรับประกันการสูญหายของข้อมูล แต่จะไม่ได้รับประกันเวลาประวิง การต่อเชื่อมประเภทนี้จะควบคุมการส่งโดยมีตัวแปรได้แก่ PCR, SCR และ MBS

4. UBR (Unspecified Bit Rate) ใช้รองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องส่งข้อมูลที่อัตราส่งที่แน่นอน และการส่งผ่านข้อมูลก็อาจเกิดการสูญเสียของเซลล์ข้อมูลได้ เช่น การสื่อสารข้อมูลคอมพิวเตอร์ที่ไม่ต้องการคุณภาพการส่งมากนัก

5. ABR (Available Bit Rate) ใช้รองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องส่งข้อมูลที่มีอัตราส่งที่แน่นอน และ ไม่ต้องการให้เกิดการสูญเสียของเซลล์ข้อมูล แต่ยอมให้มีการหน่วงเวลาในการส่งได้ เช่น การสื่อสารข้อมูลคอมพิวเตอร์ การควบคุมความคับคั่งของทราฟฟิกจะทำได้โดยอาศัยการป้อนกลับของข้อมูลที่แสดงถึงสถานะของโครงข่ายมายังผู้ส่ง เพื่อให้แหล่งกำเนิดสามารถคำนวณอัตราการส่งข้อมูลที่เหมาะสมที่โครงข่ายสามารถรับได้จริง ทำให้สามารถใช้แบนด์วิดท์ของโครงข่ายที่เหลือจากการบริการแบบอื่นได้อย่างมีประสิทธิภาพ

เมื่อเปรียบเทียบคุณสมบัติต่าง ๆ ของระดับการให้บริการต่าง ๆ จะเป็นดังตารางที่ 2.1

ตารางที่ 2.1 คุณสมบัติของการให้บริการแบบต่าง ๆ ในโครงข่าย ATM

คุณสมบัติ	ประเภทการให้บริการ				
	CBR	rt-VBR	nrt-VBR	UBR	ABR
Traffic Parameter:					
PCR and CDVT(4,5)	Specified			Specified ₂	Specified ₃
SCR, MBS, CDVT (4,5)	N/A	Specified		N/A	
MCR ₄	N/A			N/A	Specified
QoS Parameters :					
Peak-to-peak CDV	Specified		Unspecified		
MaxCTD	Specified		Unspecified		
CLR ₄	Specified			Unspecified	See Note 1
คุณสมบัติอื่น ๆ :					
Feedback	Unspecified				Specified ₆

Note :

1. CLR มีค่าต่ำเมื่อมีการปรับอัตราการส่งเซลล์ตามข้อมูลป้อนกลับจากโครงข่าย
2. ไม่ใช่ CAC และ UPC ความคม
3. แสดงค่าสูงสุดที่ส่งได้ ค่าที่ใช้ในการส่งจริงจะพิจารณาจากข้อมูลป้อนกลับจากโครงข่าย
4. เป็นตัวแปรที่บ่งบอกแน่นอนหรือบ่งบอกโดยนัยสำหรับ PVCs (Permanent VCs) หรือ SVCs (Switched VCs)
5. CDVT คือ Cell Delay Variation Tolerance ไม่เป็นค่าเฉพาะสำหรับการเชื่อมต่อใด ๆ
6. ดูรายละเอียดใน ABR Flow Control

2.2 การควบคุมการส่งข้อมูลบนการให้บริการแบบ ABR ในโครงข่าย ATM

การให้บริการแบบ ABR นั้นถูกออกแบบมาให้ใช้งานในการส่งข้อมูลคอมพิวเตอร์ โดยพยายามให้การสูญเสียข้อมูลน้อยที่สุด ในขณะที่สามารถใช้งานทรัพยากรที่เหลือจากการให้บริการแบบอื่นของโครงข่ายได้อย่างมีประสิทธิภาพมากที่สุด ด้วยเหตุนี้จึงจำเป็นต้องมีการป้อนกลับสถานะความคับคั่งมาจากโครงข่าย โดย ATM Forum Traffic Management 4.0 ได้กำหนดคุณลักษณะของการส่งข้อมูลบนการให้บริการแบบ ABR ไว้เป็นแบบการควบคุมอัตราการส่ง (Rate based Flow Control) [2] โดยปรับอัตราส่งตามข่าวสารความคับคั่งที่ป้อนกลับจากสวิทช์ในโครงข่าย สวิทช์จะตรวจสอบปริมาณทราฟฟิค และ คำนวณแบนด์วิดท์ที่เหลืออยู่จากการให้บริการแบบอื่น พร้อมทั้งคำนวณอัตราส่งที่เท่าเทียมกันสำหรับทุก ๆ การเชื่อมต่อ และ ตัดสินสถานะความคับคั่งของโครงข่าย จากนั้นจะป้อนกลับข้อ

มุลดังกล่าวกลับไปสู่แหล่งกำเนิดโดยใช้เซลล์ RM (Resource Management Cell – RM cell) ข่าวสารความคับคั่งที่อยู่ในเซลล์ RM ได้แก่

1. CI (congestion indication) เป็นค่าที่แสดงถึงสถานะความคับคั่งของโครงข่าย หากมีค่าเป็น 1 แสดงว่าเกิดความคับคั่งขึ้นในโครงข่าย และ แหล่งกำเนิดควรปรับอัตราส่งให้ลดลง เพื่อลดความคับคั่งในโครงข่าย และหากมีค่าเป็น 0 แสดงว่าไม่เกิดความคับคั่ง ทำให้แหล่งกำเนิดสามารถเพิ่มอัตราส่งได้อีก

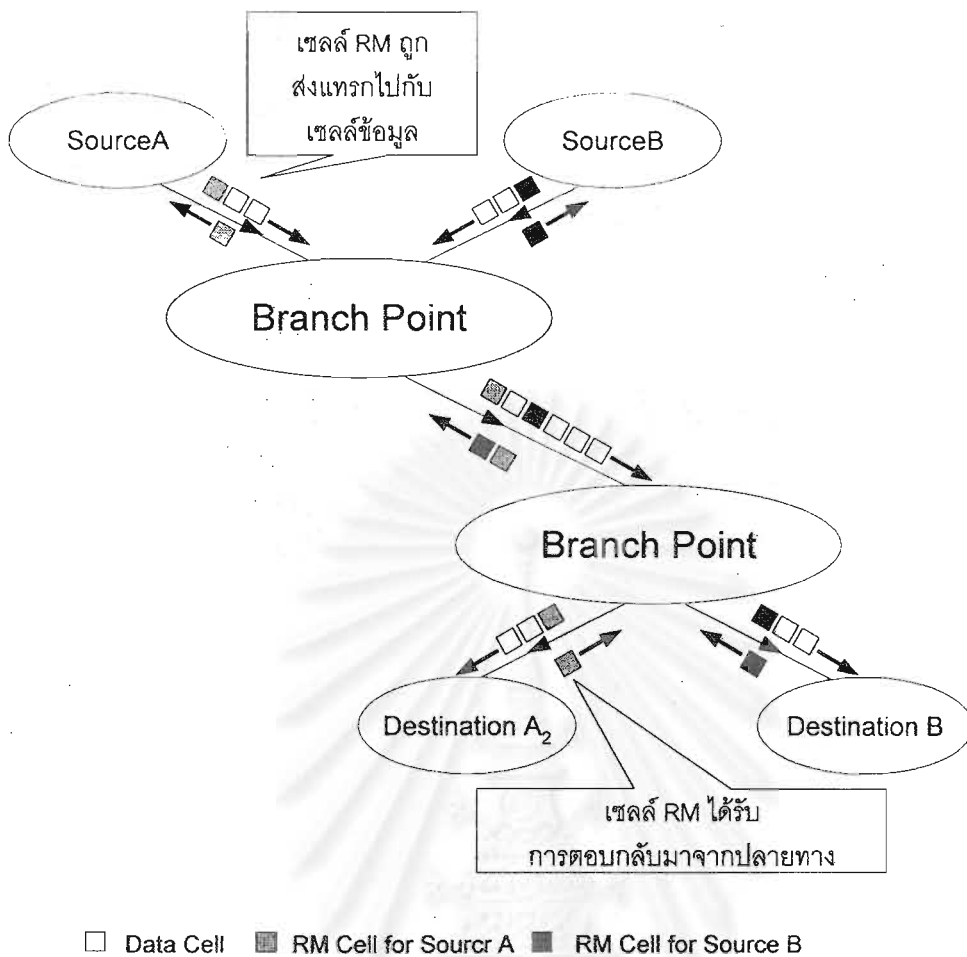
2. NI (No increase) ในบางกรณีนั้นโครงข่ายยังไม่เกิดความคับคั่งขึ้น แต่มีแนวโน้มที่จะเกิดความคับคั่งขึ้นหากมีการเพิ่มอัตราส่งขึ้นอีก ดังนั้น NI จึงมีค่าเป็น 1 เพื่อเป็นการห้ามมิให้แหล่งกำเนิดเพิ่มอัตราส่งขึ้นอีก ซึ่งอาจจะทำให้โครงข่ายเกิดความคับคั่งขึ้นอีก

3. ER (explicit rate) คือ อัตราส่งที่โครงข่ายสามารถรองรับได้โดยไม่ก่อให้เกิดความคับคั่งขึ้นในโครงข่าย ในขณะปัจจุบัน

วิธีการที่สวิตช์ใช้ในการคำนวณข่าวสารความคับคั่งดังกล่าว เรียกว่า วิธีควบคุมความคับคั่ง ซึ่งมีจุดมุ่งหมายดังต่อไปนี้

1. เพื่อหลีกเลี่ยงความคับคั่งของโครงข่าย อันจะเป็นการนำไปสู่การสูญเสียเซลล์ข้อมูล ซึ่งเป็นสิ่งที่ยอมรับไม่ได้สำหรับการให้บริการแบบ ABR
2. เพื่อให้สามารถใช้งานแบนด์วิดท์ที่เหลืออยู่จากการให้บริการแบบอื่น ได้อย่างมีประสิทธิภาพมากที่สุด
3. เพื่อให้เกิดการเท่าเทียมกันในการส่งข้อมูลระหว่างการเชื่อมต่อ

แหล่งกำเนิดทำหน้าที่สร้างเซลล์ RM และ ส่งเซลล์ RM แทรกไปพร้อมกับเซลล์ข้อมูล เพื่อใช้ในการรวบรวมข่าวสารความคับคั่งจากสวิตช์ในโครงข่าย โดยที่มีระยะห่างระหว่างการส่งเซลล์ RM 1 เซลล์ต่อการส่งเซลล์ข้อมูลเป็นค่า NRM เซลล์ RM จะถูกส่งไปตามเส้นทางพร้อมกับเซลล์ข้อมูล เรียกว่า เซลล์ FRM (Forward RM cell) และ เมื่อส่งไปถึงปลายทาง ๆ จะส่งกลับเซลล์ RM นั้นไปตามเส้นทางเดิม เรียกว่า เซลล์ BRM ในระหว่างการเดินทางของเซลล์ RM นั้น สวิตช์มีหน้าที่ใช้วิธีควบคุมความคับคั่งเพื่อปรับข้อมูลในเซลล์ RM ให้เหมาะสมกับสถานะของโครงข่าย เมื่อแหล่งกำเนิดได้รับสถานะความคับคั่งที่ได้รับการป้อนกลับจากโครงข่ายก็จะปรับอัตราการส่งข้อมูลให้เหมาะสมกับข่าวสารที่ได้รับ แบบจำลองการต่อเชื่อมประเภท ABR จะเป็นดังรูปที่ 2.1



รูปที่ 2.1 แบบจำลอง โครงข่าย ATM บนการเชื่อมต่อประเภท ABR อย่างง่าย

2.3 คุณลักษณะของแหล่งกำเนิดข้อมูล

หลังจากแหล่งกำเนิดนั้นสร้างการเชื่อมต่อระหว่างต้นทางและปลายทางสำเร็จแล้ว จะคำนวณค่า ICR (Initial Cell Rate) จาก round-tip time จากต้นทางไปสู่ปลายทาง โดยค่า ICR ที่คำนวณได้นั้น หากมีค่ามากกว่า PCR จะปรับให้เป็นค่า PCR และ หากมีค่า น้อยกว่า MCR จะถูกปรับให้เป็น MCR โดย แหล่งกำเนิดจะเริ่มส่งข้อมูลที่อัตราส่ง ICR และตั้งค่า ACR (Allowed Cell Rate) ซึ่งเป็นค่าอัตราส่งที่ได้รับอนุญาตให้ส่ง ให้เป็นค่า ICR โดยเซลล์ที่ส่งเซลล์แรกได้แก่ เซลล์ RM เพื่อเป็นการรวบรวมข่าวสารการป้อนกลับจาก โครงข่าย

เมื่อแหล่งกำเนิดข้อมูลได้รับเซลล์ RM ก็จะทำกรปรับอัตราส่งให้เหมาะสมกับข่าวสารการป้อนกลับที่ได้รับ แหล่งกำเนิดจะปรับอัตราส่งโดยการปรับค่า ACR โดยมีวิธีการปรับเป็นลำดับดังนี้

1. หากค่าบิต CI ในเซลล์ RM ที่รับได้มีค่าเป็น 1 แสดงว่าเกิดความคับคั่งขึ้นและจะปรับค่า ACR ลดลงเป็น

$$ACR = \text{maximum} (ACR \times RDF, MCR) \quad (2.1)$$

โดย RDF (Rate Decease Factor) คือ อัตราคูณลดลง
 MCR (Minimum Cell Rate) คือ อัตราส่งต่ำที่สุดที่ยอมรับได้

2. หากค่าบิต NI ในเซลล์ RM ที่รับได้มีค่าเป็น 1 แสดงว่าโครงข่ายมีแนวโน้มที่จะเกิดความคับคั่งขึ้น ดังนั้นแหล่งกำเนิดจะไม่มี การปรับค่า ACR

3. หาก บิต CI และ NI มีค่าเป็น 0 แสดงว่าโครงข่ายไม่มีความคับคั่ง และสามารถรองรับปริมาณทราฟฟิกที่เพิ่มขึ้นได้ แหล่งกำเนิดจะเพิ่มค่า ACR เป็น

$$ACR = \text{minimum} (ACR + PCR * RIF, ER, PCR) \quad (2.2)$$

โดย RIF คืออัตราคูณเพิ่มขึ้น
 PCR คือค่าอัตราส่งสูงสุดที่เป็นไปได้

การใช้ค่า ER ในการจำกัดอัตราส่งร่วมกับ PCR ด้วยนั้น เนื่องจากค่า ER เป็นค่าอัตราส่งที่โครงข่ายแนะนำไว้ ดังนั้นการเพิ่มอัตราส่งจึงไม่ควรเพิ่มไปมากกว่าค่า ER

จะเห็นว่าการปรับค่า RIF และ RDF นั้นมีความสำคัญมาก เนื่องจากมีผลต่ออัตราส่งที่เพิ่มขึ้นหรือลดลงโดยตรง อย่างไรก็ตามหากวิธีควบคุมความคับคั่งสามารถคำนวณ ER ได้อย่างมีประสิทธิภาพ ค่า RIF สามารถปรับให้มีค่าเป็น 1 เพื่อให้ค่า ACR สามารถเพิ่มเป็นค่าอัตราส่งที่โครงข่ายรองรับได้อย่างรวดเร็ว

นอกจากนี้ แหล่งกำเนิดยังต้องทำหน้าที่ควบคุมอัตราส่งให้ได้เท่ากับ ACR และควบคุมการส่งเซลล์ RM ทุก ๆ การส่งเซลล์ข้อมูล NRM-1 ตัว และสามารถส่ง out-of-rate RM cell ในกรณีที่แหล่งกำเนิดมีอัตราส่งน้อยมากซึ่งจะทำให้จำนวนเซลล์ RM น้อยเกินไปสำหรับรวบรวมสถานะของโครงข่ายในปัจจุบัน ทำให้แหล่งกำเนิดต้องส่งเซลล์ RM เพิ่มนั่นคือ out-of-rate RM cell ตารางที่ 2.2 แสดงตัวแปรสำหรับแหล่งกำเนิด และ ตารางที่ 2.3 แสดงค่าต่าง ๆ ในเซลล์ RM

ตารางที่ 2.2 ABR parameter descriptions

ตัวแปร	คำอธิบาย	หน่วย และ ขอบเขตของตัวแปร	ค่าเริ่มต้น
PCR	ค่าอัตราส่งสูงสุดที่แหล่งกำเนิดสามารถส่งได้	หน่วย Cell/sec ขอบเขต 0 ถึง 16777215	กำหนดโดยแหล่งกำเนิด และ ได้รับอนุญาตระหว่างการสร้างการเชื่อมต่อ
MCR	อัตราส่งต่ำสุดที่แหล่งกำเนิดต้องการส่ง	หน่วย Cell/sec ขอบเขต 0 ถึง 16777215	0
ICR	ค่าอัตราส่งเริ่มต้นของแหล่งกำเนิดหลังจากการสร้างการเชื่อมต่อเสร็จสิ้น	หน่วย Cell/sec ขอบเขต 0 ถึง 16777215	PCR
RIF	อัตราการเพิ่มของอัตราส่งที่แหล่งกำเนิดเมื่อไม่เกิดความคับคั่งในโครงข่าย	ไม่มีหน่วย ขอบเขต 1/32768 ถึง 1	1

Nrm	จำนวนสูงสุดของเซลล์ข้อมูลต่อการส่งเซลล์ RM 1 ครั้ง	หน่วย เซลล์ ขอบเขต 2 ถึง 256	32
Mrm	ค่าที่ใช้ควบคุมอัตราส่งของแบนด์วิดท์ระหว่างเซลล์ข้อมูล เซลล์ FRM และ เซลล์ BRM	ไม่มีหน่วย มีค่าคงที่ที่ 2	2
RDF	อัตราการลดลงของอัตราส่งที่แหล่งกำเนิดเมื่อเกิดความคับคั่งในโครงข่าย	ไม่มีหน่วย ขอบเขต 1/32768 ถึง 1	1/32768
ACR	อัตราส่งที่แหล่งกำเนิดได้รับอนุญาตให้ส่ง	หน่วย Cell/secs ขอบเขต MCR ถึง PCR	ICR
CRM	ค่าสูงสุดของการส่ง FRM เมื่อไม่ได้รับเซลล์ BRM	ไม่แน่นอน	-
ADTF	ค่า ACR จะถูกปรับให้เป็น ICR หากเวลาระหว่างการส่งเซลล์ RM มากกว่า ADTF	หน่วย second ขอบเขต 0.01 ถึง 10.23 วินาที	0.5
Tmm	เวลาที่มากที่สุดระหว่างการส่งเซลล์ RM	หน่วย milliseconds ขอบเขต $100 \cdot 2^{-7}$ ถึง $100 \cdot 2^0$	100
FRTT	ค่าเวลาประวิงการแพร่กระจายระหว่างแหล่งกำเนิดไปสู่ปลายทาง รวมถึงเส้นทางกลับ	หน่วย milliseconds ขอบเขต 0 ถึง 16.7 second	ได้จากระหว่างการสร้างการเชื่อมต่อ
TBE	จำนวนเซลล์ที่ส่งได้ระหว่างเริ่มต้นการส่ง ก่อนที่จะได้รับเซลล์ BRM ตัวแรก	หน่วย เซลล์ ขอบเขต 0 ถึง 16,777,215	16,777,215
CDF	อัตราการลดลงของอัตราส่งที่แหล่งกำเนิดเมื่อจำนวนเซลล์ FRM ที่ส่งไป โดยไม่ได้รับเซลล์ BRM มากกว่า CRM	ไม่มีหน่วย ขอบเขต 1/64 ถึง 1	1/16
TCR	ขอบเขตสูงสุดของค่าอัตราส่งที่แหล่งกำเนิดที่จะส่ง out-of-rate RM cell	มีค่าคงที่ที่ 10 Cells/seconds	10 Cells/seconds

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 2.3 Fields and their position in RM-cells

FIELD	OCTET	BIT(s)	DESCRIPTION	Initial Value	
				If source-generated	If switch-generated or destination-generated
Header	1-5	All	ATM Header	RM-VPC : VCI=6 and PTI=110 RM-VCC : PTI =110	
ID	6	All	Protocol Identifier	1	
DIR	7	8	Direction	0	1
BN	7	7	BECN Cell	0	1
CI	7	6	Congestion Indication	0	Either CI=1 or NI=1 or both
NI	7	5	No Increase	0 or 1	
RA	7	4	Request/Acknowledge	0 or set in accordance with I.371-draft	
Reserved	7	3-1	Reserved	0	
ER	8-9	All	Explicit Cell Rate	A rate not greater than PCR parameter	Any rate value
CCR	10-11	All	Current Cell Rate	ACR parameter	0
MCR	12-13	All	Minimum Cell Rate	MCR parameter	0
QL	14-17	All	Queue Length	0 or set in accordance with I.371-draft	
SN	18-21	All	Sequence Number	0 or set in accordance with I.371-draft	
Reserved	22-51	All	Reserved	6A (hex) for each octet	
Reserved	52	8-3	Reserved	0	
CRC-10	52	2-1	CRC-10	See ITU-T Recommendation I.610	
	53	All			

2.4 วิธีการควบคุมความคับคั่ง (Congestion Control Scheme)

ในปัจจุบันการควบคุมความคับคั่งแบบควบคุมอัตราส่งถูกพัฒนาขึ้นมามากมาย โดยคำนึงถึงประสิทธิภาพในการจัดการทรัพยากรของโครงข่าย, อัตราการสูญเสียเซลล์ข้อมูล, ความซับซ้อนในการใช้งานจริง และ ความสามารถในการทำให้เกิดความยุติธรรมระหว่างการเชื่อมต่อที่ผ่านสวิตช์นั้น วิธีการควบคุมความคับคั่งแต่ละอย่างจะมีคุณสมบัติ และวิธีที่การป้อนกลับข่าวสารต่างกัน หากแบ่งวิธีการควบคุมความคับคั่งตาม ข่าวสารที่ใช้ป้อนกลับ สามารถแบ่งได้เป็น 2 ประเภทใหญ่ ๆ คือ

2.4.1. วิธีการป้อนกลับบิตเดียว (Binary Feedback Schemes)

ข้อมูลการป้อนกลับที่ส่งไปบอกผู้ส่งจะเป็นลักษณะบิตเดียว เช่น วิธี EFCI (Explicit Forward Congestion Indicate) [3] สวิตช์จะตรวจสอบความคับคั่งของตัวเองจากขนาดหน่วยความจำปัจจุบัน หากขนาดหน่วยความจำมีค่ามากกว่าค่าที่กำหนดไว้จะถือว่าเกิดความคับคั่งขึ้น และ สวิตช์จะตั้งค่าบิต EFCI ในเซลล์ขาไป เพื่อบอกให้ผู้รับทราบว่าเกิดความคับคั่งขึ้นแล้ว เมื่อผู้รับได้รับเซลล์ข้อมูลที่แจ้งว่าเกิดความคับคั่งจะทำการตั้งค่าบิต CI (Congestion indicate) ของเซลล์ RM ขากลับเพื่อบอกให้แหล่งกำเนิดลดอัตราการส่งลงมา ถ้าแหล่งได้รับเซลล์ RM ขากลับที่ไม่มีการตั้งค่า CI ผู้ส่งจะทำการปรับอัตราการส่งเพิ่มขึ้นได้ และบางกรณีสามารถตั้งค่าบิต NI ได้ โดยใช้ค่าที่กำหนดไว้อีกค่าหนึ่ง เรียกว่าเป็นค่ากำหนดระดับต่ำ นอกจากนี้วิธี Relative Rate Marking ซึ่งเป็นวิธีการป้อนกลับแบบบิตเดียวอีกแบบหนึ่ง จะทำการตั้งค่าทั้งบิต CI และ NI แต่จะตั้งค่าให้กับเซลล์ BRM เท่านั้น

ข้อดีของวิธีการป้อนกลับแบบนี้คือเป็นวิธีที่ง่าย รวดเร็ว มีความซับซ้อนน้อย เนื่องจากมีการตั้งค่าเพียงบิตเดียว และ ไม่จำเป็นต้องแก้ไขข้อมูลขากลับอีกด้วย

ข้อเสียของวิธีการนี้คือหากโครงข่ายเกิดความคับคั่งขึ้น ตัวส่งไม่สามารถปรับอัตราการส่งได้ทันต่อการเปลี่ยนแปลงของโครงข่ายเนื่องจากต้องเสียเวลาในการป้อนกลับมาก ซึ่งอาจทำให้เกิดการสูญหายของข้อมูลได้ และมีค่าการครอบครองสายต่ำในกรณีที่ไมเกิดความคับคั่งอีกด้วย

นอกจากนี้ยังเกิดปัญหาเมื่อเซลล์ RM ที่มีการตั้งค่า CI ไร้สูญหายไปนโครงข่ายระหว่างเส้นทางขากลับ ทำให้ผู้ส่งไม่ได้รับเซลล์ RM ที่แสดงถึงการลดอัตราการส่ง และ โครงข่ายที่ติดขัดอยู่แล้วจะยังมีปัญหามากขึ้นไปอีก

ปัญหาเรื่องความยุติธรรม (fairness) ระหว่างการต่อเชื่อมที่มีขนาดสั้นและขนาดยาว การต่อเชื่อมที่ผ่านสวิตช์หลายตัวจะมีโอกาสการถูกลดอัตราการส่งมากกว่าการต่อเชื่อมที่ผ่านสวิตช์น้อย เนื่องจากข่าวสารการป้อนกลับจะถูกส่งผ่านหลายสวิตช์ทำให้โอกาสในการถูกตั้งค่า CI มากกว่าการต่อเชื่อมที่ผ่านสวิตช์น้อยกว่า ทำให้อัตราการส่งของแต่ละการต่อเชื่อมไม่เท่ากัน เรียกปัญหานี้ว่า beat-down วิธีแก้ปัญหาดังกล่าว ของการป้อนกลับบิตเดียวคือใช้การป้อนกลับแน่นอน (Explicit Rate Feedback Schemes)

2.4.2. วิธีการป้อนกลับค่าอัตราส่งแน่นอน (Explicit Rate Feedback Schemes)

วิธีการนี้ สวิตช์จะต้องทำหน้าที่ต่อไปนี้

1. กำหนดแบ่งแบนด์วิดท์ ให้แต่ละ VC ด้วยความเท่าเทียมกัน (fairshare calculation) โดยการปรับแต่งค่า ER ในเซลล์ RM ขากลับ

2. วัลดระดับการใช้งานของแต่ละ VC ซึ่งสามารถวัดได้โดยการวัดขนาดของคิว , อัตราการเพิ่มของขนาดหน่วยความที่สวิตช์ หรือ วัดปริมาณทราฟฟิกจริงที่ผ่านสวิตช์นั้น

3. กำหนดค่าอัตราส่งที่แนะนำ (explicit rate) เพื่อส่งให้แหล่งกำเนิด ทำการปรับอัตราส่งให้เหมาะสมต่อไป

การคำนวณ ความเท่าเทียมกันในการส่ง (fairness) นั้น มีหลักการที่ใช้งานอยู่ 2 แบบ ได้แก่

1. Proportional allocation มีหลักการโดยมีการแบ่งค่าสัดส่วนของแบนด์วิดท์ จำนวนหนึ่งที่มีค่าแน่นอนให้กับแต่ละการเชื่อมต่อ โดยสัดส่วนดังกล่าวได้มีการจัดสรรให้มีความเท่าเทียมกันไว้ก่อนแล้ว

2. Max-Min fairness มีหลักการโดยพยายามให้ทุกการเชื่อมต่อได้รับแบนด์วิดท์จำนวนเท่ากันให้มากที่สุด และหากการเชื่อมต่อใด ไม่ต้องการใช้งานแบนด์วิดท์ที่จัดสรรให้ได้เต็มที่ ก็จะไปนำแบนด์วิดท์ส่วนที่เหลือนั้นมาแบ่งเพิ่มให้กับการเชื่อมต่ออื่น ตัวอย่าง เช่น มีสายสัญญาณขนาด 100 Mbps โดยมีการเชื่อมต่อ 3 เส้นทาง แต่ละเส้นทางมีความต้องการส่งที่ 10 , 50 และ 60 Mbps ตามลำดับ ดังนั้นเมื่อคำนึงถึง Max-Min Fairness ค่าแบนด์วิดท์ที่คำนวณได้สำหรับแต่ละการเชื่อมต่อจะเป็นดังตารางที่ 2.4 โดยสังเกตว่าการเชื่อมต่อที่ 1 นั้นใช้งานแบนด์วิดท์ได้เพียง 10 ดังนั้นจึงถูกจัดสรรให้ได้รับเพียง 10 Mbps เท่านั้น ส่วนที่เหลือคือ 90 Mbps นั้นถูกแบ่งให้กับการเชื่อมต่ออีก 2 อัน

ตารางที่ 2.4 ตัวอย่างของการใช้ Max-Min Fairness กำหนดหาแบนด์วิดท์ที่เท่าเทียมกัน

VC	ความต้องการแบนด์วิดท์ (Mbps)	จัดสรรให้ (Mbps)
1	10	10
2	50	45
3	60	45

การควบคุมความคับคั่งแบบมีผู้พัฒนาหลากหลายวิธี แต่วิธีที่ได้รับความนิยมได้แก่ ERICA และ FMMRA ซึ่งมีลักษณะการทำงานดังต่อไปนี้

2.5 วิธี ERICA (Explicit Rate Indicate for Congestion Avoidance)

ERICA [4] เป็นวิธีที่ปรับค่าอัตราส่งแน่นอนโดยคำนึงถึง Max-Min Fairness โดยมีจุดประสงค์เพื่อจัดสรรแบนด์วิดท์ของโครงข่ายให้มีประสิทธิภาพและมีความเท่าเทียมกัน โดยอาศัยการตรวจสอบแบนด์วิดท์ที่ใช้งานได้ และ ความต้องการการส่งของแหล่งกำเนิด ERICA เป็นวิธีการที่ทำงานต่อหนึ่งเส้นทางขาออกหนึ่ง ๆ ของสวิตช์ สวิตช์จะคำนวณค่าตัวแปรต่าง ๆ ทุก ๆ ช่วงเวลาหนึ่งซึ่งค่าเหล่านี้จะแสดงสถานะของโครงข่าย ปริมาณทราฟฟิกขาเข้า และ จำนวนการเชื่อมต่อที่ผ่านสวิตช์นั้น จากนั้นนำ

มาคำนวณค่า Z ตามสมการ (2.3) และ ค่า Fairshare คือค่าอัตราการส่งข้อมูลที่ยุติธรรมระหว่างทุก ๆ การส่งแต่ละเส้นทาง ซึ่งคำนวณจากค่าแบนด์วิดท์ที่ใช้งานได้หารด้วยจำนวนการต่อเชื่อมที่ส่งผ่านเส้นทางขาออกหนึ่งในสวิตช์นั้น นอกจากนี้สวิตช์ยังทำการคำนวณแบนด์วิดท์ที่เหลือจากการให้บริการแบบอื่น และสามารถนำมาแบ่งสรรให้แก่การเชื่อมต่อประเภท ABR ได้ ค่า Z นั้นแสดงถึงระดับความคับคั่งของเส้นทางขาออกนั้น และเป็นตัวแปรที่สำคัญในคำนวณค่า ER เพื่อใช้ในการป้อนกลับต่อไป

$$Z = \frac{\text{แบนด์วิดท์ขาเข้า}}{\text{แบนด์วิดท์เป้าหมาย}} \quad (2.3)$$

โดย แบนด์วิดท์ขาเข้า คือ แบนด์วิดท์รวมที่ส่งเข้ามาสู่สวิตช์สำหรับทุก ๆ การเชื่อมต่อ สามารถหาได้จากการวัดปริมาณทราฟฟิกที่เข้ามาช่วงเวลาหนึ่ง

แบนด์วิดท์เป้าหมาย คือ ค่าอัตราการส่งข้อมูลขาออกที่ต้องการ มีค่าเท่ากับแบนด์วิดท์ที่สามารถใช้งานได้ คูณ ด้วย ค่าการใช้งานสายผ่านเป้าหมาย (การใช้งานสายผ่านที่ต้องการ)

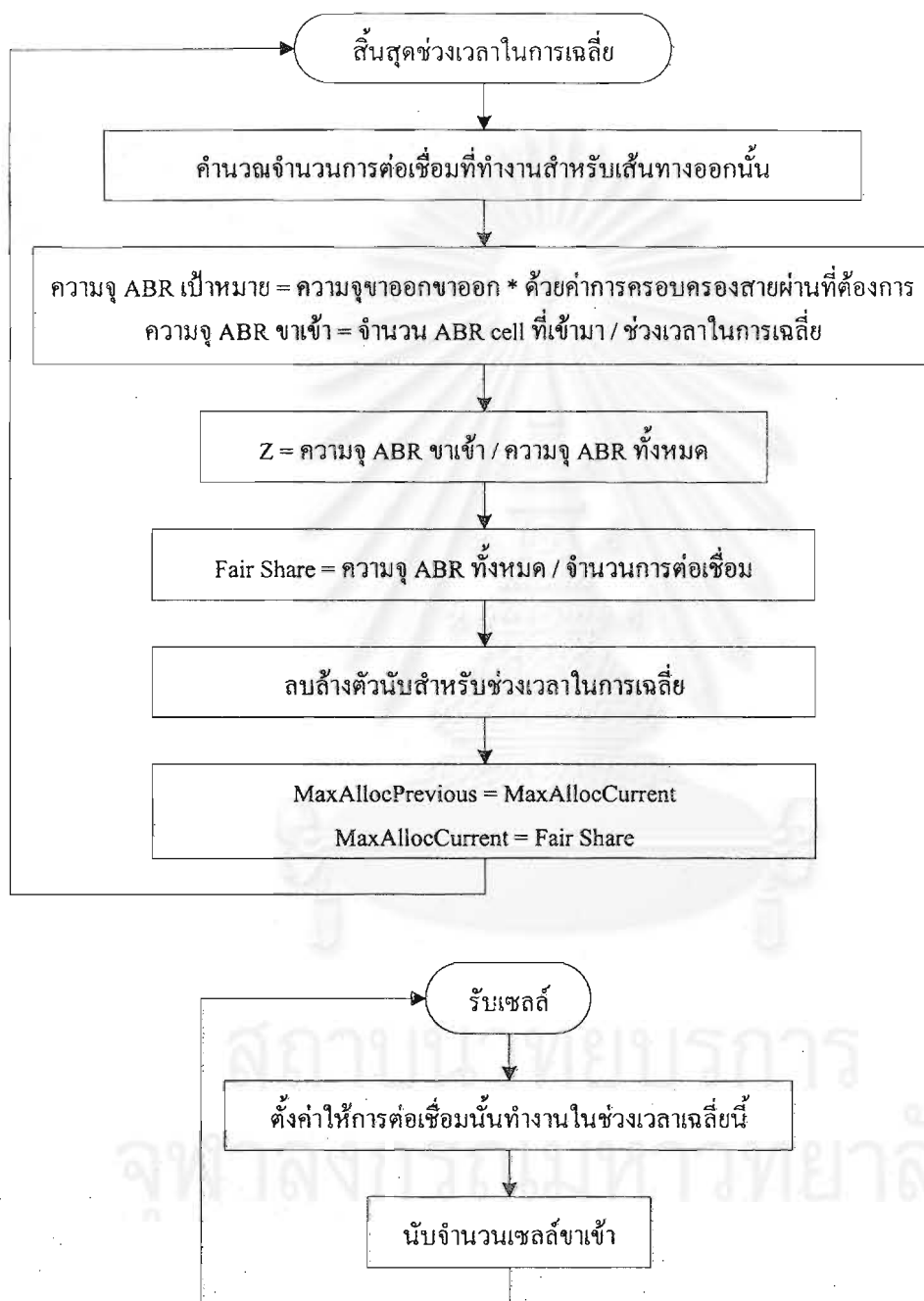
การคำนวณหาค่า Z ใหม่จะเกิดขึ้นทุกครั้งที่สิ้นสุดช่วงระยะเวลาหนึ่งเรียกว่า ช่วงเวลาในการเฉลี่ย (Averaging Interval – AI) ERICA เป็นวิธีที่พยายามจัดการทราฟฟิกให้มี ค่า Z เข้าใกล้ 1 ค่า ER จะถูกคำนวณเพื่อคำนึงถึงความเท่าเทียมกันระหว่างการเชื่อมโยงต่าง ๆ เมื่อค่า $Z < 1.1$ จะถือว่าไม่มีความคับคั่งให้ ER มีค่าประมาณกับค่าที่ส่งเดิม แต่หากค่า $Z > 1.1$ แสดงว่าเกิดความคับคั่งและจะให้ค่า ER เป็นค่า Fairshare จะเห็นว่าค่า ER ที่ได้นั้นจะเท่าเทียมสำหรับทุก ๆ การเชื่อมต่อ เมื่ออยู่ในสถานะคงตัวแล้วค่า Z จะมีค่าประมาณ 1 ถึง 1.1 ซึ่งแสดงถึงการใช้ความจุของสายส่งได้อย่างคุ้มค่า การให้ค่า Z ที่สถานะคงตัวมีค่ามากกว่า 1 เล็กน้อยเพื่อชดเชยการนำค่าการใช้งานสายสัญญาณเป้าหมายมาคูณกับแบนด์วิดท์ที่ใช้งานได้ นอกจากนี้ ERICA สามารถแก้ปัญหาเรื่องความเท่าเทียมกัน ในกรณีที่มีการเชื่อมต่อใดที่ไม่สามารถส่งได้ที่อัตราส่งเท่ากับ Fairshare ทำให้เกิด แบนด์วิดท์ส่วนที่เหลือจาก Fairshare นั้นไม่ได้ถูกใช้งาน ดังนั้น ERICA จึงเพิ่มตัวแปร VCshare เพื่อแก้ปัญหานี้ โดย VCshare จะมีค่าตามสมการที่ 2.4

$$VCshare = \frac{CCR}{Z} \quad (2.4)$$

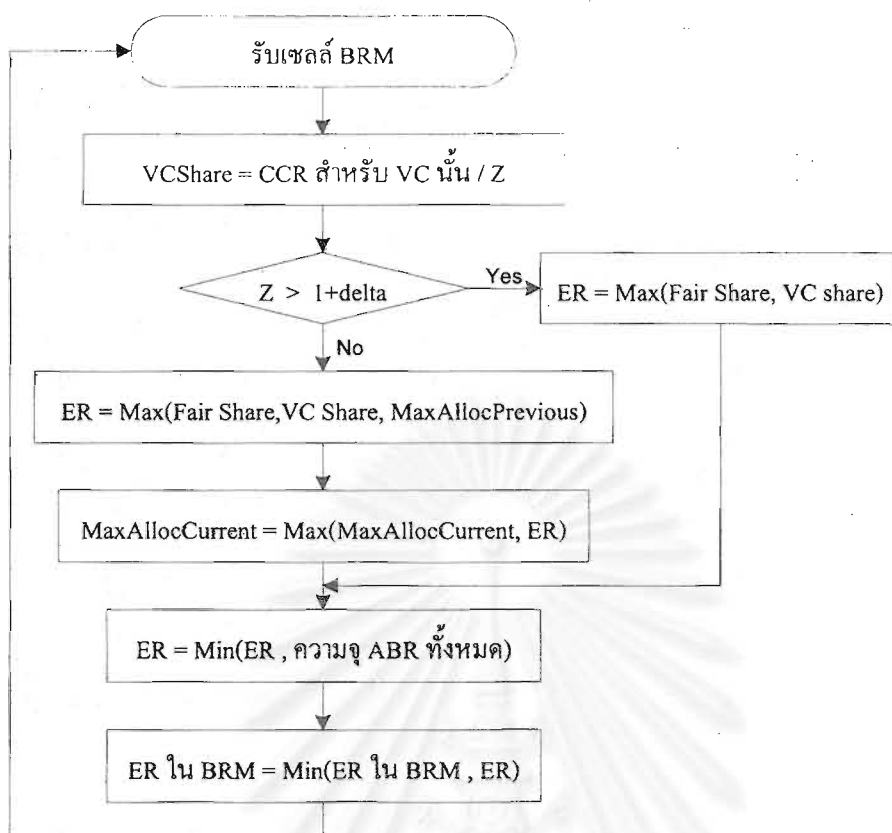
โดย CCR คือ ความเร็วในการส่งข้อมูลของผู้ส่ง

เมื่อมีแหล่งกำเนิดหนึ่งที่ใช้งานได้ไม่ถึง Fairshare จะทำให้ค่า Z น้อยกว่าหนึ่ง และค่า VCshare จะมีค่ามากขึ้น ทำให้การต่อเชื่อมอื่น สามารถเพิ่มอัตราการส่งของตนเองให้เป็น VCshare ซึ่งมากกว่า Fairshare ได้ ซึ่งเป็นการชดเชยกับความสามารถในการส่งที่เสียไปกับแหล่งกำเนิดที่ส่งได้ไม่เต็มที่ ให้กับแหล่งกำเนิดอื่น ๆ

ค่า ER ที่จะส่งกลับไปในเซลล์ RM นั้นจะเป็นค่ามากที่สุดระหว่าง VCshare และ Fairshare แต่มีค่าไม่เกินแบนด์วิดท์ที่สามารถใช้งานได้ และไม่เกินค่า ER เดิมที่อยู่ในเซลล์ RM ถ้าการทำงานของวิธี ERICA สามารถแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 วิธีการควบคุมความคับคั่งแบบ ERICA



รูปที่ 2.2 วิธีการควบคุมความคับคั่งแบบ ERICA (ต่อ)

นอกจากนี้ ERICA ยังได้รับการปรับปรุงให้เป็น ERICA+ [4] ที่สามารถควบคุมขนาดหน่วยความจำได้ โดยการปรับค่าความจุเป้าหมายให้เหมาะสมกับขนาดหน่วยความจำในปัจจุบัน ซึ่งมีค่าตามสมการ 2.5

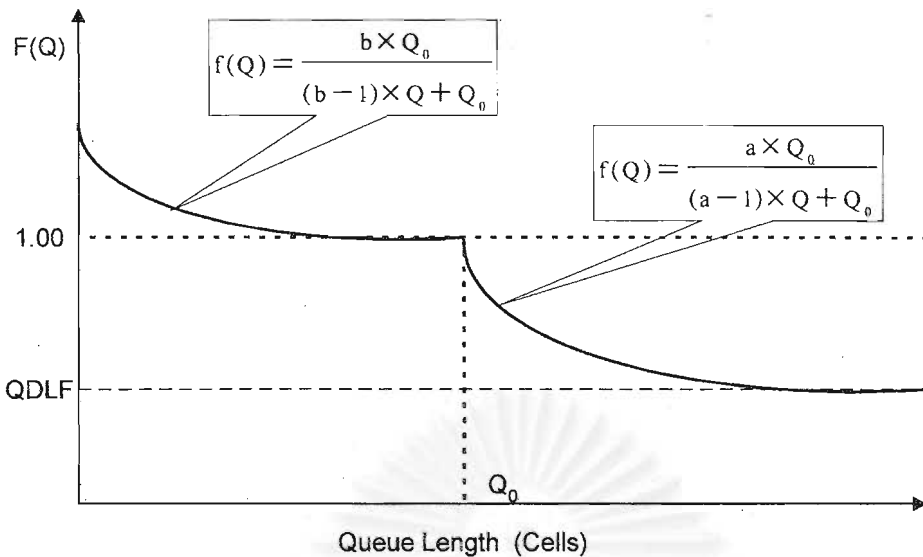
อัตราข้อมูลเป้าหมาย = $f(Q) \times$ อัตราข้อมูลทั้งหมดของ ABR

$$\text{และ } f(Q) = \begin{cases} \max(\text{QDLF}, \frac{a \times Q_0}{(a-1) \times Q + Q_0}) & \text{for } Q > Q_0 \\ \frac{b \times Q_0}{(b-1) \times Q + Q_0} & \text{for } 0 \leq Q \leq Q_0 \end{cases} \quad (2.5)$$

โดย QDLF คือ Queue Drain Limit Factor ซึ่งเป็นค่า $f(Q)$ ต่ำสุดที่ตั้งไว้

a, b คือ เป็นค่าคงที่ที่แสดงถึงความเร็วในการลดลงของค่า $f(Q)$

Q_0 คือ ขนาดหน่วยความจำที่แสดงถึงความคับคั่ง โดยถูกตั้งค่าไว้ที่ค่าที่เหมาะสม



รูปที่ 2.3 วิธีการควบคุมหน่วยความจำใน ERICA+

เมื่อนำวิธี ERICA มาใช้บนการส่งแบบหนึ่งจุดไปสู่หลายจุด ไม่จำเป็นต้องปรับแต่งตัวแปรต่าง ๆ ให้เหมาะสม และสามารถใช้ร่วมกับวิธีการรวบรวมข้อมูลการป้อนกลับทุกวิธีได้

2.6 วิธี FMMRA (Fast Max-Min Rate Allocation)

FMMRA [5,6,7] เป็นวิธีที่ปรับค่าอัตราส่งแน่นอนโดยใช้วิธีการ Max-Min Fairness การต่อเชื่อมที่ผ่านเส้นทางขาออกเดียวกันของสวิตช์จะถูกแบ่งเป็น 2 ประเภท คือ

- Bottlenecked connection เป็นการต่อเชื่อมที่ไม่สามารถปรับอัตราส่งของมันได้เท่ากับอัตราส่งยุติธรรมของสวิตช์นี้ เนื่องจากได้รับอัตราส่งที่น้อยกว่าจากสวิตช์อื่น หรือ ขีดจำกัดที่อัตราส่งสูงสุดของการต่อเชื่อมนี้
- Non-bottlenecked connection หรือ bottlenecked link เป็นการต่อเชื่อมที่อัตราส่งของมันถูกจำกัดอยู่ด้วยความจุของสายส่ง

FMMRA จะคำนวณค่า advertised rate - γ_i ซึ่งเป็นค่าที่เหมาะสมสำหรับการป้อนกลับไปเป็นค่าอัตราส่งแน่นอน ถ้าหากการต่อเชื่อมไหนไม่สามารถส่งได้เท่ากับ γ_i ก็จะถูกตั้งให้เป็น bottlenecked connections และ จะได้รับการจำกัดแบนด์วิดท์ของมันไว้เป็น bottlenecked bandwidth เพื่อที่สามารถนำแบนด์วิดท์ส่วนที่เหลือไปเพิ่มให้กับการต่อเชื่อมอื่น ๆ ได้ โดยการปรับค่า γ_i ให้เหมาะสม ตัวแปร β และ λ ใช้ในการแสดงสถานะ Bottleneck ของการต่อเชื่อมนั้น ๆ และ ขนาดแบนด์วิดท์สำหรับแต่ละการต่อเชื่อม ตามลำดับ

ตัวแปรต่าง ๆ ในวิธีการนี้จะได้รับการปรับแต่งทุก ๆ ครั้งที่ได้รับเซลล์ RM ขากลับ สวิตช์จะทำการคำนวณค่า $\Delta\lambda$, $\Delta\beta$ และ $\gamma_i(t)$ ตามสมการ 2.6

$$\begin{aligned}
 \Delta\lambda &= \begin{cases} \lambda_i^{ER} - \lambda_i \beta_i^i & \text{if } \lambda_i^{ER} < \gamma_i, \\ -\lambda_i \beta_i^i & \text{if } \lambda_i^{ER} \geq \gamma_i. \end{cases} \\
 \Delta\beta &= \begin{cases} 1 - \beta_i^i & \text{if } \lambda_i^{ER} < \gamma_i, \\ -\beta_i^i & \text{if } \lambda_i^{ER} \geq \gamma_i. \end{cases} \\
 \gamma_i(t^+) &= \begin{cases} C_i^A & \text{if } N_i = 0, \\ \frac{C_i^A - \bar{C}_i(t^+)}{N_i - \bar{N}_i(t^+)} & \text{if } N_i > \bar{N}_i, \\ \gamma_i & \text{if } N_i = \bar{N}_i \end{cases}
 \end{aligned} \tag{2.6}$$

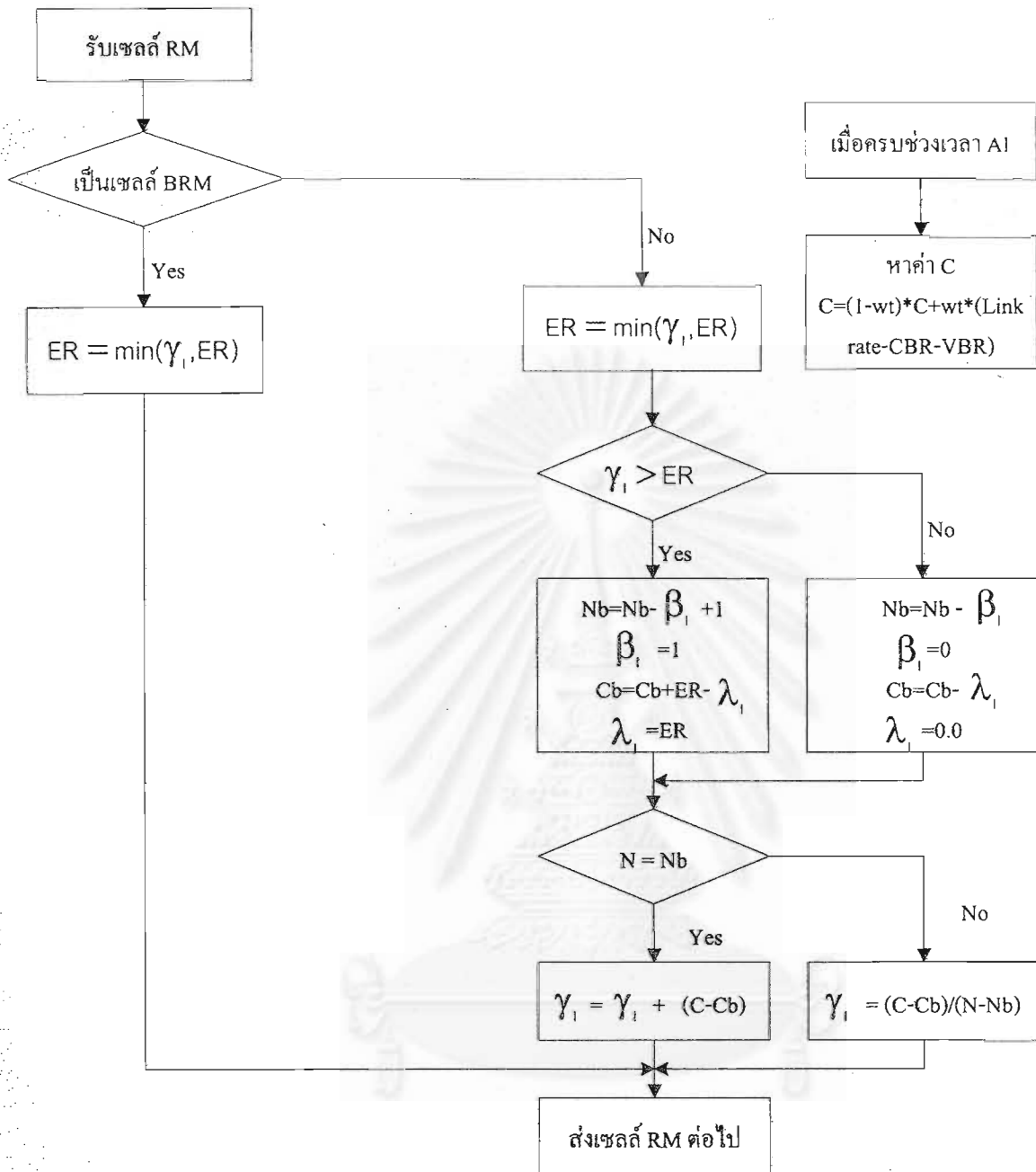
โดย λ_i^{ER} คือ ค่า ER ที่รับได้จากเซลล์ RM ขากลับที่เข้ามา
 λ_i^i คือ bottlenecked bandwidth ของ VCI นั้นที่สายส่ง I
 β_i^i คือ bottlenecked connections ของ VCI นั้นที่สายส่ง I
 N_i คือ จำนวนการต่อเชื่อมทั้งหมดที่ผ่านสายส่ง I
 $\bar{N}_i(t)$ คือ จำนวนการต่อเชื่อมที่เป็น bottlenecked connection
 C_i^A คือ ความจุทั้งหมดของการให้บริการแบบ ABR
 $\bar{C}_i(t)$ คือ ความจุ bottlenecked bandwidth ทั้งหมดที่สายส่ง I

จากนั้นสวิตช์จะคำนวณค่าตัวแปร λ , β และ N_i ใหม่สำหรับการคำนวณในรอบต่อไปและ ส่งกลับค่า ER ตามสมการ 2.7

$$ER_in_RM = \min(\gamma_i, \lambda_i^{ER}) \tag{2.7}$$

นอกจากนี้วิธี FMMRA ยังสามารถปรับแต่งค่า ER ในเซลล์ RM ทั้งทางด้านขาไปและขากลับ ทำให้สวิตช์สามารถรับข่าวสารความคับคั่งได้จากสองทางคือด้าน upstream และ downstream มีผลทำให้การป้อนกลับมีความเร็วมากยิ่งขึ้น ลำดับการทำงานของวิธี FMMRA นั้นแสดงในรูปที่ 2.4

เมื่อนำวิธี FMMRA มาใช้บนการส่งแบบหนึ่งจุดไปสู่หลายจุด จำเป็นต้องเพิ่มความซับซ้อนของตัวแปร β และ λ เพิ่มขึ้นมา โดยปกติตัวแปรทั้งสองเป็นตัวแปรที่มีทุก ๆ การเชื่อมต่อ แต่เมื่อนำมาใช้ในการส่งแบบหนึ่งจุดไปหลายจุด ซึ่งทำให้ VC หนึ่ง ๆ อาจมีเส้นทางออกได้หลายทาง ดังนั้นตัวแปรดังกล่าวจึงต้องเพิ่มปริมาณให้รองรับจำนวนเส้นทางออกให้ได้ทุกเส้นทาง โดยเพิ่มจำนวนตัวแปรให้มีมิติเท่ากับจำนวนเส้นทางขาออกสำหรับ VC นั้น



รูปที่ 2.4 วิธีการควบคุมความคับคั่งแบบ FMMRA

จุฬาลงกรณ์มหาวิทยาลัย

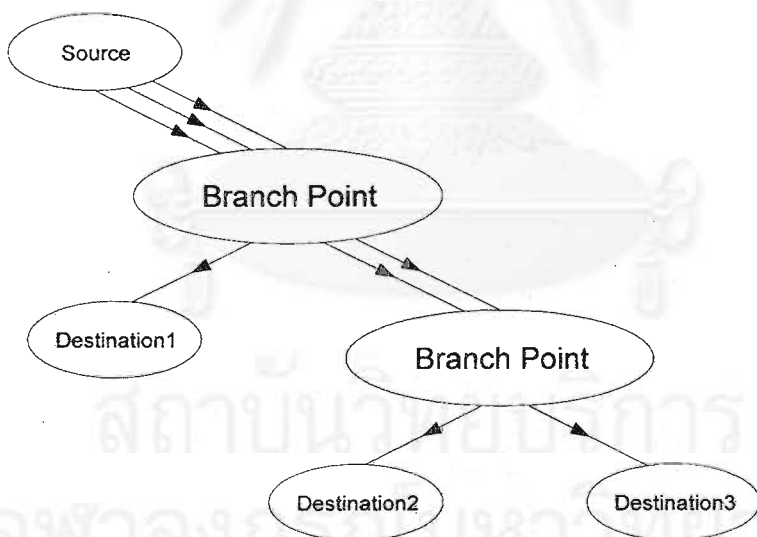


บทที่ 3

การส่งข้อมูลประเภท ABR จากหนึ่งไปจุดสู่หลายจุดในโครงข่าย ATM

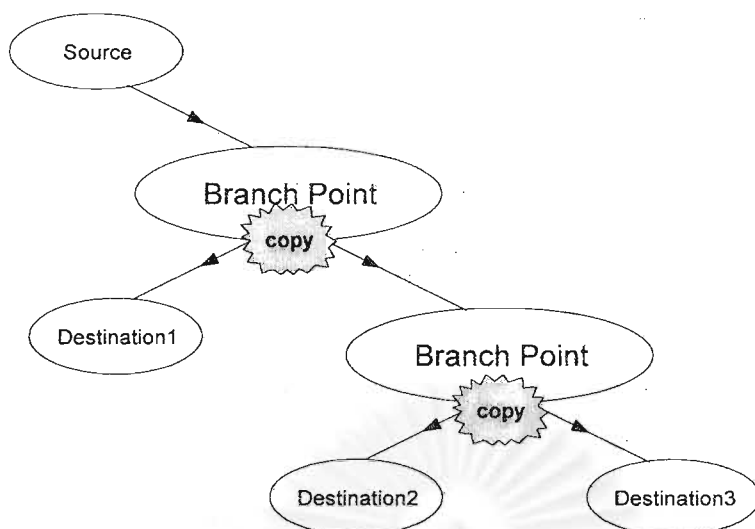
3.1 กล่าวนำ

โครงข่าย ATM นั้น ถูกออกแบบมาเพื่อรองรับการเชื่อมต่อจากหนึ่งจุดไปสู่หลายจุด เพื่อรองรับการใช้งานหลายประเภท อาทิ เช่น video conferencing การ multicast ใน LANE หรือ การสอนระยะทางไกล ฯลฯ การส่งแบบหนึ่งจุดไปสู่หลายจุดในโครงข่าย ATM นั้นต้องส่งข้อมูลเข้าสู่โครงข่าย โดยไม่เพิ่มภาระให้กับโครงข่ายโดยไม่จำเป็น ด้วยหลักการที่ว่า ผู้ส่งต้องไม่มีการส่งข้อมูลที่ซ้ำเติมไปในเส้นทางเดียวกันไปสู่ปลายทางต่าง ๆ กัน ดังรูปที่ 3.1 ดังนั้นการตัดลอกเซลล์ที่สวิตช์ระหว่างทางจึงเป็นสิ่งจำเป็น เพื่อให้ได้ประสิทธิภาพในการส่งมากที่สุดดังรูปที่ 3.2 ด้วยเหตุนี้ แหล่งกำเนิดจำเป็นต้องทราบรูปแบบโครงข่ายเป็นอย่างไรดี เพื่อสามารถสร้าง Multicast tree สำหรับส่งข้อมูล ในที่นี้ Multicast tree นั้นประกอบด้วย root นั่นคือแหล่งกำเนิดนั่นเอง , branch point ได้แก่ อุปกรณ์สวิตช์ที่ทำหน้าที่ตัดลอกข้อมูล, branch คือสายนำสัญญาณที่นำข้อมูลไปสู่ปลายทาง และ leave คือ อุปกรณ์ปลายทางที่มีหน้าที่รับข้อมูล



รูปที่ 3.1 การส่งแบบหนึ่งจุด ไปสู่หลายจุดที่ไม่มีประสิทธิภาพ

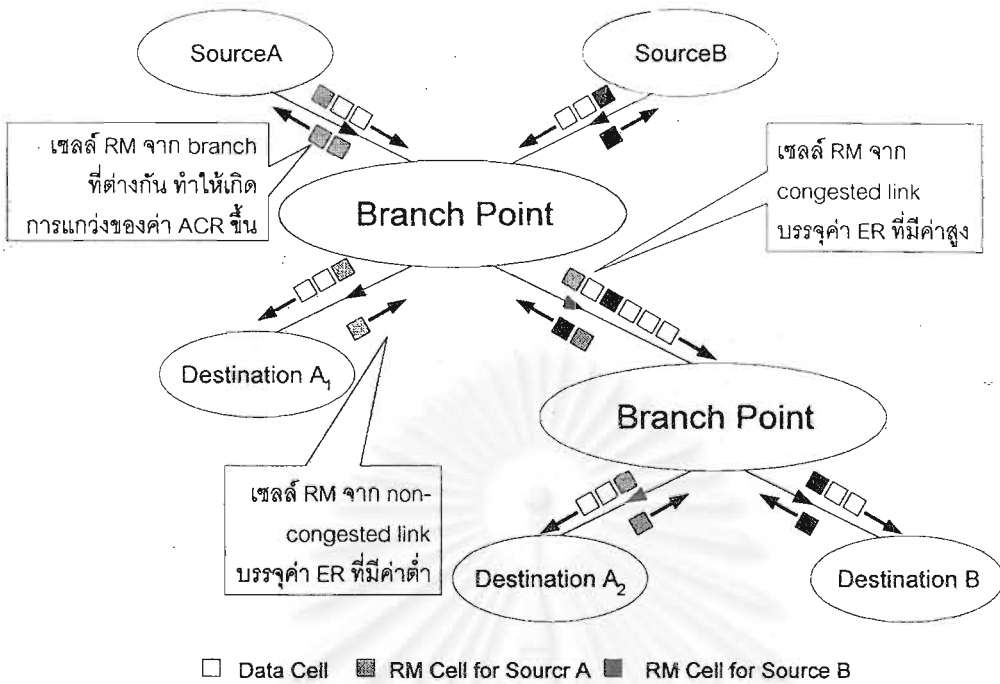
สำหรับการให้บริการแบบ ABR นั้นต้องอาศัยการป้อนกลับข่าวสารความคับคั่ง ดังนั้นหากต้องการให้การส่งจากหนึ่งจุดไปสู่หลายจุดนั้นมีประสิทธิภาพจึงจำเป็นต้องมี การควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุดสำหรับการให้บริการแบบ ABR



รูปที่ 3.2 การส่งแบบหนึ่งจุด ไปสู่หลายจุดที่มีประสิทธิภาพ

3.2 การควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุดสำหรับการให้บริการแบบ ABR

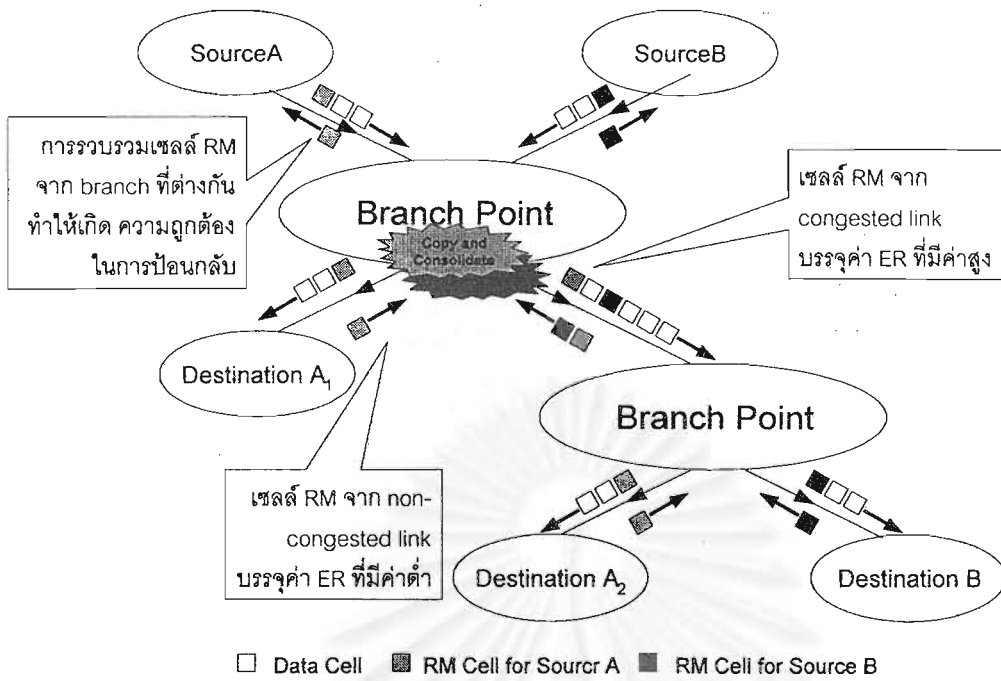
หลักการการควบคุมการส่งแบบหนึ่งจุดไปสู่หลายจุด (Point-to-multipoint connection) นั้นได้รับการพัฒนามาจาก การส่งแบบหนึ่งจุดไปสู่หนึ่งจุด โดยนำวิธีควบคุมความคับคั่งที่ใช้ในการส่งแบบหนึ่งจุดไปสู่หนึ่งจุด มาใช้ โดยการให้สวิตช์นั้นเป็นผู้คัดลอกเซลล์ต่าง ๆ ออกไปตาม branch ต่าง ๆ อย่างไรก็ตามการรวบรวมข่าวสารความคับคั่งที่ป้อนกลับจาก branch ที่ต่างกัน สำหรับแต่ละ branch point เป็นเรื่องสำคัญ เนื่องจากข่าวสารที่ป้อนกลับไปหาแหล่งกำเนิดไม่ควรจะเพิ่มขึ้นตามจำนวนของปลายทางสำหรับการเชื่อมต่อ นั้น ปัญหาดังกล่าวเรียกว่า feedback implosion โดยสามารถแสดงได้ดังรูปที่ 3.3 จะเห็นว่าเซลล์ FRM ที่สวิตช์ทำหน้าที่คัดลอกไปสู่ปลายทางที่ต่างกันนั้น ล้วนถูกตอบกลับมาจากปลายทางต่าง ๆ และหากสวิตช์นั้นปล่อยให้เซลล์ BRM นั้น ไปสู่แหล่งกำเนิดนั้น จะทำให้ค่า ACR นั้นเกิดการแกว่ง เนื่องจากค่าข่าวสารความคับคั่งที่ป้อนกลับจากแต่ละ branch นั้น ไม่เหมือนกัน ทำให้แหล่งกำเนิดได้ข่าวสารที่ไม่แน่นอน และ ค่าอัตราส่งที่แหล่งกำเนิดนั้นจะเกิดการแกว่งขึ้นมา โดยปัญหาที่เกิดขึ้นนี้เรียกว่าเกิด consolidation noise ขึ้น และจะทำให้โครงข่ายเกิดความคับคั่งขึ้น อันจะทำให้เกิดการสูญเสียเซลล์ข้อมูลได้



รูปที่ 3.3 ความผิดพลาดจากการไม่รวบรวมข่าวสารการป้อนกลับ

วิธีการควบคุมความคับคั่งในการส่งจากจุดเดียวสู่จุดเดียวจะใช้การปรับอัตราส่งที่ต่ำสุด ที่สวิตช์ตลอดเส้นทางรองรับได้ ดังนั้นในการส่งจากหนึ่งจุดไปสู่หลายจุดนั้นควรปรับอัตราส่งให้เป็นค่าต่ำที่สุดสำหรับที่ทุกเส้นทางจากต้นทางสู่ปลายทางทุกอัน สามารถรองรับได้ ดังนั้น branch point จึงจำเป็นต้องมีวิธีการรวบรวมข้อมูลความคับคั่งจากเซลล์ BRM เพื่อให้ข้อมูลที่ส่งกลับไปพร้อมเซลล์ RM มีข้อมูลความคับคั่งของโครงข่ายที่ดีที่สุดจากทุก branch วิธีการดังกล่าวเรียกว่า วิธีการรวบรวมข่าวสารการป้อนกลับ (consolidation algorithm) การรวบรวมข้อมูลความคับคั่งที่ดีควรทำให้อัตราส่งที่แหล่งกำเนิดมีค่าต่ำที่สุดที่สวิตช์ทุกตัวใน multicast tree สามารถรองรับได้ เพื่อป้องกันการเกิดความคับคั่งขึ้นและพยายามลด consolidation noise ให้มากที่สุด

อย่างไรก็ตามการป้อนกลับที่ถูกต้องสู่แหล่งกำเนิดนั้นอาจนำไปสู่ การรอข่าวสารที่จำเป็นทั้งหมดของแต่ละ branch ซึ่งจะทำให้ความเร็วในการป้อนกลับข่าวสารดังกล่าวช้ามาก โดยเฉพาะในสถานะที่โครงข่ายมีจำนวนเซลล์ RM น้อย ทำให้แหล่งกำเนิดไม่สามารถปรับอัตราส่งให้เข้าสู่อัตราส่งที่แหล่งกำเนิดสามารถรองรับได้อย่างแท้จริง และ อาจก่อให้เกิดปัญหาความคับคั่ง หรือ ปัญหาการใช้งานสายสัญญาณไม่คุ้มค่าขึ้นในโครงข่าย ปัญหาดังกล่าวเรียกว่าเกิด consolidation delay โดยวิธีการรวบรวมข่าวสารความคับคั่งที่ดีนั้นต้องมีผลตอบแทนชั่วคราวที่รวดเร็ว เพื่อลดปัญหา consolidation delay และทำให้แหล่งกำเนิดสามารถตอบสนองต่อสถานะของโครงข่ายที่เปลี่ยนไปได้ การทำงานของ consolidation algorithm สามารถแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 การทำงานของวิธีรวบรวมข่าวสารความคับคั่ง

3.3 วิธีการรวบรวมข่าวสารความคับคั่ง

วิธีการรวบรวมข่าวสารการป้อนกลับมีหลายวิธีดังได้เสนอมาก่อนหน้านี้ในบทความ [8-11] หลักการในการพิจารณาวิธีการรวบรวมข้อมูลความคับคั่งนั้นจะพิจารณาจาก ความเร็วในการป้อนกลับ และ ความถูกต้องของข้อมูลที่ป้อนกลับ ซึ่งวิธีการที่เน้นความเร็วในการป้อนกลับนั้น จะมีเวลาในการป้อนกลับที่เร็วทำให้แหล่งกำเนิดสามารถปรับอัตราส่งให้ทันต่อสถานะที่เปลี่ยนไปของโครงข่ายได้ แต่ก็ทำให้ branch point ไม่สามารถรวบรวมข่าวสารความคับคั่งจากทุก branch ก่อนการป้อนกลับ ทำให้เกิดการผิดพลาดในการป้อนกลับอันนำไปสู่การเกิด Consolidation noise ในทางกลับกันหากวิธีรวบรวมความคับคั่งนั้นเน้นความถูกต้องในการป้อนกลับ โดยการรวบรวมข่าวสารที่จำเป็นมากที่สุดก่อนการป้อนกลับนั้น ซึ่งในวิธีการแรก ๆ ที่เสนอนั้นจะพยายามให้ได้การป้อนกลับที่เร็วโดยที่ไม่สนใจความถูกต้องมากนัก แต่วิธีการใหม่ ๆ พยายามพัฒนาจากหลักการที่เน้นให้มีความถูกต้องของการป้อนกลับมากที่สุด และพัฒนาให้ได้การป้อนกลับที่เร็วขึ้น โดยไม่ต้องรอข้อมูลมากนัก อย่างไรก็ตาม จุดมุ่งหมายสำคัญของวิธีการรวบรวมข่าวสารความคับคั่งที่สำคัญคือ มีความถูกต้องในการป้อนกลับ ในขณะที่มีความรวดเร็วในการป้อนกลับ ซึ่งจะทำให้แหล่งกำเนิดสามารถตอบสนองต่อสถานะที่เปลี่ยนไปของโครงข่ายได้อย่างทันท่วงที วิธีการรวบรวมข่าวสารความคับคั่งที่พัฒนาขึ้นมานั้นมีมากมายหลายวิธี อาทิเช่น Robert algorithm, TS algorithm, RSS algorithm, Wait-for-all algorithm และ Immediate Rate Calculation

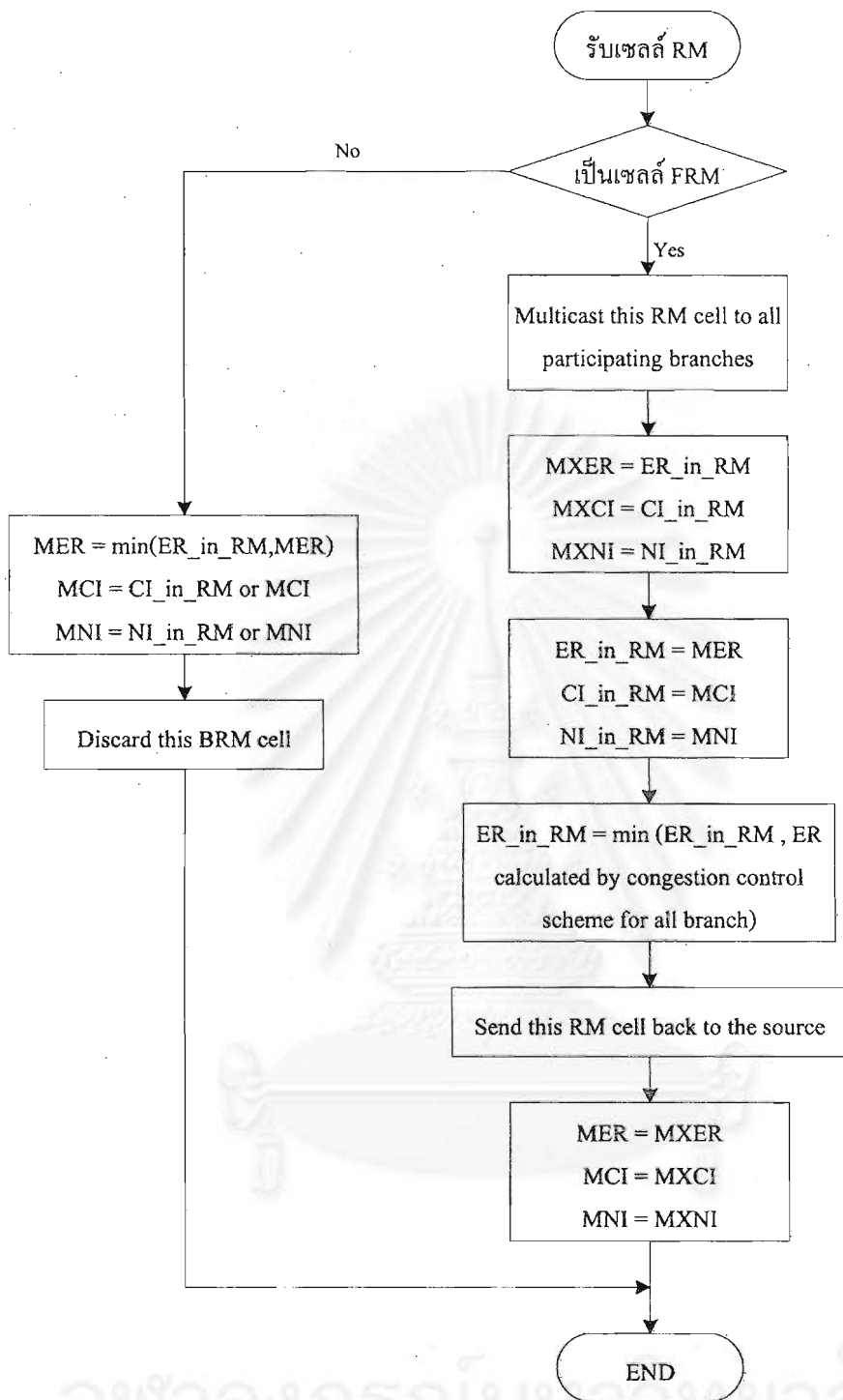
3.3.1 Robert Algorithm

Branch point จะรวบรวมข่าวสารการป้อนกลับซึ่งได้แก่ ค่า ER, CI และ NI จากเซลล์ BRM ทุกครั้งเมื่อได้รับเซลล์ BRM โดยค่าต่าง ๆ ข้างต้นจะทำการรวบรวมเก็บไว้ในตัวแปร MER, MCI และ MNI ตามลำดับ ตัวแปรทั้งสามนั้นจะจัดการโดยเป็นตัวแปรประเภท per-VC นั่นคือ สวิตช์จะมีตัวแปรดังกล่าวข้างต้นนี้เป็นจำนวนเท่ากับปริมาณการเชื่อมต่อจากหนึ่งจุดไปสู่หลายจุดที่ผ่านสวิตช์นั้น และจะจัดการค่าต่าง ๆ แยกเป็นแต่ละการเชื่อมต่อ โดยค่า MER จะเป็นค่า ER น้อยสุดจากทุก branch สำหรับในการเชื่อมต่อ นั้น ส่วน MCI และ MNI จะมีค่าเป็น 1 ก็ต่อเมื่อค่า CI และ NI จาก branch ใด branch หนึ่งมีค่าเป็น 1 ตามลำดับ

เมื่อสวิตช์ได้รับเซลล์ FRM จะสร้างเซลล์ BRM เพื่อป้อนกลับข่าวสารความคับคั่ง เซลล์ BRM ที่สร้างขึ้นใหม่นี้จะถูกบรรจุด้วยข่าวสารความคับคั่งที่รวบรวมไว้ก่อนแล้วนั้นคือ MER, MCI และ MNI นั่นเอง จากนั้นสวิตช์จะคำนวณค่า ER สำหรับการป้อนกลับจากทุก branch โดยใช้วิธีควบคุมความคับคั่งที่มีอยู่ ค่าต่าง ๆ ที่ได้จะนำมาเปรียบเทียบกับค่าที่อยู่ในเซลล์ BRM โดยค่าที่น้อยที่สุดจะถูกบรรจุอยู่ในเซลล์ BRM ก่อนการส่งเซลล์ BRM ตัวนี้ไปสู่แหล่งกำเนิด เมื่อสวิตช์ทำการส่งเซลล์ BRM ดังกล่าวไปสู่แหล่งกำเนิดแล้ว ตัวแปร MER จะถูกลบค่าให้เป็นค่า ER ที่อยู่ในเซลล์ FRM ที่เพิ่งได้รับนั่นเอง ส่วน MCI และ MNI จะถูกลบค่าให้เป็น 0 วิธีนี้นั้นมีความเร็วในการป้อนกลับสูงมาก แต่ไม่มีความถูกต้องในการป้อนกลับ โดยเมื่อ branch point ได้รับเซลล์ FRM แล้วตอบกลับข่าวสารไปแล้วนั้น หากยังไม่ได้รับเซลล์ BRM ก่อนที่ได้จะรับเซลล์ FRM ตัวถัดไปก็จะทำการป้อนกลับข่าวสารที่ผิดพลาดได้ ลำดับการของวิธีการรวบรวมข่าวสารความคับคั่งวิธีนี้จะเป็นดังรูปที่ 3.5 และแผนผังการทำงานจะอยู่ในรูปที่ 3.6

Upon the receipt of a forward RM(ER,CI,NI) cell :
 Multicast this RM cell to all participating branches;
 Let $MXER = ER$, $MXCI = CI$, $MXNI = NI$;
 Let $ER = MER$, $CI = MCI$, $NI = MNI$;
 Let $ER = \min(ER, ER \text{ calculated by congestion control scheme for all branch})$
 Send this RM cell back to the source;
 Let $MER = MXER$, $MCI = MXCI$, $MNI = MXNI$;
 Upon the receipt of a backward RM(ER,CI,NI) Cell
 Let $MER = \min(MER, ER)$, $MCI = MCI \text{ or } CI$, $MNI = MNI \text{ or } NI$
 Discard the BRM Cell

รูปที่ 3.5 ลำดับการทำงานของ Robert algorithm



รูปที่ 3.6 แผนผังการทำงานของ Robert algorithm

3.3.2 TS Algorithm

วิธีรวบรวมข่าวสารความคับคั่งวิธีนี้พัฒนามาจาก Robert Algorithm โดยวิธีการดังกล่าวนั้นเกิด Consolidation noise ขึ้น เนื่องจากความไม่ถูกต้องในการรวบรวมข่าวสารความคับคั่ง ซึ่งบางกรณีนั้นอาจมีการสร้างและส่งเซลล์ BRM ไปสู่แหล่งกำเนิด โดยใช้ค่า MER ที่ยังไม่ได้รับการปรับแต่งจากการรับเซลล์ BRM จึงมีโอกาที่ค่า MER นั้นจะเป็นค่า ER ที่มาจาก FRM ครั้งก่อน ซึ่งส่วนใหญ่จะเป็น

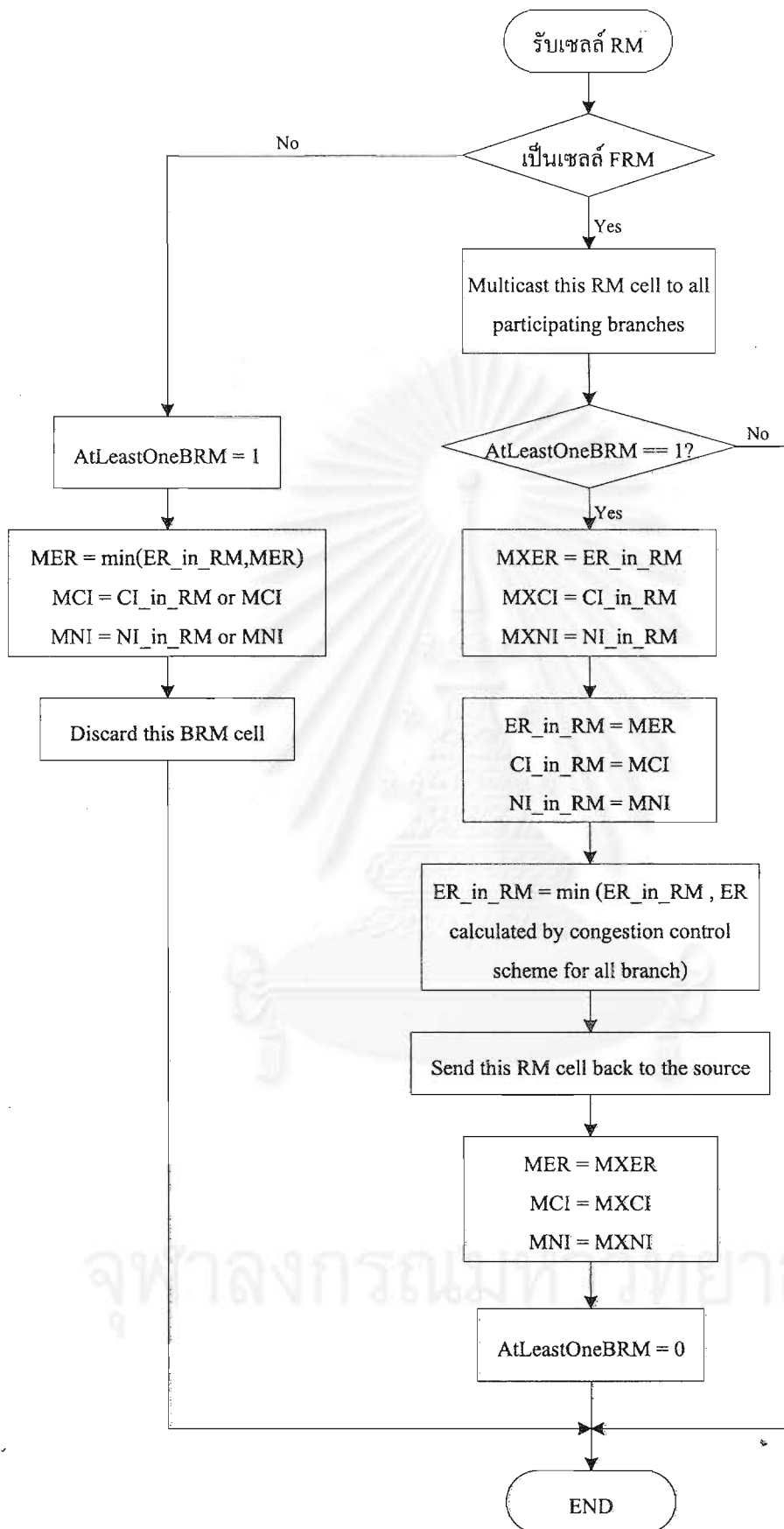
ค่า PCR นั้นเอง ดังนั้นจึงมีการเสนอวิธีให้ สวิตช์นั้นจะไม่สามารถป้อนกลับข่าวสารความคับคั่งได้ หากยังไม่ได้รับเซลล์ FRM และ ยังไม่ได้รับข่าวสารจากเซลล์ BRM อย่างน้อยหนึ่งตัวจาก branch ใด branch หนึ่งก่อน ซึ่งจะทำให้ค่า MER ที่รวบรวมไว้ที่สวิตช์นั้นจะได้รับการปรับแต่งจากการรับเซลล์ BRM ก่อนอย่างน้อยหนึ่งครั้ง เทคนิคดังกล่าวสามารถเพิ่มความถูกต้องในการรวบรวมข่าวสารการป้อนกลับได้ เนื่องจากทำให้ข่าวสารที่รวบรวมไว้ใน MER มีความถูกต้องมากขึ้นก่อนการป้อนกลับข่าวสารนั้น อย่างไรก็ตาม หาก MER ยังไม่รับการปรับแต่งจากการรับเซลล์ BRM จาก branch ที่มีปัญหาความคับคั่งจริง ๆ ก็ทำให้ค่า MER ที่เกิดขึ้นนั้นยังคงมีความผิดพลาดอยู่ นอกจากนี้จากผลการศึกษาต่อมาพบว่า หากสวิตช์ต้องเป็นฝ่ายตอบกลับเซลล์ FRM ลำดับการของวิธีการรวบรวมข่าวสารความคับคั่งวิธีนี้จะ เป็นดังรูปที่ 3.7 และแผนผังการทำงานจะอยู่ในรูปที่ 3.8

```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
  If AtLeastOneBRM then
    Let MXER = ER , MXCI = CI , MXNI = NI;
    Let ER = MER , CI = MCI , NI = MNI;
    Let ER = min (ER , ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let MER = MXER , MCI = MXCI , MNI = MXNI;
    Let AtLeastOneBRM = 0;
  Upon the receipt of a BRM(ER,CI,NI) Cell
    Let AtLeastOneBRM = 1;;
    Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
  Discard the BRM Cell

```

รูปที่ 3.7 ลำดับการทำงานของ TS algorithm



รูปที่ 3.8 แผนผังการทำงานของ Robert algorithm

3.3.3 RSS Algorithm

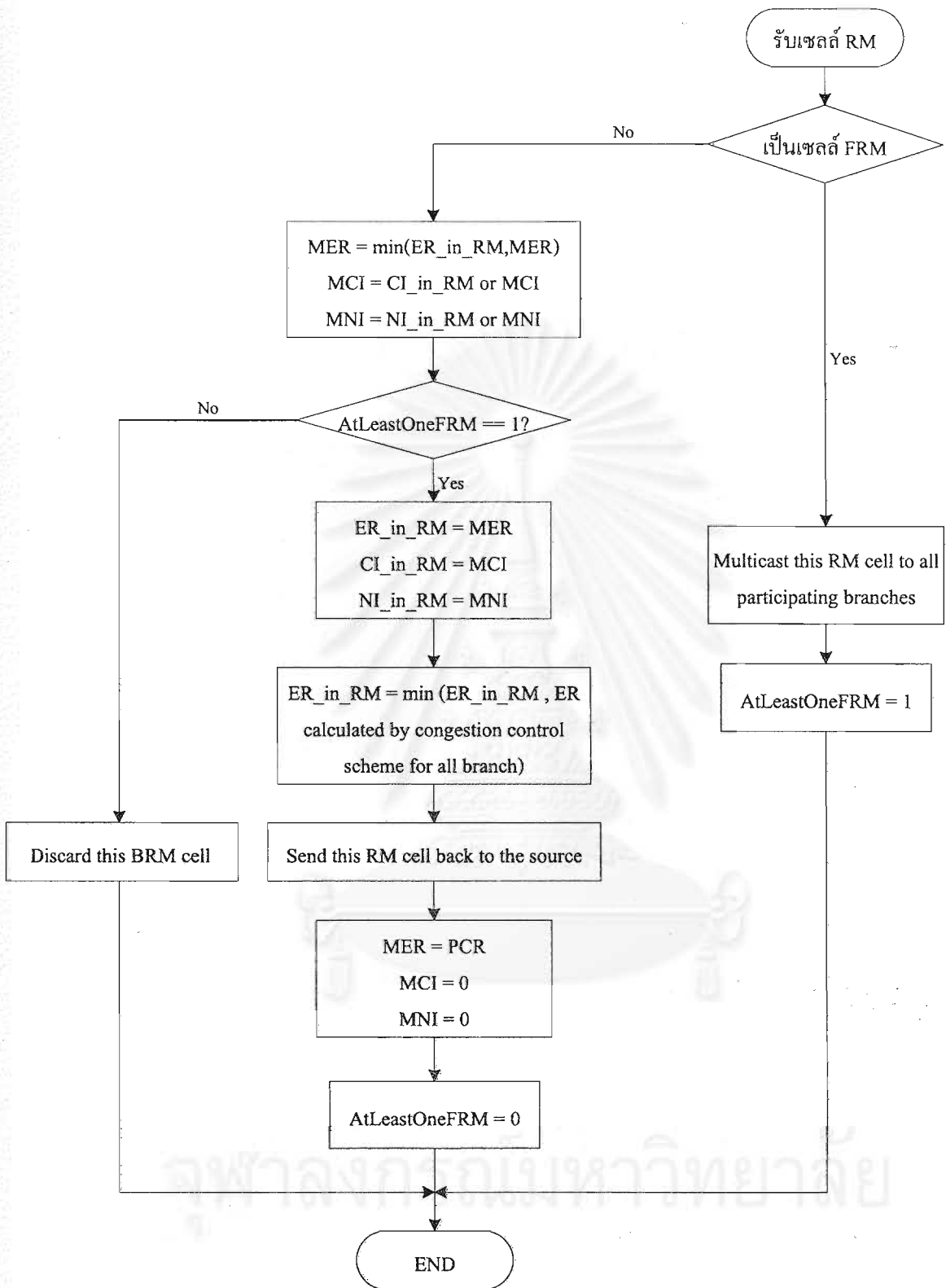
วิธีรวบรวมข่าวสารความคับคั่งวิธีนี้พัฒนามาจากวิธีการ Robert Algorithm และ TS Algorithm อีกทีหนึ่ง โดยเสนอว่าการสร้างเซลล์ BRM เพื่อตอบกลับหลังจากได้รับเซลล์ FRM ทุกครั้งนั้นเป็นการเพิ่มความซับซ้อนให้กับสวิตช์ขึ้นมา โดยสวิตช์ควรทำหน้าที่สลับเส้นทางและปรับแต่งค่าในเซลล์เพียงเท่านั้น ดังนี้ RSS Algorithm จึงได้ถูกพัฒนาขึ้น โดยใช้วิธีการตรวจนับว่าหากมีเซลล์ FRM ผ่านสวิตช์นั้นแล้วจะรอให้มีเซลล์ BRM ตัวต่อไปเพื่อปรับแต่งค่า ER, CI และ NI ในเซลล์ BRM ดังกล่าว โดยพิจารณาจากค่า MER, MCI และ MNI ที่รวบรวมไว้ที่สวิตช์ และ ค่าที่คำนวณได้จากวิธีควบคุมความคับคั่งของสวิตช์ จากนั้นจะส่งเซลล์ BRM นั้นผ่านไปหาแหล่งกำเนิดต่อไป โดยลำดับการทำงานและแผนผังการทำงานเป็นดังรูปที่ 3.9 และ 3.10 ตามลำดับ

```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
  Let AtLeastOneFRM = 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell
  Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
  If AtLeastOneFRM then
    Let ER = MER , CI = MCI , NI = MNI;
    Let ER = min (ER , ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let MER = PCR , MCI = 0 , MNI = 0;
    Let AtLeastOneFRM = 0;
  else
    Discard the BRM Cell
  
```

รูปที่ 3.9 ลำดับการทำงานของ RSS algorithm

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.10 แผนผังการทำงานของ RSS algorithm

3.3.4 Wait-for-all Algorithm

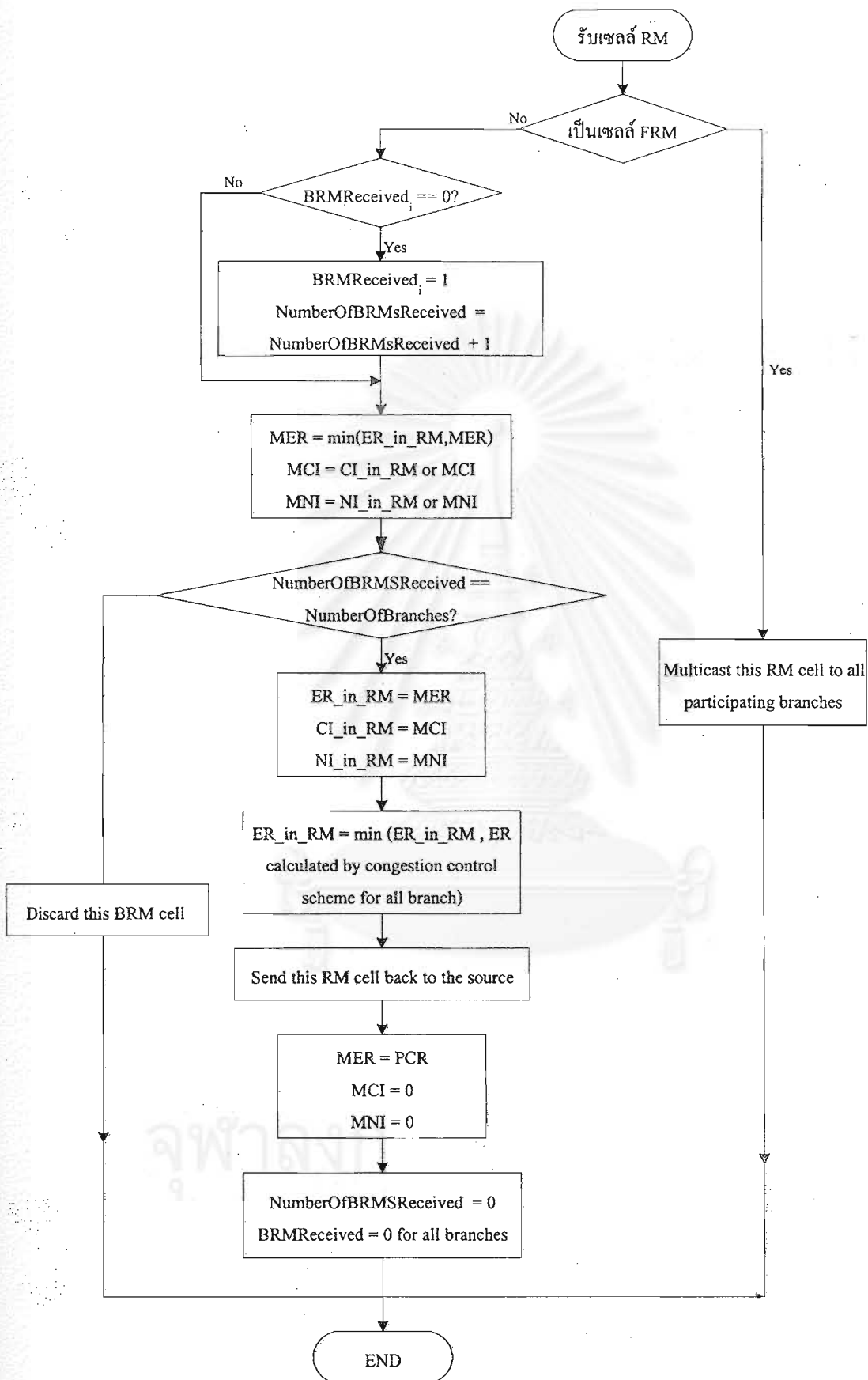
Branch point จะรวบรวมเซลล์ BRM ให้ครบตามจำนวนให้เท่ากับจำนวน branch ก่อน โดยการเก็บค่า ER ที่น้อยที่สุดจากทุก ๆ เซลล์ RM ที่เข้ามาไว้ในตัวแปร MER และเก็บค่า CI และ NI ที่แสดงถึงความคับคั่งไว้ จากนั้นจะส่งผ่านเซลล์ RM ดอกลับไปหาแหล่งกำเนิด โดยค่า ER ที่ดอกลับ คือค่าน้อยที่สุดระหว่างค่า MER ค่า ER ที่คำนวณได้จากวิธีการควบคุมความคับคั่ง การที่จะทำให้สวิตช์สามารถตรวจสอบปริมาณเซลล์ BRM ที่รับได้นั้น ต้องมีการเพิ่มตัวแปร NumberOfBRMsReceived เพื่อการตรวจนับเซลล์ BRM ที่ได้รับมาแล้ว โดยตัวแปรดังกล่าวมีการจัดการเป็น Per-VC ส่วนตัวแปร BRMReceived นั้นมีการจัดการเป็น Per-output นั่นคือมีการจัดการตัวแปรต่อหนึ่งเส้นทางขาออก โดยตัวแปร BRMReceived มีไว้เพื่อตรวจสอบการรับเซลล์ BRM สำหรับเส้นทางขาออกนั้น เมื่อจำนวนเซลล์ BRM ที่รับได้ที่สวิตช์มีค่าเท่ากับจำนวน branch ของการเชื่อมต่อนั้น สวิตช์จะทำการส่งเซลล์ BRM ที่เพิ่งได้รับนั้นไปสู่แหล่งกำเนิด โดยมีการปรับแต่งค่า ER ก่อนการป้อนกลับ จากนั้นตัวแปร MER จะถูกกลบด้างเป็นค่า PCR ของการเชื่อมต่อนั้น ส่วน CI, NI และ NumberOfBRMsReceived จะได้รับการกลบด้างให้เป็น 0 สำหรับทุก branch และ BRMReceived จะถูกกลบด้างเป็น 0 สำหรับวิธีการนี้ทำให้ความถูกต้องของการป้อนกลับ แต่มีการดอกลับที่ช้ามาก เนื่องจากต้องรอเซลล์ RM ให้กลับมาครบทุก branch ก่อน ดังนั้นหากว่ามีปลายใดปลายหนึ่งที่มี propagation delay มากจะทำให้แหล่งกำเนิดต้องรอให้เซลล์ BRM นั้นเดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อน อันจะเป็นการถ่วงให้เวลาในการป้อนกลับในช่วงแรกนั้นช้ามาก โดยเฉพาะในกรณีที่มีเซลล์ RM จำนวนน้อยอยู่ในโครงข่าย ยกตัวอย่างเช่น ในช่วงเริ่มต้นการส่งข้อมูล ลำดับการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.11 และ 3.12 ตามลำดับ

```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i :
  If not BRMReceived, then
    Let BRMReceived = 1;
    Let NumberOfBRMsReceived = NumberOfBRMsReceived + 1;
  Let MER = min (MER, ER), MCI = MCI or CI, MNI = MNI or NI
  If (NumberOfBRMSReceived == NumberOfBranches) then
    Let ER = MER, CI = MCI, NI = MNI;
    Let ER = min (ER, ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let MER = PCR, MCI = 0, MNI = 0;
    Let NumberOfBRMSReceived = 0;
    Let BRMReceived = 0 for all branch;
  else
    Discard the BRM Cell

```

รูปที่ 3.11 ลำดับการทำงานของ Wait-for-all algorithm



รูปที่ 3.12 แผนผังการทำงานของ Wait-for-all algorithm.

3.3.5 Immediate Rate Calculation Algorithm

วิธีรวบรวมข่าวสารความคับคั่งนี้พัฒนามาจาก Wait-for-all Algorithm หากแต่เพิ่มความสามารถในการตรวจจับสถานะโหลดเกินของโครงข่าย (Overload detected) โดยมีวิธีการตรวจสอบโดยให้สวิตช์เปรียบค่า ER ในเซลล์ BRM ที่เพิ่งได้รับกับค่า ER ที่ใช้ในการป้อนกลับครั้งที่แล้ว (Last_ER) ทุกครั้งที่ได้รับเซลล์ BRM และหากค่า ER ในเซลล์ BRM นั้นมีค่าน้อยกว่าค่า Last_ER คูณค่ากับ Threshold หนึ่ง ๆ แสดงว่าโครงข่ายเริ่มเกิดความคับคั่งขึ้น เนื่องจากค่าอัตราส่งที่สวิตช์แนะนำมีค่าน้อยลง ซึ่งสามารถบ่งชี้ถึงความคับคั่งของโครงข่ายได้ ดังนั้นสวิตช์จะคำนวณค่า ER ใหม่จากค่า MER และค่าที่ได้จากการใช้วิธีควบคุมความคับคั่ง เพื่อปรับแต่งค่า ER ในเซลล์ BRM ที่เพิ่งได้รับ และทำการส่งผ่านเซลล์ BRM ที่เพิ่งได้รับนั้นกลับไปสู่แหล่งกำเนิดทันที โดยไม่รอให้มีการรับเซลล์ BRM จากทุก branch ก่อน

อย่างไรก็ตาม การส่งเซลล์ BRM ที่เกิดจากการใช้ Overload detected จะทำให้ปริมาณเซลล์ BRM ที่ป้อนกลับมีจำนวนมากขึ้น และอัตราส่วนระหว่างจำนวนเซลล์ FRM ที่แหล่งกำเนิดส่งไป และจำนวนเซลล์ BRM ที่แหล่งกำเนิดได้รับ จะมีค่าเกิน 1 อันจะทำให้วิธีควบคุมข่าวสารป้อนกลับที่แหล่งกำเนิดอาจเกิดปัญหาได้ เนื่องจากแหล่งกำเนิดนั้นมีการนับปริมาณเซลล์ FRM ที่ส่งไปและยังไม่ได้รับการตอบกลับไว้ ดังนั้นหากสวิตช์มีการส่งเซลล์ BRM กลับสู่แหล่งกำเนิดมากกว่าจำนวนเซลล์ FRM ที่ได้รับอาจทำให้เกิดปัญหาในการควบคุมความคับคั่งได้ ด้วยเหตุนี้ วิธีการนี้จึงแก้ปัญหาดังกล่าวด้วยการนับจำนวนการส่งเซลล์ BRM จากการใช้ Overload detected ไว้ และในรอบการป้อนกลับถัดไปเมื่อค่า ER ที่เก็บรวบรวมไว้ได้จากทุก Branch ไม่แสดงความคับคั่ง (ไม่เกิด Overload Indication) ก็จะงดการส่งเซลล์ BRM นั้น เพื่อชดเชยการส่งเซลล์ BRM จากการใช้ Overload Indication ไป

การใช้เทคนิค Overload detected นั้นสามารถช่วยเพิ่มความเร็วในการป้อนกลับในสถานะโหลดเกินแก่วิธี Wait-for-all algorithm ได้เป็นอย่างดี อย่างไรก็ตามเทคนิคดังกล่าวยังไม่สามารถเพิ่มความเร็วในการป้อนกลับได้มากนัก เนื่องจากการตรวจสอบสถานะ Overload detected นั้น มีการตรวจสอบจากค่า ER ที่ได้จาก branch โดยไม่ตรวจสอบสถานะของสวิตช์ตัวเองก่อนการตัดสินใจ ดังนั้นเพื่อเพิ่มความเร็วในการป้อนกลับขึ้นมา วิธี Immediate rate calculation จึงเพิ่มให้มีการคำนวณสถานะความคับคั่งของสวิตช์ตัวเองก่อน โดยการให้มีการคำนวณค่า ER โดยใช้วิธีควบคุมความคับคั่งที่มี ก่อนการตรวจสอบ Overload detected ซึ่งวิธีนี้ทำให้ตรวจสอบความคับคั่งได้อย่างรวดเร็วขึ้น โดยสามารถตรวจสอบสถานะความคับคั่งของสวิตช์ตนเอง และ สถานะความคับคั่งจาก branch ต่าง ๆ ว่าเกิดสถานะโหลดเกินขึ้นในโครงข่ายหรือไม่ และดำเนินการส่งเซลล์ BRM พิเศษเพื่อป้อนกลับความคับคั่งสู่แหล่งกำเนิด ลำดับการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.13 และ 3.14 ตามลำดับ

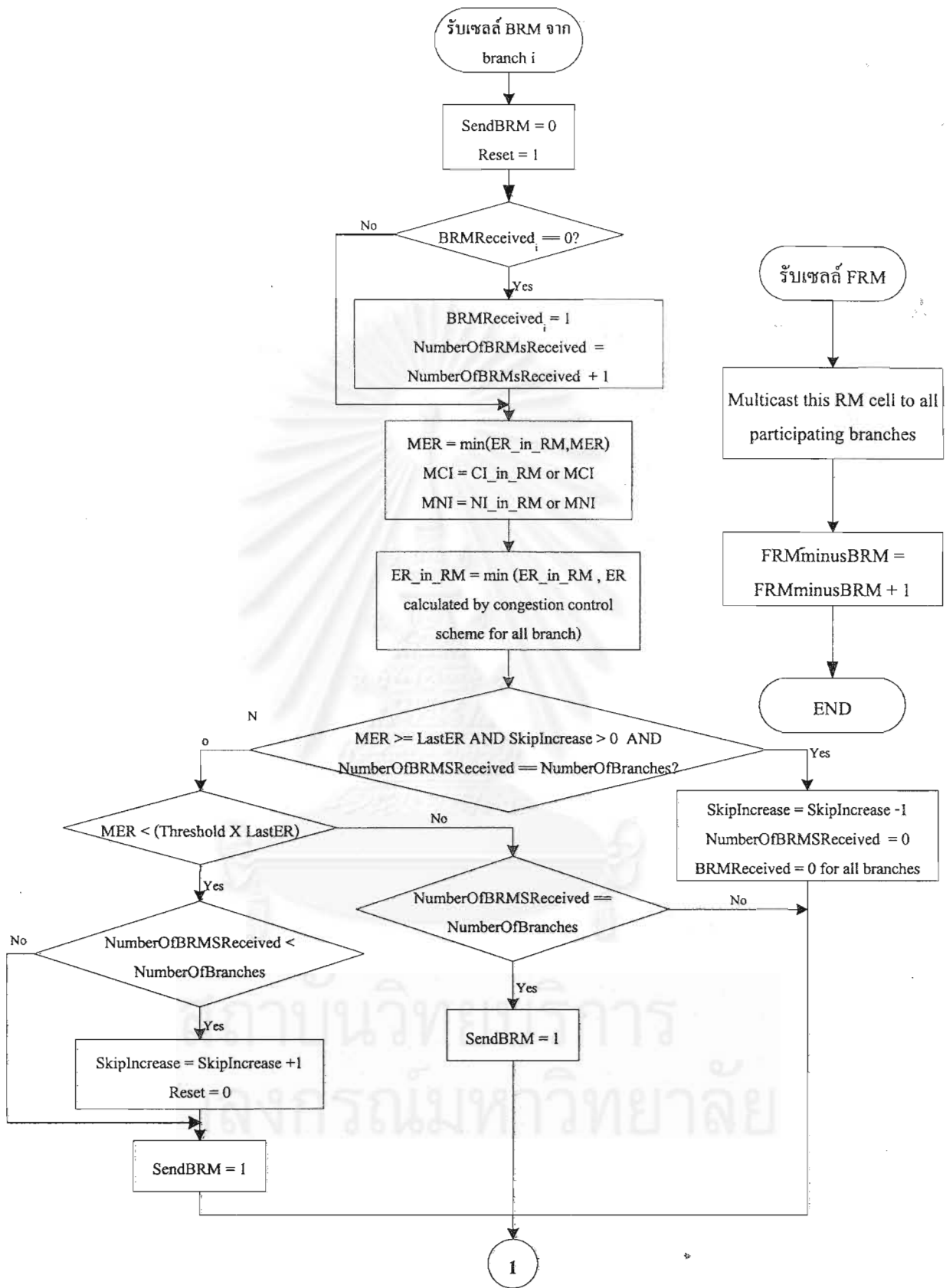
```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
  Let FRMminusBRM = FRMminusBRM + 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i :
  Let SendBRM = 0;
  Let Reset = 1;
  If not BRMReceived, then
    Let BRMReceived = 1;
    Let NumberOfBRMsReceived = NumberOfBRMsReceived + 1;
  Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI;
  Let MER = min (MER , minimum ER calculated by congestion control scheme for all branch)
  (** for faster overload detect **)
  If (MER ≥ LastER) and (SkipIncrease > 0) and (NumberOfBRMSReceived == NumberOfBranches) then
  {
    Let SkipIncrease = SkipIncrease - 1;
    Let NumberOfBRMSReceived = 0;
    Let BRMReceived = 0 for all branch;
  }
  else if (MER < (Threshold x LastER)) then      (***** overload is detected *****)
  {
    if (NumberOfBRMsReceived < NumberOfBranches) then
      Let SkipIncrease = SkipIncrease + 1;
      Let Reset = 0;
      Let SendBRM = 1;
    }
    else if (NumberOfBRMsReceived == NumberOfBranches) then
      Let SendBRM = 1;
  }
  If (SendBRM) then
  {
    Let ER = MER , CI = MCI , NI = MNI;
    Let ER = min (ER , ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let LastER = ER;
    If (Reset) then
      Let MER = PCR , MCI = 0 , MNI = 0;
      Let NumberOfBRMsReceived = 0;
      Let BRMReceived = 0 for all branches
      Let FRMminimumBRM = FRMminimumBRM - 1;
    }
  }
  else
    Discard the BRM Cell

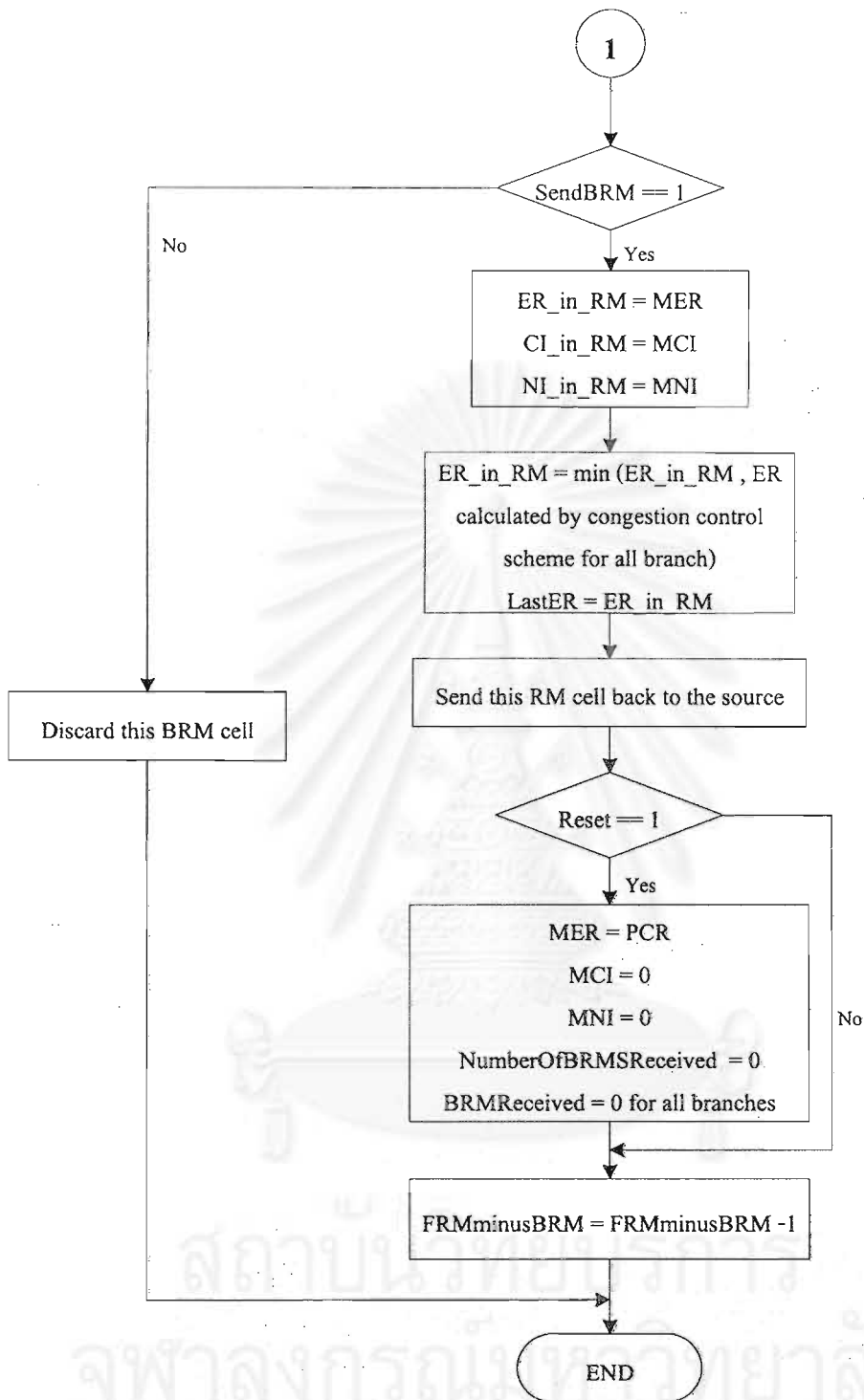
```

รูปที่ 3.13 ลำดับการทำงานของ Immediate Rate Calculation algorithm

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.14 แผนผังการทำงานของ Immediate rate calculation algorithm



รูปที่ 3.14 แผนผังการทำงานของ Immediate rate calculation algorithm (ต่อ)

3.4 ปัญหาการทำงานของวิธีควบคุมความคับคั่งร่วมกับวิธีรวบรวมข่าวสารความคับคั่ง

การศึกษาที่ผ่านมาทางด้านบริการประเภท ABR แบบหนึ่งจุดไปสู่หลายจุดที่ผ่านมานั้น เน้นการศึกษาสำหรับใช้งานกับวิธี ERICA ซึ่งยังไม่มีการศึกษาให้นำมาใช้กับวิธีการควบคุมความคับคั่งอื่น ๆ วิทยานิพนธ์ฉบับนี้จึงเสนอการนำวิธีรวบรวมข่าวสารความคับคั่งแบบต่าง ๆ มาใช้กับวิธี

FMMRA เพื่อศึกษาปัญหา Consolidation noise และความเร็วในการป้อนกลับ ซึ่งการเกิด consolidation noise ในบางสภาวะนั้นอาจมีสาเหตุมากจากการใช้งานร่วมกันระหว่าง ERICA และ วิธีรวบรวมข่าวสารความคับคั่งที่ไม่ดี

สำหรับวิธีการรวบรวมข่าวสารความคับคั่งใหม่ ๆ ที่มีอยู่ในปัจจุบันนี้เน้นที่รักษาความถูกต้องของการรวบรวมข่าวสารความคับคั่งจากเซลล์ BRM เท่านั้น ทำให้มีผลตอบสนองชั่วคราว และ consolidation delay ที่ช้ามาก วิธี Immediate rate calculation algorithm นั้นพัฒนาเทคนิคในการตรวจจับสภาวะโหลดเกินเพื่อเพิ่มความเร็วในการป้อนกลับในสภาวะโหลดเกิน อย่างไรก็ตามเทคนิคดังกล่าวเป็นการแก้ไขปัญหา consolidation delay ที่ปลายเหตุ ซึ่งสังเกตได้ว่า หากโครงข่ายอยู่ในสภาวะโหลดต่ำ วิธีการดังกล่าวยังคงมีผลตอบสนองที่ช้า ซึ่งทำให้เกิดปัญหาการใช้งานสายสัญญาณที่ไม่คุ้มค่าขึ้น รายงานวิจัยฉบับนี้เสนอสาเหตุของการเกิด consolidation noise และพยายามหาวิธีการแก้ไขปัญหาดังกล่าว โดยแนวทางการแก้ไขปัญหานั้นจะเน้นที่พัฒนาให้มีผลตอบสนองชั่วคราวที่รวดเร็ว เพื่อขจัด consolidation delay เช่นเดียวกับในวิธีการแรก ๆ หากแต่พัฒนาวิธีการใหม่ ๆ เพื่อเพิ่มความถูกต้องให้กับข่าวสารที่เก็บรวบรวมไว้ที่ branch point นั้นเอง

3.5 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 1

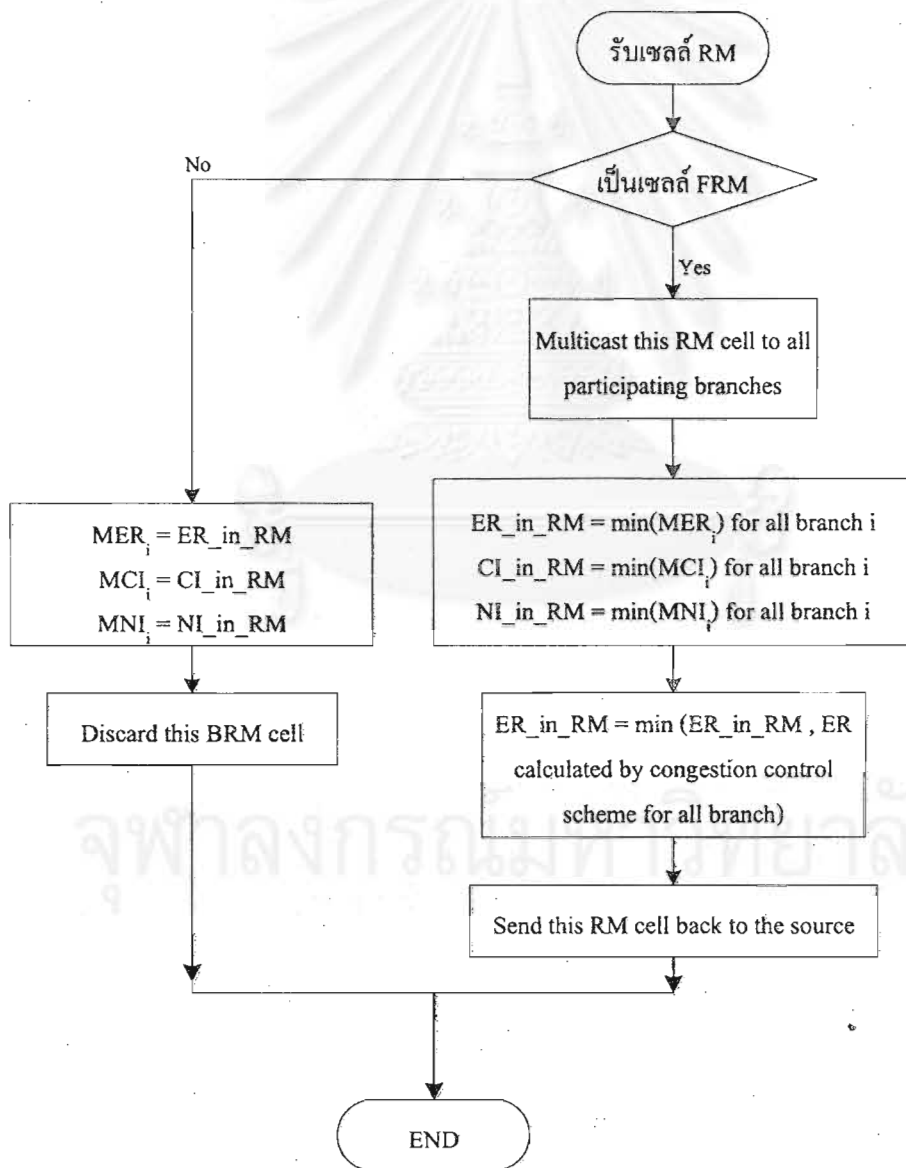
วิธีการรวบรวมความคับคั่งแบบต่าง ๆ ที่เสนอมานั้น โดยส่วนใหญ่แล้วจะพยายามพัฒนาเฉพาะทางด้านความเร็ว หรือ ความถูกต้องของข่าวสารความคับคั่ง อย่างไรก็ดีอย่างหนึ่งเท่านั้น วิธีการแรก ๆ นั้นเน้นความเร็วในการป้อนกลับในขณะที่ไม่สนใจความถูกต้องในการป้อนกลับ ส่วนวิธีใหม่ ๆ ที่พัฒนาขึ้นมาเน้นให้มีความถูกต้องของข้อมูลที่ใช้ในการป้อนกลับ แต่มีผลตอบสนองที่ช้า จึงได้มีการพัฒนาเพื่อเพิ่มความเร็วในการป้อนกลับสำหรับวิธีนี้ขึ้น โดยใช้เทคนิค overload detected ที่สามารถตรวจสอบสภาวะโหลดเกิน และทำการป้อนกลับทันที ดังที่ได้เสนอในวิธี Immediate rate calculation อย่างไรก็ตาม วิธีดังกล่าวยังมีข้อเสียในด้านผลตอบสนองชั่วคราวในสภาวะโหลดต่ำ (Underload condition) ทำให้โครงข่ายนั้นมีการใช้งานสายสัญญาณที่ไม่คุ้มค่า

วิธีการที่เสนอจึงเน้นที่ความเร็วในการป้อนกลับ ในขณะที่ปรับปรุงความถูกต้องของข่าวสารความคับคั่งที่เก็บรวบรวมไว้ที่ branch point เนื่องจากผลตอบสนองชั่วคราวของ Robert algorithm นั้นให้ผลตอบที่เร็วที่สุด วิธีการใหม่จึงนำหลักการของ Robert algorithm มา โดยนำเงื่อนไขในการป้อนกลับเซลล์ BRM มาใช้ branch point จะทำการส่งเซลล์ BRM ก็ต่อเมื่อได้รับ FRM แล้วเท่านั้น ส่วนความถูกต้องของข่าวสารที่เก็บไว้ นั้น ได้พัฒนาขึ้นมาโดยการเพิ่มจำนวนตัวแปรของข่าวสารที่รวบรวมไว้ให้มีลักษณะการเก็บ โดยเก็บแยกแต่ละ VC และ เก็บค่าจากทุก ๆ branch (per-VC and per-branch basis) ทำให้ข่าวสารที่เก็บรวบรวมไว้มีความถูกต้องและมีความข้อมูลที่ใหม่อยู่เสมอ โดยการตัดสินใจในการป้อนกลับค่า ER นั้นจะใช้ค่า ER ที่น้อยที่สุดระหว่างค่า ER ที่เก็บรวบรวมไว้จากทุก branch นั้นเอง ซึ่ง

หลักการนี้ถูกนำมาใช้สำหรับค่า CI และ NI เช่นกัน ลำดับขั้นตอนการทำงานและแผนผังการทำงานของวิธีการนี้จะเป็นดังรูปที่ 3.15 และ 3.16 ตามลำดับ

Upon the receipt of a forward RM(ER,CI,NI) cell :
 Multicast this RM cell to all participating branches;
 Let $ER = \min(MER_i)$, $CI = \min(MCI_i)$ and $NI = \min(MNI_i)$
 for all participating branches;
 Let $ER = \min(ER, ER \text{ calculated by congestion control scheme for all branches})$
 Send this RM cell back to the source;
 Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i:
 Let $MER_i = ER$, $MCI_i = CI$, $MNI_i = NI$;
 Discard the BRM Cell

รูปที่ 3.15 ลำดับการทำงานของ วิธีการที่เสนอแบบที่ 1



รูปที่ 3.16 แผนผังการทำงานของ วิธีการที่เสนอแบบที่ 1

3.6 วิธีรวบรวมความคับคั่งที่เสนอแบบที่ 2

ในวิธีรวบรวมความคับคั่งของ Robert TS และ RSS นั้นให้ผลตอบชั่วคราวที่รวดเร็วมก แต่ยังคงมี Consolidation noise อยู่ สำหรับวิธีรวบรวมความคับคั่งที่ตอบกลับเซลล์ BRM จากการตรวจสอบเซลล์ FRM นั้น หากจำนวนเซลล์ FRM ที่ branch point ได้รับมากกว่าเซลล์ BRM ที่ได้รับจาก branch หนึ่ง ๆ จะทำให้เกิดการตอบกลับเซลล์ BRM มากขึ้น สาเหตุหนึ่งของการเกิด Consolidation noise นั้นเนื่องจากการลบค่า MER ที่รวบรวมไว้จากทุก Branch หลังจากส่งเซลล์ BRM ไปแล้ว เมื่อมีการตอบกลับเซลล์ BRM มากขึ้นจะทำให้ข่าวสารที่รวบรวมไว้เกิดการผิดพลาดขึ้นเนื่องจากการลบค่า MER นั้นเอง

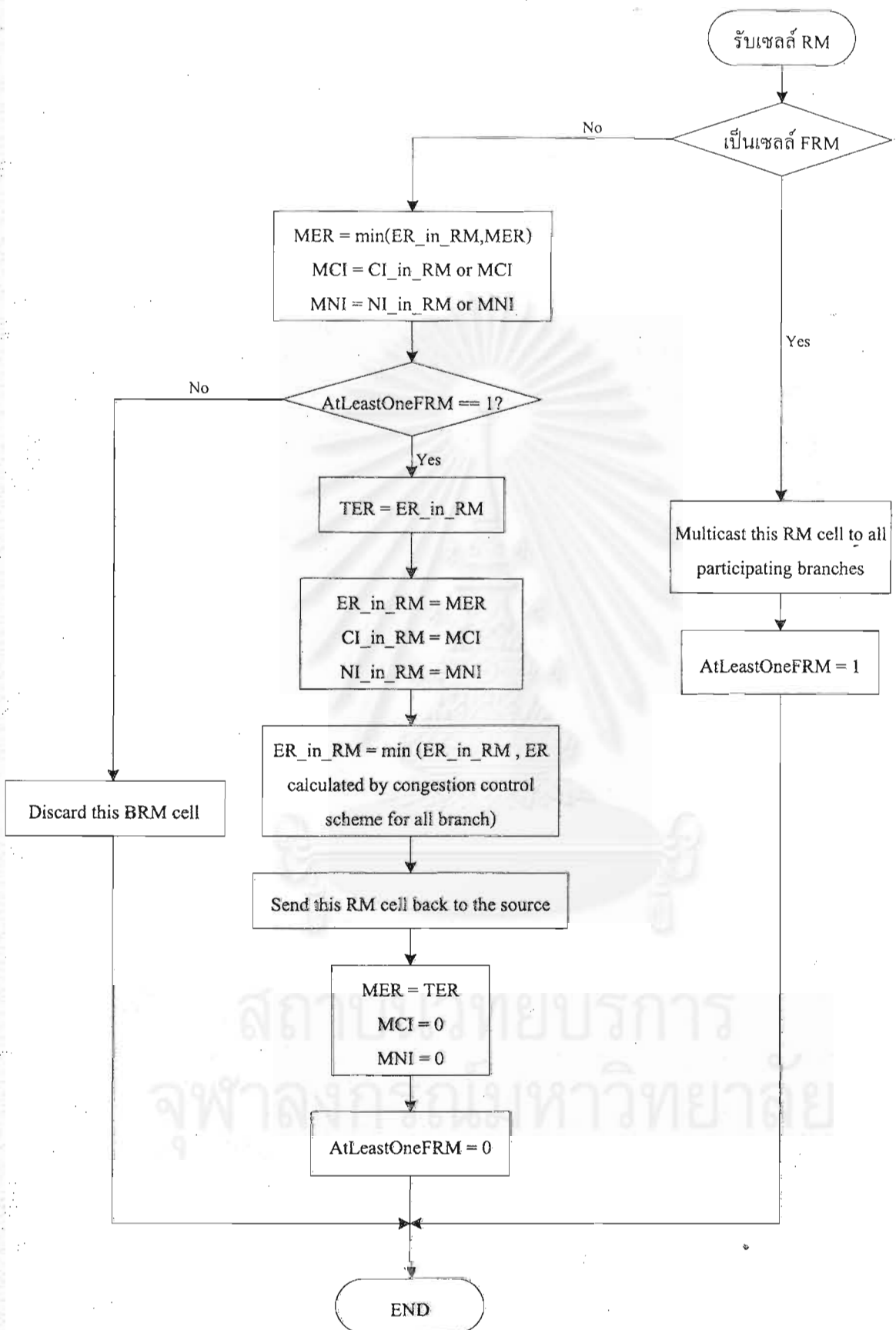
วิธีการใหม่ที่เสนออีกวิธีหนึ่งซึ่งพัฒนามาจาก RSS algorithm ที่เน้นให้มีความเร็วในการป้อนกลับ โดยที่มีหลักการทำงานพื้นฐานเหมือนกับ RSS algorithm หากแต่พัฒนาไม่ให้มีการลบค่า MER แต่นำค่า ER ในเซลล์ BRM ที่ได้รับก่อนจะตอบกลับเซลล์ RM มาเป็นค่า MER แทน ซึ่งถือเป็นการป้องกันเกิด consolidation noise อย่างหนึ่ง ตัวแปร TER นั้นถูกสร้างขึ้นมาเพื่อบันทึกค่า ER ที่ได้รับจากเซลล์ BRM ต่ำที่สุด และนำค่าดังกล่าวไปเป็นค่า MER ทันทีหลังจากป้อนกลับเซลล์ BRM กลับสู่แหล่งกำเนิดแล้ว ลำดับขั้นตอนการทำงานของวิธีการนี้จะเป็นดังนี้

```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
  Let AtLeastOneFRM = 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell
  Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
  If AtLeastOneFRM then
    Let TER = ER;
    Let ER = MER , CI = MCI , NI = MNI;
    Let ER = min (ER , ER calculated by congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let MER = TER , MCI = 0 , MNI = 0;
    Let AtLeastOneFRM = 0;
  else
    Discard the BRM Cell
  
```

รูปที่ 3.17 ลำดับการทำงานของ วิธีการที่เสนอแบบที่ 2

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.18 แผนผังการทำงานของ วิธีการที่เสนอแบบที่ 2

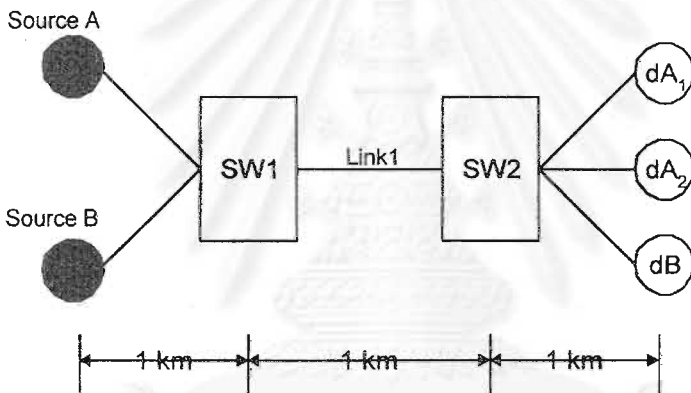
บทที่ 4

ผลการจำลองแบบและวิเคราะห์

4.1 ผลการจำลองสำหรับ One hop configuration

4.1.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ

โครงข่ายลักษณะนี้เป็นโครงข่ายรูปแบบที่ง่ายที่สุด และเป็นพื้นฐานในการทดสอบวิธีการใหม่ ๆ โครงข่ายประกอบด้วยแหล่งกำเนิด 2 แหล่ง ได้แก่ แหล่งกำเนิด A ซึ่งเป็นแหล่งกำเนิดข้อมูลสำหรับการส่งจากหนึ่งจุดไปสู่หลายจุด โดยจะส่งข้อมูลให้แก่ปลายทาง dA_1 และ dA_2 ตามลำดับ และแหล่งกำเนิด B ซึ่งส่งข้อมูลให้แก่ปลายทาง dB โดยทำหน้าที่เป็นแหล่งกำเนิดสำหรับการส่งจากหนึ่งจุดไปสู่หนึ่งจุด รูปแบบโครงข่ายจะเป็นดังรูปที่ 4.1



รูปที่ 4.1 One hop configuration

ค่าตัวแปรต่าง ๆ ที่ใช้ในรูปแบบโครงข่ายนี้ได้แก่

Source A : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

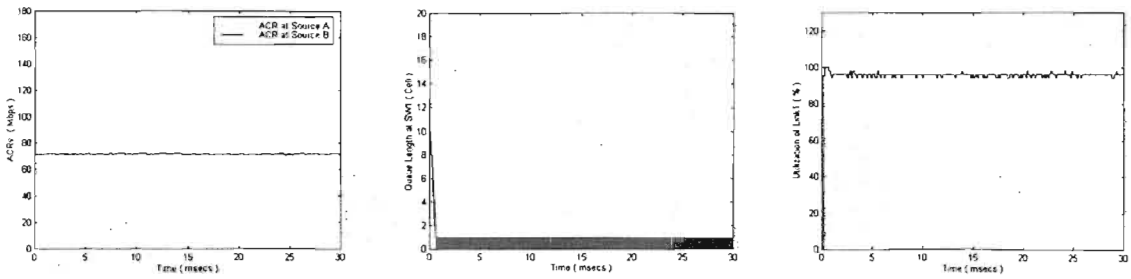
Source B : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

SW1 : Maximum queue length = 1,000

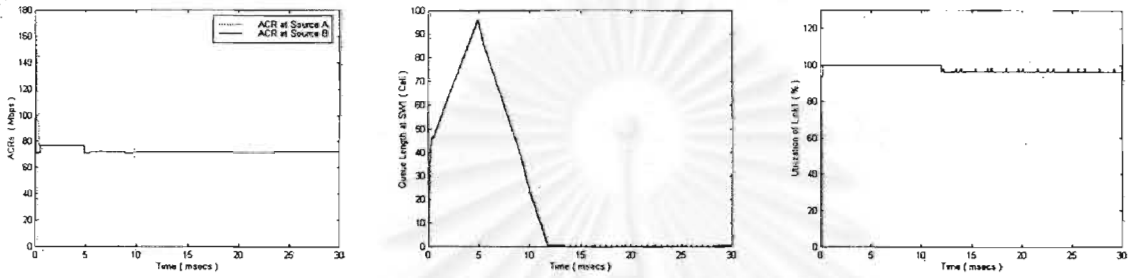
SW2 : Maximum queue length = 100

4.1.2 สมรรถนะของวิธี ERICA

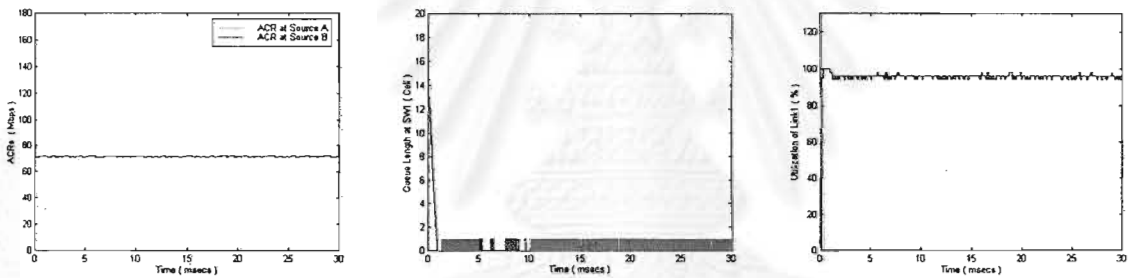
เมื่อใช้ ERICA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.2 ถึง 4.8



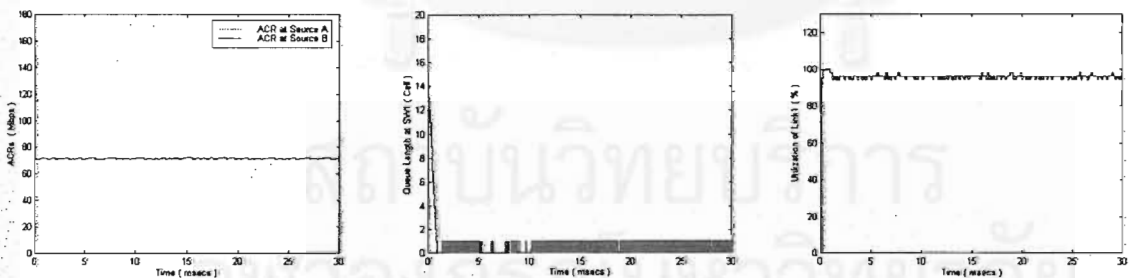
รูปที่ 4.2 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Robert Algorithm สำหรับ one hop configuration



รูปที่ 4.3 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ TS Algorithm สำหรับ one hop configuration

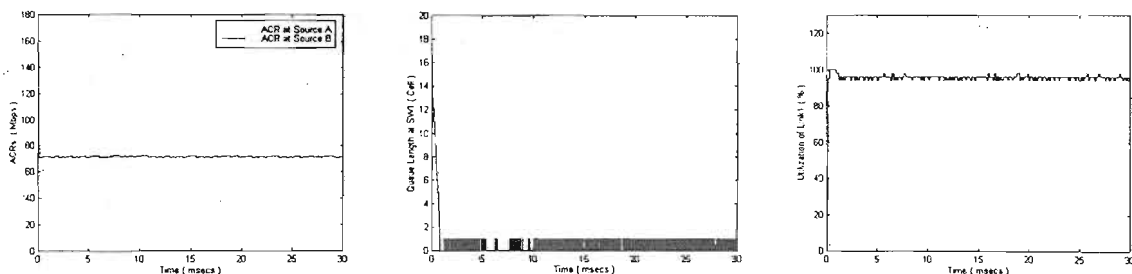


รูปที่ 4.4 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ RSS Algorithm สำหรับ one hop configuration

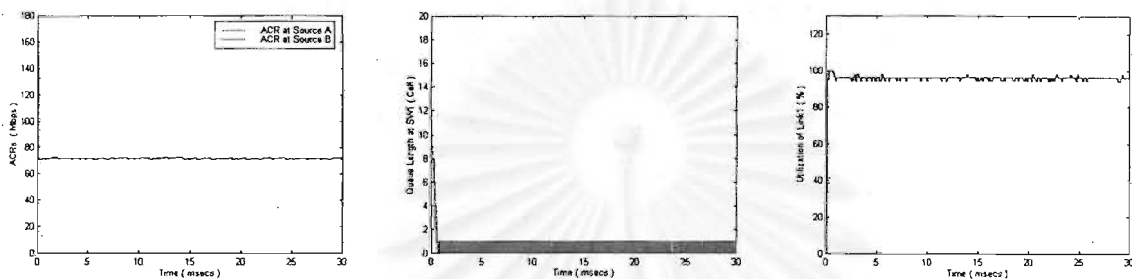


รูปที่ 4.5 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Wait-for-all Algorithm สำหรับ one hop configuration

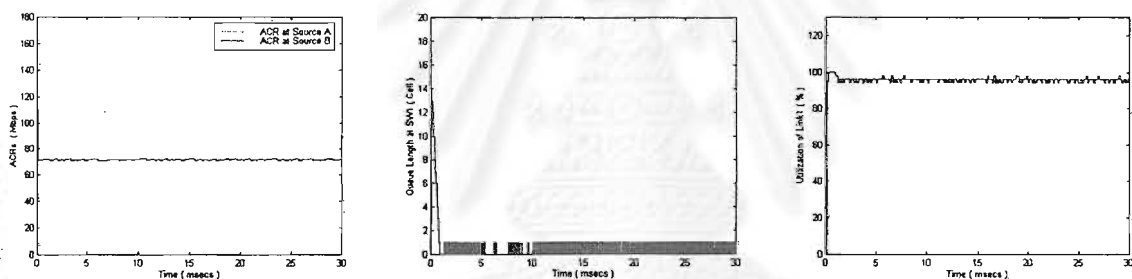




รูปที่ 4.6 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ one hop configuration



รูปที่ 4.7 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ one hop configuration



รูปที่ 4.8 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ one hop configuration

จุดประสงค์หลักของการทดสอบกับรูปแบบโครงข่ายนี้ เพื่อแสดงถึงความสามารถของวิธีรวบรวมข่าวสารความคับคั่งแต่ละวิธีว่าสามารถใช้งานวิธีการ ERICA และ FMMRA บนโครงข่าย ATM แบบง่ายที่สุดโดยไม่ก่อให้เกิดปัญหาใดๆ

จากรูปที่ 4.2 สังเกตได้ว่าเมื่อใช้ ERICA กับ Robert algorithm จะเห็นได้ว่าค่า ACR ที่แหล่งกำเนิดทั้งสองนั้นมีค่าลดลงจากค่า ICR คือ 149.76 Mbps ลงมาสู่ fairshare อย่างรวดเร็ว ทั้งนี้เนื่องจากโครงข่ายมีขนาดเล็ก และมี propagation delay น้อย ทำให้เวลาที่ใช้ในการป้อนกลับมีค่าน้อย แหล่งกำเนิดสามารถปรับอัตราส่งให้เหมาะสมที่ fairshare ได้อย่างรวดเร็ว โดยค่า fairshare ของแต่ละแหล่งกำเนิดสำหรับรูปแบบโครงข่ายนี้คือ $149.76 / 2 = 74.88$ Mbps อย่างไรก็ตาม ERICA นั้นมีการเผื่อการใช้งานสายสัญญาณไว้ที่ค่า target utilization ซึ่งในรูปแบบโครงข่ายนี้ตั้งค่าไว้ที่ 95% ERICA จะนำค่าดังกล่าวมาทำการคำนวณแบนวิดธ์ที่ใช้ได้จริงก่อนการคำนวณ fairshare และทำให้ค่า fairshare สำหรับแต่ละแหล่งกำเนิดเป็น $149.76 \times 0.95 / 2 = 71.14$ Mbps และเมื่อโครงข่ายอยู่ในสภาวะอยู่ตัวค่าอัตราส่งที่แหล่งกำเนิดมีค่างที่ค่าประมาณ fairshare นั้นเอง

สำหรับขนาดหน่วยความจำสูงสุดที่ SW1 นั้นมีค่าเพียง 10 เซลล์ ในช่วงแรกนั้น โคร่งข่ายจะอยู่ในสภาวะเกิดความคับคั่งขึ้นเนื่องจาก link1 ไม่สามารถรองรับข้อมูลที่ส่งมาจากแหล่งกำเนิดทั้งสองรวมกันได้ ทั้งนี้เนื่องจากแหล่งกำเนิดทั้งสองนั้นมีค่า ICR มากเกินไปนั่นเอง อย่างไรก็ตามการที่ขนาดหน่วยความจำสูงสุดที่ SW1 มีค่าน้อยนั้นเกิดจากการป้อนกลับที่รวดเร็ว และแหล่งกำเนิดสามารถปรับอัตราส่งได้อย่างรวดเร็ว ทำให้โคร่งข่ายหลุดพ้นจากสภาวะความคับคั่งได้รวดเร็ว

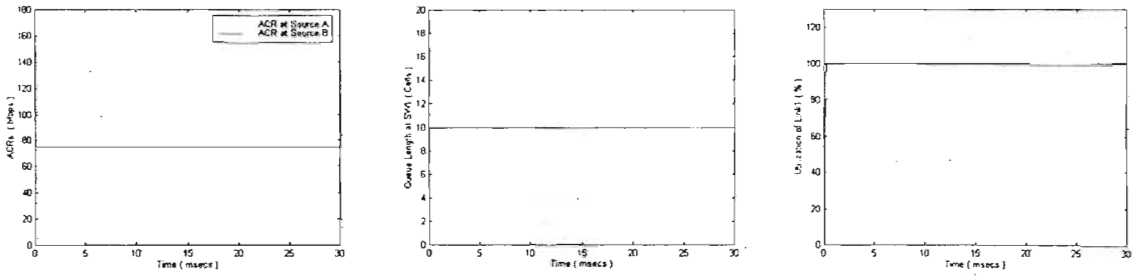
Robert Algorithm นั้นมีเวลาในการป้อนกลับรวดเร็วที่สุดในบรรดาวิธีรวบรวมข่าวสารความคับคั่ง ด้วยความสามารถในการป้อนกลับข่าวสารในเซลล์ RM ทันทีที่ได้รับเซลล์ FRM ทำให้เมื่อ SW1 ได้รับเซลล์ FRM จาก Source A จะทำการป้อนกลับข่าวสารความคับคั่งทันที และแหล่งกำเนิด A นั้นจะปรับอัตราส่งได้อย่างรวดเร็ว ทำให้โคร่งข่ายออกจากสภาวะคับคั่งได้รวดเร็ว ดังนั้นขนาดหน่วยความจำสูงสุดจึงมีค่าน้อยที่สุดด้วย ส่วนค่าการใช้งานสายสัญญาณนั้นจะมีค่าเป็น 1 ในช่วงแรก เนื่องจากในช่วงแรกแหล่งกำเนิดทั้ง 2 นั้นส่งข้อมูลที่อัตราส่งที่รวมกันสูงกว่าที่สายสัญญาณจะรองรับได้ ทำให้มีการใช้งานสายสัญญาณได้เต็มที่ โดยส่วนที่สายสัญญาณไม่สามารถรองรับได้นั้นจะนำไปเก็บไว้ในหน่วยความจำที่สวิตช์ และเมื่อสวิตช์ไม่เกิดความคับคั่ง เพราะกลไกการควบคุมความคับคั่ง ทำให้การใช้งานสายสัญญาณตกลงมาที่ประมาณ 95% ซึ่งเป็นค่า target utilization ที่ ERICA ตั้งไว้นั่นเอง

เมื่อใช้ ERICA กับ RSS algorithm, Wait-for-all algorithm, Immediate rate calculation algorithm และ วิธีการที่เสนอแบบที่ 2 จะได้ผลตามรูปที่ 4.4, 4.5, 4.6 และ 4.8 ตามลำดับ จะเห็นว่าผลการจำลองที่ได้นั้นคล้ายคลึงกับเมื่อใช้ Robert algorithm เพียงแต่ขนาดหน่วยความจำสูงสุดที่ใช้มีค่า 14 เซลล์ เนื่องจากกลไกการป้อนกลับข่าวสารความคับคั่งของวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ ข้างต้น มีการทำงานคล้ายกันในกรณีนี้ โดย SW1 จำเป็นต้องรอให้มี BRM เข้ามาก่อนจึงสามารถป้อนกลับข่าวสารความคับคั่งได้ ซึ่งทำให้เวลาที่ใช้ในการป้อนกลับครั้งแรกนั้นช้ากว่า Robert algorithm โคร่งข่ายจะอยู่ในสภาวะคับคั่งนานกว่า และทำให้ขนาดหน่วยความจำสูงสุดที่ใช้มีค่ามากกว่า อย่างไรก็ตามจะเห็นว่าขนาดหน่วยความจำสูงสุดที่ใช้มีค่าน้อยมากเพียง 14 เซลล์ ซึ่งถือว่าเป็นค่าที่ยอมรับได้

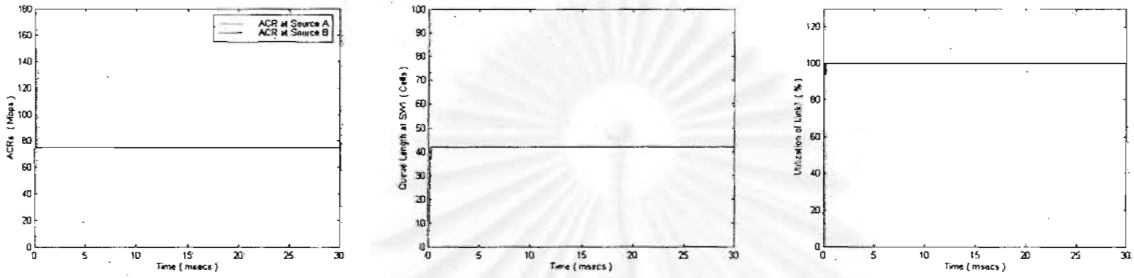
สำหรับวิธีการที่เสนอแบบที่ 1 นั้นจะได้ผลตามรูปที่ 4.7 ซึ่งจะเห็นว่าผลการจำลองนั้นคล้ายคลึงกับ Robert algorithm มาก เนื่องจากวิธีนี้ใช้เงื่อนไขในการตอบกลับข่าวสารความคับคั่งแบบเดียวกับ Robert algorithm ทำให้ผลการทดสอบสำหรับ โคร่งข่ายง่าย ๆ เช่นนี้มีผลลัพธ์ที่ไม่แตกต่างกัน

4.1.3 สมรรถนะของวิธี FMMRA

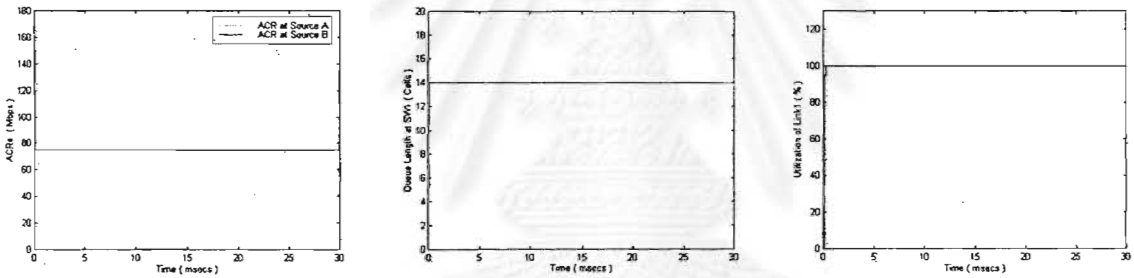
เมื่อใช้ FMMRA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.9 ถึง 4.15



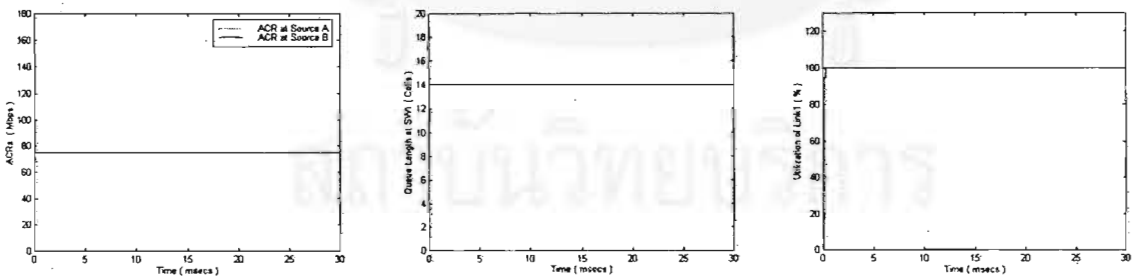
รูปที่ 4.9 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Robert Algorithm สำหรับ one hop configuration



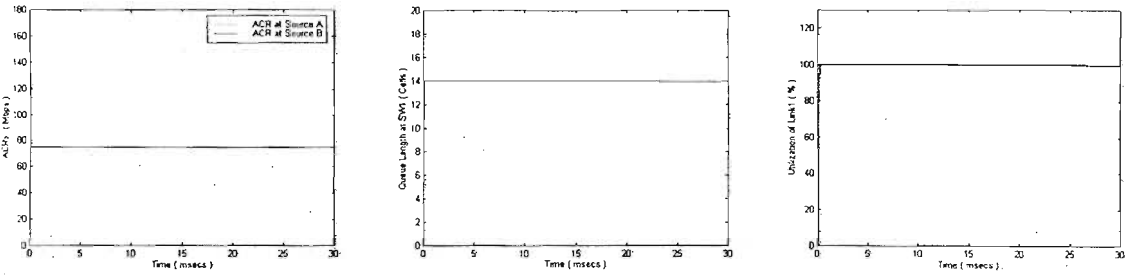
รูปที่ 4.10 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ TS Algorithm สำหรับ one hop configuration



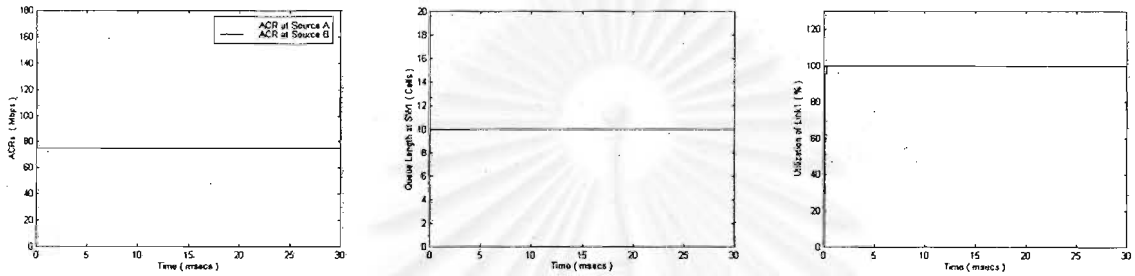
รูปที่ 4.11 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ RSS Algorithm สำหรับ one hop configuration



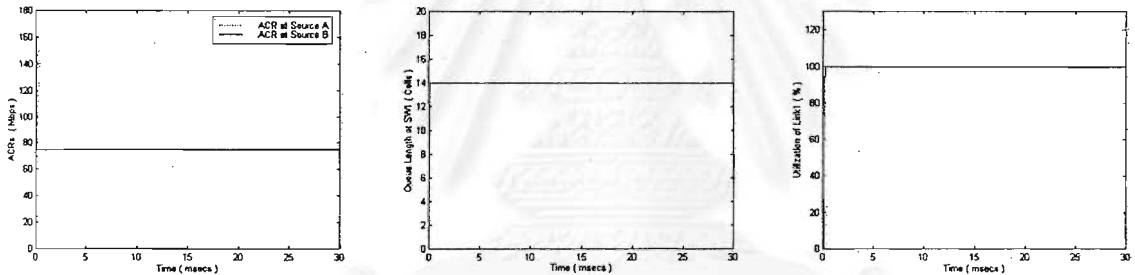
รูปที่ 4.12 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Wait-for-all Algorithm สำหรับ one hop configuration



รูปที่ 4.13 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ one hop configuration



รูปที่ 4.14 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ one hop configuration



รูปที่ 4.15 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ one hop configuration

จากรูปที่ 4.9 และ 4.15 สังเกตได้ว่าเมื่อใช้ FMMRA กับ Robert algorithm และ วิธีการที่เสนอแบบที่ 1 ซึ่งให้ผลการจำลองที่เหมือนกันในรูปแบบโครงข่ายนี้ จะเห็นได้ว่าค่า ACR ที่แหล่งกำเนิดทั้งสองนั้นมีค่าลดลงจากค่า ICR คือ 149.76 Mbps ลงมาสู่ fairshare อย่างรวดเร็วเช่นเดียวกับกรณีที่ใช้ ERICA ทั้งนี้เนื่องจากโครงข่ายมีขนาดเล็ก และมี propagation delay น้อย โดยค่า fairshare ของแต่ละแหล่งกำเนิดสำหรับรูปแบบโครงข่ายนี้คือ $149.76 / 2 = 74.88$ Mbps และเนื่องจาก FMMRA นั้นไม่มีการตั้งค่าการใช้งานสายสัญญาณที่ต้องการไว้ ทำให้ค่า fairshare นั้นอยู่ที่ 74.88 Mbps และ SW1 นำค่าดังกล่าวมาใช้ในการป้อนกลับไปสู่แหล่งกำเนิด ดังนั้นแหล่งกำเนิดจึงส่งข้อมูลด้วยอัตราดังกล่าวตลอดเวลา ซึ่งหากพิจารณาที่ SW1 จะเห็นว่าปริมาณทราฟฟิกขาเข้ารวมจากทั้งสองแหล่งกำเนิด และปริมาณความจุขาออกไป link1 นั้นมีค่าเท่ากัน ทำให้มีการใช้งานสายสัญญาณ link1 ได้เต็มที่ตามรูปที่ 4.9 ในขณะที่ขนาดหน่วยความจำที่สวิตช์นั้นมีค่าเพิ่มขึ้นในช่วงแรก เนื่องจากสถานะคับคั่งในช่วงแรกของ SW1 และมีค่าคงที่ที่ 10 เซลล์ตลอด ทั้งนี้เนื่องจากปริมาณทราฟฟิกขาเข้าและขาออกนั้นมีปริมาณเท่ากัน ส่วนขนาดหน่วยความจำสูงสุดนั้นมีค่าเป็น 10 เซลล์ เช่นเดียวกับในกรณี ERICA

สำหรับ TS algorithm จะได้ผลการจำลองตามรูปที่ 4.10 สังเกตได้ว่าค่าขนาดหน่วยความจำสูงสุดที่ SW1 นั้นมีค่าสูงกว่าเมื่อใช้ Robert algorithm ทั้งนี้เนื่องจากในช่วงแรกนั้น โครงข่ายอยู่ในสถานะคับคั่ง และ TS algorithm นั้นมีเงื่อนไขในการตอบกลับข่าวสารความคับคั่งในเซลล์ RM คือ ต้องได้รับเซลล์ FRM หลังจากได้รับเซลล์ BRM แล้ว ซึ่งทำให้ SW1 ไม่สามารถตอบกลับข่าวสารความคับคั่งได้ในขณะที่ได้รับเซลล์ FRM ตัวแรก และ SW1 จำเป็นต้องรอเซลล์ FMR ตัวที่สองจากแหล่งกำเนิด A ทำให้โครงข่ายอยู่ในสถานะคับคั่งนานขึ้น ส่งผลให้ขนาดหน่วยความจำสูงสุดมีค่ามากขึ้นไปด้วย อย่างไรก็ตามขนาดหน่วยความจำยังคงมีค่าคงที่ที่ประมาณ 40 เซลล์

เมื่อใช้ FMMRA กับ RSS algorithm, Wait-for-all algorithm, Immediate rate calculation algorithm และ วิธีการที่เสนอแบบที่ 2 จะได้ผลตามรูปที่ 4.11, 4.12, 4.13 และ 4.15 ตามลำดับ จะเห็นว่าได้ผลคล้ายคลึงกับเมื่อใช้ Robert algorithm เพียงแต่ขนาดหน่วยความจำสูงสุดที่ใช้มีค่า 14 เซลล์ เนื่องจากกลไกการป้อนกลับข่าวสารความคับคั่งของวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ ข้างต้น มีการทำงานคล้ายกันในกรณีนี้

4.1.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

ทั้ง FMMRA และ ERICA นั้นสามารถปรับอัตราส่งที่แหล่งกำเนิดไปสู่อัตราส่งที่โครงข่ายต้องการได้อย่างยุติธรรม หากแต่ ERICA นั้นมีการตั้งค่าการใช้งานสายสัญญาณที่ต้องการไว้ ทำให้สามารถลดระดับหน่วยความจำที่สวิตช์ได้ แต่ก็เป็นการทำให้การใช้งานสายสัญญาณ link1 นั้นเป็นไปได้ไม่เต็มที่ 100% ในขณะที่ FMMRA นั้นไม่มีการตั้งค่าดังกล่าว ทำให้ปริมาณกราฟฟิกขาเข้าและความจุขาออกมีค่าเท่ากัน ส่งผลให้ขนาดของหน่วยความจำที่สวิตช์นั้นมีค่าคงที่ ซึ่งเป็นผลเสีย ในกรณีที่มีการเชื่อมต่อใหม่อื่น ๆ ที่เข้ามาสู่โครงข่าย จะทำให้โครงข่ายเข้าสู่สถานะคับคั่งอีก ในขณะที่ไม่สามารถลดระดับหน่วยความจำได้

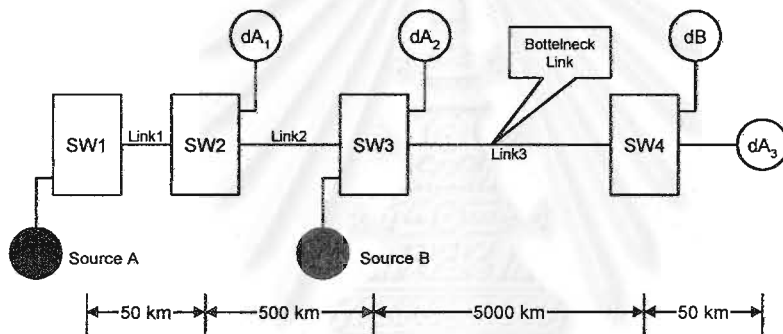
4.1.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ

ในรูปแบบโครงข่ายนี้วิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ นั้นให้ผลที่ไม่แตกต่างกันมากนัก ทั้งนี้เนื่องจากเป็นโครงข่ายที่ง่าย และมีขนาดเล็ก ทำให้ผลการจำลองต่าง ๆ มีลักษณะคล้าย ๆ กัน มีเพียง TS algorithm เท่านั้นที่ให้ผลที่แย่ที่สุด โดยมีขนาดหน่วยความจำที่สวิตช์สูงที่สุด

4.2 ผลการจำลองสำหรับ Chain configuration

4.2.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ

รูปแบบโครงข่ายนี้เป็นโครงข่ายที่เหมาะสมสำหรับการทดสอบปัญหา Consolidation noise ได้เป็นอย่างดี ประกอบด้วย VC 2 ชุด ได้แก่ VC1 ของแหล่งกำเนิด A ซึ่งเป็น point-to-multipoint connection และ VC2 ของแหล่งกำเนิด B ซึ่งเป็น point-to-point connection โดยกราฟฟิกจะถูกรวมกันที่สวิตช์ 3 ดังนั้นจะมีจุดที่อาจเกิดปัญหาความคับคั่งได้เพียงจุดเดียวคือ link3 เมื่อระบบเข้าสู่สถานะอยู่ตัวค่า ACR ที่เหมาะสมสำหรับทั้งสอง VC ควรจะมีค่าประมาณเท่ากับครึ่งหนึ่งของความจุของ link3 โดย link3 มีความจุ 149.76 Mbps จะได้ค่า fairshare เป็น $149.76/2 = 74.88$ Mbps เพื่อให้การใช้งานสายสัญญาณมีประสิทธิภาพสูงสุดและมีความเท่าเทียมกันโดยไม่เกิดความคับคั่งขึ้น รูปแบบโครงข่ายจะเป็นดังรูปที่ 4.16



รูปที่ 4.16 Chain configuration

ค่าตัวแปรต่าง ๆ ที่ใช้ในรูปแบบโครงข่ายนี้ได้แก่

Source A : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

Source B : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

SW1 : Maximum queue length = 40,000

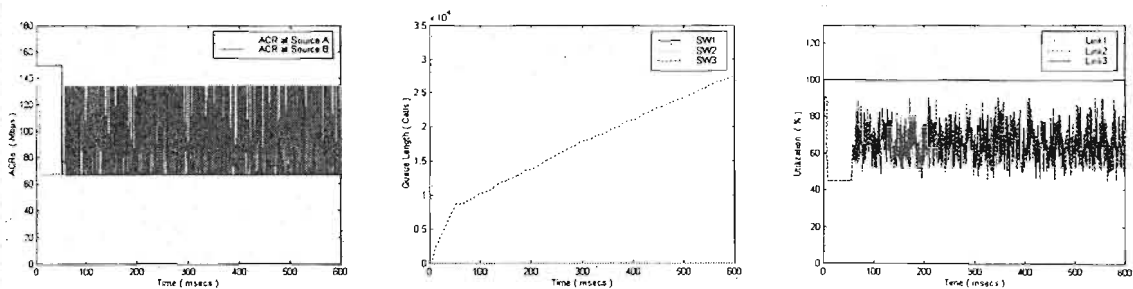
SW2 : Maximum queue length = 40,000

SW3 : Maximum queue length = 40,000

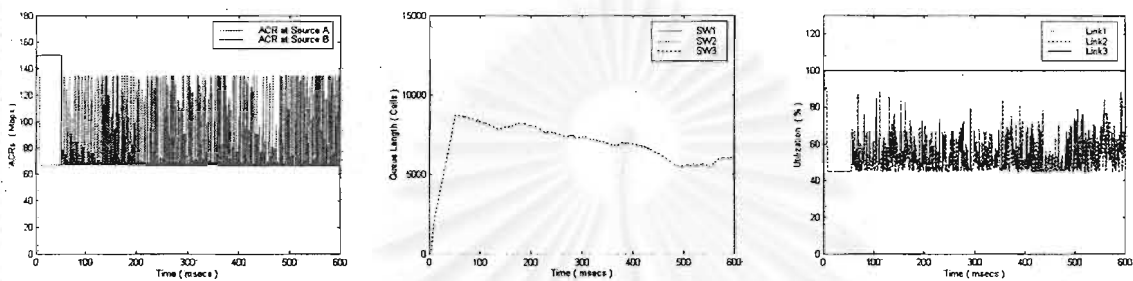
SW4 : Maximum queue length = 40,000

4.2.2 สมรรถนะของวิธี ERICA

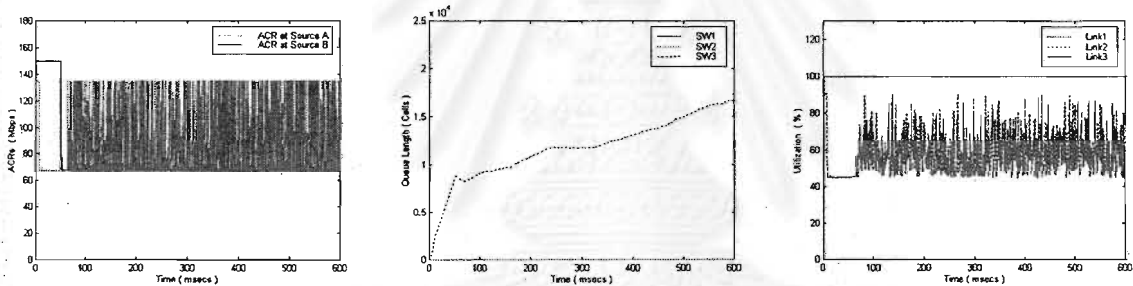
เมื่อใช้ ERICA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.17 ถึง 4.23



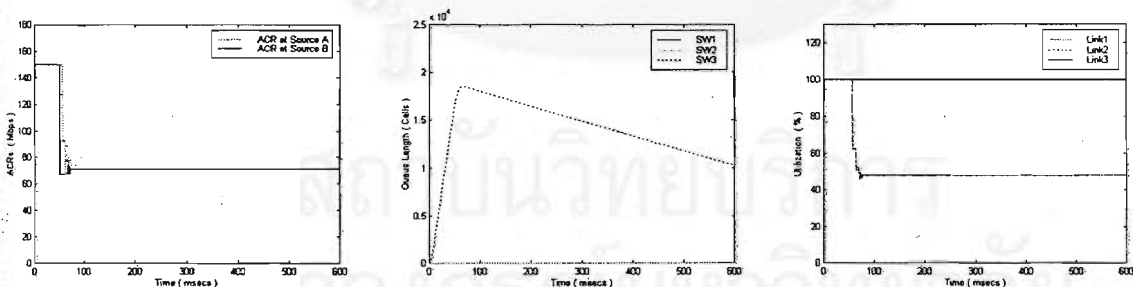
รูปที่ 4.17 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Robert Algorithm สำหรับ Chain configuration



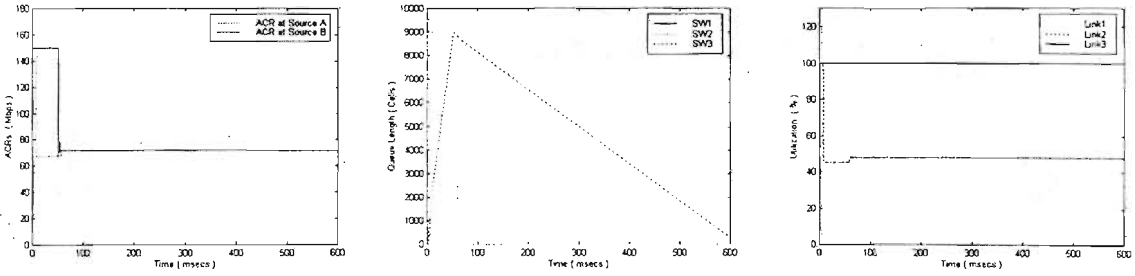
รูปที่ 4.18 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ TS Algorithm สำหรับ Chain configuration



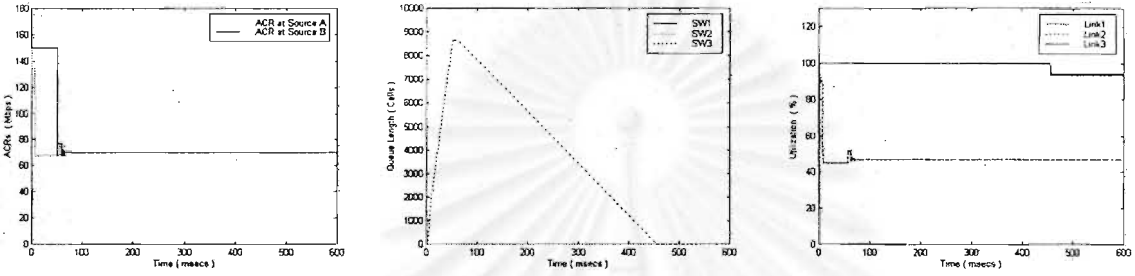
รูปที่ 4.19 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ RSS Algorithm สำหรับ Chain configuration



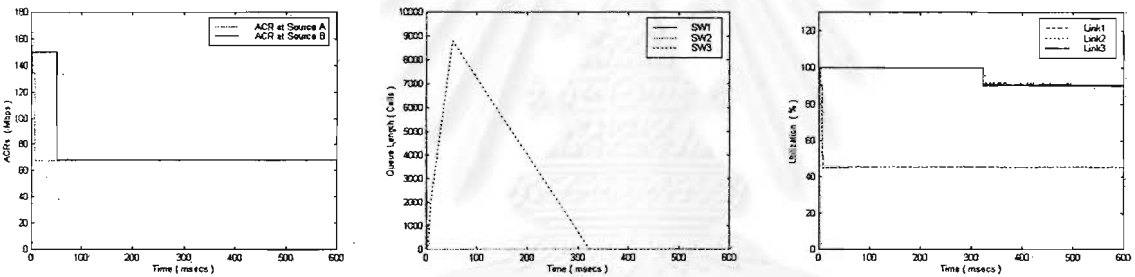
รูปที่ 4.20 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Wait-for-all Algorithm สำหรับ Chain configuration



รูปที่ 4.21 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Chain configuration



รูปที่ 4.22 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Chain configuration



รูปที่ 4.23 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Chain configuration

ค่า fairshare สำหรับแต่ละแหล่งกำเนิดที่ควรจะเป็นจะมีค่าเท่ากัน 74.88 Mbps แต่เนื่องจาก ERICA มีการกำหนดค่า target utilization เป็น 0.9 ทำให้ค่าอัตราส่งที่เท่าเทียมกันจะมีค่าเป็น 67.39 Mbps (0.9×74.88 Mbps) แต่อย่างไรก็ตามเนื่องจาก ERICA มีการทำงานโดยใช้ค่า z ที่อยู่ระหว่าง 1-1.1 ดังนั้นอัตราส่งที่เกิดขึ้นจริงจะมีค่าสูงกว่าค่าอัตราส่งเท่าเทียมกันเล็กน้อย

เมื่อพิจารณาค่า ACR ในรูปที่ 4.17 จะเห็นว่าผลตอบของ Robert algorithm นั้นมีผลตอบชั่วคราวที่รวดเร็วมาก ดังสังเกตได้จากค่า ACR ของแหล่งกำเนิด A ซึ่งเป็นการเชื่อมต่อแบบหนึ่งจุดไปสู่หลายจุด ในช่วงแรกค่า ACR มีค่า 149.76 Mbps ตามค่า ICR ของแหล่งกำเนิดนั้น เมื่อสวิตช์ 1 ได้รับเซลล์ FRM แล้ว ก็ทำการตอบกลับข่าวสารความคับคั่งทันที ด้วยการส่งกลับเซลล์ BRM ที่มีค่า ER ประมาณ 134.78 Mbps (149.76 Mbps คูณกับ target utilization) และ เมื่อแหล่งกำเนิดได้รับข่าวสารนี้จึงปรับค่า ACR ให้ลดลงมาเป็นประมาณ 134.78 Mbps ขั้นตอนนี้ใช้เวลารวดเร็วมากเนื่องจากระยะระหว่างแหล่งกำเนิด A กับ SW1 นั้นน้อยมาก และเมื่อ SW3 ได้รับเซลล์ FRM ตัวแรก ด้วยเงื่อนไขของ Robert algorithm ทำให้ SW3 ต้องตอบกลับเซลล์ BRM ทันที ซึ่งค่า ER ที่ส่งไปนั้นจะมีค่าประมาณ 67.39

Mbps โดย SW3 เป็นสวิตช์ที่มีทราฟฟิกมากที่สุดและเป็นสวิตช์ที่เกิดความคับคั่งขึ้นและจะเป็นสวิตช์ที่บ่งบอกถึงอัตราส่งที่แท้จริงที่แหล่งกำเนิดควรส่ง เมื่อแหล่งกำเนิด A ได้รับเซลล์ BRM ดังกล่าวก็จะทำการปรับลด ACR ลงมา โดยใช้เวลาในการป้อนกลับจากแหล่งกำเนิด A ไปสู่ SW3 และจาก SW3 กลับสู่ แหล่งกำเนิด A นั้นสอดคล้องกับเวลาที่ได้จากผลการจำลองคือประมาณ 6 ms โดยระยะทางที่เซลล์ RM ใช้ในเส้นทางไปและกลับจากแหล่งกำเนิดไปสู่ SW3 และจาก SW3 กลับไปสู่แหล่งกำเนิดรวมกันทั้งสิ้น 1,200 km ซึ่งคิดเป็นเวลาประมาณ 6 ms ส่วนแหล่งกำเนิด S4 นั้นมีผลตอบชั่วคราวที่ช้าเนื่องจากการเชื่อมต่อแบบหนึ่งจุดไปสู่หลายจุดทำให้ไม่จำเป็นต้องใช้วิธีรวบรวมข่าวสารความคับคั่งดังนั้นแหล่งกำเนิดจำเป็นต้องรอให้เซลล์ RM เดินทางไปสู่ปลายทาง และ เดินทางกลับมาจนถึงแหล่งกำเนิดเสียก่อนจึงสามารถปรับค่า ACR ได้

จากนั้นเมื่อแหล่งกำเนิด B เริ่มได้รับเซลล์ BRM ตัวแรกก็จะทำการปรับค่า ACR ให้ลดลง จะมีผลทำให้ค่า ACR ของ Source A เกิดการแกว่งตัวอย่างรุนแรงตามรูปที่ 4.17 โดยเรียกว่าเกิด consolidation noise ขึ้น โดย noise ที่เกิดขึ้นนั้นเกิดจากการรวบรวมข่าวสารความคับคั่งที่ไม่ดีพอ ข่าวสารความคับคั่งนั้นจะเก็บรวบรวมไว้ในตัวแปร MER และจะถูกนำมาเป็นค่า ER ใน BRM ทันทีเมื่อมีการป้อนกลับข่าวสารความคับคั่ง และค่า MER จะถูกลบล้างให้กลายเป็นค่า ER ที่อยู่ใน เซลล์ FMR นั้นเอง ซึ่งโดยปกติแล้วค่าดังกล่าวจะมีค่าเป็น PCR ในระหว่างนั้นหากมีข่าวสารจาก branch ใด branch หนึ่งก็จะนำมาเปรียบเทียบกับค่า MER โดยเลือกเก็บค่าน้อยที่สุดไว้ใน MER หากข่าวสารที่ถูกต้องจาก SW3 ยังเดินทางมาไม่ถึง SW1 และ SW2 ก่อนถึงกำหนดการป้อนกลับข่าวสารครั้งต่อไป ก็จะทำให้ SW1 และ SW2 นั้นจำเป็นต้องนำข่าวสารใน MER ไปป้อนกลับใน เซลล์ BRM ตัวใหม่ ซึ่งค่าดังกล่าวยังมีค่าสูงอยู่ทำให้แหล่งกำเนิด A นั้นปรับอัตราส่งเพิ่มขึ้นเกือบเท่ากับค่า PCR และ หากข่าวสารที่ถูกต้องจาก SW3 เดินทางมาถึง SW1 และ SW2 ก็จะทำให้ข่าวสารที่จะป้อนกลับครั้งต่อไปนั้นมีความถูกต้อง ส่งผลให้แหล่งกำเนิด A นั้นจะปรับอัตราส่งไปมาจนเกิดการแกว่งขึ้น ซึ่งใน Robert Algorithm นั้นจะทำการตอบกลับข่าวสารในเซลล์ BRM ทันที เมื่อได้รับเซลล์ FRM หากสวิตช์ 1 และ 2 ได้รับเซลล์ FRM ในขณะที่ยังไม่ได้รับข่าวสารที่ถูกต้องจากสวิตช์ 3 จะทำให้ข่าวสารที่ต้องตอบกลับทันทีในเซลล์ BRM นั้นผิดพลาด และทำให้เกิดการแกว่งของค่า ACR ขึ้น การแกว่งของค่า ACR ทำให้ Source B นั้นแกว่งตามไปด้วย และการแกว่งนั้นมีเพียงเล็กน้อย เนื่องจาก ERICA มีค่าบันทึกค่า ER ที่ได้ส่งไปในเซลล์ RM ที่สุดที่ไว้ในช่วงเวลาก่อน และหากค่า z น้อยกว่า 1.1 ก็จะนำค่านี้มาใช้ในการคำนวณด้วย ซึ่งค่านี้จะมีค่าสูง เนื่องจาก SW3 มีการคำนวณค่า ER จากค่า CCR ของแหล่งกำเนิด A ซึ่งมีค่าสูง ทำให้ค่า ER ที่บันทึกไว้มีค่าสูงตามไปด้วย และส่งผลให้ แหล่งกำเนิด B ได้รับผลกระทบจากค่า ER ที่บันทึกไว้ด้วย

อย่างไรก็ตาม การแกว่งตัวของค่า ACR ของแหล่งกำเนิด A นั้นจะไม่เกิดขึ้นหากว่าปริมาณเซลล์ FRM ที่ได้รับที่ SW1 และ SW2 มีค่าเท่า ๆ กับจำนวนเซลล์ BRM ที่ได้รับจาก branch ที่มีปัญหา

ทั้งนี้เนื่องจาก SW1 และ SW2 นั้นจะได้รับเซลล์ BRM จาก branch ที่มีปัญหา ก่อนการได้รับ FRM เสมอ ดังนั้นข่าวสารในเซลล์ BRM จาก branch ที่มีปัญหา จะทำให้ข่าวสารที่ตอบกลับใน BRM จาก SW1 และ SW2 จะเป็นค่าที่ถูกต้องจาก MER เสมอ การแกว่งตัวตามรูปที่ 4.17 นั้นเกิดจากการที่เมื่อแหล่งกำเนิด B เริ่มปรับค่า ACR ให้ลดลง ทำให้โครงข่ายเริ่มอยู่ในสถานะไม่ติดขัด ทำให้ค่า z มีค่าน้อยกว่า 1 จากนั้น ERICA นั้นจะปรับอัตราส่งของแหล่งกำเนิด A และ B ขึ้นเล็กน้อยเพื่อปรับให้ค่า z มีค่าระหว่าง 1-1.1 ในจุดนี้เองที่แหล่งกำเนิด A ปรับค่า ACR เพิ่มขึ้นนั้นจะส่งผลให้มีปริมาณ FRM ในโครงข่ายมากขึ้นด้วย ทำให้ปริมาณของ FRM ที่ SW1 และ SW2 ได้รับนั้นมากขึ้น จนมากกว่า BRM และก่อให้เกิดการตอบกลับค่า ER ที่ผิดปกติขึ้นในครั้งแรก และเหตุการณ์นั้นจะดำเนินเป็นวัฏจักรเช่นนี้ไปเรื่อย ๆ จนกลายเป็นการแกว่งตัวของค่า ACR ดังกล่าว

ขนาดหน่วยความจำที่ SW1 และ SW2 นั้นจะมีค่าเป็น 0 ดังรูปที่ 4.17 เนื่องจากปริมาณทราฟฟิกขาเข้านั้นน้อยกว่าที่สามารถรองรับได้ ส่วนหน่วยความจำที่สวิตช์ 3 นั้นจะมีค่าสูง เนื่องจากต้องรองรับทราฟฟิกสำหรับ VC ทั้ง 2 ชุดในช่วงแรก สังเกตเห็นได้ว่าขนาดของหน่วยความจำนั้นพุ่งสูงขึ้นอย่างรวดเร็ว มีสาเหตุมาจาก ค่าอัตราส่งเริ่มต้นที่แหล่งกำเนิดทั้งสองนั้นมีค่าสูง โครงข่ายจะอยู่ในสถานะคับคั่งที่แต่เริ่มต้น ทำให้ link3 ไม่สามารถรองรับความคับคั่งได้ และ ส่วนที่รองรับไม่ได้นั้นจะนำมาใส่ไว้ในหน่วยความจำที่ SW3 จนถึงที่เวลาประมาณ 50 ms นั้น จะเห็นว่าอัตราการเพิ่มขึ้นของขนาดหน่วยความจำที่สวิตช์นั้นลดลง เนื่องจากแหล่งกำเนิดทั้งสองได้ปรับอัตราส่งให้น้อยลง ทำให้มีข้อมูลส่งน้อยลง แต่อย่างไรก็ตามการเกิด consolidation noise นั้นทำให้ค่า ACR เกิดการแกว่งที่มากกว่าปริมาณทราฟฟิกที่สวิตช์สามารถรองรับได้ ทำให้ขนาดหน่วยความจำนั้นยังคงเพิ่มขึ้นเรื่อย ๆ จนไม่มีที่สิ้นสุด

สังเกตได้ว่าการใช้งาน link1 และ link2 นั้นเกิดการแกว่งขึ้น ทั้งนี้เนื่องจากเกิด consolidation noise ขึ้นนั่นเอง ทำให้ค่าอัตราส่งที่แหล่งกำเนิดนั้นแกว่งตัว ทั้ง ๆ ที่ค่าที่ควรจะเป็นสำหรับการใช้งานสายสัญญาณทั้งสองควรจะคงที่ที่ประมาณ 45% เนื่องจากสายสัญญาณรองรับได้ 149.76 Mbps แต่แหล่งกำเนิดควรสามารถส่งได้ 67.48 Mbps ส่วน link3 นั้นมีค่าใช้งาน 100% ตลอด โดยมีผลจากขนาดหน่วยความจำที่สวิตช์ยังมีอยู่ ทำให้มีข้อมูลส่งเข้าสู่ link3 ตลอดเวลาและ ทำให้การใช้งานเป็น 100 %

สังเกตได้จากรูปที่ 4.18 เมื่อนำ ERICA มาใช้ร่วมกับ TS algorithm แล้วจะเห็นว่า consolidation noise ลดลง โดยสังเกตได้จากการแกว่งตัวของค่า ACR ที่ Source A นั้นน้อยลงอย่างเห็นได้ชัด ในขณะที่ผลตอบชั่วคราวนั้นรวดเร็วเช่นเดียวกับ Robert algorithm ที่เป็นเช่นนี้เนื่องจากวิธีที่สองนั้นคล้ายกับ Robert algorithm แต่ได้เพิ่มความถูกต้องให้กับการรวบรวมข่าวสารป้อนกลับ โดยให้รับเซลล์ RM ให้ได้อย่างน้อย 1 ตัวก่อนจึงสามารถป้อนกับเซลล์ BRM ได้ อย่างไรก็ตาม consolidation noise ก็ยังคงมีผลต่อขนาดหน่วยความจำที่สวิตช์ที่ยังคงมีค่าแกว่งตัวขึ้น ๆ ลง โดยมีแนวโน้มที่จะเพิ่มขึ้นเรื่อย ๆ ในขณะที่การใช้งาน link 1 และ link2 ยังคงแกว่งตัวตามค่า ACR ของ Source A

ผลการจำลองเมื่อใช้ RSS algorithm จะเป็นดังรูปที่ 4.19 ซึ่งกลับจะเห็นว่า ค่า ACR ที่ Source A นั้นมีการแกว่งตัวมากขึ้น แสดงว่าเกิด consolidation noise เพิ่มขึ้น โดยระดับการแกว่งตัวของค่า ACR นั้นจะพอ ๆ กับเมื่อใช้ Robert algorithm ที่เป็นเช่นนี้เนื่องจาก RSS algorithm นี้ใช้หลักการเดียวกับ Robert algorithm เพียงแต่ลดความซับซ้อนในการตอบกลับเซลล์ RM ลง แต่ไม่ได้พัฒนาด้านความถูกต้องของข่าวสาร ทำให้ผลตอบชั่วคราวอยู่ในระดับที่เร็วในขณะที่ขนาดหน่วยความจำที่ SW3 นั้นมีค่าเพิ่มขึ้นเรื่อย ๆ อันเนื่องมาจาก consolidation noise ที่มาก เช่นเดียวกับการใช้งาน link1 และ link2 ที่ยังคงแกว่งตัวตามค่า ACR ของ Source A

สำหรับ Wait-for-all algorithm นั้นสามารถขจัด consolidation noise ได้ สังเกตได้จากรูปที่ 4.20 ซึ่งค่า ACR ของ Source A ที่สถานะอยู่ตัวนั้นไม่เกิดการแกว่งตัว และคงที่ที่ประมาณ 70 Mbps ส่งผลให้ค่า ACR ของ Source B มีค่าคงที่ตามไปด้วย ที่เป็นเช่นนี้เนื่องจาก Wait-for-all algorithm นี้เน้นความถูกต้องโดยรวมเซลล์ RM จากทุกกิ่งให้ครบก่อนจึงตอบกลับเซลล์ RM ทำให้ข้อมูลที่ป้อนกลับนี้ถูกต้องแน่นอน อย่างไรก็ตามผลตอบชั่วคราวของวิธีนี้นั้นจะช้ามาก โดยสังเกตได้จากค่า ACR ที่ลดลงช้ามาก เนื่องจากต้องรอเซลล์ BRM จากกิ่งที่อยู่ไกลสุดนั่นคือกิ่งที่ส่งไปยัง dA, ผลตอบชั่วคราวของวิธีการนี้จึงขึ้นอยู่กับ branch ที่ไกลที่สุด ส่งผลให้โครงข่ายอยู่ในสถานะคับคั่งในช่วงเริ่มต้นนานขึ้น ซึ่งแสดงได้จากขนาดหน่วยความจำที่ SW3 ที่พุ่งสูงขึ้นเรื่อย ๆ จนถึงจุดที่ ACR นั้นลดลงมา ขนาดหน่วยจำสูงสุดมีขนาดประมาณ 17000 ซึ่งถือว่ามีความสูงมาก อย่างไรก็ตามค่า ACR ในช่วงสถานะอยู่ตัวนั้นมีค่าคงที่ที่ค่าเหมาะสมโดยไม่เกิด consolidation noise ขึ้น ทำให้ขนาดหน่วยความจำสามารถลดลงมาได้

รูปที่ 4.21 แสดงผลการจำลองเมื่อใช้ ERICA ร่วมกับ Immediate rate calculation algorithm ซึ่งพัฒนามาจาก Wait-for-all algorithm จะเห็นได้ว่าวิธีนี้สามารถขจัด consolidation noise ไปได้เช่นกัน ในขณะที่มีผลตอบชั่วคราวที่เร็วมาก สังเกตได้จากค่า ACR ของ Source A ซึ่งลดลงอย่างรวดเร็วในช่วงแรก และคงที่ที่ประมาณ 70 Mbps ที่สถานะอยู่ตัว ทั้งนี้เนื่องจากวิธีนี้ได้เพิ่ม Overload detect ซึ่งเพิ่มความรวดเร็วในการตรวจจับความคับคั่งให้กับ Wait-for-all algorithm และทำการตอบกลับเซลล์ RM ทันทีที่เกิดสถานะ overload ซึ่งจะเห็นว่าในช่วงแรกนั้นโครงข่ายอยู่ในสถานะติดขัด และ SW3 นั้นจะอยู่ในสถานะ overload คือมีปริมาณทราฟฟิกขาเข้ามากกว่าทราฟฟิกที่รองรับได้ ดังนั้น SW3 จะทำการส่งกลับเซลล์ BRM พิเศษทันที ส่งผลให้มีการป้อนกลับที่รวดเร็ว และ แหล่งกำเนิด A สามารถรับอัตราส่งให้ลดลงอย่างรวดเร็ว เพื่อลดความคับคั่งของโครงข่ายได้ ด้วยเหตุนี้ขนาดหน่วยความจำสูงสุดที่สวิตช์ 3 จึงมีค่าไม่สูงมากนัก ในขณะที่การใช้งานสายสัญญาณ link1 link2 และ link3 นั้น มีการใช้งานได้อย่างเต็มที่เท่าที่เป็นไปได้

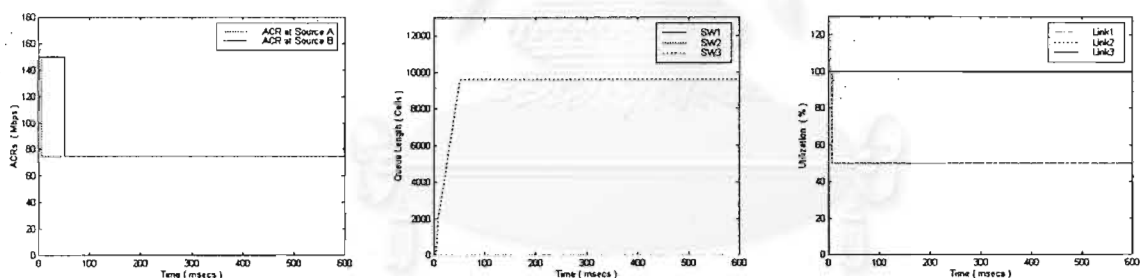
สำหรับวิธีการที่เสนอแบบที่ 1 จะเห็นได้จากรูปที่ 4.22 ว่าวิธีนี้สามารถขจัด consolidation noise ได้ ในขณะที่มีผลตอบชั่วคราวที่รวดเร็วมากเช่นเดียวกัน เนื่องจากวิธีนี้นั้นจัดเก็บค่า MER เป็นแบบ per-VC และ แบบ per-branch ทำให้ข่าวสารความคับคั่งที่เก็บไว้ใน MER มีค่าที่ถูกต้องเสมอ ดังนั้นจึงไม่

เกิดการผิดพลาดของข่าวสารที่ใช้ในการป้อนกลับ และ ไม่เกิด consolidation noise ขึ้น นอกจากนี้ยังใช้เงื่อนไขในการตอบกลับข่าวสารในเซลล์ BRM เหมือนกับ Robert algorithm ทำให้สามารถตอบกลับข่าวสารในเซลล์ BRM ได้รวดเร็วที่สุดด้วย สังเกตได้จากขนาดหน่วยความจำสูงสุดที่ใช้นั้นมีค่าน้อยกว่า Immediate rate calculation เล็กน้อย ในขณะที่การใช้งานสัญญาณมีค่าเต็มที่เท่าที่จะเป็นไปได้

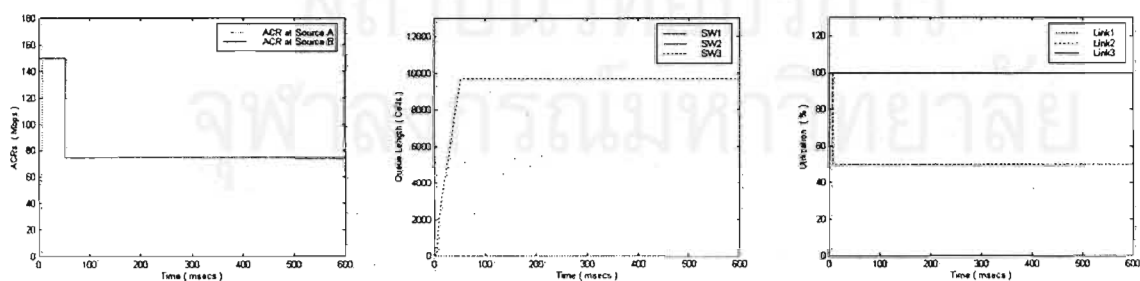
ส่วนวิธีการที่เสนอแบบที่ 2 นั้นให้ผลที่ใกล้เคียงกับวิธีการที่เสนอแบบที่ 1 สังเกตได้จากรูปที่ 4.23 โดยค่า ACR ของ source A นั้นไม่เกิดการแกว่งตัว แสดงว่าไม่เกิด consolidation noise ขึ้น ในขณะที่มีผลตอบชั่วคราวที่รวดเร็ว ทั้งนี้เนื่องจากวิธีตัดแปลงมาจาก RSS algorithm หากแต่ไม่มีการลบค่า MER ที่เก็บไว้ให้เป็นค่า PCR แต่จะตั้งค่า MER ให้เป็นค่าที่เพิ่งได้รับจากเซลล์ RM ตัวนั้น ซึ่งเป็นการลดปัญหาการลบค่า MER ให้เป็นค่าสูง ซึ่งเป็นต้นกำเนิดของการแกว่งตัวของค่า ACR นั้นเอง ส่วนผลตอบชั่วคราวมีผลตอบชั่วคราวที่รวดเร็วพอ ๆ กับ RSS algorithm เช่นกัน สังเกตได้จากค่า ACR ของ source A นั้นลดลงจากค่า ICR มาอยู่ที่ fairshare อย่างรวดเร็ว ในขณะที่ขนาดหน่วยความจำสูงสุดที่ SW3 นั้นมีค่าใกล้เคียงกับ Immediate rate calculation และ วิธีการที่เสนอแบบที่ 2 และมีการใช้งานสายสัญญาณได้อย่างเต็มที่

4.2.3 สมรรถนะของวิธี FMMRA

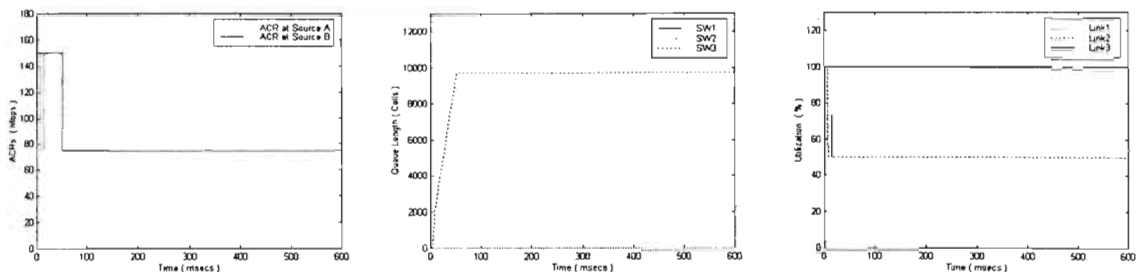
เมื่อใช้ FMMRA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะ ได้ผลดังรูปที่ 4.24 ถึง 4.30



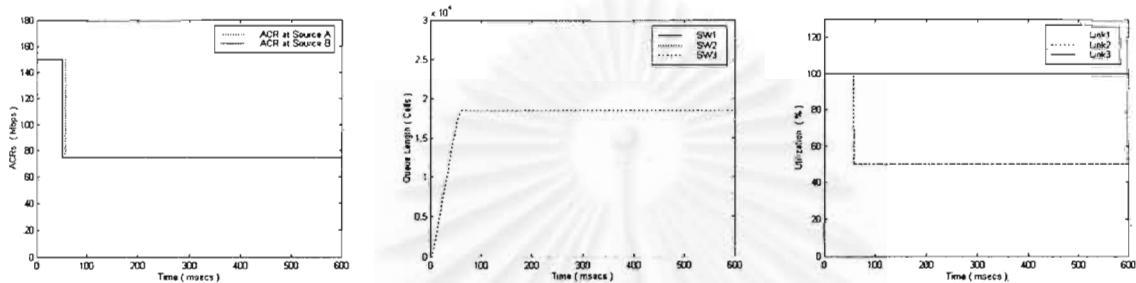
รูปที่ 4.24 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Robert Algorithm สำหรับ Chain configuration



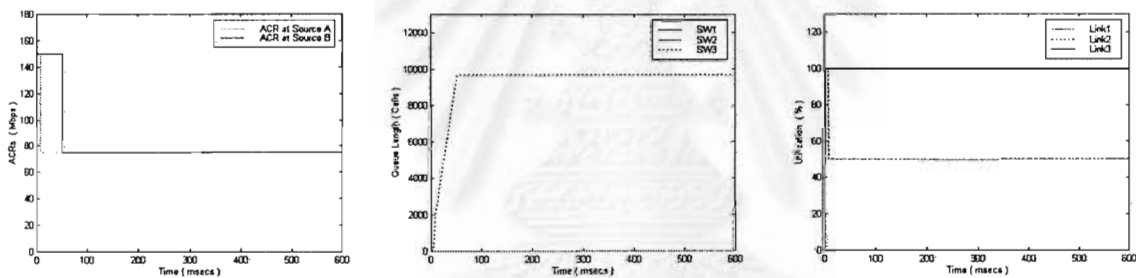
รูปที่ 4.25 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ TS Algorithm สำหรับ Chain configuration



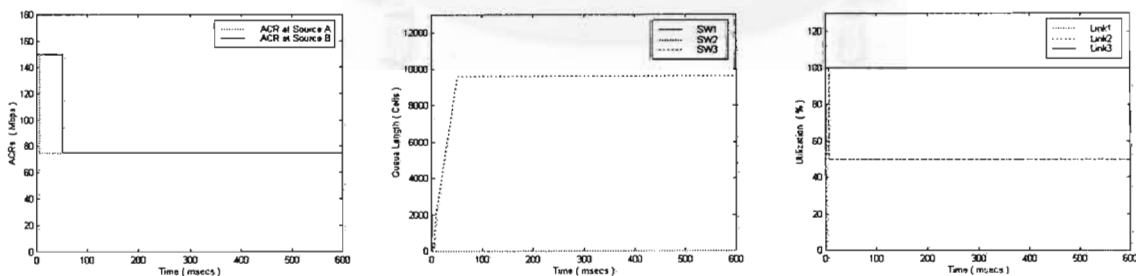
รูปที่ 4.26 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ RSS Algorithm สำหรับ Chain configuration



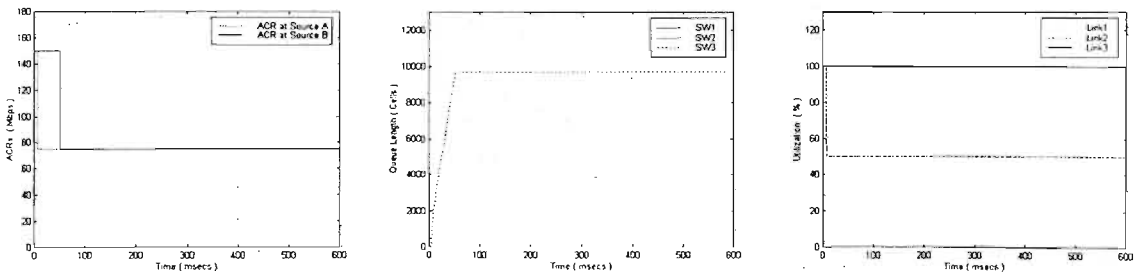
รูปที่ 4.27 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Wait-for-all Algorithm สำหรับ Chain configuration



รูปที่ 4.28 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Chain configuration



รูปที่ 4.29 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Chain configuration



รูปที่ 4.30 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Chain configuration

สังเกตได้จากรูปที่ 4.24, 4.25 และ 4.26 จะเห็นว่าเมื่อใช้ Robert algorithm, TS algorithm และ RSS algorithm ตามลำดับนั้นมีผลตอบชั่วครู่ดีมาก ซึ่งสังเกตได้จากค่า ACR ของ source A ที่ลดลงมาอยู่ที่ fairshare อย่างรวดเร็ว ในขณะที่ไม่มี consolidation noise เลย แสดงได้จากค่า ACR ของ source A ที่ไม่เกิดการแกว่งตัวเลย ทั้งนี้เนื่องจาก FMMRA นั้นให้ผลตอบชั่วครู่ที่รวดเร็วมาก และไม่มีการปรับเพิ่มค่าอัตราส่งของ source A หลังจาก source B นั้นปรับอัตราส่งลดลงมาที่ fairshare อันเป็นเหตุให้ปริมาณ FRM มากกว่า BRM และเป็นสาเหตุให้เกิดการแกว่งตัวของค่า ACR ขึ้น ดังนั้นกลไก FMMRA สามารถขจัดปัญหา consolidation noise ในรูปแบบโครงข่ายนี้ได้ และช่วยให้โครงข่ายปราศจากความคับคั่ง ซึ่งสังเกตได้จากขนาดหน่วยความจำที่ SW3 ซึ่งมีค่าไม่มากนัก จะเห็นว่าค่า ACR ของ source A ของวิธีรวบรวมความคับคั่งทั้ง 3 นั้นมีผลตอบที่คล้ายคลึงกันมาก ยังผลให้ขนาดหน่วยความจำที่ SW3 และ การใช้งานสายสัญญาณต่าง ๆ นั้นมีผลตอบคล้าย ๆ กัน

ขนาดหน่วยความจำที่สวิตช์ 3 นั้นจะเพิ่มขึ้นในช่วงแรกเนื่องจากมีสถานะไหลคเกินจากการที่แหล่งกำเนิดทั้งสองส่งข้อมูลที่อัตราส่งมากกว่าที่ link3 สามารถรองรับได้ หลังจากนั้นขนาดหน่วยความจำที่สวิตช์ 3 เริ่มมีขนาดคงที่ เนื่องจาก FMMRA นั้นทำให้โครงข่ายปรับตัวเข้าสู่สถานะอยู่ตัว โดยที่ปรับอัตราส่งให้พอดีกับปริมาณทราฟฟิกที่รองรับได้ ทำให้ปริมาณทราฟฟิกขาเข้าเท่ากับทราฟฟิกขาออกที่สวิตช์ ดังนั้นขนาดหน่วยความจำจึงมีขนาดคงที่ ซึ่งสังเกตได้จากการใช้งานสายสัญญาณของ link1 และ link2 ที่มีปริมาณ 50% ในขณะที่การใช้งาน link3 เป็น 100%

จากรูปที่ 4.27 จะเห็นว่า Wait-for-all algorithm นั้นให้ผลตอบชั่วครู่ที่ช้าเช่นเดียวกันในกรณีที่ใช้ ERICA เนื่องจากกลไกการทำงานของ Wait-for-all algorithm ทำให้แหล่งกำเนิดต้องรอให้เซลล์ BRM เดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อน จึงสามารถปรับอัตราส่งให้ลดลงได้ ส่งผลให้โครงข่ายอยู่ในสถานะคับคั่งเป็นเวลานาน และ ขนาดหน่วยความจำที่ SW3 มีค่าสูงตามไปด้วย อย่างไรก็ตาม การใช้งานสายสัญญาณยังคงเป็นไปได้อย่างเต็มที่

Immediate rate calculation algorithm นั้นปรับปรุงผลตอบชั่วครู่ของ Wait-for-all algorithm ได้เป็นอย่างดี สังเกตได้จากรูปที่ 4.28 ซึ่งค่า ACR ของ source A นั้นลดลงมาอยู่ที่ fairshare อย่างรวดเร็วใน

ช่วงแรก ทำให้โครงข่ายหลุดจากสภาวะคับคั่งอย่างรวดเร็ว สังเกตว่าผลจากจำลองที่ได้ไม่แตกต่าง Robert algorithm มากนัก

เมื่อใช้วิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 ร่วมกับ FMMRA จะได้ผลการจำลองดังรูปที่ 4.29 ซึ่งจะสังเกตได้ว่าได้ผลคล้ายคลึงกับ Robert algorithm มาก โดยไม่เกิด consolidation noise และมีผลตอบชั่วคราวที่รวดเร็วมาก

4.2.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

ทั้ง FMMRA และ ERICA นั้นสามารถปรับอัตราส่งที่แหล่งกำเนิดไปสู่อัตราส่งที่โครงข่ายต้องการได้อย่างยุติธรรมที่สุด แต่สำหรับ ERICA นั้นเกิดปัญหา consolidation noise กับวิธี Robert algorithm, TS algorithm และ RSS algorithm ทั้งนี้เนื่องจาก ERICA มีการปรับอัตราส่งเพิ่มขึ้น หรือลดลงเล็กน้อย ทำให้เกิดปัญหาในการรวบรวมข่าวสารความคับคั่ง ดังที่ได้อธิบายไปแล้ว

FMMRA นั้นสามารถลด consolidation noise ได้ เนื่องจาก FMMRA นั้นสามารถปรับอัตราส่งให้กับแหล่งกำเนิดได้อย่างรวดเร็ว และทำให้ระบบอยู่ในสภาวะอยู่ตัวอย่างรวดเร็ว

หากวิเคราะห์ด้านความซับซ้อนในการนำมาใช้ในการส่งแบบหนึ่งจุดไปสู่หลายจุดแล้วจะเห็นว่าวิธี FMMRA นั้นมีความซับซ้อนมากกว่าวิธี ERICA เนื่องจากต้องเพิ่มจำนวนตัวแปรสำหรับ β และ λ เข้าไป ทำให้มีความซับซ้อนคำนวณเพิ่มขึ้นด้วย

4.2.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีการรวบรวมข่าวสารความคับคั่งต่าง ๆ

Robert algorithm, TS algorithm และ RSS algorithm นั้นยังคงมีปัญหา consolidation noise เมื่อใช้กับ ERICA มีซึ่งวิธีแก้ไขคือ เพิ่มความถูกต้องของข้อมูลที่เก็บรวบรวมไว้ เช่นการรอข่าวสารให้ได้ครบจากทุก branch ก่อนการป้อนกลับ ดังเช่น Wait-for-all algorithm ซึ่งจะทำให้มีผลตอบชั่วคราวที่ช้าลงตามไปด้วย Immediate rate calculation จึงเสนอมาเพื่อแก้ปัญหานี้ โดยหลักการการทำงานจะมีลักษณะคล้ายคลึงกับ Wait-for-all algorithm เพียงแต่เพิ่มความสามารถในการตรวจจับสภาวะ overload ขึ้นมา

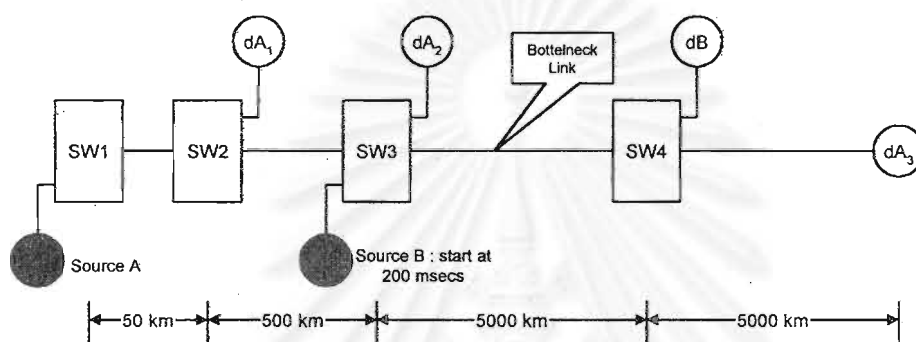
วิธีการอื่นที่ใช่แก้ปัญหาได้แก่ วิธีที่เสนอทั้งสองวิธีนั่นเอง ในวิธีการที่เสนอแบบที่ 1 นั้นมีการแก้ปัญหาโดยการเพิ่มปริมาณตัวแปรที่ใช้เก็บข่าวสารความคับคั่งให้เป็นแบบ per-VC และ per-branch ทำให้ข้อมูลที่จะป้อนกลับนั้นถูกต้องเสมอ อีกทั้งยังมีเงื่อนไขในการป้อนกลับเซลล์ BRM ที่เหมือนกับ Robert algorithm ทำให้มีผลตอบชั่วคราวที่รวดเร็วที่สุด อย่างไรก็ตามการแก้ปัญหาดังกล่าววิธีจะเพิ่มความซับซ้อนให้กับสวิทช์ เนื่องจากต้องมีการเพิ่มปริมาณตัวแปรขึ้นมา

ส่วนในวิธีการที่เสนอแบบที่ 2 นั้น มีวิธีแก้ปัญหที่ง่ายกว่า โดยการไม่ลบล้างข่าวสารความคับคั่งให้มีค่าสูง หากแต่ลบล้างให้กลายเป็นค่าที่เหมาะสมจาก BRM ที่ได้รับนั่นเอง วิธีนี้สามารถลดโอกาสการเกิด consolidation noise ได้อย่างมาก

4.3 ผลการจำลองสำหรับ Chain configuration with low ICR

4.3.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ

รูปแบบโครงข่ายนี้พัฒนามาจาก chain configuration โดยปรับอัตราส่งเริ่มต้นของแหล่งกำเนิด A มีค่าน้อย คือประมาณ 25 Mbps และ แหล่งกำเนิด B ซึ่งเป็น point-to-point connection นั้นเริ่มส่งข้อมูลที 200 msec ทั้งนี้เพื่อทดสอบผลตอบสนองชั่วคราวในสภาวะโหลดต่ำ รูปแบบโครงข่ายจะเป็นดังรูปที่ 4.31



รูปที่ 4.31 Chain configuration with low ICR

ค่าตัวแปรต่าง ๆ ที่ใช้ในรูปแบบโครงข่ายนี้ ได้แก่

Source A : PCR = 149.76 Mbps , ICR = 25 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

Source B : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125, Nrm = 32 และ แหล่งกำเนิดเริ่มส่งข้อมูลที่ 200 msecs

SW1 : Maximum queue length = 30,000

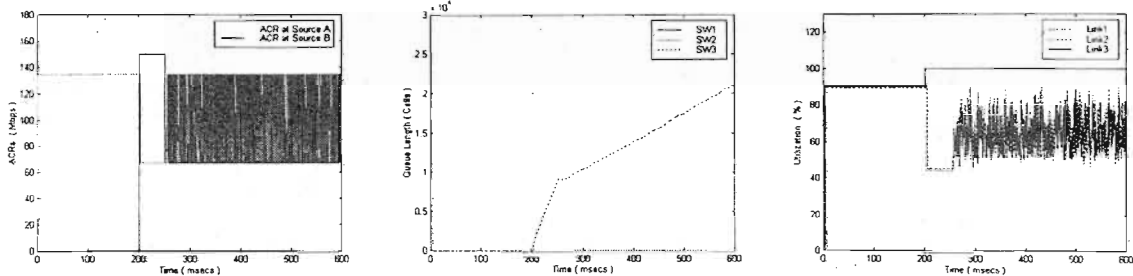
SW2 : Maximum queue length = 30,000

SW3 : Maximum queue length = 30,000

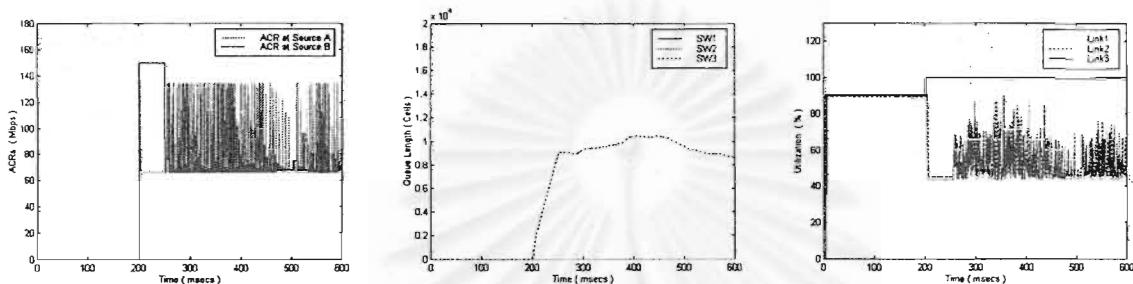
SW4 : Maximum queue length = 30,000

4.3.2 สมรรถนะของวิธี ERICA

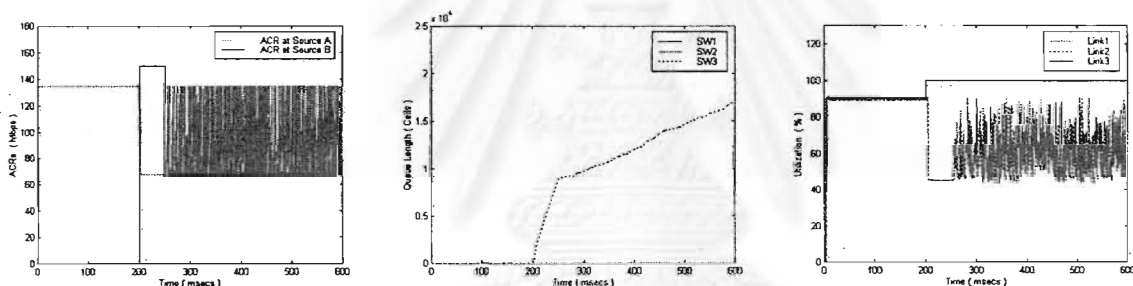
เมื่อใช้ ERICA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.32 ถึง 4.38



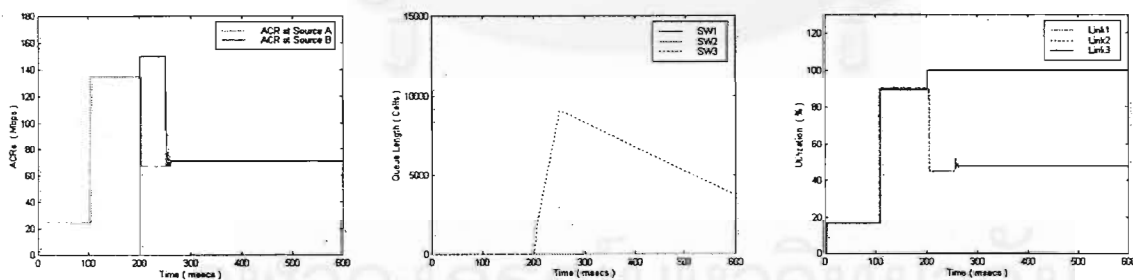
รูปที่ 4.32 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Robert Algorithm สำหรับ Chain configuration with low ICR



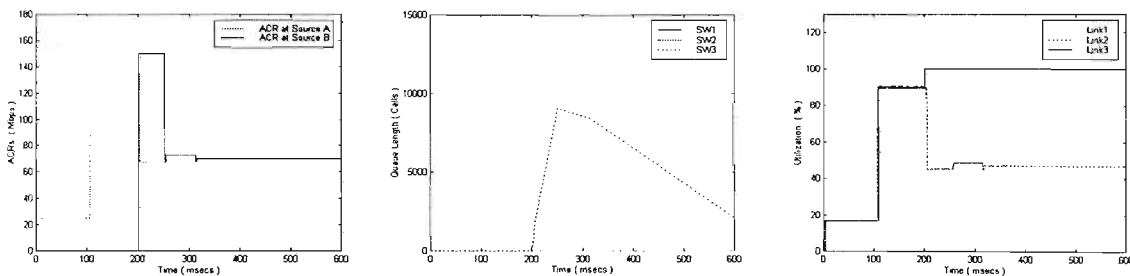
รูปที่ 4.33 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ TS Algorithm สำหรับ Chain configuration with low ICR



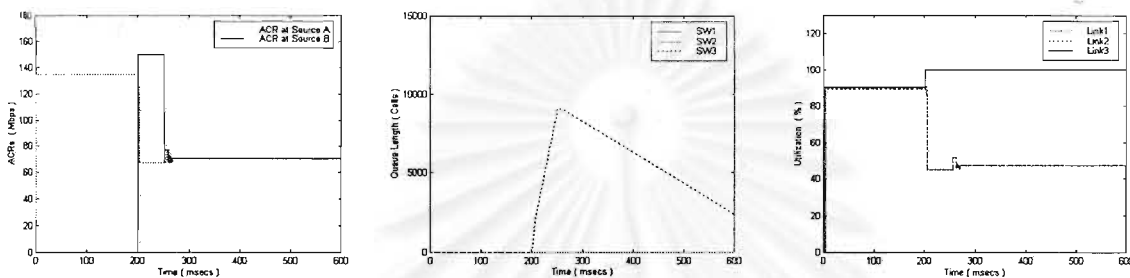
รูปที่ 4.34 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ RSS Algorithm สำหรับ Chain configuration with low ICR



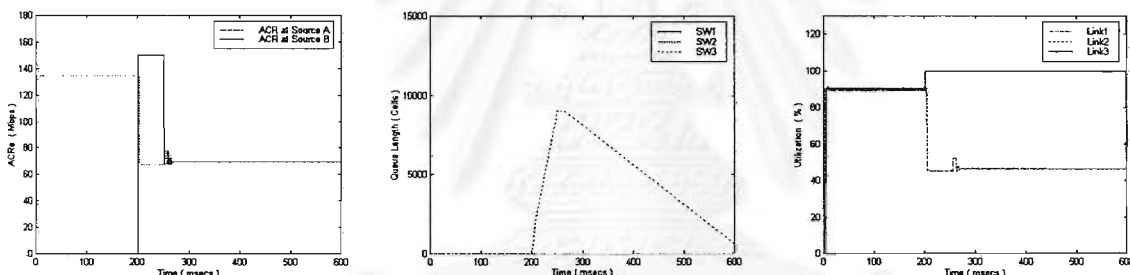
รูปที่ 4.35 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Wait-for-all Algorithm สำหรับ Chain configuration with low ICR



รูปที่ 4.36 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Chain configuration with low ICR



รูปที่ 4.37 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Chain configuration with low ICR



รูปที่ 4.38 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Chain configuration with low ICR

รูปแบบโครงข่ายนี้ถูกออกแบบมา เพื่อทดสอบสมรรถนะทางด้านผลตอบชั่วคราวของระบบนั่นเอง สังเกตได้จากรูปที่ 4.32 จะเห็นว่าเมื่อนำ Robert algorithm มาใช้งานร่วมกับ ERICA นั้นจะมีผลตอบชั่วคราวที่รวดเร็วมาก สังเกตได้จากค่า ACR ของแหล่งกำเนิด A ที่สามารถปรับตัวจากค่า ICR ที่มีค่าที่ 25 Mbps ไปสู่ค่าที่โครงข่ายสามารถรองรับได้คือที่ประมาณ 135 (150 x target utilization) Mbps นั่นเอง ที่เป็นเช่นนี้เนื่องจากเมื่อ SW1 ได้รับเซลล์ FRM ตัวแรกจากแหล่งกำเนิด A จะทำการตอบกลับเซลล์ BRM ที่บรรจุอัตราส่งที่โครงข่ายสามารถรองรับได้ในขณะนั้น และส่งกลับไปยังแหล่งกำเนิด ซึ่งขั้นตอนนี้จะใช้เวลาเร็วมาก อย่างไรก็ตามเมื่อแหล่งกำเนิด B เริ่มส่งข้อมูล ซึ่งมีผลทำให้แหล่งกำเนิด A ต้องปรับลดอัตราส่งเพื่อให้เหมาะสมกับโครงข่าย และต่อมาจึงเกิด consolidation noise ขึ้นอย่างหนักเช่นเดียวกับ chain configuration

TS algorithm ให้ผลตอบชั่วคราวที่รวดเร็วเช่นกัน เนื่องจาก SW1 มีการส่งเซลล์ BRM กลับไปหาแหล่งกำเนิดทันทีที่ได้รับเซลล์ FRM หลังจากที่ได้รับเซลล์ BRM จาก branch ใด ๆ อย่างน้อย 1 ตัว ดังนั้น SW1 จะรอเซลล์ BRM จากปลายทางที่ใกล้ที่สุดคือปลายทาง dA_1 ก่อน แล้วจึงรอให้มีเซลล์ FRM มาถึง

จึงสามารถสร้างเซลล์ BRM ส่งกลับไปสู่แหล่งกำเนิดได้ จะเห็นว่าปลายทาง dA_1 นั้นใกล้เคียงกับแหล่งกำเนิดพอสมควร ทำให้ผลตอบชั่วคราวนั้นรวดเร็วมาก อย่างไรก็ตามหลังจากแหล่งกำเนิด B เริ่มส่งข้อมูล แหล่งกำเนิด A นั้นก็เกิด consolidation noise เช่นเดียวกับ Robert's algorithm แต่ระดับของการแกว่งน้อยกว่า Robert's algorithm พอสมควร

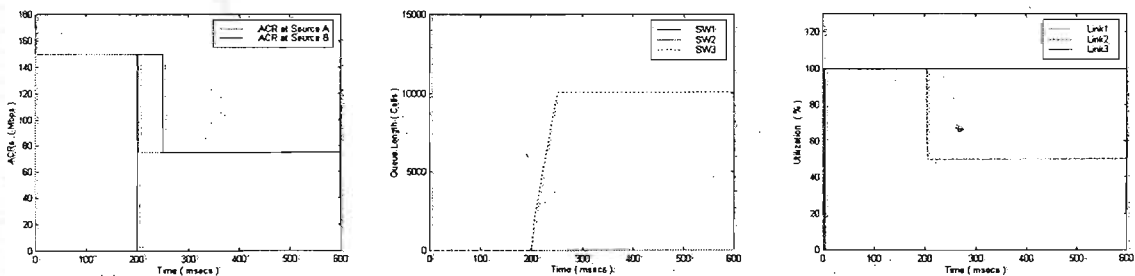
สำหรับ RSS algorithm นั้นสามารถปรับอัตราส่งไปสู่ค่าที่โครงข่ายสามารถรองรับได้อย่างรวดเร็ว ซึ่งจะเห็นว่าในช่วงแรก SW1 นั้นจะตอบกลับเซลล์ BRM เมื่อได้รับเซลล์ BRM จาก dA_1 นั้นเอง ทำให้แหล่งกำเนิดสามารถปรับอัตราส่งได้รวดเร็วเช่นเดียวกับ TS algorithm อย่างไรก็ตาม เมื่อแหล่งกำเนิด B เริ่มส่งข้อมูล ก็ยังคงมี consolidation noise เช่นเดียวกับในกรณี Robert algorithm

เมื่อใช้ Wait-for-all algorithm และ Immediate rate calculation algorithm กับรูปแบบโครงข่ายนี้ จะเห็นว่าได้ผลการจำลองที่เหมือนกัน เนื่องจากในสถานะที่โครงข่ายไม่อยู่ในสถานะไหลคเกินนั้น Immediate rate calculation algorithm จะทำงานเหมือนกับ Wait-for-all algorithm นั้นเอง สังเกตได้จากรูปที่ 4.35 และ 4.36 ว่าวิธีการทั้งสองนั้นสามารถขจัด consolidation noise ไปได้ แต่อย่างไรก็ตาม ผลตอบชั่วคราวของวิธีการทั้งสองนั้นช้ามาก สังเกตได้จากค่า ACR ที่ไม่สามารถเพิ่มขึ้นจนถึงเวลาประมาณ 100 msec ทั้งนี้เนื่องจากแหล่งกำเนิดต้องรอให้เซลล์ BRM เดินทางกลับมาจากปลายทางที่ไกลที่สุด ซึ่งได้แก่ ปลายทาง dA_3 ส่งผลให้มีการใช้งานสัญญาณต่าง ๆ อย่างไม่คุ้มค่า สังเกตได้จากค่าการใช้งานสายสัญญาณสายสัญญาณ link1, link2 และ link3 ที่มีค่าประมาณ 20 % ในช่วง 0-100 msec นั้นเอง

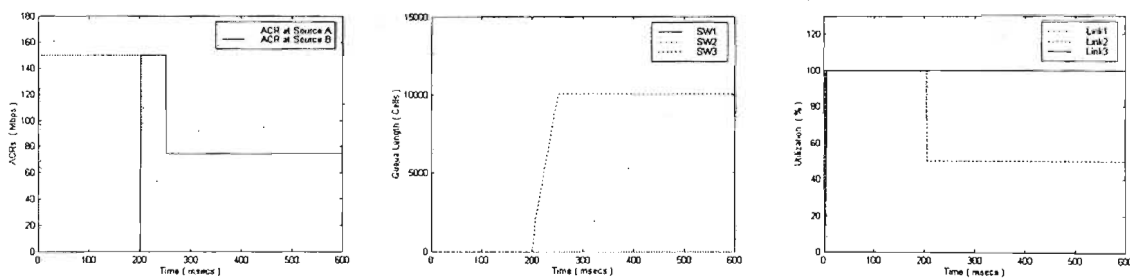
จากรูปที่ 4.37 และ 4.38 ซึ่งแสดงผลการจำลองสำหรับวิธีการที่เสนอแบบที่ 1 และ 2 นั้น จะเห็นว่าให้สมรรถนะที่ใกล้เคียงกัน โดยสามารถขจัด consolidation noise ไปได้ ในขณะที่มีผลตอบชั่วคราวที่รวดเร็ว สำหรับวิธีการที่เสนอแบบที่ 1 นั้นใช้เงื่อนไขในการตอบกลับเซลล์ BRM เหมือนกับ Robert algorithm ทำให้มีผลตอบสนองชั่วคราวที่รวดเร็วมาก ส่วนวิธีการที่เสนอแบบที่ 2 นั้นใช้เงื่อนไขการตอบกลับเซลล์ BRM เหมือนกับ RSS algorithm แต่เพิ่มการป้องกันไม่ให้เกิด consolidation noise ขึ้น ทำให้ได้สมรรถนะที่ดีมากทั้งสองวิธี

4.3.3 สมรรถนะของวิธี FMMRA

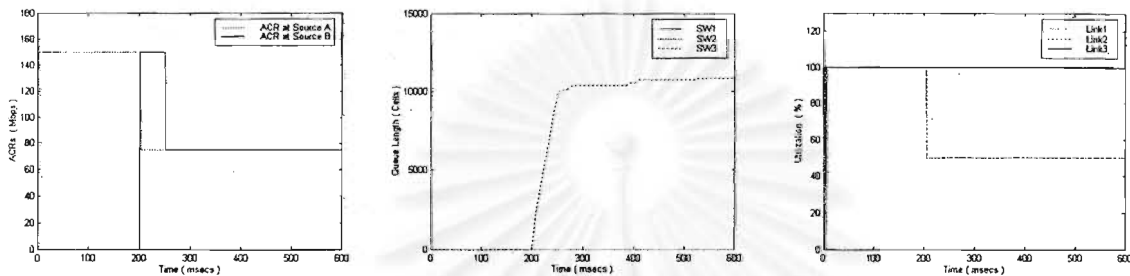
เมื่อใช้ FMMRA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะ ได้ผลดังรูปที่ 4.39 ถึง 4.45



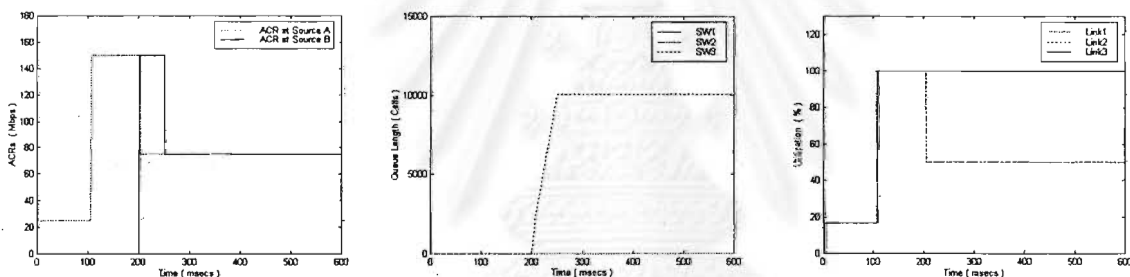
รูปที่ 4.39 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Robert Algorithm สำหรับ Chain configuration with low ICR



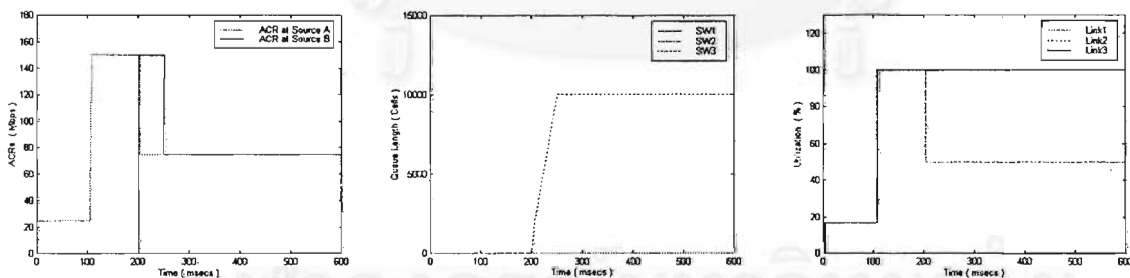
รูปที่ 4.40 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ TS Algorithm สำหรับ Chain configuration with low ICR



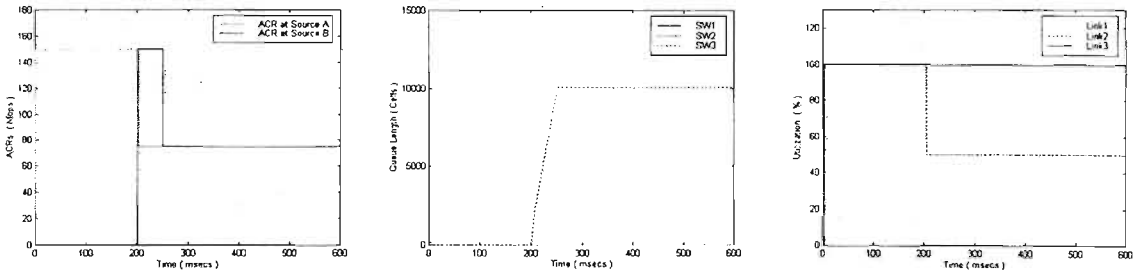
รูปที่ 4.41 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ RSS Algorithm สำหรับ Chain configuration with low ICR



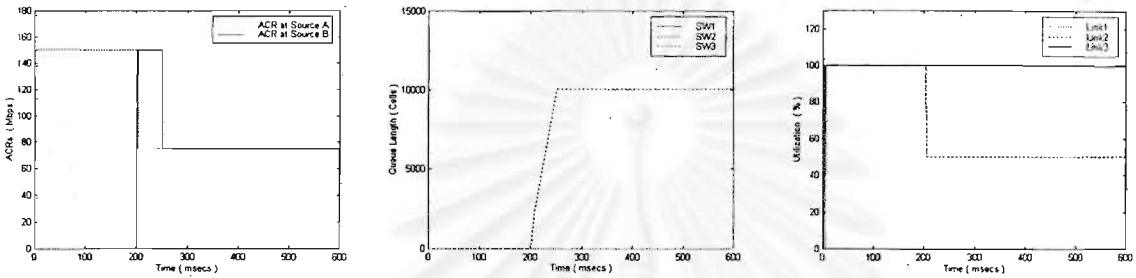
รูปที่ 4.42 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Wait-for-all Algorithm สำหรับ Chain configuration with low ICR



รูปที่ 4.43 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Chain configuration with low ICR



รูปที่ 4.44 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Chain configuration with low ICR



รูปที่ 4.45 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Chain configuration with low ICR

สังเกตได้จากรูปที่ 4.39 - 4.41 ได้ว่าเมื่อใช้ Robert algorithm, TS algorithm และ RSS algorithm ร่วมกับ FMMRA แล้วปรากฏว่าให้สมรรถนะที่ไม่แตกต่างกัน โดยที่มีผลตอบสนองชั่วคราวที่รวดเร็วในขณะที่สามารถขจัด consolidation noise ได้ ทั้งนี้เนื่องจาก FMMRA นั้นสามารถขจัด consolidation noise ไปได้ ในขณะที่ผลตอบสนองชั่วคราวของทั้งสามวิธีนั้นมีความรวดเร็วมาก สังเกตได้จากค่า ACR ของแหล่งกำเนิด A ที่ปรับเพิ่มขึ้นได้อย่างรวดเร็ว

เมื่อใช้ Wait-for-All algorithm และ Immediate rate calculation algorithm ร่วมกับ วิธี FMMRA จะให้ผลตอบที่คล้ายคลึงกัน เนื่องจากโครงข่ายอยู่ในสถานะโหลดต่ำ ทำให้วิธี Immediate rate calculation algorithm จะทำงานเหมือนกับ Wait-for-all algorithm จะเห็นว่าวิธีการทั้งสองนั้นมีผลตอบสนองชั่วคราวที่ช้ามาก ค่า ACR ของแหล่งกำเนิด A นั้นมีค่าคงที่ที่ ICR (25 Mbps) โดยไม่สามารถปรับตัวเข้าสู่อัตราส่งที่โครงข่ายสามารถรองรับได้คือที่ค่า 150 Mbps ทั้งนี้เนื่องจากวิธีการทั้งสองทำให้ แหล่งกำเนิดต้องรอให้เซลล์ BRM นั้นเดินทางกลับจากปลายทางที่ไกลที่สุดก่อนจึงสามารถปรับอัตราส่งให้เหมาะสมได้ อย่างไรก็ตาม การรอให้เซลล์ BRM เดินทางมาจากทุก branch ก่อนการป้อนกลับเซลล์ BRM นั้นสามารถขจัด consolidation noise ไปได้ โดยค่า ACR ที่แหล่งกำเนิด A นั้นไม่เกิดการแกว่งตัวขึ้น

จากรูปที่ 4.44 และ 4.45 สังเกตได้ว่าวิธีการที่เสนอทั้งสองวิธีมีผลลัพธ์ที่คล้ายคลึงกัน และคล้ายคลึงกับเมื่อใช้ Robert algorithm โดยที่ไม่มี consolidation noise เลย ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็วมาก

4.3.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

ทั้ง FMMRA และ ERICA นั้นสามารถปรับอัตราส่งที่แหล่งกำเนิดไปสู่อัตราส่งที่โครงข่ายต้องการได้อย่างยุติธรรมที่สุด แต่สำหรับ ERICA นั้นเกิดปัญหา consolidation noise กับวิธี Robert algorithm, TS algorithm และ RSS algorithm สำหรับวิธีควบคุมความคับคั่งอื่น ๆ นั้นไม่เกิด consolidation noise ได้หากแต่วิธี Wait-for-all algorithm และ Immediate rate calculation algorithm นั้นมีผลตอบสนองชั่วคราวที่ช้ามาก

เมื่อใช้ FMMRA นั้นไม่มีปัญหาเรื่อง consolidation noise เลย เนื่องจาก FMMRA นั้นสามารถปรับอัตราส่งให้กับแหล่งกำเนิดได้อย่างรวดเร็ว และทำให้ระบบอยู่ในสถานะอยู่ตัวอย่างรวดเร็ว

หากวิเคราะห์ด้านความซับซ้อนในการนำมาใช้ในการส่งแบบหนึ่งจุดไปสู่หลายจุดแล้วจะเห็นว่าวิธี FMMRA นั้นมีความซับซ้อนมากกว่าวิธี ERICA เนื่องจากต้องเพิ่มจำนวนตัวแปรสำหรับ β และ λ เข้าไป ทำให้มีความซับซ้อนคำนวณเพิ่มขึ้นด้วย

4.3.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ

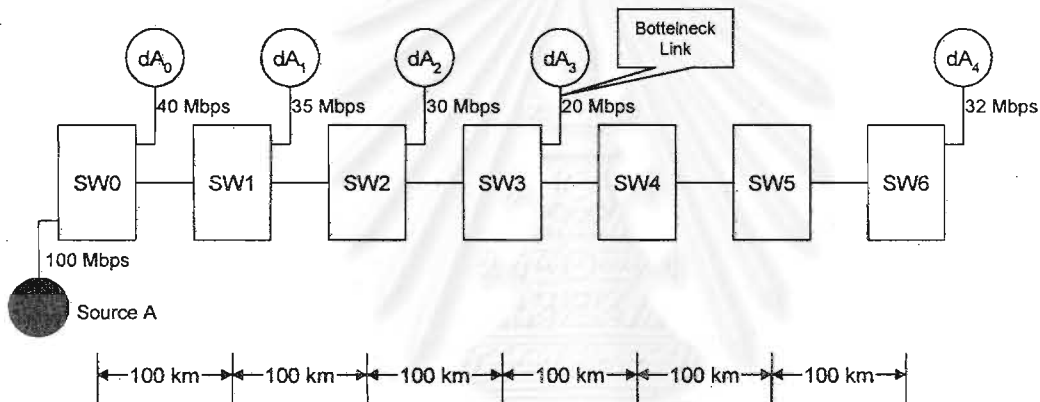
Robert algorithm, TS algorithm และ RSS algorithm นั้นยังคงมีปัญหา consolidation noise เมื่อใช้กับ ERICA ในขณะที่วิธี Wait-for-all algorithm, Immediate rate calculation algorithm, วิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 นั้นสามารถขจัด Consolidation noise ไปได้

ส่วนสมรรถนะด้านผลตอบสนองชั่วคราวนั้นสามารถแยกเป็นสองประเด็นได้แก่ ในสถานะที่โครงข่ายอยู่ในภาวะโหลดเกินนั้น วิธี Immediate rate calculation algorithm สามารถปรับอัตราส่งของแหล่งกำเนิดได้อย่างรวดเร็ว แต่เมื่อโครงข่ายอยู่ในสถานะโหลดต่ำนั้น วิธีนี้จะมีผลตอบสนองที่ช้ามาก อย่างไรก็ตาม วิธี Robert algorithm, TS algorithm, RSS algorithm, วิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 นั้นมีผลตอบสนองชั่วคราวที่เร็วมากทั้งในสถานะ โหลดเกินและ โหลดต่ำ ส่วน Wait-for-all algorithm นั้นมีผลตอบสนองชั่วคราวที่ช้าในทุกสถานะ

4.4 ผลการจำลองสำหรับ Jiang configuration

4.4.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ

Jiang ได้เสนอรูปแบบโครงข่ายขึ้นมาเพื่อทดสอบประสิทธิภาพระหว่างวิธีการรวบรวมความคับคั่งต่าง ๆ นั้นในการปรับค่าอัตราส่งของแหล่งกำเนิดว่าสามารถปรับตัวตามค่าที่โครงข่ายสามารถรองรับได้หรือไม่ โดยโครงข่ายประกอบด้วยแหล่งกำเนิด A ที่ส่งข้อมูลให้กับปลายทาง dA_0 , dA_1 , dA_2 , dA_3 และ dA_4 ตามลำดับ และ สายสัญญาณที่ส่งถึงปลายทางทั้ง 5 นั้นมีความจุ 40, 35, 30, 20 และ 32 Mbps ตามลำดับ นอกจากนี้สำหรับวิธี ERICA ที่ใช้ในโครงข่ายนี้จะตั้งค่า target utilization เป็น 1 เพื่อให้แหล่งกำเนิดสามารถใช้งานแบนวิธที่มีอยู่ของโครงข่ายได้อย่างเต็มที่ รูปแบบโครงข่ายจะเป็นดังรูปที่ 4.46



รูปที่ 4.46 Jiang configuration

ค่าตัวแปรต่าง ๆ ที่ใช้ในรูปแบบโครงข่ายนี้ได้แก่

Source A : PCR = 5 Mbps , ICR = 149.76 Mbps, MCR = 0 Mbps, RIF = 1 , RDF = 0.0125 และ Nrm = 32

SW0 : Maximum queue length = 1,000

SW1 : Maximum queue length = 1,000

SW2 : Maximum queue length = 1,000

SW3 : Maximum queue length = 1,000

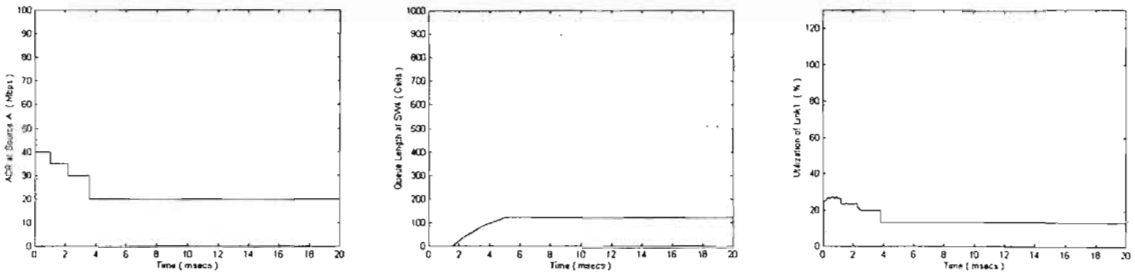
SW4 : Maximum queue length = 1,000

SW5 : Maximum queue length = 1,000

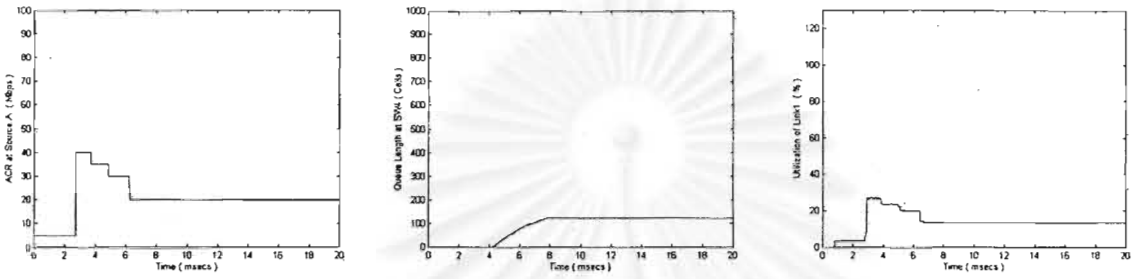
SW6 : Maximum queue length = 1,000

4.4.2 สมรรถนะของวิธี ERICA

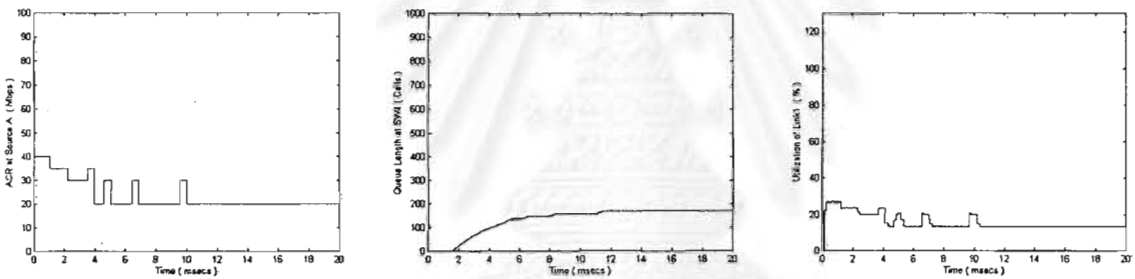
เมื่อใช้ ERICA ร่วมกับวิธีการรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.47 ถึง 4.53



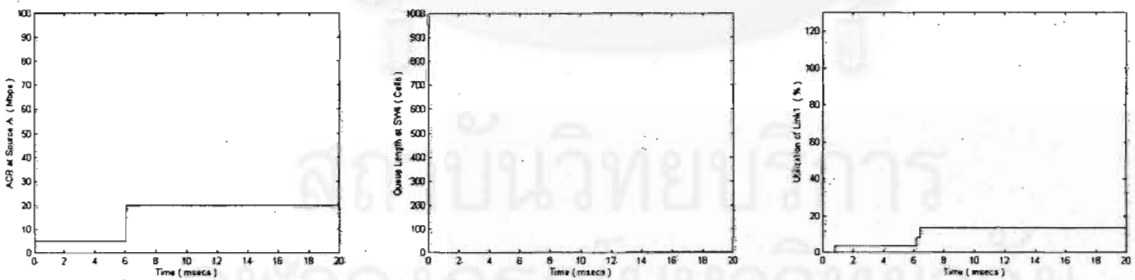
รูปที่ 4.47 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Robert Algorithm สำหรับ Jiang configuration



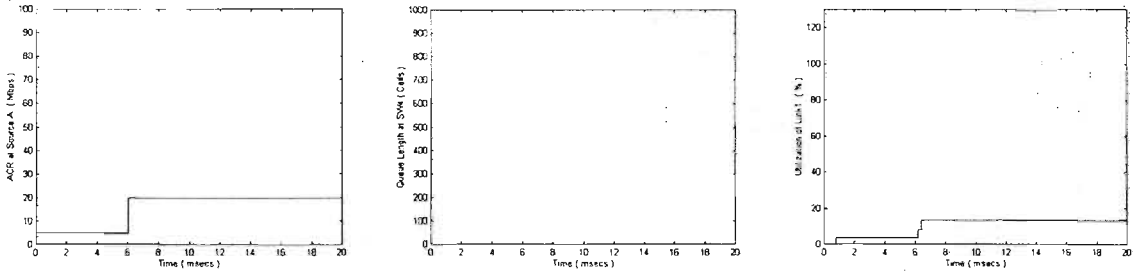
รูปที่ 4.48 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ TS Algorithm สำหรับ Jiang configuration



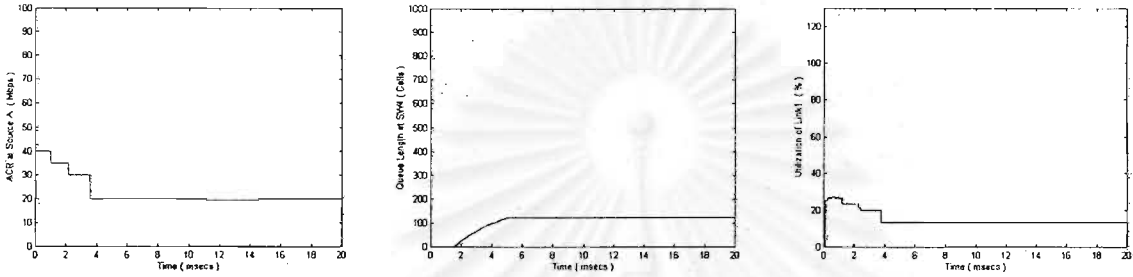
รูปที่ 4.49 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ RSS Algorithm สำหรับ Jiang configuration



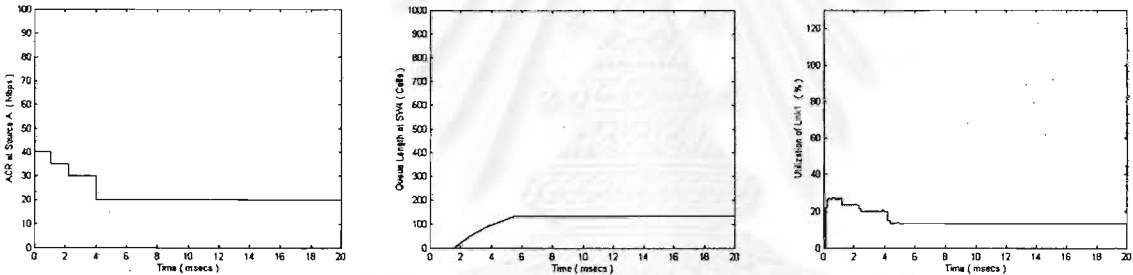
รูปที่ 4.50 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Wait-for-all Algorithm สำหรับ Jiang configuration



รูปที่ 4.51 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Jiang configuration



รูปที่ 4.52 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Jiang configuration



รูปที่ 4.53 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Jiang configuration

ในการจำลองการทำงานของโครงข่ายนี้ใช้ target utilization เป็น 100% ทำให้สวิตช์สามารถแบ่งแบนวิธซ์ของสายสัญญาณต่าง ๆ ให้กับแหล่งกำเนิดต่าง ๆ ได้อย่างเต็มประสิทธิภาพ สายสัญญาณที่มีจะเกิดความคับคั่งนั้นได้แก่ สายสัญญาณที่อยู่ระหว่าง SW3 และ ปลายทาง dA, โดยค่าอัตราส่งที่แหล่งกำเนิดควรจะส่งน้อยอยู่ที่ 20 Mbps

เมื่อพิจารณาอัตราส่งที่แหล่งกำเนิด A ตามรูปที่ 4.47 จะสังเกตได้ว่า Robert algorithm นั้นสามารถปรับอัตราส่งให้สามารถเพิ่มขึ้นเป็น 40 Mbps ได้อย่างรวดเร็ว จากนั้นที่เวลาประมาณ 1, 2 และ 3 ms จึงลดลงมาที่ 35, 30 และ 20 ตามลำดับ ทั้งนี้เนื่องจากแหล่งกำเนิดนั้นได้รับเซลล์ BRM ที่ตอบกลับมาจาก SW0, SW1, SW2 และ SW3 ตามลำดับ จึงสามารถปรับอัตราส่งให้เหมาะสมกับค่าที่สวิตช์ต่าง ๆ ดังกล่าวสามารถรองรับได้ สังเกตได้ว่าเวลาที่แหล่งกำเนิดใช้ปรับตัวตามอัตราส่งที่สวิตช์ต่าง ๆ สามารถรองรับได้นั้นรวดเร็วมาก แสดงให้เห็นถึงผลตอบชั่วคราวที่รวดเร็วของ Robert algorithm อย่างไรก็ตามการเพิ่มอัตราส่งในช่วงแรกนั้นส่งผลให้หน่วยความจำที่ SW4 นั้นมีค่าเพิ่มขึ้นในช่วงแรกจนถึงประมาณ 120 เซลล์ และมีค่าคงที่เนื่องจากค่า target utilization เป็น 100% ทำให้สวิตช์สามารถใช้งาน

สายสัญญาณได้อย่างเต็มที่ และ ปริมาณทราฟฟิกขาออกมีปริมาณเท่า ๆ กับปริมาณทราฟฟิกขาเข้า ดังนั้นจำนวนเซลล์ในหน่วยความจำจึงมีค่าคงที่

TS algorithm ในรูปที่ 4.48 นั้น มีผลตอบสนองชั่วคราวที่ช้ากว่า Robert algorithm อยู่ประมาณ 1 ms สังเกตได้จากค่าอัตราส่งของแหล่งกำเนิด A ที่เริ่มปรับตัวจากค่า ICR คือ 10 Mbps ไปสู่ 40 Mbps ที่เวลา 1 msec จากนั้นจึงปรับตัวลดลงมาที่ 35, 30 และ 20 Mbps ตามลำดับเช่นเดียวกับในกรณี Robert algorithm สาเหตุที่วิธีการนี้มีผลตอบสนองชั่วคราวที่ช้ากว่า Robert algorithm นั้น เนื่องจาก TS algorithm มีเงื่อนไขในการตอบกลับเซลล์ BRM โดยพิจารณาจากการได้รับเซลล์ FRM ในขณะที่เคยได้รับเซลล์ BRM มาแล้วอย่างน้อยหนึ่งตัวหลังจากตอบกลับเซลล์ BRM ไปแล้ว ทำให้เมื่อสวิตช์ได้รับเซลล์ FRM ตัวแรกนั้นจะยังไม่ตอบกลับเซลล์ BRM ในทันที แต่จะรอให้ได้รับเซลล์ BRM จากปลายทาง และได้รับเซลล์ FRM ตัวที่สองก่อนซึ่งช่วงเวลานี้ใช้เวลานานมากเพราะค่าอัตราส่งที่แหล่งกำเนิดมีค่าน้อย ทำให้ปริมาณเซลล์ที่สร้างขึ้นมามีน้อยส่งผลให้ปริมาณเซลล์ FRM มีน้อยตามไปด้วย แหล่งกำเนิดต้องใช้เวลา นานกว่าสามารถให้กำเนิดเซลล์ FRM ตัวที่สองได้ ทำให้การป้อนกลับเซลล์ BRM ของสวิตช์นั้นช้าตามไปด้วย

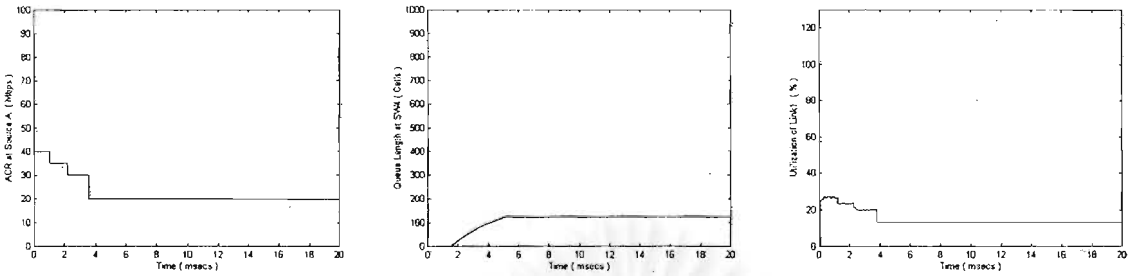
ผลการจำลองสำหรับ RSS algorithm แสดงอยู่ในรูปที่ 4.49 ซึ่งจะสังเกตว่าวิธีการนี้มีผลตอบสนองชั่วคราวที่รวดเร็วมก เนื่องจากวิธีการนี้รอเพียงให้ได้รับเซลล์ BRM หลังเพิ่งจากได้รับเซลล์ FRM จึงสามารถป้อนกลับเซลล์ BRM สู่อุปกรณ์ได้ ทำให้สวิตช์ต่าง ๆ นั้นสามารถป้อนกลับเซลล์ BRM ได้อย่างรวดเร็ว

Wait-for-all algorithm และ Immediate rate calculation นั้นมีผลการจำลองที่เหมือนกัน สังเกตได้จากรูปที่ 4.50 และ 4.51 โดยวิธีการทั้งสองนั้นให้ผลตอบสนองชั่วคราวที่ช้ามาก สังเกตได้จากค่าอัตราส่งของแหล่งกำเนิด A ที่ไม่สามารถปรับเพิ่มขึ้นจากค่า ICR ได้จนกระทั่งเวลาประมาณ 6 ms ทั้งนี้เนื่องจากในวิธีการ Wait-for-all algorithm นั้นแหล่งกำเนิดต้องรอให้เซลล์ BRM นั้นเดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อนนั่นคือปลายทาง d_A นั่นเอง ซึ่งเวลาที่แหล่งกำเนิดต้องรอนั้นสอดคล้องกับระยะทางที่เซลล์ BRM ใช้ในการเดินทางไปและกลับจากปลายทาง d_A คือ 1,200 กิโลเมตร นั่นเอง สาเหตุที่ Immediate rate calculation มีผลการจำลองที่เหมือนกับ Wait-for-all algorithm นั้น เนื่องจากโครงข่ายไม่อยู่ในสถานะโหลดเกิน ทำให้ overload detect technique นั้นไม่ทำงาน ดังนั้นวิธีดังกล่าวทำงานเหมือน Wait-for-all algorithm นั่นเอง

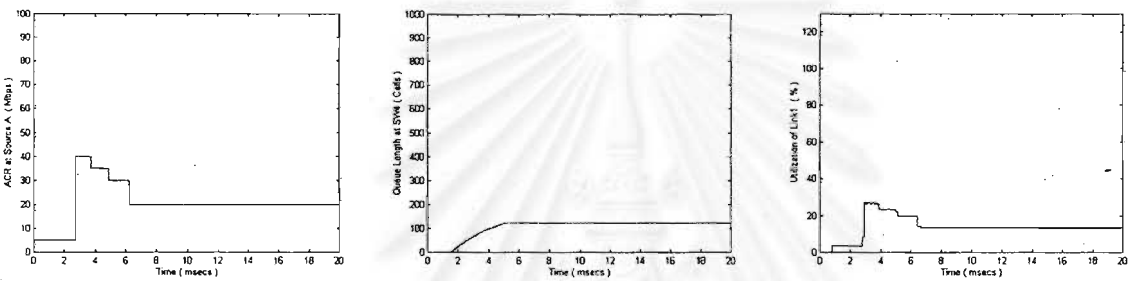
ผลการจำลองของวิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 มีลักษณะที่เหมือนกัน และเหมือนกัน Robert algorithm และ RSS algorithm เนื่องจากวิธีการที่เสนอแบบที่ 1 นั้นใช้เงื่อนไขในการตอบกลับเซลล์ BRM เหมือน Robert algorithm และ วิธีการที่เสนอแบบที่ 2 นั้นใช้เงื่อนไขในการตอบกลับเซลล์ BRM เหมือน RSS algorithm โดยสังเกตได้ว่าทั้งสองวิธีนั้นมีผลตอบสนองชั่วคราวที่รวดเร็ว โดยสามารถปรับอัตราส่งให้เหมาะสมกับค่าที่โครงข่ายสามารถรองรับได้เป็นอย่างดี

4.4.3 สมรรถนะของวิธี FMMRA

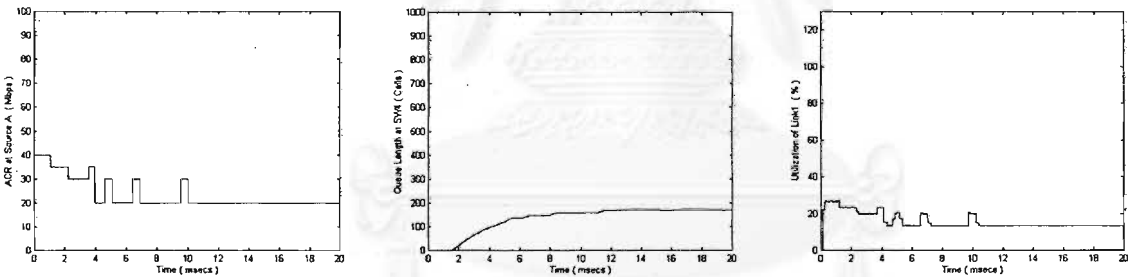
เมื่อใช้ FMMRA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.54 ถึง 4.60



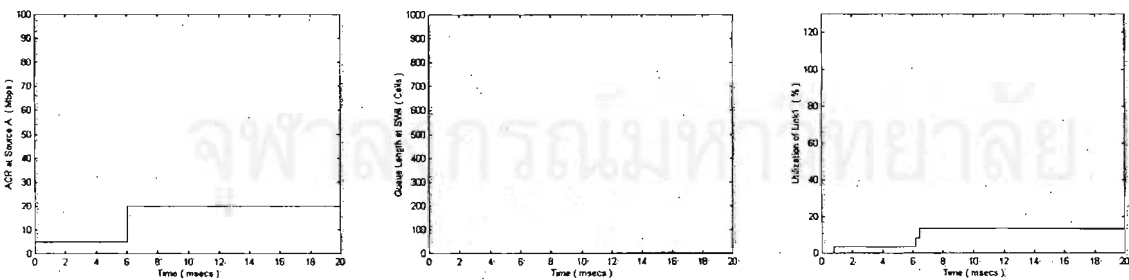
รูปที่ 4.54 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Robert Algorithm สำหรับ Jiang configuration



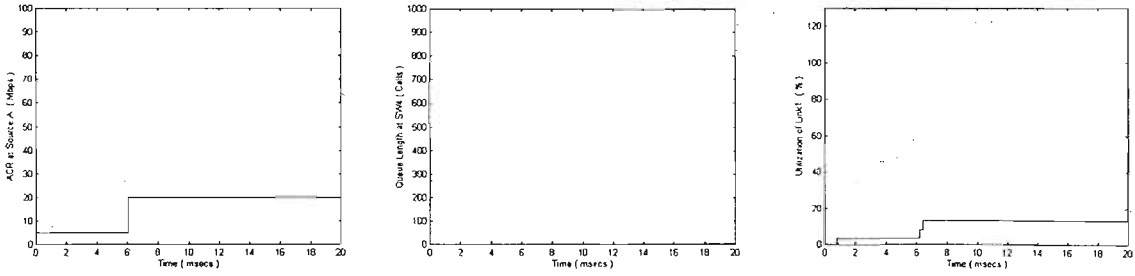
รูปที่ 4.55 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ TS Algorithm สำหรับ Jiang configuration



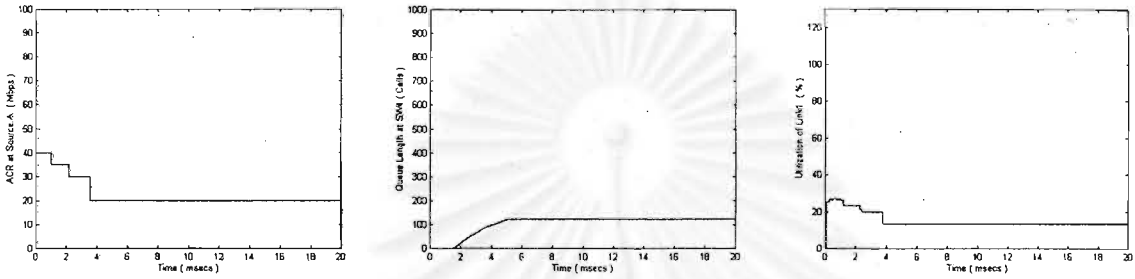
รูปที่ 4.56 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ RSS Algorithm สำหรับ Jiang configuration



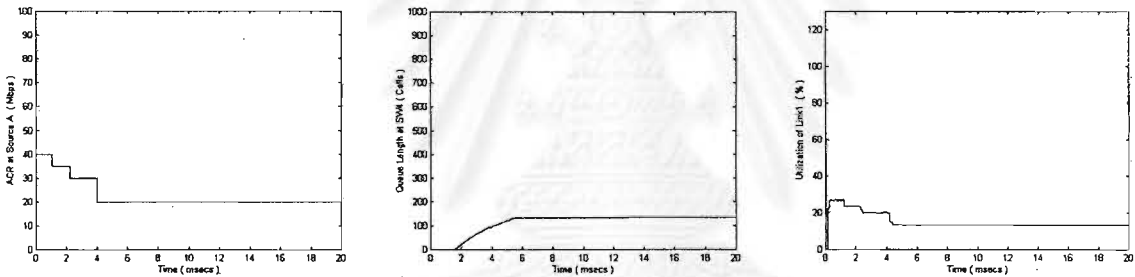
รูปที่ 4.57 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Wait-for-all Algorithm สำหรับ Jiang configuration



รูปที่ 4.58 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ Jiang configuration



รูปที่ 4.59 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ Jiang configuration



รูปที่ 4.60 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ Jiang configuration

จะเห็นว่าผลการจำลองสำหรับวิธีรวบรวมข่าวสารความคับคั่งแต่ละแบบร่วมกับ FMMRA นั้น ให้ผลที่เหมือนกับเมื่อนำมาใช้ร่วมกับ ERICA ทั้งนี้เนื่องจาก รูปแบบโครงข่ายนี้เป็นรูปแบบที่ง่ายไม่ซับซ้อน โดยมีแหล่งกำเนิดเพียงแหล่งเดียวเท่านั้น อีกทั้ง ERICA ยังมีการปรับค่า target utilization เป็น 100% ทำให้มีการใช้งานสายสัญญาณได้เต็มที่เช่นเดียวกับ FMMRA

4.4.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

ERICA และ FMMRA นั้นให้ผลการจำลองที่ไม่แตกต่างกัน เนื่องจากรูปแบบโครงข่ายนี้เป็นรูปแบบโครงข่ายที่ง่าย และ ERICA นั้นมีการตั้งค่า target utilization เป็น 100% นั่นเอง

4.4.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่างๆ

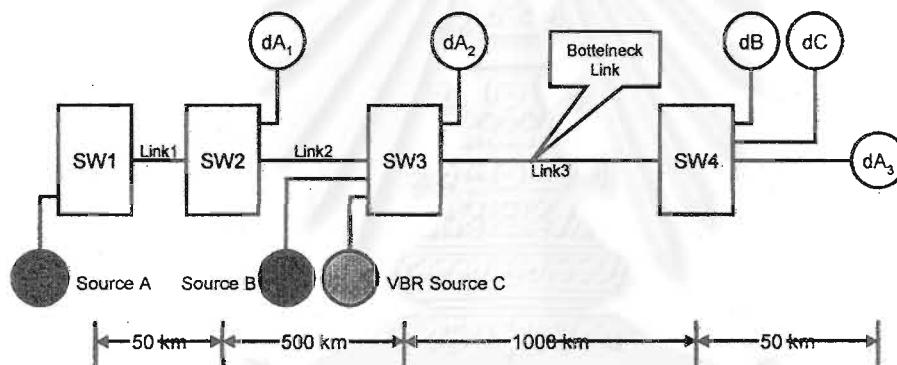
รูปแบบโครงข่ายนี้สามารถทดสอบผลตอบสนองชั่วคราวของวิธีรวบรวมข่าวสารความคับคั่งได้เป็นอย่างดี โดยวิธี Robert algorithm, RSS algorithm, วิธีการที่เสนอแบบที่ 1 และ 2 นั้นมีผลตอบสนองชั่วคราวที่รวดเร็ว โดยสามารถปรับอัตราส่งของแหล่งกำเนิดได้ตามสถานะของโครงข่าย ในขณะที่วิธี TS

algorithm นั้นสามารถปรับอัตราส่งได้ช้ากว่า ด้วยเงื่อนไขของการตอบกลับเซลล์ BRM ไม่ดีนั่นเอง ส่วน Wait-for-all และ Immediate rate calculation นั้นมีผลตอบสนองชั่วคราวที่ช้ามาก

4.5 ผลการจำลองสำหรับ VBR Chain configuration

4.5.1 รูปแบบโครงข่ายและค่ากำหนดต่าง ๆ

การส่งข้อมูลในโครงข่าย ATM จริงนั้น จะมีทราฟฟิกประเภท VBR ส่งร่วมกับทราฟฟิกประเภท ABR ด้วย ดังนั้นรูปแบบโครงข่ายนี้ไว้เพื่อทดสอบสมรรถนะภาพในการส่งข้อมูลของวิธีควบคุมความคับคั่ง และ วิธีรวบรวมข่าวสารความคับคั่ง ภายใต้สถานะที่ทราฟฟิก VBR นั้นมีการเปลี่ยนแปลงอย่างสูง อันจะทำให้แหล่งกำเนิดของ ABR นั้นต้องสามารถปรับอัตราส่งให้สอดคล้องกับทราฟฟิกที่โครงข่ายรองรับได้ โครงข่ายมีลักษณะคล้ายคลึงกับ chain configuration หากแต่เพิ่มแหล่งกำเนิด VBR ขึ้นมาเป็น source C โดย source C นี้เป็น on/off source ที่ทราฟฟิกอัตราส่งแน่นอน รูปแบบโครงข่ายจะเป็นดังรูปที่ 4.61



รูปที่ 4.61 VBR chain configuration

ค่าตัวแปรต่าง ๆ ที่ใช้ในรูปแบบโครงข่ายนี้ได้แก่

Source A : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

Source B : PCR = 149.76 Mbps , ICR = 149.76 Mbps, MCR= 0 Mbps, RIF= 1 , RDF = 0.0125 และ Nrm = 32

Source C : เป็น on/off source ที่รู้ค่าอัตราส่งแน่นอน โดยช่วงมีช่วง on เป็น 100 ms และ ช่วง off เป็น 100 ms เช่นเดียวกัน โดยในช่วง on มีอัตราส่งที่ 130 Mbps และ หยุดส่งข้อมูลในช่วง off แหล่งกำเนิดอยู่ในสถานะ ON ตั้งแต่เริ่มต้น

SW1 : Maximum queue length = 40,000

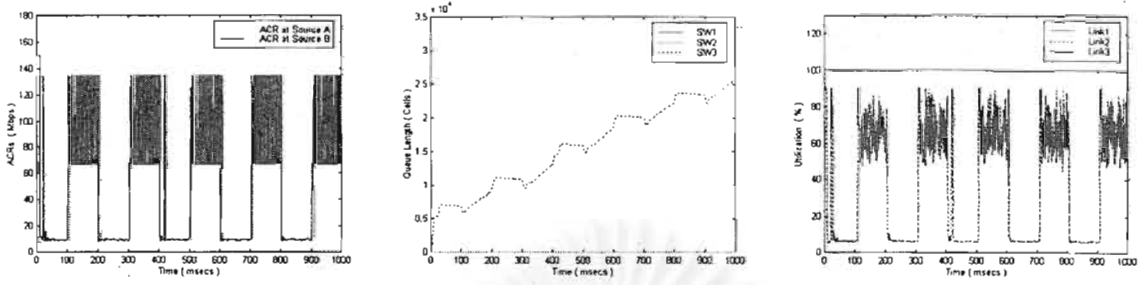
SW2 : Maximum queue length = 40,000

SW3 : Maximum queue length = 40,000

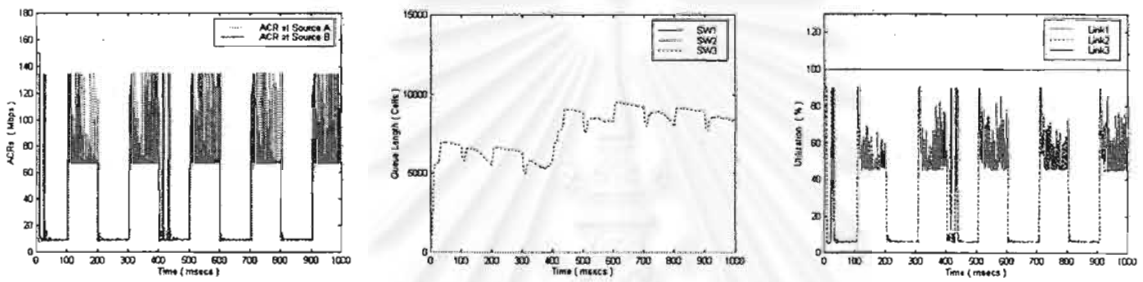
SW4 : Maximum queue length = 40,000

4.5.2 สมรรถนะของวิธี ERICA

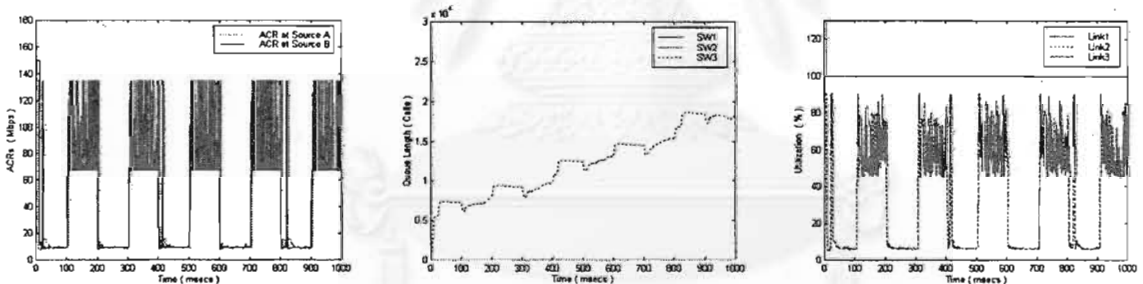
เมื่อใช้ ERICA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.62 ถึง 4.68



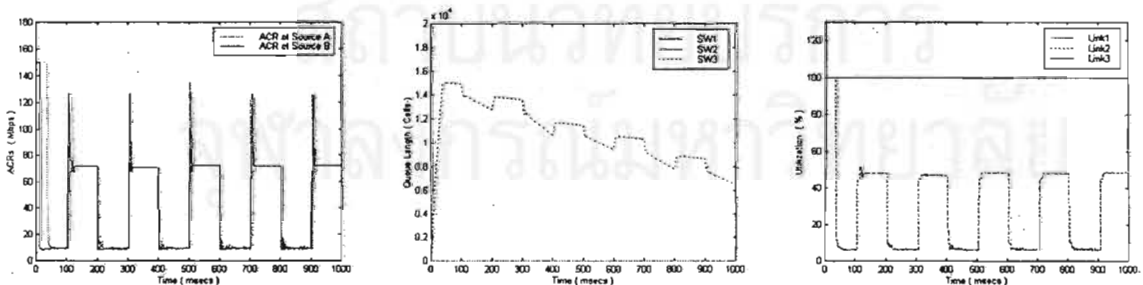
รูปที่ 4.62 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Robert Algorithm สำหรับ VBR chain configuration



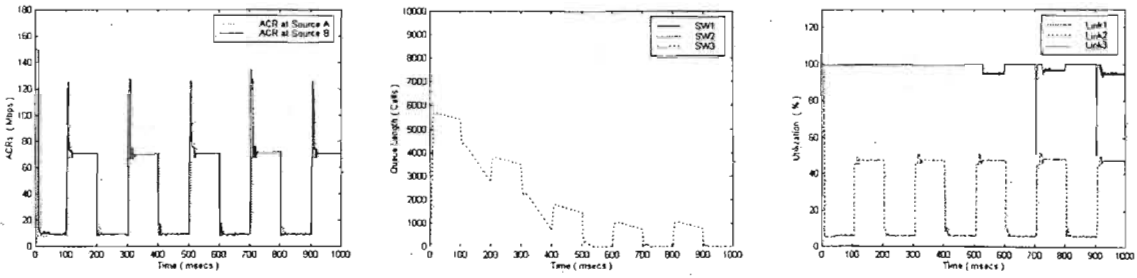
รูปที่ 4.63 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ TS Algorithm สำหรับ VBR chain configuration



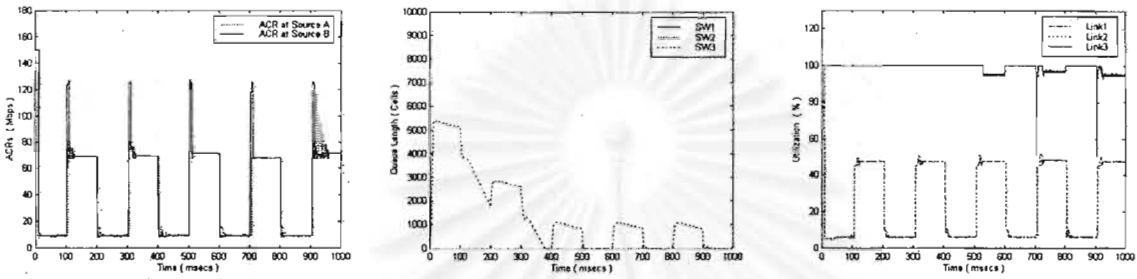
รูปที่ 4.64 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ RSS Algorithm สำหรับ VBR chain configuration



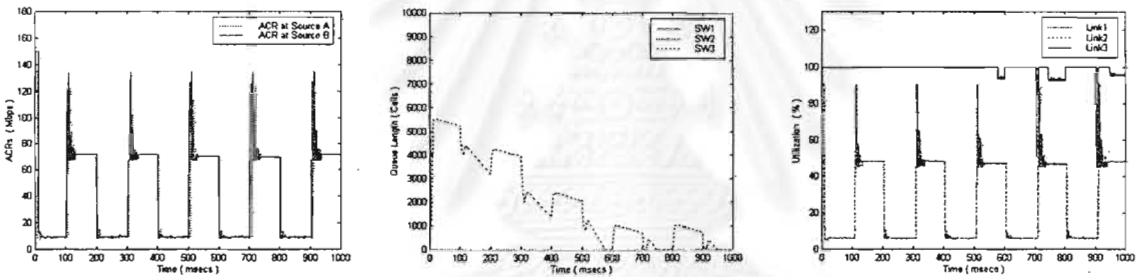
รูปที่ 4.65 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Wait-for-all Algorithm สำหรับ VBR chain configuration



รูปที่ 4.66 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ VBR chain configuration



รูปที่ 4.67 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 1 สำหรับ VBR chain configuration



รูปที่ 4.68 ผลการจำลองเมื่อใช้วิธี ERICA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ VBR chain configuration

เนื่องจากโครงข่ายนี้มี VBR ที่เป็น on/off source อยู่ด้วย ทำให้ค่า fairshare สำหรับแต่ละทุกแหล่งกำเนิดสามารถแบ่งได้เป็นสองสถานะ โดยในสถานะที่แหล่งกำเนิด VBR นั้น on อยู่ นั้น ค่า fairshare จะเป็น 9.88 Mbps ส่วนในสถานะ off จะมีค่าเป็น 74.88 Mbps อย่างไรก็ตามเนื่องจาก ERICA มีการทำงานโดยใช้ค่า z ที่อยู่ระหว่าง 1-1.1 ดังนั้นอัตราส่งที่แหล่งกำเนิดได้รับอนุญาตจริงจะมีค่าสูงกว่าค่าอัตราส่งเท่าเทียมกันเล็กน้อย

เมื่อพิจารณาค่า ACR ของแหล่งกำเนิด A ในรูปที่ 4.62 จะเห็นว่าผลตอบของ Robert algorithm นั้นมีผลตอบชั่วคราวที่รวดเร็วมาก โดยสามารถปรับค่า ACR ให้ลดลงสู่ fairshare ได้อย่างรวดเร็ว อย่างไรก็ตามวิธีการนี้ยังก่อให้เกิด consolidation noise อย่างหนัก โดยเฉพาะในสถานะที่แหล่งกำเนิด VBR นั้นเปลี่ยนสถานะจากช่วง ON ไปสู่ช่วง OFF ซึ่งจะทำให้แหล่งกำเนิด ABR ทั้งสองนั้นสามารถปรับอัตราส่งของตนเองขึ้นมาได้ จะเห็นว่า ERICA นั้นสามารถปรับอัตราส่งของแหล่งกำเนิดไปสู่ค่า fairshare ได้อย่างรวดเร็ว การปรับค่าอัตราส่งของแหล่งกำเนิดขึ้นนี้เองที่เป็นสาเหตุของการเกิด consolidation noise ในลักษณะที่คล้ายคลึงกับในกรณี Chain configuration เมื่อแหล่งกำเนิดเพิ่มอัตราส่งขึ้นอย่างรวดเร็วจะ

ทำให้ปริมาณเซลล์ FRM ที่ส่งออกจากแหล่งกำเนิดในช่วงเวลาหนึ่งเพิ่มขึ้นด้วย และทำให้สวิตช์นั้นได้รับเซลล์ FRM เพิ่มขึ้น ในขณะที่จำนวนเซลล์ BRM ที่ได้รับนั้นมีปริมาณเท่าเดิม แต่เนื่องจาก Robert algorithm นั้นใช้เงื่อนไขในการตอบกลับเซลล์ BRM ด้วยการตรวจสอบจากการรับเซลล์ FRM และจะทำการลบค่า MER ที่รวบรวมไว้ให้เป็นค่าที่ได้รับจาก FRM ซึ่งมักจะเป็นค่า PCR ที่มีค่าสูงนั่นเอง ด้วยเหตุนี้สวิตช์นั้นต้องตอบกลับเซลล์ BRM มากขึ้น และต้องลบค่า MER บ่อยขึ้น ในขณะที่ปริมาณข่าวสารการป้อนกลับที่ได้รับนั้นมีปริมาณเท่าเดิม ทำให้เกิดความผิดพลาดในการรวบรวมข้อมูลในตัวแปร MER นั่นเอง และก่อให้เกิดการแกว่งตัวของค่า ACR ที่แหล่งกำเนิด A ขึ้น Consolidation noise ที่เกิดขึ้นนั้นทำให้ขนาดหน่วยความจำที่ SW3 นั้นเพิ่มขึ้นในระดับที่ไม่สามารถควบคุมได้

ผลการจำลองของ TS algorithm ดังรูปที่ 4.63 สังเกตได้ว่าผลตอบชั่วคราวนั้นมีความรวดเร็วเช่นเดียวกับ Robert algorithm แต่ยังคงเกิด consolidation noise ขึ้น แต่อยู่ในระดับที่น้อยกว่า Robert algorithm เนื่องจากวิธีดังกล่าวตรวจสอบการได้รับเซลล์ BRM ก่อนการส่งเซลล์ BRM กลับสู่แหล่งกำเนิดทันทีหลังจากที่ได้รับเซลล์ FRM ซึ่งสามารถลด consolidation noise ได้ แต่อย่างไรก็ตาม consolidation noise ที่เกิดก่อให้เกิดความคับคั่งอย่างรุนแรงในระดับที่ยอมรับไม่ได้ สังเกตได้จากขนาดหน่วยความจำที่ SW3 มีการแกว่งตัวตลอด และมีแนวโน้มที่จะเพิ่มขึ้นเรื่อย ๆ

RSS algorithm ตามรูปที่ 4.64 นั้น ยังคงเกิด consolidation noise ในช่วงที่แหล่งกำเนิด VBR นั้น OFF อยู่ โดยระดับการเกิด consolidation noise นั้นจะมีเท่า ๆ กับ Robert algorithm ส่วนสาเหตุการเกิด consolidation noise นั้นก็เหมือนกับที่เกิดใน Robert algorithm การเกิด consolidation noise ขึ้นนั้นทำให้หน่วยความจำที่ SW3 มีปริมาณเพิ่มขึ้นตลอด แสดงให้เห็นถึงความคับคั่งที่เกิดขึ้นในโครงข่ายส่วน Wait-for-all ตามรูปที่ 4.65 นั้นสามารถขจัด consolidation noise ไปได้ แต่ก็มีผลตอบสนองชั่วคราวที่ช้ามาก ซึ่งสังเกตได้จากค่า ACR ของแหล่งกำเนิด A ซึ่งปรับตัวช้ากว่า Robert algorithm เนื่องจากแหล่งกำเนิดต้องรอให้เซลล์ BRM เดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อน การปรับตัวของค่า ACR ที่เข้าไปนี้เองทำให้โครงข่ายอยู่ในสภาวะติดขัดเป็นเวลานาน ส่งผลให้ขนาดหน่วยความจำที่ SW3 นั้นเพิ่มสูงขึ้นมาก

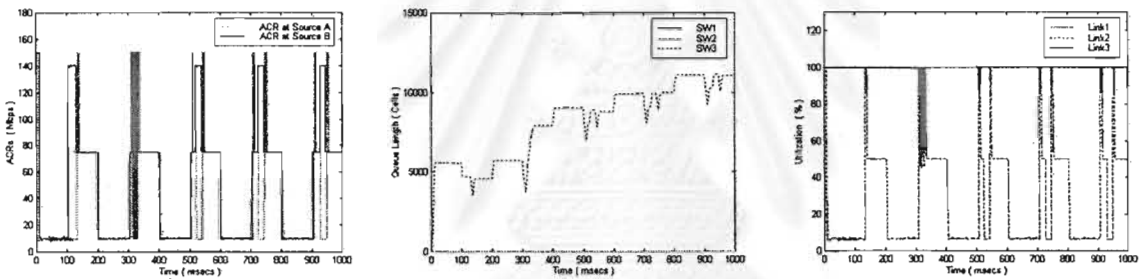
จากรูปที่ 4.66 จะเห็นว่า Immediate rate calculation algorithm นั้นสามารถขจัด consolidation noise ไปได้ ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็ว เนื่องจากโครงข่ายอยู่ในสภาวะโหลดเกิน ทำให้กลไก overload detected ของ Immediate rate calculation algorithm นั้นทำงาน โดยการส่งเซลล์ BRM กลับไปสู่แหล่งกำเนิดทันที ส่วน Consolidation noise ถูกจำกัดไปได้เนื่องจากวิธีนี้ใช้หลักการเดียวกับ Wait-for-all algorithm โดยการรวบรวมข่าวสารความคับคั่งให้ครบจากทุก branch ก่อน

วิธีการที่เสนอแบบที่ 1 นั้นสามารถขจัด consolidation noise ได้เช่นเดียวกัน ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็ว ด้วยข้อดีที่เก็บรวบรวมข่าวสารความคับคั่งจากทุก branch ไว้ ทำให้สามารถขจัด consolidation noise ไปได้ และการใช้เงื่อนไขในการตอบกลับเซลล์ BRM ที่เหมือนกับที่ใช้ใน Robert

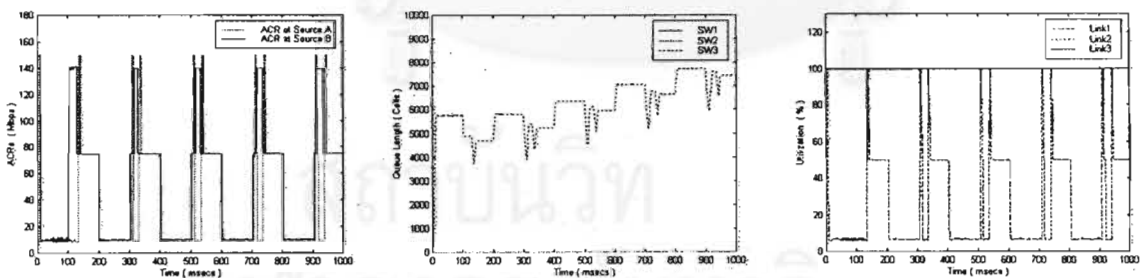
algorithm ทำให้มีผลตอบชั่วคราวที่รวดเร็ว ส่งผลให้ขนาดหน่วยความจำที่ SW3 นั้นมีค่าไม่มากเท่าไรนัก ในขณะที่การใช้งานสายสัญญาณนั้นมีประสิทธิภาพเต็มที่ ส่วนวิธีการที่เสนอแบบที่ 2 ตามรูปที่ 4.68 นั้นสามารถลดการเกิด consolidation noise ให้อยู่ในระดับที่น้อยมาก และสามารถยอมรับได้สังเกตได้จากค่า ACR ของแหล่งกำเนิด A ในระหว่างที่แหล่งกำเนิด VBR นั้นปรับการสถานะ ON ไปสู่ OFF นั้น จะเกิด consolidation noise ในช่วงแรก แต่ระดับการเกิด consolidation noise จะค่อย ๆ น้อยลงเนื่องจากวิธีการนี้ไม่ทำการลบค่า MER ที่เก็บรวบรวมไว้ให้มีค่าเป็นค่า PCR นั้นเอง หากแต่ตั้งให้เป็นค่า ER ที่เพิ่งได้รับจากเซลล์ BRM นั้นเอง วิธีการนี้ยังไม่สามารถขจัด consolidation noise ไปได้ แต่ก็ป้องกัน consolidation noise ที่เกิดขึ้นนั้นไม่ให้เกิดต่อเนื่องไปเรื่อย ๆ จนโครงข่ายเกิดความคับคั่งมากเกินไป จะเห็นว่าวิธีการนี้สามารถควบคุม consolidation noise ให้อยู่ในระดับที่สามารถควบคุมได้ โดยที่ขนาดหน่วยความจำที่สวิตช์นั้นมีค่าไม่มากนัก ในขณะที่การใช้งานสายสัญญาณนั้นมีประสิทธิภาพดี

4.5.3 สมรรถนะของวิธี FMMRA

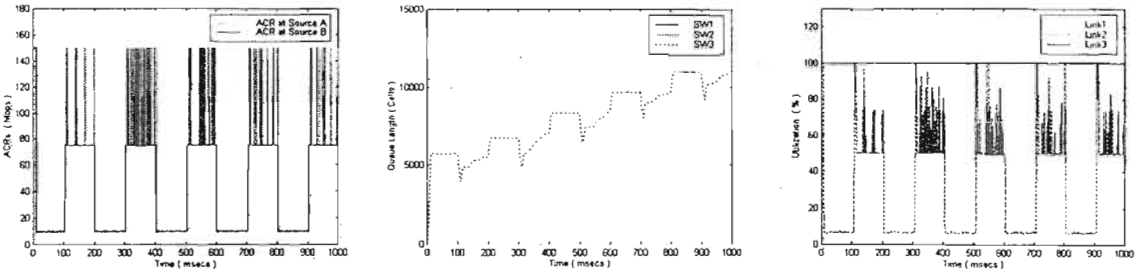
เมื่อใช้ FMMRA ร่วมกับวิธีรวบรวมความคับคั่งต่าง ๆ จะได้ผลดังรูปที่ 4.69 ถึง 4.75



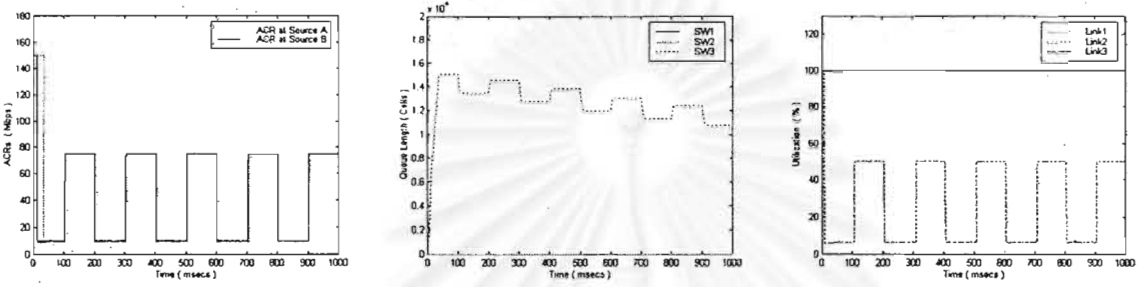
รูปที่ 4.69 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Robert Algorithm สำหรับ VBR chain configuration



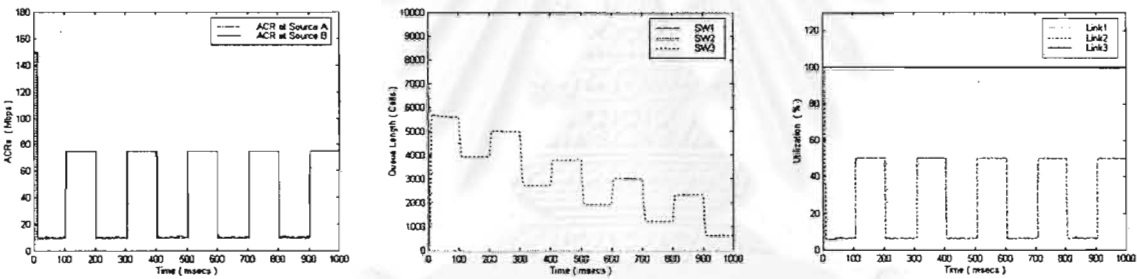
รูปที่ 4.70 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ TS Algorithm สำหรับ VBR chain configuration



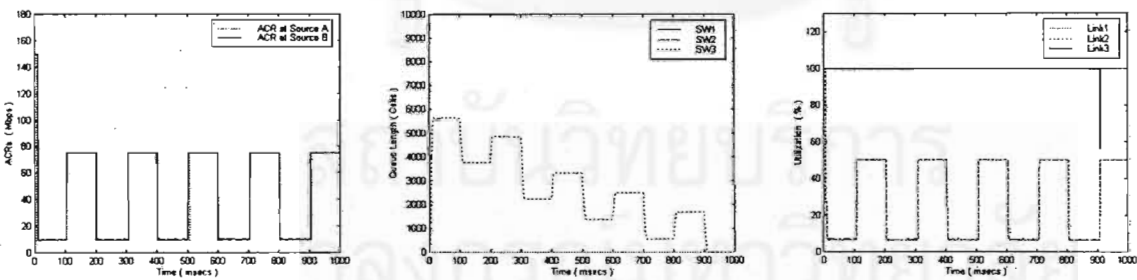
รูปที่ 4.71 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ RSS Algorithm สำหรับ VBR chain configuration



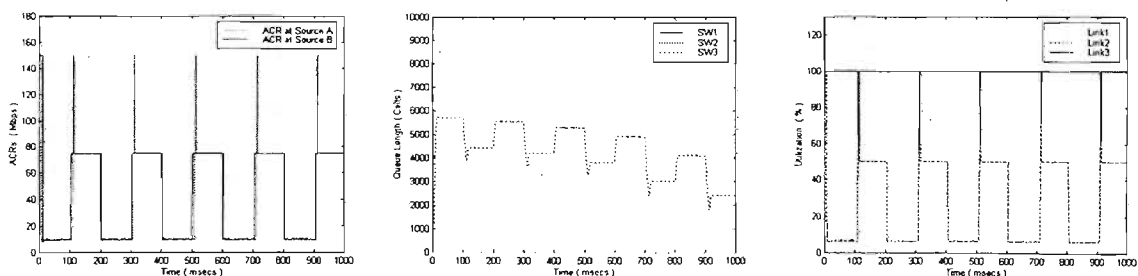
รูปที่ 4.72 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Wait-for-all Algorithm สำหรับ VBR chain configuration



รูปที่ 4.73 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับ Immediate Rate Calculation Algorithm สำหรับ VBR chain configuration



รูปที่ 4.74 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอบนที่ 1 สำหรับ VBR chain configuration



รูปที่ 4.75 ผลการจำลองเมื่อใช้วิธี FMMRA ร่วมกับวิธีการที่เสนอแบบที่ 2 สำหรับ VBR chain configuration

จากรูปที่ 4.59 สังเกตได้ว่าเมื่อนำ Robert algorithm มาใช้งานร่วมกับวิธี FMMRA นั้น จะเกิด consolidation noise ขึ้น โดยเฉพาะเมื่อ VBR source เปลี่ยนจากสถานะ ON ไปสู่สถานะ OFF นอกจากนี้ยังทำให้ FMMRA เกิดความผิดพลาดในการคำนวณค่า ER ที่แบ่งให้กับแต่ละการเชื่อมต่อในโครงข่าย สังเกตได้จากค่าอัตราส่งของแหล่งกำเนิด A นั้นมีการแกว่งตัวในลักษณะผิดปกติ โดยมีแหล่งกำเนิด B ซึ่งเป็น point-to-multipoint connection นั้นสามารถปรับอัตราส่งได้มากกว่าปกติ ส่วนแหล่งกำเนิด A ซึ่งเป็น point-to-multipoint connection นั้นไม่สามารถปรับอัตราส่งเพิ่มขึ้นได้ ทั้งนี้เนื่องจาก FMMRA นั้นมีการคำนวณค่า Bottleneck status ผิดพลาดอันเนื่องมาจากการรวบรวมข่าวสารความคับคั่งที่ผิดพลาดนั่นเอง ส่วนสาเหตุที่ทำให้เกิด consolidation noise และความผิดพลาดของข่าวสารความคับคั่งที่รวบรวมไว้ที่สวิตช์นั้น คือสาเหตุเช่นเดียวกับในกรณีที่ใช้ ERICA นั้นเอง หากแต่ ERICA นั้นคำนวณค่า ER จากค่า Fairshare และ ค่า Vcshare ที่เก็บไว้นั่นเอง ส่วนในกรณีของ FMMRA นั้น มีการนำค่า ER ที่ได้จากเซลล์ RM มามีส่วนในการคำนวณหาค่า ER ที่จะทำการป้อนกลับ วิธี Robert algorithm นี้ต้องมีการสร้างเซลล์ RM ขึ้นมา ดังนั้นค่า ER ในเซลล์ RM ที่สร้างขึ้นนี้ก็คือค่า MER ที่เก็บรวบรวมไว้จากการรับเซลล์ BRM จาก branch ต่าง ๆ นั่นเอง ด้วยเหตุนี้หากค่า ER ที่ได้จากเซลล์ RM นั้นผิดพลาดจากการรวบรวมข่าวสารความคับคั่งที่ไม่ดีพอ ก็จะทำให้ FMMRA นั้นคำนวณค่า ER ผิดพลาดตามไปด้วย

ความผิดพลาดในการคำนวณของ FMMRA และการเกิด consolidation noise นั้น ทำให้โครงข่ายไม่อยู่ในสถานะอยู่ตัว และ ค่าอัตราส่งที่แหล่งกำเนิดนั้นเกิดการแกว่งตัวอย่างรุนแรง ซึ่งนำไปสู่ความคับคั่งของโครงข่าย ดังสังเกตได้จากขนาดหน่วยความจำที่สวิตช์ที่เพิ่มขึ้นเรื่อย ๆ จนอยู่ในระดับที่ไม่สามารถควบคุมได้ ในขณะที่การใช้งานสายสัญญาณนั้นก็มีการแกว่งตัวในลักษณะคล้าย ๆ กับค่า ACR ที่แหล่งกำเนิดนั่นเอง

สำหรับ TS algorithm ในรูปที่ 4.70 นั้นมีผลการจำลองที่คล้ายคลึงกับ กรณีของ Robert algorithm โดยที่ FMMRA นั้นยังคงคำนวณค่า ER ที่ผิดพลาด และยังคงเกิด consolidation noise ขึ้น เนื่องจากวิธีนี้มีการสร้างเซลล์ BRM ก่อนการตอบกลับ แล้วนำค่า MER บรรจุเป็นค่า ER ในเซลล์ RM ก่อนที่จะให้ FMMRA คำนวณค่า ER ที่ควรจะเป็น ซึ่งหากค่า MER ที่รวบรวมไว้นั้นมีความผิดพลาดจาก consolidation noise ก็จะทำให้ FMMRA นั้นมีการคำนวณค่า ER ที่ผิดพลาดตามไปด้วย อย่างไรก็ตาม

การแกว่งตัวของค่า ACR ที่แสดงถึง consolidation noise นั้นมีน้อยลง ซึ่งมีสาเหตุจากการที่ TS algorithm ตรวจสอบการรับเซลล์ BRM ก่อนการสร้างเซลล์ BRM เพื่อการตอบกลับนั่นเอง อย่างไรก็ตามการแกว่งตัวของค่า ACR รวมถึงความผิดพลาดในการคำนวณค่า ER ของ FMMRA นั้นทำให้โครงข่ายเกิดความคับคั่งขึ้น ซึ่งสังเกตได้จากขนาดหน่วยความจำที่สวิตช์ที่เพิ่มขึ้นเรื่อยๆ จนอยู่ในระดับที่ไม่สามารถควบคุมได้

RSS algorithm ตามรูปที่ 4.71 นั้น จะเห็นว่าเกิด Consolidation noise ขึ้นในสถานะที่แหล่งกำเนิด VBR นั้น OFF เช่นเดียวกับ Robert algorithm หากแต่ไม่เกิดความผิดพลาดในการคำนวณจาก FMMRA algorithm ทั้งนี้เนื่องจาก วิธีการนี้ไม่มีการสร้างเซลล์ BRM ในการตอบกลับ หากแต่นำเซลล์ BRM ที่ได้รับตัวแรกหลังจากได้รับเซลล์ FRM มาใช้ในการตอบกลับ ทำให้สามารถใช้ค่า ER ในเซลล์ BRM ที่ได้รับนั้นมาคำนวณหาค่า ER โดยใช้ FMMRA ก่อนนำไปเปรียบเทียบกับค่า MER ข้อมูลที่ FMMRA ใช้ในการคำนวณนั้นไปค่าที่ได้รับจากโครงข่ายจริง ทำให้ไม่เกิดความผิดพลาดในการคำนวณค่า ER อย่างไรก็ตาม วิธีการนี้ยังคงเกิด consolidation noise ขึ้นด้วยสาเหตุเช่นเดียวกับในกรณีเมื่อใช้ ERICA ซึ่งการเกิด consolidation noise นี้เอง ทำให้โครงข่ายอยู่ในสถานะคับคั่ง สังเกตได้จากขนาดหน่วยความจำที่มีค่าเพิ่มขึ้นเรื่อยๆ นั่นเอง

ผลการจำลอง ของ Wait-for-all algorithm นั้น แสดงในรูปที่ 4.72 ซึ่งจะเป็นว่าวิธีนี้สามารถขจัด consolidation noise ไปได้ เนื่องจากวิธีการนี้ต้องรวบรวมข่าวสารให้ครบถ้วนก่อนการป้อนกลับเซลล์ BRM นั่นเอง อย่างไรก็ตามวิธีการนี้จะมีผลตอบสนองชั่วคราวที่ช้ามาก โดยโครงข่ายในช่วงแรกนั้นอยู่ในสถานะคับคั่ง โดยแหล่งกำเนิดสมควรลดอัตราส่งมาที่ fairshare โดยเร็วที่สุด วิธีการ Wait-for-all algorithm นั้นไม่สามารถปรับอัตราส่งให้ลดลงมาที่ fairshare ได้อย่างรวดเร็ว เนื่องจากแหล่งกำเนิดนั้นต้องรอให้เซลล์ BRM เดินทางกลับมาจากปลายทางที่ไกลที่สุดก่อน ด้วยเหตุนี้โครงข่ายจึงอยู่ในสถานะคับคั่งเป็นเวลานานในช่วงเริ่มต้น ทำให้ขนาดหน่วยความจำที่สวิตช์นั้นมีค่ามากตามไปด้วย ส่วนการใช้งานสายสัญญาณนั้นก็มิมีประสิทธิภาพดีตามที่สมควรได้รับ

ผลการจำลองของวิธี Immediate rate calculation ตามรูปที่ 4.73 และ วิธีการที่เสนอแบบที่ 1 ตามรูปที่ 4.74 นั้นให้ผลที่คล้ายคลึงกันมาก โดยสามารถขจัด consolidation noise ไปได้ ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็ว วิธี Immediate rate calculation นั้นมีการทำงานที่คล้ายคลึงกับ Wait-for-all algorithm ทำให้สามารถขจัด consolidation noise ไปได้ และ โครงข่ายอยู่ในสถานะไหลลื่นตั้งแต่เริ่มต้น ทำให้เทคนิค overload detect ของวิธีการนี้เกิดผล ทำให้มีการส่งเซลล์ BRM กลับไปที่ต้นทาง และทำให้แหล่งกำเนิดปรับอัตราส่งได้ทันทั่วทั้ง ส่วนวิธีการที่เสนอแบบที่ 1 นั้น ถึงแม้จะมีการสร้างเซลล์ BRM เพื่อการตอบกลับเช่นเดียวกับ Robert algorithm แต่ด้วยความถูกต้องของข้อมูลที่เก็บรวบรวมไว้นั้น ทำให้วิธีการนี้ไม่ก่อให้เกิด consolidation noise และไม่ทำให้เกิดความผิดพลาดในการคำนวณค่า ER ของวิธี FMMRA ด้วย ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็วเช่นเดียวกับ Robert algorithm ซึ่ง

จะเห็นว่าวิธีการ Immediate rate calculation และ วิธีการที่เสนอแบบที่ 1 นั้นทำให้แหล่งกำเนิดสามารถปรับอัตราส่งให้เหมาะสมกับโครงข่ายได้เป็นอย่างดี ทำให้โครงข่ายไม่อยู่ในสถานะคับคั่ง ส่งผลให้ขนาดหน่วยความจำสูงสุดที่สวิตช์มีค่าไม่มากนัก ส่วนการใช้งานสายสัญญาณทุกเส้นทางนั้นสามารถทำได้อย่างมีประสิทธิภาพที่สุด

สังเกตจากรูปที่ 4.75 จะเห็นว่าผลการจำลองสำหรับวิธีการที่เสนอแบบที่ 2 เมื่อใช้งานร่วมกับวิธี FMMRA นั้น ยังคงมี consolidation noise ในช่วงแรกที่แหล่งกำเนิด VBR นั้นเปลี่ยนจากสถานะ ON ไปสู่สถานะ OFF โดย consolidation noise ที่เกิดขึ้นนั้น เกิดขึ้นเฉพาะในเริ่มต้นการเปลี่ยนสถานะเท่านั้น และวิธีนี้สามารถควบคุม consolidation noise ให้อยู่ในระดับที่สามารถควบคุมได้ในที่สุด ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็วมาก ซึ่งสังเกตได้จากการปรับค่าอัตราส่งของแหล่งกำเนิดที่สามารถปรับสู่ fairshare ได้อย่างรวดเร็วในช่วงแรก ทำให้โครงข่ายไม่อยู่ในสถานะคับคั่ง และขนาดหน่วยความจำสูงสุดที่สวิตช์นั้นมีค่าน้อย ในขณะที่การใช้งานสายสัญญาณทุกเส้นทางนั้นมีประสิทธิภาพดี

4.5.4 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

วิธี ERICA และ FMMRA นั้นสามารถตอบสนองต่อปริมาณทราฟฟิกที่เปลี่ยนแปลงไปของแหล่งกำเนิด VBR โดยสามารถปรับอัตราส่งของแหล่งกำเนิดไปสู่ fairshare ได้อย่างรวดเร็ว แต่วิธี FMMRA นั้นสามารถปรับอัตราส่งได้รวดเร็วกว่าวิธีของ ERICA หากแต่ยังมีความผิดพลาดในการคำนวณค่า ER ในกรณีเมื่อใช้วิธีรวบรวมข่าวสารความคับคั่งคือ Robert algorithm และ TS algorithm

4.5.5 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ

Robert algorithm, TS algorithm และ RSS algorithm ยังคงมีปัญหาในการเกิด consolidation noise อย่างหนัก และนำไปสู่การคำนวณค่า ER ที่ผิดพลาดเมื่อใช้วิธี FMMRA ร่วมกับ Robert algorithm และ TS algorithm ในขณะที่วิธีทั้งสามนั้นมีผลตอบสนองชั่วคราวที่ดีมาก ซึ่งทำให้โครงข่ายสามารถหลุดจากสถานะคับคั่งในช่วงแรกได้อย่างรวดเร็ว วิธี wait-for-all algorithm นั้นสามารถขจัด consolidation noise ไปได้ แต่มีปัญหาเรื่องผลตอบสนองชั่วคราวที่ช้ามาก ในขณะที่วิธี Immediate rate calculation algorithm และ วิธีการที่เสนอแบบที่ 1 นั้นให้สมรรถนะที่ดีที่สุด โดยสามารถขจัด consolidation noise ไปได้ ในขณะที่มีผลตอบสนองชั่วคราวที่รวดเร็วมาก วิธีการที่เสนอแบบที่ 2 นั้นมีผลตอบสนองชั่วคราวที่รวดเร็วเช่นกัน แต่ยังไม่สามารถขจัด consolidation noise ให้หมดไปอย่างสิ้นเชิงได้ แต่สามารถลด consolidation noise ให้น้อยมากจนอยู่ในระดับที่สามารถรับได้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

วิธี ERICA และ FMMRA นั้นมีข้อดีและข้อเสียแตกต่างกันไป โดยเฉพาะเมื่อนำมาใช้ในการส่งจากหนึ่งจุดไปสู่หลายจุด โดยหัวข้อที่เราสนใจเปรียบเทียบประสิทธิภาพระหว่างสองวิธีนี้เมื่อนำมาใช้ในการส่งจากหนึ่งจุดไปสู่หลายจุด ได้แก่ Max-Min fairness, ผลตอบสนองชั่วคราว, Consolidation noise, ความซับซ้อนในการพัฒนา และความเหมาะสมในการใช้งานกับวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ

จากผลการจำลองสามารถสรุปได้ว่า วิธีควบคุมความคับคั่งนั้นมีผลต่อการทำงานของวิธีรวบรวมข่าวสารความคับคั่ง โดยที่วิธี FMMRA นั้นสามารถลดการเกิด consolidation noise ได้ในบางกรณีเมื่อเทียบกับวิธี ERICA ที่ยังคงมี consolidation noise ในสถานะการณ์เดียวกัน

5.1.1 Max-Min fairness

ในการศึกษาที่ผ่านมาได้มีการศึกษาด้าน Max-Min fairness ในการส่งจากหนึ่งจุดไปสู่หลายจุด ซึ่งผลการวิเคราะห์ที่แสดงว่า Max-Min fairness นั้นสามารถนำมาใช้ในการส่งจากหนึ่งจุดไปสู่หลายจุดได้ จากการจำลองที่ผ่านมาได้แสดงให้เห็นถึงความสามารถในการปรับอัตราส่งให้เข้าสู่ Max-Min fairness ของทั้งวิธี ERICA และ FMMRA ได้เป็นอย่างดี

5.1.2 ผลตอบสนองชั่วคราว

ในการส่งแบบหนึ่งจุดไปสู่หลายจุดนั้น วิธี FMMRA นั้นสามารถปรับอัตราส่งของแหล่งกำเนิดให้เข้าสู่ค่า fairshare ได้อย่างรวดเร็วกว่าวิธี ERICA ทำให้ FMMRA นั้นมีผลตอบสนองชั่วคราวที่รวดเร็วกว่าวิธี ERICA อย่างไรก็ตาม วิธี ERICA นั้นมีผลตอบสนองชั่วคราวที่รวดเร็วเพียงพอต่อปรับอัตราส่งของแหล่งกำเนิดให้เข้ากับสถานะของโครงข่ายได้เป็นอย่างดี ถึงแม้อยู่ในสถานะที่เลวร้ายที่สุด

5.1.3 Consolidation noise

วิธี FMMRA นั้นไม่มีปัญหาเรื่อง Consolidation noise มากนัก เนื่องจากวิธี FMMRA นั้นสามารถปรับอัตราส่งที่แหล่งกำเนิดให้เข้าสู่ fairshare ได้อย่างรวดเร็ว และถูกต้อง ในขณะที่ ERICA นั้นเกิดปัญหา Consolidation noise ขึ้น โดยเฉพาะเมื่อใช้งานกับ Robert algorithm, TS algorithm และ RSS algorithm ทั้งนี้เนื่องจาก ERICA มีการปรับอัตราส่งเพิ่มขึ้นจาก fairshare เล็กน้อย ทำให้เกิด consolidation noise ขึ้น

Consolidation noise ที่เกิดขึ้นนั้นจะเกิดขึ้นเฉพาะในช่วงที่อัตราส่งมีการปรับอัตราส่งเพิ่มขึ้นเท่านั้น ดังแสดงได้ใน VBR Chain Configuration ซึ่งจะเห็นว่าวิธี FMMRA นั้นยังคงมีปัญหา

Consolidation noise เช่นเดียวกับวิธี ERICA เนื่องจากมีการปรับอัตราส่งเพิ่มขึ้นเนื่องจากแหล่งกำเนิด VBR หยุดส่งไป ทำให้แหล่งกำเนิดแบบ ABR สามารถเพิ่มอัตราส่งได้ และทำให้เกิด Consolidation noise ขึ้น

5.1.4 ความซับซ้อนในการพัฒนา

วิธี ERICA และ FMMRA นั้นมีความซับซ้อนในการคำนวณระดับ $O(1)$ ทั้งคู่ โดยการคำนวณทุกครั้งจะทำการคำนวณรอบเดียวสำหรับการเชื่อมต่อนั้นเท่านั้น ทำให้วิธีการทั้งสองมีความซับซ้อนในระดับที่ใกล้เคียงกัน อย่างไรก็ตามวิธี FMMRA นั้นไม่สามารถนำมาใช้งานบนการส่งจากหนึ่งจุดไปสู่หลายจุดได้ทันที เนื่องจากตัวแปรต่าง ๆ ที่วิธี FMMRA ใช้นั้นเป็นตัวแปรเฉพาะการเชื่อมต่อ โดยแต่ละการเชื่อมต่ออาจมีเส้นทางขาออกมากกว่าหนึ่งเส้นทาง ทำให้สถานะของการเชื่อมต่อที่เก็บไว้ไม่เพียงพอต่อการคำนวณสำหรับหลายเส้นทางขาออก จึงจำเป็นต้องมีการเพิ่มตัวแปรต่าง ๆ สำหรับทุกการเชื่อมต่อให้เพิ่มขนาดเป็นเก็บทุกเส้นทางขาออกด้วย ทำให้ FMMRA นั้นมีความซับซ้อนมากขึ้น และมากกว่าวิธี ERICA

5.1.5 ความเหมาะสมในการใช้งานกับวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ

วิธี ERICA นั้นสามารถทำงานร่วมกับวิธีการต่าง ๆ ได้เป็นอย่างดี โดยไม่เกิดปัญหาในขบวนการคำนวณแต่อย่างใดแต่อาจเกิดปัญหา ในขณะที่วิธี FMMRA นั้นจะเกิดปัญหาในการคำนวณขึ้นโดยเฉพาะใน VBR chain configuration โดยเมื่อใช้งานร่วมกับ Robert algorithm และ TS algorithm จะเกิดความผิดพลาดในการตรวจสอบสถานะความคับคั่งของการเชื่อมต่อ อย่างไรก็ตามความผิดพลาดที่เกิดขึ้นนั้น มีสาเหตุมาจากความผิดพลาดในการคำนวณค่าการป้อนกลับของวิธีรวบรวมข่าวสารความคับคั่งด้วย

ตารางที่ 5.1 เปรียบเทียบประสิทธิภาพระหว่าง ERICA และ FMMRA

Algorithms	ERICA	FMMRA
Provide Max-Min fairness	Yes	Yes
ผลตอบสนองชั่วคราว	Fast	Very Fast
Consolidation noise	High	Low
ความซับซ้อน	Low	Medium
ความเหมาะสมในการใช้งานกับวิธีรวบรวมข่าวสารความคับคั่งต่าง ๆ	Good	Fair



5.2 เปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่ง

ในการเปรียบเทียบประสิทธิภาพระหว่างวิธีรวบรวมข่าวสารความคับคั่งนั้น จะสนใจในเรื่อง Consolidation noise, ผลตอบสนองชั่วคราว, ความซับซ้อนในการพัฒนา, RM cell overhead, Change to non responsive branch และ Change to newly joined branch

5.2.1 Consolidation noise

วิธี Robert algorithm, TS algorithm และ RSS algorithm นั้นยังคงมีปัญหาเรื่อง consolidation noise ในระดับที่ไม่สามารถรับได้เมื่อใช้งานกับวิธี ERICA ถึงแม้ว่าเมื่อนำมาใช้งานกับวิธี FMMRA นั้นจะสามารถขจัด consolidation noise ได้ก็ตาม อย่างไรก็ตาม การเกิด consolidation noise นั้นเกิดได้ในเฉพาะบางกรณีเท่านั้น โดยเฉพาะเมื่อค่า RIF มีค่าสูง ทั้งนี้เนื่องจากเมื่อค่า RIF มีค่าสูงจะทำให้แหล่งกำเนิดนั้นสามารถเพิ่มอัตราส่งขึ้นได้อย่างรวดเร็ว อันจะทำให้เกิดการแกว่งตัวอย่างรุนแรงของค่า ACR และก่อให้เกิด consolidation noise ขึ้น อย่างไรก็ตามหากค่า RIF ไม่เท่ากับ 1 นั้นจะทำให้แหล่งกำเนิดไม่สามารถปรับเพิ่มอัตราส่งได้อย่างรวดเร็ว และจะทำให้การใช้งานสายสัญญาณมีค่าน้อยตามไปด้วย วิธี RSS algorithm นั้นสามารถลด consolidation noise ได้ โดยการเพิ่มความถูกต้องของข่าวสารความคับคั่งที่ทำการป้อนกลับสู่แหล่งกำเนิด โดยไม่ทำการป้อนกลับข่าวสารความคับคั่งหากว่ายังไม่ได้รับเซลล์ BRM จาก branch ใด branch หนึ่ง อย่างไรก็ตามวิธีการดังกล่าวยังคงมีปัญหา consolidation noise อยู่เมื่อใช้งานกับวิธี ERICA

วิธี Wait-for-all algorithm, Immediate Rate Calculation algorithm และ วิธีการที่เสนอแบบที่ 1 นั้นสามารถขจัด Consolidation noise ไปได้ โดย Wait-for-all algorithm และ Immediate Rate Calculation algorithm นั้นรอให้มีการรับเซลล์ BRM จากทุก branch ก่อนการป้อนกลับเซลล์ BRM ทำให้ข่าวสารที่เก็บรวบรวมไว้ที่สวิตช์นั้นถูกต้องเสมอ ส่วนวิธีการที่เสนอแบบที่ 1 นั้นมีการจัดเก็บข่าวสารความคับคั่งที่ดี ทำให้มีความถูกต้องของข่าวสารที่รวบรวมไว้เสมอ และ ไม่เกิด consolidation noise ขึ้น

ส่วนวิธี วิธีการที่เสนอแบบที่ 2 นั้นสามารถลด Consolidation noise ให้อยู่ในระดับที่น้อยมาก ถึงแม้ว่าจะยังคงมี consolidation noise อยู่บ้างในบางกรณีเท่านั้น แต่ผลของ consolidation noise ที่เกิดขึ้นนั้นมีผลต่อโครงข่ายน้อยมาก โดยไม่ก่อให้เกิดความคับคั่งขึ้นในโครงข่าย ด้วยความสามารถในการหยุดยังไม่ให้ Consolidation noise ที่เกิดขึ้นมานั้น เกิดต่อเนื่องไปเรื่อย ๆ จนทำให้โครงข่ายอยู่ในสภาวะที่คับคั่งในที่สุด การไม่ลบล้างค่า MER ที่เก็บรวบรวมไว้หลังจากป้อนกลับเซลล์ BRM ไปแล้ว ทำให้ข่าวสารที่รวบรวมไว้มีความถูกต้องมากขึ้น และสามารถลด Consolidation noise ได้

5.2.2 ผลตอบสนองชั่วคราวในสภาวะโหลดเกิน

ผลตอบสนองชั่วคราวในสภาวะเริ่มต้นของโครงข่ายนั้นมีความสำคัญมาก เนื่องจากในช่วงเริ่มต้นนั้นหากโครงข่ายอยู่ในสภาวะคับคั่งเนื่องจากโหลดเกิน วิธีรวบรวมข่าวสารความคับคั่งที่มีผลตอบสนองชั่วคราวที่ช้านั้นจะทำให้โครงข่ายอยู่ในสภาวะคับคั่งเป็นเวลานาน ซึ่งสามารถก่อให้เกิดการสูญเสียเซลล์ข้อมูลได้

วิธี Robert algorithm และ วิธีการที่เสนอแบบที่ 1 นั้น ให้ผลตอบสนองชั่วคราวในสภาวะโหลดเกินได้รวดเร็วที่สุด เนื่องจากสามารถสร้างเซลล์ BRM ขึ้นมาเพื่อป้อนกลับได้ทันทีหลังจากรับเซลล์ FRM ตัวแรก การที่วิธีทั้งสองนั้นสามารถป้อนกลับเซลล์ BRM ได้อย่างรวดเร็ว ทำให้โครงข่ายหลุดออกจากสภาวะคับคั่งได้อย่างรวดเร็วตามไปด้วย

สำหรับ TS algorithm, RSS algorithm และ วิธีการที่เสนอแบบที่ 2 นั้นมีผลตอบสนองชั่วคราวที่รวดเร็วเช่นเดียวกัน แต่การที่วิธีการทั้งสามนั้นต้องรอให้มีเซลล์ BRM เดินทางกลับมาจากปลายทางอย่างน้อย 1 ตัวก่อนจึงสามารถป้อนกลับเซลล์ BRM ได้ ทำให้วิธีการทั้งสามนั้นมีผลตอบสนองชั่วคราวที่ช้ากว่า Robert algorithm

วิธี Immediate Rate Calculation algorithm นั้นมีผลตอบสนองชั่วคราวในสภาวะโหลดเกินที่รวดเร็ว ทั้งนี้เนื่องจากวิธีดังกล่าวใช้เทคนิค Overload detected ซึ่งสามารถตรวจจับสภาวะโหลดเกินได้อย่างรวดเร็ว และทำให้สามารถป้อนกลับเซลล์ BRM กลับไปสู่แหล่งกำเนิดได้ทันทีในสภาวะโหลดเกิน ดังนั้นแหล่งกำเนิดจึงสามารถปรับอัตราส่งมาอยู่ที่ค่าที่โครงข่ายสามารถรองรับได้อย่างรวดเร็ว ทำให้ผลตอบสนองชั่วคราวในสภาวะโหลดเกินนั้นรวดเร็วขึ้นด้วย อย่างไรก็ตามผลตอบสนองที่รวดเร็วนี้นั้นขึ้นอยู่กับ การตรวจจับสภาวะโหลดเกิน ซึ่งหากค่า threshold ที่ตั้งไว้เพื่อตรวจจับสภาวะโหลดเกินนั้นไม่ดีพอ จะทำให้ไม่สามารถตรวจจับสภาวะโหลดเกินได้ในบางกรณีไม่ได้

Wait-for-all algorithm นั้นให้ผลตอบสนองชั่วคราวในสภาวะโหลดเกินที่ช้ามาก ทั้งนี้วิธีนี้ต้องรอให้เซลล์ BRM จากทุก branch เดินทางกลับมาถึงสวิตช์ก่อน ดังนั้นในสภาวะที่ยังไม่มีเซลล์ RM ในโครงข่ายมากนักเช่นในช่วงเริ่มต้นนั้น สวิตช์จะต้องรอให้เซลล์ BRM จากปลายทางที่ไกลที่สุดเดินทางมาถึงก่อน จึงสามารถป้อนกลับเซลล์ BRM กลับไปได้ ซึ่งหากว่าปลายทางที่ไกลที่สุดนั้นมีเวลาประวิงสูงมาก ก็จะทำให้ผลตอบสนองชั่วคราวในสภาวะโหลดเกินนั้นช้าตามไปด้วย

5.2.3 ผลตอบสนองชั่วคราวในสภาวะโหลดต่ำ

ในสภาวะโหลดน้อยนั้น วิธีควบคุมข่าวสารความคับคั่งต้องสามารถปรับอัตราส่งของแหล่งกำเนิดให้เพิ่มขึ้นได้ เพื่อให้โครงข่ายสามารถใช้งานสายสัญญาณได้อย่างคุ้มค่าที่สุด ซึ่งหากว่าวิธีรวบรวมข่าวสารความคับคั่งแบบใดที่ทำให้ผลตอบสนองชั่วคราวในสภาวะโหลดต่ำช้า ก็จะทำให้เกิดการใช้งานสัญญาณที่ไม่คุ้มค่า โดยวิธี Robert algorithm และ วิธีการที่เสนอแบบที่ 1 นั้นยังคงมีผลตอบสนองใน

สภาวะโหลคน้อยที่รวดเร็วที่สุดเช่นเดียวกับในกรณีโหลคเกิน ด้วยสาเหตุเดียวกันคือ วิธีการทั้งสองสามารถตอบกลับเซลล์ BRM ทันทีที่ได้รับเซลล์ FRM ตัวแรก ในขณะที่วิธี TS algorithm, RSS algorithm และ วิธีการที่เสนอแบบที่ 2 นั้นยังคงมีผลตอบสนองชั่วคราวที่รวดเร็ว โดยในสภาวะที่ไม่มีเซลล์ RM ในโครงข่าย วิธีการดังกล่าวสามารถป้อนกลับเซลล์ BRM ได้ทันทีที่สวิทช์ได้รับเซลล์ BRM ตัวแรก

สำหรับ Wait-for-all algorithm และ Immediate Rate Calculation algorithm นั้นมีผลตอบสนองชั่วคราวที่ช้ามาก โดยเฉพาะเมื่อไม่มีเซลล์ BRM ในโครงข่าย เช่นในสภาวะเริ่มต้นการส่งข้อมูล ทั้งนี้เนื่องจากว่าวิธี Wait-for-all algorithm ทำให้สวิทช์จำเป็นต้องรอให้เซลล์ BRM เดินทางกลับมาจากปลายทางทุก branch ก่อน ซึ่งหาก branch หนึ่งที่มีเวลาประวิงมาก ก็จะทำให้เวลาที่ใช้ในการรอการตอบกลับนั้นนานตามไปด้วย ส่วนวิธี Immediate Rate Calculation algorithm นั้นจะทำงานเหมือนกับวิธี Wait-for-all algorithm เมื่อโครงข่ายอยู่ในสภาวะโหลคต่ำ ดังนั้นผลตอบสนองชั่วคราวในสภาวะโหลคต่ำจึงเหมือนกับ Wait-for-all algorithm

5.2.4 ความซับซ้อนในการพัฒนา

วิธีรวบรวมข่าวสารความคับคั่งที่ดีไม่ควรที่จะมีความซับซ้อนมากนัก เพื่อความรวดเร็ว, ง่าย และ ประหยัด ในการพัฒนาสวิทช์ที่สามารถใช้งานได้จริง ความซับซ้อนในการพัฒนาขึ้นอยู่กับปริมาณตัวแปรที่ใช้เก็บข้อมูลสำหรับแต่ละวิธี, การเพิ่มฟังก์ชันในการทำงานของสวิทช์ และ ลำดับขั้นตอนในการคำนวณ

ในวิธี Robert algorithm, TS algorithm และ วิธีการที่เสนอแบบที่ 1 นั้นมีความซับซ้อนในการพัฒนาสูง เนื่องจากว่าวิธีการทั้งสองนั้นต้องการให้สวิทช์เพิ่มฟังก์ชันในการสร้างเซลล์ BRM เพื่อการตอบกลับขึ้นมา หลักการนี้คล้ายคลึงกับหลักการของ Virtual Source/Virtual Destination (VS/VD) แต่อย่างไรก็ตามก็ถือได้ว่าเป็นค่าเพิ่มความซับซ้อนให้แก่สวิทช์อย่างมาก

วิธี RSS algorithm และ วิธีการที่เสนอแบบที่ 2 มีความซับซ้อนอยู่ในระดับที่น้อยที่สุดโดยสังเกตจากจำนวนปริมาณตัวแปรที่ต้องการใช้ที่เก็บไว้ที่สวิทช์ นั้นมีน้อยที่สุด และมีลำดับขั้นตอนในการทำงานน้อยที่สุดด้วย ในขณะที่ไม่ต้องการมีการสร้างเซลล์ BRM เพื่อการตอบกลับ ส่วนวิธี Wait-for-all algorithm นั้นมีความซับซ้อนมากกว่าทั้งสองวิธีข้างต้น เนื่องจากต้องมีการเพิ่มตัวแปรในการตรวจจับการรับเซลล์ BRM จากแต่ละ branch และ ตัวแปรในการเก็บปริมาณเซลล์ BRM ที่รับได้ ในขณะที่เพิ่มลำดับขั้นตอนการทำงานขึ้นมา

Immediate Rate Calculation algorithm มีความซับซ้อนในระดับที่มากกว่า Wait-for-all algorithm พอสมควร จากการเพิ่มเทคนิค Overload detected โดยต้องเพิ่มตัวแปรในการจดจำค่า Last_ER เพื่อนำมาเปรียบเทียบกับเทคนิค Overload detected นอกจากนี้ยังเพิ่มขั้นตอนการทำงานขึ้นมา

พอสมควร โดยเฉพาะการคำนวณโดยใช้วิธีควบคุมความคับคั่งทุกครั้งในการรับเซลล์ BRM ทุกตัว โดยไม่คำนวณเฉพาะตอนที่กำลังจะตอบกลับเซลล์ BRM

5.2.5 RM cell Overhead

RM cell Overhead คือ การสร้างเซลล์ BRM ที่เกินความจำเป็นขึ้นมาในโครงข่าย โดยปริมาณเซลล์ FRM ที่แหล่งกำเนิดสร้างขึ้นนั้นควรมีค่าประมาณเท่า ๆ กับ ปริมาณเซลล์ BRM ที่แหล่งกำเนิดได้รับ โดยปริมาณดังกล่าวแหล่งกำเนิดนั้นจะนับปริมาณเซลล์ FRM ที่สร้างขึ้นในขณะที่ยังไม่ได้รับเซลล์ BRM ตอบกลับมาจากเซลล์ FRM ตัวนั้น ซึ่งหากค่าดังกล่าวมีค่ามากกว่าค่า CRM แหล่งกำเนิดจะปรับอัตราส่งให้ลดลงเป็นค่า $ACR * CDF$ ซึ่งหากว่าวิธีรวบรวมข่าวสารความคับคั่งใด มีการผลิตเซลล์ BRM มากกว่าจำนวนเซลล์ FRM ที่ได้รับ อาจเป็นสาเหตุให้เกิดความผิดพลาดในการนับจำนวนดังกล่าว และนำไปสู่ความผิดพลาดในการควบคุมความคับคั่งได้ การตรวจสอบการป้อนกลับเซลล์ BRM ของวิธีรวบรวมข่าวสารความคับคั่งนั้นสามารถแสดงได้จากค่าอัตราส่วนระหว่างเซลล์ FRM ที่รับได้ต่อ BRM ที่ผลิตสร้างขึ้น

วิธี Robert algorithm และ วิธีการที่เสนอแบบที่ 1 นั้นมีการป้อนกลับเซลล์ BRM ทันทีที่ได้รับเซลล์ FRM ทำให้อัตราส่วนระหว่างเซลล์ FRM ต่อ เซลล์ BRM นั้นมีค่าเป็นหนึ่งเสมอ วิธี TS algorithm นั้นต้องรอให้มีรับเซลล์ BRM ก่อนจึงสามารถป้อนกลับเซลล์ BRM ทันทีที่ได้รับเซลล์ FRM ทำให้อัตราส่วนนั้นน้อยกว่า 1 เสมอ เนื่องจากเซลล์ FRM ตัวแรกนั้นจะไม่ได้รับการตอบกลับแน่นอน เนื่องจากยังไม่มีเซลล์ BRM ในโครงข่าย ส่วน RSS algorithm และ วิธีการที่เสนอแบบที่ 2 นั้นจะป้อนกลับเซลล์ BRM ทันทีที่ได้รับเซลล์ BRM ตัวแรกหลังจากได้รับเซลล์ FRM แล้ว ซึ่งถือว่าเป็นเงื่อนไขที่มีการทำงานคล้ายคลึงกับกรณีการส่งแบบหนึ่งจุดไปสู่หนึ่งจุด ซึ่งโดยรวมแล้วจะทำให้อัตราส่วนระหว่างเซลล์ FRM ต่อเซลล์ BRM นั้นมีค่าน้อยกว่าหนึ่งในช่วงที่ได้รับเซลล์ FRM แล้วแต่ยังไม่ได้รับเซลล์ BRM และมีค่าเท่ากับหนึ่งหลังจากป้อนกลับเซลล์ BRM ไปแล้วจนถึงก่อนได้รับเซลล์ FRM ตัวต่อไป

สำหรับ Wait-for-all algorithm นั้นมีอัตราส่วนระหว่างเซลล์ FRM ต่อเซลล์ BRM เป็นค่าน้อยกว่าหรือเท่ากับ 1 เนื่องจากการรอให้ได้รับเซลล์ BRM จากทุก branch ก่อนการป้อนกลับนั่นเอง ส่วนวิธี Immediate Rate Calculation algorithm นั้นหากว่าโครงข่ายอยู่ในสภาวะไหลคเกิน จะมีการป้อนกลับเซลล์ BRM เพิ่มขึ้น ดังนั้นจะทำให้อัตราส่วนระหว่างเซลล์ FRM และ เซลล์ BRM มีค่ามากกว่า 1 ในบางสภาวะ อย่างไรก็ตาม วิธีการดังกล่าวมีการแก้ไขข้อผิดพลาดนี้โดยการนับจำนวนเซลล์ BRM ที่ส่งไปด้วยเทคนิค Overload detected และจะทำการงดการป้อนกลับเซลล์ BRM ในสภาวะปกติ เป็นจำนวนเท่ากับจำนวนเซลล์ BRM ที่ส่งกินไปนั่นเอง วิธีการนี้สามารถควบคุมอัตราส่วนระหว่างเซลล์ FRM และ เซลล์ BRM ให้มีค่าเป็นน้อยกว่า หรือเท่ากับ 1 ในสภาวะอยู่ตัวได้ อย่างไรก็ตามค่าดังกล่าวอาจมีค่ามากกว่า 1 ในช่วงสภาวะไหลคเกินได้

5.2.6 Change to non responsive branch

สำหรับการส่งจากหนึ่งจุดไปสู่หลายจุดนั้น หากมี branch ใด branch หนึ่งที่เกิดความผิดพลาด และไม่สามารถตอบสนองต่อเซลล์ FRM ที่ส่งไปได้ วิธีการรวบรวมข่าวสารความคับคั่งควรที่สามารถทำงานต่อไปได้โดยปกติ วิธี Robert algorithm, TS algorithm, RSS algorithm, วิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 นั้นมีการเปลี่ยนแปลงต่อกรณีที่ branch ใด branch หนึ่งไม่ตอบสนองที่ต่ำมาก เนื่องจากวิธีการดังกล่าวนั้นใช้เงื่อนไขในการรับเซลล์ FRM เป็นตัวตัดสินใจหลักในการป้อนกลับเซลล์ BRM ทำให้หากขาดเซลล์ BRM จาก branch ใด branch หนึ่ง ก็ยังสามารถที่จะทำงานต่อไปได้โดยอาศัยข่าวสารการป้อนกลับจาก branch ที่เหลือแทน

ส่วน วิธี Wait-for-all algorithm และ Immediate Rate Calculation algorithm นั้นจะเกิดปัญหาในกรณีนี้ โดยที่วิธีการทั้งสองมีพื้นฐานในการที่ต้องรอให้มีเซลล์ BRM ตอบกลับมาจากทุก branch ก่อน ซึ่งหากว่าไม่มีเซลล์ BRM ตอบกลับมาจาก branch ใด branch หนึ่งนั้น จะทำให้วิธีทั้งสองนั้นไม่สามารถที่จะป้อนกลับเซลล์ BRM ได้ นอกจากนี้วิธีการเพิ่มเติมในการตรวจสอบกรณีดังกล่าวแยกตัวอย่างเช่น การจับเวลารอสำหรับทุก branch ซึ่งจะทำให้สวิตช์มีความซับซ้อนเพิ่มขึ้นไปอีก

5.2.7 Change to newly joined branch

ในการใช้งานจริงนั้น ต้องการมีการใช้งานที่สามารถเพิ่มเติมปลายทางเข้ามาใน multicast tree นั้น ดังนั้นวิธีการรวบรวมข่าวสารความคับคั่งที่คั้นควรมีการตอบสนองที่รวดเร็วต่อการเพิ่ม branch ใหม่นี้ โดยวิธี Robert algorithm, TS algorithm, RSS algorithm, วิธีการที่เสนอแบบที่ 1 และ วิธีการที่เสนอแบบที่ 2 ยังคงมีผลตอบสนองต่อเหตุการณ์ดังกล่าวได้เป็นอย่างดี โดยที่สามารถป้อนกลับเซลล์ BRM ได้ เนื่องจากวิธีการดังกล่าวนี้มีเงื่อนไขในการป้อนกลับเซลล์ BRM โดยใช้เซลล์ FRM ถึงแม้ว่าจะมี branch ใหม่เพิ่มเติมเข้ามา ก็ยังสามารถป้อนกลับเซลล์ BRM ไปได้อย่างรวดเร็วเป็นปกติ ในขณะที่ Wait-for-all algorithm และ Immediate Rate Calculation algorithm จะมีการเปลี่ยนแปลงในการทำงานอย่างมากหากมี branch ใหม่เพิ่มเติมเข้ามา เนื่องจากในช่วงเริ่มต้นการเพิ่ม branch ใหม่ นั้น สวิตช์ต้องรอให้เซลล์ FRM นั้นเดินทางไปและกลับมาจาก branch ใหม่ก่อน จึงสามารถป้อนกลับเซลล์ BRM ตัวต่อไปได้ ซึ่งหากว่า propagation delay ระหว่างสวิตช์ไปสู่ปลายทางใหม่นั้นมีค่าสูงจะทำให้วิธีทั้งสองนั้นต้องรอการตอบกลับจากปลายทางดังกล่าวเป็นเวลานานตามไปด้วย ซึ่งในช่วงนี้ หากโครงข่ายมีการเปลี่ยนแปลงก็จะทำให้แหล่งกำเนิดไม่สามารถตอบสนองต่อการเปลี่ยนแปลงนี้ได้ เนื่องจากไม่ได้รับเซลล์ BRM จากโครงข่ายนั่นเอง

5.3 ข้อเสนอแนะ

1. การเปรียบเทียบระหว่างวิธีควบคุมความคับคั่งเพียง 2 วิธีคือ ERICA และ FMMRA นั้นสามารถสรุปผลของวิธีการควบคุมความคับคั่งที่มีผลต่อประสิทธิภาพด้าน consolidation noise และ consolidation delay ที่เกิดขึ้นในการส่งจากหนึ่งจุดไปสู่หลายจุดได้ชัดเจนพอสมควร แต่ควรเพิ่มเติมในการทดสอบสำหรับวิธีควบคุมความคับคั่งแบบอื่น ๆ ต่อไป เพื่อสามารถสรุปผลดังกล่าวเพิ่มเติมได้

2. การทำงานร่วมกับระหว่างวิธีรวบรวมข่าวสารความคับคั่งด้วยตนเอง เป็นสิ่งที่ควรได้รับการศึกษา เนื่องจากหากในโครงข่ายประกอบด้วยสวิตช์ที่ใช้วิธีรวบรวมข่าวสารความคับคั่งที่ต่างกัน อาจทำให้เกิดการผิดพลาดหรือมีผลต่อประสิทธิภาพในการส่งข้อมูลอย่างมาก

3. วิธีการรวบรวมความคับคั่งนั้นได้มีการพัฒนาขึ้นมาใหม่ จึงควรมีการศึกษาเพิ่มเติมเพื่อศึกษาให้ครอบคลุมทุกวิธีการต่อไป

เอกสารอ้างอิง

1. ATM Forum. Traffic Management Specification. v4.0, Apr. 1996.
2. F. Bonomi and K. W. Fendick. The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service. IEEE Network, Mar.-Apr. 1995, pp. 25-39.
3. Jeremy P. Chew , Anil K . Gupta ,and Ma Tit Yap. Performance of ABR Flow Control in ATM Networks. ICICS'97 , Sep. 1997.
4. R. Jain et al. ERICA Switch Algorithm: A Complete Description. ATM Forum cont. 96-117, Aug. 1996.
5. A. Arulambalam , X. Chen and N. Ansari. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. IEEE comm. Mag. , Nov. 1996, pp. 92-100.
6. A. Arulambalam , X. Chen and N. Ansari. An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks. ICC'97 , June 1997, pp. 200-204.
7. F. Chiussi, A. Arulambalam , Y. Xia and X. Chen. Explicit Rate ABR Schemes Using Traffic Load as Congestion Indicator. IEEE Processdings Sixth International Conference , Sept. 1997, pp. 76-84.
8. L. Roberts. Rate based algorithm for point to multipoint ABR service. ATM Forum/94-0772R1, Nov. 1994.
9. H-Y. Tzeng and K-Y. Siu. On max-min fair congestion control for multicast ABR service in ATM. IEEE JSAC, vol 15, no. 3, Apr. 1997, pp. 545-555.
10. W. Ren, K-Y. Siu, and H. Suzuki. On the performance of congestion control algorithms for multicast ABR service in ATM. Proceedings of IEEE ATM'96, 1996
11. Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shivkumar Kalyanaraman, Sastri Kota, and Pradeep Samudra. Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks. Proceedings of IEEE INFOCOM 1998, volume 3, March 1998, pp. 1004-1013.
12. Alexander Gersht and Kyoo J. Lee. A Congestion Control Framework for ATM Networks. IEEE JSAC, vol 9, no. 7, Sep. 1991, pp. 1119-1129.
13. Tianji Jinag, Ellen W. Zegura and Mostafa Ammar. Improved Consolidation Algorithm for Point-to-Multipoint ABR Service. IEEE ATM workshop Proceedings, May. 1998, pp. 195-201.

14. Sonia Fahmy, Raj Jain et al. Performance analysis of ABR point-to-multipoint connections for bursty and non-bursty traffic with and without VBR background. ATM Forum cont. 97-0422, April. 1997.
15. Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, and Shivkumar Kalyanaraman. Design and Evaluation of Feedback Consolidation for ABR Point-to-Multipoint Connections in ATM Networks. Journal of Computer Communications, Vol. 22, Issue 12, 25 July 1999, pp. 1085-1103.
16. Sonia Fahmy, Raj Jain, Shivkumar Kalyanaraman, Rohit Goyal, Bobby Vandalore and Xiangrong Cai. A Survey of Protocols and Open Issues in ATM Multipoint Communications. OSU Technical Report, August 21, 1997.
17. Dong-Ho Kim, You-Ze Cho, Yoon-Young An and Yul Kwon. A Scalable Consolidation Algorithm for Point-to-Multipoint ABR Flow Control in ATM Networks. ICC'99 , vol 1, 1999, pp. 118-123.
18. Hung-Shiun, Alex Chen and Klara Nahrstedt. Feedback Consolidation and Timeout Algorithm for Point-to-Multipoint ABR Service. ICC'99, vol 1, 1999, pp. 135-139.
19. Kai-Yeung Siu and Hong-Yi Tzeng. Congestion Control For Multicast Service in ATM Networks. GLOBECOM'95, vol 1, 1995, pp. 310-314.
20. You-Ze Cho, Sang-Min Lee and Myeong-Yong Lee. An Efficient Rate-Based Algorithm for Point-to-Multipoint ABR Service. GLOBECOM'97, vol 2, 1997, pp. 790-795.

ภาคผนวก

ส่วนหนึ่งของโครงการวิจัยนี้ได้รับการตีพิมพ์ในบทความวิจัยนานาชาติจำนวน 5 ฉบับดังนี้

- S. Laohamatanee, L. Wuttisittikulkij and Prasit Prapinmongkolkarn, "Performance Classification of Consolidation Algorithms for Point-to-Multipoint ABR Service in ATM Networks," accepted for *7th IEEE Singapore International Conference on Communication Systems (ICCS)*, Singapore, November 2000.
- S. Laohamatanee, L. Wuttisittikulkij and P. Prapinmonkolkarn, "An E-FMMRA for ABR point-to-multipoint Service in ATM Network," *Proceedings of Second International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP2000)*, Bournemouth UK, July 2000, pp. 257-260.
- T. Keadtisuke, T. Jaruvitayakovit, P. Prapinmongkolkarn, and L. Wuttisittikulkij, "Improved EPD Algorithm with Dynamic Threshold Techniques for TCP in ATM Networks," *Proceedings of IEEE ISPACS'99*, Phuket, Thailand, 1999, pp. 239-242.
- S. Sookjaroen, T. Jaruvitayakovit, P. Prapinmongkolkarn, and L. Wuttisittikulkij, "Common Buffer Computation Credit Based Flow Control in ATM Networks," *Proceedings of IEEE ISPACS'99*, Phuket, Thailand, 1999, pp. 271-274.
- S. Loahamatanee, L. Wuttisittikulkij and P. Prapinmongkolkarn, "A New Feedback Consolidation Algorithm for Point-to-Multipoint ABR Service", in *Proceedings of IEEE TENCON'99*, Cheju, Korea, 1999, pp. 1387-1390.

จุฬาลงกรณ์มหาวิทยาลัย

Performance Classification of Consolidation Algorithms for Point-to-Multipoint ABR Service in ATM Networks

Lunchakorn Wuttisittikulkij, Supakiat Laohamatane, and Prasit Prapinmonkolkarn

Abstract— The main objective of designing a consolidation algorithm in point-to-multipoint available bit rate (ABR) service in ATM networks is finding how to consolidate BRM cells to avoid feedback implosion problem while providing fastest response to control traffic in the network. In this paper, we first describe the remaining unsolved problems in the existing consolidation algorithms and propose a new algorithm to solve these problems. The performance of all previously known consolidation algorithms has been investigated and classified in terms of consolidation noise, transient response, implementation complexity, ratio of FRM cells to BRM cells, the change to a non-responsive branch and change to a newly joined branch. Simulation results show that the proposed algorithm provides good and desirable performance compared to the existing algorithms.

Index Terms—ATM, ABR, consolidation algorithm

I. INTRODUCTION

The point-to-multipoint ABR (Available Bit Rate) service in ATM networks is currently considered important for supporting many emerging applications, such as data broadcasting via subscribed members. Considerable attention to this class of service is now eminent [1-9]. In point-to-point ABR service [10], a source sends a forward resource management (FRM) cell along with data cells and it is turned around into a backward resource management (BRM) cell by a destination. Along the way back to the source, each switch adjusts the explicit rate (ER) in the BRM cell to the rate that the network is able to support.

The point-to-multipoint ABR service differs from the point-to-point ABR service in that there exists a number of feedback information from all the branch points sent back to the source through BRM cells. This causes a serious problem of how to handle such a large amount of feedback information. A common technique to overcome this problem is that each branch point in the multicast tree consolidates the information in BRM cells received from downstream branch points or destinations and forwards the selected one to its upstream switch. This significantly

reduces the amount of feedback information in the system. The main challenge this problem poses is how to consolidate BRM cells in such a way that the transient response of each traffic source can be fast and simultaneously avoiding undesired consolidation noise [4,5]. So far, a number of consolidation algorithms have been proposed [1-9].

Early consolidation algorithms intend to provide fast transient response but they suffer from a consolidation noise problem. In contrast, later consolidation algorithms aim to maintain correct feedback information stored at the switches and attempt to improve the transient response especially in overload conditions. However these algorithms still have a slow transient response in underload conditions leading to congestion and low link utilization. Very recently, Supakiet and et.al. [9] have proposed a new feedback information storing method that can provide fast transient response while completely eliminating consolidation noise. However, this algorithm requires additional variables for the switches, and hence increases the complexity of the switch implementation. In this paper, we describe a new consolidation algorithm alternative that can eliminate consolidation noise, has a fast transient response for both overload and underload conditions and yet has low complexity in the switches. The details of this algorithm will be explained in section 4. Moreover, the performance of most previous consolidation algorithms has been classified and summarized in terms of consolidation noise, transient response, implementation complexity, ratio of FRM cells to BRM cells, the change to the non-responsive branch and to the newly joined branch.

II. DESIGN ISSUE

In this section, we explain several key issues to the design of a consolidation algorithm at each branch point.

1. Fast transient response is important for maximum channel utilization and prevention of congestion.
2. Consolidation noise should be eliminated to avoid unfairness and congestion in the network by maintaining correct and up-to-date feedback information stored at each branch point.

L. Wuttisittikulkij and P. Prapinmonkolkarn are with the Department of Electrical Engineering, Chulalongkorn University, Bangkok, Thailand (e-mail: lunch@ee.eng.chula.ac.th).

S. Laohamatane is with the Department of Electrical Engineering, Chulalongkorn University, Bangkok, Thailand. He is now with Bangkok Datacom company, Bangkok, Thailand.

3. Each switch branch point should not generate new BRM cells because that would add more complexity to the switch.
4. The number of variables used in each switch for consolidation purposes should be kept to a minimum to reduce the switch complexity.
5. The ratio of FRM cells generated by the source to BRM cells returned to the source should be maintained close to 1 as certain types of sources are sensitive to this ratio.
6. Consolidation algorithms should be able to continue working properly and effectively even if there is a non-responsive branch. Some consolidation algorithms have to wait for the BRM cell to return from all branches, so that the branch point is kept waiting indefinitely for the BRM cell from a non-responsive branch. During this time, if any traffic condition is changed in the network, this branch point can not send the BRM cell to the source to adjust the ACR value to the rate the network is able to support. A timeout mechanism is introduced to solve this problem. Based on this mechanism, each branch point waits for the BRM cell until the timeout value expires, and sends an extra BRM cell to the upstream node to convey feedback information to the source.
7. When there is a new branch joining an existing branch point, the existing branch point should send a BRM cell to its upstream node without waiting for the BRM cells to return from the new branch in order to maintain a good response.

III. RELATED WORK

Robert [1] first proposed a consolidation algorithm, which has the objective of achieving the fastest transient response. The basic idea of this algorithm is that each branch point consolidates the feedback information on a per-VC basis and passes it on to its upstream node through a BRM cell after the branch point has received a FRM cell. With this technique, consolidated feedback information may not reflect the actual congestion level of all the branches, because the feedback from some branches may have not yet arrived at the time when the BRM cell is sent. This could cause undesired consolidation noise. To reduce such consolidation noise, Tzeng and Siu [2] improved this algorithm by ensuring that the feedback to the upstream node is sent only when at least one BRM cell has been received from a downstream branch before receiving an FRM cell. This algorithm is able to reduce the level of consolidation noise. Ren [3] argued that generating a BRM cell causes switches to higher complexity and he also proposed two alternative algorithms. In his first algorithm, the first BRM cell received is passed to the source after a FRM cell has been received. Note that all the three algorithms described so far cause consolidation noise. For his second algorithm, Ren presented the algorithm that can eliminate the consolidation noise entirely. The algorithm is referred to as a "wait-for-all" algorithm. In this algorithm,

each branch point waits for the BRM cells, arrival from all the branches before a BRM cell is sent to the source. Although the problem of consolidation noise is completely removed, the waiting time for all feedback information may be long, causing a very slow transient response.

Recently, an improved algorithm to the "wait-for-all" algorithm has been found [4-8]. These papers introduced a fast overload indication function to the "wait-for-all" algorithm to enhance the transient response under severe network overload conditions. In an immediate rate calculation algorithm [4,5], whenever the switch receives a BRM cell with ER much lower than the previous ER, it generates an extra BRM immediately back to its upstream branch. A probabilistic aggregate algorithm [6] suggests an alternative way to improve the overload transient response by random feedback of BRM cells if there is an overload condition found in the network. Another algorithm referred to as a timeout algorithm [7] merges the immediate rate calculation algorithm and the probabilistic aggregate algorithm into a new overload detection technique. In this technique, the branch point sends extra BRM cells if the ER value from a received BRM cell is much lower than the previous ER and randomly sends extra BRM cells if the ER value from the received BRM cell is slightly lower than the previous one. In addition, this algorithm introduced a timeout mechanism to manage non-responsive branches in the network. Although these algorithms could improve the transient response in the overload condition, it occasionally generates more BRM cells than the number of FRM cells it has received. In addition, these algorithms are unable to respond to the underload condition when the source node should be entitled to increase its rate.

Kim [8] proposed a scalable algorithm to solve the problem of consolidation noise and delay. The main idea in this algorithm is that each branch point stores the feedback information on a per-branch basis for each VC and passes BRM cells returning from the most remote destination only.

Supakiat [9] proposed another way to eliminate consolidation noise by maintaining a good feedback information storing method. In Supakiat's algorithm, the branch point maintains the feedback information on a per-branch basis for each VC to eliminate consolidation noise. This approach enhances the transient response in the system by immediately sending the BRM cell after receiving a FRM cell. However, it adds more complexity in increasing the number of variables in the switches.

For convenience, in the following paragraphs, Robert's algorithm will be referred to as "A1", Tzeng's algorithm will be "A2". Ren's first and second algorithms will be denoted as "A3" and "A4", respectively. The immediate rate calculation algorithm will be referred to as "A5", the probabilistic aggregate algorithm will be known as "A6", the timeout algorithm will be called "A7", the scalable algorithm will be denoted as "A8" and finally, the Supakiat's algorithm will be known as "A9".

IV. THE NEW CONSOLIDATION ALGORITHM

The new algorithm has been developed based on the A3 algorithm because this algorithm provides good performance on all counts except for the consolidation noise. To solve the consolidation noise problem of this algorithm, we must understand how consolidation noise actually happens. For the A3 algorithm, each branch point sends BRM back to the source whenever it receives a first BRM cell after having received an FRM cell and resets its stored feedback information to its peak cell rate (PCR). If the source increases its rate and generates more FRM cells, the number of the FRM cells received at the branch point is increasing. In this situation, each branch point sends BRM cell more frequently and consolidation noise will occur if the branch points do not receive the BRM cells from the congested branch before sending the next BRM cell. Therefore, the branch points will use PCR as feedback information given in the BRM cell sent to the source, thus leading to rate oscillation and congestion in the network.

The new consolidation algorithm solves the consolidation noise problem in algorithm A3 by not resetting the stored feedback information at the branch point after it sending the BRM cell but using feedback information from a recently received BRM cell as newly stored feedback information. This can prevent consolidation noise in the next BRM cell cycle. A new variable known as TER has been introduced to memorize the ER value of the BRM cell before being updated by the congestion control algorithm. This value will be assign as new feedback information after branch point feedback of the BRM cell. The pseudo code of this algorithm can be shown as follows:

```

Upon the receipt of a forward RM(ER,CI,NI) cell :
  Multicast this RM cell to all participating branches;
  Let AtLeastOneFRM = 1;
Upon the receipt of a backward RM(ER,CI,NI) Cell
  Let MER = min (MER , ER) , MCI = MCI or CI , MNI = MNI or NI
  If AtLeastOneFRM then
    Let TER = ER;
    Let ER = MER , CI = MCI , NI = MNI;
    Let ER = min (ER , ER calculated
      By congestion control scheme for all branch)
    Send this RM cell back to the source;
    Let MER = TER , MCI = 0 , MNI = 0;
    Let AtLeastOneFRM = 0;
  else
    Discard the BRM Cell
  
```

V. PERFORMANCE ANALYSIS

A. Parameter Setting

In the simulation, the following parameter values were used.

- All links have a bandwidth of 150 Mbps.

- All sources have PCR = 150 Mbps, RIF = 1, RDF = 0.125, Nrm = 32, ICR for source A = 25 Mbps and ICR for source B and C = 150 Mbps
- All point-to-multipoint traffic flows from the root to the leaves of the tree. No traffic flows from the leaves to the root, except for the RM cells. The same applies for the point-to-point connection.
- The sources are persistent, i.e. there is always data to send.
- All switches are assumed to employ an ERICA+ [11] congestion control algorithm with the switch measurement interval set to a minimum between the time to receive 100 cells and 1 ms. Exponential queue control function was used in our simulation with T0 set to 0.0015 ms, a = 1.15, b = 1, QDLF = 0.5.

B. Simulation results

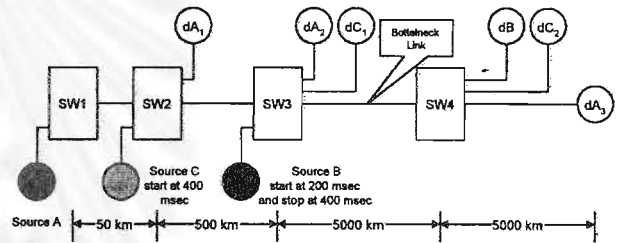


Figure 1: A test network model

We compared the consolidation noise and transient response in both overload and underload conditions with various consolidation algorithms including the new algorithm. The network configuration is shown in Figure 1. This network model is a specific configuration with very large round trip time and different transient sources. SW1 to SW4 denote four ATM switches. Source A is a point-to-multipoint connection with three receivers dA1 to dA3. The configuration has one ABR unicast connection from source B to dB. It activates after between 200 and 400 ms. A second point-to-multipoint connection sends data from source C to two receivers dC1 and dC2. It activates after 400 ms. Figure 2 to 10 show the ACR changes of source A, B and C for different consolidation algorithms. The network is in underload condition during the start phase and in overload condition when source C is active.

Figures 2 to 11 show the simulation results for the consolidation algorithms. As shown in Figure 2, algorithm A1 exhibits heavy consolidation noise due to incomplete feedback information stored at the branch point. However, it maintains a very fast transient response by immediately sending the BRM cell after receiving the FRM cell. Algorithms A2 and A3 provide a fast transient response, since they quickly send the BRM cell after receiving the first BRM cell. However, they still have a consolidation noise problem leading to rate oscillations, instability, unbounded queues and unfairness. In contrast, algorithm A4 can eliminate consolidation noise, but suffers from a slow

transient response especially when there are no BRM cells in the network because source A has to wait for the BRM cell to return from the most remote branch, the destination dA3. In the initial state, network is in the underload condition because of the low ICR value of source A. This algorithm can not increase the ACR value at source A to fairshare until after 100 ms leading to low link utilization in the start phase. Moreover, when source C is active at 400 ms with a high ICR value, the network is in the overload condition. This algorithm can not decrease the ACR value at source C to avoid congestion of the network until after 470 ms leading to cell loss at switches.

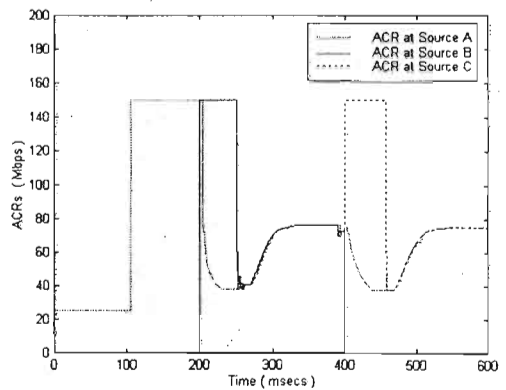


Figure 5: Results for algorithm A4

The result of algorithm A5 in Figure 6 is similar to that of algorithm A4. This algorithm can not respond to the underload condition and in this particular configuration it also fails to detect the overload condition even if equipped with the overload detection technique. Algorithm A5 detects the overload condition by comparing the current ER value with the previous one provided by the last BRM cell sent. If the ER value is lower than the last ER multiplied by the threshold level then the network is set to overload condition. The switch immediately sends an extra BRM cell to adjust the rate of the source. When source C is active, the network is in overload condition and the ERICA+ algorithm slightly decreases the rate of source A by using the queue control function. However, this decrease is below the threshold value and not fast enough to detect the overload condition, hence, the overload detection technique fails to detect the overload condition in this case. Therefore, the switch does not send an extra BRM cell to source C until it receives BRM cells from all branches.

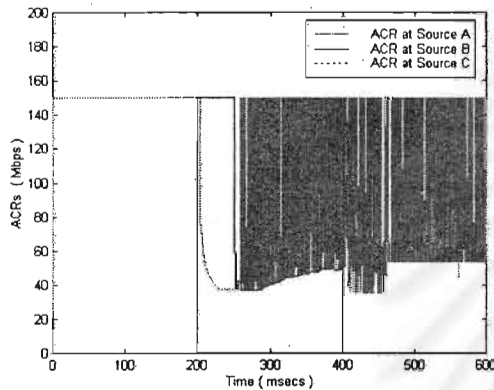


Figure 2: Results for algorithm A1

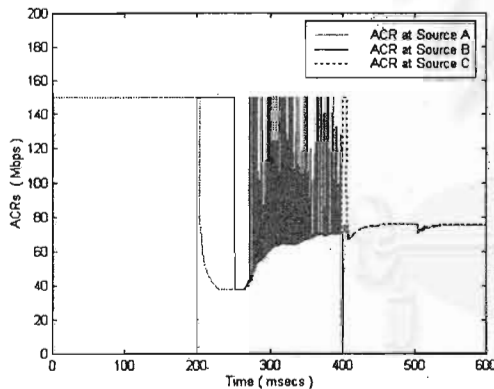


Figure 3: Results for algorithm A2

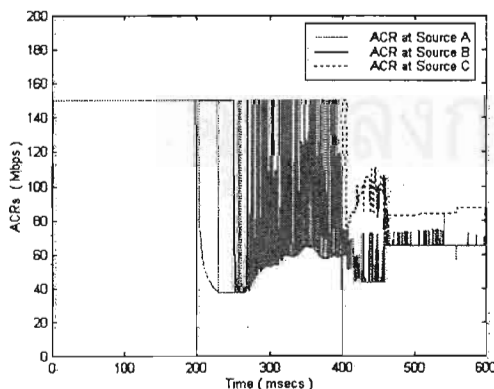


Figure 4: Results for algorithm A3

The results of algorithm A6 are identical to those of algorithm A5, since it detects the overload condition by determining the decrease in ER value in the BRM cells. However, this technique only uses information from downstream switches without any immediate rate calculation technique, which can calculate the ER value of this switch before detecting the overload condition. Therefore, algorithm A6 fails to detect the overload condition leading to a slow transient response in both overload and underload conditions. The results of algorithm A8 in Figure 9 are similar to those of algorithm A4 as the branch point must wait for the BRM cell to return from the most remote destination. Like algorithm A6, algorithm A7 uses the overload detection technique without immediate rate calculation technique and it produces a very slow transient response in both overload and underload conditions.

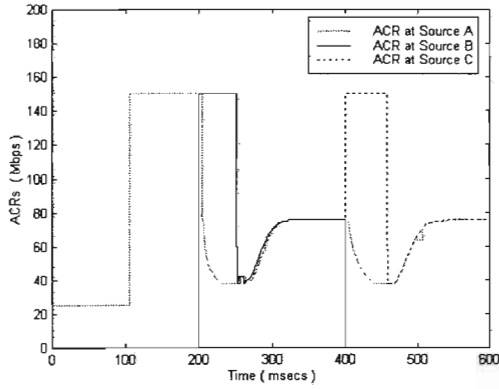


Figure 6: Results for algorithm A5

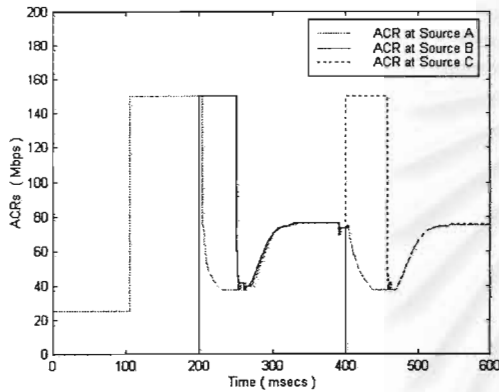


Figure 7: Results for algorithm A6

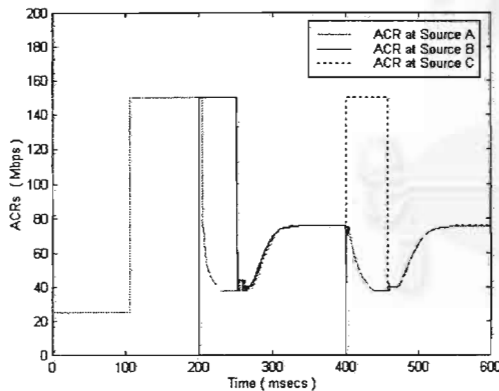


Figure 8: Results for algorithm A7

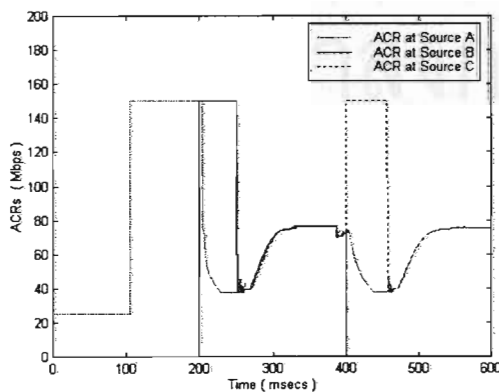


Figure 9: Results for algorithm A8

The algorithm A9 yields optimal performance in this network configuration, it can maintain stored feedback information, hence, no consolidation noise was found. Moreover, this algorithm provides a fast transient response in both overload and underload conditions because each branch point immediately sends the BRM cell to the upstream node after receiving the BRM cell. In the same manner, the proposed algorithm reduces consolidation noise by protecting stored feedback information from being reset to a high value. Like A3, this algorithm maintains a fast transient response. However, the transient response of A9 is slightly faster than the proposed algorithm.

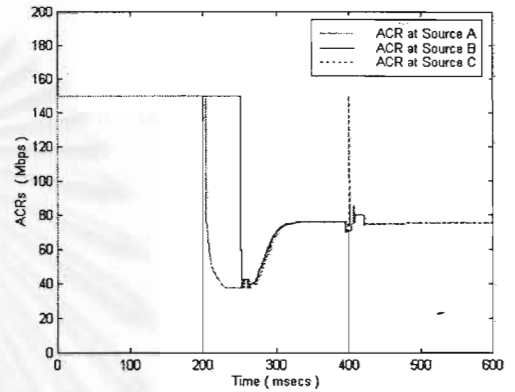


Figure 10: Results for algorithm A9

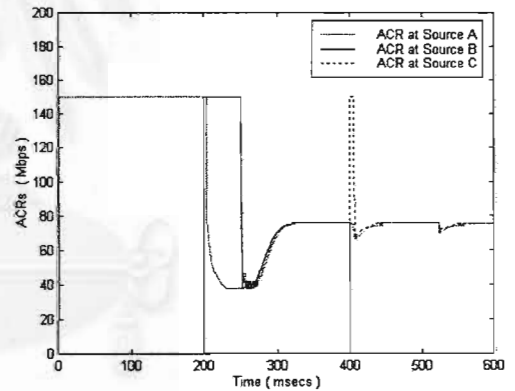


Figure 11: Results for algorithm A10

VI. COMPARISON OF THE CONSOLIDATION ALGORITHMS

This section summarizes the performance among various consolidation algorithms and compares between them in terms of consolidation noise, transient response, implementation complexity, ratio of FRM cells to BRM cells, change to non-responsive branch and change to newly joined branch.

A. Consolidation noise

Algorithms A1, A2 and A3 exhibit unacceptable consolidation noise in some cases due to incomplete feedback information stored at the branch point. Algorithms A4, A5, A6 and A7 can completely eliminate consolidation noise, since they wait for the BRM cells to return from all

branches. Algorithm A8 waits for the BRM cell from the most remote branch, this ensures that the branch point receives the BRM cells from all other branches before receiving that from the most remote one. Thus no consolidation noise is found in this algorithm. Algorithm A9 maintains good feedback information storing method, thus avoiding consolidation noise. The proposed algorithm protects the network from consolidation noise by not resetting the feedback information after sending the BRM cell.

B. Transient Response

Algorithms A1 and A9 exhibit the fastest transient response in both overload and underload conditions, since the feedback information is returned immediately after the branch point has received an FRM cell. Algorithms A2, A3 and proposed algorithm also have a very fast transient response because they can produce feedback to the BRM cell after the branch point has received the first BRM cell. Algorithm A4 has a very slow transient response especially when there are few or no BRM cells in the network, as for example during the initial state. The feedback delay of this algorithm depends on the round trip time to the most remote destination. However, when there are enough BRM cells in the network, this algorithm provides a reasonable transient response.

Algorithms A5, A6 and A7 improve the transient response in the overload condition by using various overload detection techniques. However, they still have a rather slow transient response in the underload condition. Moreover, these algorithms fail to detect the overload condition in some situations. As in algorithm A4, the algorithm A8 waits for the BRM cell to return from the most remote branch. Therefore, the transient response of this algorithm can be slow in certain situations, especially when there are very few or no BRM cells in the network.

C. Implementation complexity

Most previous studies have shows that turning around the RM cells causes a switch to high complexity. This causes algorithms A1, A2 and A9 to have a high complexity in turning around the RM cell. Algorithm A9 has the highest complexity with additional per-VC and per-branch variables to maintain the stored feedback information. Algorithm A3 introduces the simplest way to consolidate the feedback information at the branch point, since it does not turn around the RM cell, and it keeps the per-VC information minimal.

The algorithm A4 is more complex because it adds some variables to the switch to maintain the "wait-for-all" condition. Algorithm A5 adds complexity to algorithm A4 by introducing the overload detection technique, RM ratio control technique and immediate rate calculation technique. Algorithm A6 has a lower complexity than algorithm A5 but a higher one than algorithm A4, since they use the overload detection technique without both the RM ratio

control technique and the immediate rate calculation technique. Algorithm A7 merges algorithm A5 and algorithm A6 into a new algorithm with a timeout mechanism. This increases the implementation complexity in the switch.

Algorithm A8 adds complexity to algorithm A4 in increasing of per-VC and per branch variables. The proposed algorithm has slightly more complexity than A3 with additional temporary variables to memorize the information of the BRM cell before updating the feedback information.

D. Change to the newly joined

The ratio between the number of FRM cells generated by the source and the number of BRM cells received by the source should be approximately less than or equal to 1. Algorithm A1 and algorithm A9 generate a BRM cell for every FRM cell the branch point receives, therefore, the ratio of BRM cells to FRM cells is equal to 1. Algorithm A2 waits until the branch point receives at least one BRM cell before it generates and sends a BRM cell. Then the ratio of BRM cells to FRM cells is below one because the first FRM cell received will never be turned. For algorithm A3 and the proposed algorithm, each branch point sends a BRM cell whenever it receives the first BRM cell after an FRM cell, so that the ratio of BRM cells to FRM cells for this algorithm is less than or equal to one.

Algorithm A4 waits for all the BRM cells to return from all branches before feeding a BRM cell back to the upstream node, hence, the ratio of BRM cells to FRM cells for this algorithm is less than or equal to one. In algorithm A5, the overload detection technique is used and the branch points send an extra BRM cell back when the network is in the overload condition. This occasionally increases the number of BRM cells, however, the RM ratio control technique in this algorithm regulates this number to approximately the same as the number of the FRM cells received at the source in steady state.

Algorithms A6 and A7 use the overload detection technique without any RM ratio control technique, so the ratio of BRM cells to FRM cells for this algorithm is higher than one. In A8, each branch point waits for the BRM cell to return from the most remote destination and the ratio of BRM cells to FRM cells for this algorithm is less than or equal to one.

E. Change to the non-responsive branch

If some branches can not return the BRM cells to the branch point, algorithms A1, A2, A3, A9 and the proposed algorithm can properly work in this situation without any additional timeout mechanism because they use receipt of the FRM cell to determine when to feed BRM cell back to the upstream node. Therefore, if there are any FRM cells in the network, the branch point can send the BRM cell back to the source except if neither destination can return the

BRM cell. In algorithms A4, A5 and A6, the branch point must wait for the BRM cell to return from all branches. Therefore, the branch points must wait indefinitely if some branch can not return the BRM cells. This causes these algorithms to require a timeout mechanism to solve this problem

Algorithm A7 introduces a timeout mechanism to solve this problem. This causes this algorithm to work properly in this situation. However the branch point must wait for the timeout to expire before it sends an extra BRM cell. Algorithm A8 waits for the BRM cell to return from the most remote destination. If the non-responsive branch is the most remote one, then this algorithm can not work anymore. Thus, it requires an additional timeout mechanism.

F. Change to the newly joined

If there are new destinations joined to the multicast tree, the response from these branches could not effect feedback delay to the source. In this situation, algorithms A4, A5 and A6 must wait for the BRM cell to return from the newly joined branch. If the propagation delay of the new branch is very large then the transient response of these algorithms in this case is very slow and causes congestion or low link utilization in the network if there is a traffic condition changed while the branch point waits for the BRM cell from the new branch.

Algorithms A1, A2, A3, A9 and the proposed algorithm do not need to wait for the BRM to return from the new branch because the branch point can send BRM cell back to the upstream node after receiving the FRM cell. If any traffic condition is changed in the network after the new branch has been joined, these algorithms can quickly adjust the rate of the source to the rate the network is able to support. For algorithm A8, the branch point can send the BRM cells back to the upstream node and can respond to traffic conditions changed in the network because it always receives the BRM cell from the most remote destination. A7 faces a similar problem as A4, however, the timeout mechanism can reduce the waiting time by detecting the timeout of the new branch.

VII. CONCLUSION

Table 1 summarizes a performance comparison between various consolidation algorithms. The proposed algorithm provides good and desirable performance on all aspects, it can eliminate consolidation noise, has a fast transient response in both overload and underload conditions and has low implementation complexity in the switches. It works properly even when there are some branches joined to the multicast tree or not responding to the FRM cells. Algorithm A9 has very good performance with low consolidation noise and the fastest transient response, but it uses the highest implementation complexity. Algorithms A1

and A2 have a fast transient response but suffer from consolidation noise and require high complexity. Algorithm A3 provides good performance regarding transient response and implementation complexity but exhibits heavy consolidation noise. Algorithm A4 maintains reasonable performance in eliminating consolidation noise with fair complexity in the switch, but it has a very slow transient response. Algorithms A5, A6 and A7 can avoid congestion in overload conditions and remain free from consolidation noise. However, they display a very slow transient response in underload condition. Moreover, in some situations these algorithms fail to detect overload conditions in the network, leading to possible losses of cells at the switch. Algorithm A8 exhibits no consolidation noise but it has a very slow transient response and a high switch implementation complexity.

The ratio of BRM cells to FRM cells is stability maintained at one by algorithms A1 and A9. It is less than 1 in algorithm A2 and is less than or equal to one for algorithms A3, A4, A8 and the proposed algorithm. In the steady state, algorithm A5 stabilizes this ratio to the one closest to 1 in steady state.

Algorithms A4, A5 and A6 are very sensitive to either a newly joined branch or a non-responsive branch. Algorithm A7 circumvents this problem by using a timeout mechanism. However its performance largely depends on the selected timeout value. Algorithm A8 can partially manage these situations by only sending BRM cells from the most remote destination. The proposed algorithm and algorithms A1, A2, A3 and A9 can perform well in this case because they have appropriate feedback returning mechanisms.

VIII. ACKNOWLEDGEMENT

This work is supported in part by the Thailand Research Fund (TRF).

IX. REFERENCES

- [1] L. Roberts, "Rate based algorithm for point to multipoint ABR service", ATM Forum/94-0772R1, Nov. 1994.
- [2] H-Y. Tzeng and K-Y. Siu, "On max-min fair congestion control for multicast ABR service in ATM", IEEE JSAC, vol 15, no. 3, Apr. 1997, pp. 545-555.
- [3] W. Ren, K-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM", Proceedings of IEEE ATM'96, 1996.
- [4] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, Shivkumar Kalyanaraman, Sastri Kota, and Pradeep Samudra, "Feedback Consolidation Algorithms for ABR Point-to-Multipoint Connections in ATM Networks", Proceedings of IEEE INFOCOM 1998, volume 3, March 1998, pp. 1004-1013.
- [5] Sonia Fahmy, Raj Jain, Rohit Goyal, Bobby Vandalore, and Shivkumar Kalyanaraman, "Design and Evaluation of Feedback Consolidation for ABR Point-to-Multipoint Connections in ATM Networks", Journal of Computer

Communications, Vol. 22, Issue 12, 25 July 1999, pp. 1085-1103.

- [6] Tianji Jinag, Ellen W. Zegura and Mostafa Ammar, "Improved Consolidation Algorithm for Point-to-Multipoint ABR Service", IEEE ATM workshop Proceedings, May. 1998, pp. 195-201.
- [7] Hung-Shiun, Alex Chen and Klara Nahrstedt, "Feedback Consolidation and Timeout algorithm for Point-to-Multipoint ABR Service", ICC'99, vol 1, 1999, pp. 135-139.
- [8] Dong-Ho Kim, You-Ze Cho, Yoon-Young An and Yul Kwon, "A Scalable Consolidation Algorithm for Point-to-

Multipoint ABR Flow Control in ATM Networks", Proceedings of ICC'99, vol 1, 1999, pp. 118-123.

- [9] Supakiet Laohamatane, Luchakorn Wuttisittikulij and Prasit Prapinmonkolkarn, "A New Feedback Consolidation Algorithm for Point-to-Multipoint ABR Service", to appear in TENCON'99, Sept. 1999.
- [10] ATM Forum, "Traffic Management Specification", v4.0, Apr. 1996.
- [11] R. Jain et al., "ERICA Switch Algorithm: A Complete Description", ATM Forum cont. 96-117, Aug. 1996.

Table I: Comparison of consolidation algorithm

Algorithms	Robert	Tzeng	Ren	Wait-for-all	Immediate rate calculation
Consolidation noise	Very High	High	Very High	No	No
Transient response in underload conditions	Very Fast	Fast	Fast	Slow	Slow
Transient response in overload conditions	Very Fast	Fast	Fast ¹	Slow	
Generates BRMs	Yes	No	No	No	No
Change to non responsive branch	Low	Low	Low	High	High
Change to newly joined branch	Low	Low	Low	High	High
Ratio of BRM cells to FRM cells	1	< 1	≤ 1	≤ 1	Limit = 1
Resets stored feedback information at the end of cycle	Yes	Yes	Yes	Yes	Yes
Feedback information storing method	Per VC	Per VC	Per VC	Per VC	Per VC
BRM cell returning condition	Wait for FRM	Wait for FRM after BRM received	Wait for BRM after FRM received	Wait for BRM from all branches	Wait for BRM from all branches

Algorithms	Probabilistic aggregate	Timeout	Scalable	Supakiet	Proposed
Consolidation noise	No	No	No	No	Low
Transient response in underload conditions	Slow	Medium ²	Slow	Very Fast	Fast
Transient response in overload conditions	Medium ¹		Slow	Very Fast	Fast
Generates BRMs	No	No	No	Yes	
Change to non responsive branch			Medium	Low	Low
Change to newly joined branch		Medium ²		Low	Low
Ratio of BRM cells to FRM cells	> 1	> 1	≤ 1	1	≤ 1
Resets stored feedback information at the end of cycle	Yes	Yes	No	No	No
Feedback information storing method	Per VC	Per VC	Per VC and Per Branch	Per VC and Per Branch	Per VC
BRM cell returning condition	Wait for BRM from all branches or probabilistic	Wait for BRM from all branches or overload detection or probabilistic	Wait for BRM from most remote branch	Wait for FRM	Wait for FRM

Note 1. Depends on threshold 2. Depends on timeout value

4. The Call Admission Control Procedure

In possession of the three estimation methods we can build up the CAC procedure. First the traditional Chernoff bound is used. If $CLP < e^{-\gamma}$ then the call can be accepted and the procedure finishes else the Bahadur-Rao correction factor is calculated at s^* determined by the Chernoff bound. If $CLP < e^{-\gamma}$ then the call can be accepted and the procedure finishes else starting from s^* a linear search is performed to find s^* by means of $\Lambda(s)$ according to the theoretical background described earlier. If $CLP < e^{-\gamma}$ then the call can be accepted and the procedure finishes else the call must be rejected. The advantage of this three-step procedure lies in the fact that on the one hand in most cases it is enough to perform only the first one or two steps with insignificant calculation complexity to accept a new call and on the other hand step 3 provides very accurate estimation for those cases when after the first two steps we are not able to decide.

5. Simulation results

Because of the restricted size of the paper we picked up one of the simulation results to demonstrate the capabilities of the new bound. We applied the well known geometrical interpretation of CAC which was used e.g. by Hui. As a measure of performance we used the Gain Ratio which is defined in the following manner.

ACP_{method} := number of accepted traffic states by method and the GAIN of a given method as

$GAIN_{method} := ACP_{method} - ACP_{Ch}$ and $GAIN Ratio := \frac{GAIN_{Go}}{GAIN_{B-R}}$. In the simulations the capabilities of the original

Chernoff bound (Ch), the Bahadur-Rao bound (B-R) and the new method with Joint optimisation (Go) were compared as a function of the link capacity. The QoS parameter $\exp(-\gamma)$ was set up to 10^{-9} . We picked the well-known ON/OFF model and the test scenarios are shown on Table 1. As can be seen

in Figure 2, the ratio of the GAIN of Go is about 1.3-1.5 times greater than that of B-R.

	type (peak/average)	type (peak/average)
Scenario 1	Phone 64 kbps/32 kbps	Videophone 384 kbps/128 kbps
Scenario 3	Videconference 768 kbps/384 kbps	Data 4 Mbps/1 Mbps

Table 1 Test scenarios

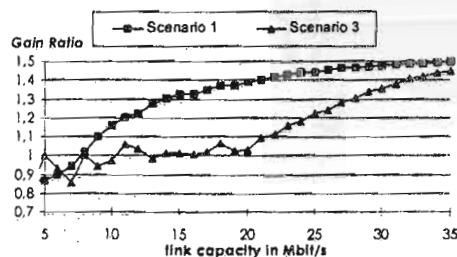


Figure 2. Comparison of B-R and Go in Scenario 1 and 3

References

- [1] A. Elwalid, D. Mitra, and R.H. Wentworth: "A New Approach for Allocating Buffers and Bandwidth to Heterogeneous Regulated Traffic in an ATM Node", *IEEE Journal on Selected Areas of Communications*, Vol.13, No. 6., pp. 1115-1127, 1995.
- [2] R. G. Gallager: *Information Theory and Reliable Communication*, John Wiley and Sons, 1968.
- [3] J. Hui: *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, 1990.
- [4] L. Pap, S. Imre, J. Leventovszky, van der Meulen, E.C. and P. Pozsgai: "Tail Distribution Estimation for Call Admission in ATM Networks", *Third Workshop on Performance Modelling and Evaluation of ATM Networks*, Bradford U.K., 3rd-5th July, pp. 46/1-46/11, 1995.
- [5] J. Roberts, U. Mucci and J. Virtamo: *Broadband Network Teletraffic, Performance Evaluation and Design of Broadband Multiservice Networks*, Springer, Berlin, Germany, 1996.

An E-FMMRA for ABR point-to-multipoint Service in ATM Network

Supakiat Laohamatane, Luchakorn Wuttisittikulij and Prasit Prapinmonkolkarn

Department of Electrical Engineering, Chulalongkorn University
Phayathai Road, Bangkok 10330, Thailand.

Tel. (662) 218-6512, Fax. (662) 251-8991, Email: lunch@ee.eng.chula.ac.th

Abstract

This paper describes how the E-FMMRA point-to-point congestion control algorithm can be applied to ABR point-to-multipoint connections in ATM networks. The performance evaluations of this system are carried out using five different consolidation techniques and the results has been compared with that of another congestion control algorithm namely ERICA. It is found that for certain types of consolidation techniques when applying with the E-FMMRA, its performance can be substantially improved in comparison to the ERICA. This indicates that different types of congestion control algorithms may not always result in the same level of performance, and it is important to carry out a separate investigation for each individual congestion control algorithm.

I. Introduction

Recent emerging point-to-multipoint applications such as data distribution, tele-conferencing and etc have created a new networking challenge to ATM network system designs. Congestion control for ABR point-to-multipoint connections is one of the key problems that need to be addressed. Previous work on this problem has shown that existing point-to-point congestion control algorithms can be applied to point-to-multipoint environments. However, this is not a straightforward matter, because there exists a large amount of feedback information from all the member branches continuously sent back to the source. This problem is known as a feedback implosion problem. To solve this problem, each branch point is required to consolidate the feedback information, so that the amount of feedback information to the source node is kept to a minimal and sufficient level. At present, it is not fully clear how the feedback information should be properly consolidated. However, a number of useful algorithms have been proposed, see [1-4] for examples. A brief overview on some important types of consolidation algorithms is shown as following.

Robert [1] first proposed a consolidation algorithm, which has an objective of achieving the fastest transient response. The basic idea of this algorithm is that each branch point consolidates the feedback information on a per-VC basis and passes it on to its upstream node through a BRM (Backward Resource Management) cell after the branch point have received a FRM (Forward Resource Management) cell. With this technique, consolidated feedback information may not reflect the actual congestion level of all the branches, because the feedback from some branches may have not yet arrived at the time when the BRM cell is sent. This could cause undesired consolidation noise. To reduce such consolidation noise, Tzeng [2] improved this algorithm by ensuring that the feedback to the upstream node is sent only when at least one BRM cell has been received from a downstream branch before receiving a FRM cell. This algorithm is able to reduce the level of consolidation noise. Ren [3] argued that generating of a BRM causes switch to higher complexity and proposed two alternative algorithms. In his first algorithm, the first BRM cell received is passed on to the source after a FRM cell has been received. Note that all the three algorithms described so far cause consolidation noise. For the second algorithm of Ren, he presented an algorithm that can eliminate the consolidation noise entirely. The algorithm is referred to as "wait-for-all" algorithm. In this algorithm, each branch point waits for BRM cells from all the branches to arrive, before a BRM cell is sent to the source. Although the problem of consolidation noise is completely removed, the waiting time for all feedback information can be long, causing very slow transient response. Fahmy [4] proposed an immediate rate calculation algorithm to improve the transient response of the wait-for-all algorithm. In this algorithm, an overload detection mechanism is added to the wait-for-all algorithm. This mechanism allows each branch point to react immediately to any rapid change of overload condition, thus enhancing the transient response.

So far, most if not all studies pay their attention to a single algorithm namely ERICA (Explicit Rate Indication for Congestion Avoidance) [5] as the underlying congestion control. It is not clear whether the results from these studies will be applicable to other congestion control algorithms. Therefore, in this paper

we present a study of another type of congestion control algorithm, known as E-FMMRA (Enhance-Fast Max-Min Rate Allocation) for point-to-multipoint ABR connections using five different consolidation algorithms.

II. The E-FMMRA for point-to-multipoint connection

The E-FMMRA [6-8] is a congestion control for ABR point-to-point connections. Each switch calculates an appropriate ER (explicit rate) to provide Max-Min fairness and passes it back to every corresponding source. At any given link in the network, connections that are competing for the bandwidth can be grouped into two categories:

- Bottlenecked connections: these connections are unable to achieve the fair rate determined by the switch because of their limitation of peak cell rate or rate allocated by other switches.
- Non-Bottlenecked connections: these connections can fully use the bandwidth allocated by the switch.

The key mechanism of E-FMMRA is that when some connections are unable to use their share of bandwidth, the remaining and unused bandwidth will be made available to other connections that need more. To achieve this, two variables β and λ are required for keeping the bottlenecked status and the bottlenecked bandwidth respectively of each connection. Based on the information, the switch will be able to assign a proper amount of bandwidth for each connection. Note that this specific mechanism is not found in the ERICA algorithm.

To apply the E-FMMRA algorithm, which is a point-to-point congestion control, to a point-to-multipoint environment, two issues must be addressed. The first issue is associated with the E-FMMRA directly. A slight modification to the variable structure is needed. The number of variables, β and λ , have to be increased from storing in a per-VC basis to storing in a per-VC and per-branch basis. For the second issue, a suitable type of consolidation algorithm has to be selected. In this paper, all the existing consolidation algorithms described above are taken for this investigation.

III. Simulation Results

A. Parameter Setting

- In the simulation, the following parameter values are used
- All links have a bandwidth of 150 Mbps.
 - All sources have PCR = 150 Mbps, RIF = 1, RDF = 0.125, Nrm = 32, ICR = 150 Mbps.
 - All point-to-multipoint traffic flows from the root to the leaves of the tree. No traffic flows from the leaves to the root, except for RM cells. The same assumption applies for the point-to-point connection.
 - The sources are persistent; i.e. there is always data to send.
 - The ERICA congestion control algorithm has a target utilization of 90% and the switch measurement interval set to a minimum of the time to receive 100 cells within 1 msec for both ERICA and E-FMMRA.

B. Network Model

A chain configuration as shown in Figure 1 is used to illustrate the problems of consolidation noise and transient response. Figures 2 and 3 show the ACR changes of source A and B when using ERICA and E-FMMRA switches respectively. As shown in Figure 2, ERICA suffers from consolidation noise when it is used with Robert's, Tzeng's and Ren's consolidation algorithms. This is because during the transient period ERICA will gradually adjust the rate of source B to its fairshare and store it in the MaxAllocCurrent register and uses it as MaxAllocPrevious in next interval. The branch point at SW3, see Figure 1, uses this value to calculate ER and slightly increases the ACR of source A. Therefore, source A will generate more FRM cells and the number of FRM cells received at the branch point will increase, requiring the branch point to send BRM cells back to the source more frequently. This temporarily shortens the feedback interval period and causes the incomplete feedback information. When the incomplete feedback information is passed on to the source, the ACR at source A will begin to oscillate. This first ACR oscillation continually affects to the next ACR oscillation. In the contrary, the wait-for-all algorithm can completely eliminate consolidation noise, but it suffers from very slow transient response. The immediate rate calculation algorithm gives the best performance for ERICA with no consolidation noise and fast transient response.

In FMMRA as shown in Figure 3, Robert's and Tzeng's algorithms act no consolidation noise while providing fast transient response. This is because E-FMMRA can adjust the source rate to its fairshare very fast and a decrease in ACR of source B does not affect ACR of source A. So the first oscillation of ACR of source A will not occur and thus no subsequent oscillation of ACR can take place. For Ren's algorithm, there still remains some consolidation noise, although it is not as much as in the case of ERICA. This is due to the mechanism of passing back the first BRM cell received to the source after having received a FRM cell. The wait-for-all and immediate rate calculation algorithms have the same results as in the case of ERICA. This is because the main objective of these algorithms is eliminating consolidation noise, thus no difference in their performance is observed.

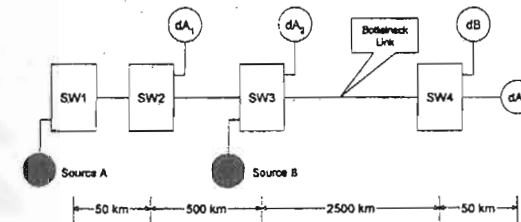


Figure 1: Network Model

IV. Conclusion

This paper has shown that the E-FMMRA congestion control algorithm can be applied to a point-to-multipoint ABR service in ATM networks. It is found that the E-FMMRA algorithm can help to reduce consolidation noise when used with some consolidation algorithms in ABR point-to-multipoint connections in comparison to the ERICA counterpart. The E-FMMRA algorithm can work satisfactorily well with three consolidation algorithms, i.e. Robert's, Tzeng's and immediate rate calculation, while the ERICA performs well only with the immediate rate calculation algorithm.

V. References

- [1] L. Roberts, "Rate based algorithm for point to multipoint ABR service," *ATM Forum/94-0772R1*, Nov. 1994.
- [2] H-Y. Tzeng and K-Y. Siu, "On max-min fair congestion control for multicast ABR service in ATM," *IEEE JSAC*, vol 15, no. 3, pp. 545-555, Apr. 1997.
- [3] W. Ren, K-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," *Proceedings of IEEE ATM'96*, 1996.
- [4] S. Fahmy, et al., "Feedback consolidation algorithms for ABR point-to-multipoint connections," *Proceedings of IEEE INFOCOM'98*, 1998.
- [5] R. Jain et al., "ERICA Switch Algorithm: A Complete Description," *ATM Forum cont. 96-117*, Aug. 1996.
- [6] A. Arulambalam, X. Chen and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks," *IEEE comm. Mag.*, Nov. 1996, pp. 92-100.
- [7] A. Arulambalam, X. Chen and N. Ansari, "An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks" *Proceedings of IEEE ICC'97*, June 1997, pp. 200-204.
- [8] F. Chiussi, A. Arulambalam, Y. Xia and X. Chen, "Explicit Rate ABR Schemes Using Traffic Load as Congestion Indication" *IEEE Proceedings Sixth International Conference*, Sept. 1997, pp. 76-84.

Analysing the versatility of the 2-MMPP traffic model

A. Nogueira, R. Valadas
Universidade de Aveiro / Instituto de Telecomunicações, 3810-193 Aveiro, Portugal
e-mail: nogueira@av.it.pt; rv@ua.pt

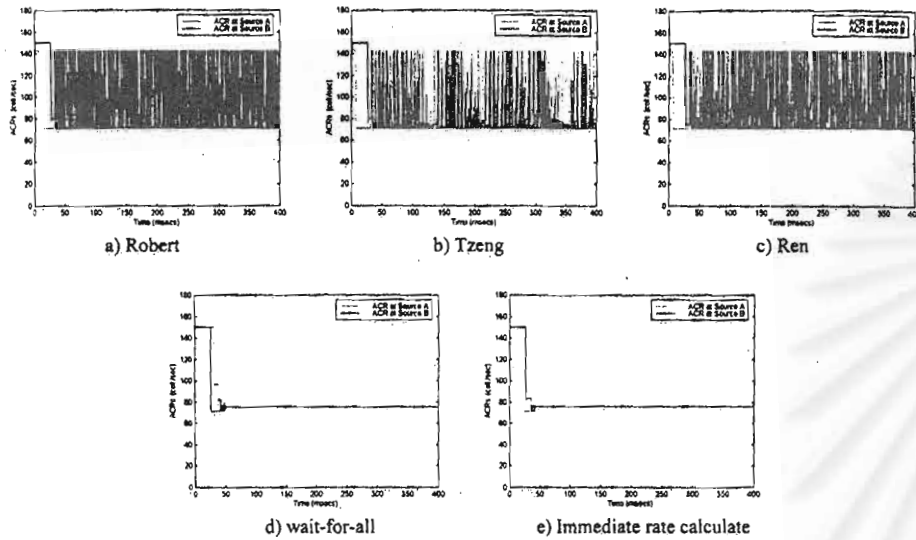


Figure 2: Allowed Cell Rate at source using ERICA

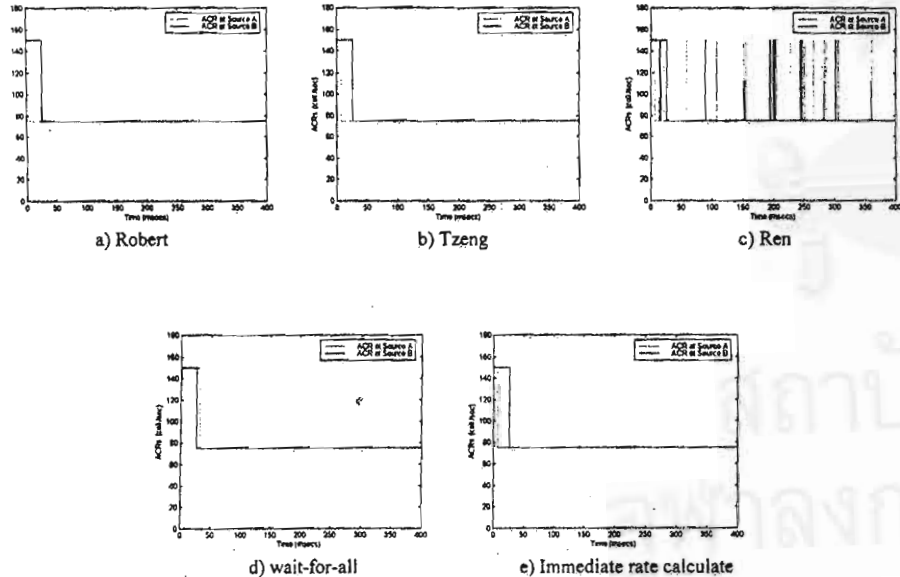


Figure 3: Allowed Cell Rate at source using E-FMRA

Summary

Cell loss is an important measure of Quality of Service (QoS) in ATM networks. Cell loss performance of ATM elements handling bursty sources must be evaluated in order to provide guaranteed QoS to users and to dimension network resources correctly.

In this paper, we make a comparative study of two fitting algorithms for 2-MMPP ATM traffic models, both based on cell interarrival times: one fits the cumulative distribution and the other fits the first three moments and auto-covariance function. The 2-MMPP is then evaluated as an approximation to 3-MMPP, 5-MMPP, IPP, IDP, Hyper-Exponential On-Off and Self-Similar traffic sources. The usefulness of both fitting procedures for cell loss prediction is evaluated and compared. The proposed algorithms can be used in the characterisation of ATM traffic streams and in connection admission control procedures.

1. Introduction

Efficient operation of ATM networks and provision of guaranteed Quality of Service (QoS) for many services to be supported in such networks will require proper modelling and control of ATM traffic. Cell loss performance of ATM elements, which is one of the fundamental measures of QoS, must be evaluated in order to provide guaranteed performance levels to users and to dimension network elements correctly.

On-Off models have been commonly used for representing bursty sources. Particularly, the IPP model has been found to adequately represent the behaviour of voice sources and it has been proposed as a model for other traffic types, including video, data and multimedia traffic. The MMPP model, with a variable number of states, is a non-renewal model that has also been widely used for modelling ATM traffic. However, findings in a number of packet-based network scenarios suggest that traffic in such networks presents a certain degree of self-similarity.

The proliferation of different traffic models and their application in various (and sometimes quite analogous) scenarios have motivated us to study the conditions of applicability for different traffic

models (and their associated estimation procedures) to certain traffic scenarios. Specifically, here we make an exhaustive study of the 2-MMPP process and two of its associated fitting procedures, analysing their suitability in approximating various types of arriving flows. In particular, we have considered 3-MMPP, 5-MMPP, IPP, IDP, On-Off sources with hyper-exponential On and Off period distributions, which are more general than exponential On-Off sources but are still quite tractable, and Self-similar traffic models.

The rest of the paper is structured as follows: in section 2, we briefly summarise various commonly used point process superposition approximations based on the 2-MMPP process; in section 3, an outline of the methodology followed for this study is provided and in section 4 the main results are presented. Finally, in section 5, we will draw some conclusions and foresee the most relevant future tasks.

2. 2-MMPP fitting procedures

In order to characterise the input stream at an ATM multiplexer, two main approaches were suggested in the literature: point process and fluid flow approximations. We will focus on non-renewal point process approximations based on the MMPP process, since such approximations have been shown to capture both the cell scale and burst scale characteristics of traffic [2]. A number of approaches have been proposed in the literature in which the superimposed stream is modelled by a 2-MMPP. These approaches differ in the choice of the traffic characteristics used to match the 2-MMPP parameters and can be divided in two groups: methods based on cell counting statistics ([2], [3] and [4]) and methods based on interarrival time statistics ([1], [5] and [6]).

Assuming interval-stationary 2-MMPP processes, where X_i represents the interarrival time between the i^{th} and $(i+1)^{\text{th}}$ cells, the distribution of the interarrival time X_i is a second order hyperexponential distribution (H_2) with complementary CDF:

$$F_c(x) = qe^{-\mu_1 x} + (1-q)e^{-\mu_2 x}, \quad 0 < q < 1 \quad (1)$$

and density function given by

$$f(x) = q\mu_1 e^{-\mu_1 x} + (1-q)\mu_2 e^{-\mu_2 x}, \quad 0 < q < 1. \quad (2)$$

An Improved EPD Algorithm with Dynamic Threshold Technique for TCP in ATM Networks

Thaksayot Keadtisuke, Tanun Jaruvitayakovit, Prasit Prapinmongkolkarn
and Lunchakorn Wuttisittikulkij

Telecommunication System Research Laboratory, Department of Electrical Engineering,
Faculty of Engineering, Chulalongkorn University, Phaya Thai Road, Bangkok 10330, Thailand
Phone. (662)218-6517, Fax. (662)251-8991, Email: a0287421@student.chula.ac.th

Abstract

This paper proposes an Early Packet Discard (EPD) algorithm with Dynamic Threshold to improve the performance of TCP applications over UBR service in ATM networks. In congested networks, the performance of TCP is degraded because a portion of the available bandwidth of the network is wasted. Early Packet Discard algorithm was proposed to improve performance of TCP. However, the EPD algorithm can not improve fairness among competing VCs in a congested network. Improvement fairness can be used with various algorithms such as EPD with per-VC accounting techniques. We propose an improved EPD algorithm using a Dynamic Threshold technique. With our technique it can achieve better fairness and throughput among multiple TCP connections. Simulation results of EPD using Dynamic Threshold technique are presented to illustrate the significant improvement performance of TCP in term of fairness and throughput compared with existing Early Packet Discard over UBR service in ATM networks.

1. Introduction

For applications like non real-time data traffic, ATM Forum has defined two different service classes, Available Bit Rate (ABR) service and Unspecified Bit Rate (UBR) service [1]. The UBR service can use any available bandwidth and is not sensitive to cell loss or delay. During congestion in the network, data cells are discarded. But their sources are not expected to reduce their transmission cell rates, because UBR service does not use closed loop feedback control. Then the higher layer protocol should be employed to control the traffic in UBR service. Since transmission control protocol (TCP) is perhaps the most widely used transport layer protocol in the existing data networks, the performance of TCP in ATM network is a major interest to ATM equipment vendors and service providers. The UBR service can not provide good performance of TCP because it does not have congestion control and reliability. Accordingly, Romanow [2] had proposed EPD algorithm for UBR service in which the throughput performance was improved but the fairness was not. To improve the fairness various algorithms such as EPD with per-VC accounting [3] or fair buffer allocation schemes [4] can be applied. However, in some cases EPD with per-VC accounting can not achieve good throughput [5]. In this paper, we propose a new algorithm that can improve both fairness and throughput performance simultaneously.

In the following section, we shall discuss the ideas of the Early Packet Discard (EPD) algorithm. The improvement of EPD with Dynamic Threshold technique,

will be described in section 3. Next section, we present a simulation model and parameters of simulation environment that used TCP over UBR service in ATM networks. In section 5, we show simulation results and analyzes the performance of TCP. Finally we summarize the Early Packet Discard algorithm with Dynamic Threshold in section 6.

2. Early Packet Discard (EPD) Algorithm

When the buffer in congested switches is full, all the arriving cells will be discarded. Basically, discarded cells belong to different packets. As a result, the rest of cells of packets are transmitted into the networks but could not be completely reconstructed to the original packets at the destination, and this is referred to as the fragmentation problem [2]. If at least one cell in the packet is discarded, the destination can not reconstruct the packet completely; this will cause packet retransmission at the source. The EPD algorithm drops the entire packets instead of partial packets. As a result, the link does not carry incomplete packets. As a consequence, the network bandwidth will be better utilized. A threshold is set at the switches. When queue length exceeds the threshold, all incoming cells from new packets are discarded. The received packets that are over the threshold are still accepted if there is a buffer space available.

The EPD algorithm used in our study is shown in Table 1. Table 1 demonstrates the efficiency and fairness for the EPD algorithm. Throughput can be improved with the EPD algorithm but it can not provide fairness. The EPD algorithm discards packets randomly so that it spreads packet loss over many sessions.

To overcome the drawbacks, we proposed improved EPD with *Dynamic Threshold technique*. The EPD with dynamic threshold techniques aims to minimize bandwidth wastage while maintaining fairness. The dynamic threshold technique is activated only when congestion is detected in the network. Dynamic Threshold technique confines packet discarding to switch buffer, it is an algorithm for buffer allocation. Next section describes the details of the EPD with Dynamic Threshold technique.

3. Early Packet Discard with Dynamic Threshold Technique

The EPD with Dynamic Threshold technique consists of four thresholds. The first threshold is called High threshold. If the current queue length exceeds this High threshold any incoming cells are discarded. The second threshold is named Low threshold. If the queue length is less than Low threshold any incoming cells are accepted into the

switch. The third threshold denotes as fair share threshold, which aims to guarantee fairness among all sessions. When the total demand exceeds the available bandwidth, the bandwidth is divided equally among the competing VC's, that is,

$$\bar{Th} = \frac{\text{High threshold}}{K} \quad (1)$$

\bar{Th} is the fair share of VC connections. Let K is the number of VC's competing for total available bandwidth. Also, let X_i is the queue length of VC $_i$. Total queue length (Q) is equal to the summary of the queue length of every VC's connection.

$$Q = \sum_{i=1}^K X_i \quad (2)$$

Among the established TCP connections, some may be idle, and some TCP connections may be active but they use less bandwidth than that being allocated. Then,

$$Q = \sum_{i > \bar{Th}} X_i + \sum_{i < \bar{Th}} X_i, \quad (3)$$

expression $X_i < \bar{Th}$ implies that the session uses more bandwidth than the fair share. Whereas $X_i > \bar{Th}$ implies that the session is idle. Some VCs may be idle causing left bandwidth in the switch. Therefore this bandwidth is shared to the session with more data. Equation (3) can now be expressed as

$$\hat{Th} = \frac{\text{High threshold} - \sum_{i < \bar{Th}} X_i}{K_{X_i > \bar{Th}}} \quad (4)$$

The Dynamic Threshold, \hat{Th} in equation (4) is used, when the queue length is between High threshold with Low threshold and Q_i is more than the fair share.

4. Simulation model and parameters

In this section, we describe the simulation environment that used TCP over UBR service in ATM networks. We simulate two different parameter sets with the network models shown in Fig 1 and Fig 2. In network model 1 scenario, there are two ATM switches, 10 sources and each link has a delay of $2\mu s$. For the first simulation set, all sources are persistent. The results are shown in Tables 1 and 2. In the other set of simulation, there are 5 bursty sources (no. 1-5) having 1MB data to transmit and 5 persistent sources (no. 6-10). In this paper, we assume that the buffer in ATM switches is 2000 cells, Low threshold of EPD with Dynamic Threshold technique is fixed at 1000 cells, TCP window size is 64 KB, minimum time out is 100 ms and packet size of TCP is 8 KB. In network model 2, there are six ATM switches, 10 sources and each link has a delay of $2\mu s$.

Table 5 shows simulation results for all persistent sources and Table 6 presents simulation results for bursty sources (no. 1-3) and persistent sources (no. 4-10).

5. Simulation results and analysis

Table 1 shows the fairness of the conventional EPD and the proposed EPD with Dynamic Threshold. From simulation results, EPD with Dynamic Threshold achieve higher fairness because the algorithm counts incoming cells and drops cells that belong to the higher usage sources. Table 2 shows the efficiency comparison of the EPD algorithm and the EPD with Dynamic Threshold using different values of the high threshold. It is shown that higher values of threshold in both algorithms leads to better efficiency of the network usage. However, the efficiency of 1900-cell threshold is lower than of 1800-cell threshold because the remaining cells of packets are transmitted into the network. The destination can not reassemble the complete packet which will cause wastefulness of bandwidth. This situation is called "fragmentation problem". Table 3 shows the fairness of the network model 1 when the sources are persistent and bursty. The EPD with Dynamic Threshold achieves higher fairness at either high or low values of high threshold because Dynamic Threshold discards cells of high usage VCs. Table 4 presents the efficiency of persistent and bursty sources. The simulation results show that EPD with Dynamic Threshold achieves higher efficiency than the EPD algorithm.

Table 5 shows efficiency and fairness of all persistent sources of the network model 2 which is named multi hop scenario. It is shown that EPD with Dynamic Threshold achieves higher efficiency and fairness than the EPD algorithm because EPD with Dynamic Threshold counts cell if any VC has amount of cells more than the fair share then cells of this VC are discarded. Fig 2 shows received packets of all persistent sources. It is shown that the number of received packets in source 10 of the EPD algorithm is less than the others because it has the smallest propagation delay. Then the corresponding destination quickly detects cell loss cause source 10 retransmits the lost packet loss again. Fig 3 (b) shows that EPD with Dynamic Threshold has the highest fairness because Dynamic Threshold discards cells of higher usage VCs. The received packets of source 10 is significantly improved by EPD with Dynamic Threshold. Table 6 shows the efficiency and fairness of the persistent and bursty sources. The simulation results show that the amount of received packets is less than the case that all sources are persistent when the sources are bursty. In addition, the efficiency and fairness of EPD with Dynamic Threshold is higher than EPD algorithm by approximately 10% and 3% respectively.

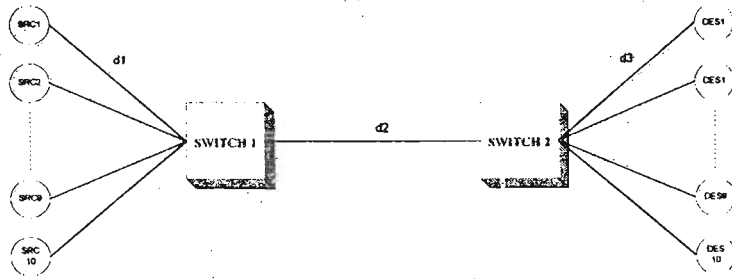


Fig1 Network model 1

Table 1 Fairness of Network model 1 for persistent sources.

Threshold (cells)	1900 (95%)	1800 (90%)	1700 (85%)	1600 (80%)	1400 (70%)	1200 (60%)	1000 (50%)
EPD algorithm	0.9436	0.9326	0.9354	0.8682	0.9576	0.9797	0.9461
EPD with Dynamic Threshold	0.9917	0.9673	0.9869	0.9615	0.9844	0.982	-

Table 2 Efficiency of Network model 1 for persistent sources.

Threshold (cells)	1900 (95%)	1800 (90%)	1700 (85%)	1600 (80%)	1400 (70%)	1200 (60%)	1000 (50%)
EPD algorithm	83.59	87.14	87.09	84.66	83.63	72.49	66.81
EPD with Dynamic Threshold	89.54	92.63	89.54	85.78	85.47	80.63	-

Table 3 Fairness of Network model 1 in case of using both persistent and bursty sources.

Threshold (cells)	1900 (95%)	1800 (90%)	1700 (85%)	1600 (80%)	1400 (70%)	1200 (60%)	1000 (50%)
EPD algorithm	0.9609	0.9858	0.9666	0.9158	0.9533	0.8766	0.93
EPD with Dynamic Threshold	0.9975	0.9929	0.9906	0.9955	0.9898	0.9525	-

Table 4 Efficiency of Network model 1 in case of using both persistent and bursty sources.

Threshold (cells)	1900 (95%)	1800 (90%)	1700 (85%)	1600 (80%)	1400 (70%)	1200 (60%)	1000 (50%)
EPD algorithm	76.06	80.62	82.46	74.44	75.33	55.2	48.83
EPD with Dynamic Threshold	91.35	90.75	90.03	83.55	78.2	63.32	-

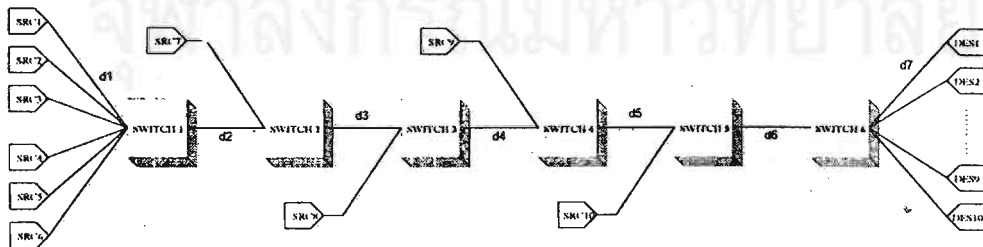


Fig. 2 Network model 2

Table 5 Simulation results of Network model 2 for persistent sources.

Algorithm	Early Packet Discard (EPD)	EPD with Dynamic Threshold
Received packet (packet)	9321	10091
Fairness index	0.9745	0.9951
Efficiency (%)	87.80	94.58

Table 6 Simulation result of Network model 2 in case of using both persistent and bursty sources.

Algorithm	Early Packet Discard (EPD)	EPD with Dynamic Threshold
Received packet (packet)	5608	6211
Fairness index	0.9629	0.9923
Efficiency (%)	82.12	92.04

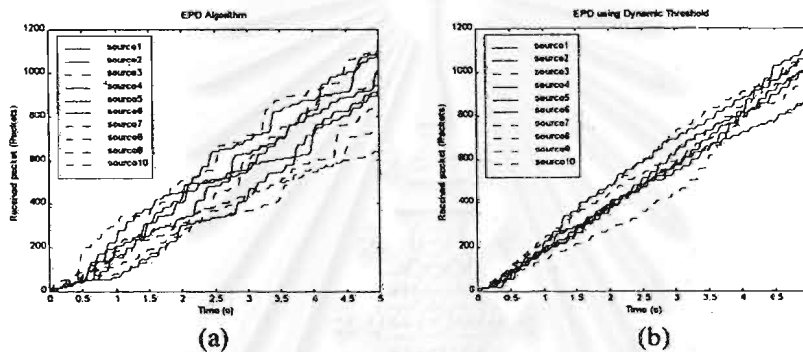


Fig 3 Received packet of persistent sources in network model 2

6. Conclusion

In this paper, we propose the improved EPD algorithm called "*Dynamic Threshold Technique*". The algorithm accomplishes good performance in case of both throughput and fairness by attempting to discard packets of the higher usage sources which cause congestion. In contrast, the conventional EPD algorithm can reach only the good throughput.

The simulation has shown that EPD with Dynamic Threshold achieves better efficiency and fairness than the conventional one in both cases, i.e. using all persistent sources and using combination of persistent and bursty sources. The proposed algorithm can fulfill in the manner of fairness because it controls buffer occupancy by counting cells in switches and discarding cells that belong to high activity sources. In addition, the better throughput can be achieved because the proposed algorithm shares bandwidth among active sources only

Acknowledgement

The research work is partly supported by Prof. Prasit Papinmongkolkarn's fund, Chulalongkorn University, and TRF.

References

- [1] "ATM Forum Traffic Management Specification Version 4.0", ATM Forum, April 1996.
- [2] A. Romanow and S. Floyd. "Dynamics of TCP Traffic over ATM Networks", IEEE Journal on Selected Areas in Communications. pages 633-641, vol.13, No.4, May 1995.
- [3] H. Li, K.-Y. Siu, H.-Y. Tzeng, C. Ikeda, and H. Suzuki. "Performance of TCP over UBR service in ATM Networks with Per-VC Early Packet Discard Schemes," Proc. Computers and Communications 1996, pages 350-357.
- [4] R.Goyal, R. Jain, S. Kalyanaraman, S. Fahmy and Seong-Cheol Kim, "UBR+: Improving Performance of TCP over ATM-UBR Service," Proc. ICC'97, June 1997.
- [5] Go Hasegawa, H. Ohsaki, M. Murata and H. Miyahara. "Performance of TCP over ABR Service Class," Proc. GLOBECOM'96 pages 1935-1941 vol3.

Common Buffer Computation Credit Based Flow Control in ATM Networks

Surachet Sookjaroen, Tanun Jaruvitayakovit, Prasit Prapinmongkolkarn
and Lunchakorn Wuttisittikulkij

Telecommunication System Research Laboratory, Department of Electrical Engineering,
Faculty of Engineering, Chulalongkorn University, Phaya Thai Road, Bangkok 10330, Thailand
Phone (662) 218-6517, Fax. (662) 251-8991, Email: yoong@ee.eng.chula.ac.th

Abstract

Credit based flow control is a hop by hop flow control that can guarantee cell loss in ATM networks. Conventional credit-based flow control scheme requires high computation complexity and has to maintain its buffer in a per virtual connection (VC) manner. In the other word, the scheme complexity highly grows in respect to the number of increasing VCs. The complexity is an order of n where n is the number of virtual connections. We propose computation base common-buffer for credit-based flow control. The proposed scheme reduces the switch complexity from an order of n to an order of 1 that is independent from the number of traversing VCs. From simulations the overall performance is still not worse and the fairness is better than the conventional scheme.

switch 1 and switch 2. From the Figure, there are two control loops between Host 1 and switch 1 as the effect from the number of traversing connections .

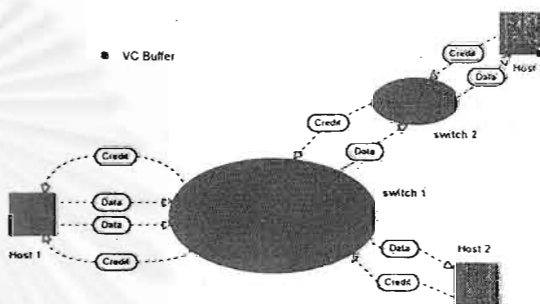


Figure 1. Credit-based flow control

1. Introduction

ATM technology was designed to support various kind of services such as video, data, multimedia and etc. Available Bit Rate (ABR) service is one of the service class standardized by the ATM forum. ABR service uses bandwidth left over from Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services, so the resource can be used efficiently. ABR service was classified into two classes which are rate-based flow control and credit-based flow control. Rate based is an end to end flow control so it can guarantee end to end delay. An example of this control scheme is the ERICA scheme [2]. Credit based flow control is a hop by hop flow control, so it cannot guarantee end to end delay but it can guarantee cell loss. Guarantee cell loss is strong point of credit based flow control especially in the network with TCP applications that have often to retransmit data when the TCP window size is larger than the switch buffer.

2. Complexity issue of credit-based

As mentioned above credit based flow control is a hop by hop flow control. Switch buffer is the main parameter for credit computation. In addition, the scheme has to compute and maintain used buffer in a per VC manner. The switch sends feedback information to inform sources regarding the network congestion via credit cells. The credit cells carry a value of total buffer space available at the downstream switch. If the feedback buffer space is small, or the upstream node cannot receive credit cells, it means that there is congestion in the network. The model of credit based flow control is shown in Figure 1.

Host 1 establishes a connection to Host 2 through switch 1 and also has another connection to Host 3 through

N23 is a well-known credit-based schemes. In the scheme, the switch buffer is divided into n sections, where n is the number of traversing VCs. Every section is divided into two zones, i.e. N2 and N3. The downstream node is eligible to send a credit cell for a VC if it has forwarded N2 data cells of that VC since the previous credit cell has been sent or the timer is expired. The credit cell contains a credit value that is equal to the number of free buffers in N2 and N3 zones. The computation of credit cell is shown in Figure 2 as credit update protocol.

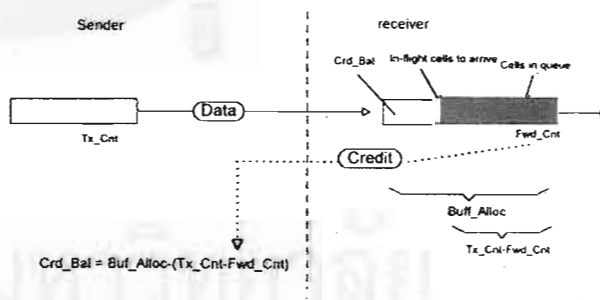


Figure 2. Credit update protocol

From Figure 2, Tx_Cnt is the inflight cell from upstream node and Fwd_Cnt is the forwarded cell from current node. Crd_Bal, which is the credit value of credit cell computed from the total switch buffer (Buf_alloc) minus the amount of switch buffer (Tx_Cnt-Fwd_Cnt) as shown in (1).

$$Crd_Bal = Buf_Alloc - (Tx_Cnt - Fwd_Cnt) \quad (1)$$

After receiving a credit cell, the upstream node is eligible to forward number of data cells of that VC to the downstream node according to the received credit information. Each time the upstream node forwards a data cell of a VC, it decrease its current credit for the VC by one. If credit value reaches zero, it means that there is no permission to send any data cell from this node. Hence, there is no data cell loss at the downstream switch.

3. Common buffer computation credit-based

The computation complexity of the conventional scheme depends on the number of VCs. In order to reduce the complexity, the common buffer computing technique is employed. The proposed scheme is developed from N23 scheme and the downstream node is eligible to send credit cells each time if it has forwarded $n \times N2$ data cells of all VC since the previous credit cell or the timer is expired. The downstream node sends credit cells back to all of corresponding upstream node with the credit value that is ratio between the available switch buffer space and n .

$$\text{Credit} = \text{Crd_Bal}/n \quad (2)$$

All upstream nodes will receive a common credit, which is the fairshare of the switch buffer. From (2) Crd_Bal is the number of free buffer space in the combined area of the N2 and N3 zones, similar to N23 schemes. When credit cells are sent, Crd_Bal will be computed as in (1) and the credit cells will be sent with credit value which is computed from (2).

After receiving credit cells, sources are eligible to send data cells according to credit value in the received credit cells. Similar to N23 scheme, each time the upstream node forward a data, it decrease its current credit for the VC by one. If the credit value reaches zero, it means that there is no permission to send any data cell.

According to the rate of sending data cells, a source approximates the times to received the next credit cell to be the time between the last received credit cell and the current credit cell. Thus the source should send data cells equal to the credit value of the current credit cell in this period of time. Allowed cell rate at the source should compute from (3).

$$\text{ACR} = \text{Credit}/\text{RT} \quad (3)$$

When RT is times between the last credit cell and the current credit cell.

For the computation complexity point of view, during the same period of times common buffer scheme computes credit update protocol less about n times than N23 scheme where n is the number of active VCs in the switch.

The proposed common buffer computation also reduce system parameters due to the computation behavior.

4. Simulation

In this section, three network scenarios are simulated to evaluate the performance of the proposed scheme as shown in Figure 3, 5 and 8. Network model 1

shown in Figure 3 is one hop network model. There are 5 sources sending data all the time to the bottleneck switch. Network model 2 shown in Figure 5 is also a one hop network model but there are 15 sources which send data around 20 percent of the whole periods that is an on-off source behavior. Finally, the network model 3 shown in Figure 8 is also one hop network model. There are 10 sources, sources 1-5 send data around 3.3 percent of the whole period and the others send data all the times. With sending data only 3.3 percent of period in source 1-5, this means that these sources are bottleneck sources.

4.1 One hop network model

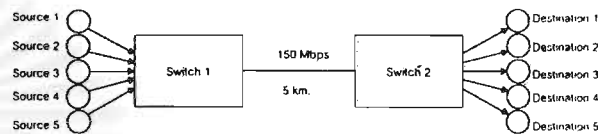


Figure 3. Network Model 1

The result of simulation network model 1 is shown in term of the total switch queue length and the received cells as in Figure 4 and Table 1 respectively.

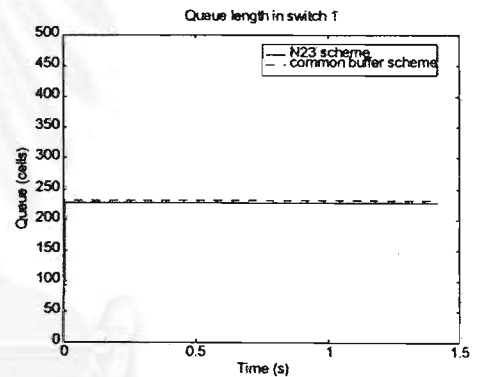


Figure 4. Queue length of switch 1

Table 1 receive cells at Destination

VC	VC 1	VC 2	VC 3	VC 4	VC 5
N23	99995	99996	99995	99994	99995
Common Buffer	99995	99994	99995	99997	99996

From network model 1, there is no significant difference between N23 scheme and common buffer scheme either switch queue length or amount of received cells because in this model there is no difference between any VCs in the switch. There is no difference on the distance between the sources and the switch and data sending on any VC.

In this scenario the performance of the common buffer is the same as that N23 scheme but the complexity is less than N23 as mentioned before. Hence using common buffer scheme is more efficient than N23 scheme in this case.

4.2 One hop network model with on-off sources

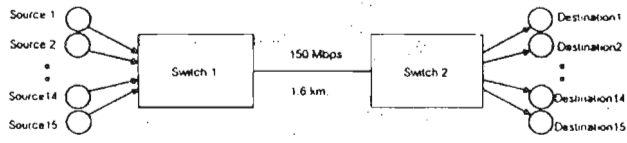


Figure 5. Network model 2

The results of network model 2 are shown in terms of received data of source 1, the total switch queue length and received cells as shown in Figure 6, 7 and Table 2, respectively.

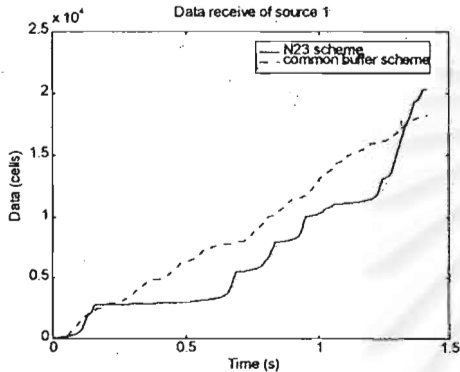


Figure 6. Data receive of source 1

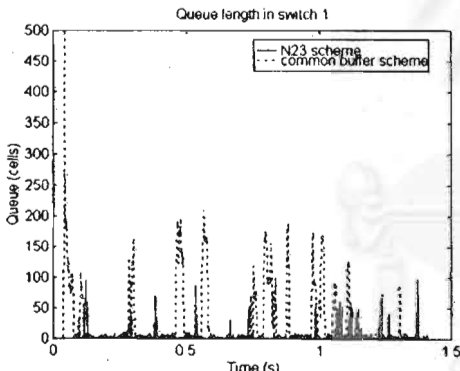


Figure 7. Queue length of switch 1

Table 2 receive cells at destination

VC	VC 1	VC 2	VC 3	VC 4	VC 5
N23	20355	18839	13558	19600	24852
Common Buffer	18226	18382	17605	18005	19705
VC	VC 6	VC 7	VC 8	VC 9	VC 10
N23	18938	22861	15914	15980	15809
Common Buffer	18326	18473	18731	20156	18279
VC	VC 11	VC 12	VC 13	VC 14	VC 15
N23	19904	21467	11713	23741	10388
Common Buffer	18149	19445	18194	17872	17885

For N23 scheme in Figure 6, the received data is not continually increased, there are some periods that the switch receives a few data or cannot receives any data from the sources so the slope of this graph is not constant. But in common buffer scheme amount of received data is continually increase and the slope is nearly constant. The difference is that for N23 scheme switch buffer is not equally shared so credit to any sources is independent and credit field of credit cell sometime is zero or close to zero in some sources. As a result, sources can not send data cells to the switch due to the suggested credit. Sources can send data cells again when receive next credit cells that have credit more than zero. For common buffer scheme, the switch buffer is equally shared to all VCs and in this scenario switch buffer is not full then credit value of credit cell that sends to all sources is not zero, so all sources can continuously send data cells to the switch.

From the switch queue length graph in Figure 7, the switch queue length of common buffer scheme is larger than N23 scheme on average. The larger value means that common buffer scheme can utilize the switch buffer more efficiently than N23 scheme because N23 scheme allocates the switch buffer of each VC and cannot use buffer of the others although they are available. For common buffer scheme, the switch buffer is common and it is not distributed so any source can achieve the available switch buffer with the sharing buffer role.

For the throughput issue, we found that in 1.4 seconds the aggregate throughput cell of the common buffer scheme is 277,433 cells while the aggregate throughput cell of N23 scheme is 273,919 cells. Hence, the throughput of common buffer scheme is more than N23 scheme about 1.28 percent in this scenario. In addition, the fairness value of N23 scheme is 0.951 but in common buffer scheme is 0.998. Hence, the proposed common buffer scheme is able to improve the fairness among all sessions as well.

4.3 One hop network model with some source bottleneck

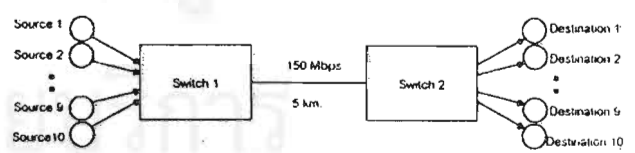


Figure 8. Network model 3

The results of simulation network model 3 are shown in term of total switch queue length and received cells as shown in Figure 9 and Table 3, respectively.

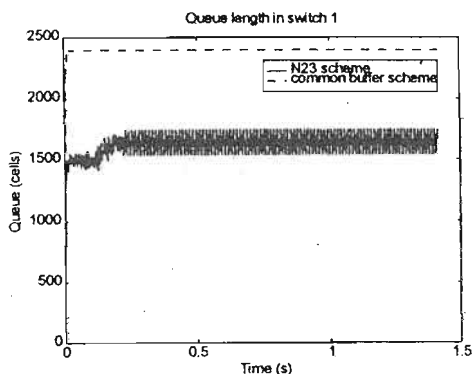


Figure 9. Queue length of switch 1

Table 3 receive cells at destination

VC	VC 1	VC 2	VC 3	VC 4	VC 5
N23	32997	32999	33001	33002	32995
Common Buffer	33003	33001	32999	32999	32998
VC	VC 6	VC 7	VC 8	VC 9	VC 10
N23	67001	67002	66999	66998	66995
Common Buffer	66999	67002	66998	67000	66998

From the switch queue length graph in Figure 9, the common buffer scheme successfully conducts the switch queue length to the work point, compared to the N23 scheme. Because the greedy sources (6-10) can utilize available buffer space that the bottleneck sources do not use. Hence, the queue length is larger than the N23 scheme. The larger value of switch queue length means that the common buffer scheme can utilize network resources more efficiently than the N23 scheme.

According to the received cells in Table 3, there is no significant difference between the common buffer scheme and the N23 scheme because in this model there is no difference between any VCs in the switch similar to network model 1 although there are bottleneck sources in VC 1-5. The bottleneck sources always send the data below the fairshare bandwidth then the result of the common buffer scheme is the same as N23 scheme in term of received cells.

In this scenario, the performance of the common buffer is the same as the N23 scheme but the proposed scheme can reduce the complexity and also able to conduct the switch queue length to the working point, compared to the conventional N23 scheme. Hence using the common buffer scheme gives more efficient buffer utilization than the N23 scheme.

5. Conclusion

In this paper, we propose common buffer scheme to reduce the complexity of credit based flow control. The complexity of the common buffer scheme is independent from the number of VCs and also smaller than complexity of the N23 scheme. In some network models that there is no difference between any VCs, the performance of common buffer scheme is the same as N23 scheme but the common

buffer scheme is still more efficient in term of complexity. In some network models such as on-off sources, common buffer scheme performs better than the conventional scheme in terms of throughput and fairness.

Acknowledgement

This research work is partly supported by Professor Prasit Prapinmongkolkarn's fund and TRF.

References

- [1] H.T.Kung, Trevor Blackwell and Alan Chapman, "ATM_Forum/94-0282," March 1994.
- [2] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy and R. Viswanathan, "ERICA Switch Algorithm: A Complete Description," ATM Forum/96-1172.
- [3] H.T.Kung, T.Blackwell and A.Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing," Proc. ACM SIGCOMM'94 Symposium on Communications Architectures, Protocol and Applications, August 31 - September 2, 1994, pp. 101-114.
- [4] H.T.Kung and A.Chapman "The FCVC (Flow Control Virtual Channels) Proposal for ATM Networks," Version 2.0, 1993 A summary appears in Proc 1993 International Conf. On Network Protocols, San Francisco, California, October 19-22, 1993 pp.116-127.
- [5] H.T.Kung and K.Chang, "Receiver-Oriented Adaptive Buffer Allocation in Credit-Based Flow Control for ATM Networks," Proc.INFOCOM'95, April 1995 pp. 239-252 vol. 1.
- [6] C. Özveren, R.Simcoe and G.Varghese, "Reliable and Efficient Hop-by-Hop Flow Control," IEEE Journal on selected areas in communications, vol. 13,no.4 May 1995.

A New Feedback Consolidation Algorithm for Point-to-Multipoint ABR Service

Supakiet Laohamatane, Lunchakorn Wuttisittikulkij and Prasit Prapinmonkolkarn

Department of Electrical Engineering, Chulalongkorn University

Phayathai Road, Bangkok, Thailand.

Tel. (662) 218-6512, Fax. (662) 251-8991, Email: lunch@ee.eng.chula.ac.th

Abstract

The main problem of point-to-multipoint ABR service in ATM network is how to consolidate feedback information from BRM cells to avoid consolidation noise while providing fast transient response. In previous studies, no known consolidation algorithm is able to achieve both requirements simultaneously. Some algorithms provide very fast transient response, but they suffer seriously from the problem of consolidation noise. Whereas the others can eliminate consolidation noise, but under certain traffic conditions they react rather slowly to traffic changes. In this paper, we present a complete solution to this problem. The key of this algorithm lies in the way in which feedback information is stored at each switch, *i.e.* information is kept on a per-branch basis for each VC. The results show that this new consolidation algorithm completely remove consolidation noise and provides very fast transient response in both overload and underload conditions. However, the proposed algorithm adds more complexity to the switches, due to extra variables being needed.

1. Introduction

The point-to-multipoint ABR (Available Bit Rate) service in ATM networks is currently considered important for supporting many emerging applications, such as data broadcasting over subscribed members. Considerable attention on this class of service is now eminent [1-4]. In point-to-point ABR service [5], a source sends forward resource management (FRM) cells along with data cells and it is turned around into backward resource management (BRM) cells by a destination. Along the way back to the source, each switch adjusts explicit rate (ER) in BRM cells to rate that the network is able to support.

The point-to-multipoint ABR service differs from the point-to-point ABR service in that there exists a number of feedback information from all the branch points sent back to the source through BRM cells. This causes a serious problem of how to handle such a large amount of feedback information. A common technique to this problem is that each branch point in the multicast tree consolidates the information in BRM cells received from downstream branch points or destinations and forwards the selected one to its upstream switch. This significantly reduces the amount of feedback information in the system. The main challenge of this problem is how to consolidate BRM cells in such a way that the transient response of each traffic source can be fast while avoiding undesired consolidation noise [4]. So far, a number of consolidation algorithms have been proposed [1-4]. However these algorithms still

have the consolidation noise problem or have slow transient response in underload condition leading to congestion and low link utilization.

In this paper, we describe a new consolidation algorithm that can eliminate consolidation noise, have fast transient response for both overload and underload conditions and can maintain the ratio of BRM cells to FRM cells close to 1. The details of this algorithm will be explained in section 3.

The organization of this paper is as follows. In section 2, we briefly describe the representative existing consolidation algorithms briefly. The new consolidation algorithm is described in section 3. Performance analysis and discussion is presented in section 4. Finally, section 5, we make conclusions.

2. Related Works

Robert [1] first proposed a consolidation algorithm which has an objective of achieving the fastest transient response. The basic idea of this algorithm is that each branch point consolidates the feedback information on a per-VC basis and passes it on to its upstream node through a BRM cell after the branch point has received a FRM (Forward Resource Management) cell. With this technique, consolidated feedback information may not reflect the actual congestion level of all the branches, because the feedback from some branches may have not yet arrived at the time when the BRM cell is sent. This could cause undesired consolidation noise. To reduce such consolidation noise, Tzeng [2] improved this algorithm by ensuring that the feedback to the upstream node is sent only when at least one BRM cell has been received from a downstream branch before receiving a FRM cell. This algorithm is able to reduce the level of consolidation noise. Ren [3] argued that generating a BRM cell causes switches to higher complexity and he also proposed two alternative algorithms. In his first algorithm, the first BRM cell received is passed to the source after a FRM cell has been received. Note that all the three algorithms described so far cause consolidation noise. For his second algorithm, Ren presented the algorithm that can eliminate the consolidation noise entirely. The algorithm is referred to as "wait-for-all" algorithm. In this algorithm, each branch point waits for BRM cells arrival from all the branches before a BRM cell is sent to the source. Although the problem of consolidation noise is completely removed, the waiting time for all feedback information may be long, causing very slow transient response.

Recently, an improved algorithm to these two techniques is found in [4]. The paper introduced a fast overload indication function to the "wait-for-all"

algorithm to enhance the transient response under severe network overload condition. Whenever the switch receives a BRM cell that has ER much lower than the previous ER, it generates an extra BRM immediately back to its upstream branch. Although this mechanism could improve the transient response, it occasionally generates more BRM cells than the number of FRM cells it has been received. In addition, this algorithm is unable to respond to the underload condition where the source node is entitled to increase its rate.

3. New Consolidation Algorithm

The main idea in the new consolidation algorithm is that each branch point maintains the feedback information on a per-branch basis for each VC, so that the feedback information from all the branch points can be kept up-to-date all time. This can prevent incorrect feedback information in BRM cell that can cause the consolidation noise. Because of correct feedback information at branch point, we can use feedback condition of Robert's algorithm that gives fastest transient response. Each branch point issues a BRM cell to its upstream switch only when it receives a FRM cell. Moreover, this ensures that the ratio of BRM cells to FRM cells is always 1. The pseudo code of this algorithm can be shown as follows:

Upon the receipt of a forward RM(ER,CI,NI) cell :

Multicast this RM cell to all participating branches;

Let $ER = \min(MER_i)$, $CI = \text{or}(MCI_i)$ and $NI = \text{or}(MNI_i)$ for all participating branches;

Let $ER = \min(ER, ER \text{ calculated by congestion control scheme for all branches})$

Send this RM cell back to the source;

Upon the receipt of a backward RM(ER,CI,NI) Cell from branch i:

Let $MER_i = ER$, $MCI_i = CI$, $MNI_i = NI$;

Discard the BRM Cell

4. Performance Analysis

A. Parameter Setting

In the simulation, the following parameter values are used

All links have a bandwidth of 150 Mbps.
 All sources have PCR = 150 Mbps, RIF = 1, RDF = 0.125, Nrm = 32, ICR for source 1 = 25 Mbps and ICR for source 2 = 150 Mbps
 All point-to-multipoint traffic flows from the root to the leaves of the tree. No traffic flows from the leaves to the root, except for RM cells. The same applies for the point-to-point connection.
 The sources are persistent, i.e. there is always data to send.
 All switches are assumed to employ ERICA [6] congestion control algorithm with the switch measurement interval set to minimum of the time to receive 100 cells and 1 ms. Target utilization of 90% is used in Network Model I and 100% in Network Model II.

B. Network Model I

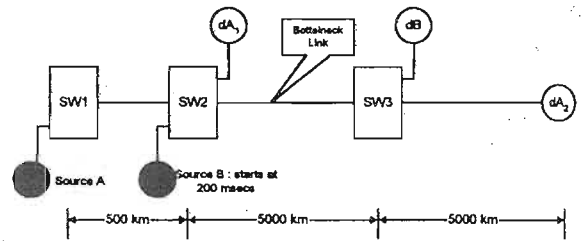


Figure 1: Network Model I

To illustrate the advantages of the new algorithm, its performance is evaluated and compared with the other 3 algorithms, Robert's algorithm, "wait-for-all" algorithm and Sonia's algorithm. A useful configuration for illustrating the consolidation noise and consolidation delay problem is shown in Figure 1. The network consists of a point-to-multipoint connection with two destinations connected to switch 2 and 3, and a point-to-point connection which starts sending data at 200 msec.

All the simulation results are summarized in Figures 4-7. It appears that Roberts' algorithm has a very fast transient response as expected. The source node can increase its rate almost instantly, as the network load condition allows. However, it exhibits serious consolidation noise, leading to rate oscillations, instability and unbounded queue, after source B becomes active. On the contrary, the "wait-for-all" algorithm can eliminate the consolidation noise, but it has very slow transient response, resulting in low link utilization during the transient period. The results from Sonia's algorithm are identical to that of "wait-for-all", because it can only detect overload condition but not the underload condition as in this configuration. Therefore, Sonia's algorithm suffers the same limitations as in the "wait-for-all" algorithm. The proposed algorithm yields superior performance to all. It can rapidly react to changes in network load conditions for both overload and underload, thus allowing maximum link utilization. The problem of consolidation noise is also completely removed. However, these advantages are achieved due to extra variables being needed in the switches.

C. Network Model II

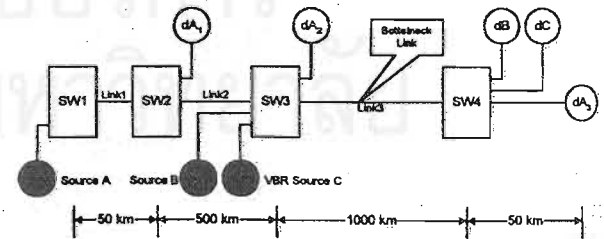


Figure 2: Network Model II

Network Model II is a VBR chain configuration as shown in Figure 2. VBR source C is an ON/OFF source. The ON period and OFF period are 100 ms. The VBR source sends data at 140 Mbps during ON period and 0 Mbps during OFF period. The bottleneck link is link3. As seen in Figure 8, Robert's algorithm has heavy consolidation noise especially in OFF period. This is because when source C stops sending data in OFF period,

source A should increase its rate leading source to generate more FRM cells and branch point would receive FRM cells more than BRM cells. For Robert's algorithm, the branch point feedbacks BRM cell when it received FRM cell, so that in this situation, the branch point must feedbacks BRM cells more often. Feedback information at branch point is cleared to ER value from FRM cells after returning BRM cell. This makes an incomplete consolidate feedback information from BRM cell and leads to consolidation noise.

For Wait-for-all algorithm, consolidation noise has been completely eliminated, however it still have very slow transient response because the source has to wait for the BRM cell to return from the farthest branch which is the destination dA_3 . Sonia's algorithm gives very fast transient response because network is in overload condition at the start phase. The overload detected technique would force branch point to feedback BRM cell to upstream branch. Like wait-for-all algorithm, Sonia's algorithm can eliminate consolidation noise because when network is not in overload condition the Sonia's algorithm acts like wait-for-all algorithm. The results for the proposed algorithm are almost the same as Sonia's algorithm with very fast transient response like Robert's algorithm and no consolidation noise founded because of good feedback information storing method.

D. Network Model III

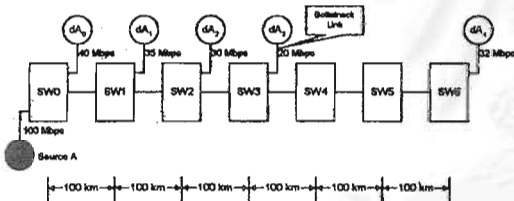


Figure 3: Network Model II

The network model III is shown in Figure 3 [7]. The bottleneck link is the one between switch SW3 and the destination dA_3 . As seen in Figure 9, the new consolidation algorithms can adjust the source rate immediately to 40 Mbps because the source A receives BRM cell from SW0. After that ACR at the source A is reduced to 35, 30 and 20 respectively according to the receiving of BRM from SW1, SW2 and SW3. Thus the network can get higher link utilization in transient state and it reaches the steady state within 3 ms. For Sonia's algorithm, the source rate is constant at ICR value (5

Mbps) until 6 ms leads to low link utilization in transient state. This is because when network is in an underload condition, Sonia's algorithm acts like wait-for-all algorithm. The source has to wait for the RM cell to return from the farthest branch which is the destination dA_4 .

5. Conclusions

In this paper, we proposed the new consolidation algorithm. The simulation results are compared with previous consolidation algorithm. It shows how consolidation noise happened and we proposed the solution of this problem by improving feedback information storing method. This can eliminate consolidation noise. In addition, this algorithm uses fastest BRM cell feedback method, results in very fast transient response in both overload and underload conditions. However, the proposed algorithm adds more complexity in increasing of variables in the switches.

Acknowledgement

This work is supported in part by TRF, Chulalongkorn University and Telecom Consortium.

References

- [1] L. Roberts, "Rate based algorithm for point to multipoint ABR service," *ATM Forum/94-0772R1*, Nov. 1994.
- [2] H-Y. Tzeng and K-Y. Siu, "On max-min fair congestion control for multicast ABR service in ATM," *IEEE JSAC*, vol 15, no. 3, pp. 545-555, Apr. 1997.
- [3] W. Ren, K-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," *Proceedings of IEEE ATM'96*, 1996.
- [4] S. Fahmy, et al., "Feedback consolidation algorithms for ABR point-to-multipoint connections," *Proceedings of IEEE INFOCOM'98*, 1998.
- [5] The ATM Forum, "The ATM forum traffic management specification version 4.0," Apr. 1996.
- [6] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "ERICA switch algorithm: A complete description." *ATM Forum 96-1172*. Aug. 1996.
- [7] T. Jiang, Ellen W. Zegura and M. Ammar, "Improved consolidation algorithm for point-to-multipoint ABR service." 0-7803-4874-5/98, *IEEE*. 1998.

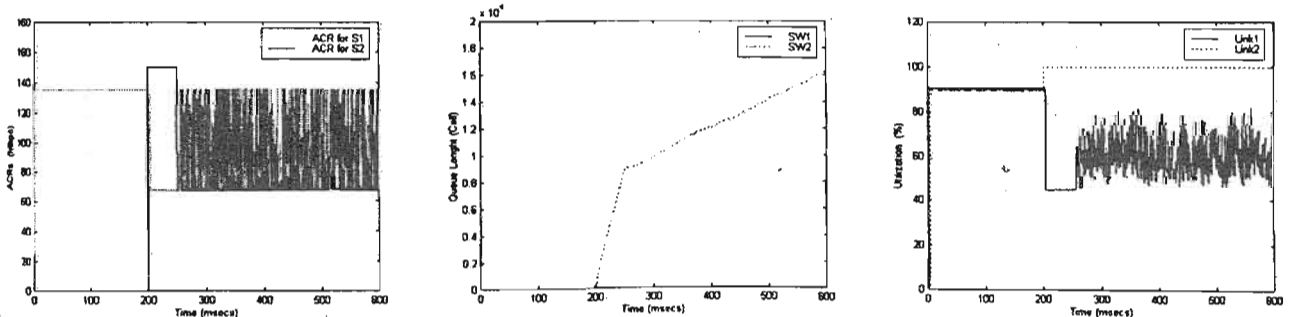


Figure 4: Simulation results for Network Model I from Roberts' algorithm

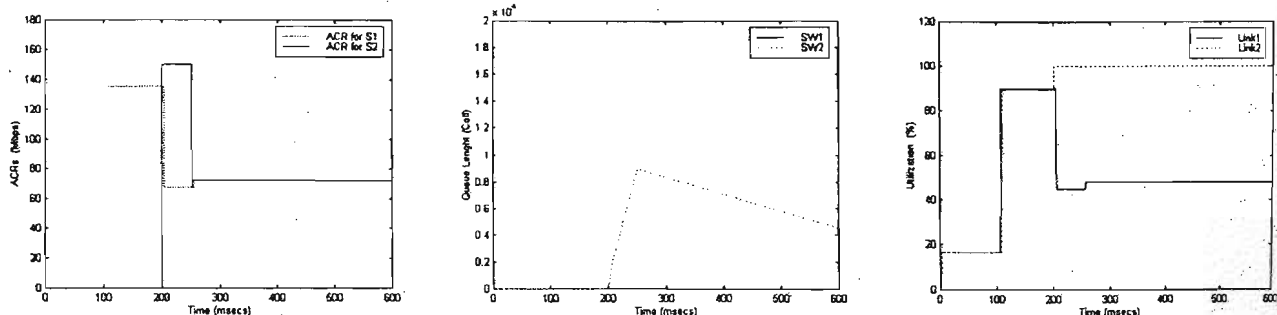


Figure 5: Simulation results for Network Model I from the "wait-for-all" algorithm

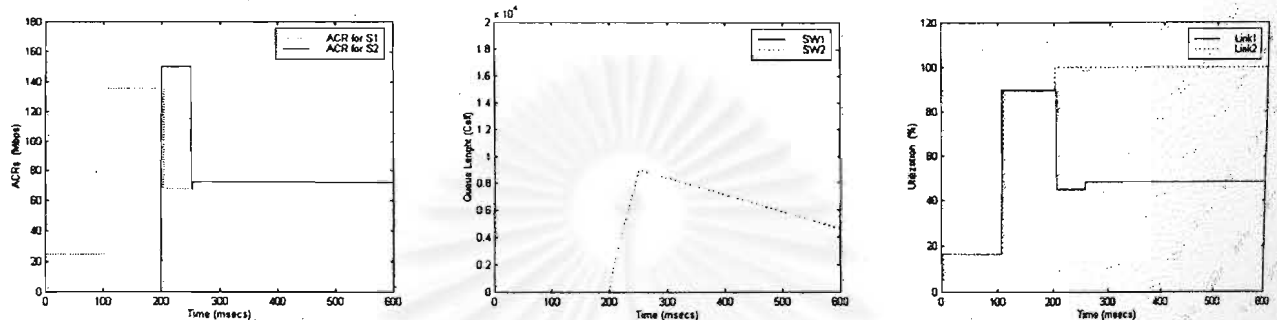


Figure 6: Simulation results for Network Model I from Sonia's algorithm

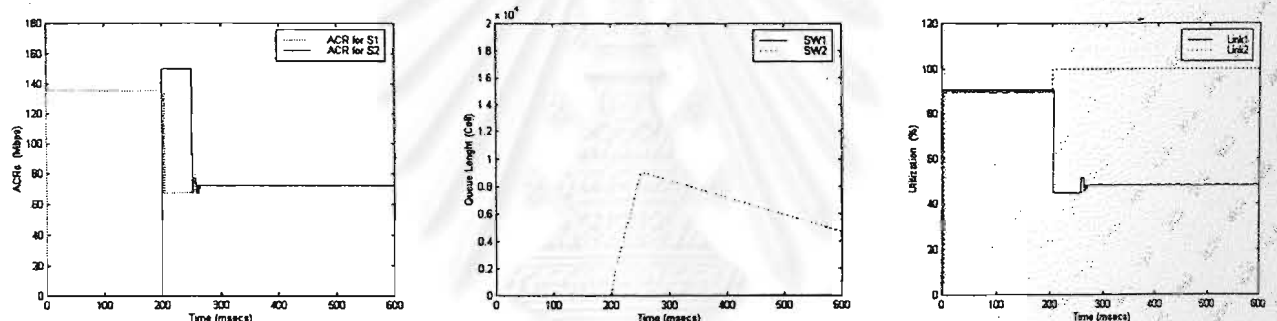


Figure 7: Simulation results for Network Model I from the proposed algorithm

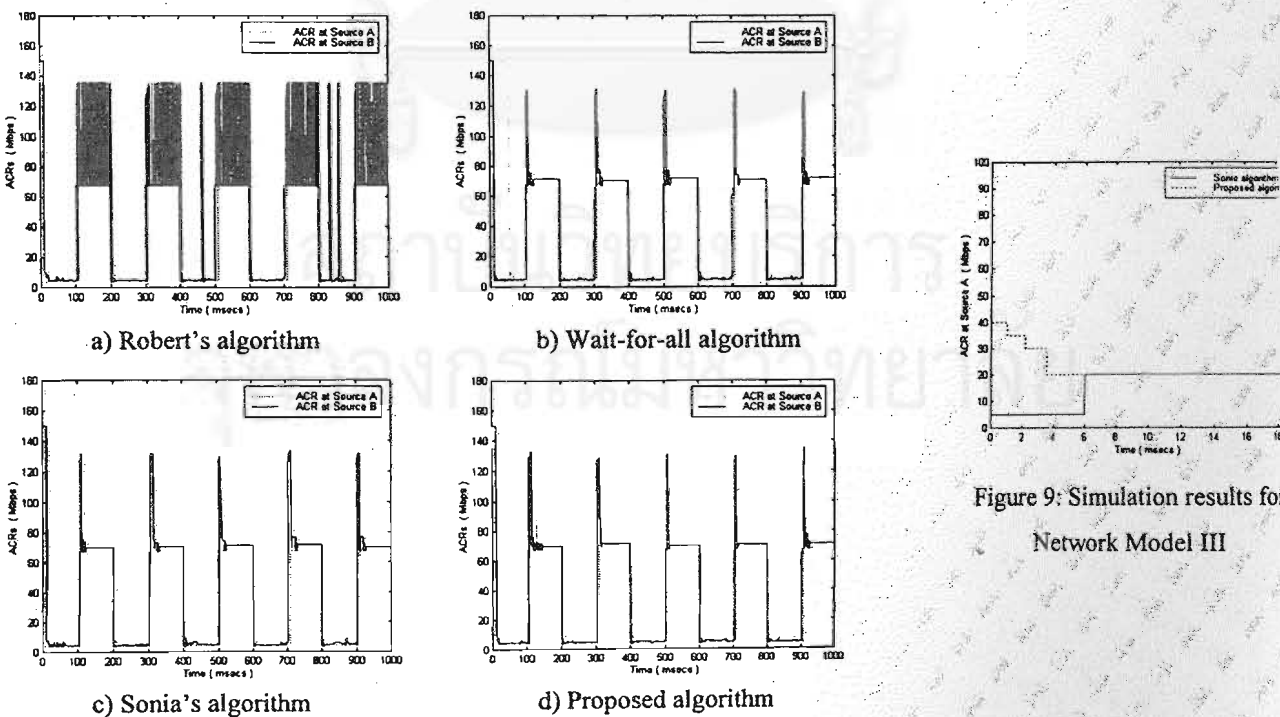


Figure 8: Simulation results for Network Model II

Figure 9: Simulation results for Network Model III