

เทคนิคการแปลงอัตราการใช้จากเซลล์เป็นเฟรม สำหรับกระบวนการควบคุมการไหลของที่ซีพี
บนบริการแบบเอปียาร์ ในโครงข่ายเอทีเอ็ม



นางสาววิณา จ่างเจริญ

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า


คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2544

ISBN 974-03-1268-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A CELL-TO-FRAME RATE CONVERSION TECHNIQUE FOR TCP FLOW CONTROL
OVER ABR BASED ATM NETWORKS



Ms.Weena Jangjaroen

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

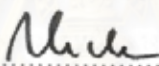
Chulalongkorn University

Academic Year 2001


ISBN 974-03-1268-3

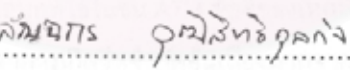
หัวข้อวิทยานิพนธ์ เทคนิคการแปลงอัตราการผลิตจากเซลล์เป็นเฟรม สำหรับกระบวนการ
ควบคุมการไหลของทีซีพี บนบริการแบบเอบีอาร์ ในโครงข่ายเอทีเอ็ม
โดย นางสาววีณา จ่างเจริญ
สาขาวิชา วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ลัญจกร วุฒิสัทติกุลกิจ
อาจารย์ที่ปรึกษาร่วม ศาสตราจารย์ ดร.ประสิทธิ์ ประพัฒน์มงคลการ


คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ


..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล)


..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.ลัญจกร วุฒิสัทติกุลกิจ)


..... อาจารย์ที่ปรึกษาร่วม
(ศาสตราจารย์ ดร.ประสิทธิ์ ประพัฒน์มงคลการ)


..... กรรมการ
(อาจารย์ สุวิทย์ นาคพิระยุทธ)

วีณา จ่างเจริญ : เทคนิคการแปลงอัตราการส่งจากเซลล์เป็นเฟรม สำหรับระบบการควบคุมการไหลของทีซีพี บนบริการแบบเอบีอาร์ ในโครงข่ายเอทีเอ็ม (A CELL-TO-FRAME RATE CONVERSION TECHNIQUE FOR TCP FLOW CONTROL OVER ABR BASED ATM NETWORKS) อ. ที่ปรึกษา: ผศ.ดร.ฉัตรกร วุฒิสัทติกุลกิจ, อ. ที่ปรึกษาร่วม: ศ.ดร.ประสิทธิ์ ประพัฒน์มงคล, 124 หน้า ISBN 974-03-1268-3

เทคโนโลยี ATM เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อรองรับการสื่อสารข้อมูลความเร็วสูงที่สามารถส่งข้อมูลหลาย ๆ ชนิดในเวลาเดียวกันได้ แต่เนื่องจากโครงข่ายที่ใช้งานในปัจจุบันไม่ได้ใช้เทคโนโลยี ATM เพียงอย่างเดียว จึงจำเป็นต้องอาศัยความสามารถในการทำงานแบบ internetworking ของ TCP มาช่วย โดยเฉพาะอย่างยิ่งในการส่งข้อมูลแบบดาตา ทั้งนี้การส่งข้อมูลแบบดาตา บนบริการแบบ ABR จะถูกควบคุมทั้งจากกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของ ABR แต่เนื่องจากกระบวนการทั้งสองทำงานแยกกันอย่างสิ้นเชิง จึงทำให้เกิดความไม่สอดคล้องกัน ซึ่งส่งผลให้ประสิทธิภาพการทำงานโดยรวมถูกจำกัด ไม่ว่าจะเกิดจากขนาดหน้าต่างของ TCP หรืออัตราการส่งข้อมูลของ ABR เทคนิคต่าง ๆ ถูกคิดค้นขึ้นเพื่อพยายามแก้ไขปัญหาดังกล่าว ซึ่งสามารถแยกแยะออกเป็น 2 กลุ่มใหญ่ ๆ คือเทคนิคการกักเก็บสัญญาณตอบกลับ และเทคนิคการปรับปรุงขนาดหน้าต่างความคับคั่ง เทคนิคการกักเก็บสัญญาณตอบกลับมีจุดเด่นที่สามารถลดขนาดหน่วยความจำที่อุปกรณ์ต้นทางได้ แต่ไม่ได้ช่วยเพิ่มประสิทธิภาพในการส่งข้อมูล ส่วนเทคนิคการปรับปรุงขนาดหน้าต่าง เป็นเทคนิคที่มุ่งเน้นการเพิ่มความสามารถในการส่งข้อมูล แต่เทคนิคนี้มีข้อเสียเมื่อรองรับทราฟฟิกมีการเปลี่ยนแปลงตลอดเวลา

สำหรับวิทยานิพนธ์ฉบับนี้ ได้เสนอเทคนิคใหม่ที่ใช้ในระบบการควบคุมการไหลของ TCP บนบริการแบบ ABR เรียกว่า เทคนิคการแปลงอัตราการส่งจากเซลล์เป็นเฟรม โดยเทคนิคนี้จะอาศัยอุปกรณ์ที่เรียกว่า Acknowledgement manager ในการควบคุมการส่งข้อมูลจากแหล่งกำเนิด TCP โดยอัตราการส่งที่ใช้ นั้น ได้มาจากการแปลงอัตราการส่งข้อมูลที่ใช้ในชั้น ATM ข้อดีของเทคนิคนี้ คือสามารถลดปริมาณหน่วยความจำที่ต้นทาง อีกทั้งยังสามารถแก้ปัญหาการเกิดขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง และอัตราการส่งได้ด้วย นั่นคือสามารถเพิ่มประสิทธิภาพในการส่งข้อมูล อย่างไรก็ตามกระบวนการทำงานของเทคนิคการแปลงอัตราการส่งแบบเซลล์เป็นเฟรมนี้ ค่อนข้างมีความซับซ้อน โดยเฉพาะอย่างยิ่งเมื่อเทียบกับกระบวนการทำงานแบบดั้งเดิม

ภาควิชา.....วิศวกรรมไฟฟ้า.....ลายมือชื่อนิสิต.....
สาขาวิชา.....วิศวกรรมไฟฟ้า.....ลายมืออาจารย์ที่ปรึกษา.....
ปีการศึกษา.....2544.....ลายมืออาจารย์ที่ปรึกษาร่วม.....

4170529421: MAJOR ELECTRICAL ENGINEERING

KEY WORD: TRANSMISSION CONTROL PROTOCOL / ASYNCHRONOUS TRANSFER MODE / AVAILABLE BIT RATE / FLOW CONTROL / CONGESTION CONTROL

WEENA JANGJAROEN: A CELL-TO-FRAME RATE CONVERSION TECHNIQUE FOR TCP FLOW CONTROL OVER ABR BASED ATM NETWORKS. THESIS ADVISOR : ASST. PROF. DR. LUNCHAKORN WUTTISITTIKULKIJ, THESIS COADVISOR : PROF. DR. PRASIT PRAPINMONGKOLKARN , 124 pp. ISBN 974-03-1268-3

Asynchronous Transfer Mode (ATM) has been known as a very high speed network that is designed for supporting a variety of applications with diverse requirements. The introduction of ATM technology has raised many questions regarding the effectiveness of using TCP over ATM networks. Strictly speaking, there is no need to have TCP if the entire network is an ATM network. In the case of data transfer over ABR service, there is the TCP window-based flow control running on the ABR rate-based congestion control. Those control mechanisms are working independently, thus making susceptible to serious performance limitations (from either window-limited or rate-limited). Therefore, various techniques to adapt TCP to ATM network environments are proposed. They are categorized into 2 groups: i.e. Acknowledgement-bucket (ACK-bucket) technique and ER-to-Window translation technique. The eminent advantage of ACK-bucket technique is minimizing the network-interface buffer requirement but it doesn't improve the quantity of throughput (the window-limited still exists). In the case of ER-to-Window technique, it can improve many aspects of performance. However, there are some drawbacks when variable rate traffic shares in the network.

The thesis proposes a new technique for TCP flow control over ABR service named Cell-to-Frame rate conversion technique. This technique requires a module called Acknowledgement manager, which controls the TCP transmission rate by mapping from the rate that used in ATM layer. Like ACK-bucket technique, the network-interface buffer requirement is minimized. In addition, this technique can suppress both window-limited and rate-limited effects. Conclusively, this technique can improve transmission performance efficiently, although it needs complexity process in the system.

Department.....Electrical Engineering..... Student's Signature.....

Field of Study.....Electrical Engineering..... Advisor's Signature.....

Academic Year.....2001..... Coadvisor's Signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลืออย่างดียิ่งของ ผู้ช่วยศาสตราจารย์ ดร. ลัญฉกร วุฒิสีทธิกุลกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และศาสตราจารย์ ดร.ประสิทธิ์ ประพัฒน์มงคล อาจารย์ที่ปรึกษาร่วม ซึ่งท่านทั้งสองได้ให้คำแนะนำ และข้อคิดเห็นต่าง ๆ ในการวิจัยด้วยดีตลอดมา ขอขอบคุณพี่ ๆ เพื่อน ๆ และน้อง ๆ นิสิตสาขาไฟฟ้าสื่อสาร โดยเฉพาะอย่างยิ่งนิสิตกลุ่มวิจัยระบบโทรคมนาคม ที่ให้คำแนะนำ และคำปรึกษาในการวิจัย อีกทั้งยังคงคอยให้กำลังใจ และความช่วยเหลือมาตลอด

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา และผู้มีพระคุณทุกท่าน ซึ่งให้การสนับสนุน และเป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

วีณา จ่างเจริญ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญภาพ.....	ณ
สารบัญตาราง.....	ด
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	5
1.3 ขอบเขตและเป้าหมายของวิทยานิพนธ์.....	6
1.4 ขั้นตอนและวิธีการดำเนินการ.....	6
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	6
2 การควบคุมความคับคั่งบนโครงข่าย ATM.....	7
2.1 ประเภทของการให้บริการในโครงข่าย ATM.....	7
2.1.1 Real time service.....	8
2.1.1.1 Constant Bit Rate (CBR).....	8
2.1.1.2 Real time Variable Bit Rate (rt-VBR).....	8
2.1.2 Non real time service.....	9
2.1.2.1 Non real time Variable Bit Rate (nrt-VBR).....	9
2.1.2.2 Unspecified Bit Rate (UBR).....	9
2.1.2.3 Available Bit Rate (ABR).....	10
2.2 บริการ ABR.....	11
2.3 การควบคุมความคับคั่ง.....	12
2.3.1 อัลกอริทึมสำหรับควบคุมความคับคั่งของบริการ ABR แบบ Binary.....	13
2.3.1.1 อี ล ก อ ริ ที ม EFICI (Explicit Forward Congestion Indication) Marking.....	13

สารบัญ (ต่อ)

	หน้า
2.3.1.2 อัลกอริทึม Relative Rate Marking.....	14
2.3.2 อัลกอริทึมสำหรับควบคุมความคับคั่งของบริการ ABR แบบ Explicit.....	15
2.3.2.1 อัล ก อ ริ ที ม ERICA (Explicit Rate Indication for Congestion Avoidance).....	15
2.3.2.2 อัล ก อ ริ ที ม ERICA+ (Explicit Rate Indication for Congestion Avoidance Plus).....	17
2.3.2.3 อัลกอริทึม FMMRA (Fast Max Min Rate Allocation)....	18
2.3.2.4 อัลกอริทึม E-FMMRA (Enhance Fast Max Min Rate Allocation).....	19
3 หลักการพื้นฐานของการควบคุมการไหลของ TCP.....	20
3.1 ลักษณะของ TCP segment.....	20
3.2 การควบคุมการส่งข้อมูล.....	22
3.2.1 การควบคุมการส่งข้อมูลของ TCP.....	22
3.2.2 หลักการที่ใช้ในการควบคุมการส่งข้อมูลของ TCP.....	25
3.2.2.1 หลักการทำงานแบบ Slow Start and Congestion Avoidance.....	26
4 การควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM.....	29
4.1 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการดั้งเดิม.....	30
4.2 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณ ตอบกลับ.....	32
4.3 การควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับรูปร่างหน้าต่างต่าง..	34
4.4 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบ เซลล์เป็นเฟรม.....	35
5 แบบจำลองการทำงาน.....	40
5.1 แบบจำลองการทำงาน (Configuration).....	40
5.1.1 Peer to peer configuration.....	41

สารบัญ (ต่อ)

	หน้า
5.1.2 VBR & ABR configuration.....	42
5.2 ข้อกำหนดที่ใช้ในการจำลองแบบ.....	43
5.2.1 แหล่งกำเนิดข้อมูล และคุณลักษณะของแหล่งกำเนิดข้อมูล (Source and Source characteristic).....	43
5.2.1.1 แหล่งกำเนิดข้อมูล TCP.....	43
5.2.1.2 แหล่งกำเนิดของบริการแบบ ABR.....	43
5.2.1.3 แหล่งกำเนิดของบริการแบบ VBR.....	44
5.2.2 ATM สวิตช์ (ATM Switch).....	44
5.2.3 ปลายทาง (Destination).....	45
5.2.4 สายส่ง (Lines).....	45
5.2.5 ข้อกำหนดอื่น ๆ.....	45
6 ผลการจำลองการทำงานและการวิเคราะห์.....	46
6.1 การจำลองการทำงานบนอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ.....	46
6.1.1 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยไม่ผ่าน TCP.....	47
6.1.1.1 การจำลองการทำงานของกระบวนการควบคุมความคับ คั่งของ ABR โดยใช้อัลกอริทึม ERICA.....	48
6.1.1.2 การจำลองการทำงานของกระบวนการควบคุมความคับ คั่งของ ABR โดยใช้อัลกอริทึม ERICA+.....	50
6.1.1.3 การจำลองการทำงานของกระบวนการควบคุมความคับ คั่งของ ABR โดยใช้อัลกอริทึม FMMRA.....	52
6.1.1.4 การจำลองการทำงานของกระบวนการควบคุมความคับ คั่งของ ABR โดยใช้อัลกอริทึม E-FMMRA.....	54
6.1.2 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้งาน ร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบ ต่าง ๆ.....	56

สารบัญ (ต่อ)

หน้า

6.1.2 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ.....	56
6.1.2.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR.....	58
6.1.2.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR.....	59
6.1.2.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	60
6.1.2.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	62
6.1.2.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และโดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR.....	63
6.1.3 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ.....	69

สารบัญ (ต่อ)

หน้า

6.1.3.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการ กักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA ใน กระบวนการควบคุมความคับคั่งของ ABR.....	69
6.1.3.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการ กักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA+ ใน กระบวนการควบคุมความคับคั่งของ ABR.....	70
6.1.3.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการ กักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม FMMRA ใน กระบวนการควบคุมความคับคั่งของ ABR.....	72
6.1.3.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการ กักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	73
6.1.3.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุม การไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดย ใช้หลักการกักเก็บสัญญาณตอบกลับ และโดยใช้อัลกอริ ทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR.....	74
6.1.4 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่าง และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ.....	74

สารบัญ (ต่อ)

	หน้า
6.1.4.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับ ปรุงค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA ใน กระบวนการควบคุมความคับคั่งของ ABR.....	75
6.1.4.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับ ปรุงค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA+ ใน กระบวนการควบคุมความคับคั่งของ ABR.....	76
6.1.4.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับ ปรุงค่าขนาดหน้าต่าง และใช้อัลกอริทึม FMMRA ใน กระบวนการควบคุมความคับคั่งของ ABR.....	77
6.1.4.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับ ปรุงค่าขนาดหน้าต่าง และใช้อัลกอริทึม E-FMMRA ใน กระบวนการควบคุมความคับคั่งของ ABR.....	78
6.1.4.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบ คุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดย ใช้การปรับปรุงค่าขนาดหน้าต่าง และโดยใช้อัลกอริทึม แบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR.	79
6.1.5 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบ เซลล์เป็นเฟรม และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุม ความคับคั่งของ ABR แบบต่าง ๆ.....	83
6.1.5.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลง อัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR.....	83

สารบัญ (ต่อ)

	หน้า
6.1.5.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลง อัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR....	84
6.1.5.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลง อัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR....	85
6.1.5.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหล ของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลง อัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม E- FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR....	86
6.1.5.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และโดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR.....	87
6.2 การจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	89
6.2.1 การจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกันโดยไม่มีการส่งข้อมูลผ่าน TCP.....	89
6.2.2 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	92
6.2.3 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ บนแบบจำลองที่มีความยาวสายส่งต่างแตกต่างกัน.....	94

สารบัญ (ต่อ)

	หน้า
6.2.4 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	96
6.2.4.1 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน และใช้ค่า RTT เฉลี่ยในการคำนวณ.....	97
6.2.4.2 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยใช้ค่า RTT มากสุดในการคำนวณ.....	99
6.2.4.3 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยใช้ค่า RTT ของแต่ละการเชื่อมต่อในการคำนวณ.....	101
6.2.5 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	103
7 บทสรุปและข้อเสนอแนะ.....	106
7.1 สรุปผลการจำลองการทำงาน.....	106
7.2 ข้อดีข้อเสียของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม.....	110
7.3 ข้อเสนอแนะ.....	111
รายการอ้างอิง.....	112

สารบัญ (ต่อ)

	หน้า
ภาคผนวก.....	114
ภาคผนวก ก การวิเคราะห์ขนาดหน้าต่างความคับคั่ง และปริมาณข้อมูลในหน่วย ความจำของอุปกรณ์ต้นทางในช่วง Slow Start.....	115
ภาคผนวก ข การวิเคราะห์ผลของความผิดพลาดที่เกิดกับข้อมูล ต่อกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ.....	120
ประวัติผู้เขียนวิทยานิพนธ์.....	124



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

	หน้า
รูปที่ 1.1	การทำงานที่ซึ่ซ้อนทับกันของกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของ ABR..... 4
รูปที่ 2.1	ลักษณะการแบ่งทรัพยากรระหว่างช่วงเวลาที่ไม่มีการเพิ่ม หรือลดจำนวนการเชื่อมต่อ..... 10
รูปที่ 2.2	การทำงานของกราฟฟิกแบบ ABR ภายใต้การควบคุมความคับคั่งแบบ Rate-Based..... 11
รูปที่ 2.3	ส่วนประกอบของเซลล์ RM..... 12
รูปที่ 2.4	ฟังก์ชันในการควบคุมระดับแถวคอยของอัลกอริทึม ERICA+..... 17
รูปที่ 3.1	ลักษณะของ TCP segment..... 20
รูปที่ 3.2	ลักษณะของ TCP เฮดเดอร์..... 21
รูปที่ 3.3	ตัวอย่างการควบคุมการส่งข้อมูลโดยใช้หลักการ Sliding Window..... 24
รูปที่ 3.4	ลักษณะการทำงานของหน่วยความจำต้นทางและปลายทาง..... 24
รูปที่ 3.5	ความสัมพันธ์ของขนาดหน้าต่างเทียบกับเวลา ของการควบคุมการส่งข้อมูลที่ใช้เทคนิค Slow Start and Congestion avoidance..... 27
รูปที่ 3.6	ตัวอย่างการส่งข้อมูลของ TCP โดยใช้หลักการ Slow Start and Congestion Avoidance..... 28
รูปที่ 4.1	เปรียบเทียบอัตราการส่งข้อมูลที่เกิดจากกระบวนการควบคุมการไหลของข้อมูลบน TCP กับกระบวนการควบคุมความคับคั่งของกราฟฟิกบนบริการแบบ ABR..... 30
รูปที่ 4.2	แสดงการทำงานของกระบวนการควบคุมการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ..... 32
รูปที่ 4.3	หลักการทำงานของกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม..... 36
รูปที่ 5.1	แบบจำลองชนิด Peer to peer..... 41
รูปที่ 5.2	แบบจำลองชนิด VBR & ABR..... 42
รูปที่ 6.1	ผลการจำลองการทำงานของแหล่งกำเนิด VBR ที่ใช้ในแบบจำลองชุดที่ 1 และ 2..... 48

สารบัญญภาพ (ต่อ)

	หน้า
รูปที่ 6.14 ความสัมพันธ์ระหว่างขนาดหน้าต่างความคับคั่งของ TCP เทียบกับเวลา....	65
รูปที่ 6.15 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR.....	69
รูปที่ 6.16 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR.....	71
รูปที่ 6.17 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	72
รูปที่ 6.18 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	73
รูปที่ 6.19 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR.....	75
รูปที่ 6.20 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR.....	76
รูปที่ 6.21 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	77
รูปที่ 6.22 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR.....	78
รูปที่ 6.23 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR.....	83

สารบัญญภาพ (ต่อ)

	หน้า
รูปที่ 6.24 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR....	84
รูปที่ 6.25 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR....	85
รูปที่ 6.26 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR..	86
รูปที่ 6.27 ผลการจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยไม่ผ่าน TCP.....	90
รูปที่ 6.28 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	92
รูปที่ 6.29 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCPชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	95
รูปที่ 6.30 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุงค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน และใช้ค่า RTT เฉลี่ยในการคำนวณ	98
รูปที่ 6.31 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุงค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน และใช้ค่า RTT มากสุดในการคำนวณ.....	99

สารบัญญภาพ (ต่อ)

	หน้า
รูปที่ 6.32 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุงค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน และใช้ค่า RTT ของแต่ละการเชื่อมต่อในการคำนวณ.....	101
รูปที่ 6.33 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรมบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน.....	103
รูปที่ ก1 แบบจำลองการทำงานของ TCP บนบริการแบบ ABR ในโครงข่าย ATM.....	115
รูปที่ ก2 ตัวอย่างการส่งข้อมูลในช่วง Slow Start.....	117
รูปที่ ก3 ความสัมพันธ์ระหว่างขนาดหน้าต่างความคับคั่งเทียบกับเวลา.....	118
รูปที่ ก4 ความสัมพันธ์ระหว่างปริมาณข้อมูลในหน่วยความจำของอุปกรณ์ต้นทางเทียบกับเวลา.....	119
รูปที่ ข1 กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้หลักการดั้งเดิม.....	120

สารบัญตาราง

	หน้า
ตารางที่ 2.1	สรุปหลักการปรับค่า ACR จากพารามิเตอร์ที่ได้รับจากเซลล์ RM..... 14
ตารางที่ 7.1	เปรียบเทียบสมรรถนะของกระบวนการควบคุมการไหลของ TCP บน บริการแบบ ABR ชนิดต่าง ๆ..... 109



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในช่วงเวลาไม่กี่ทศวรรษที่ผ่านมา เทคโนโลยีการสื่อสารข้อมูลผ่านโครงข่ายคอมพิวเตอร์ถูกพัฒนาขึ้นอย่างรวดเร็ว อีกทั้งยังได้รับการต้อนรับจากประชากรโลกอย่างอบอุ่น โดยเฉพาะอย่างยิ่งเทคโนโลยีอินเทอร์เน็ต ซึ่งกลายเป็นเทคโนโลยีที่มีบทบาทต่อวิถีชีวิตของมนุษย์ส่วนใหญ่ในปัจจุบัน ไม่ว่าจะเป็นแง่ของธุรกิจ การศึกษา หรือความบันเทิง จากความรุดหน้าของวิทยาการดังกล่าว มีผลทำให้ความสามารถในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กลายเป็นเพียงปัจจัยพื้นฐานของโครงข่ายทั่ว ๆ ไปเท่านั้น แต่สำหรับโครงข่ายที่ต้องการศักยภาพในการทำงานสูง จำเป็นต้องมีความสามารถทางด้านความเร็วในการส่งข้อมูล และความยืดหยุ่นในการรองรับทราฟฟิกแบบต่าง ๆ อย่างมีประสิทธิภาพด้วย

เทคโนโลยี ATM (Asynchronous Transfer Mode) เป็นเทคโนโลยีที่ถูกพัฒนาขึ้นเพื่อรองรับความต้องการดังกล่าว แต่ปัญหาสำคัญในการใช้งาน ATM กลับอยู่ที่ “จะนำโครงข่าย ATM ไปใช้งานร่วมกับโครงข่ายแบบเดิมที่มีอยู่แล้วได้อย่างไร ?” ทางออกง่าย ๆ ของปัญหานี้ คือการใช้ TCP (Transmission Control Protocol) เป็นตัวกลางในการเชื่อมต่อ ด้วยเหตุผลที่ว่า TCP เป็นโพรโทคอลในชั้นทรานสปอร์ตที่มีลักษณะเด่น ทั้งในแง่เหมาะสำหรับการทำงานในลักษณะ internetworking และยังเป็นโพรโทคอลที่เป็นที่ยอมรับและถูกใช้งานอย่างกว้างขวางในปัจจุบันอีกด้วย ดังนั้นการใช้งาน ATM ส่วนใหญ่จึงเป็นการใช้งานภายใต้ TCP นั่นคือใช้ TCP รองรับการทำงานในชั้นทรานสปอร์ต และ ATM รองรับการทำงานในชั้นดาตาลิงก์และชั้นเน็ตเวิร์ค ซึ่งการทำงานร่วมกันในลักษณะนี้เป็นการรวมข้อดีของทั้งสองเทคโนโลยีเอาไว้ด้วยกัน นั่นคือในแง่ของความเร็วในการติดต่อสื่อสาร และความสามารถในการรองรับ QoS (Quality of Services) ของทราฟฟิกแต่ละชนิด ซึ่งเป็นลักษณะเด่นของ ATM และความสามารถในการติดต่อสื่อสารระหว่างโครงข่าย รวมทั้งความเป็นมาตรฐานของ TCP ที่ยอมรับกันอย่างกว้างขวางในวงการสื่อสารข้อมูล อย่างไรก็ตามประเด็นที่เป็นที่กังขากลับกลายเป็นคำถามที่ว่า “TCP สามารถทำงานร่วมกับ ATM ได้ดีมากน้อยเพียงใด ?”

เนื่องจาก TCP เป็นโพรโทคอลที่ไม่ได้ถูกออกแบบมาเพื่อรองรับการทำงานร่วมกับโครงข่าย หรือโพรโทคอลชนิดใดชนิดหนึ่งโดยเฉพาะ นั่นคือ TCP มีความสามารถในการทำงานร่วมกับโพรโทคอลแบบต่าง ๆ ในชั้นต่ำกว่าได้ ซึ่งรวมไปถึง ATM ด้วย ดังนั้นจึงดูเหมือนว่าการทำงานของ

TCP บนโครงข่าย ATM ไม่น่าจะมีปัญหา แต่เมื่อศึกษาในรายละเอียดของการทำงาน ปรากฏว่า การทำงานของ TCP บนโครงข่าย ATM กลับมีความไม่สอดคล้องอะไรบางอย่าง ที่ส่งผลให้ประสิทธิภาพการทำงานโดยรวมถูกจำกัด ซึ่งจะกล่าวในรายละเอียดต่อไป

การทำงานของ TCP บนโครงข่าย ATM

เทคโนโลยี ATM จัดประเภทการให้บริการออกเป็น 5 ประเภท คือ Constant Bit Rate (CBR), real-time Variable Bit Rate (rt-VBR), non real-time Variable Bit Rate (non rt-VBR), Unspecific Bit Rate (UBR) และ Available Bit Rate (ABR) โดยบริการแบบ CBR และ VBR ถูกออกแบบมาสำหรับทราฟฟิกแบบออดิโอ และวิดีโอ ซึ่งมีข้อจำกัดในเรื่องของเวลา ในขณะที่บริการแบบ UBR และ ABR เหมาะในการใช้รองรับทราฟฟิกแบบดาตา ซึ่งมีคุณสมบัติแบบเบิรสต์ดี ดังนั้นการทำงานของ TCP บนโครงข่าย ATM อาจแบ่งได้เป็น 2 ประเภทใหญ่ ๆ คือ ใช้สำหรับรองรับทราฟฟิกแบบมัลติมีเดีย และสำหรับรองรับทราฟฟิกแบบดาตา

- **การทำงานของ TCP บนโครงข่าย ATM สำหรับรองรับทราฟฟิกแบบมัลติมีเดีย**

เมื่อพิจารณาความเป็นไปได้ในการนำ TCP มาใช้กับทราฟฟิกแบบมัลติมีเดีย (ในที่นี้หมายถึง ทราฟฟิกแบบออดิโอ และวิดีโอ) อาจกล่าวได้ว่า TCP ไม่ใช่ทรานส์พอร์ตโพรโทคอลที่เหมาะสมสำหรับทราฟฟิกแบบนี้ ทั้งนี้เพราะลักษณะการทำงานของ TCP มีความขัดแย้งกับคุณสมบัติของทราฟฟิกแบบมัลติมีเดียอย่างเห็นได้ชัด กล่าวคือในการทำงานของ TCP (โดยเฉพาะอย่างยิ่ง ในการทำงานของกระบวนการตรวจสอบความผิดพลาดในการส่งข้อมูล และกระบวนการส่งข้อมูลซ้ำ) จะคำนึงถึงความถูกต้องของข้อมูลที่ปลายทางได้รับ มากกว่าเวลาที่ใช้ในการทำงาน ในขณะที่คุณสมบัติของทราฟฟิกแบบมัลติมีเดีย เวลาที่ใช้ในการส่งข้อมูลมีความสำคัญมาก ในขณะที่ความถูกต้องของข้อมูลที่ปลายทางได้รับมีความสำคัญรองลงมา ดังนั้นทราฟฟิกแบบมัลติมีเดีย จึงเหมาะในการทำงานร่วมกับทรานส์พอร์ตโพรโทคอลชนิดอื่นมากกว่า

- **การทำงานของ TCP บนโครงข่าย ATM สำหรับรองรับทราฟฟิกแบบดาตา**

จะเห็นว่า TCP เหมาะสำหรับแอปพลิเคชันที่ต้องการความถูกต้องเป็นสำคัญ โดยไม่คำนึงถึงเวลาที่ใช้ในการสื่อสารมากนัก นั่นคือแอปพลิเคชันที่มีการส่งข้อมูลแบบดาตา ซึ่ง ATM จัดประเภทของบริการให้ทราฟฟิกแบบนี้ อยู่ในบริการแบบ UBR และ ABR

ซึ่งบริการทั้งสองแบบมีความแตกต่างกันในแง่คุณภาพของทราฟฟิกที่ต้องการ กล่าวคือ บริการแบบ UBR เป็นบริการที่ยอมรับความผิดพลาดในการส่งข้อมูลได้ระดับหนึ่ง ในขณะที่การส่งข้อมูลของบริการแบบ ABR ไม่ต้องการให้เกิดความผิดพลาดขึ้นเลย ดังนั้นลักษณะการทำงานในแง่กระบวนการควบคุมทราฟฟิกจึงแตกต่างกัน นั่นคือบริการแบบ UBR ไม่มีกระบวนการควบคุม

ปริมาณทราฟฟิก แต่มีกระบวนการตรวจจับข้อมูลทิ้ง (Dropping policy) ซึ่งทำงานที่สวิตช์ และ จะทิ้งข้อมูล เมื่อตรวจพบว่าโครงข่ายมีแนวโน้มที่จะเกิดความคับคั่งขึ้น ในขณะที่บริการแบบ ABR ควบคุมความคับคั่งของทราฟฟิก (ABR Congestion Control) โดยใช้กระบวนการย้อนกลับที่มีการตรวจสอบสภาพทราฟฟิกของโครงข่ายอย่างสม่ำเสมอ หลังจากได้รับผลการตรวจสอบจากโครงข่ายแล้วต้นทางจะนำข้อมูลนั้นมาทำการวิเคราะห์ เพื่อใช้ในการพิจารณา และปรับปริมาณข้อมูลที่จะใช้ในส่งครั้งต่อไป

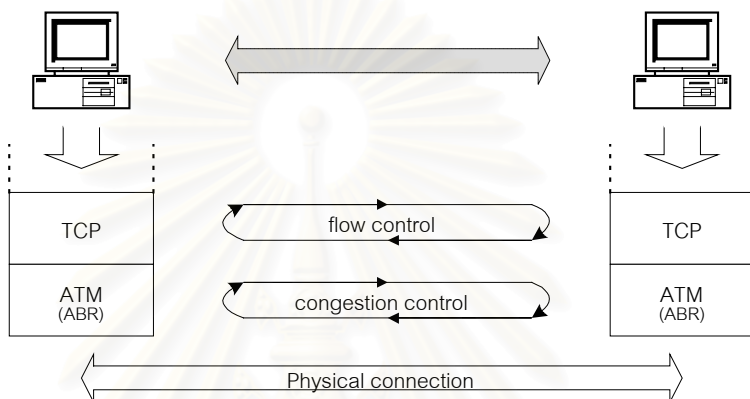
การทำงานของ TCP บนบริการแบบ UBR

เนื่องจากบริการแบบ UBR ไม่มีกระบวนการควบคุมปริมาณทราฟฟิก ดังนั้นหน้าที่ควบคุมการไหลของทราฟฟิก จึงตกเป็นของโพรโทคอลในชั้นทรานส์พอร์ต โดยกระบวนการตอบสนองความคับคั่งของบริการแบบ UBR จะอยู่ในรูปแบบของการทิ้งเซลล์ เมื่อปริมาณเซลล์ในหน่วยความจำที่สวิตช์ใกล้เต็ม ซึ่งจะเห็นได้ว่าการทำงานในลักษณะนี้ มีความเสี่ยงต่อประสิทธิภาพการทำงาน ไม่ว่าจะในแง่ของปริมาณ Throughput หรือความเท่าเทียมกัน (Fairness) ดังนั้นการพัฒนากระบวนการทำงานของ TCP บนบริการแบบ UBR จึงเป็นการพัฒนาความสามารถของกระบวนการตรวจจับข้อมูลทิ้ง ให้มีศักยภาพมากกว่านั้น สำหรับตัวอย่างกระบวนการตรวจจับข้อมูลทิ้งที่ถูกพัฒนาขึ้น ได้แก่ Early Packet Discard (EPD), Selective Drop Using per-VC Accounting และ Fair-Buffer Allocation เป็นต้น

การทำงานของ TCP บนบริการแบบ ABR

บริการแบบ ABR ในโครงข่าย ATM ถือได้ว่าเป็นบริการที่มีความโดดเด่น โดยเฉพาะอย่างยิ่งเมื่อมีการนำบริการแบบ ABR มาใช้ในเทคโนโลยีต่าง ๆ อย่างเช่น LAN Emulation รวมทั้งการที่โครงข่าย ATM เข้ามามีบทบาทกับโครงข่ายระดับโลกอย่างอินเทอร์เน็ต โดยมีบริการแบบ ABR เป็นตัวจักรสำคัญในการรองรับทราฟฟิกแบบดาตาในระบบ ทั้งนี้บริการแบบ ABR มีจุดเด่นที่กระบวนการควบคุมความคับคั่ง (ABR Congestion Control) ซึ่งเป็นกระบวนการที่พยายามใช้ทรัพยากรของโครงข่ายอย่างเต็มที่ และหลีกเลี่ยงการเกิดความคับคั่งในโครงข่าย อีกทั้งยังพยายามจัดสรรทรัพยากรให้การเชื่อมต่ออย่างยุติธรรม ทั้งนี้ก็วิจัยมากมายพยายามคิดค้น ปรับปรุง และพัฒนากระบวนการควบคุมความคับคั่งให้มีประสิทธิภาพมากยิ่งขึ้น ๆ ขึ้นไป ปัจจุบันจึงมีกระบวนการควบคุมความคับคั่งที่เป็นที่รู้จักมากมาย ยกตัวอย่างเช่น EFCI, Relative Rate Marking, ERICA, ERICA+ และ FMMRA เป็นต้น

สำหรับการทำงานของ TCP เป็นการควบคุมกระบวนการควบคุมการไหลของทราฟฟิก โดยการใช้การปรับขนาดหน้าต่าง (TCP Flow Control) และกระบวนการควบคุมความผิดพลาดของข้อมูลอย่างสอดคล้องกัน ทั้งนี้เมื่อนำ TCP มาทำงานบนบริการแบบ ABR ก็เปรียบเสมือนการนำกระบวนการควบคุมการไหลของ TCP มาซ้อนทับบนบริการควบคุมความคับคั่งของ ABR ซึ่งเมื่อพิจารณาการทำงานของทั้งสองกระบวนการ จะเห็นว่าแทบจะไม่มีความสัมพันธ์กันเลย (แสดงดังรูปที่ 1.1) ทั้ง ๆ ที่กระบวนการทั้งสองเป็นกระบวนการควบคุมปริมาณทราฟฟิกเหมือนกัน



รูปที่ 1.1 การทำงานที่ซ้อนทับกันของกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของ ABR

จากการที่ทั้งสองกระบวนการไม่มีความสัมพันธ์กันนี้เอง ส่งผลให้ประสิทธิภาพการทำงานโดยรวมของระบบถูกจำกัด การถูกจำกัดดังกล่าวแบ่งเป็น 2 ลักษณะ คือขีดจำกัดอันเนื่องมาจากหน้าต่างต่าง (Window-limited) และขีดจำกัดที่เกิดจากอัตราการส่งข้อมูล (Rate-limited) จากประเด็นดังกล่าว ส่งผลให้นักวิจัยหลาย ๆ กลุ่มพยายามปรับปรุงประสิทธิภาพการทำงานร่วมกันของกระบวนการทั้ง 2 ซึ่งสามารถแจกแจงออกเป็น 2 วิธีหลัก ๆ คือ

- การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ

หลักการการทำงานของเทคนิคนี้ คือการควบคุมปริมาณข้อมูลที่ส่งโดยการควบคุมอัตราการปล่อยสัญญาณตอบกลับสู่ TCP ต้นทาง ทั้งนี้ต้องอาศัยอุปกรณ์เสริมที่เรียกว่า ACK-Bucket ที่ทำหน้าที่กักเก็บสัญญาณตอบกลับที่ได้รับจากระบบ และจะส่งต่อไปยัง TCP ต้นทางตามจังหวะที่พิจารณาจากสภาพทราฟฟิกที่วัดโดยกระบวนการควบคุมความคับคั่งของ ABR จากหลักการเช่นนี้เองหน่วยความจำที่อุปกรณ์ต้นทางต้องการจึงมีขนาดลดลง และเป็นการลดขีดจำกัดที่เกิดจาก

อัตราการส่ง รวมทั้งยังมีผลให้สามารถลดความเสี่ยงในการเกิดความเสียหายของข้อมูลที่ต้นทางอีกด้วย

- **การควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับรูปร่างหน้าตา**

เป็นการคำนวณขนาดหน้าต่างค่าใหม่ที่เหมาะสมกับสภาพทราฟฟิกในโครงข่าย โดยการนำอัตราการส่งข้อมูลที่โครงข่ายแนะนำ (คือค่า ER ที่ได้จากระบบการควบคุมความคับคั่งของ ABR) คูณกับเวลาที่ใช้ในการส่งข้อมูลทั้งไปและกลับ (RTT) ในทางทฤษฎีเทคนิคนี้สามารถแก้ปัญหาที่เกิดจากทั้งขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง และขีดจำกัดที่เกิดจากอัตราการส่ง แต่เนื่องจากในการคำนวณค่าขนาดหน้าต่างตามหลักการนี้ จะใช้ค่า ER และค่า RTT ซึ่งถ้าผลการวัดค่าใดค่าหนึ่งผิดพลาด ก็จะทำให้ส่งผลกระทบต่อประสิทธิภาพการทำงาน อย่างไรก็ตามผู้เสนอแนวคิดนี้ไม่ได้เสนอวิธีแก้ปัญหาเมื่อเกิดความผิดพลาดในการส่งข้อมูล ซึ่งทำให้หลักการนี้ดูไม่สมบูรณ์นัก

เนื่องจากทั้งสองหลักการที่กล่าวไปข้างต้น มีจุดเด่นที่แตกต่างกัน ไม่ว่าจะในแง่ของหน่วยความจำของอุปกรณ์ต้นทางที่สามารถลดขนาดลงได้ หรือความสามารถในการแก้ปัญหาขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง ดังนั้นผู้วิจัยจึงพยายามคิดค้นระบบการควบคุมอัตราการส่งข้อมูลของ TCP บนบริการแบบ ABR ขึ้นมาใหม่ เพื่อการทำงานที่มีประสิทธิภาพเต็มที่ และเป็นการรวมจุดเด่นของเทคนิคอื่น ๆ ไว้ด้วย ระบบการนี้เรียกว่า “การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม” ซึ่งระบบการนี้ถูกออกแบบบนรากฐานที่พยายามปรับอัตราการส่งข้อมูลของ TCP ให้สอดคล้องกับอัตราการส่งข้อมูลจริงของโครงข่ายให้มากที่สุด โดยระบบการนี้ต้องอาศัยอุปกรณ์เสริมที่เรียกว่า Acknowledgement Manager ซึ่งเป็นอุปกรณ์ที่ใช้ควบคุมสัญญาณตอบกลับ เพื่อให้แหล่งกำเนิดข้อมูลค่อย ๆ ส่งข้อมูลด้วยปริมาณที่โครงข่ายสามารถรองรับได้จริง ดังนั้นระบบการทำงานแบบนี้จึงสามารถแก้ปัญหาที่เกิดจากขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง และขีดจำกัดที่เกิดจากอัตราการส่งได้

1.2 วัตถุประสงค์

1. เพื่อให้จากระบบการควบคุมการไหลของข้อมูลบน TCP และระบบการควบคุมความคับคั่งของทราฟฟิกบนบริการแบบ ABR สามารถทำงานร่วมกันได้ดีขึ้น เพื่อหาความเป็นไปได้ในการปรับปรุงประสิทธิภาพของโครงข่าย
2. เพื่อแก้ปัญหาที่เกิดจากการตั้งค่าขนาดหน้าต่างที่ใช้ในระบบการควบคุมการไหลของข้อมูลบน TCP ที่ไม่เหมาะสมกับสภาพความคับคั่งจริงของโครงข่าย

1.3 ขอบเขตและเป้าหมายของวิทยานิพนธ์

1. พยายามปรับปรุงประสิทธิภาพของกระบวนการควบคุมการไหลของ TCP ที่ใช้งานบนบริการแบบ ABR ในด้านของค่าวิสัยสามารถ ความเท่าเทียมกัน และความเร็วในการเข้าสู่จุดทำงาน โดยเปรียบเทียบกับกระบวนการควบคุมการไหลที่มีอยู่
2. แก้ปัญหาที่เกิดจากการตั้งค่าขนาดหน้าต่างที่ใช้ในกระบวนการควบคุมการไหลของข้อมูลบน TCP ที่ไม่เหมาะสมกับสภาพความคับคั่งของโครงข่ายจริง โดยใช้วิธีการแปลงค่าอัตราการส่งข้อมูลบนบริการแบบ ABR ในชั้น ATM มาเป็นอัตราการส่งข้อมูลในชั้นทรานส์พอร์ต

1.4 ขั้นตอนและวิธีการดำเนินการ

1. ศึกษามาตรฐานและทฤษฎีต่างๆของ ATM และส่วนที่เกี่ยวข้อง
2. ศึกษาวิธีการควบคุมความคับคั่งแบบต่าง ๆ ที่ใช้กับบริการแบบ ABR ในโครงข่าย ATM
3. ศึกษาวิธีการควบคุมการไหลของข้อมูล และการทำงานของ TCP
4. ศึกษาวิธีการควบคุมการไหลของข้อมูล TCP บนบริการแบบ ABR ที่มีอยู่
5. หาวิธีการปรับปรุงกระบวนการควบคุมการไหลของทราฟฟิก
6. เขียนโปรแกรมสร้างอัลกอริทึมของกระบวนการที่ปรับปรุง
7. เขียนโปรแกรมสร้างแบบจำลองขึ้นมา เพื่อนำมาทดสอบกับกระบวนการที่ปรับปรุงขึ้นมา
8. วิเคราะห์และเปรียบเทียบผลการทดสอบจากแบบจำลอง
9. ตรวจสอบและแก้ไขข้อผิดพลาด
10. สรุปผลงานที่ได้นำเสนอ
11. เขียนและพิมพ์วิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเปรียบเทียบ สรุป และวิเคราะห์ถึงความแตกต่าง ข้อดีและข้อด้อยของกระบวนการควบคุมการไหลแบบต่าง ๆ ได้
2. สามารถเพิ่มค่าวิสัยสามารถ และประสิทธิภาพของ TCP
3. สามารถลดความยุ่งยากในการทำงานของ TCP บนบริการ ABR ในโครงข่าย ATM
4. สามารถทำให้แหล่งกำเนิดแต่ละแหล่งกำเนิดส่งข้อมูลได้เท่าเทียมกันและมีประสิทธิภาพ โดยอาศัยข้อมูลจากกระบวนการควบคุมความคับคั่งของสวิตช์ที่มีประสิทธิภาพ

บทที่ 2

การควบคุมความคับคั่งบนโครงข่าย ATM

ATM (Asynchronous Transfer Rate) เป็นเทคโนโลยีการสื่อสารความเร็วสูงที่ถูกพัฒนาเพื่อรองรับทราฟฟิก (traffic) หลาย ๆ ประเภทในเวลาเดียวกัน และมีการรับรองคุณภาพการให้บริการ (QoS, Quality of Service) ของทราฟฟิกนั้นด้วย โดย ATM อาศัยการแบ่งข้อมูลให้มีขนาดคงที่เล็ก ๆ (เรียกว่าเซลล์) และหลักการพาสแพ็กเก็ตสวิตซิง (Fast Packet Switching) ซึ่งเป็นการลดความซับซ้อนในการทำงานของสวิตช์ ทั้งนี้เพื่อเพิ่มความเร็วให้แก่โครงข่ายนั่นเอง นอกจากนี้เรื่องความเร็วของโครงข่ายแล้ว ATM ยังมีจุดเด่นในแง่ความสามารถในการรองรับทราฟฟิก หรือบริการที่มีคุณลักษณะต่างกันพร้อม ๆ กันในโครงข่ายเดียวกันอีกด้วย โดย ATM อาศัยหลักการรวมทราฟฟิกเชิงสถิติ (Statistical multiplexing) บนการเชื่อมต่อเสมือน (VC, Virtual Connection) ที่มีการกำหนดเส้นทางไว้ตั้งแต่เริ่มสร้างการเชื่อมต่อ (connection orientated) รวมไปถึงการใช้กระบวนการตรวจสอบคุณลักษณะของทราฟฟิกไม่ว่าจะเป็นก่อนการสร้างการเชื่อมต่อ (CAC, Call Admission Control) หรือหลังการสร้างการเชื่อมต่อ (UPC, Usage Parameter Control) อีกทั้งยังกระบวนการควบคุมการไหลและหลีกเลี่ยงการล้นของทราฟฟิก (อาจจะอยู่ในรูปแบบของกระบวนการควบคุมความคับคั่ง (Congestion control) หรือการตรวจจับข้อมูลทิ้ง (Dropping policy) ทั้งนี้ขึ้นอยู่กับประเภทของบริการ) ซึ่งการจัดการทราฟฟิก (traffic management) [1] ไม่ว่าจะโดยกระบวนการใดก็ตาม ถูกกำหนดให้ตั้งอยู่บนมาตรฐานเดียวกันโดย ATM Forum

2.1 ประเภทของการให้บริการในโครงข่าย ATM

เนื่องจากโครงข่าย ATM ถูกออกแบบมาให้มีความสามารถในการรองรับทราฟฟิกหลากหลายประเภทในเวลาเดียวกัน ถึงแม้ว่าทราฟฟิกทุกประเภทจะต้องถูกทำให้เป็นเซลล์ ATM แล้วจึงส่งผ่านการเชื่อมต่อเสมือนหนึ่ง ๆ เหมือนกัน แต่วิธีการในการรองรับทราฟฟิกนั้นกลับต่างกัน ซึ่งขึ้นอยู่กับคุณลักษณะของทราฟฟิกและความต้องของแอปพลิเคชันที่ใช้ ยกตัวอย่างเช่นทราฟฟิกแบบ real time เช่นเสียงพูดหรือสัญญาณวิดีโอ ต้องการการหน่วงทางเวลาน้อยที่สุดเท่าที่จะเป็นไปได้ โดยพิจารณาความถูกต้องของข้อมูลเป็นเรื่องรองลงมา ในขณะที่ทราฟฟิกแบบดาตา เช่นการส่งไฟล์ผ่านโครงข่ายคอมพิวเตอร์ ความถูกต้องของข้อมูลกลับเป็นเรื่องสำคัญที่สุด โดยไม่มีใครที่จะสนใจเวลาที่ใช้ในการส่งมากนัก การรองรับทราฟฟิกที่มีคุณลักษณะแตกต่างกันดังกล่าว จึงเริ่มต้นที่การกำหนดจัดแบ่งประเภทของทราฟฟิกตามความต้องการที่ต่างกัน ซึ่งประเภทของทราฟ

พิกหรือรูปแบบการให้บริการนั้น ATM Forum เป็นผู้กำหนดโดยแจกแจงการให้บริการออกเป็นชนิด ๆ ดังนี้ [1][2][3]

2.1.1 Real time service

จุดเด่นของแอปพลิเคชันที่ต้องการบริการแบบนี้คืออยู่ที่ลักษณะการหน่วงเวลาที่แอปพลิเคชันต้องการ โดยปกติในแอปพลิเคชันแบบ real time ลักษณะการไหลของข้อมูลที่ทำให้ผู้ใช้รายใดรายหนึ่งจะมีลักษณะเหมือนกับลักษณะการไหลของข้อมูลที่ถูกส่งออกมาจากแหล่งกำเนิด เช่น ผู้ใช้ที่ต้องการข้อมูลแบบออดิโอ หรือวิดีโอ ที่มีความต่อเนื่อง ถ้าการให้บริการขาดความต่อเนื่อง หรือมีจำนวนเซลล์ที่เสียหายมาก จะทำให้คุณภาพของบริการแย่ และไม่เป็นที่พอใจ หรือถ้าเป็นแอปพลิเคชันที่มีลักษณะการโต้ตอบกันระหว่างบุคคลเวลาที่ถูกระงับเป็นสิ่งสำคัญมาก ทั้งนี้เพราะการเกิดการหน่วงเวลาเพียงแค่เสี้ยววินาที ผู้ใช้อาจรับรู้ได้ และก่อให้เกิดความรำคาญได้ ดังนั้นคุณภาพของการให้บริการแบบ real time จึงขึ้นอยู่กับเวลาประวิงภายในโครงข่าย โดย ATM Forum จัดบริการแบบ CBR และ rt-VBR มารองรับทราฟฟิกแบบ real time

2.1.1.1 Constant Bit Rate (CBR)

การให้บริการแบบ CBR เป็นลักษณะบริการที่สามารถนิยามได้ง่ายที่สุด นั่นคือทำให้บริการที่ผู้ใช้ต้องการอัตราการส่งข้อมูลที่คงที่ ตลอดช่วงเวลาในการติดต่อกับโครงข่าย และไม่ต้องมีการหน่วงทางเวลาเลย ส่วนมากการให้บริการแบบ CBR จะใช้สำหรับข้อมูลแบบออดิโอ และวิดีโอที่ไม่ถูกบีบอัด เช่นการส่งสัญญาณการประชุมทางไกล (video conferencing) สำหรับการควบคุมคุณภาพการให้บริการของโครงข่ายนั้น ทำได้โดยการที่โครงข่ายยอมรับการจองทรัพยากรของบริการแบบ CBR ที่กำหนดไว้ตั้งแต่ช่วงติดตั้งการเชื่อมต่อ (connection establish) โดยกำหนดจากค่าแบนด์วิดท์ (bandwidth) สูงสุด หรืออัตราการส่งข้อมูลสูงสุด (PCR, Peak Cell Rate) ของแหล่งกำเนิดนั่นเอง ซึ่งการรับรองคุณภาพของบริการแบบนี้ มุ่งเน้นไปในส่วนของเวลาที่ถูกระงับเป็นสำคัญ

2.1.1.2 Real time Variable Bit Rate (rt-VBR)

การให้บริการแบบ rt-VBR มีจุดประสงค์เพื่อรองรับแอปพลิเคชันที่มีความไวต่อเวลา นั่นคือต้องการความหน่วงเวลาน้อยมาก ความแตกต่างระหว่างแอปพลิเคชันที่เหมาะสมสำหรับบริการแบบ rt-VBR และ CBR คือ แอปพลิเคชันที่รองรับบริการแบบ rt-VBR จะมีอัตราการส่งข้อมูลที่ไม่แน่นอนขึ้นอยู่กับเวลา ซึ่งอาจมองแหล่งกำเนิดข้อมูลแบบ rt-VBR ว่ามีคุณสมบัติคล้าย ๆ กับแหล่งกำเนิดข้อมูลที่สร้างข้อมูลแบบเบิรสต์ดี สำหรับตัวอย่างของแอปพลิเคชันที่ใช้บริการแบบนี้ เช่น

การส่งข้อมูลแบบวีดีโอ ที่มีการบีบอัดข้อมูล ซึ่งจะทำให้อัตราการส่งข้อมูลที่ใช้นั้นมีค่าเปลี่ยนแปลงตามเวลา โดยทราฟฟิกแบบ rt-VBR จะมีการกำหนดค่า PCR ค่าอัตราการส่งข้อมูลเฉลี่ย (SCR, Sustainable Cell Rate) และค่า CTD (Cell Transfer Delay) ซึ่งใช้สำหรับช่วงที่ทราฟฟิกมีลักษณะเบิรสต์ดี

จะเห็นได้ว่าบริการแบบ rt-VBR มีความยืดหยุ่นมากกว่า บริการแบบ CBR โดยเฉพาะอย่างยิ่งเมื่อโครงข่ายสามารถรวมการเชื่อมต่อหลาย ๆ การเชื่อมต่อแบบสถิติ (statistical multiplexing) ซึ่งทำให้จำนวนการเชื่อมต่อที่สามารถรองรับได้มากขึ้นบนโครงข่ายที่มีความจุเท่าเดิม และยังคงสามารถรองรับบริการของแต่ละการเชื่อมต่อได้ตามต้องการ

2.1.2 Non real time service

จุดประสงค์ของบริการแบบ Non real time คือรองรับทราฟฟิกที่มีคุณสมบัติแบบเบิรสต์ดี และไม่มีข้อจำกัดเรื่องเวลาที่ใช้ (สามารถเกิดการหน่วงทางเวลาได้) นอกจากนี้ยังทำให้โครงข่ายมีความสามารถในการรองรับทราฟฟิกได้ยืดหยุ่นมากขึ้น และสามารถให้การรวมการเชื่อมต่อแบบสถิติได้ดีขึ้น จึงสามารถเพิ่มประสิทธิภาพของโครงข่ายด้วย

2.1.2.1 Non real time Variable Bit Rate (nrt-VBR)

เนื่องจากโครงข่ายสามารถรู้ลักษณะของการไหลของทราฟฟิกที่คาดว่าจะได้รับ ของแอปพลิเคชันแบบ Non real time บางชนิด ดังนั้นโครงข่ายจึงสามารถรองรับ QoS ในส่วนของเซลล์ที่สูญหาย และเวลาประวิงที่แท้จริงได้แอปพลิเคชันจำพวกนี้จึงเหมาะที่จะใช้การให้บริการแบบ nrt-VBR สำหรับบริการแบบนี้ระบบปลายทางไม่ว่าจะเป็นแหล่งกำเนิดข้อมูลหรือตัวรับข้อมูล จะต้องเจาะจงอัตราการส่งข้อมูลสูงสุด (PCR) อัตราการส่งข้อมูลเฉลี่ย (SCR) และ ค่าความหนาแน่นและการจับตัวกันเป็นช่วง ๆ ของเซลล์ ที่คาดว่าจะวัดได้ โดยโครงข่ายสามารถแบ่งทรัพยากร ซึ่งในที่นี้หมายถึง แบนด์วิดท์และหน่วยความจำที่ใช้ในสวิตช์ (buffer) สำหรับรองรับข้อมูลด้วยเวลาประวิงต่ำและเซลล์ที่สูญหายน้อยอย่างสัมพันธ์กัน

บริการแบบ nrt-VBR สามารถใช้สำหรับการถ่ายโอนข้อมูลที่มีความต้องการเวลาน้อย ๆ ยกตัวอย่างเช่น การสำรองที่นั่งของสายการบิน การจัดการข้อมูลเกี่ยวกับธนาคาร หรือการตรวจจับข้อมูลที่ได้จากการวัดค่าในขบวนการที่กำลังดำเนินอยู่

2.1.2.2 Unspecified Bit Rate (UBR)

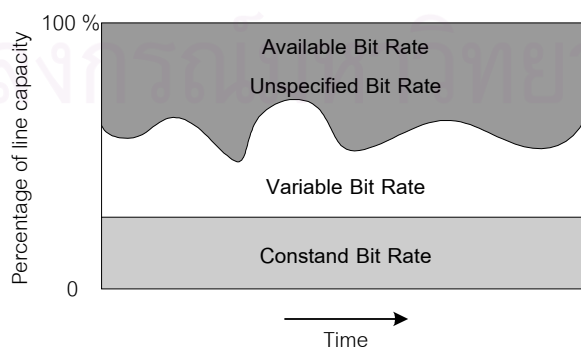
ทราฟฟิกส่วนใหญ่ในโครงข่าย ATM จะเป็นทราฟฟิกแบบ CBR และ VBR และเนื่องจากธรรมชาติของทราฟฟิกแบบ VBR จะมีลักษณะเบิรสต์ดี นั่นคือจะมีบางเวลาที่การใช้งานน้อยกว่า

ข้อตกลงที่ได้ทำไว้ จะเห็นได้ว่าจากการใช้งานของทราฟฟิกทั้งสองแบบดังกล่าว ระบบยังสามารถรองรับทราฟฟิกเพิ่มเติมได้อีก ถ้าในระบบยังมีความจุเหลือจากการใช้งาน ดังนั้นบริการแบบ UBR จึงมีไว้รองรับการใช้งานของความจุในส่วนที่เหลือนี้ การให้บริการแบบนี้จึงเหมาะสำหรับแอปพลิเคชันที่สามารถทนต่อการประวิงทางเวลา และความเสียหายที่อาจเกิดขึ้นกับเซลล์ข้อมูล ทั้งนี้ เซลล์จะถูกส่งโดยใช้เพียงหลักการพื้นฐานง่าย ๆ นั่นคือการส่งข้อมูลแบบ first-in-first-out (FIFO) เท่านั้น ดังนั้นแหล่งกำเนิดข้อมูลแบบ UBR นี้จึงไม่จำเป็นต้องมีการกำหนดค่าเริ่มต้น รวมทั้งไม่มีการใช้การควบคุมความคับคั่งซึ่งเป็นระบบแบบป้อนกลับอีกด้วย จะเห็นได้ว่าปัญหาที่พบในการใช้งานบริการแบบ UBR จึงเป็นเรื่องความสามารถในการรับประกันค่าอัตราการสูญหายของเซลล์

2.1.2.3 Available Bit Rate (ABR)

บริการแบบ ABR ถูกออกแบบสำหรับทราฟฟิกแบบ Non real time ที่ต้องการค่าอัตราการสูญหายของเซลล์อยู่ในระดับต่ำ ทั้งนี้เพื่อปรับปรุงการให้บริการแบบ UBR โดยบริการแบบ ABR จะพยายามจัดสรรทรัพยากรให้แอปพลิเคชันที่ใช้บริการแบบ ABR อย่างเท่าเทียมกัน สำหรับในกรณีที่โครงข่ายมีการใช้งานบริการอื่นอย่างเบาบาง (low load) ระบบจะพยายามแบ่งทรัพยากรให้การใช้งานบริการแบบ ABR มากขึ้น เพื่อเพิ่มประสิทธิภาพการใช้งานของโครงข่าย โดยใช้กลไกการควบคุมย้อนกลับ (feedback control mechanism) ในการกำหนดอัตราการส่งของแหล่งกำเนิด เพื่อให้มีความสอดคล้องกับเงื่อนไขการใช้งานของโครงข่ายขณะนั้น ซึ่งแหล่งกำเนิดจะทำการปรับอัตราการส่งตามข้อมูลที่ได้จากกลไกการควบคุมย้อนกลับ เพื่อป้องกันการเกิดความคับคั่งในระบบ ซึ่งเป็นสาเหตุทำให้ข้อมูลเสียหาย

โดยแอปพลิเคชันที่ใช้บริการแบบ ABR นี้ จะต้องมีระบุค่าอัตราการส่งสูงสุด (PCR) และค่าอัตราการส่งต่ำสุด (MCR, Minimum Cell Rate) ซึ่งค่า MCR นี้เองที่ใช้กำหนดแบนด์วิดท์ต่ำสุดที่ระบบรับประกันสำหรับการใช้งาน ถ้าค่า MCR ถูกกำหนดให้เท่ากับ 0 นั้นหมายความว่า จะไม่มีการรับประกันการใช้ทรัพยากรสำหรับแอปพลิเคชันนั้นเลย

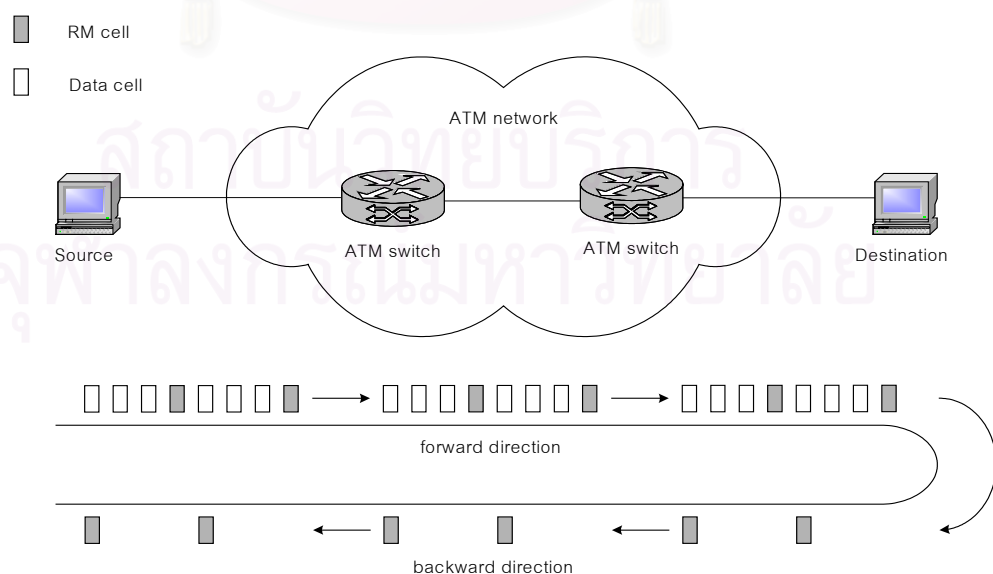


รูปที่ 2.1 ลักษณะการแบ่งทรัพยากรระหว่างช่วงเวลาที่ไม่มีการเพิ่ม หรือลดจำนวนการเชื่อมต่อ

2.2 บริการ ABR

เนื่องจากวิทยานิพนธ์ฉบับนี้เป็นการนำเสนองานวิจัยที่พยายามพัฒนาระบบการสื่อสารข้อมูลโดยมุ่งเน้นไปที่บริการแบบ ABR ในโครงข่าย ATM ดังนั้นจึงขออธิบายการทำงานของบริการแบบ ABR ในรายละเอียดพอสังเขปดังนี้

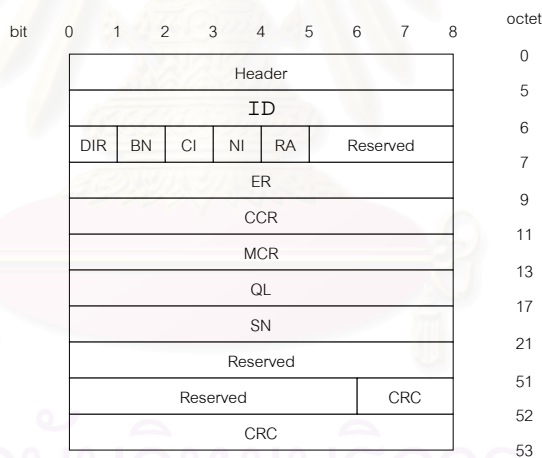
ATM Forum จัดตั้งมาตรฐานการให้บริการแบบ ABR เพื่อรองรับข้อมูลประเภทดาต้าที่ต้องการความถูกต้องสูง โดยมุ่งเน้นการใช้ทรัพยากรส่วนที่เหลือจากบริการแบบอื่นที่มีความสำคัญมากกว่า (นั่นคือ บริการแบบ CBR และ VBR) ถึงแม้ว่ายังไม่มีมาตรฐานกำหนดค่าเวลาประวิง และค่าอัตราการสูญหายของข้อมูลสำหรับบริการแบบ ABR แต่ก็มี ความพยายามลดค่าดังกล่าวให้น้อยที่สุดเท่าที่จะเป็นไปได้ ดังนั้นจึงจำเป็นต้องมีการตรวจสอบความคับคั่งของโครงข่ายเพื่อใช้ในการปรับอัตราการส่งข้อมูลที่เหมาะสม ซึ่งการควบคุมความคับคั่งสามารถแบ่งออกเป็น 2 ลักษณะ คือ Credit-Based และ Rate-Based การควบคุมความคับคั่งแบบ Credit-Based เป็นการควบคุมระหว่างอุปกรณ์ ATM สองตัวที่อยู่ติดกัน โดยด้านรับจะส่งข้อมูลไปยังด้านส่ง เพื่อบอกถึงความสามารถในการรองรับทราฟฟิกของตน (การควบคุมความคับคั่งแบบ Credit-Based เป็นการควบคุมแบบ hop-by-hop นั่นเอง) ส่วนการควบคุมความคับคั่งแบบ Rate-Based จะทำการควบคุมตั้งแต่ต้นทางจนถึงปลายทาง โดยการแทรกเซลล์พิเศษลงไปในเซลล์ข้อมูลเป็นระยะ ๆ (เท่ากับ ค่า $N_{rm}-1$ ปรกติค่า N_{rm} เท่ากับ 32) เพื่อเก็บข้อมูลความคับคั่งที่เกิดขึ้นที่สวิตช์ หรืออุปกรณ์ปลายทาง เมื่อถึงปลายทางเซลล์ดังกล่าวจะถูกส่งกลับไปที่ต้นทาง เพื่อใช้ในการปรับอัตราส่งที่เหมาะสมต่อไป เรียกเซลล์พิเศษนี้ว่า เซลล์ RM (Resource Management cell) การทำงานของการควบคุมความคับคั่งแบบ Rate-Based แสดงดังรูป 2.2



รูปที่ 2.2 การทำงานของทราฟฟิกแบบ ABR ภายใต้การควบคุมความคับคั่งแบบ Rate-Based

เนื่องจากในปัจจุบันการควบคุมความคับคั่งแบบ Rate-Based เป็นที่นิยมมากกว่าการควบคุมความคับคั่งแบบ Credit-Based มาก วิทยานิพนธ์ฉบับนี้จึงพิจารณาเฉพาะการควบคุมความคับคั่งแบบ Rate-Based เท่านั้น (โดยการอ้างถึง “การควบคุมความคับคั่ง” ในส่วนที่เหลือของวิทยานิพนธ์ฉบับนี้ จะหมายถึง การควบคุมความคับคั่งแบบ Rate-Based)

จากที่กล่าวไปข้างต้น ทราฟฟิกบนบริการแบบ ABR จะประกอบด้วยเซลล์ 2 ชนิด คือ เซลล์ข้อมูล และ เซลล์ RM ถึงแม้ว่าเซลล์ทั้งสองชนิดจะมีขนาด 53 ออกเทต (Octet) ซึ่งแบ่งเป็นส่วนของเฮดเดอร์ 5 ออกเทต และ payload อีก 48 ออกเทต เหมือนกัน แต่ความแตกต่างของเซลล์ทั้งสองชนิดนี้ (นอกจากข้อมูลในเฮดเดอร์ที่บ่งบอกชนิดของเซลล์) คือข้อมูลที่ถูกรวบรวมใน payload โดยเซลล์ข้อมูลจะบรรจุข้อมูลจากแหล่งกำเนิด ส่วนเซลล์ RM จะเป็นข้อมูลที่ใช้ในการควบคุมการส่งข้อมูล เช่น บิตแสดงถึงความคับคั่ง (CI, Congestion Indication) บิตแสดงการไม่อนุญาตให้เพิ่มอัตราการส่ง (NI, No Increase) พิวบอค่าต่าง ๆ ไม่ว่าจะเป็น ค่าอัตราการส่งสูงสุด (PCR) ค่าอัตราการส่งต่ำสุด (MCR) ค่าอัตราการส่งขณะนั้น (CCR, Current Cell Rate) หรือ ค่าอัตราการส่งที่โครงข่ายแนะนำให้ใช้ (ER, Explicit Rate) เป็นต้น ส่วนประกอบของเซลล์ RM แสดงดังรูปที่ 2.3



รูปที่ 2.3 ส่วนประกอบของเซลล์ RM

2.3 การควบคุมความคับคั่ง

กระบวนการควบคุมความคับคั่งของทราฟฟิกที่ได้รับบริการแบบ ABR เป็นการทำงานรวมกันของกลไกควบคุมย้อนกลับ และอัลกอริทึมสำหรับควบคุมความคับคั่งที่สวิตช์ โดยการออกแบบอัลกอริทึมดังกล่าว มีเป้าหมายเพื่อให้สวิตช์สามารถใช้ทรัพยากรได้อย่างมีประสิทธิภาพที่สุด ไม่ว่าในแง่ของ การตอบสนอง (Responsiveness) การทนทาน (Robustness) ความเท่าเทียมกัน

(Fairness) ของการใช้งานทรัพยากรของแต่ละแหล่งกำเนิดข้อมูล รวมไปถึงความซับซ้อนในการทำงานของอัลกอริทึม (Complexity) โดยอัลกอริทึมที่ใช้ในกระบวนการควบคุมความคับคั่งแบบ ABR นี้สามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ แบบ Binary และแบบ Explicit

2.3.1 อัลกอริทึมสำหรับควบคุมความคับคั่งของบริการ ABR แบบ Binary

อัลกอริทึมแบบนี้ อาศัยข้อมูลเพียงไม่กี่บิตในการควบคุมความคับคั่ง โดยการตรวจสอบสภาพความคับคั่งของโครงข่าย จะพิจารณาจากระดับแถวคอยในสวิตช์เปรียบเทียบกับระดับแถวคอยที่กำหนดไว้ (เรียกว่าค่า Threshold) อย่างไรก็ตามจากระบวนการควบคุมความคับคั่งแบบนี้ แหล่งกำเนิดสามารถรับรู้ได้ว่า ต้องเพิ่มหรือลดอัตราการส่งข้อมูลของตนเองเท่านั้น ดังนั้นการส่งข้อมูลจากแหล่งกำเนิดข้อมูลจึงมีลักษณะแกว่งขึ้นแกว่งลง จนกระทั่งสามารถเข้าสู่อัตราการส่งที่เหมาะสม แต่บางครั้งการแกว่งก็ไม่สามารถเข้าสู่ค่าใดค่าหนึ่งได้ การแกว่งลักษณะนี้จึงอาจก่อให้เกิดความเสียหายต่อข้อมูล หรือการขาดประสิทธิภาพในการทำงานของระบบได้ กล่าวคือ ถ้าแหล่งกำเนิดได้รับคำสั่งให้เพิ่มอัตราการส่งข้อมูล โดยมากจะเป็นการเพิ่มอัตราการส่งไปเรื่อย ๆ จนกระทั่ง ทำให้ระดับแถวคอยในสวิตช์เกินค่าที่กำหนดไว้ แล้วจึงลดลง ซึ่งในขณะที่ระบบส่งลดอัตราการส่งข้อมูล แหล่งกำเนิดก็ยังคงเพิ่มอัตราการส่งข้อมูลต่อไป ทั้งนี้เนื่องมาจากผลของการออกคำสั่ง ยังไม่ถึงแหล่งกำเนิด ทำให้แหล่งกำเนิดปฏิบัติตามคำสั่งเดิม (คือเพิ่มอัตราการส่งข้อมูลไปเรื่อย ๆ) การตอบสนองอย่างไม่ทันท่วงทีนี้เอง ทำให้ระดับแถวคอยในสวิตช์เพิ่มขึ้นเรื่อย ๆ จนกระทั่งอาจเกินความสามารถในการรองรับของสวิตช์ ซึ่งเป็นผลให้เกิดความเสียหายต่อข้อมูลได้ จากกรณีดังกล่าวจะเห็นได้ว่าความรุนแรงของปัญหาจะมากขึ้นตามจำนวนแหล่งกำเนิด และเวลาที่แหล่งกำเนิดใช้ในการตอบสนองคำสั่งจากระบบ สำหรับการปรับลดอัตราการส่งที่ทำให้เกิดผลเสียหายต่อระบบ ก็เกิดขึ้นด้วยเหตุผลทำนองเดียวกัน เพียงแต่ผลกระทบที่เกิดขึ้นเป็นเรื่องของการขาดประสิทธิภาพในการทำงานของระบบ ซึ่งเกิดจากภาวะการขาดข้อมูลในการส่ง อย่างไรก็ตามจะเห็นว่าอัลกอริทึมที่ควบคุมความคับคั่งแบบ Binary นี้ ใช้หลักการง่าย ๆ ในการควบคุมความคับคั่ง ซึ่งเป็นการลดความยุ่งยากให้แก่ระบบนั่นเอง

อัลกอริทึมที่รองรับการทำงานในลักษณะนี้ ได้แก่ EFCI Marking และ Relative Rate Marking โดยความแตกต่างของทั้งสองอัลกอริทึม อธิบายได้ดังนี้

2.3.1.1 อัลกอริทึม EFCI (Explicit Forward Congestion Indication) Marking

กระบวนการควบคุมความคับคั่ง ทำเฉพาะการส่งข้อมูลในทิศทางไปหน้าเท่านั้น เมื่อโครงข่ายเกิดความคับคั่ง (ตรวจสอบได้จากระดับแถวคอยในสวิตช์มีค่ามากกว่าระดับที่กำหนด) สวิตช์จะปรับค่าบิต EFCI (อยู่ในส่วนของเฮดเดอร์ในเซลล์ข้อมูล) ของเซลล์ที่ถูกส่งจากแหล่งกำเนิด เมื่อ

ถึงปลายทาง อุปกรณ์ปลายทางจะตรวจสอบค่าบิต EFCI ของเซลล์ตัวสุดท้ายของแต่ละชุด (ชุดในที่นี้หมายถึงชุดของเซลล์ข้อมูลที่อยู่ระหว่างเซลล์ RM) ถ้าค่าบิต EFCI ถูกปรับ (แสดงว่าโครงข่ายเกิดความคับคั่งไม่จุดใดก็จุดหนึ่ง) อุปกรณ์ปลายทางจะทำการปรับค่าบิต CI ของเซลล์ RM เซลล์ถัดไปที่ติดกัน แล้วส่งเซลล์ RM นั้นกลับไปยังแหล่งกำเนิด เพื่อลดอัตราการส่งข้อมูล ในทางตรงกันข้ามถ้าบิต EFCI ไม่ถูกปรับ บิต CI ในเซลล์ RM ก็จะไม่ถูกปรับเช่นกัน ซึ่งเป็นการทำให้แหล่งกำเนิดเพิ่มอัตราการส่งข้อมูลนั่นเอง

2.3.1.2 อัลกอริทึม Relative Rate Marking

อัลกอริทึมนี้แตกต่างจาก EFCI Marking ตรงที่สวิตช์จะทำการปรับค่าในเซลล์ RM เอง โดยไม่คำนึงถึงทิศทาง (นั่นคือจะปรับทั้งเซลล์ RM ที่มีทิศทางด้านหน้า และย้อนกลับ) โดยสวิตช์จะทำการปรับค่าบิต CI ในเซลล์ RM เมื่อเกิดความคับคั่ง และจะปรับค่าบิต NI เมื่อสวิตช์ไม่ต้องการให้แหล่งกำเนิดเพิ่มหรือลดอัตราการส่งที่ใช้อยู่

เมื่อแหล่งกำเนิดได้รับเซลล์ RM ย้อนกลับ แหล่งกำเนิดจะนำข้อมูลที่ติดมากับเซลล์ RM ย้อนกลับนี้มาปรับอัตราการส่งข้อมูล (ค่า ACR) โดยอาศัยหลักการดังต่อไปนี้

ในกรณีที่โครงข่ายแนะนำให้เพิ่มอัตราการส่งข้อมูล (พิจารณาจากบิต CI หรืออาจจะรวมบิต NI ด้วยในบางกรณี) การเพิ่มอัตราการส่งในแต่ละครั้งจะไม่เกินค่า $PCR \times RIF$ โดยเมื่อมีการปรับเพิ่มค่าอัตราการส่งเรียบร้อยแล้วค่าอัตราการส่งจะต้องไม่เกินค่า PCR โดยค่า RIF (Rate Increase Factor) เป็นค่าสัดส่วนการเพิ่มอัตราการส่ง

สำหรับการปรับลดอัตราการส่งข้อมูลของแหล่งกำเนิด ซึ่งพิจารณาจากค่าของบิต CI ในเซลล์ RM ในการปรับลดแต่ละครั้งจะเป็นสัดส่วนของอัตราการส่งปัจจุบัน นั่นคือแหล่งกำเนิดจะลดอัตราการส่งในปริมาณที่เท่ากับ $ACR \times RDF$ (RDF คือ Rate Decrease Factor เป็นสัดส่วนการลดอัตราการส่งข้อมูล) ผลจากการลดแต่ละครั้งนั้นต้องมีค่าไม่น้อยกว่า MCR ซึ่งเป็นค่าอัตราการส่งข้อมูลต่ำสุดที่โครงข่ายรับประกันไว้กับการเชื่อมต่อ

ตารางที่ 2.1 สูตรหลักการปรับค่า ACR จากพารามิเตอร์ที่ได้รับจากเซลล์ RM

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

2.3.2 อัลกอริทึมสำหรับควบคุมความคับคั่งของบริการ ABR แบบ Explicit

การควบคุมความคับคั่งโดยใช้อัลกอริทึมแบบนี้ อุปกรณ์กลางทางและปลายทางจะใช้หลักการปรับค่า ER รวมทั้งบิต CI และ NI ในเซลล์ RM เพื่อให้สอดคล้องกับสภาพทราฟฟิกของโครงข่าย หรือสวิตช์ ณ เวลานั้น ๆ โดยค่า ER ในเซลล์ RM เริ่มต้นจะถูกตั้งค่าเท่ากับ PCR ของแหล่งกำเนิดข้อมูลแต่ละตัว และเมื่อผ่านสวิตช์กลางทาง ค่า ER จะถูกปรับให้ลดลง หรือคงที่ ขึ้นอยู่กับสภาพทราฟฟิกและอัลกอริทึมที่ใช้ควบคุมความคับคั่ง เมื่อแหล่งกำเนิดได้รับเซลล์ RM ย้อนกลับแล้ว แหล่งกำเนิดจะพยายามปรับค่า ACR ให้สอดคล้องกับสภาพทราฟฟิกของโครงข่าย โดยใช้หลักการเดียวกับหลักการที่ได้กล่าวไปข้างต้น เพียงแต่ค่า ACR ที่ถูกปรับจะต้องไม่มากกว่าค่า ER (ซึ่งในกรณี Binary ค่า ER จะมีค่าเท่ากับ PCR เพราะอัลกอริทึมแบบ Binary ไม่ได้ปรับค่า ER เลย ทำให้ค่า ER ในเซลล์ RM ย้อนกลับยังคงมีค่าเท่ากับ PCR)

สำหรับความซับซ้อนในการคำนวณค่า ER ขึ้นอยู่กับอัลกอริทึมที่ใช้ ซึ่งในปัจจุบันอัลกอริทึมที่ทำงานแบบ Explicit นี้ถูกคิดค้นและพัฒนาออกมาเป็นจำนวนมาก สำหรับวิทยานิพนธ์ฉบับนี้ เลือกพิจารณาอัลกอริทึมที่เป็นที่รู้จัก และถูกกล่าวถึงมากที่สุด 4 อัลกอริทึมมาพิจารณา คือ ERICA, ERICA+, FMMRA และ E-FMMRA (ทั้งนี้เพื่อต้องการทดสอบความยืดหยุ่นต่ออัลกอริทึมที่ใช้ใน ATM ของวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้) รายละเอียดของอัลกอริทึมทั้ง 4 ขอกล่าวพอสังเขป ดังนี้

2.3.2.1 อัลกอริทึม ERICA (Explicit Rate Indication for Congestion Avoidance) [4]

เป็นอัลกอริทึมที่ใช้ในการควบคุมความคับคั่งของทราฟฟิกแบบ ABR ที่ถือได้ว่าเป็นที่รู้จักกันอย่างกว้างขวางที่สุดอัลกอริทึมหนึ่ง โดยอัลกอริทึมนี้ใช้หลักการทำงานง่าย ๆ คือ เริ่มจากกำหนดปริมาณแบนด์วิดท์ทั้งหมดที่ใช้รองรับทราฟฟิกแบบ ABR (ซึ่งกำหนดเป็นเปอร์เซ็นต์ของแบนด์วิดท์ที่เหลือจากทราฟฟิกที่มีความสำคัญมากกว่า) ดังสมการ

$$ABR\ Capacity = Target\ Utilization(U) \times Link\ Bandwidth \quad \dots (2.1)$$

โดย $ABR\ Capacity$ คือ แบนด์วิดท์ทั้งหมดสำหรับบริการแบบ ABR

$Target\ Utilization$ คือ ค่าประสิทธิภาพในการใช้งานลิงก์

$Link\ Bandwidth$ คือ แบนด์วิดท์ของลิงก์ที่เหลือจากการใช้งานของบริการ CBR และ VBR

เมื่อได้ปริมาณแบนด์วิดท์ทั้งหมดแล้ว อัลกอริทึม ERICA จะทำการแบ่งแบนด์วิดท์ให้การเชื่อมต่อทุกการเชื่อมต่ออย่างเท่าเทียมกัน โดยไม่คำนึงถึงความสามารถในการส่ง (ค่าที่ได้จากการคำนวณเรียกว่า *Fairshare*) จะเห็นว่าถ้ามีการเชื่อมต่อที่ไม่สามารถส่งข้อมูลในอัตราการส่งที่อัลก

อริทึมแนะนำได้ ทรัพยากรของโครงข่ายจะเหลือ โดยไม่มีการใช้งานใด ๆ ทั้งสิ้น อย่างไรก็ตามอัลกอริทึม ERICA แก้ปัญหาดังกล่าวโดยการเพิ่มอัตราการส่งให้แก่การเชื่อมต่อที่เหลือ โดยอาศัยค่าที่คำนวณได้จากการเปรียบเทียบปริมาณทราฟฟิเคาเข้าของสวิตช์กับปริมาณแบนด์วิดท์ทั้งหมดที่ถูกกำหนดให้มาใช้งาน เรียกค่านี้ว่าโหลดแฟกเตอร์ (Z , Load factor) นอกจากการแก้ปัญหาดังกล่าวโหลดแฟกเตอร์ยังทำหน้าที่ตรวจสอบสภาพความคับคั่งของโครงข่าย และทำการปรับอัตราการส่งเพื่อให้สอดคล้องกับปริมาณทราฟฟิเคา ณ เวลานั้น ๆ ด้วย (อัตราการส่งที่ถูกปรับโดยโหลดแฟกเตอร์ คือค่า VC Share)

$$Fairshare = \frac{ABRCapacity}{N} \quad \dots (2.2)$$

โดย N คือ จำนวนการเชื่อมต่อที่ติดต่อกับสวิตช์

$$Z = \frac{ABR Input Rate}{ABRCapacity} \quad \dots (2.3)$$

โดย $ABR Input Rate$ คือ ผลรวมของอัตราการส่งข้อมูลขาเข้าของการเชื่อมต่อแบบ ABR ทั้งหมดที่ติดต่อกับสวิตช์

$$VC Share = \frac{CCR}{Z} \quad \dots (2.4)$$

โดย CCR คือ อัตราการส่งข้อมูลปัจจุบัน (ซึ่งอาจเป็นค่าที่วัดใหม่หรือเป็นค่าที่อ่านจาก เซลล์ RM)

อัลกอริทึม ERICA จะคำนวณค่าที่ได้กล่าวไปนั้นทุก ๆ ช่วงเวลา (AI , Averaging Interval) ซึ่งช่วงเวลาดังกล่าวนี้ นอกจากจะทำตัวเหมือนกำหนดการในการคำนวณแล้ว ยังทำหน้าที่เป็นช่วงเวลาที่ใช้วัดค่าต่าง ๆ เช่นวัดจำนวนการเชื่อมต่อที่ยังคงติดต่อกับสวิตช์ เป็นต้น สำหรับการพิจารณาค่าอัตราการส่งข้อมูลที่อัลกอริทึม ERICA แนะนำนั้น ทำทุก ๆ ครั้งที่มี เซลล์ RM ย้อนกลับมาถึงสวิตช์ โดยพิจารณาจาก

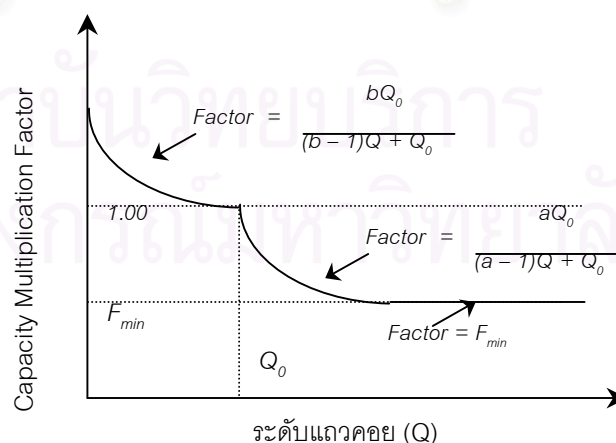
$$ER = \min\{ER, \max\{FairShare, VCShare\}\} \quad \dots (2.5)$$

โดย ER (ที่ใช้ในการเปรียบเทียบ) คือ ค่าของพิว ER ในเซลล์ RM ย้อนกลับที่รับมา

นอกจากนี้อัลกอริทึม ERICA ยังมีกระบวนการเล็ก ๆ น้อย ๆ ที่ใช้สำหรับการพิจารณาค่า ER เพื่อให้การแบ่งทรัพยากรเป็นไปอย่างยุติธรรมและมีประสิทธิภาพที่สุด สามารถศึกษารายละเอียดได้ใน [4]

2.3.2.2 อัลกอริทึม ERICA+ (Explicit Rate Indication for Congestion Avoidance Plus) [4][5]

เป็นอัลกอริทึมที่พัฒนามาจากอัลกอริทึม ERICA ในแง่ของการควบคุมระดับแฉกคอยในสวิตช์ ทั้งนี้เนื่องมาจาก ผู้พัฒนา ERICA+ เชื่อว่าการรักษาแฉกคอยให้มีค่าระดับหนึ่ง (ไม่น้อยหรือมากเกินไป) ทำให้โครงข่ายมีประสิทธิภาพการทำงานที่ดี ทั้งนี้เพื่อพยายามหลีกเลี่ยงภาวะการขาดข้อมูลในการส่ง อันเนื่องมาจากแหล่งกำเนิดใดแหล่งกำเนิดหนึ่งถูกปรับลดอัตราการส่งอย่างรวดเร็ว หรือไม่ก็เกิดการสิ้นสุดการติดต่อของการเชื่อมต่อใด ๆ ซึ่งมีผลให้สวิตช์ไม่สามารถคำนวณอัตราการส่งที่เหมาะสมค่าใหม่ ให้แก่การเชื่อมต่อที่เหลือได้ทันทั่วทั้ง (ทำให้แหล่งกำเนิดของการเชื่อมต่อที่เหลือปรับขึ้นไม่ทัน) สำหรับการควบคุมระดับแฉกคอยของ ERICA+ นั้น ทำโดยกระบวนการพิจารณาค่าแฟกเตอร์ ที่มีความสัมพันธ์โดยตรงกับระดับแฉกคอย ซึ่งค่าแฟกเตอร์ที่ได้นี้จะใช้ในการปรับค่าปริมาณแบนด์วิดท์ทั้งหมดที่ใช้อิงรับทราฟฟิกแบบ ABR ยกตัวอย่างเช่น เมื่อแฉกคอยสะสมมีปริมาณมากเกินไปที่กำหนดไว้ ค่าแฟกเตอร์ในกรณีนี้จะมีค่าน้อยกว่าหนึ่ง เพื่อเป็นการปรับค่าแบนด์วิดท์ที่เป็นเป้าหมายของการทำงานให้มีค่าน้อยลง ทำให้การเชื่อมต่อลดอัตราการส่งข้อมูล ซึ่งเป็นผลให้สวิตช์สามารถลดปริมาณแฉกคอยได้ สำหรับการทำงานของกระบวนการควบคุมระดับแฉกคอยในกรณีอื่น ๆ สามารถพิจารณาได้จากความสัมพันธ์ของระดับแฉกคอยและค่าแฟกเตอร์ที่ได้ ดังนี้



รูปที่ 2.4 ฟังก์ชันในการควบคุมระดับแฉกคอยของอัลกอริทึม ERICA+

2.3.2.3 อัลกอริทึม FMMRA (Fast Max Min Rate Allocation) [6][7]

อัลกอริทึม FMMRA เป็นอัลกอริทึมในสายพันธุ์เดียวกันกับอัลกอริทึม MIT และ UCSC [6] รวมทั้งอัลกอริทึม E-FMMRA (เป็นอัลกอริทึมที่ถูกพัฒนามาจากอัลกอริทึม FMMRA ซึ่งรายละเอียดอยู่ในส่วนถัดไป) โดยอัลกอริทึมดังกล่าวมีหลักการพื้นฐานเหมือนกัน คือใช้หลักการ *Max-Min fairness* ในการจัดสรรทรัพยากรของลิงก์ หลักการนี้จะพิจารณาความเท่าเทียมกันโดยคำนึงถึงความสามารถในการใช้ทรัพยากรของการเชื่อมต่อด้วย เมื่อจัดประเภทการเชื่อมต่อตามความสามารถในการใช้ทรัพยากรนั้น สามารถแบ่งออกเป็น 2 กลุ่มคือ

1. การเชื่อมต่อที่ติดขัด (*Bottlenecked Connection*) คือการเชื่อมต่อที่ไม่สามารถใช้แบนด์วิดท์ (ทรัพยากร) ทั้งหมดที่สวิตช์คำนวณให้ เนื่องจากถูกจำกัดด้วยอัตราการส่งสูงสุดของแหล่งกำเนิด หรืออาจถูกจำกัดโดยแบนด์วิดท์ที่แบ่งโดยอุปกรณ์ตัวอื่นในการเชื่อมต่อ

2. การเชื่อมต่อที่ไม่ติดขัด (*Non-bottlenecked Connection*) คือการเชื่อมต่อที่สามารถใช้แบนด์วิดท์ตามที่สวิตช์คำนวณได้อย่างเต็มที่

และมีสูตรในการจัดสรรทรัพยากรตามหลักการ *Max-Min fairness* ดังนี้

$$\text{Fairshare}(A_{max}) = \frac{C - C_b}{N - N_b} \quad \dots (2.6)$$

โดย	C	คือ แบนด์วิดท์ทั้งหมดของลิงก์
	C_b	คือ แบนด์วิดท์รวมของการเชื่อมต่อที่ติดขัด
	N	คือ จำนวนการเชื่อมต่อทั้งหมดที่ผ่านลิงก์
	N_b	คือ จำนวนการเชื่อมต่อที่ติดขัด

ในอัลกอริทึม FMMRA เรียกค่า *Max-Min fairness* ที่คำนวณได้ว่า *Advertised rate* ซึ่งมีรายละเอียดการคำนวณต่างไปจากสมการ (2.6) เล็กน้อย ทั้งนี้เป็นเพียงการลดความซับซ้อนในการคำนวณเท่านั้น นอกจากนี้อัลกอริทึม FMMRA ยังเพิ่มกระบวนการที่ใช้ในการแก้ปัญหาความไม่มีประสิทธิภาพในการใช้ทรัพยากรของโครงข่าย ซึ่งเกิดจากสาเหตุหลัก ๆ 2 ประการคือ การคำนวณ *Advertised rate* ผิดเนื่องมาจากการนับจำนวนการเชื่อมต่อผิดพลาด และเกิดจากธรรมชาติของแหล่งกำเนิดบางตัวที่ไม่ได้ส่งข้อมูลตลอดเวลา (ค่าอัตราการส่งข้อมูลที่บันทึกในเซลล์ RM และเป็นค่าที่นำมาใช้ในการคำนวณด้วยนั้น ไม่ได้เป็นค่าอัตราการส่งข้อมูลจริง แต่เป็นเพียงค่าอัตราการส่งข้อมูลสูงสุดที่แหล่งกำเนิดใช้เท่านั้น) ซึ่งอัลกอริทึม FMMRA แก้ปัญหาดังกล่าว โดยใช้หลักการเดียวกันกับที่ใช้ในอัลกอริทึม ERICA คือนับจำนวนการเชื่อมต่อที่ยังคงติดต่อกับสวิตช์ทุก ๆ ช่วงเวลา และใช้โหนดแพคเกจช่วยในกรณีที่ต้องการเพิ่มอัตราการส่งข้อมูลของแหล่งกำเนิด ดังนั้นค่า ER ที่ได้จากอัลกอริทึม FMMRA จึงพิจารณาจากสูตรต่อไปนี้

$$ER = \min\left\{\max\left(\frac{CCR}{\rho}, A_{max}\right), ER\right\} \quad \dots (2.7)$$

โดย ρ คือ ค่าโหลดแฟกเตอร์ ซึ่งคำนวณเหมือนกับค่า Z ในสมการที่ (2.3)
 ER (ที่ใช้ในการเปรียบเทียบ) คือ ค่าของพิว ER ในเซลล์ RM ย้อนกลับที่รับมา

นอกจากนี้อัลกอริทึม FMMRA ยังมีกระบวนการควบคุมระดับแฉวคอยของสวิตช์ โดยกำหนดระดับความคับคั่งของทราฟฟิกออกเป็น 3 ระดับ จากการเปรียบเทียบปริมาณแฉวคอยที่สะสมกับค่า QT และ DQT โดยอัลกอริทึม FMMRA จะพยายามควบคุมปริมาณแฉวคอยให้อยู่ระหว่างค่า QT และ DQT ซึ่งรายละเอียดการทำงานศึกษาได้จาก [7]

2.3.2.4 อัลกอริทึม E-FMMRA (Enhance Fast Max Min Rate Allocation) [8]

อัลกอริทึม E-FMMRA เป็นอัลกอริทึมที่พัฒนามาจากอัลกอริทึม FMMRA จึงมีกระบวนการทำงานพื้นฐานบางอย่างเหมือนกัน เช่นการคำนวณค่า *Advertised rate* เป็นต้น สำหรับจุดเด่นของอัลกอริทึม E-FMMRA อยู่ที่การนำค่า ER เฉลี่ย (เป็นการเฉลี่ยค่าตามหลัก *running weight average*) มาใช้แทนค่า CCR ที่บันทึกในเซลล์ RM ซึ่งเป็นการลดผลจากทราฟฟิกที่ไม่คงที่ โดยคำนวณตามสูตรต่อไปนี้

$$ER_{adjusted} = \frac{ER_{MAX}}{\rho} \quad \dots (2.8)$$

โดย ρ คือ ค่าโหลดแฟกเตอร์
 ER_{MAX} คือ ค่ามากที่สุดของพิว ER ที่ได้รับจากเซลล์ RM แบบย้อนกลับ โดยคิดคำนวณตามสมการ *running weight average* ดังสมการ (2.9)

$$ER_{MAX} = (1 - \text{Alpha}) \times ER_{MAX} + \text{Alpha} \times \max\{ER, ER_{adjusted}\} \quad \dots (2.9)$$

โดย Alpha คือ ค่า Averaging factor ปกติมีค่าเท่ากับ 1/8
 ER คือ ค่าของพิว ER ในเซลล์ RM ย้อนกลับที่สวิตช์รับมา

นอกจากนี้อัลกอริทึม E-FMMRA ก็มีกระบวนการควบคุมระดับแฉวคอยซึ่งเปรียบเทียบกับค่า QT และ DQT เหมือนกัน แต่แตกต่างกันที่รายละเอียดการคำนวณและการควบคุม ซึ่งสามารถศึกษารายละเอียดจาก [8]

บทที่ 3

หลักการพื้นฐานของการควบคุมการไหลของ TCP

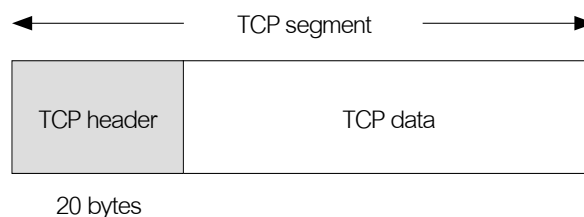
โพรโทคอลในชั้นทรานส์พอร์ตมีบทบาทสำคัญในการส่งข้อมูล ทั้งภายในเครือข่ายเองและระหว่างเครือข่าย ทั้งนี้เพราะโพรโทคอลในชั้นนี้ทำหน้าที่เป็นตัวประสานงานระหว่างแอปพลิเคชันและอุปกรณ์ในเครือข่าย นั่นคือทำให้แอปพลิเคชันสามารถใช้งานทรัพยากรของเครือข่ายได้อย่างมีประสิทธิภาพตามประเภทการใช้งาน

โพรโทคอล TCP (Transmission Control Protocol) เป็นโพรโทคอลในชั้นทรานส์พอร์ตที่เป็นที่รู้จักอย่างแพร่หลายโพรโทคอลหนึ่ง ทำหน้าที่ควบคุมการรับส่งข้อมูลระหว่างต้นทางและปลายทาง โดยการแบ่งข้อมูลออกเป็นส่วน ๆ (เรียกว่า TCP เซกเมนต์) แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล และมีการยืนยันความถูกต้องทุกครั้งที่มีการสื่อสารเพื่อรับประกันว่าข้อมูลที่รับส่งนั้นถูกต้องตรงกันทั้งสองฝ่าย ในกรณีที่มีส่วนใดส่วนหนึ่งเสียหาย ก็จะมีการส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนข้อมูลให้ต่อเนื่อง และประกอบกลับเป็นข้อมูลทั้งหมดอย่างถูกต้อง ทั้งนี้การติดต่อระหว่างต้นทางกับปลายทางจะใช้เส้นทางที่แน่นอนตลอดการเชื่อมต่อ (connection-oriented)

3.1 ลักษณะของ TCP segment

ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วน ๆ ให้มีขนาดเหมาะสมสำหรับการส่ง โดย TCP จะพิจารณาว่าขนาดเท่าใดจะทำให้การรับส่งนั้นมีประสิทธิภาพและน่าเชื่อถือสูงสุด ซึ่งจะเป็นการตกลงกันระหว่างอุปกรณ์ต้นทางและปลายทาง โดยข้อมูลที่ได้จากการแบ่งก่อนทำการส่งโดย TCP เรียกว่า TCP เซกเมนต์ (TCP segment)

รูปแบบของ TCP เซกเมนต์ จะประกอบด้วย 2 ส่วนด้วยกัน คือ ส่วนของข้อมูล (TCP data) และส่วนของเฮดเดอร์ (TCP header) แสดงดังรูป



รูปที่ 3.1 ลักษณะของ TCP segment

ส่วนของข้อมูล (TCP data) ได้มาจากการแบ่งข้อมูลของแอปพลิเคชันออกเป็นส่วน ๆ ให้มีขนาดเหมาะสมสำหรับการส่ง (จะกล่าวถึงการพิจารณาขนาดของ TCP เซกเมนต์ในส่วนต่อไป)

ส่วนของเฮดเดอร์ (TCP header) มีขนาดไม่แน่นอน เนื่องจาก TCP ไม่ได้ออกแบบมาเพื่อรองรับการทำงานร่วมกับโครงข่ายชนิดใดชนิดหนึ่งโดยเฉพาะ ทำให้ TCP จำเป็นต้องมีคุณสมบัติในการรองรับการทำงานที่มีการยืดหยุ่นสูง นอกจากนี้ TCP ยังเป็นโพรโทคอลที่ต้องการความน่าเชื่อถือในการส่งผ่านข้อมูลสูงอีกด้วย ดังนั้นจึงไม่ใช่เรื่องแปลกที่ TCP เฮดเดอร์มีขนาดค่อนข้างยาว นั่นคืออย่างน้อย 20 ออกเทต (พิจารณาได้จากรูปที่ 3.2) [2][9]

bit	0	4	10	16	31	octet				
	Source port		Destination port				0			
	Sequence number						4			
	Acknowledgement number						8			
	Data offset	Reserved	URG	ACK	PSH	RST	SYN	FIN	Window	12
	Checksum				Urgent pointer				16	
	Option + padding						20			

รูปที่ 3.2 ลักษณะของ TCP เฮดเดอร์

สำหรับการกำหนดขนาดของข้อมูลที่จะทำให้การส่งข้อมูลมีประสิทธิภาพนั้นมีปัจจัยหลายอย่างด้วยกัน ซึ่งปัจจัยสำคัญคือเรื่องของตัวกลางในการสื่อสาร ที่อยู่ในชั้นต่ำลงไปเกินกว่าที่แอปพลิเคชันสามารถควบคุมได้ ยกตัวอย่างเช่น การใช้เทคโนโลยีในระดับชั้นดาตาลิงก์ที่แตกต่างกันอย่าง Ethernet และ Token ring ก็จะมีขีดจำกัดในเรื่องขนาดของข้อมูลที่จะส่งได้ หรือ MTU (Maximum Transfer Unit) แตกต่างกัน ดังนั้นหากขนาดข้อมูลถูกพิจารณาจากการกำหนดโดยแอปพลิเคชันอย่างเดียว ก็อาจจะทำให้การสื่อสารใช้ไม่ได้ผลดีเฉพาะกับโครงข่ายแบบหนึ่ง แต่หากชั้นดาตาลิงก์เปลี่ยนไปก็就会产生ปัญหาขึ้นได้

TCP มีกลไกลดปัญหาเหล่านี้โดยจะมีการสอบถามขนาดของข้อมูลที่เหมาะสมที่เรียกว่า MSS (Maximum Segment Size) ก่อนการเริ่มส่งข้อมูล เพื่อให้ปลายทางตอบกลับมาได้ว่าสามารถทำการรับส่งข้อมูลในขนาด MSS ได้โดยไม่ต้องทำการแบ่งอีกหรือไม่ หากปลายทางเห็นว่าไม่สามารถทำได้ก็จะส่งค่า MSS กลับมาให้มีขนาดเล็กลง แล้วจึงเริ่มทำการรับส่งข้อมูลที่มีขนาดของเซกเมนต์สอดคล้องกับค่า MSS ที่ตกลงกันได้

3.2 การควบคุมการส่งข้อมูล

เทคนิคที่ใช้ในระบบการควบคุมการไหลและความผิดพลาดของการรับส่งข้อมูล โดยทั่วไปแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ เทคนิค Stop and Wait และเทคนิค Sliding Window (ซึ่งเทคนิค Go-Back-N และ Selective repeat ที่รู้จักกันดี จัดเป็นกรณีพิเศษของเทคนิค Sliding Window)

เทคนิค Stop and Wait จะทำการส่งข้อมูลที่ละแพ็กเก็ตโดยต้นทางจะไม่ส่งข้อมูลใหม่จนกว่าจะได้รับสัญญาณตอบกลับ (ACK, Acknowledgment) ที่ระบุความถูกต้องจากปลายทาง และต้นทางจะส่งข้อมูลชุดเดิมอีกครั้ง ในกรณีที่ต้นทางตีความว่าการส่งข้อมูลครั้งก่อนล้มเหลว ซึ่งอาจเกิดจากปลายทางไม่ได้รับข้อมูล หรือข้อมูลที่ปลายทางได้รับไม่ถูกต้อง ทั้งนี้จะเห็นว่ากลไกการทำงานของเทคนิค Stop and Wait จะเน้นระบบการควบคุมความผิดพลาด ทำให้ประสิทธิภาพของการสื่อสารในแง่ปริมาณข้อมูลที่ส่งได้ต่อหน่วยเวลาจึงต่ำ เทคนิค Stop and Wait จึงเหมาะที่จะใช้กับตัวกลางที่มีอัตราความผิดพลาดสูง

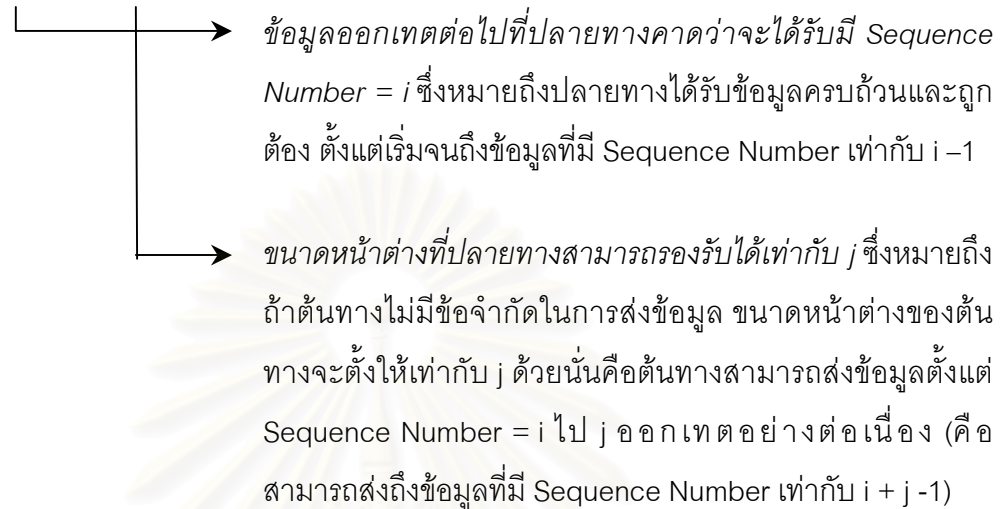
ปัจจุบันตัวกลางที่ใช้ในการสื่อสารถูกพัฒนา ให้มีความโดดเด่นทั้งในด้านความเร็วที่ใช้ในการสื่อสาร และอัตราความผิดพลาดเฉลี่ยในการส่งข่าวสารผ่านตัวกลางดังกล่าว ทำให้เทคนิค Stop and Wait ดูจะไม่เหมาะสมกับตัวกลางประเภทนี้ โดยเฉพาะอย่างยิ่งเมื่อใช้กับโครงข่ายที่มีขนาดใหญ่ (โครงข่ายขนาดใหญ่ในที่นี้หมายถึงโครงข่ายที่ครอบคลุมอาณาบริเวณกว้าง) ทำให้เวลาที่ใช้ในการส่งข้อมูลระหว่างต้นทางและปลายทางมีค่ามาก และทำให้การควบคุมการส่งข้อมูลโดยใช้เทคนิค Stop and Wait ไม่สามารถให้ประสิทธิผลสูงสุดได้ เทคนิค Sliding Window กลายเป็นทางออกของข้อบกพร่องดังกล่าว ทั้งนี้เพราะเทคนิค Sliding Window เน้นที่การควบคุมการไหลของข้อมูลปริมาณมาก โดยอาศัยหลักการกำหนดปริมาณข้อมูลที่ต้นทางสามารถส่งได้อย่างต่อเนื่อง โดยไม่จำเป็นต้องรอสัญญาณตอบกลับจากปลายทาง

3.2.1 การควบคุมการส่งข้อมูลของ TCP

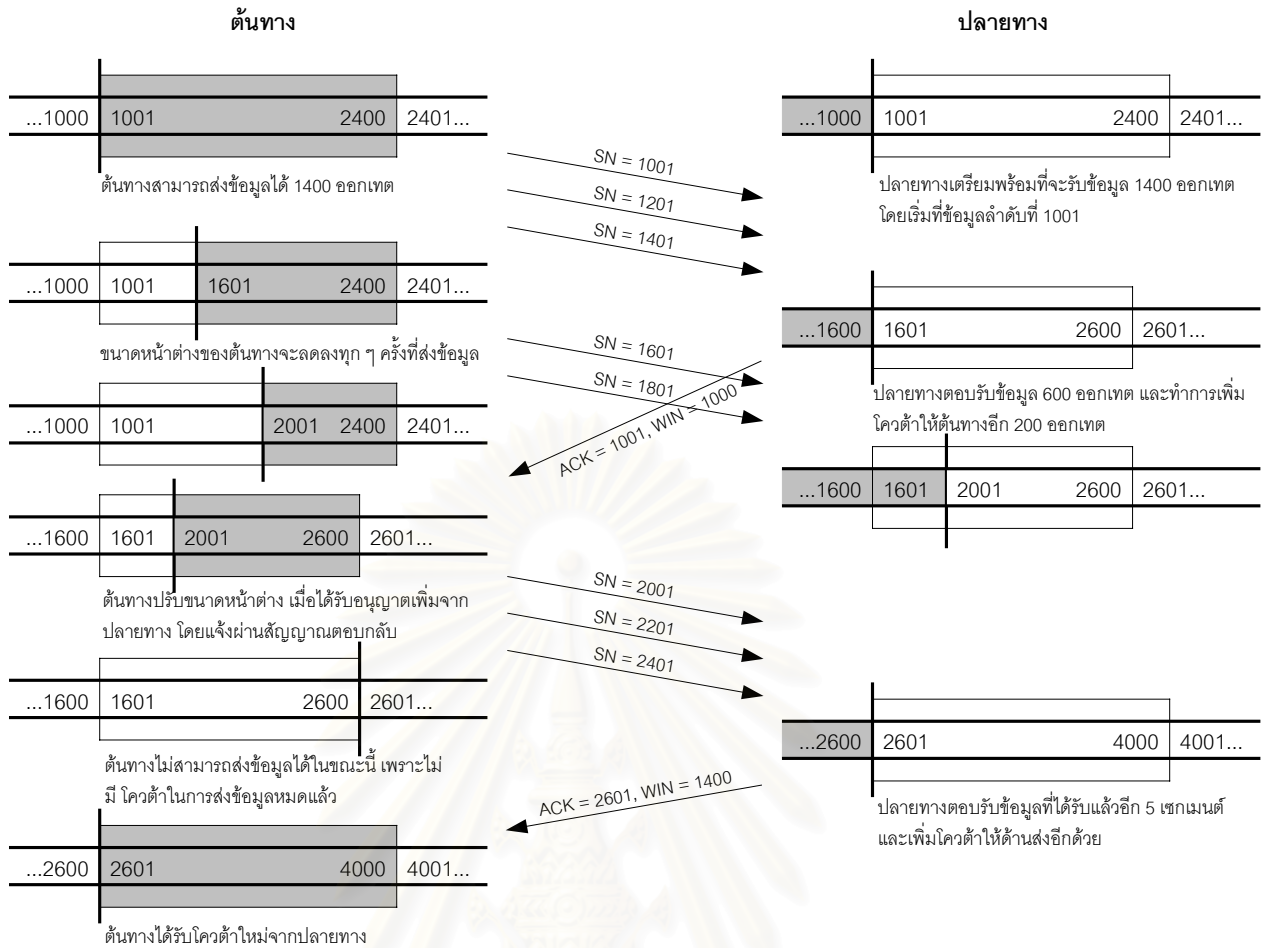
โพรโทคอล TCP เป็นโพรโทคอลหนึ่งที่ใช้เทคนิค Sliding Window ในการควบคุมการส่งข้อมูล โดยการทำงานของระบบการควบคุมการส่งข้อมูลของ TCP เริ่มจากการกำหนด Sequence Number ให้กับข้อมูลแต่ละออกเทต ที่จะส่งผ่าน TCP และมีการใส่ค่า Sequence Number ของข้อมูลออกเทตแรกที่ยังบรรจุอยู่ในส่วนข้อมูลลงในส่วนเฮดเดอร์ เพื่อระบุลำดับของแพ็กเก็ตที่ส่ง สำหรับการส่งข้อมูล ต้นทางจะมีการคำนวณค่าปริมาณข้อมูลที่ส่งได้ โดยไม่ต้องรอรับสัญญาณตอบกลับจากปลายทาง เรียกปริมาณข้อมูลนี้ว่า "ขนาดหน้าต่าง หรือ Window Size" โดย TCP จะมีกลไกในการควบคุมค่าขนาดหน้าต่างดังกล่าว (รายละเอียดกล่าวในส่วนท้าย

ของบท) ทั้งนี้ตัวแปรสำคัญในการกำหนดขนาดหน้าต่าง คือข้อมูลที่รับจากสัญญาณตอบกลับ โดยทั่วไปอยู่ในลักษณะ ($ACK = i, WIN = j$) ซึ่งสามารถตีความได้ดังนี้

($ACK = i, WIN = j$)

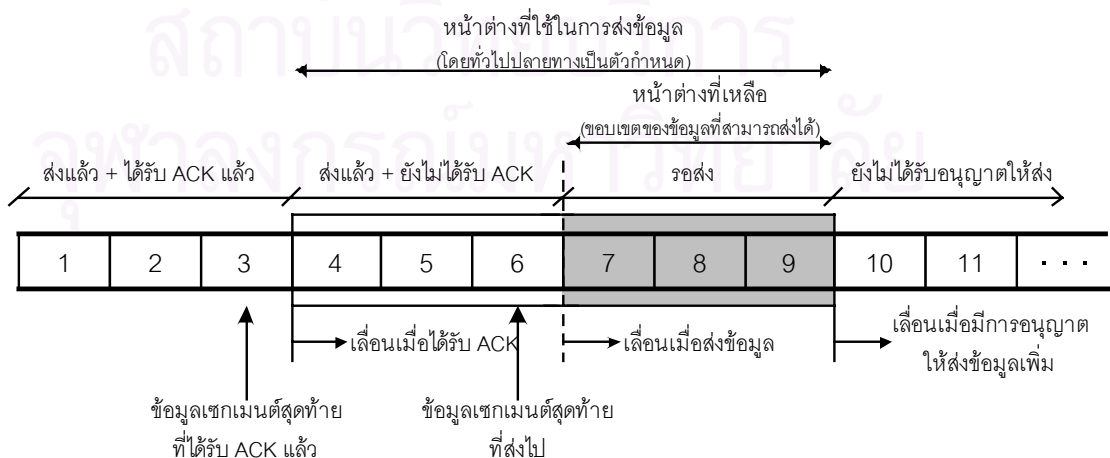


ตัวอย่างกลไกการควบคุมการส่งข้อมูลที่ใช้หลักการ Sliding Window พิจารณาได้จากรูปที่ 3.3 โดยตัวอย่างนี้มีสมมุติฐานเบื้องต้นให้ขนาดของ TCP เซกเมนต์เท่ากับ 200 ออกเขต และการส่งข้อมูลแบบทิศทางเดียว กระบวนการส่งข้อมูลเริ่มต้นจากกระบวนการสร้างการเชื่อมต่อ ซึ่งเป็นช่วงที่ต้นทางและปลายทางทำการตกลงกันก่อนมีการส่งข้อมูลจริง ไม่ว่าจะเป็นเรื่องของ Sequence Number ค่าแรกที่จะใช้ในการติดต่อ หรือเรื่องของขนาดหน้าต่างที่ยอมรับกันได้ (ในตัวอย่างจะเริ่มส่งที่ Sequence Number = 1001 และขนาดหน้าต่างเท่ากับ 1400) หลังจากที่ต้นทางส่งข้อมูลไปแล้ว 3 เซกเมนต์ หรือ 600 ออกเขต ขนาดหน้าต่างของต้นทางจะเหลือเพียง 800 ออกเขต (นั่นคือ ตั้งแต่ลำดับที่ 1601 ถึง 2400) เมื่อปลายทางได้รับข้อมูลทั้ง 3 เซกเมนต์เรียบร้อยแล้ว ปลายทางก็ทำการส่งสัญญาณตอบกลับให้แก่ต้นทาง ($ACK=1601, W=1000$) เพื่อบอกต้นทางให้รู้ว่า ปลายทางได้รับข้อมูลถึงลำดับที่ 1600 เรียบร้อยแล้ว อีกทั้งยังบอกถึงขนาดหน้าต่างที่ปลายทางอนุญาตเพิ่มมีค่าเท่ากับ 1000 (หมายถึง ต้นทางได้รับอนุญาตให้ส่งข้อมูลได้ตั้งแต่ข้อมูลลำดับที่ 1601 จนถึง 2600 ซึ่งเท่ากับ 5 เซกเมนต์) อย่างไรก็ตามก่อนที่ต้นทางจะได้รับสัญญาณตอบกลับจากปลายทาง ต้นทางซึ่งส่งข้อมูลออกไปเรื่อย ๆ ก็ได้ส่งข้อมูลไปแล้วอีก 2 เซกเมนต์ ทำให้ขนาดหน้าต่างของต้นทางก่อนได้รับสัญญาณตอบกลับเหลือเพียง 400 ออกเขตเท่านั้น แต่เมื่อข้อมูลตอบกลับมาถึงต้นทาง ต้นทางก็ปรับขนาดหน้าต่างถึงข้อมูลลำดับที่ 2601 ซึ่งทำให้มีขนาดหน้าต่างเพิ่มเป็น 600 ออกเขต อีกทั้งยังปรับปรุงหน่วยความจำต้นทางให้เริ่มจากข้อมูลลำดับที่ 1601 ด้วย (ทั้งนี้เพราะข้อมูลลำดับที่ 1601 เป็นข้อมูลออกเขตแรกที่ยังไม่ได้รับการยืนยันผลการส่งข้อมูลจากปลายทาง) และการส่งข้อมูลจะดำเนินไปในลักษณะนี้จนสิ้นสุดการส่ง

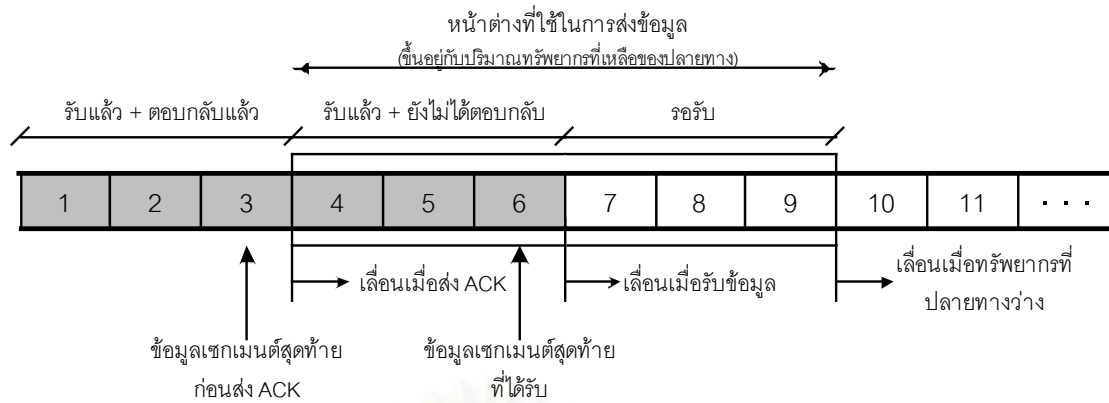


รูปที่ 3.3 ตัวอย่างการควบคุมการส่งข้อมูลโดยใช้หลักการ Sliding Window

จากตัวอย่างข้างต้น จะเห็นว่าขนาดหน้าต่างที่ต้นทางจะหดทุกครั้งที่มีการส่งข้อมูล และจะขยายขอบหน้าต่างเมื่อได้รับสัญญาณตอบกลับจากปลายทาง ที่บ่งบอกถึงการอนุญาตให้ส่งข้อมูลเพิ่ม ทั้งนี้ปรากฏการณ์ที่เกิดขึ้นทั้งด้านต้นทางและปลายทางแสดงได้ดังรูปที่ 3.4



(ก) กลไกการทำงานของหน่วยความจำต้นทาง



(ข) กลไกการทำงานของหน่วยความจำปลายทาง

รูปที่ 3.4 ลักษณะการทำงานของหน่วยความจำต้นทางและปลายทาง

จะเห็นได้ว่าค่าวิสัยสามารถที่เกิดขึ้นจากการควบคุมการส่งข้อมูลในลักษณะที่เพิ่งอธิบายไปนี้มีขีดจำกัด โดยเฉพาะอย่างยิ่งกับการเชื่อมต่อที่มีระยะเวลาประวิงเวลามาก ๆ ทั้งนี้เพราะการทำงานของโพรโทคอลแบบนี้ต้นทางจะพิจารณาขนาดหน้าต่างจากค่าขนาดหน้าต่างที่ได้รับจากปลายทางเพียงอย่างเดียว อีกทั้งการกำหนดค่าขนาดหน้าต่างของปลายทาง ก็พิจารณาจากหน่วยความจำที่ว่างจากการใช้งานเท่านั้น ซึ่งถ้าหน่วยความจำปลายทางเต็ม สัญญาณตอบกลับที่ส่งออกไป จะไม่มีการอนุญาตให้ต้นทางส่งข้อมูลใด ๆ เพิ่ม ทั้ง ๆ ที่ปลายทางเองมีศักยภาพในการจัดการกับหน่วยความจำ เพื่อให้สามารถรองรับข้อมูลได้ ก่อนที่ข้อมูลชุดใหม่ที่เป็นผลจากสัญญาณตอบกลับจะมาถึง จะเห็นได้ว่าถ้าปลายทางมีความสามารถในการรับข้อมูลทั้งหมดที่ส่งมาจากต้นทาง (นั่นคือความเร็วในการทำงานของต้นทางไม่เหนือไปกว่าปลายทาง) ประสิทธิภาพในการส่งข้อมูล ในแง่ค่าวิสัยสามารถจะสูง อีกทั้งการส่งข้อมูลดังกล่าวยังไม่เกิดผลกระทบต่อข้อมูลที่ส่งอีกด้วย ในทางกลับกัน ถ้าต้นทางมีความสามารถในการทำงานที่เร็วกว่าปลายทาง ปัญหาที่เกิดขึ้นคือ ข้อมูลบางส่วนจะถูกทิ้ง และมีความจำเป็นต้องส่งข้อมูลใหม่อีกครั้ง ทั้งนี้เห็นว่าการกำหนดอัตราการส่งข้อมูลที่เหมาะสมต้องอาศัยหลักการที่ค่อยข้างช้าช้ากว่า หลักการทำงานทั่วไปของ Sliding Window

3.2.2 หลักการที่ใช้ในการควบคุมการส่งข้อมูลของ TCP

ในปัจจุบันมีการพัฒนาเทคนิคที่ใช้ร่วมกับกระบวนการควบคุมการไหลของข้อมูลบน TCP ขึ้นมากมาย ทั้งนี้เพื่อเพิ่มประสิทธิภาพในการส่งข้อมูล ซึ่งกระบวนการดังกล่าวส่วนใหญ่จะตั้งอยู่ในหลักการอย่างน้อยหนึ่งในสามประการต่อไปนี้ Slow Start and Congestion Avoidance, Fast Retransmit and Recovery -- FRR และ Selective ACKnowledgement -- SACK [2] สำหรับใน

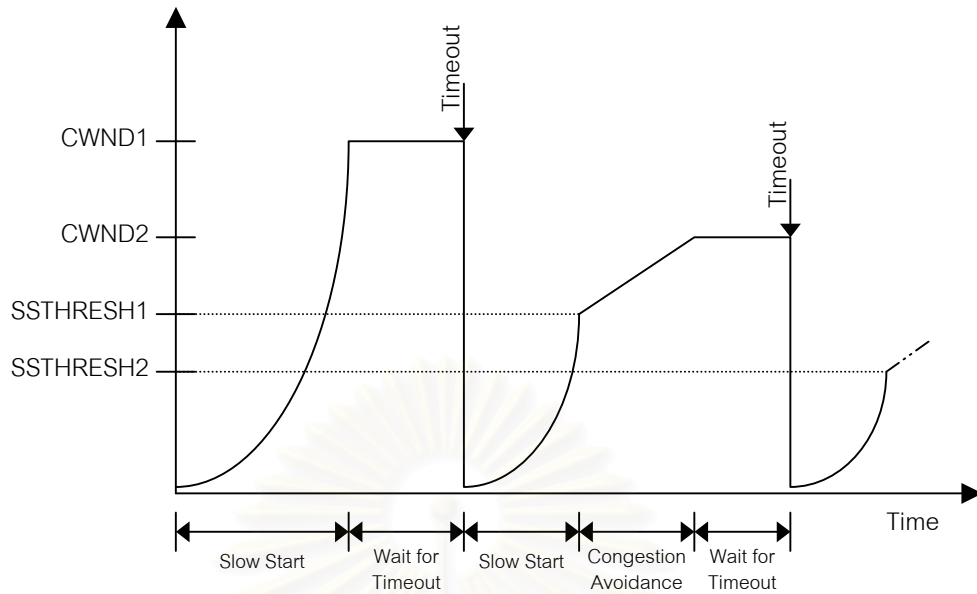
วิทยานิพนธ์ฉบับนี้จะอ้างอิงเฉพาะหลักการ Slow Start and Congestion Avoidance เท่านั้นทั้งนี้ เพราะหลักการ Slow Start and Congestion Avoidance ถือเป็นหลักการพื้นฐานที่เป็นที่รู้จัก และยังถูกใช้งานอย่างกว้างขวางอีกด้วย

3.2.2.1 หลักการทำงานแบบ Slow Start and Congestion Avoidance

หลักการของเทคนิค Slow Start and Congestion Avoidance เป็นการใช้กระบวนการ *ทดลองและตรวจสอบ* ในการควบคุมปริมาณข้อมูลที่ส่ง เพื่อหาอัตราการส่งข้อมูลที่เหมาะสมกับสภาพโครงข่าย กล่าวคือต้นทางจะเริ่ม *ทดลอง* ส่งข้อมูลที่อัตราการส่งน้อย ๆ แล้วค่อย ๆ เพิ่มปริมาณข้อมูลที่ส่งทุกครั้งที่ได้รับผลการ *ตรวจสอบ* (ผ่านทางสัญญาณตอบกลับ) ที่แสดงถึงความถูกต้องของข้อมูลที่ปลายทางได้รับ ในทางกลับกันถ้าผลการตรวจสอบระบุความเสียหายในการส่ง ไม่ว่าจะเกิดสาเหตุมาจากการสูญหายของข้อมูล หรือข้อมูลที่รับได้มีความผิดพลาด ต้นทางจะตีความผลการตรวจสอบดังกล่าว ว่าเกิดความคับคั่งขึ้น ณ จุดใดจุดหนึ่งในโครงข่าย ดังนั้นต้นทาง จะทำการลดปริมาณข้อมูลที่ส่งทันที โดยต้นทางจะเริ่มส่งข้อมูลที่อัตราการส่งน้อย ๆ แล้วค่อย ๆ เพิ่มปริมาณข้อมูลที่ส่งอีกครั้ง แต่ลักษณะการเพิ่มอัตราการส่งในกรณีหลังนี้มีรูปแบบที่แตกต่างจากการเพิ่มช่วงแรกเล็กน้อย คือมีการนำข้อมูล ณ จุดเกิดความคับคั่งมาร่วมพิจารณาด้วย

สำหรับการควบคุมปริมาณข้อมูลที่ต้นทางใช้ในการส่งนั้น จะทำโดยผ่านตัวแปรที่เรียกว่า *ขนาดหน้าต่างความคับคั่ง* (*Congestion window size -- CWND*) ซึ่งค่าขนาดหน้าต่างความคับคั่งนี้ เป็นค่าที่แสดงถึงปริมาณหน่วยข้อมูลที่ต้นทางสามารถส่งไปยังปลายทางได้ โดยไม่จำเป็นต้องรอรับสัญญาณตอบกลับ ทั้งนี้การปรับค่าขนาดหน้าต่างความคับคั่งดังกล่าว จำเป็นต้องอยู่ภายใต้ความสามารถในการรองรับของอุปกรณ์ปลายทางด้วย ซึ่งพิจารณาจากค่า *ขนาดหน้าต่างด้านรับ* (*Receiver window size -- RCWIND*) นั้นเอง (กล่าวคือ $CWND \leq RCWIND$ เสมอ)

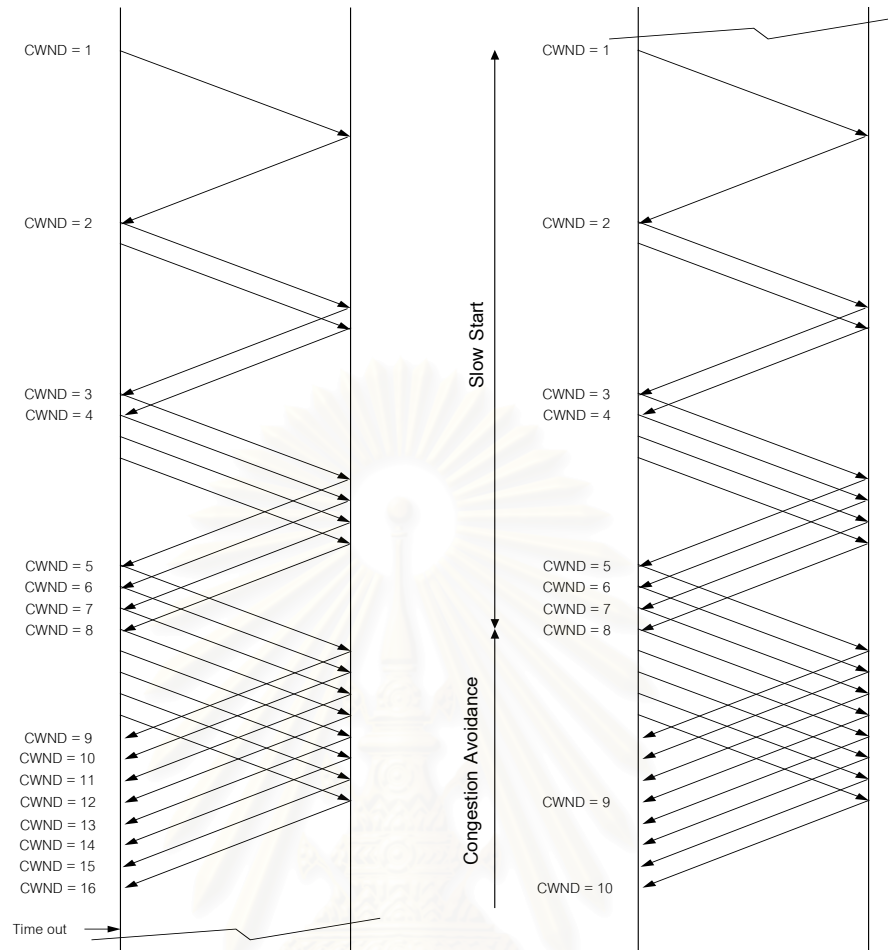
การทำงานของกระบวนการควบคุมการไหลของข้อมูล TCP โดยใช้เทคนิค Slow Start and Congestion Avoidance เริ่มต้นจาก ตั้งค่า CWND ที่ต้นทางเท่ากับหนึ่งเซกเมนต์ และ CWND นี้จะมีค่าเพิ่มขึ้นทีละหนึ่งเซกเมนต์ ทุกครั้งที่ต้นทางได้รับสัญญาณตอบกลับที่ระบุว่าการส่งข้อมูลไม่มีปัญหา โดยการเพิ่มค่า CWND จะอยู่ในขอบเขตที่กำหนดโดยความสามารถสูงสุดในการส่งข้อมูลของต้นทาง (ปรกติมีค่าเท่ากับ 64 Kbytes) และในการรับข้อมูลของปลายทาง (พิจารณาจากค่า RCWIND) จากรูปที่ 3.5 จะเห็นว่าการเพิ่มของขนาดหน้าต่างเมื่อเทียบกับเวลา มีความสัมพันธ์ในรูปแบบเอกซ์โพเนนเชียล (exponential) นั่นคือทุก ๆ รอบของการส่งข้อมูล (Round trip time -- RTT) ขนาดหน้าต่างจะเพิ่มขึ้นถึงหนึ่งเท่า เรียกการทำงานลักษณะนี้ว่า *Slow Start*



รูปที่ 3.5 ความสัมพันธ์ของขนาดหน้าต่างเทียบกับเวลา ของการควบคุมการส่งข้อมูลที่ใช้เทคนิค Slow Start and Congestion avoidance

สำหรับกรณีที่การส่งข้อมูลเกิดความผิดพลาด ต้นทางรับรู้ได้จากการรับสัญญาณตอบกลับเหมือนกันติดกัน 2 ครั้ง และไม่มีสัญญาณตอบกลับที่บ่งบอกถึงการส่งข้อมูลถูกต้องภายในเวลาที่ตั้งเอาไว้ ต้นทางจะตีความว่าเกิดความคับคั่งขึ้นในโครงข่าย ซึ่งต้นทางจะแก้ปัญหาดังกล่าว โดยการลดไหลดของโครงข่ายลง ทั้งนี้ต้นทางจะบันทึกค่าครึ่งหนึ่งของ CWND ในขณะนั้นไว้ในตัวแปรที่เรียกว่า *Slow Start Threshold* --SSTHRESH จากนั้นต้นทางจะตั้งค่า CWND เท่ากับหนึ่งเซกเมนต์ใหม่อีกครั้ง โดยการเริ่มส่งข้อมูลที่ถูกระบุว่าเสียหายก่อน แล้วจึงส่งข้อมูลชุดถัดไป โดยอัตราการส่งจะถูกควบคุมด้วยค่า CWND ที่ปรับตามหลักการ *Slow Start* ที่กล่าวไว้ข้างต้น นั่นคือค่า CWND จะค่อย ๆ เพิ่มขึ้นที่ละหนึ่งเซกเมนต์หลังจากได้รับสัญญาณตอบกลับ แต่การเพิ่มลักษณะนี้จะสิ้นสุดลงเมื่อค่า CWND มีค่าเท่ากับ SSTHRESH หลังจากนั้นการเพิ่มค่า CWND จะถูกควบคุมโดยใช้หลักการที่เรียกว่า *Congestion Avoidance* คือค่า CWND จะถูกเพิ่มขึ้นหนึ่งเซกเมนต์ เมื่อครบรอบการส่งหนึ่งรอบ (คือ ทุก ๆ RTT ขนาดหน้าต่างความคับคั่งจะเพิ่มขึ้นหนึ่งเซกเมนต์) เมื่อพิจารณาการเพิ่มดังกล่าวเทียบกับแกนเวลา จะเห็นว่าความสัมพันธ์ที่เกิดขึ้นจากการ *Congestion Avoidance* นี้จะอยู่ในรูปแบบเชิงเส้น แสดงดังรูปที่ 3.5

สำหรับตัวอย่างการส่งข้อมูล และการเพิ่มขนาดหน้าต่าง ภายใต้การทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการ *Slow Start and Congestion Avoidance*



รูปที่ 3.6 ตัวอย่างการส่งข้อมูลของ TCP โดยใช้หลักการ Slow Start and Congestion Avoidance

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM

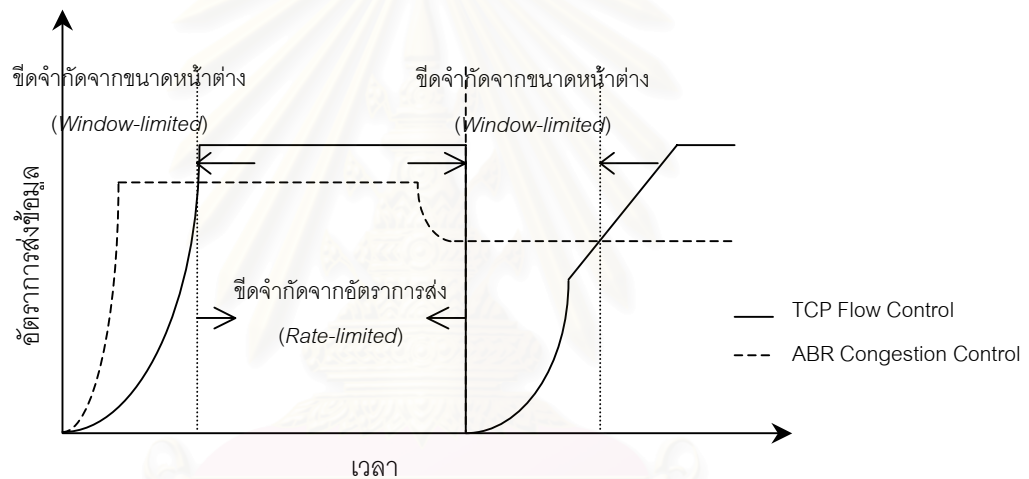
เมื่อกล่าวถึงการควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM คงจะปฏิเสธความสัมพันธ์ระหว่างกระบวนการควบคุมการไหลของ TCP ในชั้นทรานส์พอร์ต และกระบวนการควบคุมความคับคั่งของ ABR ในชั้น ATM ไปไม่ได้ ถึงแม้ว่ากระบวนการควบคุมทั้งสองชนิดดังกล่าวทำงานกันคนละระดับชั้น แต่กลับมีเป้าหมายเหมือนกัน คือพยายามจัดสรรทรัพยากรต่าง ๆ ของโครงข่ายให้นำไปใช้งานอย่างมีประสิทธิภาพที่สุด อย่างไรก็ตามเนื่องจากหลักการและวิธีการทำงานของกระบวนการควบคุมทราฟฟิกทั้งสองไม่มีความเกี่ยวข้องกัน ซึ่งเป็นผลให้เกิดความไม่สอดคล้องกันในการทำงาน ด้วยสาเหตุนี้เองประสิทธิภาพการทำงานโดยรวมของระบบ จึงเกิดขีดจำกัดขึ้น ทำให้แหล่งกำเนิดไม่สามารถส่งข้อมูลได้อย่างเต็มประสิทธิภาพ และจากประเด็นนี้เอง ทำให้กลุ่มผู้วิจัยหลายกลุ่มเล็งเห็นถึงความสำคัญและความน่าสนใจของปัญหานี้ จึงพยายามคิดค้น พัฒนาและปรับปรุงกลไกการทำงาน เพื่อให้ประสิทธิภาพการทำงานร่วมกันของกระบวนการทั้งสองดีขึ้น

เมื่อเปรียบเทียบการทำงานของกระบวนการควบคุมความคับคั่งที่ทำในชั้น ATM และกระบวนการควบคุมการไหลของ TCP จะเห็นว่ากระบวนการควบคุมความคับคั่งของ ABR สามารถเกาะติดสภาวะการเปลี่ยนแปลงทรัพยากรในโครงข่ายได้ดีกว่า ทั้งนี้เนื่องมาจากกระบวนการควบคุมความคับคั่งที่ใช้กับบริการแบบ ABR อาศัยเซลล์ RM ที่กระจายตัวอยู่ในข้อมูลอย่างสม่ำเสมอ ในการตรวจสอบสภาพความคับคั่งของโครงข่าย ซึ่งเซลล์ RM นี้จะเก็บข้อมูลที่บ่งบอกถึงสถานการณ์ในโครงข่าย ที่บันทึกโดยอุปกรณ์ที่การเชื่อมต่อผ่าน ในขณะที่กระบวนการควบคุมการไหลของ TCP ตรวจสอบสภาพของโครงข่าย โดยการสังเกตจากจังหวะการได้รับสัญญาณตอบกลับจากโครงข่าย ดังนั้นข้อมูลที่ได้รับจากกระบวนการควบคุมความคับคั่งในชั้น ATM จึงมีความแม่นยำ และทันเหตุการณ์กว่า และด้วยเหตุผลดังกล่าวทำให้นักวิจัยทั้งหมดรวมทั้งงานวิจัยชิ้นนี้ เลือกรวบรวมข้อมูลที่ได้จากกระบวนการควบคุมความคับคั่งของ ABR มาเป็นพื้นฐานในการปรับปรุงกระบวนการควบคุมการไหลของ TCP เพื่อลดปัญหาความไม่สอดคล้องกันของทั้งสองกระบวนการ และเพื่อเพิ่มประสิทธิผลของการเชื่อมต่อด้วย

เมื่อรวบรวมงานวิจัยที่เกี่ยวกับกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM สามารถแบ่งออกเป็น 4 ประเภทใหญ่ ๆ ด้วยกันคือ

4.1 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการดั้งเดิม

การควบคุมอัตราการส่งข้อมูลของ TCP โดยอาศัยหลักการดั้งเดิมนี้ เป็นการนำหลักการพื้นฐานของทั้งกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของบริการ ABR นั้นคือการใช้งาน TCP บนบริการแบบ ABR โดยไม่มีการปรับปรุงกระบวนการทำงานใด ๆ เลย เนื่องจากกระบวนการทั้งสองไม่ได้ถูกออกแบบมาให้ทำงานด้วยกันโดยตรง ซึ่งการทำงานที่เป็นอิสระต่อกันอย่างสิ้นเชิงของกระบวนการทั้งสองนี้เอง ทำให้เกิดความไม่สอดคล้องในการทำงานร่วมกัน และเป็นผลให้เกิดขีดจำกัดในการสื่อสารข้อมูลขึ้น ปัญหาดังกล่าวสามารถแสดงได้จากความสัมพันธ์ของอัตราการส่งข้อมูลในชั้นทรานส์พอร์ต ซึ่งควบคุมโดยกระบวนการควบคุมการไหลของ TCP และอัตราการส่งข้อมูลของบริการแบบ ABR ในชั้น ATM ดังรูปต่อไปนี้



รูปที่ 4.1 เปรียบเทียบอัตราการส่งข้อมูลที่เกิดจากกระบวนการควบคุมการไหลของข้อมูลบน TCP กับกระบวนการควบคุมความคับคั่งของทราฟฟิกบนบริการแบบ ABR

จากรูปจะเห็นว่าในช่วงแรกของการติดต่อสื่อสาร กระบวนการควบคุมการไหลของ TCP ค่อย ๆ ควบคุมการเพิ่มของอัตราการส่งข้อมูล (ขนาดหน้าต่าง) ตามรูปแบบของ *Slow Start* ในขณะที่กระบวนการควบคุมความคับคั่งของ ABR ปรับอัตราการส่งข้อมูลให้ถึงจุดทำงานที่คำนวณจากสภาพโครงข่าย ตามหลักการทั่วไปของกระบวนการควบคุมความคับคั่งของทราฟฟิกแบบ ABR ซึ่งได้กล่าวไว้แล้วในบทที่ 2 เมื่อมาวิเคราะห์ถึงลักษณะการปรับอัตราการส่งข้อมูลที่เกิดจากทั้งสองกระบวนการ จะเห็นว่ากระบวนการควบคุมความคับคั่งของ ABR มีการปรับอัตราการส่งข้อมูลที่รวดเร็วกว่า ทั้งนี้เนื่องมาจากระยะเวลาในการปรับแต่ละครั้งสั้นกว่า จากข้อสังเกตนี้เอง ทำให้การส่งข้อมูลผ่านการเชื่อมต่อดังกล่าวถูกจำกัดด้วยกระบวนการควบคุมการไหลของ TCP หรือ

อาจกล่าวอีกนัยหนึ่งว่าปริมาณข้อมูลที่ส่งผ่านการเชื่อมต่อ นั้น จะถูกจำกัดด้วยขนาดหน้าต่างที่กำหนดโดยกระบวนการควบคุมการไหลของ TCP ซึ่งปรากฏการณ์นี้ถูกเรียกว่า “ขีดจำกัดในการส่งข้อมูลเนื่องจากขนาดหน้าต่าง” (Window-limited) [2] นอกจากนี้จะเห็นว่าการเกิดขีดจำกัดในการส่งข้อมูลเนื่องจากขนาดหน้าต่าง ทำให้ประสิทธิภาพการใช้งานโครงข่ายลดลงด้วย

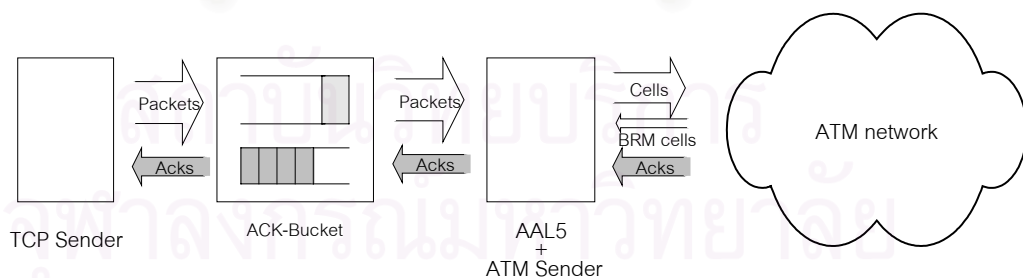
นอกจากขีดจำกัดที่เกิดจากขนาดหน้าต่างแล้ว การทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ยังเกิด “ขีดจำกัดที่เกิดจากอัตราการส่งข้อมูล” (Rate-limited) [2] อีกด้วย กล่าวคือโดยปกติกระบวนการควบคุมการไหลของ TCP จะเพิ่มขนาดหน้าต่างไปเรื่อย ๆ ราบใดที่ยังไม่ได้รับสัญญาณตอบกลับที่บ่งบอกถึงความผิดพลาดในการส่งข้อมูลจากโครงข่าย หรือยังคงไม่เกินความสามารถของตัวรับ (พิจารณาจากขนาดหน้าต่างของตัวรับ) ทั้งนี้เมื่อกระบวนการควบคุมการไหลของ TCP ปรับขนาดหน้าต่างจนมีค่ามากกว่าขนาดหน้าต่างที่สอดคล้องกับการส่งข้อมูลในชั้น ATM ทำให้ดูเหมือนว่าการเชื่อมต่อถูกจำกัดด้วยอัตราการส่งข้อมูลที่กำหนดโดยกระบวนการควบคุมความคับคั่งของ ABR แต่ในทางปฏิบัติกระบวนการควบคุมการไหลของ TCP ต่างหากที่มีการกำหนดปริมาณข้อมูลที่ให้ส่งมากเกินไป ปรากฏการณ์ดังกล่าวจึงถูกเรียกว่า ขีดจำกัดที่เกิดจากอัตราการส่งข้อมูล

สำหรับปัญหาที่อาจจะเกิดขึ้นพร้อม ๆ กับขีดจำกัดที่เกิดจากอัตราการส่งข้อมูล คือหน่วยความจำที่อุปกรณ์ต้นทางอาจมีความสามารถไม่เพียงพอที่จะรองรับข้อมูลที่เกิดจากการกำหนดขนาดหน้าต่างที่มากเกินไปของกระบวนการควบคุมการไหลของ TCP ซึ่งทำให้ข้อมูลเกิดความเสียหายตั้งแต่ยังไม่มีการส่งข้อมูลจริง ผลกระทบดังกล่าวจะมีบทบาทมากขึ้น ถ้าความสามารถในการส่งข้อมูลของการเชื่อมต่อลดลง (อาจเกิดจาก มีการเชื่อมต่อใหม่ต้องการติดต่อกับโครงข่าย ซึ่งเป็นผลให้การเชื่อมต่อที่ติดต่อกับโครงข่ายอยู่ก่อนแล้วต้องแบ่งทรัพยากรให้การเชื่อมต่อใหม่ดังกล่าว) ในขณะที่ขนาดหน้าต่างที่กำหนดโดยกระบวนการควบคุมการไหลของ TCP ยังคงไม่เปลี่ยนแปลง ทำให้ระยะห่างระหว่างขนาดหน้าต่างที่เหมาะสมกับความสามารถในการส่งข้อมูลจริง กับขนาดหน้าต่างที่ถูกกำหนดโดยกระบวนการควบคุมการไหลของ TCP กว้างขึ้นอีก และนั่นเป็นการเพิ่มโอกาสที่จะทำให้เกิดปัญหา อันเนื่องมาจากความสามารถที่มีอยู่จำกัดของอุปกรณ์ต้นทางอีกด้วย

ในกรณีของการเชื่อมต่อที่มีการตรวจพบความผิดพลาดของข้อมูล กระบวนการควบคุมการไหลของ TCP จะเริ่มตั้งค่าขนาดหน้าต่างจากค่าน้อย ๆ ใหม่อีกครั้ง (เข้าสู่ช่วงการทำงานแบบ *Slow Start*) นั่นก็หมายความว่า การเชื่อมต่อดังกล่าวอยู่ภายใต้อิทธิพลของขีดจำกัดเนื่องจากขนาดหน้าต่างอีกครั้ง นั่นคือการเชื่อมต่อไม่สามารถใช้งานข่ายเชื่อมโยงอย่างมีประสิทธิภาพตลอดช่วงระยะเวลาดังกล่าวนั่นเอง

4.2 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ [10]

จากที่ได้กล่าวไว้ข้างต้นถึงการทำงานของ TCP บนบริการแบบ ABR ในโครงข่าย ATM เป็นการดำเนินงานร่วมกันระหว่างกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของ ABR ซึ่งเมื่อพิจารณาการทำงานในแง่ที่ว่า กระบวนการควบคุมความคับคั่งของ ABR สามารถตรวจสอบและหลีกเลี่ยงการเกิดความคับคั่งภายในโครงข่ายได้ แต่ไม่ได้ครอบคลุมอุปกรณ์ส่วนที่เหลือ กล่าวคือกระบวนการควบคุมความคับคั่งของ ABR ครอบคลุมเฉพาะส่วนของสวิตช์และอุปกรณ์ปลายทาง (พิจารณาถึงส่วนของชั้น ATM) ที่การเชื่อมต่อนั้นผ่านเท่านั้น แต่ไม่มีการตรวจสอบและควบคุมความคับคั่งที่อาจเกิดขึ้นที่อุปกรณ์ส่วนที่เหลือ ซึ่งปัญหานี้จะทวีความรุนแรงขึ้นถ้าโพรโตคอลในชั้นทรานส์พอร์ต และ ATM ทำงานไม่สอดคล้องกัน ทำให้เกิดสถานะข้อมูลล้นตั้งแต่ยังไม่ได้เริ่มส่ง สำหรับการแก้ปัญหาดังกล่าวนี้ อาจทำได้โดยการเพิ่มขนาดหน่วยความจำที่อุปกรณ์ต้นทางหรือปลายทาง เพื่อลดอัตราการเสี่ยงของการล้นของข้อมูล แต่ปัญหาที่น่าขบคิดอีกข้อก็ตามมาคือขนาดหน่วยความจำที่เหมาะสมควรจะเป็นเท่าไร อีกทั้งยังต้องคำนึงถึงค่าใช้จ่ายที่เพิ่มขึ้นอีกด้วย ดังนั้นกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับจึงถูกคิดค้นขึ้น โดยมุ่งเน้นไปที่การลดขนาดหน่วยความจำและเพื่อหลีกเลี่ยงการเกิดความคับคั่งที่อุปกรณ์ต้นทาง นอกจากนี้กระบวนการควบคุมอัตราการส่งข้อมูลของ TCP แบบนี้ยังช่วยแก้ปัญหาขีดจำกัดที่เกิดจากอัตราการส่งข้อมูลดังที่ได้กล่าวข้างต้นอีกด้วย สำหรับหลักการการทำงานของกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ หรือที่รู้จักโดยทั่วไปว่า “Acknowledgement bucket” [10] แสดงคร่าว ๆ ดังรูป



รูปที่ 4.2 แสดงการทำงานของกระบวนการควบคุมการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ

กระบวนการควบคุมการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ เป็นการปรับปรุงการทำงานของกระบวนการควบคุมการไหลของ TCP โดยการเพิ่มฟังก์ชันการทำงาน

ย่อยอีก 4 ส่วน คือ Ack-Bucket, Observer, Translation Explicit Rate to Ack Sequence และ Error recovery โดยการทำงานของแต่ละส่วนสามารถอธิบายคร่าว ๆ ดังนี้

Ack-Bucket ทำหน้าที่ดักสัญญาณตอบกลับที่ได้จากโครงข่าย และจัดเก็บในหน่วยความจำที่เตรียมไว้ สำหรับกรณีที่มีการส่งข้อมูลทั้งสองทาง ซึ่งโดยมากสัญญาณตอบกลับจะถูกส่งมาพร้อมกับข้อมูล เพื่อไม่ให้เกิดผลกระทบต่อกรับข้อมูล Ack-Bucket จึงสร้างสัญญาณตอบกลับขึ้นมาใหม่ โดยสำเนาข้อมูลมาจากสัญญาณตอบกลับเดิม จากนั้น Ack-Bucket จะจัดเก็บเฉพาะสัญญาณตอบกลับตัวใหม่ และส่งส่วนของข้อมูลไปยังอุปกรณ์รับข้อมูลต่อไป

Observer ทำหน้าที่คล้ายตัวสังเกตการณ์การเพิ่มขนาดหน้าต่างของตัวส่งข้อมูล TCP เมื่อได้รับสัญญาณตอบกลับ ซึ่งในทางปฏิบัติกระบวนการนี้จะไม่มีการเพิ่มขนาดหน้าต่างจริง เพราะฉะนั้น *Observer* จึงเป็นเพียงการสังเกตจากการคำนวณและคาดเดาค่าขนาดหน้าต่างที่ควรจะเป็น ซึ่งค่าขนาดหน้าต่างดังกล่าวนี้จะถูกพิจารณาตามทฤษฎีพื้นฐานของ TCP นั่นคือเริ่มต้นด้วยการเพิ่มแบบเอกซ์โพเนนเชียล (*Slow Start*) และเมื่อค่าขนาดหน้าต่างมีค่ามากกว่าค่า *Slow Start Threshold* การเพิ่มขนาดหน้าต่างจะเปลี่ยนเป็นแบบเชิงเส้น (*Congestion Avoidance*) และจะหยุดการเพิ่มในที่สุด (*Saturation*) เมื่อขนาดหน้าต่างเท่ากับค่าขนาดหน้าต่างมากที่สุด

Translating Explicit Rate to Ack Sequence ทำหน้าที่กำหนดจังหวะการปล่อยสัญญาณตอบกลับและกำหนดปริมาณข้อมูลที่จะส่ง โดยคำนวณจากอัตราการส่งข้อมูลที่ได้จากระบบการควบคุมความคับคั่งของ ABR และผลที่ได้จาก *Observer* นั่นคือ *Translating Explicit Rate to Ack Sequence* จะใช้ผลที่ได้จาก *Observer* มากำหนดปริมาณข้อมูลในการส่งแต่ละครั้ง และใช้ค่าที่ได้จากระบบการควบคุมความคับคั่งของ ABR ซึ่งมีความสัมพันธ์โดยตรงกับอัตราการส่งข้อมูลจริง มากำหนดจังหวะการปล่อยสัญญาณตอบกลับ และจากผลของการทำงานดังกล่าวนี้เอง ทำให้ TCP สามารถส่งข้อมูลที่อัตราการส่งที่การเชื่อมต่อใช้จริง ในปริมาณที่ไม่ทำให้เกิดการสะสมของข้อมูลก่อนมีการส่งจริง (นั่นคือ สามารถลดขนาดหน่วยความจำที่ใช้ได้)

Error recovery เป็นส่วนที่ถูกเรียกใช้เมื่อเกิดความเสียหายกับข้อมูลที่ส่ง โดย *Error recovery* จะตั้งขนาดหน้าต่างใหม่ให้ไปเริ่มต้นที่ 1 แพ็กเกต และบันทึกค่า *Slow Start Threshold* เท่ากับครึ่งหนึ่งของขนาดหน้าต่างก่อนเกิดความเสียหาย

จากการทำงานทั้งหมดทำให้ *Throughput* ของกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับมีลักษณะเหมือน *Throughput* ที่ได้จากระบบการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการดั้งเดิม เนื่องจากการทำงานของ *Observer* และ *Error recovery* ยังคงยึดหลักการทำงานดั้งเดิมของ TCP รวมทั้งจังหวะของ

สัญญาณตอบกลับที่ทำให้อัตราการส่งข้อมูลสอดคล้องกับการส่งข้อมูลจริง ซึ่งเป็นผลให้ปริมาณการส่งจาก TCP สอดคล้องกับปริมาณข้อมูลที่ถูกส่งจริง นั่นคือไม่มีการตกค้างของข้อมูลก่อนส่ง และสามารถลดขนาดหน่วยความจำที่รองรับข้อมูลส่วนนี้ได้ ซึ่งเป็นการแก้ปัญหาการเกิดความคับคั่งที่อุปกรณ์ต้นทาง อันเนื่องมาจากการกำหนดขนาดหน่วยความจำไม่เหมาะสมกับขนาดหน้าต่างที่ใช้ หรือเนื่องมาจากสถานะที่การเชื่อมต่ออยู่ภายใต้เกิดขีดจำกัดเนื่องจากอัตราการส่งข้อมูล

4.3 การควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับค่าหน้าต่างต่าง [11][12]

จุดประสงค์ของการควบคุมอัตราการส่งข้อมูลของ TCP แบบนี้ คือการพยายามกำจัดทั้งขีดจำกัดที่เกิดจากหน้าต่างต่าง และขีดจำกัดที่เกิดจากอัตราการส่ง (ตามที่กล่าวไว้ข้างต้น) โดยการปรับค่าหน้าต่างต่างที่ใช้ในกระบวนการควบคุมการไหลของ TCP ให้เหมาะสมกับสภาพทราฟฟิก ซึ่งอาศัยข้อมูลที่ได้จากกระบวนการควบคุมความคับคั่งของ ABR

จากหลักการพื้นฐานของ TCP ที่เรียกว่า “*Bandwidth-Delay product*” [9] ซึ่งเป็นทฤษฎีที่ใช้ในการคำนวณหาขนาดหน้าต่าง ที่จะทำให้ได้รับ Throughput สูงสุด โดยใช้สูตรการคำนวณดังนี้

$$\text{Bandwidth} - \text{Delay product}(\text{bits}) = \text{bandwidth}(\text{bit} / \text{sec}) \times \text{round trip time}(\text{sec}) \dots (4.1)$$

จากสูตรข้างต้น ค่า Bandwidth-Delay product หาได้จากนำค่าแบนด์วิดท์ที่จุดทำงานของการเชื่อมต่อ (หรือค่าอัตราการส่งข้อมูลที่ทำให้การเชื่อมต่อมีประสิทธิภาพการทำงานสูงสุด) มาคูณกับสองเท่าของเวลาที่ข้อมูลชุดหนึ่ง ๆ ใช้ในการเดินทางผ่านการเชื่อมต่อ นั้น ซึ่งเมื่อนำหลักการนี้มาดัดแปลงให้สอดคล้องกับการส่งข้อมูลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM สมการที่ใช้จึงเปลี่ยนเป็น

$$\text{Bandwidth} - \text{Delay product}(\text{bits}) = ER(\text{bit} / \text{sec}) \times \text{round trip time}(\text{sec}) \dots (4.2)$$

โดยค่า ER (เป็นค่าที่ถูกอ่านจากฟิลด์ ER ในเซลล์ RM) คือค่าอัตราการส่งข้อมูลที่โครงข่ายแนะนำให้แหล่งกำเนิดข้อมูลใช้ ค่านี้จะถูกปรับปรุงโดยสวิตช์ ATM ที่การเชื่อมต่อที่ผ่าน ซึ่งความถูกต้องของค่า ER ขึ้นอยู่กับอัลกอริทึมที่ใช้ในสวิตช์

กระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับค่าหน้าต่างต่าง อาศัยหลักการดังกล่าวในการหาขนาดหน้าต่างที่เหมาะสม นั่นคือค่าที่สามารถให้ Throughput สูงสุด

และสอดคล้องกับการทำงานร่วมกันของ TCP และบริการแบบ ABR อีกด้วย ซึ่งสามารถเขียนเป็น สูตรการคำนวณง่าย ๆ ดังนี้

$$W = r \times RTT \quad \dots (4.3)$$

โดย W คือค่าขนาดหน้าต่างที่ได้จากการแปลงค่าอัตราการส่งข้อมูล
 r คือค่าอัตราการส่งข้อมูลในชั้น ATM (ผู้เสนอใช้ค่า ER ในการคำนวณ)
 RTT คือเวลาที่ใช้ในการส่งข้อมูลไปและกลับ

จากการสังเกตความสัมพันธ์ข้างต้น อาจเรียกวิธีการนี้ว่า การแปลงค่าอัตราการส่งข้อมูลมาเป็นขนาดหน้าต่าง (rate-to-window translation)

เมื่อได้ผลการคำนวณจากสูตรข้างต้นแล้ว นำมาเปรียบเทียบกับขนาดหน้าต่างที่กำหนด โดยปลายทาง ทั้งนี้เพื่อหลีกเลี่ยงความเสียหายที่อาจเกิดจากขีดความสามารถในการทำงานของ ปลายทางจำกัด โดยการเปรียบเทียบทำได้ดังสมการนี้

$$W_{new} = \min(W, W_{rec}) \quad \dots (4.4)$$

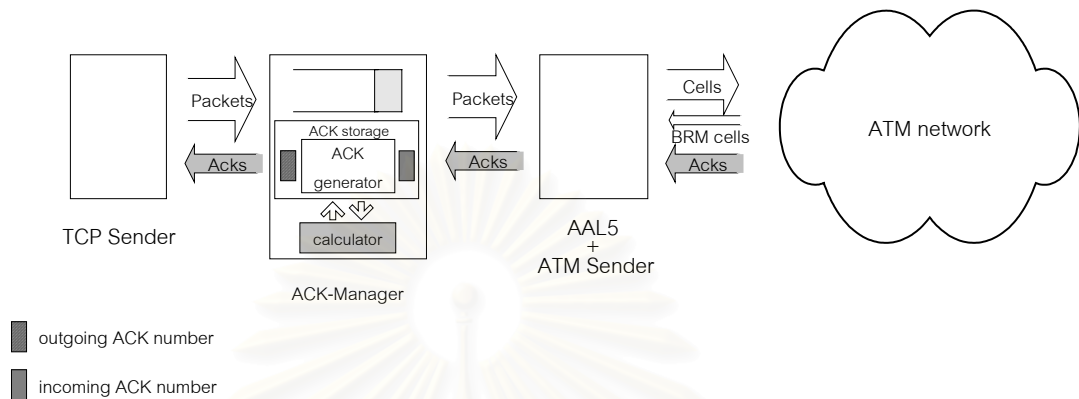
โดย W_{new} คือค่าขนาดหน้าต่างที่จะเอาไปใช้
 W คือค่าขนาดหน้าต่างที่ได้จากการคำนวณ
 W_{rec} คือค่าขนาดหน้าต่างที่ได้จากปลายทาง

ทั้งนี้การศึกษาเรื่องการควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับปรุงขนาดหน้าต่าง ที่ผ่านมาไม่ได้กล่าวถึง กระบวนการที่เกิดหลังจากเกิดความเสียหายขึ้นกับข้อมูล

4.4 การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็น เฟรม (วิธีที่เสนอ)

จากกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP บนบริการแบบ ABR ที่มีผู้เสนอมา นั้น ต่างพยายามปรับปรุงการทำงานในแง่ต่าง ๆ ไม่ว่าจะเป็นการลดขนาดหน้าต่างความจำของ อุปกรณ์ต้นทาง หรือลดปัญหาที่เกิดจากการทำงานที่ไม่สอดคล้องกันระหว่างกระบวนการควบคุม การไหลของ TCP กับการส่งข้อมูลจริง ซึ่งวิธีการที่เสนอมาแล้วนั้น สามารถปรับปรุงประสิทธิภาพ การทำงานได้ระดับหนึ่งเท่านั้น สำหรับกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การ แปลงอัตราการส่งแบบเซลล์เป็นเฟรมนี้ เป็นวิธีการที่สามารถแก้ปัญหาดังกล่าวได้ดี ทั้งนี้กระบวนการ ควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม ถูกออกแบบ ภายใต้นโยบายความคิดที่มุ่งเน้นปรับปรุงความสามารถในการควบคุมการไหลของ TCP ให้สอดคล้อง

คล้องกับความสามารถในการส่งข้อมูลจริงมากที่สุด โดยนำการแปลงอัตราการส่งข้อมูลแบบเซลล์ที่ได้จากชั้น ATM มาเป็นอัตราส่งข้อมูลแบบเฟรมในชั้น TCP มาพิจารณาเป็นหลัก โดยหลักการทำงานคร่าว ๆ ของกระบวนการนี้ สามารถพิจารณาจากรูปที่ 4.3



รูปที่ 4.3 หลักการทำงานของกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรมเป็นการปรับปรุงการส่งข้อมูลในชั้น TCP เพื่อบรรลุประสิทธิภาพการทำงานสูงสุดของระบบ โดยอาศัยอุปกรณ์เพิ่มเติมที่เรียกว่า Acknowledgement manager ควบคุมการรับส่งสัญญาณตอบกลับ ซึ่ง Acknowledgement manager จะทำการแยกสัญญาณตอบกลับที่ได้รับจากโครงข่าย และสัญญาณตอบกลับที่จะส่งต่อไปยังแหล่งกำเนิดข้อมูลในชั้น TCP ออกจากกัน โดยจะสร้างสัญญาณตอบกลับขึ้นมาใหม่และส่งออกไปในจังหวะที่เหมาะสม เพื่อเป็นการกระตุ้นให้แหล่งกำเนิดข้อมูลส่งข้อมูลที่อัตราการส่งที่แปลงมาจากอัตราการส่งแบบเซลล์ที่ใช้ในชั้น ATM กระบวนการควบคุมการไหลของ TCP แบบดั้งเดิม จะใช้ความสัมพันธ์ของการมาถึงของสัญญาณตอบกลับ เป็นตัวบ่งบอกชี้สภาพความคับคั่งของโครงข่าย เพื่อพิจารณาอัตราการส่งข้อมูลที่เหมาะสม แต่สำหรับการส่งข้อมูล TCP บนบริการ ABR ในโครงข่าย ATM นั้น ข้อมูลดังกล่าว ถูกวัดและตรวจสอบจากกระบวนการควบคุมความคับคั่งที่ทำในชั้น ATM ที่มีประสิทธิภาพการทำงานสูงอยู่แล้ว ดังนั้นจึงไม่มีความจำเป็นที่จะต้องใช้กระบวนการตรวจสอบความคับคั่งของโครงข่ายที่ทำการส่งเกิดจากจังหวะการส่งและรับสัญญาณตอบกลับอีกต่อไป

นอกจากนี้ Acknowledgement manager ยังควบคุมปริมาณการส่งของข้อมูลของแหล่งกำเนิด TCP จากการสร้างสัญญาณตอบกลับ ซึ่งเป็นการค่อย ๆ ส่งข้อมูลด้วยปริมาณที่โครงข่าย

สามารถรองรับได้จริง ด้วยเหตุนี้ทำให้ไม่มีการค้างคาของข้อมูลที่ใดที่หนึ่งในโครงข่าย และลดปัญหาการล้นของหน่วยความจำ (โดยเฉพาะที่อุปกรณ์ต้นทาง) อีกด้วย

อย่างไรก็ตามเนื่องจากสัญญาณตอบกลับไม่เพียงแต่มีความสำคัญต่อการควบคุมการไหลของ TCP เท่านั้น แต่ยังเป็นตัวแปรสำคัญในกระบวนการตรวจสอบความผิดพลาดของข้อมูลอีกด้วย สำหรับกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยการแปลงอัตราการส่งแบบเซลล์เป็นเฟรม ยังคงใช้สัญญาณตอบกลับที่ได้รับจากโครงข่ายในการพิจารณาความผิดพลาดของข้อมูลอยู่ ทั้งนี้เนื่องจากไม่มีการเปลี่ยนแปลงการทำงานของอุปกรณ์อื่น ๆ รวมทั้งอุปกรณ์ปลายทางของ TCP ด้วย นั่นคืออุปกรณ์ปลายทางจะส่งสัญญาณตอบกลับที่มีหมายเลขเดิม เมื่อตรวจสอบเจอความผิดพลาดของข้อมูล (เนื่องจากหมายเลขลำดับของสัญญาณตอบกลับคือลำดับของข้อมูลที่ตัวรับคาดว่าจะได้รับตัวต่อไป แต่เมื่อเกิดความผิดพลาดขึ้นทำให้ไม่สามารถรับข้อมูลได้ ลำดับของข้อมูลที่คาดว่าจะได้รับเป็นตัวต่อไปจึงยังคงเหมือนเดิม)

Acknowledgement manager ประกอบด้วยฟังก์ชันการทำงานย่อย 3 ส่วน คือ Calculator, Acknowledgement storage และ Acknowledgement generator รายละเอียดการทำงานของฟังก์ชันย่อยแต่ละส่วน สามารถอธิบายได้ดังต่อไปนี้

Calculator อาจเรียกว่าเป็น สมอของกระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม เพราะ Calculator ทำหน้าที่คำนวณหาอัตราการส่งข้อมูลในชั้น TCP ที่สอดคล้องกับอัตราการส่งข้อมูลที่ส่งจริง ซึ่งมีความสัมพันธ์โดยตรงกับอัตราการส่งที่ใช้ในชั้น ATM หรือข้อมูลที่ได้จากกระบวนการควบคุมความคับคั่งของ ABR Calculator จะทำงานโดยคำนวณตามสูตรต่อไปนี้

$$FR = \frac{ACR \times MTU \times (Nrm - 1)}{cell\ per\ frame \times Nrm} \dots (4.5)$$

โดย	FR (Frame Rate)	คืออัตราการส่งข้อมูลที่เทียบเป็นเฟรมต่อหน่วยเวลา
	$ACR^{[a]}$	คืออัตราการส่งข้อมูลในชั้น ATM ที่อ่านค่าจากเซลล์ RM
	MTU	คือขนาดข้อมูล TCP หนึ่งเฟรม
	Nrm	คือระยะห่างระหว่างเซลล์ RM สองตัวที่ติดกัน
	$cell\ per\ frame$	คือจำนวนเซลล์ที่เกิดจากการแบ่งเฟรมข้อมูลหนึ่งเฟรม

สำหรับการคำนวณค่า $cell\ per\ frame$ ทำโดยการหาค่าจำนวนเต็มที่น้อยที่สุดที่มากกว่าหรือเท่ากับค่าที่ได้จากการหารขนาดเฟรมข้อมูลในชั้น TCP ด้วยขนาด payload ของเซลล์ในชั้น ATM (นั่นคือ 48 ไบต์) แสดงได้ดังสมการต่อไปนี้

[a] สาเหตุที่ไม่ใช้ค่า ER นั้น เป็นเพราะค่า ER คืออัตราการส่งที่โครงข่ายแนะนำ แต่ค่า ACR คืออัตราการส่งที่ใกล้เคียงกับอัตราการส่งข้อมูลจริงมากที่สุด ดังนั้นจึงใช้ค่า ACR ในการคำนวณตามสูตรนี้

$$\text{cell per frame} = \left\lceil \frac{MTU}{48} \right\rceil \quad \dots (4.6)$$

โดย $\lceil X \rceil$ เป็นสัญลักษณ์แทนจำนวนเต็มที่น้อยที่สุดที่มากกว่าหรือเท่ากับ X

นอกจากนี้การคำนวณค่า FR ยังต้องพิจารณาความสามารถในการส่งข้อมูลของลิงก์ที่ใช้ด้วย ซึ่งคือการพิจารณาค่า FR มากที่สุดที่ได้จากการคำนวณจากค่าความจุของลิงก์นั่นเอง

$$\text{MaxFR} = \frac{\text{linkrate} \times MTU \times (Nrm - 1)}{\text{cell per frame} \times Nrm} \quad \dots (4.7)$$

โดย MaxFR คืออัตราการส่งข้อมูลสูงสุดที่เกิดจากขีดจำกัดที่เกิดจากความสามารถของลิงก์
 linkrate คือความสามารถในการส่งข้อมูลของลิงก์

Acknowledgement storage เป็นหน่วยความจำสำหรับเก็บค่าลำดับของสัญญาณตอบกลับ โดย Acknowledgement storage มีองค์ประกอบ 2 ส่วน คือส่วนของ *ACK incoming number* และส่วนของ *ACK outgoing number*

- *ACK incoming number* เก็บหมายเลขลำดับของสัญญาณตอบกลับตัวสุดท้ายที่ทำได้รับจากโครงข่าย
- *ACK outgoing number* เก็บหมายเลขลำดับของสัญญาณตอบกลับที่สร้างขึ้นเพื่อใช้ในการควบคุมการไหลของ TCP

Acknowledgement generator เป็นฟังก์ชันย่อยที่มีหน้าที่สร้างสัญญาณตอบกลับตัวใหม่และส่งไปยังแหล่งกำเนิดข้อมูลในชั้น TCP โดยใช้ผลที่ได้จากการคำนวณของ Calculator มาเป็นอัตราการส่งสัญญาณตอบกลับ เพื่อให้สัญญาณตอบกลับที่สร้างขึ้นมาใหม่ไปกระตุ้นแหล่งกำเนิดข้อมูลให้ส่งข้อมูลด้วยอัตราการส่งที่สอดคล้องกับอัตราการส่งที่ใช้ในชั้น ATM

สำหรับสัญญาณตอบกลับตัวใหม่ที่สร้างขึ้นมานี้ มีการคำนวณหมายเลขสัญญาณตอบกลับหลอก ๆ ขึ้นมา เพื่อใช้เป็นตัวกระตุ้นการส่งข้อมูล โดยการคำนวณคิดจาก

$$\text{ACK outgoing number}_{new} = \text{ACK outgoing number}_{old} + MTU \quad \dots (4.8)$$

โดย $\text{ACK outgoing number}_{new}$ คือหมายเลขสัญญาณตอบกลับที่สร้างขึ้นค่าใหม่
 $\text{ACK outgoing number}_{old}$ คือหมายเลขสัญญาณตอบกลับตัวสุดท้ายที่ส่งออกไปแล้ว
 MTU คือขนาดเฟรมข้อมูลที่ใช้ในชั้น TCP

นอกจากหมายเลขสัญญาตอบกลับที่ต้องมีการคำนวณกันใหม่แล้ว ขนาดหน้าต่างด้านรับก็มีหน้าที่สำคัญ คือเป็นตัวกำหนดปริมาณข้อมูลที่ต้นทางจะส่งต่อสัญญาตอบกลับหนึ่งตัว ทั้งนี้จึงอาศัยขนาดหน้าต่างด้านรับเป็นตัวควบคุมขนาดหน้าต่างความคับคั่งที่ต้นทาง ซึ่งค่าขนาดหน้าต่างด้านรับที่จะบันทึกลงไปในสัญญาตอบกลับตัวที่สร้างขึ้นใหม่นี้ จะใช้ค่า MTU (คือขนาดเฟรมข้อมูลที่ใช้ในชั้น TCP) เพื่อเป็นการควบคุมให้แหล่งกำเนิดข้อมูลส่งข้อมูลที่ละหนึ่งเฟรมข้อมูลต่อการรับสัญญาตอบกลับหนึ่งตัวเท่านั้น ซึ่งการทำงานแบบนี้จะทำให้แหล่งกำเนิดสามารถส่งข้อมูลในอัตราที่เท่ากับอัตราการส่งข้อมูลที่ใช้ในชั้น ATM และยังเป็นการลดขนาดหน่วยความจำของต้นทางอีกด้วย

เนื่องจากการส่งข้อมูลโดยกระบวนการควบคุมการไหลของ TCP โดยทั่วไป ใช้ขนาดหน้าต่างด้านรับเป็นตัวบ่งบอกความสามารถของอุปกรณ์ปลายทาง ดังนั้นขนาดหน้าต่างด้านรับจึงยังคงเป็นตัวแปรที่ต้องนำมาพิจารณาอยู่ สำหรับกระบวนการควบคุมการไหลของ TCP แบบนี้ ใช้ขนาดหน้าต่างด้านรับที่รับมาจากสัญญาตอบกลับ (จริง) มาใช้ในการกำหนดข้อจำกัดการส่งข้อมูล ซึ่งข้อจำกัดดังกล่าวแสดงได้ดังสมการต่อไปนี้

$$ACK\ outgoing\ number \leq ACK\ incoming\ number + W_{rec} \quad \dots (4.9)$$

นอกจากนี้ในกรณีที่มีการส่งข้อมูลแบบสองทางนั้น กระบวนการควบคุมอัตราการส่งข้อมูลของ TCP โดยการแปลงอัตราการส่งแบบเซลล์เป็นเฟรม จะใช้การสำเนาข้อมูลจากสัญญาตอบกลับที่ติดมากับเฟรมข้อมูลที่ได้รับจากโครงข่าย แล้วทำการส่งเฉพาะข้อมูลไปยังตัวรับ โดยที่ไม่มีค่าหมายเลขสัญญาตอบกลับติดไปด้วย แล้วนำข้อมูลที่ได้จากสัญญาตอบกลับนั้นมาใช้เป็นตัวแปรในการควบคุมการส่งข้อมูล ตามหลักการที่กล่าวข้างต้น

บทที่ 5

แบบจำลองการทำงาน

การทดสอบการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR จะพิจารณาจากการจำลองการทำงานภายใต้แบบจำลองของโครงข่ายลักษณะต่าง ๆ เพื่อเปรียบเทียบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ รวมถึงพิจารณาผลกระทบของพารามิเตอร์ต่าง ๆ ที่มีต่อประสิทธิภาพของระบบอีกด้วย

5.1 แบบจำลองการทำงาน (Configuration)

แบบจำลองการทำงานที่ใช้ทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR เป็นแบบจำลองที่ถูกรออกแบบ เพื่อมุ่งเน้นทดสอบประสิทธิภาพการทำงานในแง่มุมต่าง ๆ ประกอบด้วย ค่าวิสัยสามารถที่ปลายทาง (Throughput) ความเท่าเทียมกันของแหล่งกำเนิดแต่ละตัว (Fairness) การใช้งานข่ายเชื่อมโยงอย่างเต็มประสิทธิภาพ (Utilization) และปริมาณเซลล์ในแถวคอยของอุปกรณ์ต่าง ๆ ในโครงข่าย (Queue length)

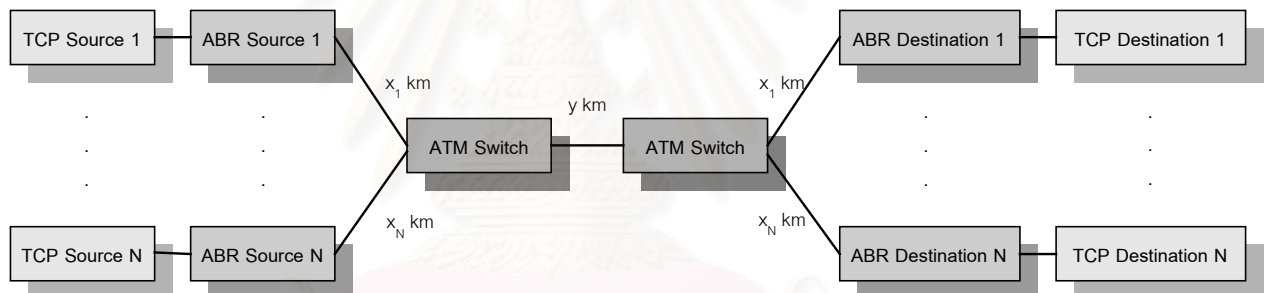
เนื่องจากปัญหาของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR คือการทำงานที่ไม่สอดคล้องกันของกระบวนการควบคุมการไหลของ TCP และกระบวนการควบคุมความคับคั่งของ ABR ดังนั้นการทดสอบการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR จึงไม่มีความจำเป็นต้องทดสอบภายใต้แบบจำลองที่หลากหลาย และมีความซับซ้อนมาก ๆ ทั้งนี้เพราะกระบวนการควบคุมการไหลของ TCP เป็นกระบวนการที่ทำงานซ้อนบนกระบวนการควบคุมความคับคั่งของ ABR ทำให้การทำงานภายใต้แบบจำลองรูปแบบต่าง ๆ มีผลต่อกระบวนการควบคุมความคับคั่งของ ABR เท่านั้น และเนื่องจากการปรับปรุงกระบวนการควบคุมการไหลของ TCP จะใช้เพียงข้อมูลที่ได้จากกระบวนการควบคุมความคับคั่งของ ABR (มีการวิเคราะห์เหตุผลไว้แล้วในบทที่ 4) ซึ่งนั่นคือความซับซ้อนของโครงข่ายแทบไม่มีความสำคัญ สิ่งสำคัญกลับขึ้นอยู่กับผลการดำเนินงานของกระบวนการควบคุมความคับคั่งเท่านั้น อย่างไรก็ตาม นอกจากกระบวนการควบคุมความคับคั่งของ ABR มีผลต่อการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR แล้ว พารามิเตอร์อื่น ๆ ของโครงข่ายก็มีผลต่อประสิทธิภาพการทำงานไม่มากนัก ดังนั้นในการพิจารณาและวิเคราะห์ประสิทธิภาพของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR จึงใช้เพียงแบบจำลองง่าย ๆ โดยเน้นไปที่การพิจารณาผลกระทบจากพารามิเตอร์ต่าง ๆ ในระบบ ที่มีต่อประสิทธิภาพการทำงานของโครงข่าย

อย่างไรก็ตาม ในการทดสอบประสิทธิภาพการทำงานจะทำการเปรียบเทียบกับโครงข่าย ATM ที่ไม่ได้ทำงานภายใต้ TCP ด้วย ทั้งนี้เพื่อแสดงคุณลักษณะของข้อมูลที่ปลายทางจะได้รับเมื่อทำงานภายใต้กระบวนการควบคุมความคับคั่งแบบ ABR อย่างเดียว โดยพิจารณาว่าร่วมกับแหล่งกำเนิดแบบส่งข้อมูลต่อเนื่อง ซึ่งกรณีดังกล่าวนี้ จะถูกนำมาใช้เป็นกรณีอ้างอิง เพราะถือได้ว่าเป็นกรณี que แสดงถึงประสิทธิภาพการทำงานสูงสุดของโครงข่ายลักษณะนั้น ๆ (เพราะเป็นการตัดปัญหา ที่เกิดจากการขาดข้อมูลในการส่ง)

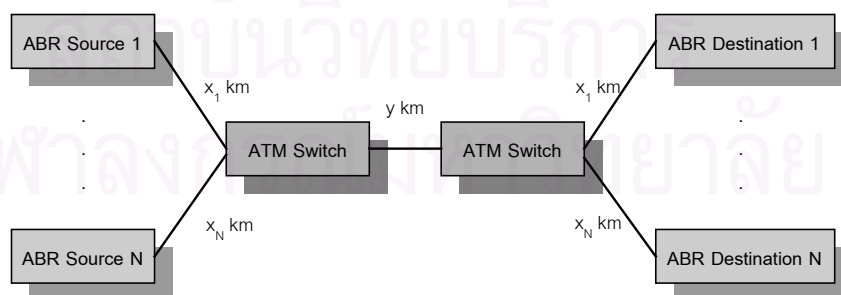
แบบจำลองการทำงานที่ใช้ทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR มีดังต่อไปนี้

5.1.1 Peer to peer configuration

แบบจำลองแบบ Peer to peer เป็นแบบจำลองการทำงานที่เชื่อมโยงระหว่างแหล่งกำเนิด N ตัวผ่านสวิตช์ 2 ตัวไปยังปลายทาง N ตัว แบ่งออกเป็น 2 แบบย่อย ๆ คือ



(ก) แบบจำลองชนิด Peer to peer ภายใต้การทำงานของ TCP
(Peer to peer configuration with TCP)

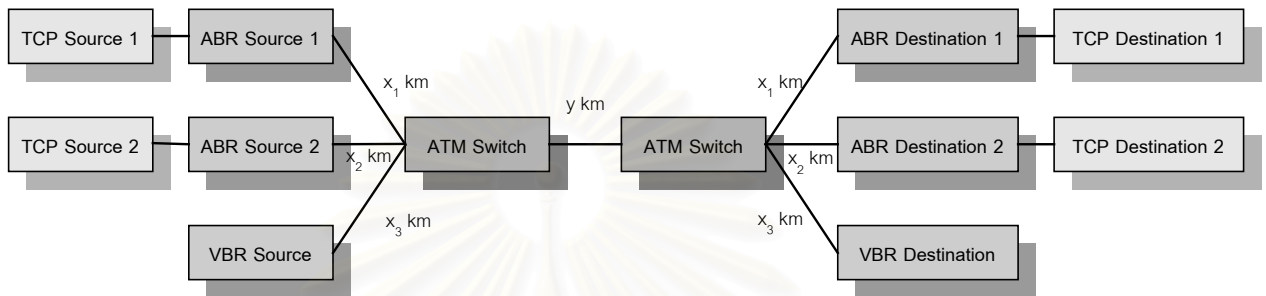


(ข) แบบจำลองชนิด Peer to peer ที่ไม่พิจารณาการทำงานในชั้น TCP
(Peer to peer configuration without TCP)

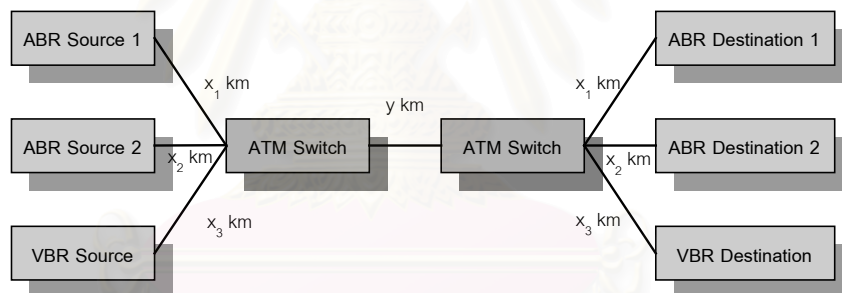
รูปที่ 5.1 แบบจำลองชนิด Peer to peer

5.1.2 VBR & ABR configuration

แบบจำลองแบบ VBR & ABR เป็นแบบจำลองที่เชื่อมโยงระหว่างแหล่งกำเนิดแบบ ABR 2 ตัว และแหล่งกำเนิดแบบ VBR 1 ตัว ส่งผ่านสวิตช์ 2 ตัว ไปยังปลายทางที่รับบริการแบบ ABR และ VBR ตามลำดับ



(ก) แบบจำลองชนิด VBR & ABR ภายใต้การทำงานของ TCP
(VBR & ABR configuration with TCP)



(ข) แบบจำลองชนิด VBR & ABR ที่ไม่พิจารณาการทำงานในชั้น TCP
(VBR & ABR configuration without TCP)

รูปที่ 5.2 แบบจำลองชนิด VBR & ABR

จากรูปที่ 5.2 แหล่งกำเนิดแบบ VBR เป็นแหล่งกำเนิดที่มีความเป็นอิสระในการส่งข้อมูลสู่โครงข่าย ซึ่งลักษณะการส่งข้อมูลจะเปลี่ยนแปลงตามเวลา ในขณะที่แหล่งกำเนิดแบบ ABR จะใช้แบนด์วิดท์ที่เหลือ (เพราะทราฟฟิกแบบ VBR ที่มีความสำคัญกว่าทราฟฟิกแบบ ABR) ทำให้แหล่งกำเนิดแบบ ABR จำเป็นต้องปรับอัตราการส่งข้อมูลให้สอดคล้องกับการเปลี่ยนแปลงของแหล่งกำเนิดแบบ VBR ดังนั้นแบบจำลองการทำงานแบบนี้ จึงเป็นการทดสอบความสามารถในการทำงานของกระบวนการควบคุมการไหลของ TCP ภายใต้การเปลี่ยนแปลงความสามารถในการส่ง

ข้อมูลของโพรโตคอลระดับต่ำกว่า กล่าวอีกนัยหนึ่งคือ แบบจำลองการทำงานแบบนี้เป็นการทดสอบผลกระทบของการเปลี่ยนแปลงอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ต่อกระบวนการควบคุมการไหลของ TCP นั้นเอง

5.2 ข้อกำหนดที่ใช้ในการจำลองแบบ

5.2.1 แหล่งกำเนิดข้อมูล และคุณลักษณะของแหล่งกำเนิดข้อมูล (Source and Source characteristic)

การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP บนโครงข่าย ATM ตามแบบจำลองข้างต้นประกอบด้วยแหล่งกำเนิดข้อมูล TCP และแหล่งกำเนิดข้อมูล ATM อีก 2 ชนิดคือ แหล่งกำเนิดข้อมูลสำหรับบริการแบบ ABR และ VBR ซึ่งแหล่งกำเนิดข้อมูลแต่ละแบบมีลักษณะต่างกัน ไม่ว่าจะเป็นรูปแบบในการส่งข้อมูล หรือพารามิเตอร์ที่ใช้ สามารถอธิบายดังนี้

5.2.1.1 แหล่งกำเนิดข้อมูล TCP

แหล่งกำเนิดข้อมูล TCP ที่ใช้ในงานวิจัยนี้ เป็นแหล่งกำเนิดที่มีข้อมูลให้ส่งเสมอ ซึ่งอัตราการส่งจะเป็นเท่าไรนั้นขึ้นอยู่กับกลไกการทำงานของโครงข่าย โดยมีค่า MTU เป็นตัวบอกขนาดของ TCP เซกเมนต์

5.2.1.2 แหล่งกำเนิดของบริการแบบ ABR

แหล่งกำเนิดของบริการแบบ ABR บนโครงข่าย ATM เป็นแหล่งกำเนิดที่มีการส่งข้อมูลสม่ำเสมอ ตามความสามารถของโครงข่าย ที่ตรวจสอบจากกระบวนการควบคุมความคับคั่ง พารามิเตอร์ที่ใช้ในการกำหนดคุณลักษณะต่าง ๆ ของแหล่งกำเนิดแบบนี้ประกอบด้วย

1. อัตราการส่งข้อมูลสูงสุด (PCR) เป็นอัตราการส่งสูงสุดที่แหล่งกำเนิดสามารถส่งได้
2. อัตราการส่งข้อมูลต่ำสุด (MCR) เป็นอัตราการส่งต่ำสุดที่แหล่งกำเนิดใช้ในการส่งข้อมูล
3. อัตราการส่งข้อมูลเริ่มต้น (ICR) เป็นอัตราการส่งที่แหล่งกำเนิดใช้เมื่อการเชื่อมต่อเริ่มทำงาน และจะใช้อีกครั้งในกรณีที่แหล่งกำเนิดไม่มีการส่งข้อมูลนาน ๆ และกลับมาส่งข้อมูลอีกครั้ง

4. แฟกเตอร์การเพิ่มอัตราการส่ง (RIF) เป็นอัตราส่วนที่ใช้ในการเพิ่มอัตราการส่งข้อมูลของแหล่งกำเนิด โดยแหล่งกำเนิดจะเพิ่มอัตราการส่งข้อมูลเท่ากับ $RIF \times PCR$ ในการเพิ่มแต่ละครั้ง แต่อัตราการส่งของแหล่งกำเนิดหลังการเพิ่มจะต้องไม่มากกว่า PCR
5. แฟกเตอร์การลดอัตราการส่ง (RDF) เป็นอัตราส่วนที่ใช้ในการลดอัตราการส่งข้อมูลของแหล่งกำเนิด โดยแหล่งกำเนิดจะลดอัตราการส่งข้อมูลเท่ากับ $RDF \times ACR$ ในการลดแต่ละครั้ง แต่อัตราการส่งของแหล่งกำเนิดหลังการลดจะต้องไม่น้อยไปกว่า MCR
6. จำนวนเซลล์ข้อมูลต่อจำนวนเซลล์ RM (Nrm) แหล่งกำเนิดจะส่งเซลล์ RM 1 เซลล์แทรกไปทุก ๆ $Nrm-1$ เซลล์ของเซลล์ข้อมูล (โดยมากถูกกำหนดให้ $Nrm = 32$)

5.2.1.3 แหล่งกำเนิดของบริการแบบ VBR

แหล่งกำเนิดของบริการแบบ VBR บนโครงข่าย ATM เป็นแหล่งกำเนิดที่มีการส่งข้อมูลที่เปลี่ยนแปลงตามเวลา ซึ่งลักษณะข้อมูลที่สร้างเป็นแบบเบริสต์ดี สำหรับการจำลองแหล่งกำเนิดแบบนี้ ทำโดยกำหนดสถานะการส่งข้อมูล 2 สถานะ คือส่ง และหยุดส่ง ซึ่งช่วงที่มีการส่งข้อมูล จะส่งที่อัตราการส่งค่าหนึ่ง ส่วนในช่วงสถานะหยุดส่งนั้นจะไม่มี การส่งข้อมูลเลย พารามิเตอร์ของแหล่งกำเนิดแบบนี้ ประกอบด้วย

1. ช่วงคาบเวลาที่มีการส่งข้อมูล (*Burst time*) เป็นช่วงเวลาที่แหล่งกำเนิดมีการส่งข้อมูลไปยังปลายทาง
2. ช่วงคาบเวลาที่หยุดส่งข้อมูล (*Silent time*) เป็นช่วงเวลาที่แหล่งกำเนิดหยุดส่งข้อมูล
3. อัตราการส่งข้อมูลของแหล่งกำเนิดเมื่ออยู่ในช่วงคาบการส่งข้อมูล (*Burst rate*)

5.2.2 ATM สวิตช์ (ATM Switch)

ATM สวิตช์ที่นำมาใช้ในการทดสอบ จำเป็นต้องมีอัลกอริทึม ที่ใช้สำหรับกระบวนการควบคุมความคับคั่งของบริการแบบ ABR ซึ่งหลักการทำงาน ร่วมทั้งความหมายและหน้าที่ของพารามิเตอร์ต่าง ๆ ของอัลกอริทึมแต่ละชนิด มีรายละเอียดอยู่ในบทที่ 2 โดยอัลกอริทึมที่จะใช้การทดสอบ มีดังต่อไปนี้

- อัลกอริทึม ERICA
- อัลกอริทึม ERICA+
- อัลกอริทึม FMMRA
- อัลกอริทึม E-FMMRA

5.2.3 ปลายทาง (Destination)

อุปกรณ์ปลายทางที่ใช้ในการทดสอบ เป็นอุปกรณ์ความเร็วสูงที่สามารถรองรับข้อมูลได้อย่างมีประสิทธิภาพ สำหรับอุปกรณ์ปลายทางของ TCP จำเป็นต้องกำหนดค่าขนาดหน้าต่างมากสุดของปลายทาง (TCP maximum receiver window size) เพื่อกำหนดความสามารถของอุปกรณ์ปลายทาง

5.2.4 สายส่ง (Lines)

สำหรับสายส่งที่ใช้ในแบบจำลองนี้จะเป็นสายส่งชนิดที่มีการประวิงเวลาขึ้นกับระยะทาง โดยสายส่งที่ใช้นี้มีแบนด์วิดท์เท่ากับ 155.52 Mbps เมื่อหัก Header ของ SONET ออกจะเหลือ 149.76 Mbps

5.2.5 ข้อกำหนดอื่น ๆ

- ข้อมูลจะส่งไปเพียงทิศทางเดียว (Unidirectional) จะไม่มีข้อมูลใด ๆ ส่งในทิศทางย้อนกลับ ยกเว้น เซลล์ RM และสัญญาณตอบกลับของ TCP (Acknowledgement)
- ไม่พิจารณาในส่วนการ Setup โคร่งข่าย โดยถือว่าโคร่งข่ายมีการ Setup เรียบร้อยแล้ว พร้อมทั้งจะส่งข้อมูลได้ทันที
- ในการจำลองการทำงานทั้งหมดนี้ ไม่มีการพิจารณากรณีที่มีการส่งข้อมูลเกิดความเสียหาย ซึ่งมีผลให้กระบวนการส่งข้อมูลซ้ำของ TCP ทำงาน อย่างไรก็ตามจะมีการวิเคราะห์ผลกระทบที่เกิดจากความเสียหายในการส่งข้อมูลต่อกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR แบบต่าง ๆ ในภาคผนวก ข.

บทที่ 6

ผลการจำลองการทำงานและการวิเคราะห์

สำหรับบทนี้เป็นการจำลองแบบกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยการจำลองจะเสนอออกมาในรูปแบบของการเปรียบเทียบ ประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ รวมทั้งเปรียบเทียบกับผลที่เกิดจากการทำงานของกระบวนการควบคุมความคับคั่งเพียงอย่างเดียว (ปราศจาก TCP) ด้วย โดยมุ่งเน้นวิเคราะห์ประสิทธิภาพการทำงานในแง่ของค่าวิสัยสามารถที่ปลายทาง (Throughput) ความเท่าเทียมกันในการใช้ทรัพยากรของแหล่งกำเนิดแต่ละตัว (Fairness) ประสิทธิภาพการใช้งานลิงก์ (Utilization) และปริมาณเซลล์ในแถวคอยของอุปกรณ์ต่าง ๆ ในโครงข่าย (Queue length)

6.1 การจำลองการทำงานบนอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ

เนื่องจากกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR เป็นการทำงานของกระบวนการควบคุมการไหลของ TCP ที่ซ้อนบนกระบวนการควบคุมความคับคั่งของ ABR ดังนั้น ผลการทำงานของแต่ละกระบวนการ จึงมีบทบาทต่อประสิทธิภาพการทำงานโดยรวมไม่น้อย ทำให้การทดสอบประสิทธิภาพการทำงาน จำเป็นต้องพิจารณาถึงผลที่เกิดจากกระบวนการควบคุมความคับคั่งเพียงอย่างเดียวก่อน แล้วจึงพิจารณาผลที่เกิดจากกระบวนการไหลของ TCP

สำหรับการจำลองการทำงานภายใต้การเปลี่ยนแปลงอัลกอริทึมที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR เป็นการศึกษาและวิเคราะห์ผลของอัลกอริทึมชนิดต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR และเพื่อสรรหาอัลกอริทึมที่เหมาะสมในการนำไปใช้วิเคราะห์ผลจากปัจจัยอื่น ๆ ต่อไป โดยแบบจำลองที่เหมาะสมสำหรับการทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมความคับคั่งของ ABR ที่เป็นที่ยอมรับมีด้วยกันหลายแบบ หนึ่งในแบบดังกล่าว คือแบบจำลองแบบ VBR & ABR ซึ่งเป็นแบบจำลองที่มีแหล่งกำเนิดแบบ VBR อยู่ในโครงข่ายด้วย เป็นผลให้ทรัพยากรในส่วนของ ABR มีการเปลี่ยนแปลงตลอดเวลา ทำให้สามารถวิเคราะห์ประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP ภายใต้โครงข่ายที่มีแหล่งกำเนิดที่มีการเปลี่ยนแปลงอีกด้วย

6.1.1 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยไม่ผ่าน TCP

การจำลองการทำงานชุดนี้ เป็นการศึกษาและวิเคราะห์ผลที่เกิดจากกระบวนการควบคุมความคับคั่งของ ABR ชนิดต่าง ๆ โดยในการจำลองจะพิจารณากรณีที่แหล่งกำเนิดมีข้อมูลมากเพียงพอที่จะส่งข้อมูลอย่างต่อเนื่อง และแหล่งกำเนิดดังกล่าวจะส่งข้อมูลผ่านมายังอุปกรณ์ในชั้น ATM โดยตรง นั่นคือไม่ส่งผ่านกระบวนการในชั้น TCP สำหรับการจำลองจะทดสอบภายใต้แบบจำลอง 2 ชุด ซึ่งแบบจำลองและพารามิเตอร์ที่ใช้ในการทดสอบ สามารถแจกแจงได้ดังต่อไปนี้

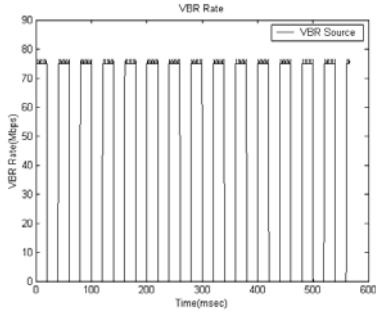
แบบจำลองชุดที่ 1

<u>แบบจำลอง</u>	ตามรูปที่ 5.2 (ข) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ VBR 1 ตัว และแหล่งกำเนิดข้อมูลแบบ ABR อีก 2 ตัว
<u>แหล่งกำเนิดข้อมูล VBR</u>	ช่วงคาบเวลาที่หยุดส่งข้อมูล = ช่วงคาบเวลาที่มีการส่งข้อมูล = 20 ms อัตราการส่งข้อมูลเมื่ออยู่ในช่วงคาบการส่งข้อมูล = 75 Mbps (50% ของความจุลิงก์)
(ลักษณะกราฟฟิกของแหล่งกำเนิดข้อมูลแบบ VBR ที่สร้างขึ้น แสดงดังรูปที่ 6.1 (ก))	
<u>แหล่งกำเนิดข้อมูล ABR</u>	PCR = 149.76 Mbps; MCR = 0 Mbps; ICR = 0.5 Mbps RIF = 1/16 [#] RDF = 1/16 [#] [#] เนื่องจากการตั้งค่า RIF และ RDF ที่ใช้ในการจำลองแบบในงานวิจัยต่าง ๆ ไม่มีมาตรฐานที่แน่นอน แต่สำหรับงานวิจัยนี้ ค่า RIF และ RDF ที่เลือกใช้มีค่าเท่ากับ 1/16 ซึ่งเป็นค่าที่มีผลให้การปรับอัตราการส่งข้อมูลที่ไม่เร็วหรือช้าจนเกินไป
<u>ความยาวของสายส่ง</u>	$x_1 = x_2 = x_3 = 10 \text{ km}$; $y = 100 \text{ km}$ (จากรูป 5.2 (ข))

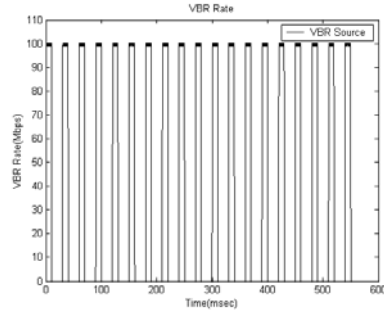
แบบจำลองชุดที่ 2

แบบจำลองชุดที่ 2 มีองค์ประกอบ และพารามิเตอร์ต่าง ๆ เหมือนแบบจำลองชุดที่ 1 ยกเว้นแหล่งกำเนิดข้อมูล VBR ที่มีการตั้งค่าให้มีลักษณะการแกว่งตัวของกราฟฟิกรุนแรงกว่าพิจารณารายละเอียดได้ดังนี้

<u>แหล่งกำเนิดข้อมูล VBR</u>	ช่วงคาบเวลาที่หยุดส่งข้อมูล = 20 ms ช่วงคาบเวลาที่มีการส่งข้อมูล = 10 ms อัตราการส่งข้อมูลเมื่ออยู่ในช่วงคาบการส่งข้อมูล = 100 Mbps (66.67% ของความจุลิงก์)
(ลักษณะกราฟฟิกของแหล่งกำเนิดข้อมูลแบบ VBR ที่สร้างขึ้น แสดงดังรูปที่ 6.1 (ข))	



(ก) อัตราการส่งข้อมูลที่เกิดจากแบบจำลองชุดที่ 1
 (คาบเวลาส่งข้อมูล = 20 ms
 คาบเวลาหยุดส่งข้อมูล = 20 ms
 อัตราการส่งข้อมูลเมื่ออยู่ในคาบการส่งข้อมูล = 75 Mbps)



(ข) อัตราการส่งข้อมูลที่เกิดจากแบบจำลองชุดที่ 2
 (คาบเวลาส่งข้อมูล = 10 ms
 คาบเวลาหยุดส่งข้อมูล = 20 ms
 อัตราการส่งข้อมูลเมื่ออยู่ในคาบการส่งข้อมูล = 100 Mbps)

รูปที่ 6.1 ผลการจำลองการทำงานของแหล่งกำเนิด VBR ที่ใช้ในแบบจำลองชุดที่ 1 และ 2

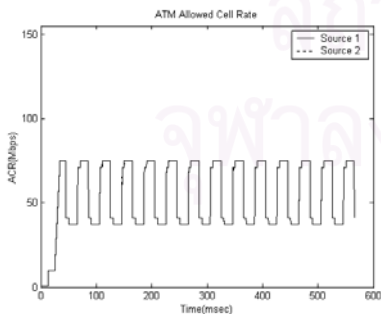
สำหรับอัลกอริทึมของกระบวนการควบคุมความคับคั่งของ ABR ที่นำมาใช้ในการจำลอง มีอยู่ 4 ชนิด คือ ERICA, ERICA+, FMMRA และ EFMMRA โดยจะพิจารณาทีละประเภทดังนี้

6.1.1.1 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้ อัลกอริทึม ERICA

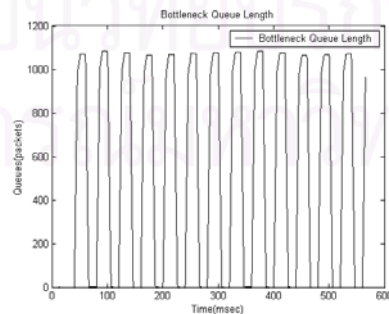
อัลกอริทึม ERICA ที่ใช้ในแบบจำลองนี้ มีการตั้งค่าดังต่อไปนี้

อัลกอริทึม ERICA ที่สวิตช์ ATM AI = 5 ms[#] ; Target Utilization = 1.0
[#] เป็นค่า AI ที่ผู้คิดค้น ERICA แนะนำให้ใช้ [5]

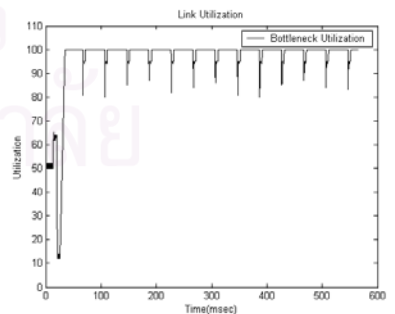
- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 1



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด แบบ ABR



(ข) ปริมาณแควคอยในสวิตช์ ATM



(ค) ประสิทธิภาพการใช้งานลิงก์

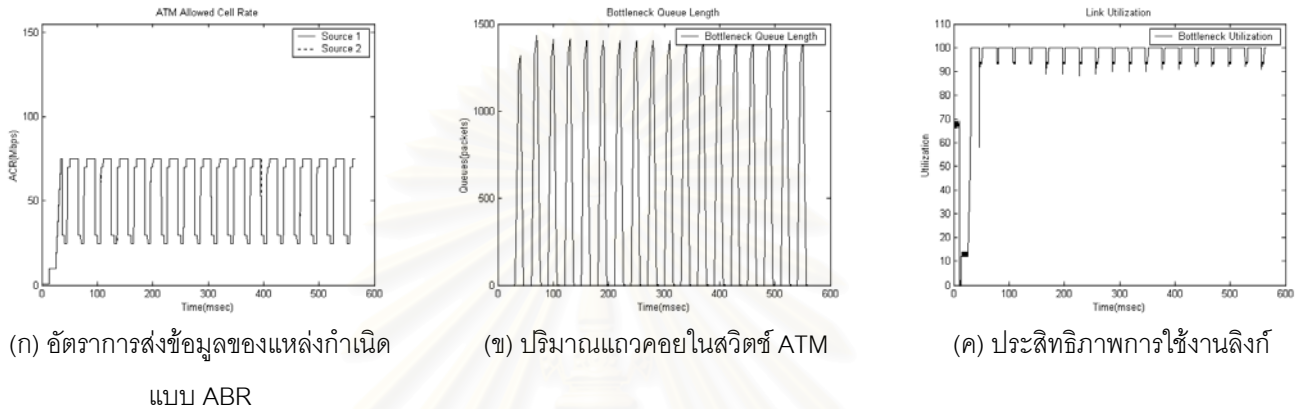
รูปที่ 6.2 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้อัลกอริทึม ERICA ภายใต้แบบจำลองชุดที่ 1

จากผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้อัลกอริทึม ERICA ภายใต้แบบจำลองที่ 1 จะเห็นว่าในช่วงแรกของการเชื่อมต่อ แหล่งกำเนิดข้อมูลแบบ VBR จะเริ่มส่งข้อมูลที่ 75 Mbps ในขณะที่แหล่งกำเนิดข้อมูลแบบ ABR จะเริ่มส่งข้อมูลที่ 0.5 Mbps (ตามค่า ICR) และเมื่อครบรอบเวลา AI (ในที่นี้เท่ากับ 5 ms) อัลกอริทึม ERICA ก็จะสามารถคำนวณค่าต่าง ๆ โดยเริ่มจากพิจารณาค่าแบนด์วิดท์ที่เหลือสำหรับบริการแบบ ABR ซึ่งเท่ากับ $149.76 - 75 = 74.76$ Mbps ดังนั้นค่า Fairshare ที่คำนวณได้จึงเท่ากับ 37.38 Mbps (มาจาก 74.76 หารด้วย 2 เพราะมีการเชื่อมต่อที่มีทราฟฟิกแบบ ABR 2 ชุด) และเป็นค่าที่ ERICA แนะนำให้แหล่งกำเนิดใช้ด้วย (นั่นคือค่า $ER = 37.38$ Mbps) ดังนั้นเมื่อแหล่งกำเนิด ABR ได้รับเซลล์ RM ย้อนกลับแล้วแหล่งกำเนิดก็จะพยายามปรับอัตราการส่งให้ได้เท่ากับค่าที่โครงข่ายแนะนำ ซึ่งในการปรับขึ้นของค่า ACR จะปรับเป็นขั้น ๆ สอดคล้องกับค่า RIF ซึ่งในการจำลองนี้ตั้งให้มีค่าเท่ากับ $1/16$ เป็นผลให้การปรับค่า ACR ในแต่ละครั้งมีค่าประมาณ 9 Mbps (คำนวณจาก $RIF \times PCR$) ทั้งนี้เมื่อพิจารณาจากรูปที่ 6.2 (ก) จะเห็นว่าแหล่งกำเนิด ABR ไม่ได้หยุดเพิ่มอัตราการส่งที่ 37.38 Mbps ตามผลการคำนวณข้างต้น ทั้งนี้เพราะเมื่อผ่าน 20 ms แรกไป แหล่งกำเนิดแบบ VBR จะหยุดส่งข้อมูลข้อมูล มีผลทำให้เมื่อถึงรอบการวัดของ ERICA (ERICA จะทำการวัดและคำนวณค่าต่าง ๆ ทุก ๆ AI) ค่า ER ใหม่ที่คำนวณได้จะมีค่าเท่ากับ 74.88 Mbps ทั้งนี้เพราะแบนด์วิดท์ที่เหลือสำหรับบริการแบบ ABR เท่ากับ 149.76 Mbps ซึ่งเท่ากับความเร็วของลิงก์ ดังนั้นจึงทำให้แหล่งกำเนิดข้อมูลแบบ ABR ปรับอัตราการส่งข้อมูลจนมีค่าประมาณ 75 Mbps ดังรูป 6.2 (ก) อย่างไรก็ตามจะเห็นว่าการปรับอัตราการส่งครั้งนี้ ต้องใช้เวลาเกือบ 40 ms กว่าที่แหล่งกำเนิดจะสามารถส่งข้อมูลที่อัตราที่โครงข่ายแนะนำ ซึ่งเป็นผลให้ช่วงเวลาดังกล่าวข่ายเชื่อมโยงไม่ได้ถูกใช้งานเต็มพิกัด (นั่นคือประมาณ 50% ในช่วงเริ่มต้นและปรับขึ้นมาที่ 65% ซึ่งสอดคล้องกับการปรับตัวเพิ่มขึ้นเล็กน้อยของ ACR แต่การใช้งานข่ายเชื่อมโยงตกลงมาอีกครั้ง คราวนี้เป็นผลของการที่แหล่งกำเนิดข้อมูลแบบ VBR หยุดส่ง อย่างไรก็ตามการใช้งานลิงก์มีค่าครบ 100% เมื่อแหล่งกำเนิด ABR สามารถส่งข้อมูลตามที่โครงข่ายแนะนำได้สำเร็จ พิจารณาการใช้งานลิงก์จากรูปที่ 6.2 (ค) ด้วยเหตุนี้เองทำให้ไม่มีการสะสมข้อมูลในแถวคอยของสวิตช์ ATM ในช่วงแรกนี้ ดังรูป 6.2 (ข)

เมื่อผ่าน 40 ms แรกไปแล้วแหล่งกำเนิดแบบ VBR จะส่งข้อมูลอีกครั้ง แต่เนื่องจากการตอบสนองที่ไม่ทันท่วงทีของแหล่งกำเนิดแบบ ABR (นั่นคือแหล่งกำเนิดแบบ ABR ยังคงส่งข้อมูลที่อัตราประมาณ 75 Mbps ทั้ง ๆ ที่ควรส่งข้อมูลที่อัตรา 37.38 Mbps เมื่อแหล่งกำเนิดแบบ VBR มีการส่งข้อมูล) จึงก่อให้เกิดการส่งข้อมูลที่เกินความสามารถในการรองรับของลิงก์ ทำให้สวิตช์ต้องรองรับข้อมูลส่วนที่ส่งเกินนี้ โดยการเก็บไว้ในแถวคอยที่สวิตช์ แสดงดังรูปที่ 6.2 (ข) และข้อมูลในแถวคอยตรงนี้เองจะถูกระบายออกในช่วงเวลาสั้น ๆ ระหว่างที่แหล่งกำเนิดข้อมูลแบบ VBR หยุดส่งไปแล้ว แต่แหล่งกำเนิดแบบ ABR ยังไม่เพิ่มอัตราการส่ง อย่างไรก็ตามข้อมูลในแถวคอยที่

สะสมดังกล่าวมีปริมาณไม่เพียงพอที่จะเติมเต็มลิงก์ตลอดช่วงเวลาที่เหลื่อมกันนี้ จึงทำให้การใช้งานลิงก์บางช่วงไม่เต็ม 100% ดังรูปที่ 6.2 (ค) ซึ่งจะเห็นว่าการสะสมข้อมูลในแถวคอย และการใช้งานลิงก์ไม่เต็มที่เกิดขึ้นเป็นจังหวะตามการเปลี่ยนแปลงของแหล่งกำเนิดข้อมูลแบบ VBR

- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 2



รูปที่ 6.3 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้อัลกอริทึม ERICA ภายใต้แบบจำลองชุดที่ 2

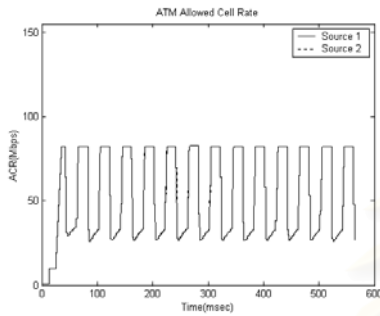
ผลการจำลองของแบบจำลองชุดที่ 2 มีลักษณะคล้ายกับผลที่เกิดจากแบบจำลองชุดที่ 1 มาก เพียงแต่ผลที่ได้จากการจำลองแบบจำลองชุดที่ 2 นี้ ทั้งค่าอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR (ค่า ACR) และระดับแถวคอยในสวิตช์ จะแกว่งตัวด้วยความถี่ที่มากกว่า (เป็นผลโดยตรงจากแหล่งกำเนิด VBR ในแบบจำลองแบบที่ 2 นี้ที่มีความถี่ในการแกว่งตัวมากกว่านั่นเอง) โดยการแกว่งตัวของค่า ACR นี้จะแกว่งอยู่ระหว่างค่าประมาณ 75 Mbps และ 25 Mbps ซึ่งเป็นค่า Fairshare ที่คำนวณได้ในกรณีนี้ สำหรับระดับแถวคอยที่สวิตช์ ATM จะมีค่ามากกว่าในกรณีที่เกิดจากแบบจำลองชุดที่ 1 เพียงเล็กน้อย (พิจารณาดังรูปที่ 6.3 (ข) เปรียบเทียบกับรูป 6.2 (ข)) ทั้งนี้เพราะแหล่งกำเนิดแบบ VBR มีอัตราการส่งข้อมูลในช่วงคาบการส่งมากกว่า นอกจากนี้ผลจากการแกว่งของอัตราการส่งข้อมูลของทั้งแหล่งกำเนิดแบบ VBR และ ABR ที่ถี่ขึ้น ทำให้ตลอดการเชื่อมต่อลิงก์ถูกใช้อย่างไม่เต็มประสิทธิภาพบ่อยขึ้นเช่นกัน (ดังรูป 6.3 (ค))

6.1.1.2 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้ อัลกอริทึม ERICA+

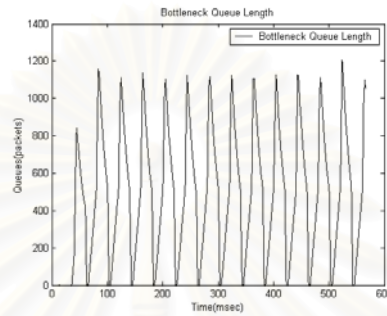
อัลกอริทึม ERICA+ ที่ใช้ในแบบจำลองนี้ มีการตั้งค่าดังต่อไปนี้

อัลกอริทึม ERICA+ ที่สวิตช์ ATM AI = 5 ms; Target Utilization = 1.0 (เหมือนกับที่ใช้กับอัลกอริทึม ERICA)
 $a = 1.15$; $b = 1.05$; $T_0 = 1.5$; $QDLF = 0.5^{\#}$
 $\#$ สำหรับพารามิเตอร์ที่ใช้กับกระบวนการควบคุมระดับแควคอยนี้ เป็นค่าที่ผู้วิจัย ERICA+ แนะนำให้ใช้ [5]

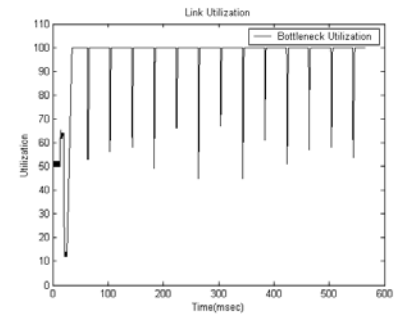
- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 1



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด
แบบ ABR



(ข) ปริมาณแควคอยในสวิตช์ ATM



(ค) ประสิทธิภาพการใช้งานลิงก์

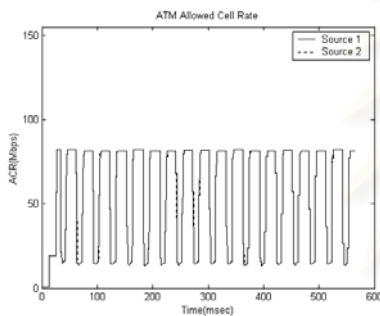
รูปที่ 6.4 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR

โดยใช้อัลกอริทึม ERICA+ ภายใต้แบบจำลองชุดที่ 1

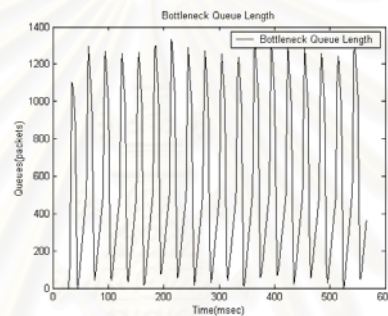
เนื่องจากอัลกอริทึม ERICA+ เป็นอัลกอริทึมที่พัฒนามาจากอัลกอริทึม ERICA ดังนั้นผลการจำลองของอัลกอริทึม ERICA+ จึงคล้ายกับ ERICA มาก โดยเฉพาะอย่างยิ่งจังหวะการปรับตัวขึ้นลงของค่า ACR แต่เนื่องจากอัลกอริทึม ERICA+ มีกลไกการควบคุมระดับแควคอย จึงทำให้ผลการคำนวณค่า ER ที่ได้มีความแตกต่างกัน นั่นคือเมื่อแหล่งกำเนิดแบบ VBR หยุดส่ง ERICA+ จะปรับอัตราการส่งให้มากกว่าค่า Fairshare (ในกรณีนี้คือมากกว่า 74.88 Mbps) ทั้งนี้เพราะกลไกการควบคุมระดับแควคอย จะพยายามควบคุมปริมาณข้อมูลในแควคอยให้อยู่ในระดับหนึ่งตลอดเวลา เพื่อพยายามหลีกเลี่ยงภาวะขาดแคลนข้อมูล ซึ่งมีผลให้การใช้งานลิงก์ไม่เต็มประสิทธิภาพ นอกจากนี้เมื่อแหล่งกำเนิดข้อมูลแบบ VBR มีการส่งข้อมูล ภาวะการลดอัตราการส่งที่ไม่ทันท่วงทีของ ABR ทำให้ระดับแควคอยเพิ่มขึ้นอย่างรวดเร็ว (ดังรูป 6.4 (ข)) และจากผลของระดับแควคอยที่มากเกินไปเอง (เกินกว่าที่ ERICA+ ต้องการ) จึงทำให้ค่า ER ที่ ERICA+ คำนวณได้มีค่าน้อยกว่า Fairshare เพื่อระบายข้อมูลในแควคอยให้อยู่ในระดับที่ ERICA+ คิดว่าเหมาะสม และนั่นทำให้ระดับแควคอยค่อย ๆ ลดลง และเมื่อระดับแควคอยเริ่มน้อยลงแล้ว ERICA+ ก็เริ่มแนะนำให้แหล่งกำเนิดข้อมูลเพิ่มอัตราการส่งขึ้นทีละน้อย (ตามรูปที่ 6.4 (ก)) เพื่อรักษาระดับแควคอยให้อยู่ในระดับที่ไม่น้อยจนเกินไป

จะเห็นได้ว่าช่วงเวลาที่แหล่งกำเนิดแบบ VBR เพ็งหยุดส่ง ข้อมูลในแถวคอยที่เคยสะสมใน สวิตช์ ATM จะถูกระบายออกอย่างรวดเร็ว ทั้งนี้เพราะความเหลื่อมของเวลาที่เกิดจากการที่แหล่ง กำเนิดแบบ VBR หยุดส่งข้อมูลไปแล้ว แต่แหล่งกำเนิดแบบ ABR ยังไม่มีการเพิ่มอัตราการส่งนั้น เอง จึงทำให้ระดับแถวคอยที่อยู่ในสวิตช์ลดลงอย่างรวดเร็ว แต่เนื่องจากความเหลื่อมดังกล่าวมาก จนทำให้ข้อมูลในสวิตช์ถูกระบายออกจนหมดก่อนที่แหล่งกำเนิดแบบ ABR จะเพิ่มอัตราการส่งให้ อยู่ในระดับที่เหมาะสม จึงทำให้เกิดภาวะขาดแคลนข้อมูลเป็นช่วงสั้น ๆ และเป็นผลให้การใช้งาน ลิงก์ไม่เต็มประสิทธิภาพ (พิจารณาได้ดังรูปที่ 6.4 (ค))

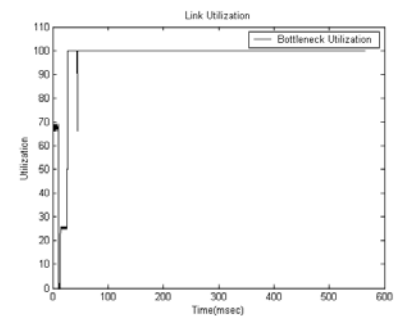
- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 2



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด แบบ ABR



(ข) ปริมาณแถวคอยในสวิตช์ ATM



(ค) ประสิทธิภาพการใช้งานลิงก์

รูปที่ 6.5 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR

โดยใช้อัลกอริทึม ERICA+ ภายใต้แบบจำลองชุดที่ 2

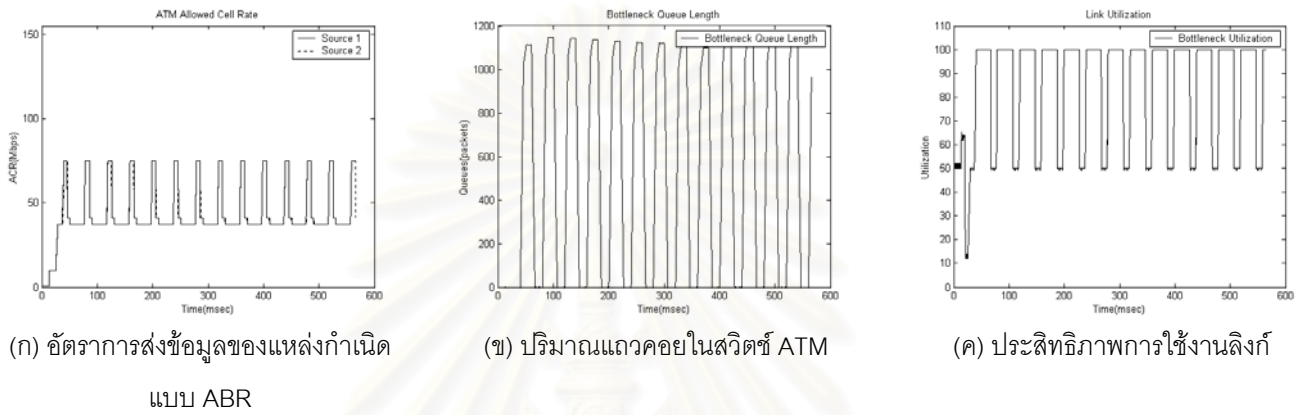
ลักษณะการแกว่งตัวของค่า ACR อยู่ที่ประมาณ 20 Mbps ถึง 80 Mbps ซึ่งเป็นไปตาม หลักการของ ERICA+ ที่มักจะปรับอัตราการส่งข้อมูลเพื่อให้สอดคล้องกับแถวคอยให้อยู่ใน ระดับหนึ่งเสมอ และ ERICA+ ก็จะทำให้การปรับอัตราการส่งข้อมูลให้ต่ำกว่าที่คำนวณได้จาก Fairshare ทั้งนี้ก็เพื่อรักษาระดับข้อมูลในแถวคอย ซึ่งการบริหารแถวคอยดังกล่าวส่งผลให้การใช้ งานลิงก์ในกรณีนี้มีประสิทธิภาพเต็มที่ (ดังรูป 6.5 (ค)) แต่สำหรับช่วงแรกของการเชื่อมต่อที่มีการ ใช้งานลิงก์อย่างไม่มีประสิทธิภาพนั้น เป็นผลจากการปรับค่าอัตราการส่งข้อมูลของแหล่ง กำเนิดในช่วงแรกเท่านั้น

6.1.1.3 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้ อัลกอริทึม FMMRA

อัลกอริทึม FMMRA ที่ใช้ในแบบจำลองนี้ มีการตั้งค่าดังต่อไปนี้

อัลกอริทึม FMMRA ที่สวิตช์ ATM	AI = 5 ms; Target Utilization = 1.0
	QT = 500 เซลล์; DQT = 1500 เซลล์

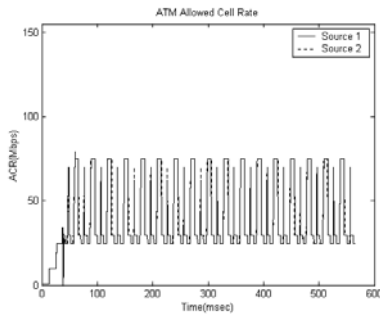
- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 1



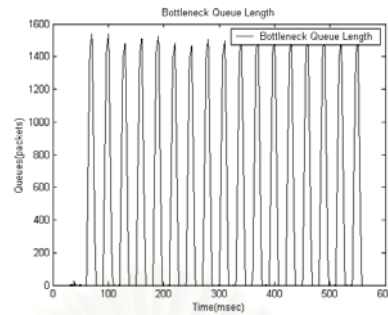
รูปที่ 6.6 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้อัลกอริทึม FMMRA ภายใต้แบบจำลองชุดที่ 1

จะเห็นได้ว่าผลการจำลองที่ได้จากการทำงานของอัลกอริทึมแบบ FMMRA มีการตอบสนองต่อการเพิ่มอัตราการส่งข้อมูลช้ากว่ากรณีที่ใช้อัลกอริทึม ERICA และ ERICA+ สังเกตได้จากช่วงเหลี่ยมที่กว้างขึ้นระหว่างเวลาที่แหล่งกำเนิดข้อมูลแบบ VBR หยุดส่งไปแล้ว แต่แหล่งกำเนิดแบบ ABR ยังไม่มีการปรับอัตราการส่ง (เปรียบเทียบรูปที่ 6.6 (ก) และ 6.1 (ก)) และเมื่อพิจารณาจากรูปที่ 6.6 (ค) จะเห็นว่าช่วงที่การใช้งานลิงก์ไม่เต็ม 100% มีความกว้างของช่วงมากขึ้นด้วย (จากการวิเคราะห์ในหัวข้อ 6.1.1.1 ซึ่งให้เห็นว่าระยะเวลาที่เหลี่ยมกันเพราะการเพิ่มอัตราการส่งของ ABR ไม่ทันท่วงที มีผลทำให้การใช้งานลิงก์ไม่เต็มประสิทธิภาพ) สำหรับสาเหตุที่ทำให้ FMMRA ปรับอัตราการส่งขึ้นได้ช้านี้เกิดจากการคำนวณค่า Advertised rate ของ FMMRA จะพิจารณาผลของระดับความคับคั่งของระบบที่วัดจากระดับแควคอย (ผ่านตัวแปร QT และ DQT ตามลำดับ) ซึ่งในช่วงเวลาที่ทำการคำนวณระดับแควคอยในสวิตช์มีค่ามาก ทำให้ FMMRA เข้าใจว่าระบบอยู่ในภาวะคับคั่ง และเป็นผลให้ FMMRA ไม่ปรับอัตราการส่งในทันที สำหรับค่า ACR ที่แหล่งกำเนิดข้อมูลใช้ในกรณีนี้ จะแกว่งขึ้นลงตามค่า Advertised rate ที่คำนวณได้เหมือนกับค่า Fairshare ในกรณีของ ERICA

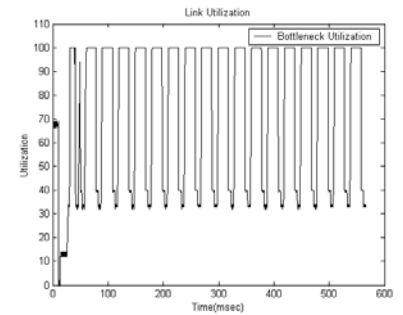
- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 2



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR



(ข) ปริมาณแควคอยในสวิตช์ ATM



(ค) ประสิทธิภาพการใช้งานลิงก์

รูปที่ 6.7 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR

โดยใช้อัลกอริทึม FMMRA ภายใต้แบบจำลองชุดที่ 2

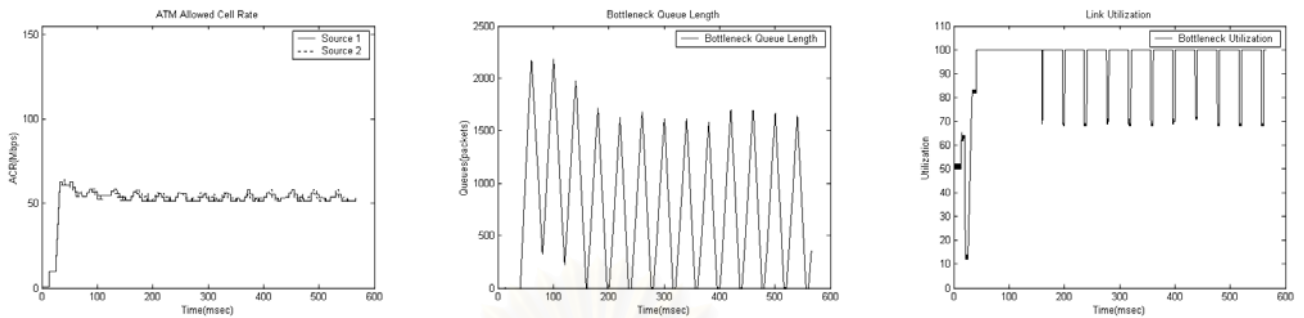
ค่าอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ที่เกิดจากการจำลองการทำงานในกรณีนี้มีลักษณะแกว่งตัวมาก ทั้งนี้เมื่อเปรียบเทียบกับผลการจำลองภายใต้แบบจำลองชุดที่ 1 จะเห็นว่าอัตราการส่งข้อมูลในกรณีนี้ (แสดงดังรูปที่ 6.7 (ก)) มีลักษณะการพุ่งสูงขึ้นเป็นช่วงสั้น ๆ มากกว่าส่วนปริมาณแควคอยในสวิตช์ที่เกิดขึ้น มีลักษณะคล้ายกับผลที่เกิดจากแบบจำลองชุดที่ 1 มาก เพียงแต่ความถี่ในการแกว่งตัวของแควคอยนั้นมีมากกว่า นอกจากนี้เมื่อเปรียบเทียบกับรูป 6.6 (ค) และรูป 6.7 (ค) ซึ่งแสดงประสิทธิภาพการใช้งานลิงก์ภายใต้แบบจำลองชุดที่ 1 และ 2 ตามลำดับ จะเห็นว่าในกรณีมีแหล่งกำเนิดแบบ VBR ที่มีความเปลี่ยนแปลงรุนแรงมาก ๆ อย่างในกรณีของแบบจำลองชุดที่ 2 จะทำให้ประสิทธิภาพการใช้งานลิงก์น้อยลง

6.1.1.4 การจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้ อัลกอริทึม E-FMMRA

อัลกอริทึม E-FMMRA ที่ใช้ในแบบจำลองนี้ มีการตั้งค่าดังต่อไปนี้

อัลกอริทึม E-FMMRA ที่สวิตช์ ATM	AI = 5 ms; Target Utilization = 1.0
	QT = 500 เซลล์; DQT = 1500 เซลล์; Alpha = 1/8 #
	# เป็นค่าที่ผู้พัฒนา E-FMMRA แนะนำให้ใช้

- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 1



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด

(ข) ปริมาณแควคอยในสวิตช์ ATM

(ค) ประสิทธิภาพการใช้งานลิงก์

แบบ ABR

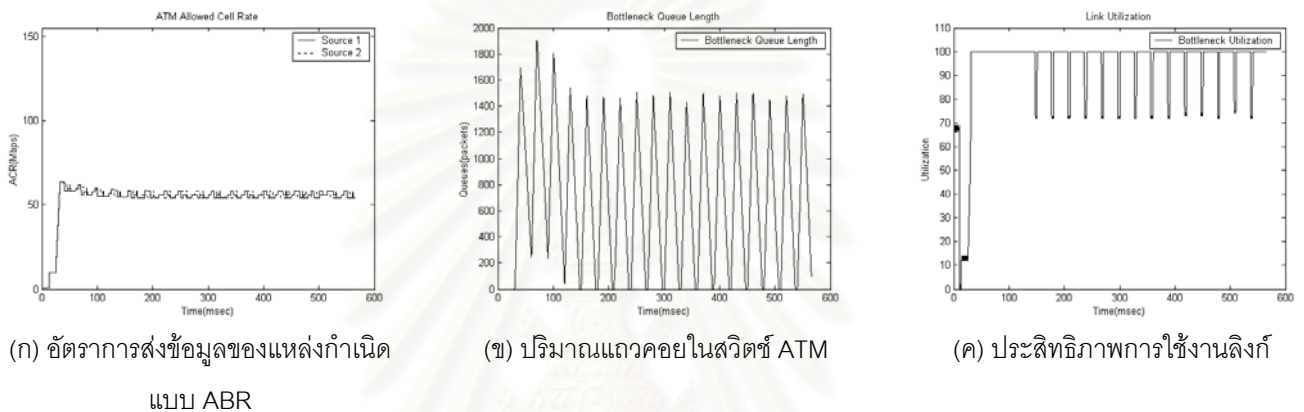
รูปที่ 6.8 ผลการจำลองการทำงานของระบบการควบคุมความคับคั่งของ ABR

โดยใช้อัลกอริทึม E-FMMRA ภายใต้แบบจำลองชุดที่ 1

จะเห็นว่าลักษณะการปรับค่า ACR ที่เกิดขึ้นจากอัลกอริทึมแบบ E-FMMRA นี้แตกต่างจากผลที่เกิดขึ้นจากอัลกอริทึมอื่น ๆ ทั้งนี้เพราะอัลกอริทึม E-FMMRA จะใช้การเฉลี่ยค่า ER ตามวิธี running weight average (ดังสมการ (2.8)) อย่างไรก็ตามในช่วงแรกที่แหล่งกำเนิด ABR พยายามปรับค่า ACR ให้เท่ากับค่า ER ที่ E-FMMRA คำนวณได้ ลักษณะการเปลี่ยนแปลงที่เกิดขึ้น จะมีผลเหมือนกับกรณีอื่น ๆ นั่นคือเป็นผลของค่า RIF ตามที่ได้วิเคราะห์ไปแล้วในหัวข้อ 6.1.1.1 เนื่องจาก E-FMMRA คำนวณค่า ER จากการนำค่าเก่ามาเฉลี่ยด้วย ทำให้ค่า ER ที่ได้แกว่งตัวในช่วงแคบ ๆ ทั้งนี้ถ้าลองคำนวณอัตราการส่งโดยเฉลี่ยแบบคร่าว ๆ ดูจะได้ว่าแหล่งกำเนิดแบบ VBR มีอัตราการส่งเฉลี่ยเท่ากับ 37.5 Mbps (โดยคำนวณจาก $(75 \times 20 + 0 \times 20) / 20$) ดังนั้นค่า ER เฉลี่ยที่ได้จึงมีค่าประมาณ 56.13 Mbps (มาจาก $(149.76 - 37.5) / 2$) และเมื่อพิจารณารูปที่ 6.8 (ก) ค่า ACR ที่ได้ นั้น จะแกว่งอยู่ประมาณ 50 กว่า ๆ Mbps ซึ่งเป็นไปตามการประมาณค่าข้างต้น อย่างไรก็ตามค่า ACR ที่ใช้ในบางช่วงมีค่าน้อยกว่าค่าเฉลี่ย ทั้งนี้เพราะสวิตช์ต้องการระบายข้อมูลในแควคอยที่มีระดับมากเกินไป ซึ่งเป็นผลจากการที่แหล่งกำเนิดข้อมูลแบบ VBR ทำการส่งข้อมูล แต่แหล่งกำเนิดแบบ ABR ไม่ได้ปรับลดอัตราการส่งลงมาก (เพราะต้องเฉลี่ยกับค่า ACR จากตอนที่แหล่งกำเนิดข้อมูลแบบ VBR หยุดส่งด้วย) นอกจากนี้เมื่อแหล่งกำเนิดแบบ VBR หยุดส่งข้อมูลแต่แหล่งกำเนิดแบบ ABR ยังคงส่งข้อมูลที่อัตราในระดับเฉลี่ย ทำให้มีการระบายข้อมูลในแควคอยเพื่อเติมเต็มลิงก์ แต่เนื่องจากปริมาณข้อมูลที่สะสมไม่มากพอ จึงทำให้เกิดภาวะขาดแคลนข้อมูลในการส่งขึ้น และเป็นผลให้การใช้งานลิงก์ไม่เต็มประสิทธิภาพดังรูปที่ 6.8 (ค) อย่างไรก็ตามจะเห็นว่าช่วงแรก ๆ ของการเชื่อมต่อการใช้งานลิงก์มีประสิทธิภาพเต็ม

100% นานระยะหนึ่ง ทั้งนี้เกิดจากการที่แหล่งกำเนิดแบบ ABR ส่งข้อมูลมากกว่าค่าเฉลี่ยมากในช่วงแรก จึงทำให้มีการสะสมข้อมูลจำนวนมาก (ประมาณ 2,200 เซลล์) และปริมาณข้อมูลระดับนี้ก็มากเพียงพอที่จะเติมเต็มลิงก์ได้ อย่างไรก็ตามเมื่ออัตราการส่งข้อมูลถูกปรับให้มีค่าน้อยลง ปริมาณข้อมูลที่สะสมก็จะน้อยลงตามไปด้วย และทำให้เกิดภาวะการใช้งานลิงก์ไม่เต็มประสิทธิภาพเป็นช่วง ๆ ในที่สุด

- ผลการจำลองและวิเคราะห์การทำงาน ที่ได้จากแบบจำลองชุดที่ 2



รูปที่ 6.9 ผลการจำลองการทำงานของกระบวนการควบคุมความคับคั่งของ ABR โดยใช้อัลกอริทึม E-FMMRA ภายใต้แบบจำลองชุดที่ 2

ผลการจำลองการทำงานของอัลกอริทึม E-FMMRA ภายใต้แบบจำลองชุดที่ 2 นี้ มีลักษณะคล้ายการจำลองภายใต้แบบจำลองชุดที่ 1 มาก ดังนั้นรายละเอียดการวิเคราะห์กรณีต่าง ๆ สามารถพิจารณาได้จากหัวข้อที่แล้ว อย่างไรก็ตามอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ที่เกิดจากแบบจำลองชุดที่ 2 นี้ มีค่าเฉลี่ยมากกว่ากรณีที่เกิดจากแบบจำลองชุดที่ 1 เล็กน้อย (เมื่อคำนวณแบบคร่าว ๆ จะได้ว่าค่าอัตราการส่งข้อมูลเฉลี่ยของแหล่งกำเนิดแบบ VBR มีค่าเท่ากับ $(100 \times 10 + 0 \times 20) / 30 = 33.33$ Mbps จึงทำให้ค่า ER เฉลี่ยแบบคร่าว ๆ มีค่าประมาณ $(149.76 - 33.33) / 2 = 58.22$ Mbps) นอกจากนี้จะเห็นว่าการแกว่งของค่า ACR ที่พิจารณาจากรูปที่ 6.9 (ก) จะแคบกว่า ทั้งนี้เพราะช่วงเวลาที่แหล่งกำเนิด VBR หยุดส่งข้อมูลสั้นกว่าในแบบจำลองชุดที่ 1 จึงทำให้ผลกระทบที่มีต่อการคำนวณค่า ER เฉลี่ยของ E-FMMRA น้อยลง

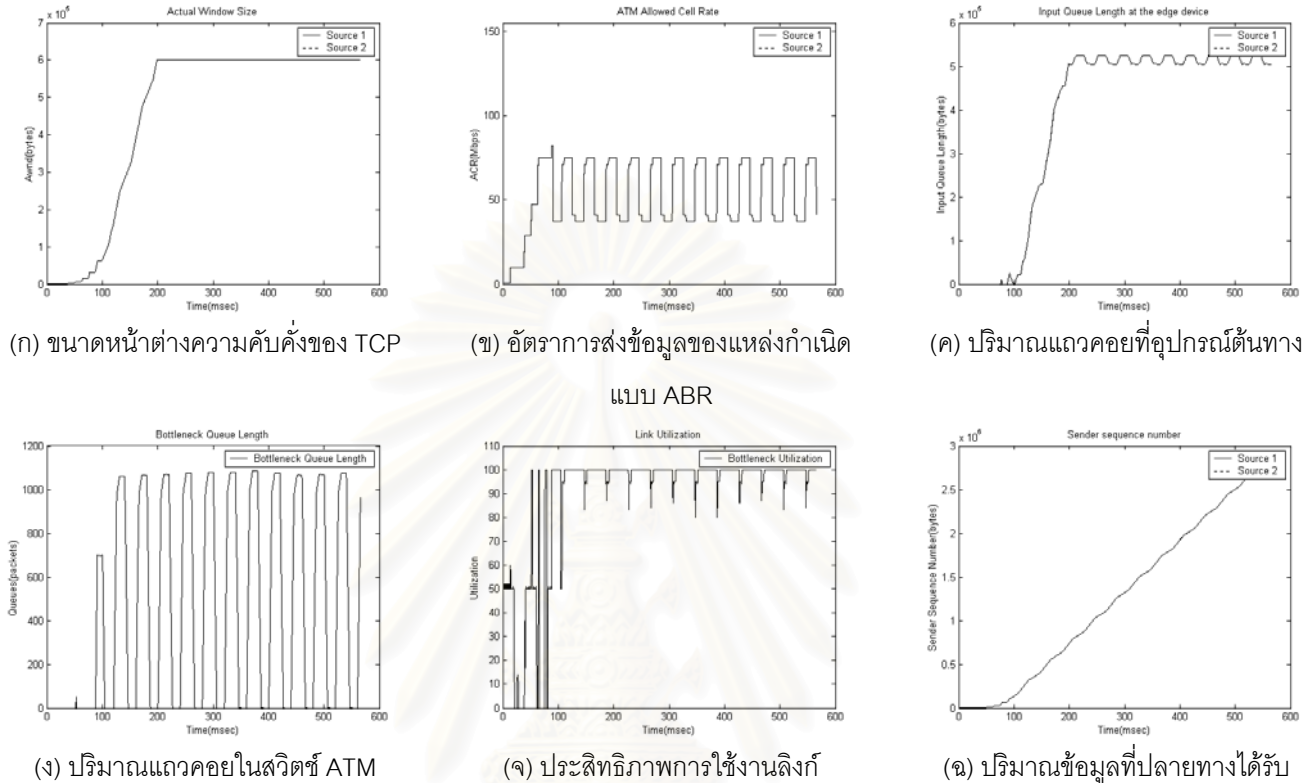
6.1.2 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ

การจำลองการทำงานชุดนี้เป็นการศึกษาและวิเคราะห์ผลของอัลกอริทึมที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR ที่มีต่อกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และเนื่องจากแบบจำลองที่ใช้เป็นแบบจำลองที่มีทราฟฟิกที่มีการเปลี่ยนแปลงตลอดเวลา ดังนั้นการจำลองการทำงานในกรณีนี้ยังเป็นการทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม ภายใต้ทราฟฟิกที่มีการเปลี่ยนแปลงอีกด้วย สำหรับแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.2 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ VBR 1 ตัว และแหล่งกำเนิดข้อมูลแบบ TCP ที่ทำงานร่วมกับแหล่งกำเนิดแบบ ABR อีก 2 ชุด
แหล่งกำเนิดข้อมูล TCP	ขนาดแพ็กเก็ต (MTU) = 1,024 ไบต์ ; ขนาดหน้าต่างผู้รับ = 600,000 ไบต์
แหล่งกำเนิดข้อมูล VBR	ช่วงคาบเวลาที่หยุดส่งข้อมูล = ช่วงคาบเวลาที่มีการส่งข้อมูล = 20 ms อัตราการส่งข้อมูลเมื่ออยู่ในช่วงคาบการส่งข้อมูล = 75 Mbps (50% ของความจุลิงก์)
(ลักษณะทราฟฟิกของแหล่งกำเนิดข้อมูลแบบ VBR ที่สร้างขึ้น แสดงดังรูปที่ 6.1 (ก))	
ส่วนอุปกรณ์อื่น ๆ	เช่น แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึมชนิดต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.1.1

หมายเหตุ เนื่องจากในหัวข้อ 6.1.1 มีการจำลองโดยใช้แบบจำลอง 2 ชุด ซึ่งประกอบด้วยแหล่งกำเนิด VBR ที่มีคุณลักษณะต่างกัน และผลจากการจำลองของแบบจำลองทั้ง 2 ชุด แสดงถึงธรรมชาติการทำงานของอัลกอริทึมชนิดต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR โดยเฉพาะในแง่ของการตอบสนองต่อทราฟฟิกที่มีการเปลี่ยนแปลงตลอดเวลา และยังแสดงให้เห็นถึงความแตกต่างที่เกิดจากลักษณะทราฟฟิกที่มีความรุนแรงต่างกันด้วย อย่างไรก็ตามสำหรับการจำลองการทำงานในหัวข้อนี้จะแสดงผลการจำลองโดยใช้แบบจำลองแบบเดียวเท่านั้น นั่นคือใช้คุณสมบัติของแหล่งกำเนิดข้อมูลแบบ VBR ตามแบบจำลองชุดที่ 1 ทั้งนี้เพราะลักษณะทราฟฟิกที่เกิดจากแหล่งกำเนิดแบบ VBR ตามแบบจำลองชุดที่ 1 นั้นมีความรุนแรงน้อยกว่า ทำให้สามารถสังเกตและวิเคราะห์ผลการจำลองได้ง่ายกว่า อย่างไรก็ตามผู้วิจัยได้ทำการจำลองการทำงานแบบจำลองที่มีคุณลักษณะของแหล่งกำเนิดข้อมูลแบบ VBR ตามแบบจำลองชุดที่ 2 แต่จะเสนอในรูปแบบของการวิเคราะห์เท่านั้น สำหรับเหตุผลที่ไม่นำผลการจำลองดังกล่าวมาแสดงในวิทยานิพนธ์ฉบับนี้เป็นเพราะผู้วิจัยต้องการลดความซ้ำซ้อนของการนำเสนอเท่านั้น

6.1.2.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการตั้งเดิม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR



รูปที่ 6.10 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการตั้งเดิม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR

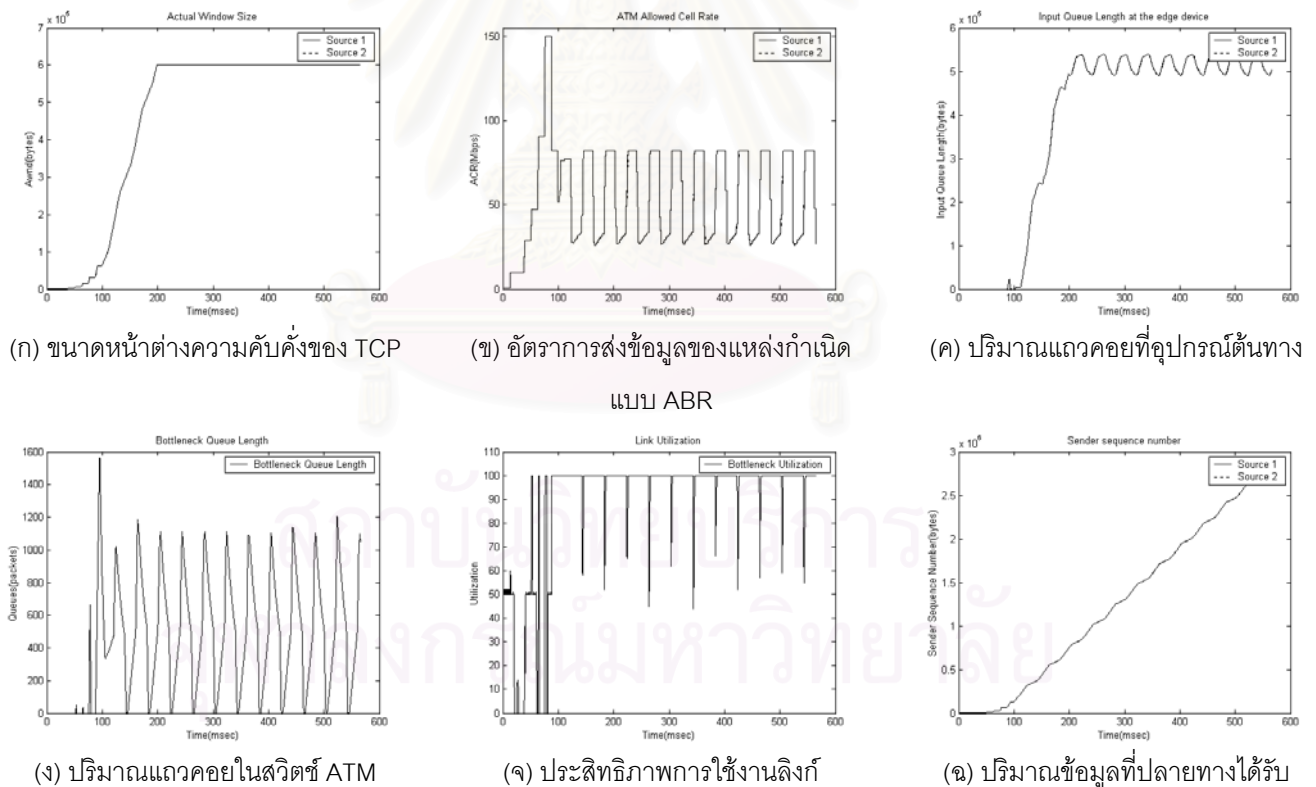
ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 2.890752×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.892800×10^6 ไบต์

จากรูปที่ 6.10 (ก) ขนาดหน้าต่างความคับคั่งของ TCP เพิ่มขึ้นเรื่อย ๆ จนกระทั่งค่าขนาดหน้าต่างคงที่อยู่ที่ 600,000 ไบต์ ส่วนค่า ACR ที่วัดได้นั้น (จากรูป 6.10 (ข)) จะเริ่มต้นที่ค่าน้อย ๆ (คือเท่ากับ 0.5 Mbps = ICR) แล้วค่อย ๆ เพิ่มขึ้นเป็นขั้น ๆ และเมื่อเวลาผ่านไปประมาณ 90 ms ค่า ACR ก็ปรับตัวอยู่ที่ประมาณ 75 Mbps หลังจากนั้นการแกว่งตัวของค่า ACR ก็มีลักษณะเหมือนกับในกรณีที่ใช้อัลกอริทึม ERICA ภายใต้แบบจำลองที่ไม่มีการส่งข้อมูลผ่าน TCP (แบบจำลองชุดที่ 1 ในหัวข้อ 6.1.1) นั่นคือค่า ACR แกว่งอยู่ระหว่างค่าประมาณ 75 และ 37 Mbps

สำหรับรูปที่ 6.10 (ค) เป็นรูปที่แสดงผลการจำลองปริมาณข้อมูลที่สะสมในแถวคอยของอุปกรณ์ต้นทาง จะเห็นว่าช่วงเริ่มต้นไม่มีการสะสมแถวคอยเลย แต่เมื่อผ่านไป 90 ms อุปกรณ์ต้นทางก็เริ่มมีการสะสมข้อมูลในแถวคอย โดยปริมาณจะค่อย ๆ เพิ่มขึ้นเรื่อย ๆ จนกระทั่งไปแกว่งตัวในช่วงแคบ ๆ ที่ระดับประมาณ 520,000 ไบต์ สำหรับระดับแถวคอยที่สวิตช์ ATM และการใช้งานลิงก์ ซึ่งแสดงในรูปที่ 6.10 (ง) และ 6.10 (จ) ตามลำดับ มีลักษณะเหมือนกรณีไม่ผ่าน TCP ยกเว้นช่วงแรก ๆ ที่ไม่มีแถวคอยสะสม หรือถ้ามีก็มีการสะสมเพียงเล็กน้อย ส่วนการใช้งานลิงก์ในช่วงแรกจะอยู่ในระดับต่ำและแกว่งตัวนานกว่า ส่วนรูปที่ 6.10 (ฉ) แสดงปริมาณข้อมูลที่ปลายทางได้รับ ซึ่งจะเห็นว่าช่วงแรก ๆ ปลายทางจะได้รับข้อมูลที่ละน้อย (ความชันของกราฟที่แสดงมีค่าน้อยมาก) แต่หลังจากเวลาผ่านไปช่วงหนึ่งปลายทางก็เริ่มได้รับข้อมูลที่อัตราสม่ำเสมอ

6.1.2.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR



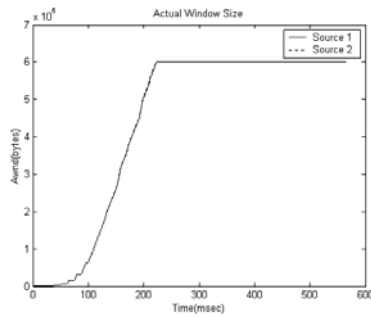
รูปที่ 6.11 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

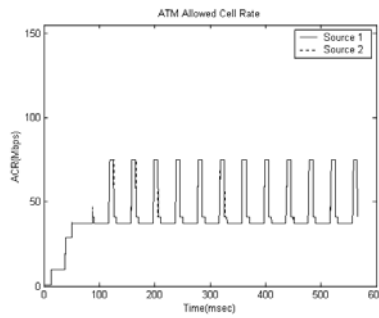
- การเชื่อมต่อที่ 1 วัดได้ 2.915328×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.915328×10^6 ไบต์

ขนาดหน้าต่างความคับคั่งของ TCP เพิ่มขึ้นเรื่อย ๆ จนกระทั่งขนาดหน้าต่างมีค่าคงที่อยู่ที่ 600,000 ไบต์ ดังรูปที่ 6.11 (ก) ส่วนอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ปริมาณแฉวคอยที่สวิตช์ ATM และการใช้งานลิงก์ ที่ได้จากการจำลองสามารถพิจารณาได้จากรูป 6.11 (ข) 6.11 (ง) และ 6.11 (จ) ตามลำดับ จะเห็นว่ามึลักษณะโดยรวมเหมือนผลการจำลองที่ได้จากกรณีไม่ผ่าน TCP (ในหัวข้อ 6.1.1.2) ยกเว้นช่วงเริ่มการเชื่อมต่อที่ค่า ACR จะเพิ่มค่าขึ้นเรื่อย ๆ จนมีค่าเกือบถึง 150 Mbps เลยทีเดียว (นั่นคือการเพิ่มค่าดังกล่าวจะใช้เวลามากกว่า และมีค่าสูงสุดสูงกว่าในกรณีที่ได้จากการจำลองในหัวข้อ 6.1.1.2) ส่วนปริมาณข้อมูลในสวิตช์ ATM ช่วงแรกไม่มีการสะสมเลย แต่พอเวลาผ่านไปช่วงหนึ่งประมาณแฉวคอยก็เพิ่มขึ้นอย่างรวดเร็วจนมีปริมาณมากกว่าในกรณีที่ไม่มี TCP ควบคุม อย่างไรก็ตามเมื่อค่า ACR เริ่มเข้าสู่จุดทำงาน ระดับแฉวคอยในช่วงเวลาดังกล่าวก็ไม่มีแตกต่างไปจากกรณีที่ไม่มี TCP เลย สำหรับประสิทธิภาพการใช้งานลิงก์ก็เช่นเดียวกัน กล่าวคือช่วงแรก ๆ ประสิทธิภาพการใช้งานลิงก์ส่วนใหญ่จะเป็นผลมาจากแหล่งกำเนิดแบบ VBR เท่านั้น แต่เมื่อผ่านช่วงเวลาหนึ่งไปแล้วระบบก็สามารถปรับตัวได้จนมีการใช้งาน 100% เกือบตลอดเวลา สำหรับรูปที่ 6.11 (ค) เป็นรูปที่แสดงผลการจำลองปริมาณแฉวคอยที่สะสมในอุปกรณ์ต้นทาง จะเห็นว่าช่วงเริ่มต้นไม่มีการสะสมแฉวคอยในส่วนนี้เลยเลย แต่เมื่อเวลาผ่านไป 90 ms ก็เริ่มมีการสะสมข้อมูลในอุปกรณ์ส่วนนี้ โดยปริมาณจะค่อย ๆ เพิ่มขึ้นเรื่อย ๆ จนกระทั่งไปแกว่งตัวระยะแคบ ๆ ที่ระดับประมาณ 480,000 – 550,000 ไบต์ (ซึ่งการแกว่งตัวดังกล่าวจะแกว่งมากกว่ากรณี ที่ใช้อัลกอริทึม ERICA) และจากรูปที่ 6.11 (ฉ) แสดงปริมาณข้อมูลที่ปลายทางได้รับ ซึ่งจะเห็นว่าช่วงแรก ๆ ปลายทางจะได้รับข้อมูลน้อยมาก แต่หลังจากผ่านเวลาไปช่วงหนึ่งปลายทางก็เริ่มได้รับข้อมูลได้มากขึ้น จนกระทั่งสามารถรับข้อมูลในอัตราที่สม่ำเสมอ นอกจากนี้จะพบว่าค่าวิสัยสามารถสุทธิที่วัดได้ของการเชื่อมต่อแต่ตัว มีค่ามากกว่ากรณีที่ใช้ัลกอริทึม ERICA (และมีค่ามากที่สุดเมื่อเปรียบเทียบกับกรจำลองโดยใช้อัลกอริทึมที่เหลือด้วย)

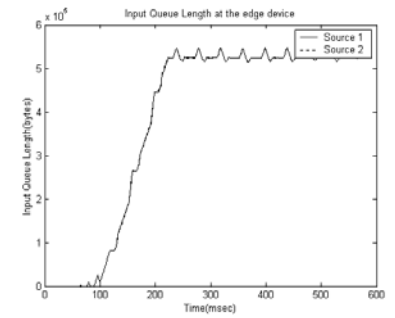
6.1.2.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

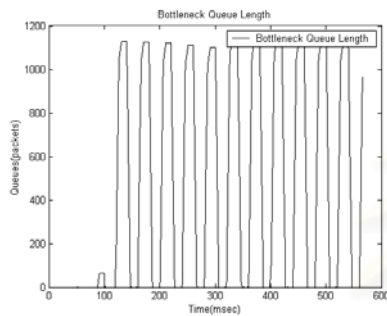


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

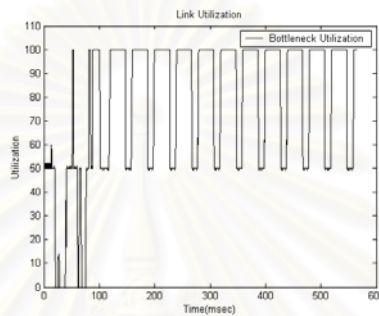


(ค) ปริมาณแควคอยที่อุปกรณ์ต้นทาง

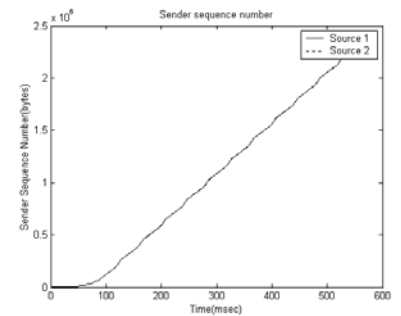
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

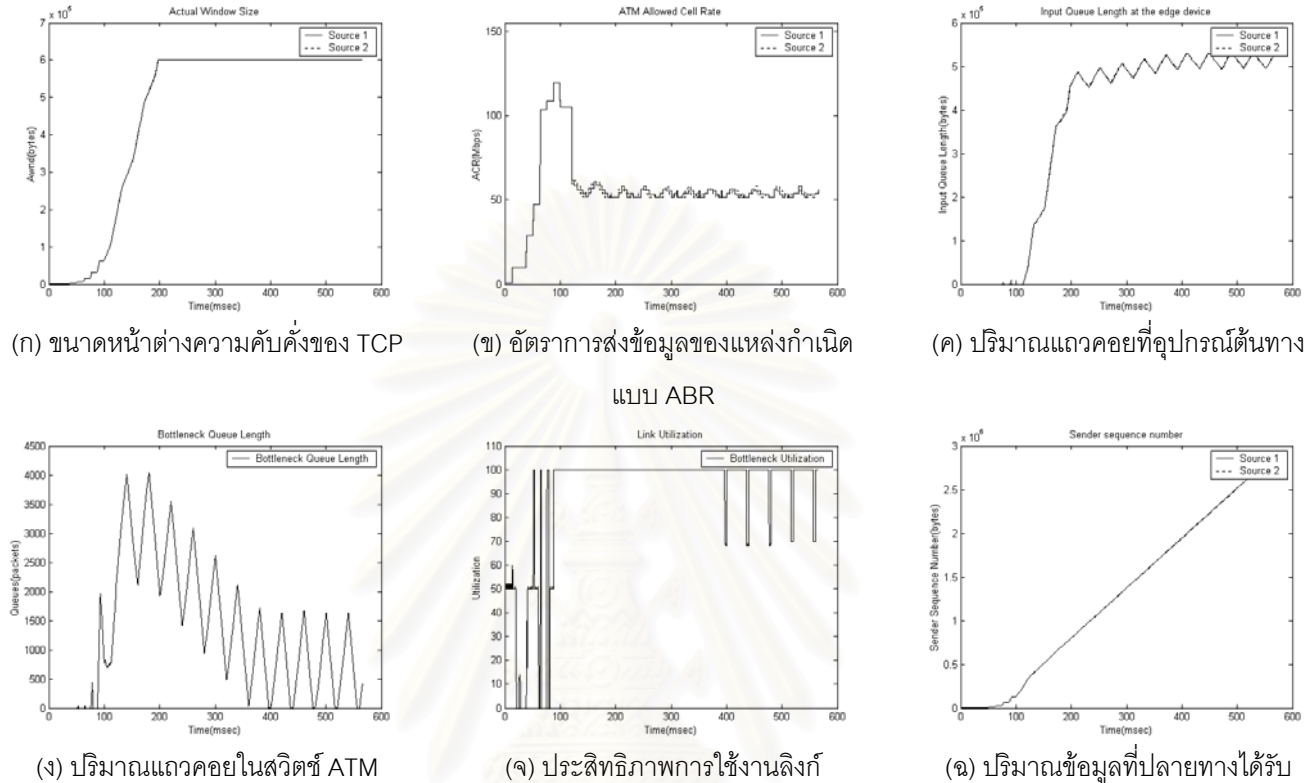
รูปที่ 6.12 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 2.378752×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.380800×10^6 ไบต์

ขนาดหน้าต่างความคับคั่งของ TCP มีความแตกต่างจากผลการจำลองที่เกิดจากอัลกอริทึม ERICA และ ERICA+ ตรงที่การเพิ่มขนาดหน้าต่างความคับคั่งของกรณีนี้ จะเพิ่มช้ากว่ากรณีอื่น ๆ เล็กน้อย ส่วนลักษณะของอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ปริมาณแควคอยในสวิตช์ ATM และประสิทธิภาพการใช้งานลิงก์ นั้นมีลักษณะเหมือนหัวข้อ 6.1.3 (กรณีที่ส่งข้อมูลผ่าน TCP) ยกเว้นช่วงแรกเท่านั้นที่ปริมาณของค่าดังกล่าวมีค่าน้อยกว่ากรณีที่ไม่มี TCP นอกจากนี้ปริมาณแควคอยในอุปกรณ์ต้นทางที่วัดได้ (แสดงดังรูปที่ 6.12 (ค)) จะเริ่มสะสมตั้งแต่ประมาณ 90 ms แรกเป็นต้นไป จนกระทั่งปริมาณแควคอยมีการแกว่งตัวในช่วงแคบ ๆ ที่ระดับประมาณ 520,000 ไบต์ นอกจากนี้เมื่อเปรียบเทียบปริมาณข้อมูลสุทธิที่ปลายทางได้รับ จะเห็นว่าปริมาณข้อมูลสุทธิที่เกิดจากการจำลองโดยใช้อัลกอริทึม FMMRA มีค่าน้อยกว่าการจำลองโดยใช้อัลกอริทึมชนิดอื่น ๆ

6.1.2.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



รูปที่ 6.13 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 2.890752×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.884608×10^6 ไบต์

ในการปรับตัวของขนาดหน้าต่างความคับคั่งของ TCP มีลักษณะเหมือนในกรณีที่ใช้อัลกอริทึม ERICA และ ERICA+ ส่วนอัตราการส่งข้อมูลของ ABR ปริมาณแฉวคคอยในสวิตช์ ATM และประสิทธิภาพการใช้งานลิงก์ ค่อนข้างแตกต่างจากการจำลองโดยใช้อัลกอริทึมชนิดอื่น ๆ นั่นคือต้องใช้เวลามากกว่าในการเข้าสู่ภาวะคงตัว กล่าวคือค่า ACR ของแหล่งกำเนิดข้อมูลแบบ ABR จะค่อย ๆ ปรับค่าขึ้นเป็นขั้น ๆ จนกระทั่งมีค่ามากถึงประมาณ 120 Mbps แล้วค่อย ๆ ตกลงมา หลังจากนั้นค่า ACR จะแกว่งอยู่ที่ประมาณ 55 Mbps โดยจะแกว่งแคบลงเรื่อย ๆ สำหรับ

ระดับแควคอยในสวิตช์ ATM เมื่อผ่าน 90 ms แรกไป ระดับแควคอยก็จะเพิ่มขึ้นเยอะมาก คือมีค่าประมาณ 4,000 ไบต์ หลังจากนั้นก็ค่อย ๆ ลดระดับลง ซึ่งผลการจำลองค่าประสิทธิภาพการปฏิบัติงานลิงก์ (ดังรูป 6.13 (จ)) แสดงถึงความสอดคล้องกับปริมาณข้อมูลในแควคอยดังกล่าว

6.1.2.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม และโดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR

เนื่องจากกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิมนี้ คือการนำกระบวนการควบคุมการไหลของ TCP ทำงานซ้อนบนกระบวนการควบคุมความคับคั่งของ ABR โดยไม่มีการเปลี่ยนแปลงการทำงานของส่วนใดเลย นั่นคือใช้อัลกอริทึมของกระบวนการควบคุมความคับคั่งของ ABR ในการควบคุมทราฟฟิกในชั้น ATM และสำหรับ TCP ก็ควบคุมอัตราการส่งข้อมูลโดยใช้ขนาดหน้าต่างความคับคั่ง ดังรูป 6.10 - 6.13 (ก) ซึ่งในช่วงแรกการเพิ่มขนาดหน้าต่างจะเป็นไปตามหลักการ Slow Start (ซึ่งมีกรณีวิเคราะห์ในภาคผนวก ก) และด้วยผลการทำงานในช่วง Slow Start นี้เอง ทำให้ช่วงแรกของการสื่อสาร เกิดภาวะขาดแคลนข้อมูลในชั้น ATM หรือเรียกอีกอย่างหนึ่งว่าเกิด “ซีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง” โดยภาวะดังกล่าวสามารถพิจารณาและวิเคราะห์ได้ดังนี้

1 ค่า ACR ที่แหล่งกำเนิดข้อมูลแบบ ABR พิจารณาตามรูป 6.10 - 6.13 (ข)

เมื่อพิจารณาอัตราการส่งข้อมูลของแหล่งกำเนิดข้อมูลแบบ ABR จะเห็นว่าค่า ACR จะเริ่มต้นที่ค่าน้อย ๆ (นั่นคือค่า ICR = 0.5 Mbps) แล้วค่อย ๆ เพิ่มขึ้นเป็นขั้น ๆ (โดยปริมาณการเพิ่มในแต่ละขั้นขึ้นอยู่กับค่า RIF) เหมือนกรณีที่ไม่ผ่าน TCP แต่จะใช้เวลาในการปรับค่ามากกว่า (เปรียบเทียบได้จากรูป 6.2 (ก) 6.4 (ก) 6.6 (ก) และ 6.8 (ก)) ทั้งนี้เนื่องมาจากแหล่งกำเนิดข้อมูลจะถูกจำกัดปริมาณข้อมูลที่ส่งจากกระบวนการควบคุมการไหลของ TCP ซึ่งทำงานในช่วง Slow Start และเมื่อข้อมูลที่ส่งผ่านโครงข่ายมีปริมาณน้อย ทำให้ปริมาณเซลล์ RM ที่แทรกตัวเข้าไปทำหน้าที่สำรวจโครงข่ายน้อยตามไปด้วย เป็นผลให้การปรับค่า ACR ต้องใช้เวลามากขึ้น (เพราะค่า ACR จะปรับตามจังหวะการมาถึงของเซลล์ RM ย้อนกลับ)

นอกจากนี้จะเห็นว่าในช่วงแรกค่า ACR จะถูกปรับค่าจนมีค่ามากกว่าค่า Fairshare หรือมากกว่าค่าที่คำนวณได้จากกรณีไม่มี TCP ทั้งนี้เกิดจากค่า ACR ที่แสดงในรูป และที่นิยามไว้ใน [1] คือค่าอัตราการส่งข้อมูลที่โครงข่ายอนุญาตให้แหล่งกำเนิดส่งได้ นั่นคือแหล่งกำเนิดสามารถส่งข้อมูลในอัตราเท่าไรก็ได้ระหว่าง 0 ถึงค่า ACR ดังนั้นค่า ACR ที่แสดงจึงไม่ใช่ค่าอัตราการส่งขอ

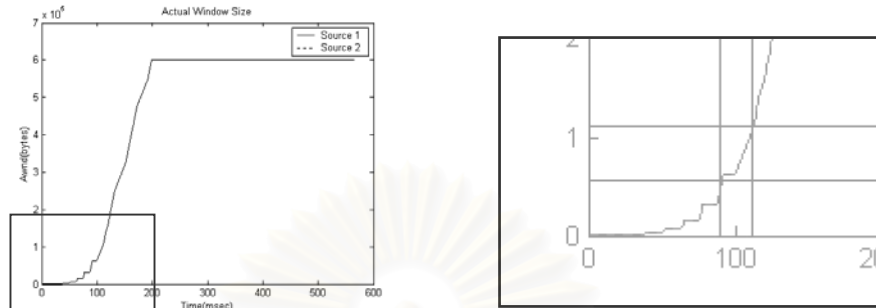
มูลที่แหล่งกำเนิดส่งจริง ๆ เสมอไป ทั้งนี้ขึ้นอยู่กับที่แหล่งกำเนิดมีข้อมูลเพียงพอที่จะส่งหรือไม่ สำหรับกรณีนี้ค่า ACR ในช่วงแรก ๆ ที่แสดงในรูปไม่ใช่ค่าอัตราการส่งจริงที่แหล่งกำเนิดใช้ส่งข้อมูล (พิจารณาจากส่วนอื่น ๆ ประกอบ เช่นประสิทธิภาพการใช้งานลิงก์ เป็นต้น) ในทางกลับกันอัลกอริทึมต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR เห็นปัญหานี้จึงแก้ปัญหาโดยการวัดอัตราการส่งข้อมูลจริงของแหล่งกำเนิดเสียเอง ทำให้ค่าที่อัลกอริทึมใช้อาจไม่เท่ากับค่า ACR ที่บันทึกไว้ในเซลล์ RM ซึ่งในกรณีนี้ค่าอัตราการส่งจริงในช่วงแรกมีค่าน้อยกว่าค่า ACR มาก ทำให้อัลกอริทึมซึ่งมีจุดมุ่งหมายในการจัดสรรทรัพยากรของโครงข่ายให้ถูกใช้อย่างเต็มประสิทธิภาพ พยายามแนะนำให้แหล่งกำเนิดเพิ่มปริมาณการส่งข้อมูลผ่านทางค่า ER (นั่นคือ ER ที่คำนวณได้จะมีค่ามาก เพราะต้องการให้แหล่งกำเนิดเพิ่มการใช้งานทรัพยากรของระบบ โดยผลการทำงานเกิดจากผลของ load factor นิยามไว้ในสมการ (2.3)) และเมื่อแหล่งกำเนิดข้อมูลได้รับเซลล์ RM ย้อนกลับที่บันทึกข้อมูลดังกล่าว แหล่งกำเนิดจึงปรับค่า ACR ให้สอดคล้องกับค่า ER ที่โครงข่ายแนะนำ ซึ่งมีค่ามาก (ดังสะท้อนออกมาทางค่า ACR) สำหรับเหตุผลที่การปรับค่า ACR ของอัลกอริทึม ERICA+ มีค่ามากกว่ากรณีอื่น ๆ เพราะ ERICA+ ใช้กระบวนการควบคุมระดับแถวคอยในการคำนวณหาค่า ER ด้วย ทั้งนี้เมื่อระดับแถวคอยในช่วงแรกมีค่าเป็นศูนย์ ทำให้ผลการคำนวณค่า ER ของ ERICA+ มีค่ามากกว่าผลที่ได้จากอัลกอริทึมอื่น ๆ เนื่องจากต้องการสะสมแถวคอยให้ได้ระดับตามต้องการ

ปัญหาที่สำคัญสำหรับกรณีนี้คือ “เมื่อไหร่จึงจะหยุดเพิ่มอัตราการส่งข้อมูล?” วิธีหาคำตอบต้องย้อนกลับไปทีกราฟของขนาดหน้าต่างความคับคั่งของ TCP กล่าวคือต้องหาจุดที่ TCP เพิ่มขนาดหน้าต่างจนกระทั่งมีปริมาณข้อมูลเพียงพอที่แหล่งกำเนิดจะส่งได้อย่างมีประสิทธิภาพ นั่นคือจุดที่ค่าขนาดหน้าต่างมีค่าเท่ากับค่า Bandwidth-Delay product (รายละเอียดสามารถศึกษาได้จากภาคผนวก ก) สำหรับการพิจารณาค่า Bandwidth-Delay product ในกรณีนี้แบ่งการพิจารณาออกเป็น 2 กรณี คือ

กรณีที่ 1 เมื่อแหล่งกำเนิดข้อมูล VBR หยุดส่งข้อมูล ค่าแบนด์วิดท์ที่การเชื่อมต่อแต่ละตัวควรได้รับ = $149.76/2 = 74.88$ Mbps ส่วนค่า RTT มีค่าเท่ากับ 12 ms (คำนวณจากระยะทางไปกลับหารด้วยความเร็วแสงในเส้นใยแก้วนำแสง) ดังนั้น Bandwidth-Delay product = $74.88 \times 12 = 898.56$ Mbit = 112.32 Mbyte

กรณีที่ 2 เมื่อแหล่งกำเนิดข้อมูล VBR ส่งข้อมูลที่อัตรา 75 Mbps ค่าแบนด์วิดท์ที่การเชื่อมต่อ ABR แต่ละตัวควรได้รับ = $(149.76-75)/2 = 37.38$ Mbps ส่วนค่า RTT มีค่าเท่ากับกรณีแรกคือ 12 ms ดังนั้น Bandwidth-Delay product = $37.38 \times 12 = 448.56$ Mbit = 56.07 Mbyte

จากนั้นเปรียบเทียบค่า Bandwidth-Delay product กับค่าขนาดหน้าต่างความคับคั่งที่ได้จากการจำลอง (เช่นจากรูปที่ 6.10 (ก)) ก็จะสามารถหาเวลาที่แหล่งกำเนิดเริ่มมีปริมาณข้อมูลมากเพียงพอที่จะส่งไปในโครงข่ายอย่างมีประสิทธิภาพ



(ก) ขนาดหน้าต่างความคับคั่งของ TCP (ข) ภาพขยายขนาดหน้าต่างความคับคั่งของ TCP ในช่วงแรกของการเชื่อมต่อ

รูปที่ 6.14 ความสัมพันธ์ระหว่างขนาดหน้าต่างความคับคั่งของ TCP เทียบกับเวลา

จากรูปที่ 6.14 (ข) แสดงให้เห็นว่าประมาณเวลาที่ 90 ms ขนาดหน้าต่างความคับคั่งของ TCP จะมีค่าเท่ากับค่า Bandwidth-Delay product (ที่เกิดภายใต้เงื่อนไขของกรณีที่ 2) นั่นคือถ้าช่วงเวลาที่เราพิจารณานี้เป็นช่วงที่แหล่งกำเนิด VBR ส่งข้อมูล ปริมาณข้อมูลที่ส่งออกจากแหล่งกำเนิด TCP จะมีมากเพียงพอสำหรับแหล่งกำเนิดแบบ ABR ส่งข้อมูลได้อย่างสม่ำเสมอและต่อเนื่อง ซึ่งในการจำลองเวลาที่ 90 ms เป็นช่วงที่แหล่งกำเนิด VBR ส่งข้อมูลพอดี (อย่างไรก็ตามแหล่งกำเนิดแบบ ABR จะมีปริมาณข้อมูลเพียงพออย่างสมบูรณ์เมื่อผ่านเวลาประมาณ 110 msไปแล้ว)

ดังนั้นจะเห็นว่าค่าอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ที่เกิดจากการจำลองภายใต้อัลกอริทึมชนิดต่าง ๆ จะเพิ่มค่าจนกระทั่งถึงเวลาประมาณ 90 ms หลังจากนั้นค่าอัตราการส่งก็จะถูกควบคุมโดยกระบวนการควบคุมความคับคั่งของ ABR อย่างเดียว ทั้งนี้จะเห็นว่าในกรณีที่ใช้อัลกอริทึม ERICA และ FMMRA มีผลการจำลองหลังเวลาดังกล่าวเหมือนกรณีที่ใช้อัลกอริทึมทั้งสองภายใต้แบบจำลองที่ไม่มี TCP เลย (คือเปรียบเทียบรูป 6.10 (ข) และ 6.12 (ข) กับ 6.2 (ก) และ 6.6 (ก) ตามลำดับ) สำหรับผลที่เกิดจากอัลกอริทึม ERICA+ นั้น เมื่อผ่าน 90 ms ไปแล้วค่า ACR ไม่ได้ลดลงจนถึงค่า Fairshare อย่างที่ควรจะเป็น ทั้งนี้เพราะกระบวนการควบคุมระดับแฉวยของ ERICA+ พยายามเพิ่มปริมาณแฉวย (จากแทบไม่มีอยู่เลย) ให้อยู่ในระดับที่เหมาะสมโดยการเพิ่มอัตราการส่งข้อมูล และเมื่อปริมาณแฉวยมีค่าอยู่ในระดับหนึ่งแล้ว การปรับค่า ACR ก็จะมีลักษณะเหมือนกรณีที่ไม่ว่าน TCP ส่วนอัตราการส่งข้อมูลที่เกิดจากการใช้งานร่วมกับอัลกอริทึม E-FMMRA นั้นต้องอาศัยเวลาในการปรับสู่ค่าคงตัว (เทียบกับผลที่เกิดจากกรณีที่ไม

ผ่าน TCP) มากกว่า ทั้งนี้เพราะค่า ER ที่คำนวณได้จาก E-FMMRA นั้น เป็นการพิจารณาภายใต้หลักการ running weight average ดังนั้นการเพิ่มค่า ACR ที่รุนแรงในช่วงแรก จึงมีผลทำให้ค่า ACR ในช่วงหลังแกว่งมากกว่าปรกติ แต่อย่างไรก็ตาม ผลกระทบดังกล่าวจะค่อย ๆ น้อยลงตามเวลา (ดังรูป 6.13 (ข))

2 ปริมาณแฉวยคองที่อุปกรณ์ต้นทาง (ในที่นี้หมายถึงปริมาณข้อมูลที่สะสมในบัฟเฟอร์ของแหล่งกำเนิด TCP ดังแบบจำลองรูป ก1 ในภาคผนวก ก) พิจารณาตามรูป 6.10 - 6.13 (ค)

ลักษณะการสะสมแฉวยคองที่อุปกรณ์ต้นทางภายใต้การใช้งานร่วมกับอัลกอริทึมแบบต่าง ๆ ของกระบวนการควบคุมความคับคั่งของ ABR มีความคล้ายคลึงกันมาก นั่นคือในช่วงแรกของการเชื่อมต่อแทบจะไม่มีข้อมูลสะสมในแฉวยคองส่วนนี้เลย เนื่องจากอุปกรณ์ต้นทางจะเริ่มมีข้อมูลสะสมในแฉวยคองเมื่อขนาดหน้าต่างความคับคั่งมากกว่า Bandwidth-Delay product (รายละเอียดในการวิเคราะห์ศึกษาได้ใน ภาคผนวก ก) ซึ่งจากการวิเคราะห์ในหัวข้อค่า ACR ของแหล่งกำเนิด แสดงให้เห็นว่าขนาดหน้าต่างความคับคั่งเริ่มมีค่ามากเพียงพอ (ที่จะทำให้การส่งข้อมูลมีประสิทธิภาพ) ที่เวลาประมาณ 90 ms และเมื่อผ่านเวลาดังกล่าวแล้ว อุปกรณ์ส่วนนี้ก็จะเริ่มมีข้อมูลมาสะสมเพิ่มขึ้นเรื่อย ๆ จนกระทั่งระดับแฉวยคองจะมีลักษณะแกว่งตัวในช่วงแคบ ๆ รอบ ๆ แฉวยคองค่าหนึ่ง โดยปริมาณข้อมูลที่สะสมในแฉวยคองส่วนนี้ขึ้นอยู่กับ ขนาดหน้าต่างความคับคั่งที่แหล่งกำเนิด TCP ใช้ และปริมาณข้อมูลที่ส่งได้จริง (มีความสัมพันธ์โดยตรงกับอัตราการส่งข้อมูล) ทั้งนี้ผลการจำลองที่เกิดขึ้น (แสดงดังรูป 6.10 - 6.13 (ค)) จึงมีลักษณะคล้ายกราฟของขนาดหน้าต่างความคับคั่งที่มีการแกว่งของค่าตลอดเวลา ซึ่งการแกว่งนี้เป็นผลมาจากปริมาณข้อมูลที่ส่งได้มีการเปลี่ยนแปลง ทำให้จังหวะการแกว่งที่เกิดขึ้นมีลักษณะคล้ายกราฟของอัตราการส่งข้อมูลแบบกลับหัว (คือ เมื่ออัตราการส่งข้อมูลอยู่ในจังหวะเพิ่มค่า ปริมาณข้อมูลในแฉวยคองส่วนนี้จะอยู่ในจังหวะลดค่า) ซึ่งมีความสอดคล้องกับการวิเคราะห์ในภาคผนวก ก นั่นคือปริมาณข้อมูลที่สะสมในแฉวยคองเกิดจากปริมาณข้อมูลที่แหล่งกำเนิด TCP ส่ง (ขนาดหน้าต่างความคับคั่ง) ลบออกจากความสามารถในการส่งข้อมูลจริง (ในที่นี้คำนวณได้จากผลคูณของอัตราการส่งข้อมูล และค่า RTT)

สำหรับผลการจำลองที่เกิดจากการใช้งานร่วมกับอัลกอริทึม E-FMMRA มีความแตกต่างจากผลการจำลองของอัลกอริทึมอื่น ๆ เล็กน้อย นั่นคือหลังจากที่ขนาดหน้าต่างความคับคั่งถูกจำกัดค่าไว้ที่ขนาดหน้าต่างปลายทางค่าเฉลี่ยของปริมาณแฉวยคองที่เกิดจากผลการทำงานของ E-FMMRA จะเพิ่มเล็กน้อยแทนที่จะมีค่าคงที่เหมือนผลของอัลกอริทึมอื่น ๆ (สังเกตจากปริมาณข้อมูลที่สะสมในแฉวยคองจะแกว่งรอบ ๆ ค่าคงที่ค่าหนึ่ง) ทั้งนี้เกิดจากผลของอัตราการส่งข้อมูลที่คำนวณโดยใช้หลักการ E-FMMRA มีค่ามากเกินไปในช่วงแรก (ซึ่งมีผลต่อค่าอัตราการส่งข้อมูลใน

เวลาถัดมา การวิเคราะห์ผลของการทำงานส่วนนี้ถูกอธิบายอย่างละเอียดแล้วในส่วนของ การวิเคราะห์ค่า ACR) ทำให้มีข้อมูลปริมาณมากถูกส่งในช่วงนี้ (ปริมาณมากเกินกว่าความสามารถ ในการรองรับของโครงข่าย พิจารณาจากปริมาณแฉกคอยที่สะสมในสวิตช์มีค่ามากในช่วงนี้) และเป็นผลให้มีข้อมูลคงค้างในอุปกรณ์ส่วนนี้น้อยกว่าปรกติ แต่เมื่อผลของค่า ER ที่มากผิดปรกติลด น้อยลง อัตราการส่งข้อมูลก็จะลดลงด้วย และเป็นผลให้ระดับแฉกคอยมีปริมาณเฉลี่ยคงที่ ดัง แสดงในรูป 6.13 (ค)

3 ปริมาณแฉกคอยในสวิตช์ ATM พิจารณาตามรูป 6.10 - 6.13 (ง)

หน่วยความจำในสวิตช์ ATM จะเริ่มมีข้อมูลสะสม เมื่อเวลาผ่านไประยะหนึ่ง นั่นคือเมื่อ ขนาดหน้าต่างความคับคั่งมีค่ามากเพียงพอที่จะไม่เกิดขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง (โดยลักษณะการสะสมที่เกิดขึ้นหลังจากช่วงเวลาดังกล่าว จะมีลักษณะคล้ายกับผลการจำลองใน กรณีไม่มี TCP) ซึ่งผลของขีดจำกัดนี้ ทำให้ปริมาณข้อมูลที่สะสมในแฉกคอยส่วนนี้แตกต่างจาก กรณีไม่มี TCP ไม่มากนักน้อยขึ้นอยู่กับอัลกอริทึมที่ใช้ กล่าวคือจากผลการทำงานของอัลกอริทึม ERICA+ ทำให้ปริมาณแฉกคอยเพิ่มมากผิดปรกติในช่วงแรก ๆ ที่หลุดจากขีดจำกัดอันเนื่องมา จากขนาดหน้าต่าง ทั้งนี้เพราะกระบวนการควบคุมระดับแฉกคอยในอัลกอริทึม ERICA+ พยายาม ควบคุมให้เกิดการสะสมข้อมูลในแฉกคอยที่สวิตช์ ATM โดยคำนวณค่า ER ที่เกินกว่าค่า ER ที่ คำนวณได้จากกรณีไม่มี TCP แต่หลังจากเริ่มมีค่าสะสมในอุปกรณ์ส่วนนี้แล้ว การคำนวณค่า ER ในรอบต่อ ๆ ไปก็จะเข้าสู่วงจรเดียวกันกับกรณีไม่มี TCP ทำให้การสะสมข้อมูลในแฉกคอยเข้าสู่ ลักษณะเดียวกันในที่สุด

ส่วนผลการจำลองที่เกิดจากอัลกอริทึม E-FMMRA จะมีการพอกพูนของข้อมูลในอุปกรณ์ ส่วนนี้มากกว่ากรณีอื่น ๆ โดยเฉพาะในช่วงเวลาประมาณ 150 ms ทั้งนี้เกิดจากผลของค่า ER ที่มี ค่ามากเกินไประยะแรกของการเชื่อมต่อ ทำให้ค่า ER ที่ได้ในช่วงต่อมา (ซึ่งเกิดจากการเฉลี่ย ค่าเก่า) มีค่ามากจนโครงข่ายไม่สามารถรองรับข้อมูลที่ส่งออกโดยใช้อัตราการส่งดังกล่าวได้หมด และข้อมูลส่วนที่โครงข่ายรองรับไม่หมดนี้ จะถูกเก็บที่สวิตช์ ATM นี้ด้วย หลังจากนั้นจะเห็นว่า ปริมาณข้อมูลจะค่อย ๆ ลดลง เพราะผลของอัตราการส่งที่มากผิดปรกติในช่วงแรกน้อยลงแล้วนั่นเอง สำหรับการสะสมข้อมูลในสวิตช์ ATM ที่เกิดจากการทำงานร่วมกับอัลกอริทึม ERICA และ FMMRA หลังจากผ่าน 90 ms ไปแล้วจะมีลักษณะเหมือนผลการจำลองในกรณีไม่มี TCP

4 ประสิทธิภาพการใช้งานลิงก์ จากรูป 6.10 - 6.13 (จ)

ผลการจำลองแสดงให้เห็นว่าในช่วงแรก ๆ ของการเชื่อมต่อ ลิงก์จะถูกใช้โดยแหล่งกำเนิด ข้อมูลแบบ VBR เสียเป็นส่วนใหญ่ เพราะแหล่งกำเนิดข้อมูลแบบ ABR ถูกจำกัดปริมาณข้อมูลที่

ส่งจากขนาดหน้าต่างความคับคั่งของ TCP จึงทำให้แหล่งกำเนิดแบบ ABR มีการส่งข้อมูลเพียงเล็กน้อยเท่านั้นในช่วงแรกนี้ (สังเกตจากกราฟผลการจำลองการใช้งานลิงก์ในช่วงนี้ จะเห็นว่าค่าประสิทธิภาพการใช้งานจะแกว่งตัวอยู่ที่ประมาณ 50% ในช่วงที่แหล่งกำเนิด VBR มีการส่งข้อมูลและมีค่าใกล้เคียงศูนย์เมื่อแหล่งกำเนิด VBR หยุดส่ง) และเมื่อขนาดหน้าต่างความคับคั่งมีค่ามากขึ้น แหล่งกำเนิด ABR ก็ส่งข้อมูล และมีการใช้งานลิงก์มากขึ้นตามลำดับ และเมื่อเวลาผ่านไปประมาณ 90 ms การใช้งานลิงก์ก็เริ่มเต็มประสิทธิภาพ และมีความคล้ายคลึงกับผลที่เกิดจากการจำลองในกรณีที่ไม่มี TCP อย่างไรก็ตามเนื่องจากผลของ ACR ในอัลกอริทึมแต่ละแบบมีความแตกต่างกัน ทำให้ผลการจำลองในแง่ประสิทธิภาพการใช้งานลิงก์มีลักษณะที่แตกต่างกันด้วย ทั้งนี้ถ้ามองความสัมพันธ์ระหว่างประสิทธิภาพการใช้งานลิงก์กับปริมาณแฉวยคอยที่สวิตช์ ATM จะเห็นว่าทราบได้ก็ตามที่ยังมีแฉวยคอยสะสมที่สวิตช์ ATM ก็เหมือนโครงข่ายยังมีข้อมูลเพียงพอที่จะเติมเต็มลิงก์ให้มีประสิทธิภาพได้ จากความสัมพันธ์นี้เอง ทำให้สามารถศึกษาประสิทธิภาพการใช้งานลิงก์ จากปริมาณแฉวยคอยที่สวิตช์ ATM ได้

5 ปริมาณข้อมูลที่ปลายทางได้รับ พิจารณาตามรูป 6.10 - 6.13 (ด)

ในช่วงแรกปลายทางได้รับข้อมูลด้วยอัตราน้อยมาก (พิจารณาจากความชันของกราฟที่แสดงผลการจำลอง) ทั้งนี้เกิดจากปริมาณข้อมูลที่ส่งถูกจำกัดด้วยขนาดหน้าต่างความคับคั่งของ TCP หลังจากนั้นอัตราการรับข้อมูลของปลายทางก็จะเพิ่มขึ้นเรื่อย ๆ จนกระทั่งมีค่าค่อนข้างคงที่ซึ่งสอดคล้องกับอัตราการส่งข้อมูล และประสิทธิภาพการใช้งานลิงก์ นอกจากนี้เมื่อพิจารณาความสามารถในการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ทั้งสองตัว จะเห็นว่าแหล่งกำเนิดแบบ ABR สามารถส่งข้อมูลได้อย่างเท่าเทียมกัน (พิจารณาจากรูป 6.10 – 6.13 (ด) จะเห็นว่าเส้นกราฟที่แสดงปริมาณข้อมูลที่ปลายทางได้รับจากแหล่งกำเนิดข้อมูลแบบ ABR ทั้ง 2 ตัวซ้อนทับกัน) เมื่อพิจารณาในแง่ค่าวิสัยสามารถ จะเห็นว่าการทำงานร่วมกับอัลกอริทึม ERICA+ จะมีค่าวิสัยสามารถสูงที่สุด และการทำงานร่วมกับอัลกอริทึม FMMRA จะมีค่าวิสัยสามารถสูงที่น้อยที่สุด

จากผลการจำลองจะเห็นว่าการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม ซึ่งมีการใช้งานร่วมกับอัลกอริทึม ERICA+ นั้นมีช่วงที่การใช้งานลิงก์ต่ำกว่า 100% แคมป์ที่สุด นอกจากนี้เมื่อจำลองการทำงานโดยใช้แหล่งกำเนิดแบบ VBR ที่มีการแกว่งตัวที่รุนแรงตามแบบจำลองแบบที่ 2 ในหัวข้อ 6.1.1 (ซึ่งไม่ได้แสดงไว้ในวิทยานิพนธ์ฉบับนี้) ผลการจำลองที่เกิดจากการใช้งานร่วมกับอัลกอริทึม ERICA+ มีการใช้งานลิงก์ที่มีประสิทธิภาพตลอดช่วงเวลาหลังจากผ่านขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างแล้ว นอกจากนี้เมื่อพิจารณาในแง่

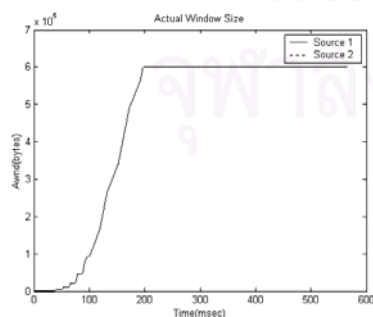
วิจัยสามารถสุทธิที่ปลายทางได้รับ การทำงานร่วมกับอัลกอริทึม ERICA+ สามารถรับข้อมูลได้มากที่สุด และถือว่ามีประสิทธิภาพที่สุด เมื่อเปรียบเทียบกับการใช้งานร่วมกับอัลกอริทึมอื่น ๆ ที่นำมาจำลอง

6.1.3 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ

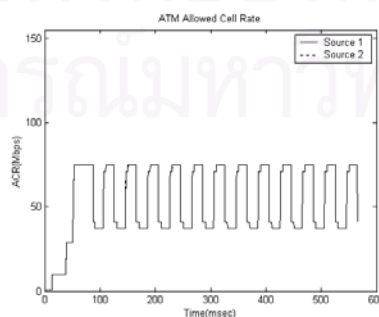
การจำลองชุดนี้เป็นการศึกษาและวิเคราะห์ผลของอัลกอริทึมที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR ที่มีต่อกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และการทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดนี้ ภายใต้กราฟฟิกที่มีการเปลี่ยนแปลงอีกด้วย สำหรับแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.2 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ VBR 1 ตัว และแหล่งกำเนิดข้อมูลแบบ TCP ที่ทำงานร่วมกับแหล่งกำเนิดแบบ ABR อีก 2 ชุด
อุปกรณ์ต่าง ๆ	เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึมชนิดต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.1.2

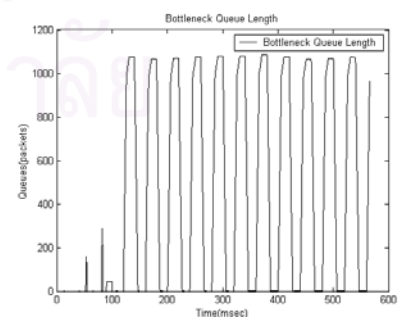
6.1.3.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR



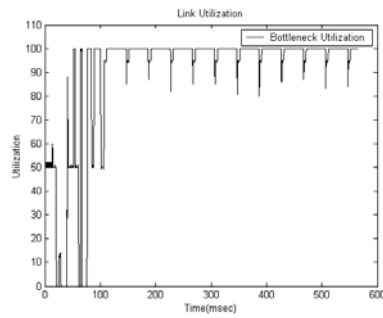
(ก) ขนาดหน้าต่างความคับคั่งเสมือนของ TCP



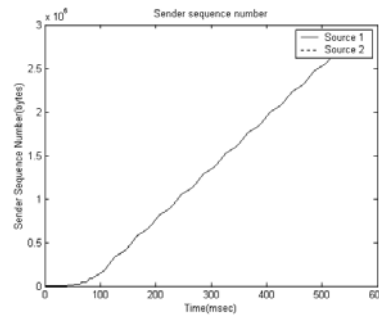
(ข) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

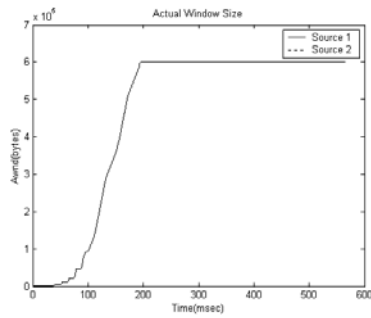
รูปที่ 6.15 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

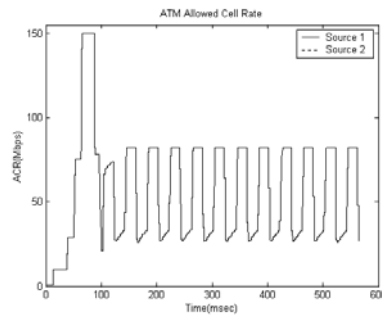
- การเชื่อมต่อที่ 1 วัดได้ 2.930688×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.930688×10^6 ไบต์

จากรูป 6.15 (ง) แสดงขนาดหน้าต่างความคับคั่งเสมือนที่ Observer คำนวณได้ ซึ่งมีลักษณะเหมือนขนาดหน้าต่างที่เกิดจากกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม (ตามรูป 6.10 (ก)) สำหรับผลการจำลองในส่วนอื่น เช่นอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ปริมาณข้อมูลที่สะสมในแถวคอยของสวิตช์ ATM ประสิทธิภาพการใช้งานลิงก์ และปริมาณข้อมูลที่ปลายทางได้รับ (แสดงดังรูป 6.15 (ข) 6.15 (ค) 6.15 (ง) และ 6.15 (จ) ตามลำดับ) มีลักษณะเหมือนผลการจำลองที่ได้จากกระบวนการควบคุมการไหลของ TCP แบบดั้งเดิม (ในหัวข้อ 6.1.2.1 ตามรูปที่ 6.10 (ข) 6.10 (ค) 6.10 (ง) และ 6.10 (จ) นั้นเอง) เพียงแต่ดูเหมือนว่าผลการจำลองที่ได้จากกรณีนี้มีการตอบสนองที่เร็วกว่า อีกทั้งค่าวิสัยสามารถสุทธิที่วัดได้ก็มีค่ามากกว่าด้วย

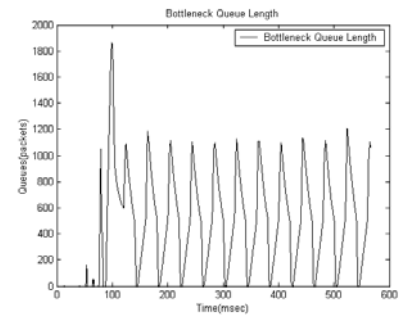
6.1.3.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งเสมือนของ



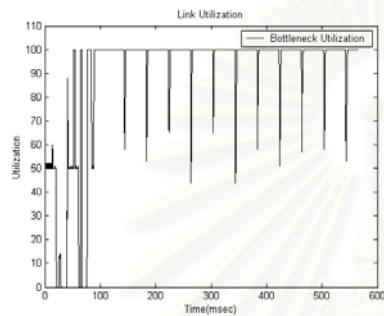
(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด



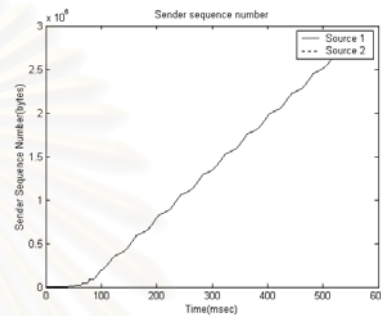
(ค) ปริมาณแควคอยในสวิตช์ ATM

TCP

แบบ ABR



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

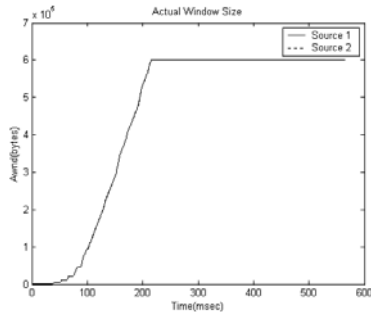
รูปที่ 6.16 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

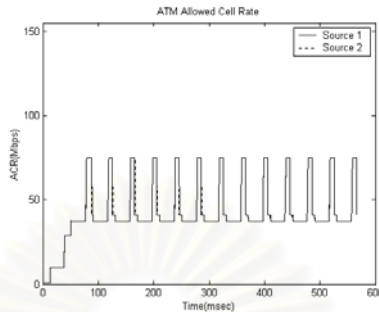
- การเชื่อมต่อที่ 1 วัดได้ 2.938880×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.938880×10^6 ไบต์

ผลการจำลองการทำงานที่ได้จากการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม ERICA+ มีลักษณะเหมือนผลที่ได้จากการจำลองการทำงานภายใต้กระบวนการควบคุมการไหลของ TCP โดยใช้อัลกอริทึมดั้งเดิม และอัลกอริทึม ERICA+ (ในหัวข้อ 6.1.2.2) เพียงแต่อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ในรูป 6.16 (ข) มีการเพิ่มค่าเร็วกว่า (ค่าที่ค่าประมาณ 150 Mbps นานกว่า) รวมทั้งปริมาณข้อมูลที่สะสมในสวิตช์ช่วงแรกมีค่ามากกว่า (นั่นคืออยู่ที่ระดับประมาณ 1,900 เซลล์)

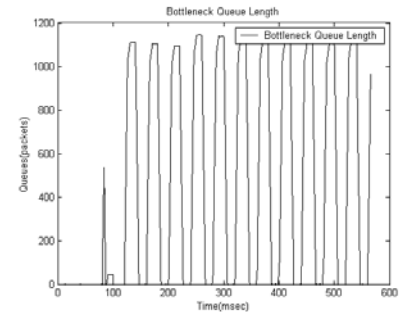
6.1.3.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งเสมือนของ



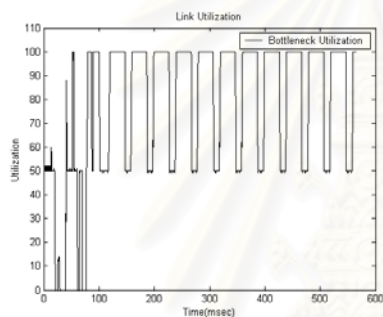
(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด



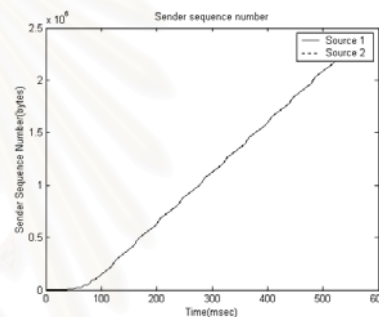
(ค) ปริมาณแฉวคอยในสวิตช์ ATM

TCP

แบบ ABR



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

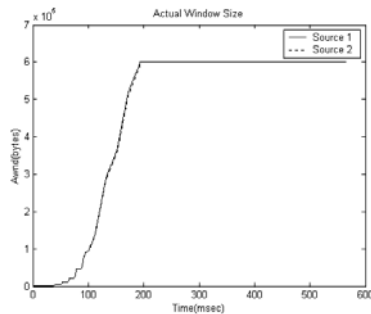
รูปที่ 6.17 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

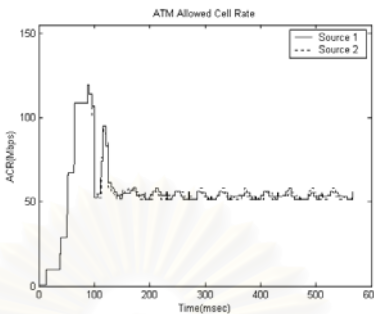
- การเชื่อมต่อที่ 1 วัดได้ 2.413568×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.415616×10^6 ไบต์

ผลการจำลองโดยรวมของกรณีนี้มีลักษณะเหมือนผลที่ได้จากการจำลองโดยใช้หลักการดั้งเดิม ร่วมกับอัลกอริทึม FMMRA (ในหัวข้อ 6.1.2.3) ยกเว้นช่วงแรกที่จะปรับค่าอัตราการส่งข้อมูลได้เร็วกว่าเล็กน้อย ทำให้ค่าวิสัยสามารถสุทธิของแต่ละการเชื่อมต่อที่วัดได้มีค่ามากกว่าในกรณีใช้หลักการดั้งเดิม

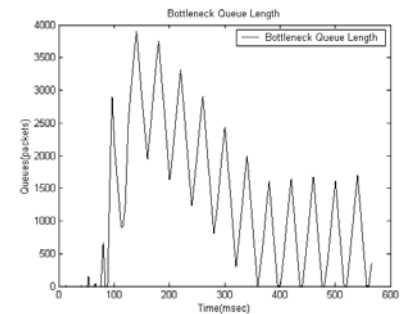
6.1.3.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งเสมือนของ

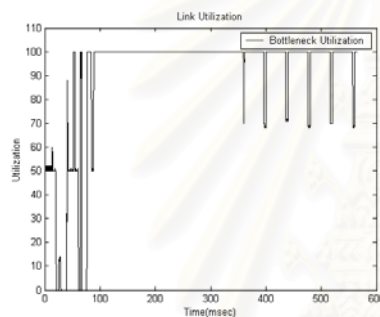


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด



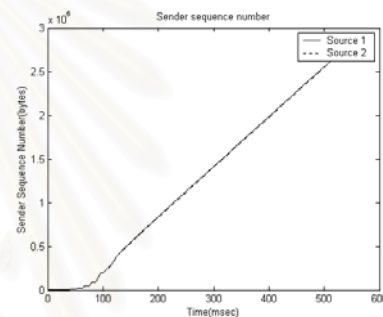
(ค) ปริมาณแฉวคอยในสวิตช์ ATM

TCP



(ง) ประสิทธิภาพการใช้งานลิงก์

แบบ ABR



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.18 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 2.929664×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.917376×10^6 ไบต์

ผลการจำลองการทำงานที่ได้จากการจำลองภายใต้เงื่อนไขนี้ มีลักษณะคล้ายผลการจำลองที่ได้จากการจำลองการทำงานภายใต้กระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม (หัวข้อ 6.1.2.4) แต่มีลักษณะแตกต่างกันเล็กน้อย นั่นคือผลการจำลองที่ได้จากหัวข้อนี้ มีการตอบสนองที่เร็วกว่า รวมทั้งค่าวิสัยสามารถสุทธิที่ปลายทางได้รับมีค่ามากกว่า

6.1.3.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ และโดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR

ผลการจำลองที่ได้จากการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูล โดยใช้หลักการกักเก็บสัญญาณตอบกลับนี้ มีลักษณะคล้ายกับผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม ทั้งนี้เพราะหลักการกักเก็บสัญญาณตอบกลับ จะใช้การทำงานในส่วนที่เรียกว่า Observer ซึ่งมีหน้าที่จำลองการเพิ่มขนาดหน้าต่างความคับคั่งของ TCP ซึ่งเหมือนหลักการที่ใช้ในหลักการแบบดั้งเดิม (คือหลักการเพิ่มแบบ Slow Start) จึงทำให้ผลที่เกิดจากหลักการกักเก็บสัญญาณตอบกลับคล้ายผลการจำลองที่เกิดจากการใช้กระบวนการควบคุมการไหลของ TCP แบบดั้งเดิม กล่าวคือผลการจำลองของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ ยังคงเห็นผลของขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างความคับคั่ง แต่ข้อดีของหลักการนี้ คือไม่มีการสะสมข้อมูลที่อุปสรรคต้นทาง (ตามหลักการของการกักเก็บสัญญาณตอบกลับ คือใช้เพียงหน่วยความจำเล็กน้อยในการเก็บข้อมูลของสัญญาณตอบกลับแทน) หรือกล่าวอีกนัยหนึ่งคือหลักการนี้ไม่ก่อให้เกิดขีดจำกัดจากอัตราการส่ง

ดังนั้นการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ ที่มีการใช้งานร่วมกับอัลกอริทึม ERICA+ ก็ยังคงเป็นอัลกอริทึมที่สามารถส่งข้อมูลได้มากที่สุด พิจารณาจากค่าวิสัยสามารถสุทธิที่ปลายทางได้รับนั่นเอง

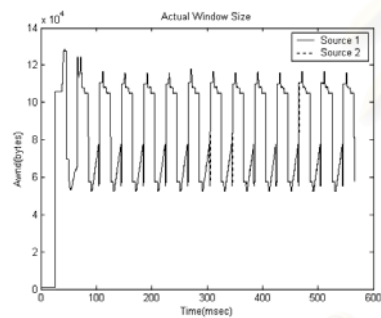
6.1.4 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ

การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง ภายใต้การใช้อัลกอริทึมชนิดต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR เป็นการนำหลักการคำนวณหาขนาดหน้าต่างที่เหมาะสมกับสภาพทราฟฟิกของโครงข่าย ซึ่งการหาขนาดหน้าต่างที่เหมาะสมดังกล่าวจำเป็นต้องใช้พารามิเตอร์ 2 ตัว คืออัตราการส่งข้อมูลที่เหมาะสม (นักวิจัยผู้คิดค้นหลักการนี้เสนอให้ใช้ค่า ER ที่บันทึกในเซลล์ RM มาคำนวณ) ส่วนพารามิเตอร์อีกตัวคือเวลาที่ใช้ในการเดินทางไปกลับของข้อมูล (RTT) โดยในการจำลองในหัวข้อนี้ใช้ค่า RTT จากการจับเวลาตั้งแต่ข้อมูลถูกปล่อยออกจากแหล่งกำเนิด TCP จนกระทั่งแหล่ง

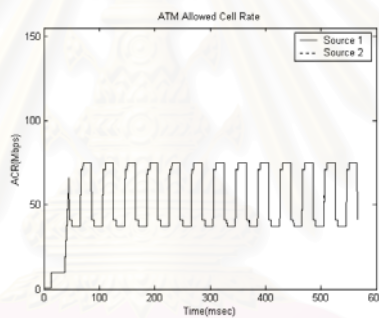
กำเนิดได้รับสัญญาณตอบกลับของเซกเมนต์ข้อมูลนั้น สำหรับแบบจำลอง และค่าพารามิเตอร์ที่ใช้ในการจำลองนี้ สามารถแจกแจงได้ดังต่อไปนี้

แบบจำลอง	ตามรูปที่ 5.2 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ VBR 1 ตัว และแหล่งกำเนิดข้อมูลแบบ TCP ที่ทำงานร่วมกับแหล่งกำเนิดแบบ ABR อีก 2 ชุด
อุปกรณ์ต่าง ๆ	เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึมชนิดต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.1.2

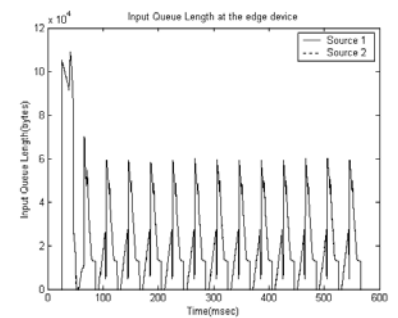
6.1.4.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

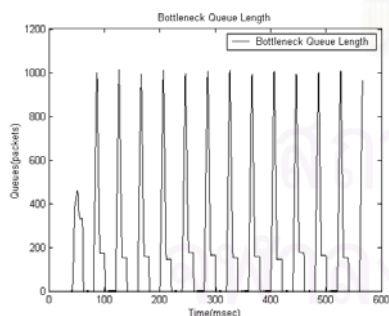


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

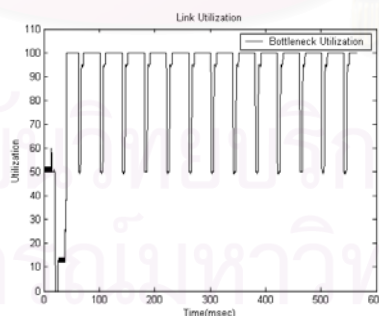


(ค) ปริมาณแควคยที่อุปกรณ์ต้นทาง

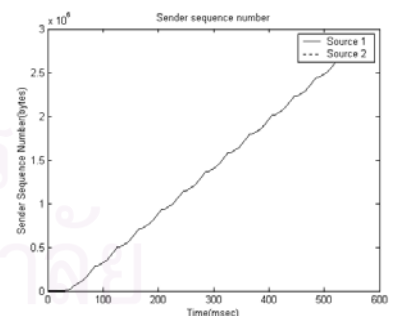
แบบ ABR



(ง) ปริมาณแควคยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

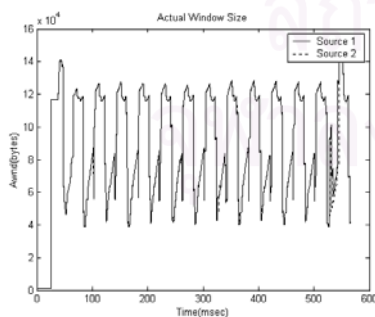
รูปที่ 6.19 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

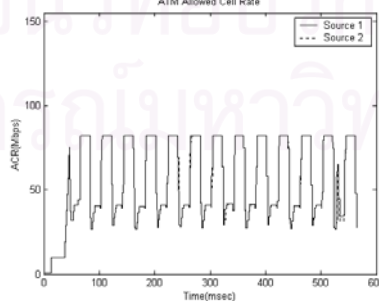
- การเชื่อมต่อที่ 1 วัดได้ 2.875392×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.875392×10^6 ไบต์

จากรูป 6.19 (ก) แสดงขนาดหน้าต่างความคับคั่งของ TCP ที่ใช้ จะเห็นว่าในช่วงเริ่มต้นของการส่งข้อมูล ขนาดหน้าต่างความคับคั่งมีค่าน้อยมาก แต่เมื่อเวลาผ่านไปประมาณ 20 ms ขนาดหน้าต่างความคับคั่งก็ปรับเพิ่มขึ้นอย่างรวดเร็ว จนมีค่าสูงสุดถึง 13×10^4 ไบต์ หลังจากนั้นค่าขนาดหน้าต่างความคับคั่งก็ลดลงอย่างฉับพลัน จนมีค่าประมาณ 57 กิโลไบต์ แล้วค่อย ๆ เพิ่มค่าทีละนิด จนเมื่อผ่านไปอีกประมาณ 20 ms ขนาดหน้าต่างความคับคั่งก็เพิ่มขึ้นอย่างรวดเร็วอีกครั้ง หลังจากนั้นค่าขนาดหน้าต่างความคับคั่งก็จะแกว่งขึ้น ๆ ลง ๆ ในลักษณะนี้ไปเรื่อย ๆ สำหรับอัตราการส่งข้อมูลของ ABR ในช่วงแรกของการเชื่อมต่อ จะค่อย ๆ เพิ่มค่าจนถึงจุดทำงาน หลังจากนั้นผลการจำลองที่ได้จะมีลักษณะเหมือนผลที่ได้จากการจำลองโดยไม่มี TCP ส่วนปริมาณข้อมูลที่ค้างในอุปกรณ์ต้นทางก็มีลักษณะแกว่งขึ้นลงเช่นกัน (ดังรูป 6.19 (ค)) โดยในช่วงแรกจะแกว่งไปถึงประมาณ 11×10^4 ไบต์ เลยทีเดียว แต่หลังจากนั้นปริมาณแควคอยก็จะแกว่งตัวอยู่ที่ค่าระหว่าง 0 ถึง 6×10^4 ไบต์ สำหรับระดับแควคอยในสวิตช์ ATM และประสิทธิภาพการใช้งานลิงก์ (แสดงดังรูป 6.19 (ง) และ 6.19 (จ) ตามลำดับ) จะเห็นว่ามีลักษณะคล้ายผลที่ได้จากกรณีที่ไม่มี TCP เพียงแต่ระดับแควคอยในสวิตช์ ATM ลดปริมาณเร็วกว่า ส่วนลิงก์จะถูกใช้งานอย่างไม่เต็มประสิทธิภาพน้อยกว่ากรณีที่ไม่มี TCP

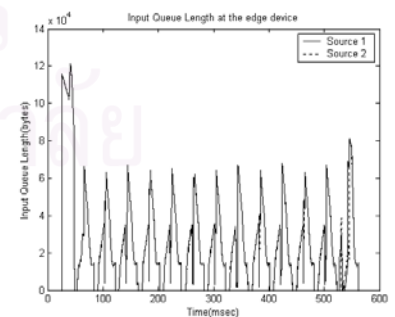
6.1.4.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

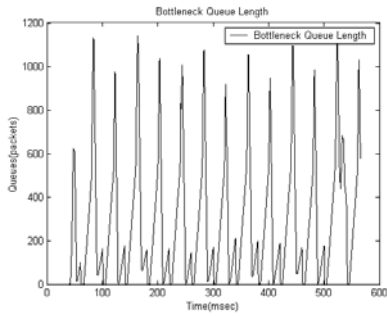


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

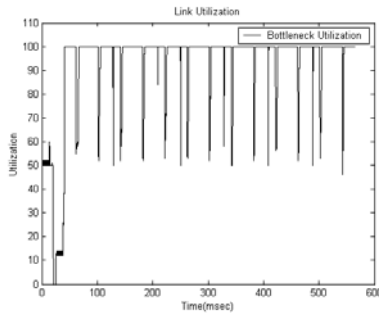


(ค) ปริมาณแควคอยที่อุปกรณ์ต้นทาง

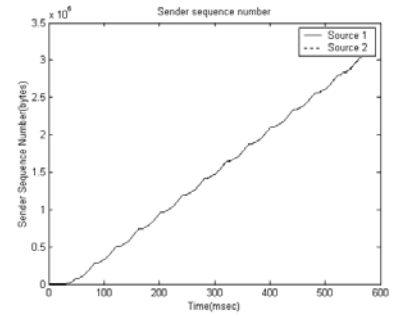
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

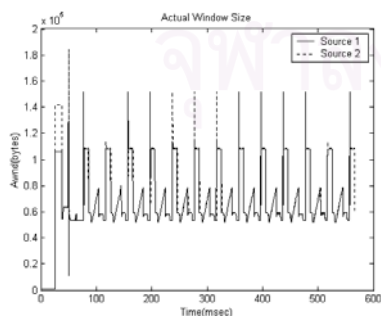
รูปที่ 6.20 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างต่าง และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุม
ความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

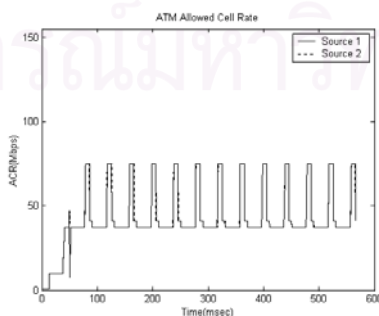
- การเชื่อมต่อที่ 1 วัดได้ 3.034112×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 3.020800×10^6 ไบต์

ผลการจำลองการทำงานที่ใช้อัลกอริทึม ERICA+ นั้นมีลักษณะคล้ายผลการจำลองของ ERICA ทั้งนี้ค่าอัตราการส่งข้อมูลที่ได้เกิดจากกระบวนการควบคุมระดับแควคอยของ ERICA+ จึงทำให้มีค่าไม่เท่ากับที่ได้จาก ERICA ซะทีเดียว นอกจากนี้เมื่อพิจารณาปริมาณแควคอยที่สวิตช์ ATM จะเห็นว่าแควคอยมีค่าเท่ากับศูนย์เป็นช่วงสั้น ๆ เท่านั้น และนั่นส่งผลให้ลิงก์ถูกใช้งานไม่เต็มประสิทธิภาพเพียงช่วงเวลาสั้น ๆ เท่านั้น

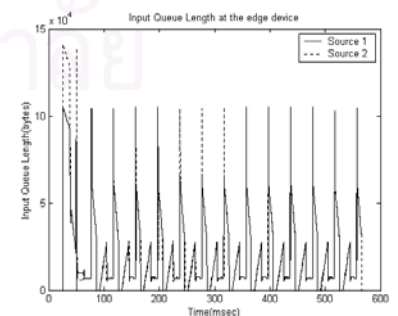
6.1.4.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างต่าง และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

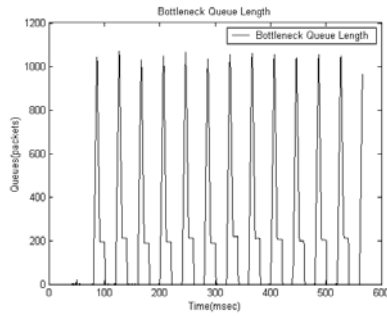


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

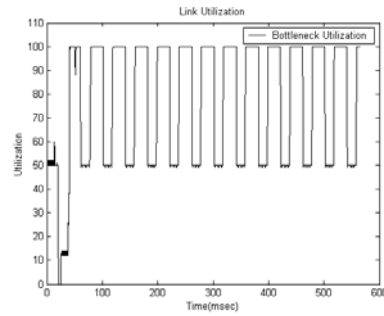


(ค) ปริมาณแควคอยที่อุปกรณ์ต้นทาง

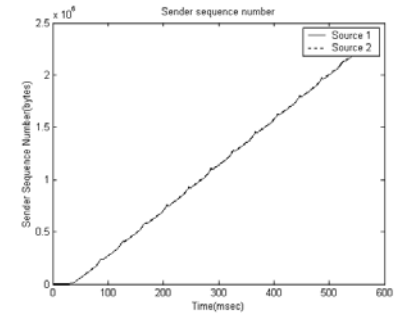
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

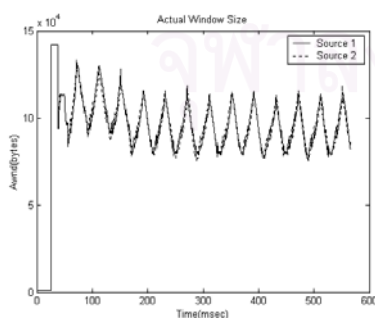
รูปที่ 6.21 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างต่าง และใช้อัลกอริทึม FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

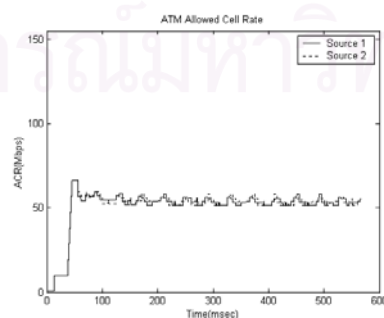
- การเชื่อมต่อที่ 1 วัดได้ 2.309120×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.318336×10^6 ไบต์

เมื่อพิจารณาผลการจำลองของอัลกอริทึม FMMRA ภายใต้กระบวนการควบคุมการไหลแบบนี้ จะเห็นว่าลักษณะโดยรวมเหมือนผลที่ได้จากอัลกอริทึม FMMRA ภายใต้แบบจำลองที่ไม่มี TCP อย่างไรก็ตามระดับแควคอยที่สวิตช์ ATM ที่เกิดจากการจำลองในกรณีนี้ จะลดลงเร็วกว่า (ดังรูป 6.21 (ง)) ส่งผลให้ประสิทธิภาพการใช้งานลิงก์มีค่าน้อยกว่ากรณีที่ไม่มี TCP หรือกรณีอื่น ๆ ด้วยเหตุนี้ทำให้ค่าวิสัยสามารถสุทธิที่ได้จากกรณีนี้ น้อยกว่าที่ได้จากกรณีอื่น ๆ

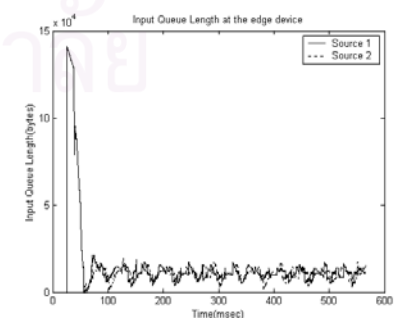
6.1.4.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างต่าง และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

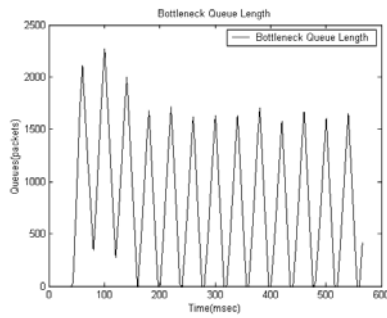


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

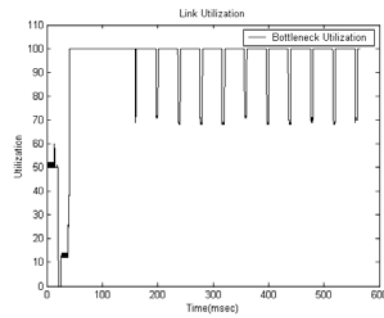


(ค) ปริมาณแควคอยที่อุปกรณ์ต้นทาง

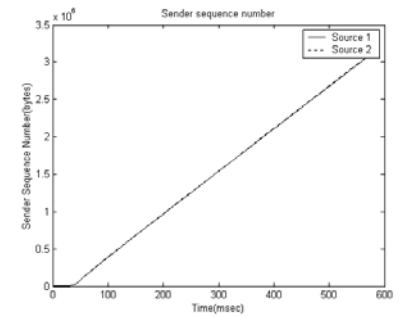
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.22 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และใช้อัลกอริทึม E-FMMRA ในกระบวนการ
ควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 3.049472×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 3.057664×10^6 ไบต์

สำหรับผลการจำลองที่เกิดจากการใช้อัลกอริทึม E-FMMRA จะมีความแตกต่างจากผล
การจำลองที่ใช้อัลกอริทึมอื่น ๆ ตามเหตุผลที่ได้กล่าวไปบ้างแล้ว อย่างไรก็ตามเนื่องจากการใช้ค่า
ER เฉลี่ยของ E-FMMRA นี้เองทำให้ผลการจำลองการทำงานของกระบวนการควบคุมการไหล
ของ TCP โดยใช้การปรับปรุ่ขนาดหน้าต่างนี้ ไม่ค่อยมีความแตกต่างจากการจำลองที่ไม่มี TCP

6.1.4.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ่ค่าขนาดหน้าต่าง และโดยใช้อัลกอริ ทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR

จะเห็นว่าผลการจำลองที่ได้ในกรณีนี้ มีความแตกต่างจากผลการจำลองที่ได้จากกรณีที่ใช้
หลักการดั้งเดิม และหลักการกักเก็บสัญญาณตอบกลับ โดยเฉพาะอย่างยิ่งในช่วงแรกของการ
เชื่อมต่อ ขนาดหน้าต่างความคับคั่งเพิ่มทันทีทันใดโดยไม่ได้ทำงานภายใต้หลักการ Slow Start ทั้ง
นี้ค่าขนาดหน้าต่างที่ใช้มีค่ามากเพียงพอที่จะส่งข้อมูลได้อย่างมีประสิทธิภาพ เนื่องจากการปรับ
ปรุ่ค่าขนาดหน้าต่างนี้ ใช้พื้นฐานความรู้เรื่อง Bandwidth-Delay product มาแปลงเป็นค่าขนาด
หน้าต่างความคับคั่ง โดยการคำนวณจากการนำค่า ER มาคูณกับค่า RTT ซึ่งค่า RTT ที่ใช้สำหรับ
กรณีนี้เป็นค่าที่ได้จากการจับเวลาตั้งแต่เซกเมนต์ข้อมูลถูกปล่อยออกจากแหล่งกำเนิด TCP จน
กระทั่งแหล่งกำเนิดได้รับสัญญาณตอบกลับของเซกเมนต์ข้อมูลนั้น เนื่องจากหลักการนี้พยายาม

ใช้ค่าขนาดหน้าต่างใกล้เคียงกับค่า Bandwidth-Delay product เพื่อหลีกเลี่ยงปัญหาที่เกิดจากขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง รวมทั้งปริมาณแฉวคอยจำนวนมากที่อุปกรณ์ต้นทางปัญหาดังกล่าวไม่ถูกกำจัดเสียทีเดียว ทั้งนี้รายละเอียดของปัญหาที่มีการวิเคราะห์ดังต่อไปนี้

1 ค่า ACR ที่แหล่งกำเนิดข้อมูลแบบ ABR พิจารณาตามรูป 6.19 – 6.22 (ข)

ค่าอัตราการส่งข้อมูลที่ได้จากการจำลองในกรณีนี้ จะมีลักษณะคล้ายกับที่เกิดจากการจำลองโดยไม่มี TCP ยกเว้นช่วงแรก ๆ อัตราการส่งที่เกิดจากการจำลองแบบนี้จะเพิ่มช้ากว่า ทั้งนี้เพราะการส่งข้อมูลของแหล่งกำเนิด TCP ในรอบแรก ต้องใช้ขนาดหน้าต่างความคับคั่งที่มีค่าน้อยอยู่ เนื่องจากแหล่งกำเนิด TCP ยังไม่ได้รับสัญญาณตอบกลับมาจากโครงข่ายเลยแม้แพ็กเก็ตเดียว ทำให้แหล่งกำเนิดไม่สามารถรู้ค่า RTT ของการเชื่อมต่อได้ เป็นผลให้ไม่สามารถคำนวณค่าขนาดหน้าต่างตามหลักการนี้ได้ จากขนาดหน้าต่างที่น้อยนี้เอง ทำให้ปริมาณข้อมูลที่ควบคุมโดยแหล่งกำเนิด TCP ในช่วงแรกนี้มีค่าน้อยกว่าความสามารถในการส่งของแหล่งกำเนิดแบบ ABR ดังนั้นอัตราการส่งของ ABR ในช่วงนี้จึงน้อยกว่าในกรณีที่ไม่มี TCP (เมื่อปริมาณข้อมูลน้อย เซลล์ RM ก็จะมีค่าน้อย และเป็นผลให้การปรับค่า ACR ช้าตามไปด้วย) แต่หลังจากที่แหล่งกำเนิด TCP คำนวณค่าขนาดหน้าต่างความคับคั่งจากผลคูณของค่า ER และ RTT ได้แล้ว ปริมาณข้อมูลที่แหล่งกำเนิด ABR ได้รับก็มากเพียงพอที่จะทำให้ไม่มีผลต่อการปรับค่า ACR ดังนั้นค่า ACR ที่ได้จึงมีลักษณะคล้ายกับผลที่ได้จากการจำลองโดยไม่มี TCP อย่างไรก็ตามอัตราการส่งข้อมูลที่เกิดจากการใช้งานร่วมกับอัลกอริทึม ERICA+ มีความแตกต่างจากกรณีที่จำลองโดยไม่มี TCP เล็กน้อย นั่นคือในช่วงที่แหล่งกำเนิดแบบ VBR ส่งข้อมูล ค่า ER ที่ได้จากการทำงานของอัลกอริทึม ERICA+ จะมีค่ามากกว่าค่าที่คำนวณได้จากกรณีที่ไม่มี TCP ทั้งนี้เพราะปริมาณข้อมูลที่สะสมในแฉวคอยที่สวิตช์ ATM ลดมากกว่าและเร็วกว่าในกรณีที่ไม่มี TCP ทำให้กระบวนการควบคุมระดับแฉวคอยในอัลกอริทึม ERICA+ ต้องเพิ่มอัตราการส่งข้อมูลในช่วงนั้นอีก (สำหรับเหตุผลของการเปลี่ยนแปลงปริมาณข้อมูลที่สะสมในแฉวคอยที่สวิตช์ ATM นั้นจะมีการวิเคราะห์ในส่วนต่อไป)

2 ขนาดหน้าต่างความคับคั่งของ TCP ปริมาณแฉวคอยในอุปกรณ์ต้นทาง และปริมาณแฉวคอยที่สวิตช์ ATM พิจารณาตามรูป 6.19 – 6.22 (ก) (ค) และ (ง) ตามลำดับ

เนื่องจากขนาดหน้าต่างความคับคั่งของแหล่งกำเนิดแบบ TCP มีความสัมพันธ์กับปริมาณข้อมูลที่สะสมในแฉวคอยที่อุปกรณ์ต้นทาง และที่สวิตช์ ATM ในการวิเคราะห์ส่วนนี้จึงจะวิเคราะห์ผลการจำลองทั้ง 3 ส่วนไปพร้อม ๆ กัน โดยจะเริ่มวิเคราะห์จากขนาดหน้าต่างความคับคั่ง ซึ่งจะเห็นว่ามีการแกว่งเหมือนการแกว่งของค่า ACR ทั้งนี้เพราะขนาดหน้าต่างความคับคั่ง

คั้งที่ใช้ในหลักการนี้เป็นผลการคำนวณจากค่า ER ซึ่งมีความสัมพันธ์โดยตรงกับค่า ACR และเนื่องจากค่า ER ที่คำนวณได้จากอัลกอริทึมต่าง ๆ ในกรณีนี้มีค่าใกล้เคียงกับค่า Fairshare ทำให้ค่าขนาดหน้าต่างมีค่าใกล้เคียงกับค่า Bandwidth-Delay product (ซึ่งวิเคราะห์การหาค่าดังกล่าวไว้แล้วในหัวข้อ 6.1.2) อย่างไรก็ตามค่าขนาดหน้าต่างที่ใช้ในการจำลองนี้มีค่าไม่เท่ากับค่า Bandwidth-Delay product เสียทีเดียว กล่าวคือในช่วงแรกของการเชื่อมต่อค่าขนาดหน้าต่างเริ่มที่ค่าน้อย (ตามที่ได้วิเคราะห์ไว้ข้างต้น) จนกระทั่งได้รับข้อมูลตอบกลับจากโครงข่ายแล้ว ขนาดหน้าต่างจะเพิ่มขึ้นทันที ในขณะที่เดียวกันค่า ACR ที่ใช้กับแหล่งกำเนิดแบบ ABR กลับค่อย ๆ เพิ่มขึ้นทีละขั้นแปรผันตามค่า RIF จึงทำให้มีข้อมูลบางส่วนตกค้าง และถูกสะสมในหน่วยความจำที่อุปกรณ์ต้นทาง (ดังรูป 6.19 – 6.22 (ค)) ด้วยผลของการที่ข้อมูลค้างอยู่ในแถวคอยส่วนนี้ ทำให้ค่า RTT ที่วัดได้มีค่ามากกว่าเวลาที่เกิดจากเคลื่อนที่ของข้อมูลจริง แต่จะรวมเวลาที่ติดค้างในหน่วยความจำด้วย ดังนั้นขนาดหน้าต่างความคับคั่งที่คำนวณจากค่า RTT ดังกล่าวจึงมีค่ามากขึ้น (พิจารณาได้จากการเกิดการกระตุกขึ้นของค่าขนาดหน้าต่างความคับคั่งในช่วงแรกของการเชื่อมต่อ)

หลังจากนั้นเมื่อแหล่งกำเนิดแบบ VBR มีการส่งข้อมูล ค่า ER ที่คำนวณได้จะลดลงทันที ในขณะที่ค่า ACR ของแหล่งกำเนิดแบบ ABR จะค่อย ๆ ลดลง ตามความสัมพันธ์ของค่า RDF ทำให้ขนาดหน้าต่างที่เกิดขึ้นช่วงนี้มีค่าไม่เพียงพอต่อความต้องการของแหล่งกำเนิดแบบ ABR แต่เนื่องจากมีข้อมูลที่ถูกเก็บไว้ในหน่วยความจำที่อุปกรณ์ต้นทางตั้งแต่ช่วงแรก ทำให้แหล่งกำเนิดแบบ ABR นำข้อมูลส่วนนั้นไปใช้ ซึ่งถือว่าการระบายข้อมูลในแถวคอยออกสู่โครงข่าย ดังนั้นระดับแถวคอยที่อุปกรณ์ต้นทางจึงลดลง แต่ข้อมูลดังกล่าวมีค่าไม่มากพอที่จะเติมเต็มความต้องการดังกล่าว ทำให้ข้อมูลในแถวคอยส่วนนี้ถูกระบายจนหมด และก่อให้เกิดภาวะขาดแคลนข้อมูลเป็นระยะสั้น ๆ ดังรูป 6.19 – 6.22 (ค) ในทำนองเดียวกันกับกรณีที่ไม่มี TCP ที่ได้วิเคราะห์ไว้ในหัวข้อ 6.1.1 นั่นคือผลจากการลดค่า ACR ที่ไม่ทันทั่วทั้ง ทำให้มีข้อมูลถูกส่งออกสู่โครงข่ายมากเกินไปจนถึงไม่สามารถรองรับได้ ทำให้ข้อมูลส่วนเกินดังกล่าวถูกกักเก็บในสวิตช์ ATM ซึ่งผลของการติดอยู่ในแถวคอยนี้ ทำให้ค่า RTT ที่ใช้ในการคำนวณรอบต่อมามีค่ามากอีกครั้ง และส่งผลให้ขนาดหน้าต่างความคับคั่งมีค่ามากด้วย (ถ้าพิจารณาจากรูป 6.19 – 6.22 (ก) ผลดังกล่าวแสดงในส่วนที่ค่อย ๆ เพิ่มขึ้นในช่วงที่มีค่าประมาณ 60 kbyte นั้นเอง) และเมื่อค่าขนาดหน้าต่างความคับคั่งมากขึ้นก็จะเกิดการสะสมที่หน่วยความจำในอุปกรณ์ต้นทางอีกครั้ง (ช่วงเพิ่มหลังจากมีค่าเป็นศูนย์) แต่เนื่องจากการส่งข้อมูลมีการเกิดภาวะขาดแคลนข้อมูลเป็นระยะสั้น ๆ (ดังที่กล่าวไว้ข้างต้น) ทำให้สวิตช์พยายามเติมเต็มถึงก็โดยการระบายข้อมูลในส่วนของสวิตช์ ATM (จริง ๆ ข้อมูลส่วนนี้จะถูกระบายเพราะช่วงเหลือมของการเพิ่มค่า ACR ตามที่ได้วิเคราะห์ในหัวข้อ 6.1.1 แต่ด้วยเหตุผลข้างต้นทำให้ข้อมูลส่วนนี้ถูกระบายเร็วกว่ากรณีไม่มี TCP) ผลจากการที่ไม่มีแถวคอย

สะสมอยู่เลยนี้ ทำให้ค่า RTT ที่วัดได้มีค่าเท่ากับค่าที่ได้จากการวัดเวลาที่ใช้ในการเคลื่อนที่ไปกลับของข้อมูลเท่านั้น (ไม่รวมเวลาที่ติดอยู่ในบัฟเฟอร์) ดังนั้นขนาดหน้าต่างความคับคั่งที่คำนวณได้จึงมีค่าเท่ากับค่า Bandwidth-Delay product แต่เป็นเพียงช่วงแคบ ๆ เท่านั้น (สังเกตจากช่วงเล็ก ๆ หลังจากที่มีขนาดหน้าต่างความคับคั่งมีการเพิ่ม) เนื่องจากแหล่งกำเนิด VBR หยุดส่งข้อมูลในเวลาต่อมา ทำให้ค่า ER ที่คำนวณได้มีค่ามากอีกครั้ง (ขนาดหน้าต่างจึงมีค่ามากขึ้นด้วย) แต่ค่า ACR ไม่ได้ปรับตัวเร็วขนาดนั้น จึงทำให้เกิดการสะสมที่หน่วยความจำในอุปกรณ์ต้นทาง และเป็นผลให้ค่าขนาดหน้าต่างรอบต่อไปมีค่ามากเกินไปมีค่ามากเกินไป ระดับแถวคอยจะลดลงเมื่อแหล่งกำเนิดแบบ VBR ส่งข้อมูล จากการทำงานดังกล่าวก่อให้เกิดลักษณะกราฟที่แกว่งขึ้นลงดังผลการจำลอง

อย่างไรก็ดีเนื่องจากหลักการคำนวณค่า ER ที่แตกต่างกันของอัลกอริทึมแบบต่าง ๆ ทำให้ผลการจำลองมีความแตกต่างกัน แต่เพียงเล็กน้อยเท่านั้น ยกเว้นผลการจำลองที่เกิดจากการใช้งานร่วมกับอัลกอริทึม E-FMMRA ทั้งนี้เพราะค่า ER ที่ใช้เป็นค่าที่ได้จากการเฉลี่ยค่าเก่าด้วย

3 ประสิทธิภาพการใช้งานลิงก์ พิจารณาตามรูป 6.19 – 6.22 (จ)

ผลการจำลองที่ได้แทบจะไม่มีอะไรแตกต่างไปจากผลการจำลองที่เกิดจากกรณีที่ไม่ได้มี TCP ยกเว้นช่วงแรกของการเชื่อมต่อที่มีการใช้งานลิงก์ไม่เต็มประสิทธิภาพมากนักว่า (ทั้งนี้มีเหตุผลเดียวกันกับที่ได้วิเคราะห์ไปแล้วในส่วนก่อนหน้า) นอกจากนี้การใช้งานลิงก์ในช่วงหลังยังมีช่วงเวลาที่ใช้งานไม่เต็ม 100% นานกว่าด้วย ทั้งนี้เป็นผลของขนาดหน้าต่างลดเร็ว ในขณะที่แหล่งกำเนิดข้อมูลแบบ ABR ลดอัตราการส่งช้ากว่า จนทำให้ข้อมูลที่เคยสะสมไว้ในหน่วยความจำต้นทางถูกนำมาใช้จนหมด จนเกิดภาวะขาดแคลนข้อมูลในระยะสั้น ๆ ขึ้นมา จนทำให้ต้องระบายข้อมูลในหน่วยความจำที่สวิตช์ ATM เพื่อเติมเต็มลิงก์ จนปริมาณข้อมูลที่สะสมมีค่าน้อยลง ทำให้เมื่อแหล่งกำเนิดแบบ VBR หยุดส่ง และแหล่งกำเนิดแบบ ABR เพิ่มอัตราการส่งไม่ทัน ข้อมูลที่จะมีเติมเต็มลิงก์ช่วงนี้ก็จะมีปริมาณน้อยลง จนทำให้ไม่พอในที่สุด และทำให้เกิดการใช้งานลิงก์ที่ไม่เต็มประสิทธิภาพมากขึ้นนั่นเอง

4 ปริมาณข้อมูลที่ปลายทางได้รับ พิจารณาตามรูป 6.19 – 6.22 (ข)

ปริมาณข้อมูลที่ปลายทางได้รับในช่วงแรกมีค่าน้อย แต่หลังจากนั้นข้อมูลที่มาถึงปลายทางก็มีปริมาณมากขึ้น และมาถึงของข้อมูลดังกล่าวมีอัตราที่ค่อนข้างคงที่ (พิจารณาจากความชันของกราฟที่มีค่าค่อนข้างคงที่)

จากผลการจำลองจะเห็นว่าการทำงานร่วมกับอัลกอริทึม E-FMMRA ถูกผลกระทบภายใต้การควบคุมโดยการปรับปรุงขนาดหน้าต่างน้อยที่สุด (พิจารณาเปรียบเทียบกับผลการจำลองในกรณี

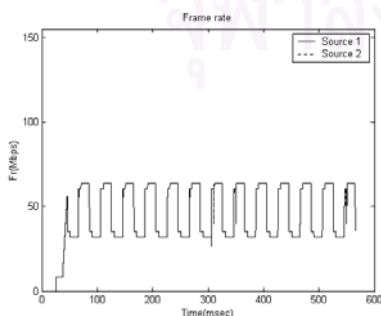
ไม่มี TCP) อย่างไรก็ตามอัลกอริทึม ERICA+ ก็เป็นอีกอัลกอริทึมหนึ่งที่ได้รับผลกระทบในการทำงานร่วมกันน้อย

6.1.5 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้งานร่วมกับอัลกอริทึมในกระบวนการควบคุมความคับคั่งของ ABR แบบต่าง ๆ

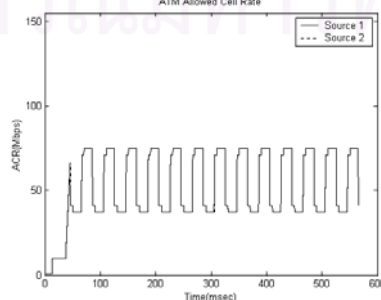
การจำลองการทำงานชุดนี้เป็นการศึกษาและวิเคราะห์ผลของอัลกอริทึมที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR ที่มีต่อกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และเนื่องจากแบบจำลองที่ใช้เป็นแบบจำลองที่มีทราฟฟิกที่มีการเปลี่ยนแปลงตลอดเวลา ดังนั้นการจำลองการทำงานในกรณีนี้ยังเป็นการทดสอบประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดนี้ ภายใต้ทราฟฟิกที่มีการเปลี่ยนแปลงอีกด้วย สำหรับแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.2 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ VBR 1 ตัว และแหล่งกำเนิดข้อมูลแบบ TCP ที่ทำงานร่วมกับแหล่งกำเนิดแบบ ABR อีก 2 ชุด
อุปกรณ์ต่าง ๆ	เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึมชนิดต่าง ๆ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.1.2

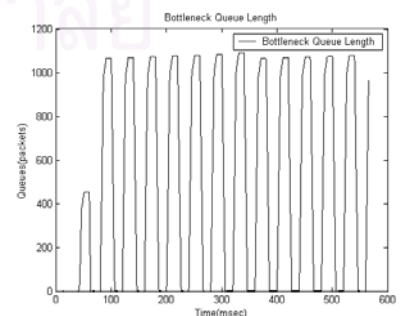
6.1.5.1 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA ในกระบวนการควบคุมความคับคั่งของ ABR



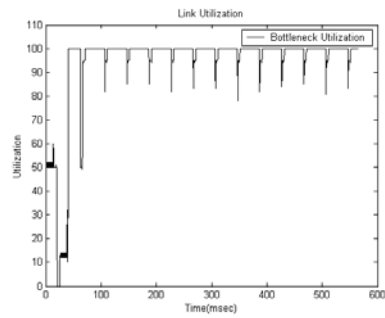
(ก) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ TCP



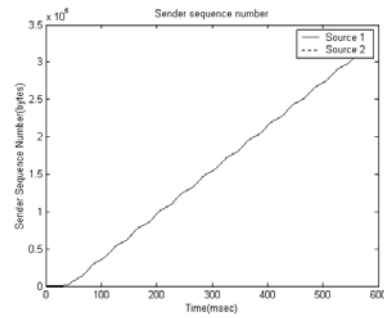
(ข) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR



(ค) ปริมาณแควคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

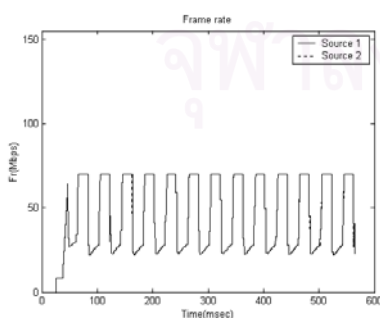
รูปที่ 6.23 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา การส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA ใน กระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 3.110912×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 3.111936×10^6 ไบต์

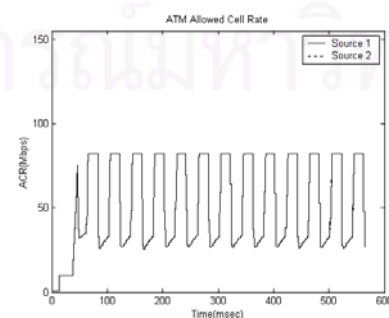
ค่าอัตราการส่งข้อมูลของแหล่งกำเนิดในชั้น TCP (ดังรูป 6.23 (ก)) มีลักษณะเหมือน อัตราการส่งข้อมูลแบบเซลล์ในชั้น ATM (ดังรูป 6.23 (ข)) เพียงแต่มีค่าน้อยกว่าเท่านั้น สำหรับผล การจำลองในส่วนอื่น ๆ ที่ได้มีลักษณะเหมือนผลการจำลองที่ได้จากกรณีที่ไม่มี TCP ยกเว้นช่วง แรก ๆ ของการเชื่อมต่อที่อัตราการส่งข้อมูล ไม่ว่าจะพิจารณาจากชั้น TCP หรือ ATM จะปรับตัว ช้ากว่า อีกทั้งประสิทธิภาพการใช้งานลิงก์ในช่วงนี้ไม่เต็มประสิทธิภาพมากกว่าด้วย (รูป 6.23 (ง))

6.1.5.2 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิด ควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR



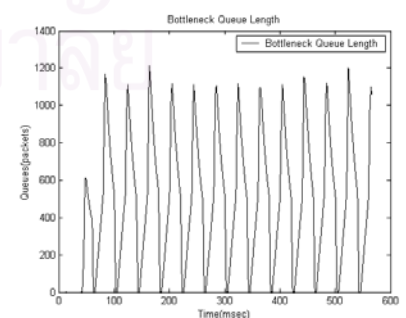
(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด

แบบ TCP

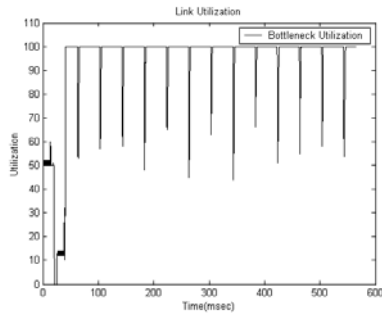


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

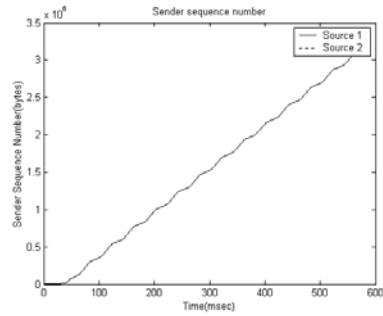
แบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

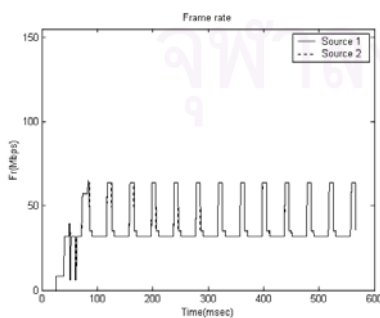
รูปที่ 6.24 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม ERICA+ ใน
กระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 3.138560×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 3.137536×10^6 ไบต์

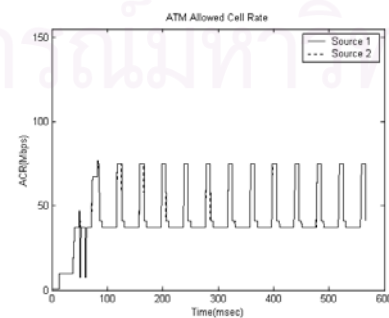
ผลการจำลองโดยรวมมีลักษณะเหมือนผลการจำลองที่ได้จากกรณีที่ไม่มี TCP ยกเว้นช่วง
เริ่มการเชื่อมต่อ ที่อัตราการส่งข้อมูลปรับตัวช้ากว่า (ดังรูป 6.24 (ข)) อีกทั้งประสิทธิภาพการใ้
งานลิงก์ที่ไม่เต็ม 100% ในช่วงแรกกว้างกว่าด้วย ดังรูป 6.24 (ง) อย่างไรก็ตามผลการจำลองค่า
อัตราการส่งข้อมูลในชั้น TCP ที่ได้มีลักษณะเหมือนค่าอัตราการส่งในชั้น ATM เพียงแต่ค่าอัตรา
การส่งในชั้น TCP มีค่าน้อยกว่า นอกจากนี้ค่าวิสัยสามารถสุทธิที่ได้ในกรณีนี้มีค่ามากกว่ากรณีอื่น

**6.1.5.3 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิด
ควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม
FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR**



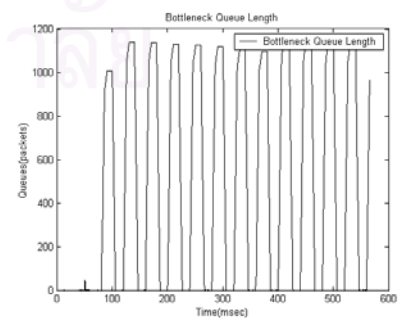
(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด

แบบ TCP

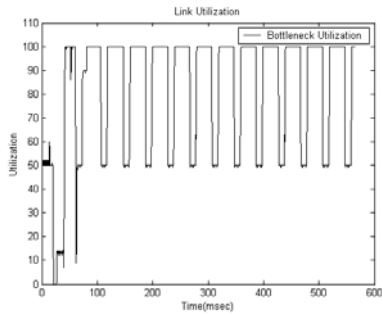


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

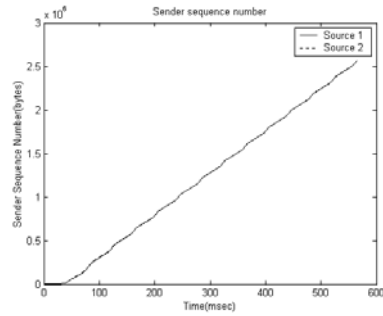
แบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

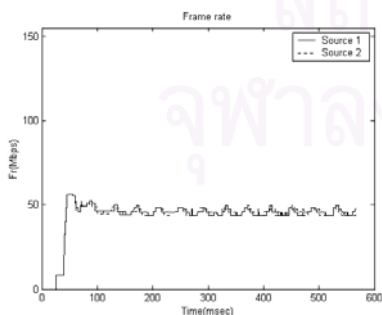
รูปที่ 6.25 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา การส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม FMMRA ใน กระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 2.571264×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 2.569216×10^6 ไบต์

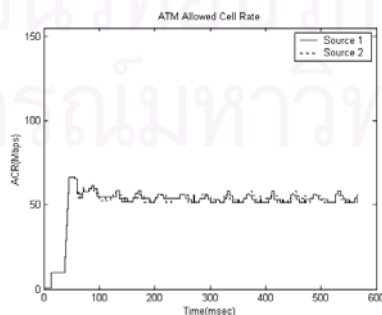
ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ที่ให้หลักการแปลง อัตราการส่งแบบเซลล์เป็นเฟรม มีแนวโน้มคล้ายผลการจำลองที่ไม่ได้ส่งข้อมูลผ่าน TCP (ยกเว้น ช่วงแรกที่แหล่งกำเนิดข้อมูลต้องอาศัยเวลาระยะหนึ่ง เพื่อปรับให้อัตราการส่งข้อมูลเข้าสู่จุด ทำงาน ทำให้ผลการจำลองที่ได้ในช่วงแรกไม่ค่อยมีประสิทธิภาพนัก)

6.1.5.4 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิด ควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม E-FMMRA ในกระบวนการควบคุมความคับคั่งของ ABR



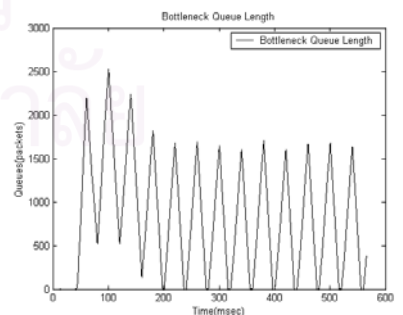
(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด

แบบ TCP

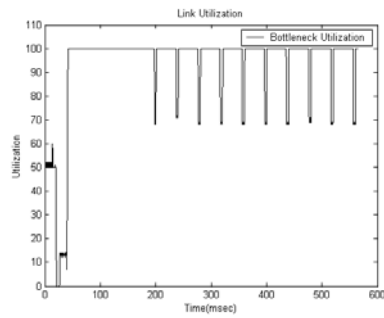


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

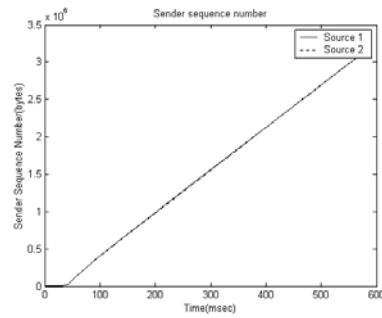
แบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.26 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา การส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้อัลกอริทึม E-FMMRA ใน กระบวนการควบคุมความคับคั่งของ ABR

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อที่ 1 วัดได้ 3.066880×10^6 ไบต์
- การเชื่อมต่อที่ 2 วัดได้ 3.066880×10^6 ไบต์

การทำงานของกระบวนการควบคุมการไหลของ TCP ที่ใช้หลักการแปลงอัตราการส่งแบบ เซลล์เป็นเฟรม และใช้งานร่วมกับอัลกอริทึม E-FMMRA มีผลการจำลองคล้ายกับกรณีที่ไม่ได้ส่ง ข้อมูลผ่าน TCP (ในหัวข้อ 6.1.1.4) ยกเว้นช่วงแรกที่แหล่งกำเนิดข้อมูลต้องอาศัยเวลาระยะหนึ่ง เพื่อปรับให้อัตราการส่งข้อมูลเข้าสู่จุดทำงาน ทำให้ผลการจำลองที่ได้ในช่วงแรกไม่ค่อยมีประสิทธิภาพนัก

6.1.5.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และ โดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR

ผลการจำลองที่ได้จากการทำงานของกระบวนการควบคุมการไหลของ TCP โดยการใช้การ แปลงอัตราการส่งแบบเซลล์เป็นเฟรม มีลักษณะคล้ายคลึงกับผลการจำลองที่ได้จากการจำลอง ภายใต้อัตราการส่งแบบไม่มี TCP (ในหัวข้อ 6.1.1) ยกเว้นเพียงช่วงแรกของการเชื่อมต่อ ที่แหล่ง กำเนิดข้อมูลสามารถส่งข้อมูลได้น้อยกว่า (พิจารณาจากอัตราการส่งที่เพิ่มขึ้นช้ากว่า เปรียบเทียบ จากรูป 6.23 – 6.26 (ข) กับรูป 6.2 (ก) 6.4 (ก) 6.6 (ก) และ 6.8 (ก) ตามลำดับ) ทั้งนี้เพราะในช่วง เริ่มส่งข้อมูล แหล่งกำเนิด TCP จะส่งข้อมูลเพียงแพ็กเก็ตเดียว ด้วยเหตุผลที่ว่า ในการส่งข้อมูล ของแหล่งกำเนิดข้อมูล TCP ต้องคำนึงถึงความสามารถในการรับข้อมูลของอุปกรณ์ปลายทาง ทั้ง

นี้ถ้าแหล่งกำเนิด TCP ส่งข้อมูลโดยพิจารณาจากข้อมูลที่ได้จากการทำงานในชั้น ATM อย่างเดียว โดยไม่คำนึงถึงข้อจำกัดดังกล่าว อัตราการส่งข้อมูลที่แหล่งกำเนิดใช้อาจไม่เหมาะสมกับความสามารถในการทำงานของ TCP ที่ปลายทาง และเป็นสาเหตุให้เกิดความเสียหายต่อข้อมูลที่ส่งได้ ทั้งนี้ความสามารถในการรับข้อมูลของอุปกรณ์ปลายทางมีความสัมพันธ์โดยตรงกับค่าขนาดหน้าต่างด้านรับ (W_{rec}) ดังนั้นแหล่งกำเนิดจึงจำเป็นต้องรอรับสัญญาณตอบกลับจากปลายทาง เพื่อนำค่าขนาดหน้าต่างด้านรับมาเป็นตัวกำหนดขอบเขตของอัตราการส่งข้อมูลต่อไป

เมื่อพิจารณาอัตราการส่งข้อมูลแบบเฟรมที่ใช้ในชั้น TCP (จากรูป 6.23 –6.26 (ก)) จะเห็นว่าผลการจำลองที่ได้มีลักษณะ รวมไปถึงจังหวะการแกว่งค่าเหมือนอัตราการส่งข้อมูลแบบเซลล์ในชั้น ATM (จากรูป 6.23 –6.26 (ข)) เพียงแต่มีค่าน้อยกว่าเท่านั้น ทั้งนี้เป็นผลมาจากกระบวนการทำงานในชั้น ATM มีการสร้างข้อมูลบางส่วนเพิ่มขึ้นมา ไม่ว่าจะเป็นข้อมูลในส่วนเฮดเดอร์ของทุก ๆ เซลล์ หรือข้อมูลไร้ความหมายที่เกิดจากกระบวนการตัดแบ่งเซลล์ (เกิดจากในกรณีที่เฟรมข้อมูลที่ได้มาจากชั้น TCP ไม่สามารถแบ่งเป็นเซลล์ได้พอดี กระบวนการสร้างเซลล์จึงจำเป็นต้องเพิ่มข้อมูลไร้ความหมายเข้าไปในเซลล์ เพื่อให้เซลล์มีขนาดคงที่) นอกจากนี้เนื่องจากแหล่งกำเนิดที่พิจารณาเป็นแหล่งกำเนิดแบบ ABR ดังนั้นทุก ๆ Nrm เซลล์ที่ส่งออกไป จะต้องมีการเพิ่มเซลล์ RM (ซึ่งสร้างขึ้นใหม่ โดยแหล่งกำเนิดแบบ ABR) หนึ่งเซลล์ส่งออกไปด้วย และด้วยเหตุนี้เองทำให้อัตราการส่งข้อมูลที่ใช้ในชั้น TCP จึงมีค่าน้อยกว่าอัตราการส่งข้อมูลในชั้น ATM

เนื่องจากว่าอัตราการส่งข้อมูลแบบเฟรมที่ใช้ในชั้น TCP เป็นผลการคำนวณที่แปลงมาจากค่าอัตราการส่งข้อมูลที่แหล่งกำเนิดข้อมูลในชั้น ATM ต้องการใช้ ดังนั้นปริมาณข้อมูลที่แหล่งกำเนิด TCP ส่งออกมาจึงเป็นปริมาณที่เพียงพอสำหรับความต้องการของแหล่งกำเนิดแบบ ABR (อย่างไรก็ตามปริมาณข้อมูลที่แหล่งกำเนิด TCP ส่งออกมาจะต้องอยู่ภายใต้ขอบเขตของความสามารถในการรับข้อมูลของอุปกรณ์ปลายทาง ดังที่ได้กล่าวไว้ข้างต้น) เปรียบเสมือนการควบคุมปริมาณทราฟฟิกในโครงข่าย ทำโดยกระบวนการควบคุมความคับคั่งของ ABR เพียงอย่างเดียว ดังนั้นผลการจำลองหลังจากผ่านช่วงแรกไปแล้วจึงมีลักษณะเหมือนผลการจำลองที่เกิดจากกรณีไม่มี TCP

เมื่อพิจารณาการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และใช้ร่วมกับอัลกอริทึมทั้ง 4 ที่ใช้ในการจำลองแบบ จะเห็นว่าอัลกอริทึม ERICA+ จัดว่าเป็นอัลกอริทึมที่สามารถควบคุมทราฟฟิกได้อย่างมีประสิทธิภาพที่สุด ทั้งนี้พิจารณาจากประสิทธิภาพการใช้งานลิงก์ และค่าวิสัยสามารถสุทธิที่ปลายทางได้รับมีค่ามาก รวมไปถึงปริมาณแควคอยในสวิตช์ ATM ก็มีค่าไม่มากหรือน้อยจนเกินไป ดังนั้นในการจำลองการทำงานในแง่อื่น ๆ จะทดสอบกับอัลกอริทึม ERICA+ เท่านั้น

6.2 การจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

ในหัวข้อนี้จะศึกษา วิเคราะห์ และเปรียบเทียบผลการทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ ภายใต้แบบจำลองที่ประกอบด้วยการเชื่อมต่อหลาย ๆ การเชื่อมต่อ โดยแต่ละการเชื่อมต่อมีระยะห่างระหว่างอุปกรณ์ต้นทางและปลายทางแตกต่างกัน กล่าวอีกนัยหนึ่งคือ เป็นการศึกษาผลของความยาวสายส่ง (ซึ่งมีความสัมพันธ์โดยตรงกับเวลาที่ใช้ในการเดินทางไป และกลับของข้อมูล หรือ RTT) ที่มีต่อประสิทธิภาพการทำงานของโครงข่าย ภายใต้การทำงานของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ

เนื่องจากการจำลองครั้งนี้ต้องการทดสอบผลที่เกิดจากค่า RTT ที่แตกต่างกันเท่านั้น จึงไม่จำเป็นต้องใช้แบบจำลองที่มีลักษณะซับซ้อน ดังนั้นแบบจำลองแบบ Peer to peer จึงถูกเลือกขึ้นมาใช้ ซึ่งแบบจำลองแบบนี้เป็นการต่อตรงระหว่างอุปกรณ์ต้นทาง สวิตช์ และอุปกรณ์ปลายทางเท่านั้น สำหรับการทดสอบจะใช้อัลกอริทึม ERICA+ ในกระบวนการควบคุมความคับคั่งของ ABR (จากผลการจำลองในหัวข้อที่แล้ว สามารถสรุปได้ว่าอัลกอริทึม ERICA+ เป็นอัลกอริทึมที่มีประสิทธิภาพ และเหมาะสมสำหรับนำมาใช้ในการเปรียบเทียบที่สุด) โดยการจำลองจะเริ่มจากการศึกษาผลที่เกิดจากกระบวนการควบคุมความคับคั่งเพียงอย่างเดียวก่อน แล้วจึงพิจารณาผลที่เกิดจากกระบวนการควบคุมการไหลของ TCP

6.2.1 การจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกันโดยไม่มีการส่งข้อมูลผ่าน TCP

การจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกันโดยไม่มี การส่งข้อมูลผ่าน TCP เป็นการจำลองที่จะพิจารณาแหล่งกำเนิดที่มีข้อมูลมากเพียงพอที่จะส่งข้อมูลได้อย่างต่อเนื่อง และแหล่งกำเนิดดังกล่าวจะส่งข้อมูลผ่านมายังอุปกรณ์ในชั้น ATM โดยตรง นั่นคือไม่ส่งผ่านกระบวนการในชั้น TCP สำหรับการจำลองจะทดสอบภายใต้แบบจำลองและพารามิเตอร์ดังต่อไปนี้

แบบจำลอง	ตามรูปที่ 5.1 (ข) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ ABR 15 ตัว แบ่งออกเป็น 3 กลุ่ม (กลุ่มละ 5 ตัว) แยกแยะแต่ละกลุ่มตามความยาวของสายส่งที่ต่างกัน
แหล่งกำเนิดข้อมูล ABR	PCR = 149.76 Mbps; MCR = 0 Mbps; ICR = 0.5 Mbps RIF = 1/16 RDF = 1/16 (พารามิเตอร์ที่ใช้กับแหล่งกำเนิดข้อมูลแบบ ABR ถูกตั้งเหมือนในหัวข้อ 6.1)
ความยาวของสายส่ง	$x_1 = x_2 = x_3 = x_4 = x_5 = 3 \text{ km}$; $x_6 = x_7 = x_8 = x_9 = x_{10} = 300 \text{ km}$; $x_{11} = x_{12} = x_{13} = x_{14} = x_{15} = 3000 \text{ km}$; $y = 1000 \text{ km}$ (จากรูป 5.1 (ข))

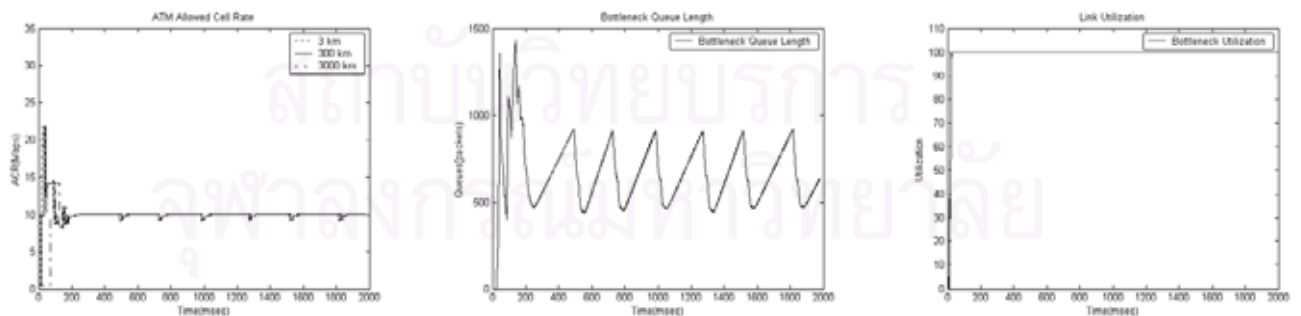
หมายเหตุ เนื่องจากแบบจำลองที่ใช้ในหัวข้อนี้ สามารถแยกแยะการเชื่อมต่อออกเป็น 3 กลุ่มใหญ่ ๆ ตามความยาวของสายส่ง และเพื่อความสะดวกในการอ้างถึง ไม่ว่าจะเป็นในส่วนของ การอธิบายผลการจำลอง หรือการวิเคราะห์การทำงาน จึงจะเรียกกลุ่มของการเชื่อมต่อ ตามความยาวของสายส่งระหว่างแหล่งกำเนิดและสวิตช์ ดังนี้

- กลุ่ม 3 km คือกลุ่มของการเชื่อมต่อที่มีความยาวของสายส่งระหว่างแหล่งกำเนิดและสวิตช์ เท่ากับ 3 km
- กลุ่ม 300 km คือกลุ่มของการเชื่อมต่อที่มีความยาวของสายส่งระหว่างแหล่งกำเนิดและสวิตช์ เท่ากับ 300 km
- กลุ่ม 3000 km คือกลุ่มของการเชื่อมต่อที่มีความยาวของสายส่งระหว่างแหล่งกำเนิดและสวิตช์ เท่ากับ 3000 km

ส่วนค่าพารามิเตอร์ที่ใช้ในอัลกอริทึม ERICA+ สำหรับการจำลองครั้งนี้ มีการตั้งค่าดังต่อไปนี้

อัลกอริทึม ERICA+ ที่สวิตช์ ATM	$AI = 5 \text{ ms}$; Target Utilization = 1.0 (เหมือนกับที่ใช้กับอัลกอริทึม ERICA) $a = 1.15$; $b = 1.05$; $T_0 = 1.5$; $QDLF = 0.5^\#$ $^\#$ สำหรับพารามิเตอร์ที่ใช้กับกระบวนการควบคุมระดับแถวคอยนี้ เป็นค่าที่ผู้วิจัย ERICA+ แนะนำให้ใช้ []
---------------------------------	--

ผลการจำลองและวิเคราะห์การทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกันโดย
ไม่ผ่าน TCP



(ก) อัตราการส่งข้อมูลของแหล่งกำเนิด

(ข) ปริมาณแถวคอยในสวิตช์ ATM

(ค) ประสิทธิภาพการใช้งานลิงก์

แบบ ABR

รูปที่ 6.27 ผลการจำลองการทำงานบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยไม่ผ่าน TCP

ผลการจำลองการทำงานกระบวนการควบคุมความคับคั่ง ที่ใช้อัลกอริทึม ERICA+ บนแบบจำลองที่มีความยาวสายส่งขนาดต่าง ๆ กัน จะเห็นว่า การเชื่อมต่อในกลุ่ม 3 km ปรับอัตราการส่งข้อมูลได้เร็วกว่าการเชื่อมต่อในกลุ่มอื่น ๆ ทั้งนี้เพราะความยาวของสายส่งข้อมูลที่ใช้ในการเชื่อมต่อในกลุ่ม 3 km นี้มีค่าน้อยที่สุด ซึ่งคือเวลาที่ข้อมูลใช้ในการเดินทางไปและกลับมีค่าน้อยที่สุดด้วย ดังนั้นกระบวนการควบคุมความคับคั่งของ ABR ที่ใช้หลักการกลไกย้อนกลับในการทำงาน จึงสามารถปรับอัตราการส่งได้เร็ว ด้วยเหตุนี้เองการเชื่อมต่อในกลุ่ม 3 km จึงสามารถปรับอัตราการส่งข้อมูลได้เร็วกว่า การเชื่อมต่อในกลุ่ม 300 km และเร็วกว่า การเชื่อมต่อในกลุ่ม 3000 km ด้วย

เมื่อพิจารณาอัตราการส่งข้อมูลที่ได้จากการจำลอง จะเห็นว่าในช่วงแรกอัตราการส่งข้อมูลของทุกแหล่งกำเนิด จะค่อย ๆ เพิ่มจากค่า ICR โดยการปรับค่ายังคงใช้หลักการพื้นฐาน ซึ่งสอดคล้องกับค่า RIF (จากเหตุผลนี้เอง ทำให้ในช่วงเวลานี้ถึงก็ไม่ได้ถูกใช้งานอย่างเต็มประสิทธิภาพ ดังรูป 6.27 (ค)) ด้วยเหตุผลที่กล่าวข้างต้น อัตราการส่งข้อมูลของการเชื่อมต่อในกลุ่ม 3 km จึงมีการปรับค่าก่อน เนื่องจากการเชื่อมต่อในกลุ่ม 3 km เป็นการเชื่อมต่อกลุ่มแรกที่มีการติดต่อกับสวิทช์ ดังนั้นสวิทช์จึงคำนวณค่า ER ที่เหมาะสมจากการเชื่อมต่อกลุ่มนี้เท่านั้น นั่นคือค่า Fairshare ที่ได้จะมีค่าประมาณ 30 Mbps (คำนวณจาก $149.76 / 5$) ทำให้แหล่งกำเนิดของการเชื่อมต่อกลุ่มนี้พยายามปรับค่า ACR ให้เท่ากับค่าดังกล่าว แต่เนื่องจากแหล่งกำเนิดไม่ได้เพิ่มค่า ACR อย่างทันทีทันใด จึงทำให้ระหว่างการเพิ่มค่านี้อัตราการส่งข้อมูลก็ได้รับการติดต่อจากกลุ่ม 300 km ทำให้ค่า Fairshare ที่คำนวณจาก ERICA+ เปลี่ยนไปมีค่าประมาณ 15 Mbps (คำนวณจาก $149.76 / 10$) แต่เนื่องจาก ERICA+ มีกระบวนการควบคุมระดับแถวคอยทำงานร่วมอยู่ด้วย ดังนั้นค่า ER ที่ได้จึงมีค่ามากกว่า 15 Mbps เพื่อเพิ่มระดับแถวคอยให้อยู่ในระดับที่ ERICA+ คิดว่าเหมาะสม ด้วยเหตุนี้เองจึงทำให้อัตราการส่งข้อมูลของการเชื่อมต่อทั้ง 2 กลุ่ม เพิ่มค่าจนมีค่าประมาณ 22 Mbps

หลังจากนั้น อัตราการส่งข้อมูลก็ลดลงจนมีค่าน้อยกว่า 15 Mbps เล็กน้อย ทั้งนี้เพื่อพยายามระบายข้อมูลในแถวคอย ที่เริ่มมีปริมาณมากเกินไป อย่างไรก็ตามการเชื่อมต่อในกลุ่ม 3000 km ก็เริ่มมีบทบาทในกระบวนการคำนวณของ ERICA+ ทำให้ค่า Fairshare ที่ได้มีค่าประมาณ 10 Mbps (คำนวณจาก $149.76 / 15$) ในขณะที่แหล่งกำเนิดในกลุ่ม 3000 km เพิ่มอัตราการส่งขึ้นเรื่อย ๆ แหล่งกำเนิดในกลุ่ม 3 km และ 300 km ก็พยายามปรับลด ในช่วงนี้เองปริมาณแถวคอยก็เพิ่มขึ้นอีกครั้ง ทำให้อัตราการส่งที่คำนวณได้จาก ERICA+ มีค่าน้อยลง (ดังรูปที่ 6.27 (ก)) หลังจากนั้นอัตราการส่งของแหล่งกำเนิดทุกตัว ก็จะถูกปรับให้มีค่ามากกว่าค่า Fairshare เล็กน้อย เพื่อรักษาระดับข้อมูลในแถวคอยให้ค่อย ๆ เพิ่มขึ้น (ตามความต้องการของอัลกอริทึม ERICA+) แต่เมื่อระดับแถวคอยเพิ่มจนเกินระดับความต้องการแล้ว อัตราการส่งข้อมูล

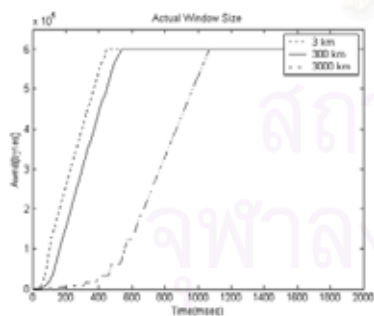
จะถูกปรับให้มีค่าน้อยลงเป็นช่วงเวลาสั้น ๆ เพื่อลดระดับแถวคอย หลังจากนั้นการทำงานในลักษณะนี้ก็จะเกิดขึ้นตลอดช่วงเวลาการเชื่อมต่อ

6.2.2 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

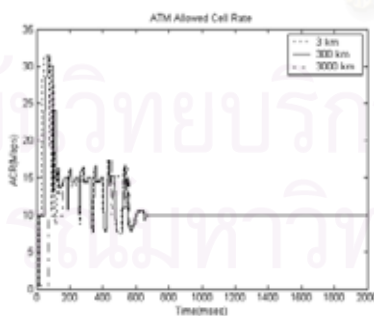
การจำลองการทำงานในหัวข้อนี้ เป็นการศึกษาและวิเคราะห์ผลของความยาวสายส่งที่มีขนาดต่างกัน ต่อกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม โดยแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.1 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ TCP ทำงานร่วมกับแหล่งกำเนิดข้อมูลแบบ ABR 15 ชุด แบ่งออกเป็น 3 กลุ่ม (กลุ่มละ 5 ชุด) แยกแยะแต่ละกลุ่มตามความยาวของสายส่งที่ต่างกัน
แหล่งกำเนิดข้อมูล TCP	ขนาดแพ็กเก็ต (MTU) = 1,024 ไบต์ ; ขนาดหน้าต่างผู้รับ = 600,000 ไบต์
ส่วนอุปกรณ์อื่น ๆ	เช่น แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึม ERICA+ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.2.1

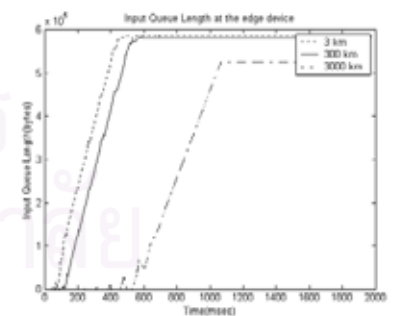
ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการดั้งเดิม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

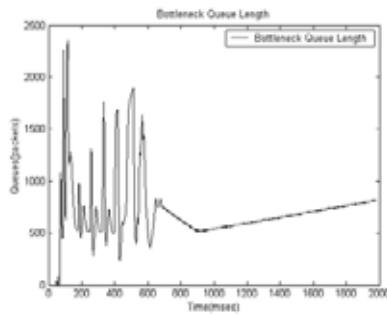


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

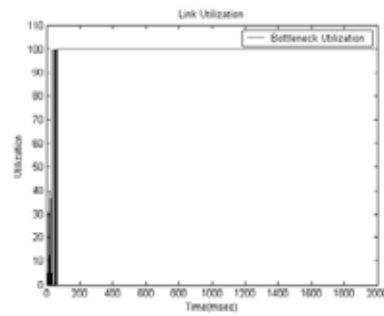


(ค) ปริมาณแถวคอยที่อุปกรณ์ต้นทาง

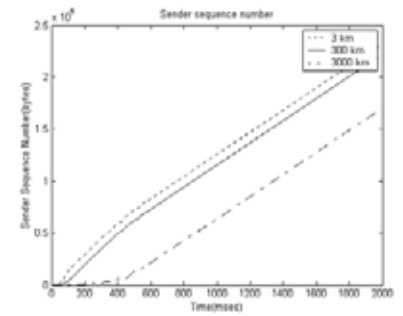
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ข) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.28 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้หลักการตั้งเดิม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.298880×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.196480×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 1.680384×10^6 ไบต์

จากรูป 6.28 (ก) แสดงขนาดหน้าต่างความคับคั่งของการเชื่อมต่อทั้ง 3 กลุ่ม โดยขนาดหน้าต่างความคับคั่งของการเชื่อมต่อในกลุ่ม 3 km เพิ่มเร็วที่สุด ทั้งนี้เพราะค่า RTT ของการเชื่อมต่อกลุ่มนี้มีค่าน้อยที่สุด ทำให้เวลาที่ใช้ในการรอสัญญาณตอบกลับจากการส่งข้อมูลเซกเมนต์หนึ่ง ๆ สั้น และเป็นผลให้แหล่งกำเนิดในกลุ่มนี้สามารถเพิ่มขนาดหน้าต่างได้เร็ว ในทำนองเดียวกัน การเชื่อมต่อในกลุ่ม 300 km และ 3000 km จะปรับขนาดหน้าต่างได้ช้ากว่าตามลำดับ

ด้วยการควบคุมปริมาณข้อมูลที่ส่งออกจากแหล่งกำเนิด TCP โดยขนาดหน้าต่างความคับคั่งนี้เอง ทำให้การปรับค่า ACR ทำได้ช้ากว่าในกรณีที่ไม่มี TCP (รายละเอียดกล่าวไว้ในหัวข้อ 6.1.2 แล้ว) จากรูป 6.28 (ข) สังเกตได้ว่าค่า ACR ที่แหล่งกำเนิดทั้ง 3 กลุ่มใช้นั้นมีค่า 3 ค่าหลัก ๆ คือ ประมาณ 30 Mbps 15 Mbps และ 10 Mbps ซึ่งค่าทั้ง 3 ค่านี้น่ามาจากการคำนวณค่า Fairshare จากจำนวนการเชื่อมต่อที่แตกต่างกัน นั่นคือ 5 10 และ 15 การเชื่อมต่อตามลำดับ (เหมือนที่ได้บรรยายไว้ในหัวข้อที่แล้ว แต่ผลการจำลองในกรณีนี้ชัดเจนกว่า ทั้งนี้เพราะการปรับเพิ่มค่า ACR ของแหล่งกำเนิดช้ากว่า เนื่องด้วยการควบคุมของหลักการ Slow Start ใน TCP) นอกจากการคำนวณจากจำนวนการเชื่อมต่อที่ต่างกันแล้ว ยังมีสาเหตุอื่น ๆ ที่ทำให้ค่า ACR มีการเปลี่ยนแปลงในลักษณะนี้ กล่าวคือเนื่องจากค่า ACR ที่แหล่งกำเนิดแสดง อาจไม่ใช่อัตราการส่งข้อมูลจริงที่แหล่งกำเนิดใช้ โดยเฉพาะกรณีที่เกิดจากการขาดแคลนข้อมูลในการส่ง (รายละเอียดได้กล่าวไว้ในหัวข้อ 6.1.2 แล้ว) และเนื่องจากภาวะขาดแคลนข้อมูลในการส่งนี้ มีความสัมพันธ์โดยตรงกับการเพิ่มขนาดหน้าต่าง (ซึ่งสัมพันธ์กับค่า RTT) ทำให้การปรับค่า ACR ในช่วงแรก ๆ มี

ลักษณะแวก์มากกว่าในกรณีที่ไม่มี TCP และด้วยเหตุนี้เองทำให้ปริมาณข้อมูลในแวก์คอยที่วัดได้ในช่วงนี้ มีลักษณะแวก์ตัวเช่นกัน (ตามรูป 6.28 (ง)) หลังจากขนาดหน้าต่างของการเชื่อมต่อทุกตัวค่ามากเพียงพอ อัตราการส่งข้อมูลที่แหล่งกำเนิดทุกตัวใช้ก็จะเข้าสู่ภาวะอยู่ตัว (เหมือนในกรณีที่ไม่มี TCP) แต่การเพิ่มระดับข้อมูลในแวก์คอยในช่วงนี้ จะเพิ่มช้ากว่าในกรณีที่ไม่มี TCP

สำหรับปริมาณข้อมูลในแวก์คอยที่อุปกรณ์ต้นทาง แสดงดังรูป 6.28 (ค) ในช่วงเริ่มต้นแวก์คอยของการเชื่อมต่อแต่ละกลุ่มมีค่าน้อย หลังจากผ่านไปชั่วระยะเวลาหนึ่งการเชื่อมต่อในกลุ่ม 3 km ก็เริ่มมีการสะสมข้อมูลในอุปกรณ์ส่วนนี้ หลังจากนั้นการเชื่อมต่อในกลุ่ม 300 km และ 3000 km ก็เริ่มสะสมตามลำดับ ทั้งนี้เกิดจากสาเหตุ 2 ประการหลัก นั่นคือ ค่า Bandwidth-Delay product ของการเชื่อมต่อในกลุ่ม 3 km มีค่าน้อยที่สุด และแหล่งกำเนิด TCP ในกลุ่ม 3 km สามารถปรับขนาดหน้าต่างความคับคั่งให้มีค่ามากเพียงพอได้เร็วที่สุด ส่วนระดับแวก์คอยที่ค่อย ๆ เพิ่มขึ้นแล้วค้างอยู่ที่ค่า ๆ หนึ่งนั้น เกิดจากขนาดหน้าต่างความคับคั่งของ TCP ค่อย ๆ เพิ่มขึ้นในลักษณะดังกล่าว รวมทั้งปริมาณข้อมูลที่แหล่งกำเนิด ABR สามารถส่งได้ก็เริ่มคงที่ สำหรับระดับแวก์คอยที่การเชื่อมต่อแต่ละกลุ่มค่านั้น คำนวณได้จากขนาดหน้าต่างปลายทางลบด้วยค่า Bandwidth-Delay product ซึ่งถือว่ามีค่าใกล้เคียงกับปริมาณข้อมูลที่แหล่งกำเนิดสามารถส่งได้ต่อหนึ่งหน่วย RTT อย่างไรก็ตาม ผู้สนใจสามารถศึกษารายละเอียดได้จากบทวิเคราะห์ในภาคผนวก ก

เมื่อพิจารณาปริมาณข้อมูลที่ปลายทางได้รับ จะเห็นว่ามีความไม่เท่าเทียมกัน กล่าวคือการเชื่อมต่อในกลุ่ม 3 km ได้รับข้อมูลที่ปลายทางมากกว่า ในกลุ่ม 300 km และ 3000 km ตามลำดับ ทั้งนี้เกิดจากช่วงแรกของการส่งข้อมูล การเชื่อมต่อในแต่ละกลุ่มอยู่ในช่วงขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างนานมากน้อยต่างกัน

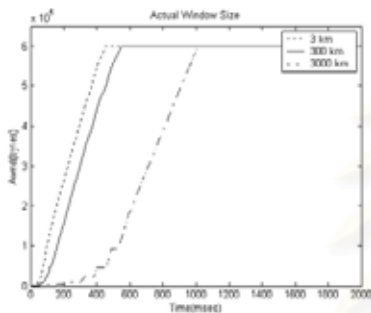
6.2.3 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ บนแบบจำลองที่มีความยาวสายส่งต่างแตกต่างกัน

การจำลองชุดนี้เป็นการศึกษาและวิเคราะห์ประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ ภายใต้แบบจำลองที่มีความยาวสายส่งขนาดต่างกัน โดยแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.1 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ TCP ทำงานร่วมกับแหล่งกำเนิดข้อมูลแบบ ABR 15 ชุด แบ่งออกเป็น 3 กลุ่ม (กลุ่มละ 5 ชุด) แยกแยะแต่ละกลุ่มตามความยาวของสายส่งที่ต่างกัน
----------	--

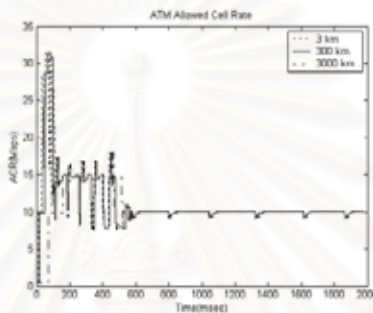
อุปกรณ์ต่าง ๆ เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึม ERICA+ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.2.2

ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน



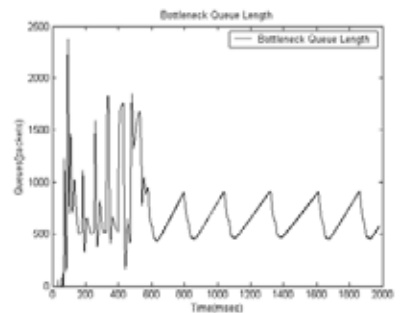
(ก) ขนาดหน้าต่างความคับคั่งเสมือนของ

TCP

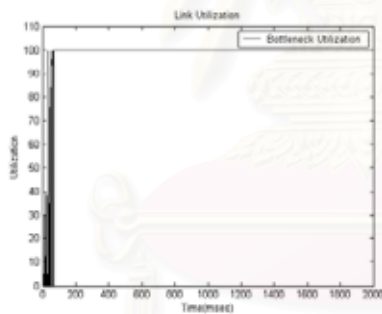


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

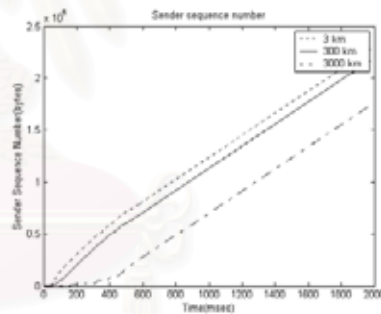
แบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.29 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับ บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.279424x10⁶ ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.173952x10⁶ ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 1.743872x10⁶ ไบต์

ผลการจำลองโดยรวมคล้ายกับผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม ทั้งนี้เพราะหลักการกักเก็บสัญญาณตอบกลับ จะใช้การทำงานในส่วนของผู้รับที่อยู่ในอุปกรณ์เสริม ACK-Bucket มีการจำลองการเพิ่มของขนาดหน้าต่างความคับคั่งของ TCP ให้เหมือนหลักการที่ใช้ในแบบดั้งเดิม ทำให้ผลที่เกิดจากหลักการกักเก็บสัญญาณตอบกลับ ยังคงเห็นผลของขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างความคับคั่ง แต่ข้อดีของหลักการกักเก็บสัญญาณตอบกลับที่เหนือกว่าหลักการดั้งเดิม คือหลักการกักเก็บสัญญาณตอบกลับนี้ จะไม่มีการสะสมข้อมูลที่อุปกรณ์ต้นทาง (ตามหลักการของการกักเก็บสัญญาณตอบกลับ คือใช้เพียงหน่วยความจำเล็กน้อยในการเก็บข้อมูลของสัญญาณตอบกลับแทน)

แต่อย่างไรก็ตามผลการทำงานของหลักการกักเก็บสัญญาณตอบกลับนี้ จะมีการตอบสนองที่เร็วกว่าแบบดั้งเดิมเล็กน้อย ทั้งนี้เนื่องเป็นผลของการทำการจำลองที่ทำการลดส่วนของบัฟเฟอร์ที่อุปกรณ์ต้นทาง ทำให้มีการตอบสนองเร็วขึ้น อย่างไรก็ตามผลของความแตกต่างนี้มีค่าน้อยมาก

6.2.4 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างบนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

การจำลองในหัวข้อนี้เป็นการศึกษาและวิเคราะห์ประสิทธิภาพการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่างภายใต้แบบจำลองที่มีความยาวสายส่งขนาดต่างกัน โดยแบบจำลองและค่าพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังต่อไปนี้

แบบจำลอง	ตามรูปที่ 5.1 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ TCP ทำงานร่วมกับแหล่งกำเนิดข้อมูลแบบ ABR 15 ชุด แบ่งออกเป็น 3 กลุ่ม (กลุ่มละ 5 ชุด) แยกแยะแต่ละกลุ่มตามความยาวของสายส่งที่ต่างกัน
อุปกรณ์ต่าง ๆ	เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึม ERICA+ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.2.2

เนื่องจากการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดนี้ อาศัยค่าเวลาที่ใช้ในการเดินทางไปกลับของข้อมูล (RTT) เป็นพารามิเตอร์หลักในการคำนวณค่าขนาดหน้าต่างที่ใช้

ในการทำงาน ซึ่งค่า RTT ในแบบจำลองนี้มีทั้งหมด 3 ค่า (พิจารณาจากความยาวสายส่งที่ใช้) นั่นคือ การเชื่อมต่อในกลุ่ม 3 km มีค่า RTT เท่ากับ 10 ms (คำนวณจากความยาวสายส่งไปกลับรวม 2012 km) การเชื่อมต่อในกลุ่ม 300 km มีค่า RTT เท่ากับ 16 ms (คำนวณจากความยาวสายส่งไปกลับรวม 3200 km) และการเชื่อมต่อในกลุ่ม 3000 km มีค่า RTT เท่ากับ 70 ms (คำนวณจากความยาวสายส่งไปกลับรวม 14000 km)

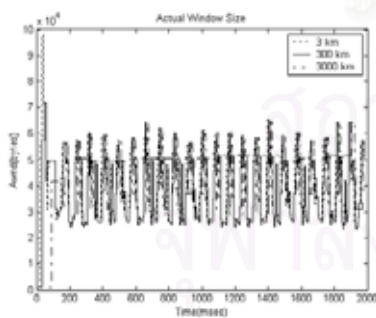
สำหรับการจำลองครั้งนี้ จะพิจารณาผลการจำลองที่เกิดจากการใช้ค่า RTT ที่เป็นค่าคงที่ใด ๆ และค่า RTT ที่วัดได้จากสัญญาณตอบกลับของแต่ละการเชื่อมต่อ ทั้งนี้เพื่อศึกษาความเป็นไปได้ในการลดความซับซ้อนในการคำนวณ รวมทั้งผลกระทบที่มีต่อศักยภาพในการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้การปรับรูปร่างหน้าต่างต่างอีกด้วย โดยค่า RTT ที่ใช้ในการจำลองจะแบ่งออกเป็น 3 กรณี คือ

กรณีที่ 1 ใช้ค่า RTT เฉลี่ย คือค่า RTT ที่เกิดจากการเฉลี่ยค่า RTT ของการเชื่อมต่อทั้งหมดในโครงข่าย (นั่นคือใช้ค่า $RTT = (10+16+70)/3 = 30$ ms)

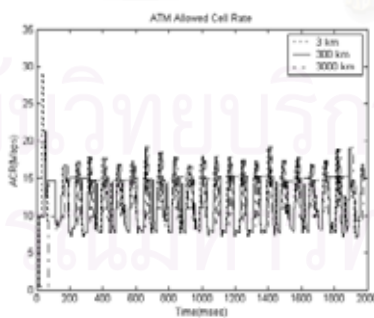
กรณีที่ 2 ใช้ค่า RTT มากสุด คือค่า RTT ของการเชื่อมต่อที่มีขนาดสายส่งยาวที่สุด (นั่นคือใช้ค่า $RTT = 70$ ms)

กรณีที่ 3 ใช้ค่า RTT ของแต่ละการเชื่อมต่อ คือใช้ค่า RTT ที่ได้จากการคำนวณเวลาเมื่อได้รับสัญญาณตอบกลับ

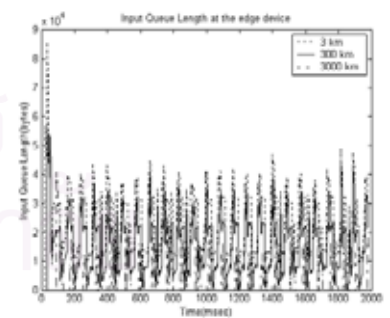
6.2.4.1 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูปร่างหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน และใช้ค่า RTT เฉลี่ยในการคำนวณ



(ก) ขนาดหน้าต่างความคับคั่งของ TCP

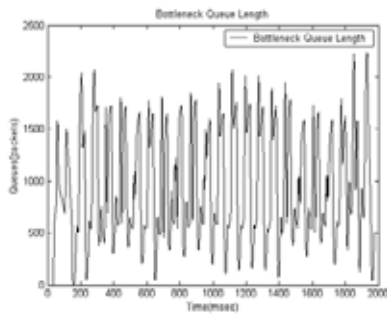


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

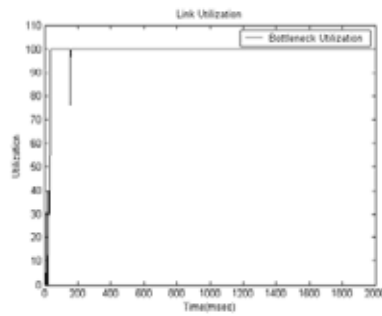


(ค) ปริมาณแควดคอยที่อุปกรณ์ต้นทาง

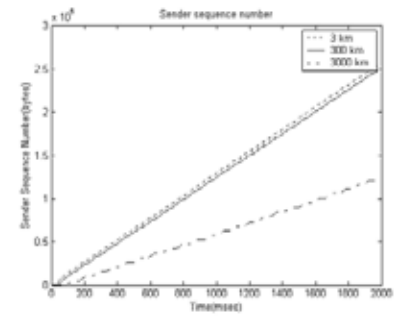
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.30 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การปรับปรุค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน
และใช้ค่า RTT เฉลี่ยในการคำนวณ

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.540544×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.485248×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 1.218560×10^6 ไบต์

การจำลองในหัวข้อนี้ จะใช้ค่า RTT เฉลี่ย (30 ms) ในการคำนวณค่าขนาดหน้าต่างความ
คับคั่งที่จะใช้กับกระบวนการควบคุมการไหลของ TCP ทั้งนี้ถ้าลองคำนวณค่า Bandwidth-Delay
product ของการเชื่อมต่อแต่ละกลุ่มแบบคร่าว ๆ (พิจารณาค่า ER จากการคำนวณค่า Fairshare
ของการเชื่อมต่อทั้ง 15 ตัว นั่นคือมีค่าประมาณ 10 Mbps) จะได้ว่าค่า Bandwidth-Delay
product ของการเชื่อมต่อในกลุ่ม 3 km มีค่าประมาณ $10 \text{ Mbps} \times 10 \text{ ms} = 100 \text{ kbit}$ ส่วนของ
การเชื่อมต่อในกลุ่ม 300 km และ 3000 km จะมีค่าประมาณ 160 kbit และ 700 kbit ตามลำดับ
ในขณะที่ขนาดหน้าต่างที่ได้จากการคำนวณในกรณีนี้มีค่าเท่ากับ 300 kbit ซึ่งจะเห็นว่ามีค่าเพียง
พอสำหรับการเชื่อมต่อในกลุ่ม 3 km และ 300 km ส่วนการเชื่อมต่อในกลุ่ม 3000 km จะเกิด
ปัญหาขาดแคลนข้อมูล กล่าวอีกนัยหนึ่งคือเนื่องด้วยขนาดหน้าต่างความคับคั่งที่มีค่าน้อยเกินไป
ทำให้แหล่งกำเนิดข้อมูลไม่สามารถส่งข้อมูลด้วยอัตราที่เหมาะสมและต่อเนื่องได้ จึงทำให้แหล่ง
กำเนิดในกลุ่ม 3000 km ส่งข้อมูลได้น้อยกว่าที่ควร และเป็นผลให้การทำงานภายใต้เงื่อนไขนี้ ไม่มี
ความเท่าเทียมกัน แสดงได้ดังรูป 6.30 (ฉ) ปริมาณข้อมูลที่ปลายทางของการเชื่อมต่อในกลุ่ม
3000 km ได้รับมีค่าน้อยกว่าปลายทางในกลุ่มอื่น ๆ มาก นอกจากนี้อัตราการรับข้อมูลยังต่ำกว่า
อีกด้วย (พิจารณาจากความชันของกราฟที่แสดง)

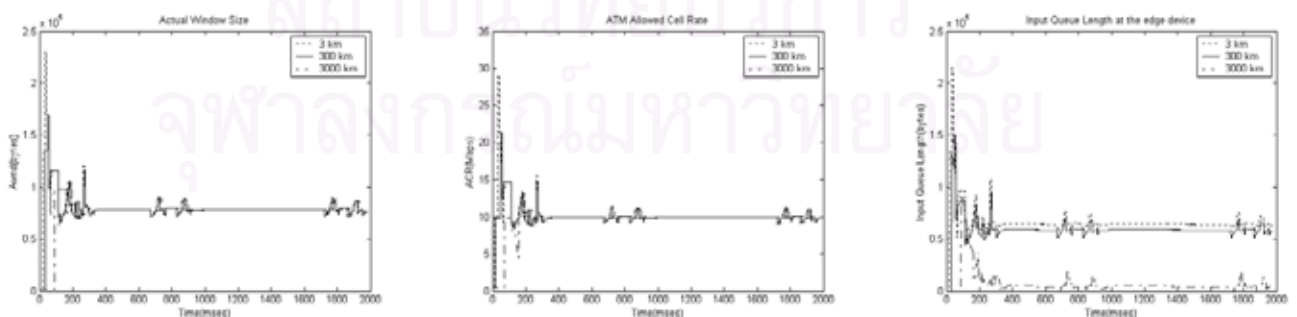
จากความขาดแคลนข้อมูลของแหล่งกำเนิดในกลุ่ม 3000 km นี้เองทำให้ค่า ER ที่คำนวณ
ได้มีลักษณะแกว่งขึ้นแกว่งลงอยู่ตลอดเวลา (แสดงดังรูป 6.30 (ข)) กล่าวคือเนื่องจากแหล่งกำเนิด

ในกลุ่ม 3000 km ไม่สามารถส่งข้อมูลได้ในอัตราที่กำหนด (นั่นคือค่า ACR ที่แหล่งกำเนิดแสดงมีค่ามากกว่าค่าอัตราการส่งข้อมูลจริง ๆ ที่แหล่งกำเนิดใช้) ทำให้ทรัพยากรของโครงข่ายไม่ได้ถูกใช้งานอย่างเต็มที่ ดังนั้น ERICA+ จึงพยายามปรับให้แหล่งกำเนิดเพิ่มอัตราการส่ง (จากรูป 6.30 (ข) สังเกตได้ว่าค่า ACR ของการเชื่อมต่อในบางช่วงมีค่ามากกว่า Fairshare) แต่อย่างไรก็ตามเนื่องหลักการดำเนินงานอย่างหนึ่งของอัลกอริทึม ERICA+ คือทราบได้ก็ตามที่ลิงก์ถูกใช้งานอย่างเต็มที่ (ค่า load factor มีค่าเท่ากับ 1 หรือใกล้เคียง) แหล่งกำเนิดทุกตัวจะถูกปรับค่าอัตราการส่งข้อมูลให้มีค่าเท่ากัน (เพื่อพยายามให้แหล่งกำเนิดทุกตัวใช้ทรัพยากรอย่างเท่าเทียมกัน) ดังนั้นอัตราการส่งข้อมูลที่แสดงดังรูป 6.30 (ข) จึงมีลักษณะแกว่งขึ้นแกว่งลงดังกล่าว นอกจากนี้ในการคำนวณค่า ER ของ ERICA+ จะคำนึงถึงการรักษาระดับข้อมูลในแถวคอยที่สวิตช์ด้วย ดังนั้นการแกว่งตัวดังกล่าวมีผลมาจากระดับแถวคอยที่แกว่งตัวด้วย

จากการแกว่งตัวของอัตราการส่งข้อมูล ทำให้ขนาดหน้าต่างที่ใช้แกว่งตัวตามไปด้วย (ดังรูป 6.30 (ก)) และเป็นผลให้ปริมาณแถวคอยที่อุปกรณ์ต้นทางแกว่งเช่นกัน อย่างไรก็ตามเพื่อพิจารณารูป 6.30 (ค) อย่างละเอียด จะเห็นว่าปริมาณแถวคอยที่สะสมในอุปกรณ์ต้นทางของแหล่งกำเนิดในกลุ่ม 3 km มีค่ามากกว่าในกลุ่ม 300 km และมากกว่าในกลุ่ม 3000 km (ซึ่งแทนจะไม่มีการสะสมเลย) ตามลำดับ

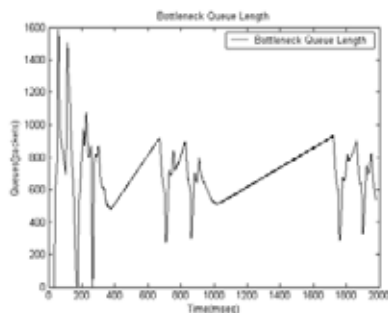
เมื่อพิจารณาระดับแถวคอยในสวิตช์ ATM จะเห็นว่า ERICA+ สามารถควบคุมให้สวิตช์มีข้อมูลในแถวคอยเกือบตลอดเวลา (ดังรูป 6.30 (ง)) ทำให้ลิงก์ถูกใช้งานอย่างมีประสิทธิภาพ (ดังรูป 6.30 (จ)) ทั้ง ๆ ที่มีการเชื่อมต่อกลุ่มหนึ่ง ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ

6.2.4.2 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับรูขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยใช้ค่า RTT มากสุดในการคำนวณ

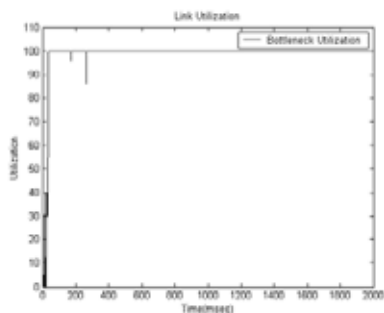


(ก) ขนาดหน้าต่างความคับคั่งของ TCP (ข) อัตราการส่งข้อมูลของแหล่งกำเนิด (ค) ปริมาณแถวคอยที่อุปกรณ์ต้นทาง

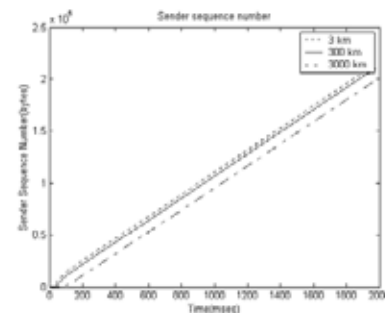
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ข) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.31 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การปรับค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน
และใช้ค่า RTT มากสุดในการคำนวณ

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.146304×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.109440×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 1.990656×10^6 ไบต์

การจำลองในหัวข้อนี้ เป็นการจำลองอีกกรณีที่ใช้ค่า RTT คงที่ใด ๆ กับกระบวนการควบคุมการไหลของ TCP โดยใช้การปรับขนาดหน้าต่าง แต่กรณีนี้ใช้ค่า RTT มากที่สุด นั่นคือเท่ากับ 70 ms ในการคำนวณ ดังนั้นค่าขนาดหน้าต่างความคับคั่งคร่าว ๆ ที่คำนวณได้จึงมีค่าประมาณ 700 kbit ทั้งนี้เมื่อเปรียบเทียบกับค่า Bandwidth-Delay product ของการเชื่อมต่อแต่ละกลุ่ม (มีการคำนวณไว้ในหัวข้อที่แล้ว) จะเห็นว่าค่าขนาดหน้าต่างที่ใช้ในกรณีนี้มีค่ามากเพียงพอสำหรับการส่งข้อมูลของการเชื่อมต่อทุกตัวในโครงข่าย

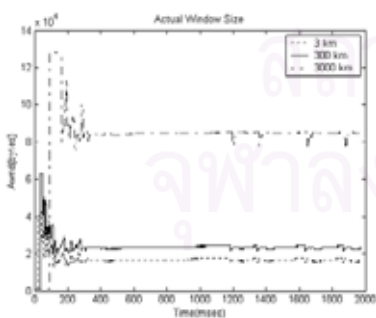
เมื่อพิจารณาอัตราการส่งข้อมูลของแหล่งกำเนิด ดังรูป 6.31 (ข) จะเห็นว่ามีความคล้ายผลการจำลองที่เกิดจากกรณีไม่มี TCP นั่นคือในช่วงแรกอัตราการส่งข้อมูลของแหล่งกำเนิดแต่ละกลุ่มจะถูกปรับให้มีค่าใกล้เคียงกับค่า Fairshare โดยเวลาที่ใช้ในการปรับขึ้นอยู่กับความยาวของสายส่งของการเชื่อมต่อแต่ละกลุ่ม หลังจากนั้นค่าอัตราการส่งข้อมูลก็มีค่าคงอยู่ที่ประมาณ 10 Mbps และเนื่องจากค่า RTT ที่ใช้ในการคำนวณครั้งนี้มีค่าเท่ากับค่า RTT มากที่สุด ซึ่งเป็นค่า RTT ที่ได้จากการคำนวณหาค่า RTT ของการเชื่อมต่อในกลุ่ม 3000 km แบบคร่าว ๆ ทั้งนี้ค่า RTT ที่การเชื่อมต่อในกลุ่ม 3000 km ใช้จริง ๆ อาจมีค่ามากกว่า 70 ms (เพราะในการส่งข้อมูลจริงนั้น เวลาที่ข้อมูลใช้ในการเดินทางไปกลับ ไม่ได้คำนวณจากความยาวของสายส่งเพียงอย่างเดียว ทั้งนี้ต้องรวมเวลาที่สูญเสียไปในแควคอยของอุปกรณ์ต่าง ๆ ด้วย) ดังนั้นการเชื่อมต่อในกลุ่ม 3000 km อาจเกิดภาวะขาดแคลนข้อมูลชั่วคราวได้ ซึ่งเป็นผลให้เกิดภาวะคล้ายคลึงกับใน

กรณีที่ใช้ค่า RTT เฉลี่ยในช่วงสั้น ๆ นั่นคือค่า ER ที่คำนวณได้จะมีค่ามากกว่าปรกติ แต่กรณีนี้จะมีผลเป็นช่วงสั้น ๆ (พิจารณาได้จากค่า ACR ในรูป 6.31 (ข) มีการกระตุกขึ้นเล็กน้อยเป็นช่วง ๆ)

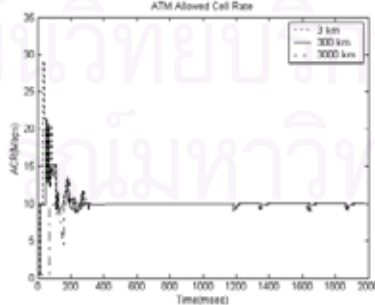
เนื่องจากค่า RTT ที่ใช้ในการคำนวณขนาดหน้าต่างกรณีนี้มีค่าคงที่ ดังนั้นค่าขนาดหน้าต่างที่ได้จากการจำลอง (ดังรูป 6.31 (ก)) จึงมีลักษณะคล้ายกราฟ ACR สำหรับปริมาณข้อมูลที่สะสมไว้ในแถวคอยของอุปกรณ์ต้นทาง จะเห็นว่าช่วงแรก ๆ ของการเชื่อมต่อปริมาณข้อมูลที่สะสมในแถวคอยส่วนนี้มีค่ามาก ทั้งนี้เพราะค่าขนาดหน้าต่างถูกคำนวณจากค่า ER ซึ่งเป็นค่าอัตราการส่งข้อมูลที่โครงข่ายแนะนำให้ใช้ แต่ไม่ใช่อัตราการส่งที่แหล่งกำเนิดใช้ในการส่งจริง จึงทำให้ขนาดหน้าต่างมีค่ามากเกินไป และก่อให้เกิดการตกค้างของข้อมูลที่อุปกรณ์ต้นทางเป็นปริมาณมาก หลังจากนั้นปริมาณแถวคอยที่อุปกรณ์ต้นทางของการเชื่อมต่อแต่ละกลุ่มมีลักษณะค่อนข้างคงที่ โดยปริมาณแถวคอยของการเชื่อมต่อในกลุ่ม 3 km มีค่ามากกว่าในกลุ่ม 300 km และมากกว่าในกลุ่ม 3000 km (ดังรูป 6.31 (ค)) ทั้งนี้เพราะปริมาณข้อมูลที่สามารรถเติมเต็มการเชื่อมต่อในแต่ละกลุ่มมีค่ามากขึ้นตามลำดับ

ด้วยเหตุที่การจำลองครั้งนี้ใช้ค่า RTT ของการเชื่อมต่อในกลุ่ม 3000 km ซึ่งเป็นค่าที่มากที่สุดในการคำนวณค่าขนาดหน้าต่าง ดังนั้นค่าขนาดหน้าต่างที่ใช้สำหรับแหล่งกำเนิดแบบ TCP ของการเชื่อมต่อทุกกลุ่ม จึงมีค่าเพียงพอที่จะให้การเชื่อมต่อในแต่ละกลุ่มทำงานอย่างเต็มประสิทธิภาพ ด้วยเหตุนี้ปริมาณข้อมูลที่ปลายทางในแต่ละกลุ่มได้รับจึงมีค่าใกล้เคียงกัน ซึ่งแสดงให้เห็นถึงคุณสมบัติความเท่าเทียมกันของกระบวนการนี้

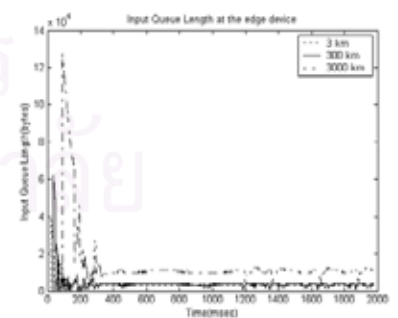
6.2.4.3 ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การปรับปรุ้งค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน โดยใช้ค่า RTT ของแต่ละการเชื่อมต่อในการคำนวณ



(ก) ขนาดหน้าต่างความคั้งของ TCP

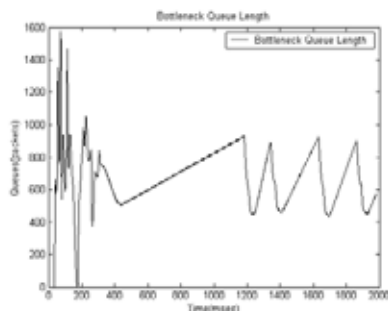


(ข) อัตราการส่งข้อมูลของแหล่งกำเนิด

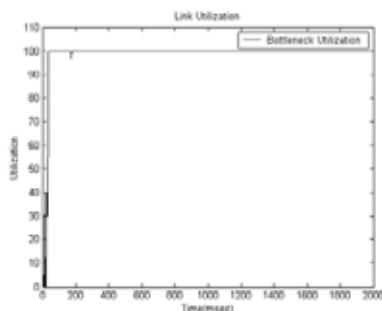


(ค) ปริมาณแถวคอยที่อุปกรณ์ต้นทาง

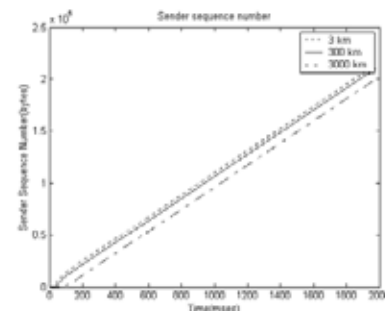
แบบ ABR



(ง) ปริมาณแควคอยในสวิตช์ ATM



(จ) ประสิทธิภาพการใช้งานลิงก์



(ฉ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.32 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การปรับค่าขนาดหน้าต่าง บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน
และใช้ค่า RTT ของแต่ละการเชื่อมต่อในการคำนวณ

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.140160×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.104320×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 2.000896×10^6 ไบต์

สำหรับการจำลองการทำงานในหัวข้อนี้ เป็นการตั้งค่า RTT ของการเชื่อมต่อแต่ละตัวใน
การคำนวณค่าขนาดหน้าต่างความคับคั่ง ดังนั้นค่าขนาดหน้าต่างความคับคั่งที่ได้จึงมีค่าใกล้เคียง
กับค่า Bandwidth-Delay product ของการเชื่อมต่อในแต่ละกลุ่ม (แสดงดังรูป 6.32 (ก)) เมื่อ
พิจารณารูป 6.32 (ข) ซึ่งแสดงค่า ACR ที่ได้จากการจำลองการทำงานในกรณีนี้ จะเห็นว่าผลการ
จำลองที่ได้มีลักษณะคล้ายผลการจำลองที่ได้จากกรณีที่ไม่มี TCP กล่าวคือในช่วงแรกค่า ACR
เกิดการแกว่งตัว เพราะการปรับค่า ACR ของแหล่งกำเนิด ABR ไม่สามารถปรับได้อย่างทันที
ทันใด อีกทั้งกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการนี้ แหล่งกำเนิด TCP ต้องรอรับ
สัญญาณตอบกลับตัวแรกจากโครงข่าย เพื่อนำค่า RTT ที่วัดได้ไปใช้ในการคำนวณค่าขนาดหน้าต่าง
ต่าง จึงทำให้การส่งข้อมูลในช่วงแรกแกว่ง และเป็นผลให้การใช้งานลิงก์ในช่วงนี้ไม่เต็มประสิทธิภาพ
อีกด้วย (ดังรูป 6.32 (จ))

นอกจากนี้การตั้งค่าขนาดหน้าต่างความคับคั่ง ที่มีค่าใกล้เคียงกับค่า Bandwidth-Delay
product นั้น มีผลทำให้หน่วยความจำที่อุปกรณ์ต้นทางแทบจะไม่มีข้อมูลค้างอยู่เลย (ดังรูป 6.32
(ค)) อย่างไรก็ตามการที่ขนาดหน้าต่างใช้ค่า ER ซึ่งอาจจะมีค่าไม่เท่ากับค่าอัตราการส่งข้อมูลจริง
จึงทำให้ระดับแควคอยในอุปกรณ์ส่วนนี้ยังผกผันบ้างไม่มากนักน้อย สำหรับการพิจารณาปริมาณ
ข้อมูลที่ปลายทางรับ (แสดงดังรูป 6.32 (ฉ)) จะเห็นว่า การเชื่อมต่อแต่ละกลุ่มสามารถรับข้อมูลได้

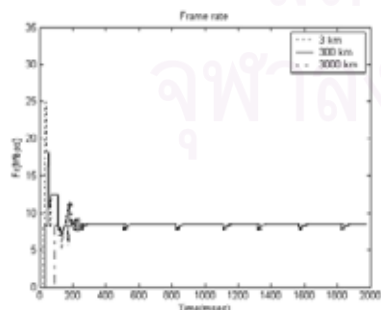
อย่างเต็มที่ และมีค่าใกล้เคียงกันมาก ทั้งนี้เป็นการแสดงถึงคุณสมบัติความเท่าเทียมกันของการทำงานภายใต้เงื่อนไขนั่นเอง

6.2.5 การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

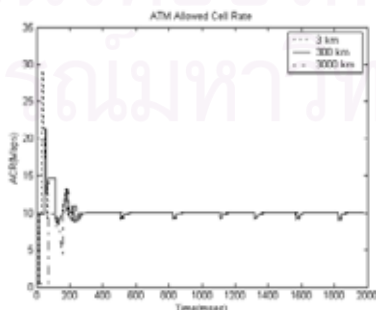
การจำลองในหัวข้อนี้เป็นการศึกษากระบวนการทำงานของการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม ภายใต้แบบจำลองที่มีความยาวสายส่งขนาดต่างกัน และเป็นการวิเคราะห์ศักยภาพในการทำงานของกระบวนการนี้ ภายใต้เงื่อนไขดังกล่าวด้วย โดยแบบจำลองและพารามิเตอร์ที่ใช้ในการจำลองสามารถแจกแจงได้ดังนี้

แบบจำลอง	ตามรูปที่ 5.1 (ก) ซึ่งประกอบด้วยแหล่งกำเนิดข้อมูลแบบ TCP ทำงานร่วมกับแหล่งกำเนิดข้อมูลแบบ ABR 15 ชุด แบ่งออกเป็น 3 กลุ่ม (กลุ่มละ 5 ชุด) แยกแยะแต่ละกลุ่มตามความยาวของสายส่งที่ต่างกัน
อุปกรณ์ต่าง ๆ	เช่น แหล่งกำเนิดข้อมูล TCP ; แหล่งกำเนิดข้อมูล VBR ; แหล่งกำเนิดข้อมูล ABR ; ความยาวของสายส่ง รวมไปถึง อัลกอริทึม ERICA+ ที่ใช้ในกระบวนการควบคุมความคับคั่งของ ABR จะถูกตั้งค่าเหมือนการจำลองในหัวข้อ 6.2.2

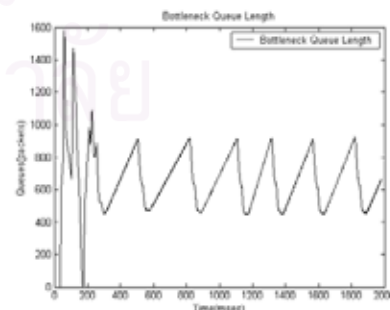
ผลการจำลองและการวิเคราะห์การทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน



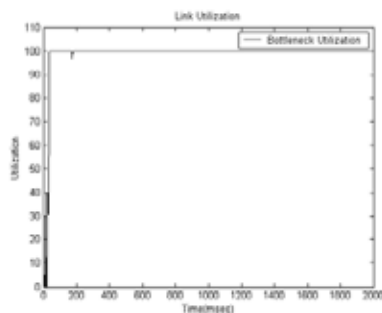
(ก) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ TCP



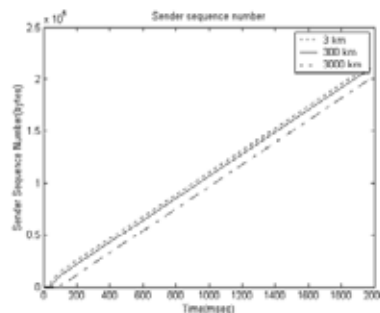
(ข) อัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR



(ค) ปริมาณแถวคอยในสวิตช์ ATM



(ง) ประสิทธิภาพการใช้งานลิงก์



(จ) ปริมาณข้อมูลที่ปลายทางได้รับ

รูปที่ 6.33 ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตรา
การส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม
บนแบบจำลองที่มีความยาวสายส่งแตกต่างกัน

ค่าวิสัยสามารถสุทธิของการเชื่อมต่อแบบ ABR ที่วัดได้

- การเชื่อมต่อในกลุ่ม 3 km วัดได้ 2.142208×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 300 km วัดได้ 2.107392×10^6 ไบต์
- การเชื่อมต่อในกลุ่ม 3000 km วัดได้ 2.007040×10^6 ไบต์

ผลการจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม มีลักษณะคล้ายคลึงกับผลการจำลองที่ได้จากการจำลองภายใต้แบบจำลองที่ไม่มี TCP (ในหัวข้อ 6.2.1) ยกเว้นเพียงช่วงแรกของการเชื่อมต่อ ที่อัตราการส่งข้อมูลเพิ่มขึ้นกว่า ทั้งนี้เพราะในช่วงเริ่มส่งข้อมูล แหล่งกำเนิด TCP จะต้องรอรับสัญญาณตอบกลับ เพื่อนำค่าขนาดหน้าต่างด้านรับมาเป็นตัวกำหนดขอบเขตของอัตราการส่งข้อมูลต่อไป (รายละเอียดส่วนนี้กล่าวไว้ในหัวข้อ 6.1.5.5 แล้ว) ซึ่งผลกระทบดังกล่าวจะเกิดนานแค่ไหน ขึ้นอยู่กับความยาวของสายส่ง

นอกจากนี้เมื่อเปรียบเทียบอัตราการส่งข้อมูลแบบเฟรมที่ใช้ในชั้น TCP (จากรูป 6.33 (ก)) และอัตราการส่งข้อมูลแบบเซลล์ในชั้น ATM (จากรูป 6.33 (ข)) จะเห็นว่ามีความคล้ายคลึงกันมาก เพียงแต่ค่าอัตราการส่งแบบเฟรมมีค่าน้อยกว่าเท่านั้น ทั้งนี้เป็นผลมาจากกระบวนการทำงานในชั้น ATM มีการสร้างข้อมูลบางส่วนเพิ่มขึ้นมา ไม่ว่าจะเพิ่มข้อมูลทีละน้อยจากกระบวนการตัดแบ่งข้อมูลเป็นเซลล์ หรือกระบวนการควบคุมความคับคั่งที่ต้องสร้างเซลล์ RM ขึ้นมา (พิจารณารายละเอียดได้จากหัวข้อ 6.1.5.5 การวิเคราะห์การจำลองการทำงานของกระบวนการควบคุมการไหลของ TCP ชนิดควบคุมอัตราการส่งข้อมูลโดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม และโดยใช้อัลกอริทึมแบบต่าง ๆ ในกระบวนการควบคุมความคับคั่งของ ABR)

เนื่องจากกระบวนการแปลงอัตราการส่งข้อมูลจากเซลล์เป็นเฟรมนี้ ไม่ก่อให้เกิดปัญหาขาดแคลนข้อมูล ทำให้การควบคุมปริมาณทราฟฟิกในโครงข่าย ดูเหมือนทำโดยกระบวนการควบคุมความคับคั่งของ ABR เพียงอย่างเดียว และเนื่องจากความสามารถในการจัดสรรทรัพยากรอย่างเท่าเทียมกัน และมีประสิทธิภาพของอัลกอริทึม ERICA+ ส่งผลให้กระบวนการควบคุมการไหลของ TCP โดยใช้หลักการนี้ มีคุณสมบัติของความเท่าเทียมกันด้วย นั่นคือปริมาณข้อมูลที่ปลายทางของการเชื่อมต่อแต่ละกลุ่มได้รับมีค่าใกล้เคียงกัน (ดังรูป 6.33 (จ)) อย่างไรก็ตามเมื่อพิจารณาโดยละเอียด ปริมาณข้อมูลที่ปลายทางของการเชื่อมต่อในกลุ่ม 3 km ได้รับมีค่ามากกว่าปลายทางของการเชื่อมต่อในกลุ่ม 300 km และมากกว่าที่พิจารณาจากกลุ่ม 3000 km ทั้งนี้มีผลมาจาก 2 ประเด็นหลัก ๆ กล่าวคือ ประเด็นแรกเกิดจากการเชื่อมต่อในกลุ่ม 3 km ปรับอัตราการส่งข้อมูลเข้าสู่จุดทำงานได้เร็วกว่าการเชื่อมต่อในกลุ่มอื่น ๆ และประเด็นที่ 2 คือปริมาณที่แสดงในรูป 6.33 (จ) หรือค่าวิสัยสมรรถนะที่วัดได้จากการจำลองนั้น เป็นค่าที่ปลายทางได้รับ ณ เวลาหนึ่ง ๆ ที่ยังไม่รวมข้อมูลที่อยู่ระหว่างการส่ง (ซึ่งปริมาณข้อมูลที่อยู่ระหว่างทางของการเชื่อมต่อในกลุ่ม 3000 km มีค่ามากที่สุด)

บทที่ 7

บทสรุปและข้อเสนอแนะ

7.1 สรุปผลการจำลองการทำงาน

ผลการจำลองและการวิเคราะห์ในบทที่ 6 เป็นการทดสอบสมรรถนะของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ ซึ่งสามารถสรุปได้ดังนี้

กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้หลักการดั้งเดิม

เกิด **ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง** ในช่วงแรกของการสื่อสาร เนื่องจากแหล่งกำเนิดข้อมูลแบบ TCP ปรับขนาดหน้าต่างโดยใช้หลักการ Slow Start ซึ่งในช่วงแรก ๆ ขนาดหน้าต่างมีค่าน้อย ทำให้ปริมาณข้อมูลที่แหล่งกำเนิดส่งออกไปถูกจำกัดด้วยขนาดหน้าต่างดังกล่าว อย่างไรก็ตามระยะเวลาที่ระบบจะอยู่ในภาวะนี้ ขึ้นอยู่กับ 2 ปัจจัย คือระยะทางระหว่างต้นทางและปลายทาง และแบนด์วิดท์ที่ระบบแบ่งให้การเชื่อมต่อนั้น ๆ กล่าวคือถ้าระยะทางระหว่างต้นทางและปลายทางมาก หรือการที่ระบบแบ่งแบนด์วิดท์ให้การเชื่อมต่อนั้นมาก ช่วงเวลาที่จะเกิดขีดจำกัดนี้จะนาน

สำหรับ **ขีดจำกัดจากอัตราการส่งข้อมูล** จะเกิดเมื่อขนาดหน้าต่างที่ควบคุมกระบวนการไหลของข้อมูลของ TCP มีค่ามากเกินไป จนทำให้เกิดการสะสมข้อมูลในหน่วยความจำที่อุปกรณ์ต้นทางเป็นจำนวนมาก

เมื่อระยะทางระหว่างต้นทางและปลายทางของการเชื่อมต่อในโครงข่ายต่างกัน การรับข้อมูลที่ปลายทางจะไม่มี **ความเท่าเทียมกัน** กล่าวคือการเชื่อมต่อที่มีระยะทางระหว่างต้นทางและปลายทางน้อย ปลายทางจะได้รับข้อมูลมากกว่าการเชื่อมต่อที่มีระยะทางระหว่างต้นทางและปลายทางน้อย ซึ่งปริมาณข้อมูลที่มากกว่านี้ จะขึ้นอยู่กับระยะเวลาที่การเชื่อมต่อง่ายกว่าอยู่ภายใต้ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง

สำหรับ **ความเร็วในการเข้าสู่จุดทำงาน** ขึ้นอยู่กับความเร็วในการเพิ่มขนาดหน้าต่าง นั่นคือขึ้นอยู่กับระยะทางระหว่างต้นทางและปลายทาง ส่วน **ค่าวิสัยสามารถสุทธิเฉลี่ย** ของกระบวนการควบคุมการไหลของ TCP แบบนี้ มีค่าน้อยกว่าค่าวิสัยสามารถสุทธิที่เกิดจากกระบวนการควบคุมการไหลของ TCP แบบอื่น ๆ

กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้หลักการกักเก็บสัญญาณตอบกลับ

ยังคงเกิด **ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง** ในช่วงแรกของการสื่อสาร เหมือนกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม ทั้งนี้เพราะหลักการกักเก็บสัญญาณตอบกลับนี้ จะควบคุมปริมาณข้อมูลที่ส่งออกจากแหล่งกำเนิดแบบ TCP โดยการเทียบกับหลักการ Slow Start ดังนั้นในช่วงแรกความสามารถในการส่งข้อมูล จึงถูกจำกัดเช่นเดียวกัน

อย่างไรก็ตามกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ จะไม่ก่อให้เกิด **ขีดจำกัดจากอัตราการส่งข้อมูล** ทั้งนี้เพราะกระบวนการนี้ จะส่งข้อมูลแพ็กเก็ตต่อแพ็กเก็ตตามความต้องการของโครงข่าย โดยการควบคุมจากสัญญาณตอบกลับ ทำให้ไม่มีการสะสมของข้อมูลในหน่วยความจำต้นทาง

ส่วนประสิทธิภาพการทำงานในแง่ของ **ความเท่าเทียมกัน ความเร็วในการเข้าสู่จุดทำงาน และควาวิสัยสามารถสุทธิเฉลี่ย** นั้นได้ผลเหมือนกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม ทั้งนี้เพราะผู้ออกแบบกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับนี้ ต้องการเพียงลดขนาดหน่วยความจำต้นทางเท่านั้น แต่ไม่มีจุดประสงค์จะเพิ่มความสามารถในการส่งข้อมูลของแหล่งกำเนิด ดังนั้นประสิทธิภาพในการทำงานโดยรวมจึงไม่แตกต่างจากผลที่เกิดจากหลักการดั้งเดิมมากนัก รวมทั้งผลของขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างที่ยังคงปรากฏอยู่ในกระบวนการควบคุมการไหลของ TCP แบบนี้ด้วย

กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การปรับปรุงขนาดหน้าต่าง

กระบวนการควบคุมการไหลของ TCP โดยใช้การปรับปรุงขนาดหน้าต่างนั้น ค่อนข้างมีปัญหาเมื่อทราฟฟิกมีการเปลี่ยนแปลงตลอดเวลา ทั้งนี้สาเหตุหลัก ๆ เกิดจากการที่หลักการนี้ใช้ค่า ER ในการคำนวณ ซึ่งค่า ER นี้คือค่าอัตราการส่งข้อมูลที่เกิดจากผลการทำงานของกระบวนการควบคุมความคับคั่งของ ABR หรือกล่าวอีกนัยหนึ่งคือค่าอัตราการส่งข้อมูลที่โครงข่ายแนะนำให้แหล่งกำเนิดใช้ ซึ่งอาจไม่ใช่ค่าที่แหล่งกำเนิดใช้ในการส่งข้อมูลจริง เพราะแหล่งกำเนิดจะไม่ปรับอัตราการส่งข้อมูลให้เท่ากับค่าที่โครงข่ายแนะนำในทันที แต่จะค่อย ๆ ปรับเป็นขั้น ๆ ไป โดยการปรับจะพิจารณาจากค่า RIF และ RDF ดังนั้นผลการคำนวณค่าขนาดหน้าต่างตามกรรมวิธีดังกล่าว จึงทำให้มีความเป็นไปได้ที่จะก่อให้เกิด **ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง และขีดจำกัดจากอัตราการส่งข้อมูล** ในช่วงสั้น ๆ กล่าวคือ **ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง** เกิดขึ้นในช่วงที่แหล่งกำเนิดพยายามลดอัตราการส่ง ซึ่งในช่วงนี้ค่า ER มีค่าน้อยกว่าค่า ACR ทำให้ผลการ

ค่านวนขนาดหน้าต่างที่ได้น้อยกว่าปริมาณข้อมูลที่แหล่งกำเนิด ABR ต้องการส่ง ส่วน **ขีดจำกัดจากอัตราการส่งข้อมูล** เกิดขึ้นในช่วงที่แหล่งกำเนิดพยายามเพิ่มอัตราการส่ง ซึ่งในช่วงนี้ค่า ER มีค่ามากกว่าค่า ACR ทำให้ขนาดหน้าต่างที่คำนวณมีค่ามาก จึงเกิดการสะสมข้อมูลในหน่วยความจำต้นทาง จะเห็นว่าปัญหาดังกล่าวจะทวีความรุนแรงมากขึ้นถ้าทราฟฟิกที่กำลังพิจารณามีการเปลี่ยนแปลงตลอดเวลา

สำหรับหลักการการทำงานของกระบวนการควบคุมการไหลของ TCP โดยใช้การปรับรูปร่างหน้าต่าง จัดได้ว่ามี **ความเท่าเทียมกัน** ระหว่างเชื่อมต่อในโครงข่าย ถึงแม้ว่าลักษณะของโครงข่ายจะประกอบด้วยการเชื่อมต่อที่มีระยะทางระหว่างต้นทางและปลายทางไม่เท่ากันก็ตาม

ส่วนเรื่อง **ความเร็วในการเข้าสู่จุดทำงาน** ขึ้นอยู่กับความเร็วในการปรับอัตราการส่งข้อมูลของแหล่งกำเนิดแบบ ABR ซึ่งเมื่อเทียบกับกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม และหลักการกักเก็บสัญญาณตอบกลับ ถือว่ามีการปรับรูปร่างหน้าต่างใช้เวลาในการเข้าสู่จุดทำงานน้อยกว่ามาก

นอกจากนี้เนื่องจากกระบวนการควบคุมการไหลของ TCP โดยใช้การปรับรูปร่างหน้าต่าง ต้องการตัวแปรซึ่งเป็นผลการตรวจสอบจากโครงข่ายถึงสองตัว ผู้คิดค้นกระบวนการนี้จึงพยายามลดความยุ่งยากดังกล่าว โดยการลดตัวแปรลงหนึ่งตัว นั่นคือกำหนดให้ค่า RTT ที่ใช้ในการคำนวณเป็นค่าคงที่ใด ๆ ถึงแม้ว่าการใช้ค่า RTT คงที่ จะเป็นการลดความยุ่งยากในการทำงานของกระบวนการนี้ แต่เป็นการลดประสิทธิภาพการทำงานของระบบด้วย กล่าวคือในกรณีที่ใช้ค่า RTT มากกว่าค่า RTT จริงของการเชื่อมต่อ การเชื่อมต่อดังกล่าวจะเกิด **ขีดจำกัดจากอัตราการส่งข้อมูล** ขึ้น ในทางกลับกันถ้าใช้ค่า RTT น้อยกว่าค่า RTT จริงของการเชื่อมต่อ การเชื่อมต่อดังกล่าวจะเกิด **ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง** ขึ้นรวมทั้งยังมีผลกระทบต่อประสิทธิภาพของการทำงานในแง่ของ **ความเท่าเทียมกัน** หรือ**คำวิสัยสามารถสุทธิเฉลี่ย** ที่ปลายทางได้รับอีกด้วย

กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

สำหรับกระบวนการควบคุมการไหลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรมนี้ เป็นกระบวนการที่พยายามควบคุมการส่งข้อมูลของแหล่งกำเนิด TCP ให้สอดคล้องกับความสามารถในการส่งข้อมูลจริงมากที่สุด กล่าวคือกระบวนการนี้จะคำนวณอัตราการส่งข้อมูลในชั้น TCP จากค่าอัตราการส่งข้อมูลที่แหล่งกำเนิด ABR จะใช้จริง (ค่าอัตราการส่งที่แหล่งกำเนิดใช้จริงนี้ จะวิเคราะห์ในลักษณะเดียวกับการปรับค่าอัตราการส่งของแหล่งกำเนิดจริง ซึ่งพิจารณาจากข้อมูลในเซลล์ RM ย้อนกลับ นอกจากนี้จะเห็นว่าค่าอัตราการส่งข้อมูลในชั้น TCP จะมีค่าน้อยกว่าค่าอัตราการส่งข้อมูลในชั้น ATM ทั้งนี้เพราะมีข้อมูลส่วนหนึ่งถูกสร้างขึ้นใหม่ เพื่อรองรับ

กระบวนการทำงานในระดับชั้น ATM) โดยแหล่งกำเนิด TCP จะควบคุมจังหวะการส่งข้อมูลแต่ละเฟรม แทนการกำหนดขนาดหน้าต่าง จากหลักการทำงานดังกล่าว จึงทำให้สามารถหลีกเลี่ยงการเกิด ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง และขีดจำกัดจากอัตราการส่งข้อมูล ไปพร้อม ๆ กัน ทั้งนี้ การหลีกเลี่ยงการเกิด ขีดจำกัดจากอัตราการส่งข้อมูล นั้นทำได้เพราะการส่งข้อมูลที่ละเฟรม ด้วยอัตราที่เหมาะสม จึงทำให้ไม่เกิดการสะสมข้อมูลที่หน่วยความจำต้นทาง ส่วนการหลีกเลี่ยงการเกิด ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง นี้ไม่ใช่เพราะวิธีการนี้ไม่ได้ใช้ขนาดหน้าต่างในการควบคุมปริมาณข้อมูลที่ส่ง แต่เป็นเพราะวิธีการนี้สามารถควบคุมปริมาณข้อมูลที่ส่งให้สอดคล้องกับความต้องการ และสภาพความคับคั่งของโครงข่ายจริงได้ โดยไม่ทำให้ประสิทธิภาพการทำงานน้อยลง

สำหรับประสิทธิภาพการทำงานในแง่ ความเท่าเทียมกัน ระหว่างการเชื่อมต่อในโครงข่าย ภายใต้การทำงานของกระบวนการควบคุมการไหลของ TCP แบบนี้ จัดได้ว่ามี ความเท่าเทียมกัน มากทีเดียว ถึงแม้ว่าลักษณะของโครงข่ายจะประกอบด้วย การเชื่อมต่อที่มีระยะทางระหว่างต้นทาง และปลายทางไม่เท่ากันก็ตาม

เนื่องจากหลักการทำงานของกระบวนการควบคุมการไหลของ TCP แบบนี้ ขึ้นอยู่กับกระบวนการควบคุมความคับคั่งของ ABR ค่อนข้างมาก ดังนั้น ความเร็วในการเข้าสู่จุดทำงาน จึงขึ้นอยู่กับความเร็วในการเข้าสู่จุดทำงานของกระบวนการควบคุมความคับคั่งของ ABR ที่ใช้

อย่างไรก็ตามสมรรถนะของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ สามารถสรุปและเปรียบเทียบในตารางต่อไปนี้

ตารางที่ 7.1 เปรียบเทียบสมรรถนะของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ

ประสิทธิภาพการทำงาน	ชนิดของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR			
	หลักการดั้งเดิม	หลักการกักเก็บสัญญาณตอบกลับ	การปรับปรุงขนาดหน้าต่าง ¹	การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม
ขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง	เกิด	เกิด	ไม่เกิด	ไม่เกิด
ขีดจำกัดจากอัตราการส่งข้อมูล	เกิด	ไม่เกิด	ไม่เกิด	ไม่เกิด
ความหนาทาน (เมื่อมีทราฟฟิกที่เปลี่ยนแปลงตลอดเวลาในโครงข่าย)	ทน	ทน	ไม่ทน	ทน

ตารางที่ 7.1 เปรียบเทียบสมรรถนะของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ (ต่อ)

ประสิทธิภาพการทำงาน	ชนิดของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR			
	หลักการดั้งเดิม	หลักการกักเก็บสัญญาณตอบกลับ	การปรับปรุงขนาดหน้าต่าง ¹	การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม
ความเท่าเทียมกัน (เมื่อระยะทางระหว่างต้นทางและปลายทางไม่เท่ากัน)	ไม่เท่าเทียมกัน	ไม่เท่าเทียมกัน	เท่าเทียมกัน	เท่าเทียมกัน
เวลาที่ใช้เพื่อเข้าสู่จุดทำงาน	ช้า	ช้า	เร็ว	เร็ว
ค่าวิสัยสามารถเฉลี่ย	น้อย	น้อย	มาก	มาก
ความยุ่งยาก ²	น้อย	มาก	ปานกลาง	มาก

หมายเหตุ

¹ สำหรับกระบวนการควบคุมการไหลของ TCP โดยการปรับปรุงขนาดหน้าต่าง ที่แสดงในตารางที่ 7.1 ข้างต้น เป็นการแสดงประสิทธิภาพสูงสุดที่กระบวนการนี้ทำได้ ยกตัวอย่างเช่นการเกิดขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง อาจเกิดขึ้นได้ โดยเฉพาะในกรณีที่โครงข่ายมีทราฟฟิกที่มีการเปลี่ยนแปลงร่วมอยู่ด้วย

² ความยุ่งยากของกระบวนการทำงาน วัดจากจำนวนฟังก์ชันที่เพิ่มขึ้นมาเมื่อเทียบกับกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการดั้งเดิม

7.2 ข้อดีข้อเสียของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

สำหรับข้อดีและข้อเสียของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรมที่เสนอนี้ สามารถแยกแยะได้ดังนี้

ข้อดีของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

1. แก้ไขปัญหาความไม่เหมาะสมของขนาดหน้าต่างที่แหล่งกำเนิดแบบ TCP ใช้ ซึ่งเป็นสาเหตุการเกิดขีดจำกัดในการส่งข้อมูล
2. สามารถลดขนาดหน่วยความจำที่อุปกรณ์ต้นทางได้
3. เมื่อเปรียบเทียบกับกระบวนการควบคุมการไหลของ TCP ชนิดอื่น ๆ กระบวนการควบคุมการไหลของ TCP แบบนี้ มีประสิทธิภาพการทำงานสูงสุด (ในที่นี้หมายถึงประสิทธิภาพการ

ทำงานของหลักการนี้ อาจไม่สูงกว่า แต่ก็ไม่ได้ด้อยกว่าเช่นกัน) ไม่ว่าในแง่ของค่าวิสัยสามารถ ความเท่าเทียมกัน หรือความเร็วในการเข้าสู่จุดทำงาน

ข้อเสียของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

1. กระบวนการทำงานมีความยุ่งยากซับซ้อนมาก แต่ประสิทธิภาพการทำงานไม่ได้เพิ่มขึ้นมากนัก

7.3 ข้อเสนอแนะ

เนื่องจากในวิทยานิพนธ์ฉบับนี้ กำหนดให้อุปกรณ์สวิตช์ ATM ที่ใช้มีหน่วยความจำในการรองรับข้อมูลมากเพียงพอที่จะไม่ทำให้ข้อมูลล้นบัฟเฟอร์ อีกทั้งในการส่งข้อมูลไม่ได้พิจารณาถึงผลของการเกิดความผิดพลาดขึ้น อย่างไรก็ตามได้มีการวิเคราะห์ปัญหาส่วนนี้แล้วในภาคผนวก ข แต่อย่างไรก็ตามน่าจะมีการทดสอบระบบ เพื่อพิจารณาสมรรถนะของกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ในกรณีพิจารณาความผิดพลาดของข้อมูลด้วย

สำหรับเรื่องที่น่าสนใจในการทำวิจัยในหัวข้อข้างเคียง นั่นคือการทำงานของ TCP หรือ IP บนโครงข่าย ATM ในแง่พัฒนาอัลกอริทึมเพื่อทำให้การทำงานร่วมกันของโพรโตคอลทั้ง 2 ชั้นมีคุณภาพมากขึ้น กล่าวคือเนื่องจากจุดเด่นอย่างหนึ่งของเทคโนโลยี ATM คือความสามารถในการรองรับข้อมูลที่มี QoS ได้อย่างมีประสิทธิภาพ ทั้งนี้กระบวนการแปลง QoS ที่ใช้ในชั้น ATM มาใช้ในชั้น TCP จัดเป็นเรื่องที่น่าสนใจ

รายการอ้างอิง

- [1] The ATM Forum. The ATM Forum Traffic Management Specification Version 4.0. Apr. 1996.
- [2] Stallings, W. High-Speed Networks: TCP/IP and ATM Design Principles. New Jersey: Prentice Hall, 1998.
- [3] Kwok, T. ATM: The New Paradigm for Internet, Intranet, and Residential Broadband Services and Applications. New Jersey: Prentice Hall, 1998.
- [4] Jain, R., et al. ERICA Switch Algorithm: A Complete Description. ATM Forum/96-1172, Aug. 1996.
- [5] Kalyanaraman, S., et al. The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks. IEEE/ACM Transactions on Networking. Vol.8, NO.1, (Feb. 2000): 87-98.
- [6] Arulambalam, A., Chen, X., and Ansari, N. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. IEEE Communications Magazine, Vol.34, Issue 11, (Nov. 1996): 92-100.
- [7] Arulambalam, A., Chen, X., and Ansari, N. An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks. Proc. ICC 97, Montreal Canada, (Jun. 1997): 200-204.
- [8] Chiussi, F. M., Arulambalam, A., Xia, Y., and Chen, X. Explicit Rate ABR Scheme Using Traffic Load as Congestion Indicator. Proc. Computer Communications and Network 97, (Aug. 1997): 76-84.
- [9] Stevens, W. R. TCP/IP Illustrated, Volume 1: The Protocols. Vol.1. Massachusetts: Addison Wesley, 1994.
- [10] Narvaez, P., and Siu, K. Y. An Acknowledgment Bucket Scheme for Regulation TCP Flow over ATM. Computer Networks and ISDN System, Vol.30, No 19, (Oct. 1998): 1775-1791.
- [11] Jung, J., and Lee, I. TCP Window Control Scheme Using ER Value in TCP over ABR Services. Electronics Letters, Vol.357, (Apr. 1999): 546-548.
- [12] Satyavolu, R., Duvedi, K., and Kalyanaraman, S. Explicit Rate Control of TCP Applications. ATM Forum/98-0152R1, Feb. 1998.

- [13] Gurski, R. J., and Williamson, C. L. TCP over ATM: Simulation Model and Performance Results. Computers and Communications, 1996, Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on, (1996): 328-335.
- [14] An, L., Ansari, N., and Arulambalam, A. TCP/IP Traffic over ATM Networks with ABR Flow and Congestion Control. Global Telecommunications Conference, 1997, GLOBECOM'97, IEEE Vol.3, (1997): 1845-1849.
- [15] Kalyanaraman, S., et al. Performance of TCP/IP over ABR Service on ATM Networks. Global Telecommunications Conference, 1996, GLOBECOM'96, Communications The Key to Global Prosperity Vol.1, (1996): 468-475.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



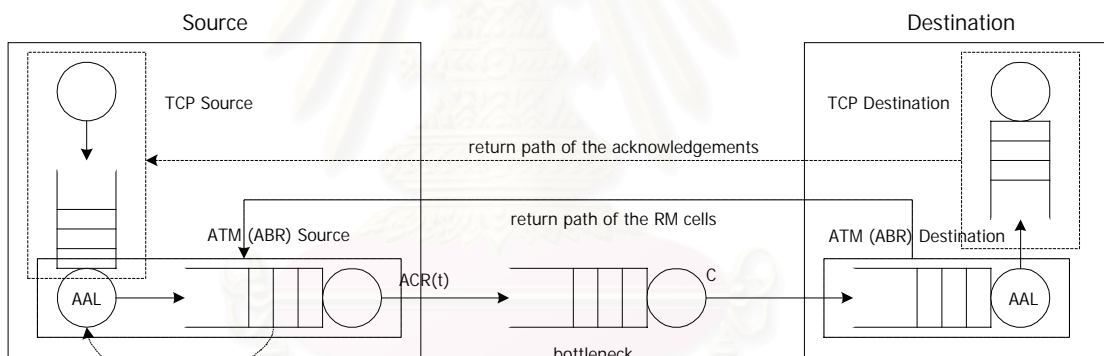
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การวิเคราะห์ขนาดหน้าต่างความคับคั่ง และปริมาณข้อมูลในหน่วยความจำของอุปกรณ์ต้นทางในช่วง Slow Start

บทวิเคราะห์บทนี้จะกล่าวถึงผลที่เกิดจากการทำงานของกระบวนการควบคุมการไหลของ TCP ในช่วง Slow Start (สามารถศึกษาหลักการพื้นฐานของกระบวนการควบคุมการไหลของ TCP ได้ในบทที่ 3) เพื่อความง่ายในการวิเคราะห์ และการทำความเข้าใจ จึงจะพิจารณาผลของกระบวนการควบคุมการไหลของ TCP บนช่องสัญญาณที่มีความสามารถในการส่งคงที่ (ซึ่งเป็นพื้นฐานในการพิจารณาผลที่เกิดจากกรณีอื่น ๆ) แต่เนื่องจากงานวิจัยฉบับนี้เป็นการศึกษากระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM แสดงดังแบบจำลองรูป ก1 ดังนั้นช่องสัญญาณที่ใช้ในการศึกษาครั้งนี้ จึงเกิดจากการพิจารณากระบวนการทำงานในชั้น ATM รวมไปถึงด้วย (นั่นคือพิจารณาตั้งแต่ AAL ต้นทางจนถึง AAL ที่ปลายทาง)



รูปที่ ก1 แบบจำลองการทำงาน of TCP บนบริการแบบ ABR ในโครงข่าย ATM

หมายเหตุ

ในแบบจำลอง ก1 ประกอบด้วยบัฟเฟอร์ 5 ตัว (ตามลำดับจากซ้ายไปขวา) กล่าวคือ

- บัฟเฟอร์ที่แหล่งกำเนิด TCP : เก็บข้อมูลทั้งหมดที่ส่งออกมาจากแหล่งกำเนิด TCP ที่ยังไม่ถูกส่งไปยังกระบวนการแบ่งเซลล์
- บัฟเฟอร์ที่แหล่งกำเนิด ABR : เก็บข้อมูลที่ถูกแบ่งเป็นเซลล์แล้ว แต่ยังไม่สามารถส่งออกไปยังโครงข่าย
- บัฟเฟอร์ที่สายส่ง : แทนบัฟเฟอร์ที่สะสมในอุปกรณ์ระหว่างทางทั้งหมด เช่นในอุปกรณ์สวิตช์ เป็นต้น
- บัฟเฟอร์ที่ ABR ปลายทาง : เก็บข้อมูลที่รับมาจากโครงข่าย (ก่อนรวมเป็นเซกเมนต์)
- บัฟเฟอร์ที่ TCP ปลายทาง : เก็บข้อมูลก่อนส่งต่อไปยังปลายทาง (กำหนดค่าขนาดหน้าต่างปลายทาง (RCVWND))

ในการวิเคราะห์การทำงานจะศึกษาผ่านตัวอย่างดังรูป ก2 ซึ่งแสดงขั้นตอนการส่งข้อมูลระหว่างต้นทาง (ด้านซ้าย) ไปยังปลายทาง (ด้านขวา) โดยนำเสนอภายใต้หน่วยเวลาแบบดิสครีต ทั้งนี้ในการวิเคราะห์กำหนดให้แหล่งกำเนิดข้อมูลมีข้อมูลมากเพียงพอ ที่จะทำให้แหล่งกำเนิดแบบ TCP ส่งออกไปได้อย่างต่อเนื่อง ซึ่งข้อมูลที่ส่งจะถูกแบ่งเป็นเซกเมนต์ (แทนด้วยสัญลักษณ์รูปสี่เหลี่ยม) แล้วค่อยส่งออกไป โดยเซกเมนต์ของข้อมูลจะเคลื่อนที่จากซ้ายไปขวา สำหรับสัญญาณตอบกลับ (ACK) จะเคลื่อนที่ในทิศทางตรงกันข้าม โดยจะแทนด้วยสัญลักษณ์รูปสี่เหลี่ยมที่มีขนาดเล็กกว่า พร้อมระบุหมายเลขเซกเมนต์ที่ตอบกลับด้วย นอกจากนี้กำหนดให้ค่าขนาดหน้าต่างด้านรับ (RCVWND) เท่ากับ 15

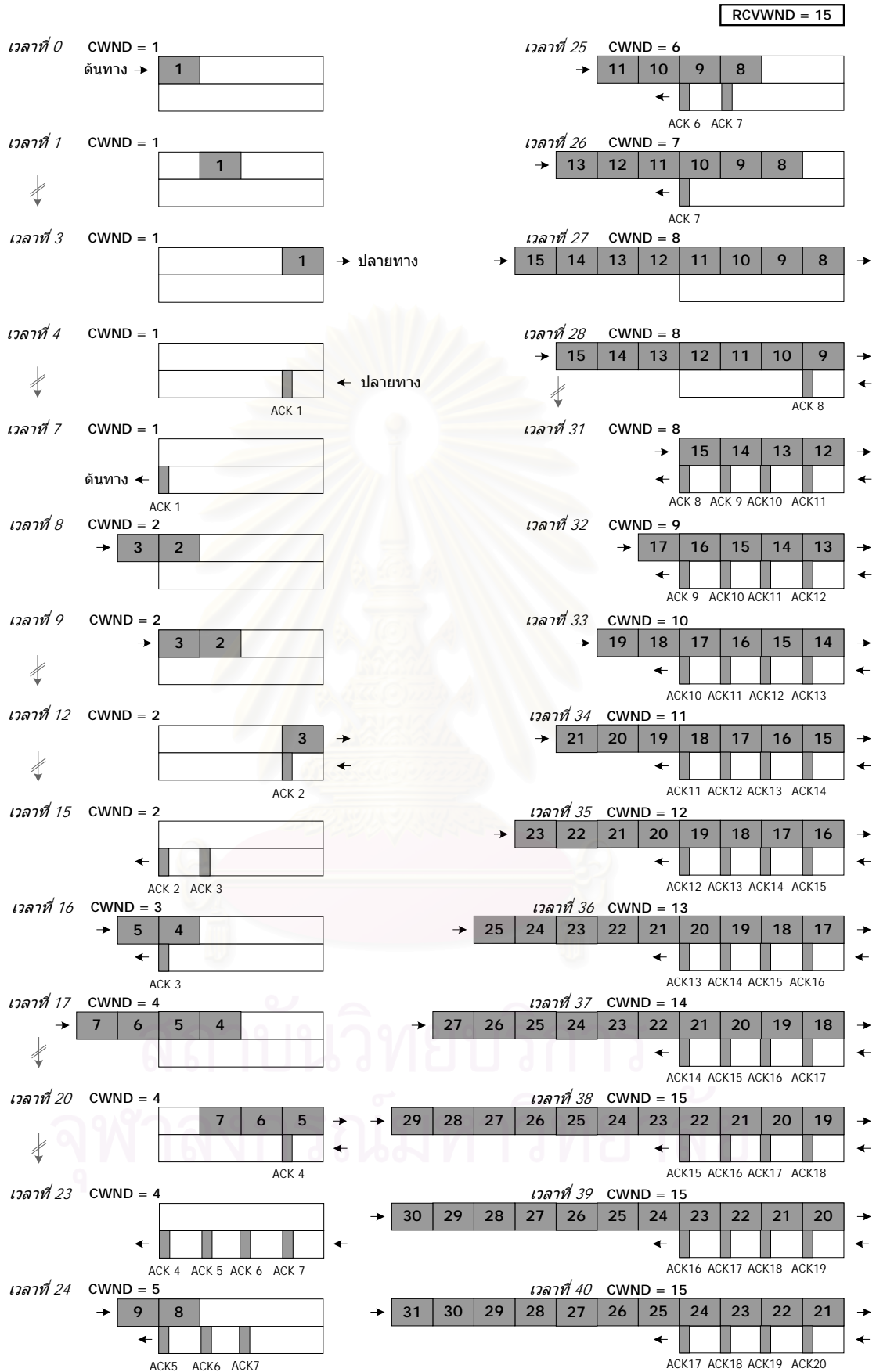
จากรูป ก2 ณ เวลาที่ 0 ต้นทางส่งข้อมูลเซกเมนต์ที่ 1 ออกไป (ค่าขนาดหน้าต่างความคับคั่ง (CWND) เท่ากับ 1) แต่เนื่องจากการทำงานอยู่ในช่วง Slow Start ต้นทางจำเป็นต้องรอรับสัญญาณตอบกลับก่อน จึงจะสามารถส่งข้อมูลเซกเมนต์ต่อไปได้ ดังนั้น ณ เวลาที่ 1 จึงไม่มีการส่งข้อมูล ทั้งนี้ ณ เวลาที่ 1 จนถึง 3 ในรูปเป็นการแสดงให้เห็นถึงการเคลื่อนที่ของเซกเมนต์ที่ 1 เท่านั้น

เมื่อข้อมูลไปถึงปลายทาง ปลายทางจะสร้างสัญญาณตอบกลับ (ACK1) แล้วส่งกลับมายังต้นทาง (ณ เวลาที่ 4) โดย ACK1 จะเคลื่อนที่มาถึงต้นทาง ณ เวลาที่ 7 (เวลาที่ใช้ในการส่งข้อมูลไปกลับ (RTT) ของกรณีตัวอย่างนี้มีค่าเท่ากับ 8 หน่วยเวลา)

เมื่อต้นทางได้รับสัญญาณตอบกลับ (ACK1) ต้นทางก็สามารถส่งข้อมูลได้อีก 2 เซกเมนต์ (เพราะ ค่า CWND ในขณะนี้เพิ่มเป็น 2 แล้ว) เนื่องจากต้นทางสามารถส่งข้อมูลได้หนึ่งเซกเมนต์ต่อหนึ่งหน่วยเวลา ดังนั้น ณ เวลาที่ 8 ต้นทางจึงสามารถส่งข้อมูลเซกเมนต์ที่ 2 เท่านั้น ส่วนข้อมูลเซกเมนต์ที่ 3 จะถูกเก็บในหน่วยความจำ และจะถูกส่งออกไป ณ เวลาที่ 9 โดยข้อมูลทั้ง 2 เซกเมนต์นี้จะไปถึงปลายทาง ณ เวลาที่ 12 และ 13 ตามลำดับ ณ เวลาเดียวกันปลายทางก็ส่งสัญญาณตอบกลับ ACK2 และ ACK3 มายังต้นทาง เพื่อตอบรับการมาถึงของข้อมูลดังกล่าว

เมื่อต้นทางได้รับ ACK2 ขนาดหน้าต่างก็จะเพิ่มขึ้นจาก 2 ไปเป็น 3 (หมายถึง ต้นทางสามารถส่งข้อมูลได้ 3 เซกเมนต์ โดยพิจารณาตั้งแต่ข้อมูลเซกเมนต์ที่ 3 แต่เนื่องจากเซกเมนต์ที่ 3 ถูกส่งไปแล้ว ดังนั้นข้อมูลที่ได้รับอนุญาตให้ส่ง ก็เหลือเพียงข้อมูลเซกเมนต์ที่ 4 และ 5)

ณ เวลาที่ 16 เซกเมนต์ที่ 4 ถูกส่งออกไป โดยเก็บเซกเมนต์ที่ 5 เอาไว้ในหน่วยความจำที่อุปกรณ์ต้นทางก่อน ณ เวลาที่ 17 ข้อมูลเซกเมนต์ที่ 5 จะถูกส่งออกไปยังโครงข่าย ในขณะที่เดียวกัน ACK3 ก็เดินทางมาถึงต้นทาง ทำให้ CWND มีค่าเพิ่มเป็น 4 และปริมาณข้อมูลที่ได้รับอนุญาตให้ส่ง ก็ขยายไปถึงเซกเมนต์หมายเลข 7 ซึ่งข้อมูลทั้ง 4 เซกเมนต์นี้จะถูกส่งออกไปจนหมด ณ เวลาที่ 19

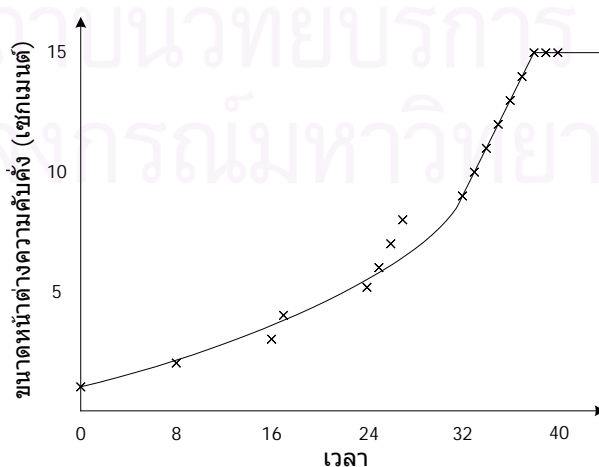


รูปที่ ก2 ตัวอย่างการส่งข้อมูลในช่วง Slow Start

ณ เวลาที่ 24 ถึง 27 ต้นทางได้รับ ACK4 ถึง ACK7 ตามลำดับ ทำให้ในที่สุด CWND ถูกปรับให้มีค่าเท่ากับ 8 และผลของการปรับขนาดหน้าต่างครั้งนี้ ทำให้ต้นทางสามารถส่งข้อมูลได้อย่างต่อเนื่อง และพอดีที่จะเติมเต็มการเชื่อมต่อ อย่างไรก็ตามเมื่อต้นทางได้รับ ACK8 ค่า CWND ก็มีค่าเพิ่มเป็น 9 ทำให้มีข้อมูลค้างอยู่ในหน่วยความจำที่อุปกรณ์ต้นทาง 1 เซกเมนต์ ในทำนองเดียวกันเมื่อ ACK9 ถึง ACK14 มาถึงต้นทาง ณ เวลาที่ 33 ถึง 38 ทำให้ค่า CWND เพิ่มเป็น 10 ถึง 15 ตามลำดับ ซึ่งมีผลให้ปริมาณข้อมูลในหน่วยความจำที่อุปกรณ์ต้นทางเพิ่มขึ้นเรื่อย ๆ จนมีปริมาณถึง 7 เซกเมนต์ดังรูป

เมื่อพิจารณาเวลาที่ 39 และ 40 จะเห็นว่าต้นทางได้รับ ACK15 และ ACK16 แต่ค่า CWND ยังคงมีค่าเท่ากับ 15 เท่านั้น ทั้งนี้เพราะขนาดหน้าต่างปลายทางมีค่าเท่ากับ 15 นั่นเอง ด้วยเหตุนี้เองทำให้ปริมาณข้อมูลในหน่วยความจำของอุปกรณ์ต้นทางมีค่าคงที่ที่ 7 เซกเมนต์

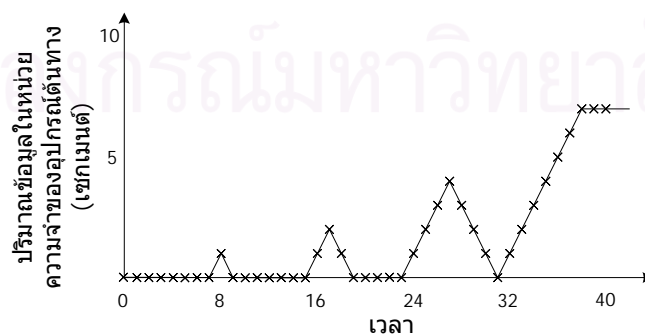
จากตัวอย่างดังกล่าวเมื่อทำการพิจารณาความสัมพันธ์ของขนาดหน้าต่างความคับคั่ง (แสดงดังรูป ก3) จะเห็นว่าลักษณะการเพิ่มของขนาดหน้าต่างความคับคั่งไม่ได้เพิ่มแบบเอกซ์โพเนนเชียลตลอดการทำงานในช่วง Slow Start ของกระบวนการควบคุมการไหลของ TCP เสมอไป โดยเฉพาะในกรณีตัวอย่างที่มีช่องสัญญาณที่มีความสามารถในการรองรับข้อมูลได้จำกัด (bottleneck) การเพิ่มของขนาดหน้าต่างที่เกิดขึ้นสามารถแบ่งออกเป็น 2 ช่วง คือช่วงแรกขนาดหน้าต่างความคับคั่งจะเพิ่มแบบเอกซ์โพเนนเชียล และช่วงที่ 2 จะมีลักษณะเป็นเชิงเส้น โดยจุดแบ่งระหว่างช่วงที่ 1 และ 2 คือจุดที่ขนาดหน้าต่างมีค่าเท่ากับผลการคำนวณ Bandwidth-Delay Product (ในตัวอย่างนี้มีค่าเท่ากับ 1 เซกเมนต์ต่อหน่วยเวลา \times 8 หน่วยเวลา = 8 เซกเมนต์) ซึ่งเป็นค่าขนาดหน้าต่างที่กำหนดปริมาณข้อมูลที่สามารถเติมเต็มช่องสัญญาณได้พอดี (พอดีในที่นี้หมายถึงมีข้อมูลส่งอย่างต่อเนื่องไม่ขาดหรือเกิน)



รูปที่ ก3 ความสัมพันธ์ระหว่างขนาดหน้าต่างความคับคั่งเทียบกับเวลา

เมื่อทำการวิเคราะห์ความสัมพันธ์ระหว่างขนาดหน้าต่างความคับคั่งเทียบกับเวลา จะเห็นว่าขนาดหน้าต่างความคับคั่งจะถูกปรับค่าตามหลักการ Slow Start นั่นคือขนาดหน้าต่างความคับคั่งจะเพิ่มขึ้น 1 เซกเมนต์ ทุกครั้งที่ได้รับสัญญาณตอบกลับหนึ่งตัว ทั้งนี้ในช่วงแรกของการเชื่อมต่อ ต้นทางจะได้รับสัญญาณตอบกลับในปริมาณที่เพิ่มขึ้น 2 เท่า ทุก ๆ รอบ RTT เพราะฉะนั้นขนาดหน้าต่างความคับคั่งก็จะเพิ่มขึ้น 2 เท่า ทุก ๆ รอบ RTT ด้วย ดังนั้นความสัมพันธ์ของขนาดหน้าต่างความคับคั่งเทียบกับเวลาในช่วงนี้จึงมีการเพิ่มขึ้นแบบเอกซ์โพเนนเชียล ทั้งนี้สัญญาณตอบกลับมิได้เพิ่มอัตราการส่งเป็น 2 เท่าเสมอไป เพราะเมื่อแหล่งกำเนิดข้อมูลสามารถส่งข้อมูลจนเต็มความสามารถในการรองรับของช่องสัญญาณแล้ว อัตราการมาถึงของสัญญาณตอบกลับจะคงที่ ทำให้การเพิ่มขนาดหน้าต่างมีอัตราคงที่ด้วย นั่นคือความสัมพันธ์ของขนาดหน้าต่างความคับคั่งเทียบกับเวลาที่มีการเพิ่มแบบเชิงเส้นนั่นเอง

นอกจากนี้การเพิ่มขนาดหน้าต่างความคับคั่งของ TCP ยังมีผลต่อการสะสมข้อมูลที่หน่วยความจำของอุปกรณ์ต้นทางอีกด้วย ทั้งนี้เมื่อพิจารณาตัวอย่าง ก2 สามารถแสดงความสัมพันธ์ระหว่างปริมาณข้อมูลในหน่วยความจำเทียบกับเวลาได้ดังรูป ก4 จะเห็นว่าในช่วงแรกปริมาณข้อมูลในหน่วยความจำจะมีลักษณะขึ้น ๆ ลง ๆ และมีค่าเป็นศูนย์เป็นช่วง ๆ ซึ่งลักษณะที่เกิดขึ้นนี้เกิดจากขนาดหน้าต่างมีค่าน้อยเกินไปจนเกิดภาวะขาดแคลนข้อมูลที่ใช้ในการส่ง แต่เมื่อขนาดหน้าต่างเพิ่มขึ้นจนทำให้มีปริมาณข้อมูลที่ส่งออกจากแหล่งกำเนิด TCP มากเกินไปจนไม่สามารถส่งข้อมูลได้หมด ทั้งนี้เพราะช่องสัญญาณถูกใช้อย่างเต็มประสิทธิภาพแล้ว (พิจารณาจากการที่แหล่งกำเนิด TCP มีขนาดหน้าต่างเท่ากับค่า Bandwidth-Delay Product) ทำให้มีข้อมูลเหลือค้างจากการส่ง และถูกสะสมในอุปกรณ์ส่วนนี้ ทั้งนี้ระดับแถวคอยในส่วนนี้สามารถคำนวณได้จากขนาดหน้าต่างที่ใช้ลบด้วยปริมาณข้อมูลที่ส่งได้ ซึ่งในที่นี้ค่าเท่ากับ Bandwidth-Delay product (จากตัวอย่างนี้ระดับแถวคอยค่าสุดท้ายที่คำนวณได้จึงเท่ากับ $15 - 8 = 7$ ซึ่งสอดคล้องกับภาพตัวอย่างที่แสดง)



รูปที่ ก4 ความสัมพันธ์ระหว่างปริมาณข้อมูลในหน่วยความจำของอุปกรณ์ต้นทางเทียบกับเวลา

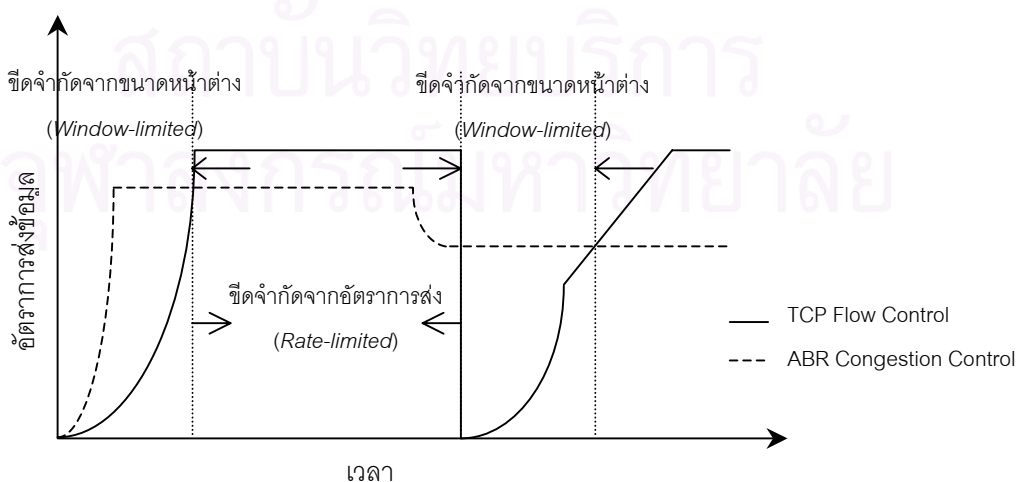
ภาคผนวก ข

การวิเคราะห์ผลของความผิดพลาดที่เกิดกับข้อมูล ต่อกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ชนิดต่าง ๆ

สำหรับในบทนี้จะเป็นการวิเคราะห์กรณีที่เกิดความผิดพลาดขึ้นกับข้อมูล หรือข้อมูลเกิดสูญหายระหว่างทาง โดยสาเหตุของความผิดพลาดที่พิจารณาในที่นี้ เกิดจากสัญญาณรบกวนที่สายส่ง และบัพเฟอร์ล้นที่อุปกรณ์ต้นทาง ส่วนสาเหตุที่เกิดจากบัพเฟอร์ล้นที่สวิตช์ ATM นั้นจะไม่พิจารณาในที่นี้ เพราะถือว่าภาวะดังกล่าวจะไม่เกิดขึ้นเลย ถ้าใช้กระบวนการควบคุมความคับคั่งของ ABR ที่มีประสิทธิภาพ ทั้งนี้กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR ในโครงข่าย ATM สามารถแบ่งออกเป็น 4 ประเภท คือ

- การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการดั้งเดิม

เนื่องจากหลักการนี้เป็นการทำงานของกระบวนการควบคุมการไหลของ TCP แบบดั้งเดิม บนกระบวนการควบคุมความคับคั่งของ ABR ดังนั้นเมื่อมีความผิดพลาดเกิดขึ้น TCP ก็ทำงานตามหลักการของ TCP แบบดั้งเดิม นั่นคือขนาดหน้าต่างความคับคั่งจะถูกแบ่งครึ่ง แล้วบันทึกลงในค่า Slow Start Threshold ก่อนจะตั้งค่าขนาดหน้าต่างให้เท่ากับหนึ่งหน่วยข้อมูล แล้วค่อย ๆ เพิ่มขนาดหน้าต่างแบบ Slow Start อีกครั้ง และเมื่อขนาดหน้าต่างเพิ่มขึ้นจนมีค่าเท่ากับ Slow Start Threshold แล้ว รูปแบบการเพิ่มค่าขนาดหน้าต่างจะเปลี่ยนไป คือจะมีลักษณะเป็นเชิงเส้นเมื่อเทียบกับเวลา ดังรูปที่ ข1



รูปที่ ข1 กระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้หลักการดั้งเดิม

จากรูปจะเห็นว่ากระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR โดยใช้หลักการดั้งเดิมนี้ จะเกิดขีดจำกัดอันเนื่องมาจากขนาดหน้าต่างทุกครั้งที่มีข้อมูลผิดพลาด หรือสูญหาย (คือเมื่อกระบวนการส่งข้อมูลซ้ำถูกใช้) ทั้งนี้จะเห็นว่าหลักการการทำงานของกระบวนการส่งข้อมูลซ้ำของ TCP เป็นการส่งข้อมูลชุดที่มีปัญหาใหม่ โดย TCP มีเชื่อว่าสาเหตุของความผิดพลาดที่เกิดกับข้อมูลที่ปลายทางได้รับนั้น มาจากความคับคั่งของอุปกรณ์ภายในโครงข่าย ดังนั้นแหล่งกำเนิดแบบ TCP จึงแก้ปัญหา โดยการลดปริมาณทราฟฟิก และเปิดโอกาสให้อุปกรณ์ในโครงข่ายระบายข้อมูลที่ค้างคาออกไปเสีย นั่นคือ TCP จะเริ่มส่งข้อมูลที่มีปริมาณน้อย ๆ ใหม่อีกครั้ง แต่เนื่องจากการทำงานของ TCP บนกระบวนการควบคุมความคับคั่งแบบ ABR มีความเป็นไปได้ในการเกิดความคับคั่งภายในโครงข่ายน้อยมาก ดังนั้นสาเหตุหลักในการเกิดความผิดพลาดของข้อมูล จึงมาจากการเกิดสัญญาณรบกวนที่สายส่ง หรือไม่ก็เกิดการล้นของบัฟเฟอร์ที่อุปกรณ์ต้นทาง ถ้าสาเหตุการเกิดความผิดพลาดเป็นเรื่องของสัญญาณรบกวนแล้ว การลดปริมาณข้อมูลตามหลักการดั้งเดิมแบบนี้ ถือได้ว่าเป็นการแก้ไขปัญหามิจุด กล่าวคือ การลดปริมาณทราฟฟิกไม่ช่วยให้อะไรดีขึ้น อีกทั้งยังส่งผลให้ประสิทธิภาพการทำงานของระบบลดลง อย่างไรก็ตามถ้าสาเหตุของการเกิดความผิดพลาดมาจากการล้นของข้อมูลที่อุปกรณ์ต้นทาง กระบวนการส่งข้อมูลใหม่ตามหลักการดั้งเดิมนี้ สามารถลดปัญหาดังกล่าวได้ กล่าวคือเมื่อ TCP ต้นทางส่งข้อมูลน้อยลง บัฟเฟอร์ที่อุปกรณ์ต้นทางก็สามารถระบายข้อมูลได้ ทำให้ระดับข้อมูลในบัฟเฟอร์ส่วนนี้ลดลง นั่นคือเป็นการลดความเสี่ยงในการเกิดการล้นของข้อมูล ทั้งนี้ทั้งนั้นความเสี่ยงในการเกิดการล้นของข้อมูลในอุปกรณ์ต้นทางนี้ สามารถพิจารณาจากการทำงานในช่วงที่เกิดขีดจำกัดอันเนื่องมาจากอัตราการส่ง (กล่าวคือ ถ้าปริมาณข้อมูลที่ส่งได้จริง มีค่าน้อยกว่าปริมาณข้อมูลที่กำหนดโดยแหล่งกำเนิดแบบ TCP มาก อัตราความเสี่ยงที่จะเกิดการล้นของข้อมูลในส่วนนี้ก็จะมากขึ้นด้วย พิจารณาได้จากกราฟวิเคราะห์ในภาคผนวก ก)

- **การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ**

เนื่องจากจุดมุ่งหมายในการออกแบบกระบวนการควบคุมการไหลของ TCP โดยใช้หลักการกักเก็บสัญญาณตอบกลับ คือต้องการลดขนาดหน่วยความจำที่ใช้ในอุปกรณ์ต้นทาง โดยกระบวนการควบคุมปริมาณการส่งข้อมูลยังคงใช้หลักการเดียวกันกับหลักการควบคุมการไหลของ TCP แบบดั้งเดิม ทั้งนี้ร่วมไปถึงกระบวนการส่งข้อมูลซ้ำด้วย เนื่องจากการควบคุมการส่งข้อมูลโดยใช้หลักการกักเก็บสัญญาณตอบกลับนี้ สามารถลดปริมาณหน่วยความจำที่ใช้ที่ต้นทางได้ ดังนั้นสาเหตุของการเกิดความผิดพลาดของข้อมูล จึงเกิดจากผลของสัญญาณรบกวนที่สายส่งเท่า

นั้น (ความเป็นไปได้ในการเกิดการล้นของข้อมูลที่อุปกรณ์ต้นทาง ในกรณีนี้ถือว่าน้อยมาก เพราะหลักการกักเก็บสัญญาณตอบกลับ จะส่งข้อมูลตามความต้องการของระบบ ที่วัดโดยกระบวนการควบคุมความคับคั่งของ ABR ทำให้ปริมาณบัพเฟอร์ที่คั่งค้างในอุปกรณ์ต้นทางมีค่าน้อยมาก) สำหรับผลของกระบวนการทำงานของหลักการกักเก็บสัญญาณตอบกลับ เมื่อมีความผิดพลาดเกิดขึ้นกับข้อมูลในระบบ จะเหมือนผลที่เกิดจากการทำงาน โดยใช้หลักการแบบดั้งเดิม นั่นคือจะเกิดขีดจำกัดอันเนื่องมาจากขนาดหน้าต่าง ขึ้นทุกครั้งที่มีการตรวจพบความผิดพลาด เนื่องจากสาเหตุของการเกิดความผิดพลาดของกรณีนี้ มาจากสัญญาณรบกวนที่สายส่ง ดังนั้นการแก้ปัญหาโดยการลดปริมาณข้อมูลที่ส่ง จึงเป็นการแก้ปัญหาที่ไม่ก่อให้เกิดประโยชน์ อีกทั้งยังเป็นการลดประสิทธิภาพในการทำงานของระบบอีกด้วย

- การควบคุมอัตราการส่งข้อมูลของ TCP โดยการปรับรูปร่างหน้าต่างต่าง

ผู้คิดค้นการทำงานของกระบวนการควบคุมการไหลของ TCP โดยการปรับรูปร่างหน้าต่างต่าง ไม่ได้กล่าวถึงกระบวนการรองรับการทำงาน เมื่อระบบตรวจพบข้อมูลที่ผิดพลาดขึ้น

- การควบคุมอัตราการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งแบบเซลล์เป็นเฟรม

สำหรับกระบวนการส่งข้อมูลซ้ำของหลักการแปลงอัตราการส่งแบบเซลล์เป็นเฟรม ที่ใช้ในกระบวนการควบคุมการไหลของ TCP บนบริการแบบ ABR นั้น จะเป็นเพียงการส่งข้อมูลเฟรมที่ตรวจสอบพบว่ามีผิดพลาดเกิดขึ้นใหม่อีกครั้ง (พิจารณาจากค่า ACK ที่ติดมากับสัญญาณตอบกลับ ซึ่งคือค่า Incoming ACK number นั้นเอง) โดยไม่ลดอัตราการส่ง ทั้งนี้เพราะสาเหตุการเกิดความผิดพลาดที่เป็นไปได้ในกรณีนี้ เกิดขึ้นจากการรบกวนของสัญญาณภายนอกต่อสัญญาณในสายส่งเท่านั้น (ส่วนสาเหตุอื่น ๆ เช่นการล้นของบัพเฟอร์ที่อุปกรณ์ต้นทาง หรือการล้นของบัพเฟอร์ที่สวิตช์ ATM มีความเป็นไปได้เล็กน้อย หรือแทบจะเป็นไปไม่ได้เลย เพราะหลักการทำงานของการส่งข้อมูลของ TCP โดยใช้การแปลงอัตราการส่งข้อมูลแบบเซลล์เป็นเฟรมนี้ เป็นการควบคุมการส่งข้อมูลจากอุปกรณ์ต้นทางให้สอดคล้องกับการส่งข้อมูลจริงมากที่สุด จึงทำให้ไม่มีการกักเก็บข้อมูลในบัพเฟอร์ที่อุปกรณ์ต้นทางก่อนส่ง ส่วนปัญหาบัพเฟอร์ที่สวิตช์ ATM นั้นจะถูกควบคุมอย่างมีประสิทธิภาพอยู่แล้ว โดยกระบวนการควบคุมความคับคั่งของ ABR)

สำหรับหลักการทำงานของกระบวนการส่งข้อมูลซ้ำในกรณีนี้ จะเริ่มจากที่ต้นทางตรวจสอบค่า Incoming ACK number ของสัญญาณตอบกลับซ้ำกัน ซึ่งบ่งบอกถึงความผิดพลาดใน

การส่งข้อมูล ดังนั้นต้นทางจำเป็นต้องส่งข้อมูลเฟรมนี้ใหม่อีกครั้ง ทั้งนี้ตัว ACK manager เป็นผู้ส่งข้อมูลดังกล่าวเอง เนื่องจากเมื่อแหล่งกำเนิดแบบ TCP ถูกบล็อกด้วยการส่ง ACK ปลอมจาก ACK manager ดังนั้นแหล่งกำเนิด TCP จึงไม่สามารถย้อนกลับไปส่งข้อมูลชุดดังกล่าวได้อีก ทั้งนี้หน้าที่ส่งข้อมูลซ้ำจึงตกมาอยู่ที่ ACK manager ดังนั้น ACK manager จึงจำเป็นต้องนี้หน่วยความจำอย่างน้อยเท่ากับขนาดหน้าต่างปลายทางของ TCP เพื่อทำหน้าที่ดังกล่าว



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาววีณา ว่างเจริญ เกิดเมื่อวันที่ 14 เมษายน พ.ศ. 2518 ที่อำเภอเมือง จังหวัดปัตตานี สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต (เกียรตินิยมอันดับสอง) ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ในปีการศึกษา 2537 และทำงานเป็นวิศวกร ที่บริษัท พลัส อินฟอร์เมชัน เซอร์วิส จำกัด หลังจากนั้นเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2541



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย