

การปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัต  
สำหรับถังร้วที่มีการจัดลำดับความสำคัญใน โครงข่ายเอทีเอ็ม โดยใช้พีชชีลอจิก



นายอภิศักดิ์ สันธนะ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

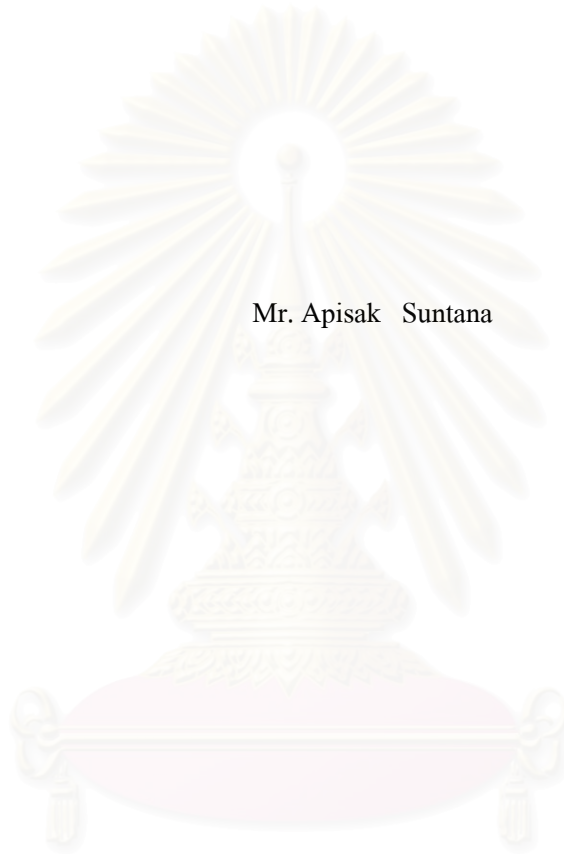
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2544

ISBN 974-03-0474-5

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DYNAMIC QUEUE LENGTH THRESHOLD SCHEDULING POLICY  
FOR A PRIORITY LEAKY BUCKET IN AN ATM NETWORK USING FUZZY LOGIC



Mr. Apisak Suntana

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 974-03-0474-5

หัวข้อวิทยานิพนธ์ การปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัตสำหรับถังรั่ว  
ที่มีการจัดลำดับความสำคัญใน โครงข่ายเอทีเอ็ม โดยใช้พีซีแอลจิก  
โดย นายอภิศักดิ์ ลั่นชนะ  
สาขาวิชา วิศวกรรมไฟฟ้า  
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง  
ของการศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
( ศาสตราจารย์ ดร.ประสิทธิ์ ประพัฒน์มงคล )

..... อาจารย์ที่ปรึกษา  
( รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล )

..... กรรมการ  
( ผู้ช่วยศาสตราจารย์ ดร.ลัญจกร วุฒิสัทธาธิกุลกิจ )

อภิศักดิ์ สันชนะ : การปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัตสำหรับถังรั่วที่มีการจัดลำดับความสำคัญในโครงข่ายเอทีเอ็มโดยใช้ฟuzzyลอจิก (DYNAMIC QUEUE LENGTH THRESHOLD SCHEDULING POLICY FOR A PRIORITY LEAKY BUCKET IN AN ATM NETWORK USING FUZZY LOGIC) อ. ที่ปรึกษา : รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล, 76 หน้า. ISBN 974-03-0474-5

วิทยานิพนธ์ฉบับนี้เสนออัลกอริทึมถังรั่วที่มีการจัดลำดับความสำคัญ (Priority Leaky Bucket Algorithm) ในโครงข่ายเอทีเอ็ม ที่ใช้ตัวควบคุมแบบฟuzzyลอจิก (Fuzzy Logic Controller) ทำหน้าที่ปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัตให้เหมาะสมกับสถานะของทราฟฟิกที่มีอัตราบิตข้อมูลที่เปลี่ยนแปลงตามเวลา (Variable Bit Rate) โดยพิจารณาถึงความต้องการของคุณภาพการบริการ (Quality of Service) ของทราฟฟิกแต่ละแบบ ได้แก่ ทราฟฟิกแบบเวลาจริง (Real-Time Traffic) ที่ต้องการเวลาการคอยในคิวโดยเฉลี่ยของเซลล์ต่ำในขณะที่ยอมให้มีอัตราส่วนการสูญหายของเซลล์ได้ในระดับหนึ่ง และทราฟฟิกแบบไม่เป็นเวลาจริง (Non Real-Time Traffic) ที่ต้องการให้มีอัตราส่วนการสูญหายของเซลล์ต่ำในขณะที่ยอมให้มีเวลาการคอยในคิวโดยเฉลี่ยของเซลล์นานกว่าได้ วิธีการปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัตโดยใช้ตัวควบคุมแบบฟuzzyลอจิกช่วยจัดสรรการใช้งานโทเค็นให้เป็นอย่างดีมีประสิทธิภาพตรงกับความต้องการของทราฟฟิกในขณะนั้น ทำให้สามารถลดอัตราส่วนการสูญหายของเซลล์ที่เกิดขึ้นในทราฟฟิกทั้งสองแบบได้เมื่อเปรียบเทียบกับการตั้งระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบคงที่

เมื่อเปรียบเทียบสมรรถนะในการจัดการกับเซลล์ที่สถานะการส่งข้อมูลในอัตราเซลล์เฉลี่ยที่ไม่เกินขีดตกลงและที่ละเมิดขีดตกลง พบว่าที่สถานะการส่งข้อมูลในอัตราเซลล์เฉลี่ยที่ไม่เกินขีดตกลง วิธีการปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิวแบบพลวัตโดยใช้ตัวควบคุมแบบฟuzzyลอจิกช่วยเพิ่มเวลาเฉลี่ยก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ในทราฟฟิกทั้งสองแบบได้ จึงทำให้ลดความน่าจะเป็นของการเกิด False Alarm ลงได้ ซึ่งต้องแลกกับเวลาเฉลี่ยก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ที่เพิ่มขึ้นในขณะที่การส่งข้อมูลในอัตราเซลล์เฉลี่ยที่ละเมิดขีดตกลงทำให้สมรรถนะในการจัดการกับเซลล์ที่ละเมิดขีดตกลงลดลงเล็กน้อย นอกจากนี้ผลการทดสอบได้แสดงถึงผลของการปรับเปลี่ยนพารามิเตอร์ของแบบจำลองได้แก่ ขนาดบัฟเฟอร์ และขนาดถังเก็บโทเค็น ซึ่งมีผลกระทบต่อสมรรถนะของแบบจำลองนี้อีกด้วย

ภาควิชา ..... วิศวกรรมไฟฟ้า .....

ลายมือชื่อนิสิต .....

สาขาวิชา ..... วิศวกรรมไฟฟ้า .....

ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา ..... 2544 .....

# # 4070567121: MAJOR ELECTRICAL ENGINEERING

KEYWORD: LEAKY BUCKET/ FUZZY LOGIC/ ATM NETWORKS.

APISAK SUNTANA: DYNAMIC QUEUE LENGTH THRESHOLD SCHEDULING POLICY FOR A PRIORITY LEAKY BUCKET IN AN ATM NETWORK USING FUZZY LOGIC. THESIS ADVISOR: ASSOC. PROF. Dr.WATIT BENJAPOLAKUL. 76 pp. ISBN 974-03-0474-5

This Thesis proposes a dynamic queue length threshold scheduling policy for priority leaky bucket algorithm in an ATM network. The proposed model uses fuzzy logic controller to make soft decision in dynamic queue length threshold scheduling policy based on traffic condition and Quality of Service (QoS). Real-time Variable Bit Rate (rt-VBR) traffic is sensitive to mean waiting time but insensitive to cell loss ratio. Non real-time Variable Bit Rate (nrt-VBR) traffic is sensitive to cell loss ratio but insensitive to mean waiting time. Dynamic queue length threshold scheduling policy using fuzzy logic increases token utilization and can meet the QoS requirements of both types of traffic. Compared to the fixed queue length threshold, this proposed model decrease cell loss ratio of both types of traffic.

The performance of controlling the violating traffic source and non violating traffic source, in terms of the average cell rate, are compared. In the case of non violating traffic source, dynamic queue length threshold scheduling policy using fuzzy logic gives a longer average time before the first loss of real-time or non real-time traffic is detected. Thus, the false alarm probability is decreased. This trade off on the violating traffic increases average time before the first loss is detected, compared to that of fixed queue length threshold. The effects of system parameters, such as the buffer size and token bucket size, on the performance of the proposed model are also investigated.

Department	Electrical Engineering	Student's signature
Field of study	Electrical Engineering	Advisor's signature
Academic year	2001	

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลืออย่างดียิ่งของรองศาสตราจารย์ ดร.วาทิต เบญจพลกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ในการวิจัยด้วยดีตลอดมา ขอขอบคุณโครงการทุนอุดหนุนและส่งเสริมวิทยานิพนธ์ระดับปริญญาโท-เอก ที่สนับสนุนโดยให้ทุนวิจัย และเพื่อนๆ นิสิต สาขาระบบโทรคมนาคมทุกๆ ท่าน ที่เป็นกำลังใจและให้ความช่วยเหลือด้วยดีมาตลอด

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา-มารดา และผู้มีพระคุณทุกท่าน ซึ่งให้การสนับสนุนและให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

นายอภิศักดิ์ สันธนะ



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

โครงข่ายเอทีเอ็ม (Asynchronous Transfer Mode หรือ ATM) เป็นโครงข่ายสื่อสารความเร็วสูงที่ออกแบบมาเพื่อให้สามารถรองรับการส่งข้อมูลได้หลายประเภท ทั้งข้อมูลที่เป็นสัญญาณเสียง สัญญาณภาพ และ สัญญาณวิดีโอ อีกทั้งยังสามารถส่งด้วยอัตราเร็วต่างกัน ซึ่งการที่จะทำให้โครงข่ายสามารถทำงานได้อย่างมีประสิทธิภาพ จำเป็นจะต้องมีตัวควบคุมที่คอยทำหน้าที่ควบคุมการส่งข้อมูลให้เป็นไปตามที่กำหนดไว้ นั่นคือการทำงานของตัวควบคุมพารามิเตอร์การใช้ (Usage Parameter Control หรือ UPC) UPC ที่ใช้มีอยู่ด้วยกันหลายชนิด แตกต่างกันไปตามลักษณะการทำงานและอัลกอริทึม (Algorithm) ที่ใช้ได้แก่ Moving Window, Fixed Windows, Exponentially Weighted Moving Average และ ถังรั่ว (Leaky Bucket) จาก [1, 2] แสดงให้เห็นว่าอัลกอริทึมถังรั่วมีประสิทธิภาพสูงในการจัดการได้รวดเร็ว และจัดการกับข้อมูลที่มีลักษณะเป็นเบิสต์ได้ดีกว่าวิธีอื่นๆ

ถังรั่วแบบใช้โทเค็น (Token Leaky Bucket) จะมีถังเก็บโทเค็น (Token Bucket หรือ Token Pool) ไว้สำหรับเก็บโทเค็นซึ่งโทเค็นจะเปรียบเสมือนเป็นเครดิตในการที่จะให้เซลล์ผ่านเข้าไปในโครงข่าย ซึ่งเซลล์จะถูกส่งเข้าไปในโครงข่ายได้ก็ต่อเมื่อต้องมีโทเค็นคู่กันเข้าไป 1 เซลล์ต่อ 1 โทเค็นและเมื่อส่งเซลล์ออกไปจำนวนโทเค็นที่มีอยู่ก็จะลดลง ซึ่งวิธีถังรั่วแบบโทเค็นมีตัวพารามิเตอร์ที่สำคัญคือ อัตราการสร้างโทเค็นและขนาดของถังเก็บโทเค็น

เนื่องจากปัญหาการประวิงเวลาในโครงข่ายที่มีความเร็วสูงนั้นมีความสำคัญมาก จึงได้มีการเสนอวิธีที่จะลดการประวิงเวลาและเพิ่มประสิทธิภาพของโครงข่าย ด้วยวิธีการใช้ถังเก็บโทเค็นร่วม (Shared Token Pool) ในถังรั่วแบบใช้โทเค็น [3] มีวัตถุประสงค์เพื่อที่จะทำให้แต่ละแหล่งกำเนิดสามารถใช้โทเค็นของแหล่งกำเนิดอื่นๆได้ เนื่องจากใช้ถังเก็บโทเค็นร่วมกัน ซึ่งผลที่ได้ทำให้การประวิงลดลง และประสิทธิภาพของโครงข่ายก็สูงขึ้น แต่วิธีนี้มีข้อเสียคือกรณีที่แหล่งกำเนิดที่มีอัตราการส่งสูงจะไปแย่งโทเค็นของแหล่งกำเนิดที่ส่งในอัตราที่ต่ำกว่า ทำให้เกิดความไม่ยุติธรรมในการใช้ทรัพยากรของโครงข่าย และทำให้แหล่งกำเนิดที่ส่งที่อัตราต่ำมีโอกาสที่จะสูญเสียเซลล์สูง

สำหรับถังรั่วชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ [4] ซึ่งจะคล้ายกับลักษณะของถังรั่วแบบใช้ถังเก็บโทเค็นร่วมแต่มีการแบ่งลำดับความสำคัญของแต่ละแหล่งกำเนิดตามชนิดของ

ทราฟฟิกออกเป็น 2 ชนิดดังนี้ แหล่งกำเนิดที่มีลำดับความสำคัญสูง และแหล่งกำเนิดที่มีลำดับความสำคัญต่ำ ซึ่งขึ้นอยู่กับระดับความต้องการการสูญหายของเซลล์และการประวิงเวลาที่แตกต่างกัน แต่จากการที่ตั้งให้ลำดับความสำคัญกับแหล่งกำเนิดที่มีลำดับความสำคัญสูงมากกว่า แหล่งกำเนิดที่มีลำดับความสำคัญต่ำ ทำให้อัตราการส่งออกของแหล่งกำเนิดที่มีลำดับความสำคัญต่ำมีค่าอัตราการสูญหายของเซลล์สูง จึงได้มีการนำหลักของ QLT (Queue Length Threshold) Scheduling มาใช้ในถังรับชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ [5] เพื่อเพิ่มประสิทธิภาพให้กับแหล่งกำเนิดที่มีลำดับความสำคัญต่ำ

วิธี QLT Scheduling ในถังรับชนิดที่มีการจัดลำดับความสำคัญนี้ได้แบ่งแหล่งกำเนิดออกเป็นทราฟฟิกแบบเวลาจริง (Real-Time Traffic) และทราฟฟิกแบบไม่เป็นเวลาจริง (Non Real-Time Traffic) โดยใช้ QLT Scheduling ในการแบ่งลำดับความสำคัญ และมีจุดเริ่มเปลี่ยนตั้งอยู่ที่บัฟเฟอร์ของทราฟฟิกแบบไม่เป็นเวลาจริง ซึ่งวิธีนี้จะดีกว่าวิธีที่ให้ลำดับความสำคัญกับแหล่งกำเนิดแหล่งใดแหล่งหนึ่งโดยเฉพาะ ทำให้สามารถจัดสรรความต้องการในการสูญหายของเซลล์และการประวิงให้กับแต่ละทราฟฟิกได้อย่างเหมาะสมตามระดับของจุดเริ่มเปลี่ยน

สังเกตได้ว่าผลของการตั้งค่า QLT ทำให้เซลล์ของทราฟฟิกแบบไม่เป็นเวลาจริงมีผลดีที่ค่า QLT ต่ำๆ ในขณะที่เซลล์ทราฟฟิกแบบเป็นเวลาจริงจะดีขึ้น โดยมีอัตราการสูญหายของเซลล์และการประวิงเวลาที่ค่า QLT สูงๆ ทำให้เป็นเหตุจูงใจในการที่จะปรับปรุงวิธีการ QLT Scheduling ในถังรับชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการที่จะลดอัตราส่วนการสูญหายและเวลาการคอยโดยเฉลี่ยของเซลล์ และทำการจัดให้เป็นไปตามความต้องการในแต่ละประเภทของทราฟฟิกสำหรับถังรับที่มีการจัดลำดับความสำคัญ เช่น ทราฟฟิกชนิดเป็นเวลาจริงมีความต้องการเวลาการคอยโดยเฉลี่ยต่ำ ในขณะที่ยอมให้มีอัตราส่วนการสูญหายของเซลล์ได้ในระดับหนึ่ง ส่วนทราฟฟิกแบบไม่เป็นเวลาจริงนั้นจะยอมให้มีเวลาการคอยโดยเฉลี่ยได้แต่ต้องการอัตราส่วนการสูญหายของเซลล์ต่ำ ซึ่งวิธีการตั้งค่า QLT ที่คงที่ทำให้ได้ค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์เหมาะกับทราฟฟิกแบบใดแบบหนึ่งเท่านั้น ถ้าตั้งค่า QLT สูงจะทำให้อัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในทราฟฟิกแบบเวลาจริงมีค่าต่ำ ส่วนอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในทราฟฟิกแบบไม่เป็นเวลาจริงมีค่าสูง และถ้าตั้งค่า QLT ต่ำค่าที่ได้ก็จะกลับกัน

เพื่อให้ได้ค่า QLT ที่เหมาะสมจึงเสนอวิธีการปรับเปลี่ยนค่า QLT โดยอาศัยค่าพารามิเตอร์ในระบบที่เกี่ยวข้องของทราฟฟิกแต่ละแบบซึ่งมีการเปลี่ยนแปลงอยู่ตลอดเวลา มาเป็นตัวแปรในการปรับค่า QLT เป็นแบบพลวัต แต่ก็เป็นการยากที่จะกำหนดได้แน่นอนว่าค่าที่นำมาพิจารณานั้น



จะต้องมากน้อยเพียงใดเพื่อที่จะนำมาตัดสินใจในการปรับค่า QLT ปัญหานี้สามารถแก้ไขได้โดยใช้ตัวควบคุมแบบฟัซซี (Fuzzy Logic Control, FLC) มาช่วยในการพิจารณาปรับเปลี่ยนค่า QLT แบบพลวัต

## 1.2 วัตถุประสงค์

1. เพื่อจัดระดับอัตราส่วนการสูญหายและเวลาการคอยโดยเฉลี่ยของเซลล์ให้เหมาะสม ทราฟฟิกแต่ละแบบซึ่งมีความต้องการแตกต่างกัน ได้แก่ ทราฟฟิกแบบเวลาจริง และทราฟฟิกแบบไม่เป็นเวลาจริง
2. เพื่อเพิ่มประสิทธิภาพการทำงานให้กับถึงรั้วที่มีการจัดลำดับความสำคัญ ด้วยการปรับระดับจุดเริ่มเปลี่ยนตามความยาวคิว (Queue Length Threshold) แบบพลวัตโดยใช้หลักการฟัซซี

## 1.3 ขอบเขตและเป้าหมายของวิทยานิพนธ์

1. ศึกษาการทำงานของ UPC ในโครงข่ายเอทีเอ็ม
2. ศึกษาการทำงานของฟัซซีและกลไกถึงรั้ว (Leaky Bucket) ในโครงข่ายเอทีเอ็ม
3. ทำการจำลองการทดสอบแบบจำลองโดยใช้กลไกการตรวจวัดของถึงรั้วแบบต่างๆ ในโครงข่ายเอทีเอ็ม เช่น กลไกถึงรั้วแบบใช้โทเค็น กลไกถึงรั้วชนิดที่มีการจัดลำดับความสำคัญแบบใช้ QLT เป็นต้น
4. ทำการจำลองการทำงานของกลไกถึงรั้วโดยใช้ฟัซซีในโครงข่ายและเขียนโปรแกรมเพื่อสร้างแบบจำลองการทดสอบ
5. เปรียบเทียบอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ใน ทราฟฟิกแบบเวลาจริงและแบบไม่เป็นเวลาจริงใน โครงข่ายเอทีเอ็ม ระหว่างกลไกถึงรั้วชนิดที่มีการจัดลำดับความสำคัญแบบที่ใช้ QLT กับวิธีที่เสนอ
6. วิจัยและสรุปผลจากอัลกอริทึมที่นำเสนอ

## 1.4 ขั้นตอนและวิธีการดำเนินการ

1. ศึกษามาตรฐานและทฤษฎีต่างๆ ของโครงข่ายเอทีเอ็มและส่วนที่เกี่ยวข้อง
2. ศึกษาองค์ประกอบต่างๆ ของโครงข่ายเอทีเอ็ม
3. ศึกษาวิธีการตรวจวัดของถึงรั้วในโครงข่ายเอทีเอ็มแบบต่างๆ
4. หาวิธีการปรับปรุงการตรวจวัดของกลไกถึงรั้วชนิดที่มีการจัดลำดับความสำคัญ

แบบ QLT

5. เขียนโปรแกรมสร้างอัลกอริทึม QLT และวิธีที่ปรับปรุงขึ้นมาใหม่
6. เขียนโปรแกรมสร้างแบบจำลองขึ้นมา เพื่อนำมาทดสอบกับอัลกอริทึมที่ปรับปรุงขึ้นมา
7. วิเคราะห์และเปรียบเทียบผลการทดสอบจากแบบจำลอง
8. ตรวจสอบและแก้ไขข้อผิดพลาด
9. สรุปผลงานที่ได้รับ
10. เขียนและพิมพ์วิทยานิพนธ์

### 1.5 ประโยชน์ที่ได้รับ

1. ทำให้ทราบถึงพื้นฐานการทำงานของโครงข่ายเอทีเอ็ม
2. ทำให้ทราบถึงวัตถุประสงค์และการทำงานของตัวควบคุมกราฟฟิกในโครงข่ายเอทีเอ็ม ด้วยตัวควบคุมพารามิเตอร์การใช้ (Usage Parameter Control หรือ UPC)
3. ทำให้ทราบถึงการทำงานในการตรวจวัดและการควบคุมเซลล์ที่ส่งเข้ามาในโครงข่ายเอทีเอ็ม ด้วยกลไกถังรับแบบต่างๆ
4. สามารถควบคุมอัตราส่วนการสูญหายและเวลาการคอยโดยเฉลี่ยของเซลล์ให้เหมาะสมกับลักษณะของแต่ละกราฟฟิกที่เข้ามาในโครงข่ายเอทีเอ็ม ด้วยกลไกถังรับชนิดที่มีการจัดลำดับความสำคัญแบบปรับค่า QLT แบบพลวัตโดยใช้พีซี
5. โปรแกรมจำลองที่พัฒนาขึ้นมา นั้น สามารถนำมาจำลองการทำงานของอัลกอริทึมถังรับในโครงข่าย ATM ได้เป็นอย่างดี สามารถใช้เป็นแนวทางในการวิจัยเพิ่มเติมต่อไป

## บทที่ 2

### โครงข่ายเอทีเอ็ม

โครงข่ายเอทีเอ็ม เป็นโครงข่ายสื่อสารความเร็วสูงที่ออกแบบมาเพื่อให้สามารถรองรับการส่งทราฟฟิกได้หลายประเภท โดยอาศัยการแบ่งข้อมูลของผู้ใช้ออกเป็นหน่วยขนาดเท่าๆกันเรียกว่า เซลล์ ภายใน 1 เซลล์มีขนาด 53 ไบต์ประกอบไปด้วยส่วนหัว (Header) 5 ไบต์ และส่วนของสัมภาระบรรทุก (Payload) 48 ไบต์ ข้อดีของการทำงานแบบนี้คือ ข้อมูลมีขนาดเล็กทำให้เกิดการประวิงเวลาน้อยและมีประสิทธิภาพสูงในการสวิตช์ข้อมูลที่มีขนาดคงที่ แม้ว่าโครงข่ายเอทีเอ็มจะสามารถส่งข้อมูลได้หลายประเภทพร้อมๆกันแต่ข้อมูลในแต่ละชนิดก็มีคุณสมบัติและความต้องการที่แตกต่างกัน ดังนั้นจึงมีการแบ่งระดับการให้บริการในโครงข่ายเอทีเอ็ม

#### 2.1 ระดับการบริการและชนิดของทราฟฟิกในโครงข่ายเอทีเอ็ม

ATM Forum Traffic Management Version 4.1 [6] ได้แบ่งระดับการบริการของโครงข่ายเอทีเอ็ม ตามประเภทของแหล่งกำเนิดข้อมูล เพื่อให้สามารถรองรับการใช้งานต่างๆได้ ระดับการบริการของโครงข่ายเอทีเอ็มเป็นดังนี้

1. *CBR (Constant Bit Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่ต้องการอัตราการส่งข้อมูลคงที่ที่อัตราค่ายอด (Peak Cell Rate, PCR) โดยไม่ต้องการให้เกิดการสูญหายของข้อมูลและเกิดการประวิงขึ้นในระหว่างการส่ง เช่น การส่งสัญญาณภาพ การส่งสัญญาณเสียง ระบบโทรศัพท์ เป็นต้น

2. *rt-VBR (Real-Time Variable Bit Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่มีอัตราการส่งข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ และต้องการใช้งานในการส่งแบบเป็นเวลาจริง เซลล์ข้อมูลที่ส่งแบบนี้จะได้รับการรับประกันการสูญหายของเซลล์ข้อมูล และมีการประวิงอยู่ในระดับต่ำตามที่ต้องการ การต่อประเภทนี้ได้รับการควบคุมการส่งโดยมีพารามิเตอร์คุณภาพของการบริการ ได้แก่ Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) และ Maximum Burst Size (MBS)

3. *nrt-VBR (Non-Real-Time Variable Bit Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่มีอัตราการส่งข้อมูลที่สามารถเปลี่ยนแปลงตามเวลาได้ และต้องการใช้งานในการส่งแบบไม่เป็นเวลาจริง เซลล์ข้อมูลที่ส่งแบบนี้จะได้รับการรับประกันการสูญหายของเซลล์ข้อมูล แต่จะไม่ได้รับประกันการประวิงที่เกิดขึ้น การต่อประเภทนี้ได้รับการควบคุมการส่งโดยมีการกำหนดพารามิเตอร์คุณภาพของการบริการไว้คือ PCR, SCR และ MBS

4. *UBR (Unspecified Bit Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องส่งข้อมูลที่อัตราข้อมูลที่แน่นอน และการส่งผ่านข้อมูลก็อาจเกิดการสูญหายของเซลล์ข้อมูลได้ โดยใช้แบนด์วิดท์ของโครงข่ายที่เหลือจากการบริการทั้ง 3 ประเภทข้างต้น เป็นการบริการที่ไม่ได้มีการกำหนดพารามิเตอร์คุณภาพของการบริการไว้

5. *ABR (Available Bit Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องส่งข้อมูลที่อัตราข้อมูลที่แน่นอน และไม่ต้องทำให้เกิดการสูญหายของเซลล์ข้อมูล แต่ยอมให้มีการประวิงในการส่งได้ มีการควบคุมความคับคั่งของทราฟฟิกโดยอาศัยการป้อนกลับของข้อมูล ทำให้สามารถปรับเปลี่ยนอัตราการส่งข้อมูลได้ตามข่าวสารที่ได้รับให้เหมาะสมกับแบนด์วิดท์ที่เหลือและสถานะของโครงข่ายในปัจจุบัน ทำให้สามารถใช้แบนด์วิดท์ของโครงข่ายที่เหลือจากการบริการแบบอื่นๆ ได้อย่างมีประสิทธิภาพ พารามิเตอร์ที่สำคัญคือ PCR และ MCR (Minimum Cell Rate)

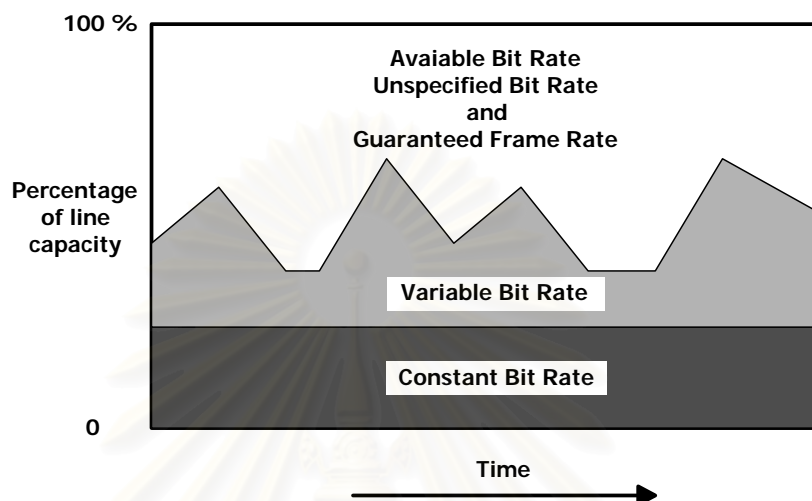
6. *GFR (Guaranteed Frame Rate)* ใช้รองรับแหล่งกำเนิดข้อมูลที่ไม่จำเป็นต้องส่งข้อมูลที่อัตราข้อมูลที่แน่นอน และยอมให้มีการประวิงในการส่งได้ ใช้แบนด์วิดท์ที่เหลือจาก CBR, rt-VBR และ nrt-VBR เช่นเดียวกับการบริการแบบ ABR และ UBR สามารถประกันอัตราข้อมูลต่ำที่สุด โดยการบริการ GFR กำหนดพารามิเตอร์คุณภาพของการให้บริการไว้ ได้แก่ PCR, Minimum Cell Rate (MCR), MBS, Maximum Frame Size (MFS) และ Cell Delay Variation Tolerance (CDVT)

จากการแบ่งระดับการบริการในโครงข่ายเอทีเอ็มข้างต้น รูปที่ 2.1 แสดงลักษณะการจัดสรรทรัพยากรของโครงข่ายในระดับการบริการต่างๆ เนื่องจากคุณภาพการบริการที่แตกต่างกัน โครงข่ายเอทีเอ็มจึงได้จัดลำดับความสำคัญของการบริการทั้ง 6 ประเภทไว้แตกต่างกัน จะพบว่าการบริการ CBR มีลำดับความสำคัญสูงสุด และสามารถใช้งานแบนด์วิดท์ของโครงข่ายได้เป็นลำดับแรก การบริการ VBR ทั้ง rt-VBR และ nrt-VBR นั้น จะมีลำดับความสำคัญเป็นลำดับที่สอง และการบริการที่มีลำดับความสำคัญที่ต่ำที่สุด คือ การบริการ ABR, GFR และ UBR ซึ่งจะใช้แบนด์วิดท์ได้เฉพาะในส่วนแบนด์วิดท์ที่เหลือจากการบริการประเภท CBR และ VBR เท่านั้น

พิจารณาระดับการบริการโดยแบ่งประเภทการบริการออกเป็น แบบเวลาจริง (Real Time) และแบบไม่เป็นเวลาจริง (Non-Real Time) เป็นดังนี้

- ทราฟฟิกแบบเวลาจริง (Real Time Traffic) ได้แก่ CBR และ rt-VBR จะมีค่าพารามิเตอร์ของทราฟฟิกเป็น ค่า Peak Cell Rate (PCR) หรือทั้งค่า Peak Cell Rate (PCR) และ Sustainable Cell Rate (SCR) ตัวอย่างของทราฟฟิกแบบนี้ได้แก่ สัญญาณเสียง, สัญญาณวิดีโอ เป็นต้น

- ทราฟฟิกแบบไม่เป็นเวลาจริง (Non-Real Time Traffic) มี 4 ประเภทได้แก่ nrt-VBR, UBR, ABR และ GFR ค่าพารามิเตอร์ของทราฟฟิกแต่ละประเภทแสดงในตารางที่ 2.1 ตัวอย่างของทราฟฟิกแบบนี้ได้แก่ สัญญาณข้อมูล, สัญญาณภาพนิ่ง



รูปที่ 2.1 ลักษณะการจัดสรรทรัพยากรของโครงข่ายเอทีเอ็ม

## 2.2 คุณภาพการบริการ (Quality of Service) ของโครงข่ายเอทีเอ็ม

เมื่อเปรียบเทียบระดับการให้บริการในแต่ละประเภทเข้ากับพารามิเตอร์ของคุณภาพการบริการ (QoS Parameters) และพารามิเตอร์ของทราฟฟิก (Traffic Parameters) ที่เกี่ยวข้องกับระดับการบริการ จะเป็นดังตารางที่ 2.1

พารามิเตอร์ของคุณภาพการบริการ (QoS Parameters) แสดงได้ดังนี้

1. Peak-to-peak Cell Delay Variation (Peak-to-peak CDV) หมายถึง ค่าความแตกต่างระหว่างเวลาประวิง ของเซลล์ในช่วงที่ยาวที่สุด และเวลาประวิงของเซลล์ในช่วงที่สั้นที่สุด
2. Maximum Cell Transfer Delay (Max CTD) หมายถึง ค่าเวลาประวิงของการรับส่งเซลล์ที่มีช่วงยาวที่สุด
3. Cell Loss Ratio (CLR) หมายถึง ค่าอัตราส่วนการสูญหายของเซลล์

ตารางที่ 2.1 คุณสมบัติของการให้บริการในแต่ละชนิดในโครงข่ายเอทีเอ็ม

คุณสมบัติ	ประเภทของการให้บริการ					
	CBR	rt-VBR	nrt-VBR	UBR	ABR	GFR
<b>พารามิเตอร์ของทราฟฟิก:</b>						
PCR และ CDVT <sub>4,5</sub>	กำหนด			กำหนด <sub>2</sub>	กำหนด <sub>3</sub>	กำหนด
SCR, MBS และ CDVT <sub>4,5</sub>	ไม่กำหนด	กำหนด		ไม่กำหนด		
MCR <sub>4</sub>	ไม่กำหนด				กำหนด	ไม่กำหนด
MCR, MBS, MFS และ CDVT <sub>4,5</sub>	ไม่กำหนด					กำหนด
<b>พารามิเตอร์ของ QoS:</b>						
Peak-to-peak CDV	กำหนด		ไม่กำหนด			
Max CTD	กำหนด		ไม่กำหนด			
CLR <sub>4</sub>	กำหนด			ไม่กำหนด	หมายเหตุ <sub>1</sub>	หมายเหตุ <sub>7</sub>
<b>คุณสมบัติอื่นๆ:</b>						
การป้อนกลับ	ไม่กำหนด				กำหนด <sub>6</sub>	ไม่กำหนด

ที่มา: Table 2-1 ใน [6]

หมายเหตุ:

1. CLR มีค่าต่ำ เมื่อแหล่งกำเนิดสามารถปรับอัตราการส่งได้ ตามการป้อนกลับของโครงข่าย
2. อาจจะไม่มีการใช้ CAC (Connection Admission Control) และ UPC (Usage Parameter Control) มาควบคุม
3. แสดงอัตราการส่งที่สูงที่สุด สำหรับอัตราการส่งจริงจะเป็นไปตามการป้อนกลับของโครงข่าย
4. พารามิเตอร์เหล่านี้ใช้แสดงการใช้งาน PVCs (Permanent Virtual Channels) หรือ SVCs (Switched Virtual Channels)
5. CDVT หมายถึง Cell Delay Variation Tolerance ไม่มีค่าเฉพาะเจาะจง
6. ดูรายละเอียดในการจัดการทราฟฟิกสำหรับการรับส่งข้อมูลแบบ ABR
7. CLR มีค่าต่ำ เมื่อเฟรมเหมาะสมในการรับรองการบริการ

นอกจากพารามิเตอร์ของคุณภาพการบริการแล้ว ยังมีพารามิเตอร์ของทราฟฟิก (Traffic Parameters) ที่เกี่ยวข้องกับการบริการ ได้แก่

1. PCR (Peak Cell Rate) หมายถึง อัตราการส่งข้อมูลที่สูงที่สุดของแหล่งกำเนิด
2. MCR (Minimum Cell Rate) หมายถึง อัตราการส่งข้อมูลที่ต่ำที่สุดของแหล่งกำเนิด
3. SCR (Sustainable Cell Rate) หมายถึง อัตราการส่งข้อมูลเฉลี่ยตลอดช่วงเวลาของการส่งข้อมูล
4. CDVT (Cell Delay Variation Tolerance) หมายถึง ระดับความทนทานต่อความแตกต่างของเวลาประวิงของเซลล์
5. MBS (Maximum Burst Size) หมายถึง จำนวนของเซลล์ที่มีจำนวนมากที่สุด ในช่วงเวลาหนึ่งที่มีเซลล์ส่งเข้าสู่โครงข่ายอย่างต่อเนื่อง

พารามิเตอร์ของคุณภาพการบริการและพารามิเตอร์ของทราฟฟิกนั้น จะเป็นพารามิเตอร์สำคัญที่แหล่งกำเนิดได้ตกลงไว้กับโครงข่าย ซึ่งโครงข่ายที่ดีจะต้องสามารถรับประกันคุณภาพของการบริการให้เป็นไปตามพารามิเตอร์ที่ได้ตกลงกันไว้นี้ และในทางตรงข้ามถ้าหากแหล่งกำเนิดละเมิดหรือไม่ปฏิบัติตามข้อตกลงดังกล่าว จะทำให้ข้อมูลอาจเกิดการสูญหาย หรืออาจมีการยกเลิกการต่อของแหล่งกำเนิดนั้นๆ ได้

### 2.3 การจัดการทราฟฟิก (Traffic Management) ในโครงข่ายเอทีเอ็ม

ในการทำงานของโครงข่ายเอทีเอ็มนั้นมีความยุ่งยากและซับซ้อนเนื่องจากต้องรองรับข้อมูลหลายชนิดอีกทั้งแต่ละชนิดยังมีความต้องการที่แตกต่างกัน ดังนั้นโครงข่ายเอทีเอ็มจึงต้องมีการจัดการกับทราฟฟิกที่เข้ามาโดยมีส่วนของการจัดการทราฟฟิก (Traffic Management) ในการทำงานที่จะจัดสรรทรัพยากรที่มีอยู่ให้ได้ตามความต้องการและเกิดประสิทธิภาพสูงที่สุด รวมทั้งต้องป้องกันการเกิดความคับคั่งขึ้นในโครงข่าย ซึ่งการจัดการดังกล่าวจะมีการจัดการทราฟฟิกอยู่หลายชนิดด้วยกัน ขึ้นกับลักษณะและหน้าที่การทำงานที่แตกต่างกัน ได้แก่ Connection Admission Control (CAC), Usage Parameter Control (UPC), Selective Cell Discard, Traffic Shaping, Explicit Forward Congestion Indication (EFCI), Generic Flow Control, ABR Flow Control การจัดการทราฟฟิกต่างๆสามารถแบ่งตามลักษณะการทำงานได้เป็น 2 แบบ คือ แบบวงจรเปิด และแบบวงจรปิด

### แบบวงจรเปิด (Open Loop)

เป็นการทำงานที่จะคอยควบคุมไม่ให้เกิดปัญหาขึ้นมา โดยทำการพิจารณาตัดสินใจในการที่จะรับหรือไม่รับข้อมูลใหม่ที่เข้ามาในระบบ อาศัยหลักเกณฑ์ในการพิจารณาโดยไม่มีการนำสถานะของระบบในขณะที่ทำงานมาร่วมพิจารณา ได้แก่การทำงานของ CAC, UPC, Selective Cell Discard และ Traffic Shaping

### แบบวงจรปิด (Close Loop)

เป็นการทำงานในลักษณะคอยตรวจสอบสถานะของระบบขณะทำงานและนำข้อมูลที่ได้มาเป็นข้อมูลสำคัญใช้ในการแก้ปัญหาโดยส่งข่าวกลับไปยังส่วนที่รับผิดชอบเกี่ยวกับปัญหา และทำการปรับการทำงานเพื่อแก้ไขปัญหที่เกิดขึ้นได้แก่ การทำงานของ EFCI, Generic Flow Control และ ABR Flow Control

ในหัวข้อต่อไปจะกล่าวถึงการทำงานในแบบวงจรเปิดของ UPC ซึ่งมีข้อดีคือสามารถทำงานได้เร็วและเป็นการป้องกันปัญหาที่จะเกิดขึ้น ในขณะที่การทำงานในแบบวงจรปิดเกิดผลเสียคือ ต้องส่งข้อมูลไปกลับทำให้ใช้เวลานานในการตอบสนอง จึงทำให้แก้ปัญหาได้ช้าหรือไม่ทันการณ์ และยังเพิ่มปริมาณข้อมูลที่ส่งในโครงข่ายอีกด้วย

## 2.4 Usage Parameter Control (UPC)

### 2.4.1 หน้าที่การทำงานของ UPC

UPC คือกลุ่มการทำงานของโครงข่ายเพื่อเฝ้ามองดูกราฟฟิกที่เข้ามาในโครงข่าย และบังคับให้กราฟฟิกเหล่านี้ปฏิบัติตามข้อตกลง โดยเน้นการป้องกันทรัพยากรของโครงข่ายจากผู้ใช้ที่กระทำผิดด้วยการละเมิดข้อตกลง ซึ่งการละเมิดข้อตกลงนี้มีผลกระทบกับ QoS ของการสร้างการต่ออื่นๆ ต่อไป วิธีการป้องกันนี้ใช้การตรวจสอบการละเมิดจากพารามิเตอร์ที่ได้ตกลงไว้ และตัดสินใจกระทำกับข้อมูลหรือเซลล์ด้วยวิธีใดวิธีหนึ่ง

จากหน้าที่และการทำงานทำให้ตำแหน่งของ UPC ควรจะอยู่ในส่วนที่ต่อจากแหล่งกำเนิดกราฟฟิก ก่อนที่จะเข้าไปในโครงข่ายหรืออาจเรียกว่าอยู่ในส่วนของ UNI (User to Network Interface) เพราะจะต้องทำหน้าที่ป้องกันและจัดการกับเซลล์ให้ได้ก่อนที่จะเข้าไปในโครงข่าย การเฝ้ามองดูการต่อร่วมของผู้ใช้ที่ตำแหน่ง UNI ถือเป็นการทำงานของ UPC และการเฝ้ามองดูการต่อร่วมของผู้ใช้ที่ตำแหน่ง NNI (Network to Network Interface) ถือเป็นการทำงานของ NPC (Network Parameter Control) แต่โดยทั่วไปแล้ว UPC มีการใช้งานมากกว่า NPC การเฝ้ามองดูการต่อซึ่งจะรวมทุกการต่อเข้าด้วยกันแล้วต่อไปยัง UPC โดย UPC จะพิจารณาทั้งการต่อของผู้ใช้และช่องสัญญาณซิกแนลลิง (Signaling channel)



การทำงานของ UPC อาจเรียกว่า Policing Function มีวัตถุประสงค์การทำงานดังต่อไปนี้

- ป้องกันไม่ให้เกิดการส่งข้อมูลเกินที่กำหนดไว้
- คอยจัดการกับข้อมูลที่ส่งมาเกินที่ตกลงไว้
- ป้องกันการเกิดความคับคั่งของโครงข่าย

การทำงานของ UPC บน VCC (Virtual Channel Connection) และ VPC (Virtual Path Connection) มีหลักการดังนี้

1. ตรวจสอบความถูกต้องของ VPI (Virtual Path Identifier) และ VCI (Virtual Channel Identifier) (กรณีใช้ VCI และ VPI ร่วมกันบน VCC) และเฟรมองคูลทราฟฟิกที่เข้ามาในโครงข่ายจากแต่ละ VCC เพื่อให้แน่ใจว่าแหล่งกำเนิดทราฟฟิกปฏิบัติตามข้อตกลงหรือละเมิดข้อตกลง

2. ตรวจสอบความถูกต้องของ VPI (กรณีใช้ส่งเฉพาะ VPI บน VPC) และเฟรมองคูลทราฟฟิกที่เข้ามาในโครงข่ายจากแต่ละ VPC เพื่อให้แน่ใจว่าทราฟฟิกไม่ได้ละเมิดข้อตกลง

#### 2.4.2 ความต้องการของ UPC

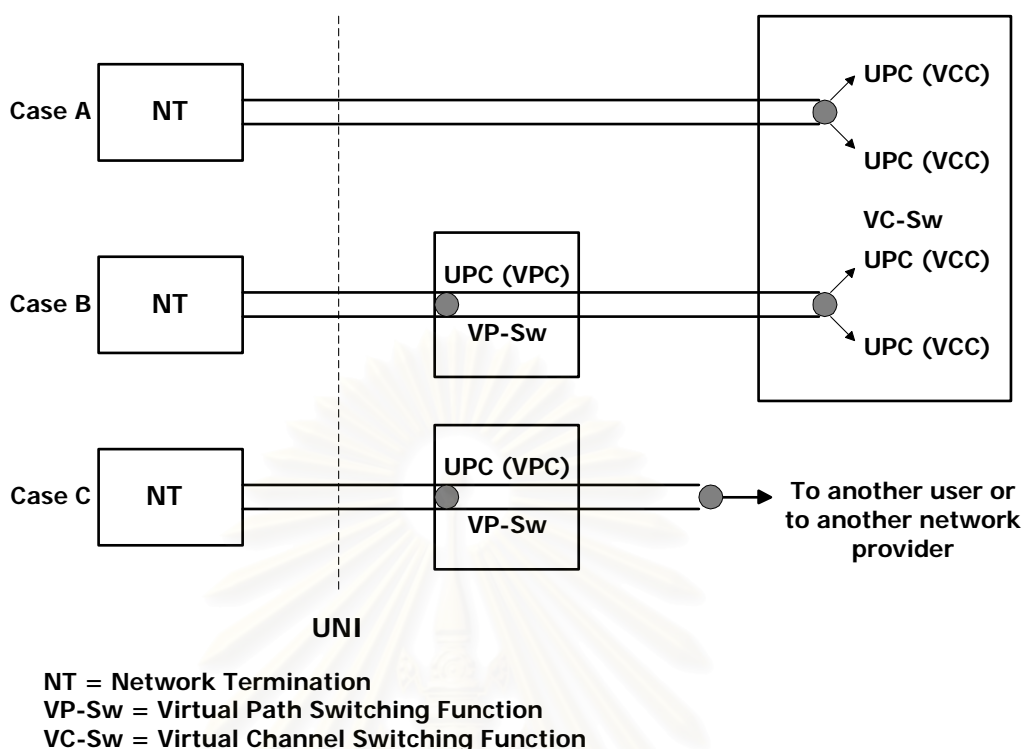
ข้อตกลงเบื้องต้นที่โครงข่ายกระทำไว้กับผู้ใช้ในตอนเริ่มต้นร้องขอการต่อ จะกลายมาเป็นตัวกำหนดหน้าที่การทำงานของ UPC ทำให้โครงข่ายสามารถรองรับคุณภาพการบริการ (QoS) ให้กับผู้ใช้ โดยไม่ทำให้เกิดความคับคั่งและลดสมรรถภาพของโครงข่าย

โดย UPC ที่ดีจะต้องมีลักษณะดังนี้

- ไม่ขัดขวางหรือรบกวนการส่งข้อมูลที่เป็นไปตามที่ตกลงไว้
- สามารถจัดการกับการส่งข้อมูลที่เกินข้อตกลงไว้ได้อย่างรวดเร็ว
- มีการทำงานที่ไม่ซับซ้อน
- ราคาถูก และการดูแลรักษาง่าย

#### 2.4.3 ตำแหน่งที่ตั้งของ UPC

ตำแหน่งที่ตั้งของ UPC จะอยู่ระหว่างสายส่งของ Virtual Path หรือ Virtual Channel กับ Network Termination แสดงได้ดังรูปที่ 2.2 [6] ในกรณีแรก (Case A) UPC จะตั้งอยู่ใน VC-Sw (Virtual Channel-Switching Function) เซลล์จะเข้ามายัง UPC และ UPC จะทำงานก่อนการทำงานของ VC-Sw ถ้ามีการต่อระหว่าง VP-Sw (Virtual Path-Switching Function) กับ VC-Sw ดังกรณีที่สอง (Case B) UPC จะทำงานอยู่ภายใน VP-Sw และ VC-Sw ตามลำดับ และทำงานก่อนการทำงานของสวิตช์ทั้งสอง ถ้ามีการต่อไปยังโครงข่ายหรือผู้ใช้อื่นๆ ผ่าน VP-Sw (Case C) UPC จะทำงานอยู่ภายใน VP-Sw ก่อนที่จะเริ่มการทำงานของสวิตช์ ถ้ามีการต่อเข้าไปยัง VC-Sw ของโครงข่ายถัดไป ก็จะมี UPC ทำงานอยู่ใน VC-Sw นั้น



รูปที่ 2.2 ตำแหน่งที่ตั้งของ UPC ในโครงข่าย ATM [6]

#### 2.4.4 การทำงานของ UPC

หลังจากที่ UPC ทำการตรวจสอบข้อมูลหรือเซลล์ที่เข้ามาเทียบกับข้อตกลงที่กำหนดไว้ในตอนแรกแล้วจะมีการทำงานดังนี้

1. ทำการปล่อยเซลล์ (Cell Passing) ให้ผ่านเข้าไปในโครงข่าย โดยเสมือนว่าเซลล์นี้ไม่ได้ละเมิดข้อตกลงที่กำหนดไว้
2. ทำการแท็กเซลล์ (Cell Tagging) เซลล์ที่จะถูกแท็กจะเป็นเซลล์ที่มีบิต CLP (Cell Loss Priority) เป็น 0 เท่านั้นโดยทำการเปลี่ยนบิต CLP จาก 0 เป็น 1 เนื่องจากเซลล์เหล่านี้ละเมิดข้อตกลงของการแท็กเซลล์ โดยที่บิต CLP ทำหน้าที่เป็นตัวบอกลำดับความสำคัญของเซลล์ บิต CLP จะเป็น 0 ในเซลล์ที่มีลำดับความสำคัญสูง CLP มีค่าเป็น 1 ในเซลล์ที่มีลำดับความสำคัญต่ำ เมื่อโครงข่ายเกิดความคับคั่งจะทำการทิ้งเซลล์ที่มีลำดับความสำคัญต่ำก่อน (CLP บิตมีค่าเป็น 1) ส่วนเซลล์ที่มีลำดับความสำคัญสูงจะถูกทิ้งก็ต่อเมื่อโครงข่ายไม่สามารถทำการรองรับข้อมูลได้อีก
3. ทำการทิ้งเซลล์ (Cell Discarding) UPC จะทำการทิ้งเซลล์ที่เข้ามา เมื่อมีการละเมิดข้อตกลงที่กำหนดไว้ และโครงข่ายไม่สามารถรองรับข้อมูลได้อีก

UPC ที่ใช้มีอยู่ด้วยกันหลายชนิด แตกต่างกันไปตามลักษณะการทำงานและอัลกอริทึม (Algorithm) ที่ใช้ได้แก่ Moving Window, Fixed Windows, Exponentially Weighted Moving Average และ ถังรั่ว (Leaky Bucket) จาก [1, 2] แสดงให้เห็นว่าอัลกอริทึมถังรั่วมีประสิทธิภาพสูงในการจัดการกับทราฟฟิกได้รวดเร็ว และจัดการกับข้อมูลที่มีลักษณะเป็นเบิสต์ได้ดีกว่าวิธีอื่นๆ ซึ่งจะกล่าวถึงรายละเอียดของอัลกอริทึมถังรั่ว แบบต่างๆในบทที่ 3 ต่อไป



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### Leaky Bucket และการทำงานของ Fuzzy Logic

Leaky Bucket จัดเป็น UPC ที่มีประสิทธิภาพมากตัวหนึ่ง มีการทำงานเป็นทั้ง Traffic Shaping และ Traffic Policing กล่าวคือ เมื่อมีการส่งข้อมูลเป็นช่วงๆด้วยความเร็วสูงมากในระยะเวลาสั้นๆที่เรียกว่า เบิร์สต์ (Burst) สลับกับการหยุดนิ่ง จะเป็นเหตุให้เกิดความคับคั่ง ถ้าผู้ใช้สามารถส่งข้อมูลในอัตราข้อมูลสม่ำเสมอ โอกาสที่จะเกิดความคับคั่งของข้อมูลก็จะลดลง การบังคับควบคุมให้ข้อมูลถูกส่งออกมาในอัตราที่ต้องการได้ จะช่วยแก้ปัญหาความคับคั่งของข้อมูลในแบบของวงจรเปิด วิธีการนี้ถูกนำมาใช้อย่างมากในโครงข่ายเอทีเอ็ม เรียกว่า การควบคุมสัญญาณของทราฟฟิกหรือ Traffic Shaping ส่วนของการตรวจทราฟฟิกหรือ Traffic Policing จะเป็นการกระทำหลังจากที่มีการกำหนดรูปแบบการส่งและได้ตกลงยอมรับกันระหว่างโครงข่ายและผู้ใช้แล้ว Traffic Policing ก็จะทำหน้าที่คอยตรวจการส่งข้อมูลให้เป็นไปตามข้อตกลงดังกล่าว ซึ่งตรงตามลักษณะการทำงานของ UPC

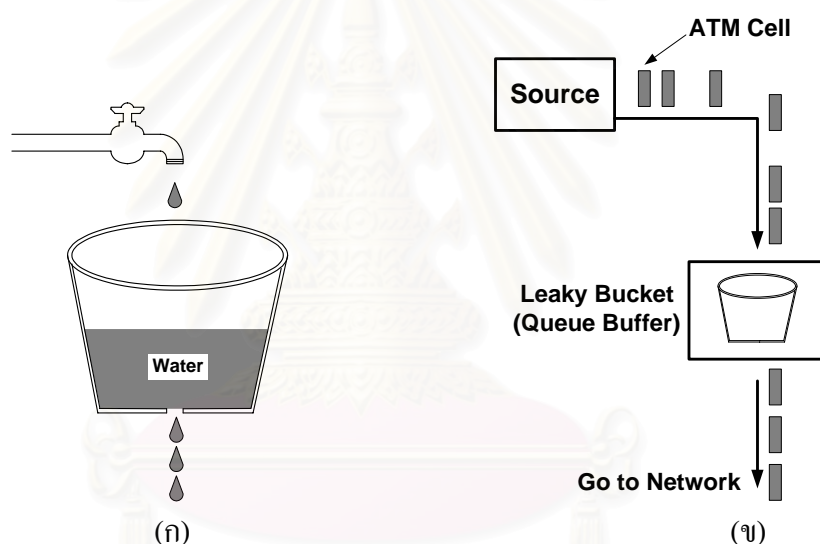
#### 3.1 อัลกอริทึมถังรั่ว (Leaky Bucket Algorithm)

J. S. Tuner เป็นผู้นำเสนอ “อัลกอริทึมถังรั่ว” (Leaky Bucket Algorithm) [7] ในปี 1986 โดยมีความหมายตามชื่อคือ สมมุติว่ามีถังใส่น้ำใบหนึ่งซึ่งมีรูรั่วเล็กๆที่ก้นถังไปรองน้ำที่ก้นถังไหลอยู่ ดังแสดงในรูปที่ 3.1(ก) ไม่ว่าอัตราการไหลเข้าของน้ำจะเป็นเท่าใดก็ตามอัตราการไหลออกของน้ำจะมีค่าคงที่เสมอ ขณะเดียวกันค่าของน้ำในถังมีอยู่สองค่าเสมอ คือ หนึ่งเป็นค่าตัวเลขเมื่อมีน้ำอยู่ในถัง และสองมีค่าเป็นศูนย์เมื่อไม่มีน้ำอยู่ในถังเลย แต่ถ้ามีน้ำเต็มถังและยังคงใส่น้ำลงไปอีก น้ำก็จะไหลล้นออกทางด้านข้างซึ่งถือว่าเป็นส่วนที่สูญหาย อัลกอริทึมถังรั่วมีพารามิเตอร์ที่สำคัญคือขนาดของถัง (Bucket size) และอัตรารั่ว (Leaky rate)

ในการส่งข้อมูลเป็นเซลล์ในโครงข่ายเอทีเอ็ม มีการทำงานของ Leaky Bucket โดยมีลักษณะเป็นการส่งข้อมูลจากแหล่งกำเนิดผ่านเข้าตัว Leaky Bucket ซึ่งประกอบด้วยคิวบัฟเฟอร์ขนาดคงที่ค่าหนึ่ง เปรียบเสมือนกับขนาดของถังน้ำ เมื่อมีเซลล์ผ่านมาถึงก็จะเข้าแถวคอยอยู่ในคิวบัฟเฟอร์ตามลำดับ บัฟเฟอร์ก็จะทำการปล่อยเซลล์ออกไปยังโครงข่ายที่อัตราคงที่ค่าหนึ่งๆ เปรียบเหมือนอัตราการรั่วของถังน้ำ โดยลักษณะของการส่งข้อมูลออกจะเรียงตามลำดับการเข้ามาของเซลล์ในบัฟเฟอร์ โดยใช้หลักการที่เรียกว่าวิธี FIFO (First In First Out) และขณะเดียวกันถ้าบัฟเฟอร์เต็ม เซลล์ที่เข้ามาที่หลังก็จะล้นออกไป เปรียบเสมือนว่าขณะนั้นอัตราการส่งมีค่าสูงเกินกว่าที่ตกลง จำเป็นต้องทำการทิ้งเซลล์บางส่วนออกไปเพื่อป้องกันไม่ให้เกิดความคับคั่งในโครงข่าย

หลักการการทำงานของ Leaky Bucket นี้ถือว่าเป็นกระบวนการที่มีระบบแถวคอยของผู้ให้บริการเดี่ยว ที่มีเวลาให้บริการคงที่ (Single Server Queuing System with Constant Service Time) นั่นเอง

การทำงานของ Leaky Bucket ในโครงข่ายเอทีเอ็ม แสดงดังรูปที่ 3.1(ข) ซึ่งกลไกนี้สามารถทำการปรับปริมาณการไหลของเซลล์เอทีเอ็ม ของผู้ใช้ที่ต้องการส่งข้อมูลผ่านโครงข่าย ที่มีอัตราความเร็วของข้อมูลไม่แน่นอนให้กลายเป็นการไหลของเซลล์เข้าสู่โครงข่ายที่มีอัตราความเร็วสม่ำเสมอ จึงเป็นการลดโอกาสที่จะทำให้เกิดความคับคั่งของข้อมูลลงได้ เรียกได้ว่ามีการทำงานเป็น Traffic Shaping และการจัดการกับเซลล์ที่ล้นทิ้งออกไปเป็นลักษณะการทำงานของ Traffic Policing ในบางกรณีขณะที่มีเซลล์เกินขนาดของถัง ระบบอาจไม่ทิ้งเซลล์นั้นในทันทีหากโครงข่ายยังสามารถรองรับการส่งข้อมูลได้ แต่จะทำการแท็กเซลล์แทนเพื่อลดลำดับความสำคัญของเซลล์ๆ นั้นลง



รูปที่ 3.1 (ก) ถังรั่วที่มีการไหลเข้าออกของน้ำ (ข) ถังรั่วสำหรับการส่งเซลล์ในโครงข่ายเอทีเอ็ม

การทำงานของอัลกอริทึมถังรั่วที่กล่าวมาข้างต้นจะเห็นว่าสามารถทำงานได้กับข้อมูลที่มีขนาดเท่าๆกัน แต่ถ้าข้อมูลที่ส่งมีขนาดไม่เท่ากัน อัลกอริทึมถังรั่วก็สามารถที่จะนำมาใช้งานได้โดยอาศัยการเปลี่ยนจากวิธีการนับจำนวนของข้อมูลที่เข้ามาเป็นการนับปริมาณของข้อมูลแทน ซึ่งจะพบได้ในการส่งข้อมูลที่เป็นแพ็กเก็ตที่มีขนาดไม่คงที่

ข้อเสียที่พบในอัลกอริทึมถังรั่วคือ ไม่มีความยืดหยุ่นในค่าพารามิเตอร์ต่างๆ เช่น ขนาดของถัง อัตราการรั่ว ซึ่งมักจะใช้เป็นค่าคงที่ และกรณีที่ข้อมูลมีลักษณะเบิสต์มากๆ ซึ่งเกิดในแหล่งกำเนิดที่มีอัตราการส่งไม่คงที่ พบมากในการทำงานของโครงข่ายเอทีเอ็ม ซึ่งมีช่วงที่ทำการส่ง

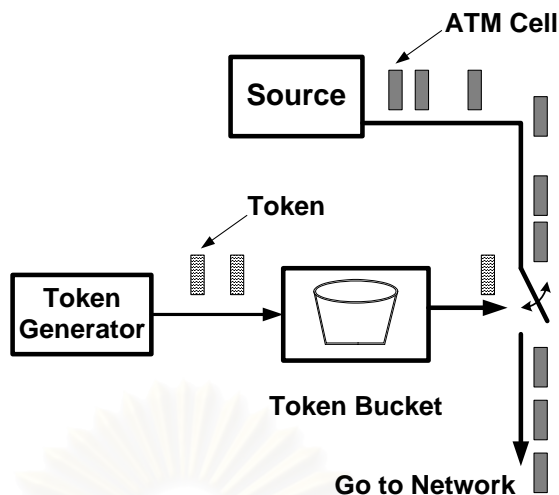
ด้วยความเร็วสูงหรือส่งเซลล์เป็นจำนวนมากกว่าปกติในช่วงระยะเวลาสั้น ทำให้มีความน่าจะเป็นของอัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio) สูง

### 3.2 อัลกอริทึมถังรั่วแบบใช้โทเค็น (Token Leaky Bucket Algorithm)

โดยทั่วไปข้อมูลที่ส่งในโครงข่ายเอทีเอ็ม นั้นจะมีลักษณะเป็นเบิร์สต์ ซึ่งจากการทำงานของอัลกอริทึมถังรั่วไม่อาจรองรับข้อมูลที่มีลักษณะการมาอย่างรวดเร็วในช่วงเวลาสั้นๆ ได้ดีพอ เนื่องมาจากการทำงานของอัลกอริทึมถังรั่วนั้นจะส่งข้อมูลเข้าสู่โครงข่ายที่อัตราคงที่ค่าหนึ่งเสมอ ไม่ว่าข้อมูลจะถูกส่งมาจากแหล่งกำเนิดด้วยความเร็วเท่าใดก็ตาม เพื่อให้เกิดความเหมาะสมกับสภาพของข้อมูลที่มีลักษณะเป็นเบิร์สต์สูงที่ถูกส่งออกมาจากแหล่งกำเนิด อัลกอริทึมถังรั่วจึงถูกนำมาปรับปรุงให้มีความเหมาะสมมากยิ่งขึ้นจนกลายมาเป็นวิธีที่เรียกว่า “อัลกอริทึมถังรั่วแบบใช้โทเค็น” (Token Leaky Bucket Algorithm)

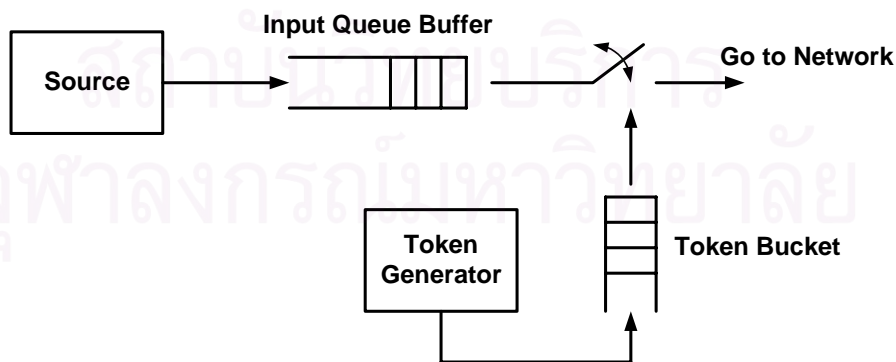
อัลกอริทึมถังรั่วแบบใช้โทเค็น ประกอบไปด้วยถังเก็บโทเค็น (Token Bucket) หรืออาจเรียกว่า บ่อเก็บโทเค็น (Token Pool) ที่มีขนาดคงที่ค่าหนึ่ง คอยทำหน้าที่เก็บรักษาโทเค็นที่ถูกส่งออกมาจากแหล่งกำเนิดโทเค็น ที่เรียกว่า Token Generator ด้วยค่าคงที่ค่าหนึ่ง และโทเค็นที่เก็บไว้ก็จะทำหน้าที่เสมือนเป็นตัวให้เครดิตกับข้อมูลหรือเซลล์ในการส่งผ่านเข้าไปในโครงข่าย ซึ่งก็คือเซลล์จะถูกส่งเข้าไปในโครงข่ายได้ก็ต่อเมื่อต้องมีโทเค็นคู่กันเข้าไป 1 เซลล์ต่อ 1 โทเค็น และเมื่อมีการส่งเซลล์ออกไปคู่กันกับโทเค็น จำนวนโทเค็นที่มีอยู่ก็จะลดลงตามจำนวนของเซลล์ ในขณะที่ไม่มีการส่งเซลล์จะมีโทเค็นถูกสร้างออกมาตลอด ถ้าถังเก็บโทเค็นเต็มโทเค็นที่ถูกสร้างออกมาทีหลังก็จะถูกทิ้งไปจนกว่าโทเค็นในถังจะถูกใช้ไปจึงจะสามารถเก็บโทเค็นที่สร้างเข้ามาใหม่ได้ ซึ่งโดยแท้จริงแล้วโทเค็นไม่ได้ถูกนำมาใช้งานอย่างอื่นเลยนอกเหนือไปจากถูกสร้างมาให้เป็นตัวกำหนดการส่งเซลล์และถูกลบทิ้งไปหนึ่งตัวต่อการส่งเซลล์หนึ่งเซลล์เท่านั้น

การทำงานของอัลกอริทึมถังรั่วแบบใช้โทเค็นแสดงได้ดังรูปที่ 3.2 มีพารามิเตอร์ที่สำคัญคือ อัตราการสร้างโทเค็นซึ่งมีผลต่ออัตราการส่งข้อมูล และขนาดของถังเก็บโทเค็นที่มีผลต่อความเร็วที่สูงที่สุดในการส่งข้อมูลเนื่องจากถังขนาดใหญ่สามารถเก็บสำรองโทเค็นได้มากซึ่งจะมีผลทำให้สามารถส่งข้อมูลได้ด้วยความเร็วสูงในทันทีเมื่อมีข้อมูลเข้ามาจำนวนมาก การทำงานของโทเค็นในลักษณะนี้ทำให้อัลกอริทึมนี้สามารถรองรับการส่งข้อมูลที่มีลักษณะเป็นเบิร์สต์สูงได้ดี เนื่องจากในช่วงเวลาที่ไม่มีการส่งข้อมูลตัวสร้างโทเค็นก็จะทำการสร้างโทเค็นออกมาตลอด เพื่อที่จะรองรับการมาของข้อมูล บวกกับการที่มีโทเค็นสะสมในถังเก็บทำให้สามารถรองรับข้อมูลได้ในปริมาณมากและส่งข้อมูลออกได้ในอัตราข้อมูลสูง ซึ่งหมายความว่าสามารถส่งเซลล์ออกได้ติดต่อกันในทันทีทันใดเท่ากับจำนวนสูงสุดของโทเค็นที่มีอยู่ในถังเก็บโทเค็น



รูปที่ 3.2 อัลกอริทึมถังรับแบบใช้โทเค้น

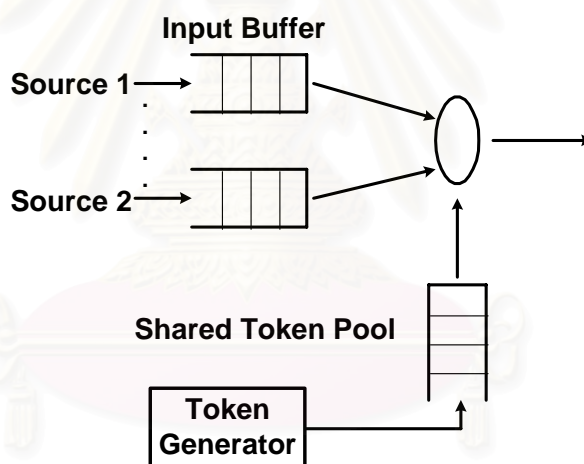
อัลกอริทึมถังรับแบบใช้โทเค้นนั้นในตอนแรกยังไม่มีการกำหนดบัฟเฟอร์ที่ด้านขาเข้าของผู้ใช้ (User Queue) ทำให้เซลล์ที่เข้ามาเมื่อไม่มีโทเค้นก็จะถูกทิ้งหรือแท็กเซลล์ไป ต่อมา I. Cidon และ I. S. Gopal [8] ได้เสนอให้ใช้บัฟเฟอร์ที่ด้านขาเข้าของข้อมูล เพื่อประนีประนอมกันระหว่างค่าความน่าจะเป็นของอัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio) ที่ลดลงในขณะที่เพิ่มเวลาการคอยโดยเฉลี่ยของเซลล์ (Mean Waiting Time) ที่คิวบัฟเฟอร์ดังแสดงในรูปที่ 3.3 ในการออกแบบขนาดบัฟเฟอร์ที่ด้านขาเข้านั้นมีการออกแบบให้มีทั้งขนาดคงที่ค่าหนึ่งและขนาดไม่จำกัด กรณีบัฟเฟอร์แบบที่มีขนาดคงที่เซลล์จะถูกทิ้งไปถ้าขณะที่ยังมีบัฟเฟอร์เต็ม ส่วนในกรณีที่ไม่มี การจำกัดขนาดของบัฟเฟอร์จะไม่มีผลกระทบต่อข้อมูลทิ้ง ถ้าข้อมูลส่งที่ความเร็วสูงจะมีปัญหาในเรื่องเวลาการรอคอยของเซลล์ในบัฟเฟอร์



รูปที่ 3.3 อัลกอริทึมถังรับแบบใช้โทเค้นที่มีบัฟเฟอร์ที่ด้านเข้า

### 3.3 หลักการของอัลกอริทึมถึงรั้วชนิดที่ใช้ถังเก็บโทเค็นร่วม

เนื่องจากปัญหาเวลาการคอยโดยเฉลี่ยของเซลล์ในโครงข่ายที่มีความเร็วสูงนั้นมีความสำคัญมาก M. Jo, S. Kimura และ Y. Ebihara [3] จึงได้มีการเสนอวิธีที่จะลดเวลาการคอยโดยเฉลี่ยของเซลล์และเพิ่มประสิทธิภาพของโครงข่ายด้วยวิธีการใช้ถังเก็บโทเค็นร่วม (Shared Token Pool) ในอัลกอริทึมถึงรั้วแบบใช้โทเค็น แสดงดังรูปที่ 3.4 มีวัตถุประสงค์เพื่อที่จะทำให้แต่ละแหล่งกำเนิดสามารถใช้โทเค็นของแหล่งกำเนิดอื่น ๆ ได้ เนื่องมาจากหลักการใช้ถังเก็บโทเค็นร่วมกัน โดยที่ขนาดของถังเก็บโทเค็นร่วมจะเท่ากับขนาดของถังทั้งสองไปรวมกันและอัตราการสร้างโทเค็นมีขนาดเท่ากับอัตราสร้างโทเค็น ของทั้งคู่รวมกัน ซึ่งผลที่ได้ทำให้เวลาการคอยโดยเฉลี่ยของเซลล์ลดลง และประสิทธิภาพของโครงข่ายสูงขึ้น แต่วิธีนี้มีข้อเสียคือ กรณีที่แหล่งกำเนิดมีอัตราการส่งสูงจะไปแย่งใช้โทเค็นของแหล่งกำเนิดที่ส่งในอัตราที่ต่ำกว่า ทำให้เกิดความไม่ยุติธรรมในการใช้ทรัพยากรของโครงข่าย และทำให้แหล่งกำเนิดที่ส่งด้วยอัตราข้อมูลต่ำมีความน่าจะเป็นของอัตราส่วนการสูญหายของเซลล์สูง



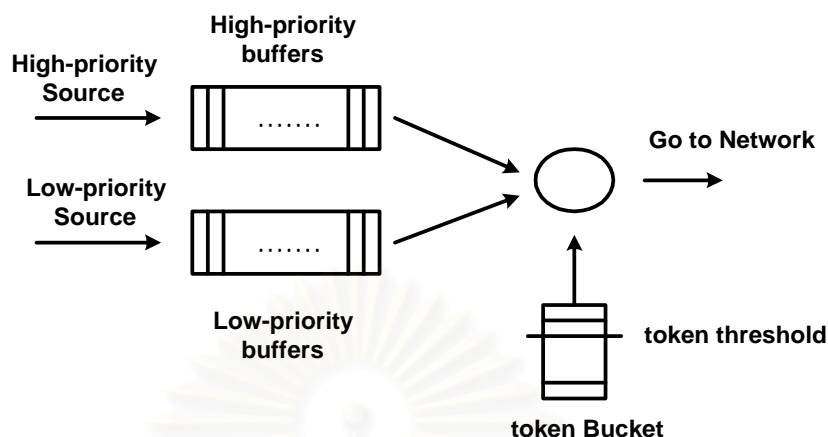
รูปที่ 3.4 ถึงรั้วแบบใช้ถังเก็บโทเค็นร่วม

#### 3.3.1 ถึงรั้วชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ (Priority Buffered Leaky Bucket)

J. Zeng และ L. F. Merakos [4] ได้เสนอการตรวจวัดในระดับวิถีเสมือน (Virtual Path) โดยใช้อัลกอริทึมถึงรั้วซึ่งเรียกว่า ถึงรั้วชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ ซึ่งจะคล้ายกับลักษณะของถึงรั้วแบบใช้ถังเก็บโทเค็นร่วมแต่มีการแบ่งลำดับความสำคัญของแต่ละแหล่งกำเนิดตามชนิดของทราฟฟิกออกเป็น 2 ชนิดดังนี้ แหล่งกำเนิดที่มีลำดับความสำคัญสูง และแหล่งกำเนิดที่มีลำดับความสำคัญต่ำ ซึ่งขึ้นอยู่กับระดับความต้องการอัตราส่วนการสูญหายของเซลล์และเวลาการ



คอยโดยเฉลี่ยของเซลล์ที่แตกต่างกัน นอกจากนี้ยังมีการตั้งระดับจุดเริ่มเปลี่ยนที่ถึงเก็บโทเค็น แสดงดังรูปที่ 3.5



รูปที่ 3.5 ถังร้วชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญ

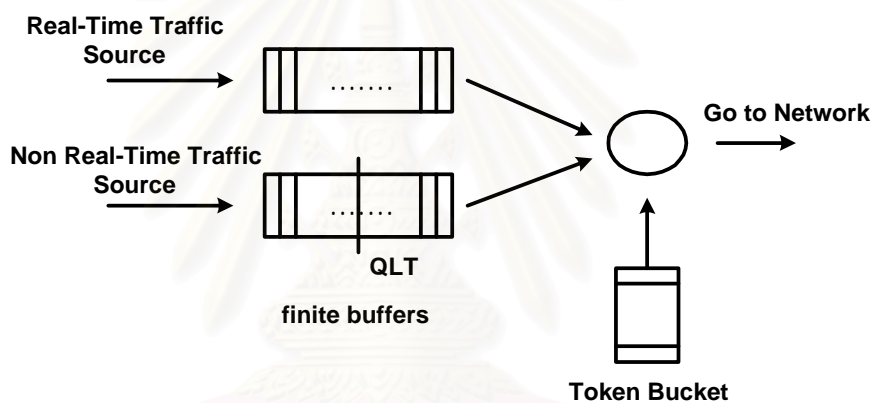
โดยมีหลักการทำงานดังนี้

1. เซลล์ที่เข้ามาจากแหล่งกำเนิดที่มีลำดับความสำคัญสูง ถ้ามีโทเค็นก็จะได้รับโทเค็นและเข้าไปในโครงข่ายได้ แต่ถ้าไม่มีโทเค็นอยู่เซลล์นั้นจะเข้าคิวรอในบัฟเฟอร์ลำดับความสำคัญสูงและจะออกไปได้เมื่อโทเค็นถูกสร้างออกมา
2. เซลล์ที่มาจากแหล่งกำเนิดที่มีลำดับความสำคัญต่ำจะออกไปได้ก็ต่อเมื่อไม่มีเซลล์ค้างอยู่ในบัฟเฟอร์ลำดับความสำคัญสูงและระดับโทเค็นในถังเก็บโทเค็นสูงกว่าค่าจุดเริ่มเปลี่ยน (Threshold)

จะเห็นว่าจุดเริ่มเปลี่ยนช่วยทำให้เวลาการคอยโดยเฉลี่ยของเซลล์ที่เกิดขึ้นในแหล่งกำเนิดที่มีลำดับความสำคัญสูงลดลง เนื่องจากมีโทเค็นสำรองไว้ให้โดยเฉพาะ ซึ่งตรงกับความต้องการของทราฟฟิกแบบเวลาจริง แต่จากการที่ตั้งให้ลำดับความสำคัญกับแหล่งกำเนิดที่มีลำดับความสำคัญสูงมากกว่าแหล่งกำเนิดที่มีลำดับความสำคัญต่ำ ทำให้อัตราการส่งออกของแหล่งกำเนิดที่มีลำดับความสำคัญต่ำมีค่าอัตราการสูญหายของเซลล์สูง จึงได้มีการคิดแก้ไขปัญหานี้โดยมีการนำหลักของ QLT (Queue Length Threshold) Scheduling มาใช้ในถังร้วชนิดที่ใช้บัฟเฟอร์ที่มีการจัดลำดับความสำคัญเพื่อเพิ่มประสิทธิภาพให้กับแหล่งกำเนิดที่มีลำดับความสำคัญต่ำ

### 3.3.2 จุดเริ่มเปลี่ยนความยาวคิว (Queue Length Threshold หรือ QLT) ในถังรับที่จับบัฟเฟอร์ที่มีการจัดลำดับความสำคัญ

D. I. Choi, B. D. Choi และ D. K. Sung [5] เสนอวิธี QLT (Queue Length Threshold) Scheduling ในถังรับชนิดที่มีการจัดลำดับความสำคัญนี้ได้แบ่งแหล่งกำเนิดออกเป็นทราฟฟิกแบบเวลาจริง (Real-Time Traffic) และทราฟฟิกแบบไม่เป็นเวลาจริง (Non Real-Time Traffic) โดยใช้หลักการ QLT Scheduling ในการแบ่งลำดับความสำคัญ และมีจุดเริ่มเปลี่ยนตั้งอยู่ที่บัฟเฟอร์ของทราฟฟิกแบบไม่เป็นเวลาจริง ซึ่งวิธีนี้จะดีกว่าวิธีที่ให้ลำดับความสำคัญกับแหล่งกำเนิดแหล่งใดแหล่งกำเนิดหนึ่งโดยเฉพาะ ทำให้สามารถจัดสรรความต้องการอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ ให้กับแต่ละทราฟฟิกได้อย่างเหมาะสมตามระดับของจุดเริ่มเปลี่ยน ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ถังรับชนิดที่มีการจัดลำดับความสำคัญด้วย QLT

มีหลักการทำงานดังนี้

1. ถ้ามีโทเค็นอยู่ในถังเก็บโทเค็น เซลล์ที่เข้ามาจะได้รับโทเค็นและเข้าไปในโครงข่ายได้ทันทีโดยไม่ขึ้นอยู่กับว่าเป็นทราฟฟิกแบบใด
2. ถ้าจำนวนเซลล์ของทราฟฟิกแบบไม่เป็นเวลาจริง มีค่าต่ำกว่าหรือเท่ากับระดับจุดเริ่มเปลี่ยน โทเค็นที่สร้างออกมาจะให้กับเซลล์ของทราฟฟิกแบบเวลาจริงก่อน
3. ถ้าเซลล์ของทราฟฟิกแบบไม่เป็นเวลาจริง มีค่าเกินระดับจุดเริ่มเปลี่ยน หรือไม่มีเซลล์ค้างในบัฟเฟอร์ของทราฟฟิกแบบเวลาจริง โทเค็นที่สร้างออกมาจะให้กับเซลล์ของทราฟฟิกแบบไม่เป็นเวลาจริง

D. I. Choi, B. D. Choi และ D. K. Sung [5] ได้ทำการเปรียบเทียบถึงวิธีที่ใช้อัตราส่วนการสูญหายที่มีการจัดลำดับความสำคัญ โดยใช้เทคนิค QLT กับวิธีที่ให้ลำดับความสำคัญกับแหล่งกำเนิดที่ไม่ได้ใช้ QLT ใน [4] ผลที่ได้พบว่าเซลล์ของกราฟฟิกแบบไม่เป็นเวลาจริง จะมีค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ต่ำลง ในขณะที่เซลล์ของกราฟฟิกแบบเป็นเวลาจริงมีประสิทธิผลลดลงเล็กน้อย ดังนั้นอาจสรุปได้ว่าผลของการตั้งค่า QLT ทำให้เซลล์ของกราฟฟิกแบบไม่เป็นเวลาจริงได้รับผลดีที่ค่า QLT ต่ำๆ ในขณะที่เซลล์ของกราฟฟิกแบบเวลาจริงจะดีขึ้น โดยมีอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ต่ำที่ค่า QLT สูงๆทำให้เป็นเหตุจูงใจในการที่จะปรับปรุงวิธีการ QLT Scheduling ในถึงวิธีที่ใช้อัตราส่วนการสูญหายที่มีการจัดลำดับความสำคัญ

วิทยานิพนธ์นี้มีจุดประสงค์หลักในการที่จะลดอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ และทำการจัดคุณภาพการบริการให้เป็นไปตามความต้องการในแต่ละประเภทของกราฟฟิก เช่น กราฟฟิกแบบเวลาจริงมีความต้องการเวลาการคอยโดยเฉลี่ยของเซลล์ต่ำ ในขณะที่ยอมให้มีอัตราส่วนการสูญหายของเซลล์ได้ในระดับหนึ่ง ส่วนกราฟฟิกแบบไม่เป็นเวลาจริงจะยอมให้มีเวลาการคอยโดยเฉลี่ยของเซลล์ได้แต่ต้องการอัตราส่วนการสูญหายของเซลล์ต่ำ ซึ่งวิธีการตั้งค่า QLT ที่คงที่ทำให้ได้ค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์เหมาะกับกราฟฟิกชนิดใดชนิดหนึ่งเท่านั้น เพื่อให้ได้ค่า QLT ที่เหมาะสมจึงเสนอวิธีการปรับเปลี่ยนค่า QLT โดยอาศัยค่าพารามิเตอร์ในระบบที่เกี่ยวข้องกับกราฟฟิกแต่ละชนิดซึ่งมีลักษณะที่เปลี่ยนแปลงอยู่ตลอดเวลา มาเป็นตัวแปรในการปรับค่า QLT เป็นแบบพลวัต

จากการทำงานของถึงวิธีที่มีการจัดลำดับความสำคัญพบว่า อัตราส่วนการสูญหายของเซลล์มีค่าสูงในช่วงที่มีอัตราการส่งข้อมูลสูง ซึ่งก็คือช่วงที่มีการส่งเซลล์เข้ามาติดๆกันหรือเป็นช่วงที่มีค่าอัตราเซลล์เฉลี่ยสูงและขณะนั้นก็เป็นช่วงที่มีบัฟเฟอร์เหลืออยู่น้อย ส่วนเวลาการคอยโดยเฉลี่ยของเซลล์ก็จะมีค่ามากในช่วงที่มีอัตราการส่งข้อมูลสูงด้วย ดังนั้นเราจึงพิจารณาค่าอัตราเซลล์เฉลี่ยและอัตราที่ว่างของบัฟเฟอร์ มาเป็นตัวแปรในการปรับค่า QLT ให้เป็นแบบพลวัต แต่ก็เป็นการยากที่จะกำหนดได้แน่นอนว่าค่าที่นำมาพิจารณานั้นจะต้องมากน้อยเพียงใดเพื่อที่จะนำมาตัดสินใจในการปรับค่า QLT ปัญหานี้สามารถแก้ไขได้โดยใช้ตัวควบคุมแบบฟัซซี (Fuzzy Logic Control, FLC) มาช่วยในการพิจารณาปรับเปลี่ยนค่า QLT เนื่องจากตัวควบคุมแบบฟัซซีสามารถให้ผลการควบคุมที่ดีเนื่องจากเป็นอัลกอริทึมที่มีการตัดสินใจคล้ายกับการตัดสินใจของมนุษย์ ในการประเมินค่าพารามิเตอร์ของระบบจะไม่ได้แสดงผลลัพธ์ออกมาเป็นค่าจริงว่า 0 หรือ 1 แต่จะมีระดับการตัดสินใจในช่วง 0 ถึง 1 ซึ่งการตัดสินใจที่ละเอียดอ่อนนี้ถือเป็นข้อเด่นของระบบควบคุมแบบฟัซซี จึงได้ตัดสินใจนำตัวควบคุมแบบฟัซซีมาเป็นตัวควบคุมในการปรับระดับ QLT ให้เหมาะสมกับสภาวะการกำเนิดกราฟฟิกจากแหล่งกำเนิดกราฟฟิกทั้งสองแบบ

### 3.4 การทำงานของ Fuzzy Logic [9]

#### 3.4.1 องค์ประกอบที่เกี่ยวข้องในการทำงานของ Fuzzy Logic มีดังนี้

**Linguistic Variable** เรียกว่า Fuzzy Set หรือ ตัวแปรเชิงภาษา เป็นตัวแปรที่จะนำมาพิจารณาในทางฟัซซี มีทั้งในส่วนของอินพุตและเอาต์พุตของฟัซซี มักจะใช้ตัวแปรที่มีผลต่อสภาวะการทำงานของระบบและผลลัพธ์ที่ต้องการ อาจมีเพียงตัวแปรเดียวหรือหลายตัวแปรขึ้นอยู่กับการออกแบบระบบฟัซซี เขียนในรูปของ  $T(X)$  เมื่อ  $X$  เป็นตัวแปรตัวหนึ่ง เช่นแทนตัวแปรของอัตราการครอบครองบัพเฟอร์ในคิวบัพเฟอร์

**Linguistic Value** เรียกว่า Term Set โดยมีลักษณะเป็นค่าที่มีการบรรยายในเชิงภาษาเป็นตัวอักษรซึ่งตัวแปรทางฟัซซีเป็นไปได้อาจมีเพียงค่าเดียวหรือหลายค่าขึ้นอยู่กับการทำงานและประสิทธิภาพที่ได้ในการควบคุม เช่นค่า Linguistic Value ของตัวแปรที่แทนอัตราการครอบครองบัพเฟอร์ในคิวบัพเฟอร์ที่มี 4 ค่าคือ สูง,ค่อนข้างสูง,ค่อนข้างต่ำและต่ำ เขียนได้ดังนี้

$$T(X) = \{High (H), Weakly High (WH), Weakly Low (WL), Low (L)\}$$

**Membership Function (MF)** เรียกว่า ฟังก์ชันการเป็นสมาชิก เป็นรูปแบบของค่า Term Set แต่ละตัวตลอดช่วงของตัวแปรเชิงภาษา ซึ่งก็คือว่าแต่ละ Term Set ของตัวแปรเชิงภาษาจะถูกกำหนดโดย Membership Function มีได้ทั้งที่เป็นค่า Discrete และค่าที่ต่อเนื่องขึ้นกับชนิดของตัวแปร โดยที่ความสูงของฟังก์ชันสมาชิกจะมีค่าสูงสุดเป็นหนึ่งเสมอ มักจะเขียนอยู่ในรูปของสมการต่างๆเช่น รูปสามเหลี่ยม, สี่เหลี่ยมคางหมู และรูปประฆังคว่ำ เป็นต้น เช่นค่า Membership Function ของตัวแปรที่เป็นอัตราการครอบครองบัพเฟอร์ในคิวบัพเฟอร์ ที่มีค่า 4 ค่า สามารถเขียนได้ในรูป

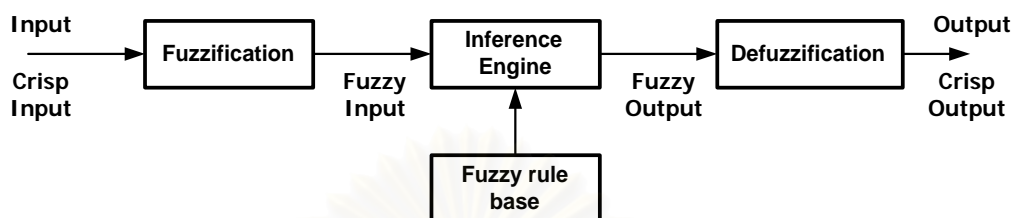
$$M(X) = \{\mu_{high}, \mu_{weaklyhigh}, \mu_{weaklylow}, \mu_{low}\}$$

**Degree of Membership** เป็นค่าที่ได้โดยนำค่าอินพุตที่เป็นค่าจุดมาเทียบกับ Membership Function ของ Fuzzy Set แล้วอ่านค่าที่ได้เป็นระดับของการเป็นสมาชิกนั้นๆ จะได้ค่าออกมาอยู่ในช่วง 0 ถึง 1 ซึ่งเป็นค่าทางฟัซซี

โดยที่  $\mu_{high}(x)$  แทนระดับการเป็นสมาชิกใน Term Set ของค่า High ที่ค่าอินพุต  $x$

### 3.4.2 โครงสร้างพื้นฐานของตัวควบคุมแบบฟัซซี

โครงสร้างโดยทั่วไปของการควบคุมแบบฟัซซีประกอบด้วย 4 ส่วนคือ Fuzzification, Fuzzy Rule Base (ฐานกฎ), Inference Engine และ Defuzzification ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 โครงสร้างพื้นฐานของตัวควบคุมแบบฟัซซี

#### **Fuzzification**

กระบวนการ Fuzzification ทำหน้าที่จัดค่าของตัวแปรอินพุตที่เข้ามาให้เหมาะสมกับที่จะนำไปใช้งาน และทำการเปลี่ยนค่าอินพุตของระบบที่มีลักษณะเป็นค่าจุด (Crisp Value) ให้เป็นค่าอินพุตทางฟัซซี ทำให้ได้ค่าสถานะของระบบในรูปของฟัซซีเซต แล้วจึงส่งค่าที่ได้ไปยังส่วน Inference Engine เพื่อเปรียบเทียบสถานะของระบบกับเงื่อนไขในกฎที่กำหนดไว้ และทำการหาการควบคุมที่เหมาะสมต่อไป

#### **Fuzzy Rule Base (ฐานกฎ)**

ฐานกฎ มีหน้าที่คือ แสดงข้อมูลของสถานะของระบบเข้ากับการควบคุมที่เหมาะสมในรูปของ Fuzzy If-Then Rule (Fuzzy Implication หรือ Fuzzy Condition Statement) โดยทั่วไปจะเขียนอยู่ในรูปของ

If **x** is **A** Then **y** is **B**

โดยที่ **A** และ **B** เป็นค่า Linguistic Value ของค่าอินพุตและเอาต์พุตตามลำดับ

**x** และ **y** เป็นค่า Linguistic Variable ของค่าอินพุตและเอาต์พุตตามลำดับ หรืออาจเขียนได้ว่า

If (สถานะของระบบ) Then (กริยาควบคุม)

ตัวอย่างเช่น กฎจากการควบคุมการทำงานของเครนยกของ

If ( Distance = Far AND Angle = Zero ) Then ( Power = Medium)

กฎดังกล่าวข้างต้นทั้งหมดจะถูกนำไปใช้ในการทำงานของ Inference Engine ซึ่งได้มีการศึกษามาอย่างกว้างขวางและมีการเสนอวิธีการต่างๆ หลายวิธีซึ่งมีความเหมาะสมแตกต่างกันตามลักษณะการใช้งาน

### ***Inference Engine***

กระบวนการ Inference Engine เป็นกลไกสำคัญที่จะนำสถานะของระบบที่ตรวจวัดได้ในรูปของค่าทางฟัซซี มาเปรียบเทียบกับกฎการควบคุมในส่วนของ Fuzzy Rule Base ซึ่งอยู่ในรูป Fuzzy If-Then Rule โดยมีวิธีการจัดการได้หลายรูปแบบ เช่นวิธีของ Mamdani, Sugeno และ Tsukamoto เป็นต้น ผลที่ได้จะใช้เป็นตัวควบคุมเพื่อให้กระบวนการดำเนินไปในลักษณะที่ต้องการ โดยอยู่ในรูปของค่าเอาต์พุตทางฟัซซี และส่งต่อไปยังส่วนของ Defuzzification

### ***Defuzzification***

กระบวนการ Defuzzification ทำหน้าที่ในการเปลี่ยนค่าเอาต์พุตของระบบที่มีลักษณะเป็นค่าทางฟัซซี ให้เป็นค่าเอาต์พุตของระบบที่มีลักษณะเป็นค่าจุด (Crisp Value) และทำการแปลงค่าจุดของตัวแปรควบคุมเป็นค่ากายภาพ ทำให้ได้ค่าที่ใช้เป็นตัวควบคุมเพื่อให้กระบวนการดำเนินไปในลักษณะที่ต้องการ

### **3.4.3 หลักพิจารณาในการเลือกองค์ประกอบของตัวควบคุมแบบฟัซซี**

เนื่องจากการทำงานของตัวควบคุมแบบฟัซซีประกอบด้วยการทำงานหลายส่วนและในแต่ละส่วนได้มีการศึกษาวิธีการทำงานอย่างกว้างขวางโดยนักวิจัยหลายท่าน มีการเสนอวิธีการต่างๆ มาหลายวิธีซึ่งมีความเหมาะสมแตกต่างกันตามลักษณะการใช้งาน ในส่วนของหัวข้อนี้จะกล่าวถึงองค์ประกอบต่างๆในการทำงานของตัวควบคุมแบบฟัซซีและวิธีการที่เลือกใช้ในแบบจำลองการทำงานที่จะนำเสนอต่อไป

#### **3.4.3.1 การเลือกฟังก์ชันการเป็นสมาชิก**

รูปร่างของฟังก์ชันการเป็นสมาชิกที่เป็นที่นิยมมากได้แก่รูปสามเหลี่ยม รูปสี่เหลี่ยมคางหมู และรูปประฆัง เนื่องจากสามารถอธิบายได้ในรูปของฟังก์ชันและพารามิเตอร์ได้ง่ายไม่ซับซ้อน ใช้หน่วยความจำในการเก็บข้อมูลของรูปร่างน้อยและจัดการได้อย่างมีประสิทธิภาพในเทอมของเงื่อนไขเวลาจริง

การทำงานของตัวควบคุมแบบฟัซซีที่จะใช้ในแบบจำลองที่จะนำเสนอต่อไป ได้เลือกใช้ฟังก์ชันการเป็นสมาชิกของทั้งตัวแปรอินพุตและเอาต์พุตเป็นรูปสี่เหลี่ยมคางหมู และรูปสามเหลี่ยม

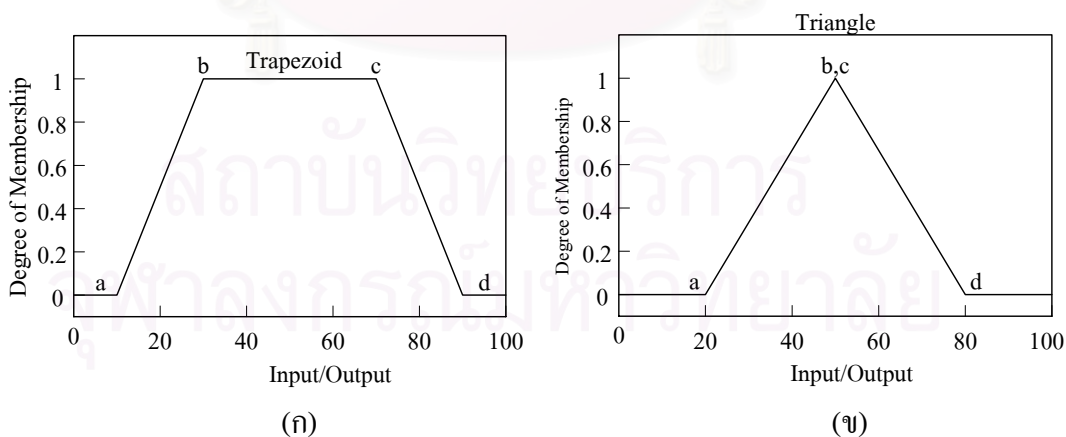
เนื่องจากให้ผลตอบแทนในการทำงานเร็วเหมาะกับการทำงานในแบบเป็นเวลาจริง [9] โดยมีพารามิเตอร์ในการกำหนดรูปร่างของฟังก์ชันอยู่ 4 ตัวคือ {a,b,c,d} ดังแสดงในสมการที่ (3.1)

$$Trapezoid(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (3.1)$$

สามารถเขียนอยู่ในรูปของค่าต่ำที่สุด “min” และค่าสูงที่สุด “max” ได้ดังสมการที่ (3.2)

$$Trapezoid(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (3.2)$$

นอกจากนี้ยังเป็นการง่ายในการที่จะใช้งานร่วมกับฟังก์ชันการเป็นสมาชิกของรูปทรงสามเหลี่ยมเนื่องจากสามารถใช้สมการเดียวกันในการทำงานได้ โดยใช้พารามิเตอร์ “b” และ “c” ในรูปสี่เหลี่ยมคางหมูเป็นตัวเดียวกันในการแทนจุดยอดของรูปสามเหลี่ยม พารามิเตอร์ในการออกแบบฟังก์ชันการเป็นสมาชิกของรูปสี่เหลี่ยมคางหมู {x;10,30,70,90} แสดงในรูปที่ 3.8 (ก) และฟังก์ชันการเป็นสมาชิกของรูปสามเหลี่ยม {x;20,50,80} แสดงในรูปที่ 3.8 (ข)

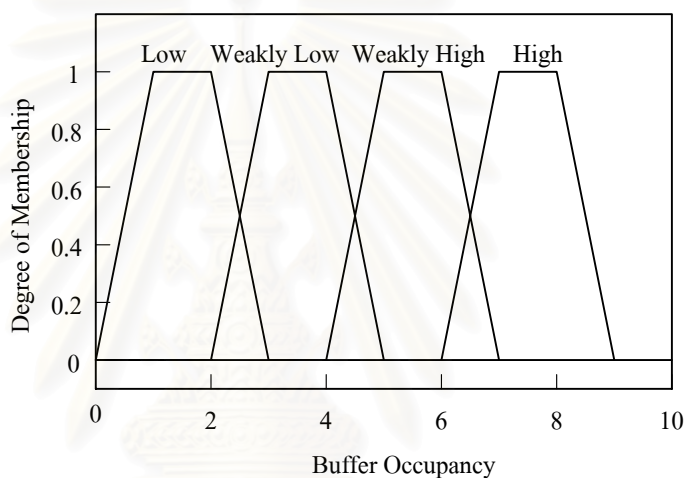


รูปที่ 3.8 พารามิเตอร์ของ MF (ก) Trapezoid{x;10,30,70,90} (ข) Triangle{x;20,50,80}

### 3.4.3.2 การเลือกค่า Linguistic Variable และ Linguistic Value

ตัวแปรเชิงภาษาที่เลือกมาพิจารณาในการตัดสินใจควบคุมระบบนั้นโดยมากจะเลือกพิจารณาจากตัวแปรที่มีผลกระทบต่อระบบและเกี่ยวข้องกับการดำเนินงานของตัวควบคุมเพื่อให้กระบวนการดำเนินไปในลักษณะที่ต้องการ ส่วนการเลือกค่า Term Set ของตัวแปรเชิงภาษาหนึ่งๆ จะประกอบด้วยจำนวนจำกัดของค่าที่แสดงได้ด้วยอักษรซึ่งตัวแปรภาษานั้นๆเป็นไปได้ เช่น ถ้าเลือกตัวแปรภาษาเป็นค่าของอัตราการครอบครองบัพเฟอร์ในคิวบัพเฟอร์แล้ว Term Set ที่เลือกใช้มักจะกำหนดเป็นเทอมของปริมาณที่มีเครื่องหมายและขนาดต่างกันเช่น Term Set ที่มี 4 ค่าคือ สูง, ค่อนข้างสูง, ค่อนข้างต่ำและต่ำ ที่มีฟังก์ชันการเป็นสมาชิกเป็นรูปสี่เหลี่ยมคางหมู ดังแสดงในรูปที่

3.9



รูปที่ 3.9 ฟังก์ชันการเป็นสมาชิกของ Term Set ค่าสูง, ค่อนข้างสูง, ค่อนข้างต่ำและต่ำ

ถ้าต้องการรายละเอียดมากขึ้นต้องทำการแบ่งค่าให้มากขึ้น อย่างไรก็ตามการเพิ่มจำนวนของ Term Set หรือตัวแปรเชิงภาษามีผลทำให้ต้องเพิ่มจำนวนกฎในส่วนของฐานกฎที่ต้องใช้มากขึ้นด้วย เช่น Term Set ของตัวแปรเชิงภาษา 2 ตัวที่มีจำนวนสมาชิกแต่ละ 10 สมาชิกจะมีจำนวนกฎสูงสุดในฐานกฎเป็น 100 ข้อเป็นต้น

### 3.4.3.3 วิธีการ Fuzzy Inference

ในกระบวนการ Fuzzy Inference นั้นได้มีการวิจัยและสรุปออกมาเป็นวิธีการทำงานหลายรูปแบบดังเช่น วิธีของ Mamdani, Sugeno และ Tsukamoto โดยทั้งสามวิธีนี้เป็นที่นิยมใช้ในการทำงานอย่างแพร่หลาย ในแต่ละวิธีก็จะมีการทำงานที่แตกต่างกัน ในที่นี้จะกล่าวถึงการทำงานของ Mamdani Inference System หรืออาจเรียกว่าวิธี Max-Min Inference ที่เลือกใช้แบบจำลองที่จะนำเสนอต่อไป การทำงานของวิธี Max-Min Inference จะประกอบด้วย 2 ส่วนคือ



- **Aggregation**

ทำหน้าที่ประมวลผลในส่วนของ “If” ในฐานกฎที่อยู่ในรูป If-Then Rule ซึ่งก็คือการหาค่าผลลัพธ์ในส่วนของเงื่อนไข If ที่แทนสถานะของระบบ ที่อยู่ในฐานกฎข้อเดียวกัน มีวิธีกระทำโดยใช้ตัวดำเนินการ min (Intersection, AND) เขียนได้ในรูปของสมการที่ (3.3)

$$\mu_{A \cap B}(u) = \min \{ \mu_A(u), \mu_B(u) \} \quad (3.3)$$

โดยที่ A และ B เป็นฟัซซีเซตใน  $U$  และ  $u \in U$  (Universe of Discourse)

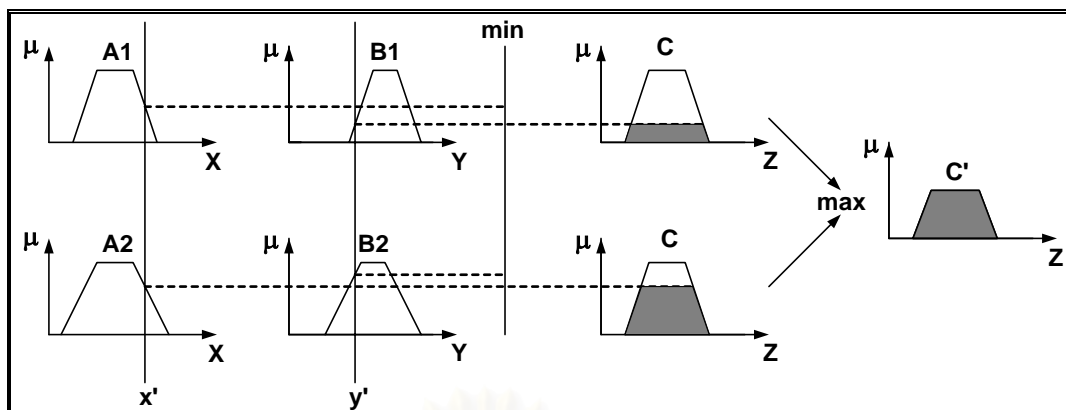
- **Composition**

ทำหน้าที่ประมวลผลในส่วนของ “Then” ในฐานกฎที่อยู่ในรูป If-Then Rule ซึ่งก็คือการหาค่าผลลัพธ์ในส่วนของเงื่อนไข Then ที่แทนกริยาควบคุม ที่อยู่ใน Term Set เดียวกันของตัวแปรภาษาทางด้านเอาต์พุต มีวิธีกระทำโดยใช้ตัวดำเนินการ max (Union, OR) เขียนได้ในรูปของสมการที่ (3.4)

$$\mu_{A \cup B}(u) = \max \{ \mu_A(u), \mu_B(u) \} \quad (3.4)$$

โดยที่ A และ B เป็นฟัซซีเซตใน  $U$  และ  $u \in U$  (Universe of Discourse)

จากการทำงานของทั้งสองส่วนจะได้ค่าของเอาต์พุตที่เป็นค่าทางฟัซซี อยู่ในรูปที่เป็นค่าของ Term Set ต่างๆของตัวแปรภาษาทางด้านเอาต์พุต ซึ่งจะต้องส่งค่าเหล่านี้ไปยังกระบวนการ Defuzzification เพื่อแปลงค่าทางฟัซซีให้เป็นค่าจริงที่สามารถนำไปใช้ในการควบคุมระบบได้ รูปที่ 3.10 แสดงตัวอย่างการทำงานของ Max-Min Inference System เทียบกับฐานกฎสองข้อ โดยมีอินพุตค่าจุดเป็นค่า  $x'$  และ  $y'$  ได้เอาต์พุต เป็นค่า  $C'$  ที่เป็นค่าทางฟัซซี เมื่อกำหนดให้ A1, A2 และ B1, B2 เป็น Term Set ของตัวแปรเชิงภาษาทางด้านอินพุต A และ B ตามลำดับ และ C เป็น Term Set ของตัวแปรเชิงภาษาทางด้านเอาต์พุต



รูปที่ 3.10 วิธีการ Max-Min Inference

#### 3.4.3.4 การกำหนดกฎในฐานกฎ [10]

การกำหนดกฎในฐานกฎทำได้หลายวิธี ซึ่งอาจนำมาใช้ร่วมกันเพื่อให้ได้การกำหนดกฎที่มีประสิทธิภาพดียิ่งขึ้น

วิธีที่ 1 เป็นวิธีที่ใช้กันอย่างแพร่หลาย โดยทำการกำหนดกฎบนพื้นฐานความรู้ ประสบการณ์ของผู้ปฏิบัติการ ผู้เชี่ยวชาญ หรือวิศวกรควบคุม

วิธีที่ 2 วิธีนี้ยังอยู่ในขั้นเริ่มต้น เป็นการหาแบบจำลองพีชชีของกระบวนการภายใต้การควบคุมขึ้นก่อน แล้วจึงทำการสร้างกฎการควบคุมพีชชีจากการวิเคราะห์แบบจำลองนั้น

วิธีที่ 3 วิธีนี้ยังอยู่ในขั้นเริ่มต้นเช่นกัน โดยมีลักษณะของการใช้วิธีการควบคุมแบบดั้งเดิม แต่ใช้แนวคิดของระบบพีชชีเข้ามาประกอบ ทำให้ได้การควบคุมที่ให้สมรรถนะคล้ายกับวิธีดั้งเดิม แต่อยู่ในรูปแบบของกฎการควบคุมพีชชี

วิธีที่ 4 เป็นวิธีการเรียนรู้ เป็นการกำหนดให้ตัวควบคุมสามารถเรียนรู้เองว่ากฎหรือพารามิเตอร์ที่สำคัญเช่นฟังก์ชันการเป็นสมาชิกที่เหมาะสมควรเป็นอย่างไร โดยการปรับตัวให้เหมาะสมขึ้น โดยดูจากสมรรถนะการทำงานในปัจจุบัน แนวทางนี้กำลังเป็นที่สนใจศึกษากันอยู่ในปัจจุบัน

### 3.4.3.5 วิธีการ Defuzzification

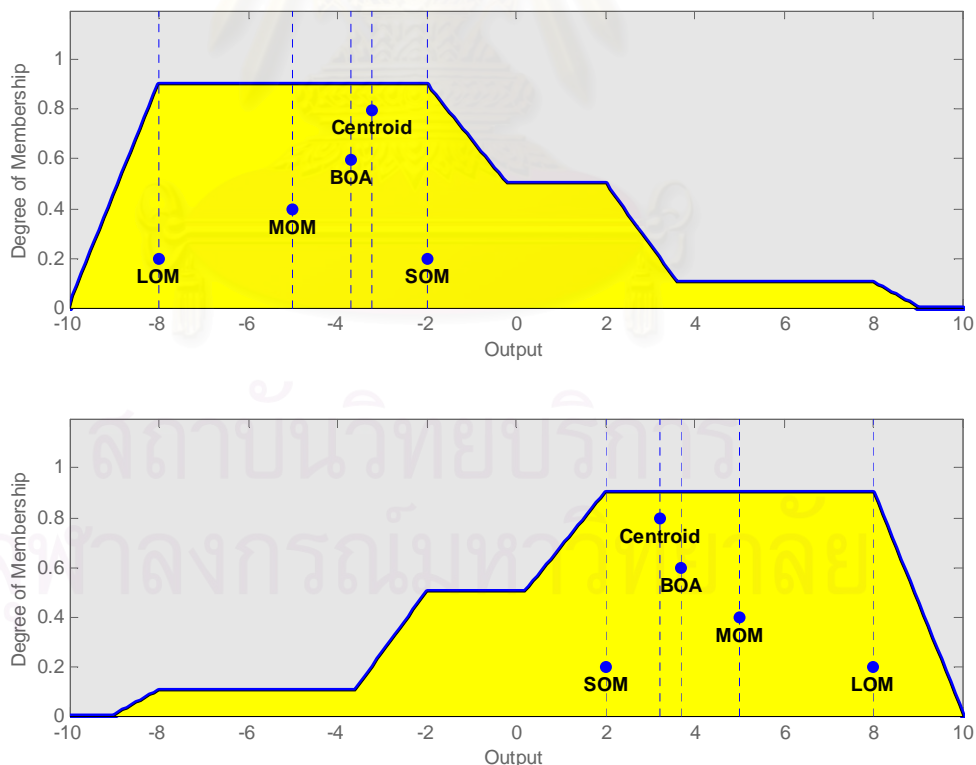
หลักการ Defuzzification มีอยู่ 5 วิธี ผลของการทำงานในแต่ละวิธีแสดงได้ดังรูปที่ 3.11

1. วิธีจุดศูนย์กลางถ่วง (Centroid) เป็นวิธีที่นิยมใช้กันมากที่สุด ค่าที่ได้หาได้จากสมการที่ (3.5)

$$u^* = \frac{\int_U \mu_U(u) u \, du}{\int_U \mu_U(u) \, du} \tag{3.5}$$

โดยที่  $\mu_U(u)$  เป็นผลรวมของค่าเอาต์พุตทางฟัซซี

2. วิธี Bisector of Area (BOA) เป็นค่าที่แบ่งพื้นที่ที่ออกเป็นสองส่วนเท่าๆกัน
3. วิธี Mean of Maximum (MOM) เป็นค่าเฉลี่ยของค่าที่มีระดับการเป็นสมาชิกสูงที่สุด
4. วิธี Smallest of Maximum (SOM) เป็นค่าน้อยที่สุดของค่าที่มีระดับการเป็นสมาชิกสูงที่สุด
5. วิธี Largest of Maximum (LOM) เป็นค่ามากที่สุดของค่าที่มีระดับการเป็นสมาชิกสูงที่สุด



รูปที่ 3.11 วิธีการ Defuzzification ในแบบต่างๆ

## บทที่ 4

### แบบจำลอง

#### 4.1 แบบจำลองแหล่งกำเนิด (Source Model)

แบบจำลองแหล่งกำเนิดที่ใช้ในวิทยานิพนธ์แบ่งออกเป็น 2 แบบคือ แบบแรกเป็นแหล่งกำเนิดที่อ้างอิงจากบทความ [5] เพื่อนำมาทดสอบความถูกต้องของแบบจำลองที่ออกแบบโดยพิจารณาถึงความถูกต้องและแนวโน้มที่ได้จากการทำงานของอัลกอริทึมถึงรัวชนิดที่มีการจัดลำดับความสำคัญ ส่วนแบบจำลองแบบที่สองใช้แบบจำลองแหล่งกำเนิดแบบเบิสต์ (Bursty Source Model) หรือเรียกว่าแบบจำลองแหล่งกำเนิดแบบเปิด-ปิด (On-Off Source Model) แบบจำลองแหล่งกำเนิดแบบนี้จะถูกนำมาใช้ในการทดสอบการทำงานของอัลกอริทึมถึงรัวชนิดต่างๆ เหตุผลในการเลือกใช้แบบจำลองแบบที่สองแทนแบบแรกนี้คือเพื่อทำให้เกิดความแตกต่างในการจำลองแบบระหว่างแหล่งกำเนิดทราฟฟิกแบบเวลาจริง (Real-Time Traffic Source) และแหล่งกำเนิด ทราฟฟิกแบบไม่เป็นเวลาจริง (Non Real-Time Traffic Source) โดยพิจารณาจากอัตราการเบิสต์ของข้อมูลที่แตกต่างกัน และแหล่งกำเนิดแบบนี้เป็นที่นิยมใช้ในการจำลองแบบการส่งข้อมูลในโครงข่ายเอทีเอ็มอย่างแพร่หลาย

##### 4.1.1 แบบจำลองแหล่งกำเนิดแบบที่ 1 [5]

ใช้ในการจำลองแบบแหล่งกำเนิดทราฟฟิก 2 ชนิดคือ แหล่งกำเนิดทราฟฟิกแบบเวลาจริง และแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง ซึ่งมีกระบวนการมาถึงของเซลล์เป็นแบบปัวส์ซง (Poisson Arrival Process) ที่มีฟังก์ชันการแจกแจงความน่าจะเป็นแบบเอ็กซ์โพเนนเชียลเชิงลบ (Negative Exponential Distribution) และมีอัตราการมาถึงของเซลล์เป็น  $\lambda$  สำหรับช่วงเวลาการมาถึงของเซลล์เป็นตัวแปรสุ่มแบบเอ็กซ์โพเนนเชียล (Exponential Random Variables) ดังสมการที่ (4.1)

$$t_i = t_{i-1} - \left(\frac{1}{\lambda}\right) \ln U \quad (4.1)$$

โดยที่  $U$  แทนค่าที่ได้จากการสุ่มแบบยูนิฟอร์ม มีค่าอยู่ในช่วง 0 ถึง 1

##### 4.1.2 แบบจำลองแหล่งกำเนิดแบบที่ 2

ใช้ในการจำลองแบบแหล่งกำเนิดทราฟฟิก 2 ชนิดคือ แหล่งกำเนิดทราฟฟิกแบบเวลาจริง และแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง โดยแบบจำลองแหล่งกำเนิดทราฟฟิกทั้งสองนี้เป็น

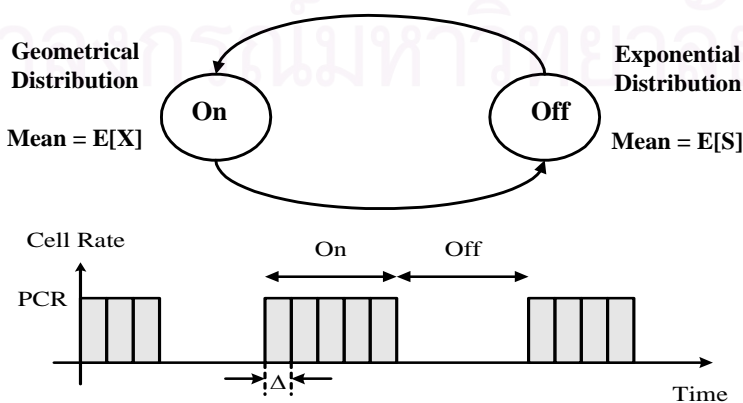
แบบ On/Off หรือ Bursty Source Model ซึ่งแบบจำลองแหล่งกำเนิดแบบนี้เป็นที่นิยมใช้ในการจำลองแบบการส่งข้อมูลทั้งทราฟฟิกแบบเวลาจริงในการส่งทราฟฟิกเสียง เช่น Packetized Voice Traffic และทราฟฟิกแบบไม่เป็นเวลาจริงในการส่งสัญญาณข้อมูล เช่น ทราฟฟิกภาพนิ่ง หรือข้อมูลทางคอมพิวเตอร์ มีลักษณะแสดงดังรูปที่ 4.1 ลักษณะของการส่งข้อมูลซึ่งในที่นี้ใช้แทนการส่งเซลล์ในโครงข่ายเอทีเอ็มแบ่งได้เป็นสองลักษณะตามชื่อคือ ช่วง “ON” หรือช่วง “Burst” จะทำการส่งข้อมูลตลอดด้วยอัตราการส่งที่ Peak Rate ในขณะที่ช่วง “OFF” หรือช่วง “Silent” จะไม่มีการส่งข้อมูลเลย โดยที่จำนวนเซลล์ที่ส่งในช่วง ON ถูกสร้างแบบสุ่มด้วยฟังก์ชันการแจกแจงแบบเรขาคณิต (Geometric Distribution) ที่ค่า Mean  $E[X]$  และมีค่าระยะห่างระหว่างเซลล์คงที่ที่ค่า “ $\Delta$ ” ( $1/$  Peak Rate) เมื่อ  $p$  คือความน่าจะเป็นของการไม่เกิดเซลล์ จะได้ว่าฟังก์ชันความน่าจะเป็นและค่าเฉลี่ยของจำนวนเซลล์ที่ส่งในช่วง ON ( $E[X]$ ) แสดงได้ดังสมการที่ (4.2) และ (4.3) [11] ตามลำดับ

$$f(x; p) = \begin{cases} p(1 - p)^x & ; x \in \{0, 1, \dots\} \\ 0 & ; otherwise \end{cases} \quad (4.2)$$

$$E(X) = \frac{1 - p}{p} \quad (4.3)$$

ส่วนระยะเวลาในช่วง OFF ถูกสร้างแบบสุ่มด้วยฟังก์ชันการแจกแจงแบบเอ็กซ์โพเนนเชียลเชิงลบ (Negative Exponential Distribution) ที่ค่า Mean  $E[S]$  เมื่อให้ค่า  $\beta$  คือระยะเวลาเฉลี่ยเฉลี่ย จะได้ฟังก์ชันความน่าจะเป็นของระยะเวลาเฉลี่ย ( $s$ ) แสดงได้ดังสมการที่ (4.4) [11] สำหรับค่าพารามิเตอร์ต่างๆ แสดงดังตารางที่ 4.1

$$f(s; \beta) = \frac{1}{\beta} e^{-\frac{s}{\beta}} \quad (4.4)$$



รูปที่ 4.1 On/Off Traffic Model

ตารางที่ 4.1 คุณสมบัติของแหล่งกำเนิด On/Off

Parameter	Real Time (Voice)	Non Real Time (Data)
Peak Rate	32 kbps	44.8 kbps
On Time	352 ms	500 ms
Off Time	650 ms	1,500 ms
Burstiness	2.85	4
Mean rate	11.2 kbps	11.2 kbps

#### 4.2 ตัววัดสมรรถนะ

ในวิทยานิพนธ์นี้พิจารณาตัววัดสมรรถนะ 3 ตัว คือ อัตราส่วนการสูญหายของเซลล์, เวลาการคอยโดยเฉลี่ยของเซลล์และค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์

อัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio)

อัตราส่วนการสูญหายของเซลล์ คำนวณได้จากสมการที่ (4.5)

$$\text{อัตราส่วนการสูญหายของเซลล์} = \frac{\text{จำนวนเซลล์ที่ส่งทั้งหมด} - \text{จำนวนเซลล์ที่ปลายทางรับได้}}{\text{จำนวนเซลล์ที่ส่งทั้งหมด}} \quad (4.5)$$

ในที่นี้การสูญหายของเซลล์จะเกิดในกรณีที่แบบจำลองแหล่งกำเนิดสร้างเซลล์ออกมาในขณะที่บัฟเฟอร์ที่ไซรองรับได้เต็มแล้วจึงทำให้เซลล์นั้นถูกทิ้งไป

เวลาการคอยโดยเฉลี่ยของเซลล์ (Mean Waiting Time)

เวลาการคอยโดยเฉลี่ยของเซลล์คำนวณได้จากสมการที่ (4.6)

$$\text{เวลาการคอยโดยเฉลี่ยของเซลล์} = \frac{\text{ผลรวมของเวลาที่แต่ละเซลล์อยู่ในบัฟเฟอร์}}{\text{จำนวนเซลล์ที่เข้ามาในบัฟเฟอร์ทั้งหมด}} \quad (4.6)$$

เวลาเฉลี่ยของเซลล์ที่อยู่ในคิวบัฟเฟอร์จะไม่พิจารณาเซลล์ที่ไม่ได้เข้ามาในบัฟเฟอร์เลย เช่น เซลล์ที่เกิดการสูญหาย ส่วนเซลล์ที่เข้ามาในบัฟเฟอร์แล้วสามารถเข้าสู่โครงข่ายได้โดยถือว่าไม่เกิดเวลาการคอยของเซลล์ เช่น กรณีที่มีโทเค็นเหลืออยู่ในขณะที่เซลล์นั้นมาถึง

ค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ (Average Time for the First Loss)

การวัดค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์วัดจากช่วงเวลาตั้งแต่ที่แหล่งกำเนิดกราฟฟิกให้กำเนิดเซลล์แล้วใช้เวลานานเท่าไรจึงจะส่งผลให้เซลล์ของแหล่งกำเนิดกราฟฟิกนั้นตรวจพบการสูญหายของเซลล์เป็นเซลล์แรก โดยแบ่งพิจารณาเป็นสองช่วงคือในช่วงที่แหล่งกำเนิดส่งเซลล์เข้าโครงข่ายที่อัตราเฉลี่ยต่ำกว่าขีดตกลงและในอัตราที่ละเมิดขีดตกลง ค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ควรจะมีความสูงในช่วงที่แหล่งกำเนิดส่งเซลล์เข้าโครงข่ายที่อัตราเฉลี่ยต่ำกว่าขีดตกลง ซึ่งแสดงว่าอัลกอริทึมที่ใช้ควบคุมนี้เป็นผลดี เนื่องจากเริ่มเกิดการสูญหายได้ช้า ทำให้ลดความน่าจะเป็นของการเกิด False Alarm ลงได้ ส่วนกรณีที่แหล่งกำเนิดส่งเซลล์ด้วยอัตราที่ละเมิดขีดตกลง ค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ก็ควรมีค่าต่ำ เพื่อที่จะทำให้สามารถจัดการทั้งเซลล์ที่ละเมิดขีดตกลงได้อย่างรวดเร็ว

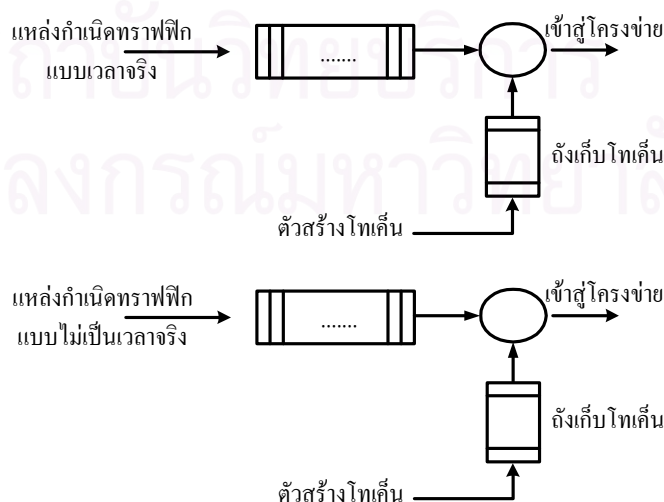
### 4.3 ข้อกำหนดของแบบจำลอง

#### 4.3.1 แบบจำลองการทำงานของ Leaky Bucket ชนิดต่างๆ

แบบจำลองที่ใช้ทดสอบการทำงานของ Leaky Bucket 4 ชนิดคือ Token Leaky Bucket (TLB), Priority Leaky Bucket (PLB), Priority Leaky Bucket ที่ใช้เทคนิค QLT (QLT) และ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT (วิธีที่เสนอ) มีข้อกำหนดแบ่งตามลักษณะการใช้งานของถังเก็บโทเค็นได้ดังนี้

#### 1. การใช้ถังเก็บโทเค็นแยกกันในแต่ละแหล่งกำเนิด

Leaky Bucket ที่มีการใช้ถังเก็บโทเค็นแยกกัน ได้แก่ Token Leaky Bucket (TLB) ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 Token Leaky Bucket

ข้อกำหนดที่ใช้มีดังนี้

1. แหล่งกำเนิด

ใช้แหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ตามหัวข้อที่ 4.1.2

2. ขนาดบัฟเฟอร์

- บัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง ( $B_r$ ) มีขนาดเท่ากับ 50 เซลล์
- บัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ( $B_{nr}$ ) มีขนาดเท่ากับ 100 เซลล์

3. ถังเก็บโทเค็น ( $M$ )

ใช้ถังเก็บโทเค็นของแต่ละแหล่งกำเนิดมีขนาดเท่ากันคือ 25 เซลล์

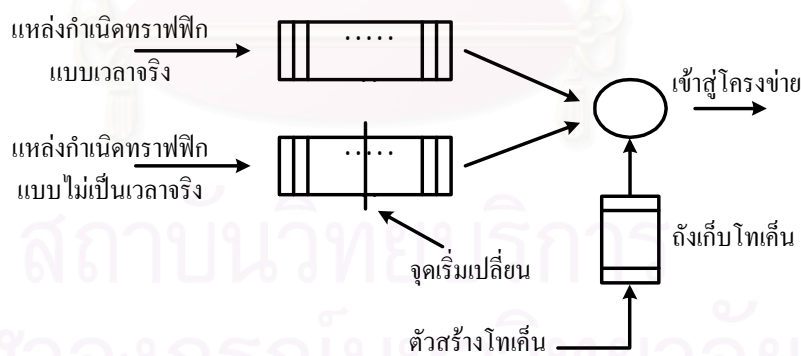
4. อัตราการสร้างโทเค็น

ใช้อัตราการสร้างโทเค็นเท่ากับ Mean Cell Rate ของแหล่งกำเนิดแต่ละแหล่ง

2. วิธีการใช้ถังเก็บโทเค็นร่วมกัน

Leaky Bucket ที่มีการใช้ถังเก็บโทเค็นร่วมกันแบ่งออกเป็น 3 วิธีได้แก่

- Priority Leaky Bucket (PLB)
- Priority Leaky Bucket ที่ใช้เทคนิค QLT (QLT)
- Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT (วิธีที่เสนอ)



รูปที่ 4.3 Leaky Bucket ที่มีการใช้ถังเก็บโทเค็นร่วมกัน

ข้อกำหนดที่ใช้มีดังนี้

1. แหล่งกำเนิด

ใช้แหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ตามหัวข้อที่ 4.1.2



2. ขนาดบัฟเฟอร์

- บัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงมีขนาดเท่ากับ 50 เซลล์
- บัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีขนาดเท่ากับ 100 เซลล์

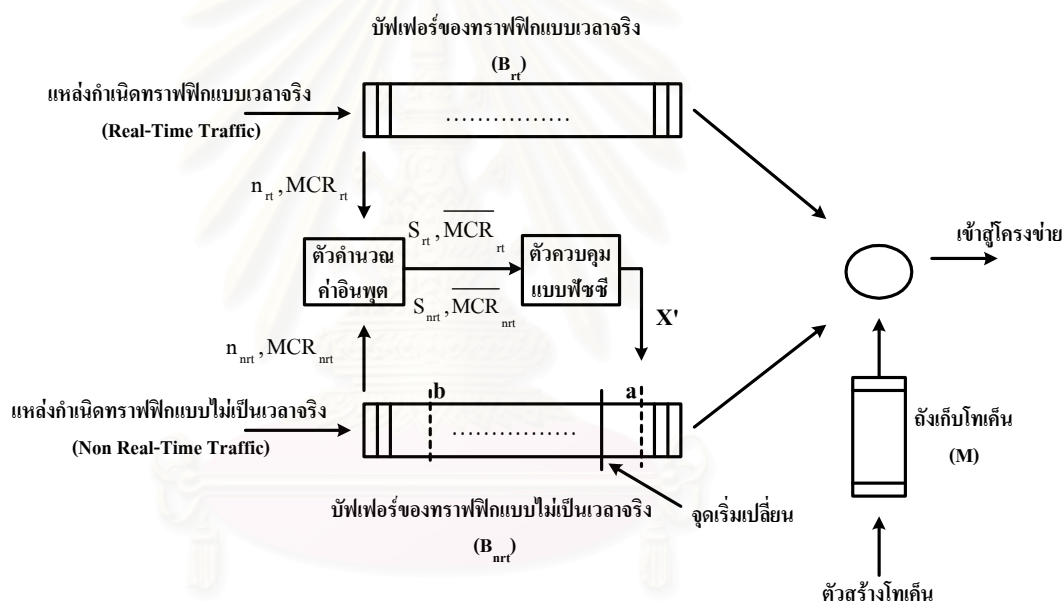
3. ถังเก็บโทเค็น

ใช้ถังเก็บโทเค็นของแหล่งกำเนิดแต่ละแหล่งร่วมกันมีขนาด 50 เซลล์

4. อัตราการสร้างโทเค็น

ใช้อัตราการสร้างโทเค็นของแหล่งกำเนิดทั้งสองชนิดร่วมกันเท่ากับผลรวมของ Mean Cell Rate ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและ Mean Cell Rate ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

4.3.2 แบบจำลองการทำงานของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT (วิธีที่เสนอ)



รูปที่ 4.4 แบบจำลองการปรับค่า QLT ด้วยตัวควบคุมแบบฟัซซี

แบบจำลองการทำงานประกอบด้วยบัฟเฟอร์ขาเข้าของกราฟฟิกแบบเวลาจริงมีและแบบไม่เป็นเวลาจริงขนาด 50 เซลล์ และ 100 เซลล์ตามลำดับ ตัวคำนวณค่าอินพุต ตัวควบคุมแบบฟัซซี และถังเก็บโทเค็นขนาด 50 เซลล์ โดยค่าของขนาดบัฟเฟอร์และขนาดถังเก็บโทเค็นใช้อัตราส่วนเดียวกับบทความ [5] แต่ทำการเพิ่มขนาดของบัฟเฟอร์และถังเก็บโทเค็นขึ้น เพื่อให้เหมาะสมกับแหล่งกำเนิดที่ใช้

ในวิทยานิพนธ์ต่อจากนี้ไปจะใช้คำย่อ MCR แทนอัตราเซลล์เฉลี่ย (Mean Cell Rate) ซึ่งมีความหมายเหมือนกับ SCR (Sustainable Cell Rate) ในหัวข้อที่ 2.2 ในวิทยานิพนธ์นี้ โดยไม่ได้หมายถึง Minimum Cell Rate ที่กล่าวถึงในหัวข้อที่ 2.2 อีกต่อไป

หลักการทํางานของตัวคํานวณคําค่าอินพุตจะกระทำในสองกรณี คือ กรณีมีเซลล์เข้ามาในบัฟเฟอร์ขาเข้าของกราฟฟิคทั้งสองแบบ เพื่อกํานวณหาคําค่าอินพุตที่ว่างของบัฟเฟอร์ (Buffer Space, S) และกรณีมีเบิรสต์ใหม่เข้ามาเพื่อกํานวณหาคําค่าอินพุตที่เป็นค่าเฉลี่ยของอัตราเซลล์เฉลี่ยในเบิรสต์ที่ผ่านมา (Average Mean Cell Rate,  $\overline{MCR}$ ) หลังจากนั้นจะทำการส่งค่าที่คํานวณได้ไปเป็นคําค่าอินพุตของตัวควบคุมแบบพีซซี โดยตัวคํานวณคําค่าอินพุตมีหลักการทํางานดังต่อไปนี้

เมื่อมีเซลล์จากกราฟฟิคแบบเวลาจริง ( $r_t$ ) หรือกราฟฟิคแบบไม่เป็นเวลาจริง ( $nr_t$ ) เข้ามาตัวคํานวณคําค่าอินพุตจะคํานวณคําค่าอินพุตที่ว่างของบัฟเฟอร์ได้ตามสมการที่ (4.7) และ (4.8)

$$S_{r_t} = B_{r_t} - n_{r_t} \quad (4.7)$$

$$S_{nr_t} = B_{nr_t} - n_{nr_t} \quad (4.8)$$

โดยที่  $n$  คือ การครอบครองบัฟเฟอร์ของกราฟฟิคแต่ละแบบ

เมื่อมีเบิรสต์ใหม่เข้ามาในกราฟฟิคแต่ละแบบตัวคํานวณคําค่าอินพุตจะทำการคํานวณค่าอัตราเซลล์เฉลี่ยในช่วงเวลาท่อนหน้าของกราฟฟิคแต่ละแบบในช่วง  $N$  เบิรสต์ได้ตามสมการที่ (4.9)

$$\overline{MCR} = \frac{\sum_{i=1}^N E(X_i)}{N \times \sum_{i=1}^N (E(X_i)\Delta + E(S_i))} \quad (4.9)$$

สำหรับค่าเอาต์พุตที่ได้จากตัวควบคุมแบบพีซซีจะเป็นค่าในช่วง 0 ถึง 1 เราจะนำค่านี้ไปแปลงให้อยู่ในค่าช่วงที่เราเลือกใช้ในการปรับค่า QLT ซึ่งผลที่ได้จึงจะเป็นค่าของ QLT ในบัฟเฟอร์ของกราฟฟิคแบบไม่เป็นเวลาจริง ตามสมการที่ (4.10)

$$X' = a + (b - a)X \quad (4.10)$$

โดยที่  $X$  คือเอาต์พุตที่ได้จากตัวควบคุมแบบพีซซี ส่วน  $X'$  คือค่า QLT ที่นำมาใช้งานจริง และค่า  $a, b$  คือ ค่าขีดจำกัดล่างและขีดจำกัดบนที่ใช้ในการปรับระดับ QLT

### 4.3.3 ข้อกำหนดการทํางานของตัวควบคุมแบบพีซซี

#### ฟังก์ชันการเป็นสมาชิก

ใช้ฟังก์ชันการเป็นสมาชิกรูปสามเหลี่ยมกับรูปสี่เหลี่ยมคางหมู

### ค่า Linguistic Variable และ Linguistic Value

Term Set ที่เลือกใช้จะกำหนดเป็นปริมาณที่มีเครื่องหมายและขนาดต่างกัน เช่น Term Set ที่มี 4 คำคือ สูง, ค่อนข้างสูง, ค่อนข้างต่ำและต่ำ ขึ้นอยู่กับว่าต้องการรายละเอียดมากเพียงใดอาจทำการแบ่งค่าให้มากขึ้นได้

### Fuzzy Inference Engine

ใช้วิธีการทำงานของ Mamdani Inference System หรืออาจเรียกว่าวิธี Max-Min Inference System

### การกำหนดกฎในฐานกฎ

ทำการกำหนดกฎอยู่บนพื้นฐานความรู้ประสบการณ์ของผู้ปฏิบัติการ

### วิธีการ Defuzzification

ใช้วิธีจุดศูนย์กลางของพื้นที่ (Centroid of Area)

ในแบบจำลองการทำงานการปรับเปลี่ยนค่า QLT โดยใช้ตัวควบคุมแบบฟัซซี ค่าอินพุตที่ใช้เป็นค่าที่ว่างของบัพเฟอร์ ( $S$ ) มี 2 คำคือ คำที่ว่างในบัพเฟอร์ของกราฟฟิกแบบเวลาจริง ( $S_r$ ) และแบบไม่เป็นเวลาจริง ( $S_{nr}$ ) มีค่าฟัซซีเซต (Fuzzy Set) ดังแสดงในสมการที่ (4.11) และ (4.12)

$$T(S_r) = \{H_r, WH_r, WL_r, L_r\} \quad (4.11)$$

$$T(S_{nr}) = \{H_{nr}, WH_{nr}, WL_{nr}, L_{nr}\} \quad (4.12)$$

ค่าอินพุตที่เป็นค่าเฉลี่ยของอัตราเซลล์เฉลี่ยในช่วง  $i$  บิตส์ที่ผ่านมา (Average Mean Cell Rate,  $\overline{MCR}$ ) ของกราฟฟิกแบบเวลาจริง ( $\overline{MCR}_r$ ) และแบบไม่เป็นเวลาจริง ( $\overline{MCR}_{nr}$ ) มีค่าฟัซซีเซต (Fuzzy Set) ดังแสดงในสมการที่ (4.13) และ (4.14)

$$\overline{T(MCR_r)} = \{H_r, M_r, L_r\} \quad (4.13)$$

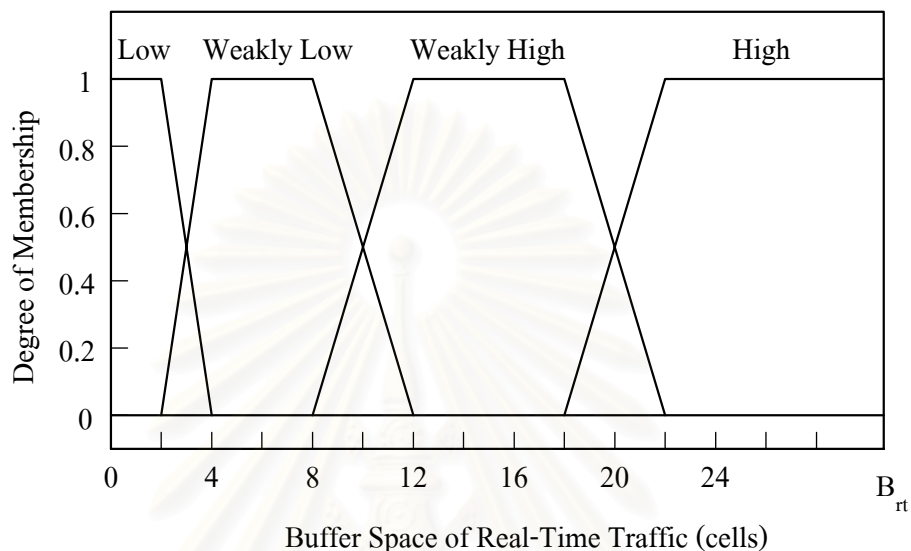
$$\overline{T(MCR_{nr})} = \{H_{nr}, M_{nr}, L_{nr}\} \quad (4.14)$$

ค่าเอาต์พุตที่ใช้หาค่าผลลัพธ์ในการปรับเปลี่ยนค่า QLT มีค่าฟัซซีเซต (Fuzzy Set) ดังแสดงในสมการที่ (4.15)

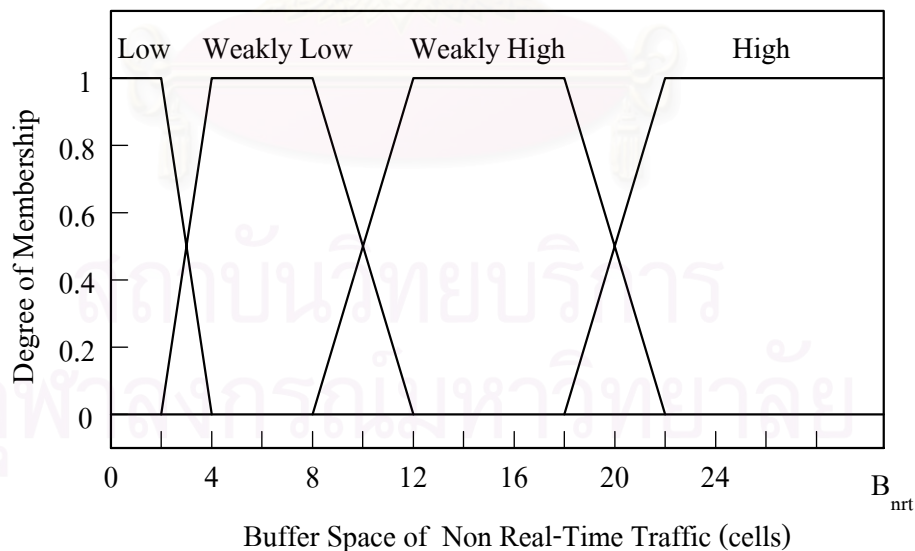
$$T(X) = \{HE, H, WH, M, WL, L, LE\} \quad (4.15)$$

(โดยที่  $HE, H, WH, M, WL, L$  และ  $LE$  หมายถึง Highest, High, Weakly High, Medium, Weakly Low, Low และ Lowest ตามลำดับ และมีค่าเรียงลำดับจากค่ามากไปน้อย)

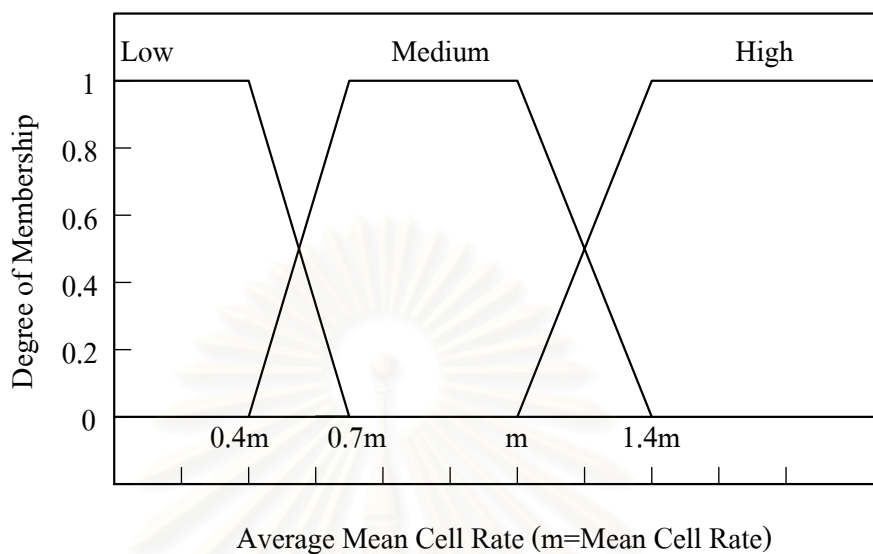
รูปที่ 4.5 และ 4.6 แสดง Input Membership Function สำหรับที่ว่างในบัฟเฟอร์ (Buffer Space) ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแบบไม่เป็นเวลาจริง ตามลำดับ รูปที่ 4.7 แสดง Input Membership Function สำหรับ Average Mean Cell Rate ของแหล่งกำเนิดกราฟฟิกทั้งแบบเวลาจริงและแบบไม่เป็นเวลาจริง ส่วนรูปที่ 4.8 แสดง Output Membership Function



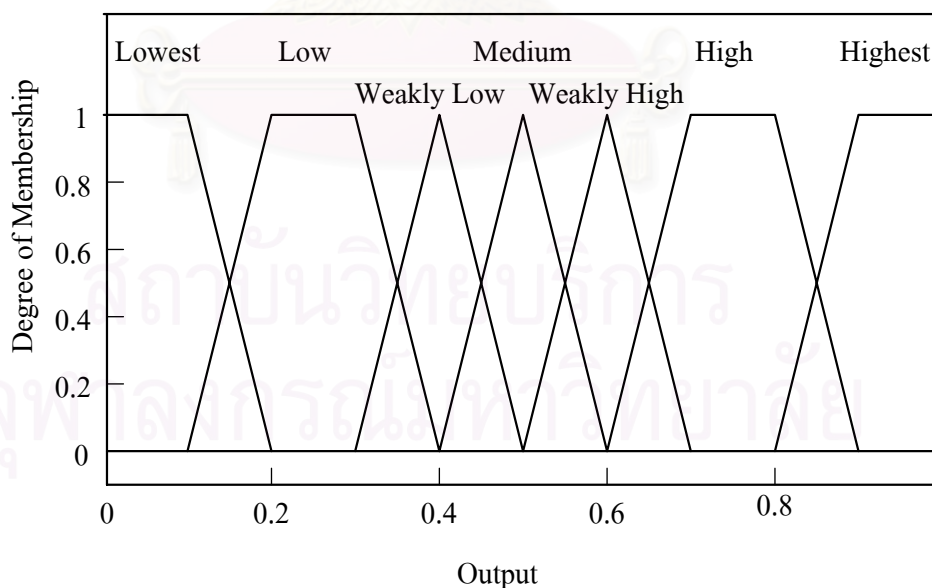
รูปที่ 4.5 Input Membership Function สำหรับที่ว่างในบัฟเฟอร์ (Buffer Space) ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง (Real-Time Traffic)



รูปที่ 4.6 Input Membership Function สำหรับที่ว่างในบัฟเฟอร์ (Buffer Space) ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง (Non Real-Time Traffic)



รูปที่ 4.7 Input Membership Function สำหรับ Average Mean Cell Rate ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง (Real-Time Traffic) และแบบไม่เป็นเวลาจริง (Non Real-Time Traffic)



รูปที่ 4.8 Output Membership Function

ตารางที่ 4.2 ฐานกฎ

<b>RULE</b>	<b>Buffer Space Real Time</b>	<b>Buffer Space Non-Real Time</b>	<b>Mean Cell Rate Non-Real Time</b>	<b>Mean Cell Rate Real Time</b>	<b>OUTPUT</b>
1	L	L	L	L	WH
2	L	L	L	M	WH
3	L	L	L	H	H
4	L	L	M	L	M
5	L	L	M	M	WH
6	L	L	M	H	WH
7	L	L	H	L	WL
8	L	L	H	M	M
9	L	L	H	H	M
10	L	WL	L	L	H
11	L	WL	L	M	H
12	L	WL	L	H	HE
13	L	WL	M	L	WH
14	L	WL	M	M	WH
15	L	WL	M	H	H
16	L	WL	H	L	M
17	L	WL	H	M	WH
18	L	WL	H	H	WH
19	L	WH	L	L	HE
20	L	WH	L	M	HE
21	L	WH	L	H	HE
22	L	WH	M	L	H
23	L	WH	M	M	H
24	L	WH	M	H	H
25	L	WH	H	L	WH
26	L	WH	H	M	H
27	L	WH	H	H	H
28	L	H	L	L	HE
29	L	H	L	M	HE
30	L	H	L	H	HE
31	L	H	M	L	HE

<b>RULE</b>	<b>Buffer Space Real Time</b>	<b>Buffer Space Non-Real Time</b>	<b>Mean Cell Rate Non-Real Time</b>	<b>Mean Cell Rate Real Time</b>	<b>OUTPUT</b>
32	L	H	M	M	HE
33	L	H	M	H	HE
34	L	H	H	L	H
35	L	H	H	M	HE
36	L	H	H	H	HE
37	WL	L	L	L	WL
38	WL	L	L	M	WL
39	WL	L	L	H	M
40	WL	L	M	L	L
41	WL	L	M	M	WL
42	WL	L	M	H	WL
43	WL	L	H	L	L
44	WL	L	H	M	L
45	WL	L	H	H	WL
46	WL	WL	L	L	M
47	WL	WL	L	M	M
48	WL	WL	L	H	WH
49	WL	WL	M	L	M
50	WL	WL	M	M	M
51	WL	WL	M	H	M
52	WL	WL	H	L	WL
53	WL	WL	H	M	M
54	WL	WL	H	H	M
55	WL	WH	L	L	WH
56	WL	WH	L	M	WH
57	WL	WH	L	H	H
58	WL	WH	M	L	WH
59	WL	WH	M	M	WH
60	WL	WH	M	H	WH
61	WL	WH	H	L	M
62	WL	WH	H	M	WH
63	WL	WH	H	H	WH

<b>RULE</b>	<b>Buffer Space Real Time</b>	<b>Buffer Space Non-Real Time</b>	<b>Mean Cell Rate Non-Real Time</b>	<b>Mean Cell Rate Real Time</b>	<b>OUTPUT</b>
64	WL	H	L	L	H
65	WL	H	L	M	H
66	WL	H	L	H	HE
67	WL	H	M	L	H
68	WL	H	M	M	H
69	WL	H	M	H	HE
70	WL	H	H	L	WH
71	WL	H	H	M	H
72	WL	H	H	H	H
73	WH	L	L	L	L
74	WH	L	L	M	L
75	WH	L	L	H	WL
76	WH	L	M	L	LE
77	WH	L	M	M	L
78	WH	L	M	H	L
79	WH	L	H	L	LE
80	WH	L	H	M	LE
81	WH	L	H	H	L
82	WH	WL	L	L	WL
83	WH	WL	L	M	WL
84	WH	WL	L	H	M
85	WH	WL	M	L	WL
86	WH	WL	M	M	WL
87	WH	WL	M	H	WL
88	WH	WL	H	L	L
89	WH	WL	H	M	WL
90	WH	WL	H	H	WL
91	WH	WH	L	L	M
92	WH	WH	L	M	M



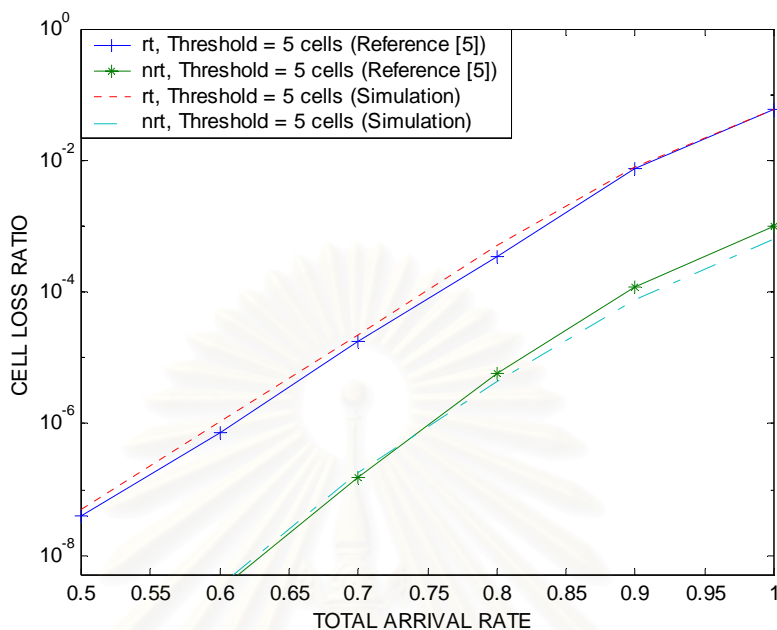
<b>RULE</b>	<b>Buffer Space Real Time</b>	<b>Buffer Space Non-Real Time</b>	<b>Mean Cell Rate Non-Real Time</b>	<b>Mean Cell Rate Real Time</b>	<b>OUTPUT</b>
93	WH	WH	L	H	WH
94	WH	WH	M	L	M
95	WH	WH	M	M	M
96	WH	WH	M	H	M
97	WH	WH	H	L	WL
98	WH	WH	H	M	M
99	WH	WH	H	H	M
100	WH	H	L	L	WH
101	WH	H	L	M	WH
102	WH	H	L	H	H
103	WH	H	M	L	WH
104	WH	H	M	M	WH
105	WH	H	M	H	WH
106	WH	H	H	L	M
107	WH	H	H	M	WH
108	WH	H	H	H	WH
109	H	L	L	L	LE
110	H	L	L	M	LE
111	H	L	L	H	LE
112	H	L	M	L	LE
113	H	L	M	M	LE
114	H	L	M	H	LE
115	H	L	H	L	LE
116	H	L	H	M	LE
117	H	L	H	H	LE
118	H	WL	L	L	L
119	H	WL	L	M	L
120	H	WL	L	H	WL
121	H	WL	M	L	LE

<b>RULE</b>	<b>Buffer Space Real Time</b>	<b>Buffer Space Non-Real Time</b>	<b>Mean Cell Rate Non-Real Time</b>	<b>Mean Cell Rate Real Time</b>	<b>OUTPUT</b>
122	H	WL	M	M	L
123	H	WL	M	H	L
124	H	WL	H	L	LE
125	H	WL	H	M	L
126	H	WL	H	H	L
127	H	WH	L	L	WL
128	H	WH	L	M	WL
129	H	WH	L	H	M
130	H	WH	M	L	WL
131	H	WH	M	M	WL
132	H	WH	M	H	WL
133	H	WH	H	L	L
134	H	WH	H	M	WL
135	H	WH	H	H	WL
136	H	H	L	L	M
137	H	H	L	M	M
138	H	H	L	H	WH
139	H	H	M	L	M
140	H	H	M	M	M
141	H	H	M	H	M
142	H	H	H	L	WL
143	H	H	H	M	M
144	H	H	H	H	M

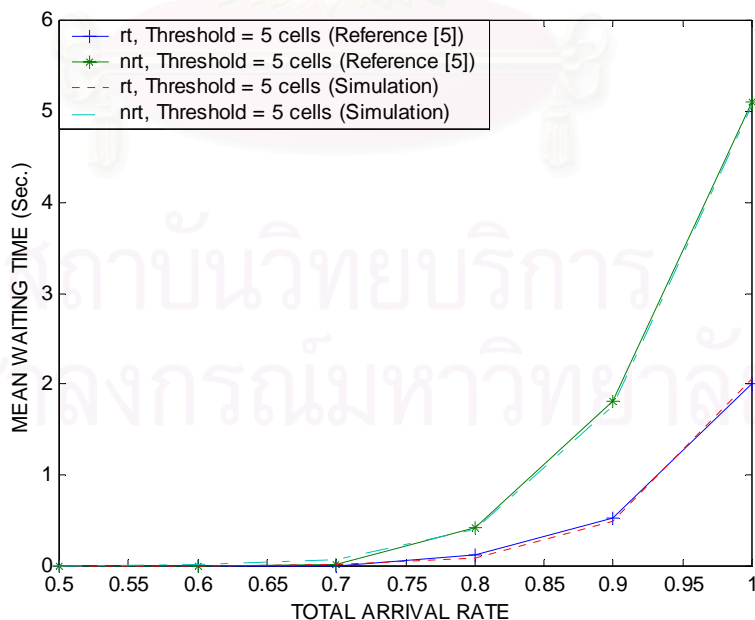
#### 4.4 การทดสอบความถูกต้องของการจำลองแบบ

ในการทดสอบความถูกต้องของการจำลองแบบการทำงานของอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญนั้นจะทำการทดสอบเปรียบเทียบผลที่ได้จากการจำลองแบบกับผลที่ได้จากบทความ [5] โดยในการจำลองการทำงานจะกำหนดค่าพารามิเตอร์ต่างๆให้ตรงกับที่ใช้ในบทความ [5] และใช้แบบจำลองแหล่งกำเนิดตามหัวข้อที่ 4.1.1 ซึ่งผลของการทดสอบจะพิจารณาถึงผลของอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ ได้ผลการทดสอบดังแสดงในรูปที่ 4.9-4.12

รูปที่ 4.9 และ 4.10 แสดงผลเปรียบเทียบอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ (ผลรวมของอัตราการมาถึงของเซลล์จากแหล่งกำเนิดกราฟฟิคทั้งสองแบบ) ค่าต่างๆตามลำดับ ในการจำลองแบบอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนไว้ที่ 50% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิคแบบไม่เป็นเวลาจริงที่มีขนาด 10 เซลล์ (ระดับจุดเริ่มเปลี่ยนอยู่ที่ 5 เซลล์) ผลที่ได้แสดงว่ามีความใกล้เคียงกับผลที่ได้จากบทความ [5] รูปที่ 4.11 และ 4.12 แสดงผลของอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ค่าต่างๆตามลำดับ ในการจำลองแบบอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนที่ 3, 5 และ 7 เซลล์ ซึ่งผลที่ได้มีความสอดคล้องกับผลที่ได้จากบทความ [5] คือ ผลของการตั้งค่า QLT ทำให้กราฟฟิคแบบเวลาจริงมีอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ค่าที่ค่า QLT สูงๆ ในขณะที่กราฟฟิคแบบไม่เป็นเวลาจริงมีผลดีในลักษณะเดียวกันเมื่อตั้งค่า QLT ที่ค่าต่ำๆ สรุปได้ว่าการจำลองแบบการทำงานของอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญมีความถูกต้อง เพราะให้ผลการทดสอบและแนวโน้มการทำงานที่ใกล้เคียงกับผลจากงานวิจัยที่อ้างอิง

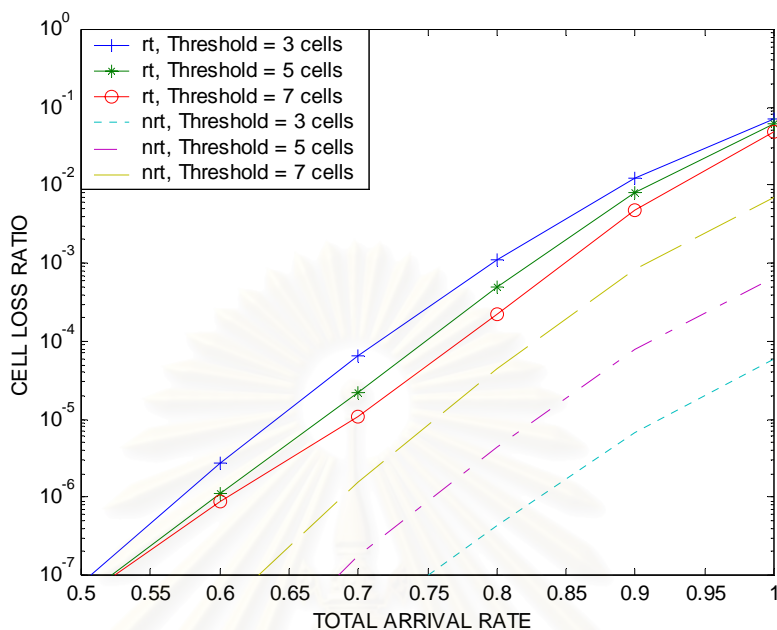


รูปที่ 4.9 ผลเปรียบเทียบอัตราส่วนการสูญหายของเซลล์ในการจำลองแบบอัลกอริทึมถึงรั้วที่มี การจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนที่ 5 เซลล์  
 rt = Real-Time Traffic, nrt = Non Real-Time Traffic



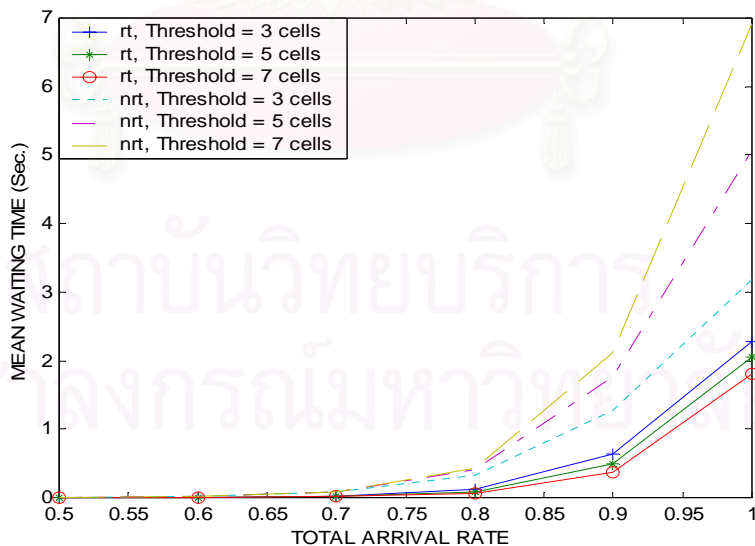
รูปที่ 4.10 ผลเปรียบเทียบเวลาการคอยโดยเฉลี่ยของเซลล์ในการจำลองแบบอัลกอริทึมถึงรั้วที่มี การจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนที่ 5 เซลล์

rt = Real-Time Traffic, nrt = Non Real-Time Traffic



รูปที่ 4.11 อัตราส่วนการสูญหายของเซลล์ในการจำลองแบบอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนที่ 3, 5 และ 7 เซลล์

rt = Real-Time Traffic, nrt = Non Real-Time Traffic



รูปที่ 4.12 เวลาการคอยโดยเฉลี่ยของเซลล์ในการจำลองแบบอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญ โดยตั้งระดับจุดเริ่มเปลี่ยนที่ 3, 5 และ 7 เซลล์

rt = Real-Time Traffic, nrt = Non Real-Time Traffic

## บทที่ 5

### ผลการทดสอบ

#### 5.1 ผลการทดสอบแบบจำลองการทำงานของอัลกอริทึมถึงรั้วชนิดต่างๆ

ในหัวข้อนี้จะแสดงผลการเปรียบเทียบการทำงานของอัลกอริทึมถึงรั้วชนิดต่างๆ ดังที่ได้กล่าวไปแล้วในบทที่ 3 ซึ่งประกอบไปด้วย Token Leaky Bucket (TLB), Priority Leaky Bucket (PLB), Priority Leaky Bucket ที่ใช้เทคนิค QLT (QLT) กับการทำงานของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ซึ่งมีข้อกำหนดของแต่ละอัลกอริทึมกำหนดไว้ในหัวข้อที่ 4.3 โดยอาศัยแบบจำลองพื้นฐานของอัลกอริทึมถึงรั้วที่มีการจัดลำดับความสำคัญมาปรับให้เป็นไปตามข้อกำหนดของอัลกอริทึมถึงรั้วชนิดต่างๆ กล่าวคือในอัลกอริทึม Token Leaky Bucket จะทำการแยกแหล่งกำเนิดทั้งสองชนิดออกจากกันและมีอัตราการสร้างโทเค็นกับขนาดของถังเก็บโทเค็นลดลงเหลืออย่างละครึ่งหนึ่ง สำหรับอัลกอริทึม Priority Leaky Bucket จะทำการตั้งระดับ QLT ไว้ที่ค่าสูงที่สุดคือที่ 100% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง เนื่องจากเป็นกรณีที่ให้ลำดับความสำคัญกับแหล่งกำเนิดกราฟฟิกแบบเวลาจริง ส่วนอัลกอริทึม Priority Leaky Bucket ที่ใช้เทคนิค QLT จะทำการทดสอบการทำงานโดยการตั้งระดับ QLT ที่ค่าต่างๆกันคือที่ระดับ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง และอัลกอริทึม Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT จะทำการปรับค่าของ QLT ในช่วง 40-90% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง โดยที่ใช้ค่าอัตราเซลล์เฉลี่ยในช่วงเวลาก่อนหน้าของกราฟฟิกแต่ละแบบในช่วง 5 เบิรสต์ (N = 5) ทำการทดสอบที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้ว (Normalized Total Arrival Rate) ค่าต่างๆ ซึ่งได้จากการเปลี่ยนแปลงอัตราการมาถึงโดยรวมของเซลล์ในแหล่งกำเนิดกราฟฟิกทั้งสองแบบ ตั้งแต่ 0.5 - 1 เท่าของอัตราการมาถึงโดยรวมของเซลล์ปกติในแหล่งกำเนิดกราฟฟิกทั้งสองแบบที่ได้ตกลงไว้กับ โครงข่าย (22.4 kbps)

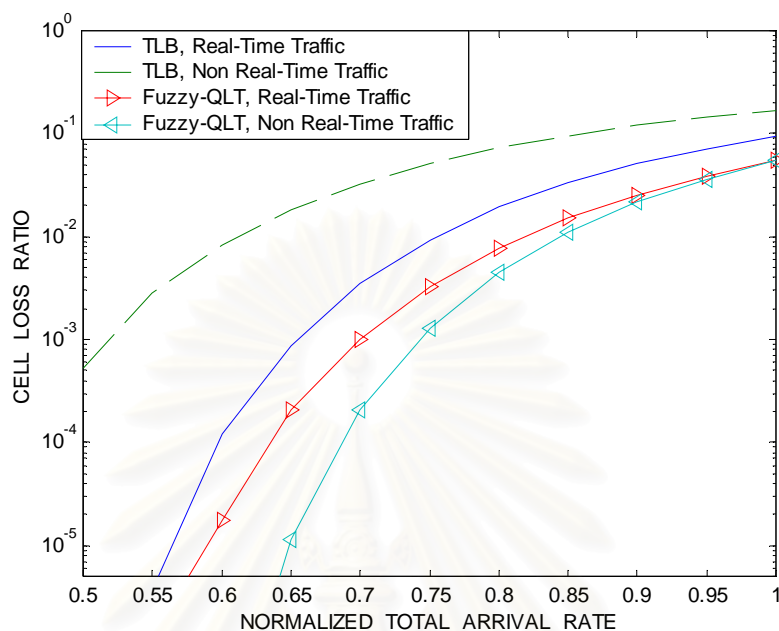
ผลการทดสอบแสดงดังรูปที่ 5.1 ถึง 5.6 โดยรูปที่ 5.1 และ 5.2 แสดงการเปรียบเทียบอัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio หรือ CLR) และเวลาการคอยโดยเฉลี่ยของเซลล์ (Mean Waiting Time) ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรั้วแบบ Token Leaky Bucket กับ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ตามลำดับ รูปที่ 5.3 และ 5.4 แสดงการเปรียบเทียบอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรั้วแบบ Priority Leaky Bucket กับ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ตามลำดับ

และรูปที่ 5.5 และ 5.6 แสดงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอ-ริทึมถึงร้วแบบ Priority Leaky Bucket ที่ใช้เทคนิค QLT โดยกำหนดระดับ QLT คงที่เท่ากับ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงกับ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ตามลำดับ ตารางที่ 5.1 และ 5.2 แสดงการเปรียบเทียบอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่ นอร์มอลไลซ์แล้วค่าต่างๆของอัลกอริทึมถึงร้วแบบ Token Leaky Bucket, Priority Leaky Bucket, Priority Leaky Bucket ที่ใช้เทคนิค QLT ที่ระดับ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงและ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ตามลำดับ

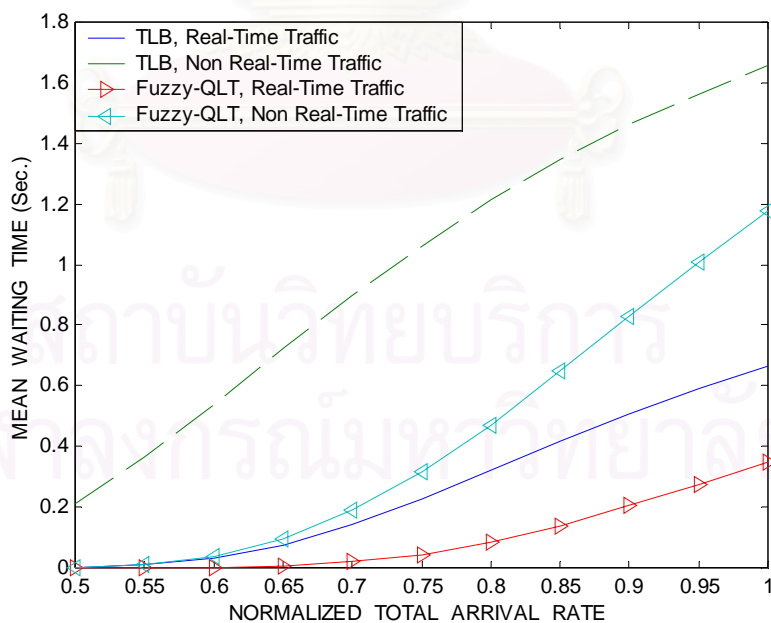


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

- ผลการจำลองแบบการทำงานของอัลกอริทึมถึงร้วแบบ Token Leaky Bucket (TLB), Priority Leaky Bucket (PLB), Priority Leaky Bucket ที่ใช้เทคนิค QLT (QLT) และ Fuzzy-QLT

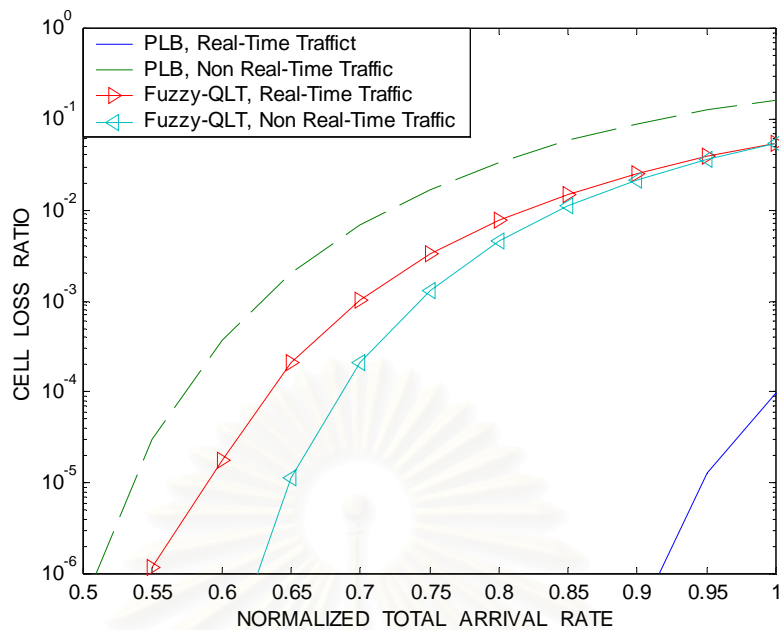


รูปที่ 5.1 เปรียบเทียบอัตราส่วนการสูญหายของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงร้วแบบ TLB กับ Fuzzy-QLT

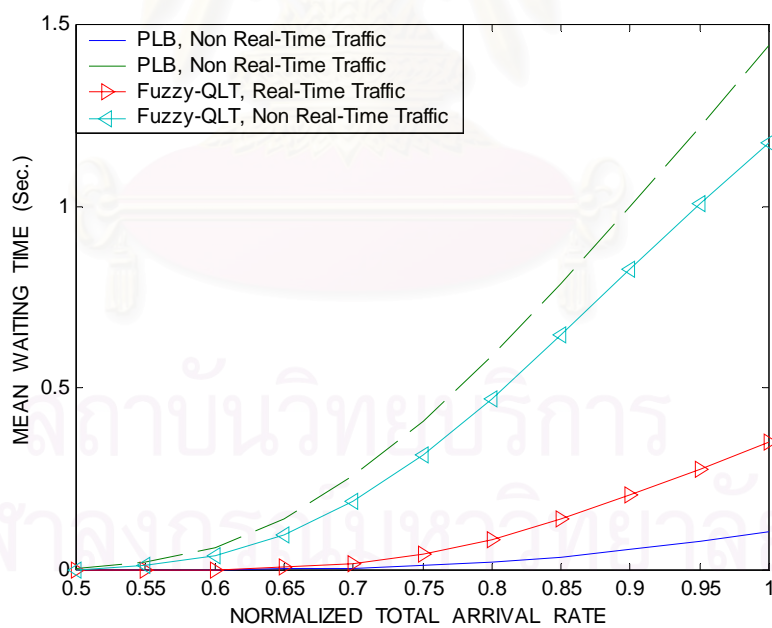


รูปที่ 5.2 เปรียบเทียบเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงร้วแบบ TLB กับ Fuzzy-QLT

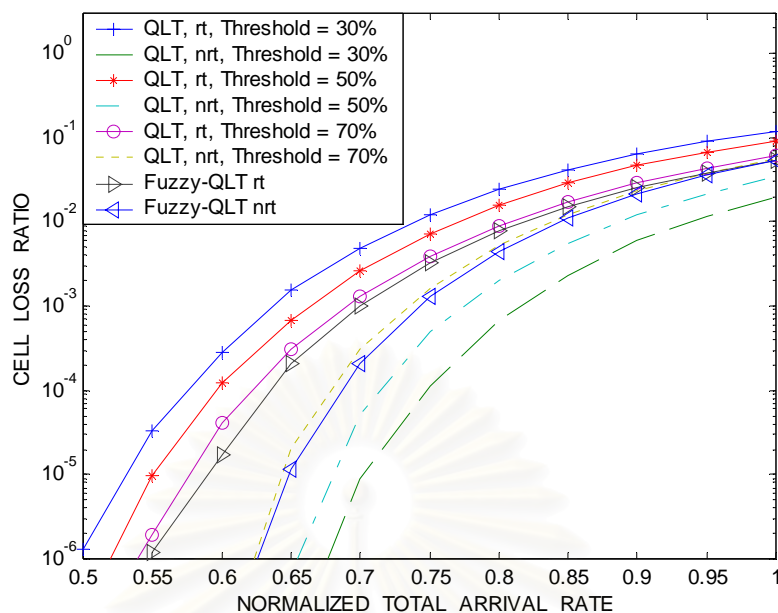




รูปที่ 5.3 เปรียบเทียบอัตราส่วนการสูญหายของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรูปแบบ PLB กับ Fuzzy-QLT

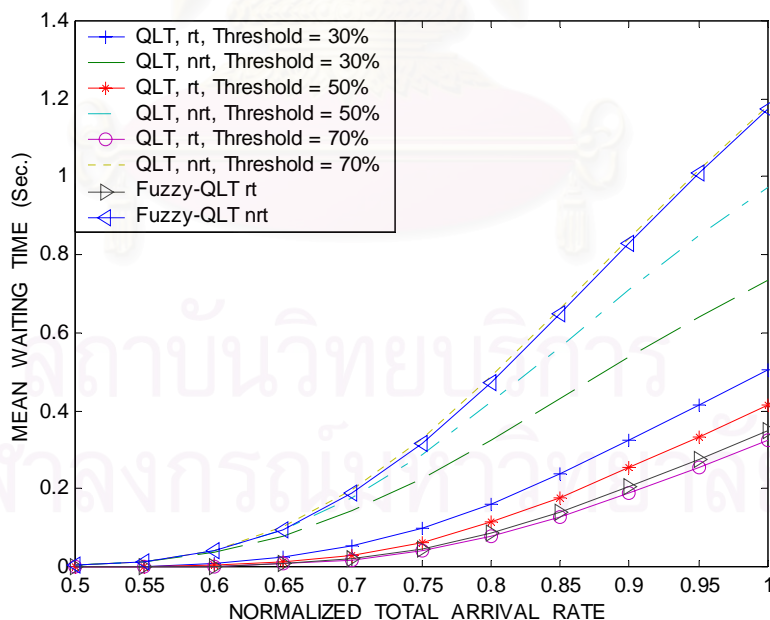


รูปที่ 5.4 เปรียบเทียบเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรูปแบบ PLB กับ Fuzzy-QLT



รูปที่ 5.5 เปรียบเทียบอัตราส่วนการสูญหายของเซลล์ ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงระบบ QLT โดยกำหนดระดับ QLT คงที่เท่ากับ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงกับ Fuzzy-QLT

rt = Real-Time Traffic, nrt = Non Real-Time Traffic



รูปที่ 5.6 เปรียบเทียบเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงระบบ QLT โดยกำหนดระดับ QLT เท่ากับ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงกับ Fuzzy-QLT

rt = Real-Time Traffic, nrt = Non Real-Time Traffic

ตารางที่ 5.1 เปรียบเทียบอัตราส่วนการสูญหายของเซลล์ ที่อัตราการมาถึงโดยรวมของเซลล์ที่  
นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรัวชนิดต่างๆ

- หมายถึง อัตราส่วนการสูญหายของเซลล์มีค่าเข้าใกล้ 0, rt = Real-Time, nrt = Non Real-Time

Algorithm	TLB		PLB		QLT		Fuzzy-QLT	
	rt	nrt	rt	nrt	rt	nrt	rt	nrt
Normalized Total Arrival Rate								
0.50	-	5.2941e-4	-	4.3929e-7	7.2506e-8	-	-	-
0.55	3.6371e-6	2.7885e-3	-	3.0379e-5	1.9496e-6	-	1.1983e-6	-
0.60	1.2195e-4	8.1673e-3	-	3.7642e-4	4.0551e-5	7.576e-8	1.7512e-5	7.577e-8
0.65	8.6605e-4	0.018087	-	2.0351e-3	3.0824e-4	2.0384e-5	2.0824e-4	1.1340e-5
0.70	0.003468	0.032254	-	6.9152e-3	1.3060e-3	3.1149e-4	1.0147e-3	2.0970e-4
0.75	0.009266	0.050616	-	0.016923	3.9130e-3	1.6144e-3	3.2615e-3	1.3010e-3
0.80	0.019243	0.072099	-	0.033754	9.0112e-3	5.1852e-3	7.7446e-3	4.4990e-3
0.85	0.033377	0.095308	4.5557e-9	0.057706	0.017264	0.012110	0.014929	0.010956
0.90	0.050888	0.120128	3.0041e-7	0.088163	0.028784	0.023134	0.02534	0.021482
0.95	0.070963	0.144795	1.2871e-5	0.124230	0.043150	0.037912	0.03859	0.036167
1.00	0.092553	0.169616	9.6312e-5	0.164028	0.060050	0.056414	0.054494	0.054570

ตารางที่ 5.2 เปรียบเทียบเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่  
นอร์มอลไลซ์แล้วค่าต่างๆ ของอัลกอริทึมถึงรัวชนิดต่างๆ

Algorithm	TLB		PLB		QLT		Fuzzy-QLT	
	rt	nrt	rt	nrt	rt	nrt	rt	nrt
Normalized Total Arrival Rate								
0.50	1.0318e-3	0.21242	9.940e-5	0.004542	5.7023e-5	2.3984e-3	5.2588e-5	2.1287e-3
0.55	8.1900e-3	0.36339	4.1671e-4	0.02151	3.4790e-4	0.013058	0.00035	0.012104
0.60	0.03114	0.53771	1.1064e-3	0.063757	1.6358e-3	0.042471	0.00167	0.03952
0.65	0.07637	0.72215	2.3434e-3	0.14218	6.1672e-3	0.10173	0.006626	0.096131
0.70	0.14464	0.89636	4.4751e-3	0.25906	0.017456	0.19646	0.019039	0.18743
0.75	0.22924	1.06291	0.01112	0.41056	0.040071	0.32717	0.044301	0.31605
0.80	0.32363	1.21172	0.02201	0.59001	0.077492	0.48701	0.084906	0.47150
0.85	0.41850	1.34435	0.03696	0.78818	0.12788	0.6628	0.13963	0.64747
0.90	0.50765	1.46085	0.05628	0.99969	0.18830	0.84368	0.20516	0.82892
0.95	0.59011	1.56419	0.07929	1.21793	0.25451	1.0192	0.27670	1.0083
1.00	0.66335	1.65497	0.10540	1.44082	0.32276	1.18197	0.35066	1.17549

## 5.2 วิเคราะห์ผลการทดสอบแบบจำลองการทำงานของอัลกอริทึมถังรั่วชนิดต่างๆ

ผลการจำลองแบบการทำงานของ Token Leaky Bucket (TLB) จากรูปที่ 5.1 และ 5.2 พิจารณาการทำงานของแหล่งกำเนิดทั้งสองแบบพบว่า แหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงมีผลของอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆสูงกว่าของแหล่งกำเนิดทราฟฟิกแบบเวลาจริง ทั้งที่บัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงมีขนาดใหญ่กว่าบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบเวลาจริง เนื่องจากอัตราการเบิรสต์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงสูงกว่าอัตราการเบิรสต์ของแหล่งกำเนิดทราฟฟิกแบบเวลาจริง ทำให้อัตราการสร้างโทเค้นไม่เพียงพอต่อเซลล์ที่เข้ามา และขนาดของบัฟเฟอร์ที่ใหญ่กว่าก็ทำให้ค่าเวลาการคอยโดยเฉลี่ยของเซลล์สูงกว่าด้วย เมื่อเปรียบเทียบผลการจำลองแบบของ TLB กับ Fuzzy-QLT พบว่าวิธี Fuzzy-QLT สามารถลดอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดทั้งสองแบบลงได้ เพราะแหล่งกำเนิดทราฟฟิกแต่ละแบบในวิธี Fuzzy-QLT สามารถแบ่งโทเค้นที่สร้างมาให้กับแหล่งกำเนิดทราฟฟิกแบบอื่นได้เนื่องจากมีการใช้ถังเก็บโทเค้นร่วมกัน และอัตราการกำเนิดโทเค้นก็ได้มาจากอัตราการกำเนิดโทเค้นของแหล่งกำเนิดทราฟฟิกทั้งสองแบบรวมกัน

สำหรับผลการจำลองแบบการทำงานของ Priority Leaky Bucket (PLB) ซึ่งเหมือนกับเป็นการตั้งค่าระดับ QLT ที่ค่าสูงที่สุด (100 %) ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง จากรูปที่ 5.3 และ 5.4 พบว่าค่าที่ได้ให้ผลดีที่สุดกับแหล่งกำเนิดทราฟฟิกแบบเวลาจริง ส่วนแหล่งกำเนิดแบบไม่เป็นเวลาจริงให้ผลที่ต่ำที่สุดทั้งในด้านอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ และเมื่อเปรียบเทียบผลการจำลองแบบของ PLB กับ Fuzzy-QLT พบว่าวิธี Fuzzy-QLT สามารถลดอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง ส่งผลให้อัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดทราฟฟิกแบบเวลาจริงมีค่าสูงขึ้น

เมื่อพิจารณาในด้านคุณภาพการบริการของแหล่งกำเนิดทราฟฟิกแต่ละแบบพบว่าวิธี TLB และ PLB ไม่สามารถรองรับความต้องการของแหล่งกำเนิดทราฟฟิกแบบเวลาจริงและไม่เป็นเวลาจริงได้ ส่วนวิธี Fuzzy-QLT สามารถลดอัตราส่วนการสูญหายของเซลล์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงลงได้ต่ำกว่าของแหล่งกำเนิดทราฟฟิกแบบเวลาจริง ส่วนเวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดทราฟฟิกแบบเวลาจริงจะมีค่าต่ำ ซึ่งตรงกับความต้องการของคุณภาพการบริการของทราฟฟิกทั้งสองแบบ

ผลจากการทำงานของ Priority Leaky Bucket ที่ใช้เทคนิค QLT (QLT) ดังแสดงในรูปที่ 5.5 และ 5.6 ซึ่งทำการทดสอบการทำงานโดยการตั้งระดับ QLT ที่ค่าต่างๆกันคือ ที่ระดับ 30%, 50%

และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง จะได้ผลที่มีแนวโน้มสอดคล้องตามบทความ [5] กล่าวคือ ที่ระดับ QLT เท่ากับ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง จะทำให้แหล่งกำเนิดทราฟฟิกแบบเวลาจริงมีค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ต่ำกว่าที่ระดับ QLT เท่ากับ 50% และ 30% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริง ส่วนแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงจะได้ผลในลักษณะตรงกันข้าม เมื่อเปรียบเทียบวิธี Priority Leaky Bucket ที่ใช้เทคนิค QLT ที่ตั้งระดับ QLT ที่ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงกับวิธี Fuzzy-QLT พบว่าวิธี Fuzzy-QLT ให้ค่าอัตราส่วนการสูญหายของเซลล์ของทราฟฟิกทั้งสองชนิดลดลง โดยที่เวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดทราฟฟิกแบบเวลาจริงมีค่าเพิ่มขึ้นเล็กน้อย



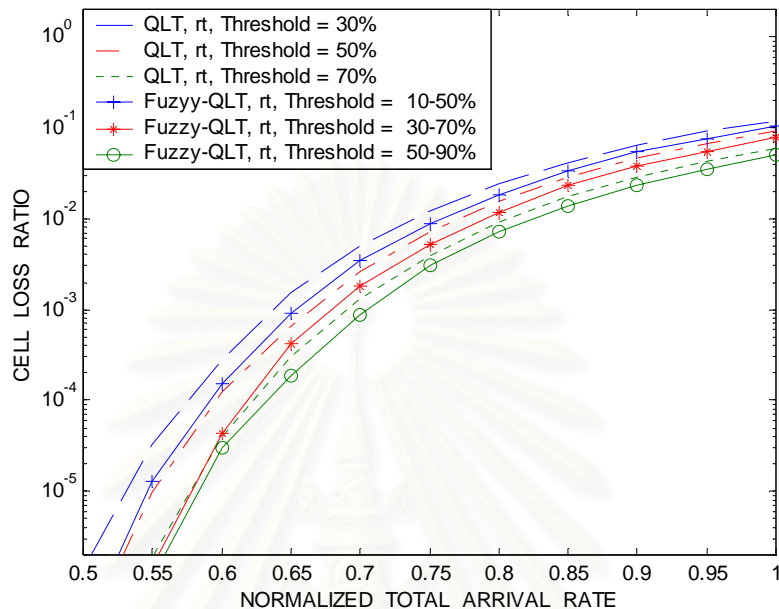
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### 5.3 ผลการทดสอบแบบจำลองการทำงานของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT (วิธีที่เสนอ)

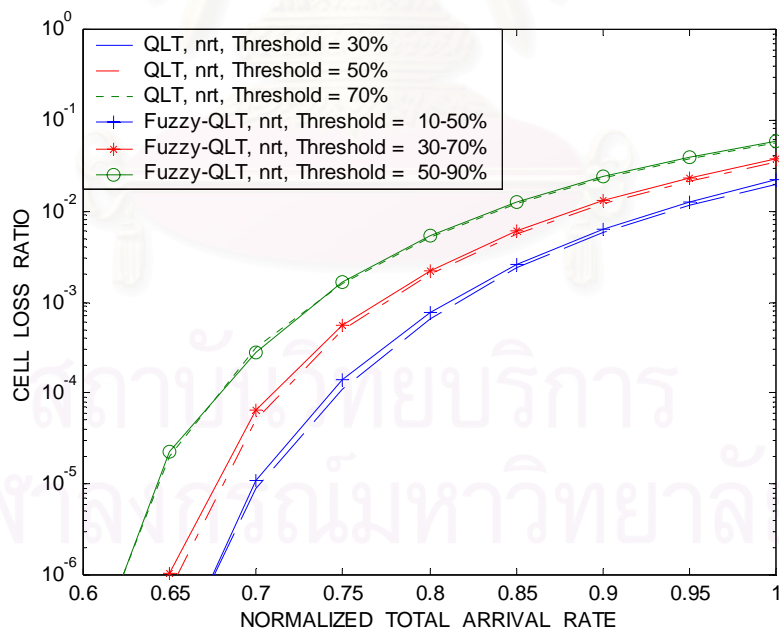
#### 5.3.1 ผลการทดสอบการปรับค่า QLT ในช่วงต่างๆ

ในส่วนนี้จะศึกษาผลของอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ โดยใช้ตัวควบคุมแบบฟัซซีทำการปรับระดับ QLT ในช่วงต่างๆ คือ 10-50%, 30-70% และ 70-90% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงเพื่อเปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่ที่ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง ตามลำดับ ผลการทดสอบดังแสดงในรูปที่ 5.7 และ 5.8 ซึ่งแสดงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ ตารางที่ 5.3 ถึง 5.8 แสดงผลของการปรับปรุงประสิทธิภาพการทำงานของ Fuzzy-QLT ในด้านอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ โดยการคำนวณหาค่าเปอร์เซ็นต์ของค่าต่างๆ ในตารางที่ 5.3 ถึง 5.8 คำนวณจากผลต่างของค่าที่ได้จากกรณีที่ไม่ใช้ตัวควบคุมแบบฟัซซีและค่าที่ได้จากกรณีที่ใช้ตัวควบคุมแบบฟัซซีที่อัตราการมาถึงโดยรวมของเซลล์ค่าหนึ่ง แล้วนำค่าผลต่างนั้นหารด้วยค่าที่ได้จากกรณีที่ไม่ใช้ตัวควบคุมแบบฟัซซีและเทียบให้เป็นเปอร์เซ็นต์ ในส่วนของค่าเฉลี่ยของเปอร์เซ็นต์หาได้จากการคำนวณค่าเฉลี่ยของทุกๆ อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้ว

- ผลการจำลองแบบการทำงานของอัลกอริทึมถังรั่วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT โดยปรับระดับ QLT ในช่วงต่างๆเปรียบเทียบกับกรณีตั้งระดับ QLT คงที่ที่ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงตามลำดับ



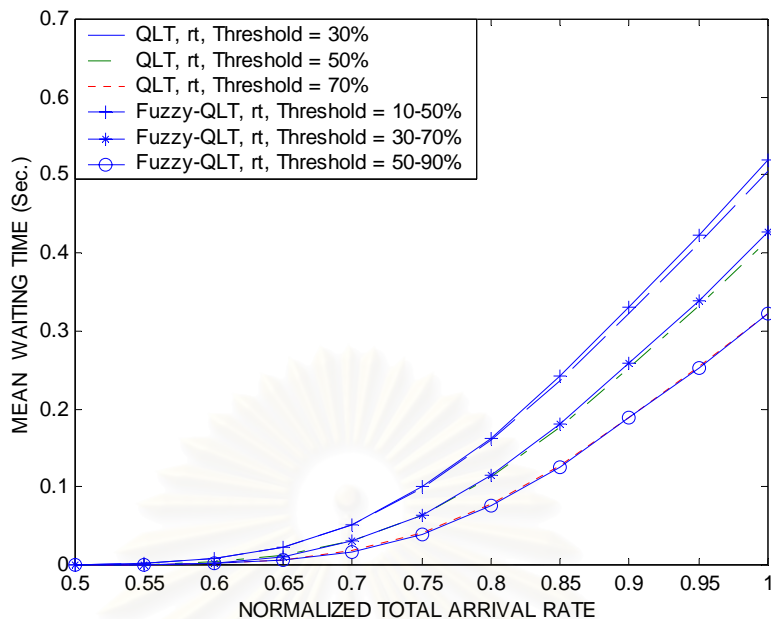
(ก) แหล่งกำเนิดกราฟฟิกแบบเวลาจริง



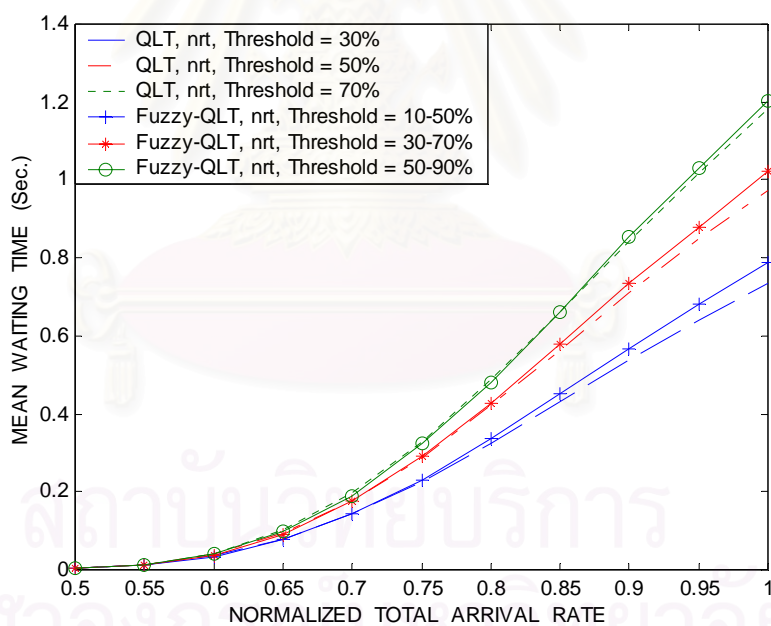
(ข) แหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

รูปที่ 5.7 อัตราส่วนการสูญหายของเซลล์ในอัลกอริทึมถังรั่วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อปรับระดับ QLT ในช่วงต่างๆ เปรียบเทียบกับการ ตั้งระดับ QLT คงที่

rt = Real-Time Traffic, nrt = Non Real-Time Traffic



(ก) แหล่งกำเนิดกราฟฟิกแบบเวลาจริง



(ข) แหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

รูปที่ 5.8 เวลาการคอยโดยเฉลี่ยของเซลล์ในอัลกอริทึมถังรั่วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อปรับระดับ QLT ในช่วงต่างๆ เปรียบเทียบกับการตั้งระดับ QLT คงที่

rt = Real-Time Traffic, nrt = Non Real-Time Traffic



ตารางที่ 5.3 เปอร์เซ็นต์การลดลงของอัตราส่วนการสูญหายของเซลล์ เมื่อใช้ตัวควบคุมแบบพีซีซีปรับระดับ QLT ในช่วง 10-50% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่ที่ 30% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

NORMALIZED TOTAL ARRIVAL RATE	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Real-Time Traffic)	% Improvement (Real-Time Traffic)	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	1.2949e-004	46.079	-	-
0.65	6.2613e-004	40.698	-1.2000e-009	-1.3711
0.70	1.5106e-003	30.837	-1.8998e-006	-21.1290
0.75	3.1668e-003	26.325	-2.8101e-005	-25.0680
0.80	5.5412e-003	23.113	-9.4442e-005	-14.270
0.85	7.6797e-003	18.500	-2.9882e-004	-12.7420
0.90	1.0682e-002	16.697	-4.4130e-004	-7.4725
0.95	1.3704e-002	15.178	-1.0240e-003	-8.7794
1.00	1.4131e-002	11.901	-1.8105e-003	-9.0567
	ค่าเฉลี่ย	25.480	ค่าเฉลี่ย	-13.7360

ตารางที่ 5.4 เปอร์เซ็นต์การลดลงของค่าเวลาการคอยโดยเฉลี่ยของเซลล์ เมื่อใช้ตัวควบคุมแบบพีซีซีปรับระดับ QLT ในช่วง 10-50% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่ที่ 30% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

NORMALIZED TOTAL ARRIVAL RATE	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Real-Time Traffic)	% Improvement (Real-Time Traffic)	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	3.0537e-004	4.0603	1.7319e-003	4.9577
0.65	6.3920e-004	2.7658	1.7343e-003	2.2164
0.70	-2.3250e-004	-0.44617	-7.8000e-004	-0.5491
0.75	-1.2174e-003	-1.2379	-4.9230e-003	-2.1722
0.80	-2.1350e-003	-1.3288	-1.0630e-002	-3.2793
0.85	-5.3460e-003	-2.2572	-1.9859e-002	-4.6128
0.90	-8.3940e-003	-2.6004	-3.0011e-002	-5.5756
0.95	-8.9780e-003	-2.1717	-3.9263e-002	-6.1221
1.00	-1.5610e-002	-3.0934	-5.1459e-002	-6.9846
	ค่าเฉลี่ย	-0.7010	ค่าเฉลี่ย	-2.4579

ตารางที่ 5.5 เปอร์เซ็นต์การลดลงของอัตราส่วนการสูญหายของเซลล์ เมื่อใช้ตัวควบคุมแบบพีซีซีเปลี่ยนแปลงระดับ QLT ในช่วง 30%-70% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 50% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

NORMALIZED TOTAL ARRIVAL RATE	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Real-Time Traffic)	% Improvement (Real-Time Traffic)	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	7.8250e-005	64.004	-	-
0.65	2.5407e-004	38.022	-3.6573e-007	-54.059
0.70	8.0560e-004	30.856	-1.1931e-005	-23.014
0.75	2.0690e-003	28.387	-7.0896e-005	-14.482
0.80	4.0155e-003	25.542	-1.7300e-004	-8.5658
0.85	5.9318e-003	20.654	-5.8125e-004	-10.3960
0.90	8.8781e-003	19.249	-9.3570e-004	-7.7917
0.95	1.2094e-002	18.016	-1.8459e-003	-8.6051
1.00	1.3862e-002	15.212	-3.1426e-003	-9.2263
	ค่าเฉลี่ย	28.8824	ค่าเฉลี่ย	-17.017

ตารางที่ 5.6 เปอร์เซ็นต์การลดลงของค่าเวลาการคอยโดยเฉลี่ยของเซลล์เมื่อใช้ตัวควบคุมแบบพีซีซีเปลี่ยนแปลงระดับ QLT ในช่วง 30%-70% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 50% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

NORMALIZED TOTAL ARRIVAL RATE	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Real-Time Traffic)	% Improvement (Real-Time Traffic)	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	2.8023e-004	8.5454	2.6470e-003	6.5209
0.65	3.1930e-004	2.7919	3.3521e-003	3.5651
0.70	-1.4530e-004	-0.4864	1.9390e-003	1.0964
0.75	-6.9680e-004	-1.1044	-1.8150e-003	-0.6295
0.80	-5.3700e-004	-0.4735	-4.2640e-003	-1.0124
0.85	-3.4600e-003	-1.9491	-1.3692e-002	-2.4272
0.90	-5.3840e-003	-2.1314	-2.3706e-002	-3.3408
0.95	-5.2190e-003	-1.5663	-3.2543e-002	-3.8378
1.00	-1.0546e-002	-2.5350	-4.7426e-002	-4.8644
	ค่าเฉลี่ย	0.1212	ค่าเฉลี่ย	-0.5477

ตารางที่ 5.7 เปอร์เซ็นต์การลดลงของอัตราส่วนการสูญหายของเซลล์ เมื่อใช้ตัวควบคุมแบบพีชซีเปลี่ยนแปลงระดับ QLT ในช่วง 50%-90% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

NORMALIZED TOTAL ARRIVAL RATE	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Real-Time Traffic)	% Improvement (Real-Time Traffic)	อัตราส่วนการสูญหายของเซลล์ที่ลดลง (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	1.0111e-005	24.934	-9.9000e-011	-0.1307
0.65	1.2265e-004	39.790	-2.1459e-006	-10.5270
0.70	4.3597e-004	33.381	3.3899e-005	10.8830
0.75	7.9853e-004	20.407	-4.5220e-005	-2.8010
0.80	1.8876e-003	20.948	-2.1946e-004	-4.2324
0.85	3.5023e-003	20.286	-3.7100e-004	-3.0634
0.90	5.1488e-003	17.887	-1.1788e-003	-5.0954
0.95	7.7355e-003	17.927	-1.8187e-003	-4.7971
1.00	1.0203e-002	16.991	-3.1485e-003	-5.5810
	ค่าเฉลี่ย	23.6167	ค่าเฉลี่ย	-2.8161

ตารางที่ 5.8 เปอร์เซ็นต์การลดลงของเวลาการคอยโดยเฉลี่ยของเซลล์ เมื่อใช้ตัวควบคุมแบบพีชซีเปลี่ยนแปลงระดับ QLT ในช่วง 50%-90% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาดบัพเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง

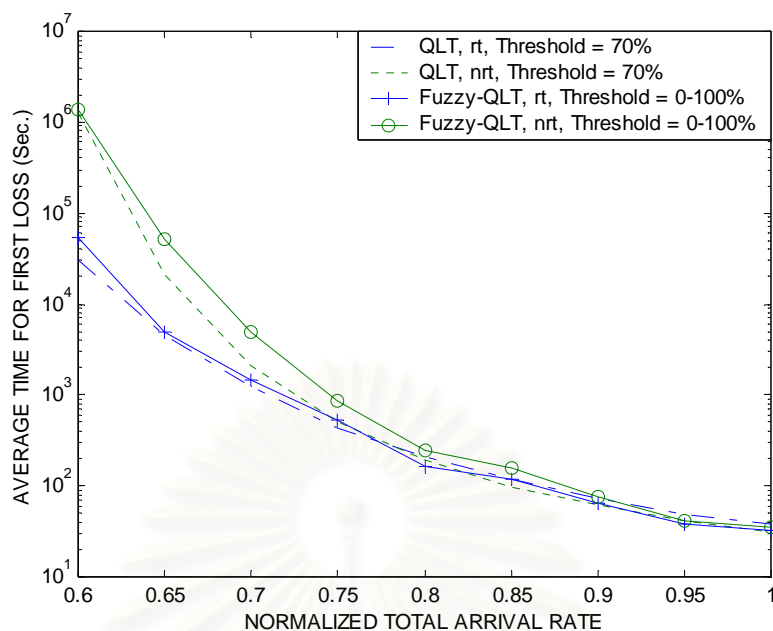
NORMALIZED TOTAL ARRIVAL RATE	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Real-Time Traffic)	% Improvement (Real-Time Traffic)	เวลาการคอยโดยเฉลี่ยของเซลล์ที่ลดลง (Sec.) (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	4.5730e-005	2.7955	1.7853e-003	4.2035
0.65	3.6023e-004	5.8410	4.6219e-003	4.5431
0.70	1.1003e-003	6.3030	7.7610e-003	3.9503
0.75	3.7840e-004	0.9443	2.8020e-003	0.8564
0.80	9.1930e-004	1.1863	4.7450e-003	0.9743
0.85	1.6920e-003	1.3231	2.4870e-003	0.3752
0.90	-4.7200e-004	-0.2507	-8.4630e-003	-1.0031
0.95	1.2940e-003	0.5084	-9.9100e-003	-0.9723
1.00	1.3350e-003	0.4136	-2.1440e-002	-1.8139
	ค่าเฉลี่ย	2.1182	ค่าเฉลี่ย	1.2348

### 5.3.2 ผลการทดสอบหาค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์

ในส่วนนี้จะทำการทดสอบหาค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์เซลล์แรก (Average Time for the First Loss) โดยใช้ตัวควบคุมแบบพีชซีทำการปรับระดับ QLT ในช่วง 0-100% ของขนาดบัพเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงเพื่อเปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่ที่ 70% ของขนาดบัพเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ผลการทดสอบดังแสดงในรูปที่ 5.9 และ 5.10 ซึ่งแสดงค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ที่อัตราการส่งเซลล์เฉลี่ยที่ต่ำกว่าข้อตกลงและที่ละเมิดข้อตกลงตามลำดับตารางที่ 5.9 และ 5.10 แสดงผลของเปอร์เซ็นต์การเพิ่มขึ้นและลดลงของค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ที่อัตราการส่งเซลล์เฉลี่ยที่ต่ำกว่าข้อตกลงและที่ละเมิดข้อตกลงตามลำดับ



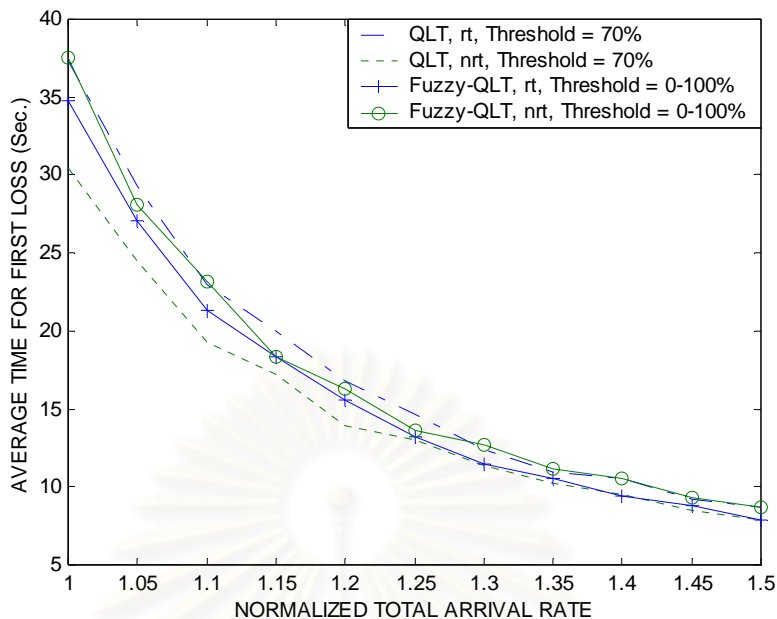
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.9 เปรียบเทียบค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหายที่อัตราการมาถึงโดยรวมของเซลล์ที่  
นอร์มอลไลซ์แล้วค่าต่างๆ ของ Fuzzy-QLT กับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาด  
บัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงเมื่ออัตราการส่งเซลล์ต่ำกว่าขีดตกลง

ตารางที่ 5.9 เปรี่เซนต์การเพิ่มขึ้นของค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหาย เมื่อใช้ตัวควบคุม  
แบบฟuzzyเปลี่ยนแปลงระดับ QLT เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาด  
บัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงเมื่ออัตราการส่งเซลล์ต่ำกว่าขีดตกลง

NORMALIZED TOTAL ARRIVAL RATE	เวลาเริ่มตรวจพบเซลล์ สูญหายที่เพิ่มขึ้น (Sec.) (Real-Time Traffic)	% Improvement (Real-Time Traffic)	เวลาเริ่มตรวจพบเซลล์ สูญหาย ที่เพิ่มขึ้น (Sec.) (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
0.60	24542	81.043	1.1907e+005	9.3208
0.65	251.66	5.4815	30073	140.41
0.70	256.38	21.067	2814.5	133.14
0.75	95.221	22.343	352.32	68.568
0.80	-43.789	-20.98	51.669	26.907
0.85	0.489	0.41628	60.878	62.521
0.90	-7.5675	-10.575	14.074	22.402
0.95	-10.47	-21.672	-0.7868	-1.8695
1.00	-5.9283	-15.422	4.1799	13.211
	ค่าเฉลี่ย	6.8557	ค่าเฉลี่ย	52.7344



รูปที่ 5.10 เปรียบเทียบค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหายที่อัตราการมาถึงโดยรวมของเซลล์ที่  
 นอร์มอลไลซ์แล้วค่าต่างๆ ของ Fuzzy-QLT กับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาด  
 บัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงเมื่ออัตราการส่งเซลล์ละเมิดข้อตกลง

ตารางที่ 5.10 เปรี่เซนต์การลดลงของค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหาย เมื่อใช้ตัวควบคุม  
 แบบฟัซซีเปลี่ยนแปลงระดับ QLT เปรียบเทียบกับกรณีกำหนดระดับ QLT คงที่เป็น 70% ของขนาด  
 บัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริงเมื่ออัตราการส่งเซลล์ละเมิดข้อตกลง

NORMALIZED TOTAL ARRIVAL RATE	เวลาเริ่มตรวจพบเซลล์สูญหายที่ลดลง (Sec.) (Real-Time Traffic)	% Improvement (Real-Time Traffic)	เวลาเริ่มตรวจพบเซลล์สูญหายที่ลดลง (Sec.) (Non Real-Time Traffic)	% Improvement (Non Real-Time Traffic)
1.05	2.2788	8.4293	-3.6145	-12.852
1.10	1.6476	7.7155	-3.8173	-16.513
1.15	1.7361	9.4894	-1.1957	-6.506
1.20	1.2305	7.9071	-2.3598	-14.516
1.25	1.4092	10.685	-0.6601	-4.8415
1.30	0.9478	8.2555	-1.2977	-10.25
1.35	0.4903	4.6702	-0.9636	-8.6039
1.40	1.064	11.276	-0.969	-9.2211
1.45	0.41097	4.6591	-0.78384	-8.4316
1.50	0.8501	10.788	-0.80365	-9.2077
	ค่าเฉลี่ย	8.3875	ค่าเฉลี่ย	-10.0939

## 5.4 วิเคราะห์ผลการทดสอบแบบจำลองการทำงานของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT

### 5.4.1 วิเคราะห์ผลการทดสอบการปรับค่า QLT ในช่วงต่างๆ

เมื่อพิจารณารูปที่ 5.7 และ 5.8 และตารางที่ 5.3 ถึง 5.8 ซึ่งทำการเปลี่ยนแปลงช่วงการปรับระดับ QLT ของอัลกอริทึมถึงรั้วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT ในช่วง 10-50%, 30-70% และ 50-90% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เพื่อเปรียบเทียบกับอัลกอริทึมถึงรั้วแบบ Priority Leaky Bucket ที่ใช้เทคนิค QLT ซึ่งทำการตั้งระดับ QLT คงที่ที่ 30%, 50% และ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง ตามลำดับ พบว่าอัลกอริทึมถึงรั้วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT สามารถลดอัตราส่วนการ สูญหายของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆของแหล่งกำเนิด ทราฟฟิกแบบเวลาจริงลงได้มากในทุกๆ ช่วงที่ทำการปรับค่า ส่วนอัตราส่วนการสูญหายของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงมีค่าเพิ่มขึ้นเพียงเล็กน้อย โดยที่มีเวลาการคอยโดยเฉลี่ยของเซลล์ที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆของแหล่งกำเนิดทราฟฟิกแบบเวลาจริงและแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงเปลี่ยนแปลงเล็กน้อย เมื่อพิจารณาค่าเฉลี่ยของเปอร์เซ็นต์ใน ตารางที่ 5.3 ถึง 5.8 เห็นได้ว่าผลการทดสอบแบบจำลองการทำงานของอัลกอริทึมถึงรั้วแบบ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT จะดีขึ้นอย่างเห็นได้ชัดเจนในการปรับค่า QLT ที่ช่วง 50-90% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง โดยมีค่าเฉลี่ยของเปอร์เซ็นต์การปรับปรุงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของแหล่งกำเนิดทราฟฟิกแบบเวลาจริงเท่ากับ 23.61% และ 2.11% ตามลำดับสำหรับค่าเฉลี่ยของเปอร์เซ็นต์การปรับปรุงอัตราส่วนการสูญหายของเซลล์ของแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงจะมีค่าต่ำลงเป็น -2.81% ซึ่งถือว่ามีค่าน้อยมากเมื่อเทียบกับค่าที่ได้จากแหล่งกำเนิดทราฟฟิกแบบเวลาจริง

### 5.4.2 วิเคราะห์ผลการทดสอบหาค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์

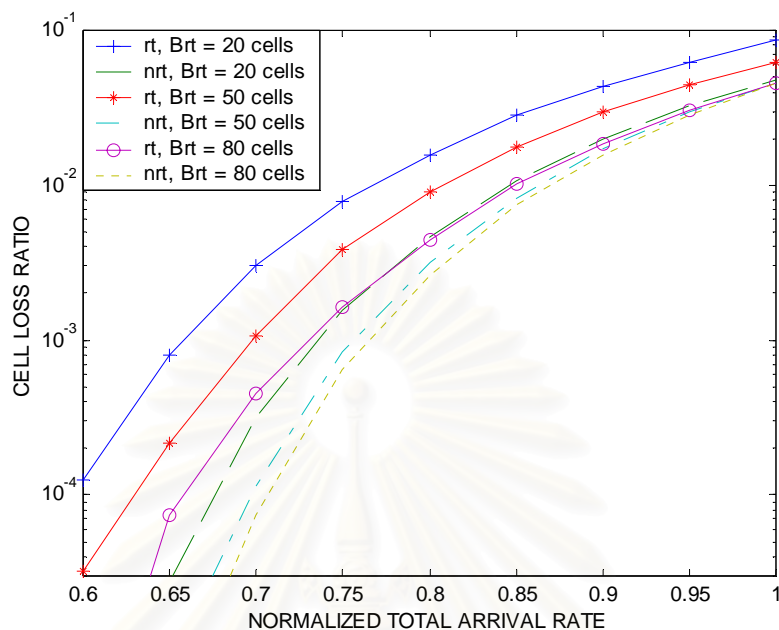
รูปที่ 5.9 แสดงค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ (Average Time for the First Loss) ของแหล่งกำเนิดทราฟฟิกแบบเวลาจริงและแหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงที่อัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆกัน เมื่ออัตราการส่งเซลล์เฉลี่ยมีค่าต่ำกว่าข้อตกลง และตารางที่ 5.9 แสดงเปอร์เซ็นต์การเพิ่มขึ้นของค่าเฉลี่ยเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์เมื่ออัตราการส่งเซลล์เฉลี่ยมีค่าต่ำกว่าข้อตกลง พบว่าการใช้ตัวควบคุมแบบพีซีซีทำการปรับระดับ QLT แบบพลวัต ทำให้แหล่งกำเนิดทราฟฟิกแบบไม่เป็นเวลาจริงมีค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์สูงในทุกอัตราการมาถึงโดยรวม

ของเซลล์ที่นอร์มอลไลซ์แล้วเมื่อเปรียบเทียบกับการตั้งระดับ QLT คงที่ที่ค่า 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ดังนั้นความน่าจะเป็นที่เซลล์จะเกิดการสูญหายจึงมีค่าน้อยซึ่งตรงกับความต้องการในกรณีที่แหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์ต่ำกว่าข้อตกลงข้อมต้องการอัตราส่วนการสูญหายของเซลล์ที่ต่ำ ส่วนกรณีของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงจะเห็นได้ว่าค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ในกรณีที่ใช้ตัวควบคุมแบบ ฟิชชีมีค่าเพิ่มขึ้นในช่วงอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วมีค่าต่ำถึงปานกลาง เมื่อเปรียบเทียบกับการตั้งระดับ QLT คงที่ที่ค่า 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ส่วนช่วงอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วมีค่าอัตราการส่งเซลล์เฉลี่ยใกล้เคียงกับค่าที่ตกลงไว้จะมีค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ต่ำกว่ากรณีการตั้งระดับ QLT คงที่ที่ค่า 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง แสดงให้เห็นว่าความน่าจะเป็นของเซลล์ที่จะเกิดการสูญหายมีค่าสูง ซึ่งถือว่าเป็นข้อเสียในกรณีที่กราฟฟิกส่งเซลล์ด้วยอัตราการส่งเซลล์ต่ำกว่าข้อตกลง แต่จะเป็นผลดีเมื่อพิจารณาในกรณีที่แหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์เฉลี่ยสูงกว่าข้อตกลง ซึ่งแหล่งกำเนิดกราฟฟิกแบบเวลาจริงมีค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ต่ำ ดังแสดงในรูปที่ 10 และตารางที่ 10 โดยแหล่งกำเนิดกราฟฟิกแบบเวลาจริงจะสามารถจัดการทิ้งเซลล์ได้รวดเร็วตรงตามความต้องการ ในขณะที่กราฟฟิกแบบไม่เป็นเวลาจริงจะมีค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์สูงซึ่งไม่เป็นผลดีเมื่อแหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์สูงกว่าข้อตกลง เนื่องจากกราฟฟิกไม่สามารถจัดการกับแหล่งกำเนิดที่ส่งเซลล์ด้วยอัตราการส่งเซลล์สูงกว่าข้อตกลงได้อย่างรวดเร็ว เมื่อพิจารณาแนวโน้มของกราฟโดยรวมทั้งในกรณีที่แหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์ต่ำกว่าข้อตกลงและสูงกว่าข้อตกลง พบว่าค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์จะมีค่าสูงมากเมื่ออัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วมีค่าต่ำ ทำให้มีความน่าจะเป็นที่จะเกิดการสูญหายของเซลล์มีค่าน้อยเพราะแหล่งกำเนิดกราฟฟิกให้กำเนิดกราฟฟิกด้วยอัตราที่ต่ำจึงมีโทเค็นเพียงพอที่จะรองรับ แต่เมื่อแหล่งกำเนิดกราฟฟิกให้กำเนิดกราฟฟิกในอัตราที่สูงขึ้น ทำให้ค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์จะมีค่าลดลงอย่างต่อเนื่อง ดังนั้นความน่าจะเป็นที่จะเกิดการสูญหายของเซลล์มีค่าสูงขึ้นเนื่องจากจำนวนโทเค็นมีอยู่อย่างจำกัดและโทเค็นเหล่านี้ต้องได้รับการจัดสรรให้กับเซลล์ที่กำเนิดจากแหล่งกำเนิดกราฟฟิกด้วยอัตราที่สูง เพราะแหล่งกำเนิดกราฟฟิกที่กำเนิดกราฟฟิกด้วยอัตราที่สูงจะมีความน่าจะเป็นที่มีที่ว่างในบัฟเฟอร์มีค่าต่ำบ่อยๆ

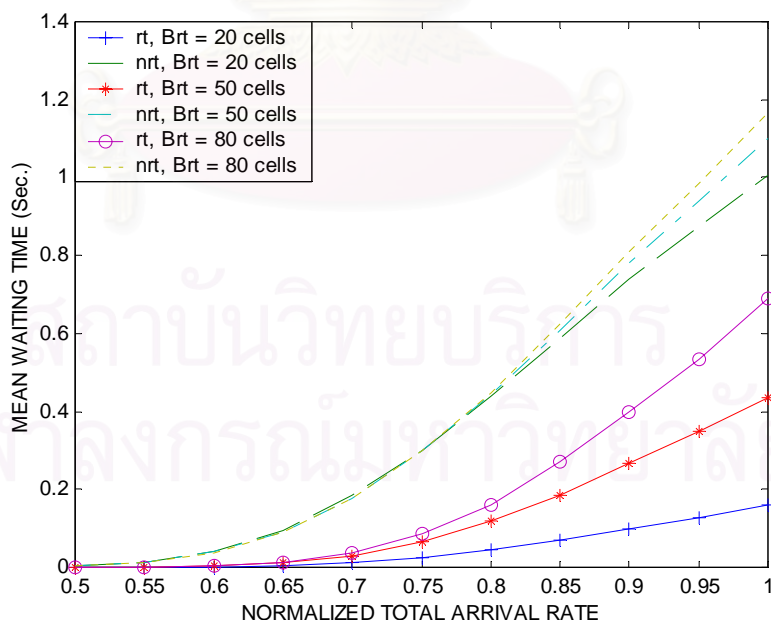


## 5.5 ผลการปรับเปลี่ยนค่าพารามิเตอร์ของแบบจำลองที่มีผลต่อสมรรถนะ

### 5.5.1 ผลการปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงที่มีต่อสมรรถนะ

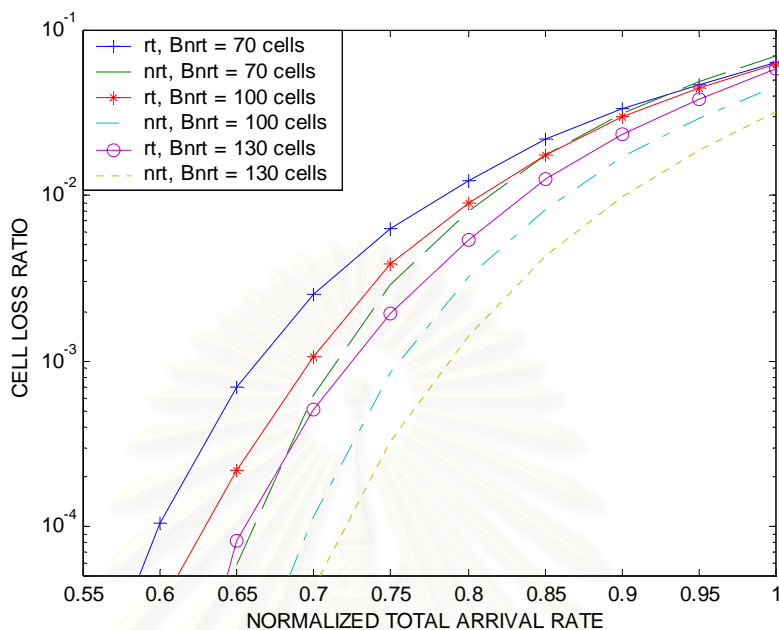


รูปที่ 5.11 ความสัมพันธ์ระหว่างอัตราส่วนการสูญหายของเซลล์ กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง

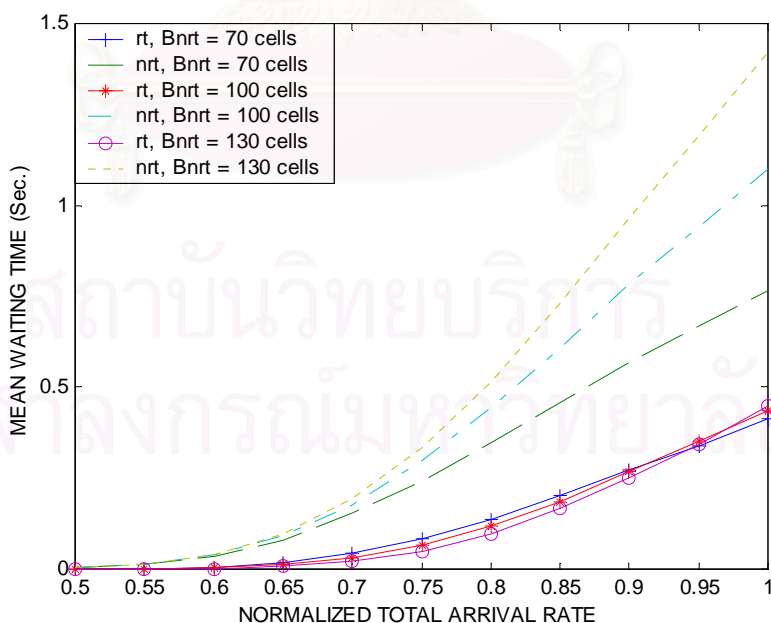


รูปที่ 5.12 ความสัมพันธ์ระหว่างเวลาการคอยโดยเฉลี่ยของเซลล์กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง

5.5.2 ผลการปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงที่มีต่อสมรรถนะ

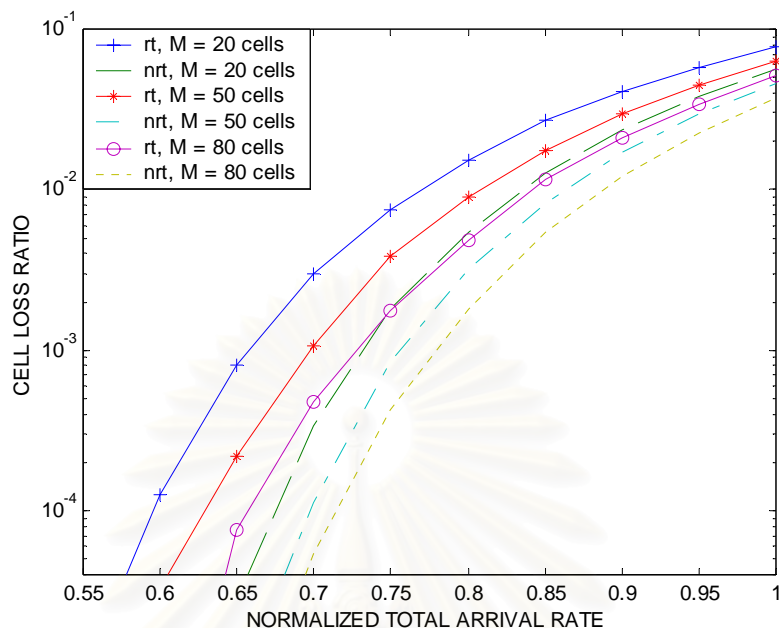


รูปที่ 5.13 ความสัมพันธ์ระหว่างอัตราส่วนการสูญหายของเซลล์ กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

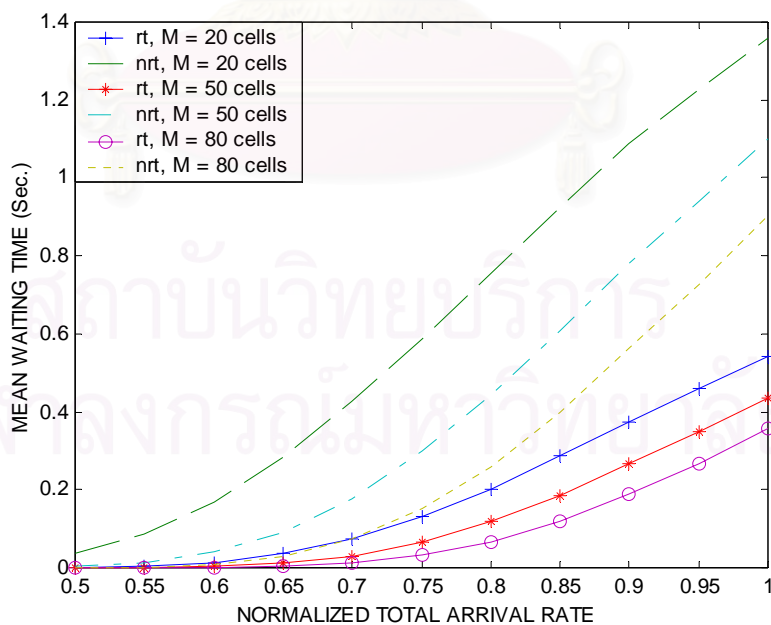


รูปที่ 5.14 ความสัมพันธ์ระหว่างเวลาการคอยโดยเฉลี่ยของเซลล์กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

### 5.5.3 ผลการปรับเปลี่ยนขนาดถังเก็บโทเค็นที่มีต่อสมรรถนะ



รูปที่ 5.15 ความสัมพันธ์ระหว่างอัตราส่วนการสูญหายของเซลล์กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดของถังเก็บโทเค็น



รูปที่ 5.16 ความสัมพันธ์ระหว่างเวลาการคอยโดยเฉลี่ยของเซลล์กับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ ของ Priority Leaky Bucket ที่ใช้เทคนิค Fuzzy-QLT เมื่อเปลี่ยนแปลงขนาดของถังเก็บโทเค็น

## 5.6 วิเคราะห์ผลการปรับเปลี่ยนค่าพารามิเตอร์ของแบบจำลองที่มีผลต่อสมรรถนะ

### 5.6.1 การปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง

รูปที่ 5.11 และ 5.12 แสดงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ตามลำดับ เปรียบเทียบกับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ โดยมีการปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง ( $B_n$ ) เมื่อขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงมีค่าเพิ่มขึ้น เช่น กรณีที่  $B_n = 80$  เซลล์ จะทำให้แหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีอัตราส่วนการสูญหายของเซลล์ลดลงอย่างมาก เนื่องจากแหล่งกำเนิดกราฟฟิกแบบเวลาจริงมีโอกาสที่จะมีที่ว่างในบัฟเฟอร์ (Buffer Space) มากขึ้น ทำให้มีที่รองรับเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงมากขึ้น ตัวควบคุมแบบพีซีจึงสามารถเลือกเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงเข้าสู่โครงข่ายได้มากขึ้น แต่การเพิ่มขนาดบัฟเฟอร์นี้จะทำให้ค่าเวลาการคอยโดยเฉลี่ยของเซลล์เพิ่มมากขึ้น สำหรับแหล่งกำเนิดกราฟฟิกแบบเวลาจริงนั้นเซลล์จะต้องเข้าคิวอยู่ในบัฟเฟอร์โดยเฉลี่ยนานขึ้นเนื่องจากขนาดบัฟเฟอร์ที่ใหญ่ขึ้นนั่นเอง สำหรับแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีเวลาการคอยโดยเฉลี่ยของเซลล์เพิ่มขึ้นเล็กน้อย ในทางกลับกันการปรับลดขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง เช่นกรณีที่มี  $B_n = 30$  เซลล์ สำหรับแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงจะมีอัตราส่วนการสูญหายของเซลล์เพิ่มขึ้นแต่มีเวลาการคอยโดยเฉลี่ยของเซลล์ลดลง เนื่องจากบัฟเฟอร์มีขนาดเล็กรองรับเซลล์ได้น้อย แต่ละเซลล์จึงมีความน่าจะเป็นที่จะสูญหายสูงและแต่ละเซลล์ก็อยู่ในบัฟเฟอร์ไม่นาน นอกจากนี้จะเห็นได้ว่าการปรับขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงนี้จะส่งผลต่ออัตราส่วนการสูญหายของเซลล์และค่าเวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงอย่างมาก แต่มีผลกระทบเพียงเล็กน้อยกับแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ซึ่งสังเกตได้จากเส้นกราฟที่มีการเปลี่ยนแปลงไม่มากนัก

### 5.6.2 การปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง

รูปที่ 5.13 และ 5.14 แสดงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ตามลำดับ เปรียบเทียบกับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ โดยมีการปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริง ( $B_{nr}$ ) แนวโน้มจะเหมือนกับกรณีการปรับเปลี่ยนขนาดบัฟเฟอร์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริง กล่าวคือ เมื่อเพิ่มขนาดบัฟเฟอร์  $B_{nr}$  ให้มากขึ้น อัตราส่วนการสูญหายของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีค่าลดลงอย่างมาก แต่เวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีค่าเพิ่มขึ้น สำหรับเวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงในช่วงอัตราการมาถึงโดยรวมของเซลล์ที่

นอร์มอลไลซ์แล้วมีค่าต่ำจนถึงปานกลางจะมีค่าลดลงเมื่อขนาดบัพเฟอร์  $B_{\text{net}}$  สูงขึ้น แต่เวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงในช่วงอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วมีค่าสูงๆ จะมีค่าเพิ่มขึ้น ในทางกลับกันถ้าลดขนาดบัพเฟอร์  $B_{\text{net}}$  อัตราส่วนการสูญหายของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงและแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงจะมีค่าเพิ่มขึ้น แต่ค่าเวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีค่าลดลง และเวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงในช่วงอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วมีค่าต่ำจนถึงปานกลางจะมีค่าเพิ่มขึ้นเมื่อขนาดบัพเฟอร์  $B_{\text{net}}$  สูงขึ้นแต่เวลาการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกแบบเวลาจริงในช่วงอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์ที่ค่าสูงๆ จะมีค่าลดลง

### 5.6.3 การปรับเปลี่ยนขนาดถังเก็บโทเค็น

รูปที่ 5.15 และ 5.16 แสดงอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ตามลำดับ เปรียบเทียบกับอัตราการมาถึงโดยรวมของเซลล์ที่นอร์มอลไลซ์แล้วค่าต่างๆ โดยมีการปรับเปลี่ยนขนาดถังเก็บโทเค็น (Token Pool) พบว่าเมื่อเพิ่มขนาดของถังเก็บโทเค็นจะส่งผลให้อัตราส่วนการสูญหายของเซลล์ของแหล่งกำเนิดกราฟฟิกทั้งแบบเวลาจริงและแบบไม่เป็นเวลาจริงมีค่าลดลง และการคอยโดยเฉลี่ยของเซลล์ของแหล่งกำเนิดกราฟฟิกทั้งแบบเวลาจริงและแบบไม่เป็นเวลาจริงก็มีค่าลดลงด้วย เนื่องจากถังเก็บโทเค็นที่มีขนาดใหญ่ขึ้นนี้สามารถเก็บโทเค็นเพื่อใช้จับคู่กับเซลล์ของแต่ละกราฟฟิกได้มากขึ้น จะเห็นว่าเมื่อพิจารณาถึงอัตราส่วนการสูญหายของเซลล์ การปรับขนาดถังเก็บโทเค็นมีผลต่อกราฟฟิกแบบเวลาจริงมากกว่ากราฟฟิกแบบไม่เป็นเวลาจริง แต่ในกรณีที่พิจารณาถึงเวลาการคอยโดยเฉลี่ยของเซลล์การปรับขนาดถังเก็บโทเค็นจะมีผลต่อกราฟฟิกแบบไม่เป็นเวลาจริงมากกว่า ซึ่งก็สอดคล้องตามคุณลักษณะของแต่ละกราฟฟิกคือ กราฟฟิกแบบเวลาจริงจะต้องคำนึงถึงเวลาการคอยโดยเฉลี่ยของเซลล์ที่ต่ำโดยยอมให้มีการสูญหายของเซลล์ได้ในระดับหนึ่งและกราฟฟิกแบบไม่เป็นเวลาจริงต้องการอัตราส่วนการสูญหายของเซลล์น้อยแต่ยอมให้มีเวลาการคอยโดยเฉลี่ยของเซลล์ได้นานขึ้น

สภามหาวิทยาลัย  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 6

### บทสรุปและข้อเสนอแนะ

#### 6.1 สรุปผลการทดสอบ

การเปรียบเทียบการทำงานของอัลกอริทึมตรวจจับแบบโทเค็นในแบบพื้นฐานกับแบบที่มีการจัดลำดับความสำคัญนั้น วิธีการจัดลำดับความสำคัญในการใช้งานโทเค็นนั้นมีผลต่อสมรรถนะของระบบทำให้ค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์มีค่าลดลงอย่างมากสาเหตุหลักมาจากการที่วิธีการจัดลำดับความสำคัญในการใช้งานโทเค็นนั้นมีข้อดีตรงที่แหล่งกำเนิดกราฟิกแต่ละแหล่งสามารถใช้โทเค็นที่สร้างขึ้นมาจากอีกฝ่ายหนึ่งได้เนื่องจากมีการใช้ถึงเก็บโทเค็นร่วมกัน และอัตราการกำเนิดโทเค็นได้มาจากอัตราการกำเนิดโทเค็นของแหล่งกำเนิด กราฟิกทั้งสองแบบรวมกัน

ในการจัดลำดับความสำคัญที่ใช้เทคนิค QLT นั้นการตั้งระดับ QLT มีผลทำให้สามารถลดค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์ในแหล่งกำเนิดกราฟิกแบบไม่เป็นเวลาจริงลงได้โดยที่มีผลกระทบต่อแหล่งกำเนิดกราฟิกอีกฝ่ายหนึ่งที่ทำงานร่วมกันคือแหล่งกำเนิดกราฟิกแบบเวลาจริงซึ่งมีผลการทำงานที่ลดต่ำลง ค่าอัตราส่วนการสูญหายของเซลล์และเวลาการคอยโดยเฉลี่ยของเซลล์สูงขึ้น แต่ผลการทำงานที่ลดลงของแหล่งกำเนิดกราฟิกแบบเวลาจริงนั้นลดลงไม่มากเมื่อเปรียบเทียบกับสมรรถนะที่เพิ่มขึ้นอย่างมากในแหล่งกำเนิด กราฟิกแบบไม่เป็นเวลาจริง ซึ่งผลที่ได้ถือว่าคุ้มค่า เนื่องจากประสิทธิภาพการทำงานที่ลดลงของแหล่งกำเนิดกราฟิกแบบเวลาจริงนั้นลดลงไม่มากและอยู่ในขั้นที่ยอมรับได้ตามค่าระดับความต้องการของกราฟิก

การใช้ตัวควบคุมแบบพีซีซีมาทำการปรับระดับของ QLT แบบพลวัตในอัลกอริทึมตรวจจับแบบจัดลำดับความสำคัญนั้น จะทำให้สามารถจัดระดับของ QLT ให้เหมาะสมกับสถานะของ กราฟิกที่เข้ามา โดยที่แหล่งกำเนิดกราฟิกทั้งสองแบบที่ใช้ในแบบจำลองนั้นมีอัตราการมาถึงโดยรวมของเซลล์แบบไม่คงที่ ผลที่ได้จากการปรับระดับ QLT แบบพลวัตมีผลดีขึ้นเมื่อพิจารณาเทียบกับการตั้งระดับ QLT คงที่ที่ระดับ 70% ของขนาดบัฟเฟอร์ของแหล่งกำเนิดแบบไม่เป็นเวลาจริง เนื่องจากทำให้อัตราส่วนการสูญหายของเซลล์ลดต่ำลง และค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์มีค่าสูงขึ้นในกรณีที่แหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์เฉลี่ยต่ำกว่าข้อตกลง จึงทำให้ความน่าจะเป็นของการเกิด False Alarm ลดลง สมรรถนะที่ลดลงมีเพียงในส่วน ของเวลาการคอยโดยเฉลี่ยของเซลล์ที่มีค่ามากขึ้นในแหล่งกำเนิดกราฟิกแบบเวลาจริงซึ่งต้องการ

เวลาการคอยโดยเฉลี่ยของเซลล์ที่ต่ำค่า และค่าเฉลี่ยของเวลาก่อนที่จะเริ่มตรวจพบการสูญหายของเซลล์ในแหล่งกำเนิดกราฟฟิกแบบไม่เป็นเวลาจริงมีค่าสูงขึ้น ในกรณีนี้แหล่งกำเนิดส่งเซลล์ด้วยอัตราการส่งเซลล์เฉลี่ยสูงกว่าข้อตกลง สำหรับผลการพิจารณาช่วงของการปรับค่า QLT ที่ช่วงต่างๆ เห็นได้ชัดว่าให้ผลการทำงานที่ดีขึ้น โดยที่อัตราส่วนการสูญหายของเซลล์ลดต่ำลง ในขณะที่ค่าเวลาการคอยโดยเฉลี่ยของเซลล์แทบจะไม่เปลี่ยนแปลง ผลการทำงานที่ดีขึ้นจะเห็นได้ชัดเจนในการปรับช่วงค่า QLT ที่ค่าสูงๆ (50%-90% ของขนาดบัพเฟอร์) และท้ายที่สุดในส่วนของการปรับเปลี่ยนค่าพารามิเตอร์ของแบบจำลอง ซึ่งได้แก่ขนาดของบัพเฟอร์และขนาดของถังเก็บโทเค็น สรุปได้ว่า ถ้ากำหนดค่าขนาดบัพเฟอร์ของกราฟฟิกแต่ละแบบให้สูงขึ้น ก็จะทำให้อัตราส่วนการสูญหายของเซลล์ลดต่ำลงในบัพเฟอร์ของกราฟฟิกแต่ละแบบ และลดอัตราส่วนการสูญหายของเซลล์ในกราฟฟิกทั้งสองแบบลงได้เมื่อทำการเพิ่มขนาดของถังเก็บโทเค็น ขณะที่ค่าเวลาการคอยโดยเฉลี่ยของเซลล์สูงขึ้นเนื่องมาจากการเพิ่มค่าพารามิเตอร์ดังกล่าว

## 6.2 ข้อดีข้อเสีย

สรุปข้อดีและข้อเสียของการใช้ตัวควบคุมแบบพีซีมาทำการปรับระดับของ QLT แบบพลวัตในอัลกอริทึมถังรับแบบจัดลำดับความสำคัญได้ดังนี้

### ข้อดี

1. ปรับปรุงประสิทธิภาพในส่วนของการสูญหายของเซลล์ ที่เกิดขึ้นในกราฟฟิกทั้งสองแบบ เมื่อเปรียบเทียบกับการตั้งระดับ QLT ที่คงที่ ซึ่งตรงกับความต้องการของ กราฟฟิกแบบไม่เป็นเวลาจริงที่ให้ลำดับความสำคัญกับการสูญหายของเซลล์ก่อน
2. ทำให้สามารถจัดสรรการใช้โทเค็นให้เป็นไปอย่างมีประสิทธิภาพมากขึ้น เนื่องมาจากการปรับระดับ QLT แบบพลวัต ทำให้เหมาะกับกราฟฟิกที่มีอัตราการมาถึงของเซลล์แบบไม่คงที่
3. การทำงานของตัวควบคุมแบบพีซี มีข้อดีคืออาศัยสมการทางคณิตศาสตร์ที่ไม่ซับซ้อนอธิบายค่าต่างๆในลักษณะของเชิงภาษา และออกแบบกฎการทำงานและค่าพารามิเตอร์โดยขึ้นกับพื้นฐานความรู้ ความชำนาญของผู้ออกแบบเป็นหลัก

### ข้อเสีย

1. การลดค่าอัตราส่วนการสูญหายของเซลล์ มักจะทำให้ค่าเวลาการคอยโดยเฉลี่ยของเซลล์สูงขึ้น ซึ่งควรพิจารณาถึงความคุ้มค่าของประสิทธิภาพที่เพิ่มขึ้น และความต้องการพื้นฐานของแต่ละแหล่งกำเนิดประกอบด้วย โดยเฉพาะในแหล่งกำเนิดกราฟฟิกแบบเวลาจริงที่ให้ลำดับความสำคัญสูงกับค่าเวลาการคอยโดยเฉลี่ยของเซลล์
2. ในการออกแบบและทดสอบนั้นถ้าหากปรับเปลี่ยนไปใช้งานกับกราฟฟิกประเภทอื่น เช่น การส่งภาพ หรือวิดีโอ ซึ่งมีอัตราข้อมูลและ Burst Rate ที่สูงขึ้น หรือมีการปรับเปลี่ยนค่า

พารามิเตอร์ของแบบจำลอง จะต้องทำการออกแบบตัวควบคุมแบบฟัซซี่ใหม่ เพื่อให้เหมาะกับ ทราฟฟิกและแบบจำลองนั้นๆ

### 6.3 ข้อเสนอแนะ

ในการออกแบบในส่วนของตัวควบคุมแบบฟัซซี่นั้น ไม่มีบรรทัดฐานที่แน่นอนในการออกแบบรูปร่างและระดับการซ้อนทับกันของฟังก์ชันสมาชิกแต่ละฟังก์ชัน ซึ่งในงานวิจัยนี้อาศัยพื้นฐานความรู้ของผู้ออกแบบและวิธีลองผิดลองถูกในการปรับแต่งฟังก์ชันสมาชิกและฐานกฎ ซึ่งเป็นการทดสอบปรับแต่งโดยการพิจารณาจากผลตอบที่ได้รับจากการปรับแต่งค่าแต่ละครั้งแล้วนำมาปรับแต่งเพื่อเลือกค่าพารามิเตอร์เริ่มต้นใหม่ ทำให้เสียเวลาในการปรับแต่งมาก เพื่อลดเวลาในการปรับแต่งค่าพารามิเตอร์ต่างๆข้างต้นอาจนำเทคนิคการปรับตัวเองมาประยุกต์ใช้กับตัวควบคุมแบบฟัซซี่ เช่นการใช้ตัวควบคุมแบบฟัซซี่แบบที่สามารถปรับตัวเองได้ (Adaptive Fuzzy Control) หรือตัวควบคุมแบบฟัซซี่ในรูปของโครงข่ายประสาท (Neuro-Fuzzy) ฯลฯ



## รายการอ้างอิง

1. E. P. Rathgeb. Modelling and Performance Comparison of Policing Mechanisms for ATM Network. IEEE Journ. On Sel. Areas in Comm. 9, 3(April 1991):325-334
2. H. Hemmer and P. T. Huth. Evaluation of Policing Function in ATM Networks. Queueing, Performance and Control in ATM. ITC-13. (June 1991):111-116
3. M. Jo, S. Kimura and Y. Ebihara. Flow Control for Delay-Sensitive Service with Shared Token Pool Scheme. Information Networking 1998 (ICOIN-12). (1998):411-414
4. J. Zeng and L. F. Merakos. Analysis of a Priority Leaky Becket for Virtual Path Traffic in ATM Network. Global Telecommunication Conference. 1, (1994):606-611
5. D. I. Choi, B. D. Choi and D. K. Sung. Performance Analysis of Priority Leaky Bucket Scheme with Queue-Length-Threshold Scheduling Policy. Communications IEE Proceeding Published. 145, (1998):395-401
6. ATM Forum. ATM Traffic Management Specification Version 4.1. April 1999.
7. J. S. Tuner. New Direction in Communications (or Which Way to the Informaton Age?). IEEE Commun. Magazine. 25, (Oct. 1986):8-15
8. I. Cidon and I. S. Gopal. PARIS: An Approach to Integrated High-Speed Private Networks. Int. J. of Digital and Analog Cabled Systems. 1, (April 1988):77-78
9. J. -S. R. Jang, C. -T. Sun and E. Mizutani. Neuro-Fuzzy and Soft Computing. Prentice-Hall, 1997.
10. สุวลัย กลั่นความดี. ระบบควบคุมกระบวนการพลวัตแบบฟัซซี่ลอจิก. กรุงเทพมหานคร, พฤศจิกายน 2537.
11. A. M. Law and W. D. Kelton. Simulation Modelling and Analysis. McGraw-Hill, 1991.

## ประวัติผู้เขียน

นายอภิศักดิ์ สัตถณะ เกิดเมื่อวันที่ 10 ตุลาคม พ.ศ. 2518 ที่อำเภอเมือง จังหวัดนครราชสีมา สำเร็จ การศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี การศึกษา 2539 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2540



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย