

การประยุกต์ใช้งานพีซีในอินเทอร์เน็ตเวิร์ก
สำหรับหน้าที่ควบคุมพารามิเตอร์การใช้ในโครงข่ายเอทีเอ็ม



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต


สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2543

ISBN 974-13-0138-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AN APPLICATION OF FUZZY NEURAL NETWORKS
FOR USAGE PARAMETER CONTROL FUNCTION IN ATM NETWORKS



Ms. Aimaschana Niruntasukrat

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 2000

ISBN 974-13-0138-3

หัวข้อวิทยานิพนธ์

การประยุกต์ใช้งานฟิชซีนิวรอลเน็ตเวิร์กสำหรับหน้าที่ควบคุมพารามิเตอร์
การใช้ในโครงข่ายเอทีเอ็ม

โดย

นางสาวเอมอัชชา นรินทร์สุขรัตน์

สาขาวิชา

วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยรับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ



คณบดีคณะวิศวกรรมศาสตร์

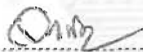
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์



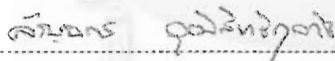
ประธานกรรมการ

(ศาสตราจารย์ ดร.ประสิทธิ์ ประพัฒน์มงคล)



อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล)



กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ฉัตรกร วุฒิสิตติกุลกิจ)

จุฬาลงกรณ์มหาวิทยาลัย

เอมอัชชา นีรันตสุขรัตน์ : การประยุกต์ใช้งานฟัซซีนิวรอลเน็ตเวิร์กสำหรับหน้าที่ควบคุมพารามิเตอร์
การใช้ในโครงข่ายเอทีเอ็ม (AN APPLICATION OF FUZZY NEURAL NETWORKS FOR USAGE
PARAMETER CONTROL FUNCTION IN ATM NETWORKS) อ. ที่ปรึกษา : รศ. ดร. วาทีต
เบญจพลกุล, 104 หน้า. ISBN 974-13-0138-3.

วิทยานิพนธ์ฉบับนี้เสนอตัวตรวจจราจรฟฟิก (Traffic Policer) สำหรับการควบคุมพารามิเตอร์การใช้
(Usage Parameter Control: UPC) ในโครงข่ายเอทีเอ็ม โดยใช้ฟัซซีนิวรอลเน็ตเวิร์ก วิธีที่เสนอมิโครงสร้างพื้นฐาน
เป็นกลไกถังรั้วแบบโทเคิน (Token Leaky Bucket Mechanism) ที่มีฟัซซีนิวรอลเน็ตเวิร์กทำหน้าที่ควบคุมอัตรา
การกำเนิดโทเคิน โดยอาศัยคุณสมบัติการควบคุมที่ยืดหยุ่นของฟัซซีลอจิก ทำให้การตรวจอัตราเซลล์เฉลี่ยของ
แหล่งกำเนิดเดี่ยวประเภท packet voice มีค่าใกล้เคียงกับอัตราที่ตกลงไว้ ปรับปรุงประสิทธิภาพของตัวตรวจใน
แง่ความสามารถในการเลือก (Selectivity) รวมทั้งมีการตอบสนอง (Responsiveness) ที่รวดเร็วยิ่งขึ้น อาศัยความ
สามารถในการจับคู่ (map) และการคำนวณที่รวดเร็วของนิวรอลเน็ตเวิร์กเพื่อทำหน้าที่แทน inference engine ใน
ฟัซซีลอจิก ทำให้การทำงานซับซ้อนน้อยลงและรวดเร็วยิ่งขึ้น

เมื่อเปรียบเทียบประสิทธิภาพในแง่ความสามารถการเลือกของกลไกที่เสนอ, กลไกถังรั้วแบบไม่มีบัฟเฟอร์
และกลไกหน้าต่างฟัซซีลอจิกกับความน่าจะเป็นการละเมิดอุดมคติ (Ideal Violation Probability) ได้รากที่สอง
ของผลรวมค่าผิดพลาดกำลังสองเป็น 0.011550, 1.023511 และ 0.161944 ตามลำดับ ส่วนในแง่การตอบสนอง
กลไกที่เสนอเริ่มละทิ้งเซลล์แรกเมื่อแหล่งกำเนิดส่งเซลล์ด้วยอัตราเซลล์เฉลี่ยเป็น 1.5 เท่าของอัตราที่ตกลงไว้หลังจาก
จากที่ปล่อยเซลล์ผ่านเข้าไปยังโครงข่ายแล้ว 359 เซลล์ เปรียบเทียบกับ 309 และ 731 เซลล์ของกลไกถังรั้วแบบไม่
มีบัฟเฟอร์ และกลไกหน้าต่างฟัซซีลอจิกตามลำดับ ผลการทดสอบแสดงให้เห็นว่า ตัวตรวจจราจรฟฟิกฟัซซีนิวรอล
เน็ตเวิร์กมีความเข้มงวดมากกว่ากลไกที่เปรียบเทียบ แม้ว่าจะเริ่มละทิ้งเซลล์แรกช้ากว่าถังรั้วแบบไม่มีบัฟเฟอร์ซึ่ง
เป็นกลไกที่ตอบสนองเร็วที่สุด แต่พฤติกรรมตอบสนองต่อการละเมิดระยะยาวดีกว่ามาก ในขณะที่กลไกที่เสนอยัง
มีความสามารถในการเลือกสูงที่สุด

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชาวิศวกรรมไฟฟ้า
สาขาวิชาวิศวกรรมไฟฟ้า
ปีการศึกษา 2543

ลายมือชื่อนิสิต..... *1020102 วาทีต*
ลายมือชื่ออาจารย์ที่ปรึกษา..... *Qud*

#4170662121 MAJOR ELECTRICAL ENGINEERING

KEY WORD: Fuzzy Neural Network / Usage Parameter Control / ATM Network

AIMASCHANA NIRUNTASUKRAT : AN APPLICATION OF FUZZY NEURAL NETWORKS FOR USAGE PARAMETER CONTROL FUNCTIONS IN ATM NETWORKS. THESIS ADVISOR : ASSOC. PROF. WATIT BENJAPOLAKUL. 104 pp. ISBN 974-13-0138-3.

This thesis proposes a novel traffic policer for UPC function in ATM Networks. The architecture of this policer is based on the Token Leaky Bucket mechanism using a fuzzy neural network, which dynamically updates the token generation rate. With the inherit of formalized approximate reasoning offered by fuzzy logic, the control process would be smooth and flexible, and would make selectivity improvement possible. With the inherits of mapping capability and fast parallel calculation offered by a neural network, the complexity in inference process of fuzzy logic is eliminated by replacing it with the neural network. The performance evaluation of the proposed policer is done with a packet voice source model and is compared to a conventional method (unbuffered leaky bucket) and the fuzzy logic window. In selectivity aspect, we compare cell loss ratio caused by policers with ideal violation probability in terms of Root of Sum Square Error (RSSE). With the proposed mechanism, unbuffered leaky bucket and fuzzy logic window, the RSSEs obtained are 0.011550, 1.023511 and 0.161944, respectively. The comparison of the responsive behavior in terms of the average number of cells from the violating source allowed by the policer to pass into the network before detecting the first cell is made. The proposed scheme takes action after 359 cells, while the unbuffered LB and fuzzy logic window do so after 309 and 731 cells, respectively. From the results obtained, the fuzzy neural network approach performs well in both aspects. Despite the slower rejection time compared to the unbuffered LB, the percentage of cells detected by the proposed policer grows at the highest rate in long term comparison. Moreover, the fuzzy neural network approach yields the highest performance in selectivity among other mechanisms.

Department..... Electrical Engineering..... Student's signature *102012 inipak*

Field of study..... Electrical Engineering..... Advisor's signature *Orab*

Academic year 2000

กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณผู้ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ คือ รองศาสตราจารย์ ดร. วาทีต เบญจพลกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความช่วยเหลือ แนะนำแนวทาง และให้ข้อคิด ที่เป็นประโยชน์แก่ผู้วิจัยมาโดยตลอด ขอขอบคุณโครงการศิษย์ก้นกุฏิที่สนับสนุนโดยให้ทุนการศึกษาและวิจัย ครูและอาจารย์ของผู้วิจัยทุกท่านที่ให้ความรู้ประสิทธิภาพวิชา และเพื่อนนิสิต สาขาระบบโทรคมนาคมทุกคน ที่ให้กำลังใจในการทำงานรวมถึงคำแนะนำ

ทำยนี้ ผู้วิจัยกราบขอบพระคุณบิดามารดาซึ่งสนับสนุนผู้วิจัยเสมอมาจนกระทั่งสำเร็จการศึกษา

เอมอัชชา นีรันตสุขรัตน์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

| | หน้า |
|---|------|
| บทคัดย่อภาษาไทย..... | ง |
| บทคัดย่อภาษาอังกฤษ..... | จ |
| กิตติกรรมประกาศ..... | ฉ |
| สารบัญ..... | ช |
| สารบัญรูป..... | ฐ |
| สารบัญตาราง..... | ถ |
| บทที่ | |
| 1 บทนำ..... | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 1 |
| 1.2 วัตถุประสงค์..... | 3 |
| 1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์..... | 4 |
| 1.4 ขั้นตอนและวิธีดำเนินงาน..... | 4 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ..... | 4 |
| 2 การควบคุมทราฟฟิกในโครงข่ายเอทีเอ็ม..... | 6 |
| 2.1 กล่าวนำ..... | 6 |
| 2.2 การควบคุมการอนุญาตการเรียก..... | 7 |
| 2.3 การควบคุมพารามิเตอร์การใช้..... | 8 |
| 2.3.1 กลไกถังรั้ว..... | 9 |
| 2.3.1.1 กลไกถังรั้วแบบไม่มีบัฟเฟอร์..... | 9 |
| 2.3.1.2 กลไกถังรั้วแบบโทเค็นหรือกลไกถังรั้วแบบมีบัฟเฟอร์..... | 10 |
| 2.3.2 กลไกหน้าต่าง..... | 11 |
| 2.3.2.1 กลไกหน้าต่างกระโดด..... | 11 |
| 2.3.2.2 กลไกหน้าต่างเคลื่อนที่..... | 11 |
| 2.3.2.3 กลไกหน้าต่างกระโดดแบบจุดชนวน..... | 11 |
| 2.3.2.4 กลไกเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอ็กซ์โพเนนเชียล..... | 12 |
| 2.4 การประเมินประสิทธิภาพของกลไกตรวจทราฟฟิก..... | 13 |
| 3 แบบจำลองแหล่งกำเนิดและกลไกตรวจทราฟฟิก..... | 15 |
| 3.1 แบบจำลองแหล่งกำเนิดที่ใช้ในการทดสอบ..... | 15 |

สารบัญ (ต่อ)

| | | |
|---------|---|----|
| 3.2 | โครงสร้างของกลไกตรวจหาฟิสิกที่ใช้เปรียบเทียบ..... | 17 |
| 3.2.1 | กลไกถังรั่วแบบไม่มีปั๊มเฟอร์..... | 17 |
| 3.2.2 | กลไกหน้าต่างพีชชีลลจิก..... | 17 |
| 3.3 | โครงสร้างตัวตรวจหาฟิสิกโดยใช้พีชชีลลจิก..... | 20 |
| 3.4 | การทดสอบความถูกต้องของแบบจำลอง..... | 29 |
| 3.4.1 | การทดสอบความถูกต้องของการจำลองกลไกถังรั่ว แบบไม่มีปั๊มเฟอร์..... | 29 |
| 3.4.2 | การทดสอบความถูกต้องของการจำลองกลไกหน้าต่างพีชชีลลจิก... | 30 |
| 4 | ผลการทดสอบประสิทธิภาพกลไกการตรวจหาฟิสิกโดยใช้พีชชีลลจิกและ การวิเคราะห์ผลการทดสอบ..... | 33 |
| 4.1 | กล่าวนำ..... | 33 |
| 4.2 | ผลของขนาดของบ่อโทเคินต่อประสิทธิภาพของกลไกถังรั่วแบบโทเคิน โดยใช้พีชชีลลจิก..... | 33 |
| 4.2.1 | กลไกตรวจหาฟิสิกเมื่อใช้แบบจำลองแหล่งกำเนิดแบบที่ 1..... | 34 |
| 4.2.1.1 | ผลการทดสอบเปรียบเทียบความสามารถในการเลือก..... | 34 |
| 4.2.1.2 | ผลการทดสอบเปรียบเทียบการตอบสนอง..... | 41 |
| 4.2.1.3 | เวลาประวิงที่เกิดจากกลไกตรวจหาฟิสิก..... | 44 |
| 4.2.2 | กลไกตรวจหาฟิสิกเมื่อใช้แบบจำลองแหล่งกำเนิดแบบที่ 2..... | 44 |
| 4.2.2.1 | ผลการทดสอบเปรียบเทียบความสามารถในการเลือก..... | 44 |
| 4.2.2.2 | ผลการทดสอบเปรียบเทียบการตอบสนอง..... | 51 |
| 4.2.2.3 | เวลาประวิงที่เกิดจากกลไกตรวจหาฟิสิก..... | 53 |
| 4.3 | วิเคราะห์ผลของขนาดของบ่อโทเคินที่มีต่อประสิทธิภาพของ กลไกตรวจหาฟิสิกที่เสนอ..... | 54 |
| 4.3.1 | ความสามารถในการเลือก..... | 54 |
| 4.3.2 | การตอบสนอง..... | 54 |
| 4.3.3 | เวลาประวิง..... | 55 |
| 4.4 | วิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างกลไกตรวจหาฟิสิก ต่างๆ..... | 56 |
| 4.4.1 | ความสามารถในการเลือก..... | 56 |

สารบัญ (ต่อ)

| | |
|--|----|
| 4.4.2 การตอบสนอง..... | 58 |
| 4.4.3 เวลาประวิง..... | 59 |
| 5 ผลการทดสอบการใช้นิรอลเน็ตเวิร์กในตัวตรวจกราฟฟิก..... | 60 |
| 5.1 กล่าวนำ..... | 60 |
| 5.2 โครงสร้างของนิรอลเน็ตเวิร์กที่ใช้..... | 60 |
| 5.3 แบบจำลองแหล่งกำเนิดแบบที่ 1..... | 61 |
| 5.3.1 ผลการทดสอบเมื่อเปลี่ยนอินพุต..... | 63 |
| 5.3.2 ผลการทดสอบเมื่อเพิ่มความถี่ในการเลือกอินพุต..... | 68 |
| 5.3.3 ผลการทดสอบเมื่อเปลี่ยนวิธีฝึก..... | 73 |
| 5.3.4 ผลการทดสอบเมื่อลดจำนวนอินพุตที่ให้ค่าอินพุตซ้ำซ้อน..... | 75 |
| 5.3.5 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก..... | 80 |
| 5.3.6 ผลการทดสอบเปรียบเทียบการตอบสนอง..... | 82 |
| 5.3.7 ผลการทดสอบเปรียบเทียบเวลาประวิง..... | 83 |
| 5.4 แบบจำลองแหล่งกำเนิดแบบที่ 2..... | 84 |
| 5.4.1 การแก้ปัญหาโดยใช้นิรอลเน็ตเวิร์ก 1 ชุด..... | 84 |
| 5.4.1.1 การเลือกชุดฝึกโดยไม่ลดความซ้ำซ้อนของอินพุต ที่ให้เอาต์พุตเท่ากัน..... | 84 |
| 5.4.1.2 การเลือกชุดฝึกโดยลดความซ้ำซ้อนของอินพุตที่ให้ เอาต์พุตเท่ากัน..... | 85 |
| 5.4.2 การแก้ปัญหาโดยใช้นิรอลเน็ตเวิร์กแบบขนาน..... | 86 |
| 5.4.3 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก..... | 90 |
| 5.4.4 ผลการทดสอบเปรียบเทียบการตอบสนอง..... | 91 |
| 5.4.5 ผลการทดสอบเปรียบเทียบเวลาประวิง..... | 93 |
| 5.5 ผลการทดสอบเปรียบเทียบเวลาในการประมวลผลของตัวควบคุม พัชชีนิรอลเน็ตเวิร์กเปรียบเทียบกับตัวควบคุมพัชชีลอจิก..... | 93 |
| 6 สรุปผลการทดสอบและข้อเสนอแนะ..... | 95 |
| 6.1 สรุปผลการทดสอบ..... | 95 |
| 6.2 ข้อดีและข้อเสียของตัวตรวจถ่วงรั้วโดยใช้พัชชีนิรอลเน็ตเวิร์ก..... | 96 |
| 6.3 ข้อเสนอแนะ..... | 96 |

สารบัญ (ต่อ)

| | |
|----------------------|-----|
| รายการอ้างอิง..... | 98 |
| ภาคผนวก..... | 99 |
| ประวัติผู้เขียน..... | 104 |



จุฬา

สารบัญรูป

| | | |
|------------|---|----|
| รูปที่ 2.1 | กลไกถังรั้วแบบไม่มีฟเฟอร์..... | 10 |
| รูปที่ 2.2 | กลไกถังรั้วแบบโทเค็น..... | 10 |
| รูปที่ 2.3 | หลักการของกลไกหน้าต่างแบบต่างๆ เมื่อกำหนดให้ขีดจำกัดหน้าต่างเท่ากับ 4 (ก) หน้าต่างกระโดด (ข) หน้าต่างเคลื่อนที่ (ค) หน้าต่างแบบจุดชนวน..... | 12 |
| รูปที่ 2.4 | เส้นโค้งความน่าจะเป็นการละเมิดอุดมคติ..... | 13 |
| รูปที่ 3.1 | แบบจำลองแหล่งกำเนิดแบบเปิด-ปิด..... | 16 |
| รูปที่ 3.2 | แบบจำลองการทำงานของกลไกหน้าต่างพีชชีลอจิก..... | 17 |
| รูปที่ 3.3 | ฟังก์ชันสมาชิกสำหรับพีชชีเซตของกลไกหน้าต่างพีชชีลอจิก (ก) A_0, A_i (ข) N_i (ค) ΔN_{i+1} | 18 |
| รูปที่ 3.4 | แบบจำลองการทำงานของกลไกถังรั้วพีชชีนิวรอลเน็ตเวิร์ก..... | 20 |
| รูปที่ 3.5 | ฟังก์ชันสมาชิกสำหรับพีชชีเซตของกลไกถังรั้วพีชชีนิวรอลเน็ตเวิร์ก (ก) N_c (ข) N_T (ค) ΔN_c (ง) R_i, R_{i+1} | 23 |
| รูปที่ 3.6 | ผลเปรียบเทียบการจำลองกลไกถังรั้ว เมื่อกำหนดให้ (ก) E(S) คงที่ (ข) E(X) คงที่..... | 30 |
| รูปที่ 3.7 | ผลเปรียบเทียบการจำลองกลไกหน้าต่างพีชชีลอจิกโดยใช้แบบจำลองแหล่งกำเนิดแบบที่ 1 เมื่อกำหนดให้ (ก) E(S) คงที่ (ข) E(X) คงที่..... | 31 |
| รูปที่ 3.8 | ผลเปรียบเทียบการจำลองกลไกหน้าต่างพีชชีลอจิกโดยใช้แบบจำลองแหล่งกำเนิดแบบที่ 2 เมื่อกำหนดให้ (ก) E(S) คงที่ (ข) E(X) คงที่..... | 31 |
| รูปที่ 4.1 | ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจกราฟฟิกเมื่อแปรค่า E(X) (ก) P = 500 (ข) P = 750 (ค) P = 1,000 (ง) P = 1,250..... | 37 |
| รูปที่ 4.2 | ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจกราฟฟิกเมื่อแปรค่า E(S) (ก) P = 500 (ข) P = 750 (ค) P = 1,000 (ง) P = 1,250..... | 40 |
| รูปที่ 4.3 | การตอบสนองของกลไกตรวจกราฟฟิก เมื่อแปรค่า E(X) ของแหล่งกำเนิด..... | 42 |
| รูปที่ 4.4 | การตอบสนองของกลไกตรวจกราฟฟิก เมื่อแปรค่า E(S) ของแหล่งกำเนิด..... | 42 |
| รูปที่ 4.5 | เวลาประวิงที่เกิดจากกลไกตรวจกราฟฟิกที่โหลดปกติ..... | 44 |

สารบัญรูป (ต่อ)

| | |
|--|----|
| รูปที่ 4.6 ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจหราวฟฟิกเมื่อ แปรค่า $E(X)$ (ก) $P = 3,000$ (ข) $P = 3,600$ (ค) $P = 4,200$ (ง) $P = 4,800$ | 46 |
| รูปที่ 4.7 ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจหราวฟฟิกเมื่อ แปรค่า $E(S)$ (ก) $P = 3,000$ (ข) $P = 3,600$ (ค) $P = 4,200$ (ง) $P = 4,800$ | 49 |
| รูปที่ 4.8 การตอบสนองของกลไกตรวจหราวฟฟิกเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด.... | 52 |
| รูปที่ 4.9 การตอบสนองของกลไกตรวจหราวฟฟิกเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด.... | 52 |
| รูปที่ 4.10 เวลาประวิงที่เกิดจากกลไกตรวจหราวฟฟิกที่โหลดปกติ..... | 53 |
| รูปที่ 5.1 จำนวนฮิดเดนโนดที่ใช้ทดสอบกรณีแบบจำลองแหล่งกำเนิดแบบที่ 1..... | 62 |
| รูปที่ 5.2 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม ฟัซซีนิวรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อแปรค่า $E(X)$ ของ แหล่งกำเนิด..... | 62 |
| รูปที่ 5.3 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม ฟัซซีนิวรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อแปรค่า $E(S)$ ของ แหล่งกำเนิด..... | 63 |
| รูปที่ 5.4 จำนวนฮิดเดนโนดที่ใช้ในการทดสอบเมื่อเปลี่ยนอินพุต (ก) วิธี ก (ข) วิธี ข (ค) วิธี ค..... | 65 |
| รูปที่ 5.5 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม ฟัซซีนิวรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ก) ในกรณีแปรค่า $E(X)$ | 65 |
| รูปที่ 5.6 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม ฟัซซีนิวรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ก) ในกรณีแปรค่า $E(S)$ | 66 |
| รูปที่ 5.7 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม ฟัซซีนิวรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ข) ในกรณีแปรค่า $E(X)$ | 66 |

สารบัญรูป (ต่อ)

| | | |
|-------------|---|----|
| รูปที่ 5.8 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ข) ในกรณีแปรค่า $E(S)$ | 67 |
| รูปที่ 5.9 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ค) ในกรณีแปรค่า $E(X)$ | 67 |
| รูปที่ 5.10 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเปลี่ยนอินพุตตาม วิธี (ค) ในกรณีแปรค่า $E(S)$ | 68 |
| รูปที่ 5.11 | จำนวนฮิดเดนโนตที่ใช้ในการทดสอบเมื่อเพิ่มควมถี่ของอินพุต (ก) วิธี ก (ข) วิธี ข (ค) วิธี ค..... | 69 |
| รูปที่ 5.12 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ก) ในกรณีแปรค่า $E(X)$ | 70 |
| รูปที่ 5.13 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ก) ในกรณีแปรค่า $E(S)$ | 70 |
| รูปที่ 5.14 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ข) ในกรณีแปรค่า $E(X)$ | 71 |
| รูปที่ 5.15 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ข) ในกรณีแปรค่า $E(S)$ | 71 |
| รูปที่ 5.16 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ค) ในกรณีแปรค่า $E(X)$ | 72 |
| รูปที่ 5.17 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พีชชีนิวรอลเนตเวิร์กกับตัวควบคุมพีชชีลจิกเมื่อเพิ่มอินพุตตาม วิธี (ค) ในกรณีแปรค่า $E(S)$ | 72 |

สารบัญรูป (ต่อ)

| | |
|---|----|
| รูปที่ 5.18 จำนวนฮิตเดนโนดที่ใช้ทดสอบเมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์ก..... | 73 |
| รูปที่ 5.19 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อใช้วิธีฝึกแบบไม่ รบกวนการใช้งาน ในกรณีแปรค่า $E(X)$ | 74 |
| รูปที่ 5.20 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อใช้วิธีฝึกแบบไม่ รบกวนการใช้งาน ในกรณีแปรค่า $E(S)$ | 74 |
| รูปที่ 5.21 จำนวนฮิตเดนโนดที่ใช้ในการทดสอบเมื่อลดความซ้ำซ้อนของอินพุต (ก) วิธี ก (ข) วิธี ข (ค) วิธี ค..... | 76 |
| รูปที่ 5.22 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ก) ในกรณีแปรค่า $E(X)$ | 77 |
| รูปที่ 5.23 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ก) ในกรณีแปรค่า $E(S)$ | 77 |
| รูปที่ 5.24 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ข) ในกรณีแปรค่า $E(X)$ | 78 |
| รูปที่ 5.25 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ข) ในกรณีแปรค่า $E(S)$ | 78 |
| รูปที่ 5.26 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ค) ในกรณีแปรค่า $E(X)$ | 79 |
| รูปที่ 5.27 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุม พืชนิรอลเน็ตเวิร์กกับตัวควบคุมพืชนีลจิกเมื่อลดความซ้ำซ้อนของ อินพุตวิธี (ค) ในกรณีแปรค่า $E(S)$ | 79 |
| รูปที่ 5.28 ความสามารถในการตรวจวัดโหลดเกินเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด... | 80 |
| รูปที่ 5.29 ความสามารถในการตรวจวัดโหลดเกินเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด... | 81 |

สารบัญรูป (ต่อ)

| | |
|---|----|
| รูปที่ 5.30 การตอบสนองของกลไกตรวจหาฟิสิกเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด... | 82 |
| รูปที่ 5.31 การตอบสนองของกลไกตรวจหาฟิสิกเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด... | 82 |
| รูปที่ 5.32 ความสามารถในการตรวจวัดโหลดเกินเมื่อไม่ลดความซ้ำซ้อนของอินพุต (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 84 |
| รูปที่ 5.33 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุต วิธี ก (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 85 |
| รูปที่ 5.34 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุต วิธี ข (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 86 |
| รูปที่ 5.35 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุต วิธี ค (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 86 |
| รูปที่ 5.36 แบบจำลองของนิรโทษคดีเวิร์กแบบขนาน..... | 87 |
| รูปที่ 5.37 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 2 ชุด (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 88 |
| รูปที่ 5.38 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 3 ชุด (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 88 |
| รูปที่ 5.39 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 4 ชุด (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 88 |
| รูปที่ 5.40 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 5 ชุด (ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด. | 89 |
| รูปที่ 5.41 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด..... | 90 |
| รูปที่ 5.42 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด..... | 90 |
| รูปที่ 5.43 เปรียบเทียบการตอบสนองของกลไกตรวจหาฟิสิกเมื่อแปรค่า $E(X)$ ของ แหล่งกำเนิด..... | 91 |
| รูปที่ 5.44 เปรียบเทียบการตอบสนองของกลไกตรวจหาฟิสิกเมื่อแปรค่า $E(S)$ ของ แหล่งกำเนิด..... | 92 |

สารบัญตาราง

| | | |
|---------------|---|----|
| ตารางที่ 3.1 | พารามิเตอร์ของแบบจำลองแหล่งกำเนิดที่ใช้ในการทดสอบ..... | 16 |
| ตารางที่ 3.2 | กฎพื้นฐานพีชคณิตสำหรับ inference ของตัวควบคุมพีชคณิตลจิก ในกลไกหน้าต่างพีชคณิตลจิก..... | 19 |
| ตารางที่ 3.3 | พารามิเตอร์ของกลไกถึงรั้วพีชคณิตนิรอลเน็ตเวิร์ก..... | 21 |
| ตารางที่ 3.4 | กฎพื้นฐานพีชคณิตสำหรับ inference ของตัวควบคุมพีชคณิตนิรอลเน็ตเวิร์ก ในกลไกถึงรั้วพีชคณิตนิรอลเน็ตเวิร์ก..... | 24 |
| ตารางที่ 4.1 | อัตราส่วนการสูญหายของเซลล์ที่เกิดจากกลไกตรวจทราฟฟิก ที่ σ ต่างๆ เมื่อแปรค่า $E(X)$ | 35 |
| ตารางที่ 4.2 | อัตราส่วนการสูญหายของเซลล์ที่เกิดจากกลไกตรวจทราฟฟิก ที่ σ ต่างๆ เมื่อแปรค่า $E(S)$ | 38 |
| ตารางที่ 4.3 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของกลไก ตรวจทราฟฟิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1..... | 41 |
| ตารางที่ 4.4 | เปรียบเทียบการตอบสนองเริ่มต้นของกลไกตรวจทราฟฟิก เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1..... | 43 |
| ตารางที่ 4.5 | อัตราส่วนการสูญหายของเซลล์ที่เกิดจากกลไกตรวจทราฟฟิก ที่ σ ต่างๆ เมื่อแปรค่า $E(X)$ | 45 |
| ตารางที่ 4.6 | อัตราส่วนการสูญหายของเซลล์ที่เกิดจากกลไกตรวจทราฟฟิก ที่ σ ต่างๆ เมื่อแปรค่า $E(S)$ | 48 |
| ตารางที่ 4.7 | เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของกลไก ตรวจทราฟฟิก เมื่อทดสอบกับแบบจำลองแบบที่ 2..... | 51 |
| ตารางที่ 4.8 | เปรียบเทียบการตอบสนองเริ่มต้นของกลไกตรวจทราฟฟิก เมื่อทดสอบ กับแบบจำลองแหล่งกำเนิดแบบที่ 2..... | 53 |
| ตารางที่ 4.9 | เปรียบเทียบจำนวนโทเค็นเฉลี่ยในบ่อโทเค็นระหว่างกลไกตรวจ ทราฟฟิกที่มีขนาดของบ่อโทเค็นต่างๆ..... | 55 |
| ตารางที่ 4.10 | เปรียบเทียบอัตราकारอนุญาตเซลล์ให้ผ่านเข้าสู่โครงข่าย ณ โหลดปกติ..... | 57 |
| ตารางที่ 5.1 | เปรียบเทียบความสามารถในการเลือกของกลไกตรวจทราฟฟิก เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1..... | 81 |

สารบัญตาราง (ต่อ)

| | | |
|--------------|---|----|
| ตารางที่ 5.2 | เปรียบเทียบการตอบสนองเริ่มต้นของกลไกถังรั่วพีซีนิวรอลเน็ตเวิร์ก กับถังรั่วพีซีลोजิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1..... | 83 |
| ตารางที่ 5.3 | เปรียบเทียบเวลาประวิงเฉลี่ยต่อ 1 เซลล์ที่เกิดจากกลไกตรวจหาฟีก เมื่อใช้ตัวควบคุมต่างกัน..... | 83 |
| ตารางที่ 5.4 | เปรียบเทียบค่าผิดพลาดในการฝึกสำหรับการแบ่งชุดฝึกเป็น 2, 3, 4 และ 5 ชุด..... | 89 |
| ตารางที่ 5.5 | เปรียบเทียบความสามารถในการเลือกของกลไกตรวจหาฟีก เมื่อทดสอบกับแบบจำลองแบบที่ 2..... | 91 |
| ตารางที่ 5.6 | เปรียบเทียบการตอบสนองเริ่มต้นของกลไกถังรั่วพีซีนิวรอลเน็ตเวิร์ก กับถังรั่วพีซีลोजิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2..... | 92 |
| ตารางที่ 5.7 | เปรียบเทียบเวลาประวิงเฉลี่ยที่เกิดจากกลไกตรวจหาฟีก เมื่อใช้ตัวควบคุมต่างกัน..... | 93 |
| ตารางที่ 5.8 | เปรียบเทียบเวลาในการประมวลผลเฉลี่ยต่อครั้งของตัวควบคุม พีซีลोजิกกับตัวควบคุมพีซีนิวรอลเน็ตเวิร์ก..... | 94 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การควบคุมพารามิเตอร์การใช้ (Usage Parameter Control: UPC) หรือตัวตรวจทราฟฟิก (Traffic Policer) ในโครงข่ายเอทีเอ็มมีหน้าที่ป้องกันไม่ให้เกิดการต่อที่ดำเนินอยู่ละเมิดข้อตกลงที่ทำให้ไว้กับการควบคุมการอนุญาตการเรียก (Call Admission Control: CAC) ในเฟสเขตอัปการต่อเพื่อป้องกันความคับคั่ง (Congestion) และการละเมิดคุณภาพของบริการ (Quality of Service: QoS) ของการต่ออื่นๆ โดยจะมีมาตรการลงโทษผู้ใช้ที่กระทำผิด ตัวตรวจทราฟฟิกจึงทำหน้าที่เปรียบเสมือนตำรวจของโครงข่ายซึ่งมีความสำคัญยิ่ง มีผู้เสนอวิธีการหรือกลไกมากมายเพื่อทำหน้าที่ตัวตรวจทราฟฟิกในโครงข่ายเอทีเอ็ม โดยที่เดิมประกอบด้วยแนวทางหลัก 2 แนวทาง คือ กลไกถังรั่ว (Leaky Bucket Mechanisms) และ กลไกหน้าต่าง (Window Mechanisms) แต่ละแนวทางยังแบ่งย่อยได้อีกหลายวิธี

เนื่องจากวัตถุประสงค์หลักของการตรวจ (police) ทราฟฟิก คือเพื่อตรวจขบวนเซลล์จากแหล่งกำเนิดไม่ให้เกินตัวอธิบายทราฟฟิก (Traffic Descriptor) ได้แก่ อัตราเซลล์ค่ายอด (Peak Cell Rate) อัตราเซลล์เฉลี่ย (Mean Cell Rate) และระยะเวลาเบิร์สต์สูงสุด (Maximum Burst Duration) การตรวจทราฟฟิกที่บริบูรณ์จึงต้องสามารถควบคุมพารามิเตอร์ทุกตัวได้ แต่ในทางปฏิบัติ การตรวจอัตราเซลล์ค่ายอดกับอัตราเซลล์เฉลี่ยของทราฟฟิกแบบเบิร์สต์ (bursty traffic) พร้อมๆ กันด้วยกลไกดั้งเดิมดังกล่าวทำได้ยาก ทั้งนี้เป็นเพราะการปรับพารามิเตอร์ในโครงสร้างของกลไกมีความขัดแย้งกัน เช่น หากต้องการให้ถังรั่วตรวจอัตราเซลล์ค่ายอด เราควรกำหนดให้ขนาดตัวนับมีค่าน้อย และอัตรารั่วใกล้เคียงกับอัตราเซลล์ค่ายอด ในขณะที่การตรวจอัตราเซลล์เฉลี่ยควรให้ขนาดตัวนับมีค่ามาก และอัตรารั่วใกล้เคียงกับอัตราเซลล์เฉลี่ย ความขัดแย้งนี้ทำให้ไม่สามารถใช้กลไกเพียงกลไกเดียวเพื่อตรวจทราฟฟิกอย่างบริบูรณ์ได้ ในงานวิจัยต่อมา จึงมีการจำแนกการตรวจทราฟฟิกออกเป็น 2 แนวทาง คือการตรวจอัตราเซลล์ค่ายอด (Peak Cell Rate Policing) และการตรวจอัตราเซลล์เฉลี่ย (Mean Cell Rate Policing) แม้ว่าการตรวจอัตราเซลล์ค่ายอดนั้นจะง่าย แต่การตรวจอัตราเซลล์ค่ายอดเพียงอย่างเดียวไม่เหมาะสมกับโครงข่ายที่มีโหลดมาก เนื่องจากจะทำให้จำนวนผู้ใช้ที่โครงข่ายรองรับได้ลดลง ในขณะที่การตรวจอัตราเซลล์เฉลี่ยทำได้ยากกว่า แต่สามารถใช้ทรัพยากรโครงข่ายอย่างมีประสิทธิภาพ ดังนั้นการพัฒนาคำ

ตรวจอัตราเซลล์เฉลี่ยเพื่อทำหน้าที่เดียวๆ หรือควบคู่กับตัวตรวจอัตราเซลล์ค่ายอดสำหรับ UPC จึงเป็นแนวทางที่เหมาะสมกว่า

งานวิจัยที่ผ่านมา [1] ทดสอบเปรียบเทียบประสิทธิภาพในการตรวจอัตราเซลล์เฉลี่ยของ กลไกถึงรั้วกับกลไกหน้าต่างกระโดด (Jumping Window), หน้าต่างเคลื่อนที่ (Moving Window), หน้าต่างกระโดดแบบจุดชนวน (Triggered Jumping Window) และกลไกเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอ็กซ์โพเนนเชียล (Exponentially Weighted Moving Average: EWMA) พบว่า ถึงรั้ว และ EWMA เป็นกลไกที่มีพฤติกรรมระยะยาวเหมาะสมที่สุดสำหรับ UPC กล่าวคือสามารถทนต่อการกระเพื่อมทางสถิติระยะสั้น (Short-term Statistical Fluctuations) ของแหล่งกำเนิดทราฟฟิกได้โดยไม่ทำให้อัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio : CLR) สูงจนเกินไป เมื่อเปรียบเทียบกับกลไกอื่นๆ แต่ในแง่พฤติกรรมพลวัต EWMA มีการตอบสนองต่อการละเมิดซ้ำที่เร็วที่สุด และช้ากว่าถึงรั้วอย่างมาก จึงทำให้กลไกถึงรั้วเป็นกลไกตรวจทราฟฟิกที่ได้รับความนิยมที่สุดในปัจจุบัน อย่างไรก็ตาม แม้ว่าถึงรั้วจะมีจุดเด่นคือมีโครงสร้างที่ไม่ซับซ้อน แต่ [2] ได้วิเคราะห์ประสิทธิภาพของถึงรั้วแบบดั้งเดิมโดยใช้วิธีการทางคณิตศาสตร์ และสรุปเป็นสูตรประมาณค่าอัตราการสูญหายของเซลล์ที่เกิดจากถึงรั้ว สูตรดังกล่าวแสดงให้เห็นว่าหากนำถึงรั้วไปใช้ตรวจอัตราเซลล์เฉลี่ยจะต้องการตัวนับ (counter) ขนาดใหญ่มาก ทำให้การตอบสนองของซ้ำ และเมื่อลดขนาดตัวนับลงจะทำให้ CLR สูง จึงเหมาะที่จะนำไปใช้ตรวจอัตราเซลล์ค่ายอดเท่านั้น

ข้อจำกัดของการนำกลไกตรวจทราฟฟิกแบบดั้งเดิมไปตรวจทราฟฟิกแบบเบิสต์ คือ โครงสร้างของกลไกมีลักษณะสถิต (static) จึงไม่สามารถปรับตัวต่อการกระเพื่อมของทราฟฟิกที่เข้ามา เพื่อแก้ไขลักษณะดังกล่าว มีงานวิจัยที่เสนอให้นำปัญญาประดิษฐ์ (Artificial Intelligence: AI) มาประยุกต์ใช้กับ UPC [3] เสนอให้นำนิรอลเน็ตเวิร์ก (Neural Network) 2 ชุดมาฝึกเพื่อใช้ทำนายกระบวนการนับ (Count Process) ของเซลล์ที่จะมาถึงในหน้าต่างต่อไป และมีชุดข้อมูลที่ใช้ในการฝึกเป็นกระบวนการนับในอดีตของทราฟฟิก นิรอลเน็ตเวิร์กแต่ละชุดถูกฝึกด้วยอินพุตแตกต่างกัน ชุดแรกถูกฝึกด้วยแบบรูป (pattern) ของทราฟฟิกที่ไม่ละเมิดข้อตกลง ในขณะที่อีกชุดหนึ่งถูกฝึกด้วยแบบรูปของทราฟฟิกที่เป็นไปได้ทั้งหมดไม่ว่าจะละเมิดหรือไม่ก็ตาม ผลต่างของนิรอลเน็ตเวิร์กทั้งสองชุดใช้ตัดสินการละทิ้งเซลล์ วิธีนี้มีจุดเด่นคือกลไกสามารถตรวจทราฟฟิกได้แบบเวลาจริง แต่มีข้อเสียคือ ปัญหาในการฝึกนิรอลเน็ตเวิร์ก นอกจากจะต้องใช้เวลานานแล้ว จำนวนชุดฝึก (training set) ของอินพุตที่มีไม่จำกัดและแบบรูปของข้อมูลที่แตกต่างกันมาก ทำให้การเลือกวิธีฝึกและชุดฝึกมีผลอย่างมากต่อการจำของนิรอลเน็ตเวิร์ก การเลือกผิดอาจไม่สามารถทำให้นิรอลเน็ตเวิร์กจำแบบรูปของทราฟฟิกได้เลย และต้องมีการฝึกใหม่ทุกครั้งหากลักษณะของแหล่งกำเนิดที่ต้องการควบคุมเปลี่ยนแปลงไปเพียงเล็กน้อย ทำให้วิธีนี้มีความยุ่งยากเกินไปและไม่ยืดหยุ่น

เครื่องมืออีกอย่างหนึ่งที่ถูกเสนอให้นำมาใช้กับ UPC คือ ฟัชซีลอจิก (Fuzzy Logic) ครั้งแรก [4] มีการเสนอตัวตรวจหากราฟฟิคที่ใช้กลไกถังร่วนแบบโทเค็น โดยมีฟัชซีลอจิกทำหน้าที่ควบคุมจำนวนโทเค็นส่วนเกินที่ระบบยอมให้มีได้นอกเหนือไปจากโทเค็นปกติที่ผลิตด้วยอัตราคงที่เท่ากับอัตราเซลล์เฉลี่ยที่ตกลงไว้ เซลล์ที่ได้รับโทเค็นส่วนเกินนี้จะถูกตั้งค่าบิตลำดับความสำคัญการสูญหาย (Cell Loss Priority Bit) เป็น 1 แม้ว่าวิธีนี้จะลดอัตราส่วนการสูญหายของเซลล์ลงได้เมื่อเปรียบเทียบกับกลไกถังร่วนแบบดั้งเดิม แต่วิธีนี้ไม่ใช่การควบคุมอัตราเซลล์เฉลี่ยของแหล่งกำเนิดอย่างแท้จริง เนื่องจากตัวแปรที่ควบคุมเป็นจำนวนโทเค็นส่วนเกิน ในขณะที่โทเค็นปกติยังคงถูกผลิตด้วยอัตราเท่ากับอัตราเซลล์เฉลี่ยที่ตกลงไว้ นอกจากนี้งานวิจัยดังกล่าวมิได้มีการประเมิน ประสิทธิภาพของกลไกในด้านความสามารถในการเลือก (Selectivity) และการตอบสนอง (Responsiveness) แต่อย่างใด [5-6] เสนอตัวควบคุมฟัชซีลอจิกบนโครงสร้างพื้นฐานของกลไกหน้าต่าง มีเอาต์พุตคือขีดจำกัดของหน้าต่าง (window limit) ต่อไป วิธีนี้ทำให้การตรวจวัดโหลดเกินของระบบเข้าใกล้เส้นโค้งการละเมิดอุดมคติ แต่การใช้โครงสร้างพื้นฐานเป็นกลไกหน้าต่างทำให้ตัวควบคุมมีการตอบสนองต่อโหลดเกินในช่วงเริ่มต้นช้ากว่ากลไกถังร่วนแบบดั้งเดิมมาก

วิทยานิพนธ์นี้เสนอตัวตรวจหากราฟฟิคที่ใช้กลไกถังร่วนแบบโทเค็น และใช้ฟัชซีนิวรอลเน็ตเวิร์ก [7] ควบคุมอัตราการผลิตโทเค็นของตัวกำเนิดโทเค็น โดยมีวัตถุประสงค์เพื่อตรวจอัตราเซลล์เฉลี่ยของกราฟฟิค โดยลดอัตราส่วนการสูญหายของเซลล์และปรับปรุงประสิทธิภาพในการตรวจวัดโหลดเกินเมื่อเปรียบเทียบกับกลไกถังร่วนแบบไม่มีบัฟเฟอร์ [1] พร้อมกันนั้นจะต้องมีพฤติกรรมพลวัตไม่ด้อยกว่ากลไกหน้าต่างฟัชซีลอจิก [6]

1.2 วัตถุประสงค์

1. พัฒนาตัวตรวจอัตราเซลล์เฉลี่ยของแหล่งกำเนิดเดี่ยว เพื่อป้องกันไม่ให้เกิดความคับคั่งขึ้นในโครงข่ายเอทีเอ็ม และสามารถใช้ทรัพยากรอย่างมีประสิทธิภาพ
2. ปรับปรุงประสิทธิภาพในการตรวจวัดโหลดเกิน (Overload Detection) และลดอัตราส่วนการสูญหายของเซลล์ที่เกิดจากตัวตรวจหากราฟฟิค ในการควบคุมพารามิเตอร์การใช้ เมื่อเปรียบเทียบกับกลไกถังร่วน และปรับปรุงการตอบสนอง (Responsiveness) ให้รวดเร็วกว่ากลไกหน้าต่างฟัชซีลอจิก
3. ลดความซับซ้อนและยุ่งยากในขั้นตอน inference ของฟัชซีลอจิก โดยใช้นิวรอลเน็ตเวิร์กแบบป้อนไปหน้า (Feedforward Neural Networks) หาความสัมพันธ์ระหว่างอัตราการกำเนิดโทเค็นของตัวตรวจหากราฟฟิค (กำหนดเป็นเอาต์พุต) กับตัวแปรที่

แสดงคุณสมบัติของทราฟฟิก (กำหนดเป็นอินพุต) และใช้ backpropagation algorithm เพื่อให้ได้เอาต์พุตใกล้เคียงกับการใช้พีชชีลอจิกมากที่สุด

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์

1. พัฒนาตัวตรวจทราฟฟิกที่มีโครงสร้างพื้นฐานเป็นกังว้แบบโทเค็น โดยใช้พีชชีลอจิกและนิรอลเน็ตเวิร์กควบคุมอัตราการทำเนิดโทเค็น
2. เปรียบเทียบวิธีที่เสนอกับกลไกกังว้ [1] และกลไกหน้าต่งพีชชีลอจิก [6] ในแง่ความสามารถในการตรวจวัดไหลดเกินและการตอบสนอง

1.4 ขั้นตอนและวิธีดำเนินงาน

1. ศึกษาทฤษฎีของโครงข่ายเอทีเอ็ม
2. ศึกษากลไกการตรวจทราฟฟิกสำหรับ UPC แบบต่งๆ
3. ศึกษาอัลกอริทึมของพีชชีลอจิกและนิรอลเน็ตเวิร์ก
4. ศึกษาการประยุกต์ใช้งานพีชชีลอจิก นิรอลเน็ตเวิร์ก และพีชชีนิรอลเน็ตเวิร์กในงานที่เกี่ยวข้อง
5. ออกแบบอัลกอริทึมเพื่อปรับปรุงประสิทธิภาพของการตรวจทราฟฟิกโดยใช้พีชชีลอจิกบนโครงสร้งกังว้แบบโทเค็น
6. ทดสอบอัลกอริทึมกับแบบจำลองแหล่งกำเนิดเดี่ยวชนิด packet voice และเปรียบเทียบประสิทธิภาพของวิธีที่เสนอกับกลไกกังว้แบบต่งเดิม และกลไกหน้าต่งพีชชีลอจิก
7. ประเมินและสรุปผลการวิจัย
8. เขียนวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ปรับปรุงกลไกการตรวจทราฟฟิกให้เข้าใกล้การตรวจอัตราเซลล์เฉลี่ยที่แท้จริง โดยมีประสิทธิภาพทั้งในแง่ความสามารถในการเลือกและการตอบสนองพร้อมๆ กัน
2. สามารถนำความรู้ที่ได้จากการศึกษาอัลกอริทึมพีชชีลอจิก นิรอลเน็ตเวิร์ก และ

พีชชีนิรอรลเนตเวริกไปประกยุกตีใช้กับงานวิจัยอื่นๆ ของระบบสื่อสารได้

ในบทที่ 2 จะกล่าวถึงการควบคุมกราฟฟิกในโครงข่ายเอทีเอ็ม รวมถึงกลไกตรวจกราฟฟิก
ดั้งเดิมแบบต่างๆ บทที่ 3 เสนอแบบจำลองของแหล่งกำเนิดและตัวตรวจกราฟฟิกที่เสนอรวมถึง
ตัวตรวจกราฟฟิกที่ใช้เปรียบเทียบ ผลการทดสอบประสิทธิภาพของตัวตรวจเมื่อใช้พีชชีลอจิก
พร้อมทั้งวิเคราะห์ผลการทดสอบแสดงไว้ในบทที่ 4 บทที่ 5 แสดงผลการทดสอบการนำนิรอรล
เนตเวริกมาใช้ในตัวตรวจ และบทที่ 6 สรุปผลการทดสอบพร้อมกับข้อดี ข้อเสียและข้อเสนอแนะ



บทที่ 2

การควบคุมทราฟฟิกในโครงข่ายเอทีเอ็ม

2.1 กล่าวนำ

เอทีเอ็มเป็นแบบวิธีถ่ายโอนข้อมูลแบบแพ็กเก็ตวิธีหนึ่ง โดยหน่วยของข้อมูลในการถ่ายโอนจะมีความยาวคงที่ที่แน่นอน เรียกว่า "เซลล์" แต่ละเซลล์ประกอบด้วยฟิลด์ข่าวสารและเฮดเดอร์ที่ระบุช่องสัญญาณเสมือน (Virtual Channel) เพื่อให้การจัดเส้นทางเป็นไปอย่างถูกต้อง รวมขนาด 53 ไบต์ เอทีเอ็มถูกเสนอให้เป็นเทคโนโลยีสำหรับโครงข่ายดิจิทัลบริการรวมแถบกว้าง (B-ISDN) โดยมีข้อได้เปรียบที่เป็นการถ่ายโอนข่าวสารแบบวิธีแพ็กเก็ต (packet mode) จึงประหยัดทรัพยากรโครงข่ายเพราะจะมีการสร้างเซลล์เฉพาะเมื่อมีข่าวสารจะถ่ายโอนเท่านั้น พร้อมๆ กับแบบวิธีวงจร (circuit mode) ที่ทำให้การถ่ายโอนรวดเร็ว เนื่องจากมีการเชื่อมต่อ

ในโครงข่ายเอทีเอ็ม จะไม่มีการจัดสรรเวลา (time slot) ที่แน่นอนให้ช่องสัญญาณ โดยฟังก์ชันการเคลื่อนย้ายจะมีโครงสร้างเป็นลำดับขั้น ตัวกลางในการส่ง (transmission medium) จะให้บริการในการเคลื่อนย้ายต่อวิถีเสมือน (virtual path: VP) และวิถีเสมือนจะให้บริการช่องสัญญาณเสมือน (virtual channel: VC) ในแ่งโพรโทคอล หน้าที่ที่ถูกสนับสนุนโดยตัวกลางในการส่งจะอยู่ในชั้นกายภาพ และหน้าที่ที่สนับสนุนโดยวิถีเสมือนและช่องสัญญาณเสมือนจะอยู่ในชั้นเอทีเอ็ม

ช่องสัญญาณเสมือนเป็นศัพท์ทั่วไปที่ใช้อธิบายการสื่อสารทิศทางเดียวในการเคลื่อนย้ายเซลล์เอทีเอ็ม โดยที่เฮดเดอร์ของแต่ละเซลล์จะมีตัวระบุช่องสัญญาณเสมือน (VC Identifier: VCI) และตัวระบุวิถีเสมือน (VP Identifier: VPI) และวิถีเสมือนแต่ละวิถีจะมีค่าตัวระบุช่องสัญญาณเสมือนที่ตรงกันได้ ค่าตัวระบุช่องสัญญาณเสมือนและตัวระบุวิถีเสมือนในเฮดเดอร์ของเซลล์จะถูกเปลี่ยนทุกครั้งตามตารางการแปลงที่โหนดการสวิตช์ ข่ายเชื่อมโยงช่องสัญญาณเสมือน (VC link) มีความสามารถในการเคลื่อนย้ายเซลล์เอทีเอ็มแบบทิศทางเดียวระหว่างโหนดการสวิตช์ตัวระบุช่องสัญญาณเสมือนที่อยู่ติดกัน 2 โหนด และข่ายเชื่อมโยงช่องสัญญาณเสมือนที่ต่อเนื่องกันจะประกอบขึ้นเป็นการต่อช่องสัญญาณเสมือน (Virtual Channel Connection: VCC) จุดสิ้นสุดของการต่อช่องสัญญาณเสมือนคือจุดที่ฟิลด์ข่าวสารของเซลล์ถูกแลกเปลี่ยนระหว่างชั้นเอทีเอ็มกับผู้ให้บริการของชั้นเอทีเอ็มซึ่งก็คือชั้นการปรับตัวเอทีเอ็ม (ATM Adaptation Layer) นั่นเอง

วิถีเสมือน คือกลุ่มของข่ายเชื่อมโยงช่องสัญญาณเสมือน ที่โหนดการสวิตช์ช่องสัญญาณเสมือน ทั้งค่าตัวระบุช่องสัญญาณเสมือนและตัวระบุวิถีเสมือนจะต้องถูกเขียนใหม่ ในขณะที่โหนด

การสวิตช์วิถีเสมือนจะมีการเขียนเฉพาะค่าตัวระบุวิถีเสมือนใหม่เท่านั้น และในทำนองเดียวกัน ข่ายเชื่อมโยงวิถีเสมือนที่ต่อกันจะประกอบขึ้นเป็นการต่อวิถีเสมือน (Virtual Path Connection: VPC) ชั้นกายภาพจะให้บริการแก่ชั้นเอทีเอ็ม โดยทำหน้าที่ส่งเซลล์เอทีเอ็มที่ถูกบรรจุในสัมภาระบรรทุกวิถีการส่ง (Transmission Path Payload) ตัวอย่างของชั้นกายภาพที่รองรับเทคโนโลยีเอทีเอ็มคือ Synchronous Digital Hierarchy (SDH)

ด้วยแนวคิดในการส่งผ่านวิถีเสมือนและวงจรเสมือน เราจึงสามารถกำจัดรูปแบบช่องสัญญาณที่มีอัตราการส่งคงที่ และลำดับชั้นของวิถีแบบดิจิทัล (Digital Path Hierarchy) ไปได้ ทำให้การจัดการโครงข่ายมีความยืดหยุ่นมากขึ้น อย่างไรก็ตาม ความยืดหยุ่นที่เพิ่มขึ้นทำให้การจัดการทรัพยากรมีได้หลากหลายและมีความท้าทายในการแสวงหาแนวทางที่เหมาะสมที่สุด แนวทางในการควบคุมความคับคั่งในโครงข่ายประกอบด้วย

1. ควบคุมการเข้าถึงของผู้เช่า (Subscriber) สูโครงข่าย แนวทางการควบคุมการอนุญาตการเรียกป้องกันโครงข่ายจากโหลดส่วนเกิน และทำให้มั่นใจว่าบริการโดยเฉลี่ยที่ผู้เช่าได้รับนั้นไม่เกินไปกว่าทรัพยากรโครงข่ายที่มีอยู่
2. ป้องกันคุณภาพของบริการของผู้ใช้จากการกระเพื่อมแบบพ่นสุ่ม (Stochastic Fluctuation) จากโหลดของผู้เช่าบางราย และสามารถแจ้งให้ผู้เช่านั้นปรับอัตราส่งได้
3. ทำให้มั่นใจได้ว่าแต่ละช่องสัญญาณเสมือนหรือวิถีเสมือน ไม่เกินตัวอธิบายทราฟฟิกและทรัพยากรที่ได้รับจัดสรร ไม่ว่าจะเป็นในแง่แบนด์วิดท์หรือหน่วยความจำ

การป้องกันความคับคั่งในโครงข่ายและทำให้การใช้ทรัพยากรเป็นไปอย่างมีประสิทธิภาพ ต้องอาศัยการควบคุมทราฟฟิกที่สำคัญ 2 กลไกหลักได้แก่ การควบคุมการอนุญาตการเรียก (CAC) และการควบคุมพารามิเตอร์การใช้ (UPC)

2.2 การควบคุมการอนุญาตการเรียก (Call Admission Control: CAC)

CAC คือเซตของการกระทำอันประกอบด้วย การตัดสินใจการเซตอัปของการต่อใหม่ ซึ่งขึ้นอยู่กับลักษณะทราฟฟิกของการต่อที่คาดการณ์ว่าจะเกิดขึ้นหากอนุมัติให้มีการต่อนั้น, คุณภาพของบริการที่ร้องขอ (Requested QoS) และโหลดของโครงข่ายในขณะนั้น โดยลักษณะทราฟฟิกของการต่อใหม่คาดการณ์ได้จากตัวอธิบายทราฟฟิกของแหล่งกำเนิด (Source Traffic Descriptor) หากการร้องขอการต่อได้รับการตอบรับ จะมีการจัดสรรแบนด์วิดท์ให้ในจำนวนที่ไม่ชัดเจน (implicitly bandwidth allocation) ในโครงข่ายเอทีเอ็มที่มีความเร็วในการส่งสูงและเวลาประวิงที่ยาวนานในโครงข่าย ทำให้กลไกควบคุมแบบปฏิกิริยา (Reactive Control Mechanism) ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ CAC ซึ่งเป็นกลไกควบคุมแบบป้องกัน (Preventive

Control Mechanism) จึงเป็นกลไกหลักที่มีความสำคัญสูงสุดกลไกหนึ่ง ในขณะที่ในโครงข่ายแบบวิธีการเคลื่อนย้ายซิงโครนัส (Synchronous Transfer Mode Network) แบนด์วิดท์ของแต่ละช่องสัญญาณจะมีค่าแน่นอน โดยระยะเวลาของช่องสัญญาณแต่ละช่องจะถูกจัดสรรเป็นวัฏจักร ดังนั้น การตัดสินใจรับการต่อใหม่สามารถพิจารณาได้อย่างชัดเจนและสะดวก แต่ในโครงข่ายเอทีเอ็มทีไม่มีการจัดสรรแบนด์วิดท์ช่องสัญญาณเสมือนในแต่ละช่องโดยชัดเจน เพราะช่องสัญญาณเสมือนจะถูกมัลติเพล็กซ์ทางสถิติ (Statistical Multiplexing) การตัดสินใจรับการต่อใหม่จึงขึ้นกับการคาดการณ์ลักษณะทราฟฟิกของช่องสัญญาณเสมือนใหม่ และพิจารณามลกระทบต่อคุณภาพของบริการของการต่อที่ดำเนินอยู่ในขณะนั้น

เนื่องจากโครงข่ายตัดสินใจการต่อใหม่ภายใต้สมมติฐานว่า ผู้ใช้จะปฏิบัติตามตัวอธิบายทราฟฟิกของแหล่งกำเนิดอย่างเคร่งครัด ดังนั้นหากไม่เป็นเช่นนั้นจะทำให้โครงข่ายเกิดความคับคั่งได้ ด้วยเหตุนี้จึงต้องมีกลไกที่ป้องกันเหตุการณ์ดังกล่าวคือ UPC

2.3 การควบคุมพารามิเตอร์การใช้ (Usage Parameter Control: UPC)

UPC หรือ ตัวตรวจทราฟฟิก (Traffic Policer) คือกลไกในการเฝ้าตรวจขบวนเซลล์ที่จะเข้าสู่โครงข่าย ณ จุดต่อร่วมผู้ใช้โครงข่าย (User-to-Network Interface: UNI) เพื่อให้มั่นใจว่าช่องสัญญาณเสมือนหรือวิธีเสมือนใดๆ จะไม่ละเมิดข้อตกลงระหว่างผู้ใช้กับโครงข่าย และลงโทษทราฟฟิกที่ละเมิดข้อตกลงด้วยมาตรการต่างๆ ได้แก่

-การละทิ้งเซลล์ UPC จะละทิ้งทราฟฟิกส่วนเกินจากช่องสัญญาณเสมือนหรือวิธีเสมือนใดๆ ก่อนเข้าสู่โครงข่าย

-การแทกเซลล์ UPC จะทำเครื่องหมายเซลล์ของทราฟฟิกส่วนเกินจากช่องสัญญาณเสมือนหรือวิธีเสมือนใดๆ โดยการตั้งค่าลำดับความสำคัญการสูญหายของเซลล์ (Cell Loss Priority: CLP) จาก 0 เป็น 1 หากเกิดความคับคั่ง ณ จุดใดในโครงข่าย เซลล์ที่มีบิต CLP เป็น 1 จะถูกละทิ้งก่อนทันที แม้ว่าวิธีนี้เหมาะกับโครงข่ายที่ไม่ต้องการการควบคุมที่เข้มงวดมากนัก และเป็นการจูงใจผู้ใช้ให้เข้ามาใช้บริการ แต่การละทิ้งเซลล์ที่ถูกทำเครื่องหมายในภายหลัง อาจทำให้เกิดความไม่เป็นธรรมต่อช่องสัญญาณเสมือนแต่ละช่อง เนื่องจากการละทิ้งไม่ได้เป็นสัดส่วนกับการละเมิดของทราฟฟิกจากแต่ละช่อง

คุณลักษณะของ UPC ที่ต้องการคือ

1. ความสามารถในการเลือกสูง (High Selectivity) ในอุดมคติ UPC ควรตรวจวัดและลงโทษเฉพาะทราฟฟิกที่ละเมิดข้อตกลงเท่านั้น และโปร่งใส (transparent) ต่อทราฟฟิกที่ไม่ละเมิดข้อตกลง

2. การตอบสนองรวดเร็วต่อการกระทำผิด (High Responsiveness) เพื่อป้องกันไม่ให้เกิดการล้นของบัฟเฟอร์ขนาดเล็กในโครงข่าย
 3. นำมาใช้งานได้ง่าย (Simple Implementation) และ คุ่มค่าแก่การลงทุน
 4. ทำให้การใช้ทรัพยากรโครงข่ายเกิดประสิทธิผล (Efficient Network Resource Usage)
- กลไกดั้งเดิมที่ถูกละทิ้งให้ทำหน้าที่ตรวจวัดสำหรับ UPC แบ่งได้เป็น 2 ประเภทหลัก คือ กลไกถังรั่ว (Leaky Bucket: LB Mechanisms) และกลไกหน้าต่าง (Window Mechanisms)

2.3.1 กลไกถังรั่ว (Leaky Bucket Mechanisms)

2.3.1.1 กลไกถังรั่วแบบไม่มีบัฟเฟอร์ (Unbuffered Leaky Bucket Mechanism)

กลไกนี้เป็นอัลกอริทึมในการตรวจทราฟฟิกที่มีโครงสร้างดังแสดงในรูปที่ 2.1 พารามิเตอร์ที่กำหนดสมรรถนะของกลไกประกอบด้วยจุดเริ่มเปลี่ยนของตัวนับ (counter threshold) และ อัตรารั่ว (leak rate) ทุกครั้งที่แหล่งกำเนิดผลิตเซลล์ ตัวนับจะนับเพิ่มครั้งละหนึ่ง ขณะเดียวกันก็จะลดลงด้วยอัตราคงที่เท่ากับอัตรารั่ว เซลล์ที่มาถึงขณะที่ตัวนับนับถึงจุดเริ่มเปลี่ยนจะถูกทิ้ง

Butto et al. [2] วิเคราะห์หาสูตรประมาณอัตราส่วนการสูญหายของเซลล์ของกลไกถังรั่ว (P_L) เมื่อใช้แบบจำลองแหล่งกำเนิดแบบเปิด-ปิด ได้ดังสมการที่ (2.1)

$$P_L = \frac{b'}{b} \cdot \frac{a\lambda_1 - b'\lambda_2}{a\lambda_1 e^{-(\lambda_1/b' - \lambda_2/a)M} - b'\lambda_2} \quad (2.1)$$

โดย b คืออัตราบิตค่ายอดของแหล่งกำเนิด

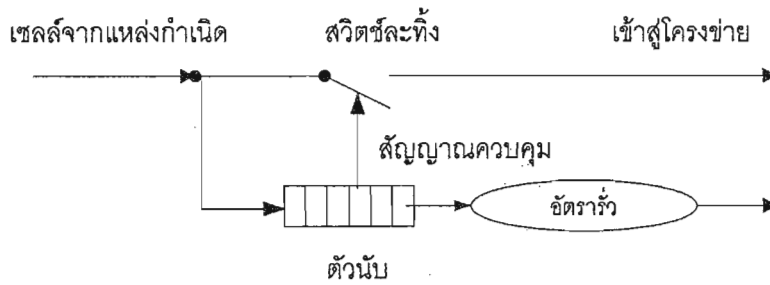
a คืออัตรารั่วของถังรั่ว (บิต/วินาที)

b' คืออัตราบิตของตัวนับที่เพิ่มขึ้นสุทธิในระยะเวลาเบิรสต์ มีค่าเท่ากับ $b - a$

λ_1, λ_2 คือส่วนกลับของระยะเวลาเบิรสต์เฉลี่ยและส่วนกลับของระยะเวลาเงียบเฉลี่ย

M คือขนาดของตัวนับ

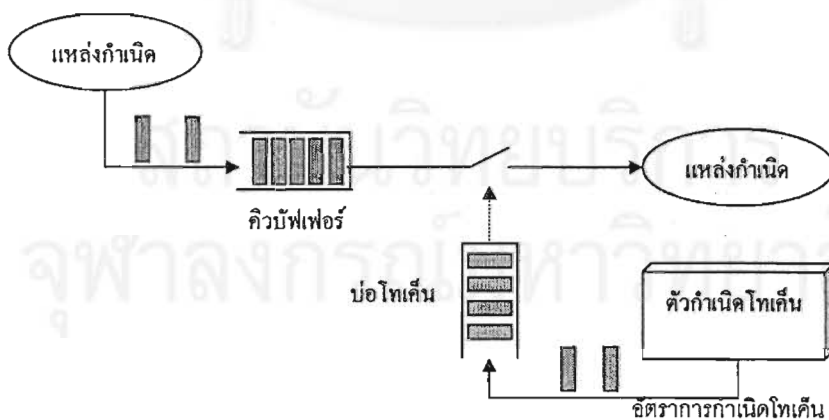
กลไกนี้มีประสิทธิภาพดีเมื่อใช้ในการตรวจอัตราเซลล์สูงสุด หรือ เมื่อ $a \approx b$ แต่เมื่อใช้ตรวจอัตราเซลล์เฉลี่ยของทราฟฟิก (Mean Cell Rate: MCR) จะทำให้อัตราส่วนการสูญหายของเซลล์ (Cell Loss Ratio: CLR) สูง การแก้ไขปัญหา CLR สูง มี 2 วิธีคือ การเพิ่มอัตราการรั่วด้วย overdimension factor (C) หรือการเพิ่มจุดเริ่มเปลี่ยนของตัวนับ วิธีแรกมีข้อเสียคือ จะทำให้ขนาดของแบนด์วิดท์ที่ต้องใช้เพิ่มขึ้นเป็นการสิ้นเปลืองทรัพยากร ส่วนวิธีที่ 2 จะทำให้การตอบสนองต่อการละเมิดช้าลง



รูปที่ 2.1 กลไกถังรั่วแบบไม่มีบัพเฟอร์

2.3.1.2 กลไกถังรั่วแบบโทเค็นหรือกลไกถังรั่วแบบมีบัพเฟอร์ (Token Leaky Bucket Mechanism or Buffered Leaky Bucket Mechanism)

การแก้ไข้ปัญหา CLR สูงในกลไกถังรั่วแบบไม่มีบัพเฟอร์ เพื่อให้สามารถนำกลไกถังรั่วไปใช้ตรวจอัตราเซลล์เฉลี่ยอย่างมีประสิทธิภาพ ทำได้โดยการเพิ่มคิวบัพเฟอร์ด้านเข้า โดยมีโครงสร้างแสดงดังรูปที่ 2.2 เซลล์จะถูกเก็บไว้ในคิวบัพเฟอร์จนกว่าจะได้รับโทเค็นจากบ่อโทเค็นจึงสามารถผ่านเข้าไปยังโครงข่ายได้ โทเค็นจะถูกผลิตด้วยอัตราคงที่ค่าหนึ่งซึ่งเรียกว่าอัตราการกำเนิดโทเค็น (Token Generation Rate) การสูญหายจะเกิดขึ้นเมื่อคิวบัพเฟอร์ล้น ดังนั้นพารามิเตอร์ที่กำหนดสมรรถนะของกลไกคือ ขนาดของคิวบัพเฟอร์ด้านเข้า ขนาดของบ่อโทเค็น และอัตราการกำเนิดโทเค็น



รูปที่ 2.2 กลไกถังรั่วแบบโทเค็น

แม้ว่าคิวบัพเฟอร์จะทำให้ CLR ลดลงได้บ้าง แต่กลไกจะมีการตอบสนองช้าลง และหากต้องการให้ CLR ลดลงมาก จะต้องใช้บัพเฟอร์ขนาดใหญ่ทำให้มีปัญหาล่าช้า

2.3.2 กลไกหน้าต่างต่าง (Window Mechanisms)

กลไกนี้มีหลักการคือ นับจำนวนเซลล์ที่จะเข้าสู่โครงข่ายในคาบเวลาแต่ละคาบ (period) หรือหน้าต่าง (window) แต่ละหน้าต่างไม่ให้เกินค่าที่กำหนดเรียกว่า ขีดจำกัดหน้าต่าง (window limit) ในกลไกจึงมีพารามิเตอร์ที่สำคัญ 2 ตัว คือ ขนาดของหน้าต่าง (window size) และขีดจำกัดหน้าต่าง

2.3.2.1 กลไกหน้าต่างกระโดด (Jumping Window Mechanism)

กลไกนี้เป็นกลไกหน้าต่างที่ง่ายที่สุด คือ หน้าต่างมีขนาดคงที่และต่อเนื่องกัน โดยที่ขีดจำกัดหน้าต่างมีค่าเท่ากันในหน้าต่างทุกๆ หน้าต่าง ข้อบกพร่องที่สำคัญที่สุดของกลไกนี้คือ หน้าต่างมีจุดเริ่มต้นที่ตายตัว หากผู้ใช้ทราบ จะสามารถลักลอบส่งเซลล์ได้สูงสุดถึง 2 เท่าของขีดจำกัดหน้าต่าง โดยแบ่งครึ่งข้อมูลส่งใน 2 หน้าต่าง การทำงานของกลไกหน้าต่างกระโดดรวมถึงข้อบกพร่องดังกล่าวแสดงดังรูปที่ 2.3(ก)

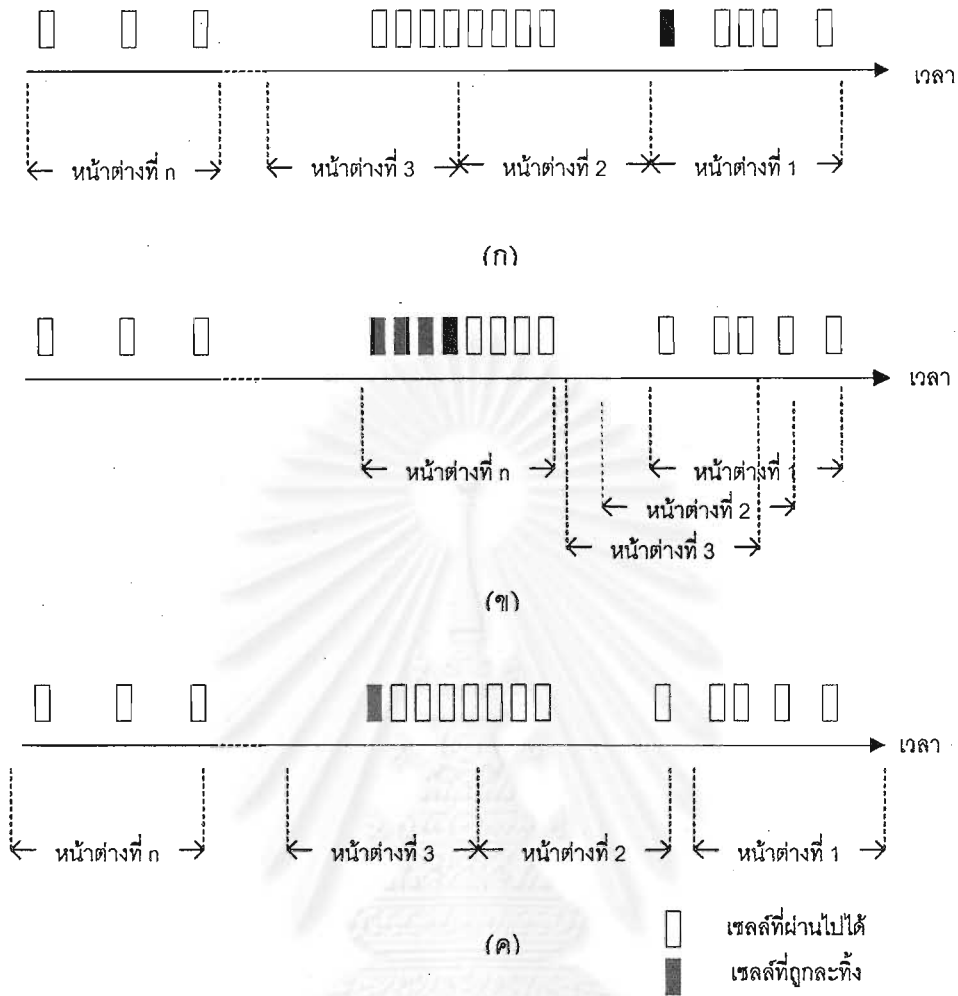
2.3.2.2 กลไกหน้าต่างเคลื่อนที่ (Moving Window Mechanism)

กลไกนี้พัฒนาจากกลไกหน้าต่างกระโดด โดยกำหนดจุดเริ่มต้นของแต่ละหน้าต่างให้พร้อมกับการมาถึงของทุก ๆ เซลล์ มองได้เหมือนหน้าต่างเลื่อนไปตามแกนเวลา ในขณะที่ขนาดของหน้าต่างและขีดจำกัดหน้าต่างยังคงคงที่ การทำงานของกลไกนี้แสดงดังรูปที่ 2.3 (ข)

แม้ว่าหน้าต่างเคลื่อนที่จะแก้ปัญหาของหน้าต่างกระโดดได้ แต่ในขณะเวลาใดๆ กลไกอาจต้องใช้หน่วยความจำสูงสุดเป็นจำนวนเท่ากับขีดจำกัดหน้าต่าง โดยเฉพาะอย่างยิ่งในการประเมินลักษณะทราฟฟิกในระยะยาว มีความจำเป็นอย่างยิ่งที่ขนาดหน้าต่างจะต้องกว้างมาก ดังนั้นวิธีนี้จึงเป็นการสิ้นเปลืองทรัพยากรหน่วยความจำเกินไป

2.3.2.3 กลไกหน้าต่างกระโดดแบบจุดชนวน (Triggered Jumping Window Mechanism)

กลไกนี้มีความคล้ายคลึงกับหน้าต่างกระโดด ข้อแตกต่างคือ จุดเริ่มต้นของหน้าต่างต่อไปจะเกิดขึ้นก็ต่อเมื่อหน้าต่างก่อนสิ้นสุดลงแล้วพร้อมๆ กับการมาถึงของเซลล์ถัดไป การทำงานของกลไกนี้แสดงไว้ในรูปที่ 2.3 (ค)



รูปที่ 2.3 หลักการของกลไกหน้าต่างแบบต่างๆ เมื่อกำหนดให้ขีดจำกัดหน้าต่างเท่ากับ 4
 (ก) หน้าต่างกระโดด (ข) หน้าต่างเคลื่อนที่ (ค) หน้าต่างกระโดดแบบจุดชนวน

2.3.2.4 กลไกเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอ็กซ์โพเนนเชียล (Exponentially Weighted Moving Average Mechanism: EWMA)

EWMA แตกต่างจากกลไกหน้าต่างชนิดอื่นๆ ที่มีขีดจำกัดหน้าต่างไม่เท่ากันในหน้าต่างแต่ละหน้าต่าง ในขณะที่ขนาดของหน้าต่างนั้นยังคงคงที่ ขีดจำกัดของหน้าต่างที่ i (N_i) คำนวณจากสมการที่ (2.2)

$$N_i = \frac{N_{i-1} - \gamma \delta_{i-1}}{1 - \gamma} \quad (2.2)$$

โดย γ คือพารามิเตอร์ควบคุมความยืดหยุ่นของกลไก เมื่อ $\gamma = 0$ กลไกนี้จะเป็นหน้าต่างกระโดด และ S_i คือผลรวมถ่วงน้ำหนักแบบเอ็กซ์โพเนนเชียลของ S_{i-1} กับจำนวนเซลล์ที่มาถึงในหน้าต่างก่อน (X_{i-1}) ซึ่ง S_i คำนวณได้จากสมการที่ (2.3)

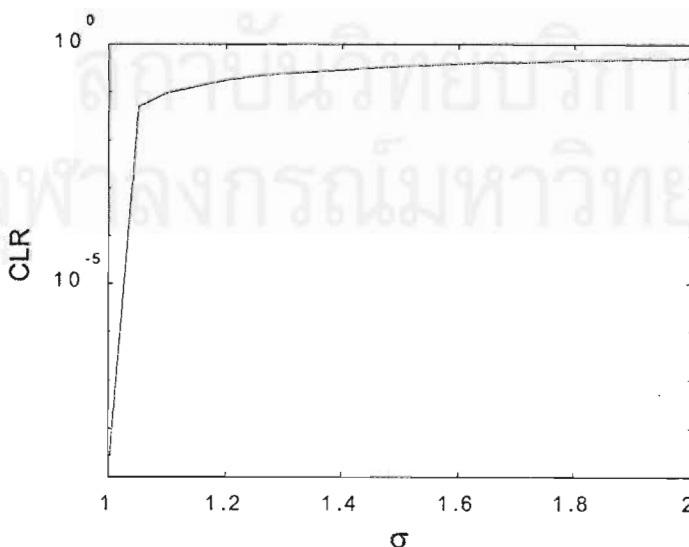
$$S_{i-1} = (1-\gamma)X_{i-1} + \gamma S_{i-2} \quad (2.3)$$

2.4 การประเมินประสิทธิภาพของกลไกตรวจกราฟฟิก

จากคุณลักษณะที่ต้องการของตัวตรวจกราฟฟิกในหัวข้อที่ 2.3 ในการประเมินคุณสมบัติความสามารถในการเลือก (Selectivity) เราเปรียบเทียบจากความสามารถในการตรวจวัดโหลดเกิน (Overload Detection Capability) กล่าวคืออัตราส่วนการสูญหายของเซลล์ (CLR) อันเนื่องมาจากตัวตรวจกราฟฟิกควรใกล้เคียงกับความน่าจะเป็นของการละเมิดอุดมคติ (Ideal Violation Probability), P_d , ดังสมการที่ (2.4) และเขียนกราฟได้ดังรูปที่ 2.4

$$P_d = \frac{\sigma - 1}{\sigma} \quad (2.4)$$

โดยที่ σ คือ อัตราเซลล์เฉลี่ยระยะยาวจากแหล่งกำเนิดที่นอร์มอลไลซ์แล้ว (Normalized Long-term Mean Cell Rate)



รูปที่ 2.4 เส้นโค้งความน่าจะเป็นการละเมิดอุดมคติ

วัตถุประสงค์อย่างหนึ่งของวิทยานิพนธ์ฉบับนี้ก็คือการพัฒนาตัวตรวจทราฟฟิกที่ตรวจอัตราเซลล์เฉลี่ย ดังนั้นหากเราสามารถออกแบบตัวตรวจทราฟฟิกให้มีความสามารถในการตรวจวัดโหลดเกินเข้าใกล้เส้นโค้งการละเมิดอุดมคติได้มากเท่าใด ย่อมจะทำให้การใช้ทรัพยากรโครงข่ายประหยัดขึ้นเท่านั้น นอกจากนี้ ตัวตรวจทราฟฟิกไม่ควรทำให้เกิด false alarm probability หรืออัตราส่วนการสูญหายของเซลล์เนื่องจากตัวตรวจทราฟฟิกสูงกว่า QoS ของทราฟฟิก

การประเมินคุณสมบัติการตอบสนอง (Responsiveness) ทำได้โดยการเปรียบเทียบจำนวนเซลล์ที่กลไกแต่ละอย่างยอมให้ผ่านเข้าสู่โครงข่ายก่อนจะมีการละทิ้งเซลล์แรก และอัตราการละทิ้งเซลล์หลังจากการตรวจพบเซลล์แรกเมื่อผู้ใช้ส่งทราฟฟิกด้วยโหลดเกินขนาดหนึ่ง



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

แบบจำลองแหล่งกำเนิดและกลไกตรวจกราฟฟิก

3.1 แบบจำลองแหล่งกำเนิดที่ใช้ในการทดสอบ

การทดสอบใช้แบบจำลองแหล่งกำเนิดแบบเบิสต์ (Bursty Source Model) หรือแบบจำลองแหล่งกำเนิดแบบเปิด-ปิด (On-Off Source Model) สำหรับกราฟฟิก packet voice

แหล่งกำเนิดจะอยู่ใน 2 สถานะคือสถานะเบิสต์และสถานะเงียบ (silent) ดังแสดงในรูปที่ 3.1 ในสถานะเบิสต์ แหล่งกำเนิดจะส่งเซลล์ด้วยอัตราคงที่เท่ากับอัตราเซลล์ค้ายอด โดยที่จำนวนเซลล์ในแต่ละเบิสต์ (X) มีการแจกแจงแบบเรขาคณิต (Geometric Distribution) เมื่อ p คือความน่าจะเป็นของการไม่เกิดเซลล์ จะได้ว่าฟังก์ชันความน่าจะเป็นและค่าเฉลี่ยของ X ($E(X)$) แสดงได้ดังสมการที่ (3.1) และ (3.2) ตามลำดับ

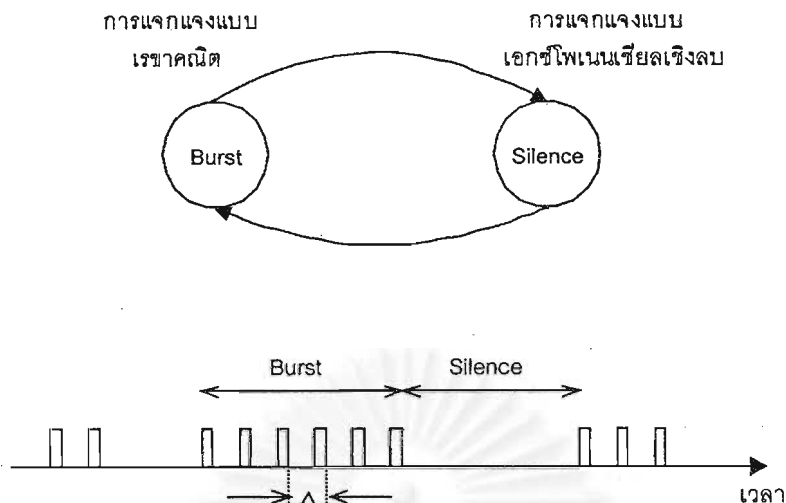
$$f(x; p) = \begin{cases} p(1-p)^x & ; x \in \{0, 1, \dots\} \\ 0 & ; otherwise \end{cases} \quad (3.1)$$

$$E(X) = \frac{1-p}{p} \quad (3.2)$$

และระยะเวลาเงียบ (s) มีการแจกแจงของแบบเอ็กซ์โพเนนเชียลเชิงลบ (Negative Exponential Distribution) เมื่อ β คือระยะเวลาเงียบเฉลี่ย ฟังก์ชันความน่าจะเป็นแสดงได้ดังสมการที่ (3.3)

$$f(s; \beta) = \frac{1}{\beta} e^{-\frac{s}{\beta}} \quad (3.3)$$

เหตุผลในการเลือกใช้แบบจำลองนี้เพราะมีความง่ายและยืดหยุ่น กล่าวคือสามารถปรับค่าพารามิเตอร์ต่างๆ เช่น อัตราเซลล์ค้ายอด อัตราเซลล์เฉลี่ย และระยะเวลาเบิสต์สูงสุด ได้อย่างเป็นอิสระต่อกัน



รูปที่ 3.1 แบบจำลองแหล่งกำเนิดแบบเปิด-ปิด

เนื่องจากต้องมีการประเมินประสิทธิภาพของกลไกโดยเปรียบเทียบกับกลไกอื่น [1, 6] แหล่งกำเนิดที่ใช้ทดสอบจึงอ้างอิงตามงานวิจัยดังกล่าวด้วย ดังนั้นพารามิเตอร์ในการจำลองแหล่งกำเนิดทราฟฟิก packet voice จะแบ่งได้เป็น 2 แบบดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 พารามิเตอร์ของแบบจำลองแหล่งกำเนิดที่ใช้ในการทดสอบ

| พารามิเตอร์ | แบบจำลองที่ 1 | แบบจำลองที่ 2 |
|--|---------------|---------------|
| ระยะเวลาระหว่างการมาถึงของแต่ละเซลล์ (Cell interarrival time : Δ) | 16 ms | 12 ms |
| จำนวนเซลล์เฉลี่ยในแต่ละเบิร์สต์ ($E(X)$) | 5 | 29 |
| ระยะเวลาเงียบเฉลี่ย (Mean silence duration : $E(S)$) | 147.72 ms | 650 ms |
| อัตราบิตสูงสุด (Peak Bit Rate) | 32 kbps | 32 kbps |
| อัตราเซลล์เฉลี่ย (Mean Cell Rate : MCR) | 21.96 cells/s | 29.06 cells/s |

โดย อัตราเซลล์เฉลี่ยของทราฟฟิกคำนวณได้ตามสมการที่ (3.4)

$$MCR = \frac{E(x)}{E(x)\Delta + E(S)} \quad (3.4)$$

3.2 โครงสร้างของกลไกตรวจตราฟิกที่ใช้เปรียบเทียบ

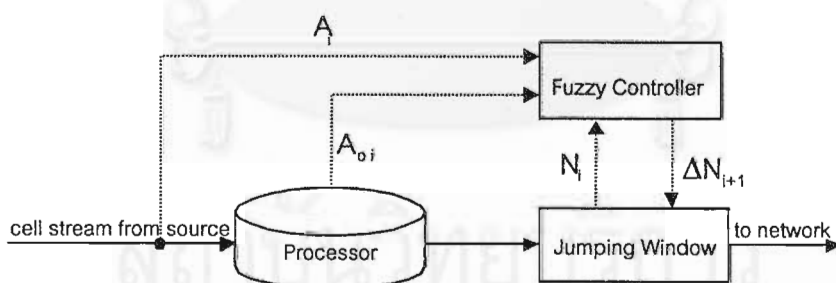
ตามวัตถุประสงค์ของวิทยานิพนธ์ กลไกที่เสนอจะถูกนำไปเปรียบเทียบประสิทธิภาพกับ กลไก 2 กลไกคือ กลไกถึงร้วแบบไม่มีบัฟเฟอร์ [1] และกลไกหน้าต่างฟัชชีลอจิก [6]

3.2.1 กลไกถึงร้วแบบไม่มีบัฟเฟอร์

หลักการพื้นฐานและการทำงานของกลไกถึงร้วแบบไม่มีบัฟเฟอร์ได้กล่าวถึงไปแล้วในหัวข้อที่ 2.3.1.1 โดยจะทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 เท่านั้น พารามิเตอร์ของกลไก [1] ประกอบด้วยอัตราร้วเท่ากับ 31.24 เซลล์ต่อวินาทีหรือเป็น 1.42 เท่าของอัตราเซลล์เฉลี่ยที่ตกลงไว้ของทราฟฟิกและขีดจำกัดตัวนับเท่ากับ 45

3.2.2 กลไกหน้าต่างฟัชชีลอจิก

กลไกหน้าต่างฟัชชีลอจิก [6] เป็นกลไกที่มีโครงสร้างพื้นฐานเป็นกลไกหน้าต่างกระโดด และฟัชชีลอจิกควบคุมการเพิ่มหรือลดของขีดจำกัดของหน้าต่างต่อไป (ΔN_{i+1}) โครงสร้างของกลไกแสดงดังรูปที่ 3.2



รูปที่ 3.2 แบบจำลองการทำงานของกลไกหน้าต่างฟัชชีลอจิก

อินพุตของตัวควบคุมฟัชชีลอจิกประกอบด้วย

- จำนวนเซลล์ที่มาถึงเฉลี่ยต่อ 1 หน้าต่างเริ่มตั้งแต่เริ่มต้นการต่อ ($A_{o,i}$) มีฟัชชีเซตประกอบด้วย

$$A_{o,i} = \{Low (L), Medium (M), High (H)\}$$
- จำนวนเซลล์ที่มาถึงในหน้าต่างที่ i (A_i) มีฟัชชีเซตประกอบด้วย

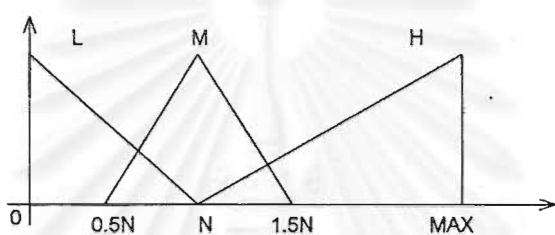
$$A_i = \{Low (L), Medium (M), High (H)\}$$
- ขีดจำกัดของหน้าต่างที่ i (N_i) มีฟัชชีเซตประกอบด้วย

$N_i = \{\text{Low (L), Medium (M), High (H)}\}$

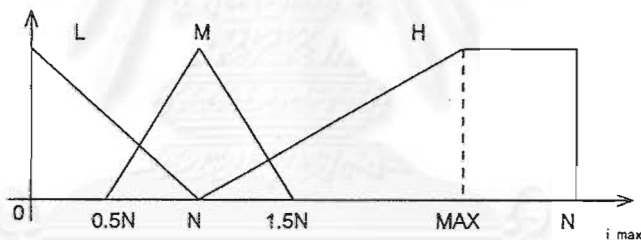
และเอาต์พุตของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กที่เสนอคือ ซีดจำกัดของหน้าต่างที่ $i+1$ (ΔN_{i+1}) มีสมาชิกฟัซซีเซตเป็น

$\Delta N_{i+1} = \{\text{Positive Big (PB), Positive Medium (PM), Positive Small (PS), Zero Equal (ZE), Negative Small (NS), Negative Medium (NM), Negative Big (NB)}\}$

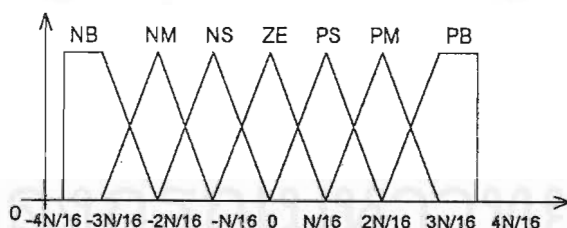
ฟังก์ชันสมาชิก (Membership Function) สำหรับฟัซซีเซตของอินพุตและเอาต์พุตแสดงดังรูปที่ 3.3(ก)-3.3(ค)



(ก)



(ข)



(ค)

รูปที่ 3.3 ฟังก์ชันสมาชิกสำหรับฟัซซีเซตของกลไกหน้าต่างฟัซซีลอจิก

(ก) A_0, A_i , (ข) N_i , (ค) ΔN_{i+1}

พารามิเตอร์ของกลไกที่ใช้ทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 ได้แก่ $N = 32$ เซลล์, $N_1 = 3.5N$, $N_{i_{max}} = 9N$, $MAX = 90$ เซลล์ และขนาดหน้าต่าง = 1,440 ms

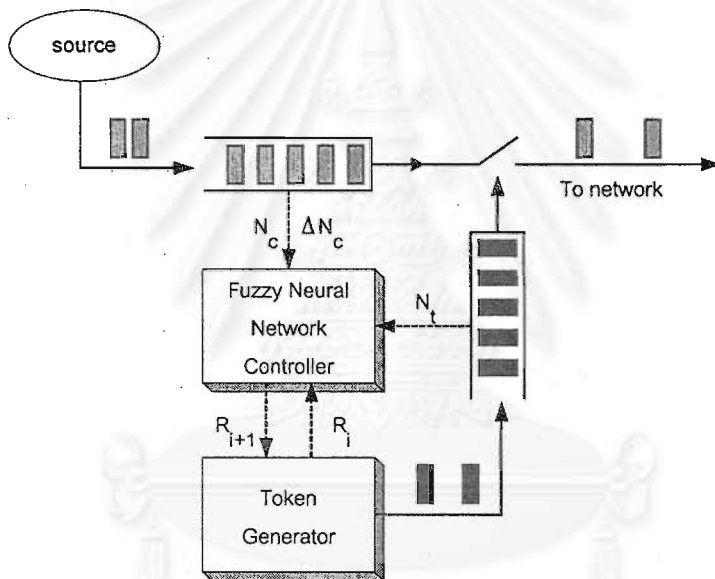
พารามิเตอร์ของกลไกที่ใช้ทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 ได้แก่ $N = 87$ เซลล์, $N_1 = 3.5N$, $N_{L_{max}} = 9N$, $MAX = 230$ เซลล์ และขนาดหน้าต่าง = 3,000 ms กฎพื้นฐานฟัซซีที่ใช้ในขั้นตอน inference แสดงไว้ในตารางที่ 3.2

ตารางที่ 3.2 กฎพื้นฐานฟัซซีสำหรับ inference ของตัวควบคุมฟัซซีลอจิกในกลไกหน้าต่างฟัซซีลอจิก

| Rule # | A_{0i} | N_i | A_i | ΔN_{i+1} |
|--------|----------|-------|-------|------------------|
| 1 | L | H | L | PB |
| 2 | L | H | M | PS |
| 3 | L | H | H | ZE |
| 4 | M | M | L | PB |
| 5 | M | M | M | PS |
| 6 | M | M | H | ZE |
| 7 | M | H | L | PB |
| 8 | M | H | M | ZE |
| 9 | M | H | H | NB |
| 10 | H | L | L | PB |
| 11 | H | L | M | PM |
| 12 | H | L | H | PS |
| 13 | H | M | L | PB |
| 14 | H | M | M | PM |
| 15 | H | M | H | ZE |
| 16 | H | H | L | NS |
| 17 | H | H | M | NM |
| 18 | H | H | H | NB |

3.3 โครงสร้างของตัวตรวจทราฟฟิกโดยใช้ฟuzzyนิวรอลเน็ตเวิร์ก

ตัวตรวจทราฟฟิกที่ใช้ฟuzzyนิวรอลเน็ตเวิร์กมีโครงสร้างดังรูปที่ 3.4 เหตุผลที่ใช้โครงสร้างพื้นฐานเป็นกลไกถึงร้วแบบโทเค็น เนื่องจากมีอัตราส่วนการสูญหายของเซลล์น้อยกว่าถึงร้วแบบไม่มีบัฟเฟอร์และมีการตอบสนองเร็วกว่ากลไกหน้าต่าง เมื่อนำฟuzzyลอจิกมาประยุกต์ใช้ร่วมด้วย จึงมีความเป็นไปได้ที่จะพัฒนาตัวตรวจบนโครงสร้างดังกล่าวให้มีประสิทธิภาพดีกว่ากลไกหน้าต่างฟuzzyลอจิกและกลไกถึงร้วแบบดั้งเดิม นอกจากนี้ พารามิเตอร์ที่ได้จากคิวบัฟเฟอร์และป่อโทเค็นยังสามารถนำมาใช้เป็นตัวแทนของลักษณะทราฟฟิกด้านเข้าได้โดยไม่ต้องมีตัวประมวลผล (Processor) เพิ่มเติมอีก



รูปที่ 3.4 แบบจำลองการทำงานของกลไกถึงร้วฟuzzyนิวรอลเน็ตเวิร์ก

การทำงานของกลไกเริ่มจากอินพุตของฟuzzyนิวรอลเน็ตเวิร์กซึ่งเป็นตัวแทนของลักษณะทราฟฟิกในขณะใดๆ ถูกสุ่มเพื่อวัดค่าทุกๆ คาบเวลาคงที่ค่าหนึ่งที่เหมาะสมเรียกว่า คาบเวลาชักตัวอย่าง (Sampling Period) อินพุตที่วัดได้จะถูกป้อนให้กับตัวควบคุมฟuzzyนิวรอลเน็ตเวิร์กเพื่อนำไปใช้คำนวณเอาต์พุตคืออัตราการกำเนิดโทเค็นในคาบเวลาต่อไป

การออกแบบกลไกเริ่มจากกำหนดพารามิเตอร์ต่างๆ ได้แก่ขนาดของคิวบัฟเฟอร์ด้านเข้า (B) ขนาดของป่อโทเค็น (P) และคาบเวลาชักตัวอย่าง (T_s) โดยกำหนดให้สัมพันธ์กับพารามิเตอร์ของ ทราฟฟิกจากแหล่งกำเนิดแต่ละแบบ ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 พารามิเตอร์ของกลไกถึงรั้วฟuzzyนิวรอลเน็ตเวิร์ก

| พารามิเตอร์ | แบบจำลองแหล่งกำเนิดแบบที่ 1 | แบบจำลองแหล่งกำเนิดแบบที่ 2 |
|-------------|-------------------------------------|---------------------------------------|
| B | 10 $= 2E_1(X)$ | 30 $\approx E_2(X)$ |
| P | 500 | 3,000 |
| T_s | 230 ms $= E_1(X)\Delta + E_1(S)$ | 1,000 ms $= E_2(X)\Delta + E_2(S)$ |

เหตุผลที่กำหนดให้ควมับฟเฟอร์ในกรณีทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 มีขนาดสัมพันธ์กับ $E(X)$ ของแหล่งกำเนิดต่ำกว่าในกรณีแบบจำลองแบบที่ 1 ก็คือว่าขนาดบัพเฟอร์สัมพันธ์โดยตรงกับเวลาประวิงของทราฟฟิก ดังนั้นการกำหนดให้ขนาดบัพเฟอร์ใหญ่จนเกินไปจึงไม่เป็นผลดีต่อสมรรถนะของกลไก ในทางปฏิบัติพบว่าขนาดของบัพเฟอร์กับความเวลาชักตัวอย่างมีผลอย่างมากต่อการกำหนดฟังก์ชันสมาชิกและกฎพื้นฐานฟuzzyในภายหลัง การเปลี่ยนขนาดของพารามิเตอร์ทั้งสองเพียงเล็กน้อย ทำให้ผลการทดสอบที่ได้แตกต่างกันมาก ดังนั้นจึงจำเป็นต้องกำหนดพารามิเตอร์ทั้งสองนี้ให้ตายตัวก่อนการหาฟังก์ชันสมาชิกและกฎพื้นฐานฟuzzy ซึ่งได้มาจากการลองผิดลองถูก (trial and error) และปรับ (tune) จนได้ผลที่น่าพอใจ ในขณะที่ขนาดของบ่อโทเค็นยังสามารถปรับเปลี่ยนให้เหมาะสมได้ภายหลัง เพราะภายใต้ฟังก์ชันสมาชิกและกฎชุดหนึ่งๆ อาจมีขนาดของบ่อโทเค็นที่เหมาะสมหรือให้สมรรถนะของกลไกที่อยู่ในช่วงที่ต้องการได้หลายค่า โดยผลการทดสอบจะต่างกันเล็กน้อย

อินพุตของตัวควบคุมฟuzzyนิวรอลเน็ตเวิร์กที่เสนอประกอบด้วย

- จำนวนเซลล์ในควมับฟเฟอร์ด้านเข้า มีฟuzzyเซตประกอบด้วย

$$N_c = \{Zero (Z), Low (L), Medium (M), High (H), Very High (VH)\}$$
- การเปลี่ยนแปลงของจำนวนเซลล์ในควมับฟเฟอร์ด้านเข้า มีฟuzzyเซตประกอบด้วย

$$\Delta N_c = \{Positive Big (PB), Positive Small (PS), Zero Equal (ZE), Negative Small (NS), Negative Big (NB)\}$$
- จำนวนโทเค็นในบ่อโทเค็น มีฟuzzyเซตประกอบด้วย

$$N_T = \{Zero (Z), Low (L), Medium (M), High (H), Very High (VH)\}$$

- อัตราการกำเนิดโทเค็น มีฟัซซีเซตประกอบด้วย

$R_t = \{ \text{Positive Big (PB), Positive Medium (PM), Positive Small (PS), Zero Equal (ZE), Negative Small (NS), Negative Medium (NM), Negative Big (NB)} \}$

และเอาต์พุตของตัวควบคุมฟัซซีนิเวรอลเน็ตเวิร์กที่เสนอคือ อัตราการกำเนิดโทเค็นมีฟัซซีเซตประกอบด้วย

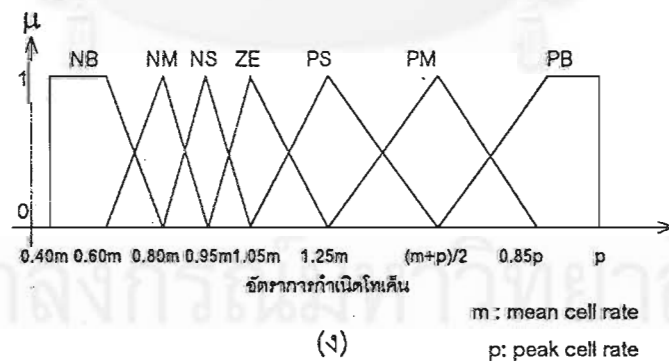
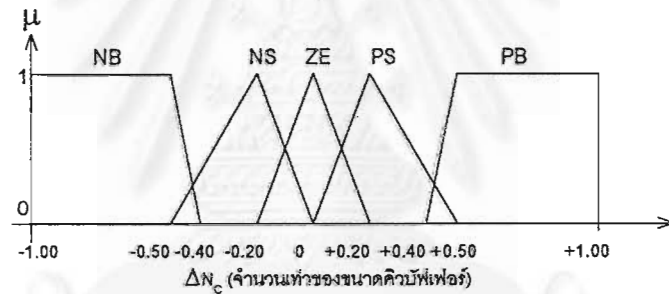
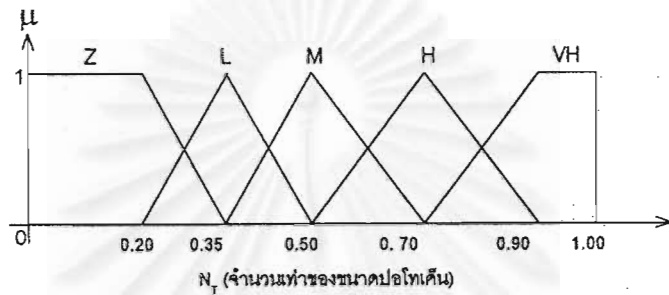
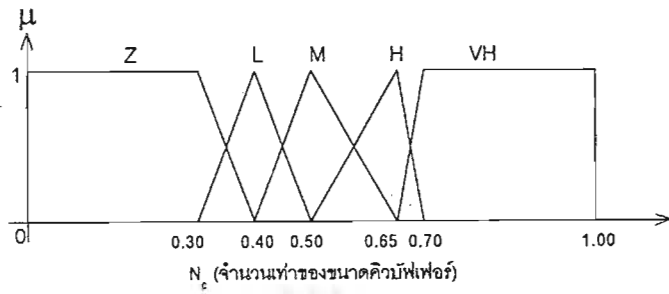
$R_{t+1} = \{ \text{Positive Big (PB), Positive Medium (PM), Positive Small (PS), Zero Equal (ZE), Negative Small (NS), Negative Medium (NM), Negative Big (NB)} \}$

เหตุผลที่ออกแบบให้อัตราการกำเนิดโทเค็นมีจำนวนสมาชิกฟัซซีเซตมากที่สุดเป็น 7 ระดับคือ เพื่อให้การปรับอัตราการกำเนิดโทเค็นเรียบ (smooth) และละเอียด ฟังก์ชันสมาชิกของอินพุตและเอาต์พุตมีลักษณะสามเหลี่ยมดังรูปที่ 3.5(ก)-3.5(ง) และกฎพื้นฐานฟัซซีแสดงดังตารางที่ 3.4

การกำหนดฟังก์ชันสมาชิกใช้วิธี hand optimization และเกณฑ์ในการตัดสินใจเพิ่มหรือลดอัตราการกำเนิดโทเค็นหรืออีกนัยหนึ่งคือเครดิตซึ่งเป็นที่มาของกฎพื้นฐานฟัซซีโดยสรุปประกอบด้วย

- ถ้าพฤติกรรมแหล่งกำเนิดในช่วงเวลาก่อนหน้าระยะยาวมีการละเมิดข้อตกลง พฤติกรรมในคาบเวลาก่อนหน้า (ระยะสั้น) มีการละเมิดข้อตกลง จะปรับลดเครดิตลงในระดับมาก
- ถ้าพฤติกรรมแหล่งกำเนิดในช่วงเวลาก่อนหน้าระยะยาวมีการละเมิดข้อตกลง พฤติกรรมในคาบเวลาก่อนหน้า (ระยะสั้น) ไม่มีการละเมิดข้อตกลง จะปรับลดเครดิตลงในระดับปานกลาง
- ถ้าพฤติกรรมแหล่งกำเนิดในช่วงเวลาก่อนหน้าระยะยาวไม่มีการละเมิดข้อตกลง พฤติกรรมในคาบเวลาก่อนหน้า (ระยะสั้น) มีการละเมิดข้อตกลง จะปรับลดเครดิตลงในระดับน้อยหรือคงที่
- ถ้าพฤติกรรมแหล่งกำเนิดในช่วงเวลาก่อนหน้าระยะยาวไม่มีการละเมิดข้อตกลง พฤติกรรมในคาบเวลาก่อนหน้า (ระยะสั้น) ไม่มีการละเมิดข้อตกลง จะปรับเพิ่มเครดิตหรือคงที่

ระดับการปรับเพิ่มหรือลดเครดิตขึ้นกับระดับของการละเมิดหรือไม่ละเมิดข้อตกลง โดยการปรับละเอียดได้จากการจูน (tune) ด้วยวิธีลองผิดลองถูก (Trial and Error) ขั้นตอน defuzzification ใช้วิธีจุดศูนย์ถ่วงฟัซซี (Fuzzy Centroid Method) ร่วมกับ Piecewise Linearities Algorithm [8] เพื่อความรวดเร็วในการหาผลลัพธ์



รูปที่ 3.5 ฟังก์ชันสมาชิกสำหรับฟัซซีเซตของกลไกถังร้วฟัซซีนิรอลเน็ตเวิร์ก

(ก) N_c (ข) N_t (ค) ΔN_c (ง) R_i, R_{i+1}

ตารางที่ 3.4 กฎพื้นฐานฟuzzy สำหรับ inference ของตัวควบคุมฟuzzy นิวรอลเน็ตเวิร์ก
ในกลไกถ่วงรัวฟuzzy นิวรอลเน็ตเวิร์ก

| Rule # | N_c | ΔN_c | N_T | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 1 | Z | NB | Z | NS | PB |
| 2 | Z | NB | Z | ZE | PM |
| 3 | Z | NB | Z | PS | PS |
| 4 | Z | NB | Z | PM | PS |
| 5 | Z | NB | Z | PB | PB |
| 6 | Z | NB | L | NS | PM |
| 7 | Z | NB | L | ZE | PS |
| 8 | Z | NB | L | PS | ZE |
| 9 | Z | NB | L | PM | ZE |
| 10 | Z | NB | L | PB | PS |
| 11 | Z | NB | M | ZE | PS |
| 12 | Z | NB | M | PS | ZE |
| 13 | Z | NB | M | PM | ZE |
| 14 | Z | NB | M | PB | ZE |
| 15 | Z | NB | H | PS | ZE |
| 16 | Z | NB | H | PM | NS |
| 17 | Z | NB | H | PB | NS |
| 18 | Z | NB | VH | PM | NM |
| 19 | Z | NB | VH | PB | NM |
| 20 | Z | NS | Z | NS | ZE |
| 21 | Z | NS | Z | ZE | PS |
| 22 | Z | NS | Z | PS | PS |
| 23 | Z | NS | Z | PM | ZE |
| 24 | Z | NS | Z | PB | NS |
| 25 | Z | NS | L | NS | PS |
| 26 | Z | NS | L | ZE | PS |

| Rule # | N_c | ΔN_c | N_T | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 27 | Z | NS | L | PS | ZE |
| 28 | Z | NS | L | PM | PS |
| 29 | Z | NS | L | PB | PS |
| 30 | Z | NS | L | PS | NS |
| 31 | Z | NS | M | PM | ZE |
| 32 | Z | NS | M | PB | PS |
| 33 | Z | NS | H | PM | NS |
| 34 | Z | NS | H | PB | NS |
| 35 | Z | NS | VH | PM | NM |
| 36 | Z | NS | VH | PB | NM |
| 37 | Z | ZE | Z | NS | ZE |
| 38 | Z | ZE | Z | ZE | PS |
| 39 | Z | ZE | Z | PS | ZE |
| 40 | Z | ZE | Z | PM | PS |
| 41 | Z | ZE | L | NS | PM |
| 42 | Z | ZE | L | ZE | PS |
| 43 | Z | ZE | L | PS | ZE |
| 44 | Z | ZE | L | PM | PS |
| 45 | Z | ZE | L | PB | NS |
| 46 | Z | ZE | M | ZE | PM |
| 47 | Z | ZE | M | PS | NS |
| 48 | Z | ZE | M | PM | PS |
| 49 | Z | ZE | M | PB | ZE |
| 50 | Z | ZE | H | PS | ZE |
| 51 | Z | ZE | H | PM | ZE |
| 52 | Z | ZE | H | PB | NS |
| 53 | Z | ZE | VH | PM | NS |
| 54 | Z | ZE | VH | PB | NM |
| 55 | Z | PS | Z | NB | NM |

| Rule # | N_c | ΔN_c | N_T | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 56 | Z | PS | Z | NM | NS |
| 57 | Z | PS | Z | NS | ZE |
| 58 | Z | PS | Z | ZE | PS |
| 59 | Z | PS | Z | PS | ZE |
| 60 | L | NB | Z | NS | PM |
| 61 | L | NB | Z | ZE | PS |
| 62 | L | NB | Z | PS | PS |
| 63 | L | NB | Z | PM | PS |
| 64 | L | NB | Z | PB | PM |
| 65 | L | NS | Z | NS | PS |
| 66 | L | NS | Z | ZE | ZE |
| 67 | L | NS | Z | PS | ZE |
| 68 | L | NS | Z | PM | PS |
| 69 | L | NS | Z | PB | PS |
| 70 | L | ZE | Z | NM | NS |
| 71 | L | ZE | Z | NS | ZE |
| 72 | L | ZE | Z | ZE | PS |
| 73 | L | ZE | Z | PS | ZE |
| 74 | L | ZE | Z | PM | ZE |
| 75 | L | ZE | Z | PB | NS |
| 76 | L | PS | Z | NB | NS |
| 77 | L | PS | Z | NM | ZE |
| 78 | L | PS | Z | NS | PS |
| 79 | L | PS | Z | ZE | PS |
| 80 | L | PS | Z | PS | ZE |
| 81 | L | PS | Z | PM | NS |
| 82 | L | PB | Z | NB | ZE |
| 83 | L | PB | Z | NM | PS |
| 84 | L | PB | Z | NS | PS |

| Rule # | N_c | ΔN_c | N_T | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 85 | L | PB | Z | ZE | PS |
| 86 | L | PB | Z | PS | ZE |
| 87 | M | NB | Z | NS | PB |
| 88 | M | NB | Z | ZE | PM |
| 89 | M | NB | Z | PS | ZE |
| 90 | M | NB | Z | PM | PS |
| 91 | M | NS | Z | NS | PS |
| 92 | M | NS | Z | ZE | PS |
| 93 | M | NS | Z | PS | ZE |
| 94 | M | NS | Z | PM | ZE |
| 95 | M | NS | Z | PB | ZE |
| 96 | M | ZE | Z | NB | NS |
| 97 | M | ZE | Z | NM | ZE |
| 98 | M | ZE | Z | NS | ZE |
| 99 | M | ZE | Z | PS | ZE |
| 100 | M | ZE | Z | PM | PS |
| 101 | M | ZE | Z | PB | NS |
| 102 | M | PS | Z | NB | NM |
| 103 | M | PS | Z | NM | NS |
| 104 | M | PS | Z | NS | ZE |
| 105 | M | PS | Z | ZE | PS |
| 106 | M | PS | Z | PS | ZE |
| 107 | M | PS | Z | PM | ZE |
| 108 | M | PS | Z | PB | NS |
| 109 | M | PB | Z | NM | PS |
| 110 | M | PB | Z | NS | PS |
| 111 | M | PB | Z | ZE | PS |
| 112 | M | PB | Z | PS | NS |
| 113 | H | NB | Z | NS | PB |

| Rule # | N_c | ΔN_c | N_r | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 114 | H | NB | Z | ZE | PM |
| 115 | H | NB | Z | PS | PM |
| 116 | H | NB | Z | PM | PS |
| 117 | H | NB | Z | PB | ZE |
| 118 | H | NS | Z | NS | PM |
| 119 | H | NS | Z | ZE | PS |
| 120 | H | NS | Z | PS | PS |
| 121 | H | NS | Z | PM | ZE |
| 122 | H | NS | Z | PB | NS |
| 123 | H | ZE | Z | NB | NM |
| 124 | H | ZE | Z | NM | NS |
| 125 | H | ZE | Z | NS | ZE |
| 126 | H | ZE | Z | ZE | PS |
| 127 | H | ZE | Z | PS | ZE |
| 128 | H | ZE | Z | PM | NS |
| 129 | H | ZE | Z | PB | NM |
| 130 | H | PS | Z | NB | NS |
| 131 | H | PS | Z | NM | ZE |
| 132 | H | PS | Z | NS | PS |
| 133 | H | PS | Z | ZE | PS |
| 134 | H | PS | Z | PM | NS |
| 135 | H | PS | Z | PB | NM |
| 136 | H | PB | Z | NB | ZE |
| 137 | H | PB | Z | NM | PS |
| 138 | H | PB | Z | NS | PS |
| 139 | H | PB | Z | ZE | NS |
| 140 | H | PB | Z | PS | NM |
| 141 | H | PB | Z | PM | NB |
| 142 | H | PB | Z | PB | NB |

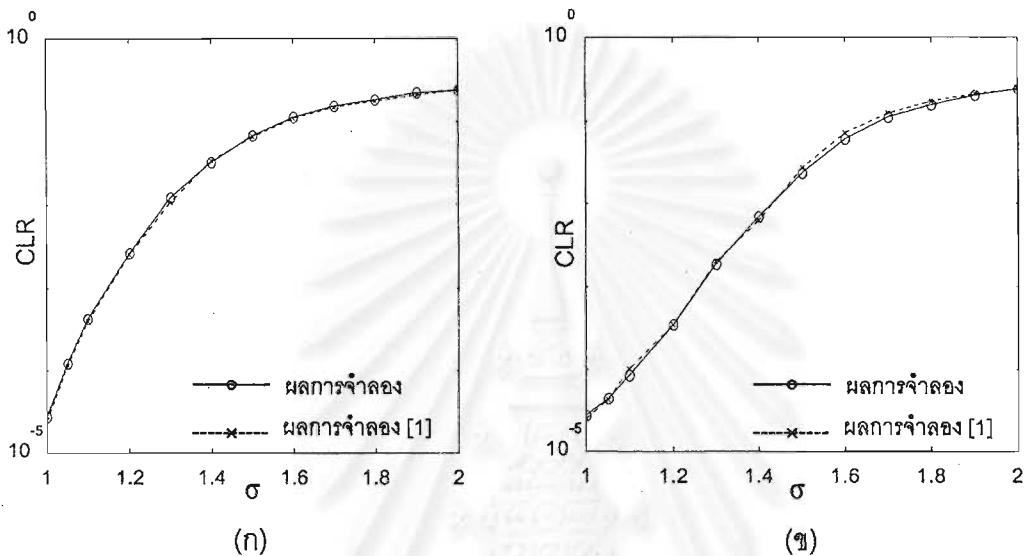
| Rule # | N_c | ΔN_c | N_T | R_i | R_{i+1} |
|--------|-------|--------------|-------|-------|-----------|
| 143 | VH | NS | Z | PS | ZE |
| 144 | VH | NS | Z | PM | PS |
| 145 | VH | NS | Z | PB | ZE |
| 146 | VH | ZE | Z | NS | PS |
| 147 | VH | ZE | Z | PS | ZE |
| 148 | VH | ZE | Z | PM | ZE |
| 149 | VH | ZE | Z | PB | PS |
| 150 | VH | PS | Z | NB | ZE |
| 151 | VH | PS | Z | NM | PS |
| 152 | VH | PS | Z | NS | PS |
| 153 | VH | PS | Z | ZE | NS |
| 154 | VH | PS | Z | PS | ZE |
| 155 | VH | PS | Z | PM | NS |
| 156 | VH | PS | Z | PB | NM |
| 157 | VH | PB | Z | NB | NS |
| 158 | VH | PB | Z | NM | ZE |
| 159 | VH | PB | Z | ZE | PS |
| 160 | VH | PB | Z | PS | NS |
| 161 | VH | PB | Z | PM | NM |
| 162 | VH | PB | Z | PB | NB |

3.4 การทดสอบความถูกต้องของการจำลอง

3.4.1 การทดสอบความถูกต้องของการจำลองกลไกถังรับแบบไม่มีบัฟเฟอร์

ทำการทดสอบความสามารถในการเลือกของกลไกด้วยการวัดค่าอัตราส่วนการสูญหายของเซลล์ของกราฟฟิกที่มีขนาดโหลดต่างๆ กำหนดให้อัตราเซลล์เฉลี่ยระยะยาวของแหล่งกำเนิดที่นอร์มอลไลซ์แล้ว $\sigma \in \{1.00, 1.05, 1.10, 1.20, \dots, 2.00\}$ ใช้วิธีแปรขนาดโหลด 2 วิธีได้แก่ แปรค่า $E(X)$ โดยให้ $E(S)$ คงที่ และแปรค่า $E(S)$ โดยให้ $E(X)$ คงที่ ทั้งนี้อัตราเซลล์ค่าออกยังคงไม่

เปลี่ยนแปลง เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 ตามรายละเอียดในหัวข้อที่ 3.1 จำนวน 15 ตัวอย่างต่อ σ 1 ค่า แต่ละตัวอย่างเป็นจำนวน 10^9 เซลล์ คำนวณค่าเฉลี่ยอัตราส่วนการสูญหายของเซลล์ที่วัดได้สำหรับ σ แต่ละค่า และเปรียบเทียบกับผลการจำลองของ [1] ได้ผลการทดสอบแสดงดังรูปที่ 3.6(ก) และ 3.6(ข)



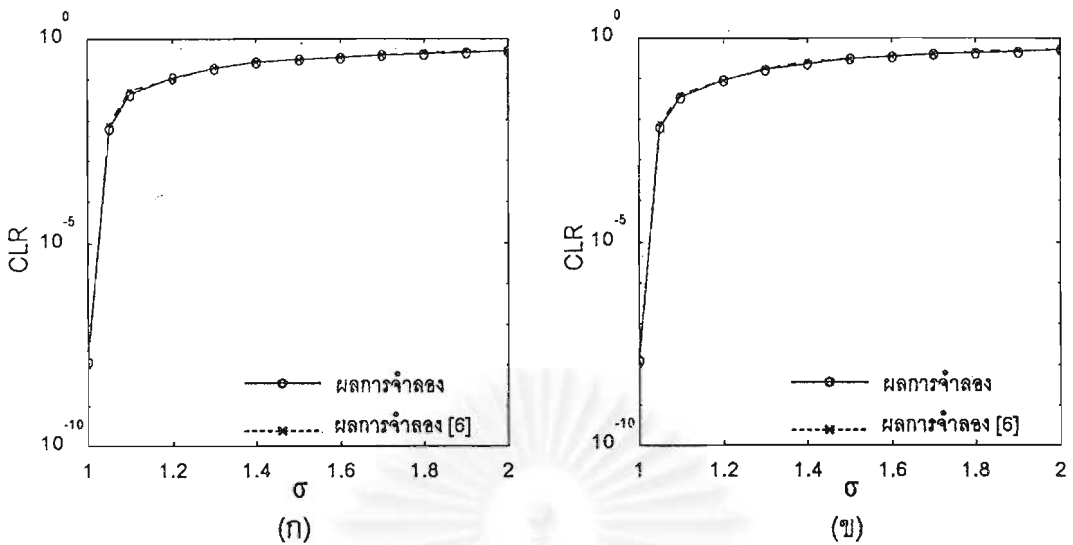
รูปที่ 3.6 ผลเปรียบเทียบการจำลองกลไกถึงรั้ว เมื่อกำหนดให้ (ก) $E(S)$ คงที่ (ข) $E(X)$ คงที่

จากรูปที่ 3.6 คำนวณค่าผิดพลาดเฉลี่ยเทียบกับผลการจำลอง [1] ได้เท่ากับ 5.64% เมื่อให้ $E(S)$ คงที่ และ 9.75% เมื่อให้ $E(X)$ คงที่

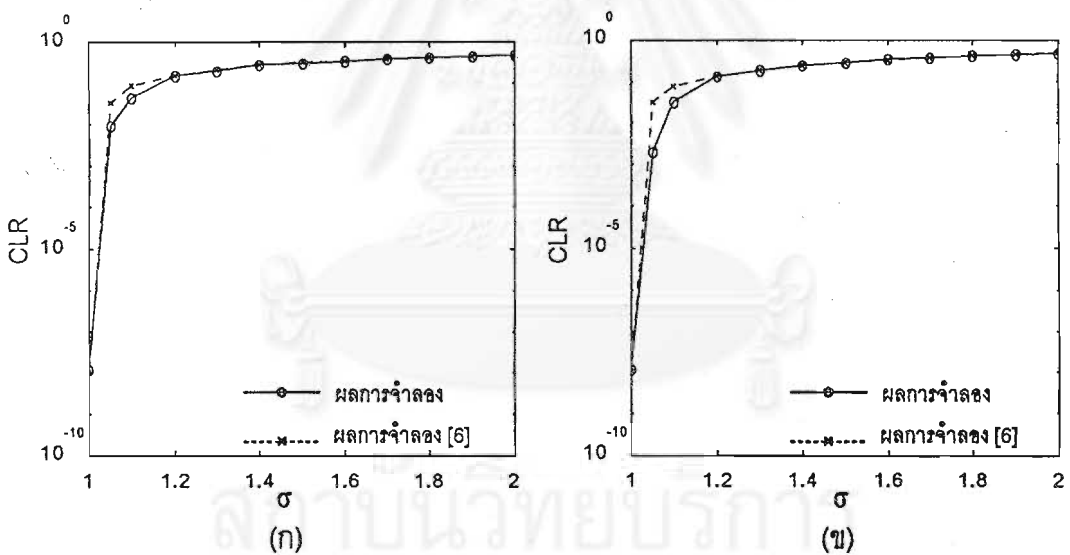
ความแตกต่างของผลการจำลองในวิทยานิพนธ์กับผลการจำลอง [1] อยู่ในระดับที่ยอมรับได้

3.4.2 การทดสอบความถูกต้องของการจำลองกลไกหน้าต่างพีซีลอคจิก

ในการทดสอบความสามารถในการเลือก เราจะกำหนดให้ $\sigma \in \{1.00, 1.05, 1.10, \dots, 2.00\}$ โดยใช้แบบจำลองแหล่งกำเนิดแบบที่ 1 และแบบที่ 2 จำนวน 15 ตัวอย่างต่อ σ 1 ค่า แต่ละตัวอย่างเป็นจำนวน 10^9 เซลล์ คำนวณเฉลี่ยอัตราส่วนการสูญหายของเซลล์ที่วัดได้สำหรับ σ แต่ละค่า และเปรียบเทียบกับผลการจำลองของ [6] ได้ผลการทดสอบแสดงดังรูปที่ 3.7(ก), 3.7(ข), 3.8(ก) และ 3.8(ข)



รูปที่ 3.7 ผลเปรียบเทียบการจำลองกลไกหน้าต่างพีชชีลจิกโดยใช้แบบจำลอง
แหล่งกำเนิดแบบที่ 1 เมื่อกำหนดให้ (ก) E(S) คงที่ (ข) E(X) คงที่



รูปที่ 3.8 ผลเปรียบเทียบการจำลองกลไกหน้าต่างพีชชีลจิกโดยใช้แบบจำลอง
แหล่งกำเนิดแบบที่ 2 เมื่อกำหนดให้ (ก) E(S) คงที่ (ข) E(X) คงที่

จากรูปที่ 3.7(ก) และ 3.7(ข) คำนวนค่าผิดพลาดเฉลี่ยของการจำลองเมื่อใช้แบบจำลอง
แหล่งกำเนิดแบบที่ 1 ได้เท่ากับ 6.13% เมื่อให้ E(S) คงที่ และ 7.33% เมื่อให้ E(X) คงที่ และจากรูป
ที่ 3.8(ก) และ 3.8(ข) เมื่อใช้แบบจำลองแหล่งกำเนิดแบบที่ 2 คำนวนค่าผิดพลาดเฉลี่ยได้เท่ากับ
12.8% เมื่อให้ E(S) คงที่ และ 14.8% เมื่อให้ E(X) คงที่

สรุปได้ว่าการจำลองแหล่งกำเนิดและแบบจำลองกลไกมีความถูกต้อง
ทดสอบรวมทั้งแนวโน้มใกล้เคียงกับงานวิจัยที่อ้างอิง

เพราะให้ผลการ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ผลการทดสอบประสิทธิภาพกลไกตรวจกราฟฟิกโดยใช้พีซีลोजิก และการวิเคราะห์ผลการทดสอบ

4.1 กล่าวนำ

คุณสมบัติหลักของตัวตรวจกราฟฟิกที่จะนำมาใช้งานคือ ความสามารถในการเลือก (Selectivity) สูง และการตอบสนอง (Responsiveness) ที่รวดเร็ว แต่ในทางปฏิบัติแล้ว แนวทางในการพัฒนาตัวตรวจที่มีคุณสมบัติดังกล่าวพร้อมกันทั้งสองประการนั้นแทบเป็นไปไม่ได้อย่างสิ้นเชิง โดยเฉพาะอย่างยิ่งในการตรวจอัตราเซลล์เฉลี่ย เนื่องจากเราต้องการเวลาในการประเมินและตัดสินใจลักษณะกราฟฟิก (traffic characteristics) ยาวนานเพื่อให้ได้ความสามารถในการเลือกสูง จึงทำให้การตรวจมีการตอบสนองช้า เพื่อแก้ปัญหานี้เราเสนอให้ใช้กลไกถังรั้วแบบโทเค็นโดยอัตรากำเนิดโทเค็นปรับค่าได้ด้วยการควบคุมจากพีซีลोजิก และใช้นิวรอลเน็ตเวิร์กในการหาความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตของ inference engine ของพีซีลोजิกเพื่อลดเวลาในการประมวลผล อย่างไรก็ตาม ก่อนขั้นตอนการฝึกนิวรอลเน็ตเวิร์กที่ใช้เวลานาน ควรทดสอบและปรับปรุงกลไกตรวจกราฟฟิกที่เสนอโดยใช้ตัวควบคุมพีซีลोजิกเสียก่อน เพื่อให้มั่นใจได้ว่ากลไกมีประสิทธิภาพตามที่ต้องการและไม่เสียเวลาฝึกโดยเปล่าประโยชน์

4.2 ผลของขนาดของบ่อโทเค็นต่อประสิทธิภาพของกลไกถังรั้วแบบโทเค็นโดยใช้พีซีลोजิก

ดังที่ได้กล่าวไปแล้วในบทที่ 3 ฟังก์ชันสมาชิกและกฎพื้นฐานพีซีลोजิกหนึ่งถูกออกแบบเพื่อให้เหมาะสมสำหรับขนาดของบัฟเฟอร์และคาบเวลาชักตัวอย่างค่าหนึ่งเท่านั้น กล่าวคือการเปลี่ยนแปลงพารามิเตอร์ดังกล่าวเพียงเล็กน้อยจะทำให้ประสิทธิภาพของกลไกเปลี่ยนแปลงไปมาก ในขณะที่บ่อโทเค็นซึ่งทำหน้าที่เพียงเก็บโทเค็นส่วนเกินในสภาวะบัฟเฟอร์ว่าง มีขนาดที่ทำให้ประสิทธิภาพ (การตรวจวัดโหลดเกินและการตอบสนอง) อยู่ในช่วงที่ต้องการได้หลายขนาดภายใต้ฟังก์ชันพีซีและกฎพื้นฐานพีซีลोजิกหนึ่ง ดังนั้นขนาดของบ่อโทเค็นที่กำหนดไว้ในหัวข้อ 4.3 เพื่อทดสอบหาฟังก์ชันสมาชิกและกฎพื้นฐานพีซีลोजิกในขั้นตอน inference อาจไม่ใช่ขนาดที่เหมาะสมที่สุด

การหาขนาดบ่อโทเค็นที่เหมาะสมแบ่งเป็น 2 กรณีคือ เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 และแบบจำลองแหล่งกำเนิดแบบที่ 2 เปรียบเทียบในการพิจารณาประกอบด้วย

1. เส้นโค้งความสามารถในการตรวจวัดโหลดเกินใกล้เคียงกับเส้นโค้งการละเมิดอุดมคติที่สุด กล่าวคือรากที่ 2 ของผลรวมค่าผิดพลาดกำลังสอง (RSSE) มีค่าต่ำที่สุด
2. CLR ที่โหลดเกินจะต้องไม่เกินความน่าจะเป็นของการละเมิดอุดมคติ
3. มี CLR ที่โหลดปกติ ($\sigma = 1.0$) ต่ำกว่ากลไกถังรับแบบไม่มีบัฟเฟอร์
4. มีการตอบสนองที่รวดเร็วที่สุด กล่าวคือมีจำนวนเซลล์ที่ยอมให้เข้าสู่โครงข่ายต่ำที่สุดนับแต่เริ่มต้นการตรวจกราฟฟิกที่ละเมิดข้อตกลง และตอบสนองเร็วกว่ากลไกหน้าต่างฟิชซีลจิกโดยต้องไม่ละเมิดเงื่อนไขข้อที่ 2 และ 3

4.2.1 กลไกตรวจกราฟฟิกเมื่อใช้แบบจำลองแหล่งกำเนิดแบบที่ 1

4.2.1.1 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก

วัดค่าอัตราส่วนการสูญหายของเซลล์ซึ่งเกิดจากกลไกตรวจกราฟฟิกที่มีขนาดบ่อโตะเต็ม (P) ต่างๆ กำหนดให้ $P \in \{500, 750, 1000, 1250\}$ ทดสอบกับกราฟฟิกที่มีอัตราเซลล์เฉลี่ยระยะยาวนอร์มอลไลซ์เทียบกับอัตราเซลล์เฉลี่ยที่ตกลงไว้ (σ) ต่างๆ กัน โดย $\sigma \in \{1.00, 1.05, 1.10, \dots, 2.00\}$ จำนวน 15 ตัวอย่างต่อ σ 1 ค่า แต่ละตัวอย่างประกอบด้วย 10^6 เซลล์ การแปรค่า σ แบ่งได้เป็น 2 วิธี คือ แปรค่า $E(X)$ โดยให้ $E(S)$ คงที่ และแปรค่า $E(S)$ โดยให้ $E(X)$ คงที่

ผลการทดสอบความสามารถในการตรวจวัดโหลดเกินของกลไกถังรับฟิชซีลจิกที่เสนอ (FLB) เปรียบเทียบกับกลไกถังรับ (LB), กลไกหน้าต่างฟิชซีลจิก (Fwindow) และเส้นโค้งความน่าจะเป็นการละเมิดอุดมคติ (ideal) เมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด แสดงได้ดังตารางที่ 4.1 โดยมีกราฟดังรูปที่ 4.1(ก)-4.1(ง) และเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด แสดงได้ดังตารางที่ 4.2 โดยมีกราฟดังรูปที่ 4.2(ก)-4.2(ง)

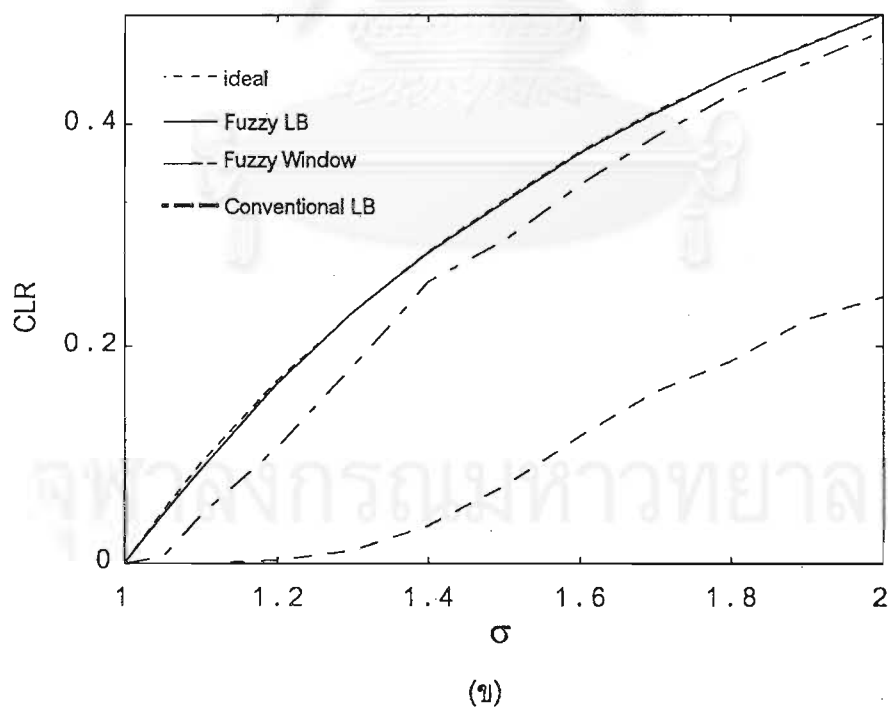
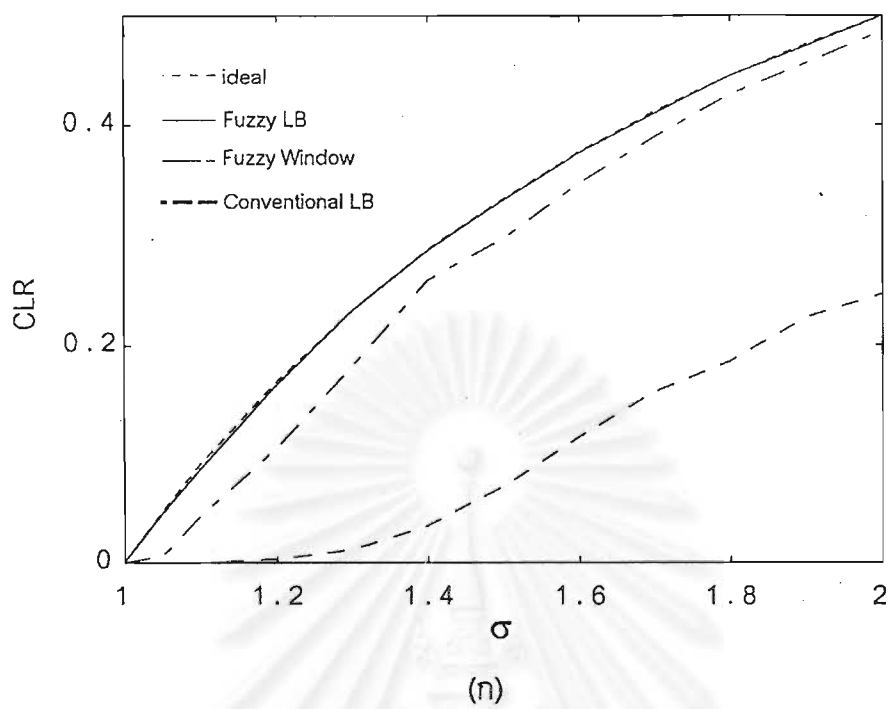
ความแตกต่างระหว่างความสามารถในการตรวจวัดโหลดเกินอุดมคติกับของกลไกวัดจากรากของผลรวมค่าผิดพลาดกำลังสอง (Root of Sum Square Error : RSSE) ซึ่งได้จากการคำนวณผลรวมของผลต่างระหว่าง CLR ที่เกิดจากกลไกตรวจกราฟฟิก (CLR_m) กับความน่าจะเป็นของการละเมิดอุดมคติ (P_d) ของกราฟฟิกแต่ละ σ ดังสมการที่ (4.1) ผลการคำนวณแสดงไว้ในตารางที่ 4.3

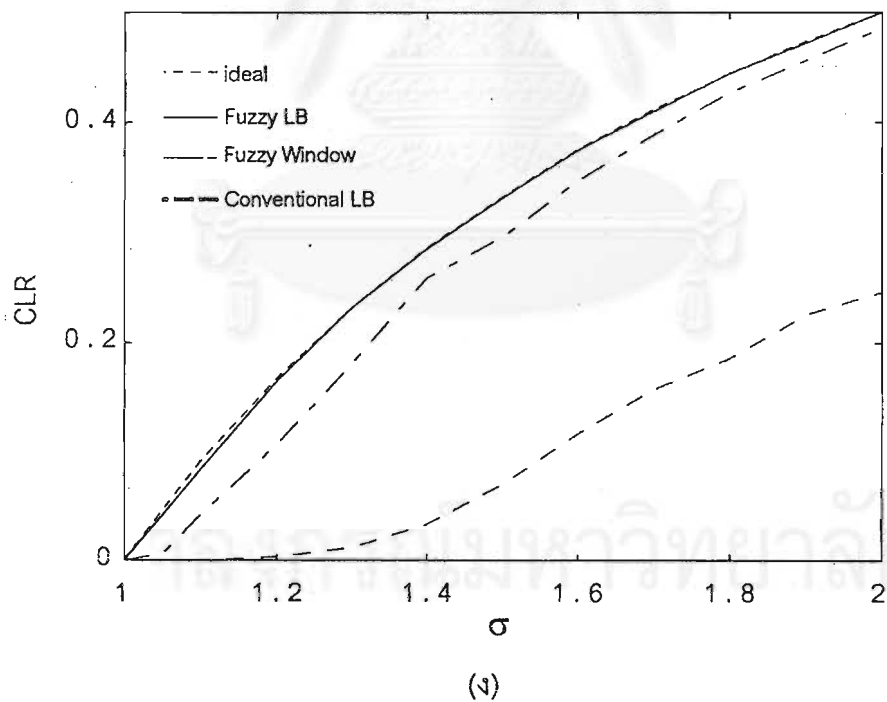
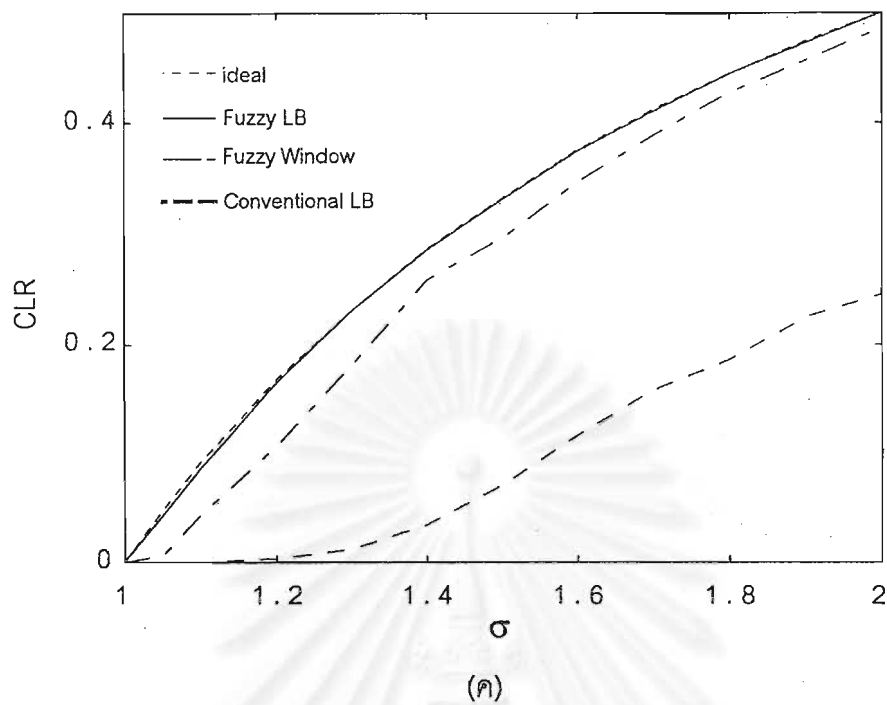
$$RSSE = \sqrt{\sum_{\sigma} (P_d - CLR_m)^2} \quad (4.1)$$

ดังนั้น หากตัวตรวจกราฟฟิกมีพฤติกรรมอุดมคติ RSSE จะมีค่าเท่ากับ 0 และจะมีค่ามากขึ้น หากตัวตรวจกราฟฟิกมีพฤติกรรมต่างไปจากพฤติกรรมอุดมคติเพิ่มขึ้น

ตารางที่ 4.1 อัตราส่วนการสูญหายของเซลล์ซึ่งเกิดจากกลไกตรวจหาฟลักที่ σ
 ต่างๆ เมื่อแปรค่า $E(X)$

| σ | อัตราส่วนการสูญหายของเซลล์ (CLR) | | | | | | |
|----------|----------------------------------|-----------------------|-------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | อุดมคติ | ถึงรั้ว | หน้าต่าง ฟลักซ์ลอจิก | ถึงรั้วฟลักซ์ลอจิก | | | |
| | | | | P=500 | P=750 | P=1,000 | P=1,250 |
| 1.00 | 0 | 2.72×10^{-5} | 1.20×10^{-6} | 5.68×10^{-7} | 2.28×10^{-7} | 1.48×10^{-7} | 1.39×10^{-7} |
| 1.05 | 0.047619 | 1.23×10^{-4} | 0.006251 | 0.045453 | 0.043901 | 0.042832 | 0.042044 |
| 1.10 | 0.090909 | 4.18×10^{-4} | 0.042459 | 0.086023 | 0.085289 | 0.084943 | 0.084089 |
| 1.15 | 0.130435 | 0.001046 | 0.075721 | 0.126144 | 0.125871 | 0.125900 | 0.125466 |
| 1.20 | 0.166667 | 0.002650 | 0.106194 | 0.164571 | 0.164284 | 0.164422 | 0.164316 |
| 1.25 | 0.200000 | 0.006442 | 0.141128 | 0.199337 | 0.199143 | 0.199102 | 0.199187 |
| 1.30 | 0.230769 | 0.012370 | 0.180980 | 0.230471 | 0.230456 | 0.230455 | 0.230473 |
| 1.35 | 0.259260 | 0.020907 | 0.222902 | 0.259112 | 0.259004 | 0.258977 | 0.258852 |
| 1.40 | 0.285714 | 0.033161 | 0.257624 | 0.284840 | 0.284569 | 0.284452 | 0.284309 |
| 1.45 | 0.310345 | 0.052291 | 0.278539 | 0.308322 | 0.308321 | 0.308316 | 0.308318 |
| 1.50 | 0.333333 | 0.070369 | 0.295815 | 0.330780 | 0.330762 | 0.330773 | 0.330725 |
| 1.55 | 0.354839 | 0.092810 | 0.320116 | 0.353285 | 0.353282 | 0.353277 | 0.253266 |
| 1.60 | 0.375000 | 0.116038 | 0.346629 | 0.374439 | 0.374422 | 0.374418 | 0.374411 |
| 1.65 | 0.393939 | 0.138423 | 0.369145 | 0.393489 | 0.393486 | 0.393486 | 0.393472 |
| 1.70 | 0.411765 | 0.157175 | 0.388776 | 0.411108 | 0.411107 | 0.411101 | 0.410993 |
| 1.75 | 0.428571 | 0.171456 | 0.408327 | 0.428220 | 0.428218 | 0.428218 | 0.428216 |
| 1.80 | 0.444444 | 0.185022 | 0.426460 | 0.444197 | 0.444193 | 0.444191 | 0.444185 |
| 1.85 | 0.459459 | 0.203943 | 0.441917 | 0.458639 | 0.458633 | 0.458629 | 0.458627 |
| 1.90 | 0.473684 | 0.227428 | 0.455653 | 0.472099 | 0.472095 | 0.472096 | 0.472063 |
| 1.95 | 0.487179 | 0.239408 | 0.469620 | 0.485416 | 0.485416 | 0.485408 | 0.485392 |
| 2.00 | 0.500000 | 0.244200 | 0.485390 | 0.499566 | 0.499565 | 0.499563 | 0.499561 |



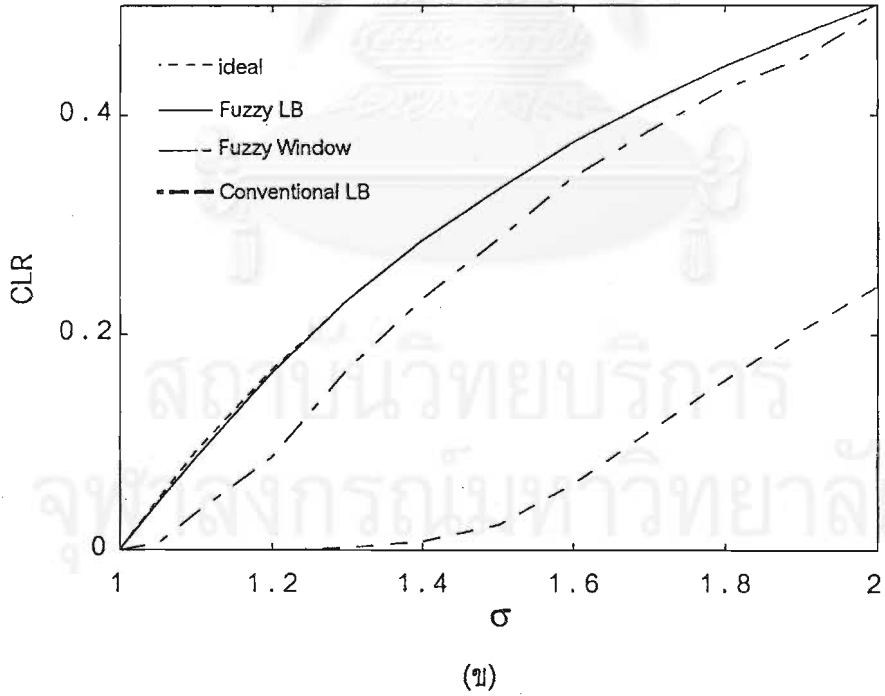
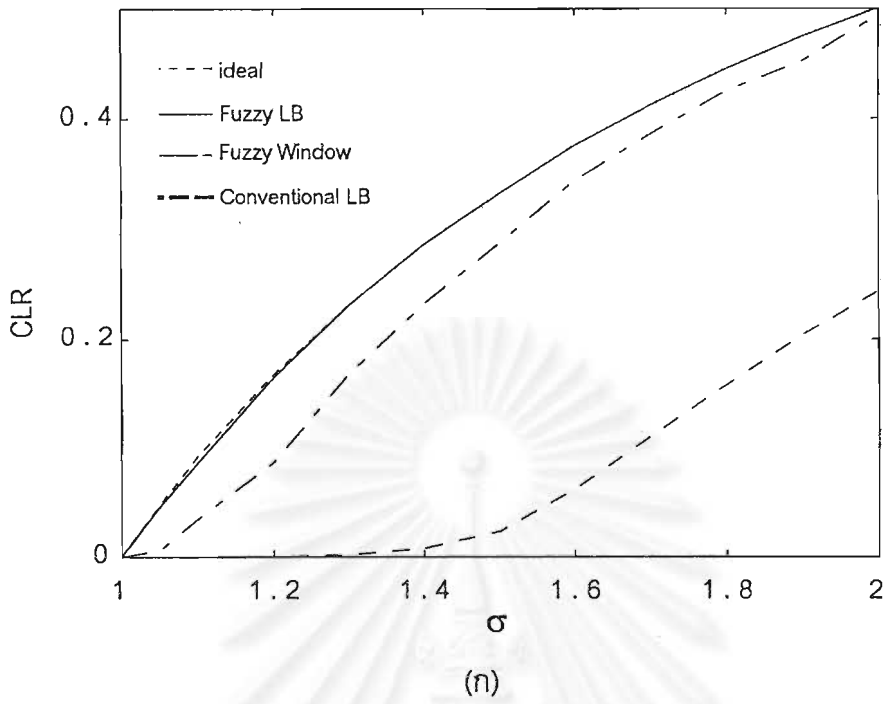


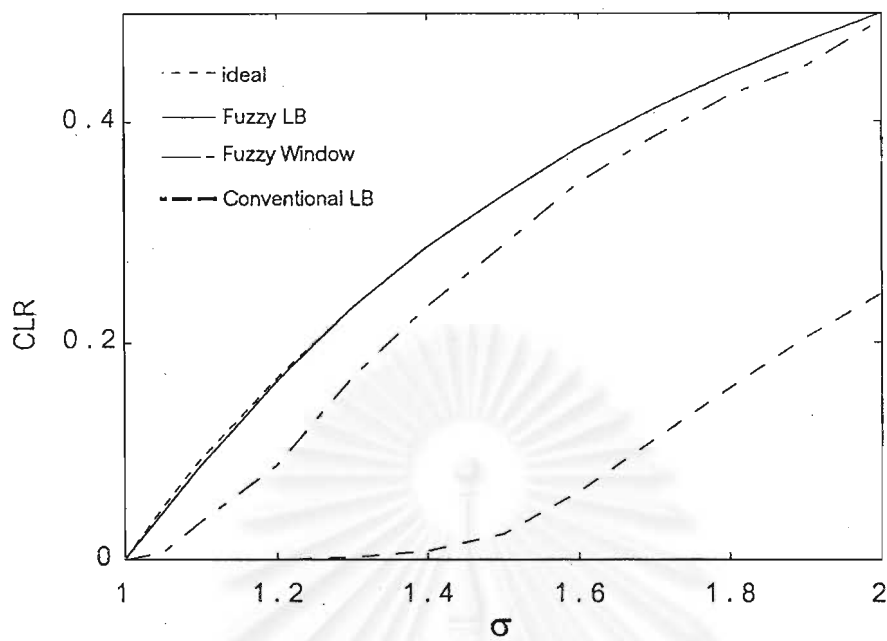
รูปที่ 4.1 ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจหาวัดไฟฟ้าเมื่อแปรค่า $E(X)$

(ก) $P = 500$ (ข) $P = 750$ (ค) $P = 1,000$ (ง) $P = 1,250$

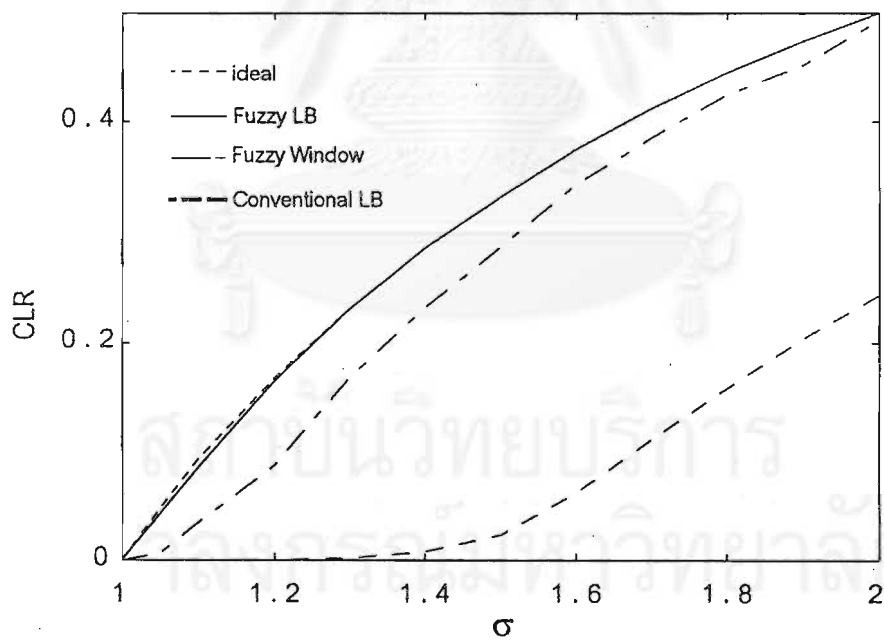
ตารางที่ 4.2 อัตราส่วนการสูญหายของเซลล์ซึ่งเกิดจากกลไกตรวจหาฟลักที่ σ ต่างๆ เมื่อแปรค่า $E(X)$

| σ | อัตราส่วนการสูญหายของเซลล์ (CLR) | | | | | | |
|----------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | อุดมคติ | ถึงรั้ว | หน้าต่าง ฟิชชีลจิก | ถึงรั้วฟิชชีลจิก | | | |
| | | | | P=500 | P=750 | P=1,000 | P=1,250 |
| 1.00 | 0 | 2.72×10^{-5} | 1.20×10^{-8} | 5.68×10^{-7} | 2.28×10^{-7} | 1.48×10^{-7} | 1.39×10^{-7} |
| 1.05 | 0.047619 | 5.20×10^{-5} | 0.006120 | 0.045327 | 0.04396 | 0.042911 | 0.041928 |
| 1.10 | 0.090909 | 8.40×10^{-5} | 0.033800 | 0.085716 | 0.084884 | 0.084937 | 0.084181 |
| 1.15 | 0.130435 | 1.50×10^{-4} | 0.058700 | 0.126247 | 0.126233 | 0.126397 | 0.125945 |
| 1.20 | 0.166667 | 0.000345 | 0.086226 | 0.164925 | 0.164942 | 0.164861 | 0.164853 |
| 1.25 | 0.200000 | 0.000900 | 0.124920 | 0.199142 | 0.199212 | 0.199130 | 0.199125 |
| 1.30 | 0.230769 | 0.001890 | 0.166720 | 0.229893 | 0.229933 | 0.229896 | 0.229926 |
| 1.35 | 0.259260 | 0.003600 | 0.202346 | 0.258841 | 0.258850 | 0.258912 | 0.258924 |
| 1.40 | 0.285714 | 0.006899 | 0.232240 | 0.285585 | 0.285585 | 0.285585 | 0.285585 |
| 1.45 | 0.310345 | 0.012900 | 0.259714 | 0.309731 | 0.309744 | 0.309700 | 0.309722 |
| 1.50 | 0.333333 | 0.023442 | 0.287270 | 0.332208 | 0.332217 | 0.332189 | 0.332194 |
| 1.55 | 0.354839 | 0.039700 | 0.316433 | 0.354130 | 0.354132 | 0.354117 | 0.354022 |
| 1.60 | 0.375000 | 0.060940 | 0.343570 | 0.374911 | 0.374915 | 0.374917 | 0.374902 |
| 1.65 | 0.393939 | 0.085400 | 0.365627 | 0.393812 | 0.393845 | 0.393954 | 0.393947 |
| 1.70 | 0.411765 | 0.110488 | 0.385410 | 0.411419 | 0.411476 | 0.411452 | 0.411403 |
| 1.75 | 0.428571 | 0.134200 | 0.405811 | 0.428200 | 0.428300 | 0.428277 | 0.428259 |
| 1.80 | 0.444444 | 0.157248 | 0.424480 | 0.444280 | 0.444277 | 0.444289 | 0.444163 |
| 1.85 | 0.459459 | 0.180900 | 0.439212 | 0.459416 | 0.459411 | 0.459410 | 0.459228 |
| 1.90 | 0.473684 | 0.204011 | 0.452290 | 0.473599 | 0.473591 | 0.473595 | 0.473568 |
| 1.95 | 0.487179 | 0.225200 | 0.468919 | 0.487110 | 0.487111 | 0.487104 | 0.487065 |
| 2.00 | 0.500000 | 0.242866 | 0.493212 | 0.499866 | 0.499857 | 0.499849 | 0.499842 |





(ค)



(ง)

รูปที่ 4.2 ความสามารถในการตรวจวัดโหนดเกินของกลไกตรวจกราฟฟิกเมื่อแปรค่า $E(S)$

(ก) $P = 500$ (ข) $P = 750$ (ค) $P = 1,000$ (ง) $P = 1,250$

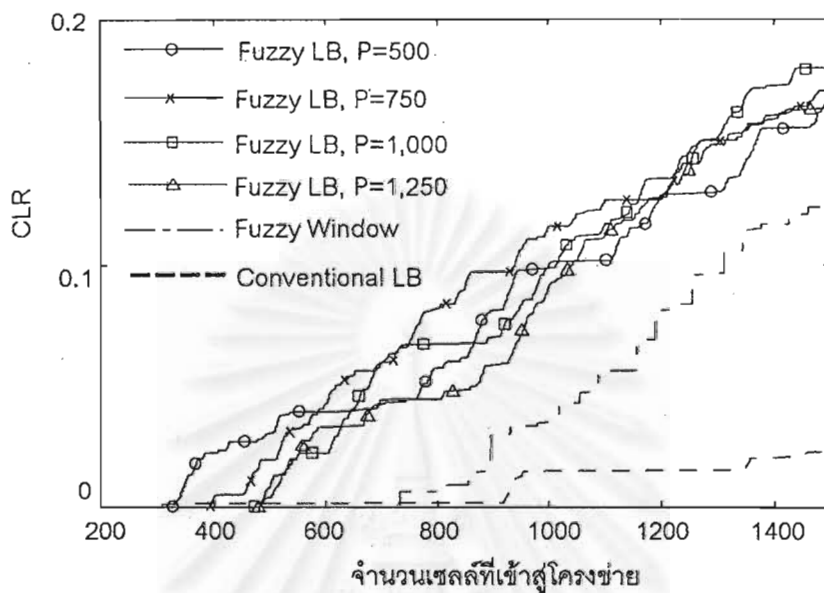
ตารางที่ 4.3 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจ
ทราบฟีกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1

| กลไก | ถึงรั้ว | หน้าต่าง ฟิชชีลลจิก | ถึงรั้วฟิชชีลลจิก | | | |
|------------------|----------|------------------------|-------------------|----------|----------|----------|
| | | | P=500 | P=750 | P=1,000 | P=1,250 |
| RSSE แปร E(X) | 1.023511 | 0.161944 | 0.008563 | 0.009743 | 0.010369 | 0.010221 |
| แปร E(S) | 1.145818 | 0.207957 | 0.007548 | 0.008612 | 0.009030 | 0.010270 |

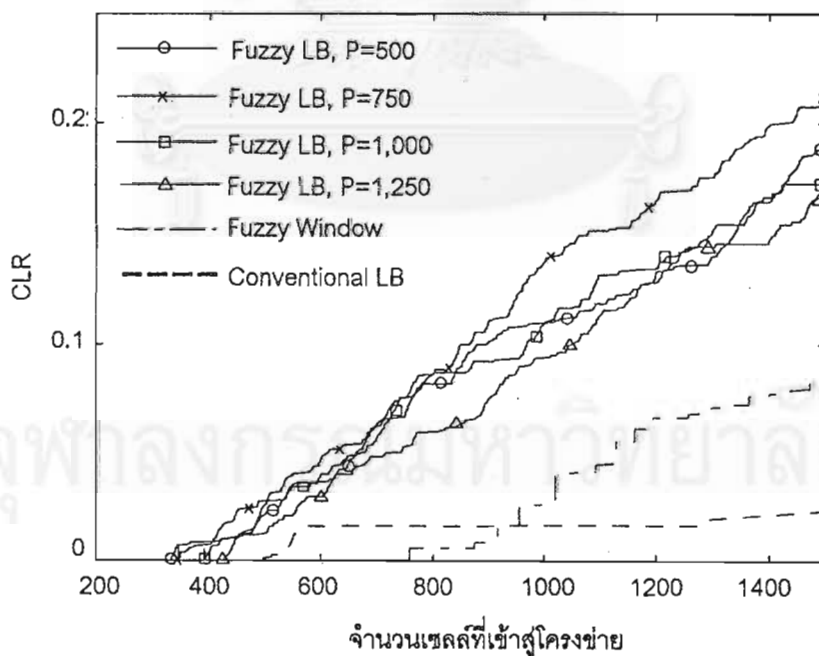
จากตารางที่ 4.3 RSSE มีค่าต่ำสุดเมื่อ $P = 500$ ในขณะที่จากตารางที่ 4.1 และ 4.2 CLR ที่โหลดปกติมีค่าต่ำสุดเมื่อ $P=1,250$ กลไกถึงรั้วฟิชชีลลจิกสำหรับบ่อโทเค็นทุกขนาด มี RSSE ต่ำกว่าถึงรั้วและหน้าต่างฟิชชีลลจิกมาก กล่าวคือ ต่ำกว่าถึงรั้วโดยเฉลี่ย 100 เท่า และต่ำกว่าหน้าต่างฟิชชีลลจิกโดยเฉลี่ย 20 เท่า ดังจะเห็นได้จากรูปที่ 4.1(ก)-4.2(ง) ที่เส้นโค้งความสามารถในการตรวจวัดโหลดเกินของกลไกที่เสนอใกล้เคียงกับเส้นโค้งอุดมคติว่ากลไกอื่นๆ มาก แต่การเพิ่มประสิทธิภาพการวัดโหลดเกินดังกล่าวทำให้ CLR ที่โหลดปกติ หรือ false alarm probability เพิ่มขึ้นไปด้วย คือสูงกว่าหน้าต่างฟิชชีลลจิก 22 เท่า แต่ยังคงต่ำกว่าถึงรั้วเกือบ 100 เท่า และ CLR ดังกล่าวยังคงอยู่ในระดับที่ยอมรับได้สำหรับทราบฟีกประเภท packet voice คือ 10^{-3}

4.2.1.2 ผลการทดสอบเปรียบเทียบการตอบสนอง

ผลการวัดอัตราส่วนการสูญหายของเซลล์เทียบกับจำนวนเซลล์ที่ตัวตรวจยอมให้ผ่านเข้าไปยังโครงข่าย เมื่อให้ $\sigma = 1.50$ [6] และเปรียบเทียบระหว่างกลไกถึงรั้วโดยใช้ฟิชชีลลจิกที่ขนาดบ่อโทเค็น (P) ต่างๆ กับกลไกถึงรั้วแบบไม่มีบัฟเฟอร์ซึ่งเป็นกลไกที่มีการตอบสนองเร็วที่สุดและกลไกหน้าต่างฟิชชีลลจิก โดย $P \in \{500, 750, 1000, 1250\}$ แสดงดังรูปที่ 4.3 เมื่อแปรค่า E(X) และรูปที่ 4.4 เมื่อแปรค่า E(S) การเปรียบเทียบจำนวนเซลล์ที่กลไกตรวจทราบฟีกยอมให้ผ่านเข้าสู่โครงข่ายก่อนละทิ้งเซลล์แรกแสดงไว้ในตารางที่ 4.4



รูปที่ 4.3 การตอบสนองของกลไกตรรกะฟัฟก เมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



รูปที่ 4.4 การตอบสนองของกลไกตรรกะฟัฟก เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

ตารางที่ 4.4 เปรียบเทียบการตอบสนองเริ่มต้นของกลไกตรวจหาฟลัก เมื่อทดสอบกับ
แบบจำลองแหล่งกำเนิดแบบที่ 1

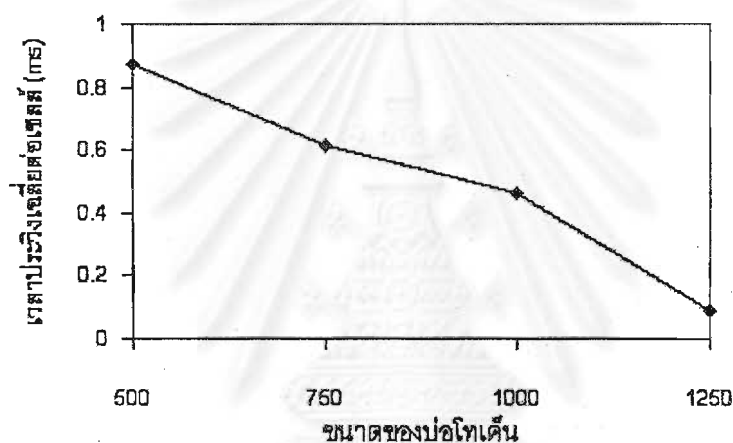
| จำนวนเซลล์ที่เข้าสู่ โครงข่ายก่อนการละ ทิ้งเซลล์แรก | ถึงรั้ว | หน้าต่าง ฟลักซ์- ลोजิก | ถึงรั้วฟลักซ์ลोजิก | | | |
|---|---------|------------------------------|--------------------|-------|---------|---------|
| | | | P=500 | P=750 | P=1,000 | P=1,250 |
| เมื่อแปรค่า E(X) | 309 | 731 | 328 | 394 | 472 | 482 |
| เมื่อแปรค่า E(S) | 496 | 757 | 334 | 346 | 392 | 424 |

จากตารางที่ 4.4 เปรียบเทียบปฏิกิริยาตอบสนองต่อการละเมิดเริ่มต้น พบว่าการเพิ่มขนาดของบ่อโทเค็นทำให้การตอบสนองเริ่มแรกช้าลง ทั้งที่ขนาดของบ่อโทเค็นไม่ใช่ตัวแปรที่มีผลโดยตรงต่อการตอบสนอง หากแต่เป็นอัตราการทำเนิดโทเค็นและขนาดของบัฟเฟอร์ แต่เนื่องจากกำหนดให้สัดส่วนของจำนวนของโทเค็นในบ่อเทียบกับขนาดของบ่อเป็นอินพุตตัวหนึ่งของตัวควบคุม ดังนั้นแม้ว่าจำนวนโทเค็นสะสมเท่ากัน แต่อินพุตดังกล่าวจะไม่เท่ากัน จากการกำหนดกฎฟลักซ์ด้วยวิธี hand optimization ในช่วงเวลาสะสมโทเค็น หากสัดส่วนของโทเค็นสะสมเทียบกับขนาดของบ่อมีค่ามากจะให้อัตราการทำเนิดโทเค็นต่ำกว่าในกรณีที่สัดส่วนของโทเค็นสะสมเทียบกับขนาดของบ่อมีค่าน้อย ทั้งนี้เพื่อป้องกันให้ตัวควบคุมไม่ผลิตโทเค็นเกินความจำเป็น ดังนั้นจึงมีแนวโน้มที่เมื่อกำหนดให้ขนาดของบ่อโทเค็นเพิ่มขึ้นจะทำให้จำนวนโทเค็นสะสมในช่วงเวลาใดๆ สูงขึ้น และเป็นผลให้เมื่อกราฟฟลักซ์ลोजิกออกตกลงเข้าสู่โครงข่าย กลไกที่มีบ่อโทเค็นใหญ่กว่าจะตรวจวัดพบช้ากว่า อย่างไรก็ตาม จากรูปที่ 4.3 และ 4.4 พฤติกรรมการตอบสนองต่อการละเมิดระยะยาวของกลไกถึงรั้วฟลักซ์ลोजิกที่มีขนาดบ่อโทเค็นแตกต่างกัน ไม่แตกต่างกันมากนักและเห็นได้ว่าอัตราการผลิตเซลล์ภายหลังการตรวจวัดพบเซลล์แรกเพิ่มขึ้นอย่างรวดเร็ว เมื่อเปรียบเทียบกับกลไกถึงรั้วและหน้าต่างฟลักซ์ลोजิก แม้ว่าการตรวจพบเซลล์แรกจะช้ากว่าถึงรั้วอยู่เล็กน้อยในกรณีที่แหล่งกำเนิดละเมิดข้อตกลงโดยวิธีแปรค่า E(X) แต่พฤติกรรมการตอบสนองโดยรวมของถึงรั้วฟลักซ์ลोजิกยังคงดีกว่ากลไกอื่นมาก

เมื่อเปรียบเทียบลักษณะของกราฟฟลักซ์ลोजิกที่ละเมิดข้อตกลง ถึงรั้วแบบดั้งเดิมจะตอบสนองต่อการเพิ่มกราฟฟลักซ์ลोजิกโดยวิธีลดระยะเวลาเรียบเฉลี่ยหรือ E(S) ช้ากว่าวิธีเพิ่มจำนวนเซลล์เฉลี่ยในแต่ละเบิรสต์หรือ E(X) มาก เพราะกราฟฟลักซ์ลोजิกที่ลดระยะเวลาเรียบลงในขณะที่จำนวนเซลล์เฉลี่ยยังคงที่ ยังคงมีช่วงเวลาพักให้ตัวนับได้ลดลงบ่อยครั้งกว่ากราฟฟลักซ์ลोजิกที่ละเมิดข้อตกลงด้วยการเพิ่มจำนวนเซลล์เฉลี่ย แต่เมื่อนำฟลักซ์ลोजิกมาใช้ ทำให้การตอบสนองต่อการละเมิดทั้ง 2 แบบ แตกต่างกันอย่างเห็นได้ชัดทั้งในหน้าต่างฟลักซ์ลोजิกและถึงรั้วฟลักซ์ลोजิก

4.2.1.3 เวลาประวิงที่เกิดจากกลไกตรวจกราฟฟีก

เนื่องจากโครงสร้างของกลไกตรวจกราฟฟีกที่เสนอมีคิวบัฟเฟอร์ด้านเข้า ทำให้ไม่สามารถหลีกเลี่ยงการประวิงเวลาเนื่องจากการรอคอยได้ จึงเป็นเหตุผลที่ต้องจำกัดขนาดของคิวบัฟเฟอร์ให้น้อยที่สุดเท่าที่จะเป็นไปได้ โดยไม่ทำให้ CLR ที่ไหลตกติ หรือ false alarm probability สูงจนเกินไปดังที่กล่าวไปแล้วในหัวข้อที่ 3.3 ผลการทดสอบวัดเวลาประวิงเฉลี่ยต่อ 1 เซลล์ของกราฟฟีกแสดงได้ดังรูปที่ 4.5 ในขณะที่กลไกที่ใช้เปรียบเทียบคือถังรับแบบไม่มีบัฟเฟอร์และหน้าต่างฟิชชิลอจิกไม่มีเวลาประวิงที่เกิดจากกลไก



รูปที่ 4.5 เวลาประวิงที่เกิดจากกลไกตรวจกราฟฟีกที่ไหลตกติ

จากรูปที่ 4.5 พบว่าเวลาประวิงเฉลี่ยต่อเซลล์ลดลงเมื่อขนาดบ่อโทเค็นเพิ่มขึ้น อย่างไรก็ตาม เวลาประวิงเฉลี่ยสูงสุดที่ $P=500$ ยังคงมีค่าไม่เกิน 1 ms

4.2.2 กลไกตรวจกราฟฟีกเมื่อใช้แบบจำลองแหล่งกำเนิดแบบที่ 2

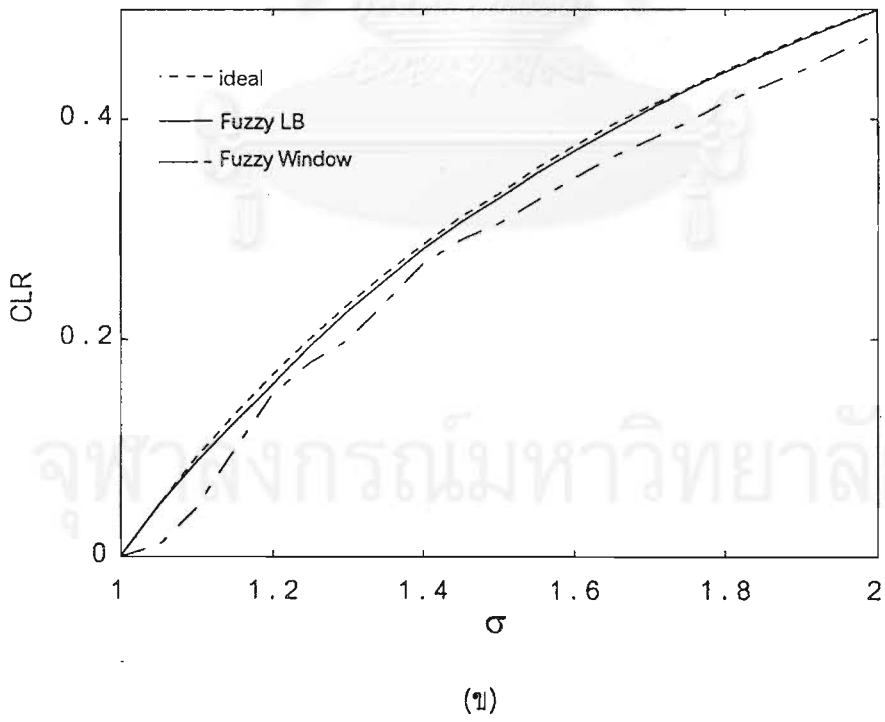
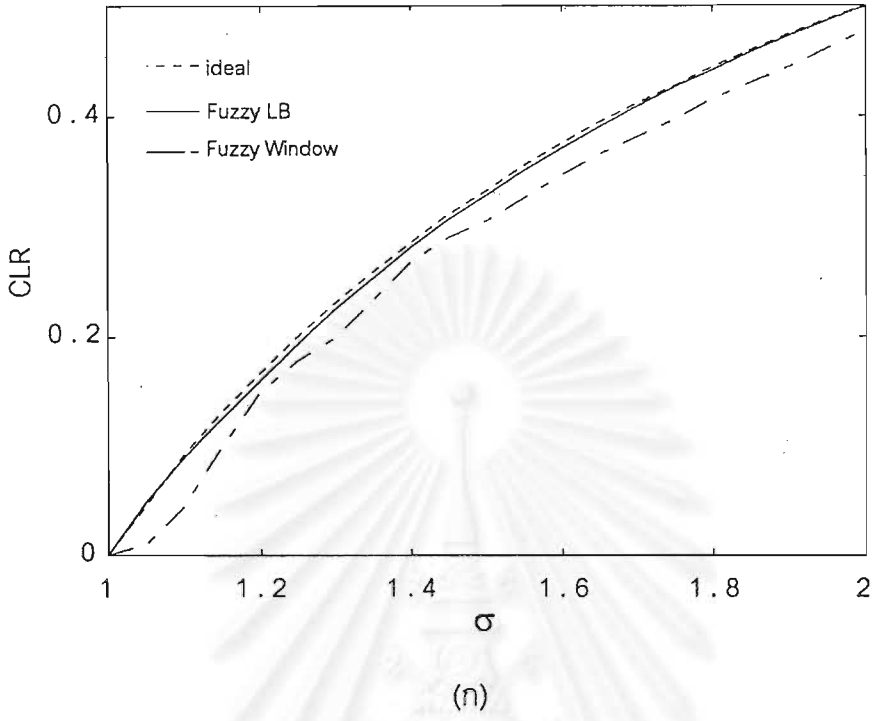
4.2.2.1 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก

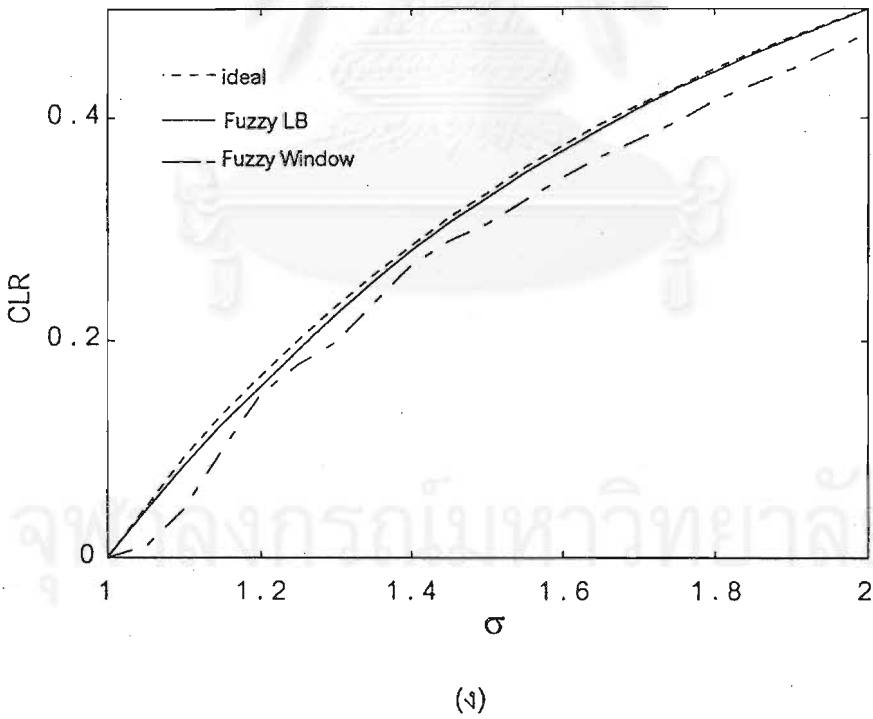
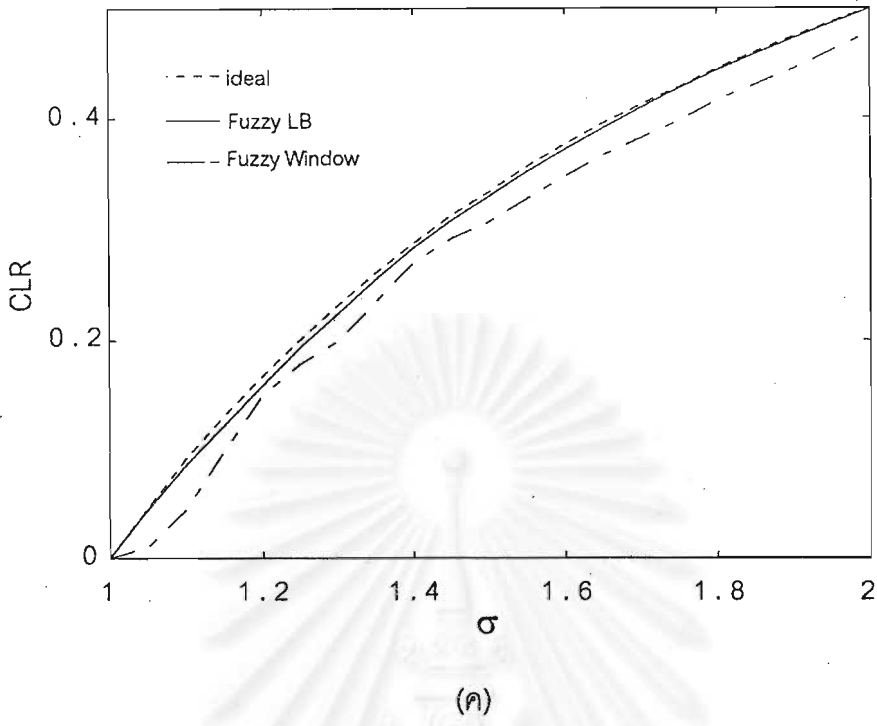
การทดสอบในที่นี้ทำโดยการวัดค่าอัตราส่วนการสูญหายของเซลล์ที่ขนาดบ่อโทเค็น (P) ต่างๆ กำหนดให้ $P \in \{3000, 3600, 4200, 4800\}$ และทดสอบกับกราฟฟีกที่มี $\sigma \in \{1.00, 1.05, \dots, 2.00\}$ จำนวน 15 ตัวอย่างต่อ σ 1 ค่า แต่ละตัวอย่างมีจำนวน 10^9 เซลล์ เพื่อเปรียบเทียบกับความน่าจะเป็นการละเมิดอุดมคติ

ผลการทดสอบเปรียบเทียบความสามารถในการตรวจวัดโหลดเกินกับกลไกหน้าต่าง พืชซีลोजิกและเส้นโค้งความน่าจะเป็นการละเมิดอุดมคติเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิดแสดง ดังตารางที่ 4.5 โดยเขียนกราฟได้ดังรูปที่ 4.6(ก)-4.6(ง) และเมื่อแปรค่า $E(S)$ แสดงดังตารางที่ 4.6 โดยเขียนกราฟได้ดังรูปที่ 4.7(ก)-4.7(ง) จากนั้นคำนวณรากที่สองของผลรวมค่าผิดพลาดกำลัง สอง (Root of Sum Square Error : RSSE) ได้ดังตารางที่ 4.7

ตารางที่ 4.5 อัตราส่วนการสูญหายของเซลล์ซึ่งเกิดจากกลไกตรวจกราฟฟิกที่ σ ต่างๆ เมื่อแปรค่า $E(X)$

| σ | อัตราส่วนการสูญหายของเซลล์ (CLR) | | | | | |
|----------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | อุดมคติ | หน้าต่าง พืชซีลोजิก | ถึงรั้วพืชซีลोजิก | | | |
| | | | P=3,000 | P=3,600 | P=4,200 | P=4,800 |
| 1.00 | 0 | 1.14×10^{-8} | 7.47×10^{-7} | 4.48×10^{-7} | 4.52×10^{-7} | 3.62×10^{-7} |
| 1.05 | 0.047619 | 0.033753 | 0.049302 | 0.047034 | 0.047636 | 0.044460 |
| 1.10 | 0.090909 | 0.087029 | 0.087903 | 0.086122 | 0.085176 | 0.083660 |
| 1.15 | 0.130435 | 0.127244 | 0.124421 | 0.123470 | 0.123311 | 0.121244 |
| 1.20 | 0.166667 | 0.155668 | 0.159536 | 0.158720 | 0.158857 | 0.157186 |
| 1.25 | 0.200000 | 0.181736 | 0.192633 | 0.192217 | 0.192120 | 0.191118 |
| 1.30 | 0.230769 | 0.211353 | 0.223924 | 0.223672 | 0.223388 | 0.223132 |
| 1.35 | 0.259260 | 0.247115 | 0.253810 | 0.253736 | 0.253397 | 0.253372 |
| 1.40 | 0.285714 | 0.278909 | 0.281407 | 0.281346 | 0.281128 | 0.281108 |
| 1.45 | 0.310345 | 0.297832 | 0.306133 | 0.306040 | 0.306142 | 0.305887 |
| 1.50 | 0.333333 | 0.310451 | 0.328640 | 0.328589 | 0.328943 | 0.328509 |
| 1.55 | 0.354839 | 0.325479 | 0.350124 | 0.350022 | 0.350314 | 0.349877 |
| 1.60 | 0.375000 | 0.342940 | 0.370700 | 0.370690 | 0.370678 | 0.370697 |
| 1.65 | 0.393939 | 0.360739 | 0.390620 | 0.390601 | 0.390607 | 0.390607 |
| 1.70 | 0.411765 | 0.378184 | 0.410291 | 0.409162 | 0.409291 | 0.409288 |
| 1.75 | 0.428571 | 0.394947 | 0.426322 | 0.426315 | 0.426327 | 0.426295 |
| 1.80 | 0.444444 | 0.411373 | 0.442081 | 0.442105 | 0.442086 | 0.442066 |
| 1.85 | 0.459459 | 0.427920 | 0.457331 | 0.457322 | 0.457328 | 0.457309 |
| 1.90 | 0.473684 | 0.444393 | 0.471905 | 0.471903 | 0.471903 | 0.471903 |
| 1.95 | 0.487179 | 0.460713 | 0.485622 | 0.485619 | 0.485619 | 0.485617 |
| 2.00 | 0.500000 | 0.476444 | 0.498259 | 0.498255 | 0.498255 | 0.498238 |



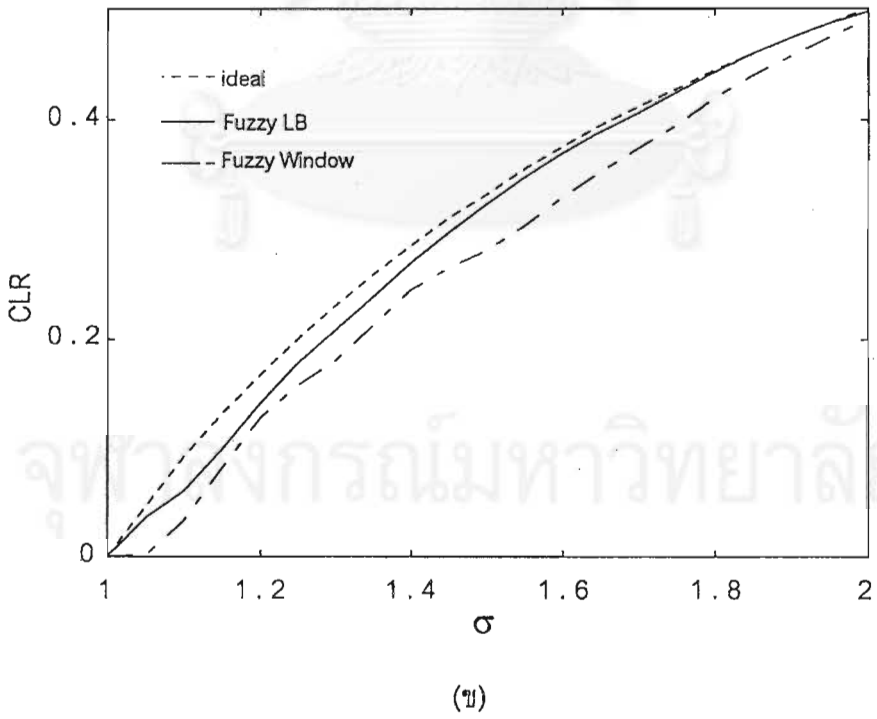
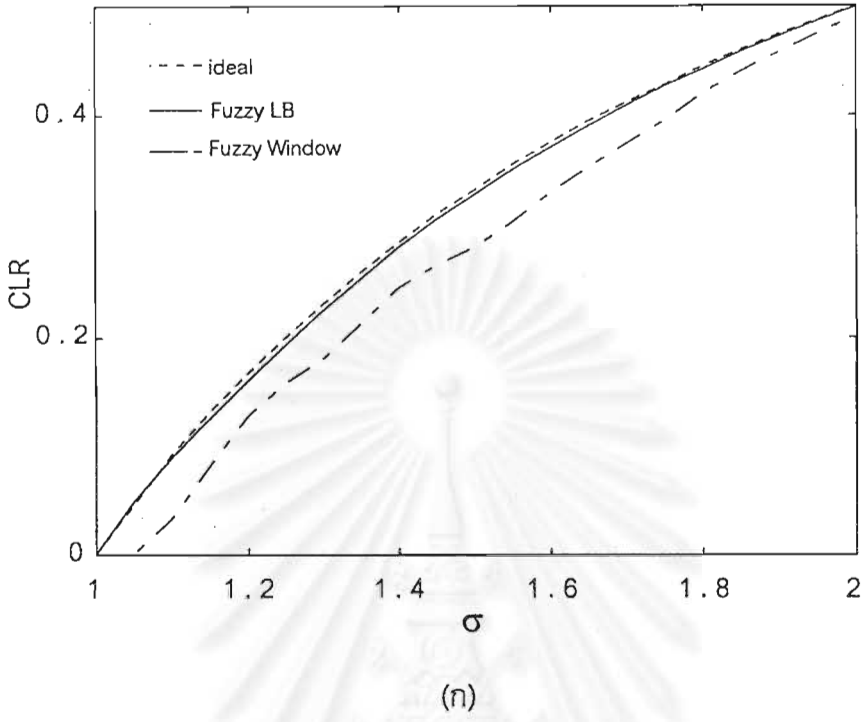


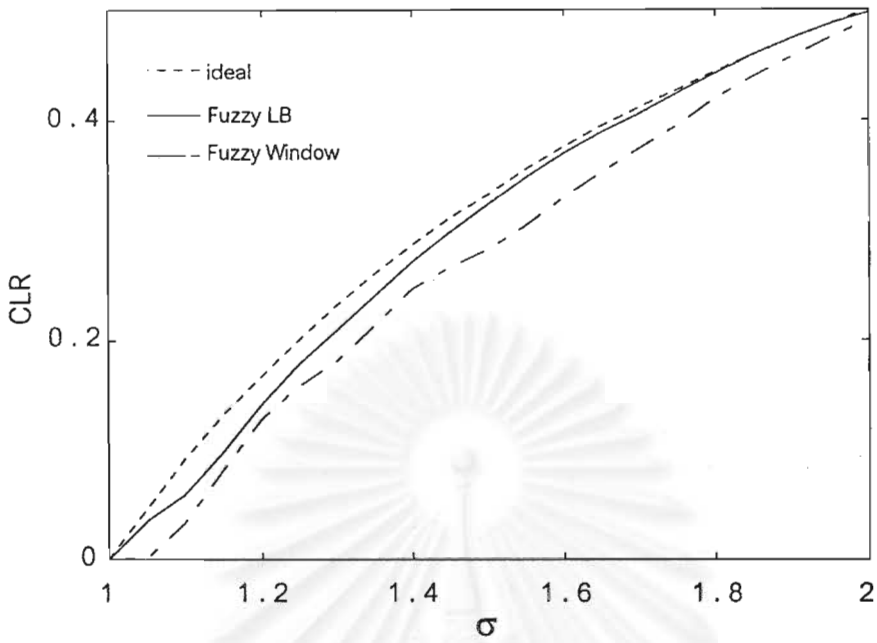
รูปที่ 4.6 ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจกราฟฟิกเมื่อแปรค่า $E(X)$

(ก) $P = 3,000$ (ข) $P = 3,600$ (ค) $P = 4,200$ (ง) $P = 4,800$

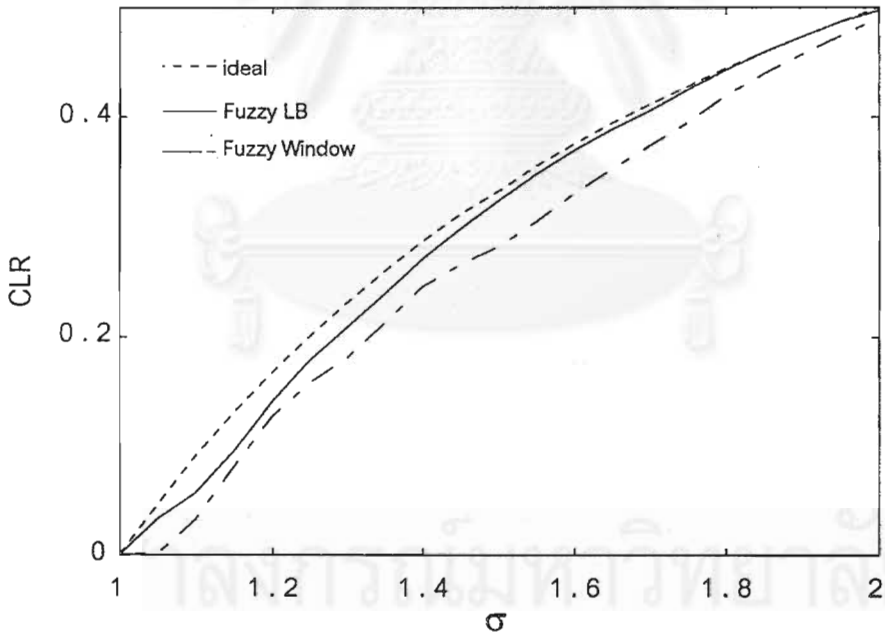
ตารางที่ 4.6 อัตราส่วนการสูญหายของเซลล์ซึ่งเกิดจากกลไกตรวจหาฟลักที่ σ
 ต่างๆ เมื่อแปรค่า E(S)

| σ | อัตราส่วนการสูญหายของเซลล์ (CLR) | | | | | |
|----------|----------------------------------|------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| | อุดมคติ | หน้าต่าง พีชชีลลจิก | ถึงรั้วพีชชีลลจิก | | | |
| | | | P=3,000 | P=3,600 | P=4,200 | P=4,800 |
| 1.00 | 0 | 1.14×10^{-8} | 7.47×10^{-7} | 4.48×10^{-7} | 4.52×10^{-7} | 3.62×10^{-7} |
| 1.05 | 0.047619 | 0.009758 | 0.038269 | 0.036460 | 0.034946 | 0.033321 |
| 1.10 | 0.090909 | 0.04346 | 0.061624 | 0.059876 | 0.058322 | 0.057209 |
| 1.15 | 0.130435 | 0.0987 | 0.098812 | 0.097724 | 0.096546 | 0.095774 |
| 1.20 | 0.166667 | 0.150212 | 0.141856 | 0.141331 | 0.140714 | 0.140102 |
| 1.25 | 0.200000 | 0.1772 | 0.177710 | 0.177523 | 0.177302 | 0.176867 |
| 1.30 | 0.230769 | 0.198006 | 0.208636 | 0.208503 | 0.208544 | 0.208413 |
| 1.35 | 0.259260 | 0.2327 | 0.239644 | 0.239547 | 0.239635 | 0.239608 |
| 1.40 | 0.285714 | 0.268363 | 0.270224 | 0.269976 | 0.269955 | 0.269621 |
| 1.45 | 0.310345 | 0.2898 | 0.298350 | 0.298333 | 0.298312 | 0.298298 |
| 1.50 | 0.333333 | 0.305184 | 0.324316 | 0.324299 | 0.324303 | 0.324278 |
| 1.55 | 0.354839 | 0.325 | 0.348237 | 0.348221 | 0.348200 | 0.348186 |
| 1.60 | 0.375000 | 0.346276 | 0.369713 | 0.369698 | 0.369698 | 0.369678 |
| 1.65 | 0.393939 | 0.3644 | 0.388921 | 0.388847 | 0.388815 | 0.388795 |
| 1.70 | 0.411765 | 0.380828 | 0.406852 | 0.406840 | 0.406756 | 0.406570 |
| 1.75 | 0.428571 | 0.3979 | 0.424840 | 0.424826 | 0.424812 | 0.424788 |
| 1.80 | 0.444444 | 0.414765 | 0.442336 | 0.442329 | 0.442316 | 0.442316 |
| 1.85 | 0.459459 | 0.4301 | 0.458736 | 0.458717 | 0.458725 | 0.458668 |
| 1.90 | 0.473684 | 0.444717 | 0.473534 | 0.473524 | 0.473501 | 0.472455 |
| 1.95 | 0.487179 | 0.46 | 0.486455 | 0.486362 | 0.486379 | 0.486223 |
| 2.00 | 0.500000 | 0.477141 | 0.497097 | 0.496850 | 0.496722 | 0.495556 |





(ค)



(ง)

รูปที่ 4.7 ความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจกราฟฟิกเมื่อแปรค่า $E(S)$

(ก) $P = 3,000$ (ข) $P = 3,600$ (ค) $P = 4,200$ (ง) $P = 4,800$

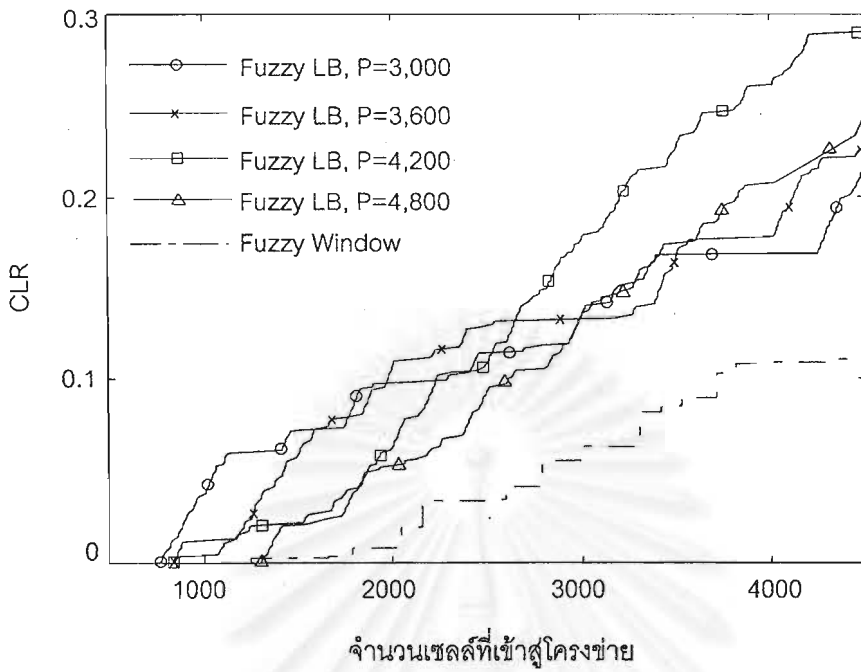
ตารางที่ 4.7 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของกลไกตรวจ
ทราฟฟิก เมื่อทดสอบกับแบบจำลองแบบที่ 2

| กลไก | หน้าต่าง พีชชีลอจิก | ถังรั่วพีชชีลอจิก | | | |
|--------------------|------------------------|-------------------|----------|----------|----------|
| | | P=3,000 | P=3,600 | P=4,200 | P=4,800 |
| RSSE E(S) คงที่ | 0.130790 | 0.019124 | 0.020493 | 0.020825 | 0.023615 |
| E(X) คงที่ | 0.181735 | 0.067424 | 0.069386 | 0.071187 | 0.072953 |

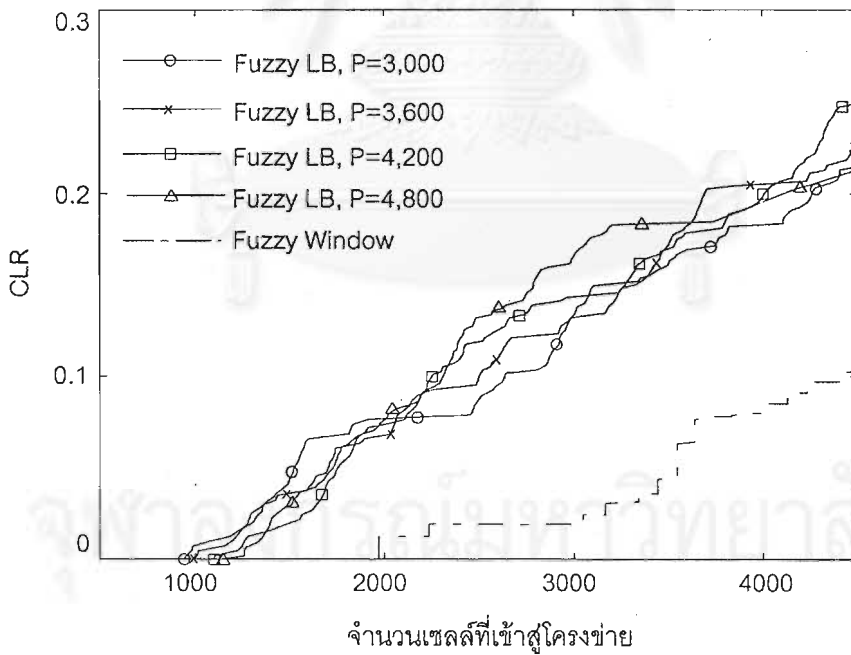
จากตารางที่ 4.7 RSSE มีค่าต่ำสุดเมื่อ $P = 3,000$ ในขณะที่ CLR ที่โหลดปกติหรือ false alarm probability มีค่าต่ำสุดเมื่อ $P=4,800$ กลไกถังรั่วพีชชีลอจิกสำหรับปอโตเค็นทุกขนาดยังคงมี RSSE ต่ำกว่าหน้าต่างพีชชีลอจิก กล่าวคือ ต่ำกว่าหน้าต่างพีชชีลอจิกโดยเฉลี่ย 2-3 เท่า แต่เมื่อเปรียบเทียบกับผลการทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 พบว่า RSSE มีค่าเพิ่มขึ้น โดยจากรูปที่ 4.6(ก)-4.7(ง) สามารถสังเกตความแตกต่างของเส้นโค้งความสามารถในการตรวจวัดโหลดเกินของกลไกที่เสนอกับเส้นโค้งอุดมคติชัดเจน โดยเฉพาะอย่างยิ่งช่วงขนาดโหลดเกินต่างๆ นอกจากนี้ false alarm probability ยังมีค่าเพิ่มขึ้นด้วย แม้ว่า CLR ดังกล่าวยังคงอยู่ในระดับที่ยอมรับได้สำหรับทราฟฟิกประเภท packet voice คือ 10^{-3} ซึ่งจะได้วิเคราะห์ต่อไป

4.2.2.2 ผลการทดสอบเปรียบเทียบการตอบสนอง

การทดสอบทำโดยการวัดอัตราส่วนการสูญหายของเซลล์เทียบกับจำนวนเซลล์ที่ตัวตรวจยอมให้ผ่านเข้าไปยังโครงข่ายเมื่อ $\sigma = 1.50$ เปรียบเทียบกับกลไกหน้าต่างพีชชีลอจิก โดย $P \in \{3000, 3600, 4200, 4800\}$ ผลการทดสอบเมื่อแปรทราฟฟิกโหลดด้วยการแปรค่า $E(X)$ แสดงดังรูปที่ 4.8 และเมื่อแปรค่า $E(S)$ แสดงดังรูปที่ 4.9 จำนวนเซลล์ที่กลไกตรวจทราฟฟิกยอมให้ผ่านเข้าสู่โครงข่ายก่อนละทิ้งเซลล์แรกแสดงไว้ในตารางที่ 4.8



รูปที่ 4.8 การตอบสนองของกลไกตรรกะฟัฟกเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



รูปที่ 4.9 การตอบสนองของกลไกตรรกะฟัฟกเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

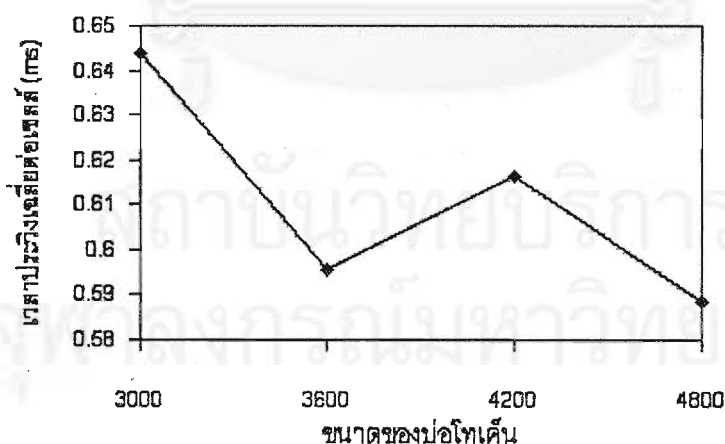
ตารางที่ 4.8 เปรียบเทียบการตอบสนองเริ่มต้นของกลไกตรวจกราฟฟิก เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2

| จำนวนเซลล์ที่เข้าสู่โครงข่ายก่อนการละทิ้งเซลล์แรก | หน้าต่างพีชชีลอจิก | ตั้งรั้วพีชชีลอจิก | | | |
|---|--------------------|--------------------|---------|---------|---------|
| | | P=3,000 | P=3,600 | P=4,200 | P=4,800 |
| เมื่อแปรค่า E(X) | 1,244 | 768 | 845 | 838 | 1,301 |
| เมื่อแปรค่า E(S) | 1,967 | 943 | 962 | 1,097 | 1,134 |

จากตารางที่ 4.8 ตั้งรั้วพีชชีลอจิกสามารถตรวจพบเซลล์แรกได้เร็วกว่าหน้าต่างพีชชีลอจิก ยกเว้นเมื่อ $P = 4,800$ และกราฟฟิกละเมิดข้อตกลงด้วยการเพิ่ม E(X) โดยยังคงเห็นได้ว่าการเพิ่มขนาดบ่อโทเค็นทำให้การตรวจพบเซลล์แรกมีแนวโน้มช้าลง ซึ่งได้วิเคราะห์เหตุผลไปแล้วในหัวข้อที่ 4.1.1.2

4.2.2.3 เวลาประวิงที่เกิดจากกลไกตรวจกราฟฟิก

ผลการวัดเวลาประวิงเฉลี่ยต่อ 1 เซลล์ของกราฟฟิกแสดงได้ดังรูปที่ 4.10



รูปที่ 4.10 เวลาประวิงที่เกิดจากกลไกตรวจกราฟฟิกที่ไหลลดปกติ

จากรูปที่ 4.10 โดยแนวโน้มพบว่าการเพิ่มขนาดบ่อโทเค็นทำให้เวลาประวิงลดลง แม้ว่าเวลาประวิงเมื่อกลไกมี $P = 4,200$ มากกว่ากลไกที่มี $P = 3,600$ ซึ่งจะได้วิเคราะห์ต่อไป

4.3 วิเคราะห์ผลของขนาดของบ่อโทเค็นที่มีต่อประสิทธิภาพของกลไกตรวจกราฟฟิกที่เสนอ

4.3.1 ความสามารถในการเลือก

จากผลการทดสอบ พบว่า CLR ที่ไหลดทุกๆ ขนาดมีแนวโน้มลดลงเมื่อขนาดของบ่อโทเค็นเพิ่มขึ้น โดยในกรณีที่ทดสอบพบว่าบ่อโทเค็นขนาดเล็กที่สุดจะมีพฤติกรรมการตรวจวัดไหลดเกินใกล้เคียงกับพฤติกรรมอุดมคติที่สุด ขณะที่เส้นโค้งการตรวจวัดไหลดเกินของกลไกที่มีบ่อโทเค็นใหญ่ขึ้นมาจะมีเส้นโค้งที่ห่างออกจากเส้นโค้งอุดมคติขึ้นเรื่อยๆ แต่ไม่อาจสรุปได้ว่าควรกำหนดให้บ่อโทเค็นมีขนาดเล็กที่สุด สังเกตได้จากรูปที่ 4.6(ก) และตารางที่ 4.5 เมื่อตัวตรวจมีขนาดบ่อโทเค็นเท่ากับ 3,000 ทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 พบว่า CLR ที่ไหลดเกินขนาดต่ำๆ มีค่ามากกว่าความน่าจะเป็นการละเมิดอุดมคติ ย่อมหมายถึงกลไกละทิ้งเซลล์มากเกินไปหรือละเมิด QoS ของผู้ใช้บริการนั่นเอง การเลือกขนาดของบ่อโทเค็นเมื่อพิจารณาจากความสามารถในการตรวจวัดไหลดเกินจึงต้องไม่สูงเกินไปหรือต่ำเกินไป แต่ต้องมีการทดสอบก่อนเพื่อให้ได้พฤติกรรมใกล้เคียงเส้นโค้งอุดมคติที่สุดโดยไม่ละเมิด QoS ของผู้ใช้

เมื่อพิจารณาในแง่ของผู้ใช้บริการที่ไม่ละเมิดข้อตกลงหรือส่งด้วยไหลดขนาดปกติ พบว่าบ่อโทเค็นที่ใหญ่ขึ้นจะทำให้ false alarm probability ลดลง เนื่องจากตัวควบคุมพีชชีลจิกถูกออกแบบให้เพิ่มเครดิตด้วยการเพิ่มอัตราการกำเนิดโทเค็นให้กับแหล่งกำเนิดที่ส่งด้วยอัตราเซลล์ต่ำกว่าหรือเท่ากับอัตราเซลล์เฉลี่ยที่ตกลงไว้ ดังนั้นกลไกที่มีบ่อโทเค็นขนาดใหญ่จึงสามารถเก็บสะสมโทเค็นไว้ใช้ในกรณีที่เกิดภาวะเบิรสต์ได้มากกว่า ดังนั้นเราจึงไม่สามารถเลือกขนาดของบ่อโทเค็นที่ทำให้ได้ false alarm probability ต่ำสุดและความสามารถในการตรวจวัดไหลดเกินสูงสุดพร้อมๆ กันได้ เนื่องจากธรรมชาติที่ขัดแย้งกันของกลไก

4.3.2 การตอบสนอง

ขนาดของบ่อโทเค็นมีผลโดยตรงต่อการตอบสนอง จากแนวโน้มในตารางที่ 4.4 และ 4.8 การเพิ่มขนาดของบ่อโทเค็นทำให้การตอบสนองช้าลง ดังที่กล่าวไปแล้วในหัวข้อที่ 4.3.1 คือกลไกมีการเพิ่มเครดิตให้กับแหล่งกำเนิดที่เคารพข้อตกลงเรื่อยๆ แต่หากมีการละเมิดเกิดขึ้นเป็นระยะเวลาหนึ่ง กลไกจะลดอัตราการกำเนิดโทเค็นลง โดยความรุนแรงในการลดขึ้นกับระดับการละเมิด กระนั้นก็ตาม การละทิ้งไม่สามารถเกิดขึ้นได้ทันทีทันใด เนื่องจาก 2 ปัจจัยคือ การมีอยู่ของคิวบัฟเฟอร์ด้านเข้าและโทเค็นสะสมในบ่อ ในกรณีนี้คิวบัฟเฟอร์มีขนาดน้อยมากเมื่อเทียบกับจำนวนเซลล์ที่จะมาถึงในแต่ละเบิรสต์เมื่อเกิดการละเมิด ดังนั้นขนาดของบ่อโทเค็นที่ใหญ่และการสะสม

โทเค็นที่เกิดขึ้นในช่วงเวลาไหลตปกติ จึงเปรียบเหมือนโมเมนตัมที่ถ่วงให้การตอบสนองของกลไกช้าลง แนวโน้มของพฤติกรรมนี้ไม่ต่างจากถ่วงร้วแบบโทเค็นปกติแต่อย่างใด เพียงแต่การเปลี่ยนแปลงอัตราการทำเนิดโทเค็นของตัวควบคุมพีชชีลอจิกจะช่วยให้การตอบสนองเร็วขึ้นได้ อย่างไรก็ตามก็ดี ในการทดสอบกับแบบจำลองแบบที่ 2 โดยแปรขนาดไหลตด้วยวิธีเพิ่มค่า $E(X)$ พบว่ากลไกที่มี $P=4,200$ มีการตอบสนองเร็วกว่า $P=3,600$ ซึ่งสอดคล้องกับ CLR ที่ $\sigma=1.50$ ในตารางที่ 4.5 ทั้งนี้เพราะการออกแบบให้จำนวนโทเค็นในบ่อโทเค็นเป็นอินพุตค่าหนึ่งของตัวควบคุมพีชชีลอจิก โดยที่ crisp value จะถูกเปลี่ยนให้เป็นค่าพีชชีด้วยฟังก์ชันสมาชิกที่คิดค่าสัมพันธ์กับขนาดของบ่อโทเค็น ขนาดของบ่อโทเค็นบางค่าอาจทำให้เกิดอัตราการทำเนิดโทเค็นเปลี่ยนแปลงในลักษณะที่ทำให้การสะสมโทเค็นไม่สัมพันธ์กับขนาดของบ่อและมีพฤติกรรมกระโดดจากแนวโน้มที่ควรจะเป็น

4.3.3 เวลาประวิง

ขนาดของบ่อโทเค็นมีความสัมพันธ์ผกผันกับเวลาประวิง กล่าวคือการเพิ่มขนาดบ่อโทเค็นจะทำให้เวลาประวิงลดลง แต่ยังพบข้อยกเว้นที่จุดเดิมคือกลไกที่มี $P=4,200$ เมื่อทดสอบกับแบบจำลองที่ 2 ที่ใช้วิธีแปรค่า $E(X)$ เป็นที่สังเกตได้ว่าพฤติกรรมดังกล่าวสอดคล้องกับ CLR ณ ไหลตปกติ ($\sigma=1.0$) จากตารางที่ 4.5, 4.6 คือเมื่อกลไกมี $P=4,200$ กลับมี false alarm probability สูงกว่ากลไกที่มี $P=3,600$ และผลการตอบสนองที่กล่าวไปแล้วในหัวข้อที่ 4.3.2 เพื่อยืนยันสมมติฐานในกรณีดังกล่าวจึงทดสอบหาจำนวนโทเค็นเฉลี่ยในบ่อโทเค็น เปรียบเทียบระหว่างกลไกที่มีขนาดบ่อต่างๆ ในสภาวะไหลตปกติ ได้ผลดังตารางที่ 4.9

ตารางที่ 4.9 เปรียบเทียบจำนวนโทเค็นเฉลี่ยในบ่อโทเค็นระหว่างกลไกตรวจกราฟฟิกที่มีขนาดของบ่อโทเค็นต่างๆ

| | | | | |
|------------------------------|-------|-------|-------|-------|
| ขนาดของบ่อโทเค็น (โทเค็น) | 3,000 | 3,600 | 4,200 | 4,800 |
| จำนวนโทเค็นสะสมเฉลี่ย | 1,882 | 2,825 | 2,793 | 3,370 |

จากตารางที่ 4.9 เห็นได้ว่าจำนวนโทเค็นเฉลี่ยเมื่อ $P=4,200$ น้อยกว่า $P=3,600$ เล็กน้อย จึงเป็นเหตุผลที่ทำให้กลไกที่ขนาดบ่อโทเค็นดังกล่าวมี false alarm probability สูงกว่า, การตอบสนองเร็วกว่าและมีเวลาประวิงสูงกว่ากลไกที่มี $P=3,600$

4.4 วิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างกลไกตรวจกราฟฟิกต่างๆ

4.4.1 ความสามารถในการเลือก

ความสามารถในการเลือกประกอบด้วย 2 ส่วนคือ false alarm probability ต่ำ หรือเท่ากับศูนย์ในกรณีอุดมคติ และความสามารถในการตรวจวัดโหลดเกินใกล้เคียงกับความน่าจะเป็นการละเมิดอุดมคติ

พิจารณาศามารถในการตรวจวัดโหลดเกิน จากผลการทดสอบในตารางที่ 4.3 และ 4.7 เห็นได้ว่ากลไกที่เสนอมีพฤติกรรมใกล้เคียงอุดมคติมากที่สุด โดยเฉพาะอย่างยิ่งเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 ดังจะเห็นได้จากรูปที่ 4.1(ก)-4.1(ง) และรูปที่ 4.2(ก)-4.2(ง) ในขณะที่เมื่อทดสอบกับแบบจำลองแบบที่ 2 แม้ว่าจะยังคงได้ผลที่ดีกว่ากลไกหน้าต่างพีชชีลोजิก แต่ความแตกต่างกับพฤติกรรมอุดมคติเพิ่มมากขึ้น

เมื่อพิจารณา false alarm probability พบว่ากลไกถึงรั้วพีชชีลोजิกมี CLR ที่สภาวะโหลดปกติในระดับชั้น 10^7 ดีกว่าถึงรั้วในระดับ 10^2 เท่า แต่ด้อยกว่าหน้าต่างพีชชีลोजิก 10 เท่า นอกจากนี้ เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 false alarm probability เพิ่มขึ้นด้วย ในขณะที่ false alarm probability ของหน้าต่างพีชชีลोजิกในแหล่งกำเนิดทั้ง 2 แบบ ไม่แตกต่างกันเท่าใดนัก ดังนั้นจึงแบ่งวิเคราะห์เป็น 2 ประเด็นคือ

1. เหตุใดกลไกที่เสนอก็มี false alarm probability สูงกว่าหน้าต่างพีชชีลोजิก
2. เหตุใดกลไกที่เสนอก็มีความสามารถในการเลือกลดลงเมื่อเปลี่ยนแบบจำลอง

แหล่งกำเนิดที่ใช้ทดสอบ

ในประเด็นที่ 1 เนื่องจากการออกแบบตัวควบคุมพีชชีลोजิก มุ่งเน้นการทำให้กลไกมีเส้นโค้งการตรวจวัดโหลดเกินใกล้เคียงกับอุดมคติที่สุด ดังจะเห็นได้ว่าเมื่อโหลดเพิ่มขึ้นจากโหลดปกติเพียงเล็กน้อย CLR ที่เกิดจากตัวตรวจเพิ่มขึ้นอย่างรวดเร็วจนใกล้เคียงความน่าจะเป็นการละเมิดอุดมคติมาก ดังนั้นอัตราการกำเนิดโทเค็นเฉลี่ยจึงสูงกว่าอัตราเซลล์เฉลี่ยที่ตกลงในเฟสเซตอัปการต่อเพียงเล็กน้อย และใช้บ่อโทเค็นขนาดใหญ่และการควบคุมอัตราการกำเนิดโทเค็นในลักษณะเพิ่ม-ลดเครดิต เพื่อไม่ให้ false alarm probability สูงเกินไป ในขณะที่หน้าต่างพีชชีลोजิกเลือกที่จะให้มี low false alarm probability ต่ำที่สุดเท่าที่จะเป็นไปได้ซึ่ง false alarm probability นี้เป็น

ปัญหาหลักปัญหาหนึ่งของกลไกหน้าตาต่างกระโดดแบบดั้งเดิม โดย [6] ใช้วิธีเพิ่มขีดจำกัดหน้าตาต่างขึ้นเรื่อยๆ ในสภาวะไหลดปกติ โดยมีขีดจำกัดการเพิ่มสูงสุดถึง 9 เท่าของจำนวนเซลล์เฉลี่ยที่คาดว่าจะมาถึงในสภาวะไหลดปกติ เปรียบเทียบอัตราการกำเนิดโทเค็นเฉลี่ยของกลไกที่เสนอกับอัตราของการอนุญาตให้เซลล์ผ่านเข้าไปในโครงข่ายคิดจากขีดจำกัดหน้าตาต่างเฉลี่ยของกลไกหน้าตาต่างพีชชีลอจิก ที่ไหลดปกติ แสดงดังตารางที่ 4.10

ตารางที่ 4.10 เปรียบเทียบอัตราการอนุญาตเซลล์ให้ผ่านเข้าสู่โครงข่าย ณ ไหลดปกติ

| กลไก | แบบจำลองแหล่งกำเนิดที่ 1 | แบบจำลองแหล่งกำเนิดที่ 2 |
|----------------------|--------------------------|--------------------------|
| ถังรั่วพีชชีลอจิก | 1.15 MCR | 1.57 MCR |
| หน้าตาต่างพีชชีลอจิก | 1.78 MCR | 7.32 MCR |

ความแตกต่างของอัตราการอนุญาตที่แสดงให้เห็นในตารางที่ 4.10 จึงเป็นเหตุผลที่ทำให้ false alarm probability ของหน้าตาต่างพีชชีลอจิกจึงต่ำกว่าถังรั่วพีชชีลอจิก แต่การเพิ่มขีดจำกัดหน้าตาต่างก็มีข้อเสีย คือเมื่อเกิดไหลดเกินเล็กน้อย กลไกจะไม่สามารถลดขีดจำกัดลงได้ทัน ผลคือความสามารถในการตรวจวัดไหลดเกินด้อยกว่ากลไกถังรั่วพีชชีลอจิกมาก

ในประเด็นที่ 2 เนื่องจากแบบจำลองทั้งสองมี burstiness เท่ากับ 2.85 โดยแตกต่างกันที่อัตราเซลล์เฉลี่ยและอัตราเซลล์สูงสุดดังตารางที่ 4.3 และเนื่องจากการกำหนดอัตราการกำเนิดโทเค็นสัมพันธ์กับอัตราเซลล์เฉลี่ยและอัตราเซลล์สูงสุดของแหล่งกำเนิด ดังนั้นประสิทธิภาพของกลไกเมื่อทดสอบกับแหล่งกำเนิดทั้งสองแบบจึงแตกต่างกัน จากตารางที่ 4.3 และ 4.7 พบว่าความสามารถในการตรวจวัดไหลดเกินของถังรั่วพีชชีลอจิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 ลดลงเมื่อเปรียบเทียบการทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1 และจากตารางที่ 4.1 และ 4.2 เปรียบเทียบกับตารางที่ 4.5 และ 4.6 พบว่า false alarm probability เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 ก็เพิ่มขึ้นด้วย ประสิทธิภาพของกลไกที่ต่างกันนั้นเกิดจากปัจจัย 2 ปัจจัยได้แก่

1. อัตราการกำเนิดโทเค็นสัมพันธ์ที่ต่างกันดังแสดงในตารางที่ 4.10
2. ผลของขนาดคิวบัฟเฟอร์ด้านเข้าสัมพันธ์กับจำนวนเซลล์เฉลี่ยในแต่ละเบิรสต์ที่แตกต่างกัน โดย $B_1 = 2E_1(X)$ และ $B_2 = E_2(X)$ เหตุผลที่กำหนดเช่นนี้เพราะการเพิ่มขนาดของบัฟเฟอร์จะทำให้เวลาประวิงเพิ่มขึ้น เมื่อทดสอบโดยเพิ่มขนาดคิวบัฟเฟอร์ $B_2 = 2E_2(X) = 60$ เมื่อให้ $P_2 = 4,800$ พบว่าเวลาประวิงเฉลี่ยต่อ 1 เซลล์เพิ่มขึ้นเป็น 2.23 ms ซึ่งมากเกินไปจะยอมรับได้

ปัจจัยทั้งสองอาจส่งผลต่อประสิทธิภาพของกลไกในแต่ละขณะไม่เท่ากัน โดยในสภาวะไหลตกติผลของขนาดบัพเฟอร์อาจมากกว่าจึงทำให้ false alarm probability เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2 สูงกว่าแบบที่ 1 แต่ในสภาวะไหลเกินมากๆ บัพเฟอร์จะอยู่ในสภาวะล้นเกือบตลอดเวลา ผลของอัตราการกำเนิดโทเค็นจึงส่งผลต่อประสิทธิภาพมากกว่า และเป็นผลให้ถึงรั้วฟิชชีลจิกเมื่อทดสอบแบบจำลองแหล่งกำเนิดแบบที่ 2 ที่มีอัตราการกำเนิดโทเค็นสัมพัทธ์สูงกว่า มีความสามารถในการตรวจวัดไหลเกินต่ำกว่า

4.4.2 การตอบสนอง

กลไกถึงรั้วฟิชชีลจิกมีการตอบสนองเร็วกว่ากลไกหน้าต่างฟิชชีลจิกในทุกๆ ลักษณะของทราฟฟิกยกเว้นเพียงกลไกที่มีขนาดของบ่อโทเค็นสูงๆ เช่น $P = 4,800$ ในกรณีแหล่งกำเนิดแบบที่ 2 ถึงรั้วฟิชชีลจิกมีการตอบสนองรวดเร็วกว่าหน้าต่างฟิชชีลจิก เนื่องจาก

1. ผู้วิจัย [6] กำหนดกฎพื้นฐานฟิชชีให้มีการเพิ่มขีดจำกัดหน้าต่างอย่างมากในสภาวะไหลตกติ และยังออกแบบให้ใช้ฟิชชีลจิกควบคุมจำนวนการเพิ่ม/ลดของขีดจำกัด ไม่ใช่การควบคุมขีดจำกัดโดยตรง ดังนั้นจึงต้องใช้เวลาในการลดขีดจำกัดหน้าต่างลงเพื่อให้เกิดการลงโทษ นอกจากนี้ทุกกลไกล้วนมีข้อจำกัดในการแบ่งแยกสภาวะไหลตกติและไหลเกินเพียงเล็กน้อยออกจากกันอย่างเด็ดขาด ดังนั้นการเลือกให้อัตราการอนุญาตเซลล์เข้าสู่โครงข่ายในสภาวะไหลตกติสูงกว่าอัตราเซลล์เฉลี่ยที่ตกลงไว้มาก หากเป็นทราฟฟิกที่มีไหลเกินน้อยๆ จึงไม่อาจบังคับให้กลไกตัดสินใจเป็นสภาวะปกติหรือไม่อย่างทันทีทันใด เป็นผลให้กลไกไม่สามารถลดอัตราการอนุญาตลงได้ทันท่วงที ในขณะที่กลไกถึงรั้วฟิชชีลจิกเป็นการควบคุมอัตราการกำเนิดโทเค็นโดยตรงจึงใช้เวลาน้อยกว่า นอกจากนี้ยังออกแบบฟังก์ชันสมาชิกและกฎฟิชชี โดยพยายามให้อัตราการอนุญาตให้เซลล์ผ่านเข้าไปสู่โครงข่ายของกลไกถึงรั้วฟิชชีลจิกมีค่าแกว่งอยู่รอบๆ อัตราเซลล์เฉลี่ยที่ตกลงไว้ ทำให้สามารถตรวจวัดและมีปฏิกิริยาต่อไหลเกินขนาดน้อยๆ ได้เร็วกว่า

2. คาบเวลาชักตัวอย่างของกลไกถึงรั้วต่ำกว่ากลไกหน้าต่าง คือเท่ากับ 1 on-off cycle โดยกลไกหน้าต่างมีคาบเวลาชักตัวอย่างเท่ากับ 6 on-off cycle สำหรับแบบจำลองแบบที่ 1 และเท่ากับ 3 on-off cycle สำหรับแบบจำลองแบบที่ 2 เหตุผลที่สามารถออกแบบให้คาบเวลาชักตัวอย่างของกลไกต่ำได้คือโครงสร้างพื้นฐานเป็นถึงรั้ว และพารามิเตอร์ที่ใช้แทนอินพุตของตัวควบคุมเป็นสภาวะสะสม ดังนั้นแม้จะมีคาบเวลาชักตัวอย่างสั้น การใช้พารามิเตอร์ของกลไกหลายๆ ตัวสามารถบ่งบอกลักษณะทราฟฟิกทั้งระยะสั้นและระยะยาวได้ เช่น จากกฎพื้นฐานฟิชชีข้อที่ 6 $N_c = Z$, $\Delta N_c = NB$, $N_r = L$, $R_r = NS$ หมายความว่า ในคาบเวลาก่อนหน้า กลไกมีอัตราการกำเนิดโทเค็นต่ำกว่าปกติ แต่กระนั้นจำนวนเซลล์ในบัพเฟอร์ก็ลดลงมาก และยังเหลือโทเค็นในบ่ออีกด้วย

จึงบอกได้ว่าพฤติกรรมระยะสั้นของแหล่งกำเนิดในคาบเวลาก่อนหน้าคือ มีอัตราเซลล์ต่ำกว่าอัตราที่ตกลงไว้มาก แต่พฤติกรรมระยะยาวที่ผ่านมาก็คือมีเซลล์คงค้างและเพิ่งถูกลบโทษ ดังนั้นจึงเพิ่มเครดิตไว้ในระดับหนึ่งแต่ไม่ถึงขั้นสูงสุด $R_{i+1} = PM$ ในขณะที่กลไกหน้าต่าง แม้ว่าจะมีจำนวนเซลล์เฉลี่ยที่มาถึงในแต่ละหน้าต่างเริ่มตั้งแต่เริ่มการต่อเป็นตัวแทนลักษณะระยะยาวของทราฟฟิก แต่อินพุตอื่นๆ ที่เป็นตัวแทนของลักษณะทราฟฟิกในคาบเวลาก่อนหน้า เช่นจำนวนเซลล์ที่มาถึง ไม่ได้เป็นสภาวะสะสม จึงต้องการคาบเวลาที่ยาวนาน เพื่อให้แน่ใจได้ว่าอินพุตดังกล่าวเป็นตัวแทนที่ใกล้เคียงความจริงที่สุด แม้ว่าจะมีการนำพีชชีลอจิกมาใช้ แต่เนื่องจากกลไกพื้นฐานเป็นหน้าต่างกระโดดทำให้ไม่อาจหลีกเลี่ยงปัญหาดังกล่าวได้ กลไกที่ใช้พีชชีลอจิกช่วยควบคุมที่มีโครงสร้างพื้นฐานเป็นหน้าต่างจึงมีคุณสมบัติการตอบสนองดีกว่ากลไกควบคุมด้วยพีชชีลอจิกที่มีโครงสร้างพื้นฐานเป็นถังรั่ว

4.4.3 เวลาประวิง

ข้อเสียที่สำคัญที่สุดของกลไกที่เสนอ คือมีเวลาประวิงที่เกิดจากการรอคอยของเซลล์ในบัฟเฟอร์ แม้ว่าจะพยายามลดขนาดบัฟเฟอร์และเพิ่มขนาดบ่อโทเค็น แต่เพื่อให้ได้คุณสมบัติอื่นๆ ตามที่ต้องการ ก็ไม่อาจหลีกเลี่ยงการประวิงเวลาดังกล่าว ในขณะที่กลไกถังรั่วและหน้าต่างพีชชีลอจิกไม่มีเวลาประวิง

ผลการทดสอบการใช้โครงข่ายประสาทเทียมในตรวจจับการฉ้อโกง

5.1 กล่าวนำ

เนื่องจากตัวควบคุมอัตราการทำผิดที่เกินมีอินพุตถึง 4 ตัว และมีกฎพื้นฐานพีชคณิตทั้งสิ้น 162 กฎ จึงทำให้ขั้นตอน inference มีความยุ่งยากและต้องใช้เวลาในการประมวลผลนาน ในขณะที่โครงข่ายเอทีเอ็มต้องการการควบคุมกราฟิกใกล้เคียงเวลาจริงมากที่สุด เพื่อป้องกันปัญหาเวลาประวิงในการควบคุม เราเสนอโครงข่ายประสาทเทียมชนิด multilayer perceptron ทำหน้าที่จับคู่อินพุตกับเอาต์พุตของ inference engine โครงข่ายประสาทเทียมจะถูกฝึกแบบชี้แนะ (Supervised Training) ด้วยชุดข้อมูลอินพุต-เอาต์พุตจนกระทั่งได้ค่าถ่วงน้ำหนักที่ค่อนข้างคงที่และทำให้ผลรวมค่าผิดพลาดกำลังสอง (SSE) ซึ่งคำนวณได้ตามสมการที่ (5.1) มีค่าไม่เกิน 10^{-5}

$$SSE = \sum_{p=1}^P \sum_{i=1}^N (t_i - o_i)^2 \tag{5.1}$$

โดยที่ t_i คือเอาต์พุตเป้าหมายของเอาต์พุตโนดที่ i , o_i คือเอาต์พุตที่ได้จากการฝึกของเอาต์พุตโนดที่ i , N คือจำนวนเอาต์พุตโนด และ P คือจำนวนชุดฝึก จากนั้นนำค่าถ่วงน้ำหนักที่ได้มาทำหน้าที่จับคู่ (map) อินพุต-เอาต์พุตแทนขั้นตอน inference ในตัวควบคุมพีชคณิตลอจิก และทดสอบเปรียบเทียบประสิทธิภาพของตัวตรวจพีชคณิตโครงข่ายประสาทเทียมกับตัวตรวจพีชคณิตลอจิกที่ใช้ inference engine โดยพิจารณาเป็น 2 กรณี คือ แบบจำลองแหล่งกำเนิดแบบที่ 1 และแบบจำลองแหล่งกำเนิดแบบที่ 2

5.2 โครงสร้างของโครงข่ายประสาทเทียมที่ใช้

โครงข่ายประสาทเทียมชนิด multilayer perceptron ที่ใช้มี 3 ชั้นประกอบด้วย ชั้นอินพุต ชั้นเอาต์พุต และชั้นฮิดเดน 1 ชั้น จำนวนอินพุตโนดในชั้นอินพุตเท่ากับ 22 โหนด และจำนวนเอาต์พุตโนดในชั้นเอาต์พุตเท่ากับ 7 โหนด ส่วนจำนวนฮิดเดนโนดในชั้นฮิดเดนขึ้นอยู่กับข้อมูลที่ใช้ฝึกซึ่งจะต้องทำการทดสอบก่อน จึงไม่สามารถระบุจำนวนแน่นอนได้ในขณะนี้

อินพุตโนด 22 โหนด ประกอบด้วย

- ค่าพีชคณิตของจำนวนเซลล์ในคิวบ์เฟอ์ด้านเข้า (N_c) 5 โหนด
- ค่าพีชคณิตของจำนวนโหนดในบ่อโหนด (N_T) 5 โหนด

- ค่าพีชชีของจำนวนการเปลี่ยนแปลงของเซลล์ในคิวบ์เฟอรัด้านเข้า (ΔN_c) 5 โหนด
- ค่าพีชชีของอัตราการกำเนิดโทเค้นในคาบเวลา ก่อน (R_i) 7 โหนด

เอาต์พุต โหนด 7 โหนด คือ ค่าพีชชีของอัตราการกำเนิดโทเค้นในคาบเวลาต่อไป (R_{i+1})

ระหว่างชั้นอินพุตกับชั้นฮิดเดนใช้ฟังก์ชันไวงานชนิดฟังก์ชันซิกมอยด์แบบสองขั้ว (Bipolar Sigmoid Function) ดังสมการที่ (5.2)

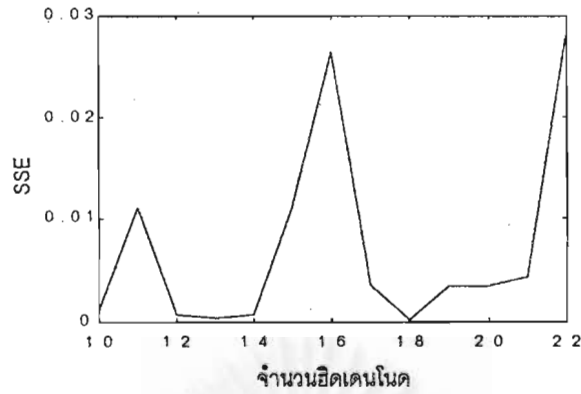
$$f_{ih}(x) = \frac{2}{1 + \exp(-x)} - 1 ; f'_{ih}(x) = \frac{1}{2} [1 + f_{ih}(x)][1 - f_{ih}(x)] \quad (5.2)$$

และระหว่างชั้นฮิดเดนกับชั้นเอาต์พุตใช้ฟังก์ชันไวงานชนิดฟังก์ชันซิกมอยด์ทวิภาค (Binary Sigmoid Function) ดังสมการที่ (5.3) เพื่อให้ได้ค่าเอาต์พุตในช่วง (0,1)

$$f_{ho}(x) = \frac{1}{1 + \exp(-x)} ; f'_{ho}(x) = f_{ho}(x)[1 - f_{ho}(x)] \quad (5.3)$$

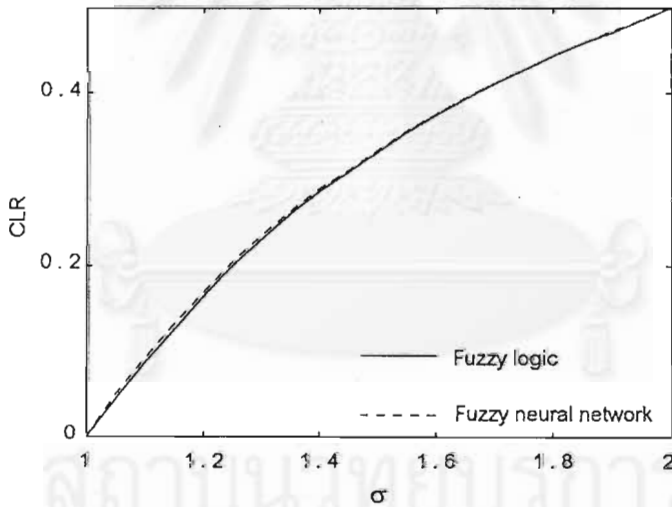
5.3 แบบจำลองแหล่งกำเนิดแบบที่ 1

จากหัวข้อที่ 4.2.1 เลือกขนาดโทเค้นที่ใช้ทดสอบตามเงื่อนไขในหัวข้อที่ 4.2 จึงเลือกให้ขนาดของบ่อโทเค้น (P) = 500 ค่าอินพุตที่เป็นค่าจริงสำหรับแบบจำลองแหล่งกำเนิดแบบที่ 1 ประกอบด้วย $N_c \in \{0,1,\dots,10\}$, $N_T \in \{0,1,\dots,500\}$, $\Delta N_c \in \{-10,-9,\dots,+10\}$ และ $R_i \in (8.80,62.50)$ เนื่องจากอินพุต R_i เป็นค่าต่อเนื่องมีจำนวนไม่จำกัด และชุดผสม (combination) ที่เกิดขึ้นระหว่าง N_c , N_T , ΔN_c และ R_i มีจำนวนมาก ดังนั้นเราจะเลือกเพียงบางค่ามาเป็นชุดฝึก โดยเลือกให้ $N_c \in \{0,1,2,\dots,10\}$, $N_T \in \{0,10,\dots,500\}$, $\Delta N_c \in \{-10,-9,\dots,+10\}$ และ $R_i \in \{10.0, 12.0,\dots,62.0\}$ ฝึกด้วย Marquardt-Lavenberg Back propagation Algorithm และค่าอินพุตดังกล่าวจะต้องผ่านขั้นตอน fuzzification เพื่อให้เป็นค่าพีชชีที่อยู่ระหว่าง (0,1) ก่อนนำไปป้อนให้กับนิวรอลเน็ตเวิร์กเป็นชุดฝึกการทดสอบหาจำนวนฮิดเดนโหนดที่เหมาะสมที่สุดได้ผลการทดสอบดังรูปที่ 5.1 คือเท่ากับ 18 โหนด เพราะทำให้ SSE ต่ำที่สุด

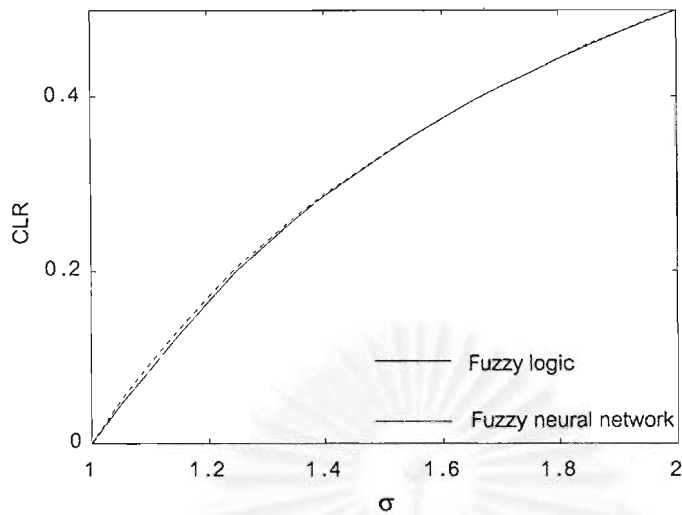


รูปที่ 5.1 จำนวนฮิดเดนโนดที่ใช้ทดสอบกรณีแบบจำลองแหล่งกำเนิดแบบที่ 1

หลังจากฝึกนิรอลเน็ตเวิร์กจนกระทั่ง SSE มีค่าค่อนข้างคงที่แล้ว จึงนำค่าถ่วงน้ำหนักสุดท้ายไปใช้กับตัวควบคุมฟัซซีนิรอลเน็ตเวิร์กและทดสอบความสามารถในการตรวจจับโหลดเกินเปรียบเทียบกับตัวตรวจหากราฟฟิคที่ใช้ตัวควบคุมฟัซซีลอจิกได้ผลดังแสดงในรูปที่ 5.2 และ 5.3



รูปที่ 5.2 เปรียบเทียบความสามารถในการตรวจจับโหลดเกินของตัวควบคุมฟัซซีนิรอลเน็ตเวิร์กกับตัวควบคุมฟัซซีลอจิกเมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



รูปที่ 5.3 เปรียบเทียบความสามารถในการตรวจจับโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

จากรูปที่ 5.2 และ 5.3 จะเห็นว่าค่าความผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟ-ฟิกที่ใช้ตัวควบคุมฟัซซีลอจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กเมื่อแปรค่า $E(X)$ และเมื่อแปรค่า $E(S)$ ของแหล่งกำเนิดมีค่าเป็น 5.14% และ 5.11% ตามลำดับ แม้ว่าเอาต์พุตที่ได้มีความแตกต่างกับเป้าหมายอยู่ไม่มากแต่ไม่สามารถยอมรับผลการฝึกได้ เพราะกลไกทำให้เกิดการละทิ้งเซลล์มากเกินไปกว่าอุดมคติ ซึ่งเป็นการละเมิด QoS ของผู้ใช้บริการ และสามารถตั้งสมมติฐานของปัญหาในการฝึกที่เกิดขึ้นได้ 4 ประเด็น คือ

1. วิธีเลือกอินพุตที่เป็นชุดฝึกไม่เหมาะสม
2. การเลือกอินพุตที่เป็นชุดฝึกเป็นจำนวนน้อยเกินไป
3. วิธีฝึกไม่เหมาะสม
4. มีจำนวนชุดผสมในการฝึกมากเกินไป

5.3.1 ผลการทดสอบเมื่อเปลี่ยนอินพุต

ต่อสมมติฐานข้อที่ 1 คือ ชุดข้อมูลที่ถูกเลือกเพื่อใช้ฝึกมีระยะห่างจากชุดข้อมูลทดสอบที่เกิดจากสภาวะการจำลองกราฟฟิกมากเกินไป ดังนั้นจึงเลือกชุดฝึกใหม่ 3 วิธี โดยกำหนดให้เป็นชุดผสมระหว่าง

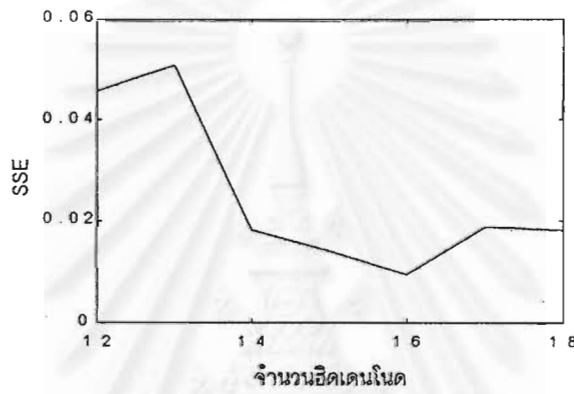
$$(ก) N_C \in \{0, 1, \dots, 10\}, N_T \in \{0, 5, 15, \dots, 495\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{10.0, 12.0, \dots, 62.0\}$$

(ข) $N_C \in \{0, 1, \dots, 10\}, N_T \in \{0, 10, \dots, 500\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{9.0, 11.0, \dots, 61.0\}$

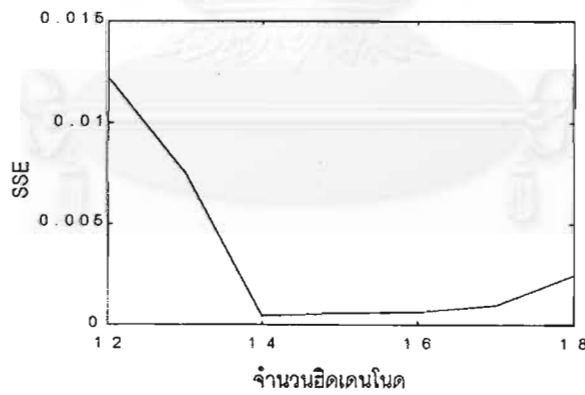
(ค) $N_C \in \{0, 1, \dots, 10\}, N_T \in \{0, 5, 15, \dots, 495\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{9.0, 11.0, \dots, 61.0\}$

หมายเหตุ : ในชุดผสมที่ $N_C \neq 0$ แล้ว $N_T = 0$ เท่านั้น และ $0 \leq |N_C + \Delta N_C| \leq \text{buffer size} = 10$

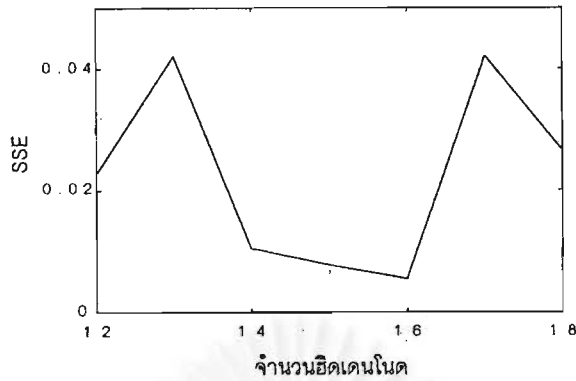
การทดสอบหาจำนวนฮิดเดนโนดของแต่ละวิธีได้ผลการทดสอบดังรูปที่ 5.4(ก)-5.4(ค)



(ก)



(ข)



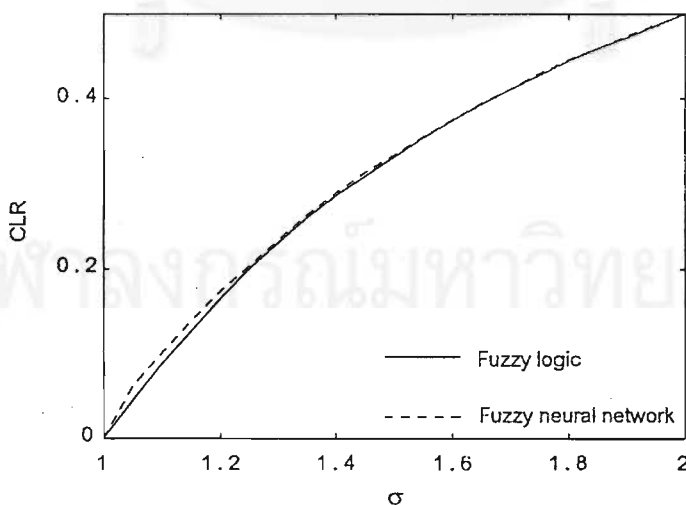
(ค)

รูปที่ 5.4 จำนวนฮิดเดนโนดที่ใช้ในการทดสอบเมื่อเปลี่ยนอินพุต

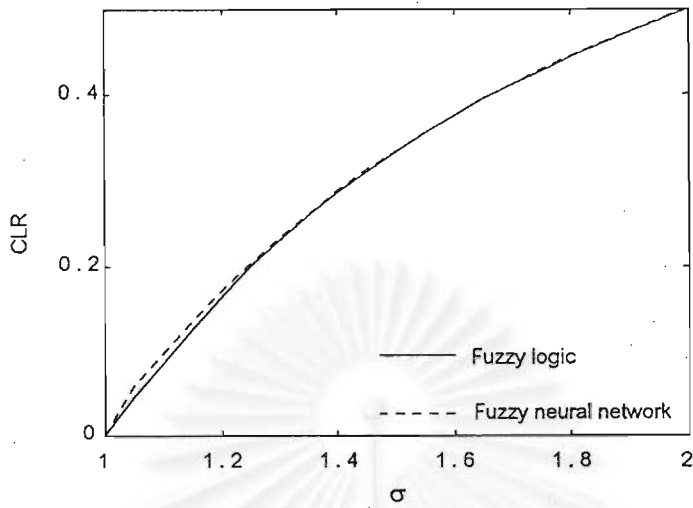
(ก) วิธี ก (ข) วิธี ข (ค) วิธี ค

จากรูปที่ 5.4(ก)-5.4(ค) ได้ว่าจำนวนฮิดเดนโนดที่เหมาะสมที่สุดสำหรับวิธี ก, ข และ ค เท่ากับ 16, 14 และ 16 โหนด ตามลำดับ ฝึกนิเวรอลเน็ตเวิร์กด้วยจำนวนฮิดเดนโนดและชุดฝึกดังกล่าว จนกระทั่ง SSE มีค่าค่อนข้างคงที่

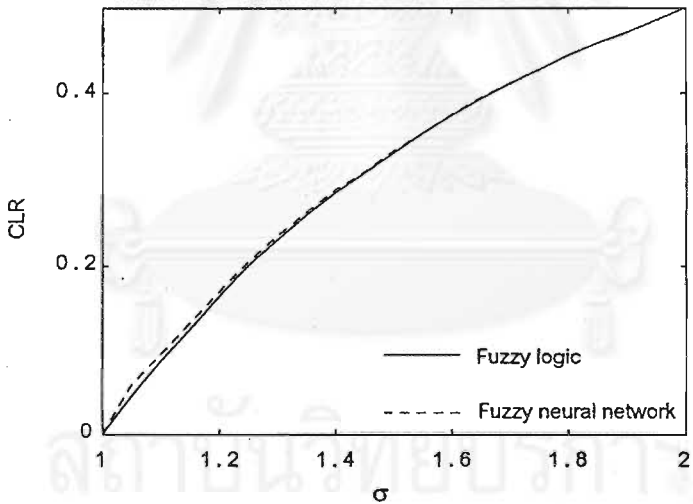
เมื่อนำค่าถ่วงน้ำหนักชุดสุดท้ายที่ได้จากการฝึกไปทำหน้าที่แทน inference engine ในตัวควบคุมอัตราการทำเนตโศเคนของตัวตรวจหาภาพก็ได้เส้นโค้งความสามารถในการตรวจวัดโผลดเกินดังรูปที่ 5.5-5.10 สำหรับวิธี (ก)-(ค) ตามลำดับ



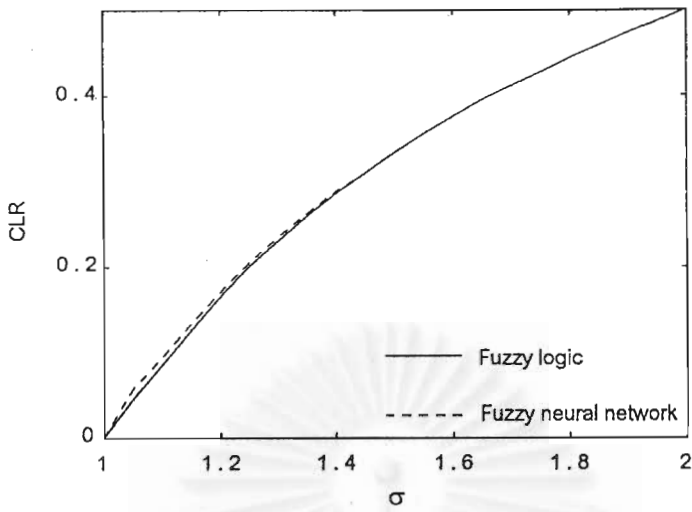
รูปที่ 5.5 เปรียบเทียบความสามารถในการตรวจวัดโผลดเกินของตัวควบคุมฟัซซีนิเวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตามวิธี (ก) ในกรณีแปรค่า $E(X)$



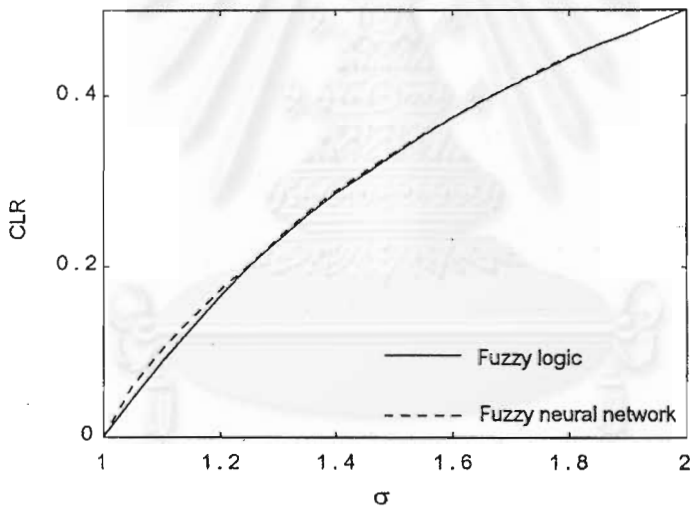
รูปที่ 5.6 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตามวิธี (ก) ในกรณีแปรค่า $E(S)$



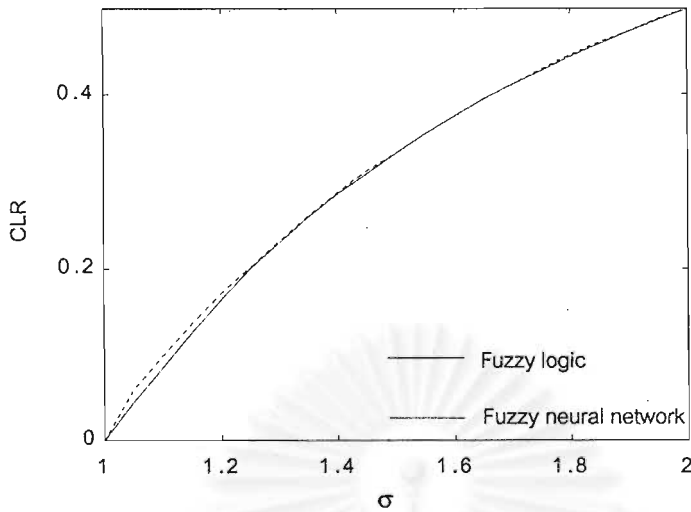
รูปที่ 5.7 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตามวิธี (ข) ในกรณีแปรค่า $E(X)$



รูปที่ 5.8 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตามวิธี (ข) ในกรณีแปรค่า $E(S)$



รูปที่ 5.9 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเปลี่ยนอินพุตตามวิธี (ค) ในกรณีแปรค่า $E(X)$



รูปที่ 5.10 เปรียบเทียบความสามารถในการตรวจจับโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อการเปลี่ยนอินพุตตามวิธี (ค) ในกรณีแปรค่า $E(S)$

จากรูปที่ 5.5-5.10 ค่าความผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจหาฟีกที่ใช้ตัวควบคุมฟัซซีลอจิกกับผลจากตัวตรวจหาฟีกที่ใช้ตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กในกรณีเลือกอินพุตวิธี (ก) เป็น 9.46%, 8.72% ในกรณีเลือกอินพุตวิธี (ข) มีค่าเป็น 5.60%, 5.35% และในกรณีเลือกอินพุตวิธี (ค) เป็น 9.87%, 9.43% เมื่อแปรค่า $E(X)$ และ $E(S)$ ของแหล่งกำเนิดตามลำดับ เห็นได้ว่าการเปลี่ยนอินพุตที่ใช้ฟีกทำให้ค่าผิดพลาดเพิ่มขึ้น การเลือกอินพุตจึงมีผลต่อประสิทธิภาพในการฟีก แต่การเปลี่ยนชุดฟีกทั้งสามวิธีดังกล่าวไม่สามารถทำให้ค่าผิดพลาดลดลงจึงต้องมีการทดสอบสมมติฐานอื่นๆ ต่อไปเพื่อให้ได้ค่าถ่วงน้ำหนักที่มีค่าผิดพลาดต่ำสุด

5.3.2 ผลการทดสอบเมื่อเพิ่มความถี่ในการเลือกอินพุต

ต่อสมมติฐานข้อที่ 2 คือ จำนวนชุดผสมที่ถูกเลือกมาเพื่อใช้ฟีกน้อยเกินไป และไม่สามารถเป็นตัวแทนของข้อมูลได้ทั้งหมด นิวรอลเน็ตเวิร์กจึงไม่สามารถให้เอาต์พุตที่ใกล้เคียงกับ inference engine ดังนั้นจึงกำหนดชุดฟีกใหม่โดยเลือกให้มีความถี่มากขึ้น แบ่งเป็น 3 ระดับ

(ก) เพิ่มความถี่การเลือกอินพุต N_T โดยชุดฟีกเป็นชุดผสมระหว่าง

$$N_C \in \{0, 1, 2, \dots, 10\}, N_T \in \{0, 5, \dots, 500\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{9.0, 11.0, \dots, 61.0\}$$

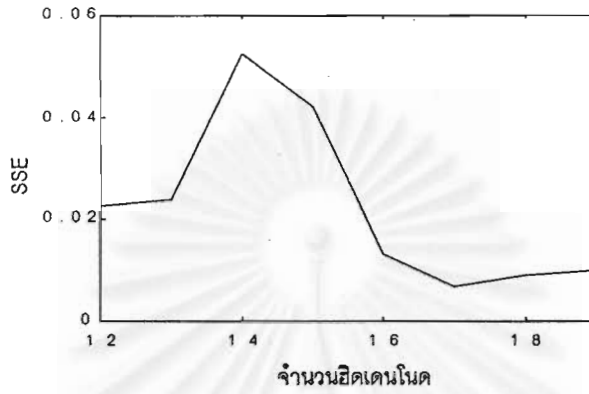
(ข) เพิ่มความถี่การเลือกอินพุต R_i โดยชุดฟีกเป็นชุดผสมระหว่าง

$$N_C \in \{0, 1, 2, \dots, 10\}, N_T \in \{0, 10, \dots, 500\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{9.0, 10.0, \dots, 62.0\}$$

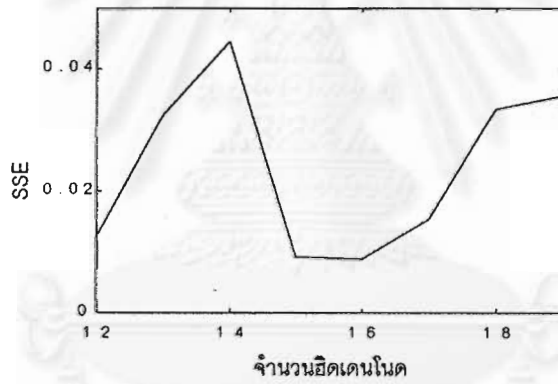
(ค) เพิ่มความถี่การเลือกอินพุต N_T และ R , โดยเลือกชุดฝึกเป็นชุดผสมระหว่าง

$$N_C \in \{0, 1, 2, \dots, 10\}, N_T \in \{0, 5, \dots, 500\}, \Delta N_C \in \{-10, -9, \dots, +10\}, R_i \in \{9.0, 10.0, \dots, 62.0\}$$

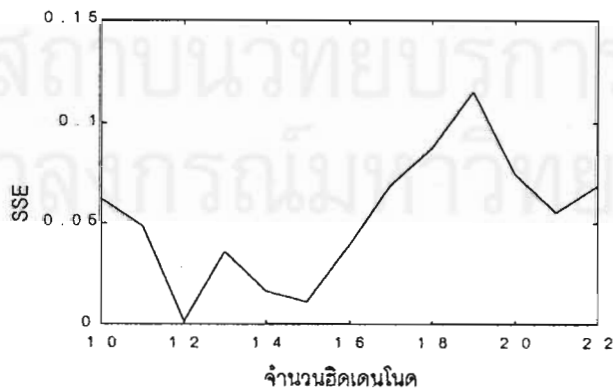
การทดสอบหาจำนวนฮิดเดนโนดของแต่ละวิธีได้ผลการทดสอบดังรูปที่ 5.11(ก)-5.11(ค)



(ก)



(ข)

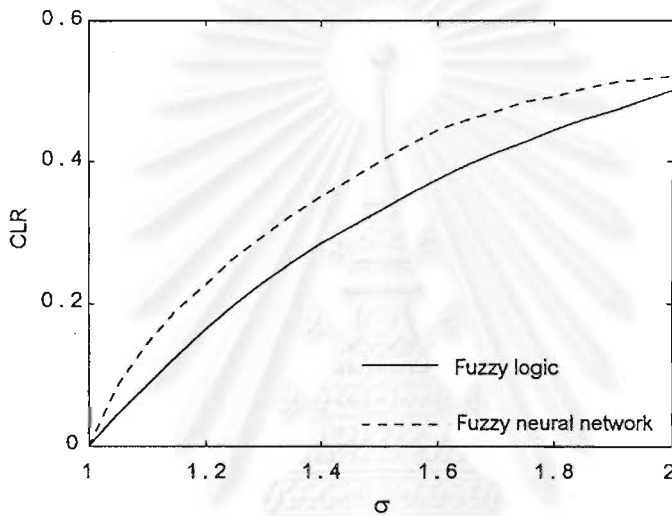


(ค)

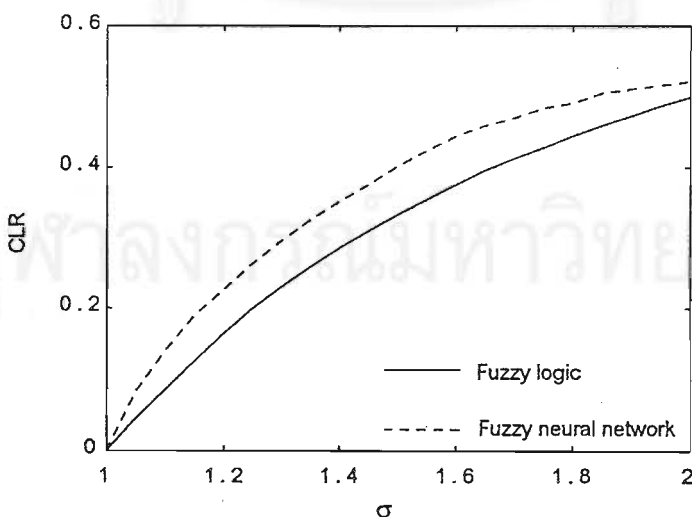
รูปที่ 5.11 จำนวนฮิดเดนโนดที่ใช้ในการทดสอบเมื่อเพิ่มความถี่ของอินพุต

(ก) วิธี ก (ข) วิธี ข (ค) วิธี ค

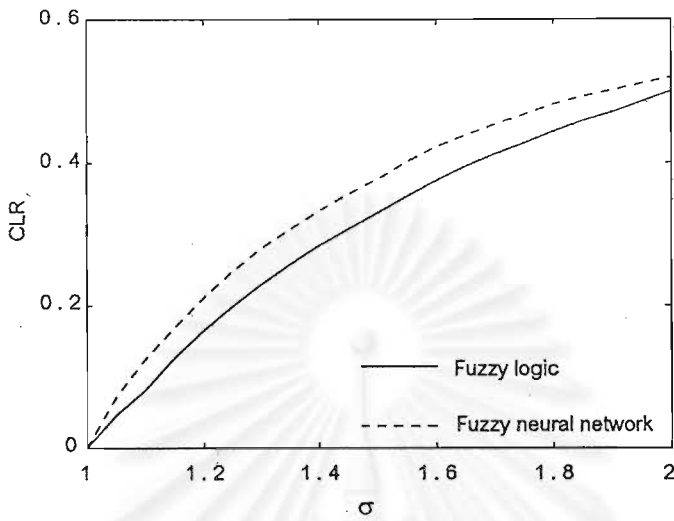
จากรูปที่ 5.11(ก)-5.11(ค) ได้ว่าจำนวนฮิดเดนโนดที่เหมาะสมที่สุดสำหรับวิธีที่ (ก), (ข) และ (ค) เท่ากับ 17, 16 และ 12 โหนดตามลำดับ เมื่อฝึกนิรอลเน็ตเวิร์กด้วยจำนวนฮิดเดนโนดและชุดฝึกดังกล่าว จนกระทั่ง SSE มีค่าค่อนข้างคงที่ พบว่าทั้งสามวิธียังคงไม่สามารถทำให้ SSE ต่ำกว่าระดับ 10^{-5} ที่ตั้งไว้ได้ และเมื่อนำค่าถ่วงน้ำหนักชุดสุดท้ายที่ได้จากการฝึกไปทำหน้าที่แทน inference engine ในตัวควบคุมอัตราการกำเนิดโทเค็นของตัวตรวจหากราฟฟิก ได้เส้นโค้งความสามารถในการตรวจวัดโหนดเกินดังรูปที่ 5.12-5.17 สำหรับวิธี (ก)-(ค) ตามลำดับ



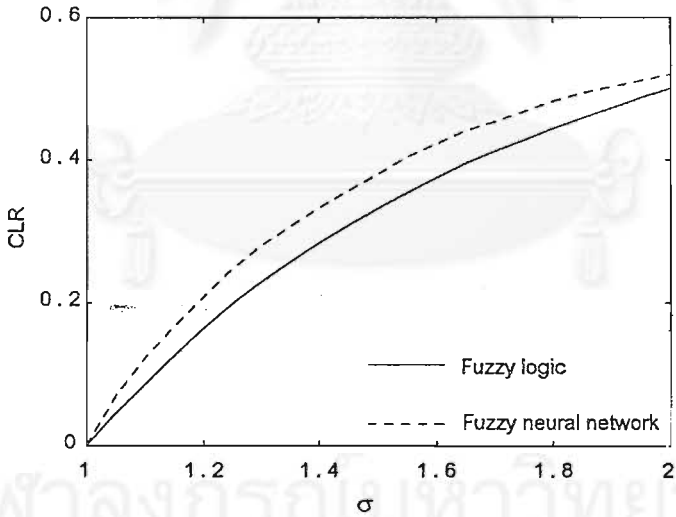
รูปที่ 5.12 เปรียบเทียบความสามารถในการตรวจวัดโหนดเกินของตัวควบคุมฟัซซีนิรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ก) ในกรณีแปรค่า $E(X)$



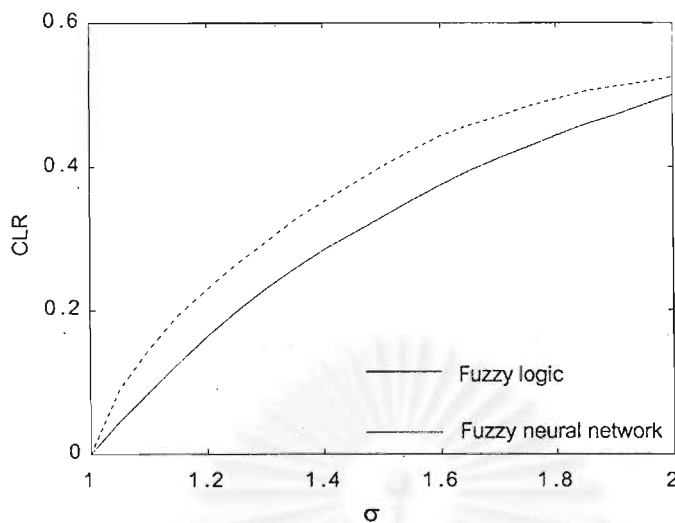
รูปที่ 5.13 เปรียบเทียบความสามารถในการตรวจวัดโหนดเกินของตัวควบคุมฟัซซีนิรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ก) ในกรณีแปรค่า $E(S)$



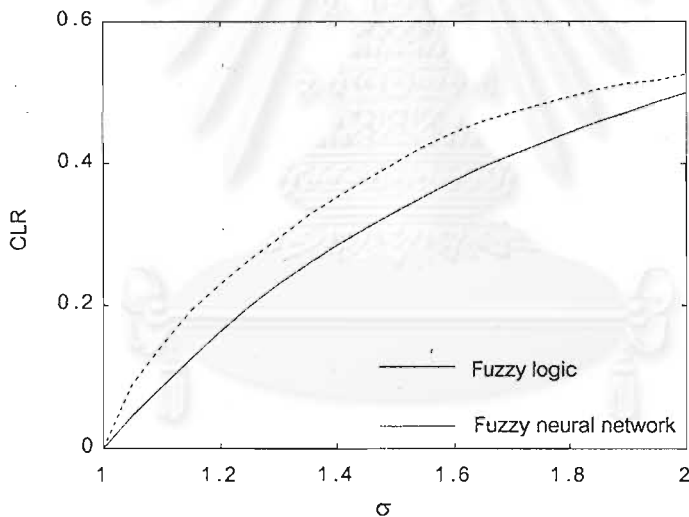
รูปที่ 5.14 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ข) ในกรณีแปรค่า $E(X)$



รูปที่ 5.15 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ข) ในกรณีแปรค่า $E(S)$



รูปที่ 5.16 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ค) ในกรณีแปรค่า $E(X)$



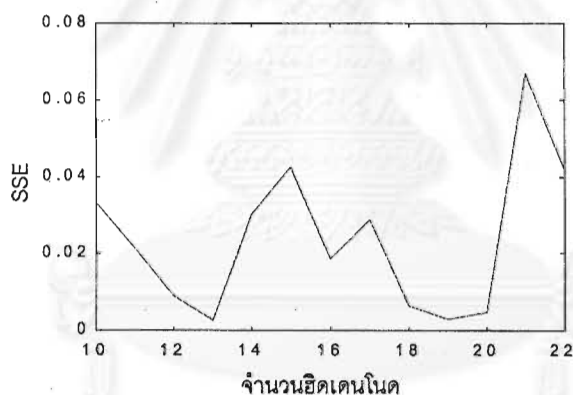
รูปที่ 5.17 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อเพิ่มอินพุตตามวิธี (ค) ในกรณีแปรค่า $E(S)$

จากรูปที่ 5.12-5.17 ค่าความผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมฟัซซีลอจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กสำหรับการเพิ่มความถี่การเลือกอินพุตวิธี (ก) ในกรณีแปรค่า $E(X)$ และ $E(S)$ มีค่าเป็น 57.30%, 56.84% สำหรับวิธี (ข) เป็น 32.34%, 31.55% และสำหรับวิธี (ค) เป็น 79.60%, 79.39% ตามลำดับ เห็นได้ว่าการเพิ่มจำนวนอินพุตกลับทำให้ค่าผิดพลาดเพิ่มมากขึ้น เพราะนิวรอลเน็ตเวิร์กไม่สามารถเรียนรู้ชุดฝึกทั้งหมดได้พร้อมๆ กัน จึงสรุปได้ว่าการเพิ่มอินพุตไม่ใช่วิธีแก้ปัญหาก็ถูกต้องในกรณีนี้

5.3.3 ผลการทดสอบเมื่อเปลี่ยนวิธีฝึก

ในการทดสอบที่ผ่านมา เราฝึกด้วยแบบวิธีหมู่ (batch mode) กล่าวคือป้อนอินพุตโดยเรียงลำดับก่อนหลังตามค่าของอินพุต และเมื่อป้อนชุดฝึกทั้งหมดครบ 1 ครั้ง จะคำนวณผลรวมความผิดพลาดทั้งหมดแล้วแพร่กระจายกลับเพื่อคำนวณค่าถ่วงน้ำหนักชุดใหม่ นับเป็น 1 รอบการฝึก ต่อสมมติฐานข้อที่ 3 ดังกล่าว หมายความว่าวิธีฝึกแบบเรียงลำดับก่อนหลังและเลือกชุดฝึกโดยมนุษย์อาจได้ค่าอินพุตที่มีระยะห่างกับค่าอินพุตจริงหรือมีจำนวนอินพุตมากเกินไป การแก้ปัญหาทำได้โดยการฝึกแบบไม่รบกวนการใช้งาน (On-line Training) กล่าวคือค่าอินพุตจะได้จากสถานะจริงในระบบ จึงเป็นการกำจัดปัญหาการเลือกชุดฝึกโดยมนุษย์ อินพุตที่ป้อนในแต่ละหมู่ (batch) จะไม่ซ้ำกันและได้จากการป้อนกราฟฟิคที่มี σ เรียงลำดับจาก 1.00 ถึง 2.00 จำนวน 10^4 เซลล์ต่อ σ 1 ค่า ให้กับตัวตรวจต่อเนื่องกัน

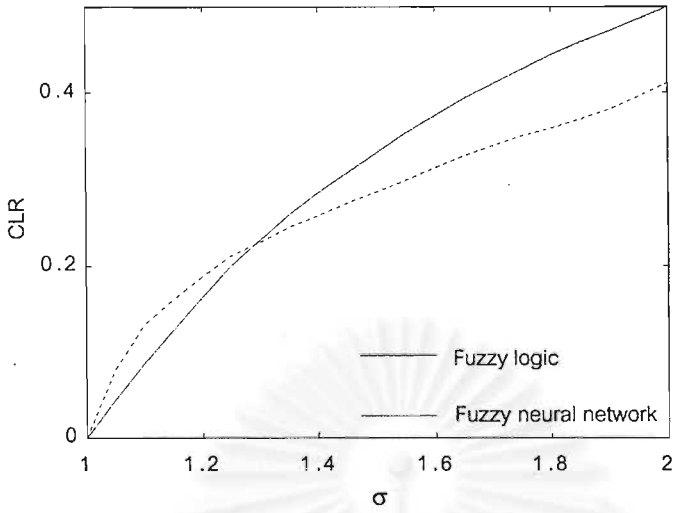
การทดสอบหาจำนวนฮิดเดนโนดได้ผลการทดสอบดังรูปที่ 5.18



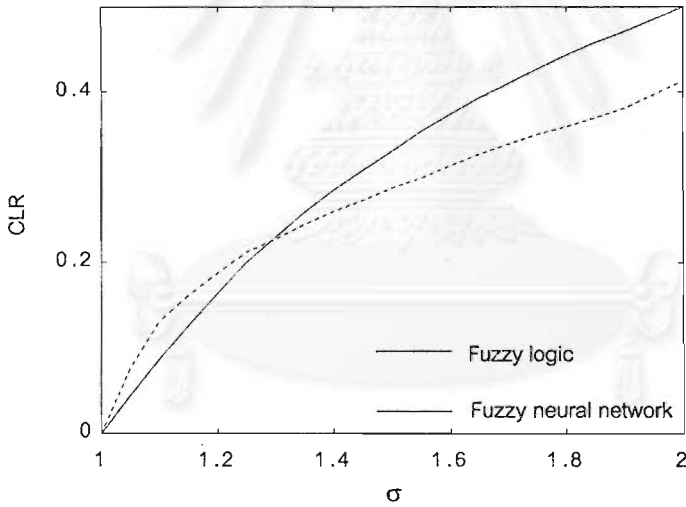
รูปที่ 5.18 จำนวนฮิดเดนโนดที่ใช้ทดสอบเมื่อเปลี่ยนวิธีการฝึกนิรอลเน็ตเวิร์ก

จากรูปที่ 5.18 ได้ว่าจำนวนฮิดเดนโนดที่เหมาะสมที่สุดเท่ากับ 13 โหนด และเมื่อฝึกนิรอลเน็ตเวิร์กด้วยจำนวนฮิดเดนโนดและชุดฝึกดังกล่าว จนกระทั่ง SSE มีค่าค่อนข้างคงที่ พบว่ายังคงไม่สามารถทำให้ SSE ต่ำกว่าระดับ 10^{-5} ที่ตั้งไว้ได้

เมื่อนำค่าถ่วงน้ำหนักชุดสุดท้ายที่ได้จากการฝึกไปทำหน้าที่แทน inference engine ในตัวควบคุมอัตราการกำเนิดโทเค็นของตัวตรวจกราฟฟิค ได้เส้นโค้งความสามารถในการตรวจวัดไหลเกินดังรูปที่ 5.19-5.20



รูปที่ 5.19 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมพีชชีนิวรอลเน็ตเวิร์ก กับตัวควบคุมพีชชีลอจิกเมื่อใช้วิธีฝึกแบบไม่รบกวนการใช้งาน ในกรณีแปรค่า $E(X)$



รูปที่ 5.20 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมพีชชีนิวรอลเน็ตเวิร์ก กับตัวควบคุมพีชชีลอจิกเมื่อใช้วิธีฝึกแบบไม่รบกวนการใช้งาน ในกรณีแปรค่า $E(S)$

จากรูปที่ 5.19-5.20 ค่าความผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีลอจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กมีค่าเป็น 88.86% และ 88.33% ตามลำดับ เห็นได้ว่าสมมติฐานดังกล่าวไม่ถูกต้อง และการเปลี่ยนวิธีฝึกไม่ได้ทำให้ค่าความผิดพลาดลดลงแต่กลับเพิ่มขึ้นเมื่อเปรียบเทียบกับกราฟฟิกโดยเรียงลำดับอินพุต ทั้งนี้เกิดจากการกระโดดของค่าอินพุตในแต่ละรอบการฝึก และการเปลี่ยนแปลงอินพุตไม่ซ้ำสำหรับ

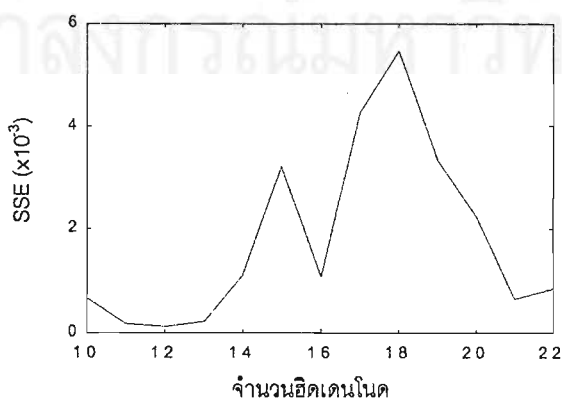
การฝึกแต่ละรอบ ทำให้นิเวรอลเน็ตเวิร์กไม่สามารถเรียนรู้ได้อย่างมีระบบ ดังนั้นการฝึกแบบไม่รบกวนการใช้งานจึงไม่ใช่ทางออกของปัญหา

5.3.4 ผลการทดสอบเมื่อลดจำนวนอินพุตที่ให้ค่าเอาต์พุตซ้ำซ้อน

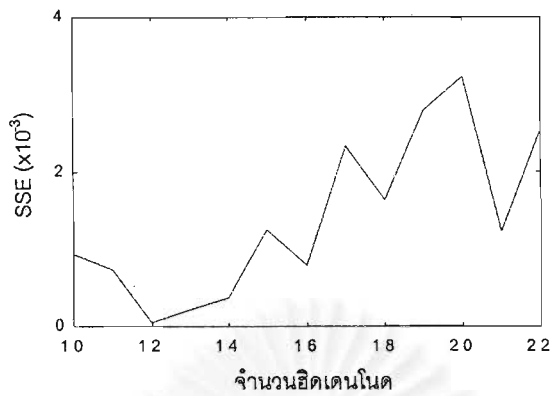
จากผลการทดสอบในหัวข้อที่ 5.3.1.2 สังเกตได้ว่าการเพิ่มอินพุต กลับทำให้ระดับคงที่ของ SSE เพิ่มขึ้น โดยที่ไม่สามารถฝึกให้ SSE ต่ำกว่า 10^{-4} ได้ ดังนั้นจึงสันนิษฐานได้ว่า หากลดจำนวนอินพุตลงอาจช่วยเพิ่มประสิทธิภาพในการฝึกและลดค่าผิดพลาดได้ เนื่องจากมีอินพุตที่มีค่าใกล้เคียงกันและให้เอาต์พุตที่ไม่ต่างกัน การเกาะกลุ่มของอินพุตจำนวนมากที่ให้ค่าเอาต์พุตเดียวกันจะทำให้การเรียนรู้ของนิเวรอลเน็ตเวิร์กไม่กระจาย เปรียบได้กับคนจดจำแต่เรื่องใดเรื่องหนึ่ง และไม่สามารถจดจำเรื่องอื่นๆ ได้ ดังนั้นจากหัวข้อ 5.3.1.1 พิจารณาชุดฝึก (ข) ซึ่งให้ค่าผิดพลาดต่ำสุดมีจำนวนชุดผสม 18,117 ชุด ลดอินพุตซ้ำซ้อนโดยการป้อนชุดผสมของอินพุตทั้งหมดให้กับ พืชซีลอจิกโดยเรียงลำดับตามค่าของอินพุตที่ต้องการลดความซ้ำซ้อน และกำจัดชุดผสมชุดกลางในชุดผสมที่มีลำดับติดกัน 3 ชุดซึ่งให้ค่าเอาต์พุตเท่ากัน วิธีนี้จึงสามารถลดจำนวนชุดผสมได้สูงสุด 50 % หากต้องการลดความซ้ำซ้อนของอินพุตมากกว่า 1 ตัว สามารถทำได้โดยนำชุดผสมที่ได้จากการลดความซ้ำซ้อนครั้งแรกมาเรียงลำดับใหม่ตามค่าอินพุตที่ 2 แล้วป้อนให้กับพืชซีลอจิก และทำซ้ำ ในการทดสอบแบ่งพิจารณาเป็น 3 กรณีคือ

- (ก) ลดความซ้ำซ้อนของ N_T เหลือชุดผสมทั้งหมด 11,714 ชุด
- (ข) ลดความซ้ำซ้อนของ R_i เหลือชุดผสมทั้งหมด 13,691 ชุด
- (ค) ลดความซ้ำซ้อนของ N_T และ R_i เหลือชุดผสมทั้งหมด 9,605 ชุด

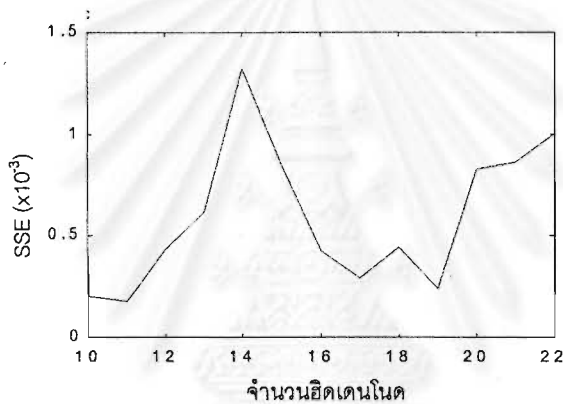
การทดสอบหาจำนวนฮิดเดนโนดของกรณี (ก)-(ค) ได้ผลการทดสอบดังรูปที่ 5.21(ก)-5.21(ค) ตามลำดับ



(ก)



(ข)

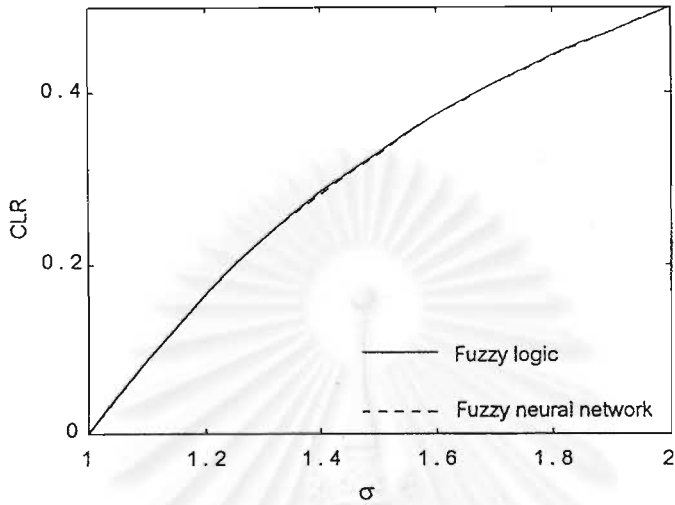


(ค)

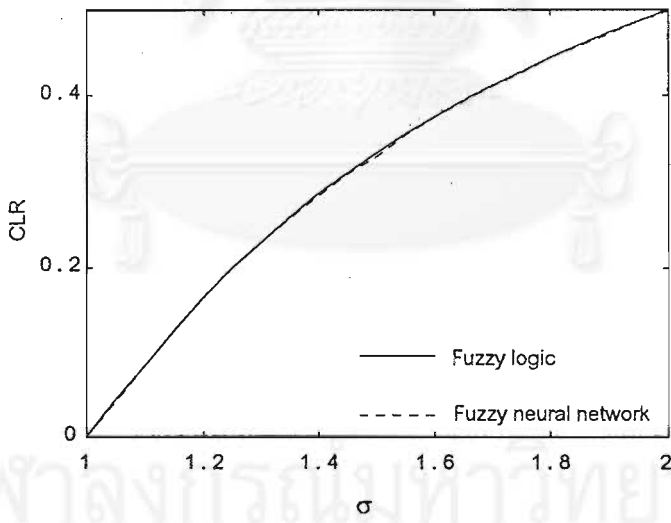
รูปที่ 5.21 จำนวนฮิตเดนโนดที่ใช้ในการทดสอบเมื่อลดความซ้ำซ้อนของอินพุต
(ก) วิธี ก (ข) วิธี ข (ค) วิธี ค

จากรูปที่ 5.21(ก)-5.21(ค) จะเห็นว่าจำนวนฮิตเดนโนดที่เหมาะสมที่สุดสำหรับกรณี (ก)-(ค) เท่ากับ 12, 12, 11 โหนด ตามลำดับ ฝึกนิรอรอลเน็ตเวิร์กด้วยจำนวนฮิตเดนโนดและชุดฝึกดังกล่าว จนกระทั่ง SSE มีค่าค่อนข้างคงที่

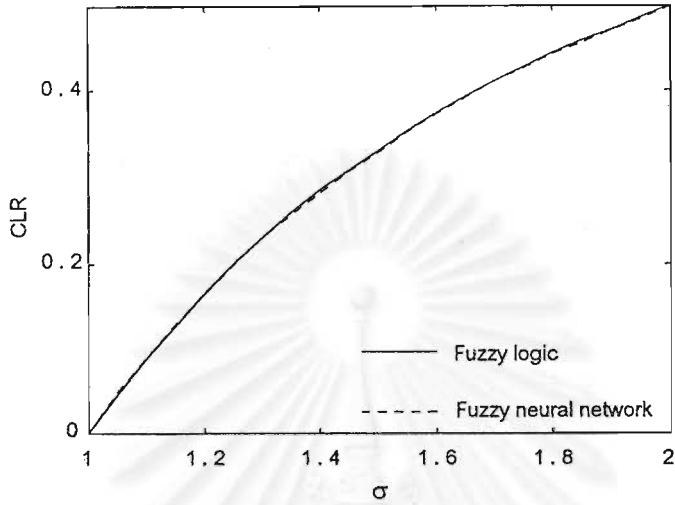
เมื่อนำค่าถ่วงน้ำหนักชุดสุดท้ายที่ได้จากการฝึกไปทำหน้าที่แทน inference engine ในตัวควบคุมอัตราการทำเนตโหนดของตัวตรวจตราฟฟิก ได้เส้นโค้งความสามารถในการตรวจวัดโหลดเกินดังรูปที่ 5.22-5.27



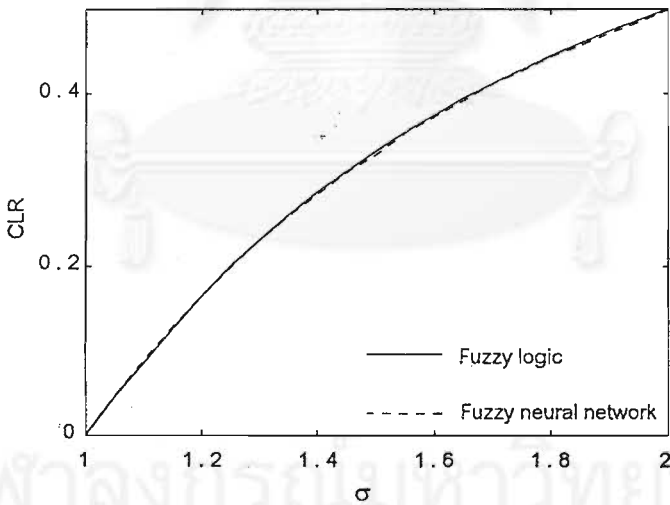
รูปที่ 5.22 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุตวิธี ก ในกรณีแปรค่า $E(X)$



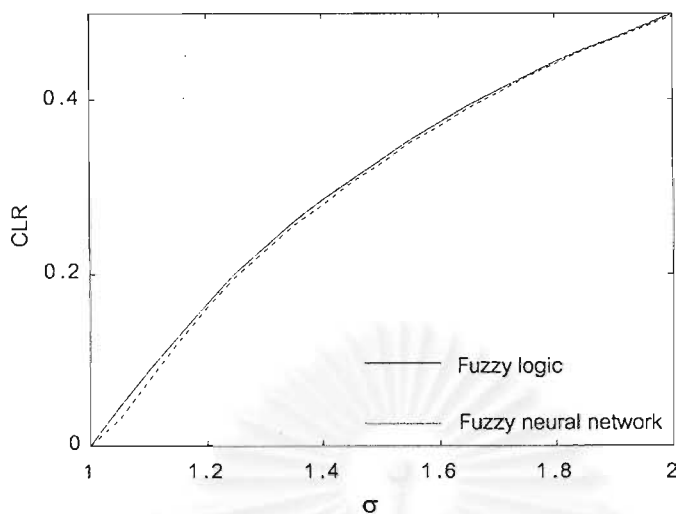
รูปที่ 5.23 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุตวิธี (ก) ในกรณีแปรค่า $E(S)$



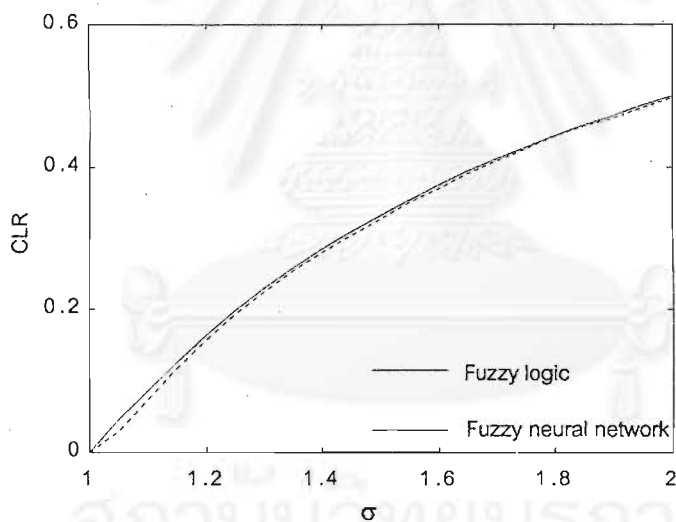
รูปที่ 5.24 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุต วิธี (ข) ในกรณีแปรค่า $E(X)$



รูปที่ 5.25 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุต วิธี (ข) ในกรณีแปรค่า $E(S)$



รูปที่ 5.26 เปรียบเทียบความสามารถในการตรวจวัดโหนดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุต วิธี (ค) ในกรณีแปรค่า $E(X)$



รูปที่ 5.27 เปรียบเทียบความสามารถในการตรวจวัดโหนดเกินของตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์ก กับตัวควบคุมฟัซซีลอจิกเมื่อลดความซ้ำซ้อนของอินพุต วิธี (ค) ในกรณีแปรค่า $E(S)$

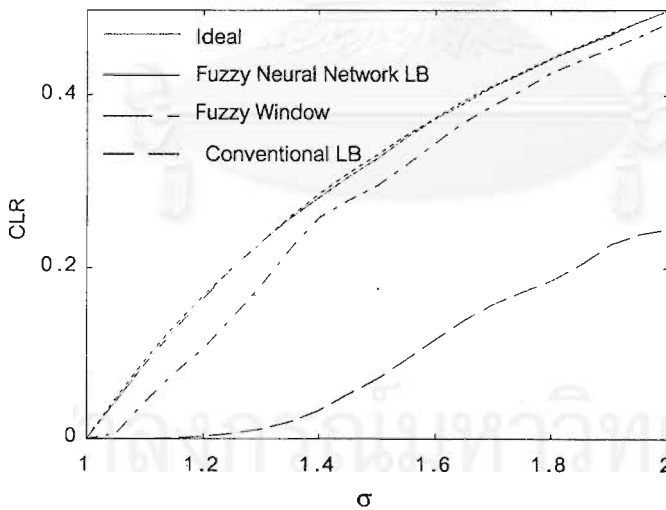
จากรูปที่ 5.22-5.27 ค่าความผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมฟัซซีลอจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กสำหรับวิธีลดความซ้ำซ้อนของอินพุตแบบ (ก) ในกรณีแปรค่า $E(X)$ และ แปรค่า $E(S)$ มีค่าเป็น 0.80%, 0.94% และสำหรับแบบ (ข) เป็น 1.97%, 1.87% และสำหรับแบบ (ค) เป็น 7.11%, 7.37% ตามลำดับ เห็นได้ว่าการลดจำนวนอินพุตลงช่วยให้ค่าผิดพลาดลดลงได้และนิวรอลเน็ตเวิร์กสามารถให้เอาต์-

พู่ที่ใกล้เคียงกับ inference engine มากในกรณี (ก) และ (ข) แต่การลดจำนวนอินพุตมากเกินไป ทำให้นิวรอลเน็ตเวิร์กไม่ได้รับข้อมูลที่เพียงพอและไม่สามารถให้ผลได้ตามที่ต้องการ ดังนั้นการเลือกชุดผสมจึงไม่ควรมากเกินไป เพราะนอกจากจะทำให้ใช้เวลาฝึกนานแล้ว นิวรอลเน็ตเวิร์กอาจไม่สามารถเรียนรู้ได้ทั้งหมดหรือเกิดปัญหาเรียนหน้าล้มหลัง และไม่ควรเลือกชุดผสมน้อยเกินไป เพื่อให้นิวรอลเน็ตเวิร์กมีข้อมูลเพียงพอสำหรับค่าถ่วงน้ำหนักที่มีประสิทธิภาพ

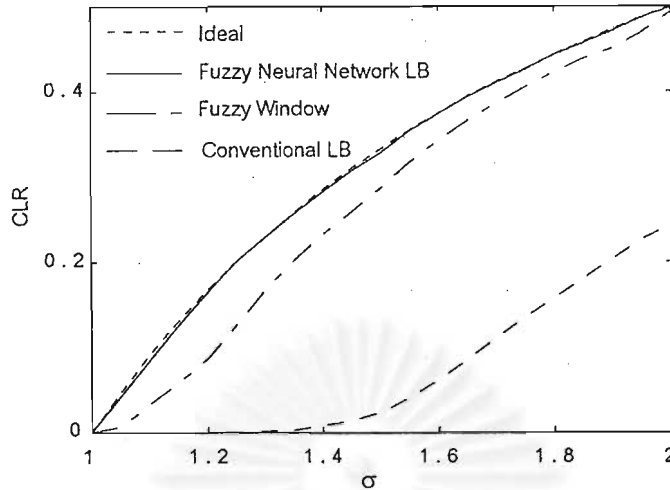
5.3.5 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก

จากการทดสอบตัวตรวจหากราฟฟิคที่ใช้ตัวควบคุมฟัซซีนิวรอลเน็ตเวิร์กที่ได้ชุดของค่าถ่วงน้ำหนักจากการฝึกในหัวข้อ 5.3.4 วิธี (ก) ซึ่งให้ค่าผิดพลาดต่ำสุด เราได้เส้นโค้งความสามารถในการตรวจวัดโหลดเกิน เปรียบเทียบกับเส้นโค้งความน่าจะเป็นการละเมิดอุดมคติ กลไกถ่วงรั้วดั้งเดิม และกลไกหน้าต่างฟัซซีลอจิกดังรูปที่ 5.28, 5.29

จากรูปได้ว่า RSSE และ false alarm probability ของตัวตรวจหากราฟฟิคถ่วงรั้วฟัซซีนิวรอลเน็ตเวิร์ก (FNN-LB) กลไกถ่วงรั้ว (LB) และกลไกหน้าต่างฟัซซีลอจิก (Fwindow) แสดงดังตารางที่ 5.1



รูปที่ 5.28 ความสามารถในการตรวจวัดโหลดเกิน เมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



รูปที่ 5.29 ความสามารถในการตรวจวัดไหลเกิน เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

ตารางที่ 5.1 เปรียบเทียบความสามารถในการเลือกของกลไกตรวจหาฟลัก
เมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1

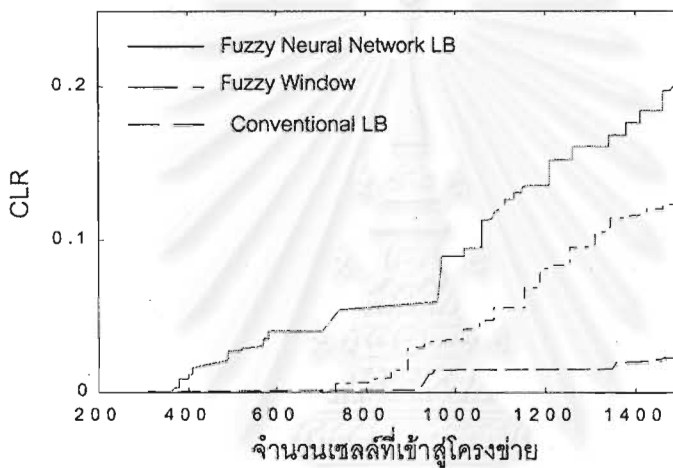
| กลไก | | ถึงรั้ว แบบดั้งเดิม | หน้าต่างพีชชีลอจิก | ถึงรั้วพีชชี- นิรอลเน็ตเวิร์ก |
|-------------------------|-------------------|------------------------|-----------------------|----------------------------------|
| RSSE | แปรค่า E (X) | 1.023511 | 0.161944 | 0.011550 |
| | แปรค่า E (S) | 1.145818 | 0.207957 | 0.011744 |
| False Alarm Probability | | 2.72×10^{-5} | 1.20×10^{-8} | 5.22×10^{-7} |

จากตารางที่ 5.1 พบว่ากลไกถึงรั้วพีชชีนิรอลเน็ตเวิร์กยังคงมีความสามารถในการตรวจวัดไหลเกินดีที่สุดโดยมี RSSE ต่ำที่สุด แต่มี false alarm probability สูงกว่ากลไกหน้าต่างพีชชีลอจิก และเมื่อเปรียบเทียบกับกลไกถึงรั้วที่ใช้ตัวควบคุมพีชชีลอจิกในตารางที่ 4.1 และ 4.2 พบว่าจะมี false alarm probability ลดลง ในขณะที่ RSSE เมื่อเปรียบเทียบกับผลจากตารางที่ 4.3 เพิ่มขึ้นอันเกิดจาก CLR ที่ขนาดไหลต่างๆ ลดลง หมายถึงพฤติกรรมของกลไกโดยรวมมีความเข้มงวดลดลงไม่ว่าที่ไหลเกินหรือไหลปกติ อธิบายได้ว่าค่าผิดพลาดในการฝึกนิรอลเน็ตเวิร์กมีแนวโน้มในทิศทางที่ทำให้อัตราการกำเนิดไหลเกินเฉลี่ยสูงขึ้น นอกจากนี้สังเกตได้ว่า

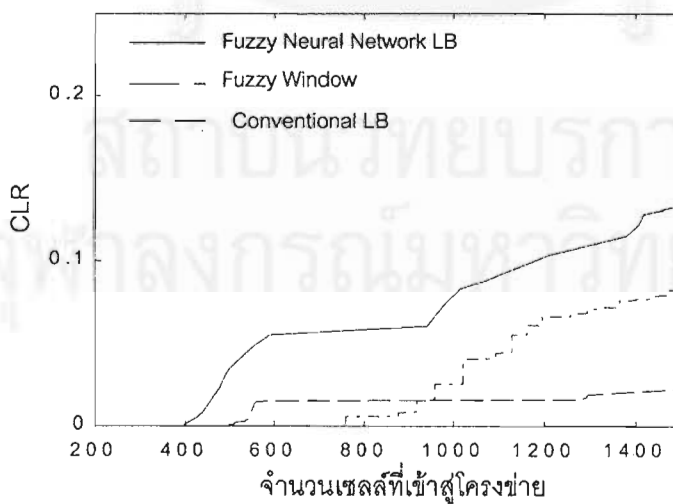
มากนัก เนื่องจากไม่ได้ฝึกแบบไม่รบกวนการใช้งาน (On-line Training) ทำให้ชุดข้อมูลที่ใช้ฝึกไม่เกี่ยวข้องกับลักษณะกราฟฟิก

5.3.6 ผลการทดสอบเปรียบเทียบการตอบสนอง

จากการทดสอบการตอบสนองที่ $\sigma = 1.50$ และเปรียบเทียบผลที่ได้จากตัวตรวจถึงรั้วฟัซซีนิวรอลเน็ตเวิร์กกับถึงรั้วและหน้าต่างฟัซซีลอจิก เราได้ผลดังแสดงในรูปที่ 5.30, 5.31



รูปที่ 5.30 การตอบสนองของกลไกตรวจจรรยาฟฟิก เมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



รูปที่ 5.31 การตอบสนองของกลไกตรวจจรรยาฟฟิก เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

จากผลการทดสอบ กลไกถ่วงรั้วพีชชีนิวรอลเน็ตเวิร์กยังคงมีพฤติกรรมพลวัตโดยรวมที่ดีกว่ากลไกหน้าต่างพีชชีลอคจิกและกลไกถ่วงรั้ว และมีการตอบสนองเริ่มต้นเร็วกว่ากลไกหน้าต่างพีชชีลอคจิก แม้ว่าจะช้าลงเล็กน้อยเมื่อเปรียบเทียบกับกรณีใช้ตัวควบคุมพีชชีลอคจิกในบทที่ 4 อธิบายได้จากการลดลงของอัตราการสูญหายของเซลล์ที่ไหลตกเกิน $\sigma = 1.50$ จากตารางที่ 5.2

ตารางที่ 5.2 เปรียบเทียบการตอบสนองเริ่มต้นของกลไกถ่วงรั้วพีชชีนิวรอลเน็ตเวิร์ก กับถ่วงรั้วพีชชีลอคจิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 1

| ลักษณะการแปรค่า ทราฟฟิกไหลตก | อัตราการสูญหายของเซลล์ที่ $\sigma = 1.50$ | | จำนวนเซลล์ที่เข้าสู่โครงข่ายก่อน การละทิ้งเซลล์แรกที่ $\sigma = 1.50$ | |
|---------------------------------|--|-----------------------------------|--|-----------------------------------|
| | ถ่วงรั้วพีชชีลอคจิก | ถ่วงรั้วพีชชี นิวรอลเน็ตเวิร์ก | ถ่วงรั้วพีชชีลอคจิก | ถ่วงรั้วพีชชี นิวรอลเน็ตเวิร์ก |
| เมื่อแปรค่า $E(X)$ | 0.330780 | 0.328320 | 328 | 359 |
| เมื่อแปรค่า $E(S)$ | 0.332208 | 0.328453 | 334 | 392 |

5.3.7 ผลการทดสอบเปรียบเทียบเวลาประวิง

จากการทดสอบเวลาประวิงเฉลี่ยของแต่ละเซลล์ในสภาวะไหลตกปกติ และเปรียบเทียบกับเมื่อใช้พีชชีลอคจิกได้ผลดังตารางที่ 5.3

ตารางที่ 5.3 เปรียบเทียบเวลาประวิงเฉลี่ยต่อ 1 เซลล์ที่เกิดจากกลไกตรวจทราฟฟิกเมื่อใช้ตัวควบคุมต่างกัน

| ตัวควบคุม | พีชชีลอคจิก | พีชชีนิวรอลเน็ตเวิร์ก |
|-----------------|-------------|-----------------------|
| เวลาประวิง (ms) | 0.87215 | 0.804140 |

จากตารางที่ 5.3 พบว่าเวลาประวิงของตัวตรวจลดลงเล็กน้อย บ่งบอกว่าตัวตรวจที่ใช้พีชชีนิวรอลเน็ตเวิร์กมีอัตราการกำเนิดโทเค็นโดยเฉลี่ยสูงขึ้นเมื่อเปรียบเทียบกับตัวควบคุมพีชชีลอคจิก สอดคล้องกับผลการทดสอบความสามารถในการเลือกและการตอบสนองที่ป่งชี้ไปในทิศทางเดียวกัน โดยในกรณีนี้ถือว่าความคลาดเคลื่อนดังกล่าวไม่ทำให้พฤติกรรมของระบบเลวร้ายลงมากแต่อย่างใด เนื่องจากเราเลือกใช้ขนาดของบ่อโทเค็นที่ทำให้ความเข้มงวดในการตรวจสูงสุด ดังนั้น

การใช้นิวรอลเน็ตเวิร์กซึ่งทำให้ตัวตรวจมีพฤติกรรมที่อ่อนปรนลง แม้ทำให้สมรรถนะการตรวจไหลตื้นลดลงบ้าง แต่ทำให้ false alarm probability และเวลาประวิงลดลงด้วย

5.4 แบบจำลองแหล่งกำเนิดแบบที่ 2

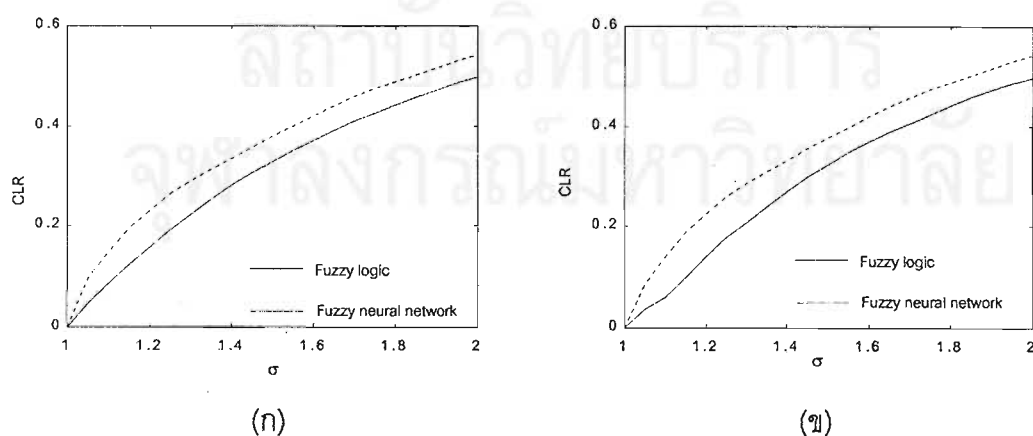
5.4.1 การแก้ปัญหาโดยใช้นิวรอลเน็ตเวิร์ก 1 ชุด

5.4.1.1 การเลือกชุดฝึกโดยไม่ลดความซ้ำซ้อนของอินพุตที่ให้เอาต์พุตเท่ากัน

จากหัวข้อที่ 4.2.2 เลือกขนาดโหนดที่ใช้ทดสอบตามเงื่อนไขในหัวข้อที่ 4.2 โดยคำนึงถึงความเข้มงวดในการตรวจเป็นเกณฑ์ จึงเลือกให้ $P = 3,600$ ค่าอินพุตที่เป็นค่าจริงสำหรับแบบจำลองแหล่งกำเนิดแบบที่ 2 ประกอบด้วย $N_C \in \{0, 1, \dots, 30\}$, $N_T \in \{0, 1, \dots, 3600\}$, $\Delta N_C \in \{-30, -29, \dots, +30\}$ และ $R_i \in (11.60, 83.30)$

เนื่องจากชุดผสมทั้งหมดมีจำนวนไม่จำกัด จึงจะเลือกบางค่ามาเป็นชุดฝึก โดยเลือกให้ $N_C \in \{0, 2, \dots, 30\}$, $N_T \in \{0, 50, \dots, 3600\}$, $\Delta N_C \in \{-30, -28, \dots, +30\}$ และ $R_i \in \{12.0, 14.0, \dots, 82.0\}$

ทำการทดสอบหาจำนวนฮิดเดนโนดที่เหมาะสมที่สุดด้วยวิธีเดียวกับหัวข้อที่ 5.3 ได้เท่ากับ 18 โหนด เพราะทำให้ SSE ต่ำที่สุดเปรียบเทียบที่จำนวนรอบฝึกเท่ากัน เมื่อใช้จำนวนฮิดเดนโนดดังกล่าวฝึกนิวรอลเน็ตเวิร์กต่อไปจนกระทั่ง SSE คงที่ พบว่าไม่สามารถทำให้ SSE ต่ำกว่าระดับ 10^{-5} ได้ เมื่อนำชุดของค่าถ่วงน้ำหนักสุดท้ายมาใช้ในตัวตรวจกราฟฟิกและทดสอบความสามารถในการตรวจวัดไหลตื้นเปรียบเทียบกับตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีซีลจิกได้ผลดังรูปที่ 5.32



รูปที่ 5.32 ความสามารถในการตรวจวัดไหลตื้นเมื่อไม่ลดความซ้ำซ้อนของอินพุต
(ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปร $E(S)$ ของแหล่งกำเนิด

จากรูปที่ 5.32 ค่าผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีลอจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กเมื่อแปรค่า $E(X)$ และเมื่อแปรค่า $E(S)$ เป็น 54.69% และ 62.99% ตามลำดับ และตัวตรวจมีพฤติกรรมละเมิด QoS ของผู้ให้บริการอีกด้วย จึงไม่สามารถยอมรับได้

เนื่องจากชุดผสมที่ใช้ฝึกมีจำนวนมากถึง 50,688 ชุด นิวรอลเน็ตเวิร์กไม่สามารถจะเรียนรู้ได้ทั้งหมด จึงไม่สามารถฝึกให้ SSE ต่ำกว่าระดับที่ต้องการได้ ดังนั้นแก้ปัญหาโดยลดจำนวนชุดฝึกลงด้วยวิธีลดความซ้ำซ้อนของชุดฝึกที่ให้เอาต์พุตเท่ากันเช่นเดียวกับหัวข้อที่ 5.3.4

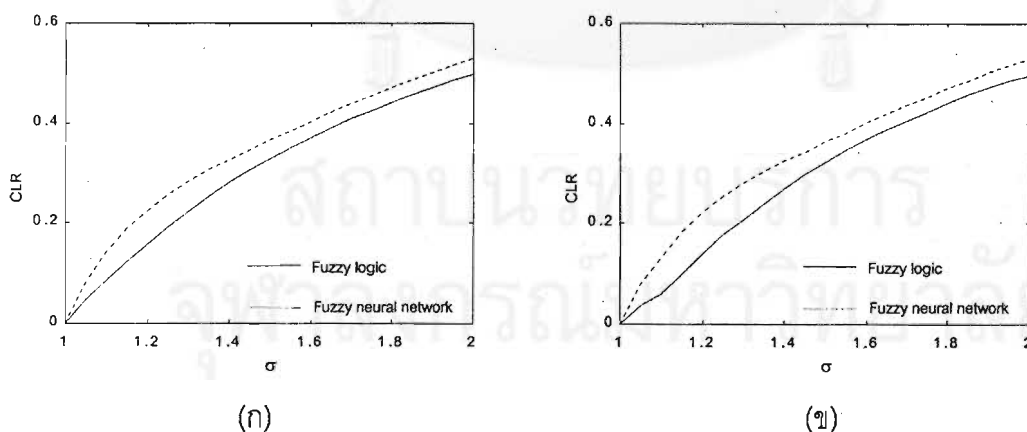
5.4.1.2 การเลือกชุดฝึกโดยลดความซ้ำซ้อนของอินพุตที่ให้เอาต์พุตเท่ากัน

การทดสอบแบ่งพิจารณาเป็น 3 กรณีคือ

- (ก) ลดความซ้ำซ้อนของ N_T เหลือชุดผสมทั้งหมด 37,694 ชุด
- (ข) ลดความซ้ำซ้อนของ R , เหลือชุดผสมทั้งหมด 40,470 ชุด
- (ค) ลดความซ้ำซ้อนของ N_T และ R , เหลือชุดผสมทั้งหมด 28,511 ชุด

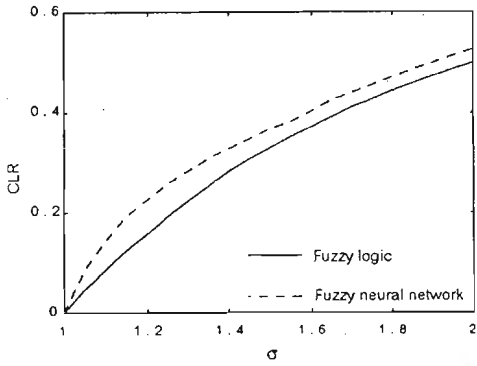
การทดสอบหาจำนวนฮิตเดโนในดของกรณี (ก)-(ค) ได้เป็น 14, 16 และ 16 ตามลำดับและเมื่อนำค่าถ่วงน้ำหนักสุดท้ายไปทดสอบความสามารถในการตรวจวัดโหลดเกิน ได้ผลดังรูปที่

5.33-5.35

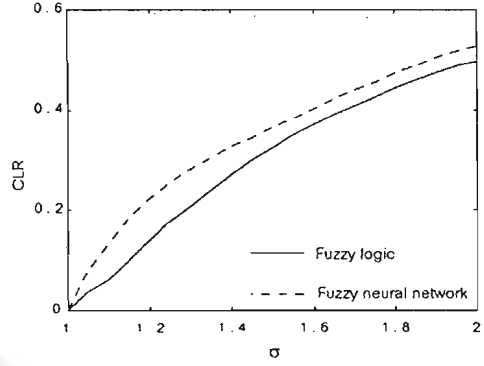


รูปที่ 5.33 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุตวิธี ก

(ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปร $E(S)$ ของแหล่งกำเนิด

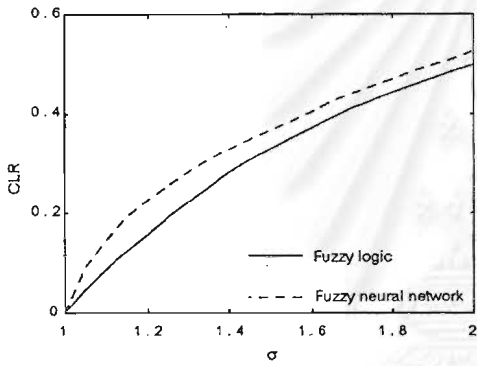


(ก)

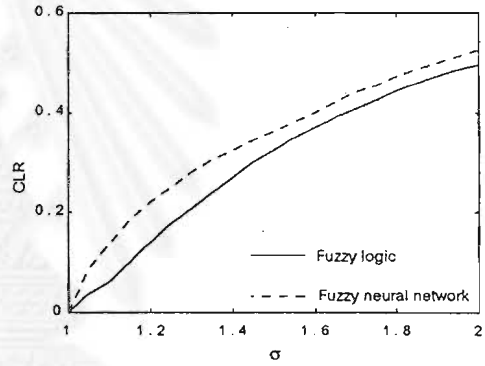


(ข)

รูปที่ 5.34 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุตวิธี ข
(ก) เมื่อแปร E(X) ของแหล่งกำเนิด (ข) เมื่อแปร E(S) ของแหล่งกำเนิด



(ก)



(ข)

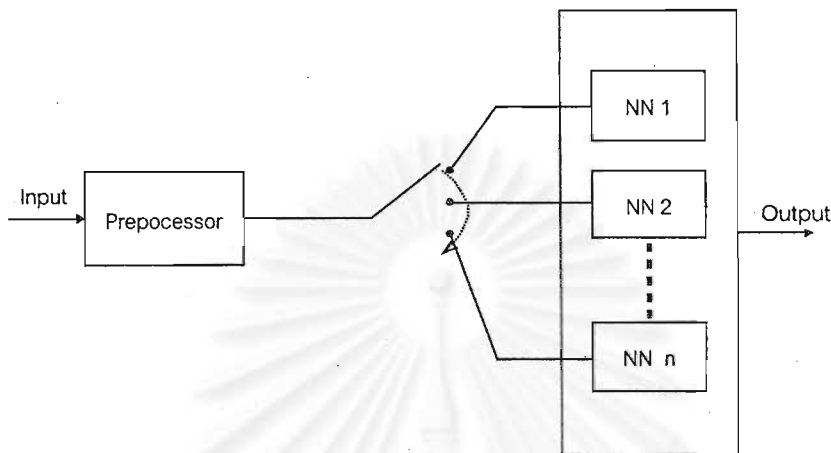
รูปที่ 5.35 ความสามารถในการตรวจวัดโหลดเกินเมื่อลดความซ้ำซ้อนของอินพุตวิธี ค
(ก) เมื่อแปร E(X) ของแหล่งกำเนิด (ข) เมื่อแปร E(S) ของแหล่งกำเนิด

จากรูปที่ 5.33-5.35 ค่าผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีลลจิกกับผลจากตัวตรวจกราฟฟิกที่ใช้ตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กสำหรับวิธี (ก) เมื่อแปรค่า E(X) และเมื่อแปรค่า E(S) เป็น 29.10%, 37.64% สำหรับวิธี (ข) เป็น 39.95%, 47.91% และสำหรับวิธี (ค) เป็น 66.42%, 74.11% ตามลำดับ ค่าผิดพลาดดังกล่าวค่อนข้างสูง จึงไม่สามารถยอมรับได้ ดังนั้นในกรณีนี้จึงไม่สามารถใช้วิธีลดอินพุตดังที่กล่าวมาเพื่อแก้ปัญหาได้

5.4.2 การแก้ปัญหาโดยใช้นิวรอลเน็ตเวิร์กแบบขนาน

เนื่องจากมีชุดผสมในการฝึกจำนวนมาก แม้ว่าจะลดจำนวนชุดผสมลงมาบ้างแล้ว แต่ก็ไม่สามารถทำให้นิวรอลเน็ตเวิร์กทำได้อย่างมีประสิทธิภาพ ทางออกของปัญหาคือการแบ่งชุดฝึก

และใช้นิวรอลเน็ตเวิร์กมากกว่า 1 ชุดในการเรียนรู้ข้อมูลเพื่อให้นิวรอลเน็ตเวิร์กชุดหนึ่งรับผิดชอบจำนวนข้อมูลลดลงโดยที่ความละเอียด (resolution) ของข้อมูลไม่ลดลง โครงสร้างของการทำงานเป็นแบบขนาน [9] ดังแสดงในรูปที่ 5.36



รูปที่ 5.36 แบบจำลองของนิวรอลเน็ตเวิร์กแบบขนาน

เราแบ่งข้อมูลเป็น 2, 3, 4 และ 5 ชุด ตามค่า R , เพื่อให้จำนวนข้อมูลในทุกๆ ชุดใกล้เคียงกันได้ดังนี้

(ก) แบ่งเป็น 2 ชุด โดย $R_{i1} \in \{11.0, 13.0, \dots, 45.0\}$ และ $R_{i2} \in \{47.0, 49.0, \dots, 83.0\}$

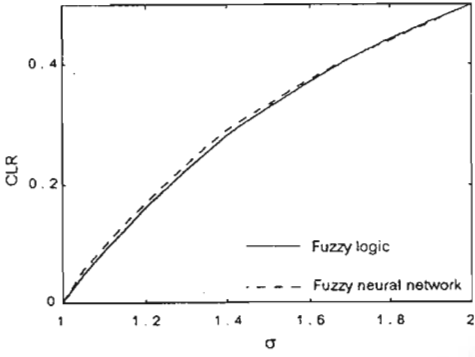
(ข) แบ่งเป็น 3 ชุด โดย $R_{i1} \in \{11.0, 13.0, \dots, 33.0\}$, $R_{i2} \in \{35.0, 37.0, \dots, 57.0\}$ และ $R_{i3} \in \{59.0, 61.0, \dots, 83.0\}$

(ค) แบ่งเป็น 4 ชุด โดย $R_{i1} \in \{11.0, 13.0, \dots, 27.0\}$, $R_{i2} \in \{29.0, 31.0, \dots, 45.0\}$, $R_{i3} \in \{47.0, 49.0, \dots, 63.0\}$ และ $R_{i4} \in \{65.0, 67.0, \dots, 83.0\}$

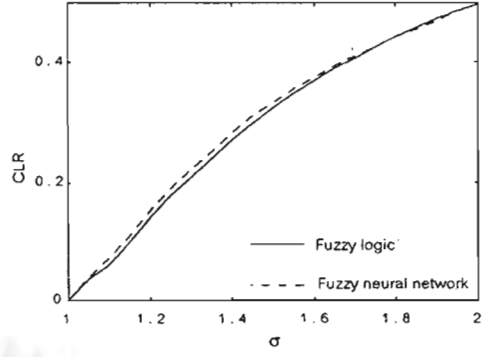
(ง) แบ่งเป็น 5 ชุด โดย $R_{i1} \in \{11.0, 13.0, \dots, 23.0\}$, $R_{i2} \in \{25.0, 27.0, \dots, 37.0\}$,

$R_{i3} \in \{39.0, 41.0, \dots, 51.0\}$, $R_{i4} \in \{53.0, 55.0, \dots, 67.0\}$ และ $R_{i5} \in \{69.0, 71.0, \dots, 83.0\}$

เราฝึกนิวรอลเน็ตเวิร์กที่มีจำนวนฮิดเดนโนดที่ได้จากการทดสอบด้วยข้อมูลในแต่ละชุด จากนั้นนำค่าถ่วงน้ำหนักสุดท้ายที่ได้ไปทดสอบความสามารถในการตรวจวัดไหลตเกิน โดยผลการทดสอบแต่ละกรณีแสดงไว้ดังรูปที่ 5.37 - 5.40 และค่าผิดพลาดเมื่อเปรียบเทียบผลจากตัวตรวจทราฟฟิกที่ใช้ตัวควบคุมพีซีกับนิวรอลเน็ตเวิร์กกับผลจากตัวตรวจทราฟฟิกที่ใช้ตัวควบคุมพีซีลอจิกแสดงไว้ดังตารางที่ 5.4



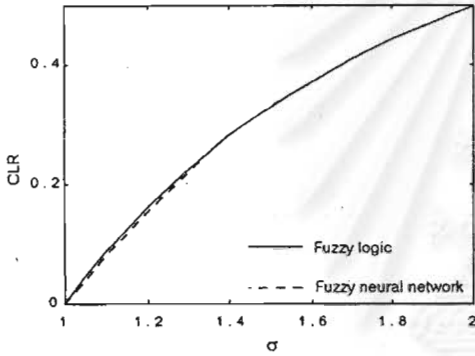
(ก)



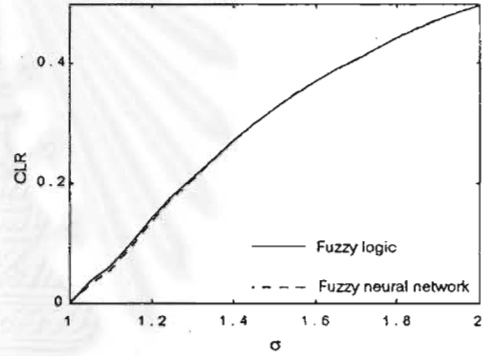
(ข)

รูปที่ 5.37 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 2 ชุด

(ก) เมื่อแปร E(X) ของแหล่งกำเนิด (ข) เมื่อแปร E(S) ของแหล่งกำเนิด



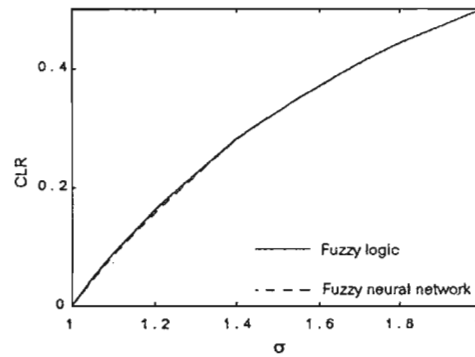
(ก)



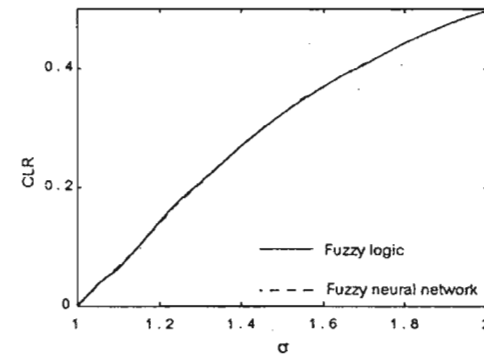
(ข)

รูปที่ 5.38 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 3 ชุด

(ก) เมื่อแปร E(X) ของแหล่งกำเนิด (ข) เมื่อแปร E(S) ของแหล่งกำเนิด



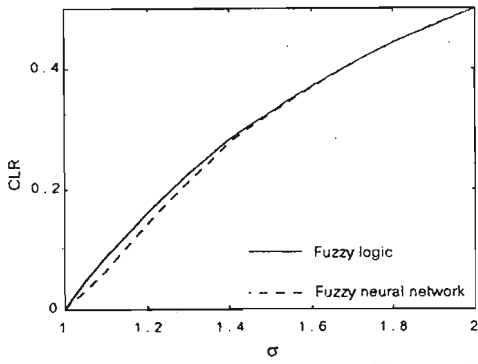
(ก)



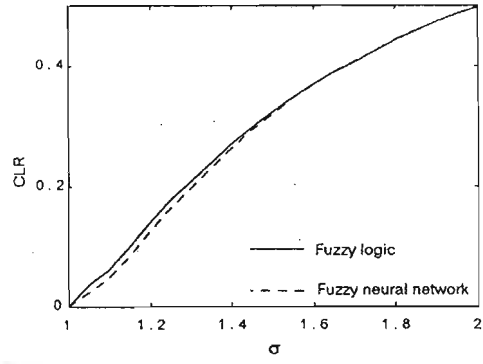
(ข)

รูปที่ 5.39 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 4 ชุด

(ก) เมื่อแปร E(X) ของแหล่งกำเนิด (ข) เมื่อแปร E(S) ของแหล่งกำเนิด



(ก)



(ข)

รูปที่ 5.40 ความสามารถในการตรวจวัดโหลดเกินเมื่อแบ่งชุดฝึกเป็น 5 ชุด

(ก) เมื่อแปร $E(X)$ ของแหล่งกำเนิด (ข) เมื่อแปร $E(S)$ ของแหล่งกำเนิด

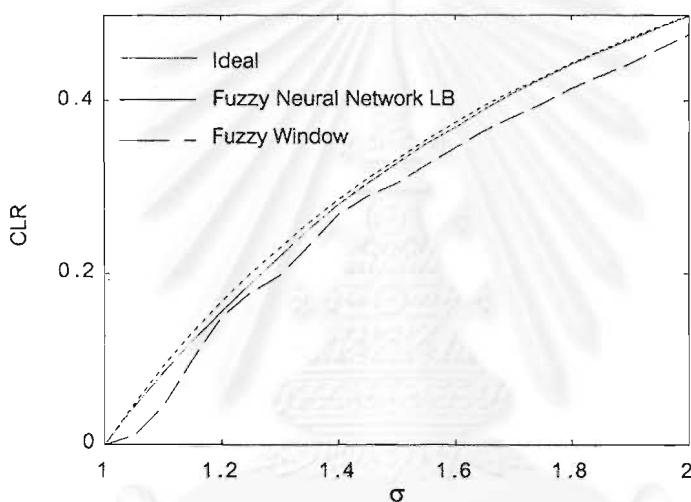
ตารางที่ 5.4 เปรียบเทียบค่าผิดพลาดในการฝึกสำหรับการแบ่งชุดฝึกเป็น 2, 3, 4 และ 5 ชุด

| จำนวนชุดของนิวรอลเน็ตเวิร์ก | | 2 | 3 | 4 | 5 |
|-----------------------------|---------------|-------|------|------|------|
| ค่าผิดพลาด (%) | แปรค่า $E(X)$ | 10.64 | 2.41 | 1.79 | 8.69 |
| | แปรค่า $E(S)$ | 11.54 | 2.44 | 1.83 | 8.34 |

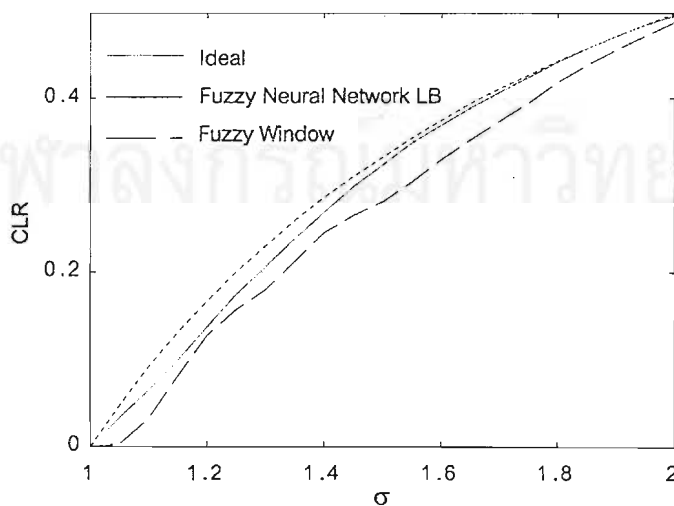
จากผลการทดสอบ พบว่าการแบ่งกลุ่มมีผลต่อประสิทธิภาพในการฝึก แต่จำนวนกลุ่มที่แบ่งไม่ได้สัมพันธ์กับค่าผิดพลาด ดังจะเห็นได้ว่าการฝึกโดยแบ่งชุดฝึกเป็น 4 ชุด ทำให้นิวรอลเน็ตเวิร์กเรียนรู้ได้ดีกว่าการแบ่งเป็น 5 ชุด สรุปได้ว่าปัจจัยสำคัญที่มีผลต่อการเรียนรู้ของนิวรอลเน็ตเวิร์กนอกเหนือจากจำนวนข้อมูลที่ฝึก คือแบบรูป (pattern) ของข้อมูลที่ป้อนให้กับนิวรอลเน็ตเวิร์ก ข้อสังเกตจากการแบ่งชุดฝึกเป็น 4 ชุดทำให้ค่าผิดพลาดต่ำที่สุดอาจเป็นเพราะแบบรูปของเอาต์พุตที่ได้จากอินพุตในแต่ละชุดเป็นไปในทิศทางเดียวกัน จากกฎพื้นฐานฟัชซีในตารางที่ 3.4 กำหนดให้ R_i ที่เป็นสมาชิกของฟัชซีเซต ZE-PS ส่วนใหญ่มีแนวโน้มที่จะให้เอาต์พุต R_{i+1} เปลี่ยนแปลงไปจากเดิมในช่วงแคบๆ ซึ่งยังคงอยู่ระหว่าง ZE-PS เช่นเดิม ซึ่ง R_i ที่มีค่าฟัชซีอยู่ในช่วงดังกล่าว ตรงกับกลุ่มที่ 2 ($R_i \in \{29.0, 31.0, \dots, 45.0\}$) พอดี เช่นเดียวกับอินพุตในกลุ่มที่ 1 ที่อยู่ในช่วง NB-NS ซึ่งมีแนวโน้มที่จะให้เอาต์พุตที่เพิ่มขึ้น และอินพุตในกลุ่มที่ 3 และ 4 ที่อยู่ในช่วง PM และ PB ซึ่งมีแนวโน้มที่จะให้เอาต์พุตลดลง ดังนั้นในกรณีที่มีข้อมูลที่จะต้องฝึกจำนวนมาก การจัดกลุ่มอย่างเหมาะสมนอกจากจะช่วยให้ข้อมูลลดลง ยังทำให้การเรียนรู้มีประสิทธิภาพยิ่งขึ้นด้วย

5.4.3 ผลการทดสอบเปรียบเทียบความสามารถในการเลือก

จากผลการทดสอบหาชุดค่าถ่วงน้ำหนักที่ให้ค่าผิดพลาดต่ำสุด ได้ว่าการแบ่งชุดฝึกเป็น 4 ชุดแล้วฝึกด้วยนิวรอลเน็ตเวิร์กแบบขนานมีความเหมาะสมที่สุด ดังนั้นใช้ค่าถ่วงน้ำหนักที่ฝึกได้ด้วยวิธีดังกล่าวมาทำหน้าที่ inference engine ในตัวตรวจกราฟฟิกรที่เสนอ และทดสอบเปรียบเทียบความสามารถในการตรวจวัดโหลดเกินและ false alarm probability กับพฤติกรรมอุดมคติ กลไกถังรั่ว และกลไกหน้าต่างพีชชีลลจิก ได้ผลการทดสอบดังแสดงในรูปที่ 5.41-5.42 โดยการเปรียบเทียบ false alarm probability และ RSSE ที่คำนวณได้จากรูปแสดงไว้ดังตารางที่ 5.5



รูปที่ 5.41 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกิน เมื่อแปรค่า $E(X)$ ของแหล่งกำเนิด



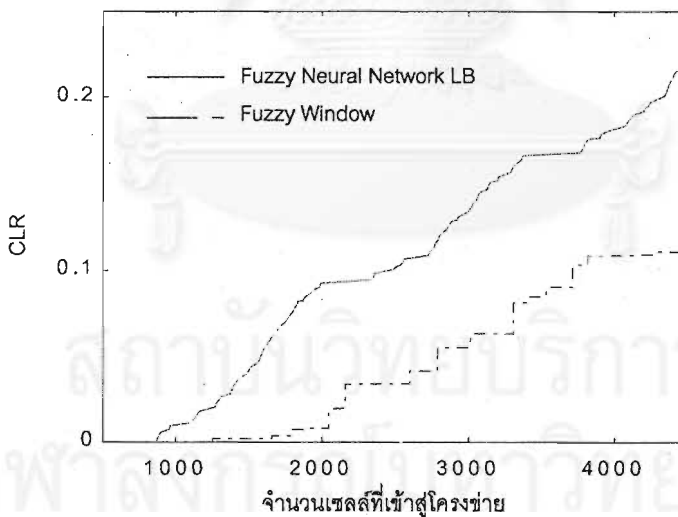
รูปที่ 5.42 เปรียบเทียบความสามารถในการตรวจวัดโหลดเกิน เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

ตารางที่ 5.5 เปรียบเทียบความสามารถในการเลือกของกลไกตรวจหาฟีกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2

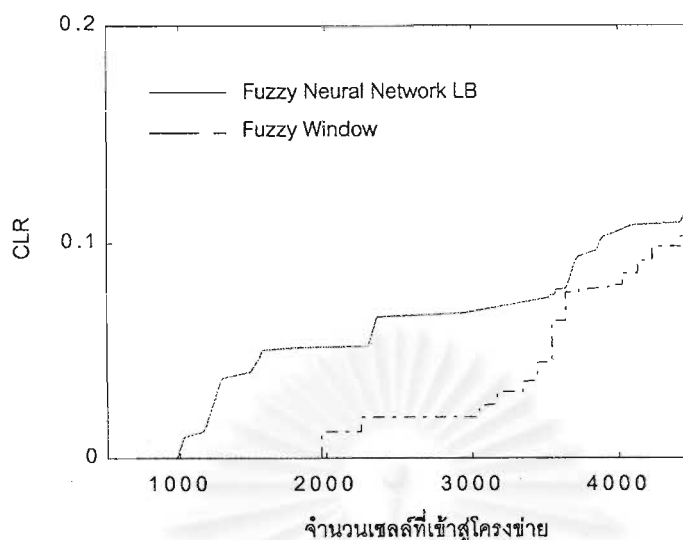
| กลไก | | หน้าต่างฟัซซีลอจิก | ถึงรั้ว ฟัซซีนิวรอลเน็ตเวิร์ก |
|-------------------------|-------------|-----------------------|----------------------------------|
| RSSE | แปรค่า E(X) | 0.130790 | 0.027591 |
| | แปรค่า E(S) | 0.181735 | 0.070226 |
| False Alarm Probability | | 1.14×10^{-8} | 3.69×10^{-7} |

5.4.4 ผลการทดสอบเปรียบเทียบการตอบสนอง

จากการทดสอบผลการตอบสนองของตัวตรวจที่ใช้ฟัซซีนิวรอลเน็ตเวิร์ก และเปรียบเทียบกับหน้าต่างฟัซซีลอจิกที่ $\sigma = 1.50$ ได้ผลการทดสอบแสดงดังรูปที่ 5.43, 5.44



รูปที่ 5.43 เปรียบเทียบการตอบสนองของกลไกตรวจหาฟีก เมื่อแปรค่า E(X) ของแหล่งกำเนิด



รูปที่ 5.44 เปรียบเทียบการตอบสนองของกลไกตรวจทราฟฟิก เมื่อแปรค่า $E(S)$ ของแหล่งกำเนิด

จากรูปที่ 5.43 และ 5.44 การตอบสนองของกลไกตรวจทราฟฟิกโดยใช้ฟuzzyนิวรอลเน็ตเวิร์กดีกว่าหน้าต่างฟuzzyลจิกมาก และหากเปรียบเทียบความเร็วในการตอบสนองเริ่มแรกกับการใช้ฟuzzyลจิกได้ผลดังตารางที่ 5.6

ตารางที่ 5.6 เปรียบเทียบการตอบสนองเริ่มต้นของกลไกถึงรั้วฟuzzyนิวรอลเน็ตเวิร์ก กับถึงรั้วฟuzzyลจิกเมื่อทดสอบกับแบบจำลองแหล่งกำเนิดแบบที่ 2

| ลักษณะการแปรค่า ทราฟฟิกไหล | อัตราการสูญหายของเซลล์ที่ $\sigma = 1.50$ | | จำนวนเซลล์ที่เข้าสู่โครงข่ายก่อน การละทิ้งเซลล์แรกที่ $\sigma = 1.50$ | |
|-------------------------------|--|----------------------------------|--|----------------------------------|
| | ถึงรั้วฟuzzyลจิก | ถึงรั้วฟuzzy นิวรอลเน็ตเวิร์ก | ถึงรั้วฟuzzyลจิก | ถึงรั้วฟuzzy นิวรอลเน็ตเวิร์ก |
| เมื่อแปรค่า $E(X)$ | 0.328589 | 0.328272 | 845 | 869 |
| เมื่อแปรค่า $E(S)$ | 0.324299 | 0.323982 | 962 | 981 |

จากตารางที่ 5.6 เห็นได้ว่า การตอบสนองเริ่มแรกมีความสัมพันธ์กับอัตราการละทิ้งเซลล์ของตัวตรวจทราฟฟิก กล่าวคือถึงรั้วฟuzzyนิวรอลเน็ตเวิร์กมีอัตราการละทิ้งเซลล์ที่ขนาดใหญ่เกิน $\sigma = 1.50$ น้อยกว่าถึงรั้วฟuzzyลจิกและทำให้ปฏิกิริยาการตอบสนองเริ่มต้นต่อทราฟฟิกที่ละเมิดข้อตกลงด้วยขนาดใหญ่มากดังกล่าวข้างต้นแต่ไม่แตกต่างกันมากนัก

5.4.5 ผลการทดสอบเปรียบเทียบเวลาประวิง

จากการทดสอบเวลาประวิงเฉลี่ยของแต่ละเซลล์ในสภาวะโหลดปกติ $\sigma = 1.0$ และเปรียบเทียบกับเมื่อใช้พีชชีลอจิกได้ผลดังตารางที่ 5.7

ตารางที่ 5.7 เปรียบเทียบเวลาประวิงของกลไกตรวจหาฟลักเมื่อใช้ตัวควบคุมต่างกัน

| ตัวควบคุม | พีชชีลอจิก | พีชชีนิวรอลเน็ตเวิร์ก |
|-----------------|------------|-----------------------|
| เวลาประวิง (ms) | 0.595321 | 0.544105 |

จากตารางที่ 5.7 ถังรั้วพีชชีนิวรอลเน็ตเวิร์กทำให้เกิดเวลาประวิงเนื่องจากการรอคอยในบัฟเฟอร์เฉลี่ยต่ำลงเมื่อเปรียบเทียบกับถังรั้วพีชชีลอจิก เพราะ false alarm probability ลดลง หมายถึงอัตราการกำเนิดโทเค็นหรืออีกนัยหนึ่งคืออัตราการอนุญาตให้เซลล์ผ่านเข้าไปในโครงข่ายเพิ่มขึ้นจึงส่งผลดีทำให้เวลาประวิงเฉลี่ยลดลงเล็กน้อย

5.5 ผลการทดสอบเปรียบเทียบเวลาในการประมวลผลของตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กเปรียบเทียบกับตัวควบคุมพีชชีลอจิก

วัตถุประสงค์ที่นำนิวรอลเน็ตเวิร์กมาใช้ในงานวิจัยคือเพื่อลดเวลาการประมวลผลในขั้นตอน inference ของพีชชีลอจิก แต่การฝึกนิวรอลเน็ตเวิร์กต้องใช้เวลาานาน ดังนั้นจึงจำเป็นต้องมีการทดสอบเพื่อพิสูจน์ถึงความคุ้มค่าของการใช้นิวรอลเน็ตเวิร์ก

จากการทดสอบจับเวลาในการประมวลผลของตัวควบคุมพีชชีลอจิกและตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กแต่ละครั้ง จำนวน 100 ครั้ง หาเวลาประมวลผลเฉลี่ย รวมทั้งเปรียบเทียบความแตกต่างของอัตราการกำเนิดโทเค็นที่ได้จากตัวควบคุมทั้งสองชนิดได้ผลดังตารางที่ 5.8

สรุปผลการทดสอบและข้อเสนอแนะ

6.1 สรุปผลการทดสอบ

ในการทดสอบแบบจำลองแหล่งกำเนิดทราฟฟิกประเภท packet voice 2 แบบ ที่มีระยะเวลาเบิรสต์ต่างกัน พบว่ากลไกตรวจทราฟฟิกโดยใช้ตัวควบคุมพีชชีนิวรอลเน็ตเวิร์กที่เสนอมีประสิทธิภาพในแง่การตรวจวัดโหลดเกินและการตอบสนองดีกว่ากลไกถังร้วแบบดั้งเดิมและกลไกหน้าต่างพีชชีลอจิก ยกเว้นในบางกรณี เช่นเมื่อทดสอบกับทราฟฟิกที่ละเมิดอัตราเซลล์เฉลี่ยด้วยการเพิ่มระยะเวลาเบิรสต์ (E(S)) ในแบบจำลองแหล่งกำเนิดแบบที่ 1 ที่กลไกมีการตอบสนองเริ่มต้นช้ากว่าถังร้วเล็กน้อย อย่างไรก็ตามก็ดี อัตราการละทิ้งเซลล์ต่อทราฟฟิกที่ละเมิดข้อตกลงในระยะยาวจะเพิ่มขึ้นเร็วกว่าทุกๆ กลไก นอกจากนี้ในการทดสอบกับการละเมิด 2 ลักษณะ คือ เพิ่มระยะเวลาเบิรสต์ กับ ลดระยะเวลาเงียบ กลไกมีปฏิกิริยาต่อการละเมิดทั้ง 2 ลักษณะใกล้เคียงกัน ในขณะที่ในกลไกถังร้วจะตอบสนองต่อการลดระยะเวลาเงียบได้ช้ากว่าการเพิ่มระยะเวลาเบิรสต์มาก

อย่างไรก็ดี แม้ว่าจะใช้พีชชีลอจิกช่วยในการตัดสินใจก็ตาม เราไม่สามารถหลีกเลี่ยงข้อจำกัดของกลไกในการแยกแยะสภาวะโหลดเกินเล็กน้อยจากสภาวะโหลดปกติอย่างเด็ดขาดได้ ดังนั้นการออกแบบให้กลไกมีพฤติกรรมที่เข้มงวดต่อการกระทำผิด จึงทำให้ false alarm probability หรือ CLR ที่โหลดปกติสูงกว่าหน้าต่างพีชชีลอจิก 1 ระดับ (10^1) แต่ยังคงต่ำกว่าถังร้วแบบดั้งเดิม 2 ระดับ (10^2) ดังนั้นกลไกถังร้วพีชชีนิวรอลเน็ตเวิร์กที่เสนอจึงเหมาะกับโครงข่ายที่มีความเสี่ยงที่จะมีการเกิดความคับคั่งและต้องการระดับความเข้มงวดในการตรวจทราฟฟิกสูง นอกจากนี้ โครงสร้างของกลไกมีบัฟเฟอร์ด้านเข้า เพื่อประโยชน์ในการลด CLR และใช้ประเมินลักษณะทราฟฟิกด้านเข้าจากจำนวนและการเปลี่ยนแปลงของจำนวนเซลล์ในบัฟเฟอร์ แต่ปัญหาที่เกิดขึ้นตามมาคือเวลาประวิง จากการทดสอบพบว่ากลไกตรวจทราฟฟิกทำให้เกิดเวลาประวิงกับทราฟฟิกในระดับไม่เกิน 1 ms และสามารถลดเวลาประวิงดังกล่าวได้ด้วยการเพิ่มขนาดบ่อโทเคินซึ่งแลกด้วยการตอบสนองและความสามารถในการตรวจวัดโหลดเกินลดลง

ในการฝึกนิวรอลเน็ตเวิร์กเพื่อทำหน้าที่จับคู่อินพุตกับเอาต์พุตของ inference engine พบว่าการเลือกชุดฝึกและจำนวนข้อมูลที่ฝึกมีผลต่อสมรรถนะการเรียนรู้ของนิวรอลเน็ตเวิร์ก และเราไม่สามารถระบุได้แน่ชัดว่าวิธีใดที่จะทำให้ค่าผิดพลาดในการฝึกต่ำที่สุด จำนวนข้อมูลน้อยเกินไปไม่สามารถทำให้นิวรอลเน็ตเวิร์กเรียนรู้ได้อย่างมีประสิทธิภาพ และหากจำนวนข้อมูลมากเกินไปก็ไม่สามารถฝึกนิวรอลเน็ตเวิร์กเพียงชุดเดียวให้เรียนรู้ข้อมูลทั้งหมดได้ เนื่องจากจำนวนค่าถ่วงน้ำหนัก

ตารางที่ 5.8 เปรียบเทียบเวลาในการประมวลผลเฉลี่ยต่อครั้งของตัวควบคุมฟuzzy ลอจิกกับตัวควบคุมฟuzzy นีวรอลเน็ตเวิร์ก

| ตัวควบคุม | แบบจำลองแหล่งกำเนิดแบบที่ 1 | แบบจำลองแหล่งกำเนิดแบบที่ 2 |
|---|-----------------------------|-----------------------------|
| ฟuzzy ลอจิก | 0.0980 sec. | 0.0977 sec. |
| ฟuzzy นีวรอลเน็ตเวิร์ก | 0.0553 sec. | 0.0543 sec. |
| ความแตกต่างของผล ลัพธ์ที่ได้จากตัวควบคุม | 2.17% | 1.85% |

จากตารางที่ 5.8 การใช้ฟuzzy นีวรอลเน็ตเวิร์กช่วยให้เวลาประมวลผลลดลง 44% เมื่อเปรียบเทียบกับการใช้ฟuzzy ลอจิก โดยคำนวณจากสมการที่ (6.1)

$$\text{ความแตกต่างของเวลาประมวลผล (\%)} = \left(\frac{t_{FNN} - t_{FZ}}{t_{FZ}} \right) \times 100 \quad (6.1)$$

เมื่อ t_{FNN} และ t_{FZ} คือเวลาในการประมวลผลของตัวควบคุมฟuzzy นีวรอลเน็ตเวิร์กและตัวควบคุมฟuzzy ลอจิกตามลำดับ ในขณะที่อัตราการทำเนิดโทเค้นที่ได้ไม่แตกต่างกันมากดังผลแสดงในตารางที่ 5.8 คำนวณจากสมการ (6.2)

$$\text{ความแตกต่างของอัตราการทำเนิดโทเค้น (\%)} = \left(\frac{R_{FNN} - R_{FZ}}{R_{FZ}} \right) \times 100 \quad (6.2)$$

เมื่อ R_{FNN} และ R_{FZ} คืออัตราการทำเนิดโทเค้นที่ได้จากตัวควบคุมฟuzzy นีวรอลเน็ตเวิร์กและตัวควบคุมฟuzzy ลอจิกตามลำดับ และการเพิ่ม preprocessor ในนีวรอลเน็ตเวิร์กแบบขนานทำให้เวลาในการประมวลผลเพิ่มขึ้นน้อยมากเมื่อเทียบกับเวลาการประมวลผลอื่นๆ ในตัวควบคุมฟuzzy นีวรอลเน็ตเวิร์กซึ่งเวลาประมวลผลรวมขึ้นกับจำนวนในดของนีวรอลเน็ตเวิร์กมากกว่า

หน้าที่ทำหน้าที่เก็บความเชื่อมโยงระหว่างอินพุตและเอาต์พุตซึ่งขึ้นกับจำนวนโนดในแต่ละชั้นถูกกำหนดให้มีจำนวนเหมาะสมค่าหนึ่งจำกัด การแก้ปัญหาจึงใช้นิวรอลเน็ตเวิร์กที่มีโครงสร้างแบบขนานและแบ่งชุดฝึกให้กับนิวรอลเน็ตเวิร์กในแต่ละชุด การฝึกนิวรอลเน็ตเวิร์กจนได้ค่าผิดพลาดต่ำเพียงพอ จะทำให้นิวรอลเน็ตเวิร์กทำหน้าที่แทน inference engine ได้ใกล้เคียงมาก และไม่มีผลกระทบต่อประสิทธิภาพในแง่ต่างๆ ของตัวตรวจแต่อย่างใด

6.2 ข้อดีและข้อเสียของตัวตรวจจั้งรั้วโดยใช้พีชชีนิวรอลเน็ตเวิร์ก

สรุปข้อดีและข้อเสียของการใช้กลไกจั้งรั้วพีชชีนิวรอลเน็ตเวิร์กเพื่อทำหน้าที่ตรวจทราฟฟิกได้ดังนี้

ข้อดี

1. ปรับปรุงประสิทธิภาพในแง่การตรวจวัดโหลดเกินและการตอบสนองให้ดีขึ้น เมื่อเปรียบเทียบกับกลไกจั้งรั้วที่ใช้กันอยู่ในปัจจุบัน และกลไกหน้าต่างพีชชีลोजิก ซึ่งเป็นแนวคิดที่เสนอให้ใช้ตัวควบคุมพีชชีลोजิกอีกวิธีหนึ่ง
2. ทำให้ false alarm probability ลดลงเมื่อเปรียบเทียบกับกลไกจั้งรั้ว
3. การใช้นิวรอลเน็ตเวิร์กแทน inference engine จะช่วยให้ประมวลผลเร็วขึ้น

ข้อเสีย

1. ในการออกแบบและทดสอบ เราใช้แบบจำลองแหล่งกำเนิดทราฟฟิกประเภท packet voice 2 แบบ แม้ว่าจะมีระยะเวลาเบิสต์ต่างกัน แต่มี burstiness เท่ากัน ดังนั้นหากทดสอบหรือนำไปใช้งานกับทราฟฟิกประเภทอื่น เช่น ภาพ หรือ วิดีโอ ซึ่งมี burstiness มากขึ้น จะต้องมีการออกแบบตัวควบคุมพีชชีลोजิกใหม่ เพื่อให้เหมาะสมกับทราฟฟิกประเภทนั้น
2. การฝึกนิวรอลเน็ตเวิร์กใช้เวลานาน และอาจต้องทดสอบหลายวิธีเพื่อให้ได้วิธีที่เหมาะสมที่สุด

6.3 ข้อเสนอแนะ

วิทยานิพนธ์นี้เป็นการเสนอแนวทางใหม่ในการนำพีชชีนิวรอลเน็ตเวิร์กมาใช้กับตัวตรวจทราฟฟิกของแหล่งกำเนิดเดี่ยว โดยทดสอบกับทราฟฟิกประเภท packet voice เท่านั้น แต่ในโครงข่ายเอทีเอ็มจริงมีทราฟฟิกอื่นๆ อีกหลายประเภท ซึ่งไม่สามารถใช้ตัวตรวจที่ออกแบบนี้ร่วมกันได้ จึงจะต้องออกแบบโครงสร้างของกลไก อาทิ ขนาดของบัฟเฟอร์และบ่อโทเค็น รวมถึงฟังก์ชันสมาชิกและกฎพื้นฐานพีชชีใหม่ให้เหมาะสมกับทราฟฟิกประเภทนั้นอีกครั้ง อย่างไรก็ตาม ผลการทดสอบได้แสดงถึงประสิทธิภาพและความเป็นไปได้ในการประยุกต์ใช้พีชชีนิวรอลเน็ตเวิร์กเพื่อ

วัตถุประสงค์ในการตรวจภาพฟิสิกที่มีลักษณะแบบเบเวสท์ ซึ่งอาจนำไปใช้กับภาพฟิสิกประเภทอื่นๆ เช่นภาพนิ่ง หรือวิดีโอได้ด้วยเช่นกัน



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. E. P. Rathgeb. Modeling and Performance Comparison of Policing Mechanisms for ATM Networks. IEEE J. Select. Areas in Commun., Vol. 9, No. 3, pp. 325-334, April 1991.
2. M. Butto, E. Cavallero and A. Tonietti. Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks. IEEE J. Select. Areas Commun., Vol. 9, No. 3, pp. 335-342, April 1991.
3. A. A. Tarraf, I. W. Habib and T. N. Saadawi. A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks. IEEE J. Select. Areas Commun., Vol. 12, No. 6, pp. 1088-1096, Aug. 1994.
4. T. D. Ndousse. Fuzzy Neural Control of Voice Cells in ATM Networks. IEEE J. Select. Areas Commun., Vol. 12, No. 9, pp. 1488-1494, Dec. 1994.
5. V. Catania, G. Ficili, S. Palazzo and D. Panno. A Fuzzy Expert System for Usage Parameter Control in ATM Networks. IEEE GLOBECOM '95, Vol. 2, pp. 1338-1342, 1995.
6. V. Catania, G. Ficili, S. Palazzo and D. Panno. A Comparative Analysis of Fuzzy versus Conventional Policing Mechanism for ATM Networks. IEEE/ACM Trans. Networking, Vol. 4, pp. 449-459, 1996.
7. A. Kwok and R. McLeod. ATM Congestion Control using a Fuzzy Neural Network. Canadian Conference on Electrical and Computer Engineering, Vol. 2, pp. 814-817, 1996.
8. T. A. Runkler and Z. E. Glesner. Efficient Algorithms for High Resolution Fuzzy Controllers with Piecewise Linearities. Proc. 3rd IEEE Conf. Fuzzy System, Vol. 1, pp. 189-191, 1994.
9. J. Laongmal and W. Benjapolakul. Aggregate Bandwidth Allocation of Heterogeneous Sources in ATM Networks. IEEE Symposium on Intelligent Signal Processing and Communication Systems, pp. 215-218, Dec. 1999.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

NOVEL SCHEME FOR POLICING MECHANISM IN ATM NETWORKS : FEEDBACK FUZZY LEAKY BUCKET

A. Niruntasukrat and W. Benjapolakul

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

Phone: +66-2-218-6482, Fax: +66-2-251-8991, E-Mail: watit@ee.eng.chula.ac.th

ABSTRACT

This paper presents a feedback fuzzy leaky bucket algorithm for Usage Parameter Control (UPC) in Asynchronous Transfer Mode (ATM) networks. By comparison with the conventional methods, leaky bucket (LB) and fuzzy window, the proposed policer has better responsiveness behavior, and selectivity performance approaches closer to the ideal violation curve than the other two methods. Therefore, the bandwidth can be used more efficiently, and more precise traffic control can be achieved.

1. INTRODUCTION

Asynchronous Transfer Mode (ATM) has been accepted as the transfer and switching techniques for the future B-ISDN networks. The ATM solutions can support various types of services (data, voice, still pictures, video, etc.). All information is packetized into fixed length packets, called cell, and transferred by the use of statistical multiplexing without flow control between the user and the network. To avoid the network congestion, ATM network utilizes two major traffic control mechanisms, Call Admission Control (CAC) and Usage Parameter Control (UPC). CAC, active in the call setup phase, decides whether to accept a new connection, depending on whether the required Quality of Service (QoS) can be guaranteed for both the existing connections and the new one. Depending on the negotiated parameters, CAC can prevent the network from congestion only if the user does not violate the traffic contract. Therefore, UPC or policing function is necessary to be implemented at User-to-Network Interface (UNI) point to monitor characteristics of traffics entering into network. The policing mechanism must operate on all connections during the entire call process, as well as, fulfill the following basic requirements: high selectivity, high responsiveness, simple implementation and resource effectiveness.

In spite of its simplicity, definitely, the use of the peak rate policing wastes more network bandwidth, compared to the use of mean rate policing. Therefore, many literatures turned to find the way to police traffics with the closest performance to the ideal mean cell rate policing in order to achieve the most efficient usage of network resources.

Several mechanisms were proposed to cope with burstiness characteristics of ATM sources, and can be categorized into two conventional groups, Leaky Bucket (LB) mechanisms and Window mechanisms. In [1], it was proved that leaky bucket had the most satisfied performance in both responsiveness and selectivity. Butto, *et al.* [2], derived the equation to approximate the Cell Loss Ratio (CLR) of leaky bucket, and came to the conclusion that LB cannot police mean cell rate and burst duration of traffic efficiently while keeping the CLR low. It

needs to be pointed out that in conventional way, to achieve both selectivity and responsiveness simultaneously is not possible.

A novel scheme of fuzzy logic approach is presented in this paper. Based on token leaky bucket structure, the proposed fuzzy policer performs high selectivity performance with satisfied responsiveness.

2. RELATED WORKS

Of all the policing mechanisms proposed, Leaky Bucket (LB) is the most popular. Despite the simplicity, it is not suited for policing burstiness traffic due to high CLR and non-flexible performance. Since it is not possible for policer to obtain complete statistics of the input traffic, in recent years, many literatures turn to one of the alternative solutions, Artificial Intelligence (AI), to solve non-flexible control problem [3,4,5,6]. Tarraf, *et al.* [3] proposed to use two artificial Neural Networks (NN) for UPC. Ndousse [4] presented a fuzzy policing mechanism called virtual Leaky Bucket. In this scheme, fuzzy logic was applied to control the number of tagged cells in addition to normal cells. However, the study seems to provide only throughput performance, but no evidences related to selectivity or responsiveness were given. Catania, *et al.*, [5,6] proposed the fuzzy expert system based on window scheme. The policer rejected any cells arriving in each window that is greater than the allowed window limit, which is dynamically updated by inference rules based on fuzzy logic. Although the performance of this mechanism was excellent in terms of selectivity, it took control action rather slowly compared to the conventional LB.

1. SOURCE MODEL AND PROPOSED FEEDBACK FUZZY POLICER

In this paper, the on-off or bursty source model of packetized voice with a peak bit rate of 32 kbps shown in Fig. 1 is used. The number of cells in burst period, $E(X)$, and the duration of silence period, $E(S)$, are geometrically distributed and negative exponentially distributed with mean values of 5 cells and 147.72 ms, respectively.

The proposed feedback fuzzy leaky bucket model is shown in Fig. 2. Fuzzy controller is employed for controlling the token generation rate. The fuzzy controller has four input linguistic parameters consisting of the number of cells in input queue buffer (N_c), the change in number of cells in input queue buffer (ΔN_c), the number of tokens in token pool (N_t) and the feedback data which is the token generation rate in the previous sampling period (R_t).

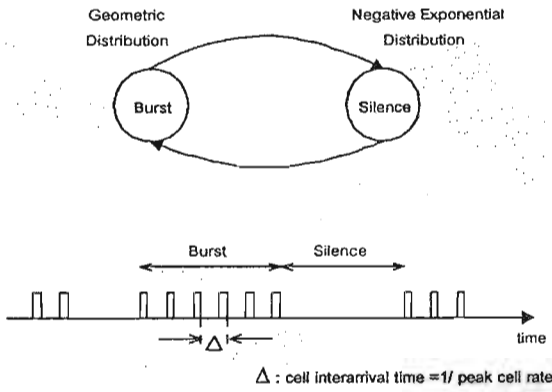


Figure 1. On-off source model

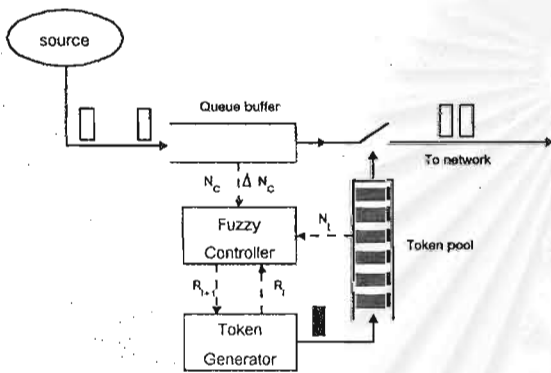


Figure 2. Feedback fuzzy policer

The membership functions of all variables, derived by adhoc method, are shown in Fig. 3. All parameters, except token generation rate, are fuzzified into 5 fuzzy set members consisting of Negative Big (NB), Negative Small (NS), Zero Equal (ZE), Positive Small (PS) and Positive Big (PB). However, two fuzzy set members of Negative Medium (NM) and Positive Medium (PM) are added for token generation rate since token generation rate is not only the input of fuzzy controller, but also the output. The more fuzzy set members are assigned, the more precise and smooth output can be gained. The fuzzy rule base is given in Table 1. The MAX-MIN inference method is used. The defuzzification technique is fuzzy centroid method, and the piecewise linearities algorithm [7] is employed for reducing the runtime and memory consumed in defuzzification process.

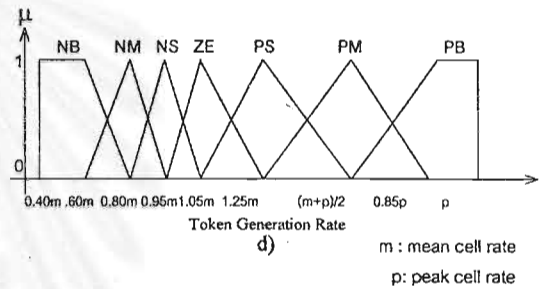
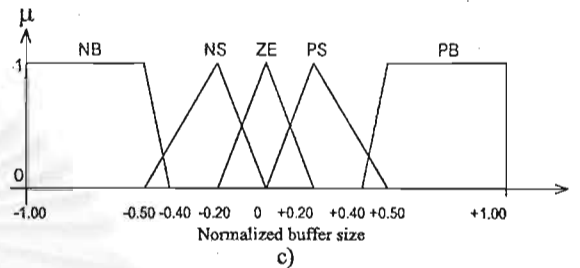
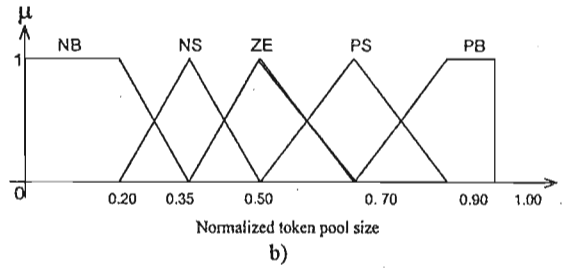
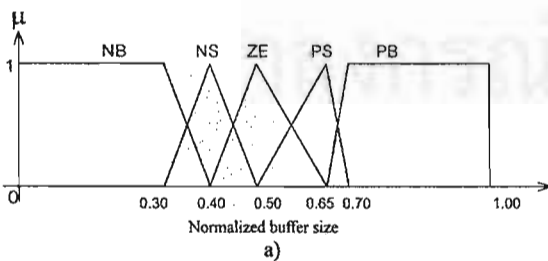


Figure 3. Membership functions of a) no. of cells in buffer (N_c) b) no. of tokens in pool (N_i) c) change in no. of cells in buffer (ΔN_c) d) token generation rate (R_i, R_{i+1})

Table 1 Fuzzy Rule Base

| No. | N_c | ΔN_c | N_i | R_i | R_{i+1} | No. | N_c | ΔN_c | N_i | R_i | R_{i+1} |
|-----|-------|--------------|-------|-------|-----------|-----|-------|--------------|-------|-------|-----------|
| 1 | NB | NB | NB | NS | PB | 24 | NB | NS | NB | PB | NS |
| 2 | NB | NB | NB | ZE | PM | 25 | NB | NS | NS | NS | PS |
| 3 | NB | NB | NB | PS | PS | 26 | NB | NS | NS | ZE | PS |
| 4 | NB | NB | NB | PM | PS | 27 | NB | NS | NS | PS | ZE |
| 5 | NB | NB | NB | PB | PB | 28 | NB | NS | NS | PM | PS |
| 6 | NB | NB | NS | NS | PM | 29 | NB | NS | NS | PB | PS |
| 7 | NB | NB | NS | ZE | PS | 30 | NB | NS | NS | PS | NS |
| 8 | NB | NB | NS | PS | ZE | 31 | NB | NS | ZE | PM | ZE |
| 9 | NB | NB | NS | PM | ZE | 32 | NB | NS | ZE | PB | PS |
| 10 | NB | NB | NS | PB | PS | 33 | NB | NS | PS | PM | NS |
| 11 | NB | NB | ZE | ZE | PS | 34 | NB | NS | PS | PB | NS |
| 12 | NB | NB | ZE | PS | ZE | 35 | NB | NS | PB | PM | NM |
| 13 | NB | NB | ZE | PM | ZE | 36 | NB | NS | PB | PB | NM |
| 14 | NB | NB | ZE | PB | ZE | 37 | NB | ZE | NB | NS | ZE |
| 15 | NB | NB | PS | PS | ZE | 38 | NB | ZE | NB | ZE | PS |
| 16 | NB | NB | PS | PM | NS | 39 | NB | ZE | NB | PS | ZE |
| 17 | NB | NB | PS | PB | NS | 40 | NB | ZE | NB | PM | PS |
| 18 | NB | NB | PS | PM | NM | 41 | NB | ZE | NS | NS | PM |
| 19 | NB | NB | PS | PB | NM | 42 | NB | ZE | NS | ZE | PS |
| 20 | NB | NS | NB | NS | ZE | 43 | NB | ZE | NS | PS | ZE |
| 21 | NB | NS | NB | ZE | PS | 44 | NB | ZE | NS | PM | PS |
| 22 | NB | NS | NB | PS | PS | 45 | NB | ZE | NS | PB | NS |
| 23 | NB | NS | NB | PM | ZE | 46 | NB | ZE | ZE | ZE | PM |

| No. | N_c | ΔN_c | N_i | R_i | $R_{c,i}$ | No. | N_c | ΔN_c | N_i | R_i | $R_{c,i}$ |
|-----|-------|--------------|-------|-------|-----------|-----|-------|--------------|-------|-------|-----------|
| 47 | NB | ZE | ZE | PS | NS | 105 | ZE | PS | NB | ZE | PS |
| 48 | NB | ZE | ZE | PM | PS | 106 | ZE | PS | NB | PS | ZE |
| 49 | NB | ZE | ZE | PB | ZE | 107 | ZE | PS | NB | PM | ZE |
| 50 | NB | ZE | PS | PS | ZE | 108 | ZE | PS | NB | PB | NS |
| 51 | NB | ZE | PS | PM | ZE | 109 | ZE | PB | NB | NM | PS |
| 52 | NB | ZE | PS | PB | NS | 110 | ZE | PB | NB | NS | PS |
| 53 | NB | ZE | PB | PM | NS | 111 | ZE | PB | NB | ZE | PS |
| 54 | NB | ZE | PB | PB | NM | 112 | ZE | PB | NB | PS | NS |
| 55 | NB | PS | NB | NB | NM | 113 | PS | NB | NB | NS | PB |
| 56 | NB | PS | NB | NM | NS | 114 | PS | NB | NB | ZE | PM |
| 57 | NB | PS | NB | NS | ZE | 115 | PS | NB | NB | PS | PM |
| 58 | NB | PS | NB | ZE | PS | 116 | PS | NB | NB | PM | PS |
| 59 | NB | PS | NB | PS | ZE | 117 | PS | NB | NB | PB | ZE |
| 60 | NS | NB | NB | NS | PM | 118 | PS | NS | NB | NS | PM |
| 61 | NS | NB | NB | ZE | PS | 119 | PS | NS | NB | ZE | PS |
| 62 | NS | NB | NB | PS | PS | 120 | PS | NS | NB | PSd | PS |
| 63 | NS | NB | NB | PM | PS | 121 | PS | NS | NB | PM | ZE |
| 64 | NS | NB | NB | PB | PM | 122 | PS | NS | NB | PB | NS |
| 65 | NS | NS | NB | NS | PS | 123 | PS | ZE | NB | NB | NM |
| 66 | NS | NS | NB | ZE | ZE | 124 | PS | ZE | NB | NM | NS |
| 67 | NS | NS | NB | PS | ZE | 125 | PS | ZE | NB | NS | ZE |
| 68 | NS | NS | NB | PM | PS | 126 | PS | ZE | NB | ZE | PS |
| 69 | NS | NS | NB | PB | PS | 127 | PS | ZE | NB | PS | ZE |
| 70 | NS | ZE | NB | NM | NS | 128 | PS | ZE | NB | PM | NS |
| 71 | NS | ZE | NB | NS | ZE | 129 | PS | ZE | NB | PB | NM |
| 72 | NS | ZE | NB | ZE | PS | 130 | PS | PS | NB | NB | NS |
| 73 | NS | ZE | NB | PS | ZE | 131 | PS | PS | NB | NM | ZE |
| 74 | NS | ZE | NB | PM | ZE | 132 | PS | PS | NB | NS | PS |
| 75 | NS | ZE | NB | PB | NS | 133 | PS | PS | NB | ZE | PS |
| 76 | NS | PS | NB | NB | NS | 134 | PS | PS | NB | PM | NS |
| 77 | NS | PS | NB | NM | ZE | 135 | PS | PS | NB | PB | NM |
| 78 | NS | PS | NB | NS | PS | 136 | PS | PB | NB | NB | ZE |
| 79 | NS | PS | NB | ZE | PS | 137 | PS | PB | NB | NM | PS |
| 80 | NS | PS | NB | PS | ZE | 138 | PS | PB | NB | NS | PS |
| 81 | NS | PS | NB | PM | NS | 139 | PS | PB | NB | ZE | NS |
| 82 | NS | PB | NB | NB | ZE | 140 | PS | PB | NB | PS | NM |
| 83 | NS | PB | NB | NM | PS | 141 | PS | PB | NB | PM | NB |
| 84 | NS | PB | NB | NS | PS | 142 | PS | PB | NB | PB | NB |
| 85 | NS | PB | NB | ZE | PS | 143 | PB | NS | NB | PS | ZE |
| 86 | NS | PB | NB | PS | ZE | 144 | PB | NS | NB | PM | PS |
| 87 | M | NB | NB | NS | PB | 145 | PB | NS | NB | PB | ZE |
| 88 | M | NB | NB | ZE | PM | 146 | PB | ZE | NB | NS | PS |
| 89 | M | NB | NB | PS | ZE | 147 | PB | ZE | NB | PS | ZE |
| 90 | M | NB | NB | PM | PS | 148 | PB | ZE | NB | PM | ZE |
| 91 | M | NS | NB | NS | PS | 149 | PB | ZE | NB | PB | PS |
| 92 | M | NS | NB | ZE | PS | 150 | PB | PS | NB | NB | ZE |
| 93 | M | NS | NB | PS | ZE | 151 | PB | PS | NB | NM | PS |
| 94 | M | NS | NB | PM | ZE | 152 | PB | PS | NB | NS | PS |
| 95 | M | NS | NB | PB | ZE | 153 | PB | PS | NB | ZE | NS |
| 96 | M | ZE | NB | NB | NS | 154 | PB | PS | NB | PS | ZE |
| 97 | M | ZE | NB | NM | ZE | 155 | PB | PS | NB | PM | NS |
| 98 | M | ZE | NB | NS | ZE | 156 | PB | PS | NB | PB | NM |
| 99 | M | ZE | NB | PS | ZE | 157 | PB | PB | NB | NB | NS |
| 100 | M | ZE | NB | PM | PS | 158 | PB | PB | NB | NM | ZE |
| 101 | M | ZE | NB | PB | NS | 159 | PB | PB | NB | ZE | PS |
| 102 | M | PS | NB | NB | NM | 160 | PB | PB | NB | PS | NS |
| 103 | M | PS | NB | NM | NS | 161 | PB | PB | NB | PM | NM |
| 104 | M | PS | NB | NS | ZE | 162 | PB | PB | NB | PB | NB |

4. SIMULATION RESULTS

To evaluate the efficiency of the proposed feedback fuzzy policer, the selectivity and responsiveness performances of the proposed policer are compared with those of the conventional LB and fuzzy window mechanisms [6]. A computer simulation has been conducted on C language, and the results were obtained with a 95% confidence of interval of the true value. The sampling interval is set at 230 ms.

The conventional LB was dimensioned by assuming the overdimension factor $C=1.42$ and counter size = 45, while the fuzzy window [6] was dimensioned by assuming a window size of $T = 1,440$ ms and window limit N is equal to 32 cells. After several times of simulations and data collecting, the feedback fuzzy LB is chosen with a buffer size of 10 and token

pool size of 500 which performs the best. The statistics obtained from every test is accumulated from a population of 10^9 cells.

In selectivity aspects, CLRs of the policers, with the ideal violation probability given by $P=(\sigma-1)/\sigma$ where σ is the long-term normalized mean cell rate of the source, are compared. To compare the CLR in each degree of mean-cell-rate violation, the test is performed by varying mean burst duration, while the mean silence duration is fixed. In Fig. 4, it is obviously shown that the feedback fuzzy policer has the performance curve closest to the ideal behavior among those of other mechanisms when $\sigma > 1$ with an average error of 4.32% compared to 39.09% of the conventional LB policer and 19.16% of the fuzzy window.

For responsiveness aspect, the evaluation, in terms of the number of cells entering the network from the violating source before the policer discards first cell, is performed. The performance comparison is conducted at traffic load $\sigma = 1.50$. As shown in Fig. 5, the feedback fuzzy LB takes action after 328 cells, faster than the fuzzy window which discards first cell at 731 cells, and slower than conventional LB, which discards first cell at 309 cells. But, after a while, the probability of discarded cells increases with the most rapid rate compared to others, while the curve is still below the upper limit of 0.333, in case of ideal behavior. The proposed scheme shows the best behavior especially in taking action after the traffic load exceeds the negotiated rate. Feedback fuzzy LB gives the violation curve closest to the ideal one, and performs the best in dynamic behavioral aspect.

To achieve all of these goals, the proposed feedback fuzzy LB is traded off with higher fault alarm probability of 10^{-7} compared to 10^{-8} of fuzzy window. However, it is still acceptable in packet voice to guarantee the QoS of 10^{-3} CLR, and the fault alarm probability is still lower than 10^{-5} of conventional LB.

In order to evaluate the performance of the feedback fuzzy LB toward a model of real packet voice source [4-6], the source which has the traffic parameters, $E(X) = 29$ cells and $E(S) = 650$ ms at the peak bit rate of 32 kbps is considered. Here, it is necessary to increase the token pool and buffer size to 3,600 tokens and 30 cells, respectively, to cope with the longer burst duration. By comparing with the fuzzy window [6], the selectivity and responsiveness performances are shown in Fig. 6 and Fig. 7. From the results, the feedback fuzzy LB approach is much closer to the ideal curve with an error of 1.3% compared to 14.5% of the fuzzy window. In addition, the proposed policer takes action faster than fuzzy window does, by discarding the first cell after letting 845 cells enter the network while fuzzy window starts to discard cells after 1,245 cells. As a result of higher responsiveness performance, the feedback fuzzy policer yields higher false alarm probability of 4.48×10^{-7} compared to 1.14×10^{-8} of fuzzy window

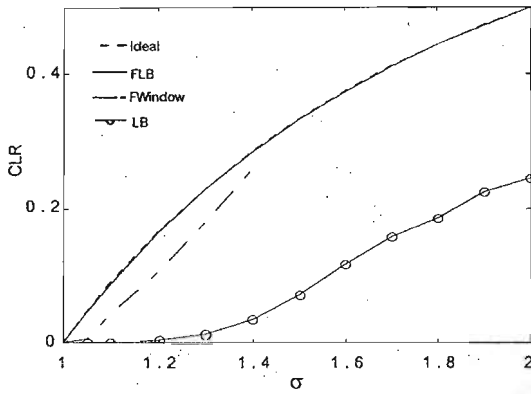


Figure 4. Comparison of selectivity performance

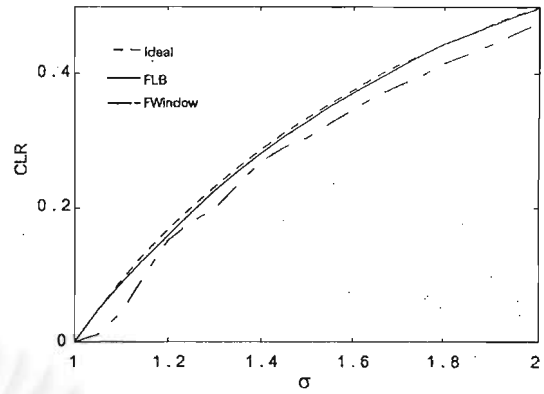


Figure 6. Selectivity performance for a real voice model

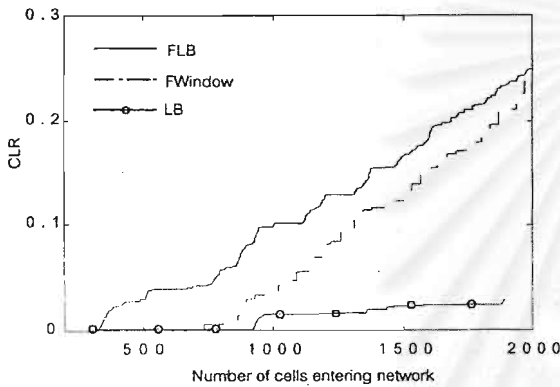


Figure 5. Responsiveness behavior

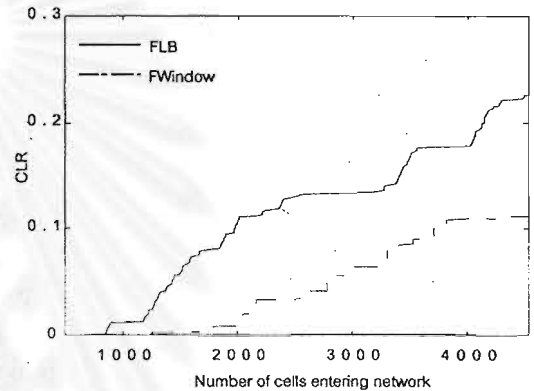


Figure 7. Responsiveness behavior for a real voice model

5. CONCLUSION

In this paper, the feedback fuzzy LB algorithm is proposed to police mean cell rate of traffic for UPC. The proposed scheme is evaluated by comparing its performance with popular conventional LB, and fuzzy window [6]. The results show that the proposed policer achieves the purpose to enhance the responsiveness and overload detection performances. It is proved that the proposed scheme, designed with more complex structure based on LB, can greatly improve overload detection performance. However, to achieve both mentioned purposes, the feedback fuzzy LB trades them off with higher fault alarm probability, therefore it is remarkable that the proposed scheme is suited for heavy-load networks due to its more strict behavior.

REFERENCES

- [1] E. P. Rathgeb. "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks". *IEEE J. Select. Areas in Commun.*, Vol. 9, No. 3, April 1991, pp. 325-334.
- [2] M. Butto, E. Cavallero and A. Tonietti. "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks". *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, April 1991, pp. 335-342.
- [3] A. A. Tarraf, I. W. Habib and T. N. Saadawi. "A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks". *IEEE J. Select. Areas Commun.*, Vol. 12, No. 6, Aug. 1994, pp. 1088-1096.
- [4] T. D. Ndousse. "Fuzzy Neural Control of Voice Cells in ATM Networks". *IEEE J. Select. Areas Commun.*, Vol. 12, No. 9, Dec. 1994, pp. 1488-1494.
- [5] V. Catania, G. Ficili, S. Palazzo and D. Panno. "A Fuzzy Expert System for Usage Parameter Control in ATM Networks". *IEEE GLOBECOM '95*, Vol. 2, 1995, pp. 1338-1342.
- [6] V. Catania, G. Ficili, S. Palazzo and D. Panno. "A Comparative Analysis of Fuzzy Versus Conventional Policing Mechanism for ATM Networks". *IEEE/ACM Trans. Networking*, Vol. 4, pp. 449-459, 1996.
- [7] T. A. Runkler and Z. E. Glesner. "Efficient Algorithms for High Resolution Fuzzy Controllers with Piecewise Linearities". *Proc. 3rd IEEE Conf. Fuzzy System*, Vol. 1, 1994, pp. 189-191.

ประวัติผู้เขียน

นางสาวเอมอัชชา นรินทร์สุวรัตน์ เกิดเมื่อวันที่ 28 สิงหาคม พ.ศ. 2520 ที่กรุงเทพมหานคร
ศึกษาระดับมัธยมศึกษาที่โรงเรียนบดินทรเดชา (สิงห์ สิงหเสนี) ตั้งแต่ พ.ศ. 2532 ถึง พ.ศ. 2537
เข้าศึกษาระดับปริญญาตรีในสาขาวิศวกรรมไฟฟ้า ณ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์-
มหาวิทยาลัย และสำเร็จการศึกษาในปีการศึกษา 2540 จากนั้นศึกษาต่อในหลักสูตรวิศวกรรม-
ศาสตรมหาบัณฑิต สาขาระบบโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย
เมื่อ พ.ศ. 2541



จุฬาลงกรณ์มหาวิทยาลัย