การพัฒนาสวิตซ์แพ็กเกตความเร็วสูงโดยใช้เทคนิคเอนเวโลปที่สามารถรับประกันคุณภาพการให้บริการสำหรับทราฟฟิก 2 ประเภท

นายสมัชชาย ศรีนนท์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ISBN 974-14-2502-3
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF A HIGH-SPEED PACKET SWITCH USING ENVELOPE TECHNIQUE SUPPORTING QUALITY OF SERVICE GUARANTEES FOR TWO TYPE TRAFFIC

Mr. Samatchai Srinon

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic year 2006

ISBN 974-14-2502-3

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาสวิตช์แพ็กเกตความเร็วสูงโดยใช้เทคนิคเอนเวโลปที่สามารถรับประกันคุณภาพการให้บริการสำหรับทราฟฟิก 2 ประเภท
โดย	นายสมัชชาย ศรีนนท์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ

คณะวิศวกรรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประสิทธิ ประพิณมงคลการ)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สุภาวดี อร่ามวิทย์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สมัชชาย ศรีนนท์ : การพัฒนาสวิตช์แพ็กเกตความเร็วสูงโดยใช้เทคนิคเอนเวโลปที่สามารถรับประกันคุณภาพการให้บริการสำหรับทราฟฟิก 2 ประเภท (DEVELOPMENT OF A HIGH-SPEED PACKET SWITCH USING ENVELOPE TECHNIQUE SUPPORTING QUALITY OF SERVICE GUARANTEES FOR TWO TYPE TRAFFIC) อ.ที่ปรึกษา: รศ. ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, 131 หน้า. ISBN 974-14-2502-3

วิทยานิพนธ์ฉบับนี้ นำเสนอการศึกษาสวิตช์แพ็กเกตความเร็วสูงที่สามารถรองรับเซลล์ที่มีระดับความสำคัญต่างกันได้ 2 ระดับ ในการที่จะทำให้สวิตช์สามารถรองรับแพ็กเกตความเร็วสูงได้นั้น ได้นำเทคนิคเอนเวโลปมาใช้เพื่อลดจำนวนครั้งของการสวิตช์ หรือลดความถี่ในการแมตช์เส้นทางระหว่างอินพุตและเอาต์พุตของสวิตช์ เทคนิคเอนเวโลปที่ใช้ในที่นี้มี 2 วิธีคือ การเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว และการเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม แม้ว่าวิธีการทั้งสองดังกล่าวทำให้ค่าเฉลี่ยการประวิงเวลาเพิ่มสูงขึ้น แต่ก็สามารถแก้ไขปัญหาคอขวดได้ ซึ่งทำให้สวิตช์สามารถรองรับอัตราบิตของสายส่งที่เพิ่มสูงขึ้นได้ การที่จะทำให้สวิตช์สามารถรองรับเซลล์ที่มีระดับความสำคัญต่างกัน 2 ระดับได้นั้น วิทยานิพนธ์นี้ได้นำเสนออัลกอริทึมการควบคุม ณ ขั้นตอนต่าง ๆ ของอัลกอริทึม iSLIP แบบดั้งเดิมจำนวนสามวิธี คือ อัลกอริทึมการควบคุมที่ request อัลกอริทึมการควบคุมที่ grant และอัลกอริทึมการควบคุมที่ accept โดยอัลกอริทึมทั้งสามที่เสนอนี้สามารถทำให้เซลล์ทั้ง 2 ระดับได้รับคุณภาพการให้บริการในเชิงของเวลาประวิงได้ตามต้องการโดยอาศัยกลไลการปรับพารามิเตอร์แบบความน่าจะเป็นในการกำหนดโอกาสการเข้าของเซลล์แต่ละระดับที่แตกต่างกัน ที่สำคัญกลไลดังกล่าวยังคงสามารถรับประกันคุณภาพการให้บริการได้แม้ว่าสภาพทราฟฟิกมีการเปลี่ยนแปลงไป

จากผลการทดสอบด้วยโปรแกรมคอมพิวเตอร์พบว่าการเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ให้ค่าเวลาประวิงที่ดีกว่าการเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม เมื่อเพิ่มขนาดความยาวเอนเวโลป พบว่าเวลาประวิงโดยเฉลี่ยก็จะเพิ่มมากขึ้นด้วย อัลกอริทึมทั้งสามที่เสนอสามารถควบคุมคุณภาพการให้บริการได้แม่นยำมากกว่าอัลกอริทึมเดิม คือ อัลกอริทึม prioritized iSLIP ในทุกๆช่วงของสภาพทราฟฟิก

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต.....
สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา 2549

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4570747221 : MAJOR ELECTRICAL ENGINEERING

KEY WORD: INPUT-QUEUED SWITCH / VIRTUAL OUTPUT QUEUE (VOQs) / /QOS / SCHEDULING / ENVELOPE

SAMATCHAI SRINON : DEVELOPMENT OF A HIGH-SPEED PACKET SWITCH USING ENVELOPE TECHNIQUE SUPPORTING QUALITY OF SERVICE GUARANTEES FOR TWO TYPE TRAFFIC. THESIS ADVISOR : ASSOC.PROF.LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., 131 pp. ISBN 974-14-2502-3

This thesis presents a study of high-speed packet switches that are capable of supporting cells with 2 different priority levels. In order to allow switch to operate at high speed, an envelope technique is applied to help reduce the switching rate or the frequency of matching between input and output of switches. Two types of the envelope technique are used, partially filled (PF) envelope and filled (F) envelope. Although these two techniques cause an increase in the average latency, they can resolve the bottleneck problem thereby enabling the switches to support higher bit rate of transmission links. To support cells with 2 priority levels, in this thesis, we present three algorithms that apply additional control mechanism at each step of the conventional iSLIP, namely control at request, control at grant and control at accept. These algorithms allow both cells with different priority to meet their quality of service (QoS) requirement in terms of delay time performance by adjusting the system parameters appropriately so that the chance of cells entering the switches are precisely controlled. More importantly, the introduced mechanism can guarantee QoS even when traffic loads changes.

From the computer simulation result, we found that partially filled (PF) envelope have the average latency better than filled (F) envelope. When we increase envelope size, the average latency will increase too. Moreover the proposed algorithms can control quality of service more precisely than existing prioritized iSLIP algorithm in every traffic condition.

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

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

Academic Year: 2006

กิตติกรรมประกาศ

การที่วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ก็เพราะได้รับความช่วยเหลือจาก รศ. ดร. ลัญฉกร วุฒิสิทธิกุลกิจ อาจารย์ที่ปรึกษา ที่ได้เสนอแนะแนวทางการวิจัย ให้คำปรึกษาในงานวิจัย ช่วยแก้ไขปัญหาต่างๆ ให้ข้อคิดเห็นต่าง ๆ ด้วยดีเสมอมา และขอขอบพระคุณครูบาอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้วิจัย

ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดามารดา ผู้มีพระคุณทุกท่านและครอบครัวของผู้วิจัย ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

ขอขอบคุณ พี่พิสิฐ วนิชชานันท์, กำพล วรดิษฐ์, พี่วรากร ศรีเชวงทรัพย์ พี่เจนจบ วีระพานิชเจริญ พี่อำนาจรัตน์ โสภณธรรมพัฒน์ นายนรินทร์ นครพันธ์ กนกภรณ์ วีสเพ็ญ นิสิตสาขาโทรคมนาคมที่ช่วยเหลือผู้วิจัยในด้านต่างๆและเป็นกำลังใจที่ดีเสมอมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญภาพ..... ฑ

บทที่

1. บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์ 2

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์ 3

1.4 ขั้นตอนและวิธีดำเนินงาน..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 4

2. สถาปัตยกรรมสวิตช์แพ็กเกต

2.1 สวิตซ์แบบรอคิวที่เอาท์พุต..... 6

2.1.1 สวิตช์แบบ SOQ (Single Output-Queued (OQ) Switches)…….. 7

2.1.2 สวิตช์แบบ MOQ (Multiple Output-Queued (MOQ) Switches)……... 7

2.2 สวิตช์แบบรอคิวที่อินพุต (IQ Switches)…….... 7

2.2.1 Multiple Input-Queued (MIQ) Switches..... 8

2.3 สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (CIOQ switches)……. 9

2.4 สวิตช์แบบต่อขนาน (Parallel Packet Switch:PPS)..... 11

3. อัลกอริทึมการจัดลำดับเซลล์ส่งผ่านสวิตซ์

อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์แบบ IQ (Scheduling algorithms for IQ switches) 15

3.1 อัลกอริทึม Parallel Iterative Matching (PIM)……... 16

3.2 อัลกอริทึม Round-Robin Matching (RRM)…….. 18

3.3 อัลกอริทึม iSLIP……... 19

3.4 อัลกอริทึม Prioritized iSLIP..... 22

3.5 อัลกอริทึมการจัดกำหนดการสำหรับการจัดสรรบริการในเซลล์สวิตช์ที่พิจารณา 24

ลำดับความสำคัญของเซลล์ที่นำเสนอ.....

4. เทคนิคเอนเวโลปที่นำเสนอ

4.1 เทคนิคเอนเวโลป..... 29

5. ผลการทดสอบ

5.1 แบบจำลองอัลกอริทึมที่นำเสนอ..... 32

5.2 ผลการทดสอบสมรรถนะของอัลกอริทึมที่นำเสนอ..... 33

5.2.1 อัลกอริทึมการควบคุมที่ request..... 33

5.2.2 อัลกอริทึมการควบคุมที่ grant..... 38

5.2.3 อัลกอริทึมการควบคุมที่ accept..... 42

5.3 การวิเคราะห์อัลกอริทึมที่นำเสนอ..... 47

5.3.1 การวิเคราะห์เชิงสมรรถนะ..... 47

5.3.1.1 สมรรถนะด้านการควบคุมคุณภาพการให้บริการ..... 47

5.3.1.2 สมรรถนะด้านช่วงการใช้งาน..... 51

5.3.1.3 สมรรถนะด้านค่าเวลาประวิง..... 53

5.4 วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์..... 57

5.5 วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองระดับความสำคัญ..... 59

5.5.1 วิธีที่ 1 เลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (Partially-Filled (PF) envelope) 59

5.5.2 วิธีที่ 2 เลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) 61

5.6 ผลการทดสอบสมรรถนะของอัลกอริทึมที่นำเสนอที่มีเทคนิคเอนเวโลป..... 64

5.6.1 อัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)..... 65

5.6.2 อัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)..... 70

5.6.3 อัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)..... 74

5.6.4 อัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)..... 78

5.6.5 อัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) 82

5.6.6 อัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม 86

(Filled (F) envelope)…….
5.7 การวิเคราะห์อัลกอริทึมที่นำเสนอที่มีเทคนิคเอนเวโลป..... 90
5.7.1 สมรรถนะด้านการควบคุมคุณภาพการให้บริการ..... 90
5.7.2 สมรรถนะด้านค่าเวลาประวิง…… 101
6. สรุปผลการวิจัยและข้อเสนอแนะ
6.1 สรุปผลการวิจัย 116
6.2 ข้อเสนอแนะ..... 117

รายการอ้างอิง 118
ภาคผนวก
ผนวก ก ประมาณค่าการแจกแจงแบบปัวส์ซงจากการแจกแจงแบบทวินาม เมื่อ **N** → ∞ 121
ผนวก ข ตัวอย่างซอร์สโคดของอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิต..... 123
ประวัติผู้เขียนวิทยานิพนธ์ 131

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง			หน้า
ตารางที่	3.1	Characterization of the Considered IQ Scheduling Algorithms…….....	18
ตารางที่	5.1	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%.....	51
ตารางที่	5.2	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%.....	51
ตารางที่	5.3	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%.....	52
ตารางที่	5.4	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%.....	52
ตารางที่	5.5	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%.....	52
ตารางที่	5.6	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%.....	53
ตารางที่	5.7	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells……	53
ตารางที่	5.8	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว เมื่อขนาดความยาวเอนเวโลป = 4 cells……	92
ตารางที่	5.9	แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วน	92

ตาราง หน้า

เวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells.....

ตารางที่ 5.11 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 93

ตารางที่ 5.12 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 94

ตารางที่ 5.13 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 94

ตารางที่ 5.14 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 94

ตารางที่ 5.15 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 94

ตารางที่ 5.16 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มี 95

ตาราง หน้า

เซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells……..

ตารางที่ 5.17 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 95

ตารางที่ 5.18 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 96

ตารางที่ 5.20 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 96

ตารางที่ 5.21 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 96

ตารางที่ 5.22 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 100

ตารางที่ 5.23 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มี 100

ตาราง หน้า

ค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells……

ตารางที่ 5.24 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells… 101

ตารางที่ 5.25 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม เมื่อขนาดความยาวเอนเวโลป = 4 cells..... 101

ตารางที่ 5.26 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells… 101

ตารางที่ 5.27 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 4 cells 102

ตารางที่ 5.28 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม เมื่อขนาดความยาวเอนเวโลป = 4 cells…… 102

ตารางที่ 5.29 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells 102

ตารางที่ 5.30 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 103

ตาราง หน้า

50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells

ตารางที่ 5.31 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells 103

ตารางที่ 5.32 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells 103

ตารางที่ 5.33 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50% โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม เมื่อขนาดความยาวเอนเวโลป = 16 cells..... 104

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ		หน้า
รูปที่ 2.1	แบบจำลองของสวิตช์พื้นฐาน…….	6
รูปที่ 2.2	สวิตช์แบบ MIQ…….	9
รูปที่ 2.3	แบบจำลองของสวิตช์แบบ CIOQ กับฟังก์ชัน back-pressure……	10
รูปที่ 2.4	สถาปัตยกรรมสวิตช์แบบ PPS based on output-queued switches……	12
รูปที่ 3.1	ตัวอย่างขั้นตอนทั้ง 3 ขั้นตอนของอัลกอริทึม PIM…….	17
รูปที่ 3.2	แสดงตัวอย่างการทำงานทั้ง 3 ขั้นตอน ของอัลกอริทึม RRM…….	19
รูปที่ 3.3	แสดงตัวอย่างการทำงานของอัลกอริทึม iSLIP……...	20
รูปที่ 3.4	แสดงผลกระทบของลำดับความสำคัญของเซลล์เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เพิ่มขึ้น ที่มีต่อ Prioritized iSLIP algorithm……	23
รูปที่ 3.5	แสดงความสัมพันธ์ระหว่างเซลล์ที่มีลำดับความสำคัญต่างกัน ใน Prioritized iSLIP algorithm เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เป็น 10%.....	23
รูปที่ 5.1	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request เวลาประวิงของเซลล์คลาส 1.....	34
รูปที่ 5.2	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request เวลาประวิงของเซลล์คลาส 2 ……..	34
รูปที่ 5.3	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request เวลาประวิงของเซลล์โดยเฉลี่ย…….	36
รูปที่ 5.4	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1.....	37
รูปที่ 5.5	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant เวลาประวิงของเซลล์คลาส 1.....	38
รูปที่ 5.6	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant เวลาประวิงของเซลล์คลาส 2.....	39
รูปที่ 5.7	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant เวลาประวิงของเซลล์โดยเฉลี่ย…….	40
รูปที่ 5.8	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1.....	41
รูปที่ 5.9	ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept เวลาประวิงของ	43

เซลล์คลาส 1.....

รูปที่ 5.10 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept เวลาประวิงของ เซลล์คลาส 2 43

รูปที่ 5.11 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept เวลาประวิงของ เซลล์โดยเฉลี่ย..... 45

รูปที่ 5.12 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept อัตราส่วนเวลา ประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 46

รูปที่ 5.13 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วน เวลาประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 100 เท่าได้ที่ offered load มีค่า 80% และมี สัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม……. 50

รูปที่ 5.14 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วน เวลาประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 10 เท่าได้ที่ offered load มีค่า 80% และมี สัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม……. 50

รูปที่ 5.15 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วนเวลา ประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 10 เท่าได้ที่ offered load มีค่า 90% และมี สัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม……. 51

รูปที่ 5.16 แสดงช่วงการใช้งานของแต่.ละอัลกอริทึม ที่ค่า offered load 80% และค่า สัดส่วนเซลล์คลาส 1 10%..... 52

รูปที่ 5.17 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ค่า offered load 80% และค่า สัดส่วนเซลล์คลาส 1 30%..... 52

รูปที่ 5.18 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ค่า offered load 80% และค่า สัดส่วนเซลล์คลาส 1 50%..... 53

รูปที่ 5.19 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดย เฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่าที่ค่าสัดส่วนเซลล์คลาส 1 10%..... 54

รูปที่ 5.20 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ย ของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่าที่ค่าสัดส่วนเซลล์คลาส 1 10% 54

รูปที่ 5.21 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมที่ให้อัตราส่วนเวลาประวิงโดย เฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1ที่ 10 เท่า ที่ค่าสัดส่วนเซลล์คลาส 55

1 30%.....

รูปที่ 5.22 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ย ของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30%..... 55

รูปที่ 5.23 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดย เฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50%..... 56

รูปที่ 5.24 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดย เฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50%..... 56

รูปที่ 5.25 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ของอัลกอริทึม iSLIP..... 57

รูปที่ 5.26 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ของอัลกอริทึม iSLIP……. 58

รูปที่ 5.27 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ของอัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1=10% 59

รูปที่ 5.28 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวของอัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1=30 % 59

รูปที่ 5.29 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวของอัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1=50 % 60

รูปที่ 5.30 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็มของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส1=10% 61

รูปที่ 5.31 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็มของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส1=30% 61

รูปที่ 5.32 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็มของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส1=50% 62

รูปที่ 5.33 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท โดยใช้อัลกอริทึม prioritized iSLIP……... 63

รูปที่ 5.34 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท 63

อัลกอริทึม prioritized iSLIP โดยวิธีเลือก เอนเวโลปที่มีเซลล์ที่หัวคิว envelope size = 4 cells……

รูปที่ 5.35 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท อัลกอริทึม prioritized iSLIP โดยวิธีเลือกเอนเวโลปบรรจุเต็ม envelope size = 4 cells…….. 64

รูปที่ 5.36 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 65

รูปที่ 5.37 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 2..... 66

รูปที่ 5.38 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย……. 68

รูปที่ 5.39 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 69

รูปที่ 5.40 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 70

รูปที่ 5.42 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 2 71

รูปที่ 5.43 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย..... 72

รูปที่ 5.44 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 73

รูปที่ 5.45 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 74

รูปที่ 5.46 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 2 …….. 75

รูปที่ 5.47 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย……. 76

รูปที่ 5.48 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 77

รูปที่ 5.49 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 79

รูปที่ 5.50 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 2 79

รูปที่ 5.51 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย……. 80

รูปที่ 5.52 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 81

รูปที่ 5.53 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 83

รูปที่ 5.54 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของ 83

เซลล์คลาส 2 .....

รูปที่ 5.55 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย……. 85

รูปที่ 5.56 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 86

รูปที่ 5.57 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 1..... 87

รูปที่ 5.58 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์คลาส 2 …….. 88

รูปที่ 5.59 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ เวลาประวิงของเซลล์โดยเฉลี่ย…….. 89

รูปที่ 5.60 ผลการทดสอบสมรรถนะของอัลกอริทึมควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป = 4 เซลล์ อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1..... 90

รูปที่ 5.61 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells……. 101

รูปที่ 5.62 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells…….. 101

รูปที่ 5.63 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells……. 102

รูปที่ 5.64 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัว 102

คิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells……

รูปที่ 5.65 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells……. 103

รูปที่ 5.66 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells…….. 103

รูปที่ 5.67 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells…….. 105

รูปที่ 5.68 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells……. 105

รูปที่ 5.69 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells…….. 106

รูปที่ 5.70 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells……. 106

รูปที่ 5.71 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells…….. 107

รูปที่ 5.72 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์หัวคิว ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells……. 107

รูปที่ 5.73 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells 109

รูปที่ 5.74 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells 109
รูปที่ 5.75 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells 110
รูปที่ 5.76 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells 110
รูปที่ 5.77 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells 111
รูปที่ 5.78 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells 111
รูปที่ 5.79 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells 112
รูปที่ 5.80 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells 113
รูปที่ 5.81 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells 113
รูปที่ 5.82 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells 114
รูปที่ 5.83 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์ 114

คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells
รูปที่ 5.84 แสดงช่วงการใช้งานของแต่ละอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์ 115
บรรจุเต็ม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์
คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความก้าวหน้าของเทคโนโลยีการส่งสัญญาณดิจิทัลผ่านสื่อนำสัญญาณเส้นใยแก้วนำแสง ได้มีการพัฒนาไปอย่างมากในช่วงสองทศวรรษที่ผ่านมา อัตราบิตของสัญญาณได้เพิ่มขึ้นอย่างรวดเร็วจาก 155.52 Mbps (STM-1) ในช่วงกลางทศวรรษ 1980s เป็น 622.08 Mbps (STM-4) 2.48 Gbps (STM-16) และ 40 Gbps (STM-256) จนถึง ณ ปัจจุบันอัตราบิตของการส่งสัญญาณอาจสูงได้ถึงในหลักของ Tbps โดยอาศัยกรรมวิธีการมัลติเพลกซ์เชิงความยาวคลื่น (WDM: Wavelength Division Multiplexing) เทคโนโลยีเหล่านี้ส่งผลให้การส่งผ่านสัญญาณความเร็วสูงเชื่อมต่อระหว่างคู่สวิตชิงโนดในโครงข่ายโทรคมนาคมเป็นไปได้จริงในทางปฏิบัติ อีกทั้งต้นทุนการรับส่งข้อมูลต่อบิตมีค่าลดลงเรื่อยๆ ดังนั้นการพัฒนาระบบสื่อสารที่มีความจุสูงมากเพื่อรองรับการใช้งานรูปแบบใหม่ๆจึงมีความเป็นไปได้และนำไปสู่การให้บริการระบบสื่อสารมัลติมีเดียผ่านโครงข่ายอินเทอร์เน็ตความเร็วสูง

ถึงแม้ว่าความจุของช่องสัญญาณสื่อสารจะได้รับการพัฒนาไปอย่างรวดเร็ว โดยอาศัยคลื่นสัญญาณแสงเป็นพาหะ แต่อุปกรณ์ของสวิตชิงโนดหรือเราเทอร์ยังคงอาศัยหลักการของอิเล็กทรอนิกส์เป็นพื้นฐานหลักสำหรับการประมวลผลสัญญาณข้อมูลในการส่งต่อแพ็กเกตข้อมูลให้เดินทางถึงจุดหมายปลายทางได้ตามต้องการ โดยทฤษฎีแล้วพบว่าอุปกรณ์อิเล็กทรอนิกส์ไม่สามารถทำงานที่ความถี่สูงเกินกว่า 40 GHz ได้ ฉะนั้น เมื่อเปรียบเทียบความจุจำนวนมหาศาลของเส้นใยแก้วนำแสงกับความจุในการสวิตช์ของสวิตช์หรือเราเทอร์อิเล็กทรอนิกส์ความเร็วสูง พบว่ามีค่าที่แตกต่างกันมาก จึงก่อให้เกิดปัญหาคอขวดขึ้น (bottleneck) ในการสวิตช์ข้อมูลผ่านสวิตช์ ปัญหาดังกล่าวนี้ส่งผลให้การประยุกต์ใช้งานเส้นใยแก้วนำแสงได้ไม่เต็มศักยภาพ การแก้ปัญหาคอขวดดังกล่าวสามารถกระทำได้โดยการพัฒนาสถาปัตยกรรมสวิตช์แพ็กเกตรูปแบบใหม่ที่ต่างไปจากในอดีตเพื่อให้สามารถทำงานร่วมกับระบบเชื่อมต่อเส้นใยแก้วนำแสงได้อย่างมีประสิทธิภาพ

นอกจากนี้สวิตช์แพ็กเกตที่พัฒนาขึ้นยังจำเป็นจะต้องมีขีดความสามารถในการรับประกันคุณภาพการให้บริการของทราฟฟิกแต่ละประเภท จึงมีความจำเป็นในการที่จะออกแบบอัลกอริทึมที่จัดลำดับเซลล์ให้ประมวลผลได้เร็วขึ้นและนำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายขึ้น ซึ่งจะทำให้สามารถลดค่าไทม์ดีเลย์ของแพ็กเกตจนกระทั่งคุณภาพการให้บริการของทราฟฟิกประเภทต่างๆอยู่ในค่าที่กำหนดไว้สำหรับโครงข่ายอินเทอร์เน็ต

ถึงแม้ว่าสวิตช์แบบรอคิวที่เอาต์พุต (Output-Queued (OQ) Switches) นั้นจะมีสมรรถนะของสวิตช์ดีกว่าสวิตช์แบบรอคิวที่อินพุต (Input-Queued (IQ) Switches) แต่การรอคิวที่เอาต์พุตก็ไม่เหมาะสมในการนำ มาใช้งานในทางปฏิบัติกับสวิตช์ความเร็วสูง เนื่องจากโครงสร้างสวิตช์ (switch fabric) และหน่วยความจำ (memory) จะต้องทำงาน (run) ที่ความเร็วเป็นจำนวน N เท่าของความเร็วแพ็กเกตที่อินพุต (line rate) ซึ่งสวิตช์ แบบ IQ นั้น สามารถลดแบนด์วิดท์ของหน่วยความจำได้เนื่องจากทำงานที่ความเร็วเท่ากับความเร็วของแพ็กเกตที่อินพุต (N คือขนาดของสวิตช์) จึงเป็นที่ดึงดูดความสนใจในการที่จะนำไปสร้างเป็นสวิตช์ความเร็วสูง และ/หรือสวิตช์ที่มีพอร์ตเป็นจำนวนมาก ทั้งนี้เพื่อให้เป็นไปตามความต้องการในการออกแบบ (satisfy) ทั้งความต้องการด้านความจุของสวิตช์ที่สูงขึ้น และยังการันตีในคุณภาพการให้บริการ สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (Combined Input and Output-Queued (CIOQ) Switches) จึงถูกนำมาพิจารณาว่าจะเป็นคำตอบของความต้องการนี้ สวิตช์แบบ CIOQ มีโครงสร้างสวิตช์ทำงานที่ความเร็วสูงกว่าความเร็วแพ็กเกตที่อินพุตเท่ากับ s เท่า ทำให้แต่ละไทม์สล็อตสามารถส่งแพ็กเกตจากอินพุตไปเอาต์พุตของสวิตช์ได้มากที่สุด s แพ็กเกต แน่นอนมันจึงมีความจำเป็นที่จะต้องมีบัฟเฟอร์ที่เอาต์พุตเพื่อให้แพ็กเกตที่ยังไม่ได้ถูกส่งออกจากสวิตช์รออยู่ในคิวจนกว่าจะได้รับการส่งออก

เทคนิคเอนเวโลป (envelope) [1] เป็นอีกวิธีหนึ่งที่ใช้ในการแก้ปัญหาคอขวดในการสวิตช์ข้อมูล โดยรวมแพ็กเกตหลายๆแพ็กเกตที่จะไปยังเอาต์พุตเดียวกันส่งไปพร้อมกันเป็นลำดับเปรียบเสมือนเป็นการเพิ่มขนาดความยาวของหน่วยข้อมูลที่จะส่งผ่านโครงสร้างสวิตซ์ แม้ว่าเทคนิคนี้อาจทำให้สมรรถนะทางด้านดีเลย์ของสวิตช์ลดลงไปบ้างแต่ก็เหมาะสมในการนำมาใช้เร่งด่วนในการแก้ปัญหานี้ ซึ่งในงานวิจัยฉบับนี้จะนำเสนอเทคนิคเอนเวโลปโดยนำเทคนิคนี้มาประยุกต์ใช้กับสวิตช์แบบต่าง ๆ ดังที่กล่าวมาแล้วข้างต้น และทำการประเมินสมรรถนะเพื่อหารูปแบบของสวิตช์ที่เหมาะสมที่จะนำเทคนิคเอนเวโลปมาประยุกต์ใช้ โดยยังคงสามารถรองรับทราฟฟิกทั้งประเภท real-time traffic และ data type traffic ในขณะเดียวกันได้ ภายใต้เงื่อนไขคือต้องรับประกันในคุณภาพการให้ บริการของทราฟฟิกแต่ละประเภท

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อพัฒนาสถาปัตยกรรมสวิตช์แพ็กเกตความเร็วสูง ที่มีขีดความสามารถในการรองรับทราฟฟิกได้สองประเภท ได้แก่ แบบเรียลไทม์และข้อมูลดาต้าได้ โดยการใช้เทคนิคเอนเวโลปควบคู่กับกลไกการสวิตซ์ที่สามารถรับประกันคุณภาพการให้บริการตามที่ทราฟฟิกแต่ละประเภทต้องการ

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์

ศึกษาปัญหาคอขวดในการสวิตช์ข้อมูล และหาวิธีการแก้ไขปัญหาโดยเน้นที่เทคนิคเอนเวโลป คิดค้นวิธีการที่จะนำเทคนิคเอนเวโลปมาประยุกต์ใช้กับสวิตช์ประเภทที่มีการวางบัฟเฟอร์ที่อินพุตของสวิตช์ ใช้อัลกอริทึมใหม่เพื่อรับประกันคุณภาพการให้บริการสำหรับทราฟฟิก 2 ระดับความสำคัญ ออกแบบสวิตช์ความเร็วสูงที่จะนำไปใช้ในโครงข่ายอินเทอร์เน็ตแบ็กโบน โดยที่จะจำกัดขอบเขตของวิทยานิพนธ์ ดังต่อไปนี้

1. สวิตช์เป็นแบบเซลล์สวิตช์

2. เป็นสวิตช์ที่มีการวางบัฟเฟอร์ที่พอร์ตอินพุตของสวิตช์

3. ทราฟฟิกโดยการมาถึงของแพ็กเกตเป็นแบบ uniform i.i.d.Bernoulli processes และแพ็กเกตมีการแจกแจงที่จะไปยังเอาท์พุตใดๆมีความน่าจะเป็นเท่ากัน

4. ทราฟฟิกเป็นประเภทเรียลไทม์และประเภทข้อมูลดาต้า

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาปัญหาคอขวดในการสวิตช์ข้อมูลในโครงข่ายอินเทอร์เน็ตแบ็กโบนและเทคนิคต่าง ๆ ที่จะนำมาใช้ในการแก้ปัญหานี้

2. ศึกษาและวิเคราะห์สมรรถนะของสวิตช์ความเร็วสูงประเภทต่างๆ ที่ใช้ในโครงข่ายอินเทอร์เน็ตความเร็วสูง

3. จำลองแบบสวิตช์ที่มีการวางบัฟเฟอร์ที่พอร์ตอินพุต และนำเทคนิคเอนเวโลปควบคู่กับกลไกการสวิตซ์ที่สามารถรับประกันคุณภาพการให้บริการมาประยุกต์ใช้

4. จำลองแบบสวิตช์ที่ใช้อัลกอริทึมการจัดลำดับเซลล์ที่นำเสนอสำหรับทราฟฟิก 2 ระดับความสำคัญเพื่อให้สามารถรับประกันคุณภาพการให้บริการได้

5. รวบรวมผลการจำลองแบบ วิเคราะห์สมรรถนะและประเมินผลที่ได้จากการจำลองแบบ

6. ออกแบบสวิตช์ความเร็วสูงที่ประยุกต์ใช้เทคนิคเอนเวโลป และใช้อัลกอริทึมการจัดลำดับเซลล์ที่นำเสนอ ซึ่งจะต้องสามารถรองรับทราฟฟิกทั้งประเภทเรียลไทม์และประเภทข้อมูลดาทาได้

7. สรุปและจัดทำรูปเล่มวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถออกแบบสวิตช์ความเร็วสูงที่ประยุกต์ใช้เทคนิคเอนเวโลปได้ เพื่อนำไปใช้กับโครงข่ายอินเทอร์เน็ตความเร็วสูง

2. สามารถแก้ปัญหาคอขวดในการสวิตช์ข้อมูลของสวิตช์ในโครงข่ายแบ็กโบนได้ ซึ่งเป็นความต้องการเร่งด่วนของผู้ให้บริการโครงข่ายอินเทอร์เน็ตในปัจจุบัน

3. รองรับการเพิ่มขนาดของโครงข่าย (scalability) ในอนาคต จากการเพิ่มขึ้นของผู้ใช้บริการโครงข่าย และนวัตกรรมใหม่ที่เกิดขึ้นในระบบอินเทอร์เน็ต ทำให้ความเร็วของแพ็กเกตและจำนวนพอร์ตของสวิตช์เพิ่มมากขึ้น

4. สามารถสนับสนุนคุณภาพการให้บริการหลายระดับสำหรับทราฟฟิกประเภทต่างๆที่ให้บริการในโครงข่ายเอทีเอ็ม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
สถาปัตยกรรมสวิตช์แพ็กเกต

ในบทนี้จะได้กล่าวถึงรายละเอียดของสถาปัตยกรรมสวิตซ์แพ็กเกตประเภทต่างๆ ที่มีอยู่ในปัจจุบันโดยทั่วไปแล้วสวิตช์แพ็กเกตจะมีโครงสร้างสวิตช์ (switch fabric) แสดงดังรูปที่ 2.1 ตามรูปเป็นแบบจำลองสวิตช์พื้นฐาน ซึ่งประกอบไปด้วย โครงสร้างการสวิตช์ (switching fabric) พอร์ตอินพุต (input port) และพอร์ตเอาต์พุต (output port) โดยสมมุติว่าโครงสร้างสวิตช์ ทำหน้าที่ส่งผ่านหน่วยข้อมูลที่เรียกว่า**เซลล์** แพ็กเกตขนาดต่าง ๆ กันจะถูกแบ่งออกเป็นเซลล์ขนาดเท่ากันส่งผ่านโครงสร้างสวิตช์ และประกอบขึ้นใหม่อีกครั้งก่อนออกจากสวิตช์ โดยฟังก์ชันการทำงานในส่วนนี้เรียกว่า Segmentation and Reassemble (SAR) function และให้เป็นหน้าที่ของไลน์การ์ด (line card) จึงถูกพิจารณาให้เป็นส่วนที่อยู่ภายนอกโครงสร้างสวิตช์ ทั้งสวิตช์เอทีเอ็มและเราเตอร์ไอพี จะมีไลน์การ์ดทำหน้าที่แบ่งย่อยแพ็กเกตออกเป็นเซลล์ขนาดเท่ากัน เซลล์จะถูกส่งผ่านโครงสร้างสวิตช์และประกอบกันขึ้นเป็นแพ็กเกตอีกครั้งก่อนออกจากสวิตช์

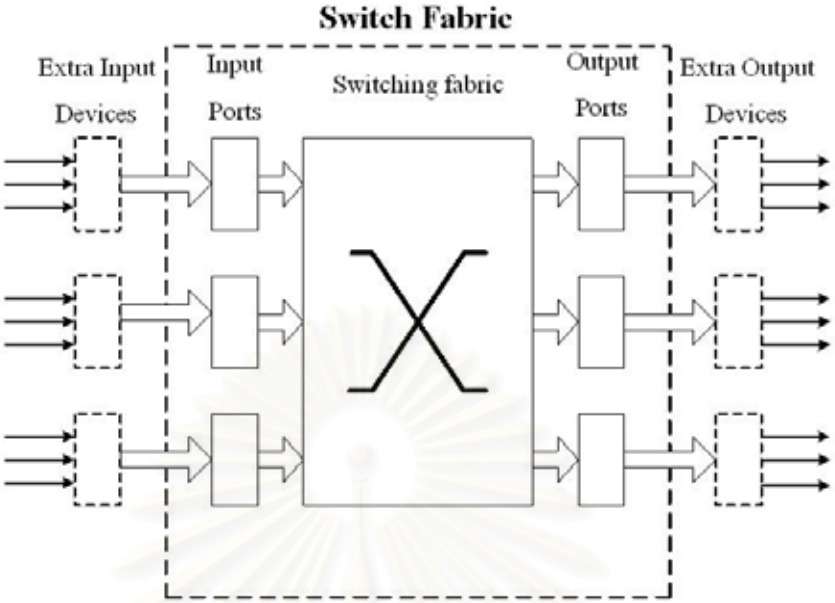
ในปัจจุบัน ไม่ว่าเป็นสวิตช์หรือเราเตอร์ที่ส่งผ่านแพ็กเกตไอพี ส่วนใหญ่ใช้โครงสร้างแบบสวิตช์เซลล์ เวลาที่ใช้ในการส่งเซลล์แต่ละเซลล์ผ่านโครงสร้างสวิตช์เรียกว่า ไทม์สล็อต (time slot) โดยโครงสร้างสวิตช์ ประกอบไปด้วยองค์ประกอบมูลฐาน 3 องค์ประกอบ ดังต่อไปนี้

1) พอร์ตอินพุต ทำหน้าที่รับหน่วยข้อมูลหรือเซลล์ ที่มาถึงที่พอร์ตอินพุตของโครงสร้างสวิตช์

2) พอร์ตเอาต์พุต ทำหน้าที่ส่งหน่วยข้อมูลออกจากพอร์ตเอาต์พุตของโครงสร้างสวิตช์

3) โครงสร้างการสวิตช์ หรือโครงข่ายเชื่อมต่อระหว่างกัน (interconnection network) ทำหน้าที่ให้การเชื่อมต่อระหว่างพอร์ตอินพุตกับเอาต์พุต เพื่อส่งผ่านหน่วยข้อมูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.1 แบบจำลองของสวิตช์พื้นฐาน [1]

ในปัจจุบันสถาปัตยกรรมสวิตช์แพ็กเกตส่วนใหญ่แบ่งออกเป็น 3 ประเภท ดังต่อไปนี้

1) สวิตช์แบบรอคิวที่เอาต์พุต (Output-Queued (OQ) Switches)

2) สวิตช์แบบรอคิวที่อินพุต (Input-Queued (IQ) Switches)

3) สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (Combined Input and Output-Queued (CIOQ) Switches)

2.1 สวิตช์แบบรอคิวที่เอาต์พุต (OQ Switches)

เป็นรูปแบบของสวิตช์ที่มีการวางบัฟเฟอร์ที่เอาต์พุตของสวิตช์ โครงสร้างสวิตช์จะทำการส่งผ่านเซลล์ ทุกๆเซลล์ที่มาถึงพอร์ตอินพุตของสวิตช์ไปยังพอร์ตเอาต์พุตโดยทันที โดยเซลล์จะรอคิวอยู่ในบัฟเฟอร์ของสวิตช์เพื่อรอการส่งออก ซึ่งจำเป็นที่สวิตช์จะต้องมีความเร็วสูงกว่าความเร็วของแพ็กเกตเป็นจำนวน N เท่าโดยที่ N คือ จำนวนพอร์ตอินพุต ซึ่งจะเห็นว่าถ้าสวิตช์มีขนาดใหญ่ขึ้นจะไม่เหมาะสมที่จะนำมาใช้ในทางปฏิบัติแต่สวิตช์แบบ รอคิวที่เอาต์พุต ก็มีข้อดีตรงที่ว่าสมรรถนะในการส่งผ่านแพ็กเกตที่สูงกว่าและค่าเฉลี่ยไทม์ดีเลย์ของแพ็กเกตที่ต่ำกว่าสวิตช์แบบรอคิวที่อินพุต [2]

2.1.1 สวิตช์แบบ SOQ (Single Output-Queued (OQ) Switches)

สวิตช์แบบ OQ เป็นสวิตช์แบบที่ใช้หน่วยความจำร่วมกัน (shared-memory switch type) ซึ่งในการจัดสรรหน่วยความจำเพื่อใช้ร่วมกันนั้นมีอยู่ 2 วิธี วิธีที่หนึ่งคือแบ่งหน่วยความจำขนาดเท่าๆ กันให้กับบัฟเฟอร์ในแต่ละเอาต์พุต (Complete partitioning) หรือวิธีที่สองคือใช้หน่วยความจำร่วมกันสำหรับบัฟเฟอร์ทุกๆ พอร์ตของเอาต์พุต (Full sharing)

ในวิธี Complete partitioning แพ็กเกตที่ไปยังเอาต์พุตเดียวกันจะถูกกำจัดทิ้งไป (lost) ถ้าบัฟเฟอร์ที่เอาต์พุตนั้นเต็ม แต่วิธี Full sharing นั้น แพ็กเกตจะยังไม่ถูกกำจัดตราบเท่าที่หน่วยความจำทั้งหมดยังไม่เต็มจึง ทำให้สมรรถนะทางอัตราการสูญเสียแพ็กเกต (lost rate)ดีกว่า จึงเป็นสิ่งที่ต้องการในการออกแบบซึ่งจะสามารถออกแบบให้สวิตช์มีขนาดหน่วยความจำเล็กลงได้มากกว่าใช้วิธีที่หนึ่ง [3]

2.1.2 สวิตช์แบบ MOQ (Multiple Output-Queued (MOQ) Switches)

สวิตช์แบบ MOQ ใช้หน่วยความจำร่วมกัน หรือโครงสร้างสวิตช์ที่มีการรอคิวที่เอาต์พุต ซึ่งการนำไปสร้างเป็นสวิตช์นั้นมีลักษณะเช่นเดียวกันกับสวิตช์แบบ OQ แต่จะมีความซับซ้อนมากกว่าในการนำไปสร้างเป็นสวิตช์ความเร็วสูงที่เหมาะสมในทางปฏิบัติ โดยใช้วิธี back-pressure จำกัดการทำงานและการเก็บเซลล์ที่มาถึงบัฟเฟอร์ที่เอาต์พุตของสวิตช์ซึ่งอยู่ภายใน line card ดังนั้นโครงสร้างบัฟเฟอร์ก็เพียงแต่ควบคุมคิวที่มีขนาดเล็กลงและมีจำนวนเซลล์รออยู่ในคิวไม่มาก ทำให้สวิตช์ประเภทนี้มีลักษณะใกล้เคียงกับสวิตช์แบบ OQ สวิตช์มากที่สุด ซึ่งการสร้างสวิตช์เลียนแบบสวิตช์แบบ OQ เรียกว่า OQ emulation เนื่องจากสวิตช์แบบ MOQ เป็นกรณีพิเศษของสวิตช์ประเภท CIOQ จึงไม่ขอลงรายละเอียดในหัวข้อนี้

2.2 สวิตช์แบบรอคิวที่อินพุต (IQ Switches)

ข้อดีของสวิตช์ประเภทนี้คือ โครงสร้างสวิตช์สามารถทำงานได้ที่ความเร็วเท่ากับความเร็วของ line rate โดยที่แพ็กเกตแต่ละแพ็กเกตที่เข้าสู่สวิตช์จะถูกแบ่งออกเป็นเซลล์ แล้วบรรจุลงในบัฟเฟอร์ที่ตำแหน่งอินพุตแต่ละอินพุตทันทีเพื่อรอการส่งผ่านสวิตช์ นอกจากนี้ สวิตช์จะไม่เกิดปัญหาของการแออัดหรือชนกันของเซลล์ภายในสวิตช์ (internal contention) เพราะสวิตช์สามารถควบคุมและจัดการในการส่งเซลล์ผ่านสวิตช์ได้ตามต้องการ ในการพิจารณาเลือกเซลล์ที่หัวคิวสำหรับส่งผ่านสวิตช์ หรือเรียกว่ากฎควบคุมการเลือก (selection policy) ซึ่งมีอยู่หลายวิธี ตัวอย่างเช่นวิธี random selection, longest selection หรือ fixed priority selection เป็นต้น [4]

เช่นเดียวกันกับการรอคิวที่เอาต์พุต เราจะสมมุติว่าการมาถึงของแพ็กเกตเป็นกระบวนการ i.i.d. Bernoulli processes และในไทม์สล็อตใดๆ ความน่าจะเป็นที่แพ็กเกตจะมาถึงอินพุตแต่ละอินพุตของสวิตช์คือ p แต่ละแพ็กเกตมีความน่าจะเป็นที่จะไปยังเอาต์พุตใดๆ เท่ากับ 1/N (N คือจำนวนเอาต์พุตพอร์ต) ตัวควบคุมการสวิตช์ (switch controller) จะทำการเลือกเซลล์ที่หัวคิว ซึ่งเซลล์ที่มาถึงอินพุตจะรอคิวอยู่ในบัฟเฟอร์แบบมาถึงก่อนได้รับการบริการก่อน (หรือ FCFS) เมื่อมีเซลล์จะไปยังเอาต์พุตใดๆ ตัวควบคุมการสวิตช์จะทำการปิดจุดเชื่อมต่อระหว่างอินพุตกับเอาต์พุตนั้นเพื่อส่งเซลล์ผ่านสวิตช์ จะเห็นว่าอาจจะมีเซลล์จำนวน k เซลล์ที่จะไปยังเอาต์พุตเดียวกัน แต่จะได้รับการพิจารณาส่งผ่านสวิตช์เพียงหนึ่งเซลล์ในแต่ละไทม์สล็อต เซลล์ที่ไม่ได้รับการส่งจะรอคิวอยู่ในบัฟเฟอร์เพื่อรอรับการเลือกในการส่งผ่านสวิตช์ในไทม์สล็อตถัดไป

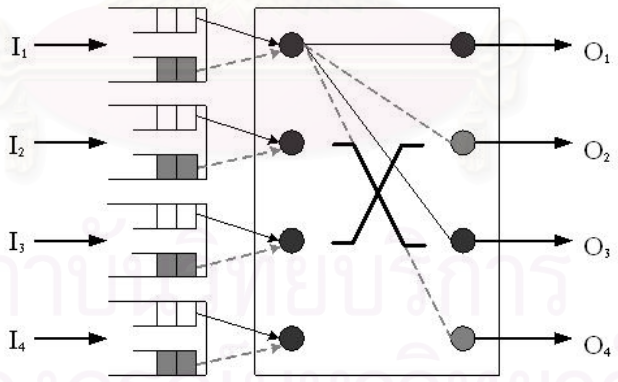
2.2.1 Multiple Input-Queued (MIQ) Switches [5]

เมื่อเปรียบเทียบกับสวิตช์แบบ OQ แล้ว สวิตช์แบบ IQ มีข้อดีที่เหนือกว่าคือ สวิตช์สามารถทำงานได้ที่ความเร็วในการสวิตช์เท่ากับ line rate ทำให้สวิตช์แบบ IQ มีความเหมาะสมอย่างยิ่งที่จะนำไปใช้กับระบบที่มีการสวิตช์ความเร็วสูง การรอคิวที่อินพุตถูกพิจารณาว่านำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายกว่าการรอคิวที่เอาต์พุตแต่เนื่องจากสวิตช์แบบ IQ ประสบปัญหาเนื่องจากการติดขัดที่หัวคิว (HOL blocking) ทำให้ค่าทรูพุตของสวิตช์ลดลงลู่เข้าสู่ค่า 0.586 เมื่อ N มีค่าเข้าสู่อนันต์ ดังนั้น จึงได้มีการศึกษาเพื่อหารูปแบบการจัดการบัฟเฟอร์ (buffering) และการจัดลำดับเซลล์ (Scheduling) เพื่อเอาชนะขีดจำกัดของค่าทรูพุตสูงสุดของสวิตช์ ตัวอย่าง เช่นวิธี Window policy, input smoothing หรือรูปแบบ Mulitple input-queueing ซึ่งแต่ละรูปแบบนั้นยอมให้เซลล์ต่าง ๆ ที่อยู่ลำดับถัดจากเซลล์ที่หัวคิว (HOL cell) สามารถส่งผ่านสวิตช์ได้ เมื่อเซลล์ที่หัวคิวไม่สามารถส่งออกได้เนื่องจากเอาต์พุตที่จะไปนั้นไม่ว่าง หรือยอมให้มีหลาย ๆ เซลล์เป็นเซลล์ที่หัวคิวในแต่ละอินพุต เพื่อเพิ่มโอกาสในการได้รับการเลือกส่งผ่านสวิตช์ รูปแบบที่ได้รับสนใจมากที่สุด คือรูปแบบ MIQ เนื่องจากให้สมรรถนะของสวิตช์ที่สูงกว่าและสามารถทำงานได้ในการสวิตช์ความเร็วสูง

ตามรูปที่ 2.3 เป็นสวิตช์แบบ MIQ ขนาด NxN (N คือจำนวนพอร์ตอินพุตและเอาต์พุต)โดยที่แต่ละอินพุตมีการจัดการแยกคิวออกเป็น m คิว ($1 \leq m \leq N$) ซึ่งแต่ละคิวคือกลุ่มของพอร์ตเอาต์พุต ในกรณีที่ m = 1 ก็คือสวิตช์แบบ SIQ และกรณีที่ m = N ก็คือสวิตช์แบบ **Virtual Output Queued (VOQ) switch** ซึ่งสวิตช์แบบ VOQ นั้นเป็นกรณีพิเศษของสวิตช์แบบ MIQ และมีความน่าจะเป็นที่จะเกิดการติดขัดที่หัวคิวเป็นศูนย์ ดังนั้นจะเห็นว่าความน่าจะเป็นที่จะเกิดการติดขัดที่หัวคิว มีค่าลดลงเมื่อจำนวนการแยกคิว m ในแต่ละอินพุตมีค่าเพิ่มขึ้น

ค่าของ m ปกติแล้วจะเป็นจำนวนเท่าของ 2 (1,2,4,…,N) หรือ mN (m อาจจะมีขนาดใหญ่กว่า N ก็ได้แต่ก็ไม่เหมาะสมในทางปฏิบัติ) ดังนั้น แต่ละคิวในแต่ละอินพุตจะบรรจุเซลล์ที่จะไปยังกลุ่มเอาต์พุตจำนวน N/m พอร์ต โดยที่กลุ่มเอาต์พุตแต่ละกลุ่มจะต้องไม่ซ้ำกับกลุ่มเอาต์พุตอื่น ซึ่งเราจะสังเกตได้ว่าเราสามารถแยกสวิตช์ออกเป็นสวิตช์ย่อยได้ m สวิตช์ โดยที่แต่ละสวิตช์ย่อยจะมีขนาด NxN/m ซึ่งสวิตช์ย่อยแต่ละตัวตรงกับกลุ่มของเอาต์พุตพอร์ต

ในแต่ละไทม์สล็อต สวิตช์แบบ MIQ สามารถส่งเซลล์ในแต่ละอินพุตผ่านสวิตช์ได้ถึง m เซลล์ แต่อย่างไรก็ตาม เมื่อรวมจำนวนเซลล์ทั้งหมดจากทุกๆ อินพุตแล้วจะต้องไม่เกิน N เซลล์ เนื่องจากแต่ละเอาต์พุตสามารถรับเซลล์ที่มาจากอินพุตได้ไม่เกิน 1 เซลล์ แม้ว่าการส่งออกเซลล์มากกว่า 1 เซลล์จากแต่ละอินพุตเป็นสิ่งที่จำเป็นด้วยเหตุผลที่ว่าจะทำให้สมรรถนะของสวิตช์สูงขึ้น แต่มันก็ต้องเพิ่มความเร็วในการสวิตช์ข้อมูลในขณะเดียวกัน ซึ่งเป็นสิ่งที่ไม่พึงปรารถนาสำหรับสวิตช์ความเร็วสูง ด้วยเหตุนี้จึงกลายเป็นปัญหาสำคัญที่จะต้องทำการแก้ไขถ้าจะนำไปใช้เป็นสวิตช์ความเร็วสูงในอนาคต ดังนั้นจึงได้มีการปรับปรุงสวิตช์แบบ MIQ โดยสมมุติว่าในแต่ละไทม์สล็อตให้แต่ละอินพุตสามารถส่งเซลล์ผ่านสวิตช์ได้ไม่เกิน 1 เซลล์ ทำให้สวิตช์สามารถทำงานได้ที่ความเร็ว line rate ซึ่งทำให้สวิตช์มีความเหมาะสมในการนำไปใช้กับระบบที่มีการสวิตช์ความเร็วสูง (high-speed switching system) และต้องการสมรรถนะของสวิตช์ที่สูงขึ้นด้วย



รูปที่ 2.2 สวิตช์แบบ MIQ [5]

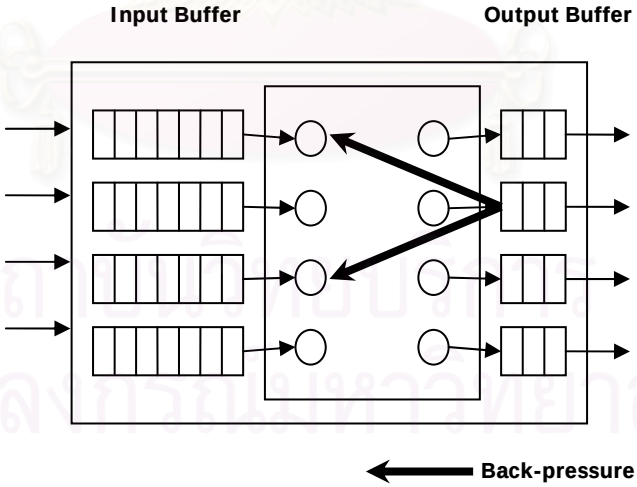
2.3 สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (CIOQ switches) [1], [6]

จากที่กล่าวมาข้างต้น สวิตช์แบบ OQ นั้นมีสมรรถนะที่ดีกว่าสวิตช์ประเภทอื่น ๆ แต่ผลเนื่องจากความ เร็วในการสวิตช์นั้นซึ่งมีความเร็ว N เท่าของ line rate จึงไม่เหมาะสมในการนำมาใช้ในทางปฏิบัติกับสวิตช์ความ เร็วสูงและ/หรือมีจำนวนพอร์ตมาก ๆ และจากข้อดีของสวิตช์

แบบ IQ ที่มีความเร็วในการทำงานของสวิตช์เท่ากับความเร็วของ line rate จึงได้มีผู้นำเสนอสวิตช์รูปแบบที่มีการวางบัฟเฟอร์ทั้งทางด้านอินพุตและเอาต์พุตรวมกับฟังก์ชันการทำงานที่เรียกว่า back-pressure และเรียกสวิตช์ประเภทที่นี้ว่าสวิตช์แบบ CIOQ

ในทางปฏิบัติ สวิตช์แบบ IQ ส่วนใหญ่มีความต้องการที่จะให้เป็นสวิตช์แบบไม่ติดขัด (nonblocking) ทำให้ต้องมีการเพิ่มความเร็วในการสวิตช์ (ตัวอย่างเช่น 2เท่าของความเร็ว line rate) ทำให้เกิดการสะสมของเซลล์ที่พอร์ตเอาต์พุต จึงจำเป็นที่จะต้องนำบัฟเฟอร์ที่เอาต์พุตมาใช้ร่วมด้วย เพื่อให้เซลล์ที่ยังไม่ได้ถูกส่งออกจากสวิตช์รอคิวอยู่ในบัฟเฟอร์ จึงถูกจัดให้อยู่ในประเภทสวิตช์แบบ CIOQ เช่นเดียวกัน

สถาปัตยกรรมสวิตช์แบบ CIOQ ตามรูปที่ 2.4 จะวางบัฟเฟอร์ที่มีหน่วยความจำความเร็วต่ำ (ราคาไม่แพง) จำนวนมากที่อินพุต และวางบัฟเฟอร์ที่มีหน่วยความจำความเร็วสูง (ราคาแพง) จำนวนไม่มากที่เอาต์พุต และเพื่อเพิ่มสมรรถนะของสวิตช์ให้สูงขึ้น จึงทำการควบคุมการรอคิวในบัฟเฟอร์ทั้งที่อินพุตและเอาต์พุต โดยให้เป็นหน้าที่ของฟังก์ชัน back-pressure ซึ่งจะส่งสัญญาณไปที่บัฟเฟอร์ที่อินพุต เพื่อห้ามมิให้อินพุตส่งออกเซลล์จากอินพุตไปยังเอาต์พุตที่มีเซลล์ในบัฟเฟอร์ที่อินพุตจะไปยังเอาต์พุตนั้น เมื่อจำนวนเซลล์ในบัฟเฟอร์ที่เอาต์พุตนั้นเต็มหรือเกินค่าเทรสโฮลด์ (threshold) เพื่อป้องกันเซลล์ล้นบัฟเฟอร์ที่เอาต์พุต ฟังก์ชัน back-pressure จึงทำให้ไม่เกิดความคับคั่ง (congestion) ขึ้นในสวิตช์



รูปที่ 2.3 แบบจำลองของสวิตช์แบบ CIOQ กับฟังก์ชัน back-pressure [6]

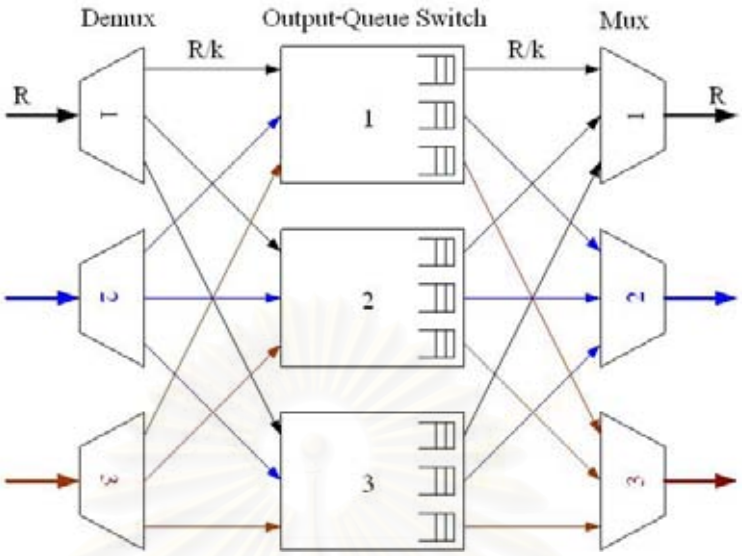
2.4 สวิตช์แบบต่อขนาน (Parallel Packet Switch:PPS) [1], [7]

สวิตช์แบบ PPS มีเป้าหมายที่จะเอาชนะขีดจำกัดทางด้านแบนด์วิดท์ของหน่วยความจำของสวิตช์แบบ OQ สวิตช์แบบ PPS ประกอบไปด้วยสวิตช์ความเร็วต่ำชนิดเดียวกันต่อขนานกัน แพ็กเกตที่มาถึงอินพุตเป็นลำดับจะถูกกระจายออกไปแพ็กเกตต่อแพ็กเกตโดยตัวดีมัลติเพลกซ์ ส่งผ่านสวิตช์ความเร็วต่ำแล้วรวมกันใหม่อีกครั้งโดยตัวมัลติเพลกซ์ที่เอาต์พุตก่อนออกจากสวิตช์ สถาปัตยกรรมสวิตช์แบบ PPS มีลักษณะคล้ายคลึงกับโครงข่ายคลอส (Clos network) ตามรูปที่ 2.5 ตัวดีมัลติเพลกซ์ สวิตช์แพ็กเกตตอนกลาง และตัวมัลติเพล็กซ์ เปรียบเทียบได้กับโครงข่ายครอส 3 ตอนแบบไม่มีบัฟเฟอร์

ถ้าเรามองทางด้านแพ็กเกตที่มาถึงสวิตช์ สวิตช์แบบ PPS ก็มีลักษณะเป็นสวิตช์ตอนเดียว (single stage packet switches) การจัดการรอคิวที่สวิตช์ตอนกลางที่ต่อขนานกันนั้น ก็มีเป้าหมายที่จะไม่ให้สวิตช์แบบ PPS มีความจำเป็นที่จะต้องทำงานที่ความเร็วเท่ากับความเร็วของ line rate ตัวดีมัลติเพลกซ์จะทำการเลือกสวิตช์ตอนกลาง (เราจะเรียกสวิตช์ตอนกลางที่ต่อขนานกันแต่ละตัวว่าเลเยอร์ (layer)) เพื่อที่จะส่งแพ็กเกตที่มาถึงไปยังเลเยอร์นั้น และแพ็กเกตจะรอคิวอยู่ในบัฟเฟอร์ของสวิตช์ตอนกลางจนกว่าจะถึงเวลาส่งออก (departure time) เมื่อเวลาส่งออกแพ็กเกตมาถึง แพ็กเกตจะถูกส่งต่อไปยังตัวมัลติเพลกซ์ซึ่งจะทำหน้าที่ส่งแพ็กเกตออกจากสวิตช์ จะเห็นได้ว่าฟังก์ชันการทำงานของตัวดีมัลติเพลกซ์และตัวมัลติเพล็กซ์มีบทบาทสำคัญในสวิตช์แบบ PPS

สวิตช์ตอนกลางนั้นเราอาจใช้สถาปัตยกรรมแบบใดๆ ก็ได้ที่เหมาะสม ตัวอย่างเช่นสวิตช์แบบ OQ หรือ สวิตช์แบบ CIOQ ตามรูปที่ 2.5 เป็นสวิตช์แบบ PPS ขนาด 3x3 โดยใช้สวิตช์ตอนกลางเป็นแบบ OQ ซึ่งแต่ละพอร์ตของสวิตช์ทำงานที่ความเร็วเท่ากับ R แต่ละพอร์ตเชื่อมต่อกับสวิตช์ OQ ตอนกลางมีทั้งหมด 3 ตัว เมื่อเซลล์มาถึงที่พอร์ตอินพุตตัวดีมัลติเพลกซ์จะทำการเลือกเลเยอร์ที่จะส่งเซลล์ไป และเนื่องจากเซลล์ต่างๆ ที่มาจากพอร์ตอินพุตมีความเร็วเท่ากับ R ถูกกระจายไป k เส้นทางทำให้แต่ละเส้นทางจะต้องทำงานที่ความเร็วไม่ต่ำกว่า R/k

ในแต่ละเลเยอร์ อาจจะเป็นสวิตช์แบบ OQ หรือ CIOQ ที่มีความเร็วในการทำงานของหน่วยความจำน้อยกว่า R เมื่อแต่ละเลเยอร์ได้รับเซลล์จากอินพุตต่างๆ ก็จะทำการส่งผ่านเซลล์ไปยังพอร์ตเอาต์พุต ถ้าในขณะนั้นเกิดความคับคั่ง (Congestion) ขึ้นเซลล์ก็จะถูกเก็บไว้ในบัฟเฟอร์ของสวิตช์ตอนกลางก่อน คอยจนกระทั่งเส้นทางที่จะไปยังตัวมัลติเพลกซ์ว่างลง จึงจะทำการส่งเซลล์ออกจากสวิตช์ตอนกลาง เมื่อเซลล์มาถึงอินพุตของตัว มัลติเพลกซ์ ตัวมัลติเพลกซ์จะทำการเลือกเซลล์ที่มาจากแต่ละเลเยอร์ส่งออกจากสวิตช์ ซึ่งบัฟเฟอร์ที่เอาต์พุตของแต่ละเลเยอร์จะต้องมีหน่วยความจำที่ทำงานด้วยความเร็วไม่น้อยกว่า R/k



รูปที่ 2.4 สถาปัตยกรรมสวิตช์แบบ PPS based on output-queued switches [7]

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์

บทนี้จะกล่าวถึงอัลกอริทึมการจัดลำดับเซลล์ส่งผ่านสวิตช์แบบต่างๆ อธิบายถึงการแมตช์ต่างๆ คุณสมบัติของอัลกอริทึมการจัดลำดับเซลล์ ยกตัวอย่างเช่น high throughput starvation free, fast และ simple to implement และอัลกอริทึมการจัดลำดับเซลล์แบบต่างๆจากในอดีตจนถึงปัจจุบัน ยังได้พูดถึงอัลกอริทึมใหม่ที่ได้นำเสนอเพื่อทราฟฟิก 2 ระดับความสำคัญด้วย

The scheduling algorithm in IQ switches (or matching) [8]-[10]

ในส่วนนี้จะอธิบายถึงกฎ (rule) ที่ใช้ในการตัดสินใจเลือกเซลล์ที่จะได้รับการส่งผ่านสวิตช์ เนื่องจากการจัดลำดับเซลล์ส่งผ่านสวิตช์ในสวิตช์ IQ สามารถแทนด้วยแบบจำลองปัญหาการแมตช์บนกราฟไบพาร์ไทท์ (bipartite graph) สถานะของสวิตช์สามารถอธิบายได้ด้วยกราฟไบพาร์ไทท์ G = [V,E] โดยที่จุดยอดคือเซต V ซึ่งแบ่งออกเป็นสองเซตย่อยคือ เซตย่อย V_i มีสมาชิก $v_i^{(k)}$ แทนอินพุตพอร์ต และเซตย่อย V_o มีสมาชิก $v_o^{(k)}$ แทนเอาต์พุตพอร์ต ด้าน (edges) แทนความต้องการที่จะส่งเซลล์ผ่านสวิตช์จากอินพุตไปยังเอาต์พุต(ด้านระหว่าง $v_i^{(m)}$ กับ $v_o^{(n)}$ บ่งชี้ถึงความต้องการที่จะส่งเซลล์จากอินพุตที่ m ไปยังเอาต์พุตที่ n) และแทนด้วยเมตริกซ์ของน้ำหนัก (weight) ด้วยสัญญลักษณ์ $w^{(mn)}$ เมตริกซ์ที่นำมาใช้นี้มีบทบาทสำคัญสำหรับอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ เพราะว่ามันสามารถบ่งชี้ได้ว่ามีอย่างน้อยหนึ่งเซลล์ที่จะได้รับบริการส่งผ่านสวิตช์ หรือแทนจำนวนเซลล์ที่จะได้รับบริการส่งผ่านสวิตช์ หรือแทนเวลานานที่สุดที่เซลล์รออยู่ในคิว

การแมตช์ (M) หมายถึงการเลือกเซตย่อยของด้านที่ตรงตามเงื่อนไข (admissible) เซตย่อยของด้านที่ตรงตามเงื่อนไขก็ต่อเมื่อไม่มีจุดยอดใดมีด้านมากกว่าหนึ่งด้านเชื่อมต่ออยู่ แสดงว่าจะไม่มีเหตุการณ์ที่เซลล์ 2 เซลล์ออกจากอินพุตเดียวกัน หรือมีเซลล์ 2 เซลล์มาถึงเอาต์พุตเดียวกันในเวลาเดียวกัน

การแมตช์จะมีขนาดสูงที่สุด (maximum size) เมื่อมีจำนวนด้านสูงสุด และการแมตช์จะมีน้ำหนักสูงที่สุด (maximum weight) เมื่อผลรวมเมตริกซ์ของด้านมีค่าสูงที่สุด การแมตช์จะมีขนาด maximal ก็ต่อเมื่อ ถ้ามีการเพิ่มด้านใด ๆ บนกราฟ G แล้วจะทำให้มันไม่ตรงตามเงื่อนไข (inadmissible)

ความต้องการของเราคือต้องการหาอัลกอริทึมการแมตช์ที่เหมาะสมที่สุดที่จะนำมาใช้ ซึ่งเราจะหาได้จากการนิยามความซับซ้อนของอัลกอริทึม (complexity) อัลกอริทึม maximum weight

matching (MWM) นั้นมีค่าความซับซ้อนคือ $O(N^3)$ ส่วนอัลกอริทึม maximum size matching มีค่าความซับซ้อนคือ $O(N^{5/2})$ ซึ่งมีความซับซ้อนน้อยกว่าแต่ประสิทธิภาพด้อยกว่าอัลกอริทึม MWM

เมตริกซ์ขนาด NxN ซึ่งมีสมาชิกคือเมตริกซ์ของด้านบนกราฟ G =[V,E] ถูกเรียกว่าเมตริกซ์ของน้ำหนัก (weight matrix) แทนด้วยสัญญลักษณ์ $W = [w^{(ij)}]$ เมตริกซ์ W มีค่าเปลี่ยนแปลงตามเวลาซึ่งสอดคล้องกับการเปลี่ยนแปลงของค่าพารามิเตอร์ในระบบซึ่งสมาชิกของเมตริกซ์ถูกนำมาคำนวณ เราจะสมมุติว่า $w^{(ij)} = 0$ เมื่อไม่มีด้านบนกราฟ G ซึ่งหมายความว่าไม่มีเซลล์จากอินพุต i ที่จะไปเอาต์พุต j รออยู่ในคิว

อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ ตามตารางที่ 1 สามารถแบ่งย่อยออกเป็นองค์ประกอบหลัก 2 องค์ประกอบคือ

(1) **Metrics computation** จะทำการคำนวณหาเมตริกซ์ของน้ำหนัก $W = [w_k^{(ij)}]$ แต่ละด้านจากที่เป็นไปได้ N^2 ด้านบนกราฟ G ซึ่งสัมพันธ์กับเมตริกซ์เนื่องจากสถานะของคิว กล่าวคือ $w_k^{(ij)}$ ขึ้นอยู่กับ $X_k^{(ij)}$ ซึ่งเป็นสถานะของคิว $Q^{(ij)}$ ในไทม์สล็อตที่ k จะเห็นได้ว่าเมตริกซ์มีพฤติกรรมเหมือนเป็นค่า priority ของเซลล์ต่างๆ ที่จะได้รับการบริการส่งผ่านสวิตช์

(2) **Matching method** จะทำการคำนวณหาค่าการแมตช์ อัลกอริทึม MSM PIM และ iSLIP มีเป้าหมายที่จะหาค่า maximum size matching ส่วนอัลกอริทึมที่เหลือพยายามที่จะหาค่า maximum weight matching

ตารางที่ 3.1 Characterization of the Considered IQ Scheduling Algorithms [8]

Algorithm	***Metric***	Matching method
MWM-QL	QL:Queue Length	Maximum Weight
MWM-CA	CA:Cell age	Maximum Weight
MSM	QO:Queue Occupancy	Maximum Size
iLQF	QL:Queue Length	Iterative Search
iOCF	CA:Cell age	Iterative Search
iSLIP, PIM	QO:Queue Occupancy	Iterative Search

สรุปได้ว่า อัลกอริทึม MWM-QL, MWM-CA และ MSM นั้น เป็นอัลกอริทึมที่เหมาะสมที่สุด (optimal-algorithms) ที่ใช้คำนวณหาค่า maximum weight หรือ maximum size matching ส่วนอัลกอริทึม iSLIP, PIM, iLQF และ iOCF นั้นคำนวณหาค่า maximal matching โดยใช้วิธีการ

วนซ้ำ (iterative search) และมีความซับซ้อนน้อยกว่าในการนำไปสร้างเป็นฮาร์ดแวร์ ในทางปฏิบัติจึงนำไปสร้างเป็นสวิตช์ความเร็วสูง

อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์แบบ IQ (Scheduling algorithms for IQ switches)

เมื่อเราใช้สวิตช์ครอสบาร์ เรามีความต้องการอัลกอริทึมในการจัดลำดับเซลล์ส่งผ่านสวิตช์ในแต่และไทม์สล็อต และทำการตัดสินใจว่าจะเชื่อมต่ออินพุตใดเข้ากับเอาต์พุตใดของสวิตช์ เพื่อจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์ ที่จุดเริ่มต้นของในแต่ละเซลล์ไทม์ ตัวจัดลำดับเซลล์ส่งผ่านสวิตช์ (Scheduler) จะทำการตรวจสอบเซลล์ที่รอคิวที่อินพุตที่จะทำการแข่งขัน และกำหนดการเชื่อมต่อระหว่างอินพุตและเอาต์พุตของสวิตช์ ที่ไม่ทำให้เซลล์เกิดการชนกัน ในทางคณิตศาสตร์ก็คือการแมตช์บนกราฟที่มี N จุดยอด (N คือจำนวนพอร์ตของสวิตช์) กระบวนการนี้มีความสลับซับซ้อนมากในการนำไปสร้างเป็นฮาร์ดแวร์ ทั้งยังต้องทำการประมวลผลอย่าง รวดเร็ว ดังตัวอย่างเช่น เพื่อที่จะให้ได้ค่าทรูพุตสูงสุดถึง 100เปอร์เซ็นต์ จะต้องใช้อัลกอริทึม maximum weight bipartite matching ซึ่งเป็นอัลกอริทึมที่คำนึงถึงลักษณะการรอคิวของเซลล์ในบัฟเฟอร์ด้วย

อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ส่วนใหญ่ใช้การประมาณค่าการแมตช์ขนาดสูงที่สุด (maximum size matching) อัลกอริทึมเหล่านี้พยายามที่จะให้จำนวนการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตมีขนาดสูงสุดในแต่ละไทม์สล็อตซึ่งจะให้ค่าการจัดสรรแบนด์วิทด์ในขณะนั้นสูงสุดการแมตช์ขนาดสูงสุดสำหรับกราฟไบพาร์ไทท์ หาได้จากการแก้ปัญหาการไหลในโครงข่ายสมมูล และเราเรียกอัลกอริทึมนี้ว่า maxsize ซึ่งให้ประสิทธิภาพลู่เข้าสู่ค่า $o(n)^{2/5}$

สำหรับปัญหาในการใช้อัลกอริทึมแบบนี้ก็คือ ถึงแม้ว่าจะการันตีว่าจะหาค่าการแมตช์ขนาดสูงสุดได้ แต่ในการนำไปประยุกต์ใช้งานนั้น มีความสลับซับซ้อนมากเกินไปในการนำไปสร้างเป็นฮาร์ดแวร์ และใช้เวลาในการประมวลผลนานเกินไป

คำถามมีอยู่ว่า อัลกอริทึมขนาดสูงสุดให้ค่าทรูพุตของสวิตช์มากที่สุดจริงหรือไม่ คำตอบคือว่า ไม่ใช่ และนอกจากนี้ อัลกอริทึมนี้ ยังทำให้เซลล์บางเซลล์รอคิวอยู่ในบัฟเฟอร์เป็นเวลานานมาก โดยไม่มีทีท่าว่าจะได้รับบริการ (starvation) และในกรณีที่ทราฟฟิกเป็นประเภทไม่ยูนิฟอร์ม ก็ไม่สามารถให้ค่าทรูพุตที่สูงมากได้ เป็นเพราะว่า มันไม่ได้พิจารณาจำนวนเซลล์ที่รอคิวอยู่ในบัฟเฟอร์ (Longest Queue First) หรือเวลาที่เซลล์ต่าง ๆ รออยู่ในคิวกว่าที่จะได้รับบริการจากสวิตช์ (Oldest Cell First)

ในทางปฏิบัตินั้น สำหรับสวิตช์ที่มีสมรรถนะสูงนั้น เราจึงมีความต้องการอัลกอริทึมที่มีคุณสมบัติดังต่อไปนี้

High Throughput อัลกอริทึมควรมีจำนวนเซลล์ที่รอคิวอยู่ในบัฟเฟอร์ต่ำ ในกรณีที่มีการรอคิวแบบ VOQ ซึ่งจะทำให้อัลกอริทึมสามารถสนับสนุนให้ได้ค่าทรูพุตถึง 100เปอร์เซ็นต์ ในแต่ละพอร์ตอินพุตและเอาต์พุต

Starvation Free อัลกอริทึมที่ดี ไม่ควรมีลักษณะที่มีเซลล์รออยู่ในคิวเป็นเวลานาน โดยไม่มีกำหนดเวลาว่าจะได้รับบริการเมื่อใด

Fast เพื่อให้ค่าแบนด์วิดท์ของสวิตช์มีค่าสูงสุด เป็นเรื่องสำคัญที่อัลกอริทึมจะต้องไม่ทำให้เกิดปัญหาคอขวด (bottleneck) โดยอัลกอริทึมจะต้องประมวลผลได้อย่างรวดเร็วในการหาการเชื่อมต่อระหว่าง อินพุตกับเอาต์พุต

Simple to implement ถ้าอัลกอริทึมสามารถประมวลผลได้อย่างรวดเร็ว ก็สามารถนำไปสร้างเป็นฮาร์ดแวร์ที่มีวัตถุประสงค์พิเศษได้โดยรวมอยู่ในชิปเดียว

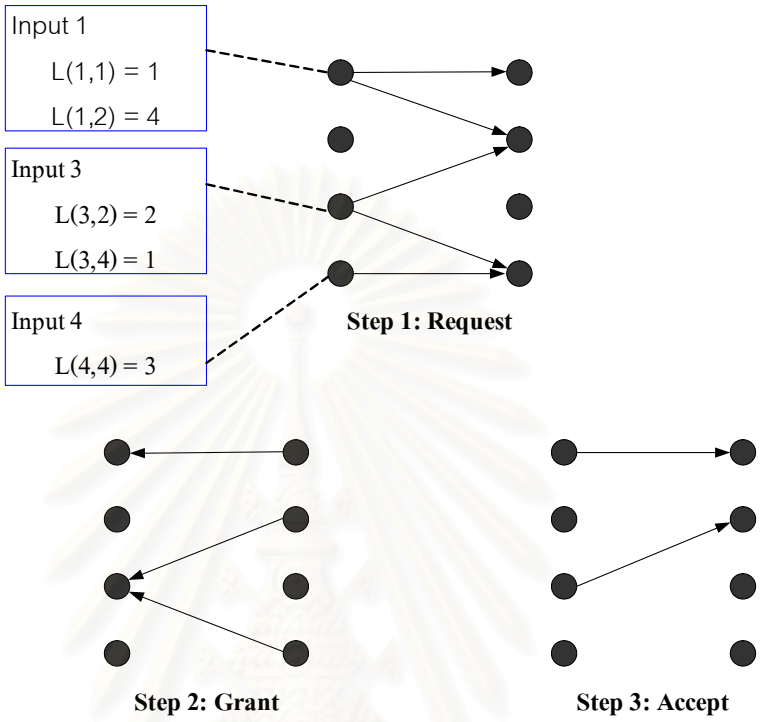
3.1 Parallel Iterative Matching

อัลกอริทึม PIM ได้รับการพัฒนาขึ้นมาจากศูนย์วิจัยระบบของ DEC สำหรับสวิตช์ขนาด 16 พอร์ตแบบ AN 2 อัลกอริทึม PIM ใช้หลักการเลือกเซลล์ที่จะส่งผ่านสวิตช์แบบสุ่มเพื่อหลีกเลี่ยงปัญหา Starvation และ เพื่อลดจำนวนการวนซ้ำซึ่งทำให้ได้การเชื่อมต่อขนาดมากที่สุด (maximal sized match) การเชื่อมต่อขนาดมากที่สุด ได้จากการเพิ่มการเชื่อมต่อในแต่ละรอบการวนซ้ำ โดยไม่ต้องยกเลิกกระบวนการเชื่อมต่อในรอบก่อนหน้านั้น โดยทั่วไปแล้วการเชื่อมต่อขนาดมากสุดมีค่าน้อยกว่าการเชื่อมต่อขนาดสูงที่สุด (maximum sized match) แต่นำไปสร้างเป็นฮาร์ดแวร์ได้ง่าย อัลกอริทึม PIM สามารถประมวลผลได้อย่างรวดเร็วในการลู่เข้าสู่ค่าการเชื่อมต่อขนาดมากที่สุดในการวนซ้ำ โดยที่ในการวนซ้ำแต่ละรอบประกอบไปด้วยขั้นตอน 3 ขั้นตอน เริ่มต้นด้วย ทุก ๆ อินพุตและเอาต์พุตของสวิตช์ จะถูกนำมาพิจารณาเชื่อมต่อกัน และเมื่ออินพุตและเอาต์พุตใดเชื่อมต่อกันแล้ว ในแต่ละรอบการวนซ้ำจะไม่นำมาพิจารณาเชื่อมต่อในรอบถัดไป แสดงดังรูปที่ 3.1 ในการวนซ้ำแต่ละรอบจะมีขั้นตอน ดังต่อไปนี้

ขั้นตอนที่ 1 **Request** ในแต่ละอินพุตของสวิตช์ที่มีเซลล์รออยู่ในคิวและยังไม่ได้รับการเชื่อมต่อจะส่ง คำร้องขอ (request) ไปยังทุก ๆ เอาต์พุต ที่มีเซลล์ที่อินพุตจะไปยังเอาต์พุตนั้น

ขั้นตอนที่ 2 **Grant** ถ้าแต่ละเอาต์พุตที่ยังไม่ได้รับการเชื่อมต่อ เมื่อได้รับคำร้องขอแล้วจะส่ง คำยินยอม (grant) โดยทำการเลือกแต่ละอินพุตที่ร้องขอมาแบบสุ่ม เพียงอินพุตเดียว

ขั้นตอนที่ 3 **Accept** ถ้าอินพุตใดได้รับคำยินยอมให้ส่งจากเอาต์พุตแล้ว จะเลือกตอบรับคำยินยอม (accept) จากเอาต์พุตแบบสุ่มเพียงเอาต์พุตเดียว เพื่อทำการเชื่อมต่อระหว่างอินพุตนั้นกับเอาต์พุตที่เลือก



รูปที่ 3.1 ตัวอย่างขั้นตอนทั้ง 3 ขั้นตอนของอัลกอริทึม PIM ที่ไม่มีการวนซ้ำจะเห็นว่าอินพุตที่ 4 กับเอาต์พุตที่ 4 ไม่ได้รับการเชื่อมต่อ แม้ว่าจะไม่เกิดการชนกันของเซลล์กับการเชื่อมต่ออื่น ๆ เลย แต่ก็จะได้รับการเชื่อมต่อถ้ามีการวนซ้ำในรอบที่ 2 [11]

สรุปได้ว่า ตัวตัดสินใจ (arbiter) ของเอาต์พุต มีอิสระในการที่จะเลือกคำร้องขอเพียงอันเดียว จากคำร้องขอของอินพุตต่าง ๆ ที่เข้าแข่งขันโดยวิธีการเลือกแบบสุ่ม ทำให้เกิดผลลัพธ์ 3 ประการ คือ ประการแรกในแต่ละรอบของการวนซ้ำจะสามารถพิจารณาให้เชื่อมต่อกันถึงประมาณ 3/4 ของการเชื่อมต่อที่เป็นไปได้ทั้งหมด ซึ่งอัลกอริทึมนี้จะลู่เข้าสู่ค่าการเชื่อมต่อขนาดมากที่สุดเมื่อจำนวนรอบของการวนซ้ำมีค่าเฉลี่ยประมาณ o(logN) ประการที่ 2 สามารถยืนยันได้ว่าทุก ๆ คำร้องขอจากอินพุตของสวิตช์ ในที่สุดจะต้องได้รับการยอมรับ (grant) จากเอาต์พุต ซึ่งเป็นสิ่งที่ยืนยันได้ว่าจะไม่มีเซลล์ใดที่รออยู่ในคิวจะเกิดภาวะ starvation ประการที่ 3 มันหมายความว่า ไม่จำเป็นต้องใช้หน่วยความจำในการเก็บขบวนการในการเชื่อมต่อที่เกิดขึ้นในรอบที่ผ่านมา ในแต่ละจุดเริ่มต้นของไทม์สล็อต กระบวนการเชื่อมต่อได้เริ่มขึ้นโดยไม่ขึ้นกับกระบวนการเชื่อมต่อในรอบที่ผ่านมา และเราสามารถทำความเข้าใจได้ง่ายขึ้น จากการวิเคราะห์สมรรถนะของอัลกอริทึม

โดยตรง ทำให้ไม่ต้องพิจารณาถึงสถานะของสวิตช์ที่เปลี่ยนแปลงตามเวลาเพียงแต่จะต้องพิจารณาถึงการครอบครองคิวที่อินพุตของสวิตช์เท่านั้น [12]

แต่การเลือกแบบสุ่มนำมาซึ่งปัญหา ประการแรกทำให้เกิดความยุ่งยากและใช้ต้นทุนสูงในการนำไปสร้างเป็นสวิตช์ความเร็วสูง เนื่องจากตัวตัดสินใจแต่ละตัวจะต้องทำการเลือกอินพุตแบบสุ่ม ซึ่งจำเป็นต้องใช้เวลา ประการที่สองเกิดความไม่เสมอภาคในการเลือก ในกรณีที่สวิตช์มีทราฟฟิกที่อินพุตเป็นแบบไม่ยูนิฟอร์ม และประการสุดท้าย อัลกอริทึม PIM ไม่เหมาะสมในการนำไปใช้ถ้าใช้การวนซ้ำเพียงรอบเดียว และนอกจากนี้ ขีดจำกัดของค่าทรูพุตสูงสุดของสวิตช์มีค่าประมาณ 63 เปอร์เซ็นต์ ซึ่งสูงกว่าอัลกอริทึมี FIFO เพียงเล็กน้อยทั้งนี้ เนื่องมาจากความน่าจะเป็นที่แต่ละอินพุตจะไม่ได้รับการยอมรับจากเอาต์พุตคือ $((N-1)/N)^N$ และเมื่อ N เพิ่มขึ้น ค่าทรูพุตมีแนวโน้มลู่เข้าสู่ค่า $1 - 1/e \approx 63$ เปอร์เซ็นต์ ถึงแม้ว่าอัลกอริทึม PIM จะทำการเชื่อมต่อได้มากขึ้น หลังจากการวนซ้ำหลาย ๆ รอบ แต่ก็ต้องใช้เวลามาก ซึ่งทำให้มีผลกระทบต่อความเร็วในการทำงานของสวิตช์ ดังนั้น เราจึงมีความต้องการอัลกอริทึมที่จะทำงานได้ดีแม้ว่าจะใช้การวนซ้ำเพียงรอบเดียว

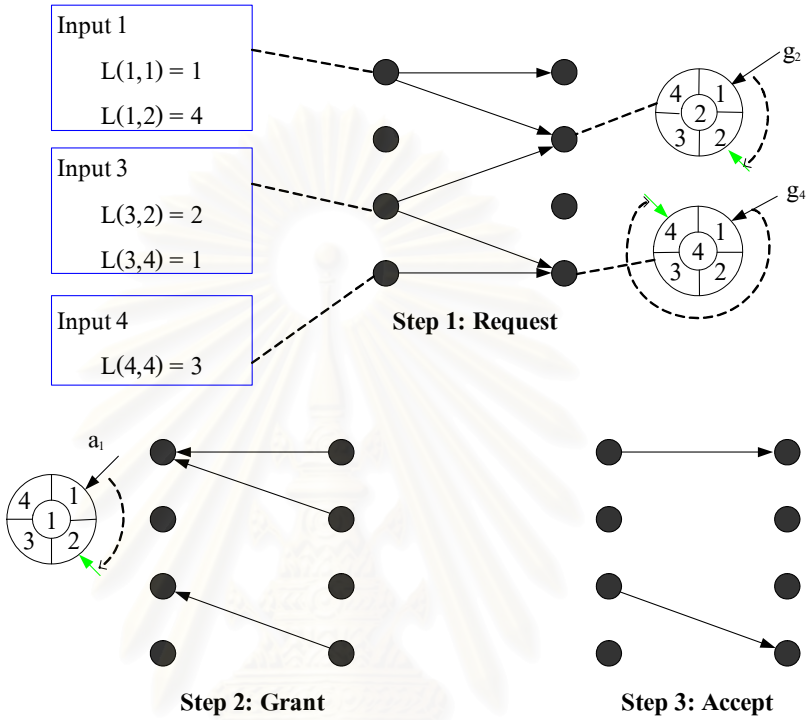
3.2 อัลกอริทึม Round-Robin Matching (RRM)

อัลกอริทึม RRM เป็นอัลกอริทึมที่ใช้การวนรอบอินพุตแต่ละอินพุตของสวิตช์แบบที่ง่ายที่สุด ดังนั้นจะทำการเปรียบเทียบอะเรย์สองมิติของตัวตัดสินใจของอัลกอริทึม RRM โดยที่แต่ละเซลล์ถูกจัดลำดับโดยวิธีการวนรอบแต่ละอินพุตและการวนรอบแต่ละเอาต์พุตอัลกอริทึม RRM นั้นสมรรถนะการทำงานไม่ดีนัก แต่ก็สามารถแก้ปัญหาความซับซ้อนและความไม่เสมอภาค (unfairness) ในการเลือกการเชื่อมต่อ ที่เกิดขึ้นในอัลกอริทึม PIM ได้ นอกจากนี้อัลกอริทึม RRM ยังนำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายกว่า และยังทำงานได้เร็วกว่าการใช้ตัวตัดสินใจเลือกเซลล์แบบสุ่มในอัลกอริทึม PIM อัลกอริทึม RRM ก็คล้ายคลึงกับแบบ PIM ประกอบด้วยขั้นตอนจำนวน 3 ขั้นตอน ดังแสดงในรูปที่ 3.2 ซึ่งทั้ง 3 ขั้นตอน มีดังต่อไปนี้

ขั้นตอนที่ 1 **Request** แต่ละอินพุตที่มีเซลล์รออยู่ในคิวและต้องการจะไปที่เอาต์พุตนั้นส่งคำร้องขอ ไปยังเอาต์พุตนั้น

ขั้นตอนที่ 2 **Grant** เมื่อแต่ละเอาต์พุตได้รับคำร้องขอ จะทำการเลือกเพียงคำร้องขอเดียวที่ปรากฏว่า มีค่า priorrity ของอินพุตสูงสุด โดยทำการวนรอบแต่ละอินพุตของสวิตช์ โดยมีตัวชี้ g_i ทำการเลือกคำร้องขอ แล้วส่งคำยินยอมไปยังอินพุตนั้น และชี้ที่อินพุตถัดไปโดยกำหนดให้เป็นอินพุต ที่มีค่า priorrity สูงสุด ตามลำดับ (modulo N) จากอินพุตที่ 1 ไป 2 ไปจนถึง N และวนซ้ำมาที่อินพุตที่ 1 ใหม่ เป็นวงรอบ

ขั้นตอนที่ 3 **Accept** เมื่อแต่ละอินพุตได้รับคำยินยอม จะเลือกตอบรับคำยินยอมจากเอาต์พุต โดยเลือกเอาต์พุตที่มีค่า priorrity สูงสุด โดยมีตัวชี้ a_i วนรอบแต่ละเอาต์พุต และชี้เอาต์พุตถัดไป (modulo N) และกำหนดให้เอาต์พุตนั้นมีค่า priorrity สูงสุดเมื่อได้เลือกเอาต์พุตนั้นและทำการเชื่อมต่อกับอินพุตแล้ว



รูปที่ 3.2 แสดงตัวอย่างการทำงานทั้ง 3 ขั้นตอน ของอัลกอริทึม RRM [9]

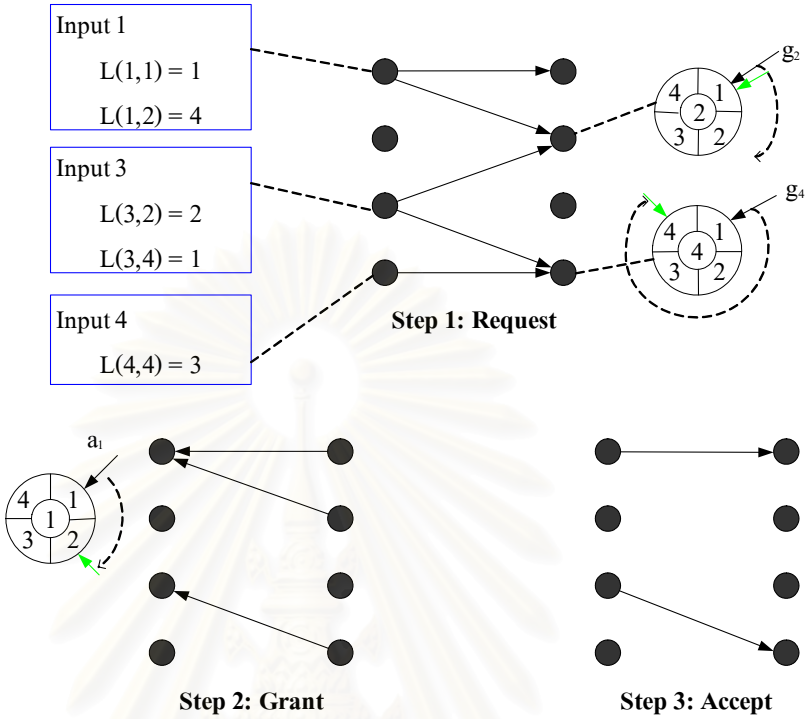
3.3 อัลกอริทึม iSLIP

อัลกอริทึม iSLIP ใช้การวนรอบอินพุตแต่ละอินพุต (round robin) เพื่อทำการตัดสินใจที่จะจัดลำดับ อินพุตและเอาต์พุตในแต่ละรอบของการวนซ้ำ คุณลักษณะสำคัญของอัลกอริทึม iSLIP ก็คือสามารถนำไปสร้างเป็นฮาร์ดแวร์ได้ง่าย และทำงานได้ดีที่ความเร็วสูง จาก [9] จะเห็นได้ว่าสมรรถนะของ iSLIP สำหรับทราฟฟิกประเภทยูนิฟอร์มมีค่าสูงมาก ถึงแม้ว่าจะใช้การวนซ้ำเพียงรอบเดียว ก็สามารถให้ค่าทรูพุตสูงถึง 100 เปอร์เซ็นต์ และเนื่องจากมีการวนซ้ำ จึงทำให้ตัวตัดสินใจของอัลกอริทึม iSLIP มีแนวโน้มไม่ซิงโครไนซ์กับตัวตัดสินใจตัวอื่น

อัลกอริทึม iSLIP แบบไม่วนซ้ำ

อัลกอริทึม iSLIP ได้รับการพัฒนาต่อมาจากอัลกอริทึม RRM โดยอัลกอริทึม iSLIP จะไม่ทำการเลื่อน ค่าตัวชี้ g_i ถ้าคำยินยอมที่ส่งไปให้อินพุตไม่ได้รับการตอบรับจากอินพุตนั้น จะเห็น

ได้ว่าอัลกอริทึม iSLIP ก็คล้าย กันกับอัลกอริทึม RRM เว้นแต่เงื่อนไขในการเลื่อนค่าของตัวชี้ g_i ในขั้นตอนที่ 2 ที่เปลี่ยนแปลงไป ดังต่อไปนี้



รูปที่ 3.3 แสดงตัวอย่างการทำงานของอัลกอริทึม iSLIP

ขั้นตอนที่ 2 **Grant** เมื่อแต่ละเอาต์พุตได้รับคำร้องขอ จะทำการเลือกเพียงคำร้องขอเดียว ที่ปรากฏว่า มีค่า priority สูงสุด โดยทำการวนรอบแต่ละอินพุต โดยมีตัวชี้ g_i ทำการเลือก แล้วส่งคำยินยอมไปยัง อินพุตนั้น และจะชี้อินพุตถัดไป โดยกำหนดให้อินพุตนั้นมีค่า priorrity สูงสุดในรอบถัดไป ก็ต่อเมื่อ คำยินยอมนั้นได้รับการตอบรับ เพื่อทำการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตในขั้นตอนที่ 3 แล้ว

คุณสมบัติของ iSLIP ที่ไม่มีการวนซ้ำ มีดังต่อไปนี้

คุณสมบัติข้อที่ 1 ให้ค่า priorrity ต่ำสุด สำหรับการเชื่อมต่อที่เป็นปัจจุบันที่สุด ทั้งนี้เนื่องมาจากเมื่อ ตัวตัดสินใจเลื่อนค่าของตัวบ่งชี้อินพุตที่ตอบรับคำยินยอมของเอาต์พุตใด และมีการเชื่อมต่อระหว่าง อินพุตและเอาต์พุตนั้นแล้ว ตัวบ่งชี้ g_i จะให้ค่า priorrity ของอินพุตนั้นต่ำสุด คือ ถ้าอินพุต i เชื่อมต่อกับเอาต์พุต j เรียบร้อยแล้ว ทั้ง a_i และ g_j จะเลื่อนค่าบ่งชี้ไปหนึ่งค่า (modulo N) ทำให้การเชื่อมต่อระหว่างอินพุต i และเอาต์พุต j มีค่า priorrity ต่ำสุด ในการเชื่อมต่อในไทม์สล็อตถัดไป

คุณสมบัติข้อที่ 2 ไม่มีการเชื่อมต่อใดเกิดกรณี starvation เนื่องจากแต่ละอินพุตจะส่งคำร้องขอต่อเนื่องกันไป จนกว่าได้รับการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตของสวิตช์

คุณสมบัติข้อที่ 3 ในกรณีที่ทราฟฟิกหนาแน่น ทุก ๆ คิวที่จะไปยังเอาต์พุตเดียวกัน จะให้ค่าทรูพุตที่เหมือนกัน ซึ่งเป็นผลที่ตามมาจากคุณสมบัติข้อที่ 2 เนื่องจากตัวบ่งชี้ของเอาต์พุต เลื่อนค่าอินพุตในตัวบ่งชี้ต่อเนื่องไปเป็นลำดับ(modulo N) จึงทำให้แต่ละอินพุตให้ค่าทรูพุตที่เหมือนกัน

อัลกอริทึม iSLIP แบบวนซ้ำ

การที่มีการวนซ้ำมากกว่าหนึ่งรอบ จะทำให้อัลกอริทึม iSLIP ให้ค่าการแมตซ์ขนาดมากที่สุดที่มีขนาดมากขึ้น เนื่องจากในแต่ละรอบการวนซ้ำ ได้เพิ่มการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตที่ไม่ได้ทำการเชื่อมต่อในรอบก่อน ๆ จึงเป็นสิ่งที่เราสามารถจะคาดการณ์ได้ว่า สมรรถนะของสวิตช์จะต้องดีขึ้นเมื่อเพิ่มจำนวนรอบให้มากขึ้นโดยในการวนซ้ำในรอบต่อไป อินพุตที่เชื่อมต่อกับเอาต์พุตในรอบปัจจุบันแล้ว จะไม่ถูกนำมาพิจารณาเชื่อมต่ออีกในรอบถัดไป หมายเหตุ เพื่อให้เกิดความเสมอภาค (fairness) ในการเลือกเซลล์ส่งผ่านสวิตช์ในแต่ละอินพุตจึงจะทำการเลื่อนค่าตัวบ่งชี้เฉพาะแต่ในรอบแรกของการวนซ้ำเท่านั้น
คุณสมบัติของ iSLIP ที่มีการวนซ้ำ มีดังต่อไปนี้

คุณสมบัติข้อที่ 1 ให้ค่า priorrity ต่ำสุดสำหรับการเชื่อมต่อที่เป็นปัจจุบันที่สุดในการวนซ้ำรอบแรกใน ไทม์สล็อตถัดไป

คุณสมบัติข้อที่ 2 ไม่มีการเชื่อมต่อใดเกิดกรณี starvation เพราะจะได้รับการเชื่อมต่อในรอบถัด ๆ ไป จนกว่าทุก ๆ อินพุตที่มีเซลล์อยู่ในหัวคิว จะได้รับการเชื่อมต่อสำเร็จ

คุณสมบัติข้อที่ 3 ในกรณีที่มีการวนซ้ำมากกว่าหนึ่งรอบ ทำให้ในกรณีที่มีทราฟฟิกหนาแน่น จะทำให้แต่ละคิวที่จะไปยังเอาต์พุตเดียวกัน มีค่าทรูพุตที่แตกต่างกัน

คุณสมบัติข้อที่ 4 อัลกอริทึมจะมีจำนวนรอบการวนซ้ำมีค่าเท่ากับ N รอบ เพราะว่าจะต้องมีอย่างน้อยหนึ่งการเชื่อมต่อที่ได้กระทำในแต่ละรอบ ถ้าไม่มีการเชื่อมต่อในรอบใดก็จะไม่มีการวนซ้ำในรอบถัดไป ทำให้จำนวนการวนซ้ำลู่เข้าสู่ค่า N

คุณสมบัติข้อที่ 5 อัลกอริทึมไม่มีความจำเป็นที่จะต้องลู่เข้าการเชื่อมต่อขนาดสูงที่สุด (maximum sized match) แต่ทางที่ดีที่สุดก็คือ การหาค่าการเชื่อมต่อขนาดมากที่สุด (maximul sized match) โดยไม่จำเป็นต้องยกเลิกการเชื่อมต่อที่ทำไปแล้วในการวนซ้ำในรอบก่อน

3.4 อัลกอริทึม Prioritized iSLIP

การใช้งานหลายอย่างในปัจจุบันใช้ชั้นของทราฟฟิกที่มีระดับความสำคัญต่างกัน อัลกอริทึม iSLIP สามารถถูกขยายไปยังความต้องการที่ระดับความสำคัญหลายๆระดับความสำคัญได้ ซึ่งเราจะเรียกว่าอัลกอริทึมนี้ว่า อัลกอริทึม Prioritized iSLIP

ในอัลกอริทึม Prioritized iSLIP แต่ละอินพุตยังคงรักษาคิว FIFO ที่แยกสำหรับแต่ละระดับความสำคัญ และสำหรับแต่ละเอาท์พุต นั่นหมายความว่า สำหรับสวิตซ์ขนาด N×N ที่มีระดับความสำคัญ P ระดับ แต่ละอินพุตจะมีคิว P×N FIFO's คิว เราจะให้ชื่อคิวระหว่างอินพุต i และเอาท์พุต j ที่ระดับความสำคัญ l ว่า $Q_l(i,j)$ ที่ซึ่ง $1 \leq i, j \leq N, 1 \leq l \leq P$ ดังที่กล่าวแล้วข้างต้น มีเพียงแค่หนึ่งเซลล์สามารถมาถึง ดังนั้นอัลกอริทึมนี้ไม่ต้องการกระบวนการเพิ่มความเร็วโดยอินพุต

อัลกอริทึม Prioritized iSLIP ให้ความสำคัญอย่างเคร่งครัดแก่การร้องขอที่มีความสำคัญสูงสุด สิ่งนี้หมายความว่า $Q_l(i,j)$ จะถูกบริการถ้าคิว $Q_m(i,j)$, $1 \leq m \leq P$ ทั้งหมดยังว่างอยู่ [9]

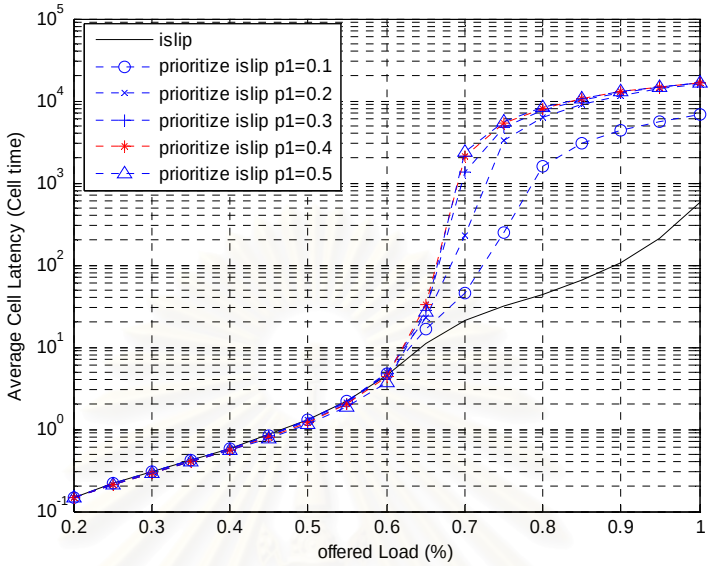
Prioritized iSLIP algorithm มีขั้นตอนการทำงานของอัลกอริทึมคือ

ขั้นที่1 : **Request** เมื่อมีเซลล์เข้ามาในบัฟเฟอร์ของอินพุต อินพุตดังกล่าวจะจัดเซลล์นั้นไว้ยัง VOQ ของลำดับความสำคัญของเซลล์นั้นๆ จากนั้นอินพุตแต่ละตัวจะพิจารณาว่ามีเซลล์ใน VOQ ที่ต้องการจะส่งหรือไม่ ถ้ามี ก็จะเลือกเซลล์ที่มีความสำคัญมากที่สุดที่จะส่งไปยังแต่ละเอาท์พุต และทำการส่ง request และลำดับความสำคัญของเซลล์ที่ต้องการจะส่งไปยังเอาท์พุตแต่ละตัวที่ต้องการ

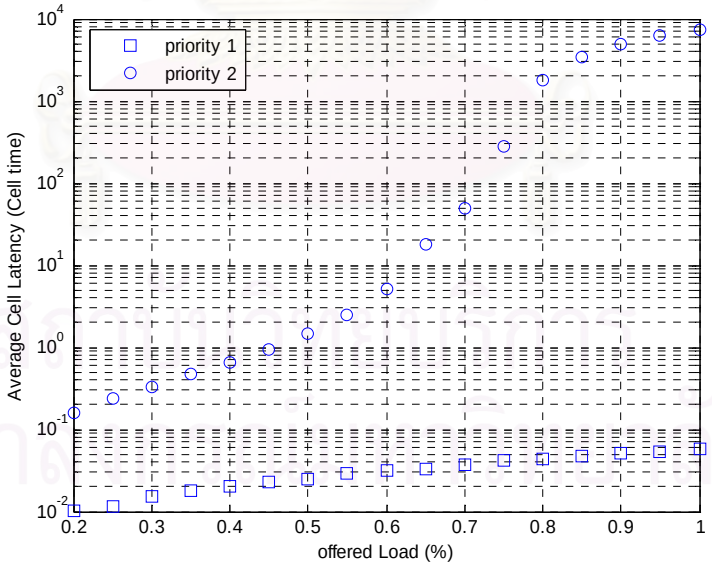
ขั้นที่ 2 : **Grant** เมื่อเอาท์พุตได้รับ request และลำดับความสำคัญจากอินพุตแล้ว เอาท์พุตตัวดังกล่าวจะพิจารณาเลือก อินพุตที่มีลำดับความสำคัญมากที่สุดที่เข้ามาเปรียบเทียบกัน แล้วเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ จากนั้นเอาท์พุตจะส่ง grant และลำดับความสำคัญของเซลล์ดังกล่าว ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว

ขั้นที่ 3 : **Accept** เมื่ออินพุตได้รับ grant และลำดับความสำคัญจากเอาท์พุตแล้ว อินพุตตัวดังกล่าวจะพิจารณาเลือก เอาท์พุตที่มีลำดับความสำคัญมากที่สุดที่เข้ามาเปรียบเทียบกัน แล้วเลือกอินพุตตามตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ จากนั้นอินพุตจะส่ง accept ไปยังเอาท์พุตตัวที่เลือก และทำการส่งเซลล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้

ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล



รูปที่ 3.4 แสดงผลกระทบของลำดับความสำคัญของเซลล์เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เพิ่มขึ้นที่มีต่อ Prioritized iSLIP algorithm



รูปที่ 3.5 แสดงความสัมพันธ์ระหว่างเซลล์ที่มีลำดับความสำคัญต่างกัน ใน Prioritized iSLIP algorithm เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เป็น 10%

จากรูปที่ 3.4 พบว่าการทำงานของอัลกอริทึม iSLIP นั้น หากทราฟฟิกที่เข้ามาเป็นแบบมีหลายลำดับความสำคัญจะใช้อัลกอริทึมแบบ Prioritzed iSLIP ซึ่งจะทำให้ประสิทธิภาพโดยรวมของระบบซึ่งแสดงด้วยค่าเวลาประวิงลดลงไปอย่างมาก โดยเฉพาะเมื่อเซลล์มีความหนาแน่นสูง และพบอีกว่า เมื่อเซลล์ที่มีลำดับความสำคัญสูงมีอัตราส่วนเพิ่มขึ้นจะยิ่งทำให้ประสิทธิภาพของอัลกอริทึมนี้ลดลงยิ่งขึ้นเนื่องจากค่าเวลาประวิงให้บริการของเซลล์ที่มีลำดับความสำคัญต่ำจะสูงมาก เนื่องจากต้องรอให้เซลล์ที่มีลำดับความสำคัญสูงกว่าได้รับบริการจนเสร็จหมดเสียก่อน ดังนั้นยิ่งมีเซลล์ที่มีลำดับความสำคัญสูงปริมาณมากยิ่งทำให้ค่าเวลาประวิงของเซลล์ที่ลำดับความสำคัญต่ำสูงมากตามไปด้วย

จากรูปที่ 3.5 พบว่าเมื่อความหนาแน่นของทราฟฟิกเพิ่มขึ้นจะทำให้ค่าเวลาประวิงของเซลล์ที่มีลำดับความสำคัญต่ำสูงขึ้นอย่างมาก ในขณะที่ค่าเวลาประวิงของเซลล์ที่มีลำดับความสำคัญสูงกว่า เพิ่มขึ้นเพียงเล็กน้อยเท่านั้น เนื่องจากเซลล์ที่มีลำดับความสำคัญต่ำต้องรอให้เซลล์ที่มีลำดับความสำคัญสูงกว่าได้รับบริการจนเสร็จหมดเสียก่อน ดังนั้นยิ่งมีเซลล์ที่มีลำดับความสำคัญสูงปริมาณมากยิ่งทำให้ค่าเวลาประวิงของเซลล์ที่ลำดับความสำคัญต่ำสูงมากตามไปด้วย ในขณะที่ข้อมูลที่มีลำดับความสำคัญสูงกว่าจะได้รับบริการในทันที จึงทำให้ค่าเวลาประวิงไม่ได้เพิ่มขึ้นมากนัก

3.5 อัลกอริทึมการจัดกำหนดการสำหรับการจัดสรรบริการในเซลล์สวิตช์ที่พิจารณาลำดับความสำคัญของเซลล์ที่นำเสนอ

อัลกอริทึมนี้จะพิจารณาลำดับความสำคัญของเซลล์โดยการสุ่มแบบถ่วงน้ำหนักหรือให้น้ำหนักในการเลือกที่เท่ากัน ขึ้นอยู่กับว่าจะควบคุมที่ขั้นตอนไหนของอัลกอริทึม iSLIP เดิม อัลกอริทึมใหม่มีทั้งหมด 3 อัลกอริทึมได้แก่

3.5.1 อัลกอริทึมการควบคุมที่ request

3.5.2 อัลกอริทึมการควบคุมที่ grant

3.5.3 อัลกอริทึมการควบคุมที่ accept

3.5.1 **อัลกอริทึมการควบคุมที่ request**

ขั้นตอนการทำงานของอัลกอริทึม

ขั้นที่1 : request เมื่อมีเซลล์เข้ามาในบัฟเฟอร์ของอินพุต อินพุตดังกล่าวจะจัดเซลล์นั้นไว้ยัง VOQ ของลำดับความสำคัญของเซลล์นั้นๆ จากนั้นอินพุตแต่ละตัวจะพิจารณาว่ามีเซลล์ใน VOQ ทั้งสองลำดับความสำคัญที่ต้องการจะส่งหรือไม่ ถ้ามี ก็จะพิจารณาว่ามีกี่ลำดับความสำคัญที่จะส่งไปยังเอาท์พุตตัวเดียวกัน หากมีลำดับเดียวก็ทำการส่ง request และลำดับความสำคัญของเซลล์ที่ต้องการจะส่งไปยังเอาท์พุตที่ต้องการ แต่หากมีเซลล์มากกว่าหนึ่งลำดับความสำคัญต้องการจะส่งไปยังเอาท์พุตเดียวกันก็จะเลือกลำดับความสำคัญโดยการสุ่มแบบถ่วงน้ำหนักของแต่ละลำดับความสำคัญ และทำการส่ง request และลำดับความสำคัญของเซลล์ที่เลือกไปยังเอาท์พุตที่ต้องการ

ขั้นที่ 2 : grant เมื่อเอาท์พุตได้รับ request และลำดับความสำคัญจากอินพุตแล้ว เอาท์พุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่มี เอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ จากนั้นเอาท์พุตจะส่ง grant และลำดับความสำคัญของเซลล์ที่เลือก ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว เอาท์พุตจะทำการสุ่มเลือกลำดับความสำคัญของเซลล์ที่จะให้บริการโดยให้น้ำหนักในการเลือกที่เท่ากัน จากนั้นเอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่เลือก และส่ง grant พร้อมด้วยลำดับความสำคัญของเซลล์ดังกล่าว ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว

ขั้นที่ 3 : accept เมื่ออินพุตได้รับ grant และลำดับความสำคัญจากเอาท์พุตแล้ว อินพุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่อินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก และทำการส่งเซลล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว อินพุตจะทำการสุ่มเลือกลำดับความสำคัญโดยให้น้ำหนักในการเลือกที่เท่ากัน จากนั้นอินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่

เลือกได้ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก จากนั้นจะทำการส่งเซลล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล

3.5.2 อัลกอริทึมการควบคุมที่ grant

เราจะไม่ควบคุมในขั้นตอน request เนื่องจากเราต้องการให้เซลล์ที่จะเข้าไปสู่การตัดสินใจในขั้นตอนต่อไปมีจำนวนมากที่สุดเท่าที่จะเป็นไปได้เนื่องจากหากควบคุมสัดส่วนการให้บริการให้เท่ากันที่ขั้นตอน request จะมีเซลล์ที่ไม่สามารถผ่านเข้ามาในขั้นตอน grant อยู่จำนวนหนึ่ง ซึ่งเป็นการตัดโอกาสในการจัดหาเส้นทางที่เหมาะสมของเซลล์เหล่านั้น ซึ่งจะทำให้มีโอกาสได้ทรูพุตที่สูงลดน้อยลง

ขั้นตอนการทำงานของอัลกอริทึม

ขั้นที่1 : request เมื่อมีเซลล์เข้ามาในบัฟเฟอร์ของอินพุต อินพุตดังกล่าวจะจัดเซลล์นั้นไว้ยัง VOQ ของลำดับความสำคัญของเซลล์นั้นๆ จากนั้นอินพุตแต่ละตัวจะพิจารณาว่ามีเซลล์ใน VOQ ที่ต้องการจะส่งหรือไม่ ถ้ามี ก็จะทำการส่ง request และลำดับความสำคัญของเซลล์ที่ต้องการจะส่งไปยังเอาท์พุตที่ต้องการโดยไม่คำนึงถึงลำดับความสำคัญของเซลล์

ขั้นที่ 2 : grant เมื่อเอาท์พุตได้รับ request และลำดับความสำคัญจากอินพุตแล้ว เอาท์พุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่มี เอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ จากนั้นเอาท์พุตจะส่ง grant และลำดับความสำคัญของเซลล์ที่เลือก ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว เอาท์พุตจะทำการสุ่มเลือกลำดับความสำคัญของเซลล์ที่จะให้บริการโดยการสุ่มแบบถ่วงน้ำหนักของแต่ละลำดับความสำคัญ จากนั้นเอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่เลือก และส่ง grant พร้อมด้วยลำดับความสำคัญของเซลล์ดังกล่าว ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว

ขั้นที่ 3 : accept เมื่ออินพุตได้รับ grant และลำดับความสำคัญจากเอาท์พุตแล้ว อินพุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่ อินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก และทำการส่งเซล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว อินพุตจะทำการสุ่มเลือกลำดับความสำคัญโดยให้น้ำหนักในการเลือกที่เท่ากัน จากนั้นอินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่เลือกได้ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก จากนั้นจะทำการส่งเซล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล

3.5.3 อัลกอริทึมการควบคุมที่ accept

เช่นเดียวกันกับอัลกอริทึมการควบคุมที่ขั้นตอน grant เราจะไม่ควบคุมสัดส่วนการให้บริการในขั้นตอน request ด้วยเหตุผลเช่นเดียวกัน แต่เราจำเป็นต้องควบคุมสัดส่วนการให้บริการที่ขั้นตอน grant เนื่องจากจะทำให้อัลกอริทึมการจัดเส้นทางให้กับเซลล์คลาสเดียวกันยังคงรูปแบบของอัลกอริทึม iSLIP อยู่

ขั้นตอนการทำงานของอัลกอริทึม

ขั้นที่1 : request เมื่อมีเซลล์เข้ามาในบัฟเฟอร์ของอินพุต อินพุตดังกล่าวจะจัดเซลล์นั้นไว้ยัง VOQ ของลำดับความสำคัญของเซลล์นั้นๆ จากนั้นอินพุตแต่ละตัวจะพิจารณาว่ามีเซลล์ใน VOQ ที่ต้องการจะส่งหรือไม่ ถ้ามี ก็จะทำการส่ง request และลำดับความสำคัญของเซลล์ที่ต้องการจะส่งไปยังเอาท์พุตที่ต้องการโดยไม่คำนึงถึงลำดับความสำคัญของเซลล์

ขั้นที่ 2 : grant เมื่อเอาท์พุตได้รับ request และลำดับความสำคัญจากอินพุตแล้ว เอาท์พุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่มี เอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ จากนั้นเอาท์พุตจะส่ง grant และลำดับความสำคัญของเซลล์ที่เลือก ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว เอาท์พุตจะทำการสุ่มเลือกลำดับความสำคัญของเซลล์ที่จะให้บริการโดยให้น้ำหนักในการเลือกที่เท่ากัน

จากนั้นเอาท์พุตจะเลือกอินพุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่เลือก และส่ง grant พร้อมด้วยลำดับความสำคัญของเซลล์ดังกล่าว ไปยังอินพุตตัวที่เลือก และรอรับ accept จากขั้นที่ 3 จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของเอาท์พุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบ โมดูโล เมื่อได้รับ accept ตอบรับจากอินพุตดังกล่าว

ขั้นที่ 3 : accept เมื่ออินพุตได้รับ grant และลำดับความสำคัญจากเอาท์พุตแล้ว อินพุตตัวดังกล่าวจะพิจารณาว่ามีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันหรือไม่ หากไม่อินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญนั้นๆ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก และทำการส่งเซล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล แต่หากมีการแย่งชิงกันของเซลล์ที่มีลำดับความสำคัญที่แตกต่างกันแล้ว อินพุตจะทำการสุ่มเลือกลำดับความสำคัญโดยการสุ่มแบบถ่วงน้ำหนักของแต่ละลำดับความสำคัญ จากนั้นอินพุตจะเลือกเอาท์พุตตาม ตัวตัดสินใจ round-robin ของลำดับความสำคัญที่เลือกได้ และส่ง accept ไปยังเอาท์พุตตัวที่เลือก จากนั้นจะทำการส่งเซล์ที่ได้รับเลือกจากอินพุตนั้นไปยังเอาท์พุตที่เลือก จากนั้นตัวชี้ตำแหน่งในตัวตัดสินใจ round robin ของอินพุตที่ลำดับความสำคัญนั้นๆ จะเปลี่ยนแปลงโดยเพิ่มขึ้นทีละหนึ่งลำดับวนไปเรื่อยๆแบบโมดูโล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
เทคนิคเอนเวโลปที่นำเสนอ

4.1 เทคนิคเอนเวโลป

เทคนิคเอนเวโลป มีแนวคิดที่ว่าจะให้หน่วยข้อมูลที่จะส่งผ่านสวิตช์นั้นมีความยาวเพิ่มขึ้นแทนที่จะส่งที่ละเซลล์ โดยวิธีรวมเซลล์ที่จะไปยังเอาต์พุตเดียวกันรวมอยู่ภายในเอนเวโลปเป็นลำดับ (บางทีเรียกว่า wrapper หรือ macrocell) ซึ่งกลุ่มเซลล์ (entity)จะถูกสวิตช์ผ่านโครงสร้างสวิตช์ ความยาวของหน่วยข้อมูลก็คือความยาว ของไทม์สล็อตที่เพิ่มขึ้นในโครงสร้างการสวิตช์ หมายความว่าเรามีเวลามากขึ้นในการที่จะตัดสินใจเลือกเซลล์สำหรับส่งผ่านสวิตช์ ซึ่งก็คือประโยชน์ที่เห็นได้ชัดเจนที่จะนำไปสร้างเป็นสวิตช์โดยเฉพาะสวิตช์ที่มีตัวเลือกเซลล์ส่งผ่านสวิตช์แบบรวมศูนย์ (centralized arbitration) เนื่องจากตัวเลือกเซลล์มีเวลาจำกัดในแต่ละไทม์สล็อตในการทำงาน ทำให้เกิดปัญหาคอขวดในการสวิตช์ขึ้น การที่ไทม์สล็อตยาวขึ้นทำให้สามารถนำสถาปัตยกรรมใดๆ ของสวิตช์ที่มีข้อจำกัดด้านเวลาในการทำงานของตัวเลือกเซลล์มาใช้ได้ (เช่นเดียวกับที่ใช้การ shared memory กับการจัดการ bit-sliced) การที่หน่วยข้อมูลยาวขึ้นนั้นทำให้องศาสูงสุดของการขนานเพิ่มขึ้นด้วย [1]

การใช้เทคนิคเอนเวโลป ถึงแม้ว่าจะแก้ปัญหาคอขวดในการสวิตช์ข้อมูลได้ แต่ก็แลกกับสมรรถนะของสวิตช์ที่จะต้องลดลง ผลที่เกิดขึ้นตามมาจากการใช้เทคนิคนี้ก็คือ ค่าเฉลี่ยไทม์ดีเลย์ของเซลล์เพิ่มสูงขึ้นในการสวิตช์ข้อมูลผ่านสวิตช์ ในการนำเอาเทคนิคเอนเวโลปมาใช้ การวางบัฟเฟอร์ที่อินพุตของโครงสร้างสวิตช์เป็นสิ่งจำเป็น ทั้งนี้ก็เพื่อที่จะจัดเตรียมเอนเวโลปก่อนที่จะได้รับการส่งผ่านสวิตช์ ด้วยเหตุผลนี้เทคนิคเอนเวโลปจึงเหมาะสมกับสถาปัตยกรรมสวิตช์แบบ IQ หรือ CIOQ มากที่สุด เทคนิคเอนเวโลปทำให้สูญเสียความต่อเนื่อง (granularity) ในการทำ back-pressure เนื่องมาจากการเพิ่มความยาวของไทม์สล็อต ซึ่งอาจทำให้คุณสมบัติทางดีเลย์ของสวิตช์เสื่อมลง

แม้ว่าเทคนิคเอนเวโลปจะทำให้สมรรถนะของสวิตช์ลดลง ซึ่งมีผลทำให้คุณภาพการให้บริการลดลงไปด้วย ดังนั้นการพิจารณาที่จะเพิ่มค่า QoS ของสวิตช์จึงเป็นสิ่งสำคัญ ซึ่งเราจะทำอย่างไรที่จะให้แต่ละแพ็กเกตที่มีความต้องการค่า QoS แตกต่างกันสามารถรวม (mixed) อยู่ในเอนเวโลปเดียวกันได้ ในแต่ละช่องสัญญาณ แต่ละเอนเวโลปจะถูกสวิตช์ผ่านโครงสร้างสวิตช์ตามลำดับ ดังนั้นทราฟฟิกประเภทต่าง ๆ สามารถที่จะรวมกันอยู่ในเอนเวโลปเดียวกันได้ เราสามารถทำให้มันง่ายขึ้นในการนำไปสร้างเป็นสวิตช์โดยการลดจำนวนการแยกคิวที่แต่ละอินพุต (เช่น สวิตช์แบบ MIQ) ซึ่งการแยกคิวเป็นสิ่งจำเป็นในการเตรียมเอนเวโลปสำหรับส่งผ่านสวิตช์

การจัดเรียงลำดับเซลล์ส่งผ่านสวิตช์ ก็เป็นสิ่งจำเป็นในการนำมาพิจารณา ในการออกแบบสวิตช์ที่ใช้เทคนิคเอนเวโลป การใช้เทคนิคเอนเวโลปทำให้ตัวจัดเรียงเซลล์ (scheduler) มีเวลามากขึ้นในการเลือกเซลล์ที่อยู่ในบัฟเฟอร์ส่งผ่านสวิตช์ทำให้สามารถนำไปสร้างเป็นสวิตช์ได้ง่ายขึ้น เราจะเลือกเอนเวโลปลักษณะใดในการส่งผ่านสวิตช์ วิธีที่หนึ่งคือเลือกเอนเวโลปที่มีสิทธิได้รับการส่งผ่านสวิตช์ก่อน โดยไม่ต้องคำนึงถึงว่าเอนเวโลปนั้นจะมีเซลล์บรรจุอยู่ภายในเต็มแล้วหรือไม่ หรือวิธีที่สองคือเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มก่อน หรือใช้ทั้งสองวิธีประกอบกัน

จะเห็นได้ว่าการเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่ไม่เต็มอาจจะทำให้ค่าทรูพุตของสวิตช์ลดลง ยิ่งขนาดเอนเวโลปยาวขึ้นเท่าใดก็ยิ่งมีผลกระทบต่อค่าทรูพุตเพิ่มมากขึ้น ซึ่งจะเกิดเฉพาะกรณีที่การจัดสรรแบนด์วิดท์นั้นไม่เพียงพอต่อความต้องการ แต่ไม่มีผลกระทบต่อค่าทรูพุตของสวิตช์ในกรณีที่ทราฟฟิกหนาแน่น ซึ่งผู้เขียนขอ เสนอว่าในการใช้วิธีเลือกเอนเวโลปที่มีสิทธิได้รับการส่งผ่านสวิตช์ก่อนนั้น ควรเลือกขนาดของเอนเวโลปที่ไม่เกินค่าเฉลี่ยของความยาวคิวของบัฟเฟอร์ที่อินพุต และควรจัดสรรแบนด์วิดท์ของหน่วยความจำให้เพียงพอต่อความต้องการผลกระทบต่อค่าทรูพุตของสวิตช์ก็จะหมดไป กล่าวโดยสรุปข้อบกพร่องของเทคนิคเอนเวโลปก็เนื่องมาจากการแยกคิวที่อินพุตเพื่อเตรียมเอนเวโลปสำหรับส่งผ่านสวิตช์ ทำให้อาจไม่เหมาะสมที่จะนำไป ใช้ในทางปฏิบัติเมื่อสวิตช์มีขนาดใหญ่มากขึ้น ซึ่งสามารถแก้ไขได้โดยลดจำนวนการแยกคิวในแต่ละอินพุต

<u>ปัญหาประการแรก</u> เราจะเลือกเอนเวโลปใดในแต่ละอินพุตเพื่อส่งผ่านโครงสร้างการสวิตช์ที่จะทำให้ค่าทรูพุตของสวิตช์สูงสุด จะเห็นว่าการใช้เทคนิคเอนเวโลปทำให้ตัวจัดลำดับเซลล์มีเวลามากขึ้นในการเลือกแต่ละเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์ จึงนำไปสร้างเป็นสวิตช์ได้ง่ายขึ้น เราจะเลือกเอนเวโลปใดในการส่งผ่านโครงสร้างการสวิตช์ วิธีที่หนึ่งคือเลือกเอนเวโลปที่มีเซลล์บรรจุที่หัวคิวก่อน (เช่นเลือกโดยวิธีสุ่มแต่ละอินพุต หรือวนรอบแต่ละอินพุต หรือตามลำดับความสำคัญของเซลล์ เป็นต้น) วิธีที่สองคือเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มก่อน

การเลือกวิธีที่หนึ่งทำให้เกิดปัญหาคือ ถ้าเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่ไม่เต็ม จะเกิดการสูญเสียแบนด์วิดท์ในการสวิตช์ทำให้ค่าทรูพุตของสวิตช์ลดลง ยิ่งเอนเวโลปยาวขึ้นเท่าใดก็ยิ่งมีผลกระทบต่อค่าทรูพุตของสวิตช์เพิ่มขึ้นเท่านั้น ในกรณีทราฟฟิกหนาแน่นจะไม่มีผลกระทบเพราะว่าเอนเวโลปมักจะมีเซลล์บรรจุอยู่เต็มอยู่เสมอ แต่เนื่องจากมีการแยกคิวที่อินพุตเพื่อเตรียมเอนเวโลปสำหรับส่งผ่านสวิตช์ ทำให้อาจไม่เหมาะสมที่จะนำไปใช้ในทางปฏิบัติเมื่อสวิตช์มีขนาดใหญ่ขึ้น

การเลือกวิธีที่สองในกรณีที่ทราฟฟิกต่ำ ๆ จะทำให้เกิดการสูญเสียแบนด์วิดท์ในการสวิตช์ เพราะว่าแต่ละคิวที่อินพุตส่วนใหญ่แล้วเอนเวโลปมีเซลล์ที่บรรจุอยู่ไม่เต็ม ซึ่งจะไม่ได้รับการส่งผ่านสวิตช์ ทำให้ทรูพุตของสวิตช์ลดลงยิ่งกว่าวิธีที่หนึ่ง แต่จะมีผลดีที่ทราฟฟิกหนาแน่นเพราะแต่ละคิวมีเซลล์บรรจุอยู่เป็นจำนวนมาก การเลือกเอนเวโลปบรรจุเต็มทำให้การสูญเสียแบนด์วิดท์ในการสวิตช์มีน้อยลง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
ผลทดสอบและการวิเคราะห์

5.1 แบบจำลองอัลกอริทึมที่นำเสนอ

ในการทดสอบสมรรถนะของเซลล์สวิตช์โดยใช้อัลกอริทึมตามที่ได้นำเสนอในบทที่ 3 นั้น จะใช้โปรแกรม Math Lab เข้ามาช่วยในการจำลองระบบ โดยสวิตช์ที่จำลองขึ้นมานั้นจะเป็นสวิตช์ขนาด 16 พอร์ต ทั้งขาเข้าและขาออก โดยแพ็กเกตที่เข้ามาในสวิตช์นี้จะได้รับการให้บริการทีละเซลล์ไปเรื่อยๆ โดยมีสมมุติฐานของแบบจำลองดังนี้

1. ความหนาแน่นของปริมาณเซลล์ที่เข้าขอใช้บริการ (offered load) มีค่าคงที่ในแต่ละการทดลอง

2. อัตราส่วนของปริมาณความหนาแน่นของข้อมูลที่มีลำดับความสำคัญแตกต่างกันทั้งสองลำดับมีค่าคงที่ในแต่ละการทดลอง

3. การเกิดขึ้นของเซลล์ที่ต้องการขอใช้บริการมีการกระจายตัวแบบ Bernoulli

4. บัฟเฟอร์ของสวิตช์มีอย่างไม่จำกัด

5. อัลกอริทึม iSLIP ที่ใช้มีการวนซ้ำเพียงรอบเดียว (1SLIP)

6. ไม่คิดคำนวณเวลาประวิงที่เกิดจากการคำนวณของระบบ

อัลกอริทึมที่ใช้ในการทดลองคืออัลกอริทึมการจัดลำดับเส้นทางภายในสวิตช์ซึ่งรองรับสภาพทราฟฟิกที่มีลำดับความสำคัญต่างกันซึ่งนำเสนอไว้ในบทที่ 3 โดยมีขอบเขตของการทดลองดังนี้

1. เซลล์ที่เข้าขอใช้บริการจะเกิดขึ้นเรื่อยๆเป็นเวลา 100,000 สล๊อตเวลาต่อ 1 พอร์ตของสวิตช์
2. ความหนาแน่นของปริมาณเซลล์ที่เข้าขอใช้บริการ (offered load) จะเพิ่มขึ้นเรื่อยๆในการทดลองแต่ละครั้ง ครั้งละ 5% จาก 20% ไปจนถึง 100%
3. มีการปรับสัดส่วนความหนาแน่นของเซลล์ข้อมูลระหว่างเซลล์ทั้งสองคลาส เริ่มต้นจากการกำหนดให้เซลล์คลาส 1มีสัดส่วน 10% และเซลล์คลาส 2 90% จากนั้นปรับเพิ่มสัดส่วนของเซลล์คลาส 1ทีละ 20% ในขณะที่ปรับลดเซลล์คลาส 2 ลงทีละ 20% จนกว่าสัดส่วนของเซลล์ทั้งสองระดับความสำคัญมีค่าเท่ากันที่ 50%

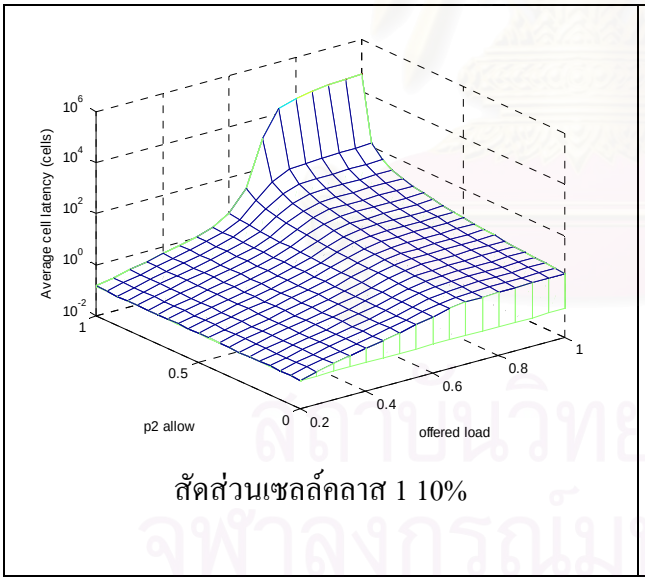
4. อัลกอริทึมที่ใช้ในการทดลองคืออัลกอริทึมการควบคุมที่ request, grant และ accept

กำหนดสัญลักษณ์ที่ใช้ดังนี้

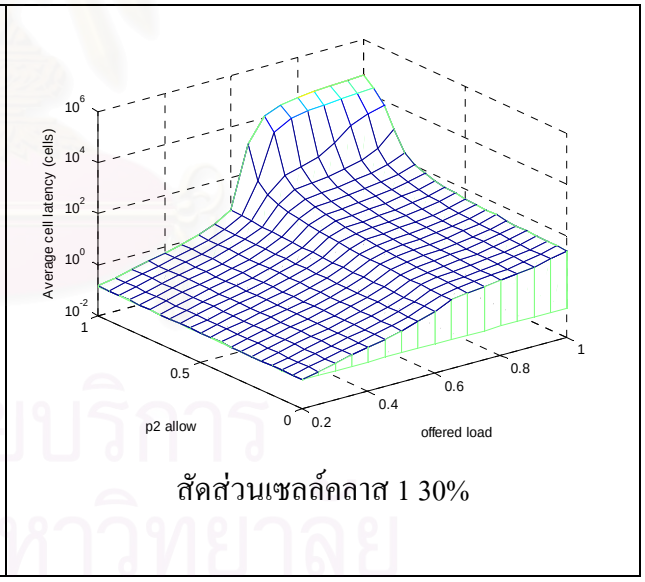
Offered load	แทนความหนาแน่นของทราฟฟิก
Percent class 1	แทนค่าสัดส่วนเซลล์คลาส 1
Average cell latency	แทนเวลาประวิงโดยเฉลี่ย
P1	แทนค่าเวลาประวิงของเซลล์ที่มีลำดับความสำคัญสูง
P2	แทนค่าเวลาประวิงของเซลล์ที่มีลำดับความสำคัญต่ำ
p_2_allow	แทนโอกาสที่เซลล์คลาส 2 จะชนะการแย่งชิง

5.2 ผลการทดสอบสมรรถนะของอัลกอริทึมที่นำเสนอ

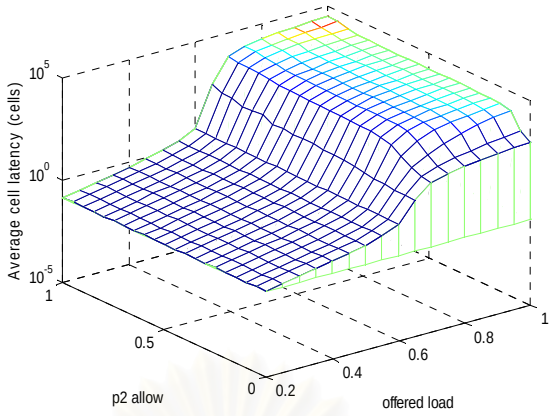
5.2.1 อัลกอริทึมการควบคุมที่ request



สัดส่วนเซลล์คลาส 1 10%

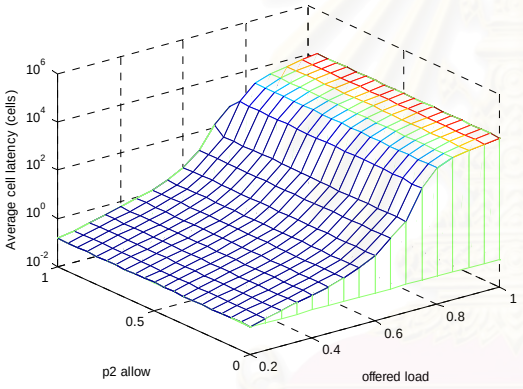


สัดส่วนเซลล์คลาส 1 30%

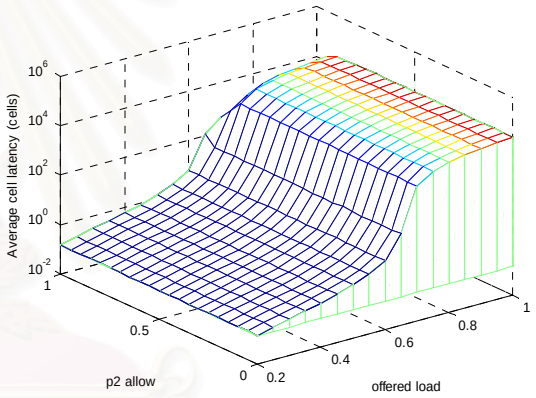


สัดส่วนเซลล์คลาส 1 50%

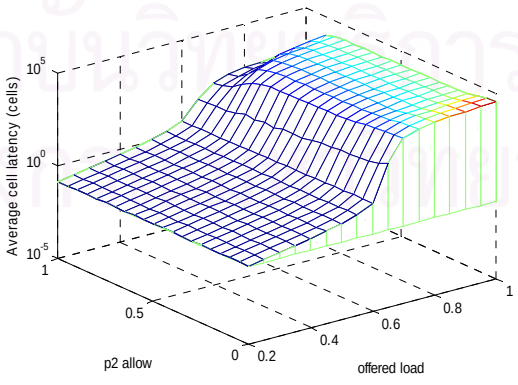
รูปที่ 5.1 เวลาประวิงของเซลล์คลาส 1



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

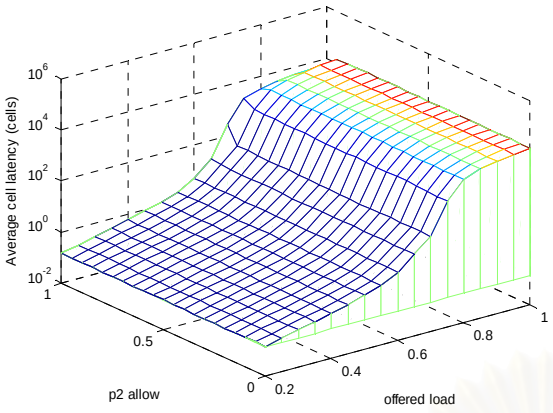


สัดส่วนเซลล์คลาส 1 50%

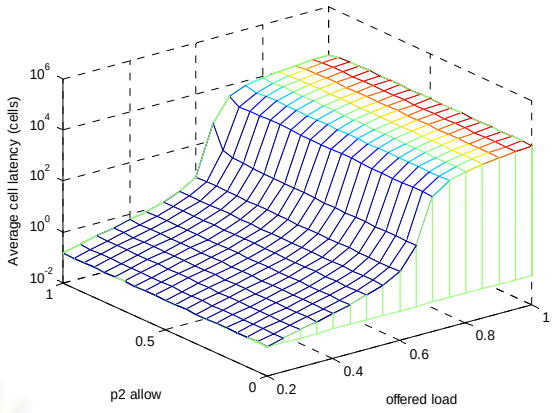
รูปที่ 5.2 เวลาประวิงของเซลล์คลาส 2

รูปที่ 5.1 และ 5.2 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ request ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เมื่อพิจารณาจากกราฟรูปใดรูปหนึ่งพบว่าการปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น ในขณะเดียวกันก็ส่งผลให้เวลาประวิงของเซลล์คลาส 2 ลดลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์แต่ละคลาสให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมเพิ่มขึ้น ส่วนค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง ณ จุดทำงานที่กำหนดให้ใช้ค่า p_2 เท่ากัน การเพิ่มขึ้นของสัดส่วนปริมาณเซลล์คลาส 1 จาก 10% เป็น 50% มีผลทำให้ค่าเวลาประวิงของเซลล์ทั้งสองคลาสเพิ่มขึ้น ลักษณะของผลการทดสอบที่ได้สามารถอธิบายได้ดังนี้ เมื่อเซลล์คลาส 1 เพิ่มขึ้น ในขณะที่ปริมาณทราฟฟิกยังคงเท่าเดิมทำให้ความรุนแรงในการแย่งใช้บริการระหว่างเซลล์ทั้งสองคลาสสูงขึ้น ส่งผลกระทบต่อเซลล์ทั้งสองคลาสทำให้โอกาสในการได้รับบริการลดลง จนทำให้ค่าเวลาประวิงของเซลล์แต่ละคลาสเพิ่มขึ้น จนเมื่อเพิ่มค่า p_2 ขึ้นเพื่อเพิ่มโอกาสในการให้บริการแก่เซลล์คลาส 2 ขึ้น จึงทำให้ค่าเวลาประวิงของเซลล์คลาส 2 ลดลง ดังจะเห็นได้จากกราฟแต่ละกราฟในรูปที่ 5.2 แต่การเพิ่ม p_2 ดังกล่าวกลับส่งผลให้เวลาประวิงของเซลล์คลาส 1 เพิ่มขึ้นดังจะเห็นได้จากแต่ละกราฟในรูปที่ 5.1 เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เท่ากันกับเซลล์คลาส 2 กราฟสมรรถนะภายใต้สัดส่วนปริมาณเซลล์คลาส 1 ที่ 50% ในรูปที่ 5.5 และ 5.6 ของเซลล์ทั้งสองคลาสมีลักษณะสมมาตรกัน และเวลาประวิงที่จุด p_2 50% ของเซลล์คลาส 1 และ 2 จะมีค่าเท่ากัน

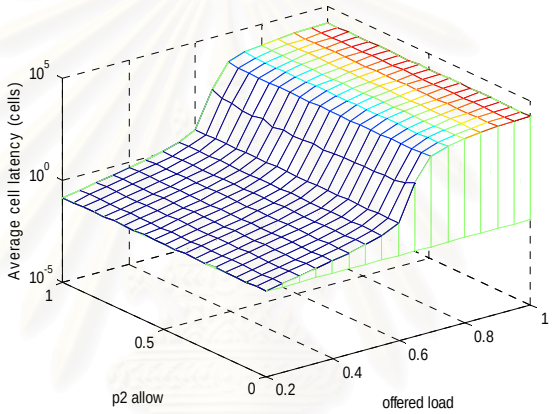
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



สัดส่วนเซลล์คลาส 1 10%



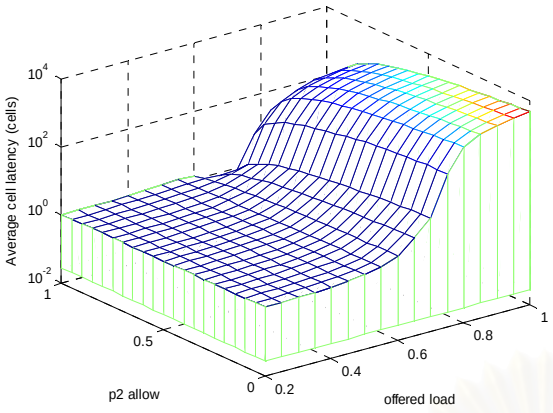
สัดส่วนเซลล์คลาส 1 30%



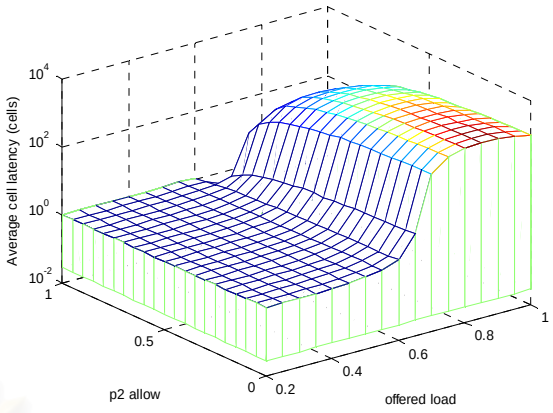
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.3 เวลาประวิงของเซลล์โดยเฉลี่ย

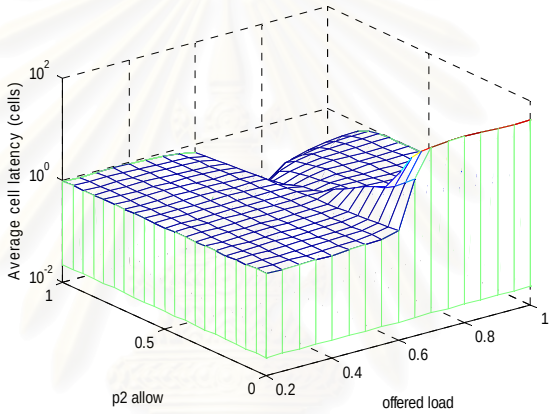
รูปที่ 5.3 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % จะเห็นว่าเวลาประวิงโดยเฉลี่ยจะมีค่าเพิ่มขึ้นเมื่อ offered load เพิ่มขึ้น เมื่อเรากำหนดให้ค่า p_2 เปลี่ยนแปลงค่าเวลาประวิงโดยเฉลี่ยโดยรวมจะเปลี่ยนแปลงน้อยมาก ซึ่งจะแตกต่างกับเวลาประวิงของเซลล์คลาส 1 และ 2 ที่จะเปลี่ยนแปลงเมื่อค่า p_2 เปลี่ยนแปลง เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลง เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request จะเปลี่ยนแปลงเพียงเล็กน้อย



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%



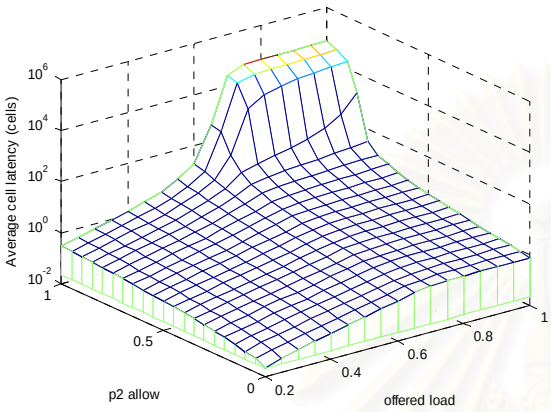
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.4 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1

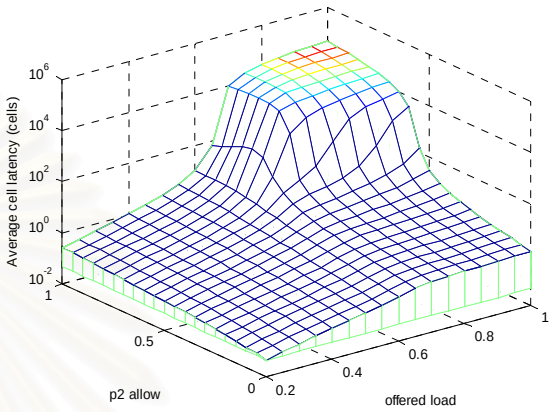
รูปที่ 5.4 แสงดอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลง ช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง ซึ่งช่วงการควบคุมจะแคบลง และที่สัดส่วนเซลล์คลาส 1 มีค่า 50% กำหนด offered load ค่าหนึ่งเราจะได้กราฟระหว่างอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 และค่า p_2 สมมาตรกันที่ค่า p_2 เท่ากับ 0.5 โดยจุดสูงสุด และต่ำสุดของกราฟจะหาได้

จากการกำหนดค่า p_2 ที่ 0% และ 100% ตามลำดับซึ่งเป็นไปตามที่คาดไว้ เนื่องจาก ณ จุดเหล่านั้นค่าเวลาประวิงของเซลล์ทั้งสองคลาสมีความต่างกันมากที่สุด

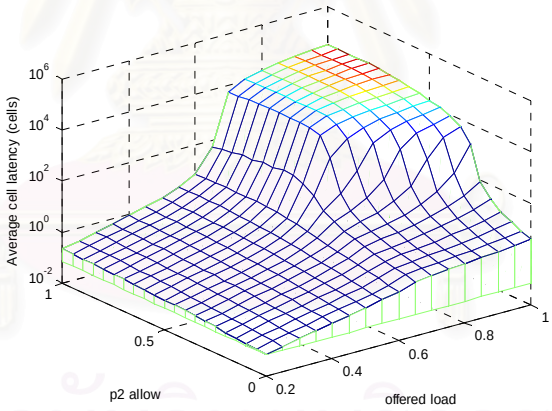
5.2.2 อัลกอริทึมการควบคุมที่ grant



สัดส่วนเซลล์คลาส 1 10%

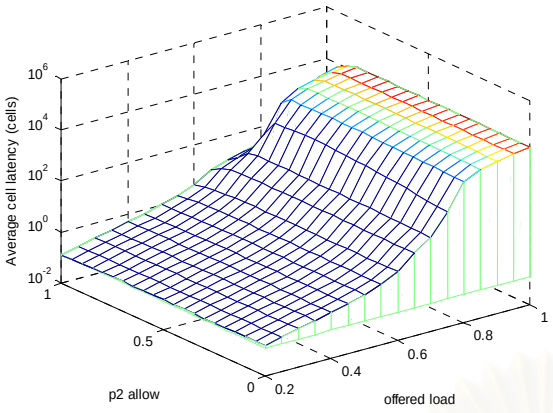


สัดส่วนเซลล์คลาส 1 30%

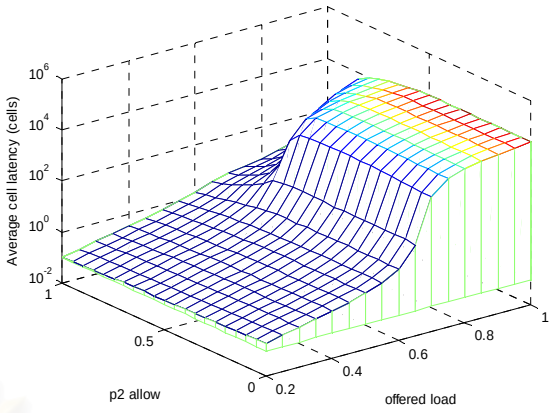


สัดส่วนเซลล์คลาส 1 50%

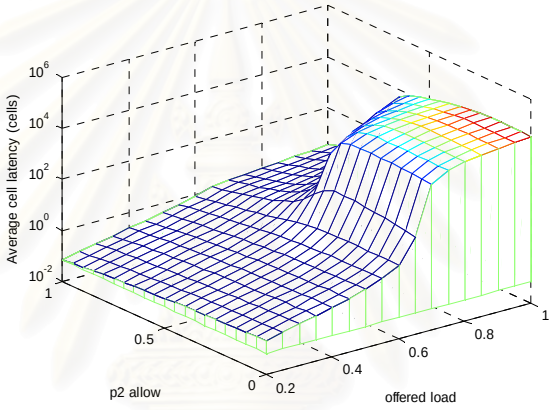
รูปที่ 5.5 เวลาประวิงของเซลล์คลาส 1



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

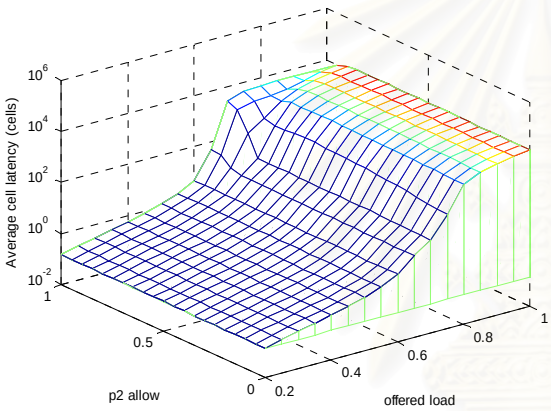


สัดส่วนเซลล์คลาส 1 50%

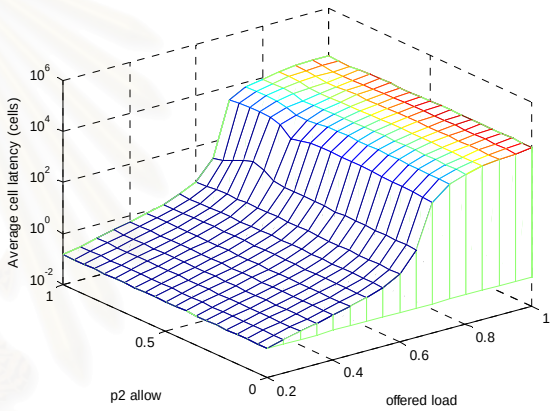
รูปที่ 5.6 เวลาประวิงของเซลล์คลาส 2

รูปที่ 5.5 และ 5.6 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ grant ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เมื่อพิจารณาจากกราฟรูปใดรูปหนึ่งพบว่าการปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น ในขณะเดียวกันก็ส่งผลให้เวลาประวิงของเซลล์คลาส 2 ลดลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์แต่ละคลาสให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมเพิ่มขึ้น ส่วนค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง ณ จุดทำงานที่กำหนดให้ใช้ค่า p_2 เท่ากัน การเพิ่มขึ้นของสัดส่วนปริมาณเซลล์คลาส 1 จาก 10% เป็น 50% มีผลทำให้ค่าเวลาประวิงของเซลล์ทั้งสองคลาสเพิ่มขึ้น

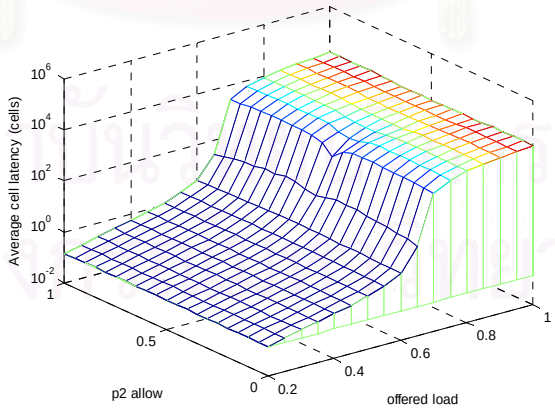
ลักษณะของผลการทดสอบที่ได้สามารถอธิบายได้ดังนี้ เมื่อเซลล์คลาส 1 เพิ่มขึ้น ในขณะที่ปริมาณทราฟฟิกยังคงเท่าเดิมทำให้ความรุนแรงในการแย่งใช้บริการระหว่างเซลล์ทั้งสองคลาสสูงขึ้น ส่งผลกระทบต่อเซลล์ทั้งสองคลาสทำให้โอกาสในการได้รับบริการลดลง จนทำให้ค่าเวลาประวิงของเซลล์แต่ละคลาสเพิ่มขึ้น จนเมื่อเพิ่มค่า p_2 ขึ้นเพื่อเพิ่มโอกาสในการให้บริการแก่เซลล์คลาส 2 ขึ้น จึงทำให้ค่าเวลาประวิงของเซลล์คลาส 2 ลดลง ดังจะเห็นได้จากกราฟแต่ละกราฟในรูปที่ 5.6 แต่การเพิ่ม p_2 ดังกล่าวกลับส่งผลให้เวลาประวิงของเซลล์คลาส 1 เพิ่มขึ้นดังจะเห็นได้จากแต่ละกราฟในรูปที่ 5.5 เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เท่ากันกับเซลล์คลาส 2 กราฟสมรรถนะภายใต้สัดส่วนปริมาณเซลล์คลาส 1 ที่ 50% ในรูปที่ 5.5 และ 5.6 ของเซลล์ทั้งสองคลาสมีลักษณะสมมาตรกัน และเวลาประวิงที่จุด p_2 50%ของเซลล์คลาส 1 และ 2 จะมีค่าเท่ากัน



สัดส่วนเซลล์คลาส 1 10%



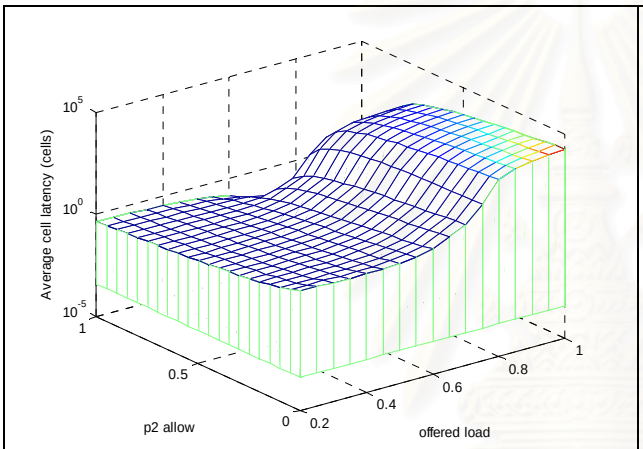
สัดส่วนเซลล์คลาส 1 30%



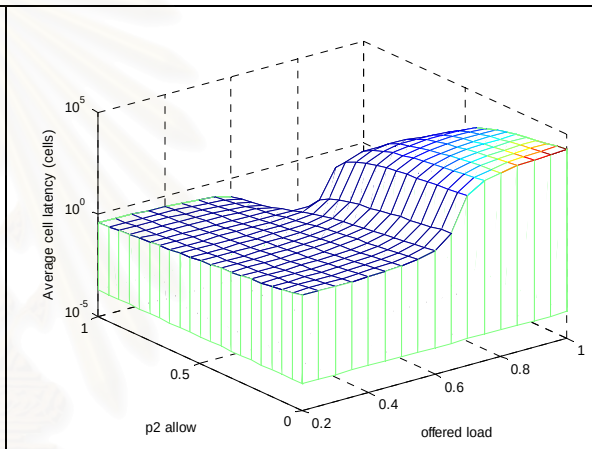
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.7 เวลาประวิงของเซลล์โดยเฉลี่ย

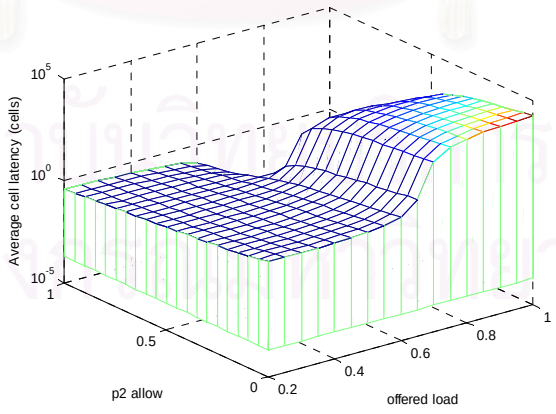
รูปที่ 5.7 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ grant ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % จะเห็นว่าเวลาประวิงโดยเฉลี่ยจะมีค่าเพิ่มขึ้นเมื่อ offered load เพิ่มขึ้น เมื่อเรากำหนดให้ค่า p_2 เปลี่ยนแปลงค่าเวลาประวิงโดยเฉลี่ยโดยรวมจะเปลี่ยนแปลงน้อยมาก ซึ่งจะแตกต่างกับเวลาประวิงของเซลล์คลาส 1 และ 2 ที่จะเปลี่ยนแปลงเมื่อค่า p_2 เปลี่ยนแปลง เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลง เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ grant จะมีแนวโน้มคล้ายกับเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request แต่ที่ต่างกันคือค่าเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ grant จะมีค่าน้อยที่สุดที่ค่า p_2 เท่ากับสัดส่วนปริมาณเซลล์คลาส 2 (สัดส่วนเซลล์คลาส 1 10% จะมีปริมาณเซลล์คลาส 1 10 ส่วน ปริมาณเซลล์คลาส 2 90 ส่วน)



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

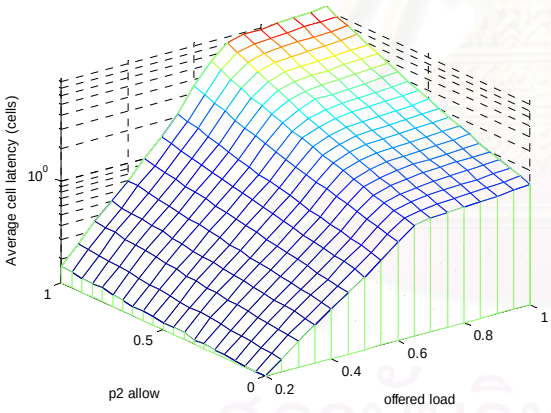


สัดส่วนเซลล์คลาส 1 50%

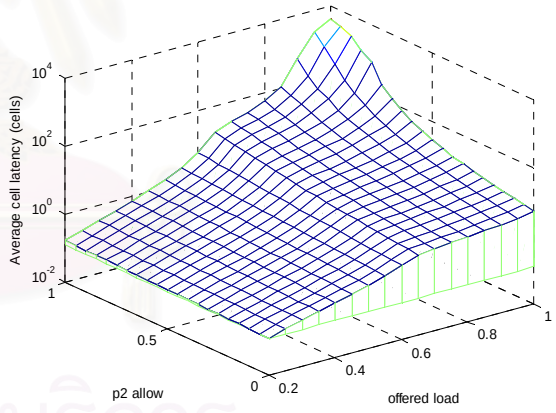
รูปที่ 5.8 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1

รูปที่ 5.8 แสงดอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ grant ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลง ช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง ซึ่งช่วงการควบคุมจะแคบลงเหมือนกับกราฟของอัลกอริทึมการควบคุมที่ request และที่สัดส่วนเซลล์คลาส 1 มีค่า 50% กำหนด offered load ค่าหนึ่งเราจะได้กราฟระหว่างอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 และค่า p_2 สมมาตรกันที่ค่า p_2 เท่ากับ 0.5 โดยจุดสูงสุด และต่ำสุดของกราฟจะหาได้จากการกำหนดค่า p_2 ที่ 0% และ 100% ตามลำดับซึ่งเป็นไปตามที่คาดไว้ เนื่องจาก ณ จุดเหล่านั้น ค่าเวลาประวิงของเซลล์ทั้งสองคลาสมีความต่างกันมากที่สุด

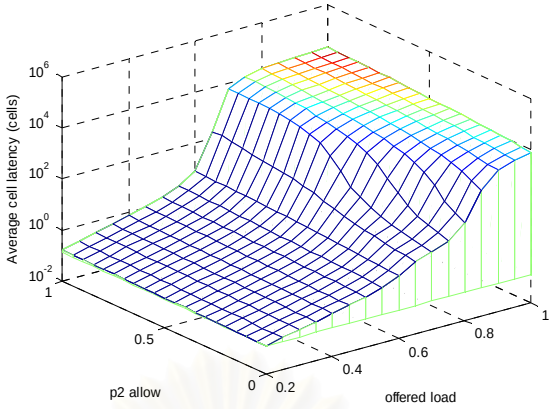
5.2.3 อัลกอริทึมการควบคุมที่ accept



สัดส่วนเซลล์คลาส 1 10%

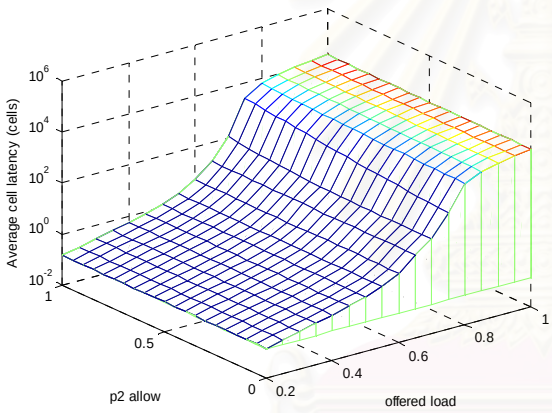


สัดส่วนเซลล์คลาส 1 30%

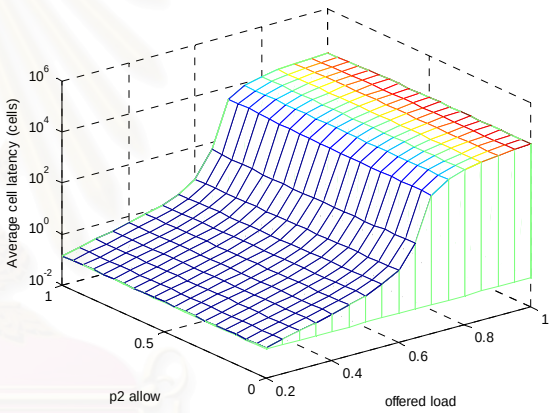


สัดส่วนเซลล์คลาส 1 50%

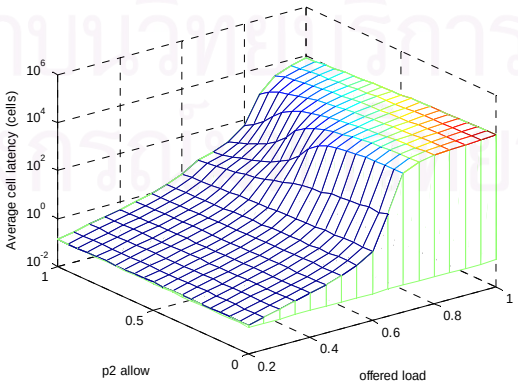
รูปที่ 5.9 เวลาประวิงของเซลล์คลาส 1



สัดส่วนเซลล์คลาส 1 10%



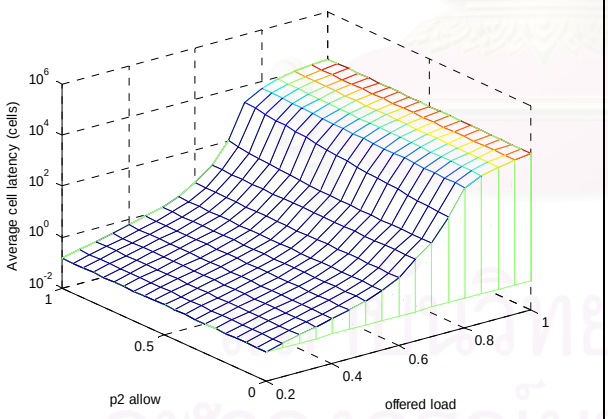
สัดส่วนเซลล์คลาส 1 30%



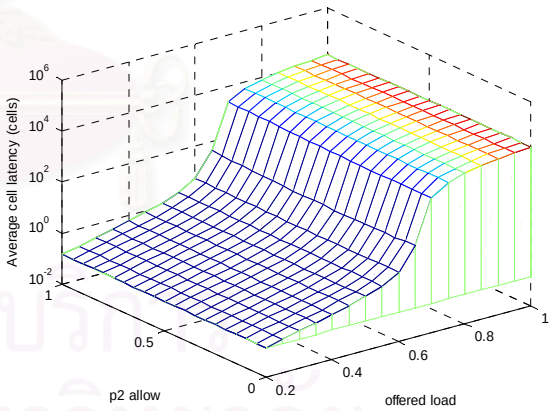
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.10 เวลาประวิงของเซลล์คลาส 2

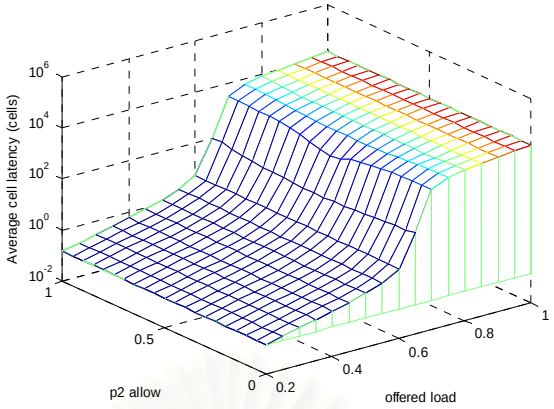
รูปที่ 5.9 และ 5.10 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ accept ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เช่นเดียวกับอัลกอริทึมควบคุมที่ request และอัลกอริทึมควบคุมที่ grant การปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น แต่ส่งผลเพียงเล็กน้อยสำหรับเวลาประวิงของเซลล์คลาส 2 ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ค่าเวลาประวิงของเซลล์คลาส 2 เราไม่สามารถควบคุมโดยการปรับค่า p_2 ได้ เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง ณ จุดทำงานที่กำหนดให้ใช้ค่า p_2 เท่ากัน การเพิ่มขึ้นของสัดส่วนปริมาณเซลล์คลาส 1 จาก 10% เป็น 50% มีผลทำให้ค่าเวลาประวิงของเซลล์ทั้งสองคลาสเพิ่มขึ้น ลักษณะของผลการทดสอบที่ได้สามารถอธิบายได้ดังนี้ เมื่อเซลล์คลาส 1 เพิ่มขึ้น ในขณะที่ปริมาณทราฟฟิกยังคงเท่าเดิมทำให้ความรุนแรงในการแย่งใช้บริการระหว่างเซลล์ทั้งสองคลาสสูงขึ้น ส่งผลกระทบต่อเซลล์ทั้งสองคลาสทำให้โอกาสในการได้รับบริการลดลง จนทำให้ค่าเวลาประวิงของเซลล์แต่ละคลาสเพิ่มขึ้น จนเมื่อเพิ่มค่า p_2 ขึ้นเพื่อเพิ่มโอกาสในการให้บริการแก่เซลล์คลาส 2 ขึ้น จึงทำให้ค่าเวลาประวิงของเซลล์คลาส 2 ลดลง ดังจะเห็นได้จากกราฟแต่ละกราฟในรูปที่ 5.10



สัดส่วนเซลล์คลาส 1 10%



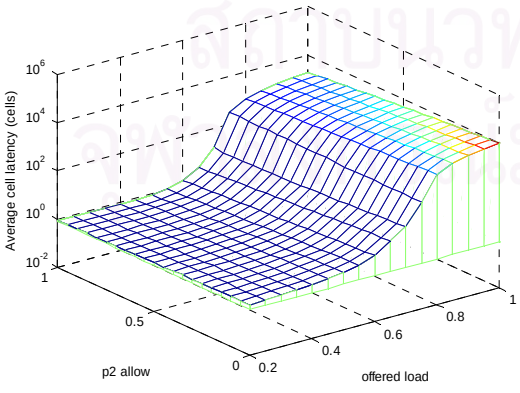
สัดส่วนเซลล์คลาส 1 30%



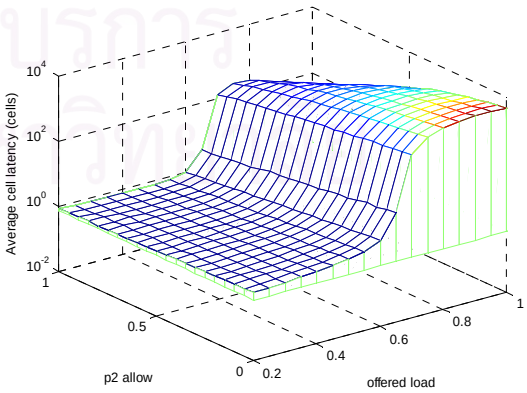
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.11 เวลาประวิงของเซลล์โดยเฉลี่ย

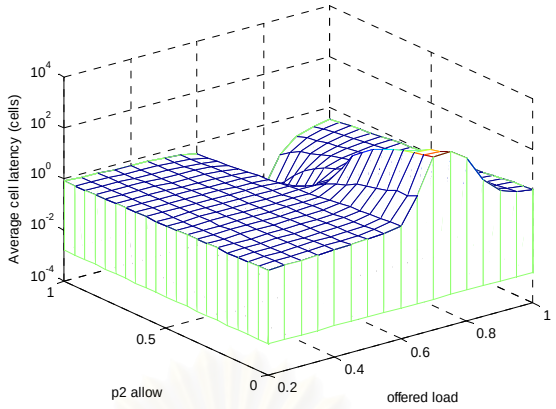
รูปที่ 5.11 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ accept ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเวลาประวิงโดยเฉลี่ยจะเพิ่มขึ้นเมื่อ offered load เพิ่มขึ้นและจะไม่เปลี่ยนแปลงตามค่า p_2 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงค่าประวิงเวลาเฉลี่ยจะเปลี่ยนแปลงเพียงเล็กน้อย จะเห็นว่าอัลกอริทึมควบคุมที่ accept การปรับค่า p_2 ไม่สามารถทำให้ค่าประวิงเวลาเฉลี่ยลดลงได้ เมื่อพิจารณาเปรียบเทียบกับอัลกอริทึมการควบคุมที่ request และ grant ดังรูปที่ 5.1, 5.2 ,5.3 ,5.5 ,5.6 และ 5.7 พบว่าในอัลกอริทึมนี้ค่า p_2 มีผลกระทบต่อค่าเวลาประวิงน้อยกว่าอัลกอริทึมการควบคุมที่ request และการควบคุมที่ grant ทั้งนี้เนื่องจากเซลล์ที่ผ่านเข้ามาถึงขั้นตอนการตัดสินใจนี้มีจำนวนน้อยลงมาก อัตราการแข่งขันกันระหว่างแต่ละคลาสจึงลดลงตามไปด้วยทำให้การเปลี่ยนแปลงน้ำหนักการสุ่มมีผลต่อการเลือกคลาสที่จะให้บริการลดลง



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%



สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.12 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1

รูปที่ 5.12 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ accept ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงแนวโน้มของกราฟจะคล้ายกับอัลกอริทึมการควบคุมที่ request และอัลกอริทึมการควบคุมที่ grant คือช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลงแคบลง และที่สัดส่วนเซลล์คลาส 1 มีค่า 50% กำหนด offered load ค่าหนึ่งเราจะได้กราฟระหว่างอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 และค่า p_2 สมมาตรกันที่ค่า p_2 เท่ากับ 0.5 โดยจุดสูงสุด และต่ำสุดของกราฟจะหาได้จากการกำหนดค่า p_2 ที่ 0% และ 100% ตามลำดับ

5.3 การวิเคราะห์อัลกอริทึมที่นำเสนอ

5.3.1 การวิเคราะห์เชิงสมรรถนะ

การวิเคราะห์สมรรถนะของอัลกอริทึมที่นำเสนอนี้ จะทำการวิเคราะห์สมรรถนะในด้านต่างๆดังนี้

5.3.1.1 สมรรถนะด้านการควบคุมคุณภาพการให้บริการ

สมรรถนะด้านเวลาประวิงของอัลกอริทึมที่นำเสนอนั้นจะวัดออกมาโดยการเปรียบเทียบเวลาประวิงโดยเฉลี่ยระหว่างแต่ละอัลกอริทึม ณ จุดที่มีอัตราส่วนเวลาประวิงของเซลล์คลาส 1ต่อเซลล์คลาส 2, offered load และ สัดส่วนปริมาณเซลล์คลาส 1 เท่ากัน

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	58.1929	0.5960
ควบคุมที่ grant	54.3794	0.6845
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.1 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	4254.55	0.9072
ควบคุมที่ grant	3042.10	0.8405
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.2 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	2115.75	0.9293
ควบคุมที่ grant	1143.59	0.6902
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.3 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	8650.45	0.8725
ควบคุมที่ grant	7217.61	0.6370
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.4 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	3232.88	0.1355
ควบคุมที่ grant	1204.59	0.4949
ควบคุมที่ accept	1265.74	0.4364

ตารางที่ 5.5 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 70% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	13328.76	0.0811
ควบคุมที่ grant	11948.42	0.3924
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

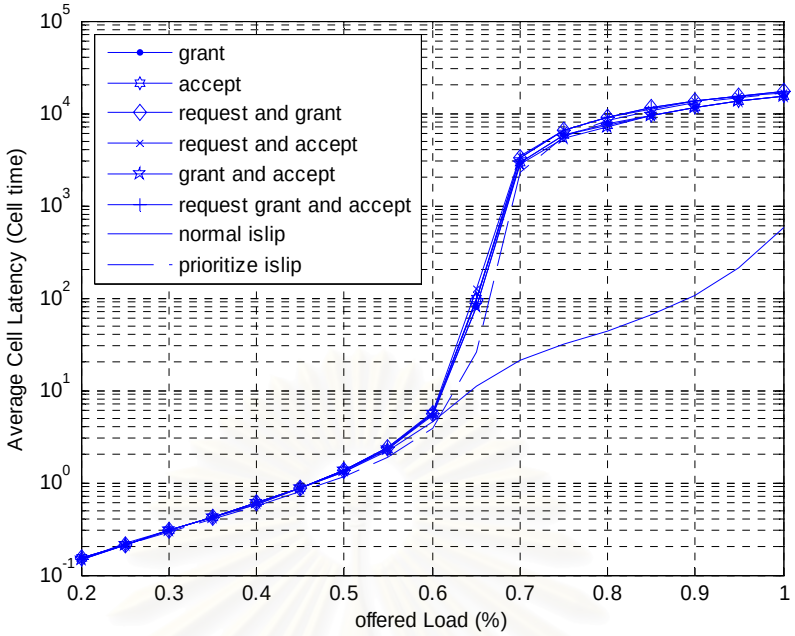
ตารางที่ 5.6 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	9429.18	0.4361
ควบคุมที่ accept	7534.18	0.0813

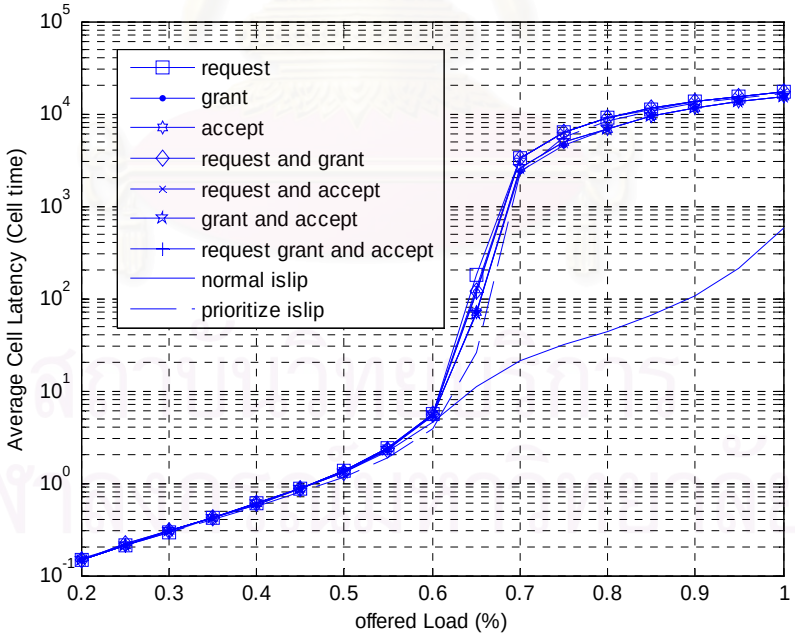
หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.7 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

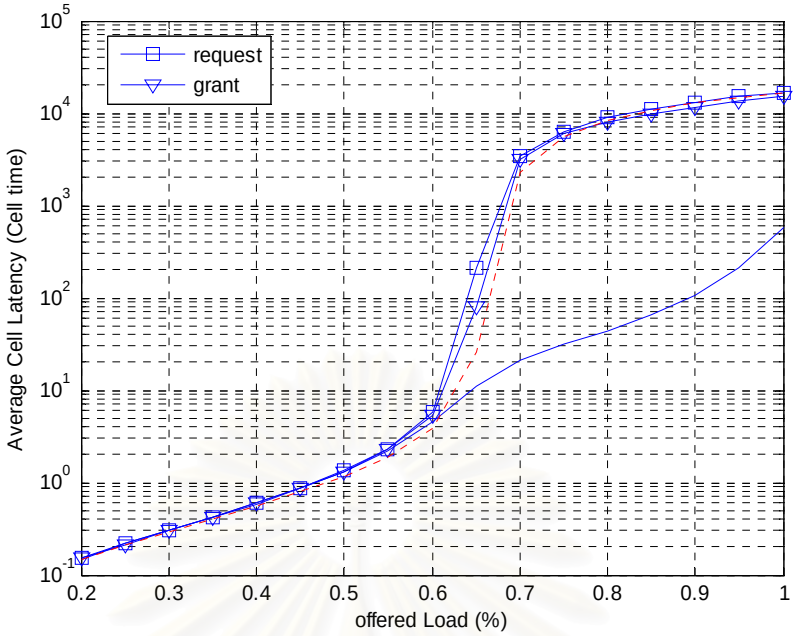
จากตารางที่ 5.1 ถึง ตารางที่ 5.7 แสดงให้เห็นอย่างชัดเจนว่า อัลกอริทึมที่นำเสนอสามารถควบคุมคุณภาพการให้บริการซึ่งในที่นี้จะแสดงออกมาในรูปของอัตราส่วนเวลาประวิงของ เซลล์คลาส 2ต่อเซลล์คลาส 1 ได้ โดยการควบคุมค่า p_2 เมื่อพิจารณาเปรียบเทียบกันในแต่ละตารางพบว่าเมื่อสภาพทราฟฟิกเปลี่ยนไป หากเรายังต้องการควบคุมคุณอัตราส่วนการให้บริการไว้เท่าเดิม เราจำเป็นจะต้องปรับค่า p_2 ใหม่เพื่อให้ได้ค่าอัตราส่วนการให้บริการที่คงที่ แต่การปรับค่า p_2 ดังกล่าวจะทำให้ค่าเวลาประวิงที่ได้เปลี่ยนไปด้วยเช่นกัน เมื่อพิจารณาเปรียบเทียบกันเองระหว่างแต่ละอัลกอริทึมในแต่ละกราฟพบว่า อัลกอริทึมการควบคุมที่ grant จะให้ค่าเวลาประวิงโดยเฉลี่ยน้อยที่สุดเนื่องจากเมื่อไม่มีการควบคุมที่ขั้นตอน request แล้วจะทำให้มีเซลล์ผ่านไปยังขั้นตอนต่อไปคือ grant และ accept เพิ่มขึ้น จึงทำให้โอกาสที่จะได้ทรูพุตในการจัดสรรเส้นทางสูงมีเพิ่มขึ้น พบว่ามีบางอัตราส่วนที่ไม่สามารถควบคุมได้ในสภาพทราฟฟิกที่ทำการทดสอบ ทั้งนี้แล้วแต่ความสามารถเกี่ยวกับช่วงการทำงานของอัลกอริทึมโดยจะกล่าวถึงในส่วนต่อไป



รูปที่ 5.13 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วน เวลาประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 100 เท่าได้ที่ offered load มีค่า 80% และมีสัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม



รูปที่ 5.14 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วน เวลาประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 10 เท่าได้ที่ offered load มีค่า 80% และมีสัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม

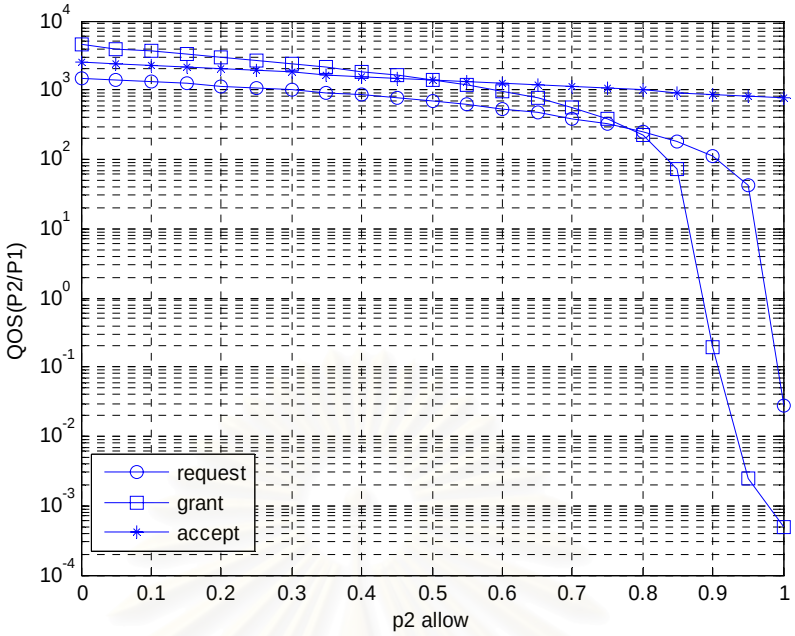


รูปที่ 5.15 แสดงผลเปรียบเทียบเวลาประวิงโดยเฉลี่ยของเซลล์ ที่ควบคุมอัตราส่วนเวลาประวิงเฉลี่ยของเซลล์คลาส 2/เซลล์คลาส 1 ที่ 10 เท่าได้ที่ offered load มีค่า 90% และมี สัดส่วนเซลล์คลาส 1 50% ในแต่ละอัลกอริทึม

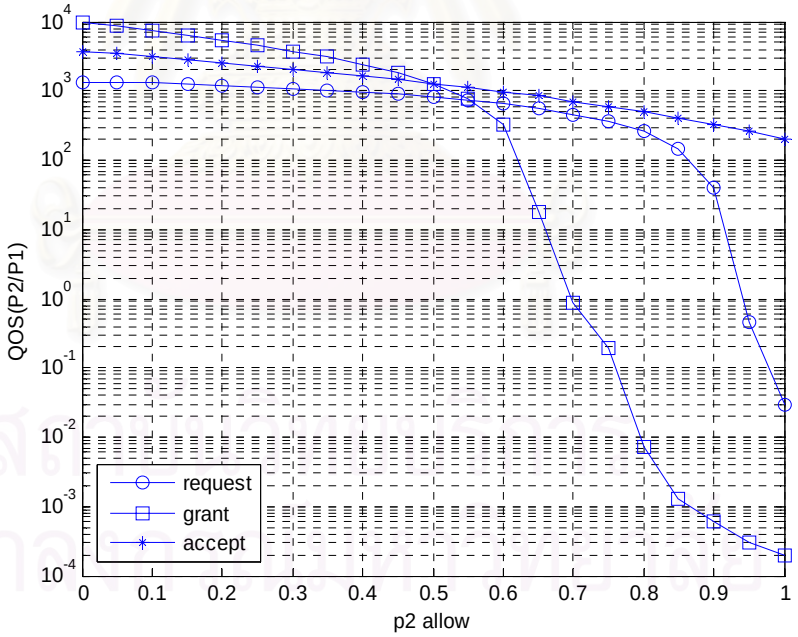
จากรูปที่ 5.13, 5.14 และ 5.15 พบว่า หากเรากำหนดค่า p_2 ตั้งแต่ต้นและไม่เปลี่ยนแปลงตามสภาพทราฟฟิก เพื่อให้ที่จุด offered load หนึ่งจะได้อัตราส่วนระหว่างเวลาประวิงเฉลี่ยของเซลล์คลาส 1 ต่อเซลล์คลาส 2 ที่ต้องการ พบว่าที่ offered load เปลี่ยนไป เราจะไม่สามารถได้รับอัตราส่วนระหว่างเวลาประวิงเฉลี่ยของเซลล์คลาส 1 ต่อเซลล์คลาส 2 ตามต้องการ ดังนั้นจึงควรปรับค่า p_2 เปลี่ยนไปเรื่อยๆตามสภาพของทราฟฟิกซึ่งจะกล่าวถึงในส่วนต่อไป

5.3.1.2 สมรรถนะด้านช่วงการใช้งาน

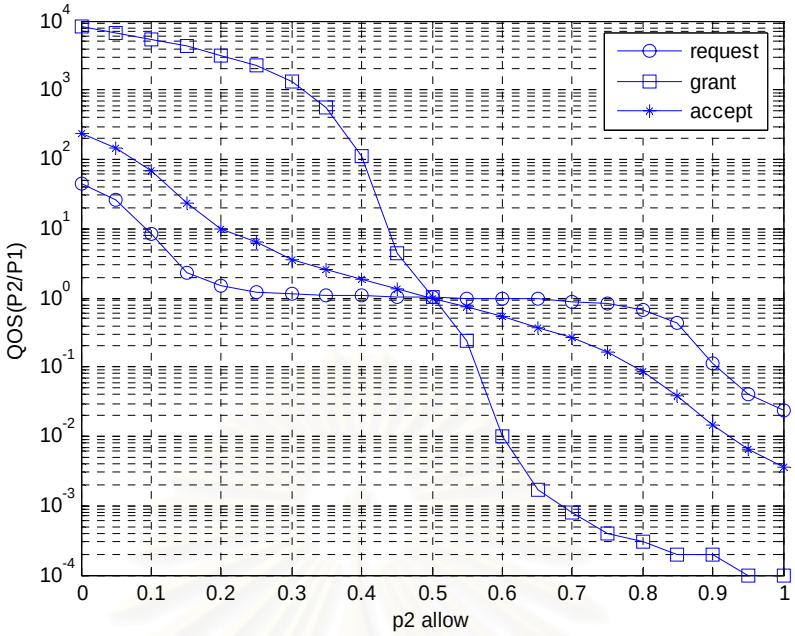
จากการวิเคราะห์ข้างต้น พบว่าในแต่ละอัลกอริทึมที่นำเสนอมีช่วงการใช้งานแตกต่างกันตามแต่สภาพของทราฟฟิกดังนี้



รูปที่ 5.16 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ค่า offered load 80% และค่า สัดส่วนเซลล์คลาส 1 10%



รูปที่ 5.17 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ค่า offered load 80% และค่า สัดส่วนเซลล์คลาส 1 30%

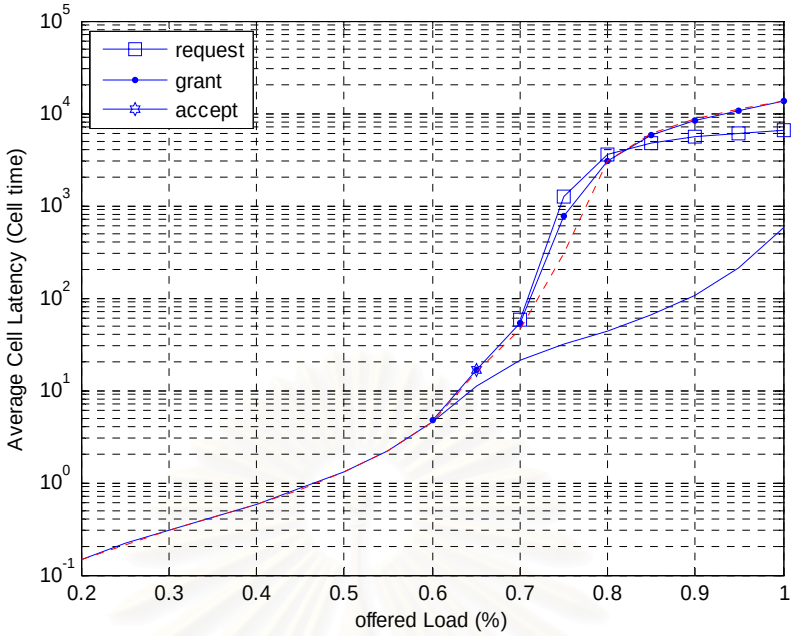


รูปที่ 5.18 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ค่า offered load 80% และค่าสัดส่วนเซลล์คลาส 1 50%

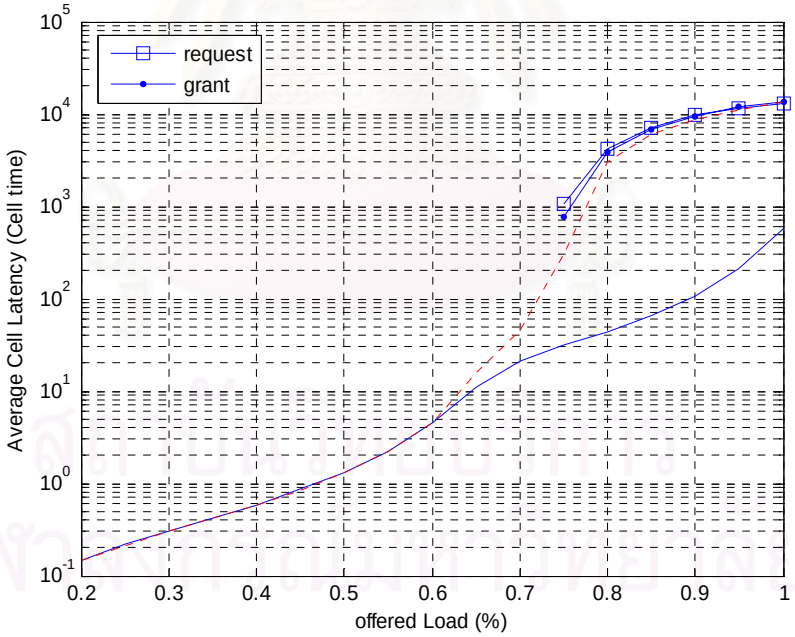
จากรูปที่ 5.16, 5.17 และรูปที่ 5.18 พบว่าอัลกอริทึมอัลกอริทึมการควบคุมที่ accept มีสมรรถนะด้านช่วงการใช้งานต่ำที่สุดเนื่องจากมีช่วง ควบคุมอัตราส่วนของเวลาประวิงเฉลี่ยระหว่างเซลล์คลาส 1 ต่อเซลล์คลาส 2 แคบที่สุด โดยค่า สัดส่วนเซลล์คลาส 1 มีผลกระทบต่อช่วงการใช้งานของอัลกอริทึมอัลกอริทึมการควบคุมที่ accept นั่นคือหากทราฟฟิกมีสัดส่วนปริมาณเซลล์คลาส 1 น้อย จะทำให้ช่วงการทำงานของอัลกอริทึมการควบคุมที่ accept แคบมากหรือแทบจะไม่สามารถควบคุมได้เลย ภายใต้สัดส่วนปริมาณเซลล์คลาส 1 ที่ 50% เซลล์ทั้งสองคลาสมีลักษณะสมมาตรกัน และเวลาประวิงที่จุด p_2 50%ของเซลล์คลาส 1 และ 2 จะมีค่าเท่ากัน เนื่องจากในสภาพทราฟฟิกดังกล่าวความสำคัญของเซลล์ทั้งสองคลาสจะเท่ากัน

5.3.1.2 สมรรถนะด้านค่าเวลาประวิง

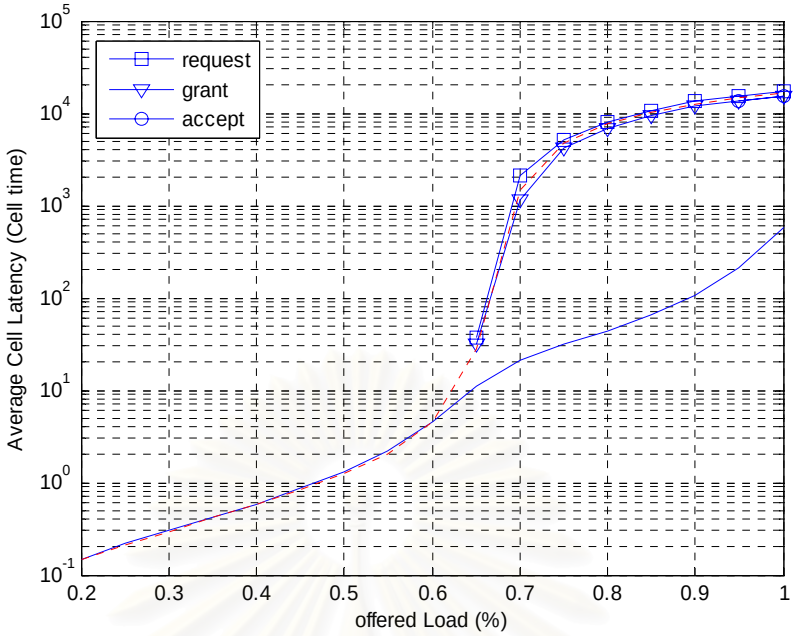
จากการวิเคราะห์ข้างต้น พบว่าหากควบคุมอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1ให้เท่ากันในแต่ละอัลกอริทึม ค่าเวลาประวิงเฉลี่ยที่ได้จะแตกต่างกันดังนี้



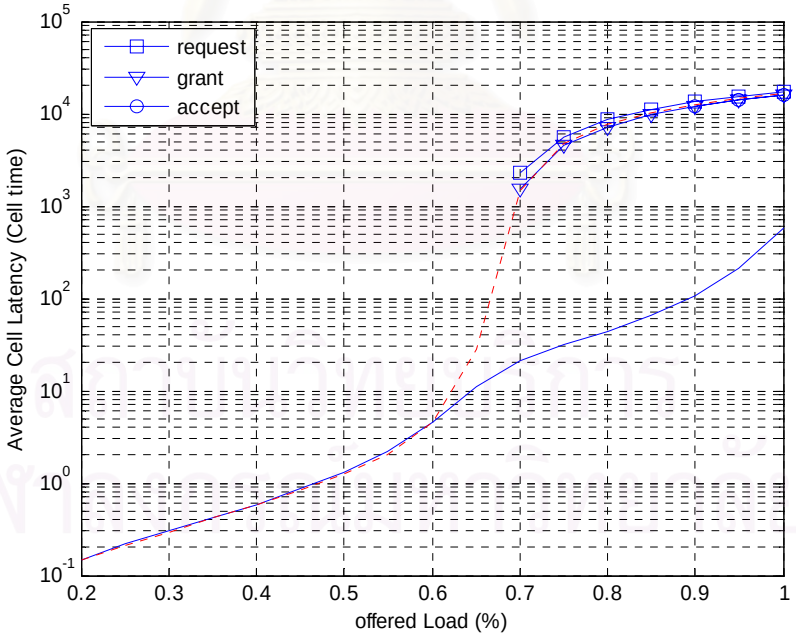
รูปที่ 5.19 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10%



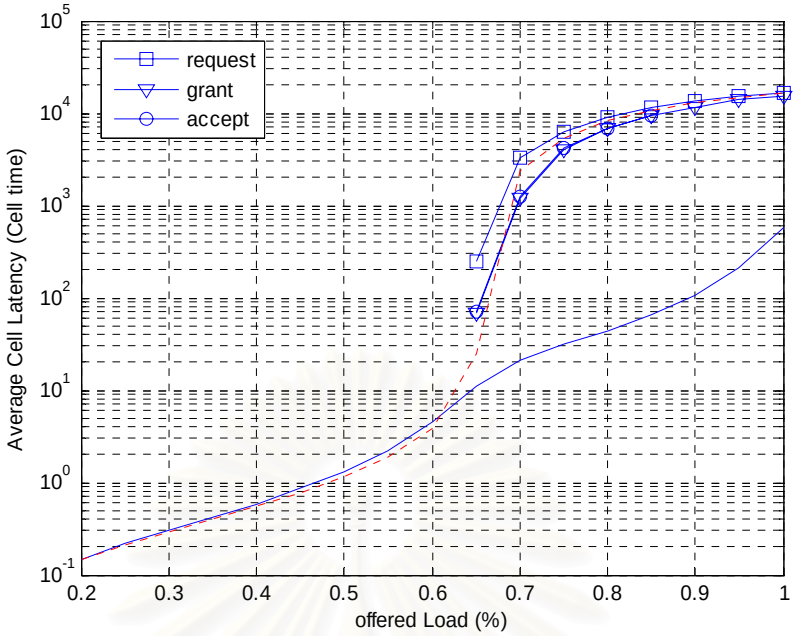
รูปที่ 5.20 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10%



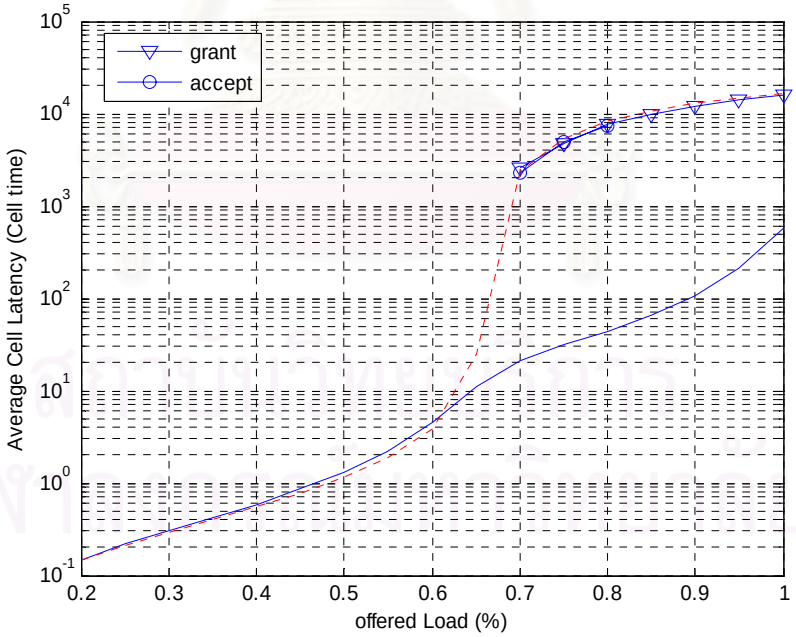
รูปที่ 5.21 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30%



รูปที่ 5.22 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30%



รูปที่ 5.23 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50%

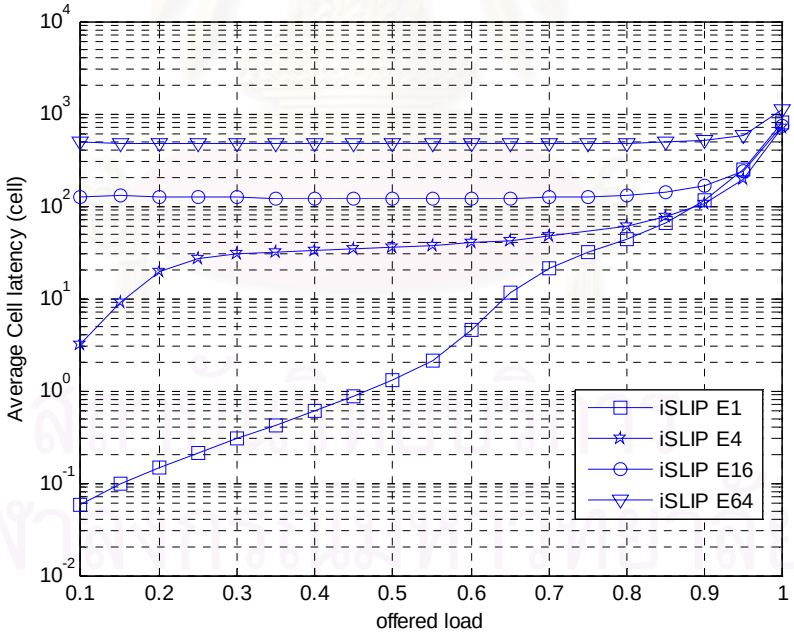


รูปที่ 5.24 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.19 ถึง 5.24 แสดงเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึม เมื่อกำหนดอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1เท่ากันแล้วเพิ่ม offered load ขึ้นจาก 20% ไปจนถึง 100% ค่าเวลาประวิงโดยเฉลี่ยในแต่ละอัลกอริทึมจะเพิ่มขึ้นตามไปด้วยดังเหตุผลที่ได้กล่าวไว้แล้วในตอนต้น ในแต่ละอัลกอริทึมสามารถทำงานในช่วง offered load และค่าสัดส่วนปริมาณเซลล์คลาส 1 ที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับค่าเวลาประวิงที่ต่างกันของเซลล์ทั้งสองคลาสที่แต่ละอัลกอริทึมสามารถควบคุมได้ เมื่อสภาพทราฟฟิกเปลี่ยนไป โดยอัลกอริทึมการควบคุมที่ grant และ accept จะมีช่วงการใช้งานที่กว้างที่สุดในทุกๆสภาพทราฟฟิก เมื่ออัตราส่วน P2 / P1 เพิ่มขึ้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มขึ้นด้วย เมื่อเพิ่มสัดส่วนเซลล์คลาส 1 จาก 10%-50% จากเหตุผลที่กล่าวแล้วในตอนต้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจึงเพิ่มสูงขึ้น ถึงแม้ว่าเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะมีค่าใกล้เคียงกับอัลกอริทึม Prioritized ISLIP แต่เราสามารถที่จะควบคุมคุณภาพการให้บริการได้อย่างแม่นยำ (ในเทอม P2 / P1)

5.4 วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์

วิธีที่ 1 ***เลือกเอนเวโลปที่มีเซลล์ที่หัวคิว*** *(Partially-Filled (PF) envelope)*

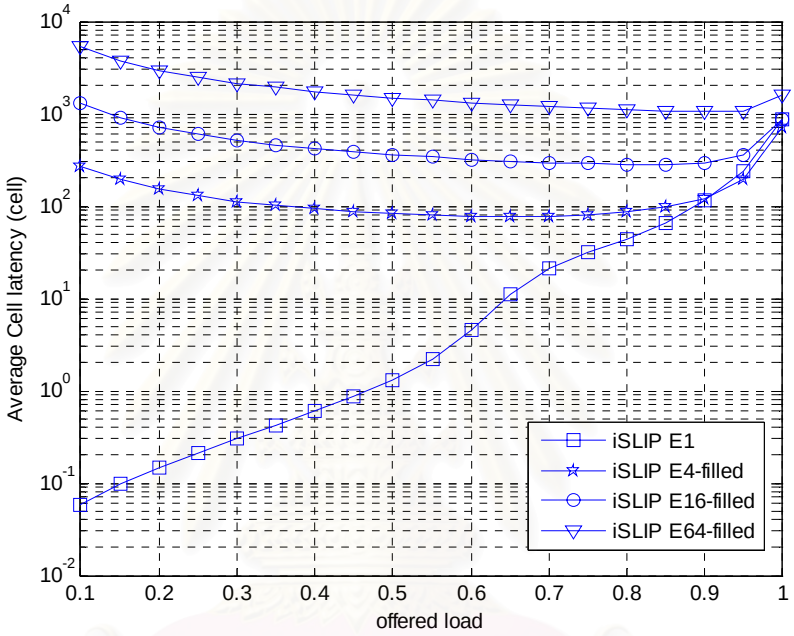


รูปที่ 5.25 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม iSLIP

จากรูปที่ 5.25 แสดงการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว พบว่าเมื่อเพิ่มขนาดความยาวของเอนเวโลป ค่าประวิงเวลาเฉลี่ยจะเพิ่มขึ้นซึ่งเราไม่สามารถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์จะหาค่าการแมตช์ได้ทัน

ตัวอย่างเช่น อัลกอริทึม iSLIP ใช้เวลาในการหาค่าการแมตช์ในแต่ละรอบประมาณ 10 ns ถ้านำไปใช้ในโครงข่ายที่มีข่ายเชื่อมโยงมีความเร็วในการรับส่งข้อมูล 40 Gbps เราจะต้องใช้เอนเวโลปขนาด 2 เซลล์ เป็นต้น เมื่อความยาวเอนเวโลปเป็น 16 เซลล์และ 64 เซลล์ ค่าประวิงเวลาเฉลี่ยมีค่าสูงในช่วง offered load ค่าต่ำ ดังนั้นจึงไม่เหมาะสมที่จะใช้กับสวิตช์ซึ่งจะต้องรองรับการเพิ่มขนาดของสวิตช์ในอนาคต

วิธีที่ 2 *เลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม* *(Filled (F) envelope)*

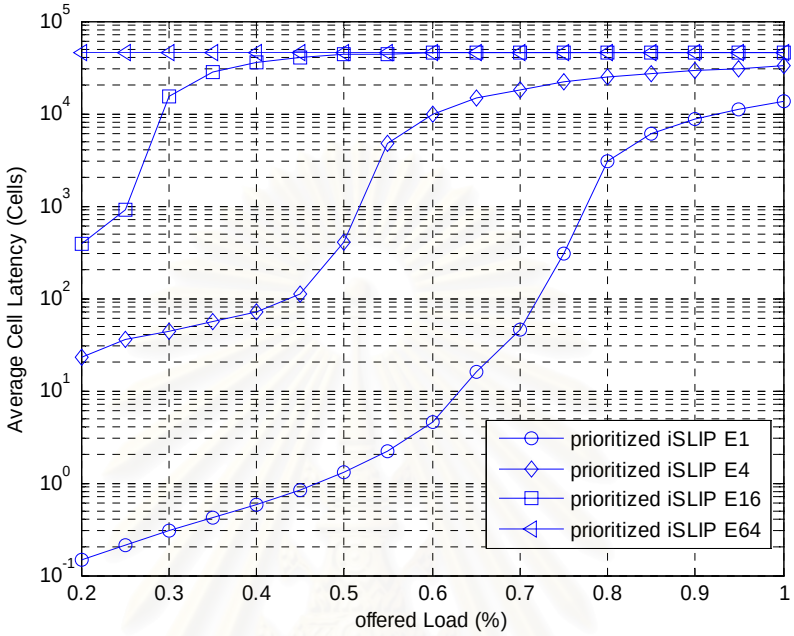


รูปที่ 5.26 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม iSLIP

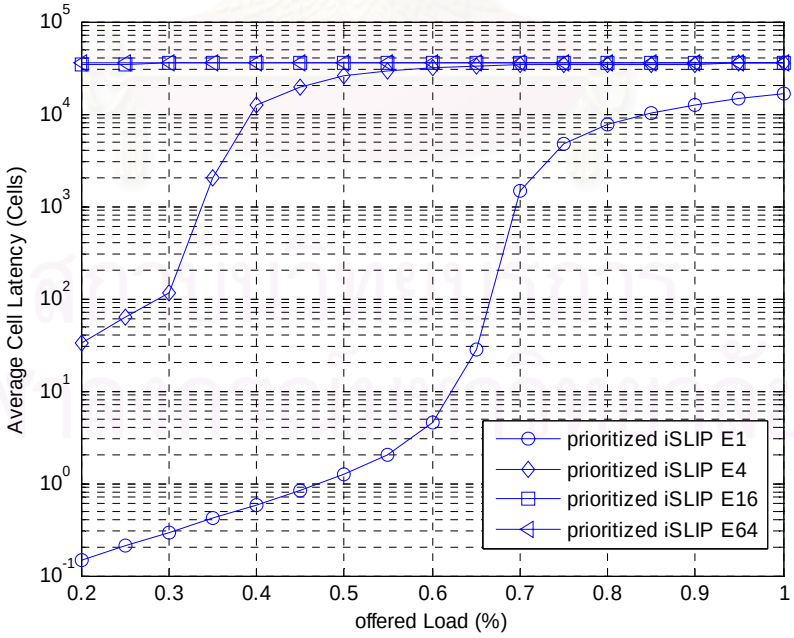
จากรูป การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มนั้น ให้ค่าเวลาประวิงเฉลี่ยสูงมากในช่วง offered load ต่ำ ๆ จึงไม่สามารถรองรับค่า QoS ได้ ยิ่งขนาดเอนเวโลปมีความยาวมากขึ้นก็จะมีค่าเฉลี่ยไทม์ดีเลย์สูงขึ้นมากเป็นอัตราส่วนสัมพันธ์กัน จึงไม่เหมาะสมที่จะนำวิธีนี้มาใช้กับสวิตช์ซึ่งจะต้องรองรับการเพิ่มขนาดของสวิตช์ในปัจจุบัน

5.5 วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองประเภทโดยใช้เทคนิคเอนเวโลป

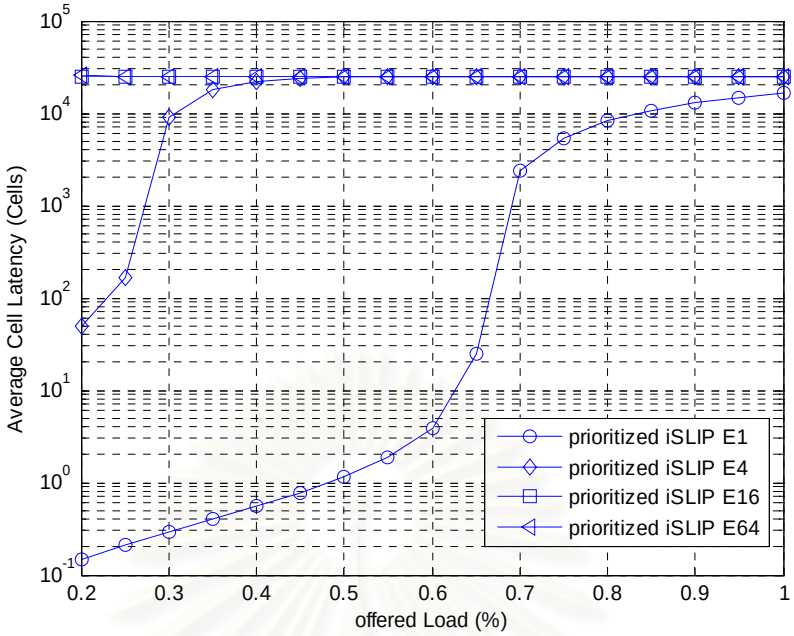
5.5.1วิธีที่ 1 *เลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (Partially-Filled (PF) envelope)*



รูปที่ 5.27 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 10 %



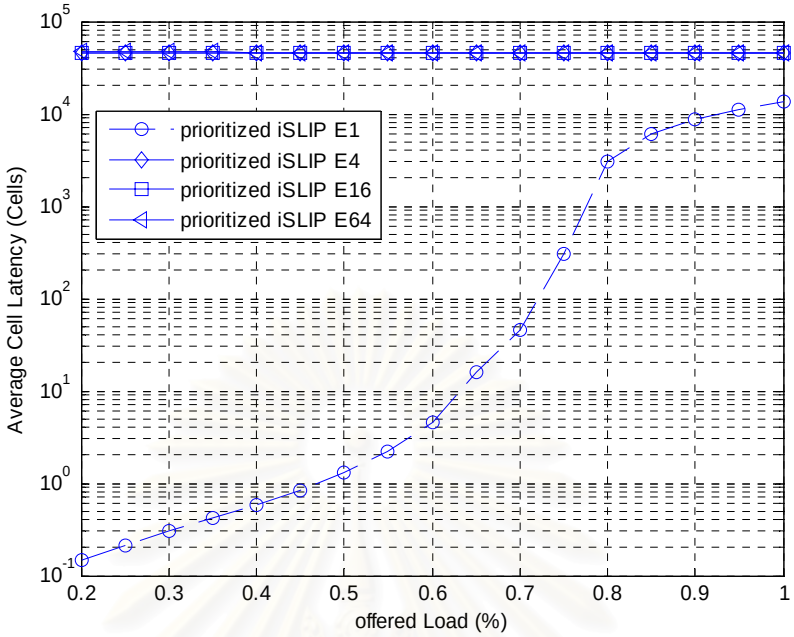
รูปที่ 5.28 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 30 %



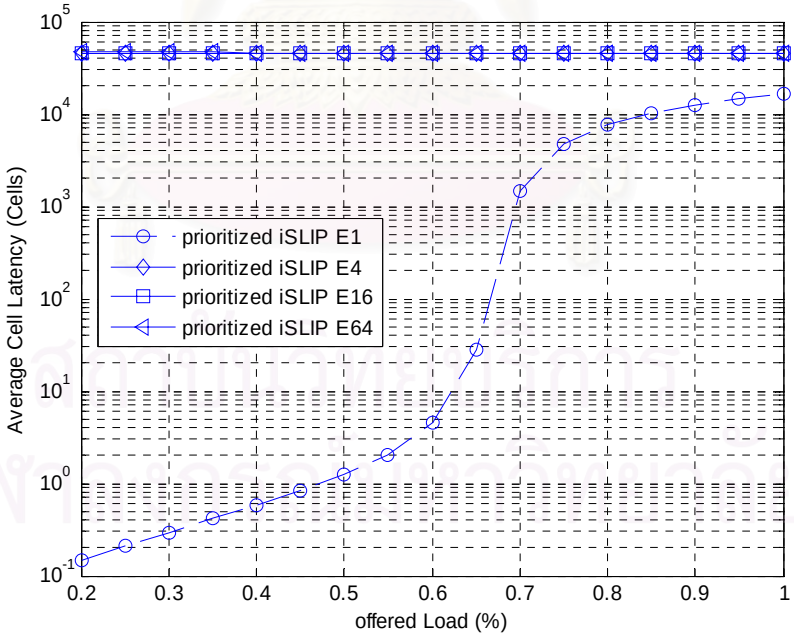
รูปที่ 5.29 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ขออัลกอริทึม prioritized iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 50 %

จากรูปที่ 5.27 5.28 และ 5.29 แสดงการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว พบว่าเมื่อเพิ่มขนาดความยาวของเอนเวโลป ค่าประวิงเวลาเฉลี่ยจะเพิ่มขึ้นซึ่งเราไม่สามารถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์จะหาค่าการแมตช์ได้ทัน ตามที่ทราบแล้วว่าอัลกอริทึม prioritized ISLIP จะมีค่าประวิงเวลาเฉลี่ยที่สูงกว่าอัลกอริทึม ISLIP เมื่อนำเทคนิคเอนเวโลปมาใช้ค่าประวิงเวลาเฉลี่ยของอัลกอริทึม prioritized ISLIP ก็จะมีค่าสูงกว่าอัลกอริทึม ISLIP เมื่อสัดส่วนเซลล์คลาส 1 เพิ่มขึ้นค่าเวลาประวิงเฉลี่ยจะสูงขึ้น เนื่องจากอัลกอริทึม prioritized ISLIP จะให้บริการเซลล์คลาส 1 ก่อนเสมอ เมื่อสัดส่วนเซลล์คลาส 1 เพิ่มขึ้น ค่าประวิงเวลาเฉลี่ยจะสูงขึ้น ซึ่งอธิบายได้ว่าค่าเวลาประวิงที่เพิ่มขึ้นจะเกิดขึ้นทั้งจากเซลล์คลาส 2 ที่ต้องรอให้เซลล์คลาส 1 ได้รับบริการก่อนและจากการแข่งขันกันเองระหว่างจำนวนเซลล์คลาส 1 ที่เพิ่มมากขึ้น

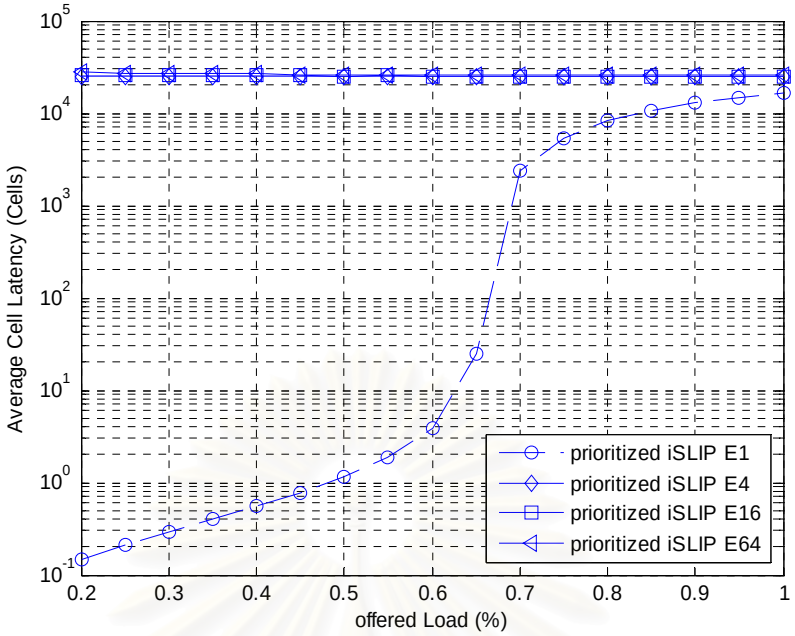
5.5.2 วิธีที่ 2 *เลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)*



รูปที่ 5.30 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 10 %



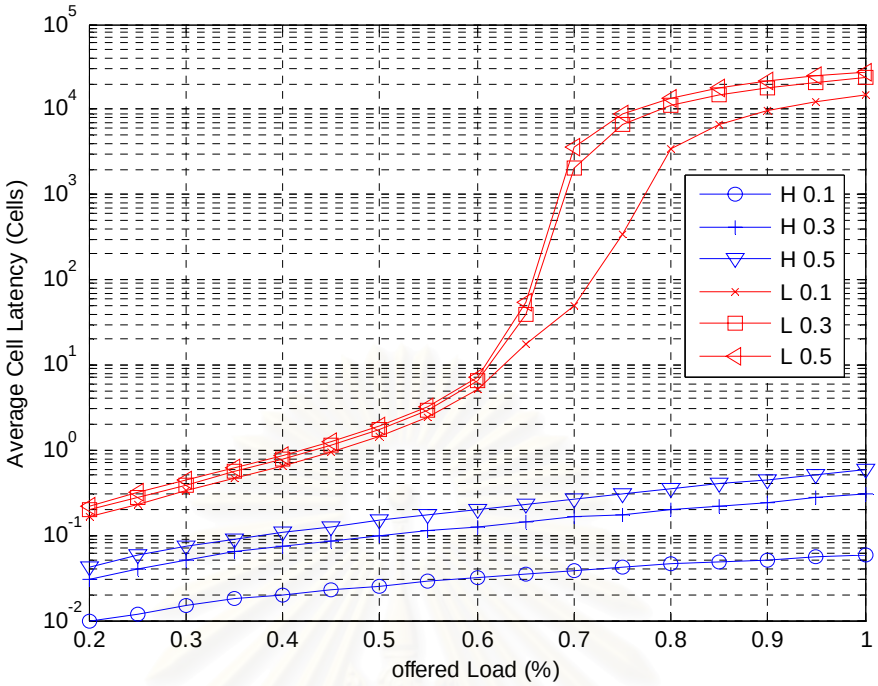
รูปที่ 5.31 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 30 %



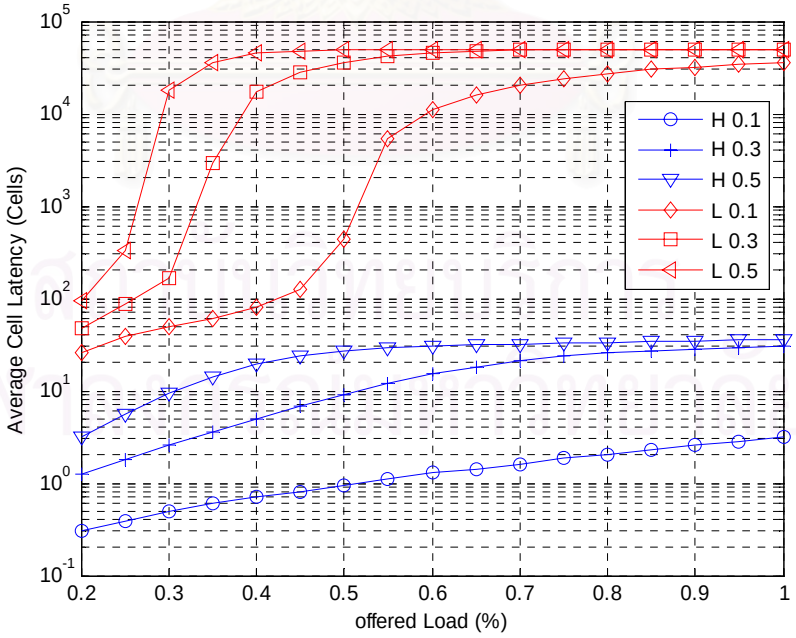
รูปที่ 5.32 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม prioritzed iSLIP สัดส่วนปริมาณเซลล์คลาส 1= 50 %

จากรูปที่ 5.30 5.31 และ 5.32 แสดงการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม พบว่าค่าเวลาประวิงโดยเฉลี่ยของเซลล์จะมีสูงในทุกๆย่านของ offered load และเมื่อเพิ่มขนาดความยาวของเอนเวโลปค่าเวลาประวิงโดยเฉลี่ยของเซลล์ที่ได้ก็มีค่าสูง ซึ่งจึงไม่สามารถรองรับค่า QoS ที่รับประกันได้

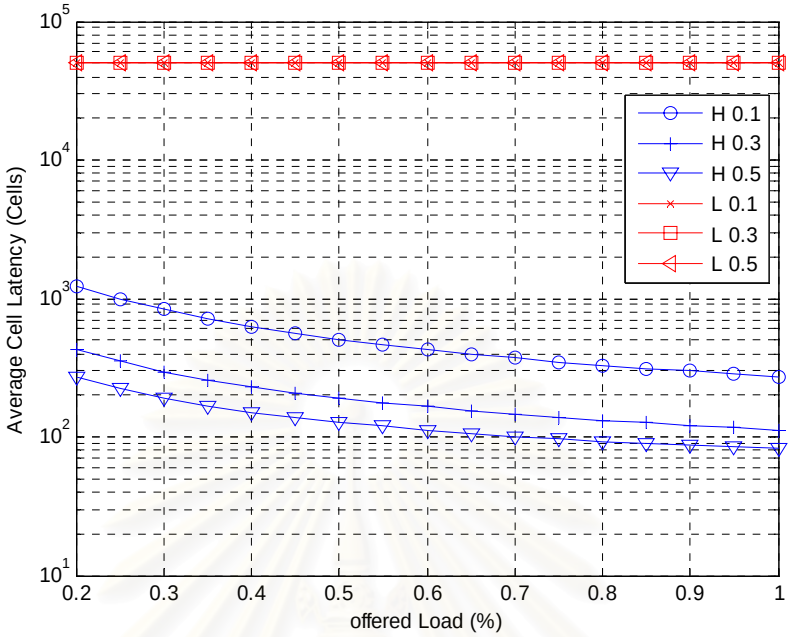
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.33 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท โดยใช้อัลกอริทึม prioritized iSLIP



รูปที่ 5.34 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท อัลกอริทึม prioritized iSLIP โดยวิธีเลือก เอนเวโลปที่มีเซลล์ที่หัวคิว envelope size = 4 cells

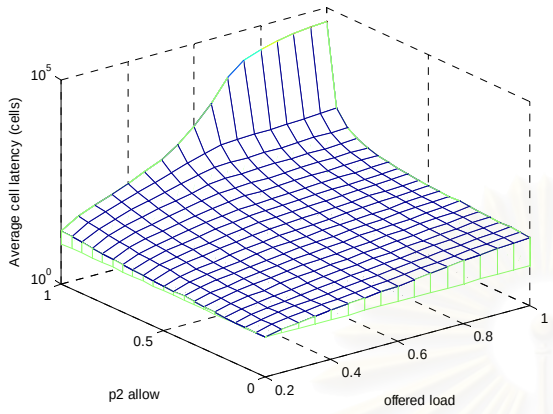


รูปที่ 5.35 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท อัลกอริทึม prioritized iSLIPโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม envelope size = 4 cells

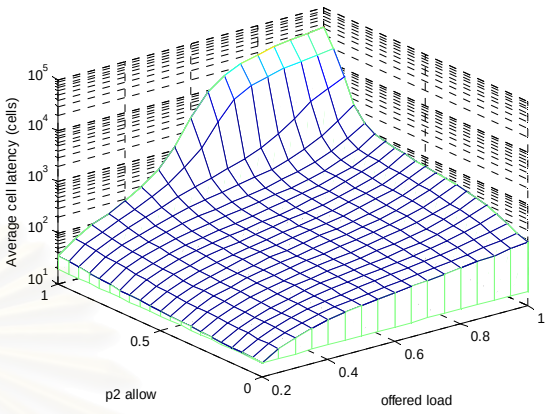
5.6 ผลการทดสอบสมรรถนะของอัลกอริทึมที่นำเสนอที่มีเทคนิคเอนเวโลป

เพื่อวิเคราะห์สมรรถนะของอัลกอริทึมที่นำเสนอที่มีเทคนิคเอนเวโลป จะใช้อัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope), โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) อัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope), โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) และอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope), โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) ซึ่งผลการทดลองมีดังนี้

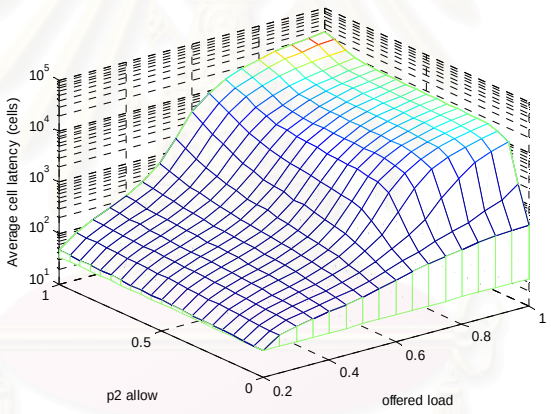
5.6.1 อัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) ขนาดความยาวเอนเวโลป 4 เซลล์



สัดส่วนเซลล์คลาส 1 10%



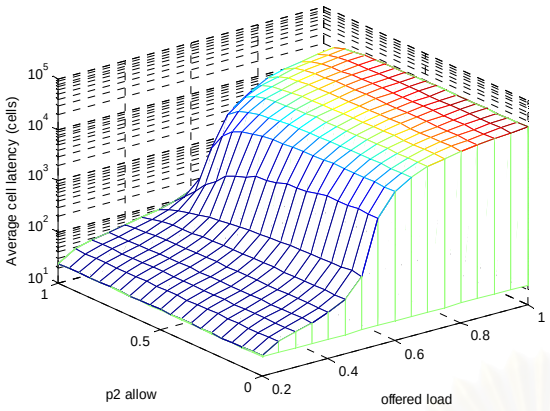
สัดส่วนเซลล์คลาส 1 30%



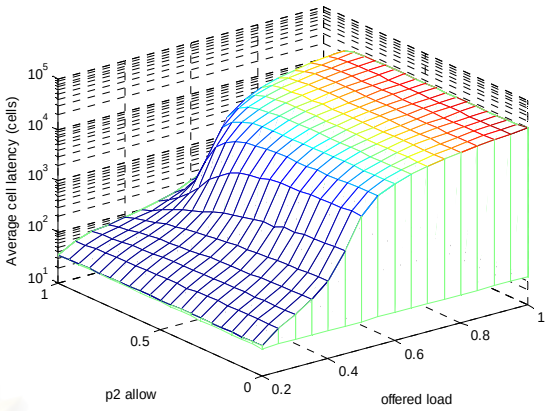
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.36 เวลาประวิงของเซลล์คลาส 1 envelope size =4 cells

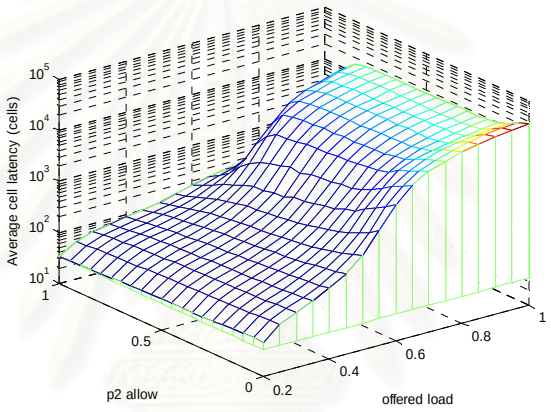
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%



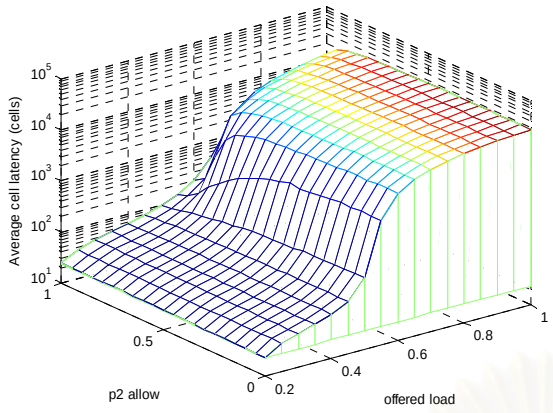
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.37 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

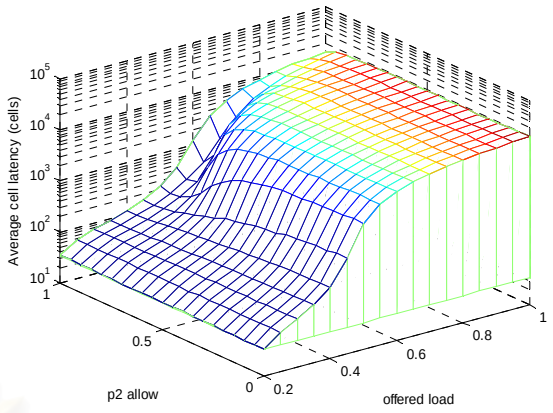
รูปที่ 5.36 และ 5.37 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเวลาประวิงของเซลล์คลาส 1 และ 2 จะมากกว่าอัลกอริทึมการควบคุมที่ request เดิมตอนที่ยังไม่มีเทคนิคเอนเวโลป ซึ่งก็จะเหมือนกับอัลกอริทึม iSLIP ที่ใช้เอนเวโลปขนาดความยาว 4 เซลล์ ดังรูปที่ 5.25 และอัลกอริทึม Prioritized iSLIP ที่ใช้เอนเวโลปขนาดความยาว 4 เซลล์ ดังรูปที่ 5.27-5.29 ที่ค่าเวลาประวิงเฉลี่ยจะเพิ่มขึ้นมากกว่าตอนที่ยังไม่มีเทคนิคเอนเวโลป และเพิ่มมากขึ้นเมื่อขนาดความยาวเอนเวโลปเพิ่มมากขึ้นซึ่งเราไม่สามาถหลีกเลี่ยงได้ เราสามารถทำนายต่อไปได้ว่าเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้นค่าเวลาประวิงของเซลล์คลาส 1 และ 2 ก็จะเพิ่มมากขึ้นด้วย ส่วนลักษณะของกราฟอื่นก็จะคล้ายคลึงกับอัลกอริทึมการควบคุมที่ request เดิม คือเมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลา

ประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง การปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น และเวลาประวิงของเซลล์คลาส 2 ลดต่ำลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมจะเพิ่มขึ้น ส่วนค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง ณ จุดทำงานที่กำหนดให้ใช้ค่า p_2 เท่ากัน การเพิ่มขึ้นของสัดส่วนปริมาณเซลล์คลาส 1 จาก 10% เป็น 50% มีผลทำให้ค่าเวลาประวิงของเซลล์ทั้งสองคลาสเพิ่มขึ้น ลักษณะของผลการทดสอบที่ได้สามารถอธิบายได้ดังว่า เมื่อเซลล์คลาส 1 เพิ่มขึ้นในขณะที่ปริมาณทราฟฟิกยังคงเท่าเดิมทำให้ความรุนแรงในการแย่งใช้บริการระหว่างเซลล์ทั้งสองคลาสสูงขึ้น ส่งผลกระทบต่อเซลล์ทั้งสองคลาสทำให้โอกาสในการได้รับบริการลดลง จนทำให้ค่าเวลาประวิงของเซลล์แต่ละคลาสเพิ่มขึ้น จนเมื่อเพิ่มค่า p_2 ขึ้นเพื่อเพิ่มโอกาสในการให้บริการแก่เซลล์คลาส 2 ขึ้น จึงทำให้ค่าเวลาประวิงของเซลล์คลาส 2 ลดลง ส่วนค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มสูงขึ้นดังจะเห็นได้จากกราฟแต่ละกราฟในรูปที่ 5.36 และ 5.37

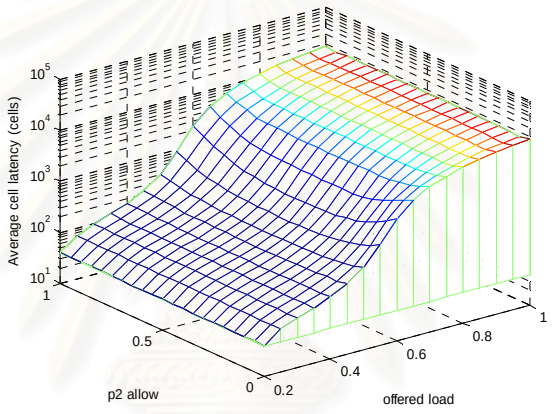
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

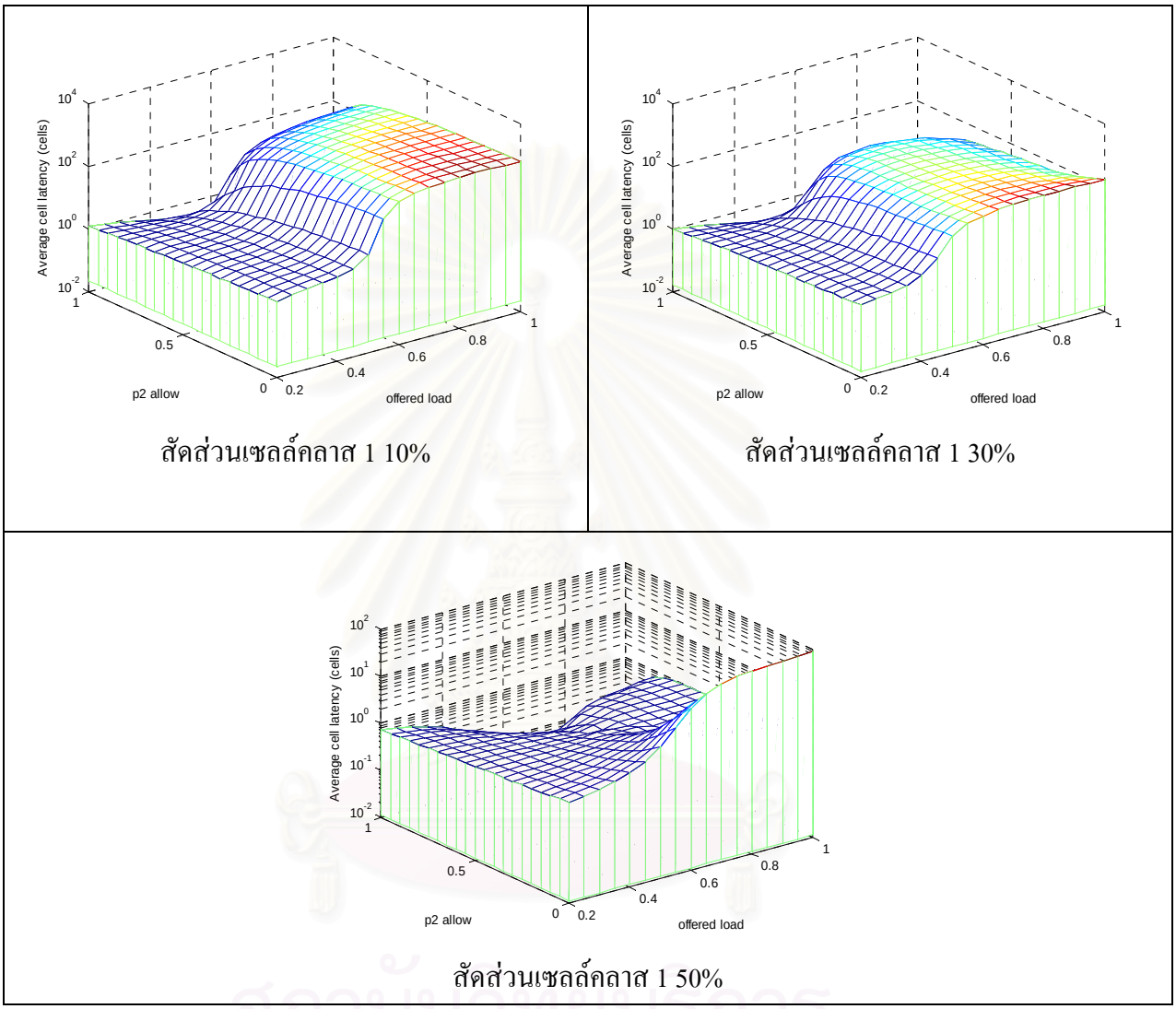


สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.38 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

รูปที่ 5.38 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่า คือ 10% , 30% และ 50 % เราสังเกตได้ว่าเวลาประวิงโดยเฉลี่ยจะมีค่าเพิ่มมากขึ้นเมื่อ offered load เพิ่มขึ้น ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เมื่อเรากำหนดให้ค่า p_2 มีค่าเพิ่มขึ้น ค่าเวลาประวิงโดยเฉลี่ยจะเปลี่ยนแปลงเพียงเล็กน้อย ซึ่งจะแตกต่างกับเวลาประวิงของเซลล์คลาส 1 และ 2 ที่จะเปลี่ยนแปลงเมื่อค่า p_2 เปลี่ยนแปลง เมื่อสัดส่วนเซลล์คลาส 1 เพิ่มมากขึ้น เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว จะเปลี่ยนแปลงเพียงเล็กน้อย เรายังพบอีกว่าเวลาประวิงของเซลล์โดยเฉลี่ยจะมากกว่าอัลกอริทึมการควบคุมที่ request เดิมตอนที่ยังไม่มีเทคนิคเอนเวโลป ซึ่งก็จะเหมือนกับอัลกอริทึม iSLIP และอัลกอริทึม Prioritized iSLIP ที่ค่าเวลาประวิงเฉลี่ยจะเพิ่มขึ้นมากกว่าตอนที่ยังไม่มี

เทคนิคเอนเวโลป และเพิ่มมากขึ้นเมื่อขนาดความยาวเอนเวโลปเพิ่มมากขึ้นซึ่งเราไม่สามาถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดความยาวเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์จะหาค่าการแมตช์ได้ทัน

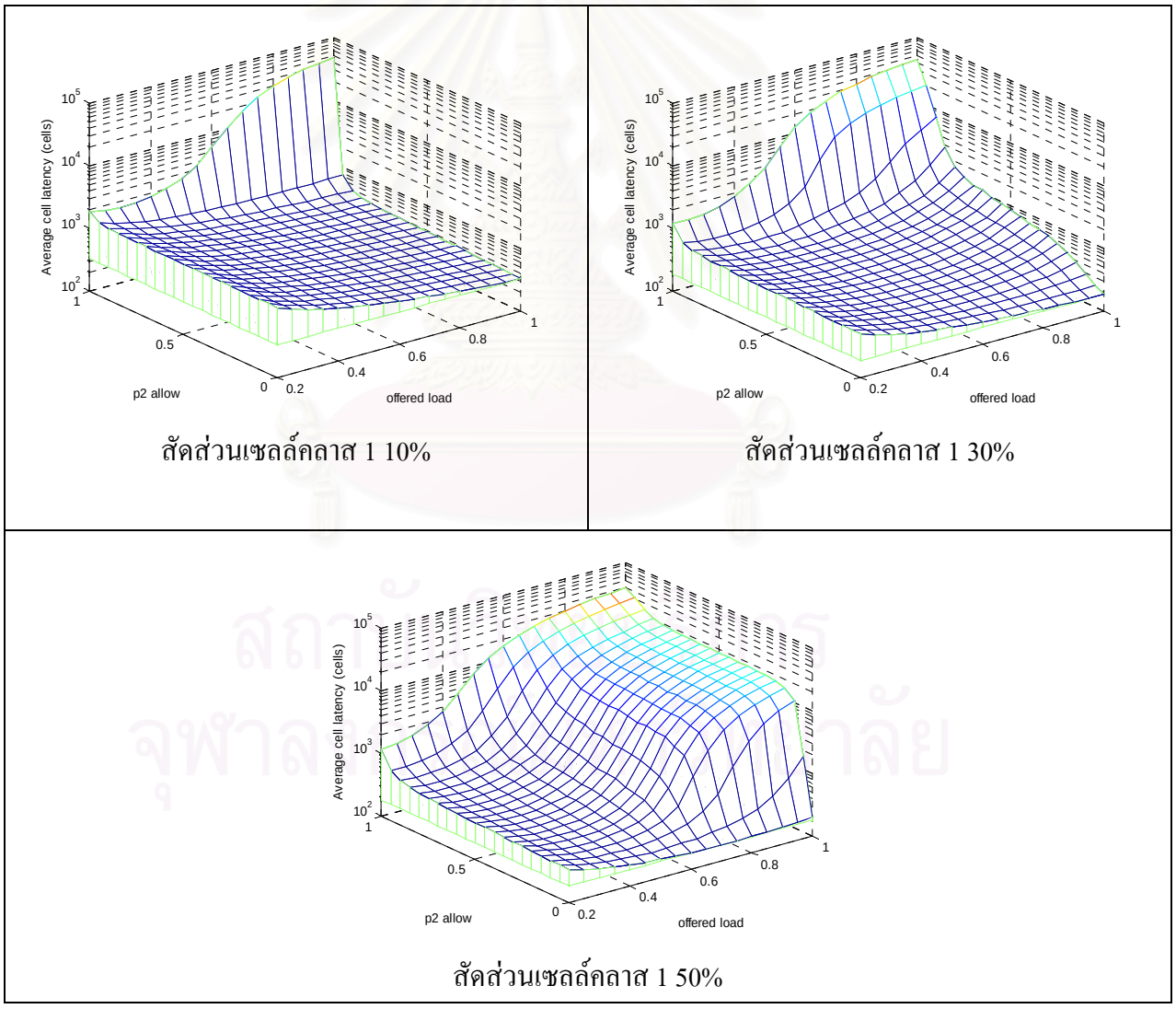


รูปที่ 5.39 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1 envelope size = 4 cells

รูปที่ 5.39 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าลักษณะของกราฟจะคล้ายคลึงกับอัลกอริทึมการควบคุมที่ request เดิม เช่น การปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่ม

หรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง และที่สัดส่วนเซลล์คลาส 1 มีค่า 50% กำหนด offered load ค่าหนึ่งเราจะได้กราฟระหว่างอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 และค่า p_2 สมมาตรกันที่ค่า p_2 เท่ากับ 0.5 โดยจุดสูงสุดและต่ำสุดของกราฟจะหาได้จากการกำหนดค่า p_2 ที่ 0% และ 100% ตามลำดับ ค่าที่เราสนใจคือที่อัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 มีค่าเป็น 10 และ 100 ซึ่งจะได้กล่างถึงในส่วนต่อไป

5.6.2 อัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)ขนาดความยาวเอนเวโลป 4 เซลล์

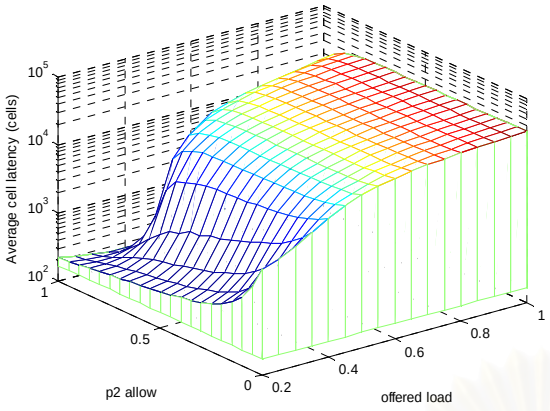


สัดส่วนเซลล์คลาส 1 10%

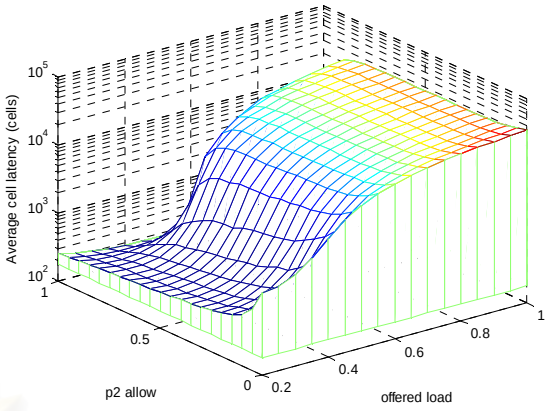
สัดส่วนเซลล์คลาส 1 30%

สัดส่วนเซลล์คลาส 1 50%

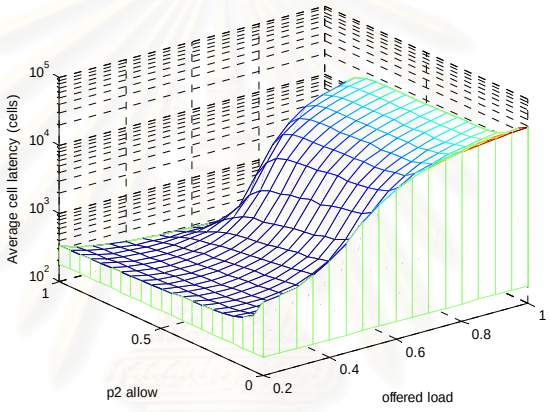
รูปที่ 5.40 เวลาประวิงของเซลล์คลาส 1 envelope size = 4 cells



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

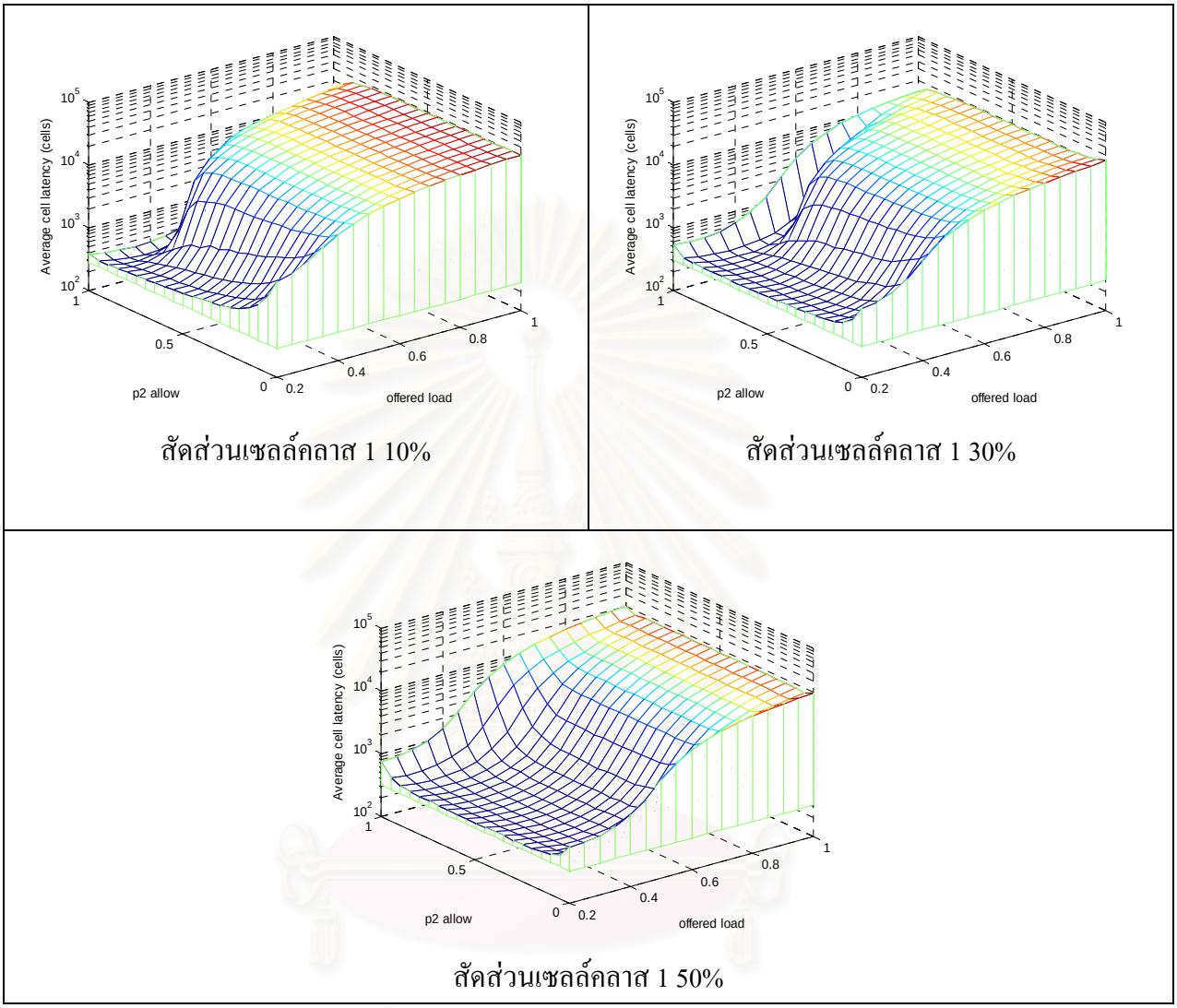


สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.42 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

รูปที่ 5.40 และ 5.42 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50% พบว่าการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มนั้น ให้ค่าเวลาประวิงของเซลล์คลาส 1 สูงมากในช่วงทราฟฟิกต่ำ ๆ จึงไม่สามารถรองรับค่า QoS ได้ ส่วนค่าเวลาประวิงของเซลล์คลาส 2 จะเพิ่มมากขึ้นเมื่อ offered load สูงขึ้น การปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น และเวลาประวิงของเซลล์คลาส 2 ลดต่ำลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมจะเพิ่มขึ้น โดยเฉพาะเมื่อสัดส่วนเซลล์คลาส 1 เป็น 50 % เวลาประวิงของเซลล์คลาส 1 จะเพิ่มมากขึ้น

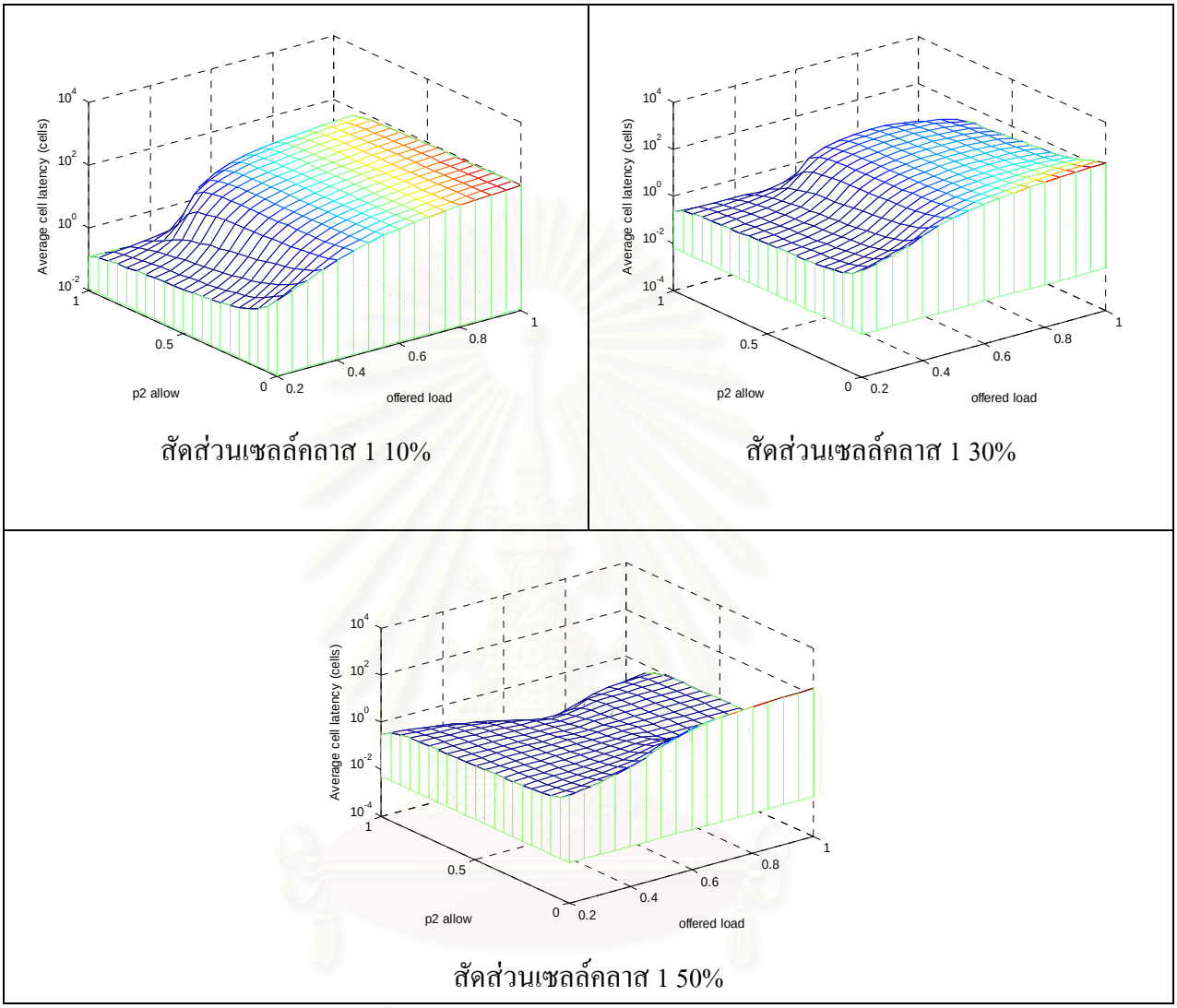
ส่วนค่าประวิงเวลาของเซลล์คลาส 2 โดยรวมจะลดลง ซึ่งจะเหมือนกับอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว



รูปที่ 5.43 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

รูปที่ 5.43 แสดงเวลาประวิงของเซลล์โดยเฉลี่ย ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50% พบว่าเวลาประวิงของเซลล์โดยเฉลี่ยจะมีค่าที่สูงในทุกๆย่านของ offered load แต่เราสามารถทำให้ต่ำลงได้ด้วยการปรับค่า p_2 ซึ่งที่สัดส่วนปริมาณเซลล์คลาส 1 เป็น 10% และ 30% ค่าเวลาประวิงโดยเฉลี่ยจะลดลงชัดเจน แต่เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เป็น 50% ค่าเวลาประวิงจะไม่เปลี่ยนแปลงมากนักเมื่อค่า p_2 เปลี่ยนแปลง เมื่อสัดส่วน

เซลล์คลาส 1 เพิ่มมากขึ้น เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม จะเปลี่ยนแปลงเพียงเล็กน้อย

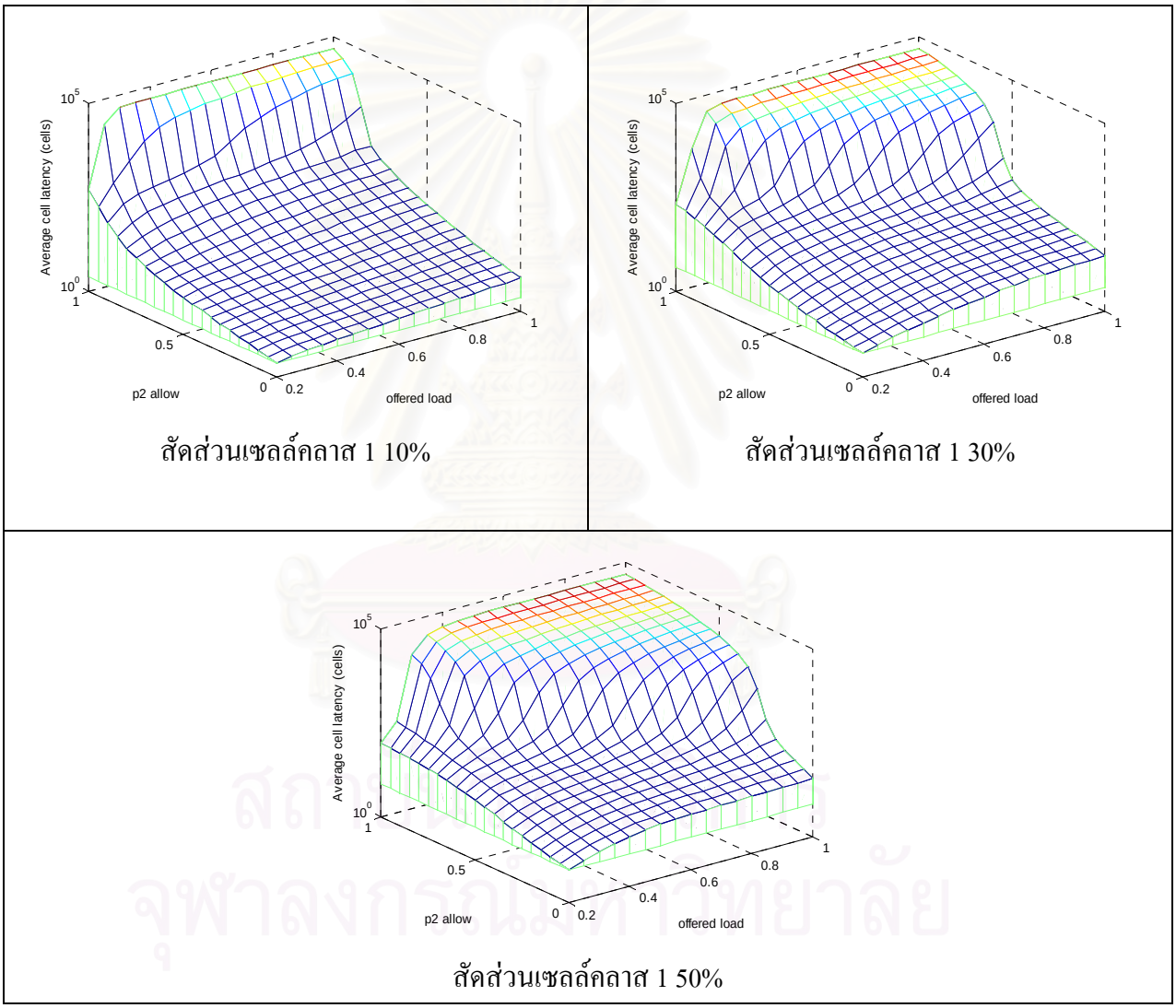


รูปที่ 5.44 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1 envelope size = 4 cells

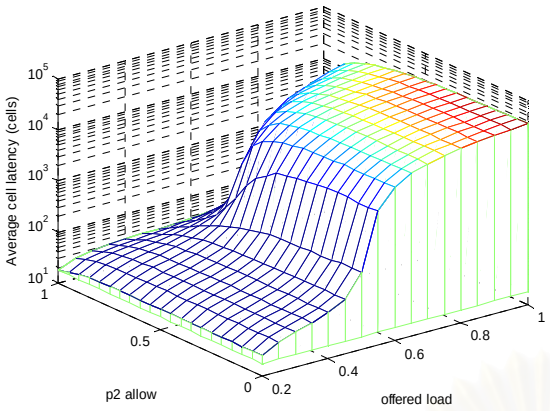
รูปที่ 5.44 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าลักษณะของกราฟจะคล้ายคลึงกับอัลกอริทึมการควบคุมที่ request เดิม เช่น การปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load

และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง ค่าที่เราสนใจคืออัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 มีค่าเป็น 10 และ 100 ซึ่งจะได้กล่าวถึงในส่วนต่อไป

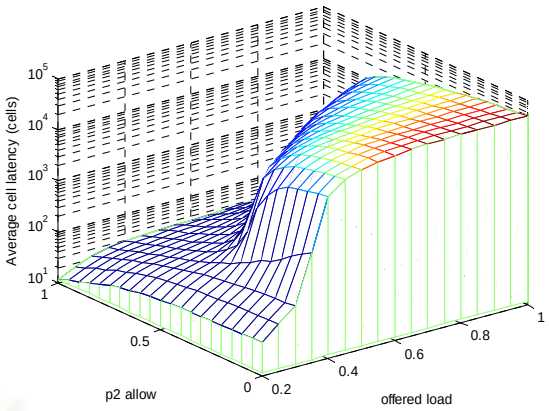
5.6.3 อัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)ขนาดความยาวเอนเวโลป 4 เซลล์



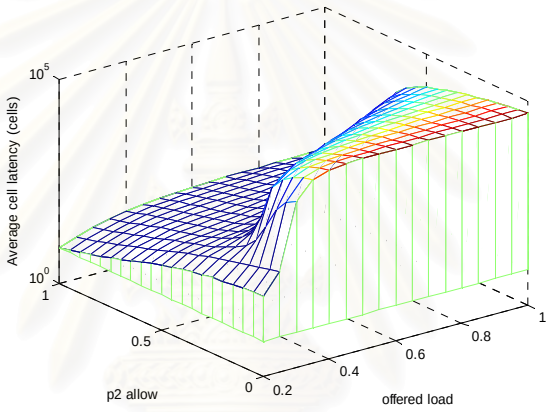
รูปที่ 5.45 เวลาประวิงของเซลล์คลาส 1 envelope size =4 cells



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

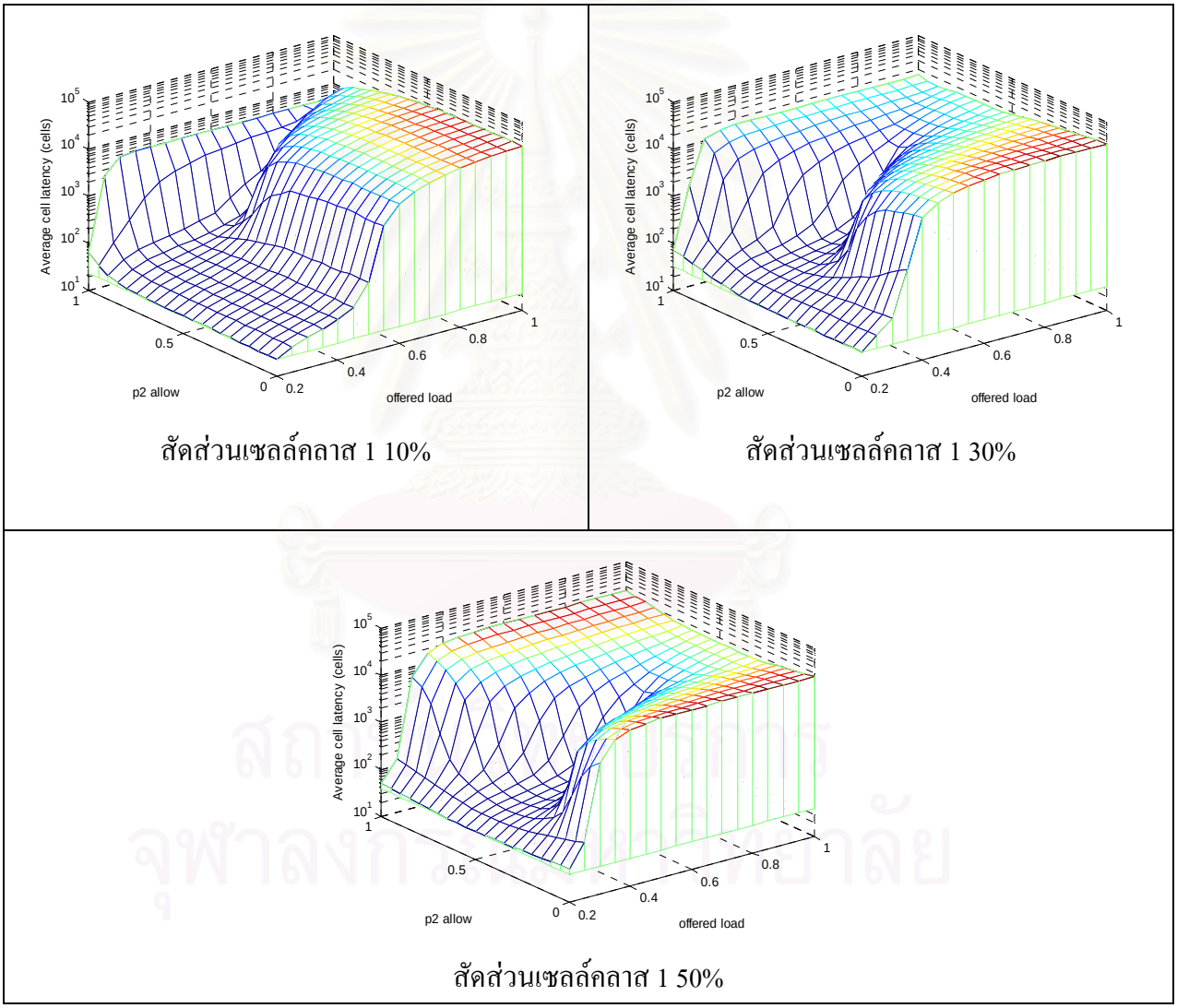


สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.46 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

รูปที่ 5.45 และ 5.46 แสดงเวลาประวิงขงเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเวลาประวิงของเซลล์คลาส 1 และ 2 จะมากกว่าอัลกอริทึมการควบคุมที่ grant เดิมตอนที่ยังไม่มีเทคนิคเอนเวโลป ทำให้เราสามารถทำนายต่อไปได้ว่าเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้นค่าเวลาประวิงของเซลล์คลาส 1 และ 2 ก็จะเพิ่มมากขึ้นด้วย ในการเพิ่มขนาดของเอนเวโลปให้ยาวขึ้นจะทำให้ค่าเวลาประวิงเฉลี่ยของเซลล์เพิ่มสูงขึ้นจนสวิตช์ไม่สามารถรับประกันในคุณภาพการให้บริการได้ ปัญหาก็คือ เราจะเพิ่มขนาดเอนเวโลปยาวไม่เกินเท่าใด จึงจะทำให้สวิตช์สามารถประมวลผลได้ทันจนสามารถขจัดปัญหาคอขวดในการสวิตช์ข้อมูลได้ และยังสามารถรับประกันในคุณภาพการให้บริการ ลักษณะ

ของกราฟอื่นก็จะคล้ายคลึงกับอัลกอริทึมการควบคุมที่ grant เดิม คือเมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง การปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น และเวลาประวิงของเซลล์คลาส 2 ลดต่ำลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมจะเพิ่มขึ้น ส่วนค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง



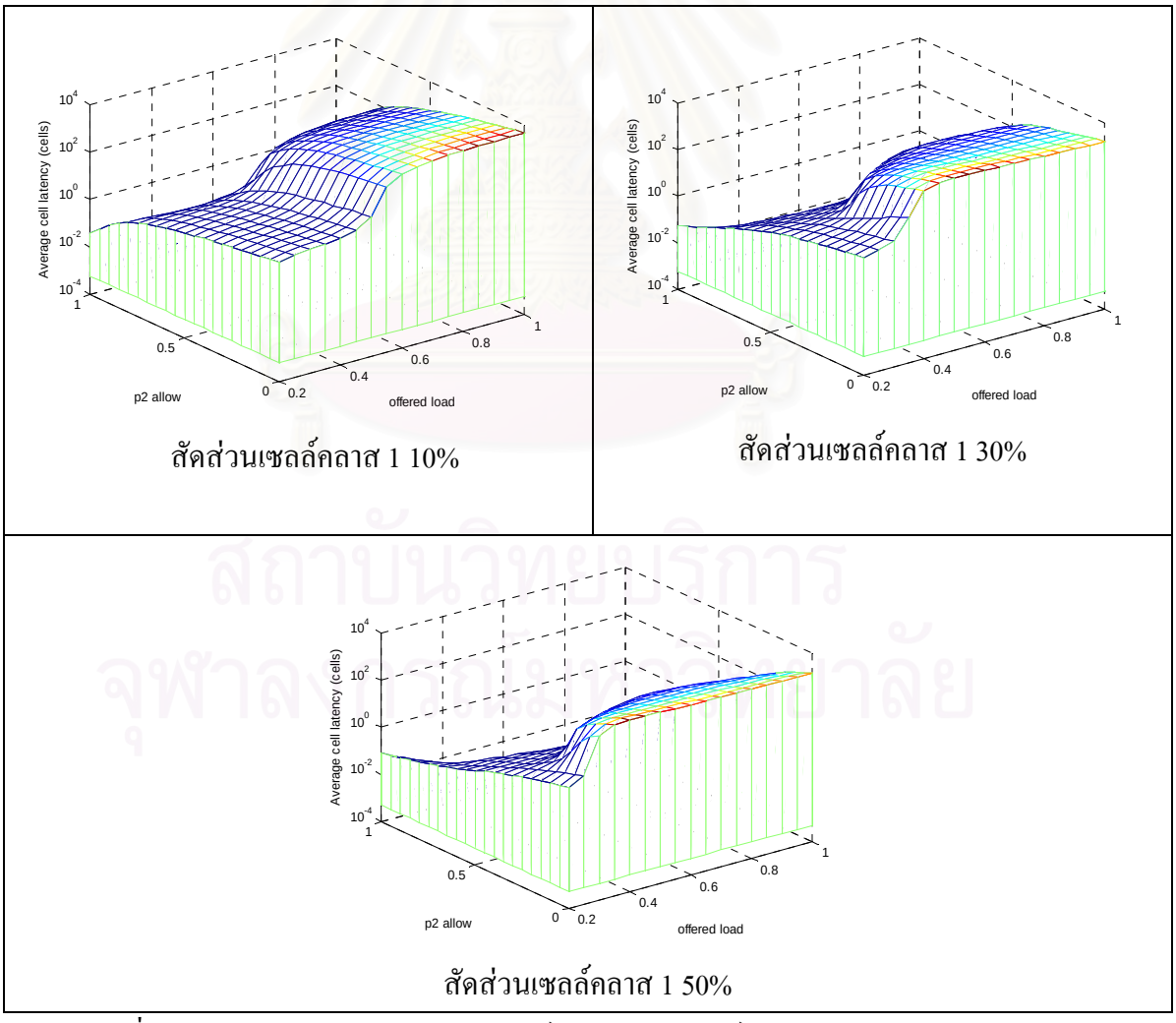
สัดส่วนเซลล์คลาส 1 10%

สัดส่วนเซลล์คลาส 1 30%

สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.47 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

รูปที่ 5.47 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่า คือ 10% , 30% และ 50 % พบว่าค่าประวิงเวลาโดยเฉลี่ยจะมีค่ามากกว่าอัลกอริทึมการควบคุมที่ grant ตอนที่ยังไม่มีเทคนิคเอนเวโลป เพราะว่าเทคนิคเอนเวโลปทำให้ตัวจัดลำดับเซลล์มีเวลามากขึ้นในการเลือกแต่ละเอนเวโลปสำหรับส่งผ่านโครงสร้างสวิตช์ แต่ต้องแลกกับสมรรถนะด้านค่าเวลาประวิงที่จะลดลง ลักษณะอื่นคือเวลาประวิงโดยเฉลี่ยจะมีค่าเพิ่มมากขึ้นเมื่อ offered load เพิ่มขึ้น ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เมื่อเรากำหนดให้ค่า p_2 มีค่าเพิ่มขึ้น ค่าเวลาประวิงโดยเฉลี่ยจะเปลี่ยนแปลงไป ซึ่งเมื่อเปรียบเทียบกับอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ค่าจะต่างกัน เมื่อสัดส่วนปริมาณเซลล์คลาส 1 เปลี่ยนแปลง เวลาประวิงโดยเฉลี่ยจะเปลี่ยนแปลงต่างจากอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว จะเห็นว่าเมื่อสัดส่วนเซลล์คลาส 1 เป็น 50% เวลาประวิงโดยเฉลี่ยต่ำที่สุดจะอยู่ที่ค่า p_2 ประมาณ 0.5



สัดส่วนเซลล์คลาส 1 10%

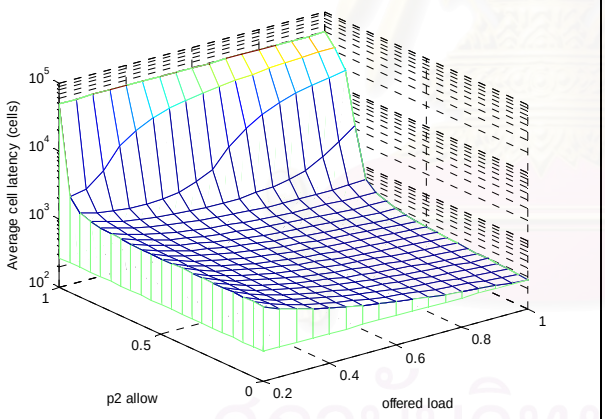
สัดส่วนเซลล์คลาส 1 30%

สัดส่วนเซลล์คลาส 1 50%

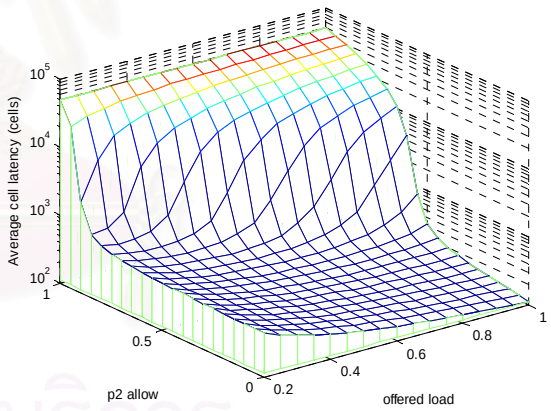
รูปที่ 5.48 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1 envelope size = 4 cells

รูปที่ 5.48 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง เหมือนกับอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว

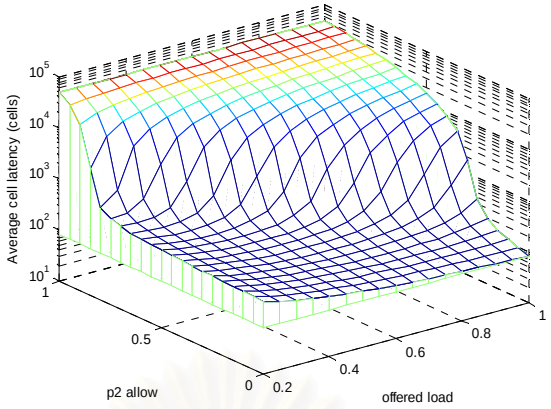
5.6.4 อัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)ขนาดความยาวเอนเวโลป 4 เซลล์



สัดส่วนเซลล์คลาส 1 10%

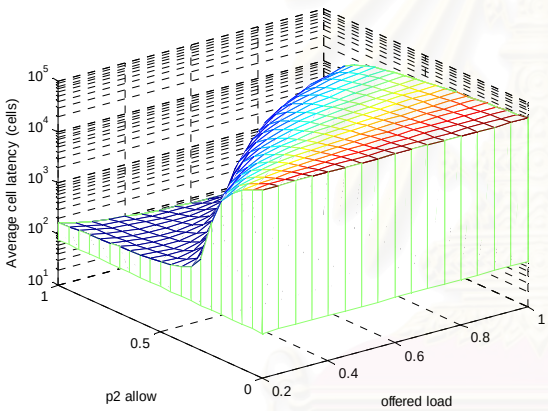


สัดส่วนเซลล์คลาส 1 30%

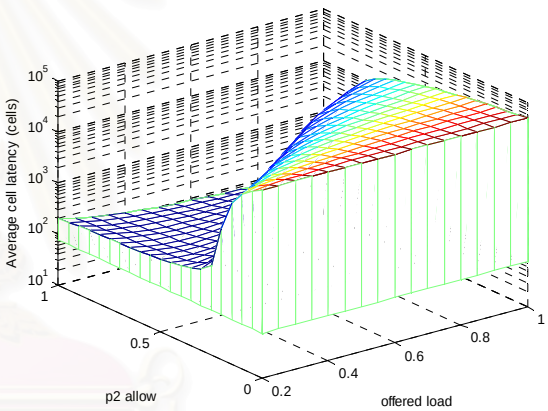


สัดส่วนเซลล์คลาส 1 50%

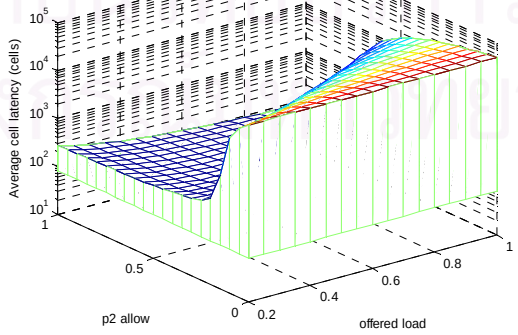
รูปที่ 5.49 เวลาประวิงของเซลล์คลาส 1 envelope size = 4 cells



สัดส่วนเซลล์คลาส 1 10%



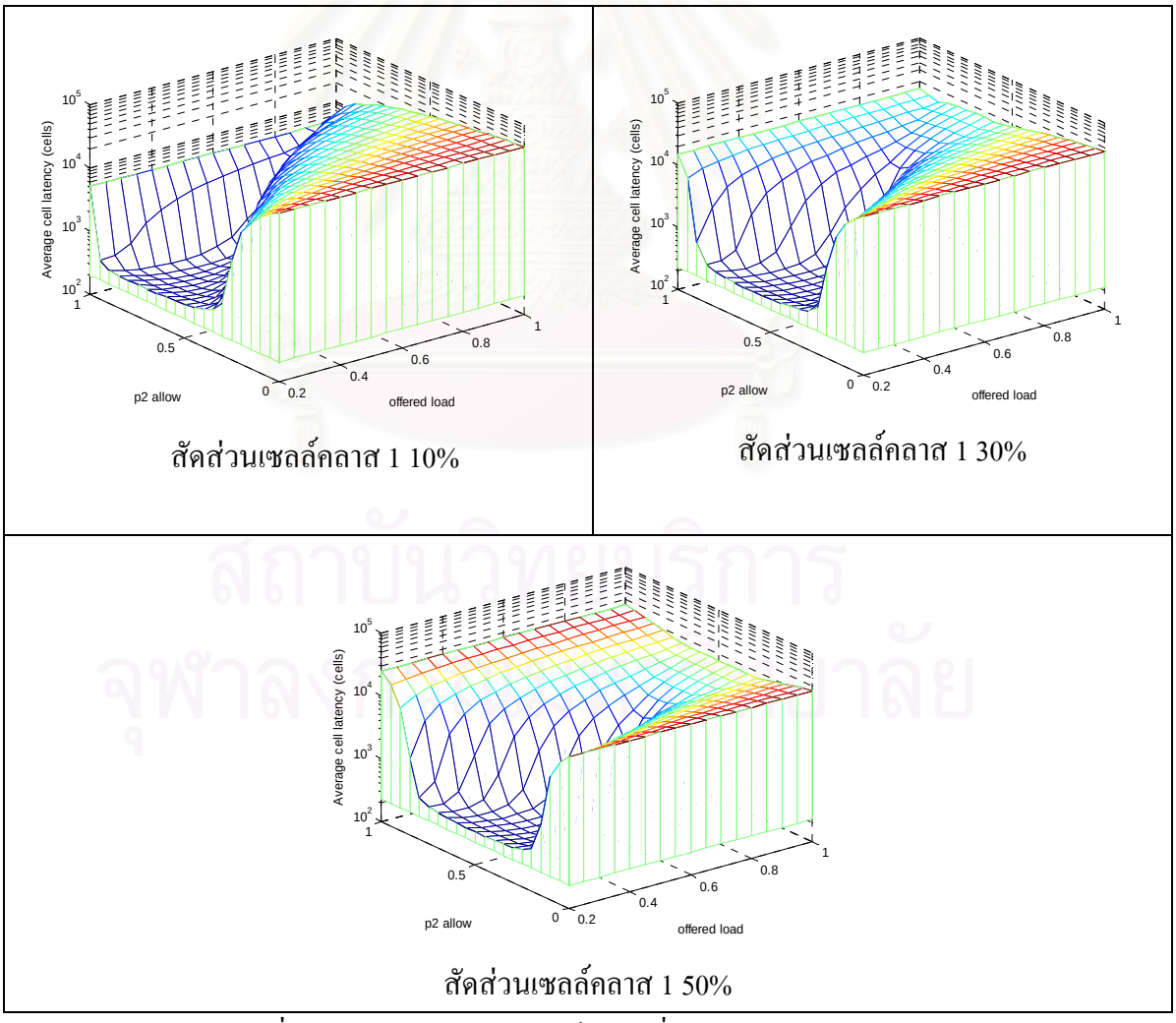
สัดส่วนเซลล์คลาส 1 30%



สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.50 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

รูปที่ 5.49 และ 5.50 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50% เราพบว่าการปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น และเวลาประวิงของเซลล์คลาส 2 ลดต่ำลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 1 โดยรวมจะเพิ่มขึ้น ส่วนค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มนั้น ให้ค่าเวลาประวิงของเซลล์คลาส 1 และคลาส 2 สูงมากในช่วงทราฟฟิกต่ำ ๆ จึงไม่สามารถรองรับค่า QoS ได้ ทำให้การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มไม่เหมาะนำมาใช้กับโครงข่ายความเร็วสูงในอนาคต



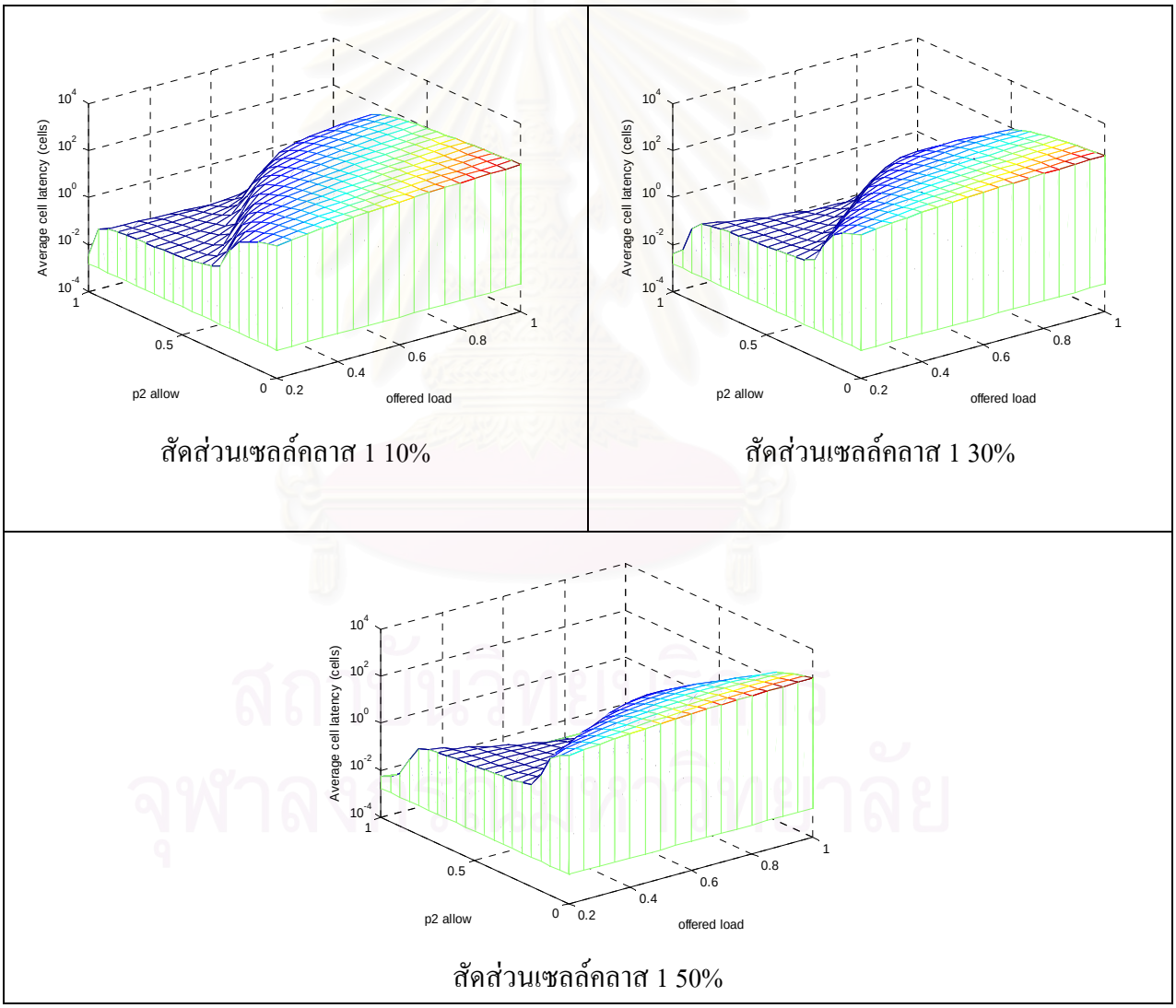
สัดส่วนเซลล์คลาส 1 10%

สัดส่วนเซลล์คลาส 1 30%

สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.51 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

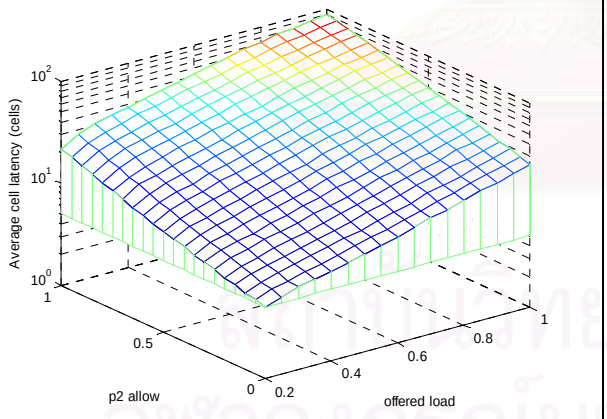
รูปที่ 5.51 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่า คือ 10% , 30% และ 50 % พบว่าเวลาประวิงของเซลล์โดยเฉลี่ยจะมีค่าที่สูงในทุกๆย่านของ offered load แต่ค่าที่ได้จะลดลงอย่างชัดเจนเมื่อค่า p_2 เปลี่ยนแปลงไปดังจะเห็นได้จากแต่ละกราฟในรูปที่ 5.51 เมื่อสัดส่วนเซลล์คลาส 1 เพิ่มมากขึ้น เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม จะเปลี่ยนแปลงและสามารถอธิบายต่อได้ว่าค่าเวลาประวิงโดยเฉลี่ยค่าต่ำๆ จะเปลี่ยนแปลงเมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลง



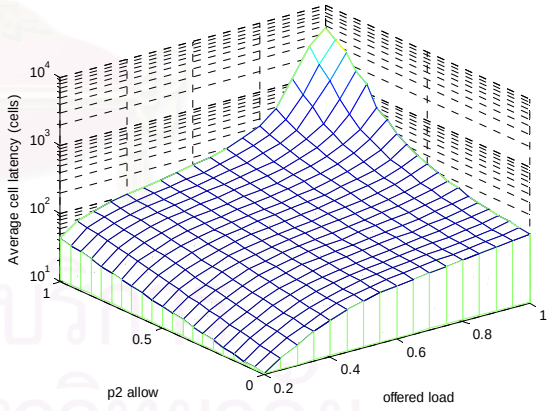
รูปที่ 5.52 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 envelope size = 4 cells

รูปที่ 5.52 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % เราพบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง และที่สัดส่วนเซลล์คลาส 1 มีค่า 50% กำหนด offered load ค่าหนึ่งเราจะได้กราฟระหว่างอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 และค่า p_2 สมมาตรกันที่ค่า p_2 เท่ากับ 0.5 โดยจุดสูงสุด และต่ำสุดของกราฟจะหาได้จากการกำหนดค่า p_2 ที่ 0% และ 100% ตามลำดับ ค่าที่เราสนใจคือที่อัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 มีค่าเป็น 10 และ 100 ซึ่งจะได้กล่าวถึงในส่วนต่อไป

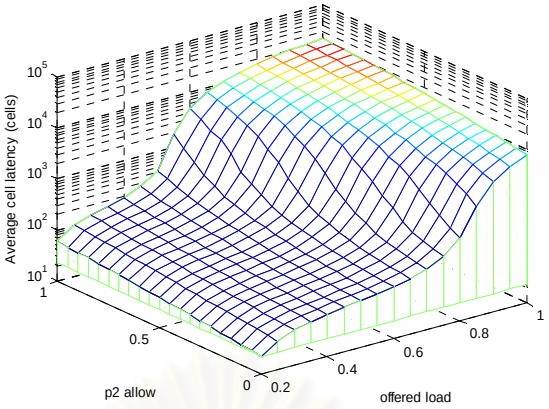
5.6.5 อัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)ขนาดความยาวเอนเวโลป 4 เซลล์



สัดส่วนเซลล์คลาส 1 10%

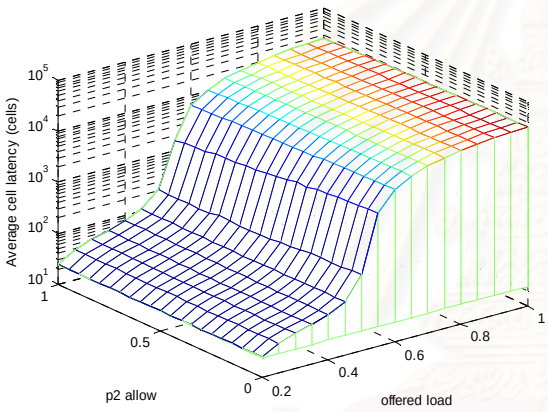


สัดส่วนเซลล์คลาส 1 30%

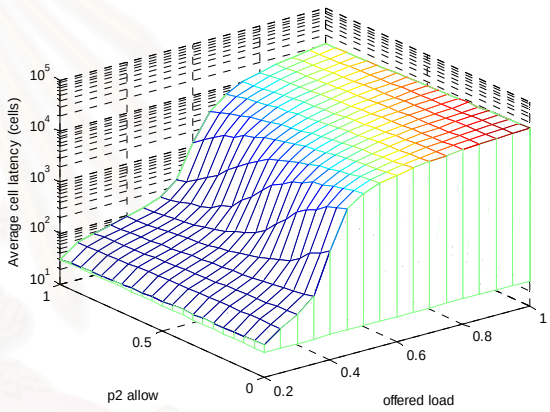


สัดส่วนเซลล์คลาส 1 50%

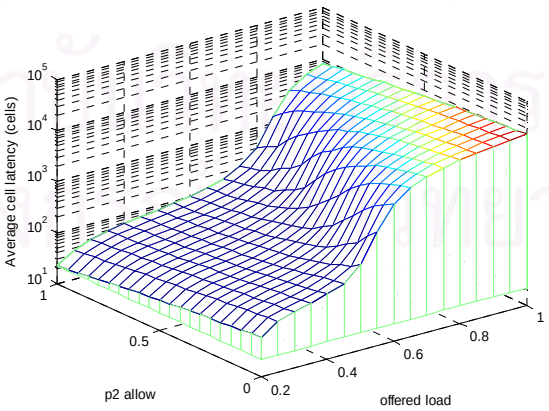
รูปที่ 5.53 เวลาประวิงของเซลล์คลาส 1 envelope size = 4 cells



สัดส่วนเซลล์คลาส 1 10%



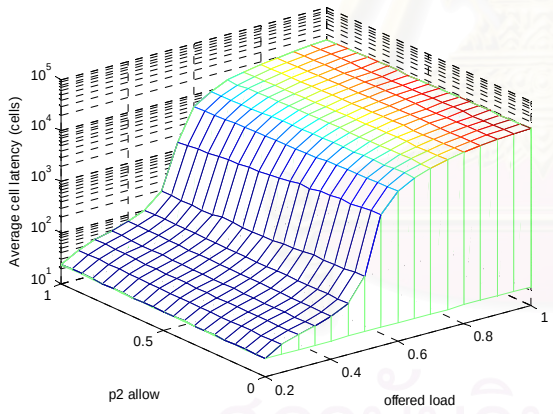
สัดส่วนเซลล์คลาส 1 30%



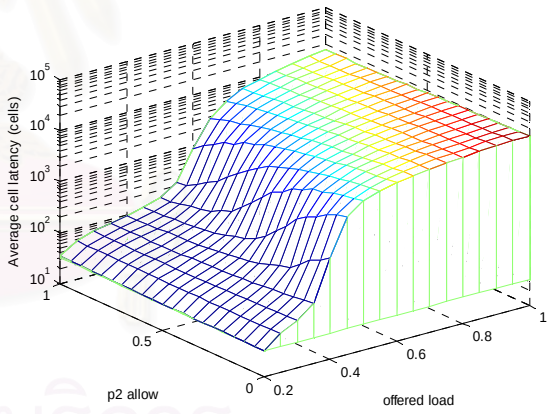
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.54 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

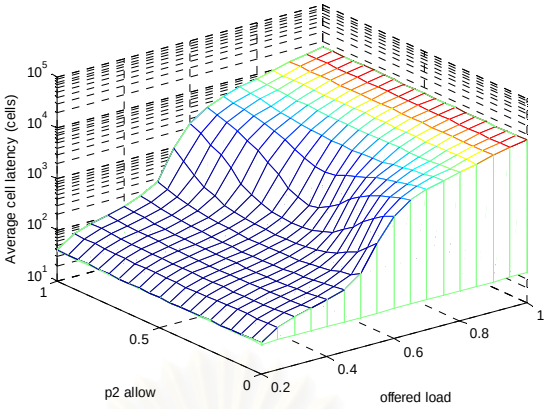
รูปที่ 5.53 และ 5.54 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าเวลาประวิงของเซลล์คลาส 1 และ 2 จะมากกว่าอัลกอริทึมการควบคุมที่ accept เดิมตอนที่ยังไม่มีเทคนิคเอนเวโลป ทำให้เราสามารถทำนายต่อไปได้ว่าเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้นค่าเวลาประวิงของเซลล์คลาส 1 และ 2 ก็จะเพิ่มมากขึ้นด้วย เมื่อปริมาณ offered load เพิ่มขึ้น ค่าเวลาประวิงของเซลล์คลาส 1 จะเพิ่มขึ้นด้วย ซึ่งเป็นไปตามที่คาดไว้เนื่องจากสวิตช์ต้องรองรับทราฟฟิกมากขึ้นนั่นเอง เช่นเดียวกับอัลกอริทึมควบคุมที่ request และอัลกอริทึมควบคุมที่ grant การปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น แต่ส่งผลเพียงเล็กน้อยเวลาประวิงของเซลล์คลาส 2 แต่ที่สัดส่วนเซลล์คลาส 1 50% เวลาประวิงของเซลล์คลาส 2 จะลดลงมากขึ้น ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง



สัดส่วนเซลล์คลาส 1 10%



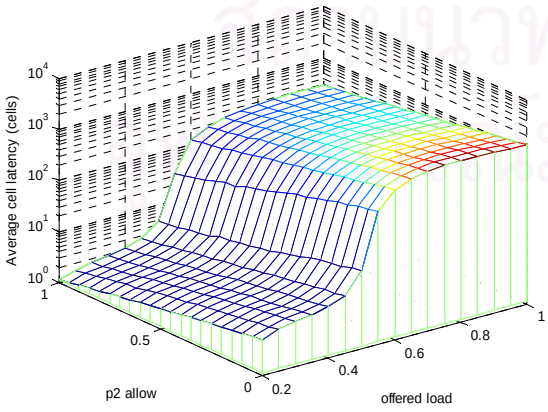
สัดส่วนเซลล์คลาส 1 30%



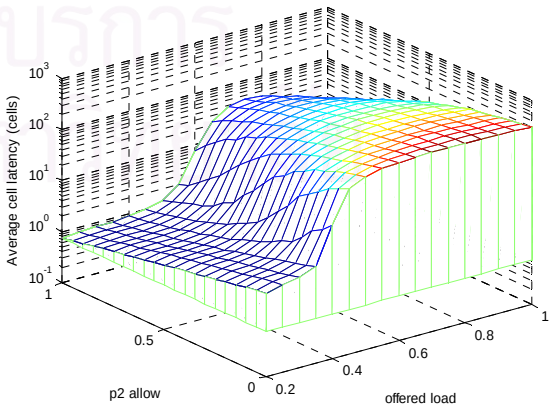
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.55 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

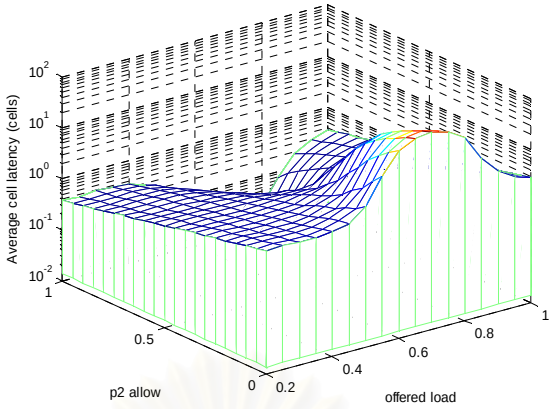
รูปที่ 5.55 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่า คือ 10% , 30% และ 50 % พบว่าเวลาประวิงโดยเฉลี่ยจะเพิ่มขึ้นเมื่อ offered load เพิ่มขึ้นและจะไม่เปลี่ยนแปลงตามค่า p_2 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงค่าประวิงเวลาเฉลี่ยจะเปลี่ยนแปลงเพียงเล็กน้อย เรายังพบว่าเวลาประวิงของเซลล์โดยเฉลี่ยจะมากกว่าอัลกอริทึมการควบคุมที่ accept เดิมตอนที่ยังไม่มีเทคนิคเอนเวโลป ซึ่งก็จะเหมือนกับอัลกอริทึม iSLIP และอัลกอริทึม Prioritized iSLIP ที่ค่าเวลาประวิงเฉลี่ยจะเพิ่มขึ้นเมื่อขนาดความยาวเอนเวโลปเพิ่มมากขึ้นและเราไม่สามารถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดความยาวเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์จะหาค่าการแมตช์ได้ทัน



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%

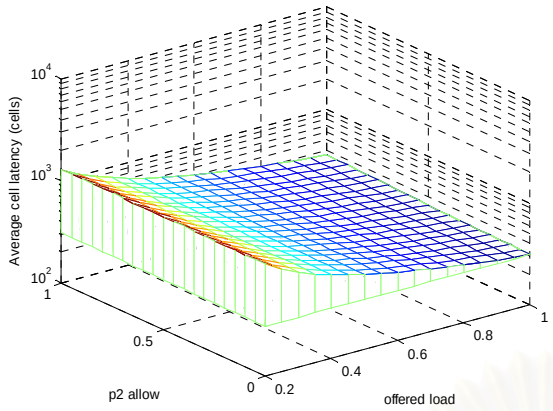


สัดส่วนเซลล์คลาส 1 50%

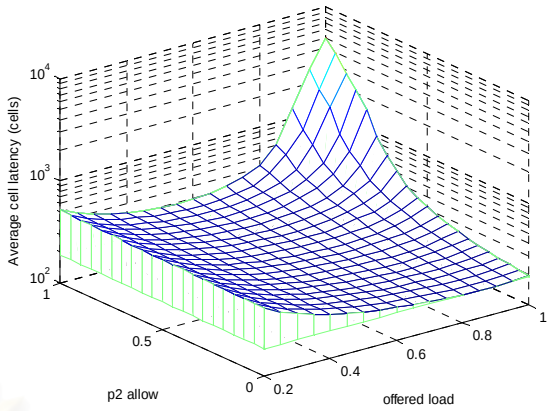
รูปที่ 5.56 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1 envelope size = 4 cells

รูปที่ 5.56 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง เหมือนกับอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวและอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว

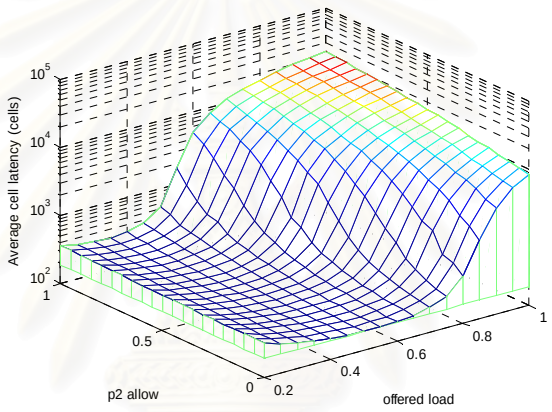
5.6.6 อัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)ขนาดความยาวเอนเวโลป 4 เซลล์



สัดส่วนเซลล์คลาส 1 10%

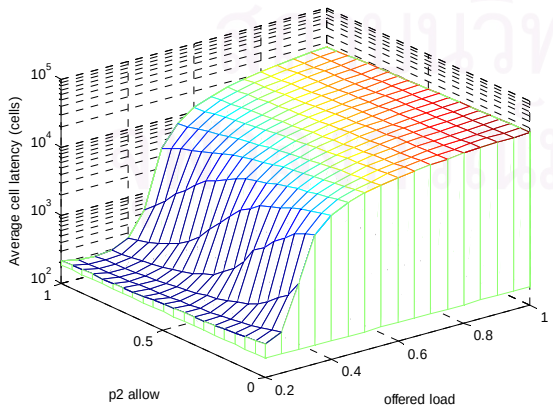


สัดส่วนเซลล์คลาส 1 30%

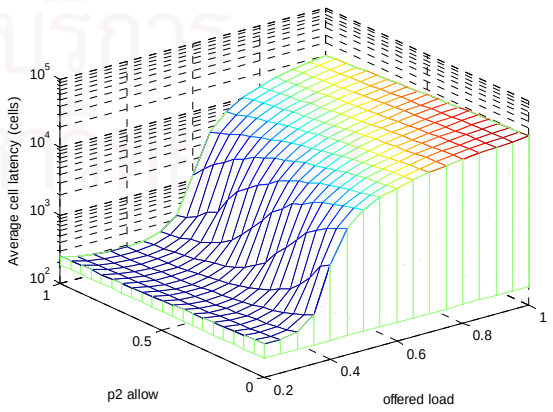


สัดส่วนเซลล์คลาส 1 50%

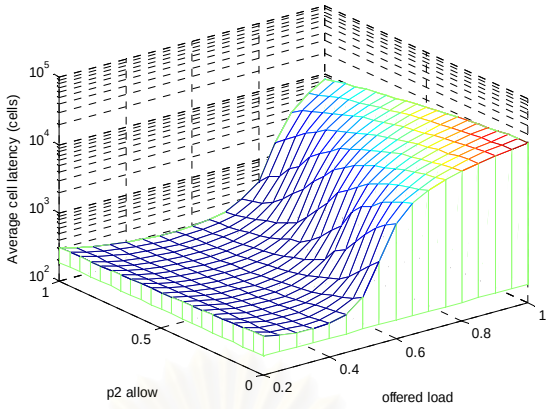
รูปที่ 5.57 เวลาประวิงของเซลล์คลาส 1 envelope size = 4 cells



สัดส่วนเซลล์คลาส 1 10%



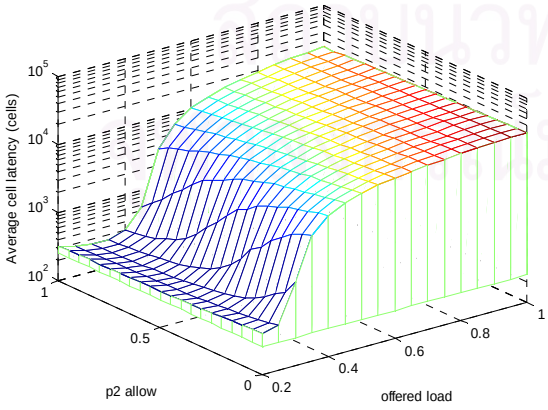
สัดส่วนเซลล์คลาส 1 30%



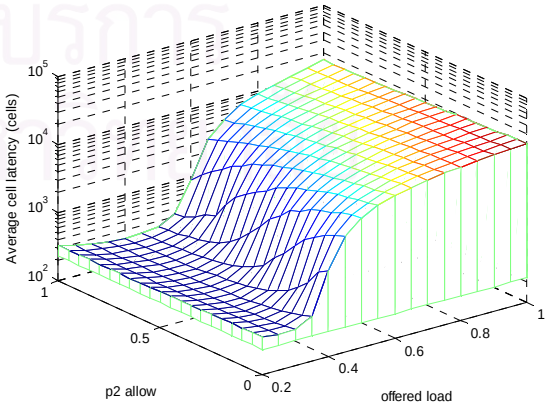
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.58 เวลาประวิงของเซลล์คลาส 2 envelope size = 4 cells

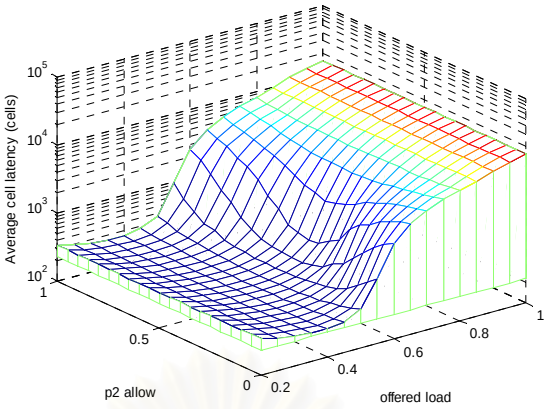
รูปที่ 5.57 และ 5.58 แสดงเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50% พบว่าการปรับขึ้นของค่า p_2 จะส่งผลกระทบทำให้ค่าเวลาประวิงของเซลล์คลาส 1 เพิ่มสูงขึ้น และเวลาประวิงของเซลล์คลาส 2 ลดต่ำลง ผลที่เกิดขึ้นนี้หมายความว่าเราสามารถควบคุมค่าเวลาประวิงของเซลล์คลาส 1 และเซลล์คลาส 2 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 เมื่อปริมาณเซลล์คลาส 1 เพิ่มขึ้นจาก 10%-50% ค่าเวลาประวิงของเซลล์คลาส 2 โดยรวมจะลดลง การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มนั้น ให้ค่าเวลาประวิงของเซลล์คลาส 1 สูงมากในช่วงทราฟฟิกต่ำ ๆ จึงไม่สามารถรองรับค่า QoS ได้ ส่วนค่าเวลาประวิงของเซลล์คลาส 2 จะเพิ่มมากขึ้นเมื่อ offered load สูงขึ้น



สัดส่วนเซลล์คลาส 1 10%



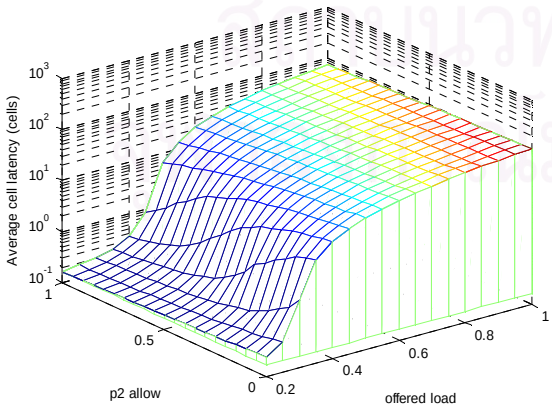
สัดส่วนเซลล์คลาส 1 30%



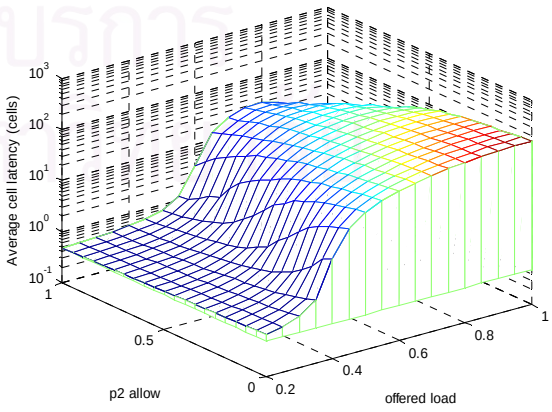
สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.59 เวลาประวิงของเซลล์โดยเฉลี่ย envelope size = 4 cells

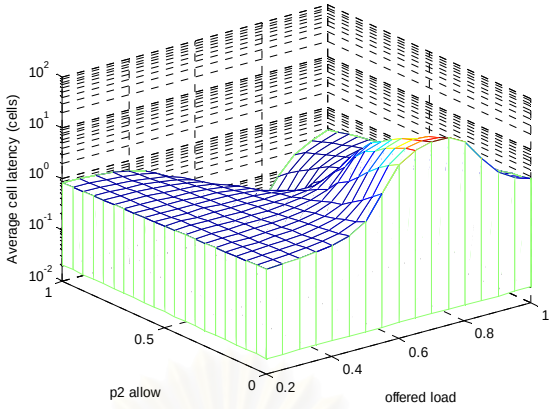
รูปที่ 5.59 แสดงเวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % เราพบว่าค่าประวิงเวลาของเซลล์โดยเฉลี่ยจะมีค่าต่ำในช่วง offered load ช่วงแรก และจะเพิ่มขึ้นเมื่อทราฟฟิกมีค่ามากขึ้น ลักษณะที่เกิดขึ้นก็จะคล้ายกับอัลกอริทึม Prioritized iSLIP โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม เมื่อ p_2 เปลี่ยนแปลง ค่าประวิงเวลาของเซลล์โดยเฉลี่ยจะเปลี่ยนแปลงเพียงเล็กน้อย เมื่อสัดส่วนเซลล์คลาส 1 เพิ่มมากขึ้น เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม จะเปลี่ยนแปลงตามค่า offered load มากกว่าเปลี่ยนแปลงตามค่า p_2



สัดส่วนเซลล์คลาส 1 10%



สัดส่วนเซลล์คลาส 1 30%



สัดส่วนเซลล์คลาส 1 50%

รูปที่ 5.60 อัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อ เซลล์คลาส 1 envelope size = 4 cells

รูปที่ 5.60 แสดงอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ accept โดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว ขนาดความยาวเอนเวโลป 4 เซลล์ ซึ่งมีค่าสัดส่วนปริมาณเซลล์คลาส 1 3 ค่าคือ 10% , 30% และ 50 % พบว่าการปรับขึ้นของค่า p_2 ให้สูงขึ้นจะส่งผลกระทบต่ออัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้ลดต่ำลง ปรากฏการณ์ที่เห็นนี้หมายความว่าเราสามารถควบคุมอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ให้เพิ่มหรือลดลงได้ด้วยการปรับค่าของ p_2 แต่ช่วงที่ควบคุมได้ดังกล่าวนั้นขึ้นอยู่กับปริมาณ offered load และสัดส่วนปริมาณของเซลล์คลาส 1 เมื่อสัดส่วนเซลล์คลาส 1 เปลี่ยนแปลงช่วงการควบคุมอัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จะแคบลง เหมือนกับอัตราส่วนเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ของอัลกอริทึมการควบคุมที่ request โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็มและอัลกอริทึมการควบคุมที่ grant โดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ค่าที่เราสนใจคือที่อัตราส่วนระหว่างค่าเวลาประวิงของเซลล์คลาส 2 ต่อเซลล์คลาส 1 มีค่าเป็น 10 และ 100 ซึ่งจะได้กล่างถึงในส่วนต่อไป

5.7 การวิเคราะห์เชิงสมรรถนะของอัลกอริทึมที่นำเสนอที่มีเทคนิคเอนเวโลป

5.7.1 สมรรถนะด้านการควบคุมคุณภาพการให้บริการ

สมรรถนะด้านเวลาประวิงของอัลกอริทึมที่นำเสนอนั้นจะวัดออกมาโดยการเปรียบเทียบเวลาประวิงโดยเฉลี่ยระหว่างแต่ละอัลกอริทึม ณ จุดที่มีอัตราส่วนเวลาประวิงของเซลล์คลาส 1 ต่อเซลล์คลาส 2, offered load และ สัดส่วนปริมาณเซลล์คลาส 1 เท่ากัน

วิธีที่ 1 วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)
เมื่อขนาดความยาวเอนเวโลป = 4 cells

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	235.23	0.3171
ควบคุมที่ grant	207.52	0.5302
ควบคุมที่ accept	204.79	0.5758

ตารางที่ 5.8 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	14163	0.8854
ควบคุมที่ grant	13000	0.7293
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.9 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	900.69	0.7121
ควบคุมที่ grant	598.9	0.5335
ควบคุมที่ accept	613.26	0.5799

ตารางที่ 5.10 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	14658	0.4012
ควบคุมที่ grant	11504	0.5470
ควบคุมที่ accept	11470	0.7025

ตารางที่ 5.11 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	436.23	0.391
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.12 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	13549	0.0870
ควบคุมที่ grant	11958	0.3838
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.13 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	-	-
ควบคุมที่ accept	10081	0.3505

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.14 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

จากตารางที่ 5.8 ถึง ตารางที่ 5.14 จะแสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอ เมื่อทราฟฟิกเปลี่ยนไปเป็นค่าต่างๆ ซึ่งทำให้เราเห็นสมรรถนะของงานวิจัยชัดขึ้นคือ อัลกอริทึมที่นำเสนอสามารถควบคุมคุณภาพการให้บริการซึ่งในที่นี้จะแสดงออกมาในรูปของ อัตราส่วนเวลาประวิงของ เซลล์คลาส 2 ต่อเซลล์คลาส 1 ได้ โดยการควบคุมค่า p_2 เมื่อพิจารณาเปรียบเทียบกันในแต่ละตารางพบว่า เมื่อสภาพทราฟฟิกเปลี่ยนไป หากเรายังต้องการควบคุมคุณอัตราส่วนการให้บริการไว้เท่าเดิม เราจำเป็นจะต้องปรับค่า p_2 ใหม่เพื่อให้ได้ค่าอัตราส่วนการให้บริการที่คงที่ แต่การปรับค่า p_2 ดังกล่าวจะทำให้ค่าเวลาประวิงที่ได้เปลี่ยนไปด้วยเช่นกัน จากตารางที่ 5.9 และอีกหลายๆตาราง พบว่ามีบางอัตราส่วนที่ไม่สามารถควบคุมได้ในสภาพทราฟฟิกที่ทำการทดสอบ ทั้งนี้แล้วแต่ความสามารถเกี่ยวกับช่วงการทำงานของอัลกอริทึม

เมื่อขนาดความยาวเอนเวโลป = 16 cells

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	5763.9	0.8551
ควบคุมที่ grant	2104.3	0.6215
ควบคุมที่ accept	4091.18	0.9456

ตารางที่ 5.15 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	13214.90	0.4479
ควบคุมที่ accept	27440.28	0.0815

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.16 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	5763.92	0.8551
ควบคุมที่ grant	1321.32	0.4923
ควบคุมที่ accept	2628.61	0.3850

ตารางที่ 5.17 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	13847.48	0.4804
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.18 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	4154.5	0.3829
ควบคุมที่ grant	811.11	0.3713
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.19 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	16584	0.8126
ควบคุมที่ grant	11158.71	0.3934
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.20 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	10535.27	0.3386
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.21 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

จากตารางที่ 5.15 ถึงตารางที่ 5.21 พบว่าการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว จะให้ค่าเฉลี่ยการประวิงเวลาที่สูงขึ้นเมื่อเอนเวโลปมีขนาดความยาวมากขึ้นซึ่งเราไม่สามารถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์หาค่าการแมตช์ได้ทัน เมื่อสภาพทราฟฟิกเปลี่ยนไป หากเรายังต้องการควบคุมคุณอัตราส่วนการให้บริการไว้เท่าเดิม เราจำเป็นจะต้องปรับค่า p_2 ใหม่เพื่อให้ได้ค่าอัตราส่วนการให้บริการที่คงที่ แต่

การปรับค่า p_2 ดังกล่าวจะทำให้ค่าเวลาประวิงที่ได้เปลี่ยนไปด้วยเช่นกัน จากตารางที่ 5.16 และอีกหลายๆตาราง พบว่ามีบางอัตราส่วนที่ไม่สามารถควบคุมได้ในสภาพทราฟฟิกที่ทำการทดสอบ ทั้งนี้แล้วแต่ความสามารถเกี่ยวกับช่วงการทำงานของอัลกอริทึม

วิธีที่ 2 วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)
เมื่อขนาดความยาวเอนเวโลป = 4 cells

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	6041.7	0.8735
ควบคุมที่ grant	5297.4	0.5802
ควบคุมที่ accept	5306.9	0.8467

ตารางที่ 5.22 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	-	-
ควบคุมที่ grant	32256	0.3013
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.23 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	2513.7	0.6451
ควบคุมที่ grant	4836.2	0.4902
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.24 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	12510	0.0357
ควบคุมที่ grant	11249	0.3258
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.25 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	1182.6	0.0564
ควบคุมที่ grant	993.87	0.3701
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.26 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	13680	0.0937
ควบคุมที่ grant	11919	0.3839
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.27 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	12510	0.0357
ควบคุมที่ grant	11249	0.3258
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.28 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

เมื่อขนาดความยาวเอนเวโลป = 16 cells

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	24375	0.0435
ควบคุมที่ grant	24132.21	0.3124
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.29 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 10%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	8520.99	0.1693
ควบคุมที่ grant	8095.15	0.3977
ควบคุมที่ accept	8062.29	0.0374

ตารางที่ 5.30 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 30%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	4382.02	0.0804
ควบคุมที่ grant	4122.22	0.3210
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.31 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 50% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	14340.20	0.0901
ควบคุมที่ grant	12664.98	0.3636
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.32 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 10 เท่า offered load มีค่า 90% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

อัลกอริทึมที่นำเสนอ	เวลาประวิงของเซลล์โดยเฉลี่ย	ค่า p_2
ควบคุมที่ request	23362.29	0.0010
ควบคุมที่ grant	23187.38	0.0537
ควบคุมที่ accept	-	-

หมายเหตุ – หมายถึง ระบบไม่สามารถควบคุมการประวิงเวลาได้ ภายใต้พารามิเตอร์ที่กำหนด

ตารางที่ 5.33 แสดงเวลาประวิงของเซลล์โดยเฉลี่ยของอัลกอริทึมที่นำเสนอเมื่ออัตราส่วนเวลาประวิงของ เซลล์คลาส 2/เซลล์คลาส 1 มีค่า 100 เท่า offered load มีค่า 80% และ สัดส่วนเซลล์คลาส 1 มีค่า 50%

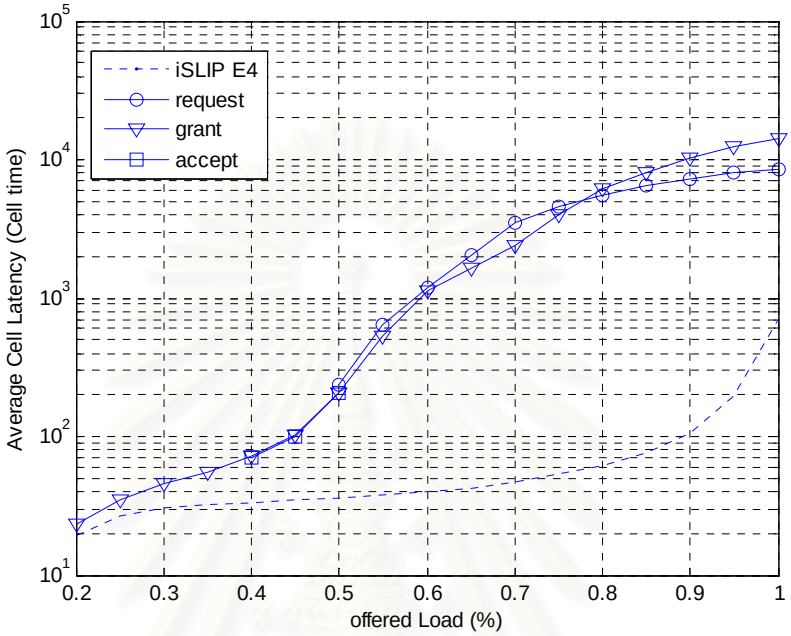
จากตารางที่ 5.22 ถึงตารางที่ 5.33 พบว่าเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมทีมีค่าสูงตลอดทุกช่วงของ offered load ทำให้คุณภาพการให้บริการของสวิตช์ต่ำมาก ยิ่งขนาดเอนเวโลปมีความยาวมากขึ้นก็จะมีค่าเฉลี่ยไทม์ดีเลย์สูงขึ้นมากเป็นอัตราส่วนสัมพันธ์กัน จึงไม่เหมาะสมที่จะนำวิธีนี้มาใช้กับสวิตช์ซึ่งจะต้องรองรับการเพิ่มขนาดของสวิตช์ในอนาคต แต่หากเรายังต้องการควบคุมคุณอัตราส่วนการให้บริการ เราก็สามารถทำการปรับค่า p_2 ใหม่ได้ เพื่อให้ได้ค่าอัตราส่วนการให้บริการตามต้องการได้ แต่การปรับค่า p_2 ดังกล่าวจะทำให้ค่าเวลาประวิงที่ได้เปลี่ยนไปด้วยเช่นกัน พบว่ามีบางตาราง มีอัตราส่วนที่ไม่สามารถควบคุมได้ในสภาพทราฟฟิกที่ทำการทดสอบ ทั้งนี้แล้วแต่ความสามารถเกี่ยวกับช่วงการทำงานของอัลกอริทึม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

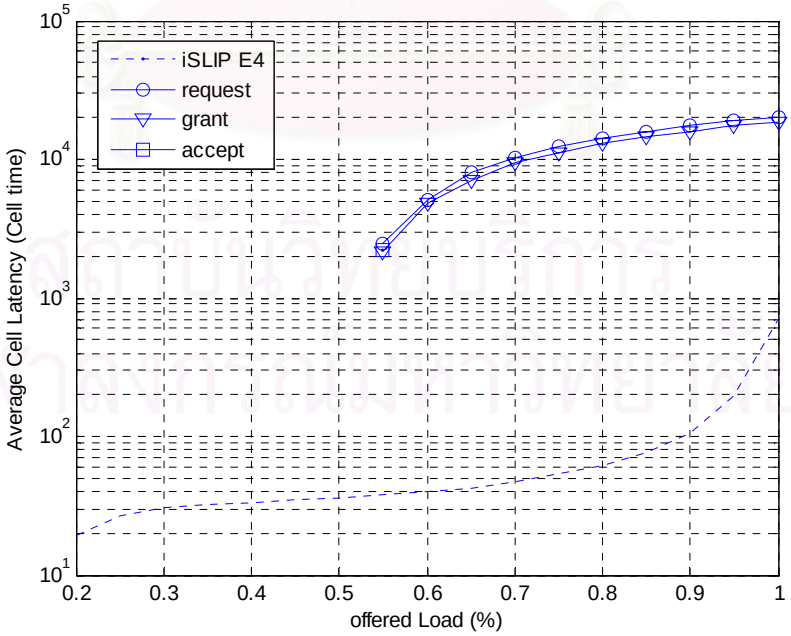
5.7.2 สมรรถนะด้านค่าเวลาประวิง

วิธีที่ 1 วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope)

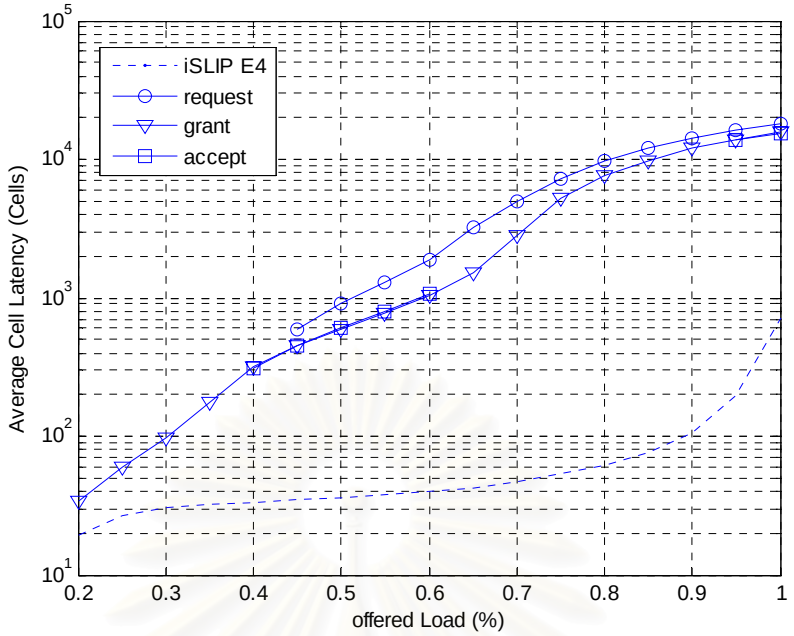
เมื่อขนาดความยาวเอนเวโลป = 4 cells



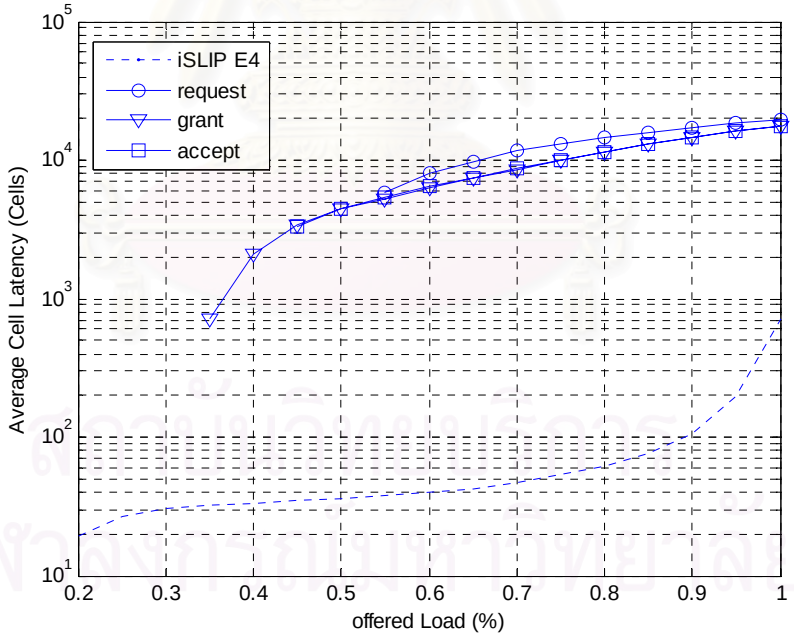
รูปที่ 5.61 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells



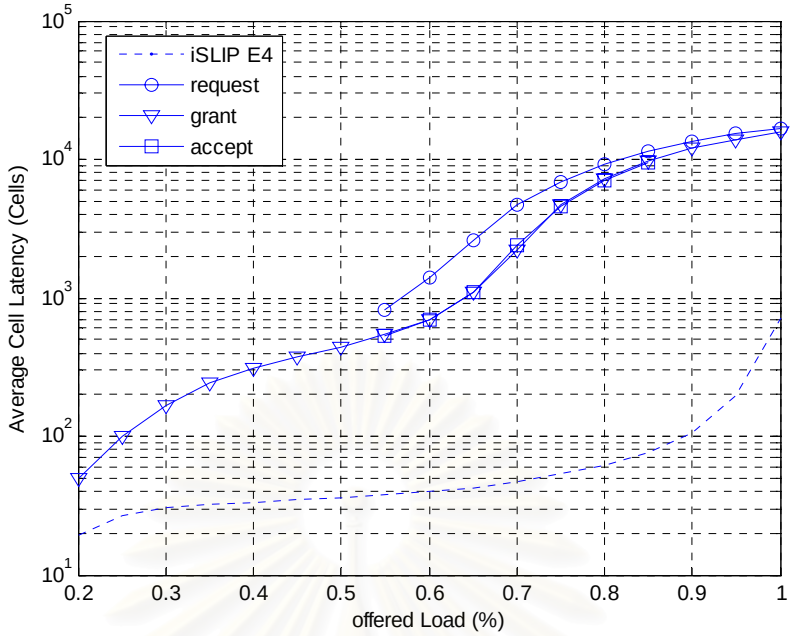
รูปที่ 5.62 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells



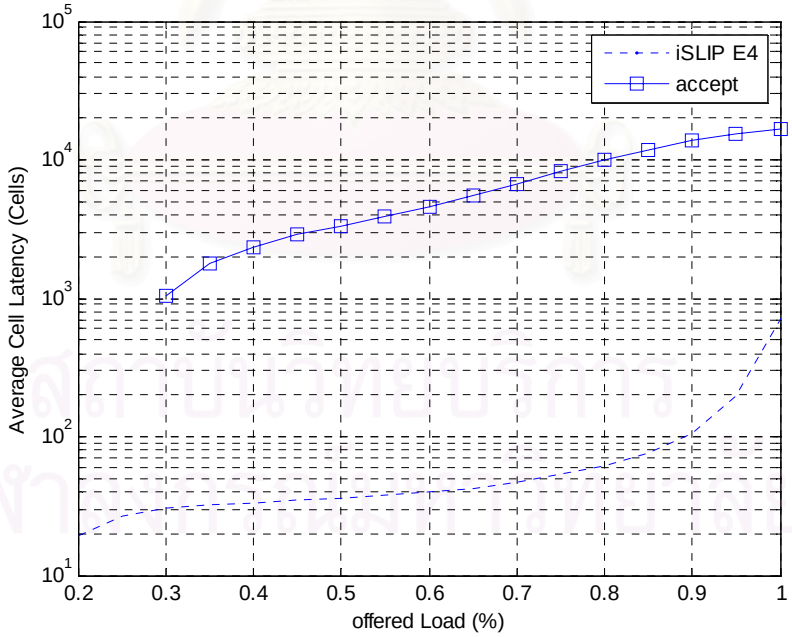
รูปที่ 5.63 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells



รูปที่ 5.64 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells



รูปที่ 5.65 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells

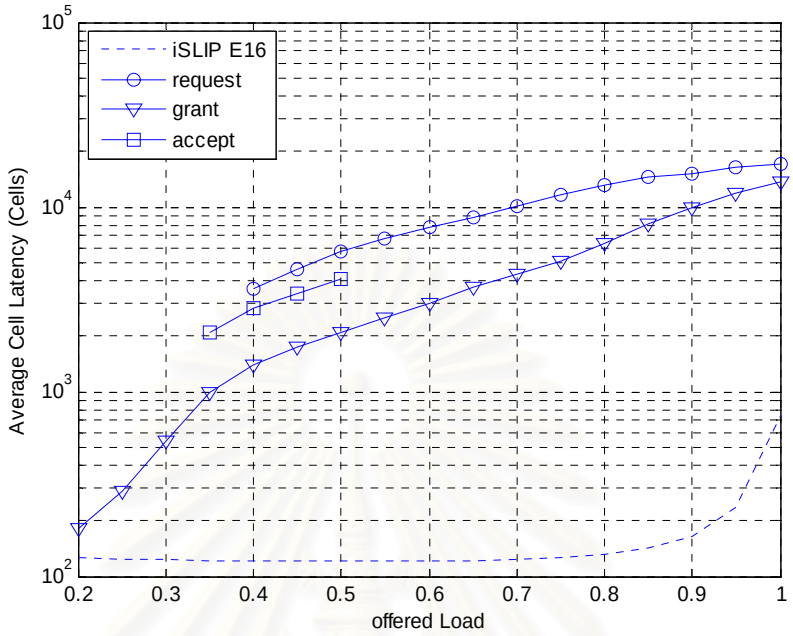


รูปที่ 5.66 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells

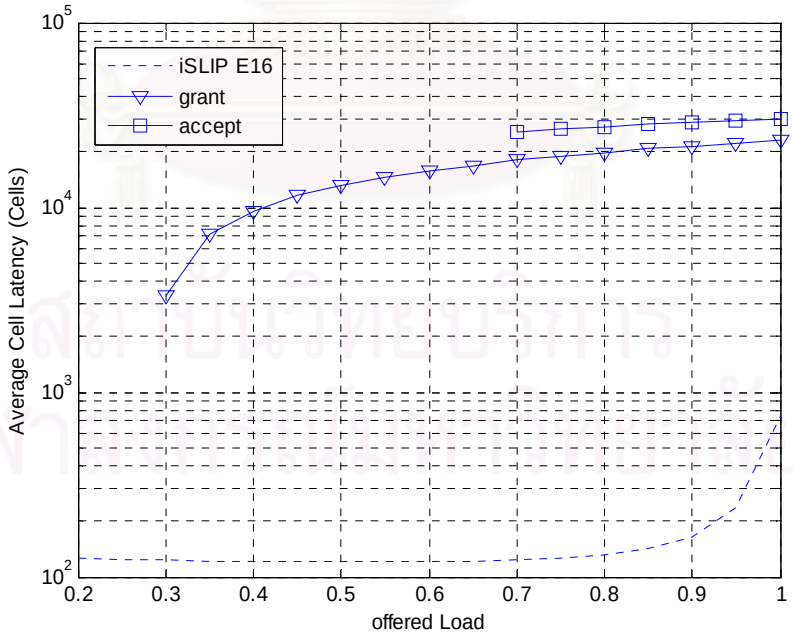
รูปที่ 5.61 ถึง 5.66 แสดงเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึม เมื่อกำหนดอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1เท่ากันแล้วเพิ่ม offered load ขึ้นจาก 20% ไปจนถึง 100% ค่าเวลาประวิงโดยเฉลี่ยในแต่ละอัลกอริทึมจะเพิ่มขึ้นตามไปด้วยดังเหตุผลที่ได้กล่าวไว้แล้วในตอนต้น ในแต่ละอัลกอริทึมสามารถทำงานในช่วง offered load และค่าสัดส่วนปริมาณเซลล์คลาส 1 ที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับค่าเวลาประวิงที่ต่างกันของเซลล์ทั้งสองคลาสที่แต่ละอัลกอริทึมสามารถควบคุมได้ เมื่อสภาพทราฟฟิกเปลี่ยนไป ซึ่งก็เหมือนกับตอนที่ยังไม่มีเทคนิคเอนเวโลป อัลกอริทึมการควบคุมที่ grant และอัลกอริทึมการควบคุมที่ accept จะมีช่วงการใช้งานที่กว้างที่สุดในทุกๆสภาพทราฟฟิก เมื่ออัตราส่วน P2 / P1 เพิ่มขึ้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มขึ้นด้วย เมื่อเพิ่มสัดส่วนเซลล์คลาส 1 จาก 10%-50% จากเหตุผลที่กล่าวแล้วในตอนต้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจึงเพิ่มสูงขึ้นและที่สัดส่วนปริมาณเซลล์คลาส 1 เป็น 10 % เมื่ออัตราส่วน P2 / P1 เปลี่ยนจาก 10 เท่า เป็น 100 เท่า ช่วงการใช้งานของอัลกอริทึมการควบคุมที่ grant และอัลกอริทึมการควบคุมที่ accept จะแคบลง ถึงแม้ว่าเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะมีค่าสูง แต่เราก็สามารถที่จะที่ควบคุมคุณภาพการให้บริการได้อย่างแม่นยำ (ในเทอม P2 / P1) พบข้อสังเกตว่าอัลกอริทึมควบคุมที่ request grant และ acceptโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวก่อนจะมีช่วงการใช้งานที่กว้างกว่าอัลกอริทึมควบคุมที่ request grant และ accept ที่ยังไม่มีเทคนิคเอนเวโลป ซึ่งเป็นข้อดี แต่ข้อเสียคือ เวลาประวิงโดยเฉลี่ยของอัลกอริทึมควบคุมที่ requestอัลกอริทึมควบคุมที่ grant และอัลกอริทึมควบคุมที่ accept ก็สูงขึ้นกว่าตอนที่ไม่มีเทคนิคเอนเวโลปและจะเพิ่มมากขึ้นเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้น ดังที่จะแสดงในผลการทดลองต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

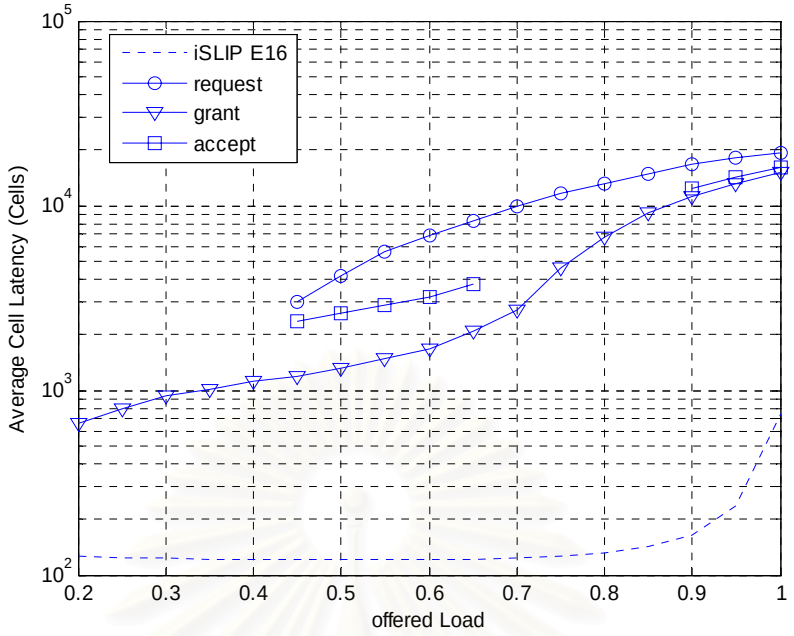
วิธีที่ 1 วิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว (partially filled (PF) envelope) เมื่อขนาดความยาวเอนเวโลป = 16 cells



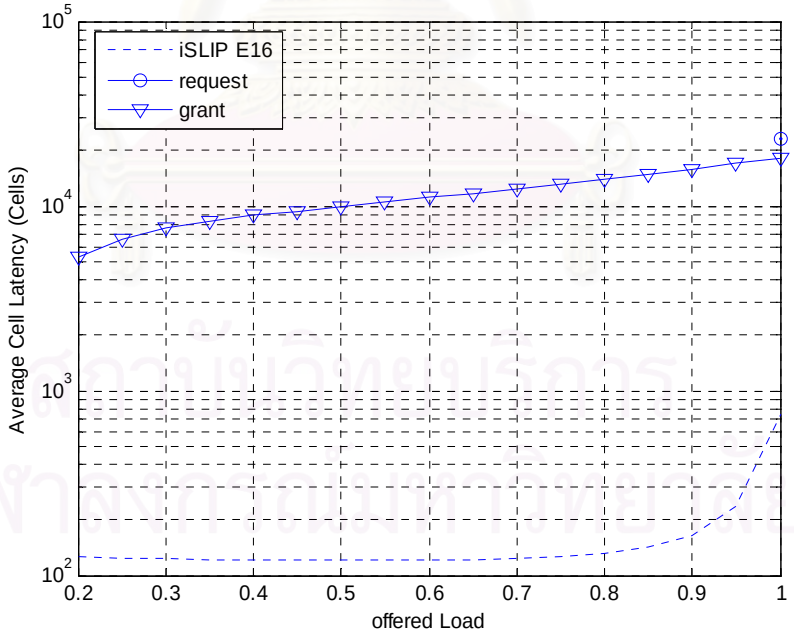
รูปที่ 5.67 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells



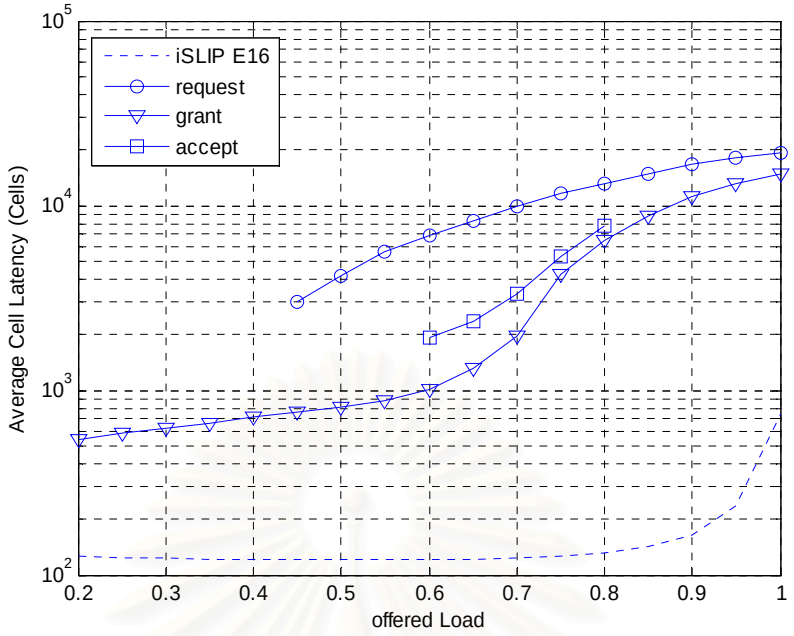
รูปที่ 5.68 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells



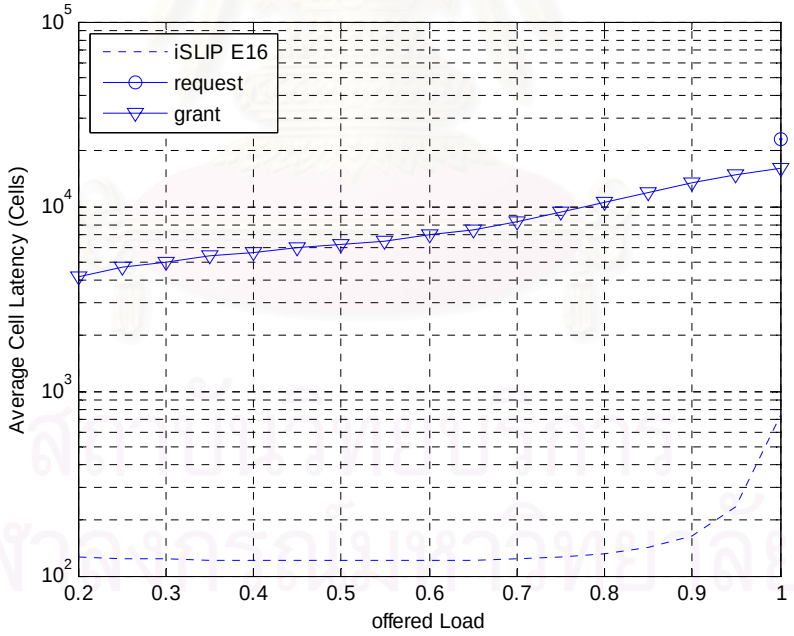
รูปที่ 5.69 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells



รูปที่ 5.70 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells



รูปที่ 5.71 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells

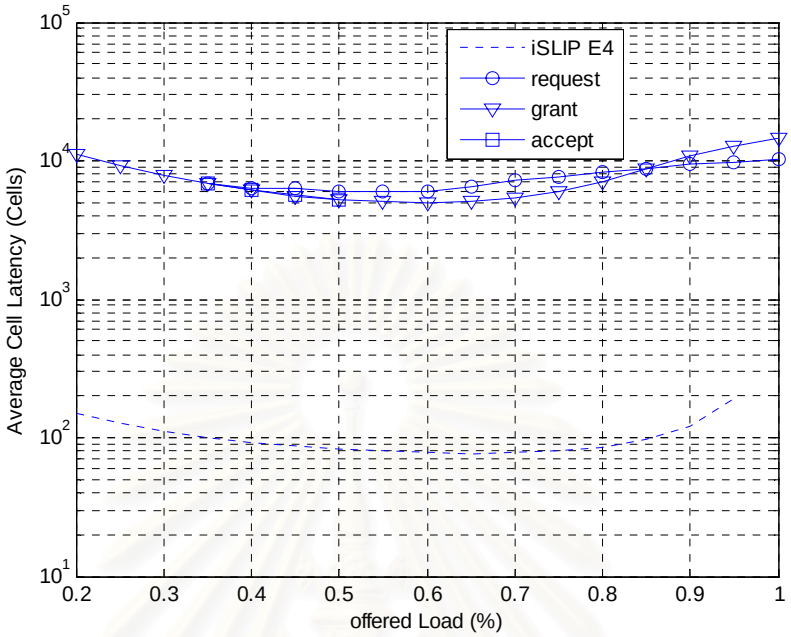


รูปที่ 5.72 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells

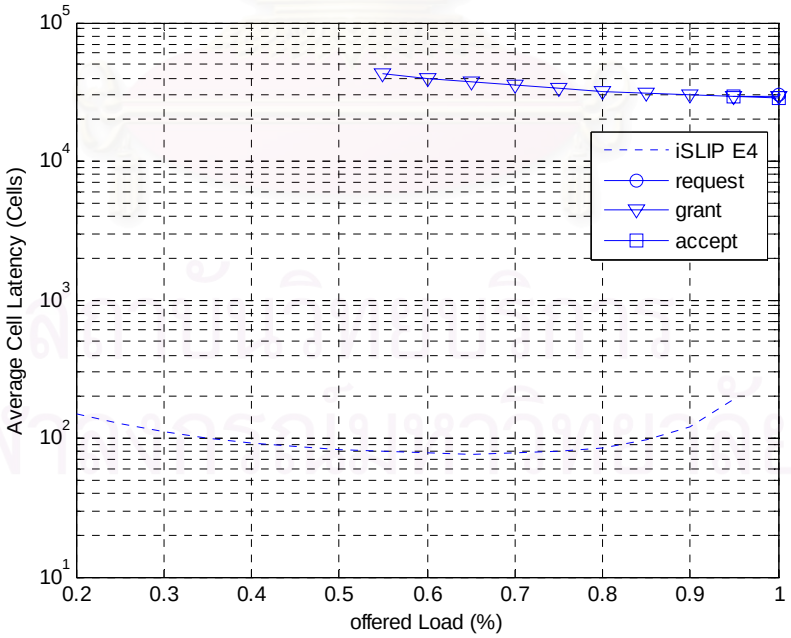
รูปที่ 5.67 ถึง 5.72 แสดงเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึม โดยมีขนาดความยาวเอนเวโลป 16 เซลล์ สัดส่วนเซลล์คลาส 1 มี 3 ค่าคือ 10% 30% และ 50% และในแต่ละสัดส่วนเซลล์คลาส 1 ที่เท่ากัน เราจะเพิ่มอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จาก 10 เท่า เป็น 100 เท่า เราพบว่าเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มขึ้นซึ่งเป็นไปตามที่คาดไว้ ตัวอย่างเช่นในรูปที่ 5.61 และ 5.67 เวลาประวิงโดยเฉลี่ยของอัลกอริทึมการควบคุมที่ request ที่ offered load =0.5 จะเพิ่มขึ้นจากค่าประมาณ 250 เซลล์ เป็น 6000 เซลล์ อัลกอริทึมการควบคุมที่ grant จะเพิ่มจากประมาณ 200 เซลล์เป็น 2000 เซลล์ และอัลกอริทึมการควบคุมที่ accept จะเพิ่มจากประมาณ 200 เซลล์เป็น 4000 เซลล์ ส่วนช่วงการใช้งานเมื่อเพิ่มความยาวเอนเวโลปจะเท่าเดิม แต่จะสามารถทำนายได้ว่าช่วงการใช้งานของทุกอัลกอริทึมจะกว้างขึ้นกว่าอัลกอริทึมเดิมที่ไม่มีเทคนิคเอนเวโลป เมื่ออัตราส่วน P2 / P1 เพิ่มขึ้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มขึ้นด้วย ส่วนช่วงการใช้งานจะแคบลง โดยอัลกอริทึมการควบคุมที่ request แคบลงเร็วมาก (sensitivity กับค่าอัตราส่วน P2 / P1 ที่เพิ่มขึ้น) เมื่อเพิ่มสัดส่วนเซลล์คลาส 1 จาก 10%-50% จากเหตุผลที่กล่าวแล้วในตอนต้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจึงเพิ่มสูงขึ้น ถึงแม้ว่าเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะมีค่าสูง แต่เราก็สามารถที่จะที่ควบคุมคุณภาพการให้บริการได้อย่างแม่นยำ (ในเทอม P2 / P1) แต่จะสามารถรับประกันคุณภาพการให้บริการได้หรือไม่นั้นขึ้นอยู่กับระดับของคุณภาพการให้บริการของโครงข่าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

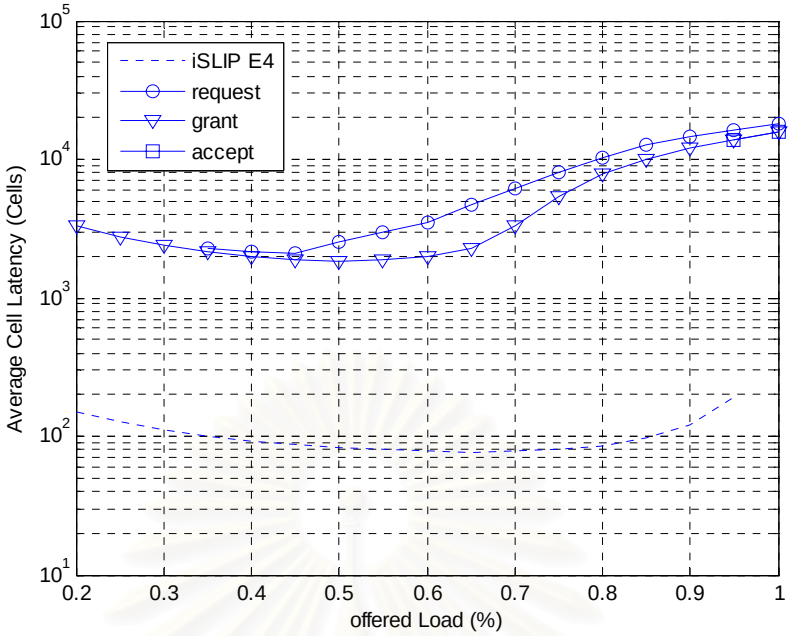
วิธีที่ 2 วิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope)
เมื่อขนาดความยาวเอนเวโลป = 4 cells



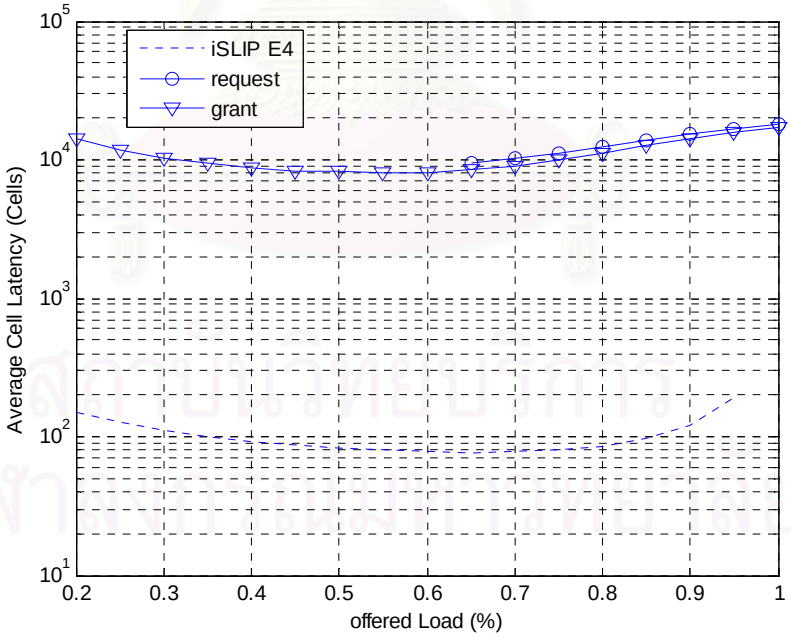
รูปที่ 5.73 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells



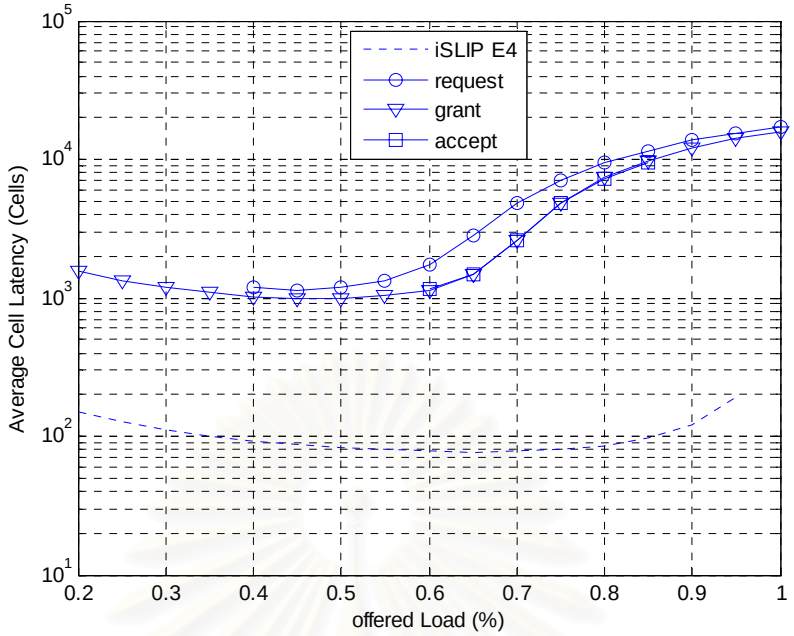
รูปที่ 5.74 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 4 cells



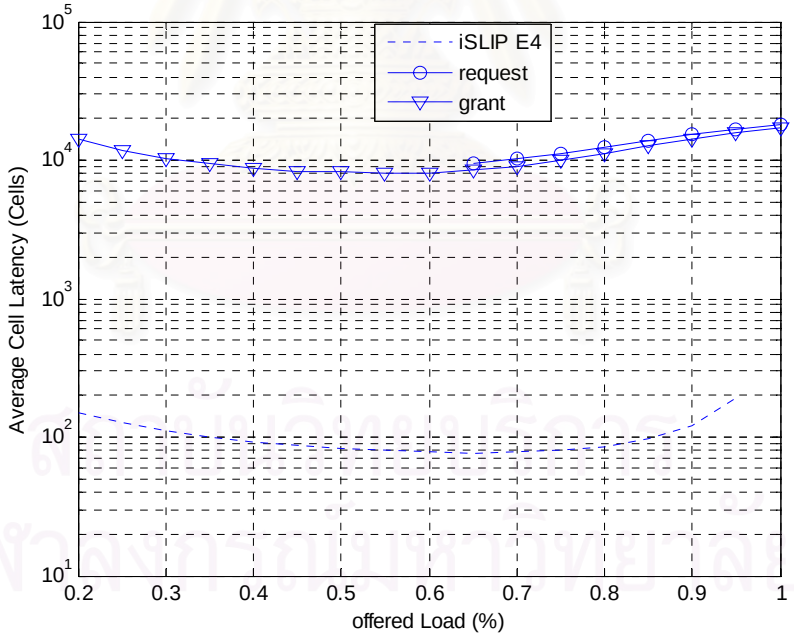
รูปที่ 5.75 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells



รูปที่ 5.76 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 4 cells



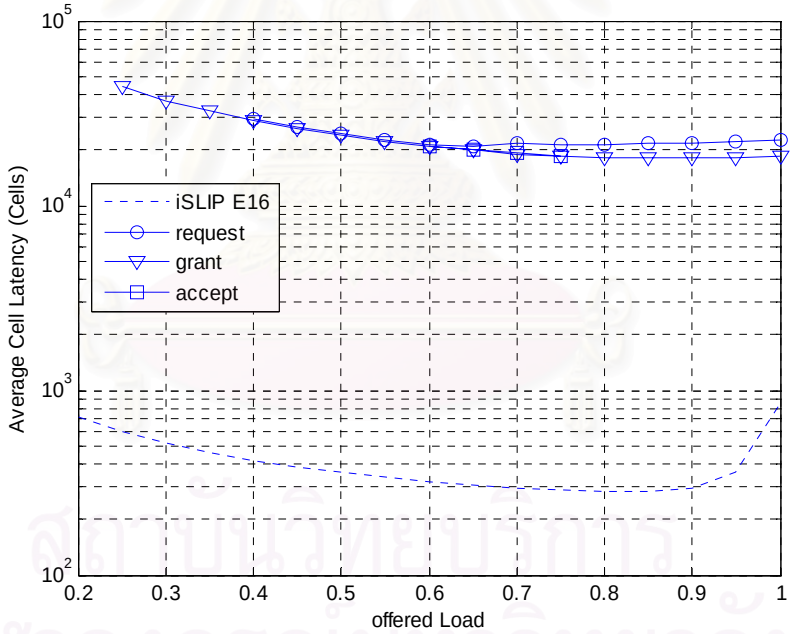
รูปที่ 5.77 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells



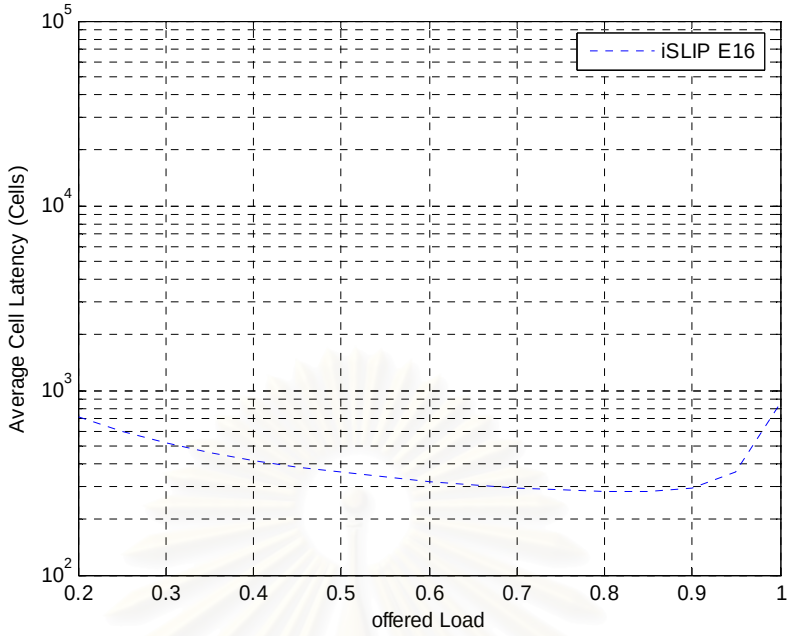
รูปที่ 5.78 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 4 cells

รูปที่ 5.73 ถึง 5.78 แสดงเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม (Filled (F) envelope) เมื่ออัตราส่วน P2 / P1 เพิ่มขึ้น เวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มขึ้น ที่สัดส่วนปริมาณเซลล์คลาส 1 เป็น 30% และ 50% อัลกอริทึมควบคุมที่ grant จะมีช่วงการใช้งานในทุกๆค่าของ offered load เมื่ออัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 เพิ่มจาก 10 เท่าเป็น 100 เท่า ซึ่งแตกต่างจากวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวก่อน แต่ด้วยเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมทีมีค่าสูงตลอดทุกช่วงของ offered load ทำให้คุณภาพการให้บริการของสวิตซ์ต่ำมาก และโครงข่ายอาจไม่สามารถรับประกันในคุณภาพการให้บริการได้ จึงไม่เหมาะสมที่จะนำวิธีนี้มาใช้กับสวิตซ์ในอนาคต และเวลาประวิงโดยเฉลี่ยจะเพิ่มมากขึ้นเมื่อขนาดความยาวเอนเวโลปเพิ่มขึ้น ดังที่จะแสดงในผลการทดลองต่อไป

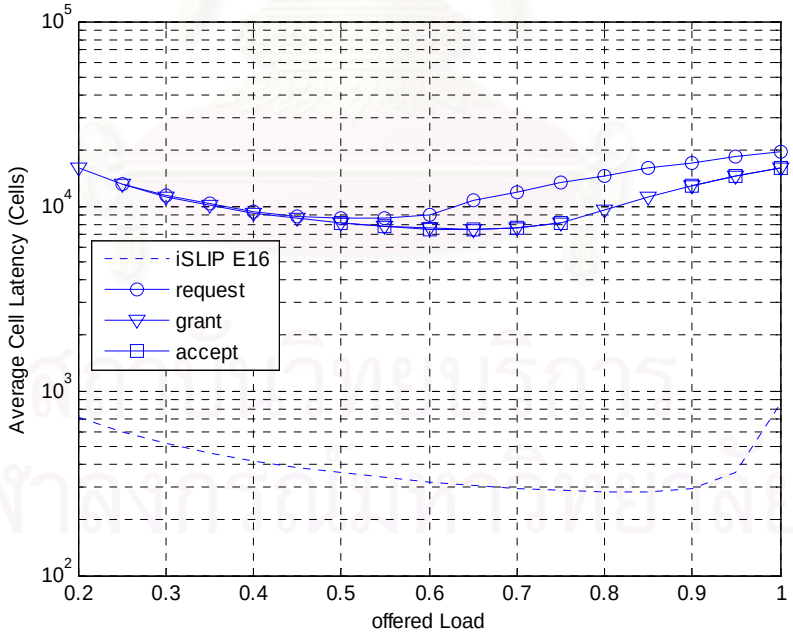
เมื่อขนาดความยาวเอนเวโลป = 16 cells



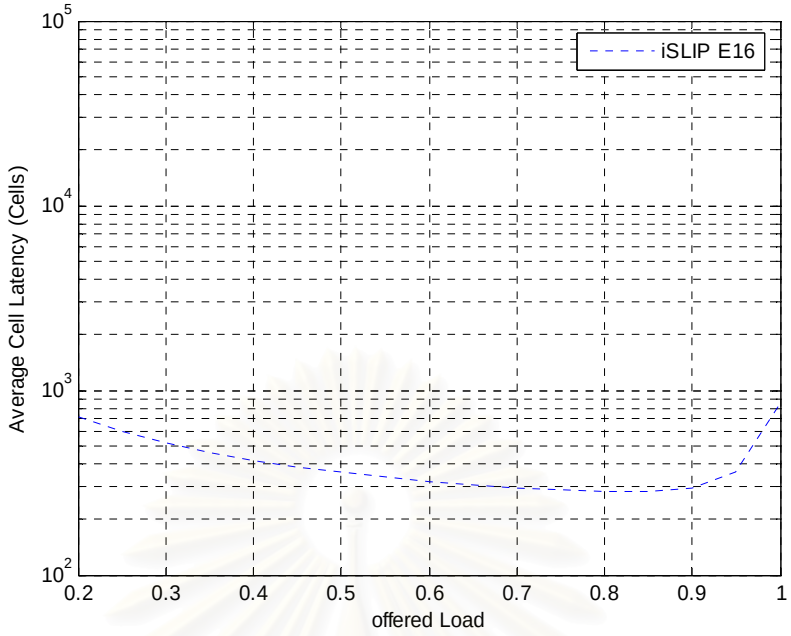
รูปที่ 5.79 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells



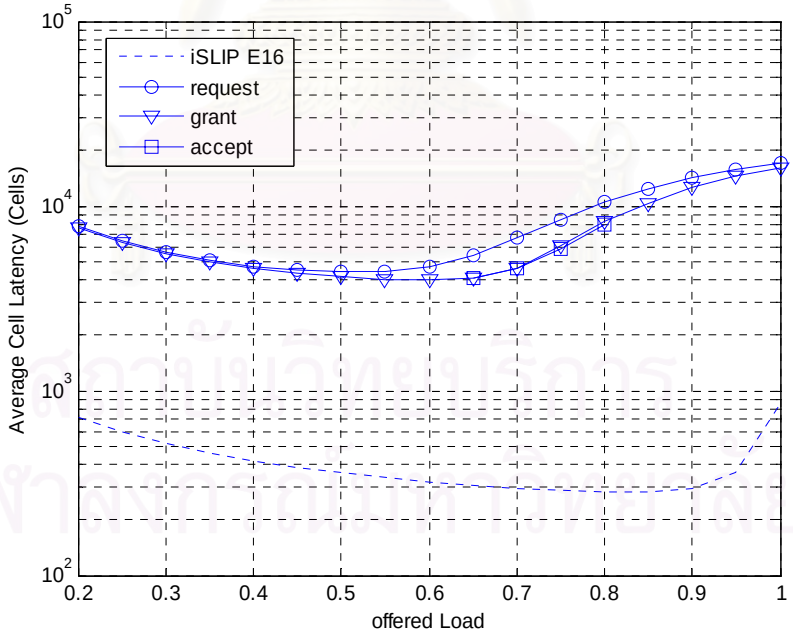
รูปที่ 5.80 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells



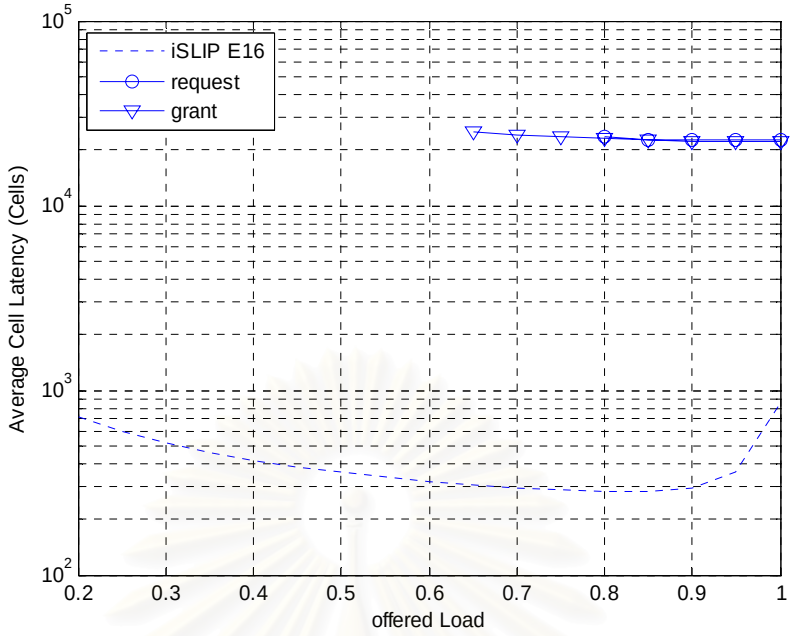
รูปที่ 5.81 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells



รูปที่ 5.82 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 30% envelope size = 16 cells



รูปที่ 5.83 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 10 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells



รูปที่ 5.84 แสดงช่วงการใช้งานของแต่ละอัลกอริทึม ที่ให้อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 50% envelope size = 16 cells

รูปที่ 5.79 ถึง 5.84 แสดงเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึม โดยมีขนาดความยาวเอนเวโลป 16 เซลล์ สัดส่วนเซลล์คลาส 1 มี 3 ค่าคือ 10% 30% และ 50% และในแต่ละสัดส่วนเซลล์คลาส 1 ที่เท่ากัน เราจะเพิ่มอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 จาก 10 เท่า เป็น 100 เท่า เราพบว่าเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมจะเพิ่มมากขึ้น ซึ่งก็เป็นไปตามที่คาดไว้ ส่วนช่วงการใช้งานก็จะลดต่ำลงเล็กน้อย แต่ที่อัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ที่ 100 เท่า ที่ค่า สัดส่วนเซลล์คลาส 1 10% envelope size = 16 cells ดังรูปที่ 5.80 ทุกอัลกอริทึมไม่สามารถทำงานได้เลย แต่ด้วยเวลาประวิงโดยเฉลี่ยของทุกอัลกอริทึมที่มีค่าสูงตลอดทุกช่วงของ offered load ทำให้คุณภาพการให้บริการของสวิตซ์ต่ำมาก และโครงข่ายอาจไม่สามารถรับประกันในคุณภาพการให้บริการได้ เช่นเดียวกับเมื่อขนาดความยาวเอนเวโลป 4 เซลล์ ดังนั้นจึงไม่เหมาะสมที่จะนำวิธีนี้มาใช้กับสวิตซ์ในปัจจุบัน

บทที่ 6
สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้เสนออัลกอริทึมการจัดกำหนดการสำหรับการจัดสรรบริการในเซลล์สวิตช์ที่พิจารณาลำดับความสำคัญของเซลล์โดยที่มีเทคนิคเอนเวโลป โดยอัลกอริทึมที่นำเสนอจะทำงานบนพื้นฐานของอัลกอริทึม iSLIP และสามารถรองรับทราฟฟิกที่พิจารณาลำดับความสำคัญของเซลล์ได้ อีกทั้งยังสามารถควบคุมคุณภาพในการให้บริการซึ่งแสดงออกมาในรูปของอัตราส่วนเวลาประวิงโดยเฉลี่ยของเซลล์คลาส 2 ต่อเซลล์คลาส 1 ได้อย่างแม่นยำ

เทคนิคเอนเวโลปนำมาใช้ในการแก้ปัญหาคอขวดในการสวิตช์ข้อมูล โดยการลดเวลาในการประมวลผลเพื่อหาการแมตช์ระหว่างอินพุตและเอาต์พุตคู่ใด ๆ ของสวิตช์ ทำให้ส่งเซลล์ผ่านโครงสร้างการสวิตช์ได้เร็วขึ้น หรือกล่าวอีกนัยหนึ่งก็คือ การเพิ่มแบนด์วิดท์ในการสวิตช์นั่นเอง ทำให้สวิตช์สามารถนำไปใช้กับแพ็กเกตที่มีความเร็วสูงขึ้นได้ ข้อเสียของเทคนิคเอนเวโลป คือจากการจัดเตรียมเอนเวโลปก่อนส่งผ่านโครงสร้างสวิตช์ ทำให้ค่าเฉลี่ยการประวิงเวลาของเซลล์มีค่าเพิ่มขึ้นตามขนาดเอนเวโลปที่เพิ่มขึ้น เราจึงไม่สามารถเพิ่มขนาดเอนเวโลปเกินขอบเขตที่กำหนดไว้สำหรับรับประกันคุณภาพการให้บริการได้ ถึงแม้ว่าจะทำให้แบนด์วิดท์ในการสวิตช์สูงขึ้นและเป็นการแก้ปัญหาคอขวดในการสวิตช์โดยตรง จะเห็นว่าทราฟฟิกประเภทเรียลไทม์มีข้อกำหนดในเรื่องการประวิงเวลาของเซลล์ที่เข้มงวดอย่างยิ่ง ดังนั้น จึงเป็นข้อสำคัญที่จะต้องนำมาพิจารณา และเป็นข้อจำกัดของเทคนิคเอนเวโลปด้วย

โดยอัลกอริทึมที่ใช้ในวิทยานิพนธ์ฉบับนี้คือ อัลกอริทึมการควบคุมที่ request ที่มีเทคนิคเอนเวโลป, อัลกอริทึมการควบคุมที่ grant ที่มีเทคนิคเอนเวโลป และอัลกอริทึมการควบคุมที่ accept ที่มีเทคนิคเอนเวโลป ซึ่งการทดสอบสมรรถนะในบทที่ผ่านมา ทำให้ทราบว่าการเลือกเอนเวโลปวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวก่อนจะเหมาะสมกับโครงข่ายในปัจจุบันมากกว่าวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม ขนาดความยาวเอนเวโลปที่ใช้นั้นจะขึ้นอยู่กับเวลาในการแมตช์ของอัลกอริทึมการจัดลำดับเซลล์ ส่วนอัลกอริทึมใหม่ที่นำเสนอก็สามารถทำงานได้ดีกว่าอัลกอริทึม Prioritized iSLIP คือสามารถควบคุมคุณภาพในการให้บริการได้อย่างแม่นยำในทุกๆค่าของสภาพทราฟฟิก การปรับค่า p_2_allow จำเป็นต้องปรับอยู่ตลอดเวลาเพื่อให้เหมาะสมกับสภาพทราฟฟิกที่เปลี่ยนแปลงอยู่ตลอดเวลา

6.2 ข้อเสนอแนะ

ในงานวิจัยที่ผ่านมา มีการนำเสนอวิธีเลือกเอนเวโลปโดยวิธีผสมผสาน (Combined Filled and Partially-Filled (CFPF) envelope) และเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต (Dynamic threshold envelope) ซึ่งการเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัตทำให้ค่าเฉลี่ยการประวิงเวลาดีกว่าการเลือกเอนเวโลปวิธีอื่น ๆ ที่นำเสนอมาก่อนหน้านั้น แต่ก็ยังไม่ได้นำมาใช้กับอัลกอริทึมการจัดลำดับเซลล์แบบที่เสนอในวิทยานิพนธ์ฉบับนี้ ดังนั้นแนวทางการวิจัยต่อไปจึงควรพัฒนาวิธีเลือกเอนเวโลปวิธีต่างๆและเพิ่มระดับความสำคัญของเซลล์ให้มากกว่า 2 ระดับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Fabio M. Chiussi and Andrea Francini, “Scalable Electronic Packet Switches”, *IEEE Journal on Selected Areas in Communications*, Vol. 21 No. 4 May 2003, pp.486-500.

2. ลัญฉกร วุฒิสิทธิกุลกิจ. *วิศวกรรมโทรคมนาคม*. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2544 : หน้า 89-115

3. Tarek N. Saadawi, Mostafa H. Ammar and Ahmed El Hakeem, “ *Fundamentals of Telecommuni-cation Networks*”, John Wiley & Sons, 1994, pp.399-447.

4. Mark J. Karol, Michael G. Hluchyj and Samuel P. Morgan, “Input versus Output Queueing on a Space-Devision Packet Switch”, *IEEE Transaction on Communication*, Vol Com-35 No. 12 December 1987, pp.1347-1356.

5. Hakyong Kim and Kiseon Kim, “Performance Analysis of the Multiple Input-Queued Packet Switch With the Restricted Rule”, *IEEE/ACM Transactions on Networking*, Vol. 11 No. 3 June 2003, pp.478-487.

6. Hiroyuki Ohsaki and Naoki Wakamiya, “Performance of an Input/Output Buffered-Type ATM LAN Switch with Back-Pressure Function”, *IEEE/ACM Transactions on Networking*, Vol. 5 No. 2 April 1997, pp.278-290.

7. Sundar Iyer and Nick W. Mckeown, “Analysis of the Parallel Packet Switch Architecture”, *IEEE/ACM Transactions on Networking*, Vol. 11 No. 2 April 2003, pp.314-324.

8. Marco A. Marsan, “Packet-Mode Scheduling in Input-Queued Cell-Based Switches”, *IEEE/ACM Transactions on Networking*, Vol. 10 No. 5 October 2002, pp.666-678.

9. Nick McKeown , “The iSLIP Scheduling Algorithm for Input-Queued Switches”, *IEEE/ACM Transactions on Networking*, Vol. 7 No. 2 April 1999, pp.188-201.

10. Nick McKeown, “Scheduling Algorithm for Input-Queued Switches”, Ph.D.dissertation, Univ. California. Berkeley, 1995.

11. Hyoung-Il Lee and Seung-Woo Seo, " A Practical Approach for Statistical Matching of Output Queueing", *IEEE Journal on Selected Areas in Communication*, Vol. 21, No. 4, May 2003, pp.616-629

12. Mounir Hamdi, "Guest Editorial High-Performance Electronic Switches/Routers for High-Speed Internet", *IEEE Journal on Selected Areas in Communications*, Vol. 21 No. 4 May 2003,

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ผนวก ก

ประมาณค่าการแจกแจงแบบปัวส์ซงจากการแจกแจงแบบทวินาม เมื่อ $N \to \infty$

การแจกแจงแบบทวินาม (Binomial Distribution)

นิยาม พิจารณาการทดลองที่มีผลการทดลองที่เป็นไปได้ 2 อย่าง คือ A กับ A^C โดยมีความน่าจะเป็น $P(A)=p$ และ $P(A^c)=1-p=q$ นอกจากนี้ สมมุติว่า $P(A)=p$ มีค่าคงตัวสำหรับการทดลองทุกครั้ง ให้ X เป็นจำนวนครั้งที่เหตุการณ์ A เกิดขึ้นจากการทดลองรวมทั้งหมด n ครั้ง ในกรณีนี้ เราจะเรียกตัวแปรสุ่ม X ว่ามีการแจกแจงแบบทวินาม ที่มีพารามิเตอร์ n และ p

ทฤษฎีบท ถ้า X เป็นตัวแปรสุ่มชนิดทวินาม ที่มีพารามิเตอร์ n และ p แล้ว

$$P[X=k]=\binom{n}{k}p^k(1-p)^{n-k},k=0,1,...,n \qquad (ก\ 1)$$

$$E[X]=np \qquad (ก\ 2)$$

$$Var(X)=npq \qquad (ก\ 3)$$

$$M(jv)=\left(q+pe^{jv}\right)^n \qquad (ก\ 4)$$

การแจกแจงแบบปัวส์ซง (Poisson Distribution)

นิยาม ให้ X เป็นตัวแปรสุ่มชนิดดิสครีตที่เป็นจำนวนเต็มค่าไม่เป็นลบ และถ้า

$$P[X=k]=\left(\frac{a^k}{k!}\right)e^{-a},k=0,1,2,... \qquad (ก\ 5)$$

เราจะเรียกตัวแปรสุ่ม X ว่ามีการแจกแจงแบบปัวส์ซง ที่มีพารามิเตอร์ $a>0$

ทฤษฎีบท ถ้า X เป็นตัวแปรสุ่มที่มีการแจกแจงแบบปัวส์ซง ที่มีพารามิเตอร์ a แล้ว

$$E[X]=a \qquad (ก\ 6)$$

$$Var(X)=a \qquad (ก\ 7)$$

$$M(jv)=\exp\left\{a\left(e^{jv}-1\right)\right\} \qquad (ก\ 8)$$

ทฤษฎีบท *การแจกแจงแบบปัวส์ซงกับการแจกแจงแบบทวินาม* [17]

ถ้า X เป็นตัวแปรสุ่มที่มีการแจกแจงแบบทวินาม ที่มีพารามิเตอร์ n และ p และถ้า $n \to \infty$ และ $p \to 0$ แต่ $np \to a$ แล้ว X จะมีการแจกแจงแบบปัวส์ซง ที่มีพารามิเตอร์ a

<u>พิสูจน์</u> $P[X=k]=\binom{n}{k}p^k(1-p)^{n-k}=\frac{n(n-1)...(n-k-1)}{k!}p^k(1-p)^{n-k}$ (ก 9)

ให้ $np=a$ ดังนั้น $p=a/n$ และ $1-p=(n-a)/n$ ซึ่งเมื่อแทนลงในเทอมสุดท้ายของสมการข้างบนนี้ จะให้

$$P[X=k]=\frac{n(n-1)...(n-k-1)}{k!}p^k(1-p)^{n-k}$$

$$=\frac{a^k}{k!}\left[(1)\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)...\left(1-\frac{k-1}{n}\right)\right]\left(1-\frac{a}{n}\right)^{-k}\left(1-\frac{a}{n}\right)^{n} \quad (10)$$

เมื่อให้ $n \to \infty$ เราพบว่า $\left(1-a/n\right)^n \to e^{-a}$ ส่วนพจน์อื่น ๆ มีลิมิตเข้าหา 1 ทั้งหมด ดังนั้น

$$\lim_{n\to\infty} P[X=k]=\left(\frac{a^k}{k!}\right)e^{-a}$$ ซ.ต.พ.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ผนวก ข

ตัวอย่างซอร์สโคดของอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์

อัลกอริทึม iSLIP

```
Clc; clear all; close all

for offered_load = 0.1:0.05:1.0
num_inport = 16;
num_outport = 16;
buffer_size = 50000;
YES = 1;
NO = -1;
EMPTY = 0;
OCCUPIED = 1;
ITERATIVE_MAX = 4;
%offered_load = 0.4;
SIM_TIME = 50000;

viqueue = zeros(num_inport,num_outport,buffer_size);
viqueue(:,:,:) = EMPTY;
viqueue_len = zeros(num_inport,num_outport);

% islip
request = zeros(num_inport,num_outport);
grant = zeros(num_inport,num_outport);
accept = zeros(num_inport,num_outport);
grant_arbiter = ones(num_outport,1);
accept_arbiter = ones(num_inport,1);


total_packet_loss = 0;
total_packet = 0;
total_delay = 0;

for time_slot = 1:SIM_TIME
%    fprintf('Time slot======= %d',time_slot)
    newpacket = (rand(1,num_inport)<offered_load);
    num_packet = sum(newpacket);
    total_packet = total_packet + num_packet;

    outport = randint(1,num_inport,[1 num_outport]).*newpacket;

    for k = 1:num_inport
        if(outport(k)~=0)
            curr_len = viqueue_len(k,outport(k));
            if(curr_len < buffer_size)
%                viqueue(k,outport(k),curr_len+1) = OCCUPIED;
                viqueue_len(k,outport(k)) = curr_len + 1;
            else
%                disp('#####Packet loss');
                total_packet_loss = total_packet_loss + 1;
            end
        end
    end
    viqueue_len;

    % islip
```

```
request(:,:) = (viqueue_len(:,:)>0);

% grant
grant = zeros(num_inport,num_outport);
actual_grant_inport = zeros(num_outport,1);
for k = 1:num_outport
    current_inport = grant_arbiter(k);
    for j = 1:num_inport
        if(request(current_inport,k)==1)
            grant(current_inport,k) = 1;
            actual_grant_inport(k) = current_inport;
            break
        else
            current_inport = current_inport+1;
            if (current_inport > num_inport)
                current_inport = 1;
            end
        end
    end
end
grant;
actual_grant_inport;

% accept
accept = zeros(num_inport,num_outport);
for j = 1:num_inport
    current_outport = accept_arbiter(j);
    for k = 1:num_outport
        if (grant(j,current_outport)==1)
            accept(j,current_outport) = 1;
            accept_arbiter(j) = current_outport+1;
            if(accept_arbiter(j)>num_outport)
                accept_arbiter(j) = 1;
            end
            break
        else
            current_outport = current_outport +1;
            if(current_outport > num_outport)
                current_outport = 1;
            end
        end
    end
end
accept;
accept_arbiter;

% update grant arbiter
for k = 1:num_outport
    packTx = find(accept(:,k));
    if(~isempty(packTx))
        if(actual_grant_inport(k)==grant_arbiter(k))
            grant_arbiter(k) = grant_arbiter(k)+1;
            if(grant_arbiter(k)>num_inport)
                grant_arbiter(k) = 1;
            end
        else
            grant_arbiter(k) = actual_grant_inport(k)+1;
            if(grant_arbiter(k)>num_inport)
                grant_arbiter(k) = 1;
            end
        end
```

```
        end
    end
    grant_arbiter;
    % sent packets out
    for j = 1:num_inport
        for k = 1:num_inport
            if(accept(j,k) == 1)
                viqueue_len(j,k) = viqueue_len(j,k)-1;
            end
        end
    end

    % collect statistics
    num_packet_inqueue = sum(viqueue_len(:));
    total_delay = total_delay + num_packet_inqueue;
end

offered_load = total_packet/(SIM_TIME*num_inport)
average_delay = total_delay/total_packet
save result

end
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึม Prioritized iSLIP

```
clc
clear all
close all

%s= rand('state');
for offered_load = 0.2:0.05:1.0
num_inport = 16;
num_outport = 16;
buffer_size = 3000;
YES  =1;
NO = 0;
EMPTY = 0;
OCCUPIED = 1;
ITERATIVE_MAX = 1;
%offered_load = 1.0;
SIM_TIME = 50000;
percent_class1 = 0.1;
num_priority = 2;

viqueue = zeros(num_inport,num_outport,buffer_size);
viqueue(:,:,:) = EMPTY;
viqueue1_len = zeros(num_inport,num_outport);
viqueue2_len = zeros(num_inport,num_outport);

% islip
request = zeros(num_inport,num_outport);
grant = zeros(num_inport,num_outport);
accept = zeros(num_inport,num_outport);
grant_arbiter = ones(num_outport,num_priority);
accept_arbiter = ones(num_inport,num_priority);


total_packet_loss = 0;
total_packet = 0;
total_packet1 = 0;
total_packet2 = 0;
total_delay = 0;
total_delay1 = 0;
total_delay2 = 0;
current_inport = 0;
current_outport = 0;

for time_slot = 1:SIM_TIME
%    fprintf('Time slot======= %d',time_slot)
    newpacket = (rand(1,num_inport)<offered_load);
    num_packet = sum(newpacket);
    total_packet = total_packet + num_packet;

    priority = ((rand(1,num_inport)>percent_class1)+1).*newpacket;

    num_packet1 = sum(priority==1);
    num_packet2 = sum(priority==2);
    total_packet1 = total_packet1+num_packet1;
    total_packet2 = total_packet2+num_packet2;

    outport = randint(1,num_inport,[1 num_outport]).*newpacket;
```

```
    for k = 1:num_inport
        if(outport(k)~=0)
            if(priority(k)==1)
                curr_len = viqueue1_len(k,outport(k));
                if(curr_len<buffer_size)
%                    viqueue(k,outport(k),curr_len+1) = OCCUPIED;
                    viqueue1_len(k,outport(k)) = curr_len + 1;
                else
                    disp('#####Packet loss');
                    total_packet_loss = total_packet_loss + 1;
                end
            else
                curr_len = viqueue2_len(k,outport(k));
                if(curr_len<buffer_size)
%                    viqueue(k,outport(k),curr_len+1) = OCCUPIED;
                    viqueue2_len(k,outport(k)) = curr_len + 1;
                else
                    disp('#####Packet loss');
                    total_packet_loss = total_packet_loss + 1;
                end
            end
        end
    end
    viqueue1_len;
    viqueue2_len;

    % islip
        request1 = (viqueue1_len(:,:)>0);
        request2 = (viqueue2_len(:,:)>0);
        request(:,:) = request1|request2;
        request_priority = request1*1 + (request2&~request1)*2;

    request;
    request_priority;

    % grant
    grant = zeros(num_inport,num_outport);
    grant_priority = zeros(num_inport,num_outport);
    actual_grant_inport = zeros(num_outport,num_priority);
    for k = 1:num_outport
        %current_inport = grant_arbiter(k);
        no_class1_request = sum(request_priority(:,k)==1);
        if(no_class1_request>0)
            current_inport = grant_arbiter(k,1);
            for j = 1:num_inport

                if((request(current_inport,k)==1)&(request_priority
                (current_inport,k)==1))
                  grant(current_inport,k) = 1;
                  actual_grant_inport(k,1) = current_inport;
                  grant_priority(current_inport,k) = 1;
                  break
                else
                  current_inport = current_inport+1;
                  if (current_inport > num_inport)
                      current_inport = 1;
                  end
                end
            end
        elseif(no_class1_request==0)
            current_inport = grant_arbiter(k,2);
```

```
        for j = 1:num_inport

              if((request(current_inport,k)==1)&(request_priority
              (current_inport,k)==2))
                grant(current_inport,k) = 1;
                actual_grant_inport(k,2) = current_inport;
                grant_priority(current_inport,k) = 2;
                break
              else
                current_inport = current_inport+1;
                if (current_inport > num_inport)
                    current_inport = 1;
                end
              end
        end
      else
            error('Request cannot be les than zero');

        end
end
grant;
actual_grant_inport;
grant_priority;

% accept
accept = zeros(num_inport,num_outport);
for j = 1:num_inport
    no_class1_grant = sum(grant_priority(j,:)==1);
    if (no_class1_grant > 0)
        current_outport = accept_arbiter(j,1);
        for k = 1:num_outport

              if((grant(j,current_outport)==1)&(grant_priority(j,
              current_outport)==1))
                accept(j,current_outport) = 1;
                accept_arbiter(j,1) = current_outport+1;
                if(accept_arbiter(j,1)>num_outport)
                    accept_arbiter(j,1) = 1;
                end
                break
              else
                current_outport = current_outport +1;
                if(current_outport > num_outport)
                    current_outport = 1;
                end
              end
        end
  elseif (no_class1_grant ==0)
        current_outport = accept_arbiter(j,2);
        for k = 1:num_outport

              if((grant(j,current_outport)==1)&(grant_priority(j,
              current_outport)==2))
                accept(j,current_outport) = 1;
                accept_arbiter(j,2) = current_outport+1;
                if(accept_arbiter(j,2)>num_outport)
                    accept_arbiter(j,2) = 1;
                end
                break
              else
                current_outport = current_outport +1;
```

```
                if(current_outport > num_outport)
                    current_outport = 1;
                end
              end
        end
    else
            error('grant cannot be les than zero');

    end
end

accept;
accept_arbiter;

% update grant arbiter
for k = 1:num_outport
    packTx = find(accept(:,k));
    if(~isempty(packTx))
        for j=1:num_inport
            if(grant_priority(j,k)==1)
                if(actual_grant_inport(k,1)==grant_arbiter(k,1))
                    grant_arbiter(k,1) = grant_arbiter(k,1)+1;
                    if(grant_arbiter(k,1)>num_inport)
                        grant_arbiter(k,1) = 1;
                    end
                    break
                else

                    grant_arbiter(k,1)=actual_grant_inport(k,1)+;
                    if(grant_arbiter(k,1)>num_inport)
                        grant_arbiter(k,1) = 1;
                    end
                    break
                end
            elseif(grant_priority(j,k)==2)
                if(actual_grant_inport(k,2)==grant_arbiter(k,2))
                    grant_arbiter(k,2) = grant_arbiter(k,2)+1;
                    if(grant_arbiter(k,2)>num_inport)
                        grant_arbiter(k,2) = 1;
                    end
                    break
                else

                    grant_arbiter(k,2)=actual_grant_inport(k,2)+;
                    if(grant_arbiter(k,2)>num_inport)
                        grant_arbiter(k,2) = 1;
                    end
                    break
                end
            end
        end
    end
end
grant_arbiter;

% sent packets out
for j = 1:num_inport
    for k = 1:num_inport
        if(accept(j,k) == 1)
            if(request_priority(j,k)==1)
                viqueue1_len(j,k) = viqueue1_len(j,k)-1;
```

```
                else
                    viqueue2_len(j,k) = viqueue2_len(j,k)-1;
                end
            end
        end
    end

    % collect statistics
    num_packet_inqueue1 = sum(viqueue1_len(:));
    total_delay1 = total_delay1 + num_packet_inqueue1;
    num_packet_inqueue2 = sum(viqueue2_len(:));
    total_delay2 = total_delay2 + num_packet_inqueue2;
end

offered_load = total_packet/(SIM_TIME*num_inport)
average_delay1 = total_delay1/total_packet1
average_delay2 = total_delay2/total_packet2
average_delay=((total_packet1*average_delay1)+(total_packet2*average_
delay2))/total_packet

data = fopen('data.txt','a+');
fprintf(data,'\n%.2f\t%.4f\t%.4f\t%.4f\n',offered_load,average_delay1
,average_delay2,average_delay);
fclose(data);
%save resultpriority2
%rand('state',s)

end
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายสมัชชาย ศรีนนท์ เกิดเมื่อวันที่ 5 กรกฎาคม พ.ศ. 2522 ที่อำเภอเมืองศรีสะเกษ จังหวัดศรีสะเกษ สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า (ไฟฟ้าสื่อสาร) มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2544 และศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า (ไฟฟ้าสื่อสาร) จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545

ประสบการณ์การทำงาน - วิศวกรไฟฟ้า บริษัทเอกชน KR precision Public company limited.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย