การปรับปรุงสวิตช์แพ็กเกตที่สามารถรองรับทราฟฟิกสองประเภทและรับประกันคุณภาพการให้บริการโดยใช้เทคนิคเอนเวโลปและการจัดการบัฟเฟอร์

เรือเอก อำนาจรัตน์ โสภณธรรมพัฒน์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2838-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

IMPROVEMENT OF PACKET SWITCHES SUPPORTING TWO-TYPE TRAFFIC WITH GUARANTEED QUALITY OF SERVICE USING ENVELOPE TECHNIQUE AND BUFFER MANAGEMENT

Lt. Amnatrat Sophonthammaphat

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 2005

ISBN 974-53-2838-3

หัวข้อวิทยานิพนธ์	การปรับปรุงสวิตช์แพ็กเกตที่สามารถรองรับทราฟฟิกสองประเภทและรับประกันคุณภาพการให้บริการ โดยใช้เทคนิคเอนเวโลปและการจัดการบัฟเฟอร์
โดย	เรือเอก อำนาจรัตน์ โสภณธรรมพัฒน์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ

คณะวิศวกรรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล)

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

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

.....กรรมการ
(อาจารย์ สุวิทย์ นาคพีระยุทธ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อำนาจรัตน์ โสภณธรรมพัฒน์, เรือเอก : การปรับปรุงสวิตช์แพ็กเกตที่สามารถรองรับทราฟฟิกสองประเภทและรับประกันคุณภาพการให้บริการ โดยใช้เทคนิคเอนเวโลปและการจัดการบัฟเฟอร์. (IMPROVEMENT OF PACKET SWITCHES SUPPORTING TWO-TYPE TRAFFIC WITH GUARANTEED QUALITY OF SERVICE USING ENVELOPE TECHNIQUE AND BUFFER MANAGEMENT) อ.ที่ปรึกษา: รศ.ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, 108 หน้า. ISBN 974-53-2838-3.

โครงข่ายอินเทอร์เน็ตแบ็กโบนประกอบไปด้วยสวิตช์หรือเราเตอร์อิเล็กทรอนิกส์ ซึ่งเชื่อมต่อระหว่างกันด้วยข่ายเชื่อมโยงที่เป็นเส้นใยแก้วนำแสง ในปัจจุบันด้วยการพัฒนาเทคโนโลยี DWDM ทำให้เส้นใยแก้วนำแสงมีความจุเกินกว่า 1 Tbps และมีความเร็วในการส่งข้อมูล 40 Gbps (OC768c) เมื่อเปรียบเทียบความจุที่สูงมากของเส้นใยแก้วนำแสงกับความจุของสวิตช์ จึงทำให้เกิดปัญหาคอขวดขึ้นในโครงข่าย ผลก็คือจึงมีความต้องการสวิตช์ที่มีความเร็วในการสวิตช์เพิ่มขึ้น นอกจากนี้ ยังมีแรงผลักดันที่ต้องการให้สวิตช์มีการรับประกันในคุณภาพการบริการ และสนับสนุนทราฟฟิกหลายประเภทในโครงข่าย ซึ่งเป็นหน้าที่อีกอย่างหนึ่งของสวิตช์ในโครงข่ายแบ็กโบน

ในวิทยานิพนธ์ฉบับนี้ จึงได้นำเสนอเทคนิคเอนเวโลป ซึ่งเป็นอัลกอริทึมจัดลำดับเซลล์สำหรับสวิตช์รอคิวที่อินพุตที่มีโครงสร้างสวิตช์แบบครอสบาร์และการจัดการบัฟเฟอร์แบบ VOQ เทคนิคเอนเวโลปถูกนำมาใช้ในการลดความถี่ในการคำนวณหาค่าการแมตช์ โดยการเพิ่มจำนวนเซลล์ในการส่งผ่านสวิตช์ในแต่ละไทม์สล็อต แต่ผลที่ตามมาก็คือทำให้ค่าเฉลี่ยการประวิงเวลาเพิ่มสูงขึ้น ดังนั้น เราสามารถใช้รูปแบบการจัดลำดับเอนเวโลปส่งผ่านสวิตช์ตามค่าเทรสโฮลด์ เพื่อลดค่าเฉลี่ยการประวิงเวลาและรับประกันทั้งการประวิงเวลาและแบนด์วิดท์ของสวิตช์ และจัดการคิวเป็นแบบ VOQ สองระนาบสำหรับทราฟฟิกสองประเภท ทั้งนี้ก็เพื่อให้การสนับสนุนคุณภาพการให้บริการ (QoS) หลายระดับ การเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต ทำงานได้ดีกว่าการเลือกเอนเวโลปที่บรรจุบางส่วนและการเลือกเอนเวโลปบรรจุเต็ม และใกล้เคียงกับการเลือกเอนเวโลปโดยวิธีผสมผสาน แต่ใช้เวลาในการหาค่าการแมตช์น้อยกว่า จึงเห็นได้ชัดเจนว่าการเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต เป็นการเลือกเอนเวโลปที่ให้สมรรถนะดีที่สุดเมื่อเทียบกับการเลือกเอนเวโลปวิธีอื่น ๆ ในทุกย่านทราฟฟิก

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต.....

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา.....

ปีการศึกษา 2548

4570652221 : MAJOR ELECTRICAL ENGINEERING

KEY WORD: PACKET SWITCH / BOTTLENECK / ENVELOPE / QOS / INTERNET BACKBONE

AMNATRAT SOPHONTHAMMAPHAT, Lt. : IMPROVEMENT OF PACKET SWITCHES SUPPORTING TWO-TYPE TRAFFIC WITH GUARANTEED QUALITY OF SERVICE USING ENVELOPE TECHNIQUE AND BUFFER MANAGEMENT. THESIS ADVISOR : ASSOC.PROF.LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., 108 pp. ISBN 974-53-2838-3.

The internet backbone is composed of high-speed electronic switches or routers which are connected via fiber-optic links. Today, with the deployment of dense wavelength-division multiplexing (DWDM) technologies, the overall transmission capacity through optical fibers has increased to over 1 Tbps with line rates of 40 Gbps (OC768c). Compared with the tremendous capacity of optical links, the switching capacity enhancement of switches becomes a relative bottleneck for the Internet backbone. As a result, there is need for building faster switches. Moreover, there is a pressing demand for guaranteed quality-of-service (QoS) and supporting multiple class traffic in the Internet, which is another responsibility of backbone switches.

In this thesis, we present the Envelope technique, it is a scheduling algorithm for input-queued switches based on a non-blocking crossbar switch fabric and virtual output queuing (VOQ). The Envelope is used in order to decrease the frequency of the matching computation by increasing the number of cells through the switch fabric. Consequently, the Envelope increases latencies in the switch so we can use the threshold envelope scheduling scheme that can reduce the latencies for providing bandwidth and delay guarantees and use two planes of VOQ for two classes of traffic for supporting multiple QoS. The dynamic threshold envelope selection performs better than the partially-filled envelope and the filled envelope and nearly equally to the mixed first-two envelope selection but uses time to match less than the mixed first-two envelope selection. Obviously, the dynamic threshold envelope selection is the best performance of envelope selection at any traffic.

Department	Electrical Engineering	Student's signature	A. Amnatrat Sophonthammaphat
Field of study	Electrical Engineering	Advisor's signature	
Academic year	2005		

กิตติกรรมประกาศ

การที่วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ก็ด้วยความช่วยเหลืออย่างดียิ่งจาก รศ.ดร. ลัญฉกร วุฒิสิทธิกุลกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ ด้วยดีเสมอมา ขอขอบพระคุณครูบาอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้วิจัย และขอบคุณเพื่อนนิสิตสาขาโทรคมนาคมที่เป็นกำลังใจและช่วยเหลือด้วยดีเสมอมา

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดามารดาและผู้มีพระคุณทุกท่าน ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ฎ

สารบัญภาพ..... ฎ

บทที่

1. บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์ 3

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์ 3

1.4 ขั้นตอนและวิธีดำเนินงาน..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 4

2. หลักการพื้นฐานของการสวิตช์แพ็กเกต

2.1 การสวิตช์แพ็กเกต..... 5

2.2 หลักการของการสวิตช์..... 9

2.2.1 การติดขัดภายใน..... 9

2.2.2 การติดขัดภายนอก..... 9

2.2.3 การติดขัดที่หัวคิว..... 10

2.2.4 การทำมัลติคาสต์..... 10

2.3 การจำแนกสถาปัตยกรรมสวิตช์แพ็กเกต..... 10

2.3.1 สวิตช์แบบแบ่งเวลา..... 11

2.3.1.1 สวิตช์แบบใช้สื่อนำสัญญาณร่วมกัน 11

2.3.1.2 สวิตช์แบบใช้หน่วยความจำร่วมกัน 12

2.3.2 สวิตช์แบบแบ่งเส้นทาง 13

2.3.2.1 สวิตช์แบบเส้นทางเดียว..... 14

2.3.2.2 สวิตช์แบบหลายเส้นทาง 19

2.4 สมรรถนะของสวิตช์..... 23

2.4.1 สวิตช์แบบรอคิวที่เอาต์พุต..... 23

2.4.2 สวิตช์แบบรอคิวที่อินพุต..... 25

2.5 สรุปท้ายบท..... 27

3. สถาปัตยกรรมสวิตช์แพ็กเกต

3.1 สวิตช์แบบรอคิวที่เอาต์พุต..... 29

3.1.1 สวิตช์แบบ SOQ..... 29

3.1.2 สวิตช์แบบ MOQ..... 30

3.2 สวิตช์แบบรอคิวที่อินพุต 31

3.2.1 อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์..... 32

3.2.2 สวิตช์แบบ MIQ..... 34

3.3 สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต..... 36

3.4 สวิตช์แบบต่อขนาน..... 37

3.5 สรุปท้ายบท..... 39

4. อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์

4.1 หลักการพื้นฐาน..... 42

4.2 การแมตช์ขนาดสูงที่สุด (maxsize)..... 44

4.2.1 อัลกอริทึม PIM..... 45

4.2.2 อัลกอริทึม RRM..... 47

4.2.3 อัลกอริทึม iSLIP..... 48

4.2.4 อัลกอริทึม DRRM..... 51

4.2.5 สมรรถนะของอัลกอริทึม maxsize 52

4.3 การแมตช์น้ำหนักสูงที่สุด (maxweight)..... 53

4.3.1 อัลกอริทึม LQF..... 54

4.3.2 อัลกอริทึม OCF..... 55

4.3.3 อัลกอริทึมแบบวนซ้ำ..... 55

4.3.3.1 อัลกอริทึม iLQF..... 56

4.3.3.2 อัลกอริทึม iOCF 57

4.3.4 สมรรถนะของอัลกอริทึม maxweight 57

4.4 สรุปท้ายบท..... 58

5. เทคนิคเอนเวโลปที่นำเสนอ

5.1 เทคนิคเอนเวโลป..... 60

5.2 แบบจำลองของสวิตช์ที่นำเสนอ..... 62

5.3 อัลกอริทึมจัดลำดับเซลล์ที่นำเสนอ..... 63
5.4 ทราฟฟิกที่ใช้ทดสอบ..... 64
5.5 ตัววัดสมรรถนะของสวิตช์..... 67
5.6 ความต้องการในการออกแบบสวิตช์แพ็กเกตความเร็วสูง..... 67
5.7 การจัดการบัฟเฟอร์ที่นำเสนอ..... 68
6. ผลการทดสอบ
6.1 อัลกอริทึมจัดลำดับเซลล์ที่ใช้ร่วมกับเทคนิคเอนเวโลป..... 69
6.2 วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างสวิตช์ 71
วิธีที่ 1 เลือกเอนเวโลปที่มีเซลล์ที่หัวคิว 71
วิธีที่ 2 เลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม 75
วิธีที่ 3 เลือกเอนเวโลปโดยวิธีผสมผสาน 77
วิธีที่ 4 เลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต 79
6.3 กรณีศึกษา: วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองประเภท..... 86
6.4 กรณีศึกษา: วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองประเภท และทราฟฟิกเป็นประเภทนอนยูนิฟอร์ม..... 91
กรณีที่ 1 กำหนดให้ $\rho = r\rho + (1-r)\rho$ โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนต่างกัน... 91
กรณีที่ 2 กำหนดให้ $\rho = r\rho + (1-r)\rho$ โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนเดียวกัน.. 93
6.5 ผลการเปรียบเทียบ 95
6.6 ผลการวิเคราะห์ปัจจัย..... 95
7. สรุปผลการวิจัย ข้อเสนอแนะ
7.1 สรุปผลการวิจัย..... 97
7.1.1 ข้อดีของเทคนิคเอนเวโลป..... 97
7.1.2 ข้อเสียของเทคนิคเอนเวโลป 98
7.2 แนวทางแก้ไข..... 98
รายการอ้างอิง 99
บรรณานุกรม..... 101
ภาคผนวก
ผนวก ก ประมาณค่าการแจกแจงแบบปัวส์ซงจากการแจกแจงแบบทวินาม เมื่อ $\mathbf{N} \to \infty$ 104
ผนวก ข ตัวอย่างซอร์สโคดของอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ 106

ประวัติผู้เขียนวิทยานิพนธ์ 108

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง หน้า

2.1 เปรียบเทียบโครงข่ายการสวิตช์วงจรกับโครงข่ายการสวิตช์แพ็กเกต 7

2.2 ค่าทรูพุตสูงสุดของสวิตช์รอคิวที่อินพุต กับการจัดการรอคิวแบบ FIFO, $\rho = 1$ 27

3.1 ค่าทรูพุตสูงสุดของสวิตช์แบบรอคิวที่อินพุตขนาด NxN กับวิธีการเลือกวินโดว์..... 31

3.2 ลักษณะของอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์แบบต่าง ๆ 34

6.1 ขนาดเอนเวโลปที่เหมาะสมสำหรับอัลกอริทึมจัดลำดับเซลล์ 95

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ หน้า

2.1 โครงข่ายการสวิตช์แพ็กเกตอย่างง่าย..... 6

2.2 การจำแนกสถาปัตยกรรมสวิตช์ 11

2.3 สวิตช์แบบใช้สื่อนำสัญญาณร่วมกัน 12

2.4 สวิตช์แบบใช้หน่วยความจำร่วมกัน 13

2.5 โครงสร้างสวิตช์แบบครอสบาร์ 15

2.6 สวิตช์แบบครอสบาร์กับรูปแบบการวางบัฟเฟอร์ 16

2.7 สวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด..... 17

2.8 โครงสร้างสวิตช์แบบบานยาน ขนาด 8x8 18

2.9 ตัวอย่างการติดขัดภายในสวิตช์แบบคลอสสามสเตจ 20

2.10 แสดงเงื่อนไขการไม่ติดขัดสำหรับสวิตช์แบบคลอสสามสเตจ 21

2.11 โครงสร้างสวิตช์แบบมัลติเพลน 22

2.12 แผนภาพการเปลี่ยนสถานะลูกโซ่มาร์คอฟเวลาไม่ต่อเนื่อง..... 24

3.1 แบบจำลองของสวิตช์พื้นฐาน 29

3.2 แสดงการติดขัดที่หัวคิว 30

3.3 สวิตช์แบบ MIQ 35

3.4 แบบจำลองสวิตช์แบบ CIOQ และฟังก์ชันแบ็กเพรสเชอร์... 36

3.5 สถาปัตยกรรมสวิตช์แบบ PPS 38

4.1 แบบจำลองของสวิตช์แบบรอคิวที่อินพุต 42

4.2 กราฟ G และการแมตช์ M บน G 44

4.3 แสดงตัวอย่างขั้นตอนทั้ง 3 ขั้นตอนของอัลกอริทึม PIM ที่ไม่มีการวนซ้ำ 46

4.4 แสดงตัวอย่างการทำงานของอัลกอริทึม RRM 48

4.5 แสดงตัวอย่างการทำงานของอัลกอริทึม iSLIP 49

4.6 แสดงตัวอย่างการทำงานของอัลกอริทึม DRRM 52

4.7 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ กับอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ..... 53

4.8 ตัวอย่างอัลกอริทึม LQF 55

4.9 สมรรถนะของสวิตช์ด้านการประวิงเวลาของอัลกอริทึม maxweight 58

5.1 แบบจำลองโครงสร้างสวิตช์และตัวตัดสินใจของเอาต์พุต 63

5.2 ตัวอย่างทราฟฟิกประเภทยูนิฟอร์มและนอนยูนิฟอร์ม 66
6.1 สมรรถนะด้านการประวิงเวลาของอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ..... 70
6.2 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม PIM..... 72
6.3 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม iSLIP 72
6.4 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม DRRM 73
6.5 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของอัลกอริทึมต่าง ๆ ที่ขนาดเอนเวโลป
เท่ากับ 4 เซลล์ 73
6.6 แสดงฮิสโทแกรมของอินพุตบัฟเฟอร์ที่โหลดขนาดต่าง ๆ 74
6.7 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม PIM 75
6.8 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม iSLIP 76
6.9 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม DRRM 76
6.10 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม PIM 77
6.11 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม iSLIP 78
6.12 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม DRRM 78
6.13 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปวิธีต่าง ๆ ของ
อัลกอริทึม PIM 80
6.14 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปวิธีต่าง ๆ ของ
อัลกอริทึม iSLIP 81
6.15 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปวิธีต่าง ๆ ของ
อัลกอริทึม DRRM 82
6.16 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปขนาดต่าง ๆ ของ
อัลกอริทึม PIM 83
6.17 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปขนาดต่าง ๆ ของ
อัลกอริทึม iSLIP..... 84
6.18 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปขนาดต่าง ๆ ของ
อัลกอริทึม DRRM..... 85
6.19 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือก
เอนเวโลปที่มีเซลล์ที่หัวคิว 87

6.20 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเอนเวโลปบรรจุเต็ม..... 87
6.21 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเอนเวโลปแบบผสมผสาน..... 88
6.22 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต..... 88
6.23 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต เมื่อโหลดมี H เป็นสัดส่วนต่าง ๆ กัน..... 90
6.24 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์กรณีทราฟฟิกสองประเภทและทราฟฟิกเป็นแบบนอนยูนิฟอร์ม กรณีที่ 1 92
6.25 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์กรณีทราฟฟิกสองประเภทและทราฟฟิกเป็นแบบนอนยูนิฟอร์ม กรณีที่ 2 94

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน โครงข่ายอินเทอร์เน็ตโดยเฉพาะที่เป็นโครงข่ายแบ็กโบน (Internet backbone) ประกอบไปด้วยสวิตช์หรือเราเตอร์อิเล็กทรอนิกส์ความเร็วสูง ซึ่งเชื่อมต่อกันด้วยเส้นใยแก้วนำแสง (fiber-optic links) ประกอบกับการนำเอาเทคนิคการมัลติเพลกซ์สัญญาณแบบ DWDM (Dense Wavelength Division Multiplexing) มาใช้ ทำให้เส้นใยแก้วนำแสงแต่ละเส้นสามารถบรรจุช่องสัญญาณที่มีหลายความยาวคลื่นจำนวนหลายช่องสัญญาณ ซึ่งแต่ละช่องสัญญาณรับส่งข้อมูลด้วยความเร็ว 2.5 (OC-48), 10 (OC-192) หรือ 40 Gbps (OC-768) ทำให้ความจุรวมในการรับส่งข้อมูลผ่านเส้นใยแก้วนำแสงแต่ละเส้นสูงเกินกว่า 1 Tbps เมื่อเปรียบเทียบความจุจำนวนมหาศาลของเส้นใยแก้วนำแสงกับความจุในการสวิตช์ของสวิตช์หรือเราเตอร์อิเล็กทรอนิกส์ความเร็วสูง จึงทำให้เกิดปัญหาคอขวด (bottleneck) ขึ้น ในการสวิตช์ข้อมูลผ่านสวิตช์ในโครงข่ายอินเทอร์เน็ตแบ็กโบน จึงเป็นความจำเป็นเร่งด่วนที่จะต้องหาวิธีแก้ไข ทั้งนี้ ก็เพื่อที่จะรับประกันคุณภาพการให้บริการของโครงข่ายอินเทอร์เน็ต ซึ่งตกเป็นความรับผิดชอบของสวิตช์หรือเราเตอร์ในโครงข่ายอินเทอร์เน็ตแบ็กโบน

ในขณะที่สวิตช์แบบรอคิวที่เอาต์พุต (Output-Queued (OQ) Switches) นั้น ให้สมรรถนะของสวิตช์ที่ดีกว่าสวิตช์แบบรอคิวที่อินพุต (Input-Queued (IQ) Switches) แต่การรอคิวที่เอาต์พุตของสวิตช์ก็ไม่เหมาะสมในการนำมาใช้เป็นสวิตช์ความเร็วสูง เนื่องจากโครงสร้างการสวิตช์ (switching fabric) และหน่วยความจำ (memory) จะต้องทำงานที่ความเร็วเป็นจำนวน N เท่าของความเร็วแพ็กเกต (line rate) โดยที่ N คือขนาดของสวิตช์ สวิตช์แบบ IQ นั้นสามารถลดแบนด์วิดท์ของหน่วยความจำได้เนื่องจากทำงานที่ความเร็วเท่ากับความเร็วแพ็กเกต จึงเป็นสิ่งที่ดึงดูดความสนใจของนักออกแบบสวิตช์ที่จะนำไปสร้างเป็นสวิตช์ความเร็วสูง หรือสวิตช์ที่มีพอร์ตเป็นจำนวนมาก นอกจากนี้ เพื่อให้เป็นไปตามความต้องการในการออกแบบ ทั้งความต้องการด้านความจุของสวิตช์ที่สูงขึ้น และรับประกันในคุณภาพการให้บริการ (Quality of Service: QoS) สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (Combined Input and Output-Queued (CIOQ) Switches) จึงได้รับการออกแบบมาเพื่อให้เป็นไปตามความต้องการดังกล่าว สวิตช์แบบ CIOQ มีโครงสร้างการสวิตช์ทำงานที่ความเร็วสูงกว่าความเร็วแพ็กเกตเท่ากับ s เท่า ทำให้ในแต่ละไทม์สล็อต (time slot) สามารถส่งแพ็กเกตจากแต่ละอินพุตไปยังเอาต์พุตของสวิตช์ได้มากที่สุด s แพ็กเกต แน่นอนจึงมีความจำเป็นที่จะต้องมีการวางบัฟเฟอร์ที่เอาต์พุตเพื่อให้แพ็กเกตที่ยังไม่ได้ถูกส่งออกจากสวิตช์รออยู่ในคิวจนกว่าจะได้รับการส่งออก [1]

ต่อมาได้มีงานวิจัยเป็นจำนวนมาก พยายามที่จะค้นหาว่าความเร็วในการสวิตช์ของสวิตช์แบบ CIOQ ที่เหมาะสมและต่ำที่สุดที่สอดคล้องกับสมรรถนะของสวิตช์แบบ OQ และเราเรียกการกระทำเช่นนี้ว่าการอีมูเลชันสวิตช์แบบ OQ (OQ emulation) เนื่องจากสวิตช์แบบ OQ นั้นให้สมรรถนะของสวิตช์ที่ดีที่สุด จึงมักจะใช้เป็นตัวเปรียบเทียบกับสมรรถนะของสวิตช์ประเภทอื่น ผู้วิจัยได้แสดงให้เห็นว่าสวิตช์แบบ CIOQ ที่ใช้การแยกคิวทางตรรกะที่อินพุตแบบ VOQ (Virtual Output Queueing) และใช้อัลกอริทึมจัดลำดับแพ็กเกตส่งผ่านสวิตช์ที่เหมาะสม จะทำให้สวิตช์สามารถรองรับทราฟฟิกได้ทุกประเภทและรองรับการเพิ่มขนาดสวิตช์ ที่ความเร็วในการสวิตช์เป็น 2 เท่าของความเร็วแพ็กเกต แต่อย่างไรก็ตาม การเพิ่มความเร็วในการสวิตช์ของโครงสร้างการสวิตช์นั้น ทำให้หน่วยความจำมีเวลาในการประมวลผลลดลง และแม้แต่ตัวจัดลำดับแพ็กเกตส่งผ่านสวิตช์ (scheduler) ก็มีเวลาประมวลผลในการตัดสินใจเลือกแพ็กเกตที่จะส่งผ่านสวิตช์ลดลง ซึ่งกระทำได้ยากในการนำไปสร้างเป็นสวิตช์ความเร็วสูง ดังนั้น จึงได้มีงานวิจัยบางชิ้น [1], [2] ได้นำเอาสมรรถนะที่ดีของสวิตช์แบบ OQ โดยจะไม่เพิ่มความเร็วในการสวิตช์มาทำการศึกษา โดยได้นำเอาสวิตช์แบบ OQ มาต่อแบบขนานกันและเรียกสวิตช์แบบนี้ว่าสวิตช์แบบต่อขนาน (Parallel Packet Switch: PPS) สวิตช์แบบต่อขนานนั้นประกอบไปด้วยส่วนต่าง ๆ ดังนี้คือ อินพุตดีมัลติเพลกเซอร์ สวิตช์ต่อขนาน และเอาต์พุตมัลติเพลกเซอร์ หลักการทำงานของสวิตช์แบบ PPS คือแพ็กเกตที่เข้ามาเป็นลำดับมีความเร็วเท่ากับ R จะถูกกระจายออกไปแพ็กเกตต่อแพ็กเกตโดยตัวดีมัลติเพลกซ์ส่งผ่านสวิตช์ที่ต่อขนานกัน k ตัว แล้วจึงรวมกันใหม่อีกครั้งโดยตัวมัลติเพลกซ์ จะเห็นว่าสวิตช์แต่ละตัวที่ต่อขนานกันทำการสวิตช์ที่ความเร็ว R/k ซึ่งต่ำกว่าความเร็วแพ็กเกตที่อินพุตของสวิตช์ ทำให้หน่วยความจำของสวิตช์ที่ต่อขนานกันตอนกลาง สามารถทำงานได้ที่ความเร็วต่ำกว่าความเร็วแพ็กเกตที่อินพุต จึงนำเอาเทคนิคการใช้หน่วยความจำร่วมกัน (share memory type) มาประยุกต์ใช้ได้ และยังรับประกันในคุณภาพการให้บริการ (QoS) ซึ่งเป็นคุณลักษณะที่ดีที่ได้จากการนำเอาสวิตช์แบบ OQ มาต่อขนานกัน

อีกวิธีหนึ่ง ที่ใช้ในการแก้ปัญหาคอขวดในการสวิตช์ข้อมูลคือ การเพิ่มขนาดความยาวของหน่วยข้อมูลที่จะส่งผ่านโครงสร้างการสวิตช์โดยรวมแพ็กเกตหลาย ๆ แพ็กเกตที่จะไปยังเอาต์พุตเดียวกันส่งไปพร้อมกันเป็นลำดับ เรียกว่าเทคนิคเอนเวโลป (envelope) ซึ่งเทคนิคนี้อาจทำให้สมรรถนะของสวิตช์ทางด้านการประวิงเวลาลดลงไปบ้าง แต่ก็เหมาะสมในการนำมาใช้เร่งด่วนในการแก้ปัญหาคอขวด ซึ่งในงานวิทยานิพนธ์ฉบับนี้ จะนำเสนอเทคนิคเอนเวโลปและการจัดการบัฟเฟอร์แบบ VOQ และนำไปประยุกต์ใช้กับอัลกอริทึมจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์แบบต่าง ๆ แล้วทำการประเมินสมรรถนะของสวิตช์ที่นำเทคนิคเอนเวโลปไปประยุกต์ใช้ เพื่อให้สามารถรองรับทราฟฟิกได้ทั้งประเภทเรียลไทม์ (real-time traffic) และประเภทดาตา (data type traffic) ภายใต้ข้อกำหนดคือ ต้องรับประกันในคุณภาพการให้บริการของสวิตช์

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อปรับปรุงและพัฒนาสถาปัตยกรรมสวิตช์แพ็กเกตความเร็วสูง ที่มีขีดความสามารถในการรองรับทราฟฟิกทั้งประเภทเรียลไทม์และประเภทดาตา โดยการใช้เทคนิคเอนเวโลปและการจัดการบัฟเฟอร์ที่สอดคล้องกัน ในการเพิ่มสมรรถนะของสวิตช์ในการสวิตช์ข้อมูลผ่านโครงสร้างการสวิตช์และรับประกันในคุณภาพการให้บริการของทราฟฟิกทั้งสองประเภท

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์

ศึกษาปัญหาคอขวดในการสวิตช์ข้อมูล และหาวิธีการที่จะแก้ไขปัญหา โดยการใช้เทคนิคเอนเวโลป คิดค้นวิธีการที่จะนำเทคนิคเอนเวโลปมาประยุกต์ใช้กับสวิตช์แพ็กเกตประเภทที่มีการวางบัฟเฟอร์ที่อินพุตของสวิตช์ เพื่อทำการปรับปรุงสวิตช์แพ็กเกตความเร็วสูงที่จะนำไปใช้ในโครงข่ายอินเทอร์เน็ตแบ็กโบน โดยจำกัดขอบเขตของวิทยานิพนธ์ ดังต่อไปนี้

1) สวิตช์เป็นแบบ**สวิตช์เซลล์** ซึ่งแพ็กเกตถูกแบ่งย่อยเป็น**เซลล์**ขนาดเท่ากันก่อนส่งผ่านโครงสร้างสวิตช์

2) เป็นสวิตช์ที่มีการวางบัฟเฟอร์ที่พอร์ตอินพุตของสวิตช์

3) การมาถึงพอร์ตอินพุตของเซลล์เป็นกระบวนการ i.i.d. Bernoulli

4) ทราฟฟิกเป็นประเภทเรียลไทม์และประเภทดาตา และสวิตช์รับประกันในคุณภาพการให้บริการกับทราฟฟิกทั้งสองประเภท

5) ทราฟฟิกที่จะส่งผ่านโครงสร้างการสวิตช์แยกออกเป็น 2 ชั้น (class) คือชั้นทราฟฟิกประเภทเรียลไทม์ โดยกำหนดให้มีระดับของลำดับความสำคัญสูง (high priority level) และ ชั้นทราฟฟิกประเภทดาตา โดยกำหนดให้มีระดับของลำดับความสำคัญต่ำ (low priority level)

1.4 ขั้นตอนและวิธีการดำเนินงาน

1) ศึกษาและวิเคราะห์สมรรถนะของสวิตช์แพ็กเกตความเร็วสูงประเภทต่าง ๆ ที่นำมาใช้ในโครงข่ายอินเทอร์เน็ต

2) ศึกษาปัญหาคอขวดในการสวิตช์ข้อมูลในโครงข่ายอินเทอร์เน็ตแบ็กโบน และเทคนิคต่าง ๆ ที่จะนำมาใช้แก้ปัญหานี้

3) จำลองแบบสวิตช์ที่มีการวางบัฟเฟอร์ที่พอร์ตอินพุตของสวิตช์ และนำเอาเทคนิคเอนเวโลปและการจัดการบัฟเฟอร์มาประยุกต์ใช้

4) วิเคราะห์สมรรถนะของสวิตช์และสรุปผลที่ได้จากการจำลองแบบ ตามข้อ 3

5) ปรับปรุงสวิตช์แพ็กเกตความเร็วสูงที่ประยุกต์ใช้เทคนิคเอนเวโลป ที่สามารถรองรับทราฟฟิกทั้งประเภทเรียลไทม์และประเภทดาตา และรับประกันในคุณภาพการให้บริการทราฟฟิกทั้งสองประเภท เพื่อที่จะนำไปใช้ในโครงข่ายอินเทอร์เน็ตแบ็กโบน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถแก้ปัญหาคอขวด ในการสวิตช์ข้อมูลของสวิตช์ในโครงข่ายอินเทอร์เน็ตแบ็กโบนซึ่งเป็นความต้องการเร่งด่วนของผู้ให้บริการโครงข่ายอินเทอร์เน็ตในปัจจุบัน

2) รองรับการเพิ่มขนาดของโครงข่าย (scalability) ในอนาคต เนื่องมาจากการเพิ่มขึ้นของผู้ใช้บริการโครงข่ายและแอปพลิเคชันใหม่ ๆ ที่เกิดขึ้นในระบบอินเทอร์เน็ต ซึ่งทำให้จำนวนพอร์ตของสวิตช์และความเร็วของแพ็กเกตเพิ่มมากขึ้น

3) สามารถปรับปรุงสวิตช์แพ็กเกตความเร็วสูง ที่ประยุกต์ใช้เทคนิคเอนเวโลปกับการจัดการบัฟเฟอร์ได้ เพื่อนำไปใช้กับโครงข่ายอินเทอร์เน็ตแบ็กโบนในอนาคตอันใกล้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
หลักการพื้นฐานของการสวิตช์แพ็กเกต

เมื่อประมาณปี ค.ศ.1970 ได้เริ่มมีงานวิจัยที่เกี่ยวกับสถาปัตยกรรมรูปแบบใหม่ของการสื่อสารข้อมูลดิจิตอลระยะทางไกล ที่เรียกว่า การสวิตช์แพ็กเกต (packet switching) ถึงแม้ว่าเทคโนโลยีการสวิตช์แพ็กเกตจะได้เริ่มขึ้นและถูกนำมาใช้ต่อเนื่องนับแต่เวลานั้น แต่ก็เป็นที่สังเกตได้ว่า หลักการพื้นฐานของการสวิตช์แพ็กเกตในปัจจุบันก็เป็นเช่นเดียวกันกับในอดีตที่ผ่านมานับแต่เวลาเริ่มต้น นอกจากนี้การสวิตช์แพ็กเกตก็ยังคงเป็นหนึ่งในไม่กี่เทคโนโลยีที่มีประสิทธิภาพในการสื่อสารข้อมูลระยะทางไกลในปัจจุบัน

2.1 การสวิตช์แพ็กเกต

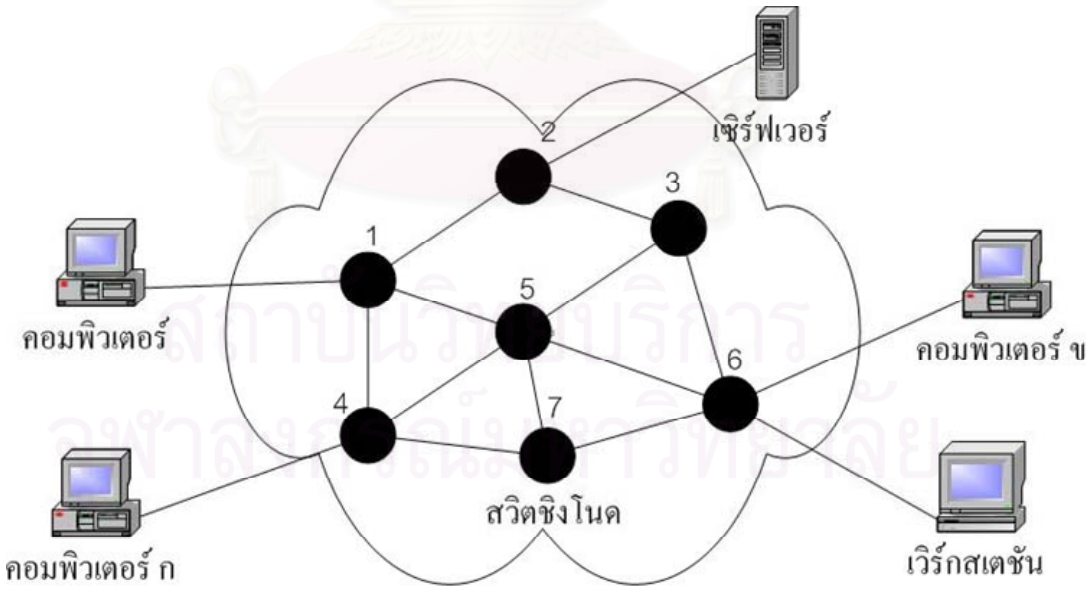
ในตอนเริ่มแรกนั้น โครงข่ายสื่อสารโทรคมนาคมระยะทางไกลใช้เทคโนโลยีการสวิตช์วงจร (circuit switching) ซึ่งเป็นเทคนิคที่ใช้ในการส่งข้อมูลเสียงของระบบโทรศัพท์ ซึ่งเมื่อมีการเรียก (call) ติดต่อระหว่างผู้ใช้ต้นทางกับปลายทาง อุปกรณ์การสวิตช์ในระบบโทรศัพท์จะหาช่องสัญญาณกายภาพ (physical channel) หรือเส้นทางผ่านจริงของสัญญาณ (physical path) เพื่อเชื่อมเส้นทางส่งข้อมูลตลอดเส้นทางระหว่างผู้ใช้ทั้งสอง ลักษณะสำคัญอย่างหนึ่งของการสวิตช์วงจรคือการที่จะส่งข้อมูลได้นั้นจะต้องสร้างการติดต่อก่อน (call setup) ซึ่งระบบโทรศัพท์จะหาช่องสัญญาณกายภาพผ่านชุมสายต่าง ๆ จะเห็นได้ว่าก่อนการส่งข้อมูลได้จะต้องเสียเวลาในการสร้างการติดต่อโดยมีการส่งสัญญาณติดต่อจนถึงปลายทาง และรอสัญญาณตอบรับกลับมาจึงจะส่งข้อมูลได้ ดังนั้นจึงไม่มีการติดขัดในการส่งข้อมูล

การส่งข้อมูลผ่านโครงข่ายนั้นแต่เดิมเป็นการส่งข้อมูลเสียง (voice) ซึ่งเมื่อได้ทำการติดต่อกันได้แล้วผู้ใช้บริการจะทำการส่งข้อมูลติดต่อกันตลอดเวลา จึงทำให้การใช้ประโยชน์จากช่องสัญญาณมีประสิทธิภาพสูง แต่ต่อมาได้มีการส่งข้อมูลประเภทดาตาในเครือข่ายคอมพิวเตอร์ ซึ่งในการติดต่อกันระหว่างผู้ใช้หรือโฮสต์นั้น เวลาส่วนใหญ่ของช่องสัญญาณจะว่าง ทำให้การส่งข้อมูลโดยการสวิตช์วงจรนั้นไม่มีประสิทธิภาพ นอกจากนั้น การส่งข้อมูลในโครงข่ายการสวิตช์วงจรจะต้องมีอัตราการส่งคงที่ตั้งแต่ต้นทางไปจนถึงปลายทาง ดังนั้น ทุก ๆ อุปกรณ์ที่อยู่ในเส้นทางการส่งจะต้องรับส่งข้อมูลด้วยอัตราการส่งเดียวกัน สิ่งนี้ไปจำกัดการใช้ประโยชน์จากโครงข่ายที่มีการเชื่อมต่อระหว่างโครงข่ายที่ประกอบไปด้วยโฮสต์คอมพิวเตอร์และเครื่องเวิร์กสเตชัน ที่มีอัตราการส่งที่แตกต่างกัน ซึ่งปัญหาเหล่านี้ แก้ไขได้โดยใช้เทคนิคการส่งข้อมูลแบบการสวิตช์แพ็กเกต เทคนิคการสวิตช์แพ็กเกตนี้ ข้อมูลจะถูกแบ่งออกเป็นเซกเมนต์เล็ก ๆ ที่มีการกำหนดขนาดสูงสุดไว้แน่นอน (เช่นประมาณ 1000 ไบต์) เรียกว่า แพ็กเกต แต่ละแพ็กเกตจะประกอบไปด้วยส่วนของ

ข้อมูลผู้ใช้กับส่วนของข้อมูลควบคุม (control information) ที่เรียกว่า เฮดเดอร์ แพ็กเกตเหล่านี้จะถูกส่งผ่านโนดต่าง ๆ แบบเก็บรอและส่งต่อ (store and forwarding) ซึ่งแต่ละโนดสามารถเปลี่ยนอัตราการส่งและเปลี่ยนรหัสต่าง ๆ ของข้อมูล ตลอดจนแก้ไขความผิดพลาดจากการส่งได้ แต่ละแพ็กเกตจะถูกส่งผ่านเส้นทางซึ่งอาจจะไม่ซ้ำกันจนกว่าจะถึงปลายทาง เมื่อถึงปลายทางแพ็กเกตข้อมูลเดียวกันจะถูกรวมเป็นข้อมูลอีกครั้งก่อนส่งให้แก่ผู้รับ

ลักษณะสำคัญอีกอย่างหนึ่งของการสวิตช์แพ็กเกตก็คือ ถึงแม้ว่าแพ็กเกตที่ส่งออกไปจะยังไม่ถึงปลายทาง แต่แพ็กเกตต่อ ๆ มาของข้อมูลชุดเดียวกันสามารถส่งต่อเนื่องกันไปเป็นลำดับเป็นการลดความล่าช้าตลอดจนเพิ่มปริมาณข้อมูลในการส่ง ซึ่งลักษณะเช่นนี้เรียกว่าไปป์ไลนิงเอฟเฟ็กต์ (pipelining effect)

ดังรูปที่ 2.1 สมมุติว่าเป็นโครงข่ายการสวิตช์แพ็กเกตอย่างง่าย เมื่อต้องการส่งแพ็กเกตจากสถานีต้นทางไปยังสถานีปลายทาง โดยที่แพ็กเกตจะประกอบด้วยเฮดเดอร์ซึ่งจะบ่งบอกหมายเลขปลายทาง แพ็กเกตจะถูกส่งจากสถานีต้นทางคือคอมพิวเตอร์ ก ไปยังโนด 4 ซึ่งจะทำการเก็บแพ็กเกตไว้ กำหนดเส้นทางขาออก และส่งแพ็กเกตออกไปยังเส้นทางขาออกเมื่อช่องสัญญาณว่างลง ไปยังโนด 5 หรือ โนด 7 ซึ่งจะส่งต่อไปยังโนด 6 จนไปถึงสถานีปลายทางคือคอมพิวเตอร์ ข



รูปที่ 2.1 โครงข่ายการสวิตช์แพ็กเกตอย่างง่าย [3]

ตารางที่ 2.1 เปรียบเทียบโครงข่ายการสวิตช์วงจรกับโครงข่ายการสวิตช์แพ็กเกต [4]

ข้อเปรียบเทียบ	การสวิตช์วงจร	การสวิตช์แพ็กเกต
เส้นทางถาวรในการส่งข้อมูล	ใช่	ไม่ใช่
ขนาดช่องสัญญาณ	คงที่	เปลี่ยนแปลงได้
แนวโน้มการใช้งานช่องสัญญาณ	ใช้งานไม่เต็มที่	ใช้งานเต็มที่
ทำงานแบบเก็บรอและส่งต่อ	ไม่ใช่	ใช่
แพ็กเกตใช้เส้นทางเดียวกันหมด	ใช่	ไม่ใช่
การสร้างการเชื่อมต่อก่อนส่งข้อมูล	ต้องทำ	ไม่ต้อง
ความคับคั่งของข้อมูลเกิดขึ้นที่	ระหว่างการสร้างการเชื่อมต่อ	เกิดขึ้นได้ทุกแพ็กเกต

จะเห็นว่า โครงข่ายการสวิตช์แพ็กเกต มีข้อดีที่เหนือกว่าโครงข่ายการสวิตช์วงจร ดังต่อไปนี้คือ

- การใช้งานช่องสัญญาณมีประสิทธิภาพสูงกว่า เนื่องจากเส้นทางการเชื่อมต่อระหว่างโนดมีลักษณะพลวัต (dynamic) แต่ละแพ็กเกตสามารถใช้งานร่วมกันได้ตลอดเวลา ซึ่งแตกต่างจากโครงข่ายการสวิตช์วงจรที่ผู้ใช้บริการโครงข่ายคู่หนึ่ง ๆ จะจับจองช่องสัญญาณตลอดเวลาแต่เวลาส่วนใหญ่นั้นเส้นทางการเชื่อมต่อระหว่างโนดว่างอยู่ แต่ไม่สามารถส่งข้อมูลของผู้ใช้รายอื่นได้

- โครงข่ายการสวิตช์แพ็กเกตสามารถเปลี่ยนแปลงอัตราการส่งได้ ทำให้สถานีต่าง ๆ ที่มีอัตราการส่งข้อมูลแตกต่างกันสามารถแลกเปลี่ยนแพ็กเกตกันได้ โดยผ่านโนดที่มีอัตราการส่งที่เหมาะสมกับแพ็กเกตนั้น ๆ

- กรณีที่เกิดความคับคั่งขึ้นในโครงข่าย ถ้าเป็นโครงข่ายการสวิตช์วงจรผลก็คือ การเรียกจะถูกทิ้งไป แต่ในโครงข่ายการสวิตช์แพ็กเกตนั้น แพ็กเกตถูกเก็บรอไว้ที่โนดรอการส่งออก ซึ่งอาจทำให้แพ็กเกตมีการประวิงเวลาเพิ่มมากขึ้นในการส่งผ่านโครงข่าย

- ลำดับความสำคัญ (priorities) ถ้าโนดได้รับแพ็กเกตไว้จำนวนหนึ่ง และเก็บรอไว้สำหรับส่งออก โนดสามารถส่งแพ็กเกตที่มีลำดับความสำคัญสูงกว่าออกก่อนแพ็กเกตที่มีลำดับความสำคัญต่ำกว่าได้

จากข้อดีที่กล่าวมาทำให้เครือข่ายคอมพิวเตอร์ซึ่งมีข้อมูลอยู่ในรูปดาตา จึงใช้การส่งข้อมูลแบบการสวิตช์แพ็กเกต มีเพียงส่วนน้อยที่เป็นแบบการสวิตช์วงจร

ในการส่งแพ็กเกตผ่านโครงข่ายการสวิตช์แพ็กเกตนั้น มีอยู่ 2 รูปแบบคือ แบบดาตาแกรม (datagram) และ แบบวงจรเสมือน (virtual circuit)

ในแบบดาตาแกรม จะไม่มีการกำหนดเส้นทางเดินของแพ็กเกตไว้ล่วงหน้า แพ็กเกตแต่ละแพ็กเกตจะมีอิสระในการเดินทางผ่านโนดต่าง ๆ ในโครงข่าย ดังนั้น แพ็กเกตที่ส่งไปยังผู้รับเดียวกันอาจเดินทางผ่านโนดในเส้นทางต่างกันได้ เมื่อไปถึงสถานีปลายทางแพ็กเกตอาจจะไม่เป็นไปตามลำดับที่ส่งออกไป จึงต้องมีการเรียงลำดับกันใหม่อีกครั้งก่อนส่งให้แก่ผู้รับ การส่งโดยวิธีนี้ทำให้เกิดความยืดหยุ่นในการส่งแพ็กเกต โดยเฉพาะในกรณีที่เกิดความคับคั่งของเส้นทางเดินบางเส้นทาง หรือเส้นทางเดินบางเส้นทางเสียหายจนใช้การไม่ได้ แพ็กเกตจะเลือกเส้นทางเดินอื่นโดยอัตโนมัติ นอกจากนี้การส่งแพ็กเกตสั้น ๆ โดยใช้ดาตาแกรมจะมีประสิทธิภาพการทำงานดีกว่าอีกวิธีหนึ่ง

วิธีส่งแพ็กเกตแบบที่สองเรียกว่า แบบวงจรเสมือน เส้นทางเดินของแพ็กเกตจะถูกกำหนดขึ้นมาก่อนการส่งข้อมูล ซึ่งมีลักษณะคล้ายกันกับการสวิตช์วงจร เส้นทางเดินของแพ็กเกตจะถูกจัดตั้งขึ้นมา โดยผู้ส่งจะต้องส่งแพ็กเกตที่เรียกว่า แพ็กเกตร้องขอการเรียก (call request packet) ไปยังทุก ๆ โนดตลอดเส้นทาง เรียกว่า วงจรเสมือน (virtual circuit) เมื่อได้รับการตอบรับจากทุกโนดแล้วจึงจะสามารถส่งแพ็กเกตได้ แต่ละแพ็กเกตจะต้องใส่รหัสวงจรเสมือนแทนหมายเลขต้นทางและปลายทางที่ใช้ในแบบดาตาแกรม เพื่อให้โนดที่อยู่ในเส้นทางทราบและจัดการส่งแพ็กเกตต่อไป วงจรเสมือนที่ใช้นี้แตกต่างจากเส้นทางเดินของข้อมูลที่ใช้ในแบบการสวิตช์วงจรสองประการ กล่าวคือ ประการแรก วงจรเสมือนไม่ได้ห้ามผู้ใช้อื่นใช้งาน ประการที่สอง วงจรเสมือนถูกจัดตั้งขึ้นทุกครั้งก่อนการส่งแพ็กเกต ดังนั้น การส่งแพ็กเกตไปยังที่เดิมในครั้งต่อไปอาจใช้วงจรเสมือนที่แตกต่างไปจากเดิมก็ได้ การยกเลิกวงจรเสมือนทำได้โดยกระบวนการการลบล้างการเรียก (call clearing)

วงจรเสมือนแบ่งออกตามขอบเขตของโครงข่ายได้เป็น 2 ประเภท คือ วงจรเสมือนภายในและวงจรเสมือนภายนอก วงจรเสมือนภายในนั้นเป็นการกำหนดเส้นทางเดินของแพ็กเกตภายในโครงข่ายย่อยโครงข่ายหนึ่งอย่างเป็นอิสระต่อวิธีการที่ใช้ในโครงข่ายย่อยอื่น ตั้งแต่เราเตอร์ตัวแรกที่รับแพ็กเกตจากโครงข่ายย่อยอื่นเข้ามา (หรือเป็นผู้ส่งแพ็กเกตเอง) ไปจนถึงเราเตอร์ตัวสุดท้ายก่อนที่จะส่งแพ็กเกตนั้นออกไปภายนอก วงจรเสมือนภายนอกนั้นเปรียบได้กับการกำหนดเส้นทางเดินวงจรเสมือนในระดับโครงข่ายย่อยเท่านั้นคือ จากโครงข่ายย่อยของผู้ส่งจะต้องผ่านโครงข่ายย่อยใดบ้างไปจนถึงโครงข่ายย่อยของผู้รับ

การเลือกส่งแพ็กเกตนั้น จะใช้วิธีใดขึ้นอยู่กับระยะเวลาในการส่งข้อมูลและปริมาณข้อมูลเป็นสำคัญ สำหรับแพ็กเกตสั้น ๆ นั้นวิธีดาตาแกรมจะมีประสิทธิภาพดีกว่า แต่ถ้ามีปริมาณแพ็กเกตมากก็ควรใช้วิธีวงจรเสมือนจะมีประสิทธิภาพดีกว่า

หลังจากการที่เราได้ศึกษาถึงส่วนประกอบของโครงข่ายสื่อสาร และวิธีการส่งผ่านข้อมูลในโครงข่ายแล้ว ในส่วนต่อไปเราจะมาศึกษาถึงรายละเอียดภายในโครงข่าย โครงข่ายสื่อสารนั้นประกอบไปด้วยโนดและข่ายเชื่อมโยงซึ่งเชื่อมต่อระหว่างโนด โนดทำหน้าที่ในการส่งผ่านและเปลี่ยนเส้นทางของข้อมูลที่เดินทางผ่านโนด อุปกรณ์สำคัญที่ทำหน้าที่นี้ก็คือ สวิตช์และเราเตอร์ สวิตช์และเราเตอร์จะต้องทำการส่งผ่านข้อมูลให้เร็วที่สุดและมากที่สุด เพื่อไม่ให้ข้อมูลเกิดการติดขัดเมื่อเดินทางผ่านโนด การออกแบบสวิตช์หรือเราเตอร์เพื่อให้มีสมรรถนะสูงสุดในการสวิตช์ข้อมูลจึงเป็นสิ่งสำคัญและจำเป็น ซึ่งในหัวข้อต่อไปเราจะได้ทำการศึกษาหลักการของการสวิตช์แพ็กเกต และสถาปัตยกรรมสวิตช์แพ็กเกตประเภทต่าง ๆ พอสังเขป

2.2 หลักการของการสวิตช์

2.2.1 การติดขัดภายใน

ในขณะที่เซลล์เดินทางผ่านโครงสร้างการสวิตช์ เป็นไปได้ว่าอาจจะเกิดปัญหามีเซลล์มากกว่าหนึ่งเซลล์จะเดินทางไปในเส้นทางเดียวกันในเวลาเดียวกัน ซึ่งเหตุการณ์นี้จะเกิดขึ้นในกรณีที่เป็นสวิตช์แบบแบ่งเวลา เนื่องจากการส่งผ่านเซลล์ระหว่างอินพุตกับเอาต์พุตคู่ใด ๆ อาจมีการใช้เส้นทางร่วมกันบางส่วนหรือทั้งหมด เราเรียกเหตุการณ์เช่นนี้ว่า การติดขัดภายใน (internal blocking) และเรียกสวิตช์ที่ประสบกับเหตุการณ์นี้ว่าสวิตช์ที่มีการติดขัด (blocking switch) สวิตช์ที่ไม่มีการติดขัดภายในจะถูกเรียกว่าสวิตช์ที่ไม่มีการติดขัด (nonblocking switch) เราสามารถแก้ปัญหาการติดขัดภายในสวิตช์ได้ โดยการวางบัฟเฟอร์ที่จุดที่จะเกิดการชนกันภายในสวิตช์ แต่การแก้ปัญหาด้วยวิธีนี้ จะทำให้สมรรถนะของสวิตช์ลดลงเนื่องจากสวิตช์จะมีการประวิงเวลาเพิ่มมากขึ้นและค่าทรูพุตของสวิตช์ลดลง

2.2.2 การติดขัดภายนอก

การติดขัดภายนอก (external blocking หรือ output conflict) เป็นเหตุการณ์ที่มีเซลล์ที่อินพุตมากกว่าหนึ่งอินพุต ต้องการจะไปยังเอาต์พุตเดียวกันในไทม์สล็อตเดียวกัน เนื่องจากแต่ละเอาต์พุตสามารถส่งออกเซลล์ได้เพียงหนึ่งเซลล์ ดังนั้น เซลล์ที่เหลืออยู่จะต้องถูกทิ้งไปหรือถูกนำไปเก็บไว้ในบัฟเฟอร์เพื่อรอการส่งออก ในสวิตช์แบบรอคิวที่เอาต์พุตนั้นจะมีการวางบัฟเฟอร์ไว้ที่พอร์ตเอาต์พุตเพื่อเก็บเซลล์ที่ยังไม่ได้ส่งออก เพื่อส่งออกจากสวิตช์ในไทม์สล็อตต่อไป

2.2.3 การติดขัดที่หัวคิว

วิธีการแก้ปัญหาการชนกันของเซลล์ที่เอาต์พุตก็คือ แทนที่จะวางบัฟเฟอร์ไว้ที่เอาต์พุต ก็เปลี่ยนเป็นการวางบัฟเฟอร์ที่อินพุตของสวิตช์แทน เมื่อเซลล์ต่าง ๆ เดินทางมาถึงที่อินพุตของสวิตช์จะต้องรอคิวอยู่ในบัฟเฟอร์ และในแต่ละไทม์สล็อตจะมีเพียงเซลล์เดียวในแต่ละอินพุตที่จะได้รับการบริการส่งผ่านสวิตช์ เราเรียกสวิตช์แบบนี้ว่าสวิตช์แบบรอคิวที่อินพุต จะเห็นว่าอาจจะมีเซลล์มากกว่าหนึ่งเซลล์ที่จะไปยังเอาต์พุตเดียวกันในไทม์สล็อตเดียวกัน แต่จะได้รับการบริการส่งผ่านสวิตช์เพียงเซลล์เดียว โดยเซลล์ที่เหลือจะต้องรออยู่ในคิวเพื่อเข้าแข่งขันในการที่จะได้รับเลือกส่งผ่านสวิตช์ในไทม์สล็อตต่อไป

อาจเป็นไปได้ที่เซลล์ที่อยู่ถัดจากหัวคิวต้องการจะไปยังเอาต์พุตที่ว่าง แต่ไม่สามารถส่งออกได้เนื่องจากเซลล์ที่หัวคิวนั้นจะไปยังเอาต์พุตที่ไม่ว่าง เหตุการณ์นี้เรียกว่า การติดขัดที่หัวคิว (Head of Line (HoL) blocking) วิธีแก้ปัญหานี้มีหลายวิธี เช่น พิจารณาเลือกเซลล์ส่งผ่านสวิตช์ลึกเข้าไปถึงเซลล์ที่อยู่ถัดจากเซลล์ที่หัวคิวเป็นจำนวน w เซลล์ หรือทำการเพิ่มความเร็วในการสวิตช์ให้สามารถส่งออกเซลล์ได้มากกว่าหนึ่งเซลล์ในแต่ละไทม์สล็อต หรือใช้การจัดการบัฟเฟอร์แบบ VOQ ซึ่งจะได้อธิบายในรายละเอียดในบทต่อไป

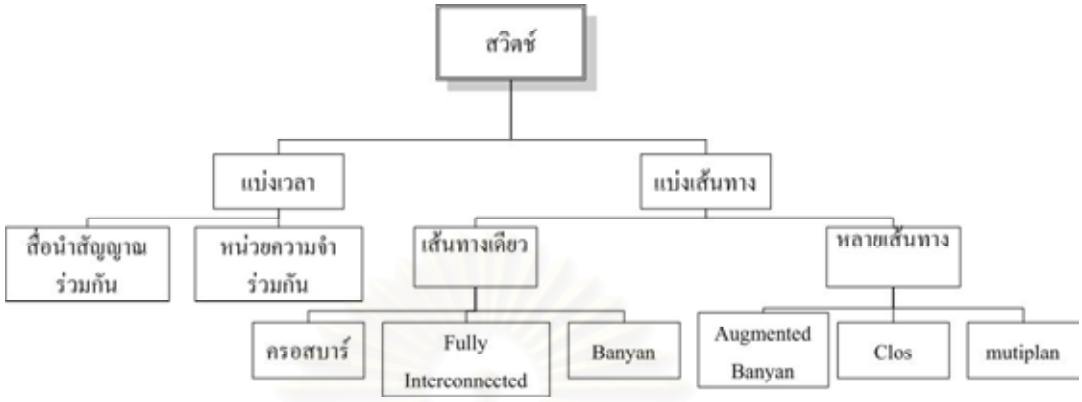
2.2.4 การทำมัลติคาสต์

เพื่อที่จะให้บริการทราฟฟิกทั้งประเภท เสียง ภาพ วิดีโอ และมัลติมีเดีย อาทิเช่นการประชุมระยะไกล สวิตช์เอทีเอ็มจึงมีความต้องการที่จะให้บริการทราฟฟิกทั้งประเภทมัลติคาสต์ (multicast) และบรอดคาสต์ (broadcast) การทำมัลติคาสต์ (multicasting) เริ่มต้นด้วยเมื่อมีเซลล์มัลติคาสต์มาถึงอินพุตของสวิตช์ โครงสร้างสวิตช์จะทำการสำเนาเซลล์มัลติคาสต์เท่าจำนวนเอาต์พุตที่เซลล์มัลติคาสต์ต้องการจะไปยังเอาต์พุตนั้น ๆ และส่งเซลล์สำเนาไปยังเอาต์พุตนั้น ๆ เทคนิคในการพิจารณาและวิธีการส่งเซลล์มัลติคาสต์ผ่านสวิตช์ยังมีอีกหลายวิธี ผู้ที่สนใจสามารถศึกษาเพิ่มเติมได้จาก [5]

2.3 การจำแนกสถาปัตยกรรมสวิตช์

โดยทั่วไปแล้วสามารถจำแนกสถาปัตยกรรมสวิตช์ (switch architecture classification) ตามเทคนิคในการสวิตช์ได้เป็น 2 กลุ่ม คือสวิตช์แบบแบ่งเวลา (time division switches) และสวิตช์แบบแบ่งเส้นทาง (space division switches) สวิตช์แบบแบ่งเวลายังแบ่งย่อยได้เป็นแบบใช้หน่วยความจำร่วมกัน (shared-memory switches) และ แบบใช้สื่อนำสัญญาณร่วมกัน (shared-medium switches) ส่วนสวิตช์แบบแบ่งเส้นทางยังแบ่งย่อยออกเป็นสวิตช์แบบเส้นทางเดียว

(single-path switches) และสวิตช์แบบหลายเส้นทาง (multiple-path switches) ทั้งสวิตช์แบบเส้นทางเดียวและแบบหลายเส้นทาง ยังแบ่งย่อยต่อไปอีก แสดงดังรูปที่ 2.2



รูปที่ 2.2 การจำแนกสถาปัตยกรรมสวิตช์ [5]

2.3.1 สวิตช์แบบแบ่งเวลา

สวิตช์แบบแบ่งเวลา ใช้โครงสร้างการสื่อสารภายในร่วมกันในการส่งข้อมูลจากพอร์ตอินพุตไปยังพอร์ตเอาต์พุตของสวิตช์ โครงสร้างการสื่อสารภายในนั้นอาจเป็น บัส (bus) , วงแหวน (ring) หรือหน่วยความจำก็ได้ ข้อเสียของเทคนิคนี้ก็คือ ความจุของสวิตช์ถูกจำกัดเนื่องมาจากโครงสร้างการสื่อสารภายใน อย่างไรก็ตาม สวิตช์ประเภทนี้ก็มีข้อดีตรงที่ว่า ทุก ๆ เซลล์ข้อมูลเดินทางผ่านโครงสร้างการสื่อสารอันเดียวกัน จึงสามารถรองรับทราฟฟิกได้ทั้งประเภทมัลติคาสต์และบรอดคาสต์

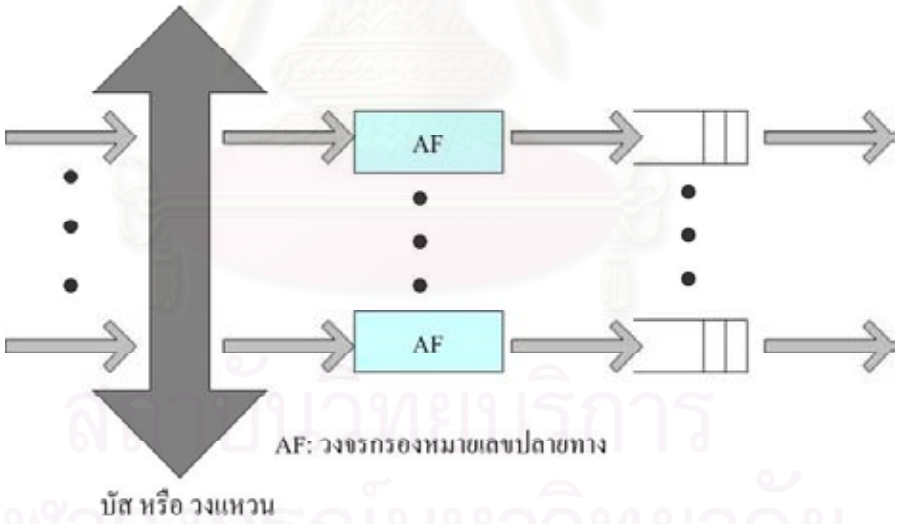
2.3.1.1 สวิตช์แบบใช้สื่อนำสัญญาณร่วมกัน

สวิตช์ที่ใช้สื่อนำสัญญาณร่วมกันนั้น เมื่อเซลล์ข้อมูลเดินทางมาถึงพอร์ตอินพุตของสวิตช์จะถูกมัลติเพลกซ์แบบแบ่งเวลา ส่งต่อไปยังสื่อนำสัญญาณความเร็วสูงซึ่งอาจจะเป็น บัส หรือวงแหวน ที่มีแบนด์วิดท์เป็นจำนวน N เท่าของความเร็วแพ็กเกต (line rate) ค่าทรูพุตของสวิตช์ประเภทนี้เป็นตัวกำหนดความจุของสวิตช์ ดังรูปที่ 2.3 จะเห็นว่าแต่ละอินพุตต่อเชื่อมกับสื่อนำสัญญาณความเร็วสูงโดยผ่านทางอินเทอร์เฟส (interface) ซึ่งประกอบไปด้วยวงจรกรองหมายเลขปลายทาง (AF: address filter) และบัฟเฟอร์ของแต่ละเอาต์พุต วงจรกรองหมายเลขปลายทางจะทำการตรวจสอบเฮดเดอร์ของแต่ละเซลล์ที่นำเข้ามา และยอมรับเฉพาะเซลล์ที่จะไปยังเอาต์พุตของตนเอง หรือมีลักษณะการทำงานแบบไม่รวมศูนย์ (decentralized) ซึ่งเป็นข้อดีที่ว่าแต่ละพอร์ตเอาต์พุตสามารถทำงานได้โดยเป็นอิสระต่อกัน จึงนำไปสร้างแยกออกจากกันได้ อย่างไรก็ตาม

ทำให้ต้องใช้อุปกรณ์ฮาร์ดแวร์และบัฟเฟอร์มากขึ้น เนื่องมาจากการแยกอินเทอร์เฟสของแต่ละเอาต์พุต

ในแต่ละไทม์สล็อตจะถูกแบ่งออกเป็น N ไทม์สล็อตย่อย ในระหว่างแต่ละไทม์สล็อตย่อยเซลล์แต่ละอินพุตจะถูกแพร่กระจายออกไปยังทุก ๆ พอร์ตเอาต์พุต จึงเป็นไปได้โดยง่ายที่เราจะทำมัลติคาสต์ แต่ละเอาต์พุตจะทำการตรวจสอบเฮดเดอร์ของแต่ละเซลล์โดยวงจรกรองหมายเลขปลายทาง และตัดสินใจเลือกเซลล์ที่จะทำการเก็บไว้ในบัฟเฟอร์ของเอาต์พุตนั้นเพื่อรอการส่งออก ข้อเสียของสวิตช์ประเภทนี้คือ ขนาดของสวิตช์ถูกจำกัดโดยความเร็วของหน่วยความจำโดยเฉพาะอย่างยิ่งในกรณีที่ทุก ๆ เซลล์จะไปยังเอาต์พุตเดียวกันซึ่งหน่วยความจำไม่อาจจะเก็บเซลล์ไว้ได้ทั้งหมดในแต่ละไทม์สล็อต หรือถูกจำกัดด้วยความเร็วของแพ็กเกตที่สูงมากเกินไป

นอกจากนี้ เนื่องจากไม่มีการใช้หน่วยความจำร่วมกันของแต่ละบัฟเฟอร์ที่พอร์ตเอาต์พุตเมื่อเกิดความคับคั่งขึ้นจากการที่ทราฟฟิกมีค่าสูง บัฟเฟอร์บางบัฟเฟอร์อาจจะเต็มและจะเริ่มทำการกำจัดเซลล์ที่มาถึงบัฟเฟอร์นั้น ในขณะเดียวกันบัฟเฟอร์อื่นอาจจะยังว่างอยู่แต่ไม่สามารถนำมาใช้กับพอร์ตที่เกิดความคับคั่งได้ จึงได้มีแก้ปัญหาโดยการสร้างสวิตช์แบบที่จะได้อธิบายต่อไปนี้ซึ่งให้ประโยชน์ในการใช้งานบัฟเฟอร์ได้ดีกว่า

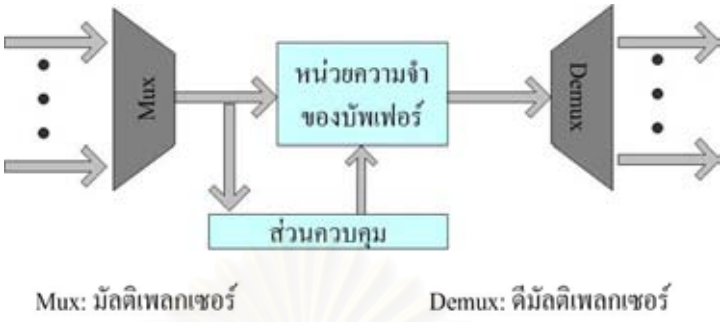


รูปที่ 2.3 สวิตช์แบบใช้สื่อนำสัญญาณร่วมกัน [5]

2.3.1.2 สวิตช์แบบใช้หน่วยความจำร่วมกัน

สวิตช์แบบที่ใช้หน่วยความจำร่วมกัน ตัวอย่างเช่นสวิตช์แบบรอคิวที่เอาต์พุต แสดงดังรูปที่ 2.4 เซลล์ที่มาถึงอินพุตของสวิตช์จะถูกมัลติเพลกซ์แบบแบ่งเวลาแล้วถูกเขียนลงในหน่วยความจำที่ใช้ร่วมกันตามลำดับ (sequentially) และถูกอ่านออกจากหน่วยความจำของสวิตช์โดยสอดคล้องกับ

จุดหมายปลายทางของเซลล์นั้น ๆ โดยให้เป็นหน้าที่ของมอดูลควบคุม (control module) เซลล์ที่เรียงกันมาเป็นลำดับนั้นจะถูกดีมัลติเพลกซ์เพื่อส่งออกจากพอร์ตเอาต์พุตต่าง ๆ ของสวิตช์



รูปที่ 2.4 สวิตช์แบบใช้หน่วยความจำร่วมกัน [5]

ไม่ว่าจะเป็นการใช้สื่อนำสัญญาณกลางร่วมกัน หรือการใช้หน่วยความจำร่วมกันของสวิตช์แบบแบ่งเวลามีข้อเสียตรงที่ว่าความเร็วในการเข้าถึงหน่วยความจำจำกัดขนาดของสวิตช์ นอกจากนี้การควบคุมการใช้หน่วยความจำร่วมกัน ทำให้สวิตช์แบบใช้หน่วยความจำร่วมกันมีความซับซ้อนมาก แต่เนื่องจากมีการใช้ประโยชน์จากบัฟเฟอร์ได้ดีกว่าจึงทำให้สวิตช์แบบใช้หน่วยความจำร่วมกันเป็นที่นิยมมากกว่าสวิตช์แบบที่ใช้สื่อนำสัญญาณร่วมกัน

2.3.2 สวิตช์แบบแบ่งเส้นทาง

สวิตช์แบบแบ่งเวลา มีโครงสร้างการสื่อสารภายในถูกใช้ร่วมกันทุก ๆ พอร์ตอินพุตและเอาต์พุต ในขณะที่สวิตช์แบบแบ่งเส้นทางนั้นเส้นทางต่าง ๆ ภายในสวิตช์จะถูกนำมาใช้ระหว่างพอร์ตอินพุตและเอาต์พุต ซึ่งเส้นทางเหล่านี้อาจจะเกิดขึ้นพร้อม ๆ กัน ดังนั้น จึงเป็นไปได้ที่จะมีเซลล์มากกว่าหนึ่งเซลล์ส่งผ่านสวิตช์ในเวลาเดียวกันได้ ความจุรวมของสวิตช์นั้นเป็นผลรวมของแบนด์วิดท์ของแต่ละเส้นทาง และจำนวนเส้นทางที่สามารถส่งผ่านสวิตช์ในขณะเดียวกันได้ ดังนั้นในทางทฤษฎีถือว่ามีค่าไม่จำกัด แต่อย่างไรก็ตามในทางปฏิบัตินั้นถูกจำกัดโดยการจำกัดทางกายภาพในการนำไปสร้างเป็นสวิตช์ เช่น จำนวนขาของอุปกรณ์ จำนวนการเชื่อมต่อ และการซิงโครไนซ์

สวิตช์แบบแบ่งเส้นทาง ยังแบ่งย่อยออกไปตามจำนวนเส้นทางระหว่างอินพุตและเอาต์พุต โดยสวิตช์แบบเส้นทางเดียวจะมีเพียงเส้นทางเดียว ในการส่งผ่านเซลล์จากอินพุตไปยังเอาต์พุตคู่ใด ๆ ในขณะที่สวิตช์แบบหลายเส้นทางจะมีเส้นทางมากกว่าหนึ่งเส้นทาง ซึ่งจะเห็นว่าสวิตช์แบบเส้นทางเดียวนั้นการจัดสรรเส้นทางเป็นไปได้ง่ายกว่า แต่สวิตช์แบบหลายเส้นทางก็มีข้อดีตรงที่ว่าทนทานต่อความผิดพร่อง (fault tolerance) ได้ดีกว่า

2.3.2.1 สวิตช์แบบเส้นทางเดียว

สวิตช์แบบเส้นทางเดียวนั้น ยังจำแนกออกเป็น สวิตช์แบบคลอสบาร์ (crossbar-based switches) สวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด (fully interconnected switches) และสวิตช์แบบบานยาน (Banyan-based switches) โดยมีรายละเอียด ดังต่อไปนี้

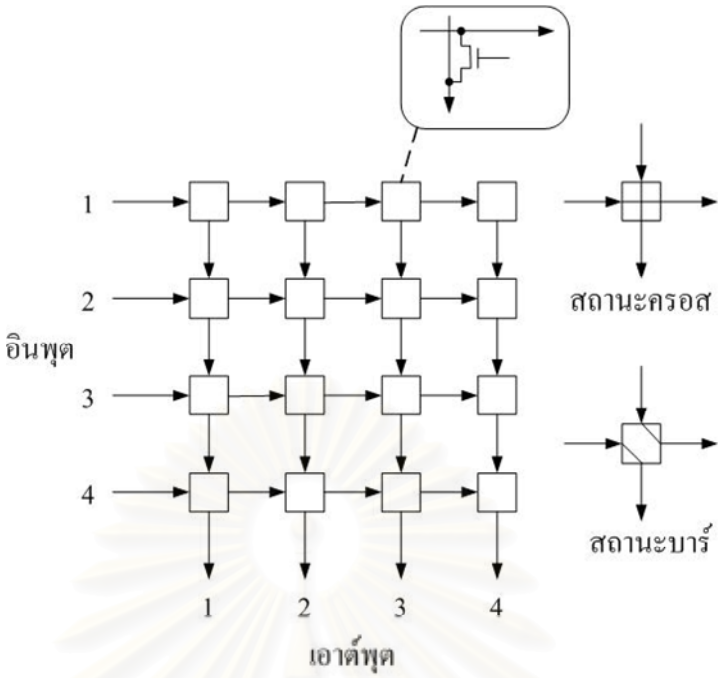
(ก) สวิตช์แบบครอสบาร์

สวิตช์แบบครอสบาร์มีลักษณะดังรูปที่ 2.5 ซึ่งประกอบไปด้วยจุดเชื่อมต่อ (cross point) จำนวน NxN จุด แต่ละจุดสำหรับเชื่อมต่อระหว่างอินพุตกับเอาต์พุตแต่ละคู่ ทั้งนี้จุดเชื่อมต่อมีความเป็นไปได้เพียงสองสถานะคือ สถานะครอส (cross (default) state) และสถานะบาร์ (bar state) การเชื่อมต่อระหว่างอินพุตที่ i และเอาต์พุตที่ j ทำได้โดยการเปลี่ยนสถานะจุดเชื่อมต่อแถวที่ i กับคอลัมน์ที่ j ไปเป็นสถานะบาร์ในขณะที่จุดเชื่อมต่ออื่น ๆ ที่อยู่ในแถวและคอลัมน์เดียวกันยังคงมีสถานะครอส จุดเชื่อมต่อสถานะบาร์จะทำการส่งเซลล์ผ่านจุดเชื่อมต่อ

ในการควบคุมสถานะของจุดเชื่อมต่อนั้น เป็นหน้าที่ของเซลล์ที่มาถึงที่พอร์ตอินพุต ที่จะทำการปิดจุดเชื่อมต่อเป็นสถานะบาร์ ให้สอดคล้องกับหมายเลขปลายทางของเซลล์ และเป็นอิสระจากกันเพื่อที่จะไปยังเอาต์พุตที่ต้องการ โดยไม่จำเป็นที่จะต้องทราบข้อมูลที่เกี่ยวข้องกับการส่งของเซลล์อื่น ๆ เราเรียกคุณสมบัตินี้ว่า การเลือกเส้นทางเอง (self–routing) ทำให้ความซับซ้อนของโครงสร้างสวิตช์ลดลง เนื่องจากฟังก์ชันการควบคุมถูกกระจายไปทุก ๆ จุดเชื่อมต่อ

สวิตช์แบบครอสบาร์มีคุณสมบัติที่น่าสนใจหลายประการคือ เป็นสวิตช์ที่ไม่มีการติดขัดภายใน นำไปสร้างเป็นสวิตช์ได้ง่าย และสามารถขยายขนาดเป็นมอดูล (modular) ได้ อย่างไรก็ตามค่าความซับซ้อนของสวิตช์ซึ่งนิยามจากจุดเชื่อมต่อที่เพิ่มขึ้นตามค่า N^2 จะทำให้เกิดปัญหาคอขวดในการสวิตช์เมื่อสวิตช์มีขนาดใหญ่ขึ้น เนื่องจากตัวตัดสินใจในการเลือกเซลล์ส่งผ่านโครงสร้างการสวิตช์ไม่สามารถทำงานได้ทันในแต่ละไทม์สล็อต

สวิตช์แบบครอสบาร์นั้นสามารถจัดวางบัฟเฟอร์ได้ 3 รูปแบบคือ วางบัฟเฟอร์ไว้ที่จุดเชื่อมต่อ วางบัฟเฟอร์ไว้ที่เอาต์พุต และวางบัฟเฟอร์ไว้ที่อินพุตของสวิตช์

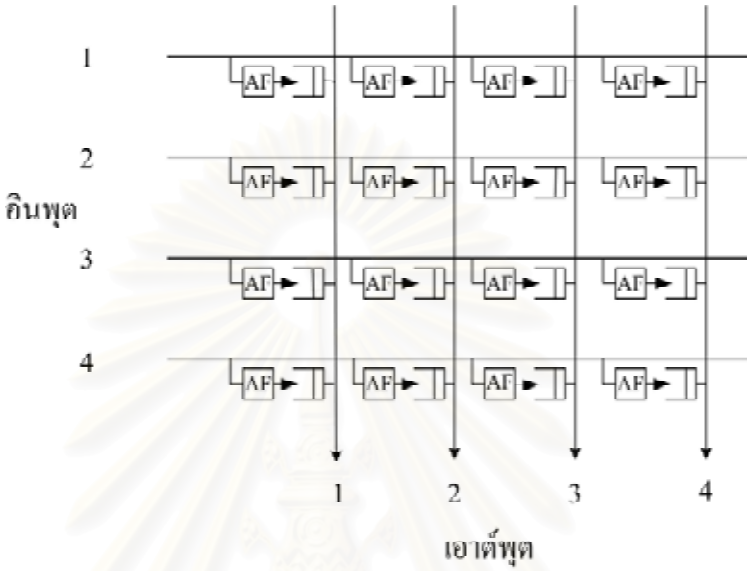


รูปที่ 2.5 โครงสร้างสวิตช์แบบครอสบาร์ [5]

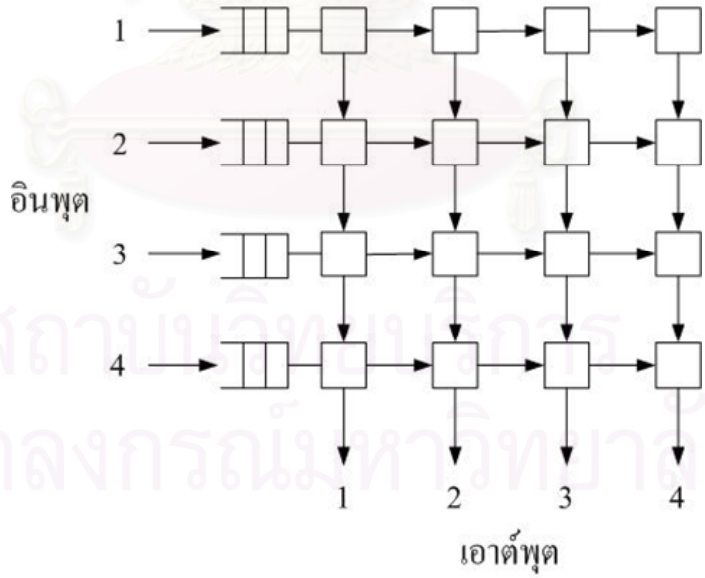
ตามรูปที่ 2.6 (ก) เป็นการวางบัฟเฟอร์ไว้ที่ทุกจุดเชื่อมต่อของสวิตช์โดยที่ในแต่ละจุดเชื่อมต่อจะประกอบไปด้วยวงจรกรองหมายเลขปลายทางและบัฟเฟอร์ วงจรกรองหมายเลขปลายทางจะยอมรับเฉพาะเซลล์ที่จะไปยังเอาต์พุตของตน โดยการทำการเปรียบเทียบหมายเลขปลายทางของแพ็กเกตว่าสอดคล้องกับหมายเลขของพอร์ตเอาต์พุตหรือไม่ ถ้าสอดคล้องกันก็จะทำการเก็บเซลล์ไว้ในบัฟเฟอร์ ในแต่ละไทม์สล็อตเซลล์ที่รอคิวอยู่ในบัฟเฟอร์ในคอลัมน์เดียวกันจะได้รับเลือกเพื่อส่งไปยังเอาต์พุตเพียงหนึ่งเซลล์ สวิตช์จึงมีคุณสมบัติที่เรียกว่า เวิร์กคอนเซิร์ฟวิง (work conserving) จึงไม่ประสบกับปัญหาการติดขัดที่หัวคิวเหมือนกับสวิตช์ที่มีการวางบัฟเฟอร์ที่อินพุต ถ้ามองอีกแง่มุมหนึ่ง การจัดวางบัฟเฟอร์วิธีนี้มีลักษณะคล้ายกันกับสวิตช์ที่มีการวางบัฟเฟอร์ที่เอาต์พุต ต่างกันแต่เพียงว่ามีการกระจายบัฟเฟอร์ในแต่ละเอาต์พุตออกไปเป็นจำนวน N บัฟเฟอร์ โดยไม่มีการใช้หน่วยความจำของบัฟเฟอร์ร่วมกัน ทำให้โครงสร้างสวิตช์แบบนี้หน่วยความจำของบัฟเฟอร์จะต้องใช้พื้นที่บนชิปมาก สิ่งนี้จึงไปจำกัดขนาดของสวิตช์ทำให้ไม่สามารถนำไปสร้างเป็นสวิตช์ที่มีขนาดใหญ่ได้บนชิปเดียว ตัวอย่างของสวิตช์ประเภทนี้ได้แก่ สวิตช์บัสเมตริกซ์ (bus matrix (BMX) switch)

ตามรูปที่ 2.6 (ข) แสดงการวางบัฟเฟอร์ไว้ที่พอร์ตอินพุตของสวิตช์ เป็นการแยกบัฟเฟอร์ออกจากจุดเชื่อมต่อเพื่อลดความซับซ้อนของวงจรการสวิตช์ เมื่อมีเซลล์มาถึงอินพุตของสวิตช์ จะถูกนำไปเก็บไว้ในบัฟเฟอร์โดยทันทีเพื่อรอการส่งผ่านโครงสร้างการสวิตช์ โครงสร้างของสวิตช์ประเภทนี้มีอยู่ด้วยกัน 2 รูปแบบคือ แบบกระจายหน้าที่การสวิตช์ไปทุก ๆ จุดเชื่อมต่อ โดยที่เมื่อ

เซลล์เดินทางมาถึงจุดเชื่อมต่อที่มีการจองไว้แล้วโดยเซลล์อื่นก่อนหน้านั้น หรือเซลล์เกิดการแย่งชิงจุดเชื่อมต่อกับเซลล์อื่นและเป็นฝ่ายแพ้ จุดเชื่อมต่อจะส่งสัญญาณไปที่พอร์ตอินพุตเพื่อห้ามมิให้ส่งออกและเก็บเซลล์นั้นไว้ในบัฟเฟอร์ก่อนเพื่อรอการพิจารณาใหม่ในไทม์สล็อตต่อไป หรือแบบรวมศูนย์การตัดสินใจในการเลือกส่งเซลล์ผ่านจุดเชื่อมต่อไว้ที่พอร์ตเอาต์พุตแต่ละพอร์ต



(ก) การวางบัฟเฟอร์ที่ทุกจุดเชื่อมต่อ



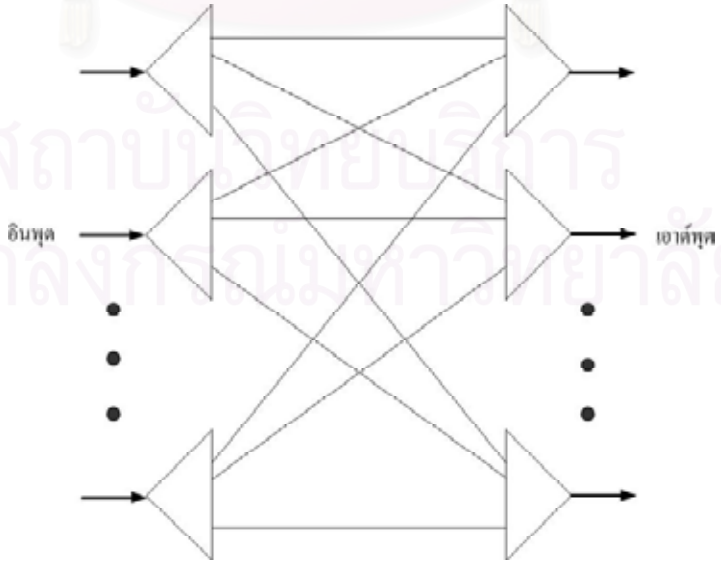
(ข) การวางบัฟเฟอร์ที่พอร์ตอินพุต

รูปที่ 2.6 สวิตช์แบบครอสบาร์กับรูปแบบการวางบัฟเฟอร์ [5]

(ข) สวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด

สวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด (fully interconnected switches) เป็นสวิตช์ที่มีการเชื่อมต่อถึงกันหมดระหว่างพอร์ตอินพุตเอาต์พุตทุก ๆ พอร์ต หมายความว่าทุก ๆ พอร์ตอินพุต จะมีเส้นทาง N เส้นทางกระจายไปยังทุก ๆ พอร์ตเอาต์พุต ดังรูปที่ 2.7 ซึ่งทำให้แต่ละเอาต์พุตจะต้องมีบัฟเฟอร์จำนวน N บัฟเฟอร์ ทำให้บัฟเฟอร์รวมมีจำนวนถึง N^2 บัฟเฟอร์ จึงมีลักษณะคล้ายกันกับสวิตช์ที่มีการวางบัฟเฟอร์ไว้ที่ทุก ๆ จุดเชื่อมต่อ สมรรถนะและความซับซ้อนของสวิตช์จึงมีลักษณะสมมูลกัน

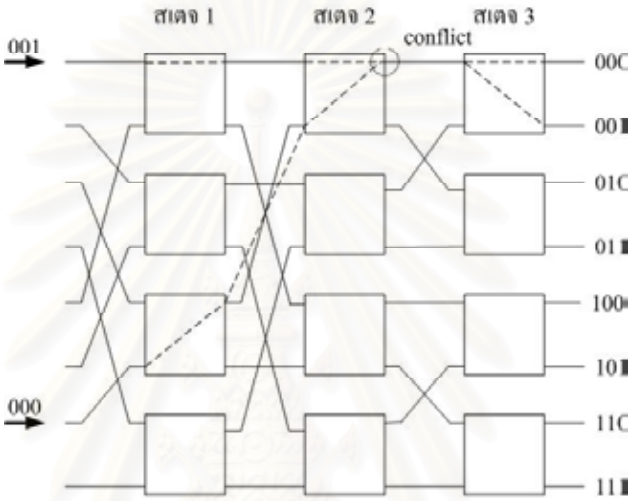
สวิตช์ประเภทนี้ จึงมีลักษณะการทำงานเช่นเดียวกันกับสวิตช์แบบใช้สื่อนำสัญญาณร่วมกัน กล่าวคือเซลล์จากอินพุตต่าง ๆ จะถูกกระจายไปยังทุก ๆ เอาต์พุต ดังนั้นอาจจะมีเซลล์มากกว่าหนึ่งเซลล์จากอินพุตต่าง ๆ สามารถที่จะไปยังเอาต์พุตเดียวกันในเวลาเดียวกันได้ จึงมีความจำเป็นที่จะต้องมีบัฟเฟอร์ที่พอร์ตเอาต์พุต เพื่อเก็บเซลล์ที่ไปยังเอาต์พุตนั้นแต่ยังไม่ได้รับการส่งออกจากสวิตช์ อย่างไรก็ตามสวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด ก็มีความแตกต่างจากสวิตช์แบบใช้สื่อนำสัญญาณร่วมกันตรงที่ว่า มีความจำเป็นที่จะต้องเพิ่มความเร็วในการสวิตช์ เนื่องจากมีการใช้การประมวลผลสัญญาณแบบขนานกระจายไปตามเส้นทาง N^2 เส้นทาง ซึ่งเป็นข้อเสียของสวิตช์ประเภทนี้ ข้อดีของสวิตช์ประเภทนี้ก็คือมีโครงสร้างแบบไม่ติดขัด และนำไปสร้างเป็นสวิตช์ได้ง่ายเช่นเดียวกับสวิตช์แบบครอสบาร์ ตัวอย่างของสวิตช์ประเภทนี้ได้แก่ สวิตช์น็อกเอาต์ (Knockout switch)



รูปที่ 2.7 สวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมด [5]

(ค) สวิตช์แบบบานยาน

สวิตช์แบบบานยาน (Banyan) เป็นสวิตช์ที่มีคุณสมบัติการเลือกเส้นทางเอง โครงสร้างสวิตช์ประกอบไปด้วย องค์ประกอบการสวิตช์ (switching element) ขนาด 2x2 ที่มีเพียงหนึ่งเส้นทางระหว่างอินพุตเอาต์พุตคู่ใด ๆ โครงสร้างแบบเดลตา (Delta). โอเมกา (Omega) และบานยานนั้น มีโทโพโลยีรูปร่างสัณฐานเหมือนกันจึงจัดอยู่ในตระกูลของสวิตช์แบบบานยาน ซึ่งให้สมรรถนะของสวิตช์ที่สมมูลกับสวิตช์แบบบานยาน แสดงดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างสวิตช์แบบบานยาน ขนาด 8x8 [6]

สวิตช์แบบบานยาน มีข้อดีหลายประการคือ ประการแรก ค่าความซับซ้อนของสวิตช์มีค่า O(NlogN) ซึ่งน้อยกว่าสวิตช์แบบเชื่อมต่อระหว่างกันทั้งหมดและแบบครอสบาร์ซึ่งมีค่าความซับซ้อน $O(N^2)$ จึงมีความเหมาะสมมากกว่าในการนำไปสร้างเป็นสวิตช์ที่มีขนาดใหญ่ ประการที่สอง มีคุณสมบัติการกำหนดเส้นทางเอง จึงไม่ต้องมีกลไกในการควบคุมการจัดสรรเส้นทางของเซลล์เพราะว่าเซลล์สามารถควบคุมการเลือกเส้นทางภายในสวิตช์ได้ด้วยตนเอง ประการที่สาม องค์ประกอบในการสวิตช์มีโครงสร้างต่อขนานกัน ทำให้แต่ละเซลล์มีเส้นทางให้เลือกหลายเส้นทาง สวิตช์จึงสามารถประมวลผลพร้อมกันได้ (ประมวลผลแบบขนาน) และประการสุดท้าย เนื่องจากมีโครงสร้างเป็นแบบมอดูลและซ้ำ ๆ กัน ทำให้การเพิ่มขนาดของสวิตช์สามารถทำได้โดยใช้องค์ประกอบการสวิตช์มูลฐานต่อซ้ำ ๆ กัน โดยไม่ต้องทำการปรับปรุงแก้ไขโครงสร้าง ซึ่งเหมาะสำหรับนำไปสร้างเป็นสวิตช์ด้วยเทคโนโลยี VLSI

ข้อเสียของสวิตช์ประเภทนี้ก็คือ เป็นสวิตช์ที่มีการติดขัดภายใน ทำให้สมรรถนะของสวิตช์ลดลงอย่างรวดเร็วเมื่อขนาดของสวิตช์เพิ่มมากขึ้น เราสามารถปรับปรุงสมรรถนะให้ดีขึ้นโดยให้องค์ประกอบการสวิตช์มีขนาด MxM (M > 2) แทนที่จะเป็น 2x2 สวิตช์ที่ใช้วิธีนี้ได้แก่ สวิตช์เดลตา

2.3.2.2 สวิตช์แบบหลายเส้นทาง

สำหรับสวิตช์แบบหลายเส้นทาง สามารถจำแนกออกเป็น สวิตช์แบบ Augmented Banyan สวิตช์แบบคลอส (Clos) และสวิตช์แบบมัลติเพลน (Multiplan)

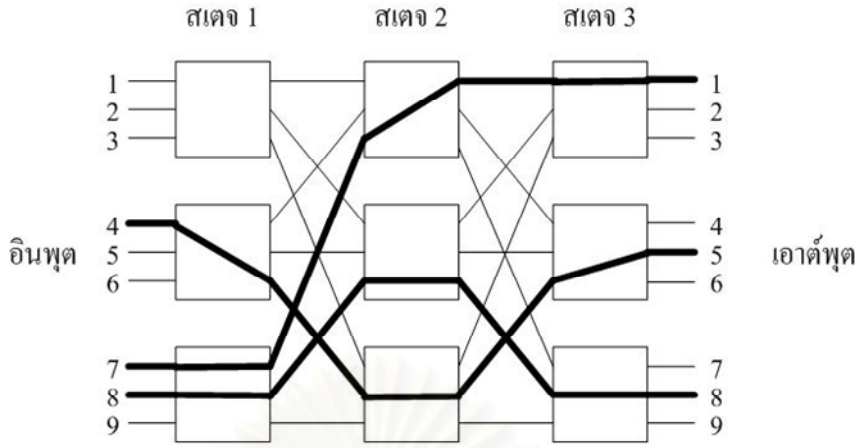
(ก) สวิตช์แบบ Augmented Banyan

โดยปกติแล้วสวิตช์แบบบานยานขนาด NxN นั้นเซลล์ต่าง ๆ จะผ่านองค์ประกอบการสวิตช์จำนวน log N สเตจ (stage) ก่อนจะไปถึงเอาต์พุตที่ต้องการ แต่สวิตช์แบบ Augment Banyan จะมีจำนวนสเตจมากกว่าสวิตช์แบบบานยาน ในสวิตช์แบบบานยานนั้นเซลล์อาจประสบปัญหาการชนกันภายในสวิตช์เนื่องจากเดินทางในเส้นทางที่ไม่เหมาะสม จึงไม่สามารถรับประกันได้ว่าเซลล์จะไปถึงเอาต์พุตที่ต้องการ แต่ในสวิตช์แบบ Augmented Banyan เซลล์ที่เริ่มเดินทางจากเส้นทางที่ไม่เหมาะสม มีทางเลือกมากกว่าที่จะเปลี่ยนแปลงเส้นทางเพื่อไปยังเอาต์พุตที่ต้องการโดยผ่านสเตจที่เพิ่มขึ้นมา แต่ถ้าเซลล์นั้นไม่สามารถไปถึงเอาต์พุตที่ต้องการได้หลังจากผ่านสเตจสุดท้ายก็จะถูกทิ้งไป

ประโยชน์ของการเพิ่มสเตจของสวิตช์ก็คือ อัตราการสูญเสียเซลล์ลดลง ทำให้สมรรถนะของสวิตช์ดีขึ้น ข้อเสียของสวิตช์ประเภทนี้ก็คือ รูปแบบการจัดสรรเส้นทางมีลักษณะซับซ้อน เนื่องจากจะต้องถูกตรวจสอบทุก ๆ สเตจที่เพิ่มขึ้นมาเพื่อจะกำหนดให้ได้ว่าเซลล์นั้นจะไปถึงพอร์ตเอาต์พุตหรือไม่ *ถ้าใช่* เซลล์นั้นจะถูกส่งไปมอดูลอินเทอร์เฟสของพอร์ตเอาต์พุต ถ้า*ไม่ใช่* เซลล์จะต้องเดินทางผ่านสเตจต่อไปเพื่อทำการทดสอบอีกครั้ง นอกจากนี้จำนวนสเตจที่เพิ่มขึ้นมาจะต้องมีขนาดใหญ่เพียงพอ การเพิ่มสเตจเข้าไปในสวิตช์แบบบานยานเป็นสาเหตุทำให้ฮาร์ดแวร์มีความซับซ้อนเพิ่มมากขึ้น ตัวอย่างของสวิตช์ประเภทนี้ ได้แก่ สวิตช์ Tandem Banyan และ สวิตช์ Dual Shuffle Exchange

(ข) สวิตช์แบบคลอส

โครงสร้างสวิตช์แบบคลอสสามสเตจนั้น ประกอบไปด้วยมอดูลสวิตช์ดังนี้ สเตจแรกอินพุตของสวิตช์จำนวน N อินพุต จะถูกแบ่งออกเป็น *r* กลุ่ม แต่ละกลุ่มมี *n* อินพุต ซึ่งจะเป็นอินพุตของมอดูลสวิตช์ของสเตจแรกซึ่งมีเอาต์พุตจำนวน *m* เอาต์พุต ถูกเชื่อมต่อกับมอดูลสวิตช์สเตจที่สองจำนวน *m* มอดูล เช่นเดียวกัน แต่ละมอดูลสวิตช์ของสเตจที่สอง จะมีเอาต์พุต *t* เอาต์พุตเชื่อมต่อกับมอดูลสวิตช์สเตจที่สาม จำนวน *t* มอดูล ซึ่งจะเห็นว่าเอาต์พุตของสวิตช์จำนวน N เอาต์พุตนั้น จะมี *t* กลุ่ม แต่ละกลุ่มมี *s* เอาต์พุต



รูปที่ 2.9 ตัวอย่างการติดขัดภายในสวิตช์แบบคลอสสามสเตจ [5]

สวิตช์แบบคลอสสามสเตจอาจมีการติดขัดภายใน แสดงดังรูปที่ 2.9 ซึ่งจะเห็นว่าอินพุตพอร์ตที่ 9 ไม่สามารถเชื่อมต่อกับเอาต์พุตพอร์ตที่ 4 หรือ 6 ได้แม้ว่าเอาต์พุตนั้นจะว่างอยู่ แต่ถ้าเราเพิ่มค่าของ m (หรือจำนวนมอดูลสวิตช์สเตจที่สอง) ความน่าจะเป็นในการติดขัดก็จะลดลง เราจะหาค่าของ m ที่ทำให้สวิตช์สามสเตจไม่มีการติดขัด ตามรูปที่ 2.10 จะเห็นว่าเป็นกรณีที่ความต้องการในการเชื่อมต่อระหว่างอินพุตและเอาต์พุตที่ก่อให้เกิดปัญหามากที่สุด เส้นทางการเชื่อมต่อระหว่างสวิตช์สเตจแรกกับสเตจที่สองนั้นไม่ว่าง เป็นจำนวน $n-1$ เส้นทาง เนื่องจากไว้รองรับความต้องการของพอร์ตอินพุตจำนวน $n-1$ พอร์ต ในขณะเดียวกันเส้นทางการเชื่อมต่อระหว่างสเตจที่สองกับสเตจที่สาม จำนวน $n-1$ เส้นทางนั้นไม่ว่าง เนื่องจากไว้รองรับความต้องการของพอร์ตเอาต์พุตทั้งหมด $n-1$ พอร์ต เพราะฉะนั้น จะต้องเพิ่มมอดูลสวิตช์อีกหนึ่งมอดูลเพื่อเชื่อมต่อพอร์ตอินพุตเอาต์พุตคู่สุดท้าย ดังนั้น ในกรณีที่ต้องการให้สวิตช์ไม่เกิดการติดขัดภายใน จะต้องมีสวิตช์สเตจที่สองจำนวนทั้งสิ้น [5], [6]

$$m \geq (n+1)+(n+1)+1 = 2n-1 \tag{2.1}$$

จำนวนจุดเชื่อมต่อทั้งหมดของสวิตช์สามสเตจ คือ

$$N_x = \left(\frac{2N}{n}(nm) + m\left(\frac{N}{n}\right)\left(\frac{N}{n}\right)\right) = 2Nm + m\left(\frac{N}{n}\right)^2 \tag{2.2}$$

แทนค่า $m = 2n-1$ ลงในสมการที่ 2.2 จะได้ว่า

$$N_x = 2N(2n-1) + (2n-1)\left(\frac{N}{n}\right)^2 \tag{2.3}$$

สำหรับสวิตช์ที่ไม่มีการติดขัด เมื่อกำหนดให้ n มีขนาดใหญ่มาก เราจะประมาณได้ว่า

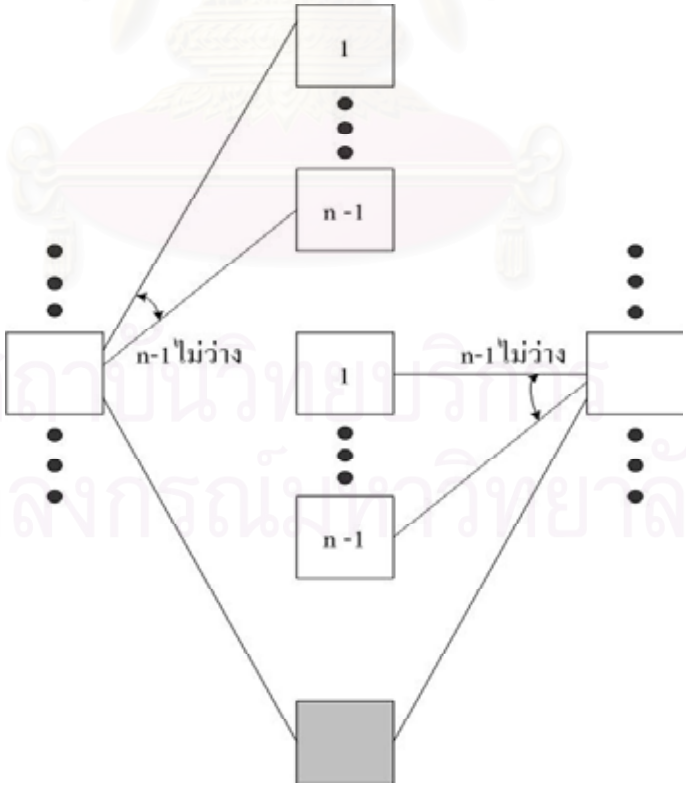
$$N_x \cong 2N(2n) + 2n\left(\frac{N}{n}\right)^2 = 4Nn + 2\left(\frac{N^2}{n}\right) \tag{2.4}$$

จากสมการที่ 2.4 จะเห็นว่าจำนวนจุดเชื่อมต่อที่ใช้ขึ้นอยู่กับค่า n เราสามารถหาค่า n ที่เหมาะสมเพื่อให้ได้จำนวนจุดเชื่อมต่อที่มีค่าน้อยที่สุดได้ โดยการหาอนุพันธ์ของสมการที่ 2.4 เทียบกับ n จะได้ค่าต่ำสุดของ n คือ

$$n \approx \left(\frac{N}{2}\right)^{\frac{1}{2}} \tag{2.5}$$

แทนสมการที่ 2.5 ลงในสมการที่ 2.4 จะได้จำนวนจุดเชื่อมต่อที่น้อยที่สุดเพื่อให้ได้สวิตช์ที่ไม่ติดขัดเลย คือ

$$N_x = 4\sqrt{2}N^{\frac{3}{2}} \Rightarrow O\left(N^{\frac{3}{2}}\right) \tag{2.6}$$

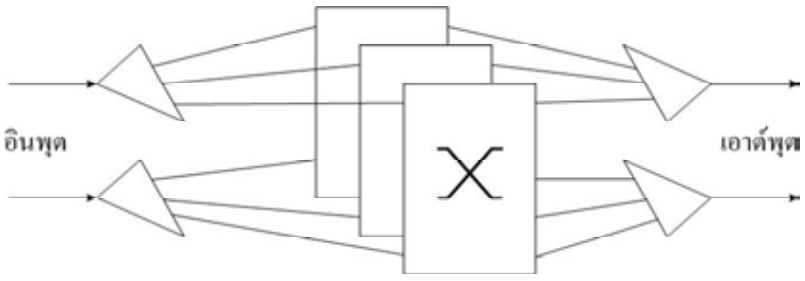


รูปที่ 2.10 แสดงเงื่อนไขการไม่ติดขัดสำหรับสวิตช์แบบคลอสสามสเตจ [5]

สวิตช์แบบคลอสสามสเตจ มีข้อดีที่ว่าสามารถลดความซับซ้อนของสวิตช์จาก $O(N^2)$ ในกรณีของสวิตช์แบบครอสบาร์เป็น $O(N^{3/2})$ และสามารถออกแบบสวิตช์ให้เป็นสวิตช์แบบไม่ติดขัดได้ นอกจากนี้ สวิตช์ยังให้ความเชื่อถือได้ดีกว่าเนื่องจากมีเส้นทางภายในเชื่อมต่อระหว่างอินพุตเอาต์พุตของสวิตช์ได้มากกว่า ข้อเสียของสวิตช์ประเภทนี้ก็คือ จะต้องมีกลไกที่ทำงานได้อย่างรวดเร็วและชาญฉลาด เพื่อจัดการเชื่อมต่อในแต่ละไทม์สล็อตให้สอดคล้องกับความต้องการของเซลล์ที่อินพุตของสวิตช์ เพื่อหลีกเลี่ยงการติดขัดภายในซึ่งอาจเกิดปัญหาคอขวดขึ้นได้เมื่อสวิตช์มีขนาดใหญ่ขึ้น ในทางปฏิบัติเป็นไปได้ยากที่จะหลีกเลี่ยงการติดขัดภายใน ถึงแม้ว่าสวิตช์จะได้รับการออกแบบให้เป็นสวิตช์แบบไม่ติดขัด ซึ่งการชนกันของเซลล์ภายในสวิตช์ทำให้ค่าทรูพุตลดลง เราสามารถแก้ปัญหานี้ได้โดย วิธีที่หนึ่ง เพิ่มจำนวนมอดูลสวิตช์ในสเตจที่สองให้มากขึ้น เพื่อเพิ่มเส้นทางภายในสำหรับส่งผ่านเซลล์ หรือวิธีที่สอง เพิ่มความเร็วในการสวิตช์ (speed up) เพื่อให้ในแต่ละไทม์สล็อตแต่ละอินพุตของสวิตช์สามารถส่งผ่านเซลล์ได้มากกว่าหนึ่งเซลล์ และวิธีสุดท้ายวางบัฟเฟอร์ไว้ที่มอดูลการสวิตช์ในสเตจที่สอง แต่จะต้องมีการจัดลำดับเซลล์กันใหม่อีกครั้งเมื่อเซลล์ไปถึงพอร์ตเอาต์พุตของสวิตช์

(ข) สวิตช์แบบมัลติเพลน

สวิตช์แบบมัลติเพลน (Multiplane) หรือสวิตช์แบบหลายระนาบ หมายถึงสวิตช์ที่มีระนาบของสวิตช์มากกว่าหนึ่งระนาบซึ่งแต่ละระนาบมีลักษณะเหมือนกัน สวิตช์แบบหลายระนาบมีวัตถุประสงค์หลักที่จะหาวิธีการเพิ่มค่าทรูพุตของระบบ โดยใช้กลไกบางอย่างในการกระจายโหลดทราฟฟิกที่เข้ามาที่อินพุตของสวิตช์ ทั้งนี้ก็เพื่อลดการชนกันของเซลล์ภายในสวิตช์ นอกจากนี้ เซลล์มากกว่าหนึ่งเซลล์สามารถส่งผ่านไปยังเอาต์พุตเดียวกันได้โดยการใช้ระนาบต่างกัน ทำให้เอาต์พุต (output line) ไม่จำเป็นต้องเพิ่มความเร็วในการทำงานให้สูงกว่าความเร็วแพ็กเกต ข้อดีอีกอย่างหนึ่งของสวิตช์แบบมัลติเพลนก็คือให้ความเชื่อถือได้เนื่องจากการเพิ่มระนาบสวิตช์นั้นทำให้ความจุในการสวิตช์ลดลงแต่จำนวนการเชื่อมต่อของสวิตช์เพิ่มมากขึ้น แต่อย่างไรก็ตาม เมื่อเซลล์ไปถึงเอาต์พุตอาจจะไม่เป็นไปตามลำดับ ตัวอย่างสวิตช์ประเภทนี้ได้แก่ สวิตช์ Parallel Banyan และ สวิตช์ Sunshine



รูปที่ 2.11 โครงสร้างสวิตช์แบบมัลติเพลน [5]

2.4 สมรรถนะของสวิตช์พื้นฐาน

ในส่วนนี้จะอธิบายสมรรถนะของสวิตช์พื้นฐาน 2 ประเภทคือ สวิตช์แบบรอคิวที่เอาต์พุต และสวิตช์แบบรอคิวที่อินพุต [6], [7]

2.4.1 สวิตช์แบบรอคิวที่เอาต์พุต

กำหนดให้เซลล์ที่มาถึงพอร์ตเอาต์พุตรออยู่ในคิวแบบ FIFO ซึ่งบัฟเฟอร์ของแต่ละเอาต์พุตนั้นแยกออกจากกัน สมมุติให้การมาถึงของเซลล์ที่แต่ละอินพุตเป็นกระบวนการ i.i.d.Bernoulli และในไทม์สล็อตใด ๆ ความน่าจะเป็นที่เซลล์จะมาถึงอินพุตแต่ละอินพุตคือ ρ และความน่าจะเป็นที่แต่ละเซลล์จะไปยังเอาต์พุตใด ๆ เท่ากับ $1/N$ (โดยที่ N คือขนาดของสวิตช์) ความสำเร็จในการส่งผ่านเซลล์นั้นเป็นอิสระต่อกัน เราจะนิยามว่า ตัวแปรสุ่ม A คือ จำนวนเซลล์ที่มาถึงเอาต์พุตในแต่ละไทม์สล็อต ดังนั้น A จึงมีความน่าจะเป็นแบบทวินาม (binomial probability) ดังสมการต่อไปนี้

$$a_k \equiv P[A=k] = \binom{N}{k}\left(\frac{\rho}{N}\right)^k\left(1-\frac{\rho}{N}\right)^{N-k} \qquad k=0,1,...,N \tag{2.7}$$

สำหรับสวิตช์ที่มีขนาดใหญ่ขึ้นนั้น เมื่อให้ $N \to \infty$ จะได้ว่าจำนวนเซลล์ที่มาถึงในแต่ละบัฟเฟอร์ในแต่ละไทม์สล็อตมีความน่าจะเป็นแบบปัวส์ซง (วิธีพิสูจน์ดู ผนวก ก) ที่มีสมการดังต่อไปนี้

$$a_k \equiv P[A=k] = \rho^k \frac{e^{-\rho}}{k!} \qquad k=0,1,...,\infty \tag{2.8}$$

โดยมีฟังก์ชันก่อกำเนิดความน่าจะเป็น (probability generating function: PGF) คือ

$$A(z) \equiv \sum_{k=0}^{N} z^k P[A=k] = \left(1-\frac{\rho}{N}+z\frac{\rho}{N}\right)^N \tag{2.9}$$

ให้ Q_m เป็นจำนวนเซลล์ที่รอคิวอยู่ในบัฟเฟอร์ของเอาต์พุตที่จุดสิ้นสุดของไทม์สล็อตที่ m และ A_m เป็นจำนวนเซลล์ที่มาถึงอินพุตของสวิตช์ในระหว่างไทม์สล็อตที่ m เราจะได้ว่า

$$Q_m = \max\left(0, Q_{m-1} + A_m - 1\right) \tag{2.10}$$

เมื่อ $Q_{m-1}=0$ และ $A_m > 0$ แล้ว จะต้องมีหนึ่งเซลล์ถูกส่งออกจากสวิตช์โดยทันทีในระหว่างไทม์สล็อตที่ m โดยไม่มีการประวิงเวลาแต่อย่างใด เราจะจำลองขนาดของ Q_m โดยใช้ลูกโซ่มาร์คอฟเวลาไม่ต่อเนื่อง (discrete time Markov chain) ดังรูปที่ 2.12 ซึ่งแสดงแผนภาพการเปลี่ยน

สถานะลูกโซ่มาร์คอฟเวลาไม่ต่อเนื่อง (discrete-time Markov chain state transition diagram) และใช้การวิเคราะห์การรอคิว (queueing analysis) เราจะได้ PGF ของขนาดของคิวที่สถานะอยู่ตัว ดังนี้

$$Q(z)=\frac{(1-\rho)(1-z)}{A(z)-z} \tag{2.11}$$

และสุดท้าย แทน $A(z)$ สมการที่ 2.9 ลงในสมการที่ 2.11 จะได้ว่า

$$Q(z)=\frac{(1-\rho)(1-z)}{\left(1-\frac{\rho}{N}+z\frac{\rho}{N}\right)^{N}-z} \tag{2.12}$$

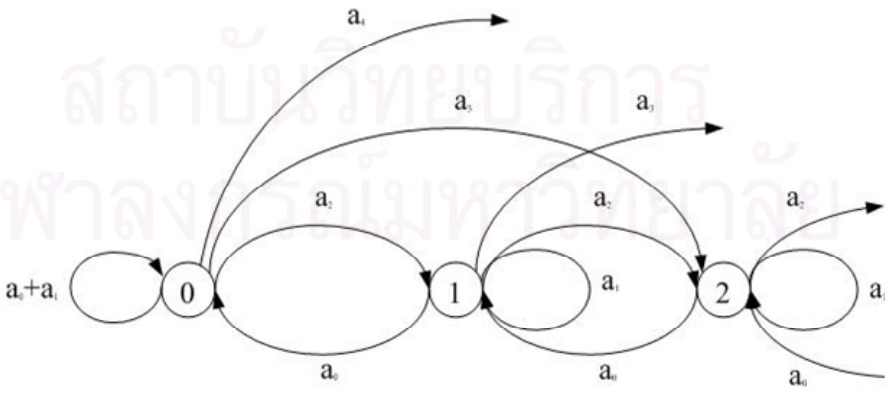
หาอนุพันธ์ของสมการที่ 2.12 เทียบกับ z และหาค่าลิมิต เมื่อ $z\to 1$ เราจะได้ค่าเฉลี่ยขนาดของคิวที่สถานะอยู่ตัว (mean steady-state queue size) หรือ $\bar{Q}$ คือ

$$\bar{Q}=\left(\frac{N-1}{N}\right)\frac{\rho^2}{2(1-\rho)}=\left(\frac{N-1}{N}\right)\bar{Q}_{M/D/1} \tag{2.13}$$

โดยที่ $\bar{Q}_{M/D/1}$ คือ ค่าเฉลี่ยขนาดของคิว (mean queue size) ของการรอคิวแบบ M/D/1

และเมื่อเราให้ขนาดของสวิตช์ $N\to\infty$ เราจะได้ว่า

$$\bar{Q}\cong\bar{Q}_{M/D/1}=\frac{\rho^2}{2(1-\rho)} \tag{2.14}$$



รูปที่ 2.12 แผนภาพการเปลี่ยนสถานะลูกโซ่มาร์คอฟเวลาไม่ต่อเนื่อง [7]

2.4.2 สวิตช์แบบรอคิวที่อินพุต

เช่นเดียวกับการรอคิวที่เอาต์พุต สมมุติให้การมาถึงของเซลล์ที่แต่ละอินพุตเป็นกระบวนการ i.i.d.Bernoulli และในไทม์สล็อตใด ๆ ความน่าจะเป็นที่เซลล์จะมาถึงอินพุตแต่ละอินพุตคือ ρ และความน่าจะเป็นที่แต่ละเซลล์จะไปยังเอาต์พุตใด ๆ เท่ากับ $1/N$ ตัวควบคุมการสวิตช์ (switch controller) จะทำการเลือกเซลล์ที่หัวคิว ซึ่งเซลล์ที่มาถึงอินพุตจะรอคิวอยู่ในบัฟเฟอร์แบบ FIFO เมื่อเซลล์จะไปยังเอาต์พุตใด ๆ ตัวควบคุมการสวิตช์จะทำการปิดจุดเชื่อมต่อระหว่างอินพุตกับเอาต์พุตนั้นเพื่อส่งเซลล์ผ่านสวิตช์ จะเห็นว่าในแต่ละไทม์สล็อตอาจจะมีเซลล์จำนวน k เซลล์ที่จะไปยังเอาต์พุตเดียวกัน แต่จะได้รับการพิจารณาส่งผ่านสวิตช์ได้เพียงหนึ่งเซลล์ เซลล์ที่ไม่ได้รับการส่งจะรออยู่ในคิวเพื่อรอรับการพิจารณาในการส่งผ่านสวิตช์ในไทมสล็อตต่อไป

เราจะวิเคราะห์ที่จุดอิ่มตัว โดยการพิจารณาเลือกเซลล์ส่งผ่านสวิตช์แบบสุ่ม สมมุติว่าการรอคิวที่อินพุตถึงจุดอิ่มตัว (กรณีโหลดหนาแน่น, $\rho=1$) เมื่อใดก็ตามที่เซลล์ส่งผ่านโครงสร้างการสวิตช์จะมีเซลล์มาแทนที่ที่หัวคิวของบัฟเฟอร์ในทันที เราจะนิยาม B_m^i คือจำนวนเซลล์ที่หัวคิวที่จะไปยังเอาต์พุตที่ i ที่ถูกบล็อกไว้ที่จุดสิ้นสุดของไทม์สล็อตที่ m และ A_m^i คือจำนวนเซลล์ที่เลื่อนไปที่หัวคิวที่จะไปเอาต์พุตที่ i ในระหว่างไทม์สล็อตที่ m ดังนั้น เราจะได้ว่า

$$B_m^i = \max\left(0, B_{m-1}^i + A_m^i - 1\right) \tag{2.15}$$

ซึ่งมีรูปแบบทางคณิตศาสตร์ เช่นเดียวกันสมการที่ 2.10

A_m^i มีการแจกแจงแบบทวินาม กล่าวคือความน่าจะเป็นที่จะมี k เซลล์ ไปแทนที่หัวคิวที่ว่าง ในระหว่างไทม์สล็อตที่ m และต้องการจะไปยังเอาต์พุตที่ i ดังสมการ ต่อไปนี้

$$P\left[A_m^i = k\right] = \binom{F_{m-1}}{k}\left(\frac{1}{N}\right)^k\left(1-\frac{1}{N}\right)^{F_{m-1}-k} \qquad k = 0,1,\ldots,F_{m-1} \tag{2.16}$$

โดยที่ $F_{m-1} \equiv N - \sum_{i=1}^{N} B_{m-1}^i$ และหาค่าเฉลี่ยที่สถานะอยู่ตัวทั้งสองด้านของสมการ จะได้ว่า

$$\overline{F} = N - \sum_{i=1}^{N} \overline{B^i} \tag{2.17}$$

F_{m-1} คือจำนวนหัวคิว (HoL) ที่ว่าง ที่จุดสิ้นสุดของไทม์สล็อตที่ $m-1$ ซึ่งก็คือจำนวนเซลล์ที่ส่งผ่านสวิตช์ในระหว่างไทม์สล็อตที่ $m-1$ และยังเท่ากับจำนวน A_m^i ในไทม์สล็อตที่ m นั่นคือ

$$F_{m-1} = \sum_{i=1}^{N} A_m^i \tag{2.18}$$

กำหนดให้ค่าทรูพุตของสวิตช์เป็นไปดังสมการ

$$\rho_0 = \frac{\overline{F}}{N} \tag{2.19}$$

โดยที่ $\overline{F}$ คือ ค่าเฉลี่ยที่สภาวะคงตัวของหัวคิวที่ว่าง

เมื่อให้ $N \to \infty$ จะได้ว่า A^i กลายเป็นการแจกแจงแบบปัวส์ซงที่มีพารามิเตอร์ ρ_0 (ดู ผนวก ก) จากการสังเกตสมการที่ 2.15 รูปแบบทางคณิตศาสตร์เป็นเช่นเดียวกับสมการที่ 2.10 ดังนั้น เราจึงนำผลที่ได้จากกรณีการรอคิวที่เอาต์พุตมาใช้ เพื่อแสดงว่าค่าเฉลี่ยที่สภาวะคงตัวของ B_m^i เมื่อ $N \to \infty$ โดยประยุกต์ใช้สมการที่ 2.14 มีค่าดังนี้ คือ

$$\overline{B^i} = \frac{\rho_0^2}{2(1-\rho_0)} \tag{2.20}$$

และใช้ค่าเฉลี่ยของสมการที่ 2.17 กับสมการที่ 2.19 เราจะได้ว่า

$$\overline{B^i} = 1-\rho_0 \tag{2.21}$$

จากสมการที่ 2.20 และสมการที่ 2.21 จะได้ว่า

$$\rho_0 = \left(2-\sqrt{2}\right) = 0.586 \tag{2.22}$$

นั่นคือ ค่าทรูพุตของสวิตช์ เมื่อสวิตช์ถึงจุดอิ่มตัว และ $N \to \infty$

ตามตารางที่ 2.2 แสดงค่าทรูพุตสูงสุดของสวิตช์รอคิวที่อินพุตขนาด 16x16 กับการจัดการรอคิวแบบ FIFO โดยกำหนดให้โหลดทราฟฟิกคือ $\rho = 1$ ผลจากการจำลองแบบสวิตช์ได้จากการเขียนด้วยโปรแกรมภาษา C++

ตารางที่ 2.2 ค่าทรูพุตสูงสุดของสวิตช์รอคิวที่อินพุต กับการจัดการรอคิวแบบ FIFO, $\rho = 1$

ขนาดสวิตช์ (N)	ทรูพุต
1	1.0000
2	0.7498
4	0.6557
8	0.6189
16	0.6021
32	0.5938
64	0.5896
128	0.5875
∞	0.5858 (ค่าตามทฤษฎี)

2.5 สรุปท้ายบท

จากการที่การสวิตช์วงจรนั้น ไม่เหมาะสมกับโครงข่ายที่มีการส่งผ่านข้อมูลแบบแพ็กเกต การสวิตช์แพ็กเกตจึงได้ถือกำเนิดขึ้นมา โดยสามารถส่งแพ็กเกตได้ทั้งแบบดาตาแกรม และแบบวงจรเสมือน โดยทั่วไปแล้วสถาปัตยกรรมสวิตช์แบ่งออกเป็น 2 กลุ่ม คือ สวิตช์แบบแบ่งเวลา และสวิตช์แบบแบ่งเส้นทาง สวิตช์แบบแบ่งเวลายังแบ่งย่อยออกเป็น แบบใช้สื่อนำสัญญาณร่วมกัน และแบบใช้หน่วยความจำร่วมกัน สวิตช์แบบแบ่งเส้นทางนั้นแบ่งย่อยออกเป็น แบบเส้นทางเดียว และแบบหลายเส้นทาง สวิตช์แบบครอสบาร์เป็นสวิตช์แบบเส้นทางเดียวที่ถูกนำมาใช้เป็นโครงสร้างการสวิตช์ของสวิตช์ความเร็วสูงทั้งสวิตช์เอทีเอ็มและเราเตอร์ไอพี เนื่องจากโครงสร้างการสวิตช์สามารถส่งผ่านเซลล์มากกว่าหนึ่งเซลล์ในเวลาเดียวกันได้ เป็นสวิตช์ที่ไม่มีการติดขัดภายในและนำไปสร้างเป็นสวิตช์ได้ง่าย ส่วนสุดท้ายได้ทำการวิเคราะห์สมรรถนะของสวิตช์พื้นฐาน 2 ประเภท คือ สวิตช์แบบรอคิวที่เอาต์พุต และสวิตช์แบบรอคิวที่อินพุต

บทที่ 3
สถาปัตยกรรมสวิตช์แพ็กเกต

หลังจากเราได้ศึกษาหลักการพื้นฐานของการสวิตช์แพ็กเกตมาแล้ว ในบทนี้จะได้อธิบายรายละเอียดของสถาปัตยกรรมสวิตช์แพ็กเกตประเภทต่าง ๆ ที่มีอยู่ในปัจจุบัน โดยทั่วไปแล้วสวิตช์แพ็กเกตจะมีโครงสร้างสวิตช์ (switch fabric) แสดงดังรูปที่ 3.1 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

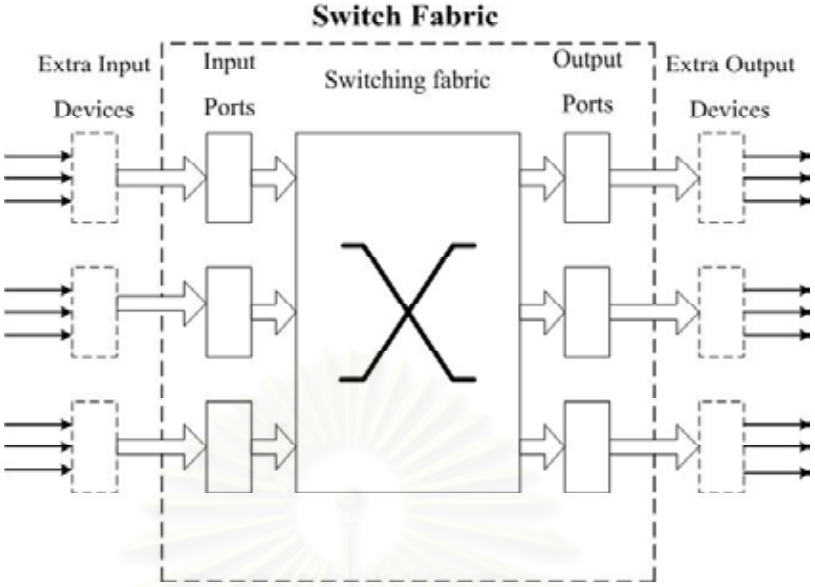
ในปัจจุบัน ไม่ว่าเป็นสวิตช์หรือเราเตอร์ที่ส่งผ่านแพ็กเกตไอพี ส่วนใหญ่ใช้โครงสร้างแบบสวิตช์เซลล์ เวลาที่ใช้ในการส่งเซลล์แต่ละเซลล์ผ่านโครงสร้างสวิตช์เรียกว่า ไทม์สล็อต (time slot) โดยโครงสร้างสวิตช์ ประกอบไปด้วยองค์ประกอบมูลฐาน 3 องค์ประกอบ ดังต่อไปนี้

1) พอร์ตอินพุต ทำหน้าที่รับหน่วยข้อมูลหรือเซลล์ ที่มาถึงที่พอร์ตอินพุตของโครงสร้างสวิตช์

2) พอร์ตเอาต์พุต ทำหน้าที่ส่งหน่วยข้อมูลออกจากพอร์ตเอาต์พุตของโครงสร้างสวิตช์

3) โครงสร้างการสวิตช์ หรือโครงข่ายเชื่อมต่อระหว่างกัน (interconnection network) ทำหน้าที่ให้การเชื่อมต่อระหว่างพอร์ตอินพุตกับเอาต์พุต เพื่อส่งผ่านหน่วยข้อมูล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1 แบบจำลองของสวิตช์พื้นฐาน [4]

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)

3.1 สวิตช์แบบรอคิวที่เอาต์พุต (OQ Switches)

เป็นสวิตช์ที่มีรูปแบบการวางบัฟเฟอร์ไว้ที่เอาต์พุตของสวิตช์ โดยสวิตช์จะทำการส่งผ่านเซลล์ทุก ๆ เซลล์ที่มาถึงพอร์ตอินพุตของสวิตช์ไปยังพอร์ตเอาต์พุตของสวิตช์โดยทันที โดยเซลล์จะรอคิวอยู่ในบัฟเฟอร์เพื่อรอการส่งออก จึงมีความจำเป็นที่สวิตช์จะต้องมีความเร็วในการสวิตช์สูงกว่าความเร็วแพ็กเกตเป็นจำนวน N เท่า (N คือขนาดของสวิตช์) จะเห็นว่าถ้าสวิตช์มีขนาดใหญ่ขึ้นจะไม่เหมาะสมที่จะนำมาใช้ในทางปฏิบัติ แต่สวิตช์แบบนี้ก็มีข้อดีตรงที่ว่าสมรรถนะในการส่งผ่านเซลล์ที่สูงกว่าและค่าเฉลี่ยการประวิงเวลาของเซลล์ที่ต่ำกว่าสวิตช์แบบรอคิวที่อินพุต

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

สวิตช์แบบ SOQ จัดวางบัฟเฟอร์เฉพาะที่พอร์ตเอาต์พุตของสวิตช์เท่านั้น และจัดเป็นสวิตช์แบบที่ใช้หน่วยความจำร่วมกัน (shared-memory switch) ซึ่งในการจัดสรรหน่วยความจำเพื่อใช้

ร่วมกันนั้น มีอยู่ 2 วิธีคือ วิธีที่หนึ่งแบ่งหน่วยความจำขนาดเท่า ๆ กันให้บัฟเฟอร์ในแต่ละเอาต์พุต เรียกว่า การจัดสรรหน่วยความจำแบบสถิตย์ (complete partitioning) หรือวิธีที่สองให้ใช้หน่วยความจำร่วมกันสำหรับบัฟเฟอร์ทุก ๆ พอร์ตเอาต์พุต เรียกว่า การจัดสรรหน่วยความจำแบบพลวัต (full sharing)

วิธีการจัดสรรหน่วยความจำแบบสถิตย์ เซลล์ที่จะไปยังเอาต์พุตเดียวกันจะถูกทิ้งไปถ้าบัฟเฟอร์ที่เอาต์พุตนั้นเต็ม ส่วนวิธีการจัดสรรหน่วยความจำแบบพลวัตเซลล์จะยังไม่ถูกทิ้งไปตราบเท่าที่หน่วยความจำทั้งหมดยังไม่เต็ม จึงทำให้สมรรถนะของสวิตช์ทางด้านอัตราการสูญเสียเซลล์ (lost rate) ดีกว่าจึงกลายเป็นสิ่งที่ต้องการในการออกแบบ ซึ่งเราสามารถออกแบบให้สวิตช์มีขนาดหน่วยความจำเล็กลงได้มากกว่าการใช้วิธีการจัดสรรหน่วยความจำแบบสถิตย์ [6]

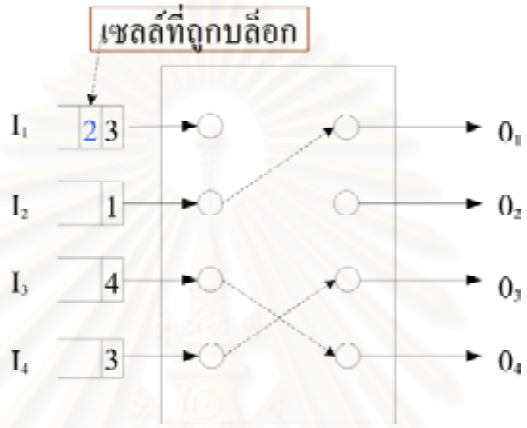
3.1.2 สวิตช์แบบ MOQ (Multiple Output-Queued (MOQ) Switches)

สวิตช์แบบ MOQ [1] มีการขยายบัฟเฟอร์ของพอร์ตเอาต์พุตไปที่พอร์ตอินพุตด้วย จัดเป็นสวิตช์ที่ใช้หน่วยความจำร่วมกัน หรือเป็นสวิตช์ที่มีโครงสร้างสวิตช์ที่มีการรอคิวที่เอาต์พุต ในการนำไปสร้างเป็นสวิตช์นั้นมีลักษณะเช่นเดียวกันกับสวิตช์แบบ OQ แต่จะมีความซับซ้อนมากกว่าในการนำไปสร้างเป็นสวิตช์ความเร็วสูงที่เหมาะสมในทางปฏิบัติ และโดยใช้ฟังก์ชันแบ็กเพรสเชอร์ (back-pressure) จำกัดการทำงานและการเก็บเซลล์ที่มาถึงบัฟเฟอร์ ซึ่งเป็นหน้าที่ของไลน์การ์ด จึงทำให้โครงสร้างของบัฟเฟอร์เพียงแต่ควบคุมคิวที่มีขนาดเล็กลงและมีจำนวนเซลล์ไม่มาก ทำให้สวิตช์ประเภทนี้มีลักษณะใกล้เคียงกับสวิตช์แบบ OQ มากที่สุด ซึ่งการสร้างสวิตช์เลียนแบบสวิตช์แบบ OQ เรียกว่า การอีมูเลชันสวิตช์แบบ OQ (OQ emulation) จะเห็นว่าสวิตช์แบบ MOQ มีลักษณะคล้ายคลึงกับสวิตช์แบบ CIOQ มาก แตกต่างกันตรงที่ว่า การวางบัฟเฟอร์ที่อินพุตนั้นมีหน้าที่แต่เพียงว่าเป็นการขยายบัฟเฟอร์ของเอาต์พุตไปที่อินพุตเพียงเท่านั้น

3.2 สวิตช์แบบรอคิวที่อินพุต (IQ Switches)

ข้อดีของสวิตช์ประเภทนี้ก็คือ โครงสร้างการสวิตช์ทำงานด้วยความเร็วในการสวิตช์เท่ากับความเร็วแพ็กเกต (line rate) โดยที่แพ็กเกตแต่ละแพ็กเกตที่เข้าสู่สวิตช์จะถูกแบ่งออก เป็นเซลล์โดยฟังก์ชัน SAR และบรรจุในบัฟเฟอร์ที่อินพุตของสวิตช์โดยทันทีเพื่อรอการได้รับบริการส่งผ่านสวิตช์ นอกจากนี้ สวิตช์จะไม่ประสบกับปัญหาการแออัดหรือชนกันของเซลล์ภายในสวิตช์ (internal contention) เพราะว่าสวิตช์สามารถควบคุมและจัดการในการส่งเซลล์ผ่านสวิตช์ได้ตามต้องการ ในการพิจารณาเลือกเซลล์ที่หัวคิวสำหรับส่งผ่านสวิตช์มีอยู่หลายวิธี ตัวอย่างเช่น วิธีเลือกแบบสุ่ม (random), แบบวนรอบ (round robin), เลือกคิวยาวที่สุด (longest) หรือเลือกตามลำดับความสำคัญที่กำหนดแน่นอน (fixed priority) เป็นต้น [7]

ในการวางบัฟเฟอร์ไว้ที่อินพุตของสวิตช์ ถ้าเราจัดการรอคิวแบบมาถึงก่อนได้รับบริการก่อน (first-in-first-out: FIFO) ในแต่ละอินพุต ทำให้เกิดปัญหาการติดขัดที่หัวคิว (HoL blocking) ซึ่งเกิดขึ้นจากการที่เซลล์ที่จะไปยังเอาต์พุตที่ว่างอยู่หลังเซลล์ที่หัวคิวที่ไม่สามารถส่งออกได้ เนื่องจากจะไปยังเอาต์พุตที่ไม่ว่าง แสดงดังรูปที่ 3.2 ตามรูปเซลล์ที่อยู่หลังเซลล์ที่หัวคิวที่พอร์ตอินพุตที่ 1 จะไปยังพอร์ตเอาต์พุตที่ 2 แต่ถูกบล็อกไว้จากเซลล์ที่หัวคิวทำให้ไม่สามารถส่งออกได้ จึงทำให้ค่าทรูพุตสูงสุดของสวิตช์ลดลง



รูปที่ 3.2 แสดงการติดขัดที่หัวคิว

เราสามารถแก้ปัญหานี้ได้ โดยการพิจารณาเลือกเซลล์ส่งผ่านสวิตช์ลึกเข้าไปถึงเซลล์ที่อยู่ถัดจากเซลล์ที่หัวคิวเป็นจำนวน w เซลล์ วิธีนี้เรียกว่าการเลือกวินโดว์ (window selection) ทำให้เกิดความยืดหยุ่นในการเลือกเซลล์ส่งผ่านโครงสร้างการสวิตช์ แม้ว่าแต่ละอินพุตจะยังคงส่งออกเซลล์ได้ไม่เกินหนึ่งเซลล์ในแต่ละไทม์สล็อต แต่ก็ไม่จำเป็นที่จะต้องเป็นเซลล์แรกที่รออยู่ในคิวที่จะต้องได้รับการส่งออก ที่จุดเริ่มต้นในแต่ละไทม์สล็อต เซลล์จำนวน w เซลล์แรกของแต่ละอินพุต จะได้รับการพิจารณาส่งผ่านโครงสร้างการสวิตช์ตามลำดับ โดยที่เซลล์ที่หัวคิวจะได้สิทธิที่จะได้รับการเลือกก่อนเป็นลำดับแรก และเนื่องจากอาจจะมีการชนกันที่เอาต์พุต บางอินพุตอาจจะไม่ได้รับเลือก เซลล์ในลำดับถัดไปของอินพุตที่ไม่ได้รับเลือกก็จะได้รับการพิจารณาเพื่อส่งผ่านโครงสร้างการสวิตช์ถ้าเอาต์พุตที่เซลล์นั้นจะไปถึงยังว่างอยู่ และถ้าเกิดการชนกันอีกและไม่ได้รับเลือก ก็จะพิจารณาเซลล์ในลำดับถัดไปจนถึง w เซลล์ จะเห็นว่า กรณี w = 1 ก็คือ กรณีที่อินพุตมีการรอคิวแบบ FIFO ที่ทำให้เกิดการติดขัดที่หัวคิวนั่นเอง

ตามตารางที่ 3.1 แสดงค่าทรูพุตสูงสุดที่ได้จากสวิตช์ขนาดต่าง ๆ และวินโดว์ (w) ขนาดต่าง ๆ กัน ซึ่งเป็นค่าที่ได้จากผลการจำลองแบบด้วยโปรแกรมภาษา C++ จะเห็นว่าค่าทรูพุตเพิ่มขึ้นเมื่อ w มีค่าเพิ่มขึ้น แต่ค่าทรูพุตที่เพิ่มขึ้นมีสัดส่วนที่ช้าลงเมื่อค่า w เพิ่มมากขึ้น เนื่องมาจากพอร์ต

เอาต์พุตที่ยังว่างอยู่หลังจากการพิจารณาเซลล์ในการส่งผ่านโครงสร้างการสวิตช์ในลำดับต่อ ๆ มาลดน้อยลงนั่นเอง

ตารางที่ 3.1 ค่าทรูพุตสูงสุดของสวิตช์แบบรอคิวที่อินพุตขนาด NxN กับวิธีการเลือกวินโดว์

	W							
N	1	2	3	4	5	6	7	8
2	0.75	0.84	0.89	0.92	0.93	0.94	0.95	0.96
4	0.66	0.76	0.81	0.85	0.87	0.89	0.91	0.92
8	0.62	0.72	0.78	0.82	0.85	0.87	0.88	0.89
16	0.6	0.71	0.77	0.81	0.84	0.86	0.87	0.88
32	0.59	0.70	0.76	0.80	0.83	0.85	0.87	0.88
64	0.56	0.70	0.76	0.80	0.83	0.85	0.86	0.88
128	0.56	0.70	0.76	0.80	0.83	0.85	0.86	0.88

3.2.1 อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์

เมื่อใช้สวิตช์แบบครอสบาร์เป็นโครงสร้างการสวิตช์ของสวิตช์แบบรอคิวที่อินพุต ในแต่ละไทม์สล็อต เราจึงมีความต้องการอัลกอริทึมในการจัดลำดับเซลล์เพื่อพิจารณาเลือกเซลล์ที่หัวคิวที่จะได้รับบริการส่งผ่านโครงสร้างการสวิตช์ การจัดลำดับเซลล์ส่งผ่านสวิตช์แทนได้ด้วยแบบจำลองปัญหาการแมตช์บนกราฟไบพาร์ไทท์ (bipartite graph matching) สถานะของสวิตช์สามารถอธิบายได้ด้วยกราฟไบพาร์ไทท์ G = [V,E] โดยที่จุดยอดคือเซตของ V ซึ่งแบ่งออกเป็นสองเซตย่อย คือ เซตย่อย V_i มีสมาชิกเท่ากับ $v_i^{(k)}$ แทนพอร์ตอินพุต และเซตย่อย V_o มีสมาชิกเท่ากับ $v_o^{(k)}$ แทนพอร์ตเอาต์พุต ด้าน (edges) แทนความต้องการที่จะส่งเซลล์จากอินพุตไปยังเอาต์พุตผ่านโครงสร้างการสวิตช์ (ด้านระหว่าง $v_i^{(m)}$ กับ $v_o^{(n)}$ บ่งชี้ถึงความต้องการที่จะส่งเซลล์จากอินพุตที่ m ไปยังเอาต์พุตที่ n) และแทนได้ด้วยเมตริกซ์น้ำหนัก (weight matrix) ด้วยสัญลักษณ์ $w^{(mn)}$ เมตริกซ์ที่นำมาใช้นี้มีบทบาทสำคัญสำหรับอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ เพราะว่าสามารถบ่งชี้ได้ว่ามีอย่างน้อยหนึ่งเซลล์ที่จะได้รับการส่งผ่านสวิตช์ หรือแทนจำนวนเซลล์ที่จะได้รับการส่งผ่านสวิตช์ หรือแทนเวลารอคิวของเซลล์ที่หัวคิวก่อนที่จะได้รับบริการส่งผ่านสวิตช์

การแมตช์ (M) หมายถึงการเลือกเซตย่อยของด้านที่ตรงตามเงื่อนไข (admissible) เซตย่อยของด้านที่ตรงตามเงื่อนไขก็ต่อเมื่อไม่มีจุดยอดใดมีด้านมากกว่าหนึ่งด้านเชื่อมต่ออยู่ แสดงว่าจะไม่

มีเหตุการณ์ที่เซลล์สองเซลล์ออกจากอินพุตเดียวกัน หรือมีเซลล์สองเซลล์มาถึงเอาต์พุตเดียวกันในเวลาเดียวกัน

การแมตช์จะมีขนาดสูงที่สุด (maximum size matching) ก็ต่อเมื่อมีจำนวนด้านสูงที่สุด และการแมตช์จะมีน้ำหนักสูงที่สุด (maximum weight matching) ก็ต่อเมื่อผลรวมของเมตริกซ์ด้านมีค่าสูงที่สุด การแมตช์จะมีขนาดมากที่สุด (maximal matching) ก็ต่อเมื่อถ้ามีการเพิ่มด้านใด ๆ บนกราฟ G แล้วจะทำให้มันไม่ตรงตามเงื่อนไข (inadmissible)

ความต้องการของเราก็คือ ต้องการหาอัลกอริทึมการแมตช์ที่เหมาะสมที่สุดที่จะนำมาใช้ ซึ่งหาได้จากการนิยามค่าความซับซ้อนของอัลกอริทึม (complexity) อัลกอริทึม MWM (maximum weight matching (MWM) algorithm) มีค่าความซับซ้อน $O(N^3)$ ส่วนอัลกอริทึม MSM (maximum size matching (MSM) algorithm) มีค่าความซับซ้อน $O(N^{5/2})$ ซึ่งมีค่าความซับซ้อนน้อยกว่า แต่ก็มีประสิทธิภาพด้อยกว่าอัลกอริทึม MWM

เมตริกซ์ขนาด NxN ซึ่งมีสมาชิกคือ เมตริกซ์ด้านบนกราฟ G = [V,E] ถูกเรียกว่า เมตริกซ์น้ำหนัก (weight matrix) แทนด้วยสัญลักษณ์ $W = [w^{(ij)}]$ เมตริกซ์ W มีค่าเปลี่ยนแปลงตามเวลา ซึ่งสอดคล้องกับการเปลี่ยนแปลงค่าพารามิเตอร์ในระบบซึ่งสมาชิกของเมตริกซ์ถูกนำมาคำนวณ เราจะสมมุติว่า $w^{(ij)} = 0$ เมื่อไม่มีด้านบนกราฟ G ซึ่งหมายความว่าไม่มีเซลล์จากอินพุตที่ i ที่จะไปยังเอาต์พุตที่ j รออยู่ในคิว

อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ ตามตารางที่ 3.2 สามารถแบ่งย่อยออกเป็นองค์ประกอบหลัก 2 องค์ประกอบ คือ

1) Metrics computation จะทำการคำนวณหาเมตริกซ์น้ำหนัก $W = [w^{(ij)}]$ แต่ละด้านจากด้านที่เป็นไปได้ N^2 ด้านบนกราฟ G ซึ่งสัมพันธ์กับเมตริกซ์เนื่องจากสถานะของคิว กล่าวคือ $w_k^{(ij)}$ ขึ้นอยู่กับ $X_k^{(ij)}$ ซึ่งเป็นสถานะของคิว $Q^{(ij)}$ ในไทม์สล็อตที่ k จะเห็นว่าเมตริกซ์มีพฤติกรรมเหมือนเป็นค่าลำดับความสำคัญของเซลล์ต่าง ๆ ที่จะได้รับการส่งผ่านสวิตช์

2) Matching method จะทำการคำนวณหาค่าการแมตช์ อัลกอริทึม MSM, PIM, iSLIP และ DRRM มีเป้าหมายที่จะหาค่าการแมตช์ขนาดสูงที่สุด ส่วนอัลกอริทึมที่เหลือพยายามที่จะหาค่าการแมตช์น้ำหนักสูงที่สุด

ตารางที่ 3.2 ลักษณะของอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์แบบต่าง ๆ [8]

Algorithm	Metice	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
PIM, iSLIP, DRRM	QO: Queue Occupancy	Iterative search

สรุปได้ว่าอัลกอริทึม MWM-QL, MWM-CA และ MSM นั้นเป็นอัลกอริทึมที่เหมาะสมที่สุด (optimal algorithms) ที่ใช้คำนวณหาค่า MWM หรือ MSM ส่วนอัลกอริทึม iSLIP, PIM, DRRM, iLQF หรือ iOCF นั้นคำนวณหาค่าขนาดมากที่สุดโดยวิธีวนซ้ำ (iterative search) และมีความซับซ้อนน้อยกว่าในการนำไปสร้างเป็นฮาร์ดแวร์ ในทางปฏิบัติจึงนำไปสร้างเป็นสวิตช์ความเร็วสูง

3.2.2 สวิตช์แบบ MIQ (Multiple Input-Queued (MIQ) Switches) [1]

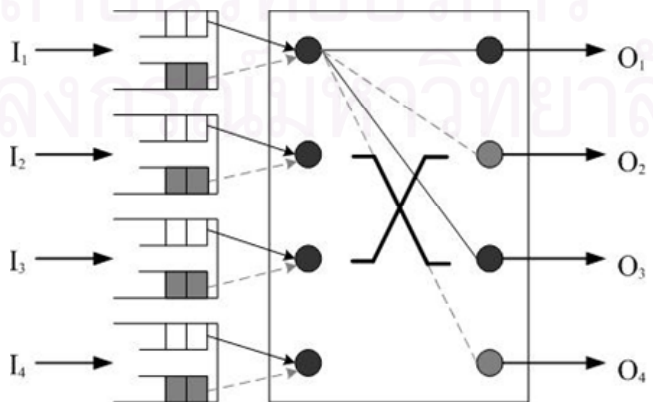
ถ้าเราเปรียบเทียบกับสวิตช์แบบ OQ จะเห็นว่าสวิตช์แบบ IQ มีข้อดีที่เหนือกว่าสวิตช์แบบ OQ ตรงที่ว่าสวิตช์สามารถทำงานได้ที่ความเร็วในการสวิตช์เท่ากับความเร็วแพ็กเกต ทำให้สวิตช์แบบ IQ มีความเหมาะสมอย่างยิ่งที่จะนำไปใช้กับระบบที่มีการสวิตช์ความเร็วสูง การรอคิวที่อินพุตถูกพิจารณาว่านำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายกว่าการรอคิวที่เอาต์พุต แต่เนื่องจากสวิตช์แบบ IQ ประสบปัญหาการติดขัดที่หัวคิว ทำให้ค่าทรูพุตสุงสุดของสวิตช์ลดลงลู่เข้าสู่ค่า 0.58 เมื่อ N มีค่าเข้าสู่อนันต์ ดังนั้น จึงได้มีการศึกษาเพื่อหารูปแบบการจัดวางบัฟเฟอร์ (buffering) และการจัดลำดับเซลล์ (scheduling) เพื่อที่จะเอาชนะขีดจำกัดของค่าทรูพุตสูงสุด ตัวอย่างเช่น วิธีการเลือกวินโดว์ วิธี input smoothing หรือวิธี multiple input-queueing (MIQ) แต่ละรูปแบบนั้นยอมให้เซลล์ต่าง ๆ ที่อยู่ในลำดับถัดจากเซลล์ที่หัวคิว (HoL cell) สามารถส่งผ่านสวิตช์ได้ เมื่อเซลล์ที่หัวคิวไม่สามารถส่งออกได้เนื่องจากเอาต์พุตที่จะไปนั้นไม่ว่าง หรือยอมให้มีหลาย ๆ เซลล์เป็นเซลล์ที่หัวคิวในแต่ละอินพุต เพื่อเพิ่มโอกาสในการที่จะได้รับบริการส่งผ่านสวิตช์ รูปแบบที่ได้รับความสนใจมากที่สุดคือ รูปแบบ MIQ เนื่องจากให้สมรรถนะของสวิตช์ที่ดี และสามารถทำงานได้ดีในการสวิตช์ความเร็วสูง

รูปที่ 3.3 เป็นสวิตช์แบบ MIQ ขนาด NxN (N คือขนาดของสวิตช์) โดยที่ในแต่ละอินพุตมีการจัดการแยกคิวทางตรรกะออกเป็น m คิว ($1 \leq m \leq N$) ซึ่งแต่ละคิวจะมีเฉพาะเซลล์ที่จะไปยัง

กลุ่มของเอาต์พุตที่กำหนด กรณีที่ m = 1 ก็คือ สวิตช์แบบ SIQ (Single Input-Queued (SIQ) Switch) และกรณีที่ m = N ก็คือสวิตช์แบบ VOQ (Virtual Output-Queued (VOQ) Switch) ซึ่งสวิตช์แบบ VOQ นั้น เป็นกรณีพิเศษของสวิตช์แบบ MIQ และมีความน่าจะเป็นที่จะเกิดการติดขัดที่หัวคิวเป็นศูนย์ จะเห็นว่าความน่าจะเป็นที่จะเกิดการติดขัดที่หัวคิวมีค่าลดลงเมื่อจำนวนการแยกคิวในแต่ละอินพุตมีค่าเพิ่มขึ้น

ปกติแล้ว ค่า m จะเป็นจำนวนเท่าของ 2 (ตัวอย่างเช่น 1, 2, 4,..., N) หรือ mN (m อาจมีขนาดใหญ่กว่า N ก็ได้แต่ก็ไม่เหมาะสมในทางปฏิบัติ) ดังนั้นแต่ละบัฟเฟอร์ในแต่ละอินพุตจะบรรจุเซลล์ที่จะไปยังกลุ่มเอาต์พุตจำนวน N/m พอร์ต โดยที่กลุ่มเอาต์พุตแต่ละกลุ่มจะต้องไม่ซ้ำกันกับกลุ่มเอาต์พุตอื่น ๆ ซึ่งเราจะสังเกตได้ว่าเราสามารถแยกสวิตช์ออกเป็นสวิตช์ย่อยได้ m สวิตช์ โดยที่แต่ละสวิตช์ย่อยจะมีขนาด NxN/m ซึ่งสวิตช์ย่อยแต่ละตัวตรงกันกับกลุ่มของพอร์ตเอาต์พุต

ในแต่ละไทม์สล็อต สวิตช์แบบ MIQ สามารถส่งเซลล์ในแต่ละอินพุตผ่านโครงสร้างการสวิตช์ได้ถึงจำนวน m เซลล์ แต่อย่างไรก็ตาม เมื่อรวมจำนวนเซลล์ทั้งหมดจากทุก ๆ อินพุตแล้วจะต้องไม่เกิน N เซลล์ เนื่องจากแต่ละเอาต์พุตรับเซลล์ที่มาจากอินพุตได้ไม่เกิน 1 เซลล์ แม้ว่าการส่งออกเซลล์มากกว่าหนึ่งเซลล์จากแต่ละอินพุตเป็นสิ่งจำเป็นที่จะทำให้สมรรถนะของสวิตช์สูงขึ้น แต่ก็ต้องเพิ่มความเร็วในการสวิตช์ข้อมูลในขณะเดียวกัน ซึ่งเป็นสิ่งที่ไม่พึงปรารถนาสำหรับสวิตช์ความเร็วสูง ด้วยเหตุนี้ จึงกลายเป็นปัญหาสำคัญที่จะต้องทำการแก้ไขถ้าจะนำไปสร้างเป็นสวิตช์ความเร็วสูงในอนาคต ดังนั้นจึงได้มีการปรับปรุงสวิตช์แบบ MIQ โดยสมมุติว่าในแต่ละไทม์สล็อตให้แต่ละอินพุตสามารถส่งออกเซลล์ได้ไม่เกินหนึ่งเซลล์ ทำให้สวิตช์สามารถทำงานได้ที่ความเร็วแพ็กเกต ซึ่งทำให้สวิตช์มีความเหมาะสมในการนำไปใช้กับระบบที่มีการสวิตช์ความเร็วสูง และต้องการสมรรถนะของสวิตช์ที่สูงขึ้นด้วย



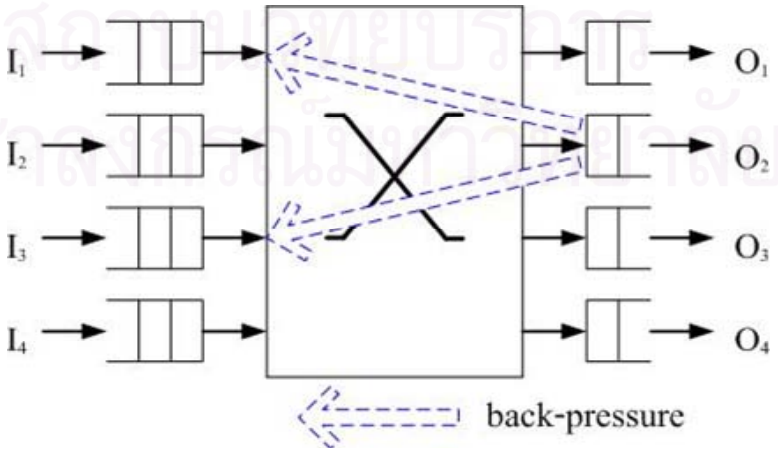
รูปที่ 3.3 สวิตช์แบบ MIQ

3.3 สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (CIOQ switches)

จากที่กล่าวมาแล้ว สวิตช์แบบ OQ นั้น เป็นสวิตช์ที่มีสมรรถนะดีกว่าสวิตช์ประเภทอื่น ๆ แต่เนื่องจากความเร็วในการสวิตช์นั้นจะต้องเป็น N เท่าของความเร็วแพ็กเกต จึงไม่เหมาะสมในการนำไปสร้างเป็นสวิตช์ความเร็วสูงหรือสวิตช์ที่มีขนาดใหญ่ และจากข้อดีของสวิตช์แบบ IQ ที่มีความเร็วในการสวิตช์เท่ากับความเร็วแพ็กเกต จึงได้มีการนำเสนอรูปแบบของสวิตช์ที่มีการวางบัฟเฟอร์ทั้งทางด้านอินพุตและเอาต์พุต [10] รวมกับฟังก์ชันการทำงานที่เรียกว่าแบ็กเพรสเชอร์ (back-pressure) และเราเรียกสวิตช์ประเภทนี้ว่า สวิตช์แบบรอคิวที่อินพุตและเอาต์พุต (CIOQ)

ในทางปฏิบัตินั้น สวิตช์แบบ IQ ส่วนใหญ่มีความต้องการที่จะให้เป็นสวิตช์แบบไม่ติดขัด (nonblocking) ทำให้ต้องมีการเพิ่มความเร็วในการสวิตช์ (เช่น เป็น 2 เท่าของความเร็วแพ็กเกต) ทำให้เกิดการสะสมของเซลล์ที่พอร์ตเอาต์พุต ด้วยเหตุนี้ จึงมีความจำเป็นที่จะต้องมีการวางบัฟเฟอร์ที่เอาต์พุตเพื่อให้เซลล์ที่ยังไม่ได้รับการส่งออกจากสวิตช์รออยู่ในคิว สวิตช์แบบ IQ ที่มีการเพิ่มความเร็วในการสวิตช์จึงเป็นสวิตช์แบบ CIOQ ด้วยเช่นเดียวกัน

สถาปัตยกรรมสวิตช์แบบ CIOQ แสดงดังรูปที่ 3.4 จะมีการวางบัฟเฟอร์ที่มีหน่วยความจำความเร็วต่ำจำนวนมากที่พอร์ตอินพุต และมีการวางบัฟเฟอร์ที่มีหน่วยความจำความเร็วสูงจำนวนไม่มากที่พอร์ตเอาต์พุต เพื่อเพิ่มสมรรถนะของสวิตช์ให้สูงขึ้นจึงทำการควบคุมการรอคิวในบัฟเฟอร์ทั้งทางด้านอินพุตและเอาต์พุต โดยให้เป็นหน้าที่ของฟังก์ชันแบ็กเพรสเชอร์ซึ่งจะส่งสัญญาณไปที่อินพุต เพื่อห้ามมิให้ส่งออกเซลล์จากอินพุตไปยังเอาต์พุตที่เซลล์ในบัฟเฟอร์จะไปยังเอาต์พุตนั้น เมื่อจำนวนเซลล์ในบัฟเฟอร์ที่เอาต์พุตนั้นเต็มหรือเกินค่าเทรสโฮลด์ (threshold) ทั้งนี้ก็เพื่อป้องกันเซลล์ล้นบัฟเฟอร์ที่เอาต์พุตจนต้องทิ้งเซลล์นั้นไป ฟังก์ชันแบ็กเพรสเชอร์จึงป้องกันไม่ให้เกิดความคับคั่งขึ้นในสวิตช์



รูปที่ 3.4 แบบจำลองสวิตช์แบบ CIOQ และฟังก์ชันแบ็กเพรสเชอร์ [10]

3.4 สวิตช์แบบต่อขนาน

สวิตช์แบบต่อขนาน (Parallel Packet Switch: PPS) เป็นเทคนิคในการนำสถาปัตยกรรมสวิตช์แบบต่าง ๆ ดังที่กล่าวมาแล้วไปประยุกต์ใช้ เพื่อให้การนำไปสร้างเป็นสวิตช์ความเร็วสูงในทางปฏิบัติเป็นไปได้ง่ายขึ้น

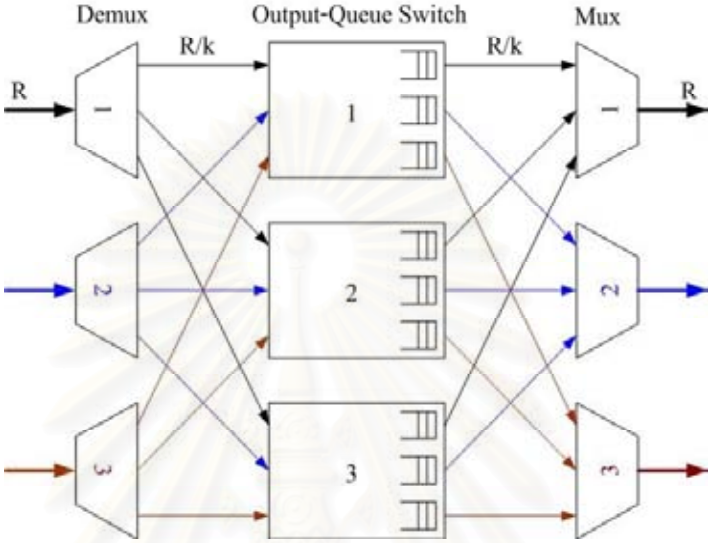
สวิตช์แบบ PPS มีเป้าหมายที่จะเอาชนะขีดจำกัดทางด้านแบนด์วิดท์ของหน่วยความจำของสวิตช์แบบ OQ สวิตช์แบบ PPS ประกอบไปด้วยสวิตช์ความเร็วต่ำชนิดเดียวกันต่อขนานกัน แพ็กเกตที่มาถึงอินพุตของสวิตช์เป็นลำดับจะถูกกระจายออกไปแพ็กเกตต่อแพ็กเกต ซึ่งเป็นหน้าที่ของตัวดีมัลติเพลกซ์ ส่งผ่านสวิตช์ความเร็วต่ำ แล้วรวมกันใหม่อีกครั้งโดยตัวมัลติเพลกซ์ที่เอาต์พุตก่อนออกจากสวิตช์ สถาปัตยกรรมสวิตช์แบบ PPS มีลักษณะคล้ายกันกับโครงข่ายคลอส (Clos network) ดังรูปที่ 3.5 จะประกอบไปด้วย ตัวดีมัลติเพลกซ์ สวิตช์ตอนกลาง และตัวมัลติเพลกซ์ เปรียบเทียบได้กับโครงข่ายคลอสสามสเตจแบบไม่มีบัฟเฟอร์

ถ้าเรามองทางด้านแพ็กเกตที่มาถึง สวิตช์แบบ PPS ก็มีลักษณะเป็นสวิตช์สเตจเดียว (single stage packet switches) การจัดการรอคิวที่สวิตช์ตอนกลางที่ต่อขนานกัน ก็มีเป้าหมายที่จะไม่ให้สวิตช์มีความจำเป็นที่จะต้องทำงานที่ความเร็วเท่ากับความเร็วแพ็กเกต ตัวดีมัลติเพลกซ์จะทำการเลือกสวิตช์ตอนกลาง (เราเรียกสวิตช์ตอนกลางที่ต่อขนานแต่ละตัวว่า เลเยอร์ (layer)) เพื่อที่จะส่งแพ็กเกตที่มาถึงไปยังเลเยอร์นั้น แพ็กเกตจะรอคิวอยู่ในบัฟเฟอร์ของสวิตช์ตอนกลางจนกว่าจะถึงเวลาส่งออก (departure time) เมื่อเวลาส่งออกมาถึงแพ็กเกตจะถูกส่งต่อไปยังตัวมัลติเพลกซ์ ซึ่งจะทำหน้าที่ส่งแพ็กเกตออกจากสวิตช์ เราจะเห็นว่าฟังก์ชันการทำงานของตัวดีมัลติเพลกซ์และตัวมัลติเพลกซ์ มีบทบาทสำคัญในสวิตช์ประเภทนี้

สวิตช์ตอนกลางนั้น เราอาจจะใช้สถาปัตยกรรมสวิตช์แบบใดก็ได้ที่เหมาะสม ตัวอย่างเช่น สวิตช์แบบ OQ หรือ สวิตช์แบบ CIOQ ดังรูปที่ 3.5 ซึ่งเป็นสวิตช์แบบ PPS ขนาด 3x3 โดยใช้สวิตช์ตอนกลางเป็นสวิตช์แบบ OQ ซึ่งแต่ละอินพุตของสวิตช์ทำงานที่ความเร็วเท่ากับ R แต่ละพอร์ตของตัวดีมัลติเพลกซ์เชื่อมต่อกับสวิตช์แบบ OQ ตอนกลางทั้งหมด 3 ตัว เมื่อเซลล์มาถึงที่อินพุตตัวดีมัลติเพลกซ์จะทำการเลือกเลเยอร์ที่จะส่งเซลล์ไป และเนื่องจากเซลล์ต่าง ๆ ที่มาจากอินพุตมีความเร็วเท่ากับ R ถูกกระจายไป k เส้นทาง ทำให้แต่ละเส้นทางจะต้องทำงานที่ความเร็วไม่ต่ำกว่า R/k

ในแต่ละเลเยอร์ อาจจะเป็นสวิตช์แบบ OQ หรือแบบ CIOQ ที่มีความเร็วในการทำงานของหน่วยความจำน้อยกว่า R เมื่อแต่ละเลเยอร์ได้รับเซลล์จากอินพุตต่าง ๆ ก็จะทำการส่งผ่านเซลล์ไปยังพอร์ตเอาต์พุต ถ้าในขณะนั้นเกิดความคับคั่งขึ้นเซลล์ก็จะถูกเก็บไว้ในบัฟเฟอร์ของสวิตช์

ตอนกลางก่อน คอยจนกระทั่งเส้นทางที่จะไปยังตัวมัลติเพลกซ์ว่างลง จึงจะทำการส่งออกเซลล์จากสวิตช์ตอนกลางไปยังตัวมัลติเพลกซ์นั้น ตัวมัลติเพลกซ์จะทำการเลือกเซลล์ที่มาจากแต่ละเลเยอร์ส่งออกจากสวิตช์ ซึ่งบัฟเฟอร์ของสวิตช์แต่ละเลเยอร์จะต้องมีหน่วยความจำที่ทำงานด้วยความเร็วไม่น้อยกว่า R/k



รูปที่ 3.5 สถาปัตยกรรมสวิตช์แบบ PPS [2]

3.5 สรุปท้ายบท

การที่เราจะทำนายการเติบโตของโครงข่ายอินเทอร์เน็ตเป็นการล่วงหน้านั้น เป็นไปได้ยากยิ่งว่าจะเป็นไปในทิศทางใด แต่ก็มีสิ่งหนึ่งที่เห็นได้อย่างชัดเจนว่า สวิตช์แพ็กเกตความเร็วสูงที่จะนำมาใช้ในอนาคตอันใกล้นั้น มีความต้องการในการออกแบบ ดังต่อไปนี้

1) เพิ่มความจุในการสวิตช์
2) สนับสนุนแพ็กเกตที่มีความเร็วสูงขึ้น
3) สนับสนุนการรับประกันคุณภาพการให้บริการ (QoS) ของทราฟฟิกแต่ละประเภทที่แตกต่างกัน (เช่น เสียง ภาพ วิดีโอ และ/หรือ มัลติมีเดีย)
4) สนับสนุนการเพิ่มขนาดของโครงข่าย (scalability)

ซึ่งความต้องการในการออกแบบเหล่านี้ ได้ท้าทายนักวิจัยทั้งหลายที่จะจัดการแก้ปัญหา เช่นสวิตช์ที่มีความจุในการสวิตช์ที่สูงขึ้นจะต้องอาศัยสถาปัตยกรรมแบบใหม่ ๆ ความเร็วแพ็กเกตที่สูงขึ้นเรื่อย ๆ จนเกินขีดความสามารถของหน่วยความจำของสวิตช์ที่มีอยู่และความต้องการใน

การสนับสนุนค่า QoS ของทราฟฟิกหลายระดับ อาจจะต้องนำเอาสมรรถนะที่ดีของสวิตช์แบบ OQ มาประยุกต์ใช้ (ตัวอย่างเช่น สวิตช์แบบ CIOQ) ส่วนการสนับสนุนการเพิ่มขนาดของโครงข่าย อาจจะต้องนำเอาโครงสร้างสวิตช์แบบหลายสเตจมาใช้ (ตัวอย่างเช่น สวิตช์แบบ PPS) การนำเส้นใยแก้วนำแสงมาใช้ในโครงข่ายอินเทอร์เน็ตสำหรับส่งผ่านข้อมูล ทำให้เกิดปัญหาคอขวดในการสวิตช์ข้อมูลเกิดขึ้น ซึ่งเป็นปัญหาเร่งด่วนที่จะต้องแก้ไข การนำเอาสถาปัตยกรรมสวิตช์แบบ PPS หรือการนำเอาเทคนิคเอนเวโลปมาประยุกต์ใช้ก็สามารถแก้ปัญหาคอขวดได้ในระดับหนึ่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
อัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์

เนื่องจากในปัจจุบัน โครงข่ายอินเทอร์เน็ตความเร็วสูงโดยเฉพาะที่เป็นโครงข่ายแบ็กโบนประกอบไปด้วยสวิตช์แพ็กเกตหรือเราเตอร์ไอพีความเร็วสูง ซึ่งได้เพิ่มจำนวนขึ้นอย่างรวดเร็ว โดยส่วนใหญ่แล้วโครงสร้างการสวิตช์ (switching fabric หรือ backplane) เป็นแบบครอสบาร์ กล่าวคือโครงสร้างการสวิตช์ถูกกำหนดโดยตัวจัดลำดับเซลล์ส่งผ่านสวิตช์แบบรวมศูนย์ (centralized scheduler) แพ็กเกตขนาดต่าง ๆ กันที่มาถึงอินพุตของสวิตช์จะถูกแบ่งออกเป็นเซลล์ส่งผ่านโครงสร้างการสวิตช์ส่วนกลาง และประกอบกันขึ้นใหม่อีกครั้งก่อนออกจากสวิตช์ การที่เรานิยมใช้โครงสร้างการสวิตช์เป็นแบบครอสบาร์ ก็เพราะว่านำไปสร้างเป็นฮาร์ดแวร์ได้ง่าย และเป็นสวิตช์ที่ไม่มีการติดขัดภายใน (internally non-blocking) ซึ่งยอมให้เซลล์มากกว่าหนึ่งเซลล์ได้รับการส่งผ่านโครงสร้างการสวิตช์พร้อมกันในเวลาเดียวกันได้ ทำให้ความคับคั่งที่เกิดขึ้นในโครงข่ายลดน้อยลง

สวิตช์แบบครอสบาร์นั้นส่วนใหญ่จะใช้การรอคิวที่อินพุต โดยที่เมื่อเซลล์มาถึงอินพุตของสวิตช์ จะถูกนำไปเก็บไว้ในบัฟเฟอร์โดยทันทีจนกว่าจะได้รับการส่งผ่านโครงสร้างการสวิตช์ จากบทความวิจัยของ Karol et al. ที่ตีพิมพ์ในปี 1987 [7] ทำให้เป็นที่รู้จักกันโดยทั่วไปว่า สวิตช์แบบรอคิวที่อินพุตขนาด NxN กับการจัดการรอคิวในบัฟเฟอร์แบบมาถึงก่อนได้รับบริการก่อน (FIFO: first-in-first-out) จะทำให้ได้ค่าทรูพุตสูงสุดจำกัดอยู่ที่ประมาณ 58.6 เปอร์เซ็นต์ที่ทราฟฟิกมีค่าสูงสุด เมื่อ N มีค่าเข้าสู่อนันต์ ซึ่งจะเป็นจริงได้ภายใต้เงื่อนไข ดังต่อไปนี้

1) การมาถึงของแพ็กเกตที่แต่ละอินพุต จะต้องมีการแจงแจงที่เหมือนกันและเป็นอิสระต่อกัน

2) กระบวนการการมาถึงที่แต่ละอินพุตนั้น เป็นอิสระจากอินพุตอื่น ๆ

3) กระบวนการการมาถึงนั้นจะต้องมีอัตราการมาถึงเดียวกัน และจุดหมายปลายทางที่จะไปยังทุก ๆ เอาต์พุตนั้น มีการแจกแจงแบบยูนิฟอร์ม

4) แพ็กเกตที่มาถึง ถูกแบ่งให้มีขนาดคงที่และเท่ากัน เรียกว่า เซลล์

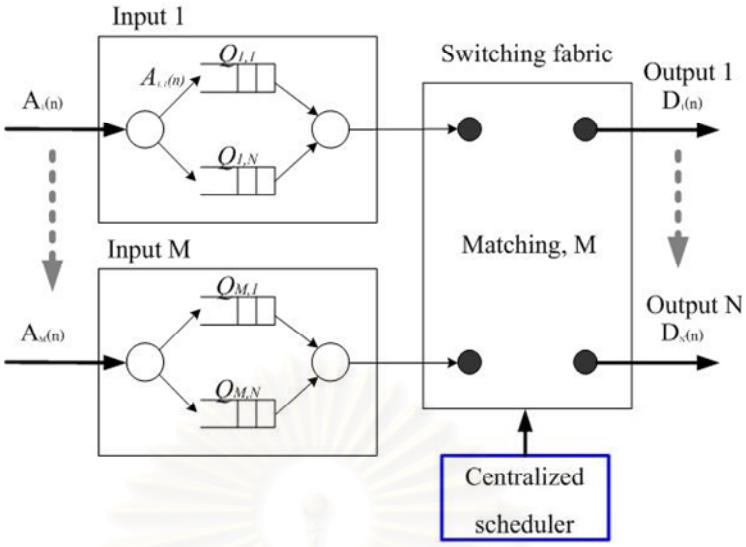
5) ค่า N มีขนาดใหญ่

เมื่อเงื่อนไขตามข้อ 1) และ ข้อ 2) เป็นจริง เราจะกล่าวว่าการมาถึงนั้นเป็นอิสระต่อกัน (independent) และเมื่อเงื่อนไขตามข้อ 3) เป็นจริงด้วย เราจะกล่าวว่าการมาถึงนั้นเป็นแบบยูนิฟอร์ม (uniform)

ค่าทรูพุตสูงสุดถูกจำกัด เพราะว่าเซลล์ที่อยู่ในลำดับหลังเซลล์ที่หัวคิวจะไปยังเอาต์พุตที่ต่างกันไม่สามารถส่งออกได้แม้ว่าเอาต์พุตนั้นจะว่างอยู่ ปรากฏการณ์นี้เรียกว่า การติดขัดที่หัวคิว (HoL blocking) ซึ่งจะเกิดขึ้นในกรณีที่สวิตช์มีการจัดการรอคิวแบบ FIFO และมีการแนะนำเทคนิคต่าง ๆ สำหรับลดการติดขัดที่หัวคิว โดยการใช้การรอคิวแบบไม่เป็น FIFO ตัวอย่างเช่น ให้พิจารณาเซลล์ที่อยู่ในลำดับหลังเซลล์ที่หัวคิว สำหรับส่งผ่านสวิตช์ได้ถึง k เซลล์ โดยแท้จริงแล้วการติดขัดที่หัวคิวสามารถขจัดให้หมดไปได้โดยสิ้นเชิงโดยใช้กลวิธีการจัดวางบัฟเฟอร์แบบง่าย ๆ ในแต่ละพอร์ตอินพุตเราจะไม่ใช้การรอคิวแบบ FIFO สำหรับทุก ๆ เซลล์เพียงชุดเดียว แต่จะใช้การรอคิวโดยแยกคิวที่จะไปยังแต่ละเอาต์พุตออกจากกันเป็นจำนวน N ชุด ดังรูปที่ 1 หลักการรอคิวแบบนี้เรียกว่า *การรอคิวเอาต์พุตเสมือน* (Virtual Output Queueing: VOQ) การติดขัดที่หัวคิวถูกกำจัดออกไปก็เพราะว่า เซลล์ที่รอคิวอยู่ในลำดับหลังเซลล์ที่จะไปยังเอาต์พุตเดียวกัน ไม่มีเซลล์ใดในบัฟเฟอร์ชุดเดียวกันจะไปยังเอาต์พุตที่ต่างกัน ทำให้เมื่อเรานำเอาการรอคิวแบบ VOQ มาใช้ จึงเป็นไปได้ว่าค่าทรูพุตสูงสุดจะเพิ่มขึ้นจากประมาณ 58 เปอร์เซ็นต์ ไปจนถึง 100 เปอร์เซ็นต์ ทั้งกรณีทราฟฟิกประเภทยูนิฟอร์มและนอนยูนิฟอร์ม

แต่การจัดการรอคิวแบบนี้ ทำให้การนำไปสร้างเป็นสวิตช์มีความซับซ้อนขึ้น เพราะจะต้องใช้การรอคิวแบบ FIFO จำนวน N ชุดในแต่ละอินพุต แต่เนื่องจากไม่ต้องการให้มีการเพิ่มความเร็วในการสวิตช์ ในแต่ละไทม์สล็อตจึงมีเซลล์ไม่เกินหนึ่งเซลล์มาถึงหรือออกจากบัฟเฟอร์ที่อินพุต ในระหว่างแต่ละไทม์สล็อตอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์จะทำการตัดสินใจในการเลือกเซลล์ที่หัวคิวที่เข้าแข่งขันเพื่อให้ได้รับการส่งผ่านโครงสร้างการสวิตช์ และกำหนดโครงสร้างการสวิตช์โดยการหาค่าการเชื่อมต่อหรือการแมตช์ (matching) บนกราฟไบพาร์ไทท์ เทคนิคต่าง ๆ ที่ถูกนำมาใช้สำหรับหาค่าการแมตช์ เช่น โครงข่ายนิวรอล (neural network) หรือ การแมตช์ด้วยการวนซ้ำ (iterative matching) เป็นต้น อัลกอริทึมเหล่านี้ ถูกออกแบบมาเพื่อให้ได้ค่าทรูพุตที่สูงขึ้น ขณะเดียวกันก็ยังสามารถนำไปสร้างเป็นฮาร์ดแวร์ได้ง่าย เมื่อทราฟฟิกเป็นประเภทยูนิฟอร์ม อัลกอริทึมเหล่านี้ทำงานได้ดี (ให้ค่าทรูพุตสูงกว่า 90 เปอร์เซ็นต์) ตัวอย่างเช่น อัลกอริทึม iSLIP จากผลการจำลองแบบแสดงให้เห็นว่าให้ค่าทรูพุตสูงสุดถึงเกือบ 100 เปอร์เซ็นต์ เมื่อทราฟฟิกเป็นประเภทยูนิฟอร์มและเป็นอิสระต่อกัน [11], [12] แต่อย่างไรก็ตาม อัลกอริทึมเหล่านี้ทำงานได้เลวลงและไม่สามารถสนับสนุนค่าทรูพุตสูงสุดได้เมื่อทราฟฟิกเป็นประเภทนอนยูนิฟอร์ม

คำถามจึงมีอยู่ว่า อะไรคือค่าทรูพุตสูงสุดที่สามารถเป็นไปได้ โดยใช้สวิตช์แบบรอคิวที่อินพุตและการจัดการคิวแบบ VOQ ดังรูปที่ 4.1



รูปที่ 4.1 แบบจำลองของสวิตช์แบบรอคิวที่อินพุต [13]

ตามรูปเป็นแบบจำลองสวิตช์แบบรอคิวที่อินพุตที่ใช้การรอคิวแบบ VOQ ซึ่งประกอบไปด้วยองค์ประกอบ 3 องค์ประกอบ คือ โครงสร้างการสวิตช์แบบไม่ติดขัด (เช่น แบบครอสบาร์) ตัวจัดลำดับเซลล์แบบรวมศูนย์ และพอร์ตอินพุตและเอาต์พุต ในส่วนต่อไปเราจะแสดงให้เห็นว่าสำหรับการมาถึงของเซลล์ที่เป็นอิสระต่อกัน (ทั้งแบบยูนิฟอร์มและนอนยูนิฟอร์ม) จะให้ค่าทรูพุตสูงสุดสูงถึง 100 เปอร์เซ็นต์ โดยใช้อัลกอริทึมแบบ MWM ซึ่งเป็นอัลกอริทึมที่คำนึงถึงระดับการรอคิวในบัฟเฟอร์ด้วย

4.1 หลักการพื้นฐาน

พิจารณาแบบจำลองของสวิตช์รอคิวที่อินพุตตามรูปที่ 4.1 ซึ่งประกอบไปด้วย M อินพุต และ N เอาต์พุต กำหนดให้ $A_i(n)$ เป็นกระบวนการมาถึงแบบสเตชันนารีและเออร์กอดิกที่อินพุต $i, 1 \le i \le M$ และเป็นกระบวนการที่ไม่ต่อเนื่องของแพ็กเกตที่มีขนาดคงที่ และเรียกว่าเซลล์ ที่จุดเริ่มต้นของแต่ละไทม์สล็อตจะมีเซลล์จำนวนศูนย์หรือหนึ่งเซลล์มาถึงที่แต่ละอินพุต แต่ละเซลล์จะมีตัวบ่งชี้ที่จะบ่งชี้เอาต์พุต $j, 1 \le j \le N$ คือเอาต์พุตที่เซลล์จะไปยังเอาต์พุตนั้น เมื่อมีเซลล์ที่จะไปยังเอาต์พุต j มาถึงที่อินพุต i ก็จะถูกเก็บไว้ในคิวแบบ FIFO ที่เรียกว่า VOQ $Q_{i,j}$ ดังนั้น เราจะนิยามเวกเตอร์การครอบครองคิวทุก ๆ คิว ที่ไทม์สล็อตที่ n ว่า

$$\underline{L}(n) = \left(L_{1,1}(n),...,L_{1,N}(n),...,L_{M,N}(n)\right)^T. \tag{4.1}$$

เช่นเดียวกันเราจะนิยาม $W_{i,j}(n)$ คือ จำนวนไทม์สล็อตที่ใช้ในการรอคิวของเซลล์ VOQ $Q_{i,j}$ ที่ไทม์สล็อตที่ n และนิยามเวกเตอร์ที่ใช้แทนเวลารอคิวของเซลล์ที่หัวคิวที่รอคิวแบบ VOQ ที่ไทม์สล็อตที่ n ว่า

$$\underline{W}(n)=\left(W_{1,1}(n),...,W_{1,N}(n),...,W_{M,N}(n)\right)^T. \tag{4.2}$$

เราจะนิยามกระบวนการการมาถึง $A_{i,j}(n)$ ว่าเป็นกระบวนการการมาถึงที่อินพุตที่ i และต้องการจะไปที่เอาต์พุตที่ j ที่อัตราการมาถึง $\lambda_{i,j}$ ดังนั้น เซตของกระบวนการการมาถึงทั้งหมดคือ $A(n)=\{A_i(n);1\le i\le M\}$ และเซตของ $A(n)$ จะถูกพิจารณาว่าตรงตามเงื่อนไข (admissible) ถ้าเป็นไปตามเงื่อนไขของสมการ ดังต่อไปนี้

$$\sum_i \lambda_{i,j}<1, \sum_j \lambda_{i,j}<1. \tag{4.3}$$

มิฉะนั้นแล้วเราจะกล่าวว่ามันไม่ตรงตามเงื่อนไข (inadmissible)

เพื่อพิจารณาหาค่าการแมตช์ M อัลกอริทึมจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์จะทำการแก้ปัญหาการแมตช์บนกราฟไบพาร์ไทท์ แสดงดังรูปที่ 4.2

ถ้า $Q_{i,j}$ นั้นไม่ว่าง $L_{i,j}(n)>0$ และมีด้านบนกราฟ G ระหว่างอินพุต i และเอาต์พุต j ซึ่งสัมพันธ์กับค่าน้ำหนัก $W_{i,j}(n)$ ในแต่ละด้าน สิ่งนี้หมายความว่าน้ำหนัก $W_{i,j}(n)$ ขึ้นอยู่กับอัลกอริทึมที่นำมาใช้ ดังนั้นเราจะพิจารณาอัลกอริทึม 2 แบบ ดังต่อไปนี้

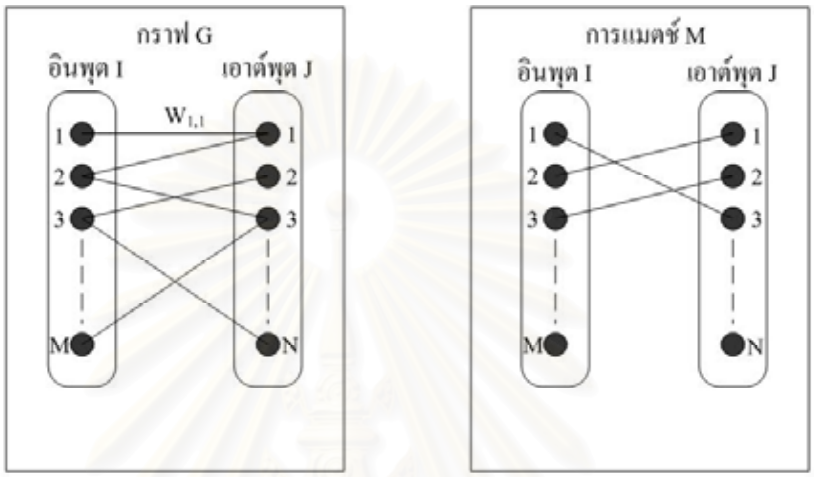
1) อัลกอริทึม MSM (Maximum Size Matching) อัลกอริทึมพยายามที่จะหาค่าการแมตช์ที่มีจำนวนด้านสูงที่สุด

2) อัลกอริทึม MWM (Maximum Weight Matching) อัลกอริทึมพยายามที่จะหาค่าการแมตช์น้ำหนักสูงที่สุด โดยที่เราจะพิจารณาเพียงกรณีที่น้ำหนัก $W_{i,j}(n)$ มีค่าเป็นเลขจำนวนเต็ม (integer) เท่ากับการครอบครองคิว $L_{i,j}(n)$ ของ $Q_{i,j}$ หรือเวลาในการรอคิว $W_{i,j}(n)$ ของเซลล์ที่หัวคิวในแต่ละ $Q_{i,j}$

เราจะเห็นได้อย่างชัดเจนว่าการแมตช์ขนาดสูงที่สุดนั้นก็คือ กรณีพิเศษของอัลกอริทึมการแมตช์น้ำหนักสูงที่สุด โดยที่ $w_{i,j}=1$ เมื่อ $Q_{i,j}$ นั้นไม่ว่าง

ดังรูปที่ 4.2 นิยามกราฟ G = [V,E] คือกราฟไม่มีทิศทาง ซึ่งเป็นกราฟที่มีการเชื่อมต่อเซตของจุดยอด V ด้วยเซตของด้าน E ด้านที่เชื่อมต่อจุดยอด $i,1\le i\le M$ และ $j,1\le j\le N$ มีความ

สัมพันธ์กับน้ำหนักที่แทนด้วยสัญลักษณ์ $w_{i,j}$ กราฟ G จะเป็นกราฟไบพาร์ไทท์ ถ้าเซตของอินพุต $I=\{i:1\leq i\leq M\}$ และ $J=\{j:1\leq j\leq N\}$ แบ่งเซต V ออกเป็นสองส่วน โดยที่ทุก ๆ ด้านจะมีปลายข้างหนึ่งอยู่ใน I และปลายอีกข้างหนึ่งอยู่ใน J การแมตช์ M บนกราฟ G ก็คือเซตย่อยของ E ซึ่งไม่มีด้าน 2 ด้านใน M ที่มีจุดยอดร่วมกัน อัลกอริทึมการแมตช์ขนาดสูงที่สุด ก็คือ การหาค่าการแมตช์ที่มีขนาดรวมสูงที่สุด (total size) หรือมีน้ำหนักรวมสูงที่สุด (total weight)



(ก) ตัวอย่างของกราฟ G สำหรับ |I|=M และ |J|=N (ข) ตัวอย่างของการแมตช์ M บน กราฟ G

รูปที่ 4.2 กราฟ G และการแมตช์ M บน G [13]

4.2 การแมตช์ขนาดสูงที่สุด

การแมตช์ขนาดสูงที่สุดหรือเรียกอย่างย่อว่า maxsize สำหรับกราฟไบพาร์ไทท์ หาได้จากการแก้ปัญหาการไหลในโครงข่ายสมมูล มีอัลกอริทึมอยู่หลายแบบสำหรับการแก้ปัญหานี้ แต่อัลกอริทึมที่มีประสิทธิภาพสูงที่สุดที่รู้จักกันในปัจจุบันนี้ ลู่เข้าสู่ค่า $O(N^{5/2})$

ปัญหาของอัลกอริทึมก็คือ ถึงแม้ว่าจะรับประกันได้ว่าจะหาค่าการแมตช์ขนาดสูงที่สุดได้ แต่ในการนำไปประยุกต์ใช้งานนั้น มีความซับซ้อนมากเกินไปในการนำไปสร้างเป็นสวิตช์ความเร็วสูง นอกจากนี้ยังใช้เวลาในการประมวลผลนานเกินไป

คำถามมีอยู่ว่า อัลกอริทึมขนาดสูงที่สุดนั้น ให้ค่าทรูพุตของสวิตช์สูงที่สุดจริงหรือไม่ คำตอบก็คือไม่ นอกจากนี้ อัลกอริทึมยังทำให้เซลล์ที่รออยู่ในคิวเกิดสภาวะสทาร์เวชัน (starvation) หรือสภาวะที่เซลล์รออยู่ในคิวเป็นเวลานานโดยไม่มีกำหนดเวลาว่าจะได้รับบริการเมื่อใด และในกรณีที่ทราฟฟิกเป็นประเภทนอนยูนิฟอร์มด้วยแล้วก็ทำให้ค่าทรูพุตสูงสุดลดลง สิ่งนี้ก็เนื่องมาจากการไม่ได้ทำการพิจารณาระดับของการรอคิวในบัฟเฟอร์

ในทางปฏิบัติสำหรับระบบที่ต้องการสวิตช์ที่มีสมรรถนะสูงนั้น เราจึงมีความต้องการในการออกแบบอัลกอริทึมที่มีคุณสมบัติ ดังต่อไปนี้

- ให้ค่าทรูพุตสูงสุด (High throughtput) อัลกอริทึมที่ดีควรมีจำนวนเซลล์ที่รอคิวอยู่ในบัฟเฟอร์ต่ำ ทำให้ในกรณีที่มีการรอคิวแบบ VOQ จะทำให้อัลกอริทึมสามารถสนับสนุนค่าทรูพุตสูงสุดได้ถึง 100 เปอร์เซ็นต์ในแต่ละคู่อินพุตเอาต์พุต

- อิสระจากสภาวะสทาร์เวชัน (Starvation free) อัลกอริทึมที่ดีไม่ควรมีลักษณะที่มีเซลล์รออยู่ในคิวเป็นเวลานานโดยไม่มีกำหนดเวลาว่าจะได้รับบริการเมื่อใด

- รวดเร็ว (Fast) เพื่อให้ค่าแบนด์วิดท์ในการสวิตช์มีค่าสูงสุด จึงเป็นเรื่องสำคัญที่อัลกอริทึมจะต้องไม่ทำให้เกิดปัญหาคอขวดในการสวิตช์ข้อมูล โดยอัลกอริทึมจะต้องประมวลผลได้อย่างรวดเร็วในการหาการเชื่อมต่อระหว่างอินพุตและเอาต์พุต

- ง่ายในการนำไปสร้างเป็นสวิตช์ (Simple to implement) ถ้าอัลกอริทึมสามารถประมวลผลได้อย่างรวดเร็ว ก็ควรจะสามารถนำไปสร้างเป็นฮาร์ดแวร์ที่มีวัตถุประสงค์พิเศษได้ง่ายภายในชิปเดียว

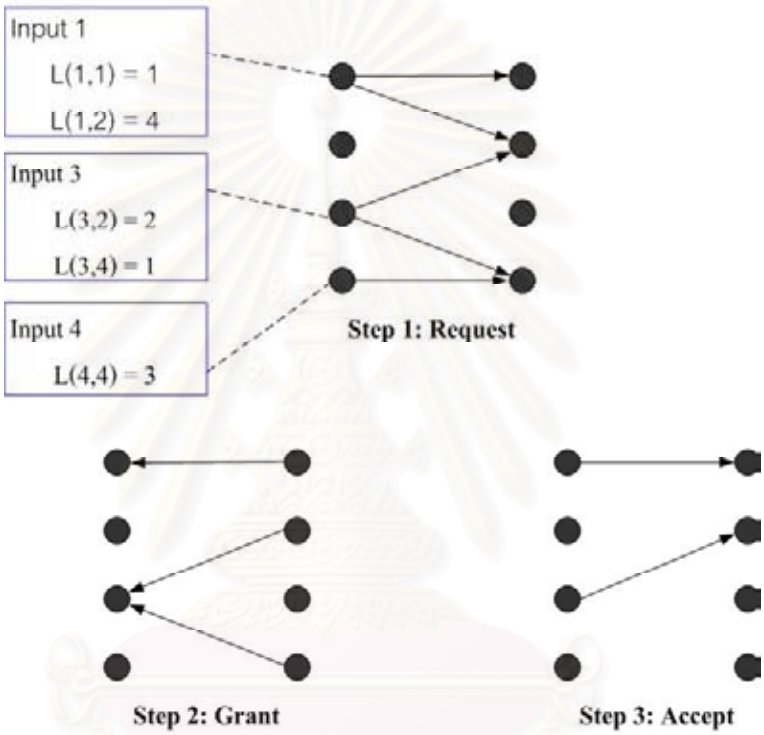
4.2.1 อัลกอริทึม PIM

อัลกอริทึม PIM (Parallel Iterative Matching) ได้รับการพัฒนาขึ้นจากศูนย์วิจัยระบบของบริษัท DEC สำหรับสวิตช์ขนาด 16 พอร์ต แบบ AN 2 อัลกอริทึม PIM ใช้หลักการพิจารณาเลือกเซลล์ที่เข้าแข่งขันเพื่อส่งผ่านโครงสร้างการสวิตช์แบบสุ่ม ทั้งนี้เพื่อหลีกเลี่ยงปัญหาสทาร์เวชันและลดจำนวนการวนซ้ำ ซึ่งจะทำให้ได้การเชื่อมต่อขนาดมากที่สุด การหาการเชื่อมต่อขนาดมากที่สุดนั้นได้มาจากการเพิ่มการเชื่อมต่อในแต่ละรอบของการวนซ้ำ โดยไม่ต้องยกเลิกกระบวนการเชื่อมต่อในรอบก่อนหน้านั้น โดยทั่วไปแล้ว การเชื่อมต่อขนาดมากที่สุดมีค่าน้อยกว่าการเชื่อมต่อขนาดสูงที่สุด แต่นำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายกว่า อัลกอริทึม PIM สามารถประมวลผลได้อย่างรวดเร็วในการลู่เข้าสู่ค่าการเชื่อมต่อขนาดมากที่สุดในการวนซ้ำ โดยที่ในการวนซ้ำแต่ละรอบจะประกอบไปด้วยขั้นตอน 3 ขั้นตอน แสดงดังรูปที่ 4.3 เริ่มต้นด้วยทุก ๆ อินพุตและเอาต์พุตของสวิตช์จะถูกนำมาพิจารณาเชื่อมต่อกัน และเมื่ออินพุตใดเชื่อมต่อกับเอาต์พุตแล้วจะไม่นำมาพิจารณาเชื่อมต่อในรอบต่อไป ในแต่ละรอบของการวนซ้ำจะมีขั้นตอน ดังต่อไปนี้

- ขั้นตอนที่ 1 *คำร้องขอ* (Request) ในแต่ละอินพุตของสวิตช์ที่มีเซลล์รออยู่ในคิว และยังไม่ได้รับการเชื่อมต่อ จะส่งคำร้องขอไปยังทุก ๆ เอาต์พุตที่มีเซลล์ที่อินพุตจะไปยังเอาต์พุตนั้น

- ขั้นตอนที่ 2 *คำยินยอม* (Grant) แต่ละเอาต์พุตที่ยังไม่ได้รับการเชื่อมต่อ เมื่อได้รับคำร้องขอแล้ว จะส่งคำยินยอมไปยังอินพุตโดยทำการเลือกจากแต่ละอินพุตที่ร้องขอมาแบบสุ่มเพียงอินพุตเดียว

- ขั้นตอนที่ 3 *ตอบรับคำยินยอม* (Accept) ถ้าอินพุตใดได้รับคำยินยอมให้ส่งจากเอาต์พุตแล้ว จะเลือกตอบรับคำยินยอมจากเอาต์พุตแบบสุ่มเพียงเอาต์พุตเดียว เพื่อทำการเชื่อมต่อระหว่างอินพุตนั้นกับเอาต์พุตที่เลือก



รูปที่ 4.3 แสดงตัวอย่างขั้นตอนทั้ง 3 ขั้นตอนของอัลกอริทึม PIM ที่ไม่มีการวนซ้ำ [12]

จะเห็นว่าอินพุตที่ 4 กับเอาต์พุตที่ 4 ไม่ได้รับการเชื่อมต่อ แม้ว่าจะไม่เกิดการชนกันของเซลล์กับการเชื่อมต่ออื่น ๆ เลย แต่ก็จะได้รับการเชื่อมต่อถ้ามีการวนซ้ำในรอบที่ 2

สรุปได้ว่าตัวตัดสินใจ (arbiter) ของเอาต์พุตมีอิสระในการที่จะเลือกคำร้องขอเพียงคำร้องขอเดียว จากคำร้องขอของอินพุตต่าง ๆ ที่เข้าแข่งขันโดยวิธีการเลือกแบบสุ่ม ทำให้เกิดผลลัพธ์ 3 ประการคือ ประการแรก ในแต่ละรอบของการวนซ้ำ จะสามารถพิจารณาให้การเชื่อมต่อได้ถึงประมาณ 3 ใน 4 ของการเชื่อมต่อที่เป็นไปได้ทั้งหมด ซึ่งอัลกอริทึมนี้จะลู่เข้าสู่ค่าการเชื่อมต่อขนาดมากที่สุดเมื่อจำนวนรอบของการวนซ้ำมีค่าเฉลี่ยประมาณ $O(\log_2 N)$ ประการที่สอง สามารถยืนยันได้ว่าทุก ๆ คำร้องขอจากอินพุต ในที่สุดจะต้องได้รับคำยินยอมจากเอาต์พุต ซึ่งเป็นสิ่งที่ยืนยันได้ว่าจะไม่มีเซลล์ใดที่รออยู่ในคิวจนเกิดสภาวะสทาร์เวชัน ประการที่สามหมายความว่าไม่มีความจำเป็น

ที่จะต้องใช้หน่วยความจำในการเก็บสถานะการเชื่อมต่อที่เกิดขึ้นในรอบที่ผ่านมา กระบวนการเชื่อมต่อได้เริ่มขึ้นโดยไม่ขึ้นกับกระบวนการเชื่อมต่อในรอบที่ผ่านมา และเราสามารถทำความเข้าใจจากการวิเคราะห์สมรรถนะของอัลกอริทึมได้โดยตรง จึงไม่ต้องพิจารณาถึงสถานะของสวิตช์ที่เปลี่ยนแปลงตามเวลา เพียงแต่จะต้องพิจารณาการครอบครองคิวที่อินพุตของสวิตช์เท่านั้น

แต่การเลือกแบบสุ่มทำให้เกิดปัญหาคือ ประการแรก ทำให้เกิดความยุ่งยากและใช้ต้นทุนสูงในการนำไปสร้างเป็นสวิตช์ความเร็วสูง เนื่องจากตัวตัดสินใจแต่ละตัวจะต้องทำการเลือกอินพุตแบบสุ่มซึ่งจำเป็นต้องใช้เวลา ประการที่สอง เกิดความไม่เท่าเทียมกัน (fairness) ในการเลือกในกรณีที่สวิตช์มีทราฟฟิกเป็นประเภทนอนยูนิฟอร์ม และประการสุดท้าย อัลกอริทึม PIM ไม่เหมาะสมในการนำไปใช้ถ้าไม่มีการวนซ้ำ นอกจากนี้ ขีดจำกัดของค่าทรูพุตสูงสุดของสวิตช์มีค่าประมาณ 63 เปอร์เซ็นต์ ซึ่งสูงกว่าอัลกอริทึม FIFO เพียงเล็กน้อย ทั้งนี้เนื่องมาจากความน่าจะเป็นที่แต่ละอินพุตจะไม่ได้รับการเลือกจากเอาต์พุตคือ $((N-1)/N)^N$ และเมื่อ N มีค่าเพิ่มขึ้น ค่าทรูพุตมีแนวโน้มลู่เข้าสู่ค่า $1 - 1/e \approx 63$ เปอร์เซ็นต์ ถึงแม้ว่าอัลกอริทึม PIM จะทำการเชื่อมต่อได้มากขึ้นหลังจากการวนซ้ำหลาย ๆ รอบ แต่ก็ต้องใช้เวลาในการประมวลผลมาก จึงมีผลกระทบต่อความเร็วในการทำงานของสวิตช์ ดังนั้น เราจึงมีความต้องการอัลกอริทึมที่จะทำงานได้ดีแม้ว่าจะไม่มีการวนซ้ำ

4.2.2 อัลกอริทึม RRM

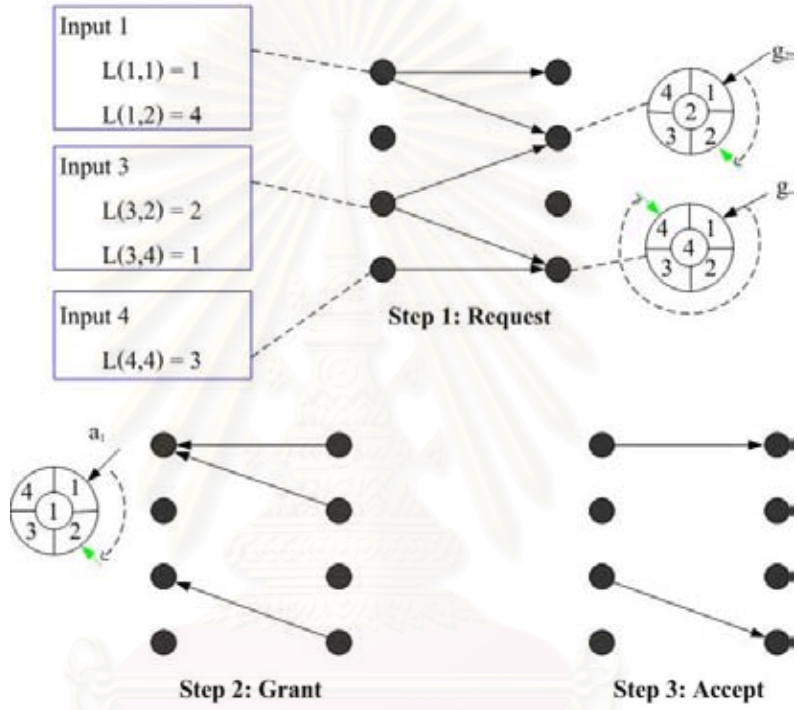
อัลกอริทึม RRM (Round-Robin Matching) เป็นอัลกอริทึมที่ใช้การวนรอบอินพุตแต่ละอินพุตของสวิตช์แบบที่ง่ายที่สุด ดังนั้น จะทำการเปรียบเทียบอะเรย์สองมิติของตัวตัดสินใจของอัลกอริทึม RRM โดยที่แต่ละเซลล์จะถูกจัดลำดับโดยวิธีการวนรอบแต่ละอินพุตและการวนรอบแต่ละเอาต์พุต อัลกอริทึม RRM มีสมรรถนะการทำงานไม่ดีนักแต่ก็สามารถแก้ปัญหาความซับซ้อนและความไม่เท่าเทียมกัน (unfairness) ในการเลือกการเชื่อมต่อที่เกิดขึ้นในอัลกอริทึม PIM ได้ นอกจากนี้ อัลกอริทึมยังสามารถนำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายกว่า และทำงานได้รวดเร็วกว่าการใช้ตัวตัดสินใจเลือกเซลล์แบบสุ่มของอัลกอริทึม PIM อัลกอริทึม RRM ก็เป็นเช่นเดียวกับอัลกอริทึม PIM ประกอบไปด้วย 3 ขั้นตอน แสดงดังรูปที่ 4.4 โดยมีรายละเอียด ดังต่อไปนี้

- ขั้นตอนที่ 1 *คำร้องขอ* แต่ละอินพุตที่มีเซลล์รออยู่ในคิวและต้องการจะไปยังเอาต์พุตนั้นส่งคำร้องขอไปยังเอาต์พุตนั้น

- ขั้นตอนที่ 2 *คำยินยอม* เมื่อแต่ละเอาต์พุตได้รับการร้องขอ จะทำการเลือกเพียงคำร้องขอเดียวที่ปรากฏว่ามีค่าลำดับความสำคัญสูงสุด โดยทำการวนรอบแต่ละอินพุตซึ่งมีตัวบ่งชี้ g_j

ทำการเลือกคำร้องขอแล้วส่งคำยินยอมไปยังอินพุตนั้น และชี้อินพุตต่อไป (modulo N) โดยกำหนดให้เป็นอินพุตที่มีค่าลำดับความสำคัญสูงสุดตามลำดับ

- ขั้นตอนที่ 3 *ตอบรับคำยินยอม* เมื่อแต่ละอินพุตได้รับคำยินยอมแล้วจะเลือกตอบรับคำยินยอมของเอาต์พุต โดยทำการเลือกเอาต์พุตที่มีค่าลำดับความสำคัญสูงสุด โดยมีตัวบ่งชี้ a_i วนรอบแต่ละเอาต์พุต และชี้เอาต์พุตถัดไป (modulo N) โดยกำหนดให้เอาต์พุตนั้นมีค่าลำดับความสำคัญสูงสุด เมื่อได้เลือกเอาต์พุตนั้นและทำการเชื่อมต่อกับอินพุตแล้ว



รูปที่ 4.4 แสดงตัวอย่างการทำงานของอัลกอริทึม RRM [12]

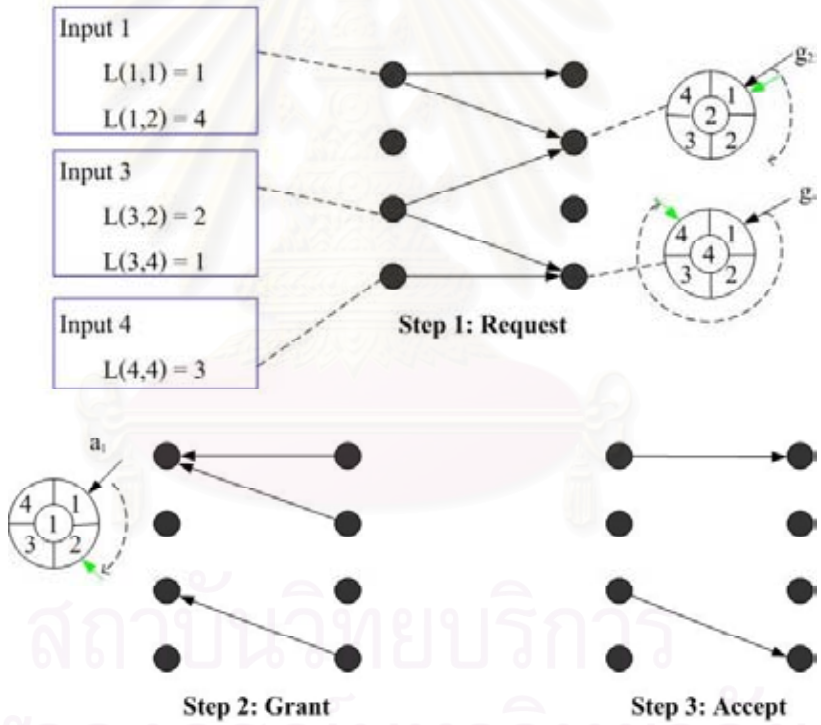
4.2.3 อัลกอริทึม iSLIP

อัลกอริทึม iSLIP ใช้หลักการวนรอบอินพุตแต่ละอินพุต (round robin) เพื่อทำการตัดสินใจที่จะจัดลำดับอินพุตและเอาต์พุตในแต่ละรอบของการวนซ้ำ คุณลักษณะที่สำคัญของอัลกอริทึมก็คือ สามารถนำไปสร้างเป็นฮาร์ดแวร์ได้ง่ายและทำงานได้ดีที่การสวิตช์ความเร็วสูง จาก [12] แสดงให้เห็นว่า สมรรถนะของอัลกอริทึมสำหรับทราฟฟิกประเภทยูนิฟอร์มมีค่าสูงมาก ถึงแม้ว่าจะไม่มีการวนซ้ำก็ให้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ และเนื่องจากการที่มีการวนซ้ำจึงทำให้ตัวตัดสินใจของอัลกอริทึมแต่ละตัว มีแนวโน้มไม่ซิงโครไนซ์กับตัวตัดสินใจอื่น ๆ

(ก) อัลกอริทึม iSLIP แบบไม่วนซ้ำ

อัลกอริทึม iSLIP ได้รับการพัฒนามาจากอัลกอริทึม RRM โดยอัลกอริทึม iSLIP จะไม่เลื่อนค่าตัวบ่งชี้ g_j ถ้าคำยินยอมที่ส่งไปให้อินพุตไม่ได้รับการตอบรับจากอินพุตนั้น จะเห็นว่าอัลกอริทึม iSLIP ก็คล้ายกันกับอัลกอริทึม PIM เว้นแต่เงื่อนไขในการเลื่อนค่าของตัวบ่งชี้ g_j ในขั้นตอนที่ 2 ที่เปลี่ยนแปลงไป ดังนี้

- ขั้นตอนที่ 2 *คำยินยอม* เมื่อแต่ละเอาต์พุตได้รับคำร้องขอแล้ว จะทำการเลือกคำร้องขอเพียงคำร้องขอเดียว ที่ปรากฏว่ามีค่าลำดับความสำคัญสูงสุด โดยการวนรอบแต่ละอินพุต โดยมีตัวบ่งชี้ g_j ทำการเลือกแล้วส่งคำยินยอมไปยังอินพุตนั้น และจะชี้อินพุตถัดไปโดยกำหนดให้อินพุตนั้นมีค่าลำดับความสำคัญสูงสุดในรอบต่อไป ก็ต่อเมื่อคำยินยอมนั้นได้รับการตอบรับ และทำการเชื่อมต่อระหว่างอินพุตและเอาต์พุต ในขั้นตอนที่ 3 แล้ว



รูปที่ 4.5 แสดงตัวอย่างการทำงานของอัลกอริทึม iSLIP

คุณสมบัติของอัลกอริทึม iSLIP แบบไม่วนซ้ำ มีดังต่อไปนี้

- ข้อที่ 1 ให้ค่าลำดับความสำคัญต่ำสุดสำหรับการเชื่อมต่อที่เป็นปัจจุบันที่สุด เนื่องจากเมื่อตัวตัดสินใจของเอาต์พุตเลื่อนค่าบ่งชี้อินพุตที่ตอบรับคำยินยอมของเอาต์พุตใด และมีการเชื่อมต่อระหว่างอินพุตและเอาต์พุตนั้นแล้ว ตัวบ่งชี้ g_j จะให้ค่าลำดับความสำคัญของ

อินพุตนั้นต่ำสุด กล่าวคือถ้าอินพุต i เชื่อมต่อกับเอาต์พุต j ได้สำเร็จ ทั้ง a_i และ g_j จะเลื่อนค่าบ่งชี้ไปหนึ่งค่า (modulo N) ทำให้การเชื่อมต่อระหว่างอินพุต i และเอาต์พุต j จึงกลายเป็นการเชื่อมต่อที่มีลำดับความสำคัญต่ำสุดในไทม์สล็อตต่อไป

- ข้อที่ 2 ไม่มีการเชื่อมต่อใดเกิดสภาวะสทาร์เวชัน เนื่องจากแต่ละอินพุตจะส่งคำร้องขอต่อเนื่องกันไป จนกว่าจะได้รับการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตนั้น

- ข้อที่ 3 ในกรณีที่ทราฟฟิกหนาแน่น ทุก ๆ คิวที่จะไปยังเอาต์พุตเดียวกัน จะให้ค่าทรูพุตที่เหมือนกัน ซึ่งเป็นผลที่ตามมาจากคุณสมบัติข้อที่ 2 เนื่องจากตัวบ่งชี้ของเอาต์พุต เลื่อนค่าอินพุตในตัวบ่งชี้ต่อเนื่องกันไปเป็นลำดับ จึงทำให้แต่ละอินพุตให้ค่าทรูพุตที่เหมือนกัน

(ข) อัลกอริทึม iSLIP แบบวนซ้ำ

การที่มีการวนซ้ำมากกว่าหนึ่งรอบ จะทำให้อัลกอริทึม iSLIP ให้ค่าการแมตช์ขนาดมากที่สุดที่มีขนาดมากขึ้น เนื่องจากในแต่ละรอบการวนซ้ำได้เพิ่มการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตที่ไม่ได้ทำการเชื่อมต่อในรอบก่อน จึงเป็นสิ่งที่เราสามารถจะคาดการณ์ได้ว่าสมรรถนะของสวิตช์จะต้องดีขึ้นเมื่อเพิ่มจำนวนรอบในการวนซ้ำให้มากขึ้น อินพุตที่เชื่อมต่อกับเอาต์พุตในรอบปัจจุบันแล้วจะไม่ถูกนำมาพิจารณาเชื่อมต่ออีกในรอบต่อไป และเพื่อให้เกิดความเท่าเทียมกันในการเลือกเซลล์ส่งผ่านสวิตช์ในแต่ละอินพุต จึงจะทำการเลื่อนค่าของตัวบ่งชี้เฉพาะในรอบแรกของการวนซ้ำเท่านั้น

คุณสมบัติของอัลกอริทึม iSLIP ที่มีการวนซ้ำ มีดังต่อไปนี้

- ข้อที่ 1 ให้ค่าลำดับความสำคัญต่ำสุด สำหรับการเชื่อมต่อที่เป็นปัจจุบันที่สุด ในการวนซ้ำรอบแรกในแต่ละไทม์สล็อต

- ข้อที่ 2 ไม่มีการเชื่อมต่อใดเกิดสภาวะสทาร์เวชัน เพราะว่าจะได้รับการเชื่อมต่อในรอบต่อไป จนกว่าทุก ๆ อินพุตที่มีเซลล์อยู่ในหัวคิว จะได้รับการเชื่อมต่อสำเร็จ

- ข้อที่ 3 ในกรณีที่มีการวนซ้ำมากกว่าหนึ่งรอบ กรณีที่ทราฟฟิกหนาแน่น จะทำให้แต่ละคิวที่จะไปยังเอาต์พุตเดียวกัน มีค่าทรูพุตที่แตกต่างกัน

- ข้อที่ 4 อัลกอริทึมจะมีจำนวนรอบของการวนซ้ำมีค่าไม่เกิน N รอบ เพราะว่าจะต้องมีอย่างน้อยหนึ่งการเชื่อมต่อที่ได้กระทำในแต่ละรอบ ถ้าไม่มีการเชื่อมต่อในรอบใด ก็จะไม่มีการวนซ้ำในรอบต่อไป ทำให้จำนวนการวนซ้ำลู่เข้าสู่ค่า N

- ข้อที่ 5 อัลกอริทึมไม่มีความจำเป็นที่จะต้องลู่เข้าสู่ค่าการเชื่อมต่อขนาดสูงที่สุด แต่ทางที่ดีที่สุดก็คือ การหาค่าการเชื่อมต่อขนาดมากที่สุด (maximal sized match) โดยไม่จำเป็นที่จะต้องยกเลิกการเชื่อมต่อที่ได้ทำไปแล้วในการวนซ้ำในรอบก่อน

4.2.4 อัลกอริทึม DRRM

รูปแบบของอัลกอริทึม DRRM (Dual Round-Robin matching) นั้นมีลักษณะการทำงานคล้ายคลึงกับอัลกอริทึม iSLIP แต่มีความซับซ้อนน้อยกว่าในการนำไปสร้างเป็นฮาร์ดแวร์ โดยอัลกอริทึม DRRM จะเริ่มต้นด้วยการเลือกแบบวนรอบที่พอร์ตอินพุต แทนที่จะเป็นที่พอร์ตเอาต์พุต ดังเช่นกรณีของอัลกอริทึม iSLIP ตัวตัดสินใจของพอร์ตอินพุต จะทำการเลือกอินพุตที่มีเซลล์รออยู่ในคิวโดยการวนรอบ เมื่อเลือกได้แล้วก็จะส่งคำร้องขอของอินพุตนั้นไปยังเอาต์พุต หลังจากนั้นตัวตัดสินใจของพอร์ตเอาต์พุต จะได้รับคำร้องขอจากอินพุตต่าง ๆ มากที่สุดถึงจำนวน N คำร้องขอ แต่ตัวตัดสินใจจะต้องเลือกเพียงหนึ่งคำร้องขอ โดยใช้วิธีการเลือกแบบวนรอบ และเมื่อเลือกได้แล้ว ก็จะส่งคำยินยอมไปยังอินพุตที่ได้รับเลือก ดังนั้น ในการพิจารณาเลือกเซลล์ส่งผ่านสวิตช์ของอัลกอริทึม DRRM ในแต่ละไทม์สล็อต จะมีขั้นตอนจำนวน 3 ขั้นตอน ดังต่อไปนี้

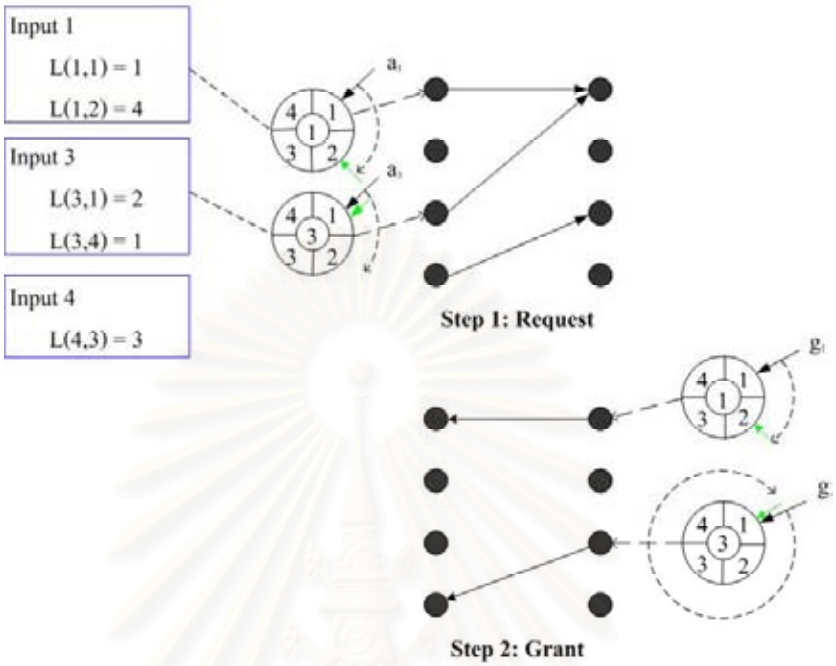
- ขั้นตอนที่ 1 *คำร้องขอ* แต่ละอินพุตที่มีเซลล์รออยู่ในคิว จะทำการเลือกส่งคำร้องขอเพียงคำร้องขอเดียวไปยังเอาต์พุตที่เลือกโดยวิธีการวนรอบ เมื่อปรากฏว่าตัวตัดสินใจของพอร์ตอินพุตเลือกเอาต์พุตที่มีค่าลำดับความสำคัญสูงสุด โดยมีตัวบ่งชี้ a_i วนรอบแต่ละเอาต์พุต

- ขั้นตอนที่ 2 *คำยินยอม* เมื่อแต่ละเอาต์พุตได้รับคำร้องขอ จะทำการเลือกเพียงคำร้องขอเดียว ที่ปรากฏว่ามีค่าลำดับความสำคัญของอินพุตสูงสุด โดยทำการวนรอบแต่ละอินพุต โดยมีตัวบ่งชี้ g_j วนรอบแต่ละอินพุต และชี้อินพุตถัดไป (modulo N) โดยกำหนดให้มีค่าลำดับความสำคัญสูงสุด เมื่อได้ส่งคำยินยอมไปยังอินพุตนั้นแล้ว

- ขั้นตอนที่ 3 *ตอบรับคำยินยอม* ตัวตัดสินใจของพอร์ตอินพุต จะชี้เอาต์พุตต่อไป (modulo N) เมื่อคำร้องขอได้รับการตอบรับโดยส่งคำยินยอมกลับมาในขั้นตอนที่ 2 เพื่อทำการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตของสวิตช์ในรอบแรก และจะกระทำซ้ำในขั้นตอนที่ 1 และขั้นตอนที่ 2 ถ้ามีการวนซ้ำมากกว่าหนึ่งรอบ

รูปที่ 4.6 แสดงตัวอย่างการทำงานของอัลกอริทึม DRRM ในขั้นตอนที่ 1 แต่ละอินพุตจะเลือก VOQ ที่มีเซลล์รออยู่ในคิว และส่งคำร้องขอไปยังเอาต์พุตที่เลือก สมมุติว่าอินพุตที่ 1 มีเซลล์ที่จะไปยังเอาต์พุตที่ 1 และ 2 แต่เนื่องจากตัวตัดสินใจเลือกแบบวนรอบซึ่งมีค่าตัวบ่งชี้ a_1 ชี้ที่เอาต์พุตที่ 1 ดังนั้นอินพุตที่ 1 จึงส่งคำร้องขอไปยังเอาต์พุตที่ 1 และเลื่อนค่าของตัวบ่งชี้ a_1 หลังจาก

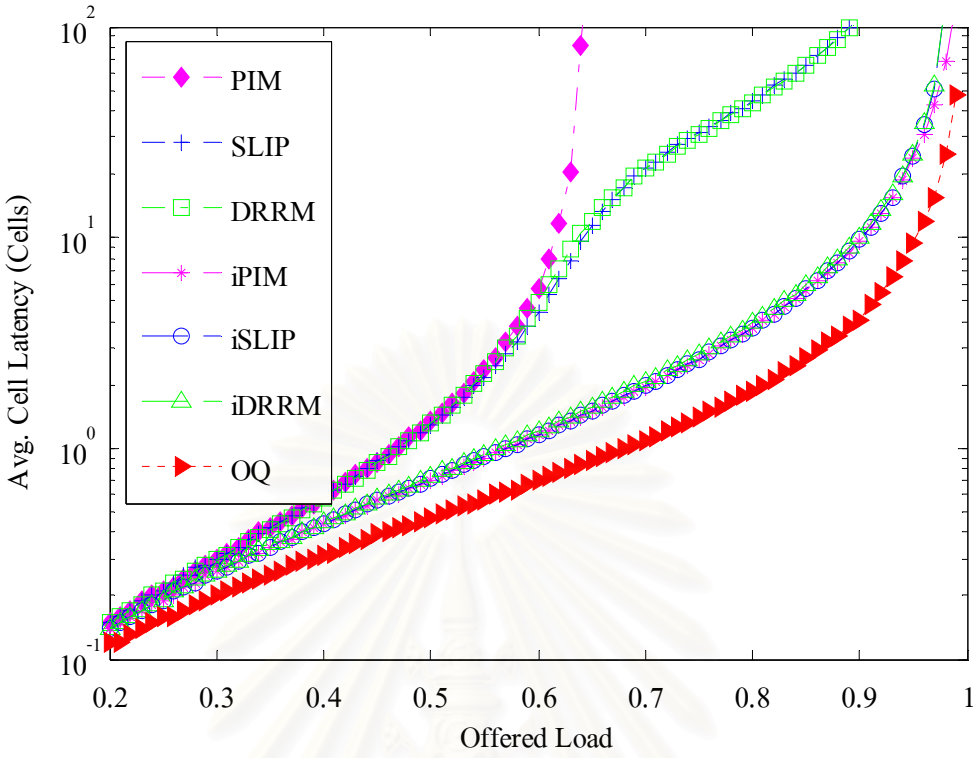
ทำการเชื่อมต่อระหว่างอินพุตกับเอาต์พุตสำเร็จ และกรณีเอาต์พุตที่ 3 ตัวบ่งชี้ g_3 ชี้อินพุตที่ 4 ตัวตัดสินใจของพอร์ตเอาต์พุต จึงส่งคำยินยอมไปยังอินพุตที่ 4 และเลื่อนค่าบ่งชี้ของ g_3 ไปชี้อินพุตที่ 1



รูปที่ 4.6 แสดงตัวอย่างการทำงานทั้ง 3 ขั้นตอนของอัลกอริทึม DRRM

4.2.5 สมรรถนะของอัลกอริทึม maxsize

ดังรูปที่ 4.7 แสดงสมรรถนะของอัลกอริทึม กรณีทราฟฟิกเป็นประเภทยูนิฟอร์ม จะเห็นว่ากรณีที่ไม่มีการวนซ้ำ อัลกอริทึม iSLIP และ DRRM จะให้ค่าทรูพุตและการประวิงเวลาดีกว่าอัลกอริทึม PIM แต่เมื่อมีการวนซ้ำแล้ว อัลกอริทึมเหล่านี้จะให้ค่าการประวิงเวลาใกล้เคียงกันและใกล้เคียงกับสวิตช์ OQ แต่เนื่องจากอัลกอริทึม iSLIP และ DRRM นำไปสร้างเป็นสวิตช์ได้ง่ายกว่าจึงนิยมใช้ในปัจจุบันมากกว่า ซึ่งในปัจจุบันอัลกอริทึมที่ใช้เทคนิคการเลือกเซลล์ส่งผ่านสวิตช์แบบวนรอบ (round robin) ถูกนำไปใช้ในทางการค้า [16] และเป็นอัลกอริทึมพื้นฐานที่นำไปใช้ในการวิจัยในเรื่องที่เกี่ยวข้องกับสวิตช์มากกว่าแบบอื่น ๆ วิทยานิพนธ์ฉบับนี้จึงใช้อัลกอริทึม iSLIP ในการจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์ที่จะนำเทคนิคเอนเวโลป และการจัดการบัฟเฟอร์ที่เหมาะสมมาประยุกต์ใช้ และใช้อัลกอริทึมจัดลำดับเซลล์อื่น ๆ เป็นตัวเปรียบเทียบสมรรถนะของสวิตช์



รูปที่ 4.7 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ กับอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ

4.3 การแมตช์น้ำหนักสูงที่สุด (maxweight)

การแมตช์น้ำหนักสูงที่สุด, M สำหรับกราฟไบพาร์ไทท์ ก็คือค่าสูงสุดของผลรวม $w_{i,j}$ โดยที่ $w_{i,j}$ คือค่าน้ำหนักของด้านที่เชื่อมต่อระหว่างจุดยอด i และ j และเราสามารถหาค่าได้โดยการแก้ปัญหาการไหลในโครงข่ายสมมูล อัลกอริทึมที่มีประสิทธิภาพมากที่สุดสำหรับการแก้ปัญหานี้ มีค่าความซับซ้อนเท่ากับ $O(N^3 \log N)$

อัลกอริทึมการแมตช์ขนาดสูงที่สุดนั้น พิจารณาแต่เพียงว่าการรอคิวที่อินพุตนั้นว่างหรือไม่ ดังนั้น ถ้าทราฟฟิกเป็นประเภทนอนยูนิฟอร์มแล้ว ก็จะเกิดการสะสมของเซลล์ในบางอินพุต โดยที่อัลกอริทึมไม่มีทางล่วงรู้ถึงสิ่งเหล่านี้ จึงไม่ได้ทำการลดการสะสมของเซลล์ที่รอคิวอยู่ที่อินพุตนั้น

หรืออีกนัยหนึ่ง อัลกอริทึมการแมตช์น้ำหนักสูงที่สุด สามารถที่จะทำการคำนวณการครอบครองคิว $L_{i,j}(n)$ ของ $VOQ_{i,j}$ หรือเวลาในการรอคิว $W_{i,j}(n)$ ของเซลล์ที่หัวคิว โดยอัลกอริทึมจะให้สิทธิในการที่จะได้รับบริการส่งผ่านโครงสร้างสวิตช์ก่อน แก่คิวที่มีจำนวนการครอบครองคิวมากกว่าหรือแก่เซลล์ที่หัวคิวที่รออยู่ในคิวนานกว่า ผลลัพธ์ก็คือ ทำให้อัลกอริทึม

สามารถให้ค่าทรูพุตสูงสุดได้ถึง 100 เปอร์เซ็นต์ สำหรับทราฟฟิกทั้งประเภทยูนิฟอร์มและนอนยูนิฟอร์ม

เราจะทำการพิจารณาการแมตช์น้ำหนักสูงที่สุด 2 รูปแบบ ที่อาจนำมาใช้จัดลำดับเซลล์ส่งผ่านสวิตช์แบบรอคิวที่อินพุต ได้แก่ อัลกอริทึม LQF และอัลกอริทึม OCF ซึ่งได้นำเสนอครั้งแรกโดย McKeown [11] และจากผลการจำลองแบบให้ค่าทรูพุตสูงสุดถึงเกือบ 100 เปอร์เซ็นต์ และต่อมาได้มีการพิสูจน์ว่าอัลกอริทึม LQF ให้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ สำหรับทราฟฟิกทุกประเภท โดยที่การมาถึงของเซลล์เป็นอิสระต่อกัน [13]

แม้ว่าในทางปฏิบัติแล้ว อัลกอริทึม LQF และ OCF จะมีความซับซ้อนมากเกินกว่าที่จะนำไปสร้างเป็นสวิตช์ความเร็วสูงได้ แต่ก็เป็นสิ่งที่สำคัญที่จะต้องนำมาศึกษา เพราะว่าอัลกอริทึมเหล่านี้ ได้แสดงให้เห็นหลักการในการจัดลำดับเซลล์ส่งผ่านสวิตช์สำหรับทราฟฟิกประเภทนอนยูนิฟอร์ม

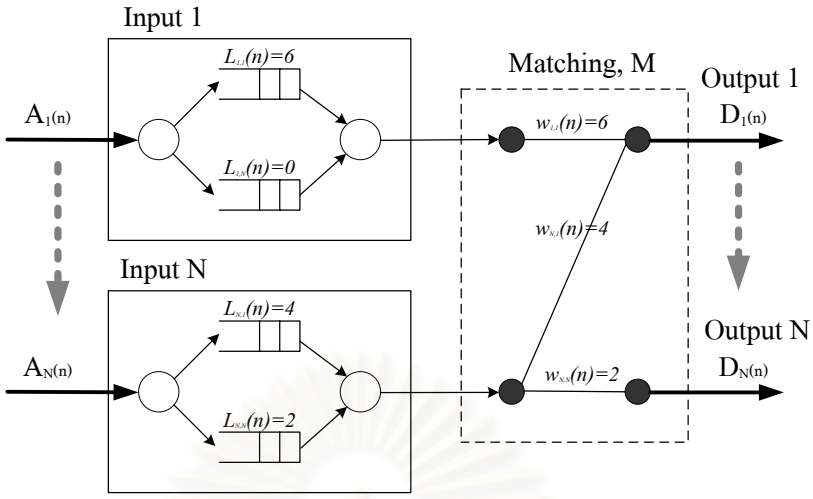
4.3.1 อัลกอริทึม LQF

อัลกอริทึม LQF (Longest Queue First (LQF) algorithm) เป็นอัลกอริทึมแรกที่ใช้ในการแก้ปัญหาค่าทรูพุตลดลงเนื่องมาจากทราฟฟิกประเภทนอนยูนิฟอร์ม ซึ่งได้รับพิสูจน์แล้วว่าสวิตช์แบบรอคิวที่อินพุตจะให้ค่าทรูพุตสูงสุดได้เช่นเดียวกับสวิตช์แบบรอคิวที่เอาต์พุต โดยจะให้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ สำหรับกระบวนการการมาถึงของเซลล์ที่เป็นอิสระต่อกัน โดยให้สิทธิกับอินพุตที่มีจำนวนการครอบครองคิวมากที่สุด ได้รับการส่งผ่านโครงสร้างสวิตช์ก่อน ทำให้บัฟเฟอร์ของแต่ละอินพุตมีจำนวนการครอบครองคิวที่สมดุลกัน (balance) ลดปัญหาความเป็นไปได้ที่เซลล์จะล้นบัฟเฟอร์ (overflow) แต่การทำวิธีนี้ ก็อาจทำให้อินพุตที่มีการครองครองคิวต่ำไม่ได้รับการบริการหรือได้รับการบริการน้อยมาก จนเกิดสภาวะสทาร์เวชันขึ้น

อัลกอริทึม LQF จะให้บริการส่งผ่านโครงสร้างสวิตช์กับอินพุตที่มีจำนวนการครอบครองคิวมากกว่าอินพุตอื่น แสดงดังรูปที่ 4.7 ดังนั้นแต่ละค่าน้ำหนักคำร้องขอ (request weight) จะมีค่าเท่ากับจำนวนการครอบครองคิวของเซลล์ และเราสามารถอธิบายได้ดังนี้

$$w_{i,j}(n) = L_{i,j}(n) \tag{4.4}$$

โดยที่ $w_{i,j}(n)$ คือน้ำหนักของคำร้องขอของอินพุตที่ i กับเอาต์พุตที่ j ในแต่ละไทม์สล็อต อัลกอริทึม LQF จะเลือกการแมตช์ M ที่จะทำให้ได้ค่าน้ำหนักรวมของการแมตช์สูงที่สุด กล่าวคือ $\sum_{(i,j)\in M} w_{i,j}$ มีค่าสูงที่สุด



รูปที่ 4.8 ตัวอย่างอัลกอริทึม LQF

4.3.2 อัลกอริทึม OCF

อัลกอริทึม OCF (Oldest Cell First (OCF) algorithm) ได้รับการออกแบบมาเพื่อให้ได้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ สำหรับทราฟฟิกประเภทนอนยูนิฟอร์ม และเพื่อแก้ปัญหาสทาร์เวชันที่เกิดขึ้นกับอัลกอริทึม LQF อัลกอริทึม OCF จะให้สิทธิแก่เซลล์ที่หัวคิวที่รออยู่ในคิวนานที่สุดได้รับการส่งผ่านโครงสร้างการสวิตช์ก่อน เหตุผลก็คือ จะได้ไม่มีเซลล์ใดรออยู่ในคิวเป็นเวลานานจนเกิดสภาวะสทาร์เวชัน เพราะว่าเซลล์ที่ยังไม่ได้รับการบริการในที่สุดก็จะกลายเป็นเซลล์ที่อยู่ในคิวนานที่สุด จนได้รับสิทธิในการส่งผ่านโครงสร้างการสวิตช์ก่อน

อัลกอริทึม OCF จะให้สิทธิแก่เซลล์ที่หัวคิวที่จะได้รับการส่งผ่านโครงสร้างการสวิตช์ก่อน โดยใช้เวลารอคิวของเซลล์ที่หัวคิวเป็นค่าน้ำหนักคำร้องขอ และเราสามารถนิยามได้ดังนี้

$$w_{i,j}(n) = W_{i,j}(n) \tag{4.5}$$

โดยที่ $W_{i,j}(n)$ คือเวลารอคิวของเซลล์ที่หัวคิว เช่นเดียวกับอัลกอริทึม LQF อัลกอริทึม OCF ใช้อัลกอริทึมน้ำหนักสูงที่สุดในการหาค่าการแมตช์ของน้ำหนักสูงที่สุด

4.3.3 อัลกอริทึมแบบวนซ้ำ

เนื่องจากอัลกอริทึมน้ำหนักสูงที่สุดนั้น มีความซับซ้อนมากเกินไปที่จะนำไปสร้างเป็นสวิตช์ จึงได้มีการนำเสนออัลกอริทึมที่มีความซับซ้อนน้อยกว่า ดังนั้น เราจะทำการพิจารณาอัลกอริทึม LQF และ OCF แบบวนซ้ำ ซึ่งจะให้ค่าการแมตช์น้ำหนักมากที่สุด (maximal weight matching) และมีลักษณะเช่นเดียวกับอัลกอริทึม iSLIP ดังที่กล่าวมาแล้ว ที่ทำดังนี้ก็เพื่อให้สามารถ

นำไปสร้างเป็นสวิตช์ได้ง่าย และอัลกอริทึมทำงานได้อย่างรวดเร็วในการหาค่าการแมตช์น้ำหนักมากที่สุด อัลกอริทึมเหล่านี้เรียกว่า iLQF และ iOCF

4.3.3.1 อัลกอริทึม iLQF

เช่นเดียวกับอัลกอริทึม iSLIP อัลกอริทึม iLQF เป็นอัลกอริทึมแบบวนซ้ำที่ประกอบไปด้วยตัวตัดสินใจของอินพุต N อินพุต และตัวตัดสินใจของเอาต์พุต N เอาต์พุต กับการประมวลผลแบบขนาน ตัวจัดลำดับเซลล์ส่งผ่านสวิตช์จะพิจารณาให้สิทธิแก่เซลล์ที่รออยู่ในคิวที่มีจำนวนการครอบครองคิว $L_{i,j}(n)$ มากกว่าให้ได้รับการบริการก่อน

ที่จุดเริ่มต้นของแต่ละไทม์สล็อต กระบวนการหาค่าการแมตช์จะเริ่มต้นขึ้น ทุก ๆ อินพุตและเอาต์พุตที่ยังไม่ได้รับการเชื่อมต่อเมื่อถึงจุดสิ้นสุดของการวนซ้ำ จะได้รับสิทธิที่จะได้รับการเลือกในการวนซ้ำรอบต่อไป การเชื่อมต่อที่ได้ทำขึ้นในแต่ละรอบของการวนซ้ำ จะไม่ถูกยกเลิกเมื่อมีการวนซ้ำในรอบต่อไป แม้ว่าจะทำให้ได้ขนาดของการแมตช์มากขึ้นกว่าเดิม ในแต่ละรอบของการวนซ้ำ จะประกอบไปด้วยขั้นตอน 3 ขั้นตอน ดังต่อไปนี้

ขั้นตอนที่ 1 *คำร้องขอ* แต่ละอินพุตที่ยังไม่ได้รับการเชื่อมต่อ จะส่งคำร้องของความยาว b บิต ($2^b \geq L_{max}$) ซึ่งบ่งชี้ถึงจำนวนเซลล์ที่อยู่ในคิวที่จะไปยังเอาต์พุตนั้น ไปยังเอาต์พุตนั้น

ขั้นตอนที่ 2 *คำยินยอม* เมื่อแต่ละเอาต์พุตที่ยังไม่ได้รับการเชื่อมต่อได้รับคำร้องขอ จะเลือกคำร้องขอที่มีค่าสูงสุด (บ่งชี้ถึงจำนวนเซลล์ที่อยู่ในคิว) ถ้ามีค่าเท่ากันให้ทำการเลือกแบบสุ่ม และส่งคำยินยอมไปยังอินพุตที่เลือก

ขั้นตอนที่ 3 *ตอบรับคำยินยอม* เมื่อแต่ละอินพุตได้รับคำยินยอม จะเลือกตอบรับคำยินยอมที่คำร้องขอมีค่าสูงสุด ถ้ามีค่าเท่ากันให้ทำการเลือกแบบสุ่ม

คุณสมบัติของอัลกอริทึม iLQF มีดังต่อไปนี้

- ข้อที่ 1 อินพุตที่มีจำนวนเซลล์อยู่ในคิวมากที่สุด จะต้องได้รับการบริการส่งผ่านโครงสร้างการสวิตช์เสมอโดยไม่ขึ้นกับจำนวนรอบของการวนซ้ำ เพราะว่าในการวนซ้ำรอบแรกอินพุตนี้จะได้รับคำยินยอมที่มีค่าสูงสุด ซึ่งทำให้จะต้องได้รับการเชื่อมต่ออย่างแน่นอน
- ข้อที่ 2 เช่นเดียวกันกับกรณีอัลกอริทึม iSLIP จะมีจำนวนรอบของการวนซ้ำไม่เกิน N รอบ เพราะว่าแต่ละรอบจะต้องมีอย่างน้อยหนึ่งการเชื่อมต่อ ถ้าไม่มีการเชื่อมต่อในรอบใดก็จะไม่มีการวนซ้ำในรอบต่อไป ทำให้จำนวนรอบของการวนซ้ำลู่เข้าสู่ค่า N

- ข้อที่ 3 ในกรณีโหลดงานที่เป็นแบบไม่ตรงตามเงื่อนไข อาจจะทำให้อินพุตเกิดสภาวะสทาร์เวชันได้เช่นเดียวกับกรณีของอัลกอริทึม LQF

4.3.3.2 อัลกอริทึม iOCF

อัลกอริทึม iOCF สามารถกำจัดปัญหาสทาร์เวชันที่เกิดขึ้นในอัลกอริทึม iLQF ได้ โดยใช้วิธีเลือกเซลล์ที่หัวคิวที่มีเวลารอคิวนานที่สุด ที่จะได้รับการบริการส่งผ่านโครงสร้างการสวิตช์ อัลกอริทึม iOCF แตกต่างจากอัลกอริทึม iLCF ตรงที่คำร้องขอของอินพุตแต่ละอินพุต บ่งชี้ค่าของเวลารอคิว $W_{i,j}(n)$ ของเซลล์ที่หัวคิว แทนที่จะเป็นจำนวนเซลล์ที่อยู่ในคิว $L_{i,j}(n)$

คุณสมบัติของอัลกอริทึม iOCF มีดังต่อไปนี้

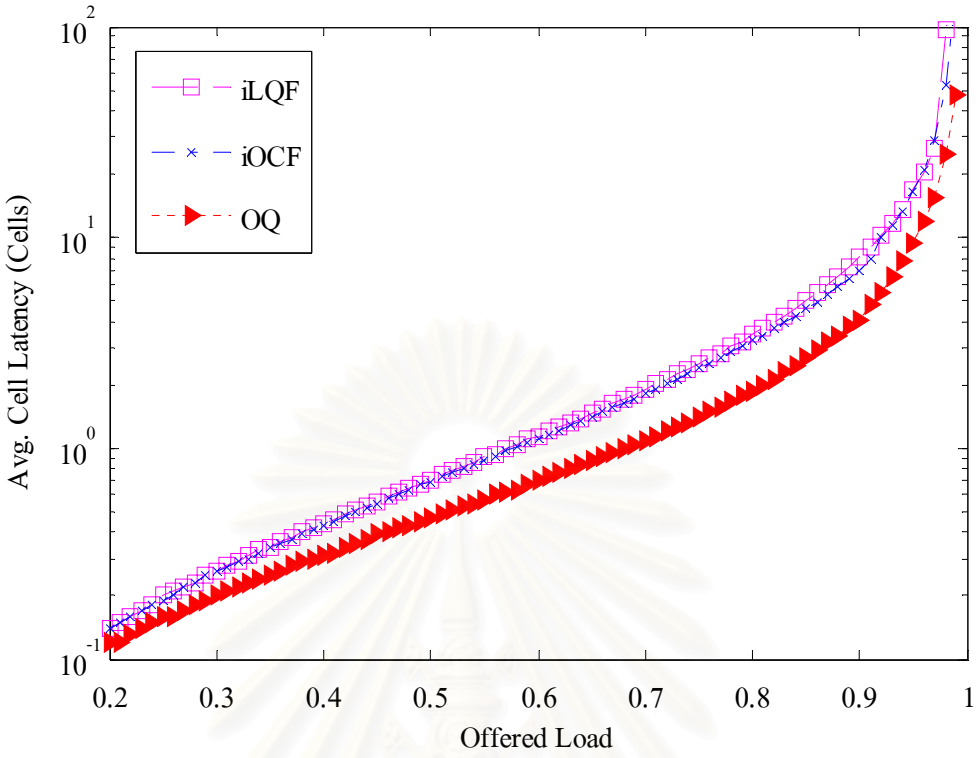
- ข้อที่ 1 เซลล์ที่หัวคิวของอินพุตที่มีเวลารอคิวนานที่สุด จะได้รับบริการส่งผ่านโครงสร้างสวิตช์ก่อนโดยไม่ขึ้นกับจำนวนรอบของการวนซ้ำ เนื่องจากในการวนซ้ำรอบแรกอินพุตนั้นจะได้รับคำยินยอมที่มีค่าสูงสุด ซึ่งทำให้จะต้องได้รับการเชื่อมต่ออย่างแน่นอน

- ข้อที่ 2 เช่นเดียวกับอัลกอริทึม iLQF อัลกอริทึม iOCF จะมีจำนวนรอบของการวนซ้ำไม่เกิน N รอบ

- ข้อที่ 3 ไม่มีอินพุตใด ที่จะเกิดสภาวะสทาร์เวชัน

4.3.4 สมรรถนะของอัลกอริทึม maxweight

ตามรูปที่ 4.9 จะเห็นว่าทั้งอัลกอริทึม iLQF และ iOCF ให้สมรรถนะที่ดีด้านค่าเฉลี่ยการประวิงเวลาของเซลล์ใกล้เคียงกับสวิตช์ OQ และให้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ ภายใต้เงื่อนไขทราฟฟิกเป็นประเภทยูนิฟอร์ม แต่เนื่องจากความซับซ้อนในการออกแบบฮาร์ดแวร์ของสวิตช์ประเภทนี้จึงไม่สามารถนำมาใช้โดยตรงในทางปฏิบัติได้ ตัวอย่างที่เห็นได้อย่างชัดเจนคือสวิตช์จะต้องส่งค่าคำร้องขอความยาว b บิตเพื่อส่งค่าความยาวคิวหรือเวลาที่รออยู่ในคิว แทนที่จะเป็นหนึ่งบิตอย่างเช่นอัลกอริทึม maxsize ที่นำไปใช้ในทางปฏิบัติเช่นอัลกอริทึม iSLIP ซึ่งส่งเพียงค่าการครอบครองคิวเท่านั้น (ตัวอย่างเช่น บิต 1 แทนมีเซลล์อยู่ในคิว และบิต 0 แทนว่าง) และให้สมรรถนะทางด้านการประวิงเวลาที่ใกล้เคียงกัน



รูปที่ 4.9 สมรรถนะของสวิตช์ด้านการประวิงเวลาของอัลกอริทึม maxweight

4.4 สรุปท้ายบท

สวิตช์แบบรอคิวที่อินพุตส่วนใหญ่มีโครงสร้างการสวิตช์แบบครอสบาร์ เมื่อเราใช้โครงสร้างแบบครอสบาร์ จึงมีความจำเป็นที่จะต้องมีตัวจัดลำดับเซลล์ส่งผ่านสวิตช์ ในการเลือกเซลล์ที่เข้าทำการแข่งขันเพื่อให้ได้รับบริการส่งผ่านสวิตช์และกำหนดโครงสร้างการสวิตช์ ได้มีการแนะนำอัลกอริทึมจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์หลายรูปแบบ โดยมีการจัดการบัฟเฟอร์แบบ VOQ อัลกอริทึม MSM ที่ทำการหาค่าการแมตช์ขนาดมากที่สุด (maximal size match) ได้แก่ อัลกอริทึม PIM, RRM, iSLIP และ DRRM ซึ่งจากการประเมินสมรรถนะของสวิตช์นั้นสามารถให้ค่าทรูพุตสูงสุดถึง 100 เปอร์เซ็นต์ ภายใต้เงื่อนไขทราฟฟิกต้องเป็นประเภทยูนิฟอร์ม ในปี 1993 อัลกอริทึม PIM ได้ถือกำเนิดขึ้นมาจากบริษัท DEC เป็นอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์โดยใช้วิธีเลือกเซลล์แบบสุ่มผลก็คือทำให้สวิตช์ประมวลผลได้ช้าจึงไม่เหมาะสมเมื่อเป็นสวิตช์ความเร็วสูงที่ส่งผ่านข้อมูลที่ความเร็วกิกะบิตต่อวินาที ต่อมาในปี 1995 McKeown ได้เสนออัลกอริทึมที่ทำงานได้เร็วกว่า นำไปสร้างเป็นสวิตช์ได้ง่ายกว่า และประสิทธิภาพสูงกว่าเรียกว่า อัลกอริทึม iSLIP และต่อมาในปี 1998 Chao [14] ได้ทำการปรับปรุงโดยลดขั้นตอนที่ซ้ำซ้อนของอัลกอริทึม iSLIP ลง และเรียกอัลกอริทึมนี้ว่า อัลกอริทึม DRRM

แต่เนื่องจากอัลกอริทึมเหล่านี้ ได้พัฒนาขึ้นภายใต้สมมุติฐานที่ว่าทราฟฟิกเป็นประเภทยูนิฟอร์ม ดังนั้นเมื่อทราฟฟิกเป็นประเภทนอนยูนิฟอร์มแล้วทำให้สมรรถนะของสวิตช์ลดลง จึงได้มีการนำเอาหลักการของอัลกอริทึม MWM มาทำการศึกษา และได้นำเสนออัลกอริทึม iLQF และ iOCF ซึ่งสามารถรองรับทราฟฟิกได้ทั้งประเภทยูนิฟอร์มและนอนยูนิฟอร์ม แต่อัลกอริทึมเหล่านี้ก็ยังมีความซับซ้อนมากเกินไป จนไม่เหมาะสมที่จะนำไปใช้ในทางปฏิบัติในการสร้างเป็นสวิตช์ความเร็วสูง ผู้วิจัยด้านสวิตช์จึงยังคงใช้อัลกอริทึมแบบ MSM โดยพยายามที่จะปรับปรุงอัลกอริทึมรูปแบบเดิมให้สามารถรองรับทราฟฟิกประเภทนอนยูนิฟอร์ม แน่นอนสมรรถนะของสวิตช์ไม่อาจถึงจุดสูงสุดตามรูปแบบเดิมได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
เทคนิคเอนเวโลปที่นำเสนอ

5.1 เทคนิคเอนเวโลป

เทคนิคเอนเวโลป มีแนวคิดที่ว่าจะให้หน่วยข้อมูลที่จะส่งผ่านโครงสร้างการสวิตช์นั้นมีความยาวเพิ่มขึ้นแทนที่จะส่งทีละเซลล์ โดยวิธีรวมเซลล์ที่จะไปยังเอาต์พุตเดียวกันรวมอยู่ภายในเอนเวโลปเป็นลำดับ ซึ่งกลุ่มเซลล์ (entity) จะถูกสวิตช์ผ่านโครงสร้างการสวิตช์ ความยาวของหน่วยข้อมูลก็คือ ความยาวของไทม์สล็อตที่เพิ่มขึ้นในโครงสร้างการสวิตช์ หมายความว่าเราจะมีเวลามากขึ้นในการตัดสินใจเลือกเซลล์สำหรับส่งผ่านสวิตช์ซึ่งก็เป็นประโยชน์ที่เห็นได้อย่างชัดเจนที่จะนำไปสร้างเป็นสวิตช์ความเร็วสูง โดยเฉพาะอย่างยิ่งสวิตช์ที่มีตัวเลือกเซลล์ส่งผ่านสวิตช์แบบรวมศูนย์ ทั้งนี้เนื่องจากในแต่ละไทม์สล็อตสวิตช์ประเภทนี้มีเวลาจำกัดในการประมวลผลหาค่าการแมตช์ ทำให้เกิดปัญหาคอขวดในการสวิตช์ข้อมูล การที่ไทม์สล็อตยาวขึ้นทำให้เราสามารถนำสถาปัตยกรรมสวิตช์ที่มีข้อจำกัดในด้านเวลาในการประมวลผลหาค่าการแมตช์ เพื่อจัดลำดับเซลล์ส่งผ่านสวิตช์มาใช้ได้ เช่นเดียวกับที่ในอดีตเราใช้วิธีการใช้หน่วยความจำร่วมกัน (shared memory) และการจัดการ bit-sliced

การนำเอาเทคนิคเอนเวโลปมาใช้ ถึงแม้ว่าจะสามารถแก้ปัญหาคอขวดในการสวิตช์ข้อมูลได้ แต่ต้องแลกกับสมรรถนะทางด้านการประวิงเวลาของสวิตช์ที่จะเลวลง ทั้งนี้เนื่องมาจากเทคนิคเอนเวโลปทำให้ค่าเฉลี่ยการประวิงเวลาของเซลล์เพิ่มสูงขึ้น คุณภาพการให้บริการของสวิตช์จึงลดลงไปด้วย ดังนั้น การพิจารณาในการเพิ่มค่า QoS ของสวิตช์จึงเป็นประเด็นที่สำคัญ ทั้งนี้ก็เพื่อที่จะรับประกันในคุณภาพการให้บริการของสวิตช์ ตัวอย่างเช่น [1] การให้เซลล์แต่ละเซลล์ที่มีความต้องการค่า QoS ต่างกันสามารถรวม (mixed) อยู่ภายในเอนเวโลปเดียวกันได้ และเพื่อให้เป็นไปได้ในทางปฏิบัติเราสามารถใช้วิธีลดการแยกคิวในแต่ละอินพุต หรือใช้การจัดการบัฟเฟอร์ของสวิตช์แบบ MIQ เป็นต้น

จะเห็นว่าในการนำเทคนิคเอนเวโลปมาประยุกต์ใช้ การวางบัฟเฟอร์ที่อินพุตของสวิตช์เป็นสิ่งจำเป็น ทั้งนี้ก็เพื่อจัดเตรียมเอนเวโลปก่อนได้รับบริการส่งผ่านสวิตช์ เทคนิคนี้จึงจำเป็นต้องนำไปใช้กับสวิตช์ที่มีการวางบัฟเฟอร์ที่พอร์ตอินพุต เช่น สวิตช์แบบ IQ และแบบ CIOQ สำหรับสวิตช์แบบ CIOQ จะทำให้สูญเสียความต่อเนื่องในการทำแบ็กเพรสเชอร์ (back-pressure) เนื่องจากการเพิ่มความยาวของไทม์สล็อต จึงอาจทำให้คุณสมบัติทางด้านการประวิงเวลาของสวิตช์เลวลง

<u>ปัญหาประการแรก</u> เราจะเลือกเอนเวโลปใดในแต่ละอินพุตเพื่อส่งผ่านโครงสร้างการสวิตช์ที่จะทำให้ค่าทรูพุตของสวิตช์สูงสุด [18] จะเห็นว่าการใช้เทคนิคเอนเวโลปทำให้ตัวจัดลำดับ

เซลล์มีเวลามากขึ้นในการเลือกแต่ละเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์ จึงนำไปสร้างเป็นสวิตช์ได้ง่ายขึ้น เราจะเลือกเอนเวโลปใดในการส่งผ่านโครงสร้างการสวิตช์ วิธีที่หนึ่งคือเลือกเอนเวโลปที่มีเซลล์บรรจุที่หัวคิวก่อน (เช่นเลือกโดยวิธีสุ่มแต่ละอินพุต หรือวนรอบแต่ละอินพุต หรือตามลำดับความสำคัญของเซลล์ เป็นต้น) วิธีที่สองคือเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มก่อน หรือวิธีที่สามคือเลือกเอนเวโลปโดยวิธีผสมผสานทั้งสองวิธีแรกเข้าด้วยกัน

การเลือกวิธีที่หนึ่งทำให้เกิดปัญหาคือ ถ้าเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่ไม่เต็ม จะเกิดการสูญเสียแบนด์วิดท์ในการสวิตช์ทำให้ค่าทรูพุตของสวิตช์ลดลง ยิ่งเอนเวโลปยาวขึ้นเท่าใดก็ยิ่งมีผลกระทบต่อค่าทรูพุตของสวิตช์เพิ่มขึ้นเท่านั้น ในกรณีทราฟฟิกหนาแน่นจะไม่มีผลกระทบเพราะว่าเอนเวโลปมักจะมีเซลล์บรรจุอยู่เต็มอยู่เสมอ แต่เนื่องจากมีการแยกคิวที่อินพุตเพื่อเตรียมเอนเวโลปสำหรับส่งผ่านสวิตช์ ทำให้อาจไม่เหมาะสมที่จะนำไปใช้ในทางปฏิบัติเมื่อสวิตช์มีขนาดใหญ่ขึ้น

การเลือกวิธีที่สองในกรณีที่ทราฟฟิกต่ำ ๆ จะทำให้เกิดการสูญเสียแบนด์วิดท์ในการสวิตช์ เพราะว่าแต่ละคิวที่อินพุตส่วนใหญ่แล้วเอนเวโลปมีเซลล์ที่บรรจุอยู่ไม่เต็ม ซึ่งจะไม่ได้รับการส่งผ่านสวิตช์ ทำให้ทรูพุตของสวิตช์ลดลงยิ่งกว่าวิธีที่หนึ่ง แต่จะมีผลดีที่ทราฟฟิกหนาแน่นเพราะแต่ละคิวมีเซลล์บรรจุอยู่เป็นจำนวนมาก การเลือกเอนเวโลปบรรจุเต็มทำให้การสูญเสียแบนด์วิดท์ในการสวิตช์มีน้อยลง

การเลือกวิธีที่สามหรือวิธีผสมผสาน ได้ลดการสูญเสียแบนด์วิดท์ที่ทราฟฟิกต่ำด้วยการเลือกเอนเวโลปบรรจุเต็มก่อนแล้วจึงเลือกเอนเวโลปที่มีเซลล์อยู่ที่หัวคิว และที่ทราฟฟิกหนาแน่นนั้น การเลือกเอนเวโลปบรรจุเต็มก่อนทำให้การสูญเสียแบนด์วิดท์ในการสวิตช์น้อยที่สุดเช่นเดียวกับวิธีที่สอง จึงให้สมรรถนะของสวิตช์ทางด้านการประวิงเวลาดีกว่าทั้งสองวิธีดังที่กล่าวมาแล้ว แต่เนื่องจากในแต่ละไทม์สล็อตจะต้องประมวลผลในการเลือกทั้งเอนเวโลปบรรจุเต็ม และเอนเวโลปที่มีเซลล์ที่หัวคิว ทำให้ปริมาณงานที่จะต้องทำการประมวลผลเพิ่มขึ้นเป็นสองเท่า จึงใช้เวลาในการประมวลผลหาค่าการแมตช์นานกว่าทั้งสองวิธีดังกล่าวข้างต้น ดังนั้น เราจึงพยายามที่จะหาวิธีเลือกเอนเวโลปที่ให้สมรรถนะด้านการประวิงเวลาไม่ด้อยกว่าทั้งสามวิธี และใช้เวลาประมวลผลในการหาค่าการแมตช์น้อยกว่าวิธีที่สาม

วิทยานิพนธ์ฉบับนี้ ได้นำเสนอวิธีเลือกเอนเวโลปโดยวิธี **เทรสโฮลด์เอนเวโลปแบบพลวัต** ซึ่งมีแนวคิดที่ว่า พิจารณาเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่บางส่วนโดยกำหนดเป็นค่าเทรสโฮลด์ของจำนวนเซลล์ที่บรรจุอยู่ในเอนเวโลป และเปลี่ยนแปลงค่าเทรสโฮลด์ไปตามค่าทราฟฟิก ในกรณีทราฟฟิกต่ำ ๆ จำนวนเซลล์ในเอนเวโลปมีน้อย เราจึงกำหนดค่าเอนเวโลปน้อยเพื่อลดการสูญเสียแบนด์วิดท์ที่เกิดขึ้นจากการเลือกในวิธีที่หนึ่ง และในกรณีที่ทราฟฟิกมีค่าเพิ่มขึ้นเราจะ

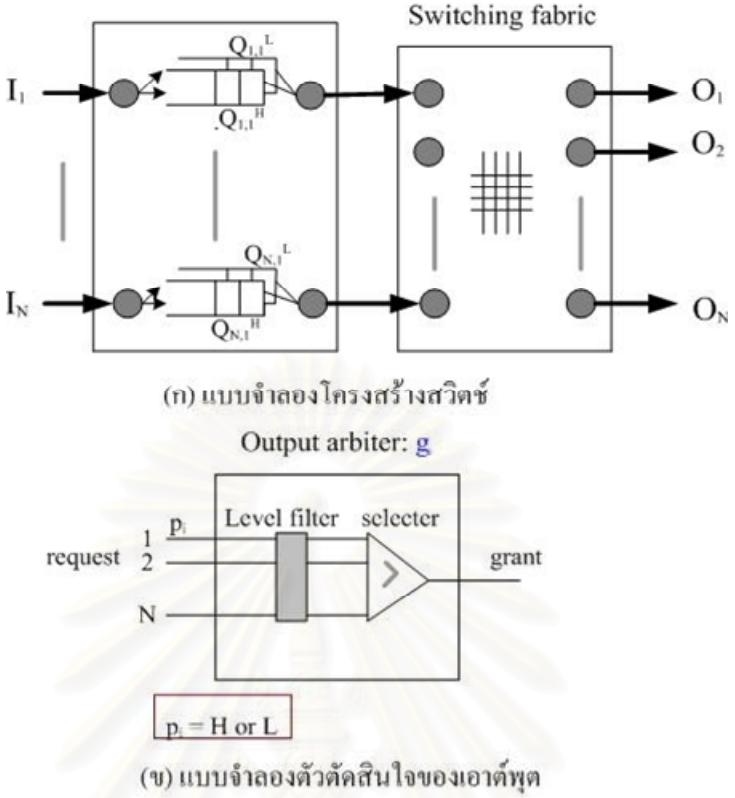
เปลี่ยนแปลงค่าเทรสโฮลด์ให้สูงขึ้นตามค่าทราฟฟิก จนเมื่อทราฟฟิกหนาแน่นเราจะเปลี่ยนแปลงค่าเทรสโฮลด์เป็นเท่ากับเอนเวโลปที่บรรจุเต็ม ซึ่งได้อาศัยประโยชน์จากการเลือกเอนเวโลปวิธีที่สอง โดยที่เราไม่ต้องใช้เวลาในประมวลผลหาค่าการแมตช์ที่เพิ่มขึ้น เช่นที่เกิดขึ้นเมื่อเราเลือกเอนเวโลปวิธีที่สาม เราใช้การคำนวณหาค่าเทรสโฮลด์ที่ให้สมรรถนะด้านการประวิงเวลาของสวิตช์ที่ดีที่สุด โดยใช้การจำลองแบบด้วยโปรแกรมคอมพิวเตอร์ แล้วจึงกำหนดเป็นค่าเทรสโฮลด์ที่ทราฟฟิกนั้น ๆ การวิเคราะห์ผลจากการทดสอบจะได้นำเสนอในบทต่อไป

ปัญหาประการที่สอง เอนเวโลปควรจะมีขนาดยาวเท่าใดจึงจะเหมาะสมที่สุด การเพิ่มขนาดความยาวของเอนเวโลปโดยไม่มีขอบเขตจำกัด ถึงแม้ว่าจะทำให้สวิตช์สามารถประมวลผลในการหาค่าการแมตช์ได้ทัน แต่เมื่อเราเพิ่มขนาดเอนเวโลปมากขึ้นจนสวิตช์สามารถประมวลผลได้ทันแล้วก็ไม่มีความจำเป็นอันใดที่จะเพิ่มขนาดเอนเวโลปต่อไปอีก ทั้งนี้ ในการเพิ่มขนาดของเอนเวโลปให้ยาวขึ้นจะทำให้ค่าเฉลี่ยการประวิงเวลาของเซลล์เพิ่มสูงขึ้นจนสวิตช์ไม่สามารถรับประกันในคุณภาพการให้บริการได้ ปัญหาก็คือ เราจะเพิ่มขนาดเอนเวโลปยาวไม่เกินเท่าใด จึงจะทำให้สวิตช์สามารถประมวลผลได้ทันจนสามารถขจัดปัญหาคอขวดในการสวิตช์ข้อมูลได้ และยังสามารถรับประกันในคุณภาพการให้บริการ

ดังนั้น ในการนำเทคนิคเอนเวโลปมาประยุกต์ใช้กับสวิตช์จะต้องแก้ปัญหาหลัก 2 ประการ คือ**วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์** และ**การหาขนาดเอนเวโลปที่เหมาะสม** ทั้งนี้ก็มีวัตถุประสงค์ที่จะให้สวิตช์สามารถรับประกันในคุณภาพการให้บริการทราฟฟิกทั้งประเภทเรียลไทม์และประเภทดาตาได้ ในการเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์จะใช้แบบจำลองที่ได้ออกแบบขึ้นตามหัวข้อ 5.2 และใช้อัลกอริทึมจัดลำดับเซลล์ตามหัวข้อ 5.3 แล้วจึงทำการเปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ เมื่อใช้วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์ที่ได้นำเสนอในวิทยานิพนธ์ฉบับนี้ กับวิธีเลือกเอนเวโลปที่ได้มีการนำเสนอไว้ก่อนหน้านี้ทั้งสามวิธีโดยการวิเคราะห์ผลที่ได้จากการประมวลผลด้วยโปรแกรมคอมพิวเตอร์ ในส่วนของขนาดเอนเวโลปที่เหมาะสมนั้น สามารถคำนวณได้โดยตรงจากข้อมูลของข่ายเชื่อมโยง ความยาวของเซลล์ และเวลาที่ใช้ในการประมวลผลหาค่าการแมตช์ของอัลกอริทึมจัดลำดับเซลล์

5.2 แบบจำลองของสวิตช์ที่นำเสนอ

แบบจำลองของสวิตช์อย่างง่าย ที่จะใช้เป็นต้นแบบในการเขียนโปรแกรมจำลองแบบด้วยภาษา C++



รูปที่ 5.1 แบบจำลองโครงสร้างสวิตช์และตัวตัดสินใจของเอาต์พุต

แบบจำลองโครงสร้างสวิตช์ตามรูปที่ 5.1 (ก) มีโครงสร้างการสวิตช์แบบครอสบาร์ซึ่งเป็นโครงสร้างการสวิตช์ที่ไม่มีการติดขัดภายใน ส่วนการจัดการคิวที่อินพุตนั้นให้แยกคิวทางตรรกะเป็น 2 ระนาบ คือ $Q_{i,j}^{H}$ กับ $Q_{i,j}^{L}$ โดยที่ H คือระนาบที่บรรจุเซลล์ที่มีลำดับความสำคัญสูงหรือทราฟฟิกประเภทเรียลไทม์ และ L คือระนาบที่บรรจุเซลล์ที่มีลำดับความสำคัญต่ำหรือทราฟฟิกประเภทดาตา $Q_{i,j}$ คือจำนวนเซลล์ที่รออยู่ในคิวของอินพุตที่ i และ N คือขนาดของสวิตช์ ส่วนรูปที่ 5.1 (ข) แสดงตัวตัดสินใจของเอาต์พุตที่ j หรือ g_j

5.3 อัลกอริทึมจัดลำดับเซลล์ที่นำเสนอ

สำหรับอัลกอริทึมจัดลำดับเซลล์ส่งผ่านสวิตช์ที่ใช้สำหรับคำนวณหาค่าการแมตช์นั้น มีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 *คำร้องขอ* (request): $r_{i,j}P_i$

ขั้นตอนที่ 2 *คำยินยอม* (grant): $g_j = \max_i (r_{i,j}P_i)$

ขั้นตอนที่ 3 *ตอบรับคำยินยอม* (accept): $a_i = \max_j (g_j)$

โดยที่

$r_{i,j}$ คือคำร้องขอของอินพุตที่ *i* ที่จะไปยังเอาต์พุตที่ *j*

P_i คือลำดับความสำคัญที่อินพุตที่ i มีค่าลำดับความสำคัญสูงเมื่อ P_i = H หรือมีค่าลำดับความสำคัญต่ำเมื่อ P_i = L

g_j คือคำยินยอมที่ส่งไปให้อินพุตที่ *i* ของเอาต์พุตที่ *j*

a_i คือตอบรับคำยินยอมของเอาต์พุตที่ *j* ของอินพุตที่ *i*

โดยที่ขั้นตอนทั้ง 3 ขั้นตอนมีความหมาย ดังนี้

ขั้นตอนที่ 1 ***คำร้องขอ***: แต่ละอินพุตส่งคำร้องของไปยังทุก ๆ เอาต์พุตที่มีเซลล์อยู่ในคิวและจะไปยังเอาต์พุตนั้น ถ้าคำร้องขอมีลำดับความสำคัญ ให้อินพุตส่งลำดับความสำคัญเท่ากับค่าลำดับความสำคัญสูงสุดของเซลล์ที่อยู่ในคิวไปยังเอาต์พุตนั้น

ขั้นตอนที่ 2 ***คำยินยอม***: เมื่อแต่ละเอาต์พุตได้รับคำร้องขอของแต่ละอินพุต จะเลือกคำร้องขอที่มีค่าลำดับความสำคัญสูงสุด โดยมีตัวบ่งชี้ g_j^P ทำการเลือกคำร้องขอโดยวิธีวนรอบอินพุต (round-robin) และจะชี้อินพุตถัดไปเมื่อคำยินยอมได้รับการตอบรับคำยินยอมจากอินพุตนั้นแล้ว

ขั้นตอนที่ 3 ***ตอบรับคำยินยอม***: เมื่อแต่ละอินพุตได้รับคำยินยอมจะเลือกคำยินยอมที่ปรากฏว่ามีค่าลำดับความสำคัญสูงสุด โดยมีตัวบ่งชี้ a_i^P ทำการเลือกโดยวิธีวนรอบเอาต์พุต และจะชี้เอาต์พุตถัดไป เมื่อได้ตอบรับคำยินยอมและทำการเชื่อมต่อระหว่างอินพุตและเอาต์พุตนั้นแล้ว

หมายเหตุ 1) ถ้าเป็นอัลกอริทึมแบบวนซ้ำ ให้ทำซ้ำขั้นตอนที่ 2 และ 3 ในแต่ละรอบการวนซ้ำ

หมายเหตุ 2) สามารถแทนการเลือกโดยวิธีวนรอบเอาต์พุตหรืออินพุตของอัลกอริทึม iSLIP หรืออัลกอริทึม DRRM เป็นวิธีเลือกแบบสุ่มของอัลกอริทึม PIM ก็ได้ แล้วแต่ว่ามีวัตถุประสงค์ที่จะใช้อัลกอริทึมจัดลำดับเซลล์แบบใดร่วมกับเทคนิคเอนเวโลป

5.4 ทราฟฟิกที่ใช้ทดสอบ

ทราฟฟิกที่จะใช้ในการทดสอบกับแบบจำลองโครงสร้างสวิตช์ ที่นำเทคนิคเอนเวโลปมาประยุกต์ใช้ มี 2 ประเภท คือ ทราฟฟิกประเภทยูนิฟอร์ม และทราฟฟิกประเภทนอนยูนิฟอร์ม

1) ทราฟฟิกประเภทยูนิฟอร์ม

ทราฟฟิกประเภทยูนิฟอร์มนั้นจะต้องเป็นไปตามเงื่อนไขคือ การมาถึงของแพ็กเกตที่พอร์ตอินพุตของสวิตช์เป็นกระบวนการ i.i.d.Bernoulli และกระบวนการการมาถึงของแต่ละอินพุตจะต้องมีอัตราการมาถึงเดียวกัน และมีจุดหมายปลายทางที่จะไปยังเอาต์พุตแต่ละเอาต์พุตด้วยความน่าจะเป็นเท่า ๆ กัน (ยูนิฟอร์ม) เนื่องจากในความเป็นจริงไม่มีทราฟฟิกเป็นประเภทยูนิฟอร์ม ดังนั้น การที่เราจะทดสอบเฉพาะทราฟฟิกประเภทยูนิฟอร์มจึงยังไม่มีข้อมูลเพียงพอที่จะสรุปได้ว่าแบบจำลองโครงสร้างสวิตช์ที่ได้ออกแบบขึ้นนั้น สามารถแก้ปัญหาคอขวดได้จริง

2) ทราฟฟิกประเภทนอนยูนิฟอร์ม

ถ้าทราฟฟิกที่ไม่เป็นประเภทยูนิฟอร์มแล้วก็จะเป็นประเภทนอนยูนิฟอร์มทั้งสิ้น ดังนั้นทราฟฟิกประเภทนอนยูนิฟอร์มจึงมีรูปแบบมากมาย โดยเฉพาะอย่างยิ่งถ้าสวิตช์มีขนาดใหญ่ขึ้น จะมีรูปแบบของทราฟฟิกมากมายมหาศาล ตัวอย่างเช่น ถ้าเราพิจารณาสวิตช์ขนาด NxN กับการมาถึงพอร์ตอินพุตของแพ็กเกตเป็นกระบวนการ i.i.d.Bernoulli และยังได้จำกัดเงื่อนไขเพิ่มเติมโดยกำหนดให้

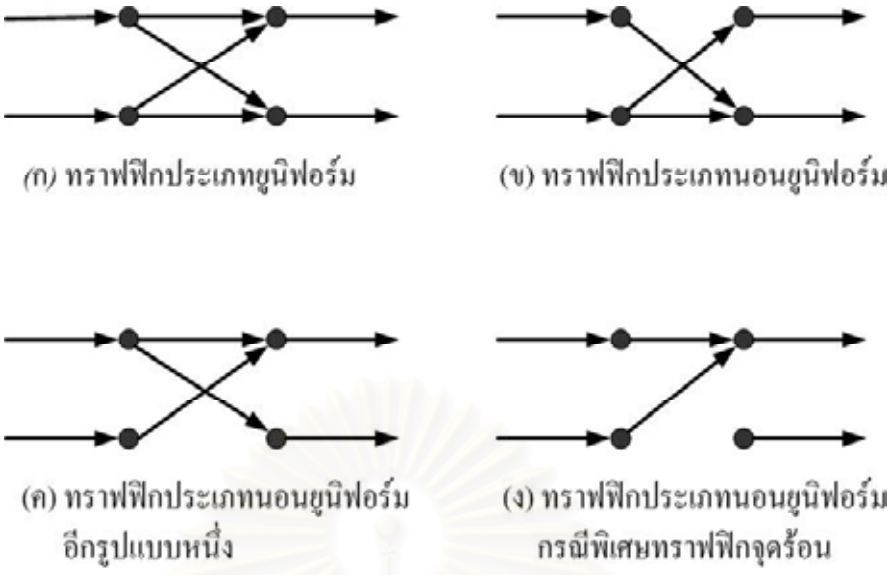
$$\sum_{i=1}^{N}\lambda_{i,j}=1,\sum_{j=1}^{N}\lambda_{i,j}=1 \text{ และ } \lambda_{i,j}=1,0 \qquad (5.1)$$

จะมีรูปแบบทราฟฟิกประเภทนอนยูนิฟอร์มให้พิจารณาถึง N! รูปแบบ ยิ่งถ้าไม่จำกัดเงื่อนไขแล้ว จะมีรูปแบบมากกว่าที่กล่าวมาเป็นจำนวนมาก

เพื่อให้เห็นภาพได้ชัดเจนขึ้น จะขอยกตัวอย่างสวิตช์ขนาด 2x2 ซึ่งมีรูปแบบของทราฟฟิกแสดงดังรูปที่ 5.2

ตามรูปที่ 5.2 (ก) ทราฟฟิกประเภทยูนิฟอร์ม (ข) และ (ค) ทราฟฟิกประเภทนอนยูนิฟอร์มโดยกำจัดด้านใดด้านด้านหนึ่งออกจากทราฟฟิกประเภทยูนิฟอร์ม และ (ง) ทราฟฟิกประเภทนอนยูนิฟอร์มที่มีชื่อเฉพาะว่า ทราฟฟิกจุดร้อน (hot-spot traffic)

เนื่องจากอัลกอริทึมจัดลำดับเซลล์ดังที่กล่าวมาแล้ว ส่วนใหญ่มีสมรรถนะที่ดีภายใต้เงื่อนไขทราฟฟิกเป็นประเภทยูนิฟอร์ม แต่เมื่อทราฟฟิกเป็นประเภทนอนยูนิฟอร์มแล้วจะทำให้ค่าทรูพุตของสวิตช์ลดลงอย่างมาก ซึ่งการที่ค่าทรูพุตลดลงเป็นผลกระทบโดยตรงต่อคุณภาพการให้บริการของสวิตช์



รูปที่ 5.2 ตัวอย่างทราฟฟิกประเภทยูนิฟอร์มและนอนยูนิฟอร์ม

เราไม่สามารถทดสอบทราฟฟิกประเภทนอนยูนิฟอร์มเป็นกรณีทั่วไปได้ เราทดสอบได้แต่เป็นกรณีพิเศษ เนื่องจากทราฟฟิกประเภทนอนยูนิฟอร์มแต่ละรูปแบบก็ให้สมรรถนะของสวิตช์ที่แตกต่างกันออกไป ดังจะพิจารณาโดยละเอียดในบทต่อไป ทราฟฟิกกรณีพิเศษที่เป็นประเภทนอนยูนิฟอร์มที่นิยมใช้ในการทดสอบก็คือทราฟฟิกจุดร้อน ซึ่งเซลล์ในแต่ละอินพุตจะถูกส่งไปยังเอาต์พุตใดเอาต์พุตหนึ่งของสวิตช์เพียงเอาต์พุตเดียว ทำให้เอาต์พุตนั้นรับภาระโหลดเกิน 100 เปอร์เซ็นต์ ค่าทรูพุตของสวิตช์จึงลดลงอย่างรวดเร็ว ตัวอย่างของทราฟฟิกจุดร้อน เช่น การที่ผู้ใช้บริการอินเทอร์เน็ตจะต้องติดต่อกับผู้ให้บริการอินเทอร์เน็ต เพื่อจะขอใช้บริการอินเทอร์เน็ตในช่วงเริ่มต้นการติดต่อ เป็นต้น

วิทยานิพนธ์ฉบับนี้ ได้นำเสนอทราฟฟิกจุดร้อนที่ใช้ในการทดสอบโดยมีรูปแบบ ดังนี้

<u>กรณีที่ 1</u> ทราฟฟิก $\rho = r\rho + (1-r)\rho$ โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนต่างกัน

ρ หมายถึงโหลดที่เข้ามาสู่พอร์ตอินพุตของสวิตช์และ r คือสัดส่วนของทราฟฟิกที่แต่ละอินพุตที่จะไปยังเอาต์พุตจุดร้อนที่เป็นเอาต์พุตที่แตกต่างกันทั้งหมด และ $1-r$ คือเซลล์อื่น ๆ ที่อินพุตแต่ละอินพุต ที่จะไปยังเอาต์พุตที่ไม่ใช่เอาต์พุตจุดร้อนด้วยการแจกแจงแบบยูนิฟอร์ม

<u>กรณีที่ 2</u> ทราฟฟิก $\rho = r\rho + (1-r)\rho$ โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนเดียวกัน

ρ หมายถึงโหลดที่เข้ามาสู่พอร์ตอินพุตของสวิตช์ และ r คือสัดส่วนของทราฟฟิกที่แต่ละอินพุตที่จะไปยังเอาต์พุตจุดร้อนที่เป็นเอาต์พุตเดียวกัน และ $1-r$ คือเซลล์อื่น ๆ ที่อินพุตแต่ละอินพุต ที่จะไปยังเอาต์พุตที่ไม่ใช่เอาต์พุตจุดร้อนด้วยการแจกแจงแบบยูนิฟอร์ม

5.5 ตัววัดสมรรถนะของสวิตช์

ตัวแปรที่ใช้ในการประเมินค่าสมรรถนะของสวิตช์ก็คือ คุณภาพการให้บริการ หรือ QoS ซึ่งประกอบไปด้วยพารามิเตอร์ ดังต่อไปนี้

1) ค่าทรูพุตหรือแบนด์วิดท์ของสวิตช์ (Throughput or Bandwidth)

2) ค่าเฉลี่ยการประวิงเวลาของเซลล์ (Average cell delay or Latency)

3) อัตราการสูญเสียเซลล์ (Cell loss rate)

ซึ่งพารามิเตอร์เหล่านี้ จะต้องมีค่าเป็นไปตามที่กำหนดไว้สำหรับรับประกันคุณภาพการให้บริการของสวิตช์ ดังนั้น อัลกอริทึมจัดลำดับเซลล์ตามแบบจำลองจะต้องมีสมรรถนะในการรับประกันค่าพารามิเตอร์ดังกล่าว

เนื่องจากวิทยานิพนธ์ฉบับนี้ได้กำหนดให้**บัฟเฟอร์มีขนาดไม่จำกัด** และในกรณีทราฟฟิกสองประเภทก็จะพิจารณาเฉพาะ**ลำดับความสำคัญเชิงเวลา** (time priority หรือ delay priority) ทำให้ไม่ต้องพิจารณาถึงอัตราการสูญเสียเซลล์ซึ่งถือว่าน้อยมากจนสามารถตัดทิ้งได้ เราจะใช้ตัววัดสมรรถนะเฉพาะค่าทรูพุตของสวิตช์และค่าเฉลี่ยการประวิงเวลาของเซลล์ ในการวัดสมรรถนะของสวิตช์ กรณีที่ค่าทรูพุตของสวิตช์เป็น 100 เปอร์เซ็นต์หรือเกือบจะ 100 เปอร์เซ็นต์ในทุกค่าทราฟฟิกนั้น เราจะแสดงเฉพาะกราฟค่าเฉลี่ยการประวิงเวลาของเซลล์เท่านั้น ซึ่งแฝงความหมายของค่าทรูพุตของสวิตช์อยู่แล้ว ตัวอย่างเช่น ถ้าค่าเฉลี่ยการประวิงเวลาของเซลล์ยังอยู่ในช่วงแกนตั้งของกราฟแสดงว่าค่าทรูพุตเป็น 100 เปอร์เซ็นต์หรือเกือบ 100 เปอร์เซ็นต์ ทำให้กราฟที่ได้จากการทดสอบจะแสดงแต่เพียงค่าเฉลี่ยการประวิงเวลาของเซลล์เป็นส่วนใหญ่

5.6 ความต้องการในการออกแบบสวิตช์แพ็กเกตความเร็วสูง

ในโครงข่ายอินเทอร์เน็ตแบ็กโบนนั้น สวิตช์ที่ใช้ในโครงข่ายเป็นสวิตช์ความเร็วสูง ซึ่งเหมาะสมกับแพ็กเกตความเร็วสูงที่ส่งผ่านกันในโครงข่ายการสวิตช์ ดังนั้น สวิตช์แพ็กเกตความเร็วสูง (High Speed Packet (HSP) switches) จึงมีความต้องการในการออกแบบ ดังนี้

1) การจัดการแบนด์วิดท์ (bandwidth management or scheduling algorithm)

2) การจัดการบัฟเฟอร์ (buffer management or buffering or queueing)

ในการจัดการแบนด์วิดท์อัลกอริทึมที่ใช้จัดลำดับเซลล์ส่งผ่านสวิตช์ก็คือ เทคนิคเอนเวโลป ส่วนในการจัดการบัฟเฟอร์นั้นจะใช้การรอคิวแบบ VOQ ซึ่งเป็นรูปแบบที่นิยมใช้กันในงานวิจัย

ของนักวิจัยต่าง ๆ และที่ใช้ในการออกแบบสวิตช์ที่ผลิตในทางการค้าในปัจจุบัน (เช่น Cisco 12000 series router [16]) โดยมีรูปแบบและการทำงานดังที่กล่าวมาแล้วในบทก่อน

5.7 การจัดการบัฟเฟอร์ที่นำเสนอ

เมื่อเราใช้สวิตช์แบบรอคิวที่เอาต์พุตนั้น เราจะใช้การรอคิวของเซลล์ก่อนส่งออกจากสวิตช์โดยวิธี FIFO แต่เมื่อเราเปลี่ยนมาใช้สวิตช์แบบรอคิวที่อินพุต การใช้วิธี FIFO จะทำให้สมรรถนะของสวิตช์ลดลงอย่างมาก เนื่องจากถึงแม้ว่าบางเอาต์พุตจะว่างลงแต่เซลล์ที่อินพุตก็ยังไม่สามารถส่งไปยังเอาต์พุตนั้นได้เนื่องจากอยู่หลังเซลล์ที่ไม่ได้ไปยังเอาต์พุตเดียวกัน ดังที่อธิบายมาก่อนหน้านี้แล้ว แต่มาได้มีการแก้ปัญหานี้โดยใช้วิธีวินโดว์อิง โดยให้พิจารณาส่งผ่านสวิตช์ได้ถึง k เซลล์ในแต่ละคิวของอินพุต แต่ก็ทำให้อัลกอริทึมยุ่งยากซับซ้อนมากและเสียเวลาประมวลผลมากขึ้น จึงได้มีการพัฒนาต่อมาจนเป็นการรอคิวแบบ VOQ โดยใช้การแยกคิวแต่ละคิวที่จะไปยังแต่ละเอาต์พุตที่อินพุตในทางตรรกะ ซึ่งได้นำไปใช้ในทางการค้า และในงานวิจัยด้านสวิตช์จำนวนมากในปัจจุบัน เราจึงได้นำการรอคิวแบบ VOQ มาใช้ในการจัดการคิวในวิทยานิพนธ์ฉบับนี้ด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
ผลการทดสอบ

6.1 อัลกอริทึมจัดลำดับเซลล์ที่ใช้ร่วมกับเทคนิคเอนเวโลป

เทคนิคในการเลือกเซลล์ส่งผ่านโครงสร้างการสวิตช์แบบครอสบาร์ ที่มีใช้กันอยู่ตั้งแต่อดีตจนถึงปัจจุบัน หลัก ๆ แล้วมี 3 วิธีคือ

1) วิธีมาถึงก่อนได้รับบริการก่อน (FIFO)

2) วิธีสุ่ม (random) ด้วยอัลกอริทึม PIM

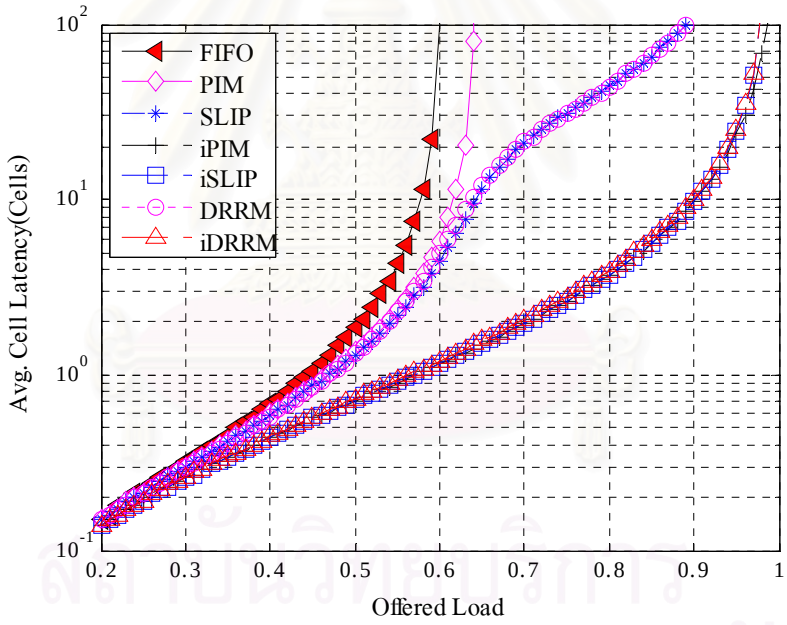
3) วิธีวนรอบ (round-robin) ด้วยอัลกอริทึม iSLIP และ DRRM

จะทำการเปรียบเทียบ โดยดูผลจากกราฟซึ่งเป็นผลจากการจำลองแบบด้วยโปรแกรมภาษา C++ ตามรูปที่ 5.3

ตามรูปที่ 5.3 จะเห็นว่าอัลกอริทึม PIM และอัลกอริทึม iSLIP ให้ค่าเฉลี่ยการประวิงเวลาเกือบเท่ากันและดีกว่าอัลกอริทึม FIFO มาก ถึงแม้ว่าอัลกอริทึม PIM ซึ่งใช้วิธีเลือกเซลล์ส่งผ่านสวิตช์แบบสุ่มจะใช้เวลาประมวลผลในแต่ละครั้งนานกว่าอัลกอริทึม iSLIP ซึ่งใช้วิธีเลือกเซลล์แบบวนรอบแต่ละอินพุต แต่เนื่องจากเทคนิคเอนเวโลปนั้นเราพิจารณาเลือกเซลล์ส่งผ่านสวิตช์ทีละเอนเวโลป เมื่อพิจารณาในแง่ของการส่งเซลล์ผ่านโครงสร้างการสวิตช์ทีละเอนเวโลป จึงสามารถนำเอาอัลกอริทึม iSLIP และ PIM มาประยุกต์ใช้กับเทคนิคเอนเวโลปได้ แต่เนื่องจากอัลกอริทึม PIM ใช้เวลาในการประมวลผลเพื่อหาค่าการแมตช์นานกว่าอัลกอริทึม iSLIP ถ้าจะนำไปใช้ร่วมกับเทคนิคเอนเวโลปจะต้องกำหนดให้ขนาดเอนเวโลปยาวกว่าการใช้อัลกอริทึม iSLIP ซึ่งทำให้ค่าเฉลี่ยการประวิงเวลาสูงกว่า ด้วยเหตุนี้อัลกอริทึม iSLIP จึงมีความเหมาะสมมากกว่าที่จะนำไปใช้ร่วมกับเทคนิคเอนเวโลปเพราะให้ค่าเฉลี่ยการประวิงเวลาต่ำกว่าอัลกอริทึม PIM ซึ่งเป็นการวิเคราะห์ขณะที่ยังไม่ได้นำไปใช้ร่วมกับเทคนิคเอนเวโลป

อัลกอริทึม DRRM และอัลกอริทึม iSLIP ให้ค่าเฉลี่ยการประวิงเวลาเท่ากันในทุกย่านทราฟฟิก อัลกอริทึม iSLIP ใช้เทคนิคทางฮาร์ดแวร์ในการลดเวลาประมวลผลหาค่าการแมตช์ (เทคนิคไปป์ไลน์) ส่วนอัลกอริทึม DRRM ใช้เทคนิคทางซอฟต์แวร์ในการลดเวลาในการประมวลผลหาค่าการแมตช์ โดยใช้การสื่อสารแบบสองทางด้วยการเริ่มการวนซ้ำในขั้นตอนการร้องขอแทนที่จะเป็นขั้นตอนการตอบรับคำยินยอมเหมือนดังเช่นอัลกอริทึม iSLIP เราจึงสามารถที่จะนำเอาอัลกอริทึม iSLIP และอัลกอริทึม DRRM ไปใช้ร่วมกับเทคนิคเอนเวโลปได้เช่นเดียวกัน

ตามรูปที่ 6.1 อัลกอริทึม PIM, iSLIP และ DRRM ให้สมรรถนะด้านการประวิงเวลาของสวิตช์ไม่แตกต่างกันในทุก ๆ ย่านทราฟฟิก และเนื่องจากการนำเทคนิคเอนเวโลปมาประยุกต์ใช้นั้นมีการเพิ่มจำนวนเซลล์ที่จะส่งผ่านโครงสร้างการสวิตช์ ในการประมวลผลหาค่าการแมตช์ในแต่ละครั้ง ซึ่งมีลักษณะแปลกแยกออกไปจากการวิเคราะห์ผลที่ได้จากกราฟตามรูปที่ 6.1 ซึ่งการนำเอาอัลกอริทึมจัดลำดับเซลล์มาใช้ร่วมกับเทคนิคเอนเวโลป อาจจะทำให้สมรรถนะด้านการประวิงเวลาของสวิตช์เปลี่ยนแปลงไปจากผลที่ได้จากกราฟ ดังนั้น ในการวิเคราะห์สมรรถนะของสวิตช์เมื่อนำเอาเทคนิคเอนเวโลปมาประยุกต์ใช้ จึงนำเอาอัลกอริทึมจัดลำดับเซลล์ทั้ง 3 อัลกอริทึมดังกล่าวมาวิเคราะห์หาสมรรถนะของสวิตช์ ทั้งนี้ก็มีวัตถุประสงค์ที่จะหาอัลกอริทึมจัดลำดับเซลล์ที่เหมาะสมที่สุดเมื่อนำไปใช้ร่วมกับเทคนิคเอนเวโลป



รูปที่ 6.1 สมรรถนะด้านการประวิงเวลาของอัลกอริทึมจัดลำดับเซลล์แบบต่างๆ

6.2 วิธีเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์

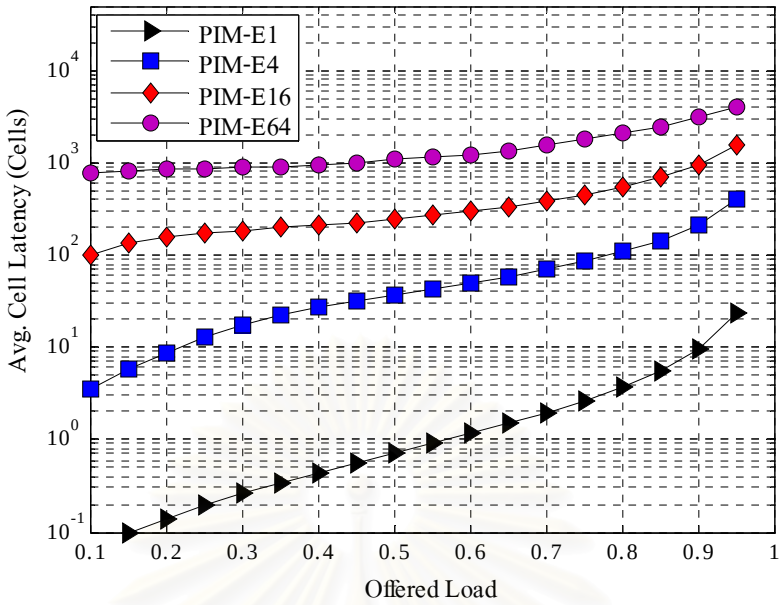
วิธีที่ 1 *เลือกเอนเวโลปที่มีเซลล์ที่หัวคิว* *(Partially-Filled (PF) envelope)*

ผลการจำลองแบบอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ ตามรูปที่ 6.2 6.3 และ 6.4 จะให้ผลคล้ายคลึงกันคือ การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว จะให้ค่าเฉลี่ยการประวิงเวลาที่สูงขึ้นเมื่อเอนเวโลปมีขนาดความยาวมากขึ้นซึ่งเราไม่สามารถหลีกเลี่ยงได้ เราทำได้แต่เพียงกำหนดให้ขนาดเอนเวโลปต่ำที่สุดเท่าที่จะทำให้สวิตช์หาค่าการแมตช์ได้ทัน ตัวอย่างเช่น อัลกอริทึม iSLIP ใช้เวลาในการหาค่าการแมตช์ในแต่ละรอบประมาณ 10 ns ถ้านำไปใช้ในโครงข่ายที่มีข่ายเชื่อมโยงมีความเร็วในการรับส่งข้อมูล 40 Gbps เราจะต้องใช้เอนเวโลปขนาด 2 เซลล์ เป็นต้น

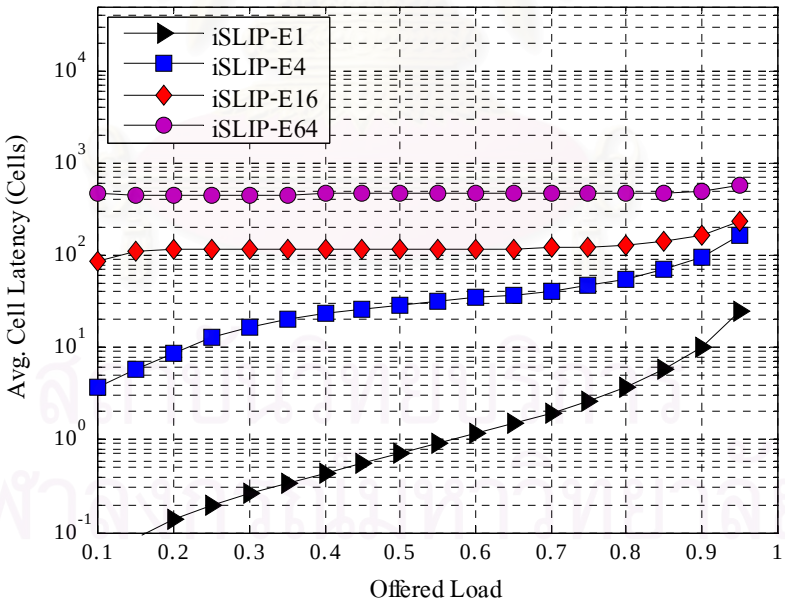
ตามรูปที่ 6.5 แสดงการเปรียบเทียบค่าเฉลี่ยการประวิงเวลาของสวิตช์ จะเห็นว่าที่ทราฟฟิกต่ำอัลกอริทึมทั้งสามแบบให้ค่าเฉลี่ยการประวิงเวลาเท่ากัน แต่ที่ทราฟฟิกสูงอัลกอริทึม iSLIP ให้ค่าเฉลี่ยการประวิงเวลาต่ำกว่าอัลกอริทึมอื่น ๆ เล็กน้อย สรุปว่าถ้าเราเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว อัลกอริทึม iSLIP จะเป็นอัลกอริทึมที่เหมาะสมที่สุดสำหรับนำไปใช้ร่วมกับเทคนิคเอนเวโลป

ตามรูปที่ 6.6 แสดงให้เห็นว่าในกรณีที่ทราฟฟิกต่ำ เซลล์ที่รออยู่ในคิวจำนวน 1 เซลล์ มีมากที่สุด และที่ทราฟฟิกสูงเซลล์ที่รออยู่ในคิวจำนวน 3 เซลล์มีมากที่สุด ผลจากกราฟแสดงให้เห็นว่าที่ทราฟฟิกต่ำควรใช้ขนาดเอนเวโลปน้อย ตัวอย่างเช่นที่ทราฟฟิก 0.2 ควรใช้ขนาดเอนเวโลปเท่ากับ 1 เซลล์ และที่ทราฟฟิกสูง ควรใช้ขนาดเอนเวโลปมากขึ้น ตัวอย่างเช่นที่ทราฟฟิก 0.8 ควรใช้ขนาดเอนเวโลปเท่ากับ 3 เซลล์ จึงจะทำให้การสูญเสียแบนด์วิดท์ในการสวิตช์เนื่องจากเทคนิคเอนเวโลปลดน้อยลง

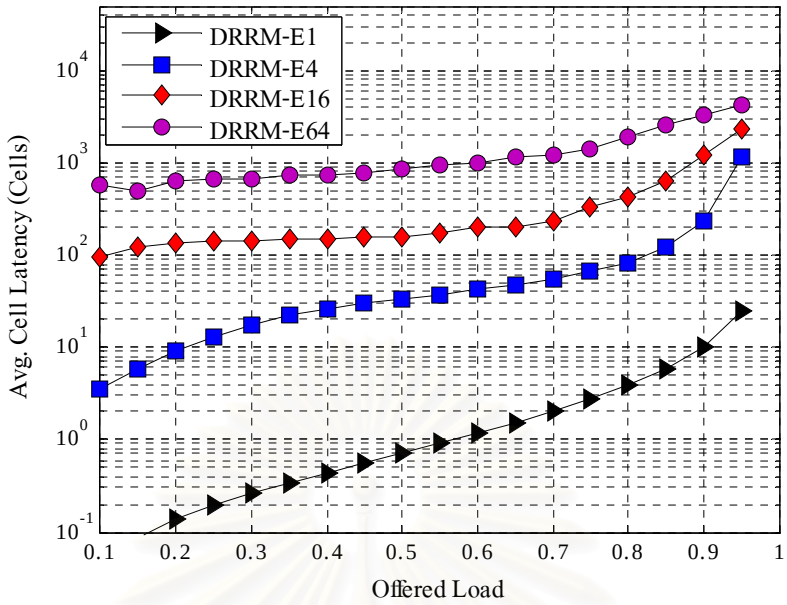
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



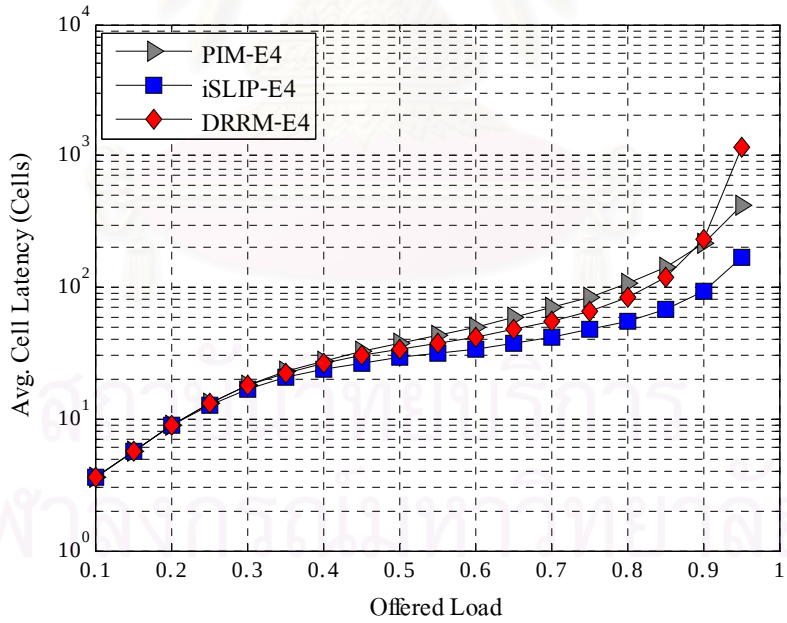
รูปที่ 6.2 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม PIM



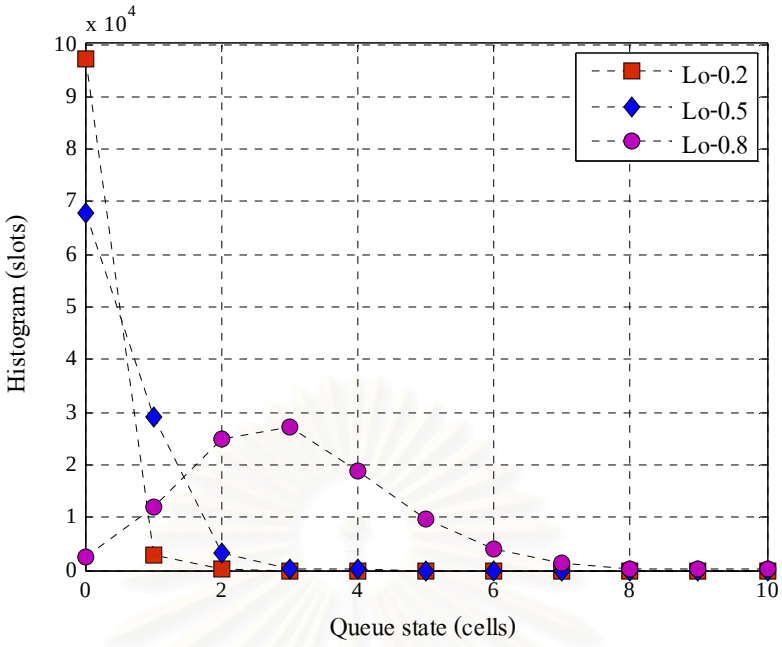
รูปที่ 6.3 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม iSLIP



รูปที่ 6.4 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 1 ของอัลกอริทึม DRRM



รูปที่ 6.5 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของอัลกอริทึมต่าง ๆ ที่ขนาดเอนเวโลปเท่ากับ 4 เซลล์

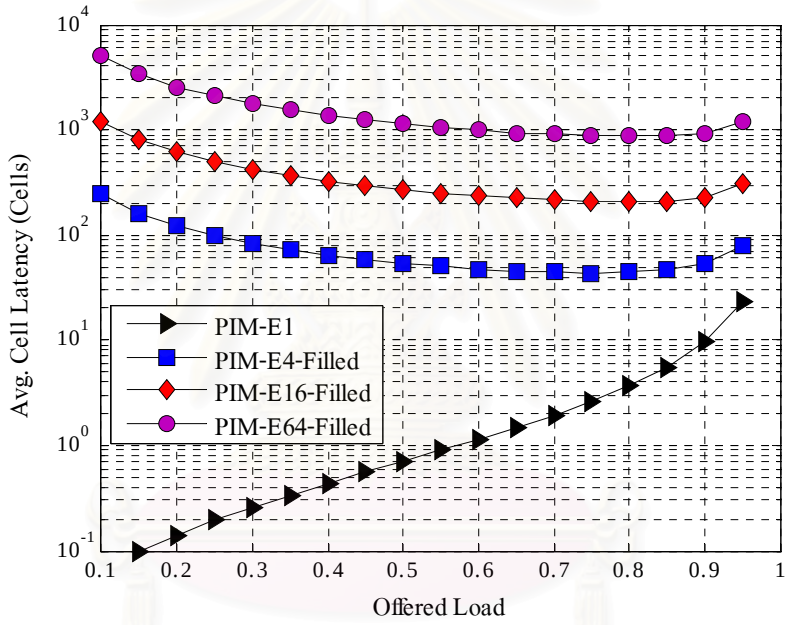


รูปที่ 6.6 แสดงฮิสโทแกรมของอินพุตบัฟเฟอร์ที่โหลดขนาดต่าง ๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

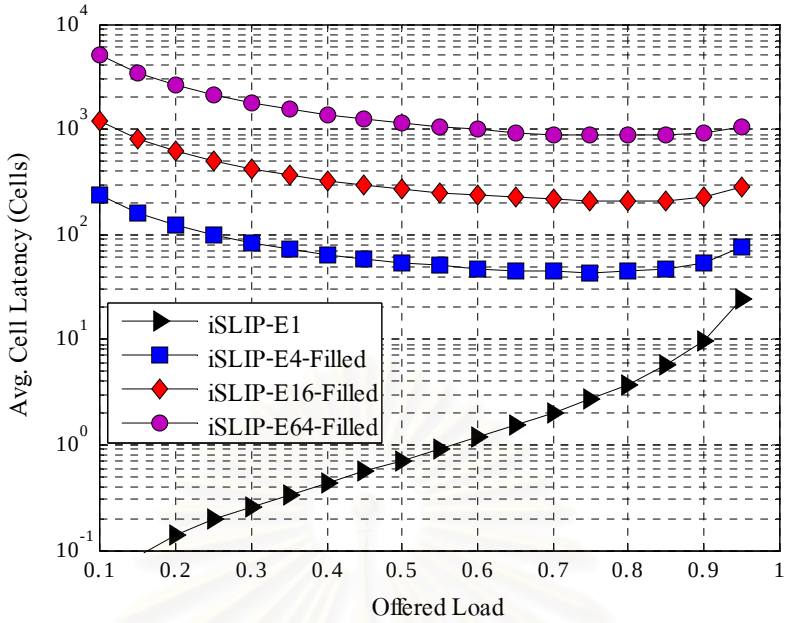
วิธีที่ 2 *เลือกเอนเวโลปที่มีเซลล์บรรจุเต็ม* *(Filled (F) envelope)*

ผลการจำลองแบบอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ ตามรูปที่ 6.7 6.8 และ 6.9 จะให้ผลอย่างเดียวกันคือ การเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่เต็มนั้น ให้ค่าเฉลี่ยการประวิงเวลาสูงมากในช่วงทราฟฟิกต่ำ ๆ จึงไม่สามารถรองรับค่า QoS ได้ ยิ่งขนาดเอนเวโลปมีความยาวมากขึ้นก็จะมีค่าเฉลี่ยไทม์ดีเลย์สูงขึ้นมากเป็นอัตราส่วนสัมพันธ์กัน จึงไม่เหมาะสมที่จะนำวิธีนี้มาใช้กับสวิตช์ซึ่งจะต้องรองรับการเพิ่มขนาดของสวิตช์ในอนาคต

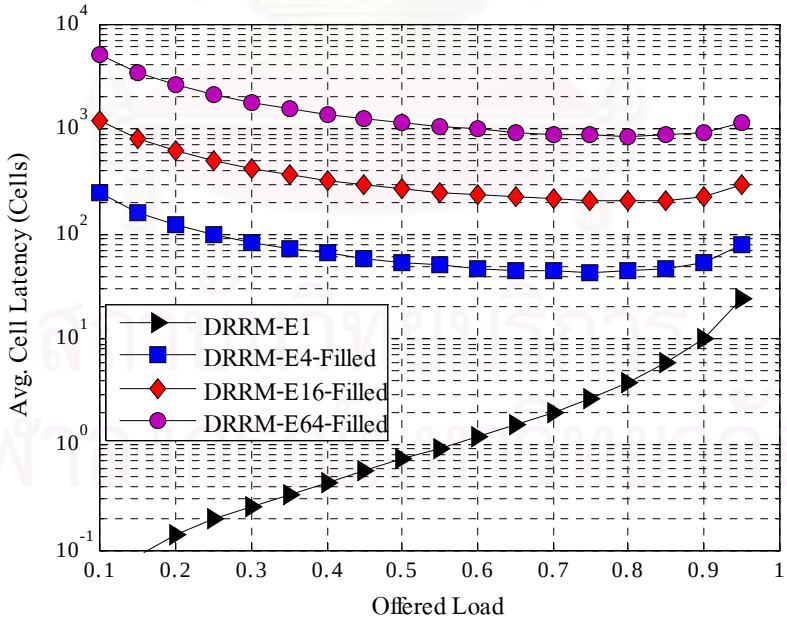


รูปที่ 6.7 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม PIM

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



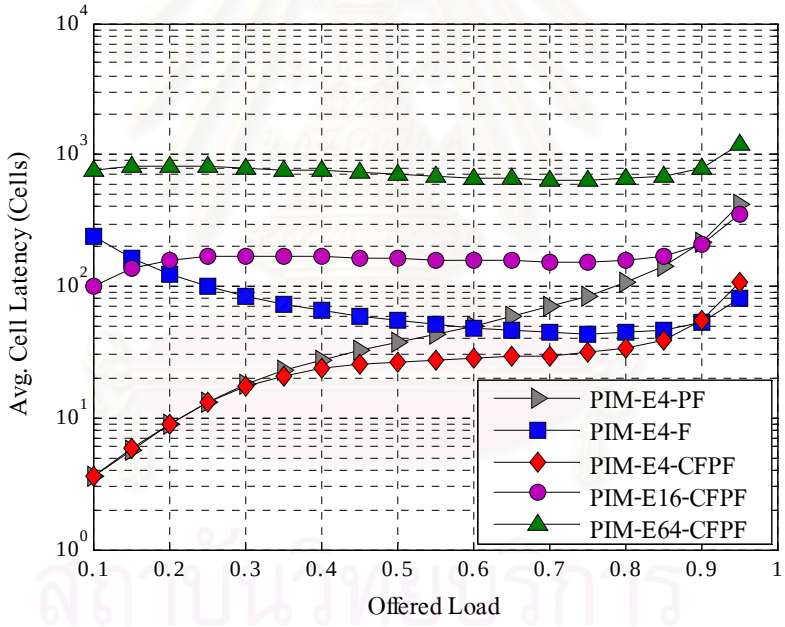
รูปที่ 6.8 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม iSLIP



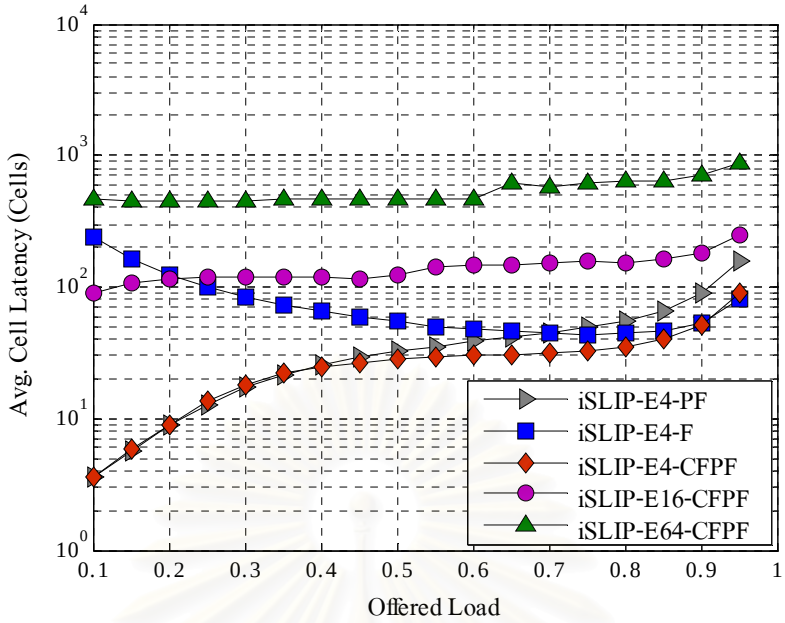
รูปที่ 6.9 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 2 ของอัลกอริทึม DRRM

วิธีที่ 3 *เลือกเอนเวโลปโดยวิธีผสมผสาน* *(Combined Filled and Partially-Filled (CFPF) envelope)*

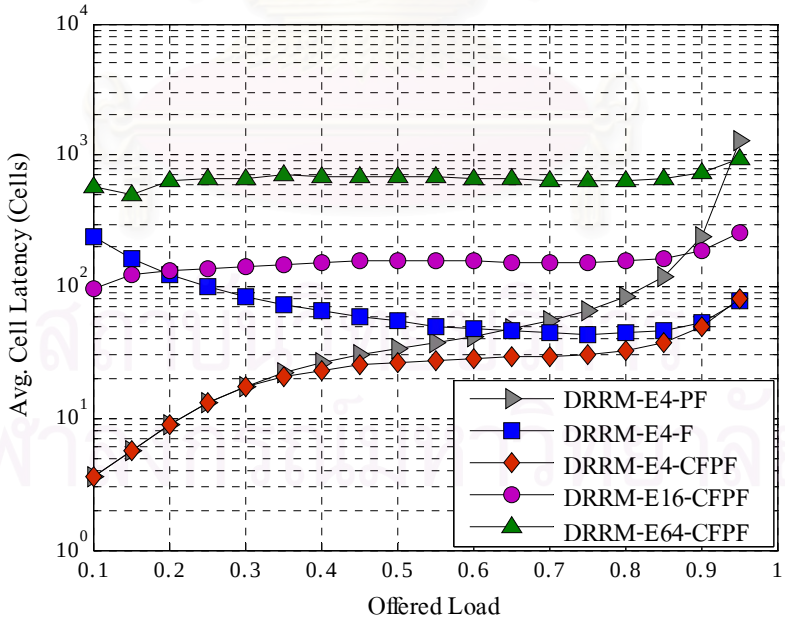
ผลการจำลองแบบอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ ตามรูปที่ 6.10 6.11 และ 6.12 พร้อมทั้งเปรียบเทียบกับวิธีที่ 1 และวิธีที่ 2 จะได้ว่าการเลือกวิธีที่ 3 จะให้ค่าเฉลี่ยการประวิงเวลาดีกว่าวิธีอื่น ๆ แต่วิธีที่ 3 จะต้องใช้เวลาในการหาค่าการแมตช์นานกว่าวิธีอื่น เนื่องจากต้องหาค่าการแมตช์ทั้งการเลือกเอนเวโลปบรรจุเต็มก่อน แล้วจึงเลือกเอนเวโลปที่มีเซลล์ที่หัวคิวตามลำดับด้วย ปริมาณงานจึงเพิ่มเป็นสองเท่าของวิธีที่ 1 หรือวิธีที่ 2 ซึ่งจะส่งผลกระทบโดยตรงต่อขนาดของเอนเวโลปที่จะใช้ในการส่งผ่านสวิตช์ซึ่งจะต้องมีขนาดเพิ่มขึ้น ขนาดเอนเวโลปที่เพิ่มขึ้นทำให้ค่าเฉลี่ยการประวิงเวลาสูงกว่าวิธีที่ 1 และวิธีที่ 2 ในทุกย่านทราฟฟิก



รูปที่ 6.10 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม PIM



รูปที่ 6.11 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม iSLIP



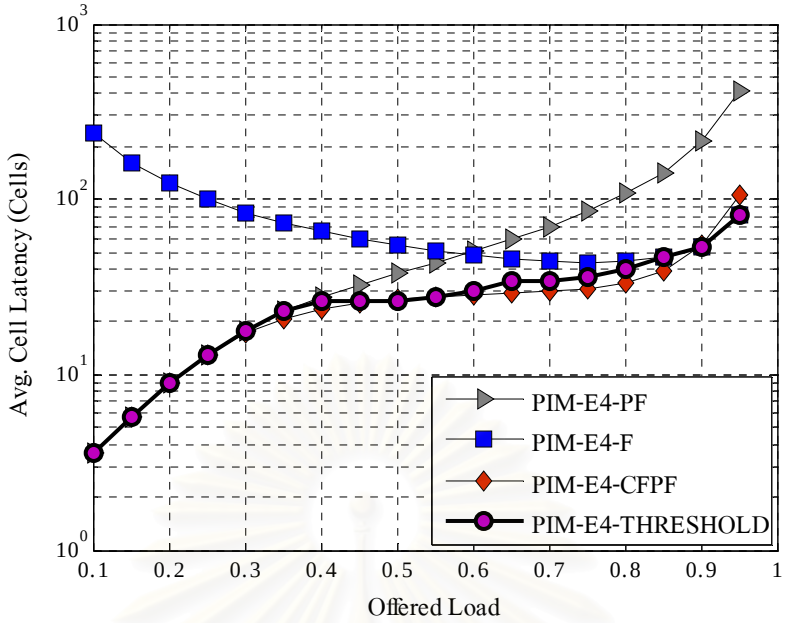
รูปที่ 6.12 เปรียบเทียบสมรรถนะด้านการประวิงเวลาเมื่อใช้วิธีที่ 3 ของอัลกอริทึม DRRM

วิธีที่ 4 *เลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต* *(Dynamic threshold envelope)*

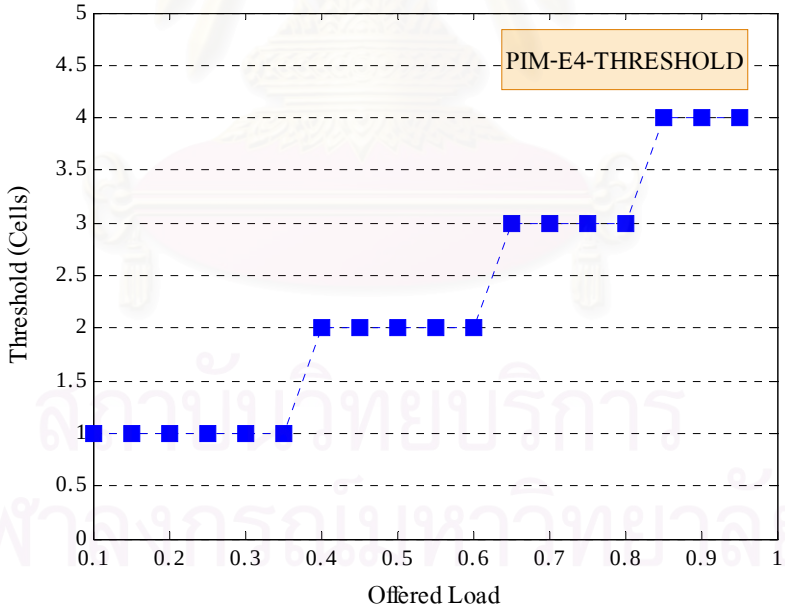
ในการเลือกเอนเวโลปส่งผ่านโครงสร้างการสวิตช์ในแต่ละไทม์สล็อต จะเห็นว่าวิธีเลือกเอนเวโลปโดยพิจารณาแต่เพียงเอนเวโลปที่มีเซลล์ที่หัวคิวหรือไม่นั้น ให้ค่าเฉลี่ยการประวิงเวลาต่ำกว่าวิธีอื่นที่ทราฟฟิกต่ำ แต่ให้ค่าเฉลี่ยการประวิงเวลาสูงกว่าวิธีอื่นเมื่อทราฟฟิกสูง ส่วนการเลือกเอนเวโลปโดยวิธีเลือกเอนเวโลปบรรจุเต็มนั้น ให้ค่าเฉลี่ยการประวิงเวลาสูงกว่าวิธีอื่นที่ ทราฟฟิกต่ำและให้ค่าเฉลี่ยการประวิงเวลาต่ำใกล้เคียงกับวิธีอื่นที่ทราฟฟิกสูง ดังนั้น เราจึงเอาข้อดีและข้อเสียของวิธีเลือกเอนเวโลปทั้งสองวิธีมาพิจารณา วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการเลือกเอนเวโลปที่เรียกว่าวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต ซึ่งมีแนวคิดว่าการเลือกเอนเวโลปนั้นจะพิจารณาเลือกเอนเวโลปที่มีเซลล์บรรจุอยู่บางส่วนที่ไม่ต่ำกว่าค่าเทรสโฮลด์ที่กำหนดไว้ และค่าเทรสโฮลด์มีการเปลี่ยนแปลงตามค่าทราฟฟิก ทั้งนี้ก็มีวัตถุประสงค์ที่จะให้สวิตช์มีแบนด์วิดท์ในการสวิตช์สูงสุด ซึ่งทำให้ค่าเฉลี่ยการประวิงเวลาดีกว่าทั้งสองวิธีที่กล่าวมาแล้ว และเมื่อเปรียบเทียบกับการเลือกเอนเวโลปโดยวิธีผสมผสาน จะเห็นว่าให้สมรรถนะด้านการประวิงเวลาใกล้เคียงกัน แต่การเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต ใช้เวลาในการหาค่าการแมตช์น้อยกว่าการเลือกเอนเวโลปโดยวิธีผสมผสาน

ตามรูปที่ 6.13 6.14 และ 6.15 แสดงผลการจำลองแบบอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ ที่ใช้ร่วมกับการเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต ในการเลือกเอนเวโลปส่งผ่านโครงสร้างการสวิตช์ ซึ่งให้ค่าสมรรถนะทางด้านการประวิงเวลาของสวิตช์ที่สอดคล้องกันคือ จะให้ค่าเฉลี่ยการประวิงเวลาดีกว่าวิธีที่ 1 และ 2 และให้ค่าเฉลี่ยการประวิงเวลาใกล้เคียงกับวิธีที่ 3

ตามรูปที่ 6.16 6.17 และ 6.18 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ โดยเลือกเอนเวโลปขนาดต่าง ๆ กันของอัลกอริทึมจัดลำดับเซลล์แบบต่าง ๆ ที่ใช้ร่วมกับการเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต ซึ่งให้ผลจากกราฟเป็นไปในทิศทางเดียวกันคือ จะเห็นว่ายิ่งเพิ่มความยาวเอนเวโลปมากขึ้นเท่าใด ค่าเฉลี่ยการประวิงเวลาก็จะเพิ่มมากขึ้นเป็นสัดส่วนสัมพันธ์กัน เพื่อที่จะรับประกันในคุณภาพการให้บริการเราจึงไม่สามารถเพิ่มขนาดเอนเวโลปตามใจชอบได้ เราจะต้องระมัดระวังไม่เพิ่มขนาดเอนเวโลปจนค่าเฉลี่ยการประวิงเวลาสูงเกินไป จนไม่สามารถรับประกันในคุณภาพการให้บริการได้

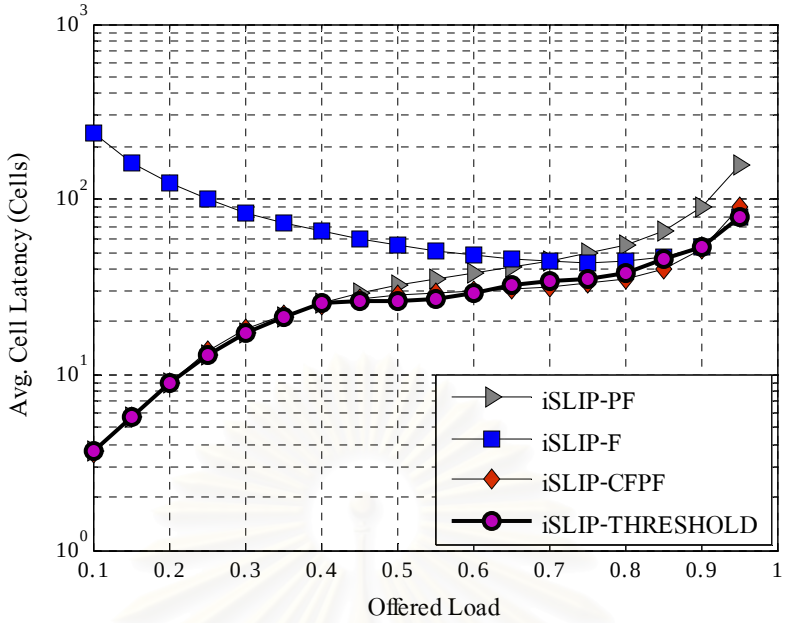


(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลา เอนเวโลปขนาด 4 เซลล์

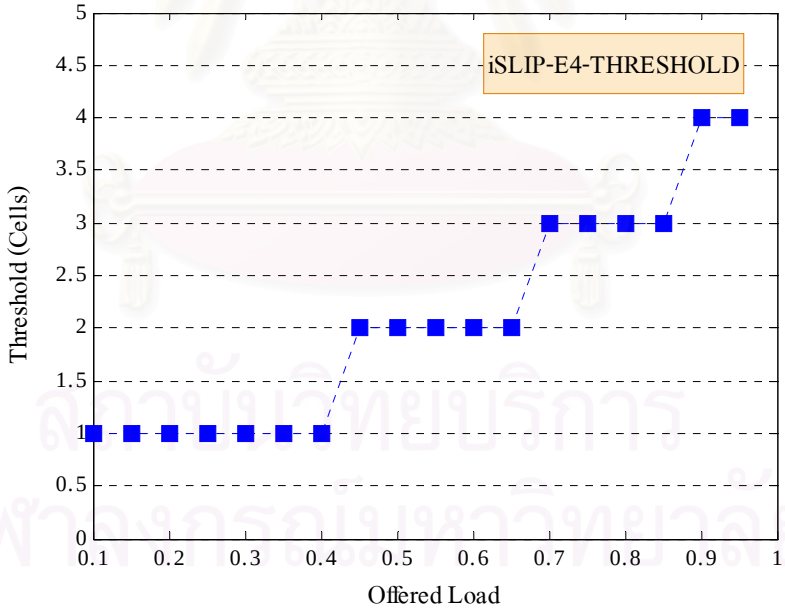


(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.13 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปวิธีต่าง ๆ ของอัลกอริทึม PIM

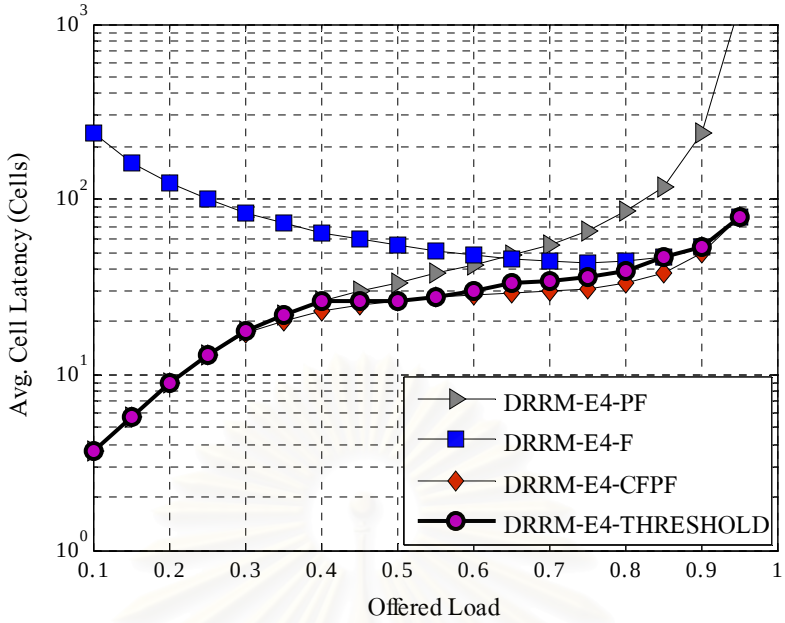


(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลา เอนเวโลปขนาด 4 เซลล์

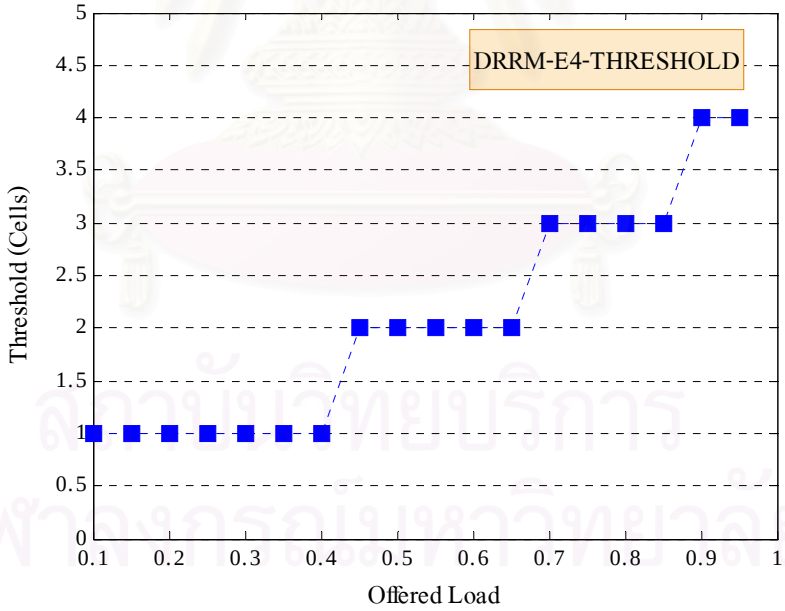


(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.14 เปรียบเทียบสมรรถนะด้านการประวิงเวลาโดยเลือกเอนเวโลปวิธีต่าง ๆ ของอัลกอริทึม iSLIP

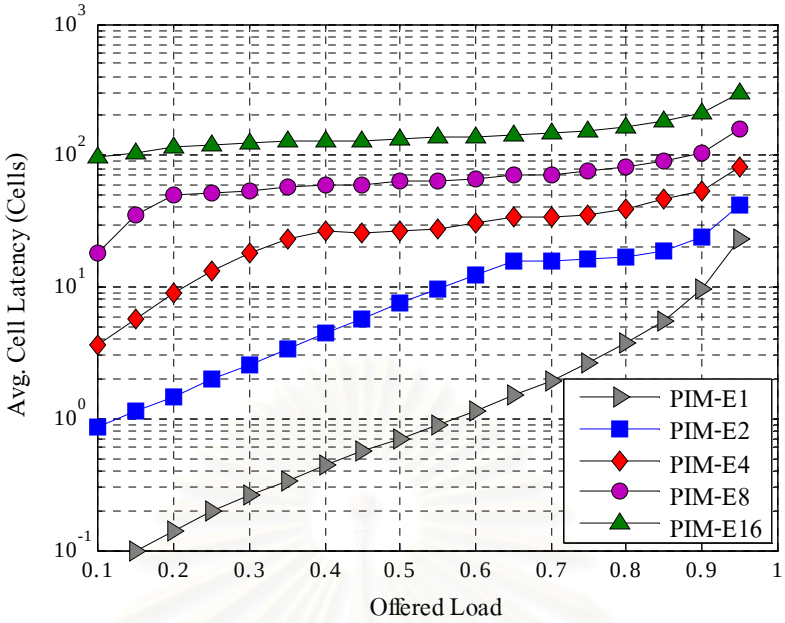


(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลา เอนเวโลปขนาด 4 เซลล์

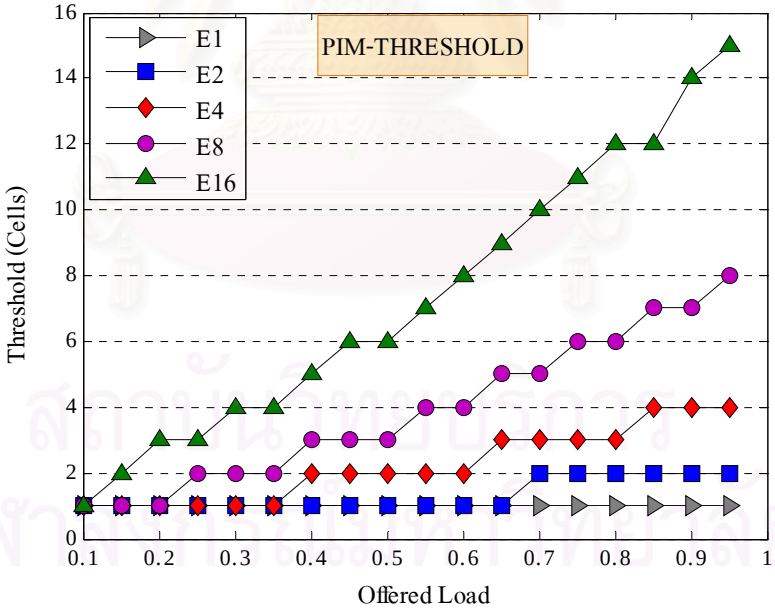


(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.15 เปรียบเทียบสมรรถนะด้านการประวิงเวลา โดยเลือกเอนเวโลปวิธีต่าง ๆ ของอัลกอริทึม DRRM

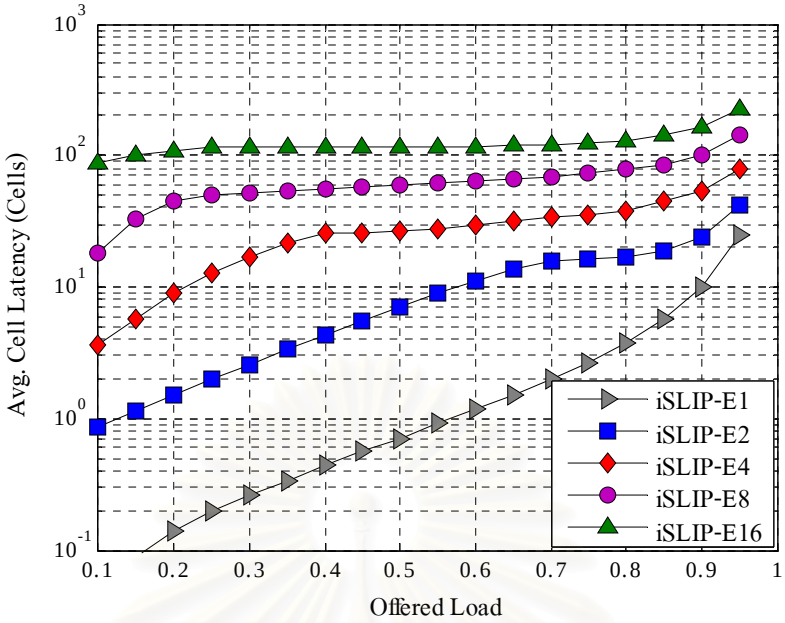


(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาที่เอนเวโลปขนาดต่าง ๆ

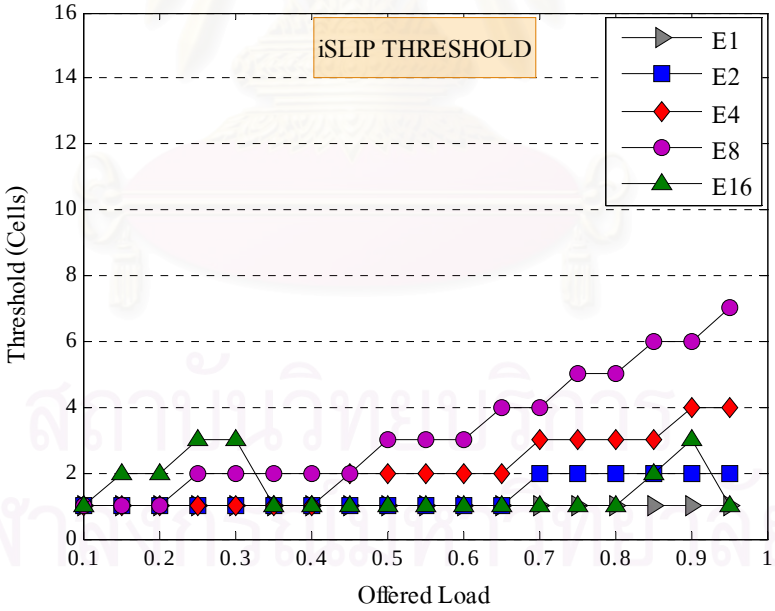


(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.16 เปรียบเทียบสมรรถนะด้านการประวิงเวลา โดยเลือกเอนเวโลปขนาดต่าง ๆ ของอัลกอริทึม PIM

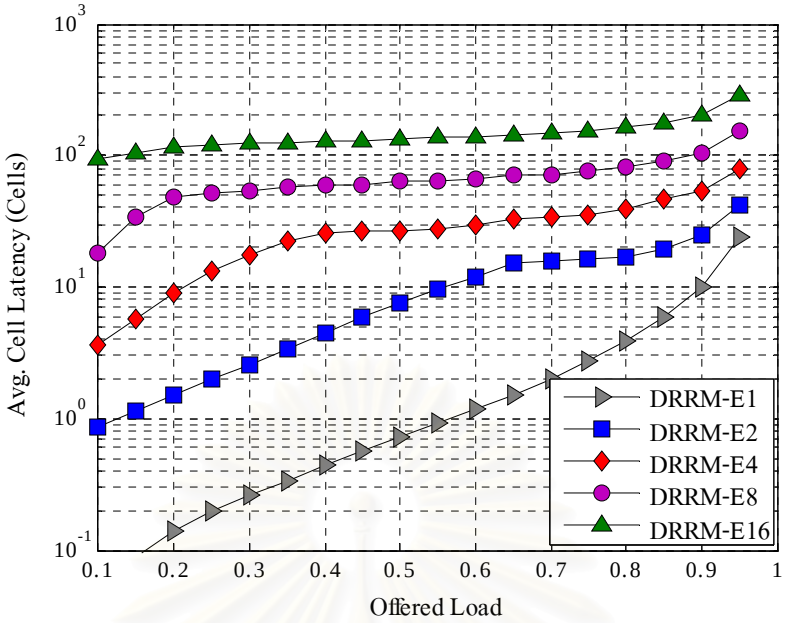


(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาที่เอนเวโลปขนาดต่าง ๆ

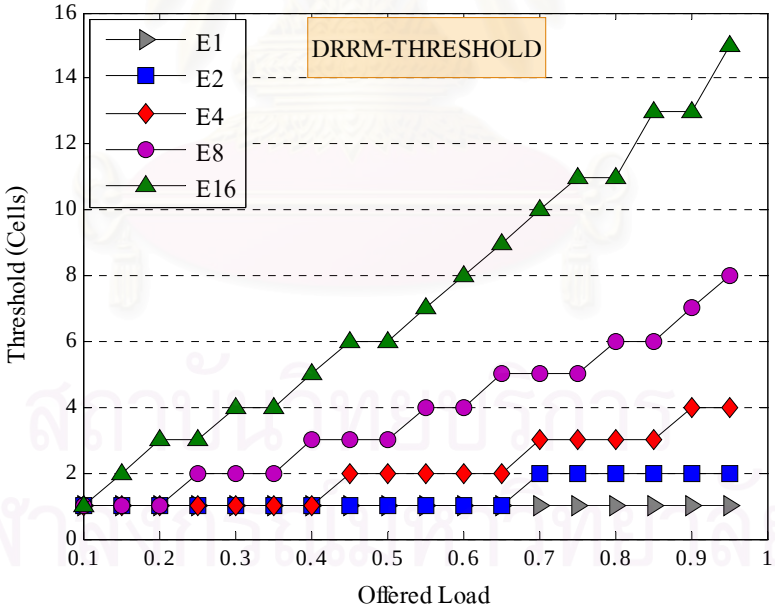


(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.17 เปรียบเทียบสมรรถนะด้านการประวิงเวลา โดยเลือกเอนเวโลปขนาดต่าง ๆ ของอัลกอริทึม iSLIP



(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาที่เอนเวโลปขนาดต่าง ๆ



(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

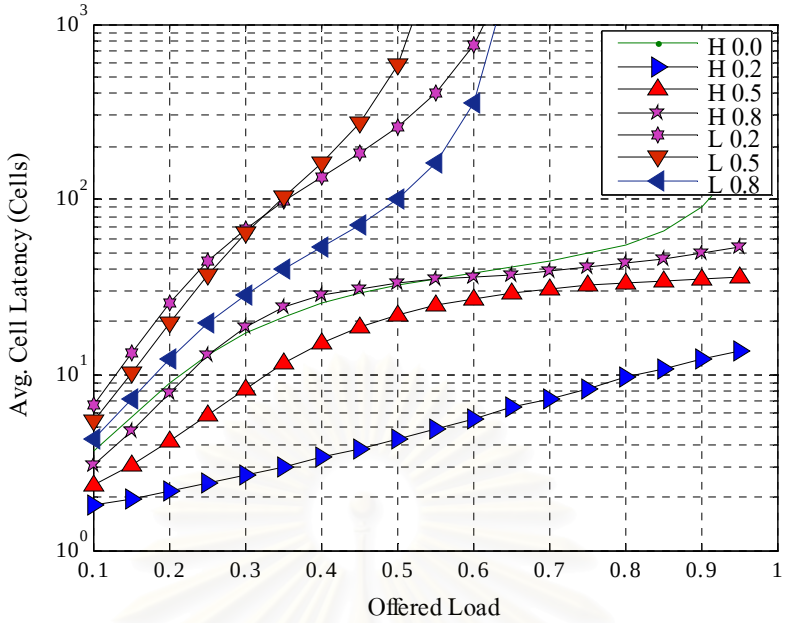
รูปที่ 6.18 เปรียบเทียบสมรรถนะด้านการประวิงเวลา โดยเลือกเอนเวโลปขนาดต่าง ๆ ของอัลกอริทึม DRRM

6.3 กรณีศึกษา: วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองประเภท

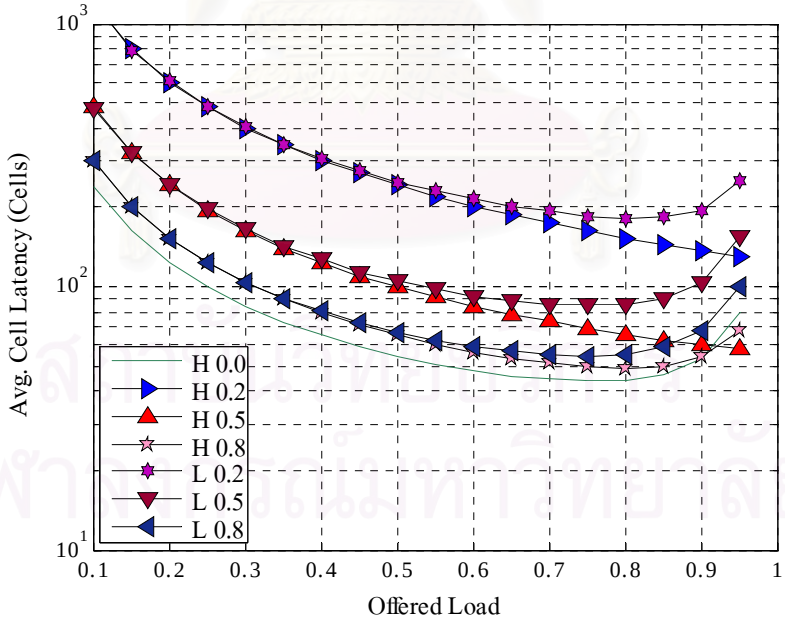
ตามรูปที่ 6.19 กำหนดให้ H แทนทราฟฟิกที่มีลำดับความสำคัญสูงหรือทราฟฟิกประเภทเรียลไทม์ และ L แทนทราฟฟิกที่มีลำดับความสำคัญต่ำหรือทราฟฟิกประเภทดาตา จะเห็นได้ว่าการใช้การเลือกเอนเวโลปโดยวิธีที่ 1 นั้น สวิตช์สามารถรับประกันคุณภาพการให้บริการสำหรับทราฟฟิกลำดับความสำคัญสูง แต่กรณีที่ทราฟฟิกลำดับความสำคัญต่ำนั้นไม่สามารถรับประกันคุณภาพการให้บริการได้ ส่วนตามรูปที่ 6.20 นั้นให้ค่าเฉลี่ยการประวิงเวลาสูงทั้งย่านทราฟฟิกต่ำและทราฟฟิกสูง โดยเฉพาะทราฟฟิกที่มีลำดับความสำคัญต่ำ สวิตช์จึงไม่สามารถรับประกันในคุณภาพการให้บริการได้

ตามรูปที่ 6.21 เราได้นำเสนอการเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัตสามารถรับประกันคุณภาพการให้บริการในทุกย่านทราฟฟิก โดยดูผลจากกราฟซึ่งให้ค่าเฉลี่ยการประวิงเวลาใกล้เคียงกับการเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต ที่เซลล์ไม่มีการจัดแบ่งตามลำดับความสำคัญ กล่าวโดยสรุปการเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัตสามารถรองรับได้ทั้งกรณีที่ไม่มีการจัดแบ่งและมีการจัดแบ่งประเภทของทราฟฟิกตามลำดับความสำคัญ ดังนั้น สวิตช์หรือเราเตอร์ที่นำเทคนิคเอนเวโลปไปใช้ จึงสามารถรับประกันในคุณภาพการให้บริการแก่ทราฟฟิกในโครงข่ายที่มีสองระดับได้

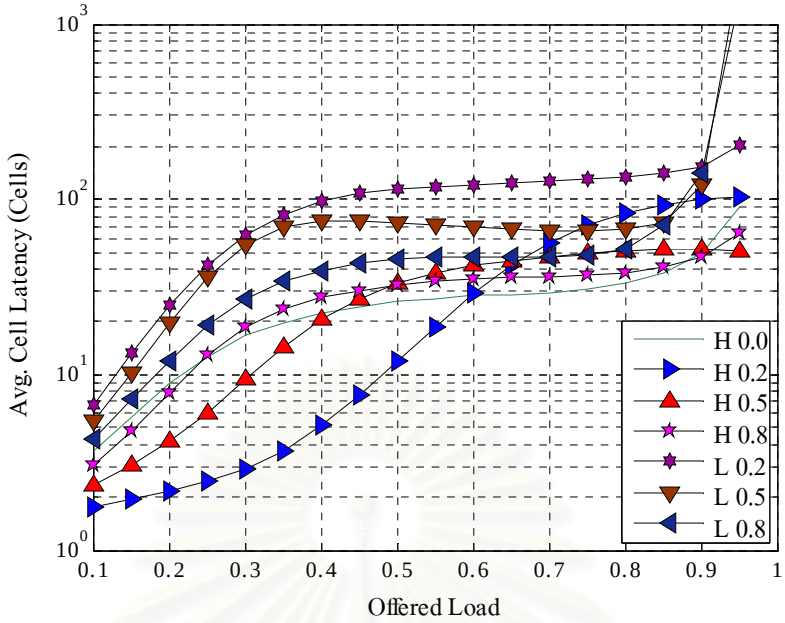
ตามรูปที่ 6.22 (ก) (ข) และ (ค) แสดงค่าเฉลี่ยการประวิงเวลาของเซลล์ และ 6.22 (ง) แสดงค่าเทรสโฮลด์ เมื่อโหลดมีทราฟฟิก H มีค่า 20 50 และ 80 เปอร์เซ็นต์ จะเห็นว่าที่ H มีสัดส่วนในโหลดน้อย เราจะต้องปรับค่าเทรสโฮลด์เมื่อทราฟฟิกเพิ่มขึ้นให้เร็วขึ้นเพื่อชดเชยไม่ให้ทราฟฟิก L มีค่าเฉลี่ยการประวิงเวลาสูงมากเกินไปจนไม่สามารถรับประกันในคุณภาพการให้บริการได้ ทำให้เส้นกราฟมีลักษณะกระโดดเป็นช่วง ๆ แต่เมื่อโหลดมีสัดส่วนของ H มีค่ามากกว่า L ทำให้ H มีการแข่งขันกันมากขึ้นที่จะได้รับเลือกค่าเฉลี่ยการประวิงเวลาจึงสูงขึ้น และ L มีการแข่งขันกันน้อยลงค่าเฉลี่ยการประวิงเวลาจึงลดลง จึงไม่จำเป็นที่จะต้องเร่งเปลี่ยนค่าเทรสโฮลด์เพื่อชดเชยกรณีที่ทราฟฟิก L มีการประวิงเวลาสูงเกินไป ทำให้เส้นกราฟของค่าเฉลี่ยการประวิงเวลาต่อเนื่องกันมากขึ้น



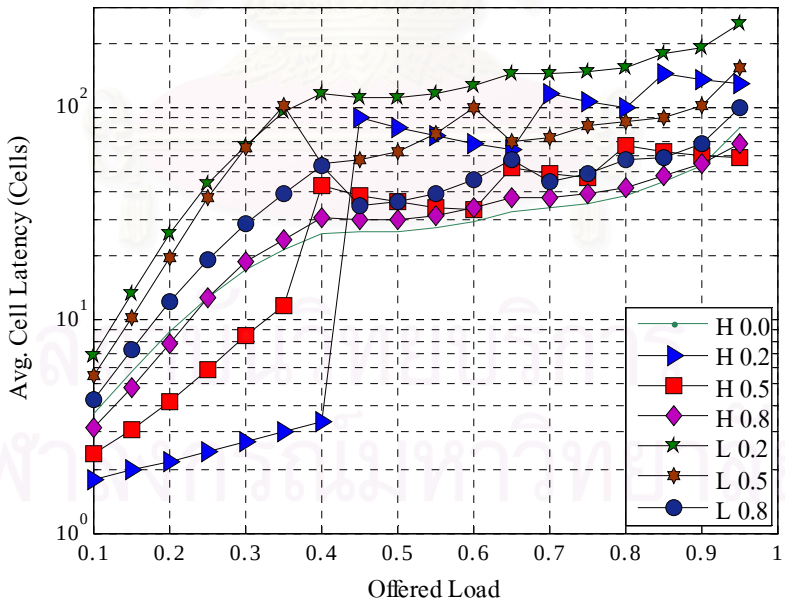
รูปที่ 6.19 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเอนเวโลปที่มีเซลล์ที่หัวคิว



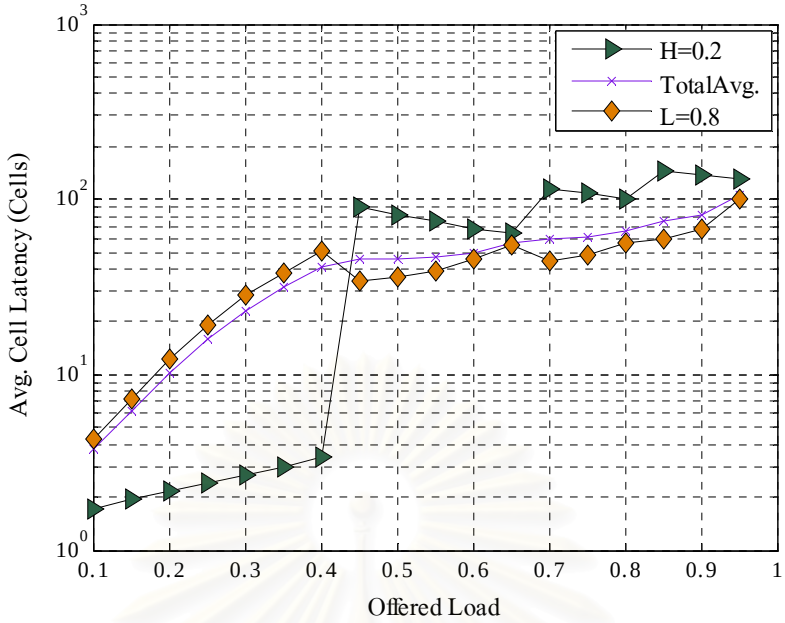
รูปที่ 6.20 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท โดยวิธีเลือกเอนเวโลปบรรจุเต็ม



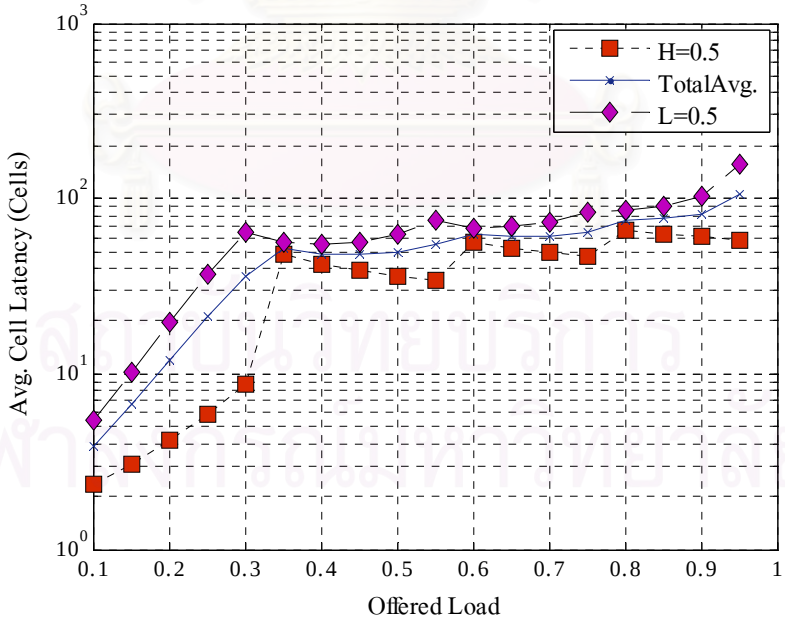
รูปที่ 6.21 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภท โดยวิธีเลือกเอนเวโลปแบบผสมผสาน



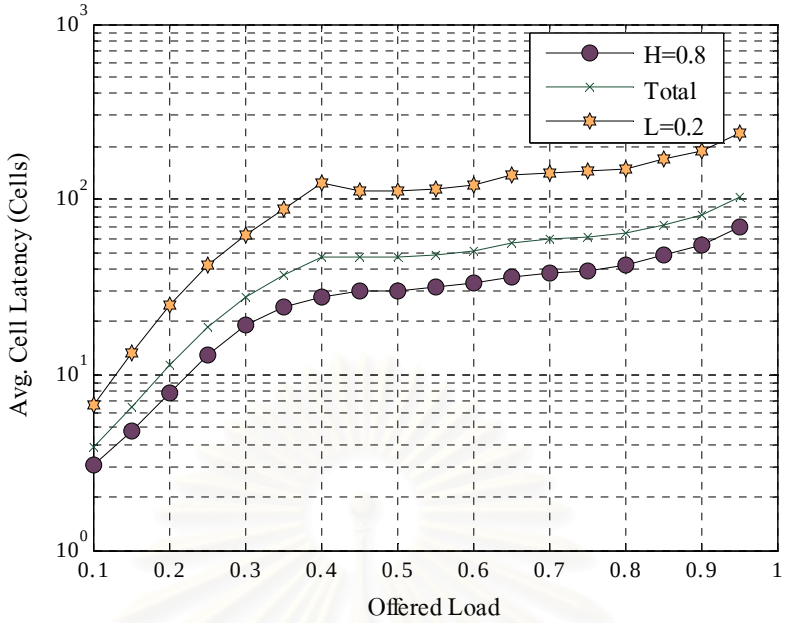
รูปที่ 6.22 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต



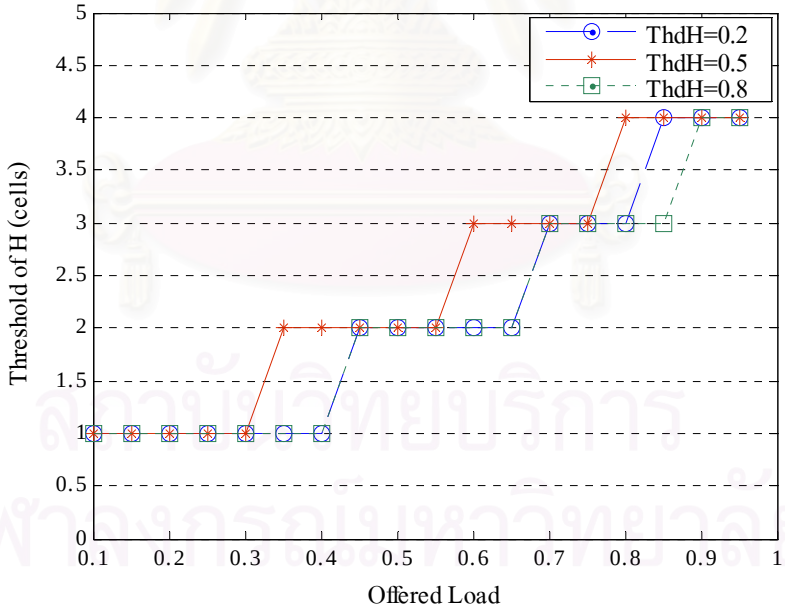
รูปที่ 6.23 (ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ เมื่อโหลดมี H = 20%



รูปที่ 6.23 (ข) เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ เมื่อโหลดมี H = 50%



รูปที่ 6.23 (ค) เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์ เมื่อโหลดมี H = 80%



รูปที่ 6.23 (ง) แสดงค่าเทรสโฮลด์ที่ H ค่าต่าง ๆ

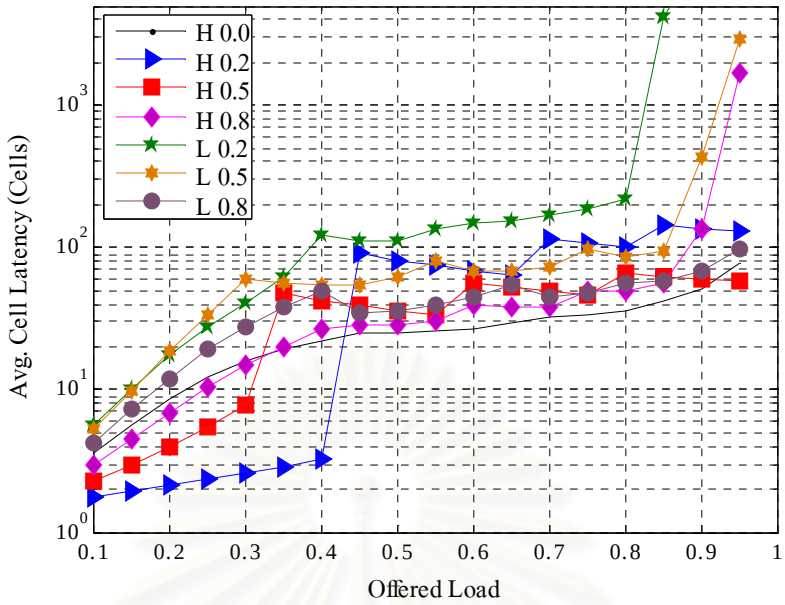
รูปที่ 6.23 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของทราฟฟิกสองประเภทโดยวิธีเลือกเทรสโฮลด์เอนเวโลปแบบพลวัต เมื่อโหลดมี H เป็นสัดส่วนต่าง ๆ กัน

6.4 กรณีศึกษา: วิเคราะห์สมรรถนะของสวิตช์กรณีทราฟฟิกสองประเภท และทราฟฟิกเป็นประเภท*นอนยูนิฟอร์ม*

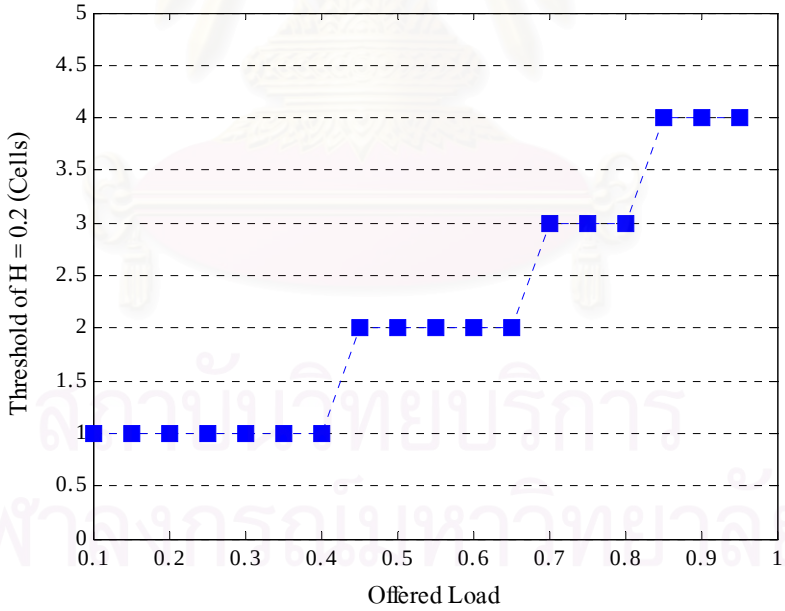
กรณีที่ 1 *กำหนดให้* $\rho = r\rho + (1-r)\rho$ *โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนต่างกัน*

ในการทดสอบกับทราฟฟิกประเภทนอนยูนิฟอร์ม จะใช้รูปแบบทราฟฟิกเป็นแบบเอาต์พุตจุดร้อน (hot-spot output) ที่นิยมใช้กันมากในการวิเคราะห์สมรรถนะของสวิตช์ในงานวิจัยด้านสถาปัตยกรรมสวิตช์ โดยให้สัดส่วนของเซลล์ที่แต่ละอินพุตจะไปยังเอาต์พุตใดเอาต์พุตหนึ่งที่ไม่ซ้ำกันเท่ากับ *r,* $(1/N \le r \le 1)$ หมายความว่าถ้า *r = 1/N* ก็คือทราฟฟิกเป็นประเภทยูนิฟอร์ม เราจะทำการวิเคราะห์ผลจากการจำลองแบบ โดยให้อัตราส่วน r = 0.2 โดยแสดงดังรูปที่ 6.24 จะเห็นว่าการเลือกเอนเวโลปส่งผ่านสวิตช์โดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัต ให้สมรรถนะทางด้านการประวิงเวลาของสวิตช์ที่ยังสามารถยอมรับได้ในช่วงทราฟฟิกต่ำและปานกลาง (ทราฟฟิกน้อยกว่า 0.8) แต่ในกรณีที่ทราฟฟิกหนาแน่นมาก ค่าเฉลี่ยการประวิงเวลาสูงมากจนไม่สามารถรับประกันในคุณภาพการให้บริการสำหรับทราฟฟิกนี้ได้ กล่าวโดยสรุป ในกรณีทราฟฟิกประเภทนอนยูนิฟอร์ม เราไม่สามารถออกแบบให้สวิตช์รองรับค่าคุณภาพการให้บริการได้ในทุกช่วงทราฟฟิก โดยเฉพาะช่วงทราฟฟิกเข้าใกล้ค่า 1.0 แต่ยังสามารถนำเทคนิคนี้ไปใช้ได้ถ้าทราฟฟิกมีค่าไม่เกิน 0.8 การควบคุมค่าทราฟฟิกไม่ให้ผ่านสวิตช์เกินค่า 0.8 คงจะยกให้เป็นหน้าที่ผู้บริหารจัดการโครงข่ายที่จะนำวิธีอื่น ๆ มาใช้ในการควบคุมทราฟฟิกในระบบ เมื่อต้องนำเทคนิคเอนเวโลปมาประยุกต์ใช้กับสวิตช์หรือเราเตอร์ในโครงข่ายอินเทอร์เน็ตแบ็กโบนในโอกาสต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์



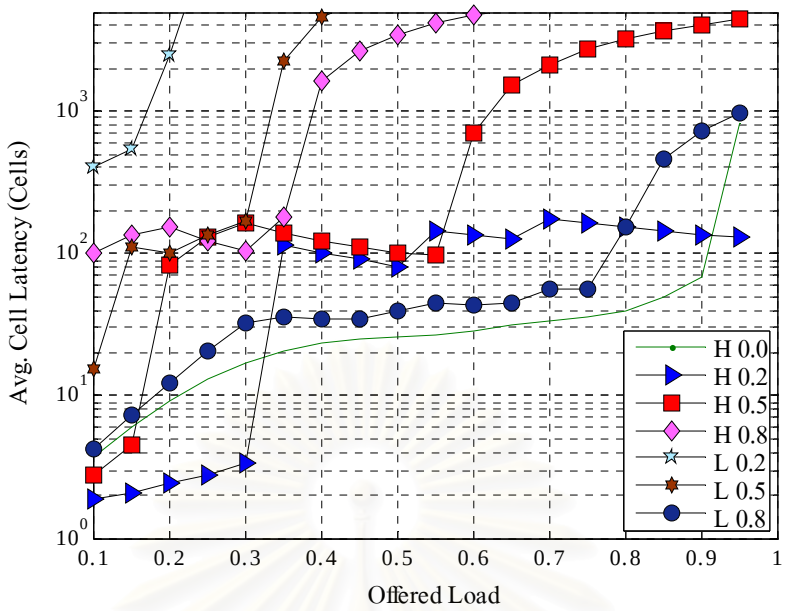
(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.24 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์กรณีทราฟฟิกสองประเภท และทราฟฟิกเป็นแบบนอนยูนิฟอร์ม กรณีที่ 1

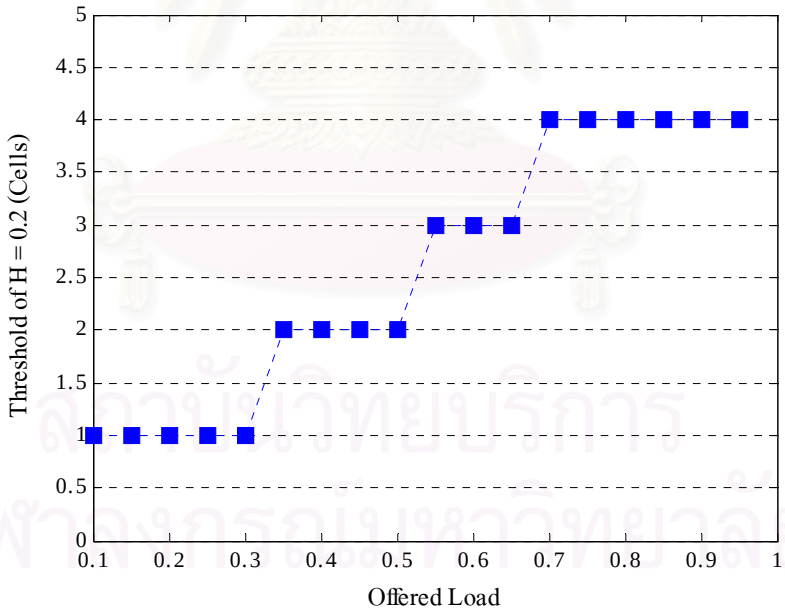
กรณีที่ 2 *กำหนดให้* $\rho = r\rho + (1-r)\rho$ *โดยที่แต่ละอินพุตมีเอาต์พุตจุดร้อนเดียวกัน*

จากการทดสอบทราฟฟิกนอนยูนิฟอร์มกรณีที่ 1 จะเห็นว่าสามารถรับประกันในคุณภาพการให้บริการในย่านทราฟฟิกไม่เกินค่า 0.8 แต่เมื่อเรากำหนดเงื่อนไขให้เอาต์พุตจุดร้อนเป็นพอร์ตเดียวกันของทุกอินพุต วิเคราะห์ผลจากกราฟ ตามรูปที่ 6.25 จะเห็นว่าสามารถรับประกันคุณภาพการให้บริการเฉพาะกรณีที่ทราฟฟิกประเภทมีลำดับความสำคัญสูง (H = 0.2) แต่ในกรณีที่ทราฟฟิกที่มีลำดับความสำคัญสูงเพิ่มสัดส่วนมากขึ้น (H = 0.5 และ H = 0.8) ไม่สามารถรับประกันในคุณภาพการให้บริการได้เลย สาเหตุเนื่องจากเอาต์พุตจุดร้อนในกรณีนี้รับภาระโหลดเกินมากกว่ากรณีที่ 1 เป็นจำนวน N เท่า สมรรถนะของสวิตช์ทางด้านค่าเฉลี่ยการประวิงเวลาจึงลดลงรวดเร็วกว่าทราฟฟิกนอนยูนิฟอร์มกรณีที่ 1 จะเห็นว่าการกระจายโหลดจึงเป็นสิ่งจำเป็นเพื่อที่จะไม่ให้เอาต์พุตแต่ละเอาต์พุตของสวิตช์รับภาระโหลดเกินมากเกินไป จากการวิเคราะห์ทั้งสองกรณีแสดงให้เห็นว่า รูปแบบของทราฟฟิกนอนยูนิฟอร์มแต่ละรูปแบบก็ให้สมรรถนะด้านการประวิงเวลาของสวิตช์ที่แตกต่างกัน การวิเคราะห์กรณีทราฟฟิกที่เป็นประเภทนอนยูนิฟอร์มจึงไม่สามารถวิเคราะห์เป็นกรณีทั่วไปได้เหมือนกับทราฟฟิกที่เป็นประเภทยูนิฟอร์ม เราจึงวิเคราะห์สมรรถนะของสวิตช์เมื่อใช้เทคนิคเอนเวโลปได้เป็นแต่ละกรณี ๆ เท่านั้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



(ก) เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์



(ข) แสดงค่าเทรสโฮลด์เอนเวโลป

รูปที่ 6.25 เปรียบเทียบสมรรถนะด้านการประวิงเวลาของสวิตช์กรณีทราฟฟิกสองประเภท และทราฟฟิกเป็นประเภทนอนยูนิฟอร์ม กรณีที่ 2

6.5 ผลการเปรียบเทียบ

ตามตารางที่ 6.1 เปรียบเทียบขนาดเอนเวโลปต่ำสุดของอัลกอริทึมจัดลำดับเซลล์ ซึ่งได้จากผลการคำนวณขนาดเอนเวโลป จากข้อมูลของความเร็วในการรับส่งข้อมูลของข่ายเชื่อมโยง ขนาดแพ็กเกตไอพีต่ำสุด และความเร็วในการประมวลผลหาค่าการแมตช์ของอัลกอริทึมจัดลำดับเซลล์ดังนี้

ตารางที่ 6.1 ขนาดเอนเวโลปที่เหมาะสมสำหรับอัลกอริทึมจัดลำดับเซลล์

Year	Line	Line rate (bps)	64-bytes cell (ns)	Envelope size (cells)
1995	T1	1.5 M	-	1
1996	OC3c	155.52 M	-	1
1997	OC12c	622.08 M	823.2	1
1999	OC48c	2.5 G	204.8	1
2000	OC192c	10.0 G	51.2	1
2002	OC768c	40.0 G	9.6	2

หมายเหตุ ใช้อัลกอริทึม iSLIP, run-time $\cong$ 10 ns (1- iterative)

6.6 ผลการวิเคราะห์ปัจจัย

การวิเคราะห์ผลด้วยการจำลองแบบด้วยคอมพิวเตอร์ ให้ผลถูกต้องในระดับของการศึกษาเพื่อทำวิทยานิพนธ์ ซึ่งเรากำหนดสถานะการณ์ขึ้นที่คาดว่าสถานะการณ์จริงในโครงข่ายทราฟฟิกจะเป็นเช่นนั้น และเพื่อให้สามารถศึกษาวิเคราะห์ในเชิงการจำลองแบบได้ สิ่งที่ต้องการก็คือความถูกต้องในระดับศึกษาวิจัย แต่โดยความเป็นจริงแล้วพฤติกรรมมนุษย์มีลักษณะเป็นอนิจจัง มีความเปลี่ยนแปลง ไม่แน่นอน ไม่สามารถหารูปแบบได้ และเนื่องจากมนุษย์เป็นผู้ให้กำเนิดทราฟฟิกในระบบ ทราฟฟิกที่สื่อสารกันอยู่ในระบบจึงเป็นเช่นนั้นด้วย เราทำได้แค่การประมาณการ (หรือคาดคะเน) ว่า ทราฟฟิกมีลักษณะโดยรวมใกล้เคียงกับที่เรากำหนดด้วยรูปแบบทางคณิตศาสตร์ แพ็กเกตโดยลักษณะแล้วมีการแจกแจงเข้าใกล้การแจกแจงแบบ pareto distribution หรือ heavy-tailed traffic จึงใช้ความเป็นอิสระต่อกันของทราฟฟิก (independent) ในการกำเนิดแพ็กเกตที่ใช้ในการทดสอบแบบจำลองโดยสมมุติว่าใกล้เคียง และสุดท้ายมีแต่การนำไปติดตั้งและทดลองกับ

ระบบจริงจึงจะสามารถแก้ปัญหาปัจจัยอื่น ๆ ที่มองไม่เห็นได้ และนำไปใช้ได้กับระบบจริงซึ่งอยู่นอกเหนือขอบเขตของงานวิทยานิพนธ์ฉบับนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7
สรุปผลการวิจัย และข้อเสนอแนะ

7.1 สรุปผลการวิจัย

7.1.1 ข้อดีของเทคนิคเอนเวโลป

เทคนิคเอนเวโลปนำมาใช้ในการแก้ปัญหาคอขวดในการสวิตช์ข้อมูล โดยการลดเวลาในการประมวลผลเพื่อหาการแมตช์ระหว่างอินพุตและเอาต์พุตคู่ใด ๆ ของสวิตช์ ทำให้ส่งเซลล์ผ่านโครงสร้างการสวิตช์ได้เร็วขึ้น หรือกล่าวอีกนัยหนึ่งก็คือ การเพิ่มแบนด์วิดท์ในการสวิตช์นั่นเอง ทำให้สวิตช์สามารถนำไปใช้กับแพ็กเกตที่มีความเร็วสูงขึ้นได้ อัลกอริทึมที่เหมาะสมที่สามารถนำมาใช้ในการจัดลำดับเซลล์ที่เข้าแข่งขันส่งผ่านโครงสร้างการสวิตช์ร่วมกับเทคนิคเอนเวโลปก็คือ อัลกอริทึม iSLIP ซึ่งเมื่อนำมาใช้กับเทคนิคเอนเวโลปจะต้องปรับปรุงอัลกอริทึมจัดลำดับเซลล์ในการเลือกเอนเวโลปสำหรับส่งผ่านโครงสร้างการสวิตช์ ซึ่งวิธีเลือกเอนเวโลปมีผลกระทบโดยตรงต่อการประวิงเวลาของเซลล์ที่เพิ่มขึ้น การเลือกเอนเวโลปโดยวิธีเทรสโฮลด์เอนเวโลปแบบพลวัตทำให้ค่าเฉลี่ยการประวิงเวลาดีกว่าการเลือกเอนเวโลปวิธีอื่น ๆ ที่นำเสนอมาก่อนหน้านั้น กรณีที่ทราฟฟิกมีหลายระดับ ได้แก้ปัญหาโดยให้ทราฟฟิกที่มีระดับความเร่งด่วนในการส่งข้อมูลสูงกว่าก็จะมีลำดับความสำคัญสูงกว่า ซึ่งจะได้รับเลือกก่อนตามลำดับความสำคัญของทราฟฟิก ทำให้เราสามารถควบคุมคุณภาพการให้บริการสำหรับทราฟฟิกหลายระดับ ให้อยู่ในขอบเขตที่กำหนดไว้ได้ ทั้งนี้ก็เพื่อที่จะให้สวิตช์สามารถรับประกันคุณภาพการให้บริการสำหรับทราฟฟิกประเภทนั้น ๆ ได้ นอกจากนี้ ยังได้ศึกษาผลกระทบต่อสมรรถนะของสวิตช์ เมื่อทราฟฟิกเป็นประเภทนอนยูนิฟอร์มเมื่อนำเทคนิคเอนเวโลปมาประยุกต์ใช้ด้วย

7.1.2 ข้อเสียของเทคนิคเอนเวโลป

จากการจัดเตรียมเอนเวโลปก่อนส่งผ่านโครงสร้างสวิตช์ ทำให้ค่าเฉลี่ยการประวิงเวลาของเซลล์มีค่าเพิ่มขึ้นตามขนาดเอนเวโลปที่เพิ่มขึ้น เราจึงไม่สามารถเพิ่มขนาดเอนเวโลปเกินขอบเขตที่กำหนดไว้สำหรับรับประกันคุณภาพการให้บริการได้ ถึงแม้ว่าจะทำให้แบนด์วิดท์ในการสวิตช์สูงขึ้นและเป็นการแก้ปัญหาคอขวดในการสวิตช์โดยตรง จะเห็นว่าทราฟฟิกประเภทเรียลไทม์มีข้อกำหนดในเรื่องการประวิงเวลาของเซลล์ที่เข้มงวดอย่างยิ่ง ดังนั้น จึงเป็นข้อสำคัญที่จะต้องนำมาพิจารณา และเป็นข้อจำกัดของเทคนิคเอนเวโลปด้วย

7.2 ข้อเสนอแนะ

ถึงแม้ว่าในปัจจุบัน จะมีการใช้ข่ายเชื่อมโยงทางแสงในโครงข่ายอินเทอร์เน็ตแบ็กโบนกันอย่างแพร่หลาย และได้มีผลงานวิจัยสวิตช์แสง (optical switch) ที่จะนำมาใช้เป็นสวิตช์ในโครงข่ายดังกล่าว แต่ยังคงไม่สามารถนำมาใช้แทนสวิตช์อิเล็กทรอนิกส์ในทางปฏิบัติได้ เนื่องจากยังไม่สามารถหาวิธีจัดการบัฟเฟอร์แสงได้ การใช้สวิตช์อิเล็กทรอนิกส์กับข่ายเชื่อมโยงทางแสง ทำให้เกิดปัญหาคอขวดในการสวิตช์ข้อมูลซึ่งเป็นข้อจำกัดของสวิตช์ประเภทนี้ บรรดานักออกแบบโครงข่าย จึงได้พยายามหาวิธีแก้ไขมาอย่างต่อเนื่องจนถึงปัจจุบัน นอกจากเทคนิคเอนเวโลปซึ่งได้นำเสนอในวิทยานิพนธ์ฉบับนี้แล้ว ในงานวิจัยทางด้านสถาปัตยกรรมสวิตช์ ได้มีการนำเสนอวิธี load balancing และ massive parallelism และในงานวิจัยทางด้านโครงข่าย ได้มีการนำเสนอโครงข่ายการสวิตช์เบิรสต์แสง (optical burst switching network) ซึ่งเราสามารถจะนำเทคนิคเอนเวโลปไปพัฒนาต่อไป เพื่อให้ทำหน้าที่ในการสะสมเซลล์เบิรสต์และจัดลำดับเซลล์ส่งผ่านโครงสร้างการสวิตช์ของเราเตอร์อินเกรส (ingress router) ก่อนส่งผ่านโครงข่ายอินเทอร์เน็ตแบ็กโบนที่มีการสวิตช์เบิรสต์ (burst switching internet backbone) ที่จะนำมาใช้ในอนาคตอันใกล้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Fabio M. Chiussi and Andrea Francini, "Scalable Electronic Packet Switches," IEEE Journal on Selected Areas in Communication, 21, 4 (May 2003): 486-500.

2. Sundar Iyer and Nick W. Mckeown, "Analysis of the Parallel Packet Switch Architecture," IEEE/ACM Transactions on Networking, 11, 2 (April 2003): 314-324.

3. William Stallings, Data and Computer Communication, 6th ed., New Jersey, Prentice Hall International, 2000: pp. 277-314.

4. Andrew S. Tanenbaum, Computer Networks, 4th ed., New Jersey, Pearson Education, 2003: pp. 146-151.

5. H. Jonathan Chao, Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers, A Wiley-Interscience Publication, John Wiley & Sons, 2001.

6. Tarek N. Saadawi, Mostafa H. Ammar and Ahmed El Hakeem, Fundamentals of Telecommunication Networks, John Wiley & Sons, 1994: pp. 399-477.

7. Mark J. Karol, Michael G. Hluchyj and Samuel P. Morgan, "Input versus Output Queueing on a Space-Devision Packet Switch," IEEE Transaction on Communication, com-35, 12 (December 1987): 1347-1356.

8. Marco A. Marsan, "Packet-Mode Scheduling in Input-Queued Cell-Based Switches," IEEE/ACM Transactions on Networking, 10, 5 (October 2002): 666-678.

9. Hakyong Kim and Kiseon Kim, "Performance Analysis of the Multiple Input-Queued Packet Switch with the Restricted Rule," IEEE/ACM Transactions on Networking, 11, 3 (June 2003): 478-487.

10. Hiroyuki Ohsaki and Naoki Wakamiya, "Performance of an Input/Output Buffer-Type ATM LAN Switch with Back-Pressure Function," IEEE/ACM Transactions on Networking , 5, 2 (April 1997): 278-290.

11. Nick Mckeown, "Scheduling Algorithm for Input-Queued Switches," Ph.D. dissertation, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, 1995.

12. Nick Mckeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches," IEEE/ACM Transactions on Networking, 7, 2 (April 1999): 188-201.

13. Nick Mckeown, Venkat Anantharam and Jean Walrand, "Achieving 100% Throughput in an Input-Queued Switch," Proceeding of IEEE INFOCOM'96, 1 (1996): 296-302.

14. Jonathan Chao, "Saturn: A Terabit Packet Switch Using Dual Round-Robin," IEEE Communications Magazine (December 2000): 78-84.

15. Nick Mckeown, Martin Izzard, Adisak Mekkittikul, William Ellersick and Mark Horowitz, " The Tiny Tera: A Packet Switch Core," Proceedings of Hot Interconnects IV, Stanford (August 1996): 161-173.

16. Available from: http :// **www.cisco.com** (white paper).

17. Athanasios Papoulis, Probability, Random variables and Stochastic Processes, 4th ed., New York, McGraw-Hill, 2002: pp. 105-113.

18. Koushik Kar, Dimitrios Stiliadis, T. V. Lakshman and Leandros Tassiulas, "Scheduling Algorithms for Optical Packet Fabrics," IEEE Journal on Selected Areas in Communications, 21, 7 (September 2003): 1143-1155.

19. Adisak Makkittikul, "Scheduling Non-uniform Traffic in High Speed Packet Switches and Routers," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, CA, 1998.

20. Pablo M. Fernandez, "Circuit Switching in The Internet," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, CA, 2003.

บรรณานุกรม

1. Ronald W. Wolff, "Stochastic Modeling and The Theory of Queues," New Jersey, Prentice-Hall, 1989.

2. Perter G. Harrison and Naresh M. Patel, "Performance Modelling of Communication Networks and Computer Architectures," Great Britain, Addison-Wesley Publishing Company, 1993.

3. H. Jonathan Chao and Xiaolei Guo, "Quality of Service Control in High-Speed Networks," A Wiley-Interscience Publication, John Wiley & Sons, 2001.

4. Salah Aidarous and Thomas Plevyak, "Managing IP Network: Challenges and Opportunities," A Wiley-Interscience Publication, John Wiley & Sons, 2003.

5. Isaac Keslassy, "The Load-Balanced Router," Ph.D. dissertation, Department of Electrical Engineering, Stanford Univ., CA, 2004.

6. H. Jonathan Chao, "Next Generation Routers," Proceedings of The IEEE, 90, 9 (September 2002): 1518-1558.

7. Ge Nong and Mounir Hamdi, "On the Provision of Quality-of-Service Guarantees for Input Queued Switches," IEEE Communication Magazine (December 2000): 62-69.

8. Cyriel Minkenberg and Ton Engbersen, "A Combined Input and Output Queued Packet-Switched System Based on PRIZMA Switch-on-a-Chip Technology," IEEE Communication Magazine (December 2000): 70-76.

9. Kohei Shiomoto et al., "Scalable Multi-QoS IP+ATM Switch Router Architecture," IEEE Communication Magazine (December 2000): 86-92.

10. Xiao Zhang and L.N. Bhuyan, "Deficit Round-Robin Scheduling for Input-Queued Switches," IEEE Journal on Selected Areas in Communications, 21, 4 (May 2003): 584-594.

11. Matthew Andrews and Milan Vojnovic, “Scheduling Reserved Traffic in Input-Queued Switches: New Delay bounds via Probabilistic Techniques,” IEEE Journal on Selected Areas in Communications, 21, 4 (May 2003): 595-605.

12. Robert B. Magill, Charles E. Rohrs and Robert L. Stevenson, “Output-Queued Switch Emulation by Fabrics with Limited Memory,” IEEE Journal on Selected Areas in Communications, 21, 4 (May 2003): 606-614.

13. Anthony C. Kam and Kai-Yeung Siu, “Linear-Complexity Algorithms for QoS Support in Input-Queued Switches with No Speedup,” IEEE Journal on Selected Areas in Communications, 17, 6 (June 1999): 1040-1056.

14. Pattabhiraman Krishna et. al., “On the Speedup Required for Work-Conserving Crossbar Switches,” IEEE Journal on Selected Areas in Communications, 17, 6 (June 1999): 1057-1066.

15. Donpaul C. Stephens, Jon C. R. Bennett and Hui Zhang, “Implementing Scheduling Algorithms in High-Speed Networks,” IEEE Journal on Selected Areas in Communications, 17 (June 1999): 1145-1158.

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

```
%Input-queued switch with iSLIP algorithm using MATLAB program.

clear
SIM_TIME = 10000;
NUM_PORT = 16;
BUFFER_SIZE = 16;
ITERATIVE_MAX = 4;
offerLoad = [0:0.1:1];
input = zeros(1,NUM_PORT);
x = zeros(1,11);
y = zeros(1,11);
z = zeros(1,11);
%%%main program*********
for n = 1:11
   lengthQ = zeros(NUM_PORT);
   genPack = 0; txPack = 0; lossPack = 0; totalDelay = 0;
   grt = zeros(1,NUM_PORT); act = zeros(1,NUM_PORT);
   for m = 1:SIM_TIME
      %generated cell at inputport***
      input = rand(1,NUM_PORT) < offerLoad(n);
      genPack = genPack + sum(input);
      for i = 1:NUM_PORT
         if input(i) > 0
            j = randint(1,1,[1,NUM_PORT]);
            lengthQ(i,j) = lengthQ(i,j) + 1;
            if lengthQ(i,j) > BUFFER_SIZE
               lengthQ(i,j) = BUFFER_SIZE;
               lossPack = lossPack + 1;
            end
         end
      end
      %Step 1: Request **********
      request = sign(lengthQ);
      %Step 2: Grant ************
      for iterative =1:ITERATIVE_MAX
         grant = zeros(NUM_PORT);
         g = zeros(1,NUM_PORT);
         g = grt;
         for j = 1:NUM_PORT
            i = 1;
            while (i <= NUM_PORT)
               g(j) = rem(g(j),NUM_PORT) + 1;
               if request(g(j),j) > 0
                  grant(g(j),j) = 1;
                  i = NUM_PORT + 1;
               else
                  grant(g(j),j) = 0;
                  i = i + 1;
               end
            end
         end
```

```
            %step 3: Accept ************
            accept = zeros(NUM_PORT);
            for i = 1:NUM_PORT
               j = 1;
               while (j <= NUM_PORT)
                  act(i) = rem(act(i),NUM_PORT) + 1;
                  if grant(i,act(i)) > 0
                     accept(i,act(i)) = 1;
                     request(i,:) = 0;
                     request(:,act(i)) = 0;
                     if iterative < 2
                        grt(act(i)) = g(act(i));
                     end
                     j = NUM_PORT + 1;
                  else
                     accept(i,act(i)) = 0;
                     j = j + 1;
                  end
               end
            end
            txPack = txPack + sum(sum(accept));
            lengthQ = lengthQ - accept;
         end
         totalDelay = totalDelay + sum(sum(lengthQ));
      end
      % display output data ******
      x(n) = genPack/(SIM_TIME*NUM_PORT);
      y(n) = txPack/(SIM_TIME*NUM_PORT);
      if txPack ~= 0
         z(n) = totalDelay/txPack;
      else
         z(n) = 0;
      end
   end
   disp('  GenPacket:TranPacket:AvgLatency(cells)');
   disp([x'y'z']);
   figure(1)
   plot(x,y,'b:+');
   axis([0 1 0 1]);
   xlabel('offerload,p:');
   ylabel('Switch throughtput');
   title('Input-Queued switch');
   legend('iSLIP');
   figure(2)
   semilogy(x,z,'m-*');
   axis([0.2 1 0.01 1000]);
   xlabel('offerload,p:');
   ylabel('AvgLatency(cells)');
   legend('iSLIP');
   data = fopen('data.txt','w');
   fprintf(data,'%.4f       %.4f        %.4f\n',x',y',z');
   fclose(data);
   %%%end main program.
```

ประวัติผู้เขียนวิทยานิพนธ์

เรือเอก อำนาจรัตน์ โสภณธรรมพัฒน์ เกิดเมื่อวันที่ 24 พฤศจิกายน พ.ศ. 2513 ที่อำเภอขาณุวรลักษบุรี จังหวัดกำแพงเพชร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า (ไฟฟ้ากำลัง และไฟฟ้าสื่อสาร) มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2542 ปริญญาตรีนิติศาสตรบัณฑิต มหาวิทยาลัยรามคำแหง ในปีการศึกษา 2545 และศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า (ไฟฟ้าสื่อสาร) จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545

ประสบการณ์การทำงาน - กองร้อยต่อสู้อากาศยานที่ 2 กองพันต่อสู้อากาศยานที่ 23 กรมต่อสู้อากาศยานที่ 2 หน่วยบัญชาการต่อสู้อากาศยานและรักษาฝั่ง กองทัพเรือ

- ประจำหมวดวางแผนงานและวิเคราะห์ แผนกมาตรฐานและวิเคราะห์ กองควบคุมคุณภาพ อู่ทหารเรือพระจุลจอมเกล้า กรมอู่ทหารเรือ กองทัพเรือ

- นายช่าง แผนกออกแบบต่อเรือ กองออกแบบ อู่ทหารเรือพระจุลจอมเกล้า กรมอู่ทหารเรือ

ตำแหน่งหน้าที่ปัจจุบัน หัวหน้าหมวดไฟฟ้าและอิเล็กทรอนิกส์ แผนกจัดหา ศูนย์พัสดุช่าง กรมอู่ทหารเรือ ถ.อรุณอมรินทร์ แขวงศิริราช เขตบางกอกน้อย กรุงเทพ ฯ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย