



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)
ปริญญา

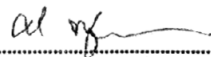
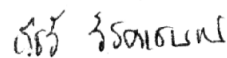
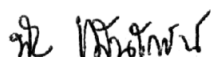
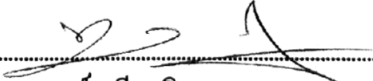
วิศวกรรมไฟฟ้า สาขา วิศวกรรมไฟฟ้า
ภาควิชา

เรื่อง การส่งข้อมูลต่างลักษณะบนระบบที่มีพลังงานจำกัด

Supporting Heterogeneous Traffic in Energy Limited Systems

นามผู้วิจัย นางสาวนฤมล ปฐมมาธิศ

ได้พิจารณาเห็นชอบโดย

ประธานกรรมการ ()
กรรมการ ()
กรรมการ ()
หัวหน้าภาควิชา ()

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

()
รองศาสตราจารย์วินัย อาจคงหาญ, M.A.
คณบดีบัณฑิตวิทยาลัย

วันที่ 28 เดือน พฤษภาคม พ.ศ. 2549

วิทยานิพนธ์

เรื่อง

การส่งข้อมูลต่างลักษณะบนระบบที่มีพลังงานจำกัด

Supporting Heterogeneous Traffic in Energy Limited Systems

โดย

นางสาวนฤมล ปฐมมาณิส

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2549

ISBN 974-16-1122-6

นฤมล ปฐมมาณิส 2549: การส่งข้อมูลต่างลักษณะบนระบบที่มีพลังงานจำกัด
ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) สาขาวิศวกรรมไฟฟ้า
ภาควิชาวิศวกรรมไฟฟ้า ภาชานกรรมการที่ปริภษา: ผู้ช่วยศาสตราจารย์ศิริวัฒน์ พูนวสิน,
M.S.EE. 159 หน้า
ISBN 974-16-1122-6

เราพิจารณาโหนดๆ หนึ่งที่มีการส่งหน่วยข้อมูลซึ่งตระหนักถึงการบริโภคพลังงาน โดยโหนด
ที่กล่าวถึงนี้จะสามารถรองรับข้อมูลสองชนิดที่มีความแตกต่างกัน ชนิดแรกคือข้อมูลที่ต้องการการ
รับประกันคุณภาพด้านเวลา (หน่วยข้อมูลที่ถูกจำกัดเวลาส่งด้วยเส้นตาย) และแบบสองคือข้อมูลที่ไม่
ต้องการรับประกันคุณภาพด้านเวลา (หน่วยข้อมูลที่ไม่มีการกำหนดเส้นตาย) ซึ่งข้อมูลทั้งสองต่างมีความ
ต้องการที่คล้ายกัน นั่นคือชนิดแรกต้องการส่งให้ทันเส้นตาย ส่วนชนิดที่สองต้องการให้มีค่า
ความหน่วงเวลาน้อยที่สุด แสดงว่าเมื่อส่งข้อมูลทั้งสองผ่านช่องสัญญาณเดียวกัน มันก็ต่างต้องพยายาม
แย่งใช้แบนด์วิธที่มีอยู่เพื่อตอบสนองความต้องการของแต่ละฝ่าย ดังนั้นงานวิจัยนี้จึงนำเสนอวิธีการ
จัดลำดับประกอบกับการจัดสรรการใช้พลังงานไว้หลายวิธี วิธีการเหล่านี้แบ่งการทำงานออกเป็นสอง
ส่วน ส่วนแรกเป็นการจัดลำดับโดยเริ่มจากจัดลำดับภายในชนิดของข้อมูลประเภทเดียวกัน ซึ่งการ
จัดลำดับภายในแต่ละชนิดไม่จำเป็นต้องใช้วิธีการจัดลำดับเดียวกัน ต่อมานำข้อมูลที่ผ่านมาการจัดลำดับ
ภายในแล้วทั้งสองมาจัดลำดับรวมกันอีกครั้งด้วยวิธีการจัดลำดับ ซึ่งก็ไม่จำเป็นต้องใช้วิธีการ
จัดลำดับเดียวกับการจัดลำดับภายใน ส่วนที่สองเป็นนำหน่วยข้อมูลที่เรียงลำดับสมบูรณ์แล้วมา
กำหนดหาอัตราบิตที่ใช้ในการส่งให้กับแต่ละหน่วยข้อมูล อัตราบิตเหล่านี้สามารถแปรค่าได้ตาม
กระบวนการ DMS ซึ่งจะปรับเปลี่ยนอัตราบิตต่อสัญลักษณ์ เพื่อการประหยัดพลังงานตามอัตราเร็วใน
การส่งที่กำหนด งานวิจัยนี้ได้ทำการจำลองวิธีการต่างๆ ดังกล่าว แล้วนำมาเปรียบเทียบหาข้อดีข้อด้อย
ผลการทดลองแสดงให้เห็นว่าไม่เพียงแต่วิธีการสามารถจัดสรรการใช้ช่องสัญญาณให้กับข้อมูลทั้งสอง
ชนิดได้อย่างยุติธรรม อีกทั้งยังสามารถลดการบริโภคพลังงานลงได้ โดยเฉพาะอย่างยิ่งกรณีที่มีปริมาณ
โหนดบางจะสามารถประหยัดพลังงานลงได้กว่าร้อยละ 90 เมื่อเทียบกับการส่งข้อมูลแบบไม่
ตระหนักถึงการใช้พลังงาน

นฤมล ปฐมมาณิส

ลายมือชื่อนิสิต



ลายมือชื่อประธานกรรมการ

16, มย 49

Naruemon Pathommanit 2006: Supporting Heterogeneous Traffic in Energy Limited Systems. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Department of Electrical Engineering. Thesis Advisor: Assistant Professor Sirivat Poonvasin, M.S.EE. 159 pages.
ISBN 974-16-1122-6

We address a resource allocation problem within an energy-aware transmission node. In particular, we consider an energy-aware packet transmission node that supports packets of two types: real-time packets and non real-time packets. In addition, an energy-aware heterogeneous packet scheduling scheme is proposed. It is composed of two scheduling stages: the first stage, based on FCFS and EDF, manages each packet type independently and on the *per type* basis, while the second merges the two packet types using PGPS and a radio power management technique called Dynamic Modulation Scaling (DMS). The method has been evaluated using simulation and compared against one in which energy savings schemes are not incorporated. The result shows that not only does it provide a fair share of bandwidth (low rejected packets for real-time packets and low delay for non real-time packets) between these streams, but it also saves the overall transmission energy, particularly in the light load case, where as much as 90% savings can be obtained.

Naruemon Pathommanit

Student's signature

S. Poonvasin 16/02/06

Thesis Advisor's signature

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ อาจารย์ศิริวัฒน์ พูนวสิน ประธานกรรมการที่ปรึกษา ที่ได้ให้ความช่วยเหลือต่างๆ จนสำเร็จเป็นงานวิจัยชิ้นนี้ ไม่ว่าจะเป็นการให้ความรู้ในเชิงวิชาการ การฝึกให้คิด และวิเคราะห์สิ่งที่ได้จากผลการทดลอง ช่วยตรวจและแก้ไขข้อบกพร่องของงาน ฝึกให้ค้นคว้าและทำความเข้าใจกับงานวิจัยที่เกี่ยวข้อง และอื่นๆ อีกมากมาย ขอกราบขอบพระคุณอาจารย์วัชร วิระเชนทร์ กรรมการที่ปรึกษาวิชาเอก และ ดร.พีระยศ แสนโกชณ์ กรรมการที่ปรึกษาวิชาการ ที่กรุณาให้คำปรึกษา และคำแนะนำในการทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณ คุณนริศ กุณาศล นิสิตปริญญาโท มหาวิทยาลัยเกษตรศาสตร์ คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ ที่ได้ให้แนวคิดและความช่วยเหลือในด้านการทำการจำลองการทำงานระบบ (simulation)

ขอขอบคุณ คุณพ่อ คุณแม่ พี่ และน้อง ที่คอยเป็นกำลังใจ และสนับสนุนการทำวิทยานิพนธ์จนสำเร็จลุล่วงได้

นฤมล ปฐมมาณิศ
ธันวาคม 2548

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(4)
สารบัญภาพ	(6)
คำอธิบายสัญลักษณ์ที่ใช้ใน pseudocode และอักษรย่อ	(15)
คำนำ	1
การตรวจเอกสาร	5
ตัวแปรสุ่มแบบต่อเนื่อง (Continuous Random Variable)	5
การจัดลำดับ (Scheduling)	8
วิธี First Come First Serve	8
วิธี Earliest Deadline First	8
วิธี Packet-by-packet Generalized Processor Sharing	10
การมอดูเลต (Modulation)	15
การมอดูเลตเชิงเลขแบบเอ็มอาร์เคียวเควดเรเจอร์ (M-ary QAM)	17
วิธี Dynamic Modulation Scaling	23
วิธี Speed Scheduling of Real-time Packets	27
งานวิจัยที่เกี่ยวข้อง	28
อุปกรณ์และวิธีการ	30
อุปกรณ์	30
วิธีการ	30
ขั้นตอนที่ 1 (กระบวนการจัดลำดับ)	31
วิธี FCFS	32
วิธี FEF	33
วิธี FEP	33
ขั้นตอนที่ 2 (กระบวนการประหยัดพลังงาน)	34
วิธี HSNQ	37
วิธี LI	38
วิธี QB	40
วิธี RRP	42

สารบัญ (ต่อ)

	หน้า
วิธี LS	46
วิธี RRP w/ RPS	47
วิธี LS w/ RPS	50
ข้อมูลที่ใช้ในการทดสอบ	51
ผลและการวิจารณ์	54
ขั้นตอนที่ 1 (กระบวนการจัดลำดับ)	54
การวิเคราะห์ผลในเชิงการแปรค่าปริมาณโหลด (α / μ_{max})	54
การวิเคราะห์ผลในเชิงการแปรค่าสัดส่วนระหว่าง	
หน่วยข้อมูลนอนทิวโอเอสต่อทิวโอเอส (n_{NQoS} / n_{QoS})	57
การเปรียบเทียบกับงานวิจัยอื่น	60
ข้อสังเกตเพื่อการพัฒนาส่วนของการประหยัดพลังงาน	61
ขั้นตอนที่ 2 (กระบวนการประหยัดพลังงาน)	62
การวิเคราะห์ผลในเชิงการแปรค่าปริมาณโหลด (α / μ_{max})	63
การวิเคราะห์ผลในเชิงการแปรค่าความเร่งรีบของ	
หน่วยข้อมูลทิวโอเอส (t_{min} / T)	77
การวิเคราะห์ผลในเชิงการแปรค่าสัดส่วนระหว่าง	
หน่วยข้อมูลนอนทิวโอเอสต่อทิวโอเอส (n_{NQoS} / n_{QoS})	80
การวิเคราะห์ผลด้วยตัววัดประสิทธิภาพ	84
สรุป	87
เอกสารและสิ่งอ้างอิง	91
ภาคผนวก	94
ตัวอย่างวิธีการจัดลำดับ	95
ตัวอย่างการจัดลำดับหน่วยข้อมูลด้วยวิธี FEF	95
ตัวอย่างการจัดลำดับหน่วยข้อมูลด้วยวิธี FEP	100
ตัวอย่างวิธีการประหยัดพลังงาน	104
ตัวอย่างการประหยัดพลังงานด้วยวิธี HSNQ	104
ตัวอย่างการประหยัดพลังงานด้วยวิธี LI	110
ตัวอย่างการประหยัดพลังงานด้วยวิธี QB	117

สารบัญ (ต่อ)

	หน้า
ตัวอย่างการประหยัดพลังงานด้วยวิธี RRP	125
ตัวอย่างการประหยัดพลังงานด้วยวิธี LS	135
ตัวอย่างการประหยัดพลังงานด้วยวิธี RRP w/ RPS	149
ตัวอย่างการประหยัดพลังงานด้วยวิธี LS w/ RPS	153
ตัวอย่างการเลือกทั้งหน่วยข้อมูลคิวโอเอสที่จำเป็นต่องานมากกว่า หนึ่งหน่วยของวิธี RRP w/ RPS และวิธี LS w/ RPS	157

สารบัญตาราง

ตารางที่	หน้า
1 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง เพื่อแสดงกระบวนการจัดลำดับด้วยวิธี EDF	9
2 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง เพื่อแสดงกระบวนการจัดลำดับด้วยวิธี PGPS	12
3 รายละเอียดในแต่ละขั้นตอนของการส่งและรับข้อมูลจากต้นทางไปยังปลายทาง	16
4 วิธีการจัดลำดับภายในแต่ละบัฟเฟอร์ของวิธี FCFS FEF และ FEP	32
5 จำนวนบิตต่อสัญลักษณ์ที่ใช้ในการมอดูเลตสัญญาณสำหรับแต่ละหน่วยข้อมูล ซึ่งคำนวณด้วยวิธี HSNQ	37
6 จำนวนบิตต่อสัญลักษณ์ที่ใช้ในการมอดูเลตสัญญาณสำหรับแต่ละหน่วยข้อมูล ซึ่งคำนวณด้วยวิธี QB	41

สารบัญตาราง (ต่อ)

ตารางผนวกที่	หน้า
1 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี FEF และ FEP	95
2 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี HSNQ และ LI	104
3 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี QB, RRP และ LS	117
4 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี RRP w/ RPS และ LS w/ RPS	149
5 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี RRP w/ RPS และ LS w/ RPS ในกรณีที่ต้องเลือกทั้งหลายหน่วยข้อมูล	157

สารบัญภาพ

ภาพที่		หน้า
1	ลำดับการส่งออกหน่วยข้อมูลตัวอย่างที่ผ่านการจัดลำดับด้วยวิธี EDF	9
2	ลำดับการส่งออกหน่วยข้อมูลตัวอย่างที่ผ่านการจัดลำดับด้วยวิธี FCFS	10
3	แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ให้กับหน่วยข้อมูลของช่องสัญญาณขาเข้าทั้ง N ช่อง โดยหน่วยข้อมูลจากช่องสัญญาณขาเข้าที่ i จะได้รับแบนด์วิธเท่ากับ g_i	11
4	แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ด้วยวิธี GPS แบบอย่างละครั้งให้กับหน่วยข้อมูลจากช่องสัญญาณขาเข้าทั้งสอง	13
5	กระบวนการจำลองการส่งชุดข้อมูลตัวอย่างด้วยวิธี GPS แบบอย่างละครั้ง	13
6	การจัดลำดับชุดข้อมูลตัวอย่างด้วยวิธี PGPS	13
7	แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ด้วยวิธี GPS แบบหนึ่งต่อสองให้กับหน่วยข้อมูลจากช่องสัญญาณขาเข้าทั้งสอง	14
8	กระบวนการจำลองการส่งชุดข้อมูลตัวอย่างด้วยวิธี GPS แบบหนึ่งต่อสอง	14
9	การจัดลำดับชุดข้อมูลตัวอย่างด้วยวิธี PGPS	14
10	กระบวนการพื้นฐานในการส่งและรับข้อมูล	15
11	แผนภาพตำแหน่ง (constellation) เมื่อทำการมอดูเลตทางแอมพลิจูดของพัลส์แบบเอ็มอาร์เคย์	18
12	ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b)	24
13	ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ที่รัศมีการส่งสั้น	25
14	ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ที่รัศมีการส่งไกล	26
15	โครงสร้างกระบวนการทำงานในงานวิจัย	30
16	โครงสร้างกระบวนการจัดลำดับในงานวิจัย	31
17	กระบวนการทำงานของวิธี HSNQ	37

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
18	การคำนวณหาอัตราบิดใหม่เพื่อให้ได้อัตราบิดที่ต่ำที่สุดสำหรับ หน่วยข้อมูลคิวโอเอส ที่ทำให้สามารถส่งทุกหน่วยข้อมูล ในบัฟเฟอร์ร่วมได้ทัน สำหรับวิธี LI	38
19	กระบวนการทำงานของวิธี LI	40
20	กระบวนการทำงานของวิธี QB	41
21	การคำนวณหาอัตราบิดใหม่เพื่อให้ได้อัตราบิดที่ต่ำที่สุดที่สามารถส่ง ทุกหน่วยข้อมูลในบัฟเฟอร์ร่วมได้ทัน สำหรับวิธี RRP และวิธี LS	43
22	กระบวนการทำงานของวิธี RRP	45
23	กระบวนการทำงานของวิธี LS	46
24	การคำนวณหาอัตราบิดใหม่เพื่อให้ได้อัตราบิดที่ต่ำที่สุดที่สามารถส่ง ทุกหน่วยข้อมูลในบัฟเฟอร์ร่วมได้ทัน สำหรับวิธี RRP w/ RPS และวิธี LS w/ RPS	47
25	กระบวนการทำงานของวิธี RRP w/ RPS	49
26	กระบวนการทำงานของวิธี LS w/ RPS	50
27	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลออนคิวโอเอส โดยแสดงผลที่ปริมาณ โหลดต่างๆ (α / μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{QoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร	55
28	เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเทียบกับ หน่วยข้อมูลคิวโอเอสทั้งหมด โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับรูป 27	55
29	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลออนคิวโอเอสโดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลออนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{QoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.2$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร	57
30	เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเทียบกับ หน่วยข้อมูลคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่าง หน่วยข้อมูลออนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{QoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 29	58

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
31	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนทิวโอเอส โดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนทิวโอเอสต่อหน่วยข้อมูลทิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha/\mu_{max} = 0.6$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 58
32	เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลทิวโอเอสเมื่อเทียบกับ หน่วยข้อมูลทิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่าง หน่วยข้อมูลนอนทิวโอเอสต่อหน่วยข้อมูลทิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 31 59
33	เปอร์เซ็นต์การเกิดช่องสัญญาณว่าง โดยแสดงผลที่ปริมาณโหลดต่างๆ (α/μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{NQoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 61
34	เปอร์เซ็นต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α/μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{NQoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 64
35	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลทิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α/μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 64
36	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนทิวโอเอส โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α/μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 65
37	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนทิวโอเอส โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α/μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 66
38	เปอร์เซ็นต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α/μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 66
39	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลทิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α/μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 67

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
40	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	68
41	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	69
42	เปอร์เซ็นต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ ผนึกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้ พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	69
43	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	70
44	เปอร์เซ็นต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	71
45	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	71
46	เปอร์เซ็นต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ ผนึกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้ พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	73
47	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	74
48	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผล ที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34	74

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
49	เปอร์เซ็นต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 75
50	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 76
51	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลบนคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 76
52	เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสโดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.2$, $n_{QoS} / n_{QoS} = 1$ และ จ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 78
53	เปอร์เซ็นต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52 78
54	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลบนคิวโอเอส โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52 79
55	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52 79

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
56	เปอร์เซ็นต์การใช้พลังงานเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึก วิธีประหยัดพลังงาน (ตอนที่ ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้ พลังงานที่ 100%) โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูล นอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.2$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 80
57	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 56 81
58	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 56 81
59	เปอร์เซ็นต์การใช้พลังงานเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึก วิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้ พลังงานที่ 100%) โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูล นอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.6$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร 82
60	เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 59 82
61	ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผล ที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 59 83
62	ตัววัดประสิทธิภาพ (η) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34 85

สารบัญภาพ (ต่อ)

ภาพผนวกที่		หน้า
1	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEF ณ วินาทีที่ 9	97
2	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEF ณ วินาทีที่ 10	98
3	แผนภาพการส่งออกหน่วยข้อมูลตัวอย่างที่ช่องสัญญาณขาออก เมื่อใช้วิธีจัดลำดับ FEF	99
4	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 6	100
5	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 6.5	101
6	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 7	101
7	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 8	102
8	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 9	102
9	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 10	102
10	กระบวนการจัดลำดับทั้งสามบัพเฟอรัด้วยวิธี FEP ณ วินาทีที่ 11	103
11	แผนภาพการส่งออกหน่วยข้อมูลตัวอย่างที่ช่องสัญญาณขาออก เมื่อใช้วิธีจัดลำดับ FEP	103
12	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 0	105
13	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 1	105
14	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 2	106
15	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 3	106
16	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 4	106
17	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 5	107
18	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 6	107
19	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 7	108
20	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 8	108
21	การคำนวณหาอัตราบิดด้วยวิธี HSNQ ณ วินาทีที่ 9	108
22	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 0	110
23	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 1	110
24	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 2	111
25	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 3	111
26	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 4	112

สารบัญภาพ (ต่อ)

ภาพผนวกที่		หน้า
27	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 5	112
28	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 6	112
29	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 7	113
30	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 8	115
31	การคำนวณหาอัตราบิดด้วยวิธี LI ณ วินาทีที่ 9	115
32	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 0	118
33	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 1	119
34	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 2	119
35	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 3	120
36	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 4	121
37	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 5	121
38	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 6	122
39	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 7	123
40	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 8	123
41	การคำนวณหาอัตราบิดด้วยวิธี QB ณ วินาทีที่ 9	124
42	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 0	125
43	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 1	126
44	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 2	127
45	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 3	128
46	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 4	128
47	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 5	129
48	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 6	130
49	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 7	132
50	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 8	133
51	การคำนวณหาอัตราบิดด้วยวิธี RRP ณ วินาทีที่ 9	133
52	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 0	135
53	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 1	137
54	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 2	138

สารบัญภาพ (ต่อ)

ภาพผนวกที่		หน้า
55	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 3	138
56	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 4	140
57	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 5	140
58	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 6	143
59	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 7	145
60	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 8	147
61	การคำนวณหาอัตราบิดด้วยวิธี LS ณ วินาทีที่ 9	148
62	การคำนวณหาอัตราบิดด้วยวิธี RRP w/ RPS ณ วินาทีที่ 0	150
63	การคำนวณหาอัตราบิดด้วยวิธี RRP w/ RPS ณ วินาทีที่ 1	150
64	การคำนวณหาอัตราบิดด้วยวิธี RRP w/ RPS ณ วินาทีที่ 2	152
65	การคำนวณหาอัตราบิดด้วยวิธี LS w/ RPS ณ วินาทีที่ 0	153
66	การคำนวณหาอัตราบิดด้วยวิธี LS w/ RPS ณ วินาทีที่ 1	155
67	การคำนวณหาอัตราบิดด้วยวิธี LS w/ RPS ณ วินาทีที่ 2	156

คำอธิบายสัญลักษณ์ที่ใช้ใน pseudocode

i	=	หน่วยข้อมูลที่ได้รับการจัดลำดับแล้วในบัฟเฟอร์รวม (merge buffer)
b_i	=	จำนวนบิตต่อสัญลักษณ์ ที่จะใช้ในการส่งหน่วยข้อมูล i ซึ่งแปรค่าได้ตั้งแต่ b_{min} ถึง b_{max} โดย $b_i \in I^+$
L_i	=	จำนวนบิตของหน่วยข้อมูล i
d_i	=	เส้นตาย (deadline) ของหน่วยข้อมูล i
s_i	=	เวลาที่หน่วยข้อมูล i เริ่มส่งได้
R_s	=	อัตราสัญลักษณ์ (symbol rate)
b_j^*	=	จำนวนบิตต่อสัญลักษณ์ ที่จะใช้ในการส่งหน่วยข้อมูลคิวโอเอส j ซึ่งแปรค่าได้ตั้งแต่ b_{min} ถึง b_{max} โดย $b_j \in I^+$

คำอธิบายอักษรย่อ

WSNs	=	Wireless Sensor Networks
QoS	=	Quality of Service
NQoS	=	Non-Quality of Service
DMS	=	Dynamic Modulation Scaling
FCFS	=	First Come First Serve
EDF	=	Earliest Deadline First
PGPS	=	Packet-by-packet Generalized Processor Sharing
GPS	=	Generalized Processor Sharing
WFQ	=	Weighted Fair Queuing
FEF	=	F irst come first serve for NQoS packets and E arliest deadline first for QoS packets and F irst come first serve for all packets
FEP	=	F irst come first serve for NQoS packets and E arliest deadline first for QoS packets and P acket-by-packet generalize processor sharing for all packets
HSNQ	=	Highest transmission Speed for Non-Quality-of-service packets

LI	=	Lowest Impact for average delay of non-quality-of-service packets and rejection rate of quality-of-service packets
QB	=	Quality-of-service traffic based
RRP	=	Reducing Rejected Packets of quality-of-service packets
LS	=	Lowest transmission Speed
RRP w/ RPS	=	Reducing Rejected Packets of quality-of-service packets with Rejected Packet Selection
LS w/ RPS	=	Lowest transmission Speed with Rejected Packet Selection

การส่งข้อมูลต่างลักษณะบนระบบที่มีพลังงานจำกัด

Supporting Heterogeneous Traffic in Energy Limited Systems

คำนำ

ความเจริญก้าวหน้าทางวิทยาศาสตร์และเทคโนโลยีมีความเกี่ยวพันกันอย่างลึกซึ้ง อย่างเช่น กล้องส่องทางไกลทำให้เข้าใจดาราศาสตร์มากยิ่งขึ้น กล้องจุลทรรศน์ทำให้เรารู้จักกับแบคทีเรีย และ ดาวเทียมก็ทำให้เราเข้าใจโลกมากยิ่งขึ้น จะเห็นว่าแนวโน้มของความเจริญก้าวหน้าเป็นไปในรูปแบบทวีคูณภายใต้เทคโนโลยีสารกึ่งตัวนำ จากจำนวนทรานซิสเตอร์ (transistors) บนชิพ (chip) ที่มีมากขึ้น ดังนั้นการประมวลผลและความจุของชิพจึงมีประสิทธิภาพเพิ่มขึ้นเป็นสองเท่าในทุกๆ หนึ่งถึงสองปี ตามกฎของมัวร์ (Moore's Law) และในขณะที่ประสิทธิภาพเพิ่มขึ้นนี้ นักวิจัยก็กำลังประยุกต์ใช้เทคโนโลยีไปพร้อมๆ กัน

ขนาดของหน่วยความจำและส่วนของการประมวลผลที่มีขนาดเล็กลง พร้อมทั้งมีราคาถูกลง และมีประสิทธิภาพสูงขึ้นในทุกๆ ปี จึงเป็นแรงผลักดันทำให้นักวิจัยคิดค้นอุปกรณ์มากมายเพื่อนำมาประยุกต์ใช้กับเทคโนโลยีในปัจจุบันและอนาคต แต่อย่างไรก็ตามการทำงานในทุกๆ ระบบล้วนแล้วแต่ต้องใช้พลังงานในการขับเคลื่อนทั้งสิ้น ดังนั้นส่วนของพลังงานจึงเป็นอีกประการหนึ่งที่ต้องได้รับการพัฒนาควบคู่กันไป ไม่ว่าจะเป็นพัฒนาแหล่งจ่ายพลังงานให้มีประสิทธิภาพในการจ่ายไฟที่นานขึ้นและมีเสถียรภาพขึ้น หรือพัฒนาวิธีการใช้พลังงานที่มีอยู่เดิมนั้นให้มีประสิทธิภาพมากยิ่งขึ้น

อุปกรณ์ประเภทหนึ่งที่แสดงให้เห็นข้อจำกัดด้านพลังงานอย่างชัดเจนได้แก่ อุปกรณ์ตรวจจับ (sensor nodes) ซึ่งเป็นอุปกรณ์ที่มีความสามารถในการตรวจจับสิ่งที่ต้องการ และประมวลผลข้อมูล พร้อมทั้งติดต่อสื่อสารกันระหว่างอุปกรณ์ โดยภายในอุปกรณ์ประกอบด้วยส่วนต่างๆ ซึ่งสามารถแยกออกเป็น 4 ระบบย่อย (Raghuathan *et al.*, 2002 a) ได้แก่ ระบบการคำนวณ ระบบการสื่อสาร ระบบการตรวจจับ และระบบการจ่ายพลังงาน โดยมีรายละเอียดดังนี้

ระบบการคำนวณ จะประกอบด้วยไมโครคอนโทรลเลอร์ (microcontroller) ซึ่งมีหน้าที่ควบคุมการทำงานของกลุ่มตัวตรวจจับ (sensors) รวมทั้งการประมวลผลของระเบียบวิธีการสื่อสาร (protocol) และข้อมูลที่ได้จากตัวตรวจจับ หน่วยควบคุม (microcontroller unit) ในปัจจุบันสามารถรองรับการ

ทำงานได้อย่างหลากหลาย ไม่ว่าจะเป็น โหมดการทำงาน (active mode) โหมดว่างงาน (idle mode) และโหมดพักการทำงาน (sleep mode) ซึ่งมีไว้เพื่อวัตถุประสงค์ในการจัดการพลังงาน โดยแต่ละโหมดมีการบริโภคพลังงานที่แตกต่างกัน เช่น ไมโครโปรเซสเซอร์ StrongARM จากบริษัท Intel ใช้พลังงานประมาณ 50 มิลลิวัตต์ ในโหมดว่างงาน แต่ใช้พลังงานเพียง 0.16 มิลลิวัตต์ ในโหมดพักการทำงาน เพื่อใช้ในการประมวลผลคำสั่ง ในขณะที่ไมโครโปรเซสเซอร์ ATmega103L AVR จากบริษัท Atmel ใช้พลังงานเพียง 16.5 มิลลิวัตต์ แต่ก็มีสมรรถนะที่ต่ำกว่า อย่างไรก็ตามการเปลี่ยนโหมดย่อมต้องใช้พลังงานและเสียเวลาในการเปลี่ยนแปลงโหมด ดังนั้นการเปลี่ยนแปลงโหมดเพื่อประหยัดพลังงานจำเป็นต้องคำนึงถึงพลังงานที่สูญเสียจากเหตุผลดังกล่าวด้วย

ในระบบการสื่อสารนั้น เป็นติดต่อสื่อสารแบบไร้สายระหว่างโหนดข้างเคียงและโลกภายนอก หลายปัจจัยที่มีผลต่อการบริโภคพลังงาน ไม่ว่าจะเป็นชนิดของการมอดูเลต (modulation) อัตราการส่งข้อมูล (data rate) ระยะทางในการส่งข้อมูล (transmission distance) และรอบในการดำเนินการ (operational duty cycle) โดยปกติการส่งสัญญาณจะแบ่งการทำงานออกเป็น 4 โหมด ได้แก่ โหมดส่งสัญญาณ (transmit mode) โหมดรับสัญญาณ (receive mode) โหมดส่งสัญญาณโดยไม่มีข้อมูล (idle mode) และโหมดพักการส่งสัญญาณ (sleep mode) (Raghunathan *et al.*, 2002 a) จากการสังเกตในการส่งสัญญาณส่วนมากนั้น การส่งสัญญาณโดยไม่มีข้อมูลจะบริโภคพลังงานค่อนข้างมาก (Raghunathan *et al.*, 2002 a) ซึ่งพอๆ กับพลังงานที่ใช้ในโหมดรับสัญญาณ ดังนั้นในกรณีที่ไม่มีข้อมูล จึงมักจะปิดระบบการสื่อสารเพื่อประหยัดพลังงาน แต่การเปลี่ยนโหมดการทำงานก็จำเป็นต้องใช้พลังงาน เช่น การเปลี่ยนจากโหมดพักการทำงานมาเป็นโหมดส่งสัญญาณ จำเป็นต้องใช้พลังงานส่วนหนึ่งในการกระตุ้นเครื่องส่งสัญญาณ (transmitter) ให้ทำการส่งหน่วยข้อมูล (packets) (Raghunathan *et al.*, 2002 a)

ส่วนระบบการตรวจจับสนั้น มีความหลากหลาย จึงทำให้วัดค่าพารามิเตอร์ (parameter) ได้หลายอย่าง เช่น อุณหภูมิ ความสว่าง เสียง ภาพ และความชื้น เป็นต้น โดยจะทำหน้าที่แปลงการเปลี่ยนแปลงสัญญาณทางกายภาพเป็นสัญญาณไฟฟ้า ซึ่งจะเป็นแบบแอนะล็อก (analog) หรือดิจิทัล (digital) ก็แล้วแต่อุปกรณ์ ในระบบการตรวจจับจะมีการบริโภคพลังงานจากหลายส่วน ไม่ว่าจะเป็นการดำเนินการชักตัวอย่างสัญญาณ (signal sampling) การดำเนินการเปลี่ยนสัญญาณทางกายภาพเป็นสัญญาณทางไฟฟ้า เป็นต้น เมื่อตัวตรวจจับ (sensor) ทำหน้าที่ในการตรวจจับที่ต่างกัน จะส่งผลให้การบริโภคพลังงานต่างกันด้วย โดยทั่วไปตัวตรวจจับแบบพาสซีฟ (passive sensors) เช่น ตัวตรวจจับอุณหภูมิ จะบริโภคพลังงานเพียงเล็กน้อยเมื่อเทียบกับระบบการทำงานส่วนอื่น แต่ตัวตรวจจับแบบแอคทีฟ (active sensors) เช่น ตัวตรวจจับภาพ จะบริโภคพลังงานค่อนข้างสูง (Raghunathan *et al.*, 2002 a)

จากที่กล่าวมาทั้ง 3 ระบบ ล้วนแล้วแต่ต้องใช้พลังงานเป็นตัวขับเคลื่อนทั้งสิ้น จะเห็นได้ว่าระบบการจ่ายพลังงานนั้นมีความสำคัญเป็นอย่างยิ่ง แหล่งพลังงานที่กล่าวถึงนี้ส่วนใหญ่แล้วจะเป็นแบตเตอรี่ (battery) โดยแบตเตอรี่จะจ่ายพลังงานให้กับทุกระบบของโหนดตรวจจับ (sensor nodes) ดังนั้นอายุการใช้งานของโหนดตรวจจับจึงขึ้นอยู่กับพลังงานในแบตเตอรี่

จากความสามารถของโหนดตรวจจับ (sensor nodes) และราคาต่อหน่วยที่ถูกลงมา จึงมีผู้นำไปใช้งานด้วยการวางโหนดตรวจจับจำนวนหนึ่ง กระจายกระจายลงในพื้นที่ที่ต้องการ โดยต้องให้ความหนาแน่นของโหนดเพียงพอที่จะประมวลผลสิ่งที่ต้องการตรวจจับ พร้อมทั้งติดต่อสื่อสารข้อมูลระหว่างโหนด และส่งข้อมูลไปยังโลกภายนอกได้ ด้วยวิธีการนี้ทำให้เกิดการใช้งานกลุ่มของโหนดตรวจจับกันอย่างแพร่หลาย ที่รู้จักกันในชื่อเครือข่ายตรวจจับไร้สาย (Wireless Sensor Networks, WSNs) (Culler *et al.*, 2004; Estrin *et al.*, 2001; Intanagonwiwat *et al.*, 2003; Lewis, 2004) ซึ่งมีประโยชน์อย่างมากมาย ได้แก่ การเตือนภัยการเกิดไฟป่า การตรวจตราพื้นที่ตามแนวชายแดนและพื้นที่วิกฤต เป็นต้น

จากจุดเด่นของเครือข่ายตรวจจับไร้สาย จึงเป็นที่คาดว่าในอนาคตอันใกล้นี้ เครือข่ายตรวจจับไร้สายจะถูกผนึกเข้าเป็นส่วนหนึ่งของเครือข่าย (network) ที่มีอยู่ในปัจจุบัน ดังนั้นเครือข่ายตรวจจับไร้สายจึงควรรองรับข้อมูลที่หลากหลายอย่างมีประสิทธิภาพ แต่ด้วยข้อจำกัดด้านพลังงานของเครือข่ายตรวจจับไร้สายดังที่ได้กล่าวในตอนต้น จึงอาจไม่เหมาะสมถ้าจะใช้กระบวนการเดียวกับเครือข่ายทั่วไปในการจัดการข้อมูล

งานวิจัยชิ้นนี้มุ่งเน้นเพื่อหาวิธีการในการรองรับข้อมูลที่มีความหลากหลาย รวมทั้งลดการบริโภคพลังงานในโหนดหนึ่งๆ ลง แต่วิธีการประหยัดพลังงานในงานวิจัยนี้แปรผันโดยตรงกับอัตราบิต (bit rate) ซึ่งแปลว่าถ้าเราสามารถลดความเร็วในการส่งข้อมูลลงยอมใช้พลังงานในการส่งข้อมูลลดลงตาม

ในการทดสอบ กำหนดให้มีหน่วยข้อมูลเข้ามายังโหนดๆ หนึ่ง ซึ่งสามารถแบ่งออกเป็น 2 ชนิด ได้แก่ หน่วยข้อมูลที่ต้องการการรับประกันคุณภาพด้านเวลา (Quality of Service packets, QoS packets) ซึ่งต่อไปจะเรียกว่าหน่วยข้อมูลคิวโอเอส (QoS packets) และหน่วยข้อมูลที่ไม่จำเป็นต้องรับประกันคุณภาพด้านเวลา (Non-Quality-of-Service packets, NQoS packets) ซึ่งต่อไปจะเรียกว่าหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) โดยหน่วยข้อมูลชนิดแรกเป็นหน่วยข้อมูลที่ต้องการการรับประกันคุณภาพด้านเวลา (QoS packets) หรือหน่วยข้อมูลที่มีเส้นตาย (deadline) ซึ่งต้องประมวลผลและส่งออก

ข้อมูลให้เสร็จสมบูรณ์ก่อนที่จะถึงเส้นตาย (deadline) ที่กำกับหน่วยข้อมูลนั้น มิฉะนั้นจะไม่สามารถนำหน่วยข้อมูลนั้นมาใช้งานได้และต้องทิ้งไป ซึ่งเราจะเรียกหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตายว่าหน่วยข้อมูลเสีย (rejected packets) และหน่วยข้อมูลชนิดที่สองเป็นหน่วยข้อมูลที่ไม่จำเป็นต้องรับประกันคุณภาพด้านเวลา (NQoS packets) ซึ่งไม่มีเส้นตายกำกับ แต่ก็มีความต้องการให้ความหน่วงเวลา (delay) มีค่าน้อยที่สุดเท่าที่เป็นไปได้ ดังนั้นการสนองความต้องการให้กับหน่วยข้อมูลทั้งสองประเภทจึงต้องอยู่บนพื้นฐานความยุติธรรม และให้ความสำคัญกับหน่วยข้อมูลทั้งสองเท่าๆกัน โดยไม่จำเป็นว่าจะต้องให้ความสำคัญกับการเกิดหน่วยข้อมูลเสีย (rejected packets) ที่เกิดขึ้นกับหน่วยข้อมูลคิวโอเอส (QoS packets) มากกว่าความหน่วงเวลาเฉลี่ย (average delay) ที่เกิดขึ้นกับหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) แต่อย่างไรก็ตาม

การประหยัดพลังงานสามารถทำได้หลายวิธี แต่ในงานวิจัยนี้ เราจะใช้การปรับค่าอัตราบิดในการส่งแต่ละหน่วยข้อมูลโดยยึดตามกระบวนการ Dynamic Modulation Scaling (DMS) (Schurgers *et al.*, 2001a) การส่งหน่วยข้อมูลด้วยอัตราบิดสูงย่อมต้องใช้พลังงานงานมาก ดังนั้นถ้าเราส่งหน่วยข้อมูลด้วยอัตราบิดที่ต่ำลงย่อมส่งผลให้ระบบบริโภคพลังงานลดลง แต่การส่งหน่วยข้อมูลด้วยอัตราบิดที่ต่ำลงจะส่งผลให้อัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส (QoS packets) มีมากขึ้น และค่าความหน่วงเวลาเฉลี่ย (average delay) ของหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) มีค่ามากขึ้น

เป้าหมายของงานวิจัยนี้คือคิดค้นวิธีการจัดลำดับการใช้ช่องสัญญาณ ระหว่างหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอสโดยยึดหลักความยุติธรรมและไม่ทำให้ฝ่ายใดฝ่ายหนึ่งเกิดการแน่นิ่ง (starving) พร้อมทั้งยังบริโภคพลังงานอย่างคุ้มค่าด้วยวิธีการปรับเปลี่ยนอัตราบิดตามหลักการ DMS ซึ่งความคุ้มค่านี้จะส่งผลให้การบริโภคพลังงานใดๆ ก็เพื่อทำให้หน่วยข้อมูลคิวโอเอส (QoS packets) มีอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) น้อย และหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) มีค่าความหน่วงเวลาเฉลี่ย (average delay) น้อยนั่นเอง

การตรวจเอกสาร

ตัวแปรสุ่มแบบต่อเนื่อง

ในการจำลองการทำงานของระบบหนึ่งๆ นั้น จำเป็นต้องใช้ข้อมูลต่างๆ มาทดสอบประสิทธิภาพของระบบนั้น แต่ลักษณะของข้อมูลที่ใช้ในการทดสอบที่ดีและเป็นที่ยอมรับมักจะเป็นข้อมูลที่มีลักษณะแบบสุ่ม (random) ซึ่งการสุ่มนั้นก็มีอยู่หลายแบบ แต่ละแบบก็เหมาะสมแตกต่างกัน ในที่นี้จะกล่าวถึงตัวแปรสุ่มแบบต่อเนื่อง (continuous random variable) (Yates and Goodman, 1999) ที่สำคัญ 3 ชนิด ได้แก่ ตัวแปรสุ่มแบบสม่ำเสมอ (Uniform Random Variable) ตัวแปรสุ่มแบบเอกซ์โพเนนเชียล (Exponential Random Variable) และตัวแปรสุ่มแบบเกาส์ (Gaussian Random Variable) ซึ่งมีรายละเอียดดังนี้

ตัวแปรสุ่มแบบสม่ำเสมอ (Uniform Random Variable)

ตัวแปรสุ่มแบบสม่ำเสมอ (X) มีลักษณะการกระจายตัวของตัวแปรคงที่เท่าๆ กันทุกตำแหน่ง ในช่วงที่กำหนด ในที่นี้ช่วงที่กำหนดคือ $[a, b]$ รายละเอียดของการกระจายตัวของตัวแปร X เป็นดังนี้

1. ฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF)

$$f_X(x) = \begin{cases} \frac{1}{b-a} & a \leq x < b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

2. ฟังก์ชันการแจกแจงสะสม (Cumulative Distribution Function, CDF)

$$F_X(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & x > b \end{cases} \quad (2)$$

3. ค่าเฉลี่ย (expected value)

$$E[X] = \frac{b+a}{2} \quad (3)$$

4. ค่าเบี่ยงเบน (variance)

$$Var[X] = \frac{(b-a)^2}{12} \quad (4)$$

ตัวแปรสุ่มแบบเอกซ์โพเนนเชียล (Exponential Random Variable)

ตัวแปรสุ่มแบบเอกซ์โพเนนเชียล (X) มีลักษณะการกระจายตัวของตัวแปรในรูปแบบเอกซ์โพเนนเชียล รายละเอียดของการกระจายตัวของตัวแปร X เป็นดังนี้

1. ฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF)

$$f_X(x) = \begin{cases} ae^{-ax} & x \geq 0 \\ 0 & otherwise \end{cases} \quad (5)$$

2. ฟังก์ชันการแจกแจงสะสม (Cumulative Distribution Function, CDF)

$$F_X(x) = \begin{cases} 1 - ae^{-ax} & x \geq 0 \\ 0 & otherwise \end{cases} \quad (6)$$

3. ค่าเฉลี่ย (expected value)

$$E[X] = \frac{1}{a} \quad (7)$$

4. ค่าเบี่ยงเบน (variance)

$$Var[X] = \frac{1}{a^2} \quad (8)$$

ตัวแปรสุ่มแบบเกาส์ (Gaussian Random Variable)

ตัวแปรสุ่มแบบเกาส์ (X) มีลักษณะการกระจายตัวของตัวแปรอยู่รอบๆ ค่าเฉลี่ยในรูปแบบระฆังคว่ำแบบสมมาตร รายละเอียดของการกระจายตัวของตัวแปร X เป็นดังนี้

1. ฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF)

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \quad (9)$$

2. ค่าเฉลี่ย (expected value)

$$E[X] = \mu \quad (10)$$

3. ค่าเบี่ยงเบน (variance)

$$Var[X] = \sigma^2 \quad (11)$$

การจัดลำดับ

การจัดลำดับ (scheduling) เป็นวิธีการที่สำคัญอย่างหนึ่ง ซึ่งช่วยเพิ่มประสิทธิภาพในกระบวนการทำงาน ส่วนใหญ่แล้วจะใช้ในแง่ของการเพิ่มความเร็วในการทำงาน โดยทำให้เครื่องจักรสามารถทำงานได้อย่างต่อเนื่อง เช่น ในโรงงานแห่งหนึ่งมีหลายสายการผลิตแต่จำเป็นต้องใช้เส้นทางในการลำเลียงสิ่งของบางส่วนร่วมกัน เพื่อให้สามารถลำเลียงสิ่งของร่วมกันได้อย่างมีประสิทธิภาพจึงจำเป็นต้องมีกระบวนการจัดลำดับสิ่งของที่จะลำเลียงจากแต่ละสายให้เหมาะสม เป็นต้น

การจัดลำดับในงานวิจัยชิ้นนี้เป็นการจัดลำดับหน่วยข้อมูล (packets) ที่เข้ามายังโหนด (node) ดังนั้นแนวคิดที่สามารถประยุกต์ใช้เพื่อจัดการปัญหาด้านการจัดลำดับจึงมีหลายวิธี (Zhang, 1995; Towsley, 1993) ดังนี้

วิธี First Come First Serve (FCFS)

วิธี FCFS (Towsley, 1993) เป็นการเรียงลำดับตามเวลาที่หน่วยข้อมูลเข้ามายังโหนด (arrival time) ซึ่งหมายความว่าหน่วยข้อมูลใดมาถึงโหนดก่อนก็ได้ส่งออกก่อน และหน่วยข้อมูลใดมาถึงช้ากว่าก็จะส่งในลำดับถัดมา วิธีนี้นิยมใช้ค่อนข้างมากเนื่องจากมีความซับซ้อนน้อย

วิธี Earliest Deadline First (EDF)

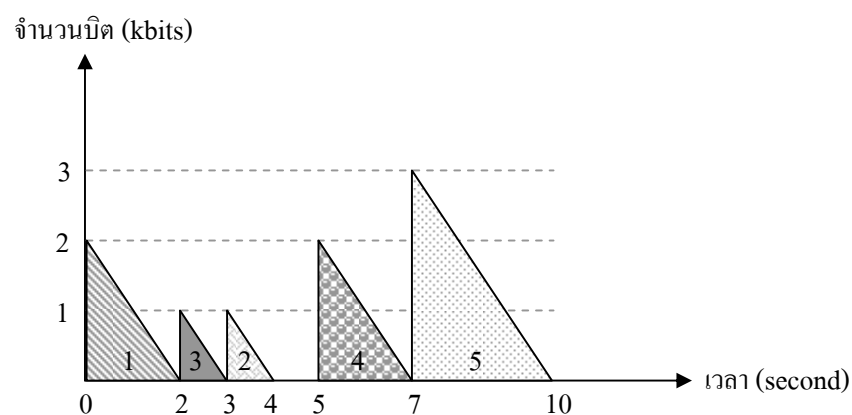
วิธี EDF (Towsley, 1993) ใช้จัดลำดับได้เฉพาะหน่วยข้อมูลคิวโอเอส (QoS packets) หรือหน่วยข้อมูลที่มีเส้นตาย (deadline) นั่นเอง โดยวิธีนี้เป็นการเรียงลำดับตามเส้นตาย (deadline) ของหน่วยข้อมูล ซึ่งหมายความว่าหน่วยข้อมูลใดมีเส้นตายใกล้หรือมีความเร่งรีบมาก ก็ให้ส่งหน่วยข้อมูลนั้นก่อน และหน่วยข้อมูลใดมีเส้นตายไกลกว่าหรือมีความเร่งรีบน้อยกว่า ก็ให้ส่งหน่วยข้อมูลนั้นในลำดับถัดมา วิธีการนี้ช่วยลดปริมาณหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตายหรือหน่วยข้อมูลเสีย (rejected packet) ลงได้ เนื่องจากการเรียงลำดับตามความเร่งรีบของแต่ละหน่วยข้อมูล ดังตัวอย่าง

สมมุติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางที่ 1 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราบิตของช่องสัญญาณขาออก 1 kbps

ตารางที่ 1 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง เพื่อแสดงกระบวนการจัดลำดับด้วยวิธี EDF

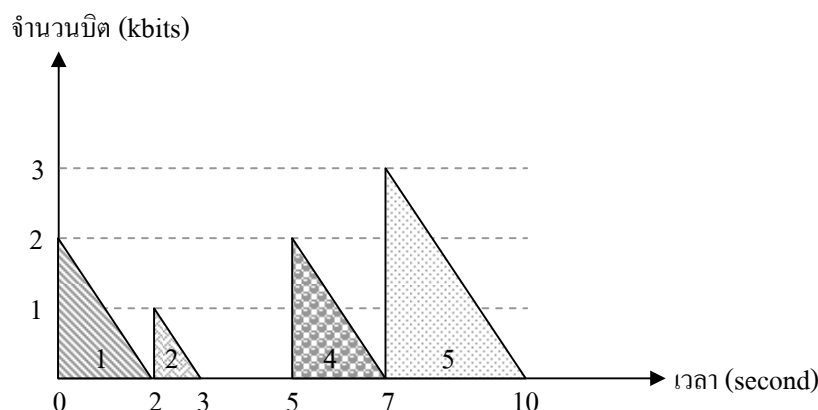
หน่วยข้อมูลที่	เวลาที่หน่วยข้อมูลมาถึง (second)	จำนวนบิตของหน่วยข้อมูล (kbits)	เส้นตาย (second)
1	0	2	7
2	1	1	6
3	2	1	3
4	5	2	10
5	6	3	10

การจัดลำดับหน่วยข้อมูลจากตารางที่ 1 ด้วยวิธี EDF ทำให้ได้ลำดับการส่งและระยะเวลาในการส่งดังแสดงในภาพที่ 1 โดยรูปสามเหลี่ยมมุมฉากแทนหน่วยข้อมูลซึ่งหมายเลขภายในสามเหลี่ยมบอกลำดับการเข้ามายังโหนดของหน่วยข้อมูลนั้น ความสูงของสามเหลี่ยมใช้แทนจำนวนบิตของหน่วยข้อมูลนั้น และฐานของแต่ละสามเหลี่ยมบ่งบอกถึงเวลาที่ต้องใช้ในการส่งหน่วยข้อมูล ความชันของสามเหลี่ยมแสดงถึงอัตราบิตของช่องสัญญาณขาออก



ภาพที่ 1 ลำดับการส่งออกหน่วยข้อมูลตัวอย่างที่ผ่านการจัดลำดับด้วยวิธี EDF

ถ้าเราลองเปรียบเทียบการจัดลำดับระหว่างวิธี EDF กับวิธี FCFS ดังแสดงในภาพที่ 1 และ 2 ตามลำดับ จะเห็นว่าการส่งออกชุดของหน่วยข้อมูลเดียวกัน ถ้าส่งออกหน่วยข้อมูลคิวโอเอสที่ผ่านการจัดลำดับด้วยวิธี FCFS จะทำให้บางหน่วยข้อมูล (packets) ไม่สามารถส่งได้ทันเส้นตาย (deadline) และจำต้องตัดหน่วยข้อมูลนั้นทิ้งไป (rejected packets)



ภาพที่ 2 ลำดับการส่งออกหน่วยข้อมูลตัวอย่างที่ผ่านการจัดลำดับด้วยวิธี FCFS

จากรูปทั้งสอง หน่วยข้อมูลชุดเดียวกัน แต่ทำการจัดลำดับด้วยวิธีที่แตกต่างกันย่อมทำให้หน่วยข้อมูลที่สามารถส่งออกได้แตกต่างกัน โดยหน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย (deadline) เมื่อใช้วิธี FCFS เพราะวิธี FCFS ให้ความสำคัญกับเวลาที่หน่วยข้อมูลนั้นมาถึง (arrival time) แต่สำหรับหน่วยข้อมูลคิวโอเอส (QoS packets) ซึ่งต้องการการรับประกันคุณภาพด้านเวลานั้น ควรให้ความสำคัญกับเส้นตาย (deadline) เพื่อรักษาปริมาณของหน่วยข้อมูลดี (accepted packets) ให้ได้มากที่สุด จึงสรุปได้ว่าวิธี EDF เป็นวิธีการจัดลำดับการส่งออกข้อมูลที่เหมาะสมกับหน่วยข้อมูลคิวโอเอส

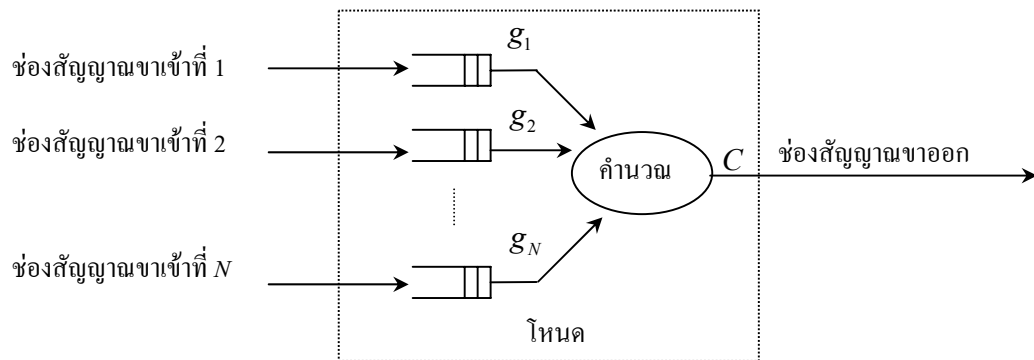
วิธี Packet-by-packet Generalized Processor Sharing (PGPS)

วิธี PGPS โดย Parekh and Gallager (1993) หรือวิธี Weighted Fair Queuing (WFQ) โดย Demers et al. (1990) เป็นวิธีการจัดลำดับอีกวิธีหนึ่ง ซึ่งรับประกันว่าข้อมูลที่เข้ามาจากแต่ละช่องสัญญาณขาเข้า (incoming channel) จะได้รับแบนด์วิธของช่องสัญญาณขาออกอย่างแน่นอน

วิธี PGPS เป็นวิธีการจัดลำดับหน่วยข้อมูลโดยพิจารณาจากเวลาส่งสำเร็จ (completion time) ของหน่วยข้อมูลที่ได้จากการจำลองส่งหน่วยข้อมูล (packets) เหล่านั้นด้วยวิธี Generalize Processor Sharing (GPS)

วิธี GPS เป็นกระบวนการที่เหมือนกับแบ่งแบนด์วิธ (bandwidth) ของช่องสัญญาณขาออก (outgoing channel) ออกเป็นส่วนๆ ซึ่งจะเป็นที่ส่วนนั้นขึ้นอยู่กับจำนวนช่องสัญญาณขาเข้า (incoming channel) ที่ขณะนั้นมีหน่วยข้อมูลรอการส่งออก แบนด์วิธที่แบ่งให้กับแต่ละช่องสัญญาณขาเข้าไม่จำเป็นต้องเท่ากันทุกช่อง ซึ่งแบนด์วิธจะเป็นเท่าไรนั้นก็ตามแต่เรากำหนด ดังนั้นหน่วยข้อมูลจากแต่ละช่องสัญญาณขาเข้าจะสามารถส่งออกได้ตามแบนด์วิธที่ได้รับ เสมือนกับหน่วยข้อมูลจากทุกช่องสัญญาณขาเข้าได้ส่งผ่านช่องสัญญาณขาออก (outgoing channel) ไปพร้อมๆ กัน การแบ่งแบนด์วิธของวิธี GPS เป็นดังนี้

สมมติให้ช่องสัญญาณขาเข้าที่ขณะนั้นมีหน่วยข้อมูลรอการส่งออกทั้งหมด N ช่อง และหน่วยข้อมูลเหล่านั้นต้องส่งผ่านช่องสัญญาณขาออกเพียงช่องสัญญาณเดียว ดังแสดงในภาพที่ 3



ภาพที่ 3 แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ให้กับหน่วยข้อมูลของช่องสัญญาณขาเข้าทั้ง N ช่อง โดยหน่วยข้อมูลจากช่องสัญญาณขาเข้าที่ i จะได้รับแบนด์วิธเท่ากับ g_i

เมื่อแบ่งแบนด์วิธของช่องสัญญาณขาออก (C) ออกเป็นส่วนๆ แล้ว หน่วยข้อมูลจากแต่ละช่องสัญญาณขาเข้าจะได้รับการรับประกันความสามารถในการระบายข้อมูลออก โดยหน่วยข้อมูลจากช่องสัญญาณขาเข้าที่ i จะได้รับแบนด์วิธเท่ากับ g_i ดังสมการ (12) โดย ϕ_i เป็นค่าถ่วงน้ำหนัก (weight) ที่เรากำหนดให้กับแต่ละช่องสัญญาณขาเข้า ซึ่งไม่จำเป็นต้องมีค่าเท่ากัน

$$g_i = \frac{\phi_i}{\sum_{j=1}^N \phi_j} \times C \quad (12)$$

$$\sum_{i=1}^N g_i = C \quad (13)$$

หลังจากนั้นจะทำการจำลองส่งข้อมูลด้วยวิธี GPS โดยแต่ละหน่วยข้อมูลจากช่องสัญญาณขาเข้าจะได้ส่งออกด้วยอัตราบิตที่ได้รับ จากการจำลองนี้ทำให้เราทราบเวลาส่งสำเร็จ (completion time) ของแต่ละหน่วยข้อมูล เวลาส่งสำเร็จดังกล่าวจะถูกนำไปใช้ในวิธี PGPS นั่นคือในการส่งจริงเราจะทำการเรียงลำดับการส่งหน่วยข้อมูลตามเวลาส่งสำเร็จที่คำนวณจากวิธี GPS จากก่อนไปหลังตามลำดับ ถ้าเกิดกรณีที่หน่วยข้อมูลมีเวลาส่งสำเร็จเท่ากัน เราจะให้หน่วยข้อมูลที่เข้ามายังโหนด (arrival time) ก่อนได้รับลำดับก่อน

เพื่อให้เข้าใจการจัดลำดับของวิธี PGPS ได้ง่ายขึ้น จึงแสดงตัวอย่างการจัดลำดับ โดยสมมติให้โหนดๆ หนึ่งมีหน่วยข้อมูลเข้ามาจาก 2 เส้นทาง ($N = 2$) และหน่วยข้อมูลเหล่านั้นจะถูกส่งออกด้วยอัตราบิตของช่องสัญญาณออกเท่ากับ 1 kbps ชุดข้อมูลตัวอย่างแสดงในตารางที่ 2

ตารางที่ 2 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง เพื่อแสดงกระบวนการจัดลำดับด้วยวิธี PGPS

ช่องสัญญาณขาเข้าที่ 1			ช่องสัญญาณขาเข้าที่ 2		
หน่วยข้อมูลที่	เวลาที่หน่วยข้อมูลมาถึง	จำนวนบิตของหน่วยข้อมูล	หน่วยข้อมูลที่	เวลาที่หน่วยข้อมูลมาถึง	จำนวนบิตของหน่วยข้อมูล
	(second)	(kbits)		(second)	(kbits)
1-1	1	2	2-1	0	3
1-2	2	1	2-2	5	2
1-3	3	1	2-3	9	2
1-4	11	2	-	-	-

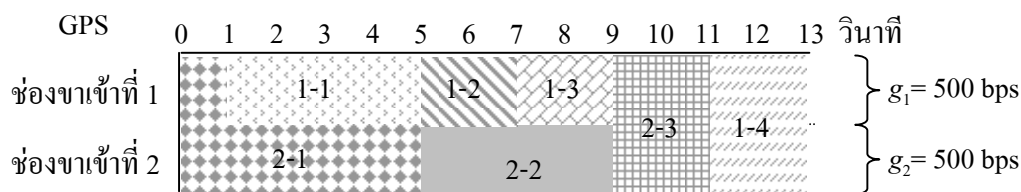
ที่มา: Parekh and Gallager (1993)

ถ้ากำหนดให้ $\phi_1 = \phi_2$ แล้ว แบนด์วิธ $g_1 = g_2 = 0.5$ kbps ดังภาพที่ 4



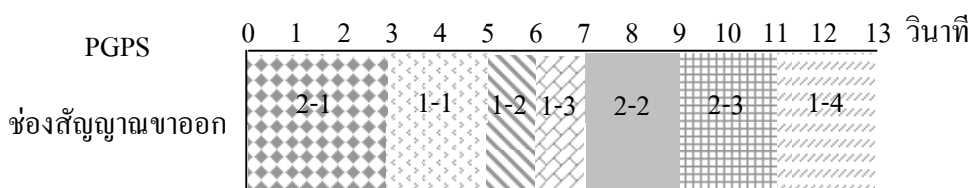
ภาพที่ 4 แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ด้วยวิธี GPS แบบอย่างละครึ่งให้กับหน่วยข้อมูลจากช่องสัญญาณขาเข้าทั้งสอง

เมื่อจำลองการส่งหน่วยข้อมูลจากตารางที่ 2 ด้วยอัตราบิต g_1 และ g_2 ตามวิธี GPS จะได้ลักษณะการส่งดังภาพที่ 5 โดยแนวตั้งแสดงแบนด์วิธของแต่ละช่องสัญญาณขาเข้าได้รับ พื้นที่สีเหลี่ยมจึงแสดงถึงจำนวนบิตของหน่วยข้อมูล ดังนั้นแนวนอนแสดงถึงเวลาซึ่งจะบอกถึงเวลาที่หน่วยข้อมูลนั้นเริ่มส่งจนกระทั่งส่งสำเร็จ



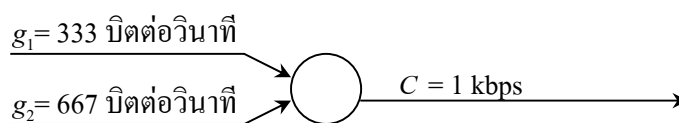
ภาพที่ 5 กระบวนการจำลองการส่งชุดข้อมูลตัวอย่างด้วยวิธี GPS แบบอย่างละครึ่ง

จากการจำลองส่งด้วยวิธี GPS ในภาพที่ 5 ทำให้เราทราบเวลาส่งสำเร็จ (completion time) ของทุกหน่วยข้อมูล ดังนั้นเราจึงสามารถจัดลำดับส่งด้วยวิธี PGPS ซึ่งเป็นการเรียงลำดับหน่วยข้อมูลจากเวลาส่งสำเร็จที่ได้จากวิธี GPS จากนั้นน้อยไปมากตามลำดับ ดังภาพที่ 6



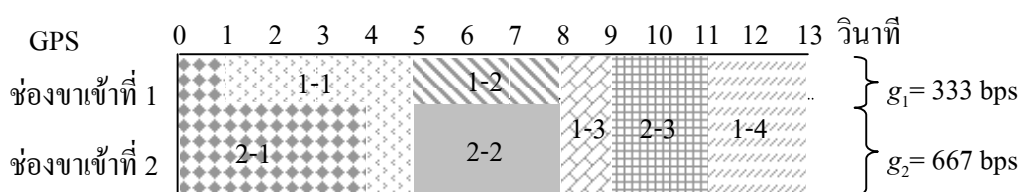
ภาพที่ 6 การจัดลำดับชุดข้อมูลตัวอย่างด้วยวิธี PGPS

ถ้ากำหนดให้ $2\phi_1 = \phi_2$ แล้ว แบนด์วิธ $g_1 = 333$ บิตต่อวินาที และ $g_2 = 667$ บิตต่อวินาที ดังภาพที่ 7



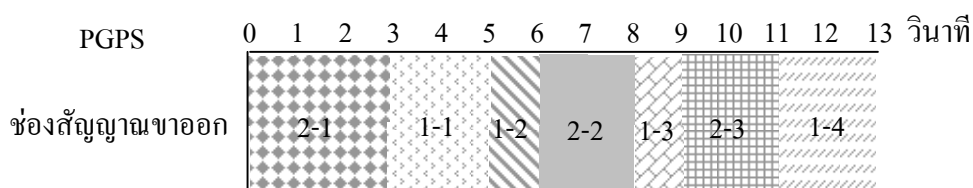
ภาพที่ 7 แผนภาพการจัดสรรแบนด์วิธของช่องสัญญาณขาออก (C) ด้วยวิธี GPS แบบหนึ่งต่อสอง ให้กับหน่วยข้อมูลจากช่องสัญญาณขาเข้าทั้งสอง

เมื่อจำลองการส่งหน่วยข้อมูลจากตารางที่ 2 ด้วยอัตราบิต g_1 และ g_2 ตามวิธี GPS จะได้ลักษณะการส่งดังภาพที่ 8 โดยแนวตั้งแสดงแบนด์วิธที่แต่ละช่องสัญญาณขาเข้าได้รับ พื้นที่สีเหลี่ยมจึงแสดงถึงจำนวนบิตของหน่วยข้อมูล ดังนั้นแนวนอนแสดงถึงเวลาซึ่งจะบอกถึงเวลาที่หน่วยข้อมูลนั้นเริ่มส่งจนกระทั่งส่งสำเร็จ



ภาพที่ 8 กระบวนการจำลองการส่งชุดข้อมูลตัวอย่างด้วยวิธี GPS แบบหนึ่งต่อสอง

จากการจำลองส่งด้วยวิธี GPS ในภาพที่ 8 ทำให้เราทราบเวลาส่งสำเร็จ (completion time) ของทุกหน่วยข้อมูล ดังนั้นเราจึงสามารถจัดลำดับส่งด้วยวิธี PGPS ซึ่งเป็นการเรียงลำดับหน่วยข้อมูลจากเวลาส่งสำเร็จที่ได้จากวิธี GPS จากนั้นน้อยไปมากตามลำดับ ดังภาพที่ 9



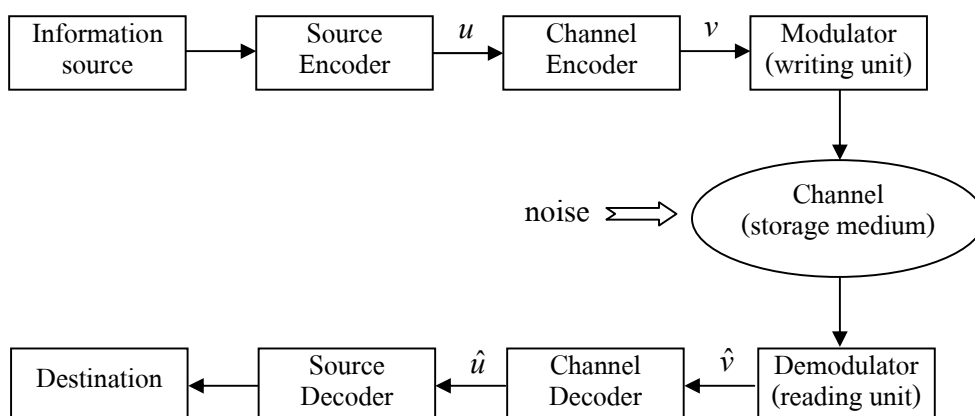
ภาพที่ 9 การจัดลำดับชุดข้อมูลตัวอย่างด้วยวิธี PGPS

วิธีการจัดลำดับ PGPS มีความเหมาะสมกับข้อมูลที่มาจากหลายเส้นทาง เนื่องจากรับประกันการให้ช่องสัญญาณกับหน่วยข้อมูล ว่า ณ เวลาใดๆ หน่วยข้อมูลต้องได้รับแบนด์วิธไม่น้อยกว่าแบนด์วิธที่แบ่งให้ (g_i) และในกรณีที่มีข้อมูลจากเส้นทางหนึ่งเข้ามาจำนวนมาก ก็ไม่ทำให้หน่วยข้อมูลจาก

เส้นทางที่เหลือนั้น (starving) อีกทั้งยังสามารถปรับค่าถ่วงน้ำหนัก (ϕ_i) ให้เหมาะสมกับสถานการณ์ได้

การมอดูเลต

การส่งข้อมูลหนึ่งๆ จากผู้ส่ง (transmitter) ไปยังผู้รับ (receiver) นั้น ต้องผ่านกระบวนการมากมายเพื่อให้ส่งข้อมูลได้อย่างมีประสิทธิภาพ ดังภาพที่ 10



ภาพที่ 10 กระบวนการพื้นฐานในการส่งและรับข้อมูล

กระบวนการส่งและรับข้อมูลประกอบด้วยหลายขั้นตอน ซึ่งแต่ละขั้นตอนก็มีหน้าที่แตกต่างกัน ซึ่งมีรายละเอียดดังตารางที่ 3 จากหน้าที่ในตารางที่ 3 จะเห็นว่ากระบวนการในการส่งข้อมูลที่ขาดไม่ได้คือการมอดูเลต (modulation) เนื่องจากการแปลงบิตข้อมูลให้อยู่ในรูปสัญญาณไฟฟ้า เพื่อให้สามารถส่งข้อมูลผ่านช่องสัญญาณได้ การแปลงบิตข้อมูลเป็นสัญญาณไฟฟ้านั้นจะต้องใส่ข้อมูลไว้บนคลื่นพาห์ (carrier) ด้วยวิธีการสวิตชิง (switching, keying) โดยข้อมูลนั้นจะอยู่บนคลื่นพาห์ (carrier) ในรูปแอมพลิจูด (amplitude) ความถี่ (frequency) และ/หรือ เฟส (phase) ด้วยวิธีการมอดูเลตสัญญาณพื้นฐานคือ การมอดูเลตเชิงเลขทางแอมพลิจูด (amplitude-shift keying, ASK) การมอดูเลตเชิงเลขทางความถี่ (frequency-shift keying, FSK) และ/หรือ การมอดูเลตเชิงเลขทางเฟส (phase-shift keying, PSK) ตามลำดับ (Haykin, 2001) วิธีการพื้นฐานนี้เป็นการใส่หนึ่งบิตข้อมูลลงในหนึ่งสัญญาณ (signal) นั่นคือระยะเวลาในการส่งหนึ่งสัญญาณจะส่งออกข้อมูลได้เพียงหนึ่งบิต ดังนั้นระยะเวลาในการส่งข้อมูลหนึ่งบิต (T_{bit}) จะมีค่าเท่ากับระยะเวลาในการส่งข้อมูลหนึ่งสัญญาณ (T)

ตารางที่ 3 รายละเอียดในแต่ละขั้นตอนของการส่งและรับข้อมูลจากต้นทางไปยังปลายทาง

ขั้นตอน	รายละเอียด
แหล่งข้อมูล (Information source)	แหล่งข้อมูลจริงๆ ที่ยังไม่ได้ผ่านกระบวนการเข้ารหัส
ตัวเข้ารหัสแหล่งข้อมูล (Source Encoder)	เป็นการเข้ารหัสข้อมูลเพื่อลดส่วนซ้ำซ้อน (redundancy) ซึ่งเป็นจำนวนบิตที่ไม่จำเป็นต้องมีก็ได้ ขั้นตอนนี้จะทำให้เหลือเฉพาะบิตข้อมูลที่จำเป็น ซึ่งเราสามารถแปลงข้อมูลกลับมาให้เป็นเหมือนเดิมได้ด้วยกระบวนการถอดรหัส
ตัวเข้ารหัสช่องสัญญาณ (Channel Encoder)	เป็นการเข้ารหัสข้อมูลโดยเพิ่มบิตที่จำเป็นลงไป เพื่อให้ข้อมูลมีความทนทานต่อสัญญาณรบกวน
มอดูเลเตอร์ (Modulator)	เป็นการแปลงบิตข้อมูลเป็นสัญญาณทางไฟฟ้า เพื่อให้สามารถส่งข้อมูลออกไปได้
ดีมอดูเลเตอร์ (Demodulator)	เป็นการแปลงสัญญาณทางไฟฟ้าที่ได้รับจากช่องสัญญาณ (channel) มาเป็นบิตข้อมูล
ตัวถอดรหัสช่องสัญญาณ (Channel Decoder)	เป็นการถอดรหัสข้อมูลที่รับมา ซึ่งส่วนใหญ่แล้วจะมีสัญญาณรบกวน (noise) ปะปนเข้ามากับข้อมูล ดังนั้น กระบวนการนี้จะทำให้เราทราบว่าข้อมูลมีการผิดพลาด (error) หรือไม่ ซึ่งวิธีการเข้ารหัสบางรูปแบบสามารถแก้การผิดพลาดที่เกิดขึ้นได้
ตัวถอดรหัสแหล่งข้อมูล (Source Encoder)	เป็นการถอดรหัสข้อมูลโดยเพิ่มส่วนซ้ำซ้อน (redundancy) เข้าไปเพื่อให้ได้ข้อมูลเดิมก่อนการเข้ารหัสออกมา
แหล่งปลายทาง (Destination)	ข้อมูลที่ได้รับเป็นข้อมูลที่เครื่องส่ง (transmitter) ต้องการส่ง โดยถ้าข้อมูลที่รับได้เป็นข้อมูลเดียวกันกับเครื่องส่ง (transmitter) ส่งทุกประการ ก็แสดงว่าข้อมูลที่รับได้นี้ถูกต้อง แต่ในบางกรณี ก็อาจได้ข้อมูลไม่ตรงกันเนื่องจากมีสัญญาณรบกวน (noise) เข้ามาปนอยู่กับสัญญาณข้อมูล

ต่อมาได้มีการพัฒนาวิธีการมอดูเลตเพื่อให้ส่งออกข้อมูลในหนึ่งหน่วยเวลาได้มากขึ้น โดยการใส่ข้อมูลจำนวน n บิต ลงในหนึ่งสัญญาณ ดังนั้นระยะเวลาในการส่งข้อมูลหนึ่งบิต (T_{bit}) จะมีค่าน้อยกว่าระยะเวลาในการส่งข้อมูลหนึ่งสัญญาณ (T) ทั้งหมด n เท่า วิธีการมอดูเลตเหล่านี้ได้แก่ การมอดูเลตเชิงเลขทางแอมพลิจูดแบบเอ็มอาร์อี (M-ary amplitude-shift keying, M-ary ASK) การมอดูเลต

เชิงเลขทางความถี่แบบเอ็มอาร์เอ (M-ary frequency-shift keying, M-ary FSK) และ/หรือ การมอดูเลตเชิงเลขทางเฟสแบบเอ็มอาร์เอ (M-ary phase-shift keying, M-ary PSK) และเพื่อเพิ่มประสิทธิภาพจึงมีการรวมวิธีการมอดูเลตให้อยู่ในรูปแบบไฮบริด (hybrid) (Haykin, 2001) ตัวอย่างเช่นการมอดูเลตเชิงเลขทางแอมพลิจูด-เฟสแบบเอ็มอาร์เอ (M-ary amplitude-phase keying, M-ary APK) ซึ่งเป็นวิธีการใส่ข้อมูลลงคลื่นพาห์ในรูปแบบแอมพลิจูดและเฟส โดยรูปแบบที่น่าสนใจของวิธีมอดูเลตนี้คือ การมอดูเลตเชิงเลขแบบเอ็มอาร์เอควอดเรเจอร์ (M-ary quadrature amplitude modulation, M-ary QAM)

ในการส่งข้อมูลดิจิทัล (digital) ผ่านช่องสัญญาณแบบแบนด์พาส (bandpass channel) ที่มีแบนด์วิธ (bandwidth) จำกัด แต่มีกำลังงาน (power) มากเพียงพอแล้ว วิธีการเข้าสัญญาณแบบเอ็มอาร์เอ (M-ary signaling schemes) จะเป็นที่ยอมรับกว่าวิธีการเข้าสัญญาณแบบไบนารี (binary signaling schemes) เนื่องจากการใช้งานจริง การหาช่องสัญญาณที่มีแบนด์วิธพอดีกับความต้องการในการส่งออกข้อมูลนั้นยากมาก โดยส่วนใหญ่แล้วแบนด์วิธของช่องสัญญาณจะมีขนาดเล็กกว่าความต้องการในการส่งออกข้อมูล เราจึงใช้วิธีการเข้าสัญญาณแบบเอ็มอาร์เอ เพื่อเพิ่มประสิทธิภาพในการส่งออกข้อมูลของช่องสัญญาณนั้นให้เพียงพอกับความต้องการ

ในกรณีที่แบนด์วิธของช่องสัญญาณมีขนาดไม่เพียงพอกับความต้องการในการส่งออกข้อมูล เราสามารถแก้ไขข้อจำกัดนี้ด้วยวิธีการเข้าสัญญาณแบบเอ็มอาร์เอ โดยหนึ่งสัญญาณ (signal) ที่จะส่งออกนั้นแทนข้อมูล n บิต ($n = \log_2 M$) วิธีนี้จะทำให้สามารถส่งข้อมูลผ่านช่องสัญญาณได้มากขึ้นจากเดิม n เท่า แต่ความสามารถในการส่งที่เพิ่มขึ้นนี้จำเป็นต้องแลกกับพลังงานในการส่งสัญญาณที่สูงขึ้น และ/หรือ ความผิดพลาดในการส่งข้อมูล (bit error rate, BER) ที่สูงขึ้น เราจะแสดงให้เห็นความสัมพันธ์ของค่าทั้งสามโดยใช้วิธีการมอดูเลตเชิงเลขแบบเอ็มอาร์เอควอดเรเจอร์ (M-ary QAM) เนื่องจากง่ายต่อการวิเคราะห์

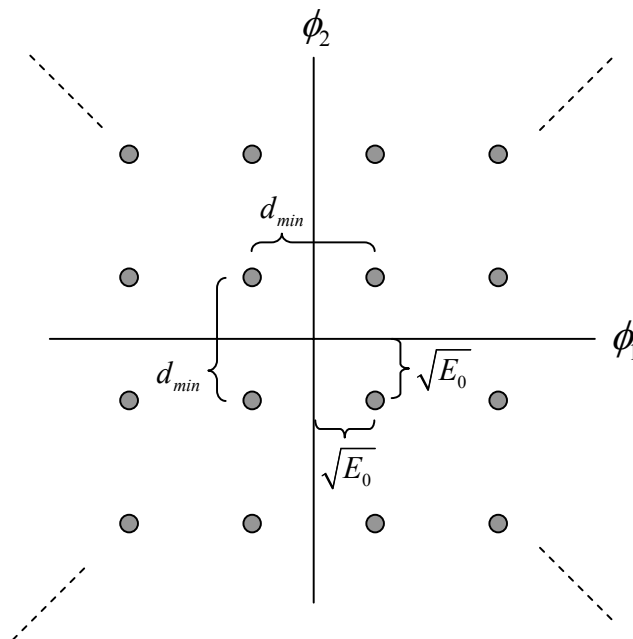
การมอดูเลตเชิงเลขแบบเอ็มอาร์เอควอดเรเจอร์ (M-ary QAM)

การมอดูเลตเชิงเลขแบบเอ็มอาร์เอควอดเรเจอร์ (M-ary QAM) (Haykin, 2001) เป็นการประกอบกันของการมอดูเลตทางแอมพลิจูดของพัลส์แบบเอ็มอาร์เอ (M-ary pulse amplitude modulation, M-ary PAM) 2 แกนที่ตั้งฉากกัน ดังสมการ (14) และ (15)

$$\phi_1(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t), \quad 0 \leq t \leq T \quad (14)$$

$$\phi_2(t) = \sqrt{\frac{2}{T}} \sin(2\pi f_c t), \quad 0 \leq t \leq T \quad (15)$$

ในแต่ละแกนมีการแปรค่าแอมพลิจูดตามวิธีการมอดูเลตทางแอมพลิจูดของพัลส์แบบเอ็มอาร์เอช จึงได้ตำแหน่งของข้อมูลบนระนาบ (ϕ_1, ϕ_2) ดังภาพที่ 11



ภาพที่ 11 แผนภาพตำแหน่ง (constellation) เมื่อทำการมอดูเลตทางแอมพลิจูดของพัลส์แบบเอ็มอาร์เอช

ข้อมูลที่ i ให้แทนด้วยสัญลักษณ์ s_i โดยตำแหน่งของข้อมูลนั้นอยู่บนระนาบ (ϕ_1, ϕ_2) ณ คู่ลำดับ $(a_i d_{min}/2, b_i d_{min}/2)$ โดยที่ d_{min} คือระยะทางที่สั้นที่สุดระหว่างข้อมูลตัวที่อยู่ในตำแหน่งติดกัน $a_i, b_i \in I$ และ $i = 1, 2, \dots, M$ และให้ $d_{min}/2 = \sqrt{E_0}$ ซึ่ง E_0 เป็นพลังงานในการส่งสัญญาณที่มีแอมพลิจูดต่ำที่สุด จะได้สัญญาณแบบเอ็มอาร์เอชคิวเอ็ม (M-ary QAM) ที่ใช้ในการส่งออกข้อมูลที่ i ดังสมการ (16) โดย $0 \leq t \leq T$ และ $k = 0, \pm 1, \pm 2, \dots$

$$s_i(t) = \sqrt{\frac{2E_0}{T}} a_i \cos(2\pi f_c t) - \sqrt{\frac{2E_0}{T}} b_i \sin(2\pi f_c t) \quad (16)$$

เราสามารถหาความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM) ได้ตามลำดับดังนี้

1. ความน่าจะเป็นในการตัดสินใจถูกต้อง (probability of correct detection, P_c) เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม

$$P_c = (1 - P_e')^2 \quad (17)$$

โดย P_e' คือความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์เมื่อเข้าสู่สัญญาณด้วยวิธีแอลอาร์ซีพีเอ็ม (L-ary PAM) ซึ่ง $L = \sqrt{M}$

$1 - P_e'$ คือความน่าจะเป็นในการตัดสินใจถูกต้อง เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์ซีพีเอ็ม (M-ary PAM)

$(1 - P_e')^2$ คือความน่าจะเป็นในการตัดสินใจถูกต้อง เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์ซีพีเอ็ม (M-ary PAM) โดยตัดสินใจถูกต้องทั้งแกน ϕ_1 และแกน ϕ_2

2. ความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM)

$$P_e' = \left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc}\left(\sqrt{\frac{E_0}{N_0}}\right) \quad (18)$$

3. ความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM)

$$P_e = 1 - P_c = 1 - (1 - P_e')^2 \quad (19)$$

เมื่อสมมุติว่า P_e' มีค่าน้อยมากเมื่อเทียบกับพจน์ที่เหลือ

$$P_e \approx 2P_e' \quad (20)$$

$$P_e \approx 2\left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc}\left(\sqrt{\frac{E_0}{N_0}}\right) \quad (21)$$

4. พลังงานที่ใช้ในการส่งแต่ละสัญญาณเมื่อเข้าสัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM) มีค่าไม่เท่ากัน โดยขึ้นอยู่กับตำแหน่งของสัญญาณนั้นบนระนาบ (ϕ_1, ϕ_2) เราสามารถหาพลังงานเฉลี่ยที่ใช้ในการส่งสัญญาณใดๆ (E_{av}) ได้ดังนี้

$$\begin{aligned} E_{av} &= 2 \left[\frac{2E_0}{\sqrt{L}} \sum_{i=1}^{L/2} (2i-1)^2 \right] \\ &= \frac{2(L^2-1)E_0}{3} \\ &= \frac{2(M-1)E_0}{3} \end{aligned} \quad (22)$$

5. ความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) เมื่อเข้าสัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM)

$$P_e \approx 2 \left(1 - \frac{1}{\sqrt{M}} \right) \operatorname{erfc} \left(\sqrt{\frac{3E_{av}}{2(M-1)N_0}} \right) \quad (23)$$

เราสามารถหาอัตราบิตผิดพลาด (bit error rate, BER) ได้จากความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) การมอดูเลตแบบเอ็มอาร์เคอีเอ็มจะมีจุดข้อมูล (message point) บนระนาบ (ϕ_1, ϕ_2) ทั้งหมด M จุด แต่ละจุดแทนข้อมูลขนาด $\log_2 M$ บิต การจัดเรียงจุดข้อมูล (message points) บนระนาบจะใช้หลักการรหัสเกรย์ (Gray code) โดยจุดข้อมูล (message point) ที่อยู่ติดกันจะมีบิตข้อมูลต่างกันเพียงหนึ่งบิตเท่านั้น ในกรณีที่ความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) มีค่าน้อยแล้ว การตัดสินใจผิดพลาดเกือบทั้งหมดจะเป็นการตัดสินใจเลือกจุดข้อมูล (message point) ที่อยู่ติดกับจุดข้อมูลที่ต้องการ ดังนั้น การเกิดความผิดพลาดของสัญลักษณ์ (symbol error) จะเป็นความผิดพลาดที่เกิดจากการผิดเพียงบิตเดียวของสัญลักษณ์นั้น

เนื่องจากหนึ่งจุดข้อมูล (message point) มีขนาด $\log_2 M$ บิต หรืออีกความหมายหนึ่งคือ $\log_2 M$ บิตต่อหนึ่งสัญลักษณ์ เราจึงหาความสัมพันธ์ระหว่างความน่าจะเป็นในการเกิดความผิดพลาดของสัญลักษณ์ (probability of symbol error, P_e) กับอัตราบิตผิดพลาด (bit error rate, BER) ได้ดังนี้

$$\begin{aligned}
P_e &= P\left(\bigcup_{i=1}^{\log_2 M} \{i^{\text{th}} \text{ bit is in error}\}\right) \\
&\leq \sum_{i=1}^{\log_2 M} P(i^{\text{th}} \text{ bit is in error}) \\
&= \log_2 M \cdot \text{BER}
\end{aligned} \tag{24}$$

จากสมการ (23) และ (24) เมื่อเข้าสู่สัญญาณด้วยวิธีเอ็มอาร์เคอีเอ็ม (M-ary QAM) แล้วอัตราบิตผิดพลาด (bit error rate, BER) จะมีค่าดังสมการ (25)

$$\text{BER} = \frac{2}{\log_2 M} \left(1 - \frac{1}{\sqrt{M}}\right) \text{erfc} \left(\sqrt{\frac{3E_{av}}{2(M-1)N_0}} \right) \tag{25}$$

ถ้ากำหนดให้ b คือจำนวนบิตต่อสัญลักษณ์ซึ่งมีค่าเท่ากับ $\log_2 M$ และ SNR คืออัตราส่วนสัญญาณต่อสัญญาณรบกวน (signal-to-noise ratio) ซึ่งมีค่าเท่ากับ $E_{av}/2N_0$ แล้ว ความสัมพันธ์ระหว่างอัตราบิตผิดพลาด (bit error rate, BER) อัตราส่วนสัญญาณต่อสัญญาณรบกวน (signal-to-noise ratio, SNR) และจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) เป็นดังสมการ (26)

$$\text{BER} = \frac{2}{b} \left(1 - \frac{1}{2^{b/2}}\right) \text{erfc} \left(\sqrt{3 \cdot \frac{\text{SNR}}{2^b - 1}} \right) \tag{26}$$

การส่งออกสัญญาณใดๆ มีการใช้กำลังงาน (power) ในการดำเนินการ 2 ส่วน คือกำลังงานในส่วนการส่งคลื่นวิทยุ (P_s) และกำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_e) โดยกำลังงานในส่วนแรกใช้ในการส่งคลื่นวิทยุ (P_s) ซึ่งเป็นตัวกำหนดสมรรถนะ (performance) เช่นอัตราบิตผิดพลาด (BER) ระยะทางระหว่างเครื่องส่ง (transmitter) และเครื่องรับ (receiver) เป็นต้น

$$P_s = \frac{P_n}{A} \cdot \text{SNR} = \frac{N_0 \beta R_s}{A} \cdot \text{SNR} \tag{27}$$

- โดย P_n คือกำลังงานของสัญญาณรบกวน (noise power)
 A คือการสูญเสียในการส่งทั้งหมด (transmission loss)
 N_0 คือความหนาแน่นกำลังเชิงสเปกตรัมของสัญญาณรบกวน (noise power spectral density)
 β คือการสูญเสียอันเนื่องมาจากอุปกรณ์ เช่น ความไม่เป็นอุดมคติของวงจรกรอง (filter)
 R_s คืออัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (symbol rate)

เราสามารถจัดรูปแบบสมการกำลังงานในส่วนการส่งคลื่นวิทยุ (P_S) ใหม่ เพื่อแสดงให้เห็นความสัมพันธ์กับจำนวนบิตต่อสัญลักษณ์ (b) ได้ดังสมการ (28) โดยค่า C_S แสดงถึงคุณภาพในการส่งข้อมูล ซึ่งแปรค่าตามจำนวนบิตต่อสัญลักษณ์ (b) แต่ค่าที่แปรนั้นมีค่าน้อยมากจนถือว่าเป็นค่าคงที่

$$P_S = C_S \cdot R_S \cdot (2^b - 1) \quad (28)$$

$$C_S = \frac{N_0 \cdot \beta}{A} \cdot \frac{1}{3} \cdot \left[\operatorname{erfc}^{-1}(1/2) \cdot \left(1 - \frac{1}{2^{b/2}} \right) \cdot b \cdot \operatorname{BER} \right] \quad (29)$$

กำลังงานในส่วนที่สองเป็นกำลังงานที่ใช้ในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) ไม่ว่าจากการฟิลเตอร์ (filtering) การมอดูเลต (modulating) การแปลงผันสัญญาณ (upconverting) และอื่นๆ

$$P_E = \left[C_E + C_R \cdot \frac{R_{S_{\max}}}{R_S} \right] \cdot R_S \quad (30)$$

โดย C_E เป็นค่าคงที่ที่เกิดจากการใช้พลังงานจากวงจรอิเล็กทรอนิกส์ (electronic circuit)

C_R เป็นค่าที่เกิดจากความเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (symbol rate)

$R_{S_{\max}}$ คืออัตราเร็วสูงสุดในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (maximum symbol rate)

เนื่องจากข้อจำกัดของอุปกรณ์ที่รับส่งข้อมูลในปัจจุบัน การเปลี่ยนอัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (R_S) ทำให้คาบการชักตัวอย่าง (sampling period) ไม่คงที่ จึงไม่สามารถชักตัวอย่าง (sampling) ได้ในจังหวะที่ถูกต้อง ส่งผลให้ไม่สามารถนำข้อมูลออกมาได้อย่างถูกต้อง ดังนั้นการส่งออกข้อมูลในปัจจุบันจึงจำเป็นต้องส่งด้วยอัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (symbol rate) ที่คงที่ จากสมการ (30) จึงถูกแปลงรูปใหม่ดังสมการ (31) โดย C_{ER} เป็นค่าคงที่ในการสูญเสีย (loss) จากวงจรอิเล็กทรอนิกส์และที่อัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (symbol rate, R_S)

$$P_E = [C_E + C_R] \cdot R_S = C_{ER} \cdot R_S \quad (31)$$

จากที่กล่าวแล้วข้างต้นว่า กรณิที่แบนด์วิธของช่องสัญญาณมีขนาดไม่เพียงพอกับความต้องการในการส่งออกข้อมูล เราสามารถใช้วิธีการเข้าสัญญาณแบบเอ็มอาร์บี โดยการส่งหนึ่งสัญญาณ (signal) เป็นการส่งออกข้อมูลจำนวน b บิต ($b = \log_2 M$) วิธีนี้ทำให้ส่งข้อมูลได้มากขึ้นจากเดิม b เท่า แต่

ความสามารถในการส่งที่เพิ่มขึ้นนี้ เมื่อเราต้องการให้ความผิดพลาดในการส่งข้อมูล (bit error rate, BER) คงเดิม เราจึงจำเป็นต้องแลกกับพลังงานในการส่งสัญญาณที่สูงขึ้น

วิธี Dynamic Modulation Scaling

กระบวนการปรับเปลี่ยนจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ถูกเรียกว่าวิธี Dynamic Modulation Scaling (DMS) โดยการปรับเปลี่ยนนี้มีผลต่อระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) และพลังงานที่ใช้ในการส่งหนึ่งบิตข้อมูล (E_{bit}) ดังสมการ

$$E_{bit} = (P_S + P_E) \cdot T_{bit} \quad (32)$$

$$E_{bit} = [C_S \cdot (2^b - 1) + C_{ER}] \cdot \frac{1}{b} \quad (33)$$

$$T_{bit} = \frac{1}{b \cdot R_S} \quad (34)$$

เพื่อให้เห็นความสัมพันธ์ของจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) และพลังงานที่ใช้ในการส่งหนึ่งบิตข้อมูล (E_{bit}) ได้ชัดเจนขึ้น จึงแสดงความสัมพันธ์ของสมการให้อยู่ในรูปแบบกราฟ โดยภาพที่ 12 ใช้ค่ากำหนดจากงานวิจัยของ Schurgers et al. (2001a)

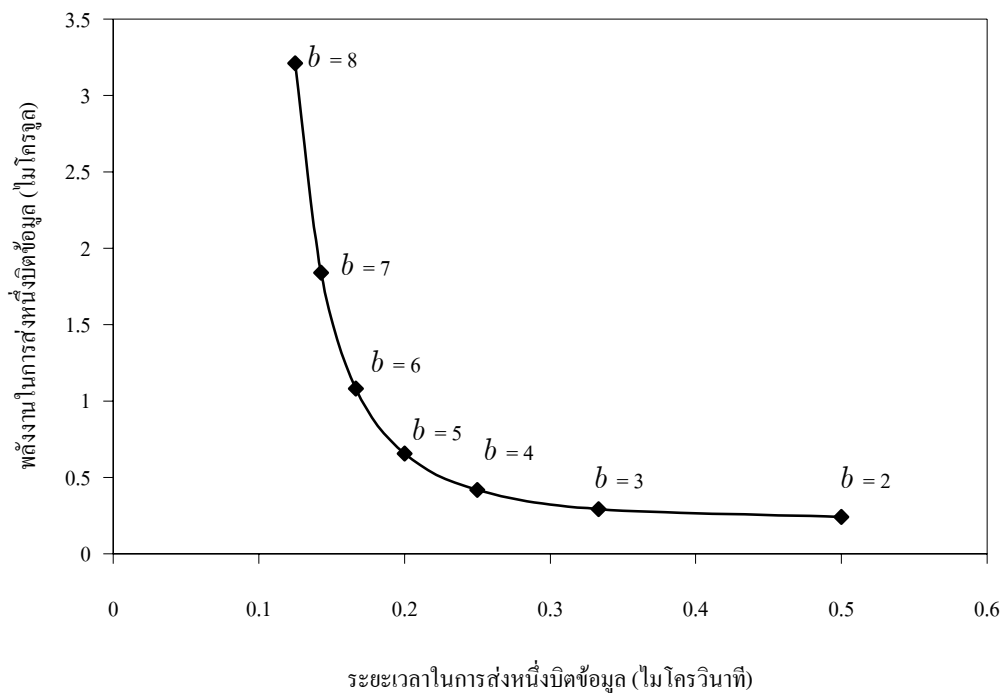
กำหนดให้

อัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (R_S) มีค่าเท่ากับ 1 MHz

อัตราบิตผิดพลาด (BER) มีค่าเท่ากับ 10^{-5}

ค่าคงที่ในการส่งคลื่นวิทยุ (C_S) มีค่าเท่ากับ 10^{-7} (ที่ $b = 4$)

ค่าคงที่ในวงจรอิเล็กทรอนิกส์ (C_{ER}) มีค่าเท่ากับ 1.8×10^{-7} จูล



ภาพที่ 12 ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b)

จากภาพที่ 12 แสดงให้เห็นถึงความคุ้มค่าเมื่อลดการใช้พลังงานลงโดยแลกกับระยะเวลาในการส่งที่เพิ่มขึ้น ดังนั้นถ้าเราต้องการส่งปริมาณบิตข้อมูลต่อหนึ่งหน่วยเวลาให้มากขึ้น นั่นคือระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) น้อยลง ก็ต้องยอมสูญเสียพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) สูงขึ้น เพื่อรักษาระดับอัตราผิดพลาด (BER) ให้คงเดิม ในทางกลับกัน ถ้าเราต้องการให้อัตราการใช้พลังงานต่ำก็จำเป็นต้องให้ระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) สูงขึ้น

ระยะทางในการส่งข้อมูลก็เป็นอีกปัจจัยหนึ่งที่เกี่ยวข้องกับการใช้พลังงาน ในกรณีที่เครื่องส่ง (transmitter) และเครื่องรับ (receiver) อยู่ใกล้กันแล้ว กำลังงานในการส่งออกสัญญาณจะมีค่าน้อย จากที่ได้กล่าวในตอนต้นว่ากำลังงานนี้เกิดขึ้นจากสองส่วน โดยกำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) จะมีค่าคงที่ไม่ขึ้นอยู่กักระยะทาง แต่กำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_S) จะแปรค่าตามระยะทาง ซึ่งการส่งสัญญาณระยะใกล้ กำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_S) จะมีค่าน้อยกว่ากำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) จึงทำให้ไม่สามารถลดการใช้พลังงานลงได้แม้ว่าจะใช้จำนวนบิตต่อสัญลักษณ์ (b) ต่ำในการส่งออกสัญญาณ ดังภาพที่ 13 ค่าของตัวแปรที่นำมาเขียนกราฟได้มาจาก Yu and Prasanna (2003)

กำหนดให้

ระยะห่างระหว่างเครื่องส่งและเครื่องรับมีค่าเท่ากับ 7 เมตร

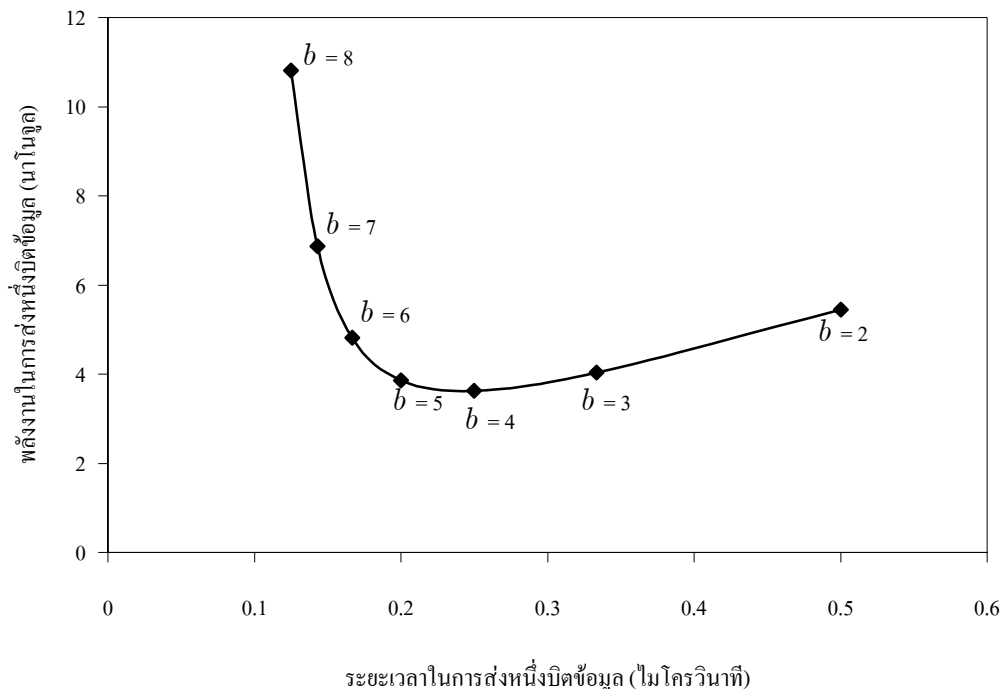
อัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (R_s) มีค่าเท่ากับ 1 MHz

กำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_s) มีค่าเท่ากับ 1 มิลลิวัตต์ (สำหรับ 4-QAM)

กำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) มีค่าเท่ากับ 10 มิลลิวัตต์

ค่าคงที่ในการส่งคลื่นวิทยุ (C_s) มีค่าเท่ากับ 3×10^{-10} (สำหรับ 4-QAM)

ค่าคงที่ในวงจรอิเล็กทรอนิกส์ (C_{ER}) มีค่าเท่ากับ 10^{-8} จูล



ภาพที่ 13 ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ที่รัศมีการส่งสั้น

จากภาพที่ 13 การส่งข้อมูลในรัศมีการส่งสั้น ถ้าเราขอมสูญเสียระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) นานขึ้นเพื่อรักษาระดับอัตราผิดพลาด (BER) พร้อมทั้งต้องการลดการปริมาณการใช้พลังงานลง แต่ผลที่ได้กลับไม่เป็นดังคาด เนื่องจากพลังงานในส่วนของวงจรอิเล็กทรอนิกส์จะต้องสูญเสียลงที่ และเมื่อเราลดพลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุโดยใช้จำนวนบิตต่อสัญลักษณ์น้อยๆ จนกระทั่งกำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_s) จะมีค่าน้อยกว่ากำลังงานในส่วนวงจรอิเล็กทรอนิกส์ (P_E) มาก ทำให้พลังงานที่ใช้ในการส่งสัญญาณเมื่อเทียบกับบิตแล้วไม่คุ้มค่า ซึ่งต่างจาก

การส่งข้อมูลระยะไกล เพราะพลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุขึ้นอยู่กับหลายปัจจัย โดยปัจจัยหนึ่งก็คือระยะทางในการส่งข้อมูล จึงทำให้พลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุยังมีค่ามาก แม้ว่าจะใช้จำนวนบิตต่อสัญลักษณ์น้อยๆ ก็ตาม ดังภาพที่ 14 โดยค่าของตัวแปรที่นำมาเขียนกราฟได้มาจาก Yu and Prasanna (2003)

กำหนดให้

ระยะทางระหว่างเครื่องส่งและเครื่องรับมีค่าเท่ากับ 32 เมตร

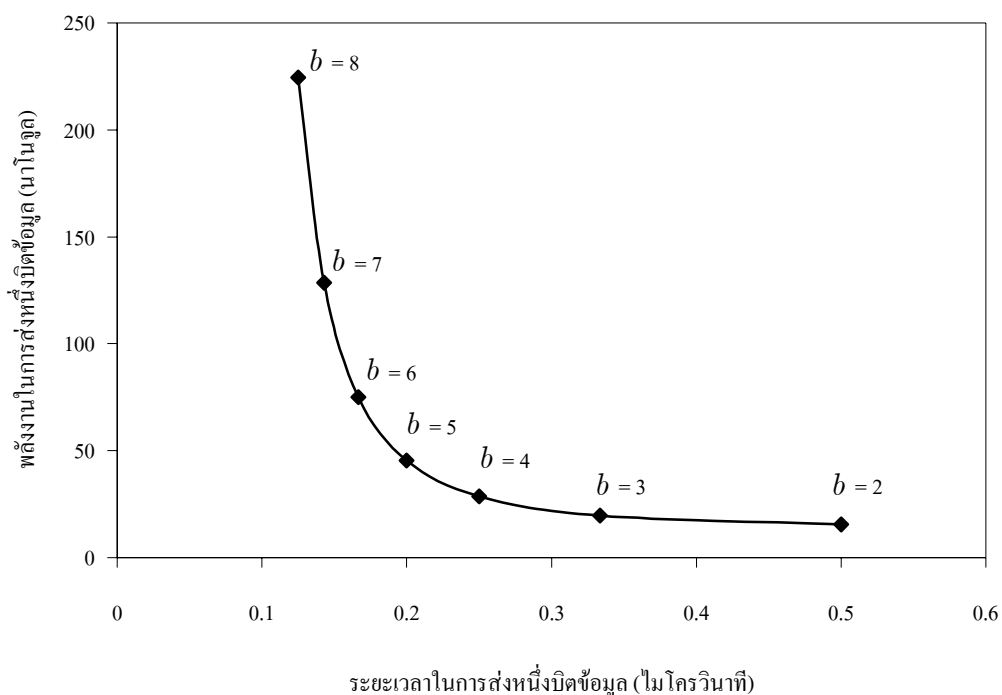
อัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (R_s) มีค่าเท่ากับ 1 MHz

กำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_s) มีค่าเท่ากับ 20 มิลลิวัตต์ (สำหรับ 4-QAM)

กำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) มีค่าเท่ากับ 10 มิลลิวัตต์

ค่าคงที่ในการส่งคลื่นวิทยุ (C_s) มีค่าเท่ากับ 7×10^{-9} (สำหรับ 4-QAM)

ค่าคงที่ในวงจรอิเล็กทรอนิกส์ (C_{ER}) มีค่าเท่ากับ 10^{-8}



ภาพที่ 14 ความสัมพันธ์ระหว่างพลังงานในการส่งหนึ่งบิตข้อมูล (E_{bit}) กับระยะเวลาในการส่งหนึ่งบิตข้อมูล (T_{bit}) เมื่อเปลี่ยนแปลงจำนวนบิตต่อหนึ่งสัญลักษณ์ (b) ที่รหัสมีการส่งไกล

วิธี Speed Scheduling of Real-time Packets

จากคุณสมบัติของวิธี DMS งานวิจัยของ Poonvasin (2003) จึงได้นำเสนอการส่งหน่วยข้อมูลคิวโอเอส (QoS packets) ด้วยวิธีการปรับอัตราบิต (bit rate) เพื่อประโยชน์ในการประหยัดพลังงาน อัตราบิตที่ใช้ในการส่งหน่วยข้อมูลคิวโอเอส (QoS packets) จะถูกปรับตามความเร่งรีบของหน่วยข้อมูลนั้น โดยคำนึงถึงจำนวนหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตาย (rejected packet) ควบคู่กับการประหยัดพลังงาน ซึ่งการคำนวณหาอัตราบิต (bit rate) ที่ใช้ส่งแต่ละหน่วยข้อมูล พิจารณาจากปริมาณงาน ในที่นี้คือจำนวนบิตของหน่วยข้อมูลใดๆ และเส้นตาย (deadline) ของหน่วยข้อมูลที่เร่งรีบที่สุด โดยความเร่งรีบที่มากที่สุดจะเป็นตัวชี้ว่าหน่วยข้อมูลนี้มีโอกาสส่งไม่ทันเส้นตายมากที่สุดเมื่อเทียบกับหน่วยข้อมูลอื่นๆ ในบัฟเฟอร์ (buffer)

หลักการทำงานของงานวิจัย Poonvasin (2003) คือสมมติให้หน่วยข้อมูลคิวโอเอส (QoS packets) จำนวน n หน่วยข้อมูล ถูกเรียงลำดับการส่งลงในบัฟเฟอร์ (buffer) ตามเส้นตาย (deadline) จากน้อยไปมาก (วิธี EDF) และนำหน่วยข้อมูลเหล่านั้นเข้าสู่การคำนวณหาอัตราบิต ($S_j(t)$) ที่สามารถประหยัดพลังงาน พร้อมทั้งคำนึงถึงปริมาณหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตาย (rejected packet) อัตราบิตดังกล่าวนี้อัตราบิตที่ต่ำที่สุดที่ใช้ในการส่งข้อมูล ณ เวลาใดๆ ซึ่งมีข้อแม้ว่า จะต้องทำให้ทุกหน่วยข้อมูลสามารถส่งได้ทันเส้นตาย ดังสมการ (35)

$$S_j(t) = \max_{1 \leq k \leq n} \left\{ \left(\frac{1}{d_k - t} \right) \sum_{j=1}^k R_j(t) \right\} \quad (35)$$

$$S_j(t) \leq S_{max} \quad (36)$$

โดย d_k คือเส้นตาย (deadline) ของหน่วยข้อมูล k

$R_j(t)$ คือจำนวนบิตของหน่วยข้อมูล j ที่ค้างในบัฟเฟอร์ ณ เวลาใดๆ

S_{max} คืออัตราบิตสูงสุดซึ่งขึ้นอยู่กับกำลังงานสูงสุดของแหล่งจ่ายพลังงาน (P_{max})

ในกรณีส่งหน่วยข้อมูลใดๆ ด้วยอัตราบิตสูงสุดแล้วยังไม่สามารถส่งได้ทันเส้นตาย ก็ให้ตัดหน่วยข้อมูลนั้นทิ้งไป (rejected packet) อัตราบิต $S_j(t)$ ที่ใช้ในการส่งแต่ละหน่วยข้อมูลไม่จำเป็นต้องมีค่าเท่ากัน และอัตราบิตนี้จำเป็นต้องคำนวณใหม่รวมทั้งปรับอัตราบิตใหม่ทุกครั้งที่มีหน่วยข้อมูลใหม่

เข้ามาหรือส่งหน่วยข้อมูลใดๆ สำเร็จ วิธีการนี้สามารถประหยัดพลังงานลงได้ เนื่องจากทุกหน่วยข้อมูล จะได้รับการส่งด้วยอัตราบิตต่ำที่สุด ($S_r(t_0)$) ที่สามารถส่งได้ทันเส้นตาย

งานวิจัยที่เกี่ยวข้อง

งานวิจัยจำนวนมากที่คิดค้นมาเพื่อรองรับการใช้งานบนเครือข่ายตรวจสอบไร้สาย (Wireless Sensor Networks) และเกือบทั้งหมดของงานวิจัยเหล่านี้จะให้ความสนใจไปที่การประหยัดพลังงาน งานวิจัยชิ้นนี้เป็นการส่งข้อมูลโดยเน้นการประหยัดพลังงานรวมทั้งรักษาคุณภาพของข้อมูล ซึ่งข้อมูลนี้มีทั้งหน่วยข้อมูลคิวโอเอส (QoS packets) และหน่วยข้อมูลนอนคิวโอเอส (Non-Quality-of-Service packets, NQoS packets) ดังนั้นความแตกต่างกันของข้อมูลจึงจำเป็นต้องมีกระบวนการจัดลำดับเพื่อความยุติธรรม ซึ่งได้มีผู้ทำวิจัยเกี่ยวกับการจัดลำดับข้อมูลทั้งสองประเภทนี้ คือ Chipalkatti *et al.* (1989) โดยกระบวนการจัดลำดับสามารถให้สิทธิพิเศษ (priority) กับหน่วยข้อมูลประเภทใดประเภทหนึ่งได้ โดย จะให้สิทธิพิเศษ (priority) กับหน่วยข้อมูลคิวโอเอสในกรณีที่หน่วยข้อมูลประเภทนี้มีปริมาณหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตาย (rejected packet) เกินกว่าค่าที่กำหนด (threshold) และจะให้สิทธิพิเศษ (priority) กับหน่วยข้อมูลนอนคิวโอเอสในกรณีที่หน่วยข้อมูลประเภทนี้ค้างค้ำอยู่ในบัฟเฟอร์เกินกว่าค่าที่กำหนด (threshold) แต่ค่าที่กำหนด (threshold) ทั้งในกรณีที่ให้สิทธิพิเศษ (priority) กับหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส ต่างก็เป็นค่าที่ผู้ทดลองคิดขึ้น งานวิจัยชิ้นนี้จึงมีข้อดีที่แตกต่างจากงานวิจัยของ Chipalkatti *et al.* (1989) ดังนี้

1. วิธีการจัดลำดับในงานวิจัยของ Chipalkatti *et al.* (1989) ไม่ได้มุ่งเน้นเพื่อรองรับวิธีการประหยัดพลังงาน
2. การจัดลำดับในงานวิจัยนี้เป็นการจัดลำดับที่ให้สิทธิพิเศษ (priority) ขึ้นอยู่กับหน่วยข้อมูล เช่นข้อมูลใดมาถึงก่อนก็จะได้รับการส่งก่อน หรือข้อมูลใดมีความเร่งรีบมากกว่าก็ได้รับการส่งก่อน เป็นต้น โดยไม่จำเป็นต้องกำหนดค่าคงที่ใดๆ (threshold)
3. การจัดลำดับการใช้ช่องสัญญาณของหน่วยข้อมูลทั้งสองประเภท ที่มีความต้องการต่างกัน จำต้องคำนึงถึงความยุติธรรม และให้ทั้งสองฝ่ายได้ใช้ช่องสัญญาณร่วมกันโดยไม่ทำให้ฝ่ายใดฝ่ายหนึ่งเกิดการเนิ่น (starving) ดังนั้นวิธีการจัดลำดับนี้จึงได้ประยุกต์ข้อดีของวิธี PGPS ซึ่งเป็นวิธีการที่ได้รับการยอมรับว่ามีความยุติธรรมรวมทั้งไม่ก่อให้เกิดการเนิ่น (starving) เพื่อให้การจัดลำดับมีประสิทธิภาพมากขึ้น

การประหยัดพลังงานบนเครือข่ายสามารถทำได้หลายรูปแบบ ไม่ว่าจะเป็นการจัดโหมดการทำงานให้เหมาะสมกับลักษณะข้อมูล (Raghunathan *et al.*, 2002 b) หรือการหาเส้นทางที่เหมาะสมเพื่อประหยัดพลังงาน (Shah and Rabaey, 2002; Younis *et al.*, 2004) เป็นต้น แต่มีเพียงงานวิจัยของ Raghunathan *et al.* (2002 b) และงานวิจัยของ Schurgers *et al.* (2001 b) ที่ใกล้เคียงกับงานวิจัยนี้คือ ใช้ DMS เพื่อช่วยในการประหยัดพลังงาน โดยงานวิจัยของ Raghunathan *et al.* (2002 b) ได้ทำการจัดลำดับหน่วยข้อมูลนอนคิวโอเอสด้วยวิธีจัดลำดับ PGPS แล้วทำการประหยัดพลังงานโดยใช้วิธี DMS ซึ่งการปรับอัตราบิต (bit rate) ในการส่งหน่วยข้อมูลจะพิจารณาจาก 2 ส่วนคือ ส่วนแรกพิจารณาจากค่าคงที่ที่ผู้ใช้เป็นผู้กำหนด โดยกำหนดตามค่าหน่วงเวลา (delay) ของหน่วยข้อมูล ซึ่งปกติเป็นค่าที่บอกว่าจะระบบจะเริ่มไม่สามารถยอมรับได้ถ้าหน่วยข้อมูลเกิดการประวิงเวลานานกว่าค่านี้ และส่วนที่สองพิจารณาจากปริมาณโหลด (load) ส่วนงานวิจัยของ Schurgers *et al.* (2001 b) ได้ทำการจัดลำดับหน่วยข้อมูลคิวโอเอสด้วยวิธีจัดลำดับ EDF แล้วทำการประหยัดพลังงานโดยใช้วิธี DMS ซึ่งการปรับอัตราบิตในการส่งหน่วยข้อมูลจะถ่วงน้ำหนัก (weight) อัตราบิตสูงสุดด้วยค่าที่เหมาะสมเพื่อช่วยลดการใช้พลังงานลง แต่ทั้ง 2 งานวิจัย ไม่ได้พิจารณากรณีที่ข้อมูลบนเครือข่ายมีทั้งหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส ดังนั้นเราจึงนำเสนองานวิจัยชิ้นนี้ซึ่งมุ่งเน้นการประหยัดพลังงานและยุติธรรมกับทั้งสองฝ่าย

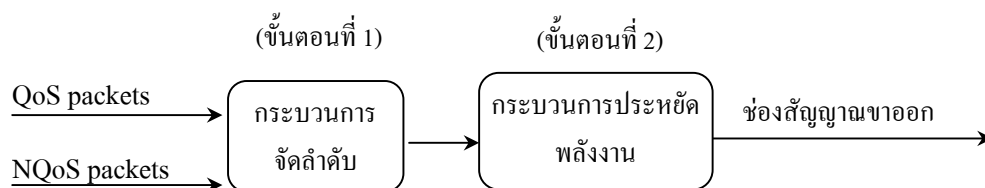
อุปกรณ์และวิธีการ

อุปกรณ์

งานวิจัยชิ้นนี้ทำการทดสอบแนวคิดด้วยวิธีทำการจำลอง (simulation) โดยใช้ภาษาซี (C language) บนเครื่องคอมพิวเตอร์ AMD Duron 1.2 GHz

วิธีการ

งานวิจัยชิ้นนี้มุ่งเน้นการประหยัดพลังงาน เพื่อยืดอายุการใช้งานของแบตเตอรี่รวมทั้งเพิ่มความสามารถในการจัดการกับข้อมูลที่มีความหลากหลาย งานวิจัยนี้จึงได้คิดค้นวิธีการจัดลำดับการส่งออกข้อมูลระหว่างหน่วยข้อมูลคิวโอเอส (QoS packets) และหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) โดยยึดความยุติธรรมเป็นหลัก รวมทั้งลดอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ย (average delay) ของหน่วยข้อมูลนอนคิวโอเอส ให้น้อยที่สุดเท่าที่เป็นไปได้ ซึ่งในส่วนการทดสอบวิธีจัดลำดับนี้จะไม่คำนึงถึงการใช้ทรัพยากรพลังงานของโหนดตรวจจับ แต่หลังจากจัดลำดับแล้วก็จะเข้าสู่กระบวนการประหยัดพลังงานด้วยการปรับค่าอัตราบิตที่เหมาะสมให้กับแต่ละหน่วยข้อมูลโดยงานวิจัยชิ้นนี้ได้เลือกใช้วิธี Dynamic Modulation Scaling (DMS) การกำหนดอัตราบิตดังกล่าวนี้จะต้องคำนึงถึงหน่วยข้อมูลคิวโอเอสในแง่อัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) และคำนึงถึงหน่วยข้อมูลนอนคิวโอเอสในแง่ความหน่วงเวลาเฉลี่ย (average delay) และยังรักษาคุณภาพของข้อมูล เช่น อัตราบิตผิดพลาด (BER) อีกด้วย



ภาพที่ 15 โครงสร้างกระบวนการทำงานในงานวิจัย

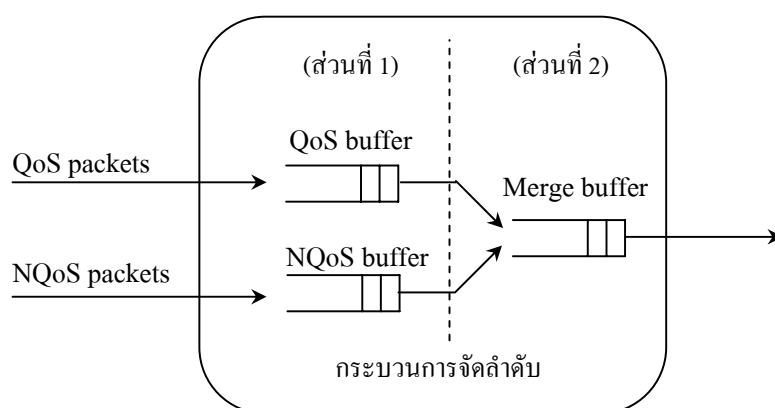
ภาพที่ 15 แสดงแบบจำลองระบบของงานวิจัยนี้ โดยเป็นส่วนที่เพิ่มเติมจากส่วนการทำงานพื้นฐานของโหนดตรวจจับ ซึ่งได้แบ่งการทำงานออกเป็น 2 ขั้นตอน โดยแต่ละขั้นตอนมีเป้าหมายดังนี้ ขั้นตอนแรกเป็นกระบวนการจัดลำดับข้อมูลเพื่อให้หน่วยข้อมูลทั้งสองประเภทได้ลำดับการส่งออกที่

ยุติธรรม และขั้นตอนที่สองเป็นกระบวนการประหยัดพลังงาน เพื่อลดการบริโภคพลังงานจากการทำงานพื้นฐานของโหนดตรวจจับ ซึ่งแต่ละขั้นตอนมีวิธีการทำงานดังนี้

ขั้นตอนที่ 1 (กระบวนการจัดลำดับ)

ขั้นตอนนี้ มีจุดประสงค์หลักอยู่ที่การจัดสรรความยุติธรรมในการจัดลำดับเพื่อการส่งออกข้อมูลระหว่างหน่วยข้อมูลสองประเภท โดยยังไม่สนใจการบริโภคพลังงาน ดังนั้นการทดสอบแนวคิดของขั้นตอนนี้จึงเป็นการส่งออกข้อมูลที่ได้จัดลำดับเรียบร้อยแล้ว โดยไม่ผ่านขั้นตอนที่สอง ซึ่งทำให้อัตราบิตที่ใช้ในการส่งข้อมูลจึงไม่สามารถปรับค่าได้ โดยอัตราบิตที่ใช้ในการส่งออกทุกหน่วยข้อมูลจะมีค่าคงที่ที่อัตราบิตสูงสุด

ในการส่งออกข้อมูลทั้งสองชนิดผ่านช่องสัญญาณเดียวกัน (outgoing channel) นั้น เป็นที่แน่นอนว่า ถ้าหน่วยข้อมูลหนึ่งได้ส่งแล้วหน่วยข้อมูลที่เหลือก็ต้องคอยจนกระทั่งช่องสัญญาณว่างจึงมีโอกาสได้ส่ง ความจริงนี้ทำให้เราไม่สามารถลดอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส (QoS packets) และลดค่าความหน่วงเวลาเฉลี่ย (average delay) ของหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) ลงได้ในเวลาเดียวกัน ไม่ว่าจะใช้แนวคิดใดๆ มาช่วยในการจัดลำดับก็ตาม



ภาพที่ 16 โครงสร้างกระบวนการจัดลำดับในงานวิจัย

เพื่อการศึกษาปัญหาดังกล่าว กระบวนการจัดลำดับในภาพที่ 16 จึงได้ถูกกำหนดโครงสร้างเพื่อจัดการกับข้อมูลทั้งสองประเภท โดยกระบวนการจัดลำดับสามารถแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกเป็นการจัดลำดับในแต่ละประเภท นั่นคือหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) ถูกจัดลำดับลงใน

บัฟเฟอร์นอนคิวโอเอส (NQoS buffer) และหน่วยข้อมูลคิวโอเอส (QoS packets) ถูกจัดลำดับลงในบัฟเฟอร์คิวโอเอส (QoS buffer) ซึ่งการจัดลำดับของทั้งสองบัฟเฟอร์ไม่เกี่ยวข้องกัน และส่วนที่สองเป็นการจัดลำดับร่วมกันระหว่างหน่วยข้อมูลในสองบัฟเฟอร์แรก โดยการเลือกหน่วยข้อมูลที่อยู่ในลำดับแรกของคิวโอเอสและนอนคิวโอเอสบัฟเฟอร์มาจัดลำดับลงบัฟเฟอร์รวม (Merge buffer) การจัดลำดับหน่วยข้อมูลของทั้งสามบัฟเฟอร์ไม่จำเป็นต้องใช้วิธีการจัดลำดับแบบเดียวกัน

งานวิจัยชิ้นนี้นำเสนอการเปรียบเทียบวิธีการจัดลำดับระหว่างหน่วยข้อมูลนอนคิวโอเอส และหน่วยข้อมูลคิวโอเอส ไว้ทั้งหมด 3 วิธี ได้แก่ **F**irst **C**ome **F**irst **S**erve for all packets (FCFS), **F**irst come first serve for NQoS packets and **E**arliest deadline first for QoS packets and **F**irst come first serve for all packets (FEF) และ **F**irst come first serve for NQoS packets and **E**arliest deadline first for QoS packets and **P**acket-by-packet generalize processor sharing for all packets (FEP) ดังแสดงในตารางที่ 4

ตารางที่ 4 วิธีการจัดลำดับภายในแต่ละบัฟเฟอร์ของวิธี FCFS FEF และ FEP

METHOD	SCHEDULING METHOD		
	NQoS buffer	QoS buffer	Merge buffer
FCFS	FCFS	FCFS	FCFS
FEF	<i>FCFS</i>	<i>EDF</i>	<i>FCFS</i>
FEP	<i>FCFS</i>	<i>EDF</i>	<i>PGPS</i>

วิธี FCFS

การจัดลำดับหน่วยข้อมูลของทั้งสามบัฟเฟอร์ ใช้วิธี FCFS โดยเรียงลำดับตามเวลาที่หน่วยข้อมูลนั้นมาถึง (arrival time) ดังนั้นลำดับของหน่วยข้อมูลในบัฟเฟอร์รวม (merge buffer) จะเรียงตามเวลาที่ข้อมูลนั้นมาถึงโหนด หรือก็คือทุกหน่วยข้อมูลที่เข้ามายังโหนดจะเข้าแถวรอคิวเรื่อยๆ ตามลำดับการมาถึงนั่นเอง

วิธี FEF

วิธีนี้เป็นการรวมจุดเด่นของวิธี EDF ในการจัดลำดับข้อมูลคิวโอเอส และวิธี FCFS เข้าด้วยกัน โดยการจัดลำดับหน่วยข้อมูลนอนคิวโอเอสในบัฟเฟอร์นอนคิวโอเอส (NQoS buffer) ใช้วิธี FCFS ดังนั้นลำดับหน่วยข้อมูลนอนคิวโอเอสในบัฟเฟอร์นอนคิวโอเอสจะเรียงตามเวลาที่หน่วยข้อมูลนอนคิวโอเอสนั้นมาถึงโหนด ส่วนการจัดลำดับหน่วยข้อมูลคิวโอเอสในบัฟเฟอร์คิวโอเอส (QoS buffer) ใช้วิธี EDF ดังนั้นลำดับหน่วยข้อมูลคิวโอเอสในบัฟเฟอร์คิวโอเอสจะเรียงตามเส้นตาย (deadline) และสำหรับการจัดลำดับหน่วยข้อมูลทั้งสองลงในบัฟเฟอร์รวม (merge buffer) จะเป็นการเลือกหน่วยข้อมูลระหว่างหน่วยข้อมูลนอนคิวโอเอสในลำดับแรกของบัฟเฟอร์นอนคิวโอเอสและหน่วยข้อมูลคิวโอเอสในลำดับแรกของบัฟเฟอร์คิวโอเอส โดยเลือกว่าระหว่างสองหน่วยข้อมูลนี้หน่วยข้อมูลใดมีเวลาที่มาถึงโหนดนี้ก่อน

อย่างไรก็ตามการจัดลำดับหน่วยข้อมูลคิวโอเอสในบัฟเฟอร์คิวโอเอสด้วยการเรียงตามเส้นตายนั้น ทำให้ลำดับหน่วยข้อมูลในบัฟเฟอร์มีค่าเวลาที่หน่วยข้อมูลคิวโอเอสมาถึงโหนด (arrival time) ไม่ได้เรียงจากน้อยไปมาก ดังนั้นการเลือกหน่วยข้อมูลจากหัวของแต่ละบัฟเฟอร์ลงบัฟเฟอร์รวม (merge buffer) ด้วยวิธี FCFS ส่วนใหญ่แล้วหน่วยข้อมูลนอนคิวโอเอสจะมีค่าเวลาในการมาถึงโหนด (arrival time) น้อยกว่าหน่วยข้อมูลคิวโอเอส ดังนั้นหน่วยข้อมูลนอนคิวโอเอสจึงมักได้รับการเลือกเข้าสู่บัฟเฟอร์รวมก่อน ด้วยเหตุผลนี้หน่วยข้อมูลคิวโอเอสจึงมักส่งไม่ทันเส้นตาย เราจึงเพิ่มกระบวนการอีกเล็กน้อย โดยในกระบวนการจัดลำดับในบัฟเฟอร์คิวโอเอส ทุกๆ หน่วยข้อมูลคิวโอเอสที่ทำการแทรกหน้าหน่วยข้อมูลคิวโอเอสที่ได้รับการจัดลำดับเรียบร้อยแล้ว จะถูกปรับค่าเวลาที่หน่วยข้อมูลนั้นมาถึงโหนด (arrival time) ของหน่วยข้อมูลคิวโอเอสที่แทรกให้มีความเท่ากับหน่วยข้อมูลคิวโอเอสที่ถูกแทรก เพื่อให้ง่ายต่อการเข้าใจวิธี FEF จึงแสดงตัวอย่างเพิ่มเติมในส่วนของภาคผนวก

วิธี FEP

วิธีจัดลำดับ FEP จะคล้ายกับวิธี FEF แต่แตกต่างกันที่ใช้วิธี PGPS ในการจัดลำดับหน่วยข้อมูลในบัฟเฟอร์รวม (merge buffer) วิธี PGPS สามารถแก้ไขความไม่ยุติธรรมจากการปรับค่าเวลาการมาถึง (arrival time) ของหน่วยข้อมูลคิวโอเอส (QoS packets) ที่เกิดขึ้นกับการจัดลำดับในบัฟเฟอร์รวม (merge buffer) ของวิธี FEF ในงานวิจัยนี้เรากำหนดให้หน่วยข้อมูลคิวโอเอส (QoS packets) และหน่วยข้อมูลนอนคิวโอเอส (NQoS packets) ได้รับแบนด์วิธ (bandwidth) ในการส่งที่เท่ากันตลอดเวลา ($\phi_{QoS} = \phi_{NQoS}$) เพื่อให้โอกาสในการส่งของทั้งสองฝ่ายเท่ากัน การทดสอบแนวคิดเป็นการจำลองการส่งแบบ on-line ดังนั้นการจัดลำดับด้วยวิธี PGPS ระหว่างหน่วยข้อมูลสองชนิด อาจทำให้เกิดเหตุการณ์ที่หน่วยข้อมูลนอนคิวโอเอสแทรกหน้าหน่วยข้อมูลคิวโอเอสที่ถูกจัดลำดับในบัฟเฟอร์รวมแล้วได้ หรือ

ในทางกลับกันอาจทำให้เกิดเหตุการณ์ที่หน่วยข้อมูลคิวโอเอสแทรกหน้าหน่วยข้อมูลนอนคิวโอเอสที่ถูกจัดลำดับในบัฟเฟอร์พร้อมแล้วได้ เพื่อให้ง่ายต่อการเข้าใจวิธี FEP จึงแสดงตัวอย่างไว้ในภาคผนวก

ขั้นตอนที่ 2 (กระบวนการประหยัดพลังงาน)

หลังจากจัดลำดับหน่วยข้อมูลลงบัฟเฟอร์พร้อมเรียบร้อยแล้ว ก็ทำการส่งต่อมายังขั้นตอนนี้ ซึ่งเป็นขั้นตอนในการหาอัตราบิตที่เหมาะสมสำหรับการส่งออกแต่ละหน่วยข้อมูล จุดประสงค์ของขั้นตอนนี้ก็เพื่อประหยัดพลังงานพร้อมกับลดอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ย (average delay) ของหน่วยข้อมูลนอนคิวโอเอส ให้น้อยที่สุดเท่าที่เป็นไปได้ ในงานวิจัยนี้ เราทำการศึกษาไว้ 7 วิธี คือ

1. Highest transmission Speed for Non-Quality-of-service packets (HSNQ)
2. Lowest Impact for average delay of non-quality-of-service packets and rejection rate of quality-of-service packets (LI)
3. Quality-of-service traffic based (QB)
4. Reducing Rejected Packets of quality-of-service packets (RRP)
5. Lowest transmission Speed (LS)
6. Reducing Rejected Packets of quality-of-service packets with Rejected Packet Selection (RRP w/ RPS)
7. Lowest transmission Speed with Rejected Packet Selection (LS w/ RPS)

ในการหาอัตราบิตที่ใช้ในการส่งแต่ละหน่วยข้อมูลของทั้ง 7 วิธีดังกล่าวนี้ อัตราบิตที่ใช้ส่งต้องมีค่าไม่เกินอัตราบิตสูงสุด (maximum bit rate) ที่โหนดสามารถรองรับได้ และสำหรับหน่วยข้อมูลคิวโอเอสที่แม้จะส่งด้วยอัตราบิตสูงสุดแล้วก็ยังไม่สามารถส่งได้ทันเส้นตาย ก็ให้ทิ้งหน่วยข้อมูลนั้นทิ้งไป

กระบวนการจัดลำดับในขั้นตอนที่ 1 นั้น ในบัฟเฟอร์รวมจะทราบค่าว่าแต่ละหน่วยข้อมูลมาถึงโหนดที่เวลาเท่าไร และสามารถส่งสำเร็จได้ที่เวลาเท่าใด แต่เวลาที่ส่งสำเร็จที่ทราบนั้นเป็นการส่งหน่วยข้อมูลที่ได้รับการจัดลำดับแล้วด้วยอัตราบิตสูงสุด ดังนั้นในแต่ละครั้งที่หน่วยข้อมูลเข้ามาถึงโหนด หน่วยข้อมูลนั้นก็จะเข้าสู่กระบวนการจัดลำดับซึ่งเป็นการหาตำแหน่งที่อยู่ให้กับหน่วยข้อมูลนั้นในบัฟเฟอร์รวม เมื่อนำหน่วยข้อมูลใหม่ลงตำแหน่งในบัฟเฟอร์เรียบร้อยแล้ว เราก็จะทราบว่าหน่วยข้อมูลนั้นมีเวลาเหลือหรือไม่ ดังนั้นเราจึงสามารถปรับอัตราบิตที่เหมาะสมสำหรับแต่ละหน่วยข้อมูลได้

การประหยัพลังงานในงานวิจัยนี้ทำได้โดยปรับอัตราบิตในการส่งข้อมูลให้ช้าลง แต่การปรับอัตราบิตจะทำให้เกิดข้อได้เปรียบเสียเปรียบ (tradeoff) ระหว่างพลังงานในการส่งข้อมูลกับระยะเวลาในการส่งข้อมูล ดังแสดงในภาพที่ 12 ดังนั้นวิธีการประหยัพลังงานที่เหมาะสมน่าจะเป็นการลดอัตราบิตในการส่งในกรณีที่หน่วยข้อมูลนั้นสามารถคอยได้ แต่ด้วยข้อจำกัดของการส่งออกข้อมูลแบบ on-line ทำให้เราไม่สามารถคาดเดาว่าหน่วยข้อมูลที่ตามจะมีปริมาณมากน้อยเพียงใด หรือมีความเร่งรีบเพียงใด ดังนั้นการลดอัตราบิตในการส่งหน่วยข้อมูลหนึ่งๆ ย่อมส่งผลกระทบต่ออัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ย (average delay) ของหน่วยข้อมูลนอนคิวโอเอส อย่างหลีกเลี่ยงไม่ได้ อีกทั้งยังมีข้อจำกัดด้านกำลังงานภายในโหนดจึงทำให้อัตราบิตถูกจำกัดตามไปด้วย ด้วยเหตุผลดังกล่าวนี้เราจึงพิจารณาวิธีการซึ่งเห็นว่าคุ้มค่าระหว่างข้อได้เปรียบเสียเปรียบ (tradeoff) ของการลดอัตราบิตเพื่อประหยัพลังงาน กับผลกระทบแง่เวลาที่เกิดขึ้นกับหน่วยข้อมูลทั้งสองประเภท (ในงานวิจัยนี้ ผลกระทบที่เกิดขึ้นกับหน่วยข้อมูลแบ่งออกเป็น 2 อย่าง ได้แก่ สำหรับหน่วยข้อมูลคิวโอเอส ผลกระทบนี้คืออัตราการเกิดหน่วยข้อมูลเสียที่เพิ่มขึ้น และสำหรับหน่วยข้อมูลนอนคิวโอเอส ผลกระทบนี้คือค่าความหน่วงเวลาที่เพิ่มขึ้น)

หน่วยข้อมูลคิวโอเอสเป็นหน่วยข้อมูลที่ถูกจำกัดด้วยเส้นตาย จึงทำให้เราหาอัตราบิตที่เหมาะสมได้ นั่นคือทำให้หน่วยข้อมูลคิวโอเอสเหล่านั้นส่งได้ทันเส้นตาย แต่สำหรับหน่วยข้อมูลนอนคิวโอเอสนั้น เนื่องจากเป็นหน่วยข้อมูลที่ไม่ถูกจำกัดด้วยเส้นตาย ทำให้เราไม่ทราบว่าควรจะอัตราบิตเท่าไรจึงจะเหมาะสม เราจึงแบ่งแนวคิดออกเป็นสองประเภทใหญ่ๆ แนวคิดแรกเป็นแนวคิดในการประหยัพลังงานที่ต้องมีผลกระทบต่อค่าความหน่วงเวลาเฉลี่ยน้อยที่สุด นั่นคือให้ส่งทุกหน่วยข้อมูลนอนคิวโอเอสด้วยอัตราบิตสูงสุด เพื่อให้ผลจากการประหยัพลังงานกระทบต่อค่าความหน่วงเวลาเฉลี่ยให้น้อยที่สุด แนวคิดนี้ได้แก่วิธี HSNQ และ LI แนวคิดที่สองเป็นแนวคิดเป็นแนวคิดที่ยินยอมให้หน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดความเร็วในการส่งหน่วยข้อมูลนอนคิวโอเอส ซึ่งจะส่งผลต่อค่าความหน่วงเวลาเฉลี่ย โดยจะขึ้นอยู่กับความเร่งรีบของหน่วยข้อมูลคิวโอเอส นั่นคือหน่วยข้อมูลถูกกำกับด้วยเส้นตายที่สั้น ก็จะทำให้ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสมีค่าน้อยตามแนวคิดนี้ได้แก่วิธี QB, RRP, LS, RRP w/ RPS และ LS w/ RPS

วิธีการประหยัพลังงาน HSNQ และ วิธี LI มีความคล้ายคลึงกัน โดยทั้งสองวิธีพยายามให้หน่วยข้อมูลนอนคิวโอเอสเกิดค่าความหน่วงเวลาเฉลี่ยน้อยที่สุด พร้อมๆ กับการประหยัพลังงาน โดยการส่งหน่วยข้อมูลนอนคิวโอเอสด้วยอัตราบิตสูงสุด แต่สำหรับการส่งหน่วยข้อมูลคิวโอเอส เราสามารถปรับลดอัตราบิตลงได้ โดยอัตราบิตที่ใช้ส่งหน่วยข้อมูลคิวโอเอสจะเป็นอัตราบิตที่ต่ำที่สุด ที่ทำให้หน่วยข้อมูลคิวโอเอสเหล่านั้นส่งได้ทันเส้นตาย ดังนั้นการประหยัพลังงานของสองวิธีนี้จึงขึ้นอยู่กับ

ความเร่งรีบของหน่วยข้อมูลคิวโอเอส วิธี LI เป็นการพัฒนางานวิธี HSNQ โดยลดอัตราการเกิดหน่วยข้อมูลเสียด้วยกระบวนการคำนวณซ้ำเพื่อเพิ่มอัตราบิตให้สูงขึ้นเท่าที่จำเป็น ซึ่งจะได้กล่าวต่อไปอย่างละเอียดในทั้งสองวิธี

วิธีการประหยัดพลังงาน QB, RRP และ LS มีความคล้ายคลึงกันคือ ให้อัตราบิตที่ใช้ในการส่งหน่วยข้อมูลบนคิวโอเอสขึ้นอยู่กับอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลคิวโอเอส จึงทำให้ทั้งสามวิธีน่าจะประหยัดพลังงานลงได้มาก วิธี RRP และ LS เป็นการพัฒนางานวิธี QB โดยลดอัตราการเกิดหน่วยข้อมูลเสียด้วยกระบวนการคำนวณซ้ำเพื่อเพิ่มอัตราบิตให้สูงขึ้นเท่าที่จำเป็น ซึ่งจะได้กล่าวต่อไปอย่างละเอียด

วิธีการประหยัดพลังงานสองวิธีสุดท้ายที่จะนำเสนออีกคือ RRP w/ RPS และ LS w/ RPS ซึ่งจะคล้ายกับวิธี RRP และ LS ตามลำดับ แต่ต่างกันเพียงเล็กน้อยในกรณีที่จำเป็นต้องทิ้งหน่วยข้อมูลคิวโอเอสบางหน่วยข้อมูล วิธี RRP w/ RPS และ LS w/ RPS จะคัดเลือกหน่วยข้อมูลคิวโอเอสที่ทำให้เกิดความเร่งรีบที่สุด ซึ่งจะกล่าวต่อไป

การปรับเปลี่ยนอัตราบิตด้วยวิธี DMS เป็นการปรับเปลี่ยนที่จำนวนบิตต่อสัญลักษณ์ตามที่กล่าวในส่วนของการตรวจเอกสาร ดังนั้นเราจึงใช้จำนวนบิตต่อสัญลักษณ์เพื่อบ่งแทนอัตราบิตที่ใช้ในการส่งข้อมูล

วิธีการประหยัดพลังงาน 7 วิธี ที่จะกล่าวต่อไปเกี่ยวข้องกับสมการ เราจึงกำหนดตัวแปรเพื่อใช้ร่วมกันในทุกๆ วิธีการประหยัดพลังงาน ดังนี้

กำหนดให้

- i คือหน่วยข้อมูลที่ได้รับการจัดลำดับแล้วในบัฟเฟอร์รวม (merge buffer)
- b_i คือจำนวนบิตต่อสัญลักษณ์ ที่จะใช้ในการส่งหน่วยข้อมูล i ซึ่งแปรค่าได้ตั้งแต่ b_{min} ถึง b_{max}
โดย $b_i \in I^+$
- L_i คือจำนวนบิตของหน่วยข้อมูล i
- d_i คือเส้นตาย (deadline) ของหน่วยข้อมูล i
- s_i คือเวลาที่หน่วยข้อมูล i เริ่มส่งได้
- R_S คืออัตราสัญลักษณ์ (symbol rate)

วิธี HSNQ

วิธี HSNQ จะไม่เน้นการประหยัดพลังงานมากนัก แต่จะพยายามคงค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสให้มีค่าน้อยที่สุด จึงใช้อัตราบิตสูงสุดในการส่งหน่วยข้อมูลนอนคิวโอเอส แต่สำหรับหน่วยข้อมูลคิวโอเอส (QoS packets) ให้ส่งด้วยอัตราบิตที่ทำให้ส่งได้ทันเส้นตาย (deadline) ของหน่วยข้อมูลนั้นพอดี แต่ต้องไม่เกินกว่าอัตราบิตสูงสุด อัตราบิตที่ใช้ในการส่งแต่ละหน่วยข้อมูลดังแสดงในตารางที่ 5 และกระบวนการทำงานของวิธี HSNQ เป็นดังขั้นตอนในภาพที่ 17 และเพื่อให้เข้าใจการทำงานของวิธี HSNQ ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ตารางที่ 5 จำนวนบิตต่อสัญลักษณ์ที่ใช้ในการมอดูเลตสัญญาณสำหรับแต่ละหน่วยข้อมูล ซึ่งคำนวณด้วยวิธี HSNQ

NQOS PACKETS	QOS PACKETS
$b_i = b_{max}$	$b_i = \left\lceil \frac{1}{R_s} \cdot \frac{L_i}{d_i - s_i} \right\rceil$ โดย $b_{min} \leq b_i \leq b_{max}$

ภาพที่ 17 กระบวนการทำงานของวิธี HSNQ

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัดพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวม ให้คำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลใหม่ตามตารางที่ 5 ซึ่งถ้าเป็นหน่วยข้อมูลคิวโอเอสและไม่สามารถส่งได้ทันเส้นตาย ให้ทิ้งหน่วยข้อมูลใหม่

ขั้นที่ 3 ถ้าหน่วยข้อมูลใหม่นี้เมื่อเข้าสู่กระบวนการจัดลำดับแล้วได้ตำแหน่งในลำดับก่อนหน้าหน่วยข้อมูลใดๆ ที่ถูกจัดลำดับในบัฟเฟอร์รวมก่อนหน้านี้ที่หน่วยข้อมูลใหม่นี้เข้ามายังโหนด ให้ทำการคำนวณอัตราบิตใหม่ให้กับทุกหน่วยข้อมูลในบัฟเฟอร์รวม ที่อยู่ในลำดับตั้งแต่หน่วยข้อมูลใหม่จนกระทั่งหน่วยข้อมูลสุดท้ายที่อยู่ในบัฟเฟอร์รวม โดยใช้การคำนวณตามตารางที่ 5 และถ้ามีหน่วยข้อมูลคิวโอเอสใดไม่สามารถส่งได้ทันเส้นตาย ก็ให้ทิ้งหน่วยข้อมูลใหม่ในกรณีที่หน่วยข้อมูลใหม่เป็นหน่วยข้อมูลคิวโอเอส หรือให้ทิ้งหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตายในกรณีที่หน่วยข้อมูลใหม่เป็นหน่วยข้อมูลนอนคิวโอเอส

ขั้นที่ 4 กลับไปที่ ขั้นที่ 1

วิธี LI

กระบวนการทำงานพื้นฐานของวิธี LI จะคล้ายกับวิธี HSNQ แต่เพิ่มการคำนวณซ้ำขึ้นมาในกรณีที่คำนวณหาอัตราบิตด้วยวิธี HSNQ แล้วทำให้บางหน่วยข้อมูลคิวโอเอสไม่สามารถส่งได้ทันเส้นตาย การคำนวณใหม่นี้จะทำให้สามารถลดอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส (QoS packets) และอัตราบิตใหม่นี้เป็นอัตราบิตที่ต่ำที่สุดสำหรับแต่ละหน่วยข้อมูล ซึ่งทำให้ส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมได้

ลักษณะการคำนวณซ้ำประกอบด้วยการทำงาน 2 ส่วน โดยส่วนแรกเป็นการตรวจสอบความเป็นไปได้ในการส่งหน่วยข้อมูลได้สำเร็จทั้งหมด และเมื่อตรวจสอบว่า ณ ขณะนี้ทุกข้อมูลที่อยู่ในบัฟเฟอร์สามารถส่งออกได้ทั้งหมด ก็ให้ทำต่อในส่วนที่สองซึ่งเป็นการหาอัตราบิตที่ต่ำที่สุดที่ทำให้ส่งทุกหน่วยข้อมูลที่ผ่านมาการตรวจสอบจากส่วนแรกได้สำเร็จทั้งหมด การคำนวณซ้ำเป็นดังภาพที่ 18 และมีกระบวนการทำงานดังภาพที่ 19 และเพื่อให้เข้าใจการทำงานของวิธี LI ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ภาพที่ 18 การคำนวณหาอัตราบิตใหม่เพื่อให้ได้อัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่
ทำให้สามารถส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมได้ทัน สำหรับวิธี LI

กำหนดให้ หน่วยข้อมูล i เป็นหน่วยข้อมูลใดๆ ในบัฟเฟอร์รวมที่ผ่านการจัดลำดับอย่างสมบูรณ์แล้ว โดยเริ่มจากหน่วยข้อมูลที่ 1 จนถึงหน่วยข้อมูลที่ m ซึ่งจะประกอบด้วยหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส และกำหนดให้ Q เป็นเซตของหน่วยข้อมูลคิวโอเอสทั้งหมดที่อยู่ในบัฟเฟอร์รวม และ Q' เป็นเซตของหน่วยข้อมูลนอนคิวโอเอสทั้งหมดที่อยู่ในบัฟเฟอร์รวม

/* กระบวนการส่วนแรกนี้เป็นการตรวจสอบว่าถ้าส่งทุกหน่วยข้อมูลด้วยอัตราบิตสูงสุดแล้วจะสามารถส่งทุกหน่วยข้อมูลได้หรือไม่ ถ้าไม่ได้ก็ให้ทิ้งหน่วยข้อมูลที่มีความเร่งรีบที่สุดออกไป */

FOR ($i = 1$ to m) AND ($i \in Q$)

$$\text{IF } \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} < 0 \right) \text{ OR } \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} > b_{\max} \cdot R_S \right)$$

drop new packet

END IF

END FOR

/*จบกระบวนการส่วนแรก..... */

/* หลังจากตรวจสอบว่าสามารถส่งทุกข้อมูลได้จริง ก็เข้าสู่กระบวนการที่สอง ซึ่งเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้สามารถส่งทุกหน่วยข้อมูลได้ */

SET $q=1$ AND $r \in Q'$ AND $j \in Q$

DO

SET $k_0 = -1$

$$k_0 = \operatorname{argmax}_{q \leq k \leq m, k \in Q} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q - \left(\frac{\sum_{r=q}^k L_r}{b_{\max} \cdot R_S} \right)} \right\}$$

$$p = \frac{1}{R_S} \cdot \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q - \left(\frac{\sum_{r=q}^{k_0} L_r}{b_{\max} \cdot R_S} \right)}$$

FOR ($i = q$ to k_0)

IF (pkt i is a QoS pkt)

$$b_i = p$$

END IF

ELSE

$$b_i = b_{\max}$$

END ELSE

$$q = k_0 + 1$$

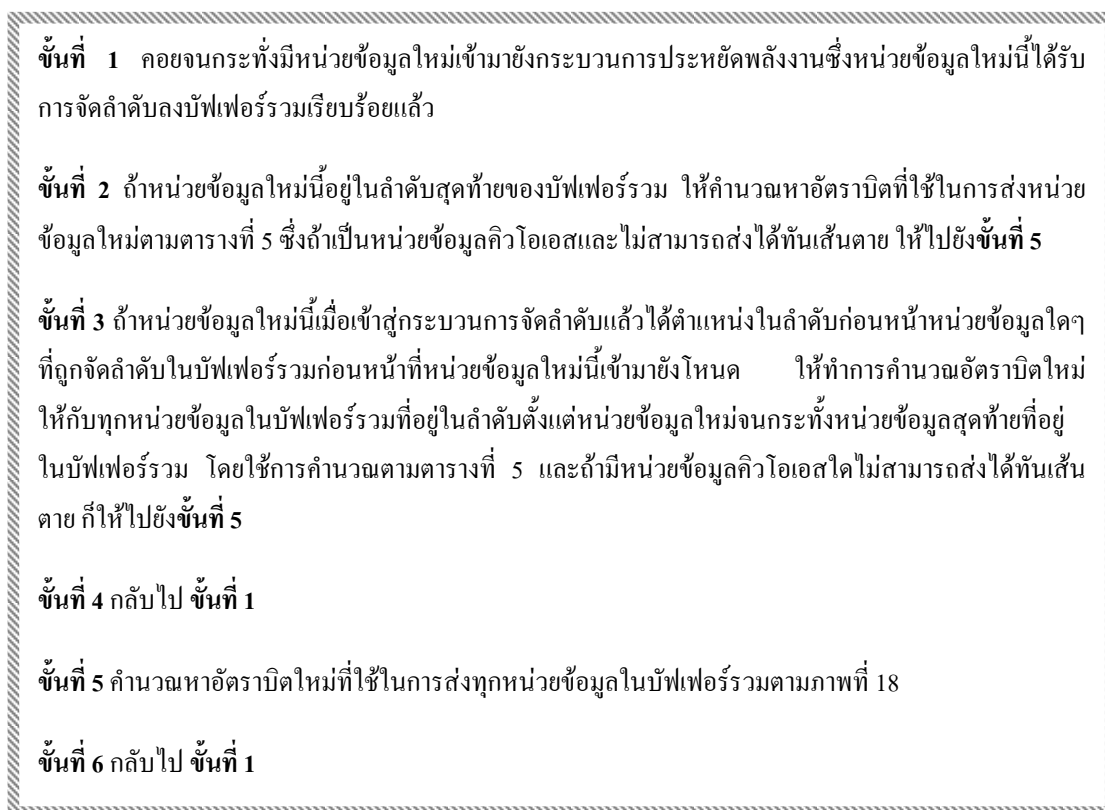
END FOR

UNTIL $[(k_0 \geq m) \text{ OR } (k_0 == 1)]$

RETURN

/*จบกระบวนการหาอัตราบิต..... */

ภาพที่ 19 กระบวนการทำงานของวิธี LI



วิธี QB

กระบวนการทำงานของวิธีนี้จะคล้ายกับวิธี HSNQ แต่แตกต่างกันที่การคำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลคิวโอเอส (NQoS packets) โดยอัตราบิตนี้ได้จากอัตราบิตเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยข้อมูล ที่อยู่ในลำดับก่อนหน้าหน่วยข้อมูลคิวโอเอสนี้ ซึ่ง 10 หน่วยข้อมูลนี้จะรวมทั้งหน่วยข้อมูลที่สามารถส่งได้และหน่วยข้อมูลที่ถูกทิ้ง โดยในกรณีที่คำนวณการส่งหน่วยข้อมูลคิวโอเอสใดๆ ด้วยอัตราบิตสูงสุดแล้วไม่สามารถส่งหน่วยข้อมูลนั้น เราจะเก็บค่าอัตราบิตของหน่วยข้อมูลคิวโอเอสนั้นไว้ที่อัตราบิตสูงสุด และทิ้งหน่วยข้อมูลนั้นไป ดังตารางที่ 6 และมีกระบวนการทำงานดังภาพที่ 20 และเพื่อให้เข้าใจการทำงานของวิธี QB ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ตารางที่ 6 จำนวนบิตต่อสัญลักษณ์ที่ใช้ในการมอดูเลตสัญญาณสำหรับแต่ละหน่วยข้อมูลซึ่งคำนวณด้วยวิธี QB

NQOS PACKETS	QOS PACKETS
$b_i = \left\lceil \frac{\sum_{j=i-x}^{i-1} b_j^*}{10} \right\rceil$ โดย $b_j^* \in \{\text{QoS packets}\}$ x คือหน่วยข้อมูลที่ i ย้อนหลังไป x หน่วย จะมีหน่วยข้อมูลคิวโอเอสทั้งหมด 10 หน่วย	$b_i = \left\lceil \frac{1}{R_s} \cdot \frac{L_i}{d_i - s_i} \right\rceil$ โดย $b_{\min} \leq b_i \leq b_{\max}$

ภาพที่ 20 กระบวนการทำงานของวิธี QB

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัดพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวม ให้คำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลใหม่ตามตารางที่ 6 ซึ่งถ้าเป็นหน่วยข้อมูลคิวโอเอสและไม่สามารถส่งได้ทันเส้นตาย ให้ทิ้งหน่วยข้อมูลใหม่

ขั้นที่ 3 ถ้าหน่วยข้อมูลใหม่นี้เมื่อเข้าสู่กระบวนการจัดลำดับแล้วได้ตำแหน่งในลำดับก่อนหน้าหน่วยข้อมูลใดๆ ที่ถูกจัดลำดับในบัฟเฟอร์รวมก่อนหน้านี้ที่หน่วยข้อมูลใหม่นี้เข้ามายังไหน ให้ทำการคำนวณอัตราบิตใหม่ให้กับทุกหน่วยข้อมูลในบัฟเฟอร์รวม ที่อยู่ในลำดับตั้งแต่หน่วยข้อมูลใหม่จนกระทั่งหน่วยข้อมูลสุดท้ายที่อยู่ในบัฟเฟอร์รวม โดยใช้การคำนวณตามตารางที่ 6 และถ้ามีหน่วยข้อมูลคิวโอเอสใดไม่สามารถส่งได้ทันเส้นตาย ก็ให้ทิ้งหน่วยข้อมูลใหม่ในกรณีที่หน่วยข้อมูลใหม่เป็นหน่วยข้อมูลคิวโอเอส หรือให้ทิ้งหน่วยข้อมูลที่ไม่สามารถส่งได้ทันเส้นตายในกรณีที่หน่วยข้อมูลใหม่เป็นหน่วยข้อมูลนอนคิวโอเอส

ขั้นที่ 4 กลับไปที่ **ขั้นที่ 1**

วิธี RRP

กระบวนการทำงานพื้นฐานของวิธีนี้จะคล้ายกับวิธี QB แต่เพิ่มการคำนวณซ้ำขึ้นมาในกรณีที่คำนวณหาอัตราบิตด้วยวิธี QB แล้วทำให้บางหน่วยข้อมูลคิวโอเอสไม่สามารถส่งได้ทันเส้นตาย การคำนวณใหม่นี้จะทำให้สามารถลดอัตราการเกิดหน่วยข้อมูลเสีย (rejection rate) ของหน่วยข้อมูลคิวโอเอส (QoS packets) และอัตราบิตใหม่นี้เป็นอัตราบิตที่ต่ำที่สุดสำหรับแต่ละหน่วยข้อมูล ซึ่งทำให้ส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมได้ ลักษณะการคำนวณซ้ำประกอบด้วยการทำงาน 2 ส่วน โดยส่วนแรกเป็นการตรวจสอบความเป็นไปได้ในการส่งหน่วยข้อมูลได้สำเร็จทั้งหมด และเมื่อตรวจสอบว่า ณ ขณะนี้ทุกข้อมูลที่อยู่ในบัฟเฟอร์สามารถส่งออกได้ทั้งหมด ก็ให้ทำต่อในส่วนที่สองซึ่งเป็นการหาอัตราบิตที่ต่ำที่สุดที่ทำให้ส่งทุกหน่วยข้อมูลผ่านการตรวจสอบจากส่วนแรกได้สำเร็จทั้งหมด การคำนวณซ้ำเป็นดังภาพที่ 21 และมีกระบวนการทำงานดังภาพที่ 22 และเพื่อให้เข้าใจการทำงานของวิธี RRP ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ภาพที่ 21 การคำนวณหาอัตราบิดใหม่เพื่อให้ได้อัตราบิดที่ต่ำที่สุดที่สามารถส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมได้ทัน สำหรับวิธี RRP และวิธี LS

กำหนดให้ หน่วยข้อมูล i เป็นหน่วยข้อมูลใดๆ ในบัฟเฟอร์รวมที่ผ่านการจัดลำดับอย่างสมบูรณ์แล้ว โดยเริ่มจากหน่วยข้อมูลที่ 1 จนถึงหน่วยข้อมูลที่ m ซึ่งจะประกอบด้วยหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส และกำหนดให้ Q เป็นเซตของหน่วยข้อมูลคิวโอเอสทั้งหมดที่อยู่ในบัฟเฟอร์รวม

/* กระบวนการส่วนแรกนี้เป็นการตรวจสอบว่าถ้าส่งทุกหน่วยข้อมูลด้วยอัตราบิดสูงสุดแล้วจะสามารถส่งทุกหน่วยข้อมูลได้หรือไม่ ถ้าไม่ได้ก็ให้ทั้งหน่วยข้อมูลที่มีปัญหาออกไป */

FOR ($i = 1$ to m) AND ($i \in Q$)

$$\text{IF } \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} < 0 \right) \text{ OR } \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} > b_{\max} \cdot R_S \right)$$

IF (new packet == QoS packet)

drop new packet

RETURN

ELSE

drop packet i

END IF

END IF

END FOR

/*จบกระบวนการส่วนแรก..... */

```

/*      หลังจากตรวจสอบว่าสามารถส่งทุกข้อมูลได้จริง ก็เข้าสู่กระบวนการที่สอง ซึ่งเป็นการหาอัตราบิตที่ต่ำ
        ที่สุดที่ทำให้สามารถส่งทุกหน่วยข้อมูลได้ */

SET  $q=1$ 
DO
    SET  $k_0 = -1$ 

    
$$k_0 = \operatorname{argmax}_{q \leq k \leq m, k \in Q} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}$$


    
$$p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_S} \right\rceil$$


    FOR ( $i = q$  to  $k_0$ )
         $b_i = p$ 
         $q = k_0 + 1$ 
    END FOR

    UNTIL  $[(k_0 \geq m) \text{ OR } (k_0 == 1)]$ 

    RETURN

/* .....จบกระบวนการหาอัตราบิต..... */

```

ภาพที่ 22 กระบวนการทำงานของวิธี RRP

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัคพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวม ให้คำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลใหม่ตามตารางที่ 6 ซึ่งถ้าเป็นหน่วยข้อมูลคิวโอเอสและไม่สามารถส่งได้ทันเส้นตาย ให้ไปยังขั้นที่ 5

ขั้นที่ 3 ถ้าหน่วยข้อมูลใหม่นี้เมื่อเข้าสู่กระบวนการจัดลำดับแล้วได้ตำแหน่งในลำดับก่อนหน้าหน่วยข้อมูลใดๆ ที่ถูกจัดลำดับในบัฟเฟอร์รวมก่อนหน้านี้ที่หน่วยข้อมูลใหม่นี้เข้ามายังไหนค ให้ทำการคำนวณอัตราบิตใหม่ให้กับทุกหน่วยข้อมูลในบัฟเฟอร์รวมที่อยู่ในลำดับตั้งแต่หน่วยข้อมูลใหม่จนกระทั่งหน่วยข้อมูลสุดท้ายที่อยู่ในบัฟเฟอร์รวม โดยใช้การคำนวณตามตารางที่ 6 และถ้ามีหน่วยข้อมูลคิวโอเอสใดไม่สามารถส่งได้ทันเส้นตาย ก็ให้ไปยังขั้นที่ 5

ขั้นที่ 4 กลับไป ขั้นที่ 1

ขั้นที่ 5 คำนวณหาอัตราบิตใหม่ที่ใช้ในการส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมตามภาพที่ 21

ขั้นที่ 6 กลับไป ขั้นที่ 1

วิธี LS

กระบวนการทำงานของวิธีนี้จะคล้ายกับวิธี RRP แต่มีความแตกต่างกันที่การคำนวณซ้ำโดยวิธีนี้จะทำการคำนวณซ้ำทุกครั้งที่มีหน่วยข้อมูลใหม่เข้ามา ยกเว้นกรณีที่หน่วยข้อมูลใหม่นี้เป็นหน่วยข้อมูลนอนคิวโอเอสที่ถูกจัดลำดับให้อยู่ในตำแหน่งสุดท้ายของบัฟเฟอร์รวม กระบวนการทำงานเป็นดังภาพที่ 23 และเพื่อให้เข้าใจการทำงานของวิธี LS ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ภาพที่ 23 กระบวนการทำงานของวิธี LS

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัคพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้เป็นหน่วยข้อมูลนอนคิวโอเอสและถูกจัดให้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวมให้คำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลใหม่นี้ตามตารางที่ 6 และหลังจากคำนวณเสร็จก็ให้กลับไปยังขั้นที่ 1

ขั้นที่ 3 คำนวณอัตราบิตใหม่ให้กับทุกๆ หน่วยข้อมูลในบัฟเฟอร์รวมตามภาพที่ 21

ขั้นที่ 4 กลับไป ขั้นที่ 1

วิธี RRP w/ RPS

วิธีนี้จะทำงานเหมือนกับ RRP เกือบทุกประการยกเว้นขั้นตอนของการเลือกว่าจะทิ้งหน่วยข้อมูลใด ซึ่งจะเกิดขึ้นในกรณีที่คำนวณการส่งทุกหน่วยข้อมูลด้วยอัตราบิตสูงสุดแล้วยังมีบางหน่วยข้อมูลคิวโอเอสในบัฟเฟอร์ไม่สามารถส่งได้ทันเส้นตาย (กระบวนการที่กล่าวนี้เกิดขึ้นในกระบวนการแรกในภาพที่ 21) การเลือกทิ้งหน่วยข้อมูลนี้ เป็นการเลือกทิ้งหน่วยข้อมูลที่เห็นว่าจะต้องใช้ความเร็วในการส่งสูงที่สุดเมื่อเทียบกับหน่วยข้อมูลอื่นๆ (ความเร็วสูงสุดนี้จะมีค่ามากกว่า maximum bit rate เสมอ) เพื่อให้ส่งได้ทันเส้นตาย ดังนั้นการคำนวณซ้ำของวิธี RRP w/ RPS จึงมีความแตกต่างการคำนวณซ้ำของวิธี RRP ดังภาพที่ 24 และมีกระบวนการทำงานดังภาพที่ 25 และเพื่อให้เข้าใจการทำงานของวิธี RRP w/ RPS ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ภาพที่ 24 การคำนวณหาอัตราบิตใหม่เพื่อให้ได้อัตราบิตที่ต่ำที่สุดที่สามารถส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมได้ทัน สำหรับวิธี RRP w/ RPS และวิธี LS w/ RPS

กำหนดให้ หน่วยข้อมูล i เป็นหน่วยข้อมูลใดๆ ในบัฟเฟอร์รวมที่ผ่านการจัดลำดับอย่างสมบูรณ์แล้ว โดยเริ่มจากหน่วยข้อมูลที่ 1 จนถึงหน่วยข้อมูลที่ m ซึ่งจะประกอบด้วยหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส และกำหนดให้ Q เป็นเซตของหน่วยข้อมูลคิวโอเอสทั้งหมดที่อยู่ในบัฟเฟอร์รวม

/* กระบวนการส่วนแรกนี้เป็นการตรวจสอบว่าถ้าส่งทุกหน่วยข้อมูลด้วยอัตราบิตสูงสุดแล้วจะสามารถส่งทุกหน่วยข้อมูลได้หรือไม่ ถ้าไม่ได้ก็ให้ทิ้งหน่วยข้อมูลที่มีความเร่งรีบที่สุดออกไป */

FOR ($i = 1$ to m) AND ($i \in Q$)

$$\text{IF} \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} < 0 \right) \text{OR} \left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} > b_{\max} \cdot R_s \right)$$

drop packet i

END IF

END FOR

/*จบกระบวนการส่วนแรก..... */

```

/*      หลังจากตรวจสอบว่าสามารถส่งทุกข้อมูลได้จริง ก็เข้าสู่กระบวนการที่สอง ซึ่งเป็นการหาอัตราบิตที่ต่ำ
        ที่สุดที่ทำให้สามารถส่งทุกหน่วยข้อมูลได้ */

SET  $q=1$ 
DO
    SET  $k_0 = -1$ 
    
$$k_0 = \operatorname{argmax}_{q \leq k \leq m, k \in Q} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}$$

    
$$p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_s} \right\rceil$$

    FOR ( $i = q$  to  $k_0$ )
         $b_i = p$ 
         $q = k_0 + 1$ 
    END FOR
UNTIL  $[(k_0 \geq m) \text{ OR } (k_0 = 1)]$ 
RETURN

/* .....จบกระบวนการหาอัตราบิต..... */

```

ภาพที่ 25 กระบวนการทำงานของวิธี RRP w/ RPS

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัคพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวม ให้คำนวณหาอัตราบิดที่ใช้ในการส่งหน่วยข้อมูลใหม่ตามตารางที่ 6 ซึ่งถ้าเป็นหน่วยข้อมูลคิวโอเอสและไม่สามารถส่งได้ทันเส้นตาย ให้ไปยังขั้นที่ 5

ขั้นที่ 3 ถ้าหน่วยข้อมูลใหม่นี้เมื่อเข้าสู่กระบวนการจัดลำดับแล้วได้ตำแหน่งในลำดับก่อนหน้าหน่วยข้อมูลใดๆ ที่ถูกจัดลำดับในบัฟเฟอร์รวมก่อนหน้านี้ที่หน่วยข้อมูลใหม่นี้เข้ามายังโหนด ให้ทำการคำนวณอัตราบิดใหม่ให้กับทุกหน่วยข้อมูลในบัฟเฟอร์รวมที่อยู่ในลำดับตั้งแต่หน่วยข้อมูลใหม่จนกระทั่งหน่วยข้อมูลสุดท้ายที่อยู่ในบัฟเฟอร์รวม โดยใช้การคำนวณตามตารางที่ 6 และถ้ามีหน่วยข้อมูลคิวโอเอสใดไม่สามารถส่งได้ทันเส้นตาย ก็ให้ไปยังขั้นที่ 5

ขั้นที่ 4 กลับไป ขั้นที่ 1

ขั้นที่ 5 คำนวณหาอัตราบิดใหม่ที่ใช้ในการส่งทุกหน่วยข้อมูลในบัฟเฟอร์รวมตามภาพที่ 24

ขั้นที่ 6 กลับไป ขั้นที่ 1

LS w/ RPS

วิธีนี้จะทำงานเหมือนกับ LS เกือบทุกประการ ยกเว้นขั้นตอนของการเลือกที่จะทิ้งหน่วยข้อมูลใด ซึ่งจะเกิดขึ้นในกรณีที่คำนวณการส่งทุกหน่วยข้อมูลด้วยอัตราบิตสูงสุดแล้วยังมีบางหน่วยข้อมูลคิวโอเอสในบัฟเฟอร์ไม่สามารถส่งได้ทันเส้นตาย (กระบวนการที่กล่าวนี้เกิดขึ้นในกระบวนการแรกในภาพที่ 21) การเลือกทิ้งหน่วยข้อมูลนี้ เป็นการเลือกทิ้งหน่วยข้อมูลที่เห็นว่าจะต้องใช้ความเร็วในการส่งสูงที่สุดเมื่อเทียบกับหน่วยข้อมูลอื่นๆ (ความเร็วสูงสุดนี้จะมีค่ามากกว่า maximum bit rate เสมอ) เพื่อให้ส่งได้ทันเส้นตาย ดังนั้นการคำนวณซ้ำของวิธี LS w/ RPS จึงมีความแตกต่างการคำนวณซ้ำของวิธี LS ดังแสดงในภาพที่ 24 และกระบวนการทำงานของวิธี LS w/ SRP เป็นดังภาพที่ 26 และเพื่อให้เข้าใจการทำงานของวิธี LS w/ RPS ได้ชัดเจนขึ้น จึงได้แสดงตัวอย่างการทำงานไว้ในภาคผนวก

ภาพที่ 26 กระบวนการทำงานของวิธี LS w/ RPS

ขั้นที่ 1 คอยจนกระทั่งมีหน่วยข้อมูลใหม่เข้ามายังกระบวนการประหยัคพลังงานซึ่งหน่วยข้อมูลใหม่นี้ได้รับการจัดลำดับลงบัฟเฟอร์รวมเรียบร้อยแล้ว

ขั้นที่ 2 ถ้าหน่วยข้อมูลใหม่นี้เป็นหน่วยข้อมูลนอนคิวโอเอสและถูกจัดให้อยู่ในลำดับสุดท้ายของบัฟเฟอร์รวมให้คำนวณหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลใหม่นี้ตามตารางที่ 6 และกลับไปยัง**ขั้นที่ 1**

ขั้นที่ 3 คำนวณอัตราบิตใหม่ให้กับทุกๆ หน่วยข้อมูลในบัฟเฟอร์รวมตามภาพที่ 24

ขั้นที่ 4 กลับไป **ขั้นที่ 1**

ข้อมูลที่ใช้ในการทดสอบ

กำหนดให้อัตราบิตของช่องสัญญาณขาออก (μ) สามารถแปรค่าได้ตามหลักการ DMS ที่อัตราสัญลักษณ์ (symbol rate, R_s) คงที่ที่ 1 MHz การแปรค่าอัตราบิตขึ้นอยู่กับจำนวนบิตต่อสัญลักษณ์ (b) ในการมอดูเลต โดยจำนวนบิตต่อสัญลักษณ์แปรค่าตั้งแต่ $b_{min} = 2$ บิตต่อสัญลักษณ์ จนถึง $b_{max} = 10$ บิตต่อสัญลักษณ์ และหน่วยข้อมูลที่ต้องการส่งออกจากโหนดๆ หนึ่งมีลักษณะดังนี้

1. ชนิดของหน่วยข้อมูล

โอกาสของหน่วยข้อมูลใดๆ ที่เข้ามายังโหนด ณ เวลาหนึ่งๆ จะเป็นหน่วยข้อมูลคิวโอเอส หรือนอนคิวโอเอสมีค่าตามสมการ Bernoulli ดังนี้

$$P_X(x) = \begin{cases} p = \frac{n_{NQoS}}{n_{total}} & ; x \in \{NQoS \text{ packets}\} \\ 1-p = \frac{n_{QoS}}{n_{total}} & ; x \in \{QoS \text{ packets}\} \end{cases} \quad (37)$$

โดย n_{NQoS} คือจำนวนหน่วยข้อมูลนอนคิวโอเอสทั้งหมดที่ใช้ในการทดสอบ
 n_{QoS} คือจำนวนหน่วยข้อมูลคิวโอเอสทั้งหมดที่ใช้ในการทดสอบ
 $n_{NQoS} + n_{QoS}$ คือจำนวนหน่วยข้อมูลทั้งหมดที่ใช้ในการทดสอบ

การทดสอบในงานวิจัยนี้แสดงให้เห็นผลของวิธีการจัดลำดับและวิธีการประหยัดพลังงาน เมื่อมีการแปรสัดส่วนระหว่างปริมาณหน่วยข้อมูลนอนคิวโอเอสและปริมาณหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) โดยสัดส่วนที่ใช้ทดสอบมีค่าเท่ากับ 0.5, 1.0 และ 2.0

2. จำนวนบิตในหนึ่งหน่วยข้อมูล

ในหน่วยข้อมูลหนึ่งๆ ที่ใช้ในการทดสอบจะมีจำนวนบิตไม่เท่ากัน โดยจำนวนบิตของแต่ละหน่วยข้อมูลมีการกระจายตัวแบบ Poisson ที่จำนวนบิตเฉลี่ย (L_{av}) 400 บิต

3. ระยะเวลาห่างระหว่างการเข้ามาของแต่ละหน่วยข้อมูล (interarrival time)

ระยะเวลาห่างระหว่างหน่วยข้อมูลที่เข้ามายังโหนด มีกระจายตัวแบบ Exponential ซึ่งมีค่าเฉลี่ย (mean interarrival time) ตามสมการ (38) ซึ่ง α คืออัตราการเข้ามาของบิตข้อมูล (arrival rate, หน่วย bps)

$$\text{mean interarrival time} = \frac{L_{av}}{\mu_{max} \cdot (\alpha / \mu_{max})} \quad (38)$$

การทดสอบในงานวิจัยนี้แสดงให้เห็นผลของวิธีการจัดลำดับและวิธีการประหยัดพลังงานที่ปริมาณโหลดต่างๆ โดยปริมาณโหลดกำหนดขึ้นจากสัดส่วนของอัตราการเข้ามาของบิตข้อมูลต่ออัตราบิตสูงสุดของช่องสัญญาณขาออก (α / μ_{max}) ซึ่งให้มามีค่าเท่ากับ 0.1, 0.2, ..., 1.0 ค่าเหล่านี้แสดงถึงความคับคั่งของหน่วยข้อมูลที่ปริมาณต่างๆ จากนั้นน้อยไปมาก

4. เส้นตาย (deadline)

ระยะเวลาตั้งแต่หน่วยข้อมูลคิวโอเอสเข้ามายังโหนดจนกระทั่งถึงเส้นตายมีการกระจายตัวแบบ Exponential ที่ค่าเฉลี่ย (T)

ช่วงเวลาดังแต่หน่วยข้อมูลคิวโอเอสเข้ามายังโหนด จนกระทั่งถึงเส้นตายที่มีความเร่งรีบที่สุด โดยช่วงเวลานั้นเป็นช่วงเวลาที่ทำให้ส่งหน่วยข้อมูลคิวโอเอสที่มีจำนวนบิต L_{av} ได้ทันพอดี ช่วงเวลานี้แทนด้วย t_{min} ซึ่งมีค่าดังสมการ

$$t_{min} = \frac{L_{av}}{R_s \cdot b_{max}} \quad (39)$$

ดังนั้นถ้าช่วงเวลาดังแต่หน่วยข้อมูลคิวโอเอสที่มีจำนวนบิต L_{av} มาถึงโหนดจนกระทั่งถึงเส้นตายมีค่าน้อยกว่า t_{min} แล้วแสดงว่าเป็นไปไม่ได้เลยที่จะสามารถส่งหน่วยข้อมูลนี้ได้ทัน งานวิจัยนี้จึงกำหนดให้ช่วงเวลาดังแต่หน่วยข้อมูลคิวโอเอสมาถึงโหนดจนกระทั่งถึงเส้นตาย (T) มีค่ามากกว่า t_{min} โดยให้ช่วงเวลานี้ (T) แปรค่าตาม t_{min}/T ซึ่งมีค่าเท่ากับ 0.1, 0.2, ..., 1.0 ค่าเหล่านี้แสดงถึงความเร่งรีบของหน่วยข้อมูลคิวโอเอสที่มีความเร่งรีบจากน้อยไปมาก

ในการทดสอบ เราใช้วิธีการมอดูเลตเชิงเลขแบบเอ็มอาร์เควอดเรเจอร์ (M-ary quadrature amplitude modulation, M-ary QAM) ในการมอดูเลตบิตข้อมูลเป็นสัญญาณ โดยหนึ่งสัญญาณจะมีจำนวนบิตสูงสุด (b_{max}) ที่ 10 บิตต่อสัญลักษณ์ ซึ่งจำนวนบิตสูงสุดนี้ถูกจำกัดจากกำลังงานสูงสุด ($P_S + P_E$) ตามสมการ (28) และ (31) และทดสอบการบริโภคพลังงานโดยค่าคงที่ตามงานวิจัยของ Yu and Prasanna (2003) ซึ่งมีรายละเอียดดังนี้ ระยะห่างระหว่างเครื่องส่งและเครื่องรับมีค่าเท่ากับ 32 เมตร อัตราเร็วในการส่งข้อมูลต่อหนึ่งสัญลักษณ์ (R_S) มีค่าเท่ากับ 1 MHz กำลังงานที่ใช้ในส่วนการส่งคลื่นวิทยุ (P_S) มีค่าเท่ากับ 20 มิลลิวัตต์ กำลังงานในส่วนของวงจรอิเล็กทรอนิกส์ (P_E) มีค่าเท่ากับ 10 มิลลิวัตต์ ค่าคงที่ในการส่งคลื่นวิทยุ (C_S) มีค่าเท่ากับ 7×10^{-9} และค่าคงที่ในวงจรอิเล็กทรอนิกส์ (C_{ER}) มีค่าเท่ากับ 10^{-8}

ผลและวิจารณ์

ขั้นตอนที่ 1 (กระบวนการจัดลำดับ)

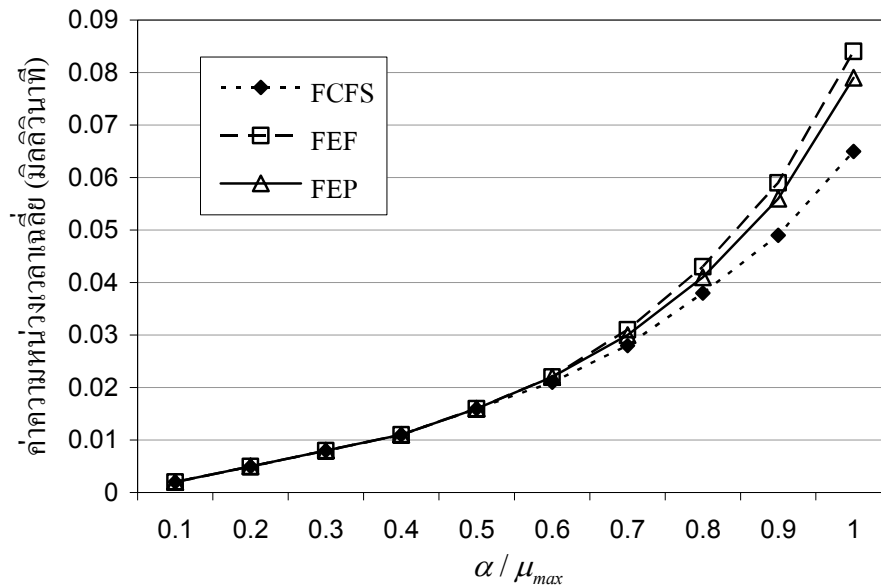
เป้าหมายสำคัญของการส่งข้อมูลในงานวิจัยชิ้นนี้ คือต้องการให้ความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส (average delay of NQoS packets) และอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส (rejection rate of QoS packets) มีค่าต่ำทั้งคู่ แต่ในความเป็นจริงสำหรับการส่งออกข้อมูลชุดเดียวกันด้วยความเร็วในการส่งข้อมูล (bit rate) ที่เท่ากันแล้ว ไม่ว่าจะใช้วิธีการใด ก็ล้วนแต่ต้องเจอกับข้อจำกัดที่ว่า เมื่อต้องการให้อัตราการเกิดหน่วยข้อมูลเสียลดลง แสดงว่าต้องให้หน่วยข้อมูลนอนคิวโอเอสบางหน่วยเลื่อนการส่งออกไป เพื่อให้โอกาสหน่วยข้อมูลคิวโอเอสได้ใช้ช่องสัญญาณก่อน ในทางกลับกัน เมื่อต้องการให้ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสต่ำลง แสดงว่าต้องให้หน่วยข้อมูลคิวโอเอสบางหน่วยเลื่อนการส่งออกไป เพื่อให้โอกาสหน่วยข้อมูลนอนคิวโอเอสได้ใช้ช่องสัญญาณก่อน แต่การทำเช่นนี้จะทำให้หน่วยข้อมูลคิวโอเอสที่ถูกเลื่อนการส่งเหล่านั้นไม่สามารถส่งข้อมูลได้ทันเส้นตาย (deadline) ส่งผลให้เกิดหน่วยข้อมูลเสียมากขึ้น

จากข้อจำกัดนี้ จึงได้พยายามหาวิธีการที่มีความยุติธรรมในการแบ่งสรรการใช้ช่องสัญญาณในการส่งหน่วยข้อมูลทั้งสองประเภท และสามารถลดอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสและลดค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสให้ได้มากที่สุดเท่าที่เป็นไปได้

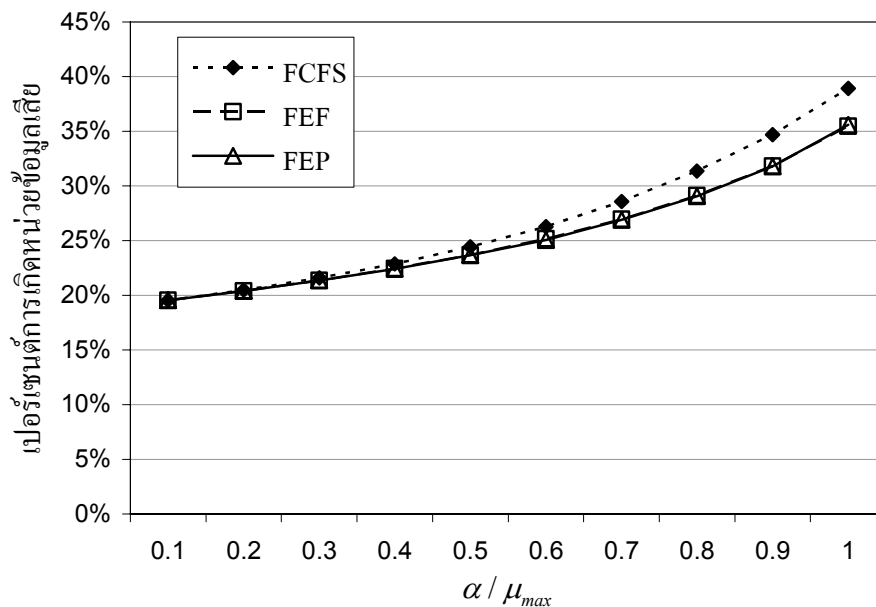
การทดสอบในขั้นตอนนี้ทำโดยไม่มีการปรับเปลี่ยนอัตราบิตแต่อย่างใด ดังนั้นการส่งหน่วยข้อมูลหนึ่งๆ ออกจากโหนดจึงใช้อัตราบิตคงที่ ที่อัตราบิตสูงสุด ซึ่งในการทดสอบนี้ให้อัตราบิตสูงสุดที่ 10 Mbps (อัตราบิตสูงสุด = $b_{min} \cdot R_s$)

การวิเคราะห์ผลในเชิงการแปรค่าปริมาณโหลด (α / μ_{max})

จากผลการทดลอง เมื่อมีปริมาณข้อมูลที่ต้องการส่งออกจากโหนดมากขึ้น (α / μ_{max} สูงขึ้น) ย่อมทำให้ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสและอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสสูงขึ้นตาม แต่เมื่อใช้วิธีการจัดลำดับที่แตกต่างกัน ก็ทำให้ผลที่ได้มีค่าต่างกันดังภาพที่ 27 และ 28 ตามลำดับ



ภาพที่ 27 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{NQoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 28 เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเทียบกับหน่วยข้อมูลคิวโอเอสทั้งหมด โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับรูป 27

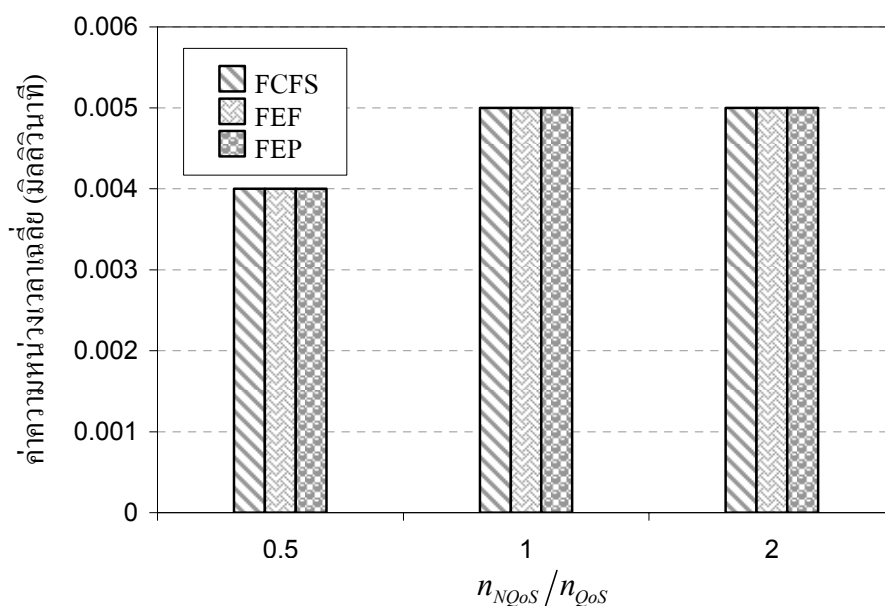
สำหรับวิธี FCFS แม้ว่าความหน่วงเวลาเฉลี่ยจะมีค่าต่ำที่สุด แต่อัตราการเกิดหน่วยข้อมูลเสียก็มีค่าสูงมาก ที่เป็นเช่นนี้เพราะหน่วยข้อมูลคิวไอเอสถูกกำกับด้วยเส้นตาย ดังนั้นการจัดลำดับในบัฟเฟอร์คิวไอเอสด้วยวิธี FCFS จึงไม่เหมาะสม โดยส่งผลให้เกิดหน่วยข้อมูลเสียจำนวนมาก และเมื่อหน่วยข้อมูลเหล่านี้ถูกตัดทิ้ง ปริมาณหน่วยข้อมูลรวมที่ส่งผ่านช่องสัญญาณจึงมีค่าน้อยลง ส่งผลให้หน่วยข้อมูลนอนคิวไอเอสได้รับช่องสัญญาณมากขึ้น ดังนั้นความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลคิวไอเอสจึงมีค่าต่ำ

จาก Towsley (1993) วิธี EDF มีความเหมาะสมในการส่งข้อมูลที่ถูกรับกำกับด้วยเส้นตาย ดังนั้นวิธี FEF จึงนำวิธี EDF มาประยุกต์ใช้เป็นการแก้ปัญหาที่เกิดขึ้นกับวิธี FCFS ซึ่งผลที่ได้ก็สามารถลดอัตราการเกิดหน่วยข้อมูลเสียลงตามที่ตั้งใจดังภาพที่ 28 แต่การจัดลำดับหน่วยข้อมูลคิวไอเอสในบัฟเฟอร์คิวไอเอสนั้น มีหน่วยข้อมูลคิวไอเอสบางหน่วยถูกแก้ไขค่าเวลาการมาถึงโหนด (arrival time) ตามที่กล่าวไว้ในส่วนอุปกรณ์และวิธีการ ผลเสียที่เกิดขึ้นจากการปรับค่าเวลาการมาถึงโหนด (arrival time) คือความไม่ยุติธรรมที่เกิดขึ้นกับหน่วยข้อมูลนอนคิวไอเอส เช่น ในกรณีมีหน่วยข้อมูลคิวไอเอสที่มีเส้นตายใกล้ได้ถูกจัดเข้าบัฟเฟอร์คิวไอเอส แล้วมีหน่วยข้อมูลคิวไอเอสที่มีเส้นตายกระชั้นชิดเข้ามาอย่างต่อเนื่อง การจัดลำดับในบัฟเฟอร์คิวไอเอสก็จะปรับค่าเวลาการมาถึงโหนด (arrival time) ของหน่วยข้อมูลที่มีเส้นตายกระชั้นให้มีค่าเท่ากับหน่วยข้อมูลคิวไอเอสที่มีเส้นตายไกล เนื่องจากเรียงตามเส้นตาย กรณีนี้ เมื่อมีหน่วยข้อมูลนอนคิวไอเอสเข้ามา หน่วยข้อมูลนอนคิวไอเอสนี้จะไม่ได้รับช่องสัญญาณในการส่งออกเลยแม้แต่หน่วยข้อมูลเดียว เนื่องจากการจัดลำดับในบัฟเฟอร์รวมเรียงตามค่าเวลาการมาถึง (arrival time) ด้วยวิธี FCFS และในกรณีนี้ค่าเวลาการมาถึง (arrival time) ของหน่วยข้อมูลคิวไอเอสที่เข้ามาอย่างต่อเนื่องจะถูกปรับให้น้อยลงกว่าหน่วยข้อมูลนอนคิวไอเอสเสมอ แม้ว่าในความเป็นจริงหน่วยข้อมูลนอนคิวไอเอสจะมาถึงโหนดก่อนหน่วยข้อมูลคิวไอเอสก็ตาม

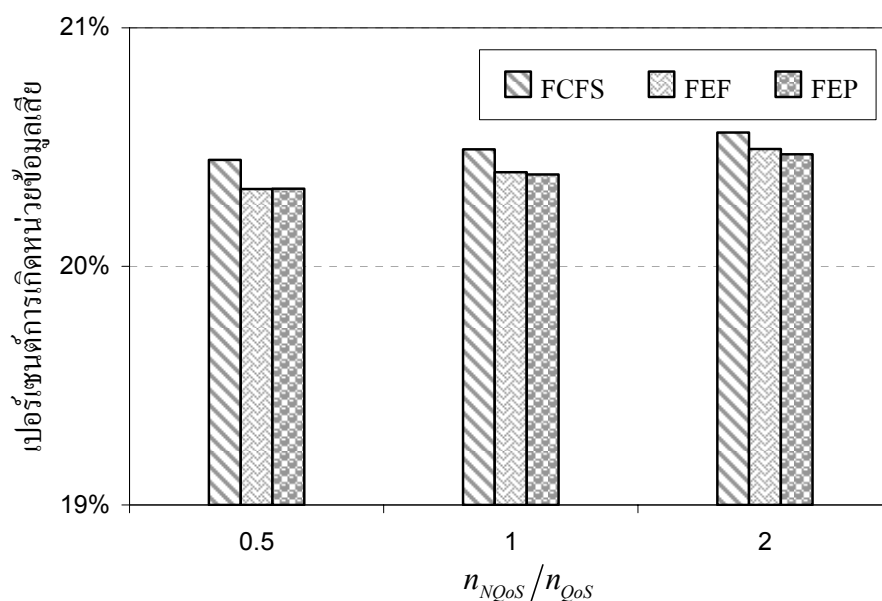
จากเหตุผลนี้ เมื่อเราใช้วิธี EDF ในการจัดลำดับในบัฟเฟอร์คิวไอเอส แล้วหน่วยข้อมูลจากสองบัฟเฟอร์มาจัดลำดับรวมด้วยวิธี FCFS แม้ว่าจะสามารถลดอัตราการเกิดหน่วยข้อมูลเสียลงจนเป็นที่น่าพอใจ แต่การปรับค่าเวลาการมาถึงโหนด (arrival time) ทำให้เกิดความไม่ยุติธรรมขึ้น ดังนั้นเราจึงนำเสนอวิธี FEP ซึ่งสามารถแก้ไขความไม่ยุติธรรมนี้โดยใช้วิธี PGPS แทนวิธี FCFS ในการจัดลำดับหน่วยข้อมูลสองชนิดลงในบัฟเฟอร์รวม ซึ่งการจัดลำดับหน่วยข้อมูลจากบัฟเฟอร์นอนคิวไอเอสและบัฟเฟอร์คิวไอเอสด้วยวิธี PGPS จะรับประกันว่าหน่วยข้อมูลจากสองบัฟเฟอร์จะได้รับช่องสัญญาณอย่างแน่นอน วิธี FEP นี้ได้กำหนดให้หน่วยข้อมูลจากสองบัฟเฟอร์ได้รับแบนด์วิธเท่าๆ กัน ($\phi_{QoS} = \phi_{N_{QoS}}$)

การวิเคราะห์ผลในเชิงการแปรค่าสัดส่วน ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อคิวโอเอส
 (n_{NQoS}/n_{QoS})

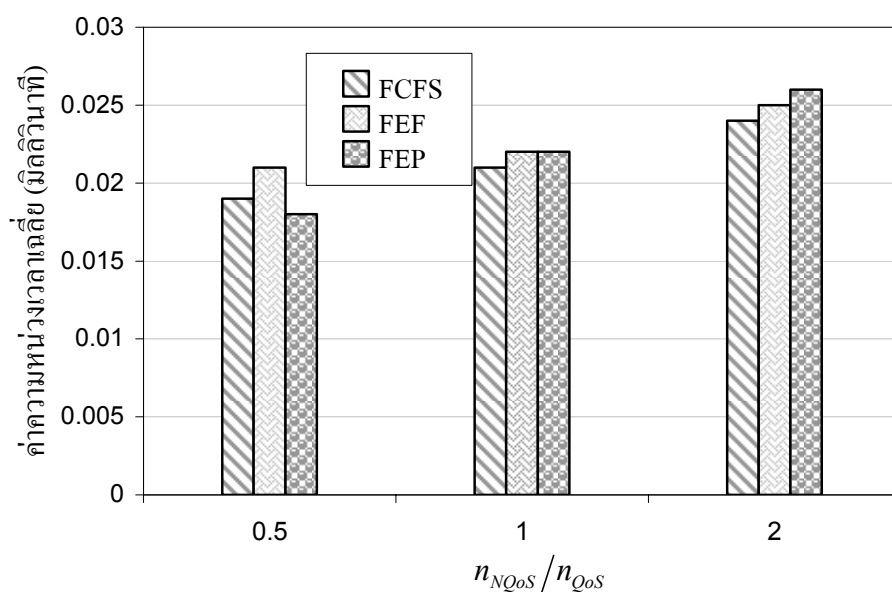
วิธี FEP มีการแบ่งแบนด์วิธด้วยวิธี GPS ให้กับหน่วยข้อมูลทั้งสองประเภทอย่างละครึ่ง จากการคาดการณ์เบื้องต้นก่อนการทดลอง ถ้าสัดส่วนระหว่างปริมาณหน่วยข้อมูลนอนคิวโอเอสและปริมาณหน่วยข้อมูลคิวโอเอสมีค่าไม่เท่ากัน อาจทำให้เกิดความไม่เหมาะสมเมื่อทำการแบ่งแบนด์วิธแบบครั้งต่อครั้ง โดยในกรณีที่ปริมาณหน่วยข้อมูลนอนคิวโอเอสมากกว่าปริมาณหน่วยข้อมูลคิวโอเอส อาจส่งผลให้ความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสมีค่ามากเมื่อเทียบกับวิธีอื่น และในทางกลับกัน ในกรณีที่ปริมาณหน่วยข้อมูลคิวโอเอสมากกว่าปริมาณหน่วยข้อมูลนอนคิวโอเอสอาจส่งผลให้อัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสมีค่ามากเมื่อเทียบกับวิธีอื่น แต่จากผลการทดลองไม่เป็นเช่นนั้น ดังภาพที่ 29, 30, 31 และ 32



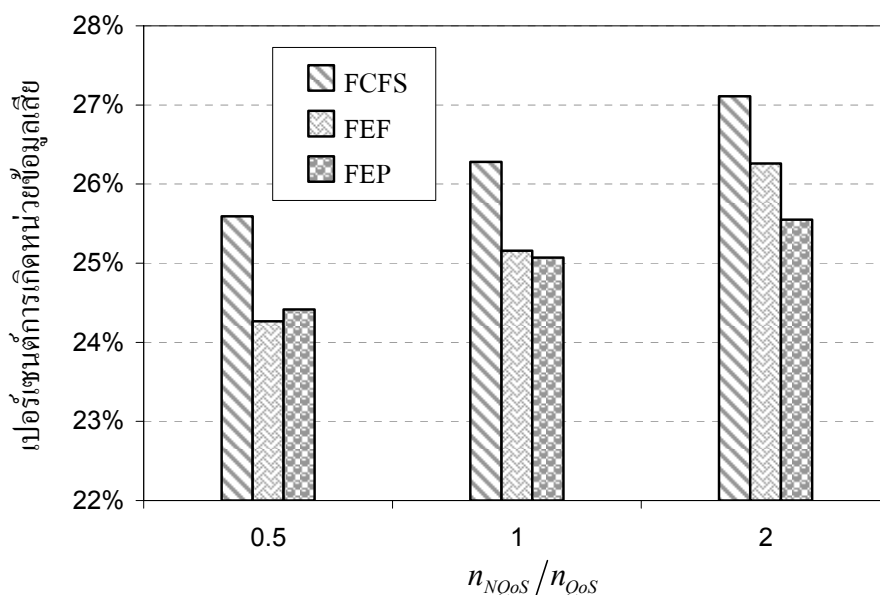
ภาพที่ 29 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.2$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 30 เปอร์เซ็นต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเทียบกับหน่วยข้อมูลคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 29



ภาพที่ 31 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.6$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 32 เปอร์เซนต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเทียบกับหน่วยข้อมูลคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 31

ในกรณีปริมาณโหลดเบาบาง ผลที่ได้จากการจัดลำดับด้วยวิธีต่างๆ มีค่าใกล้เคียงกันมาก ดังภาพที่ 29 และ 30 เนื่องจากเหตุการณ์ส่วนใหญ่ที่เกิดขึ้นคือไม่มีหน่วยข้อมูลค้างในบัฟเฟอร์ เมื่อมีหน่วยข้อมูลใดๆ เข้ามายังโหนดก็สามารถส่งออกได้ทันทีโดยไม่ต้องผ่านกระบวนการจัดลำดับ ดังนั้นการแปรเปลี่ยนสัดส่วน ระหว่างปริมาณหน่วยข้อมูลนอนคิวโอเอสต่อปริมาณหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) จึงไม่เห็นความแตกต่างระหว่างวิธีการจัดลำดับแบบต่างๆ

ในกรณีโหลดคับคั่ง การแปรสัดส่วน n_{NQoS}/n_{QoS} จะเห็นผลของการจัดลำดับอย่างชัดเจน ดังรูป 21 และ 22 ที่สัดส่วนของหน่วยข้อมูลคิวโอเอสมากกว่านอนคิวโอเอสสองเท่า ($n_{NQoS}/n_{QoS} = 0.5$) วิธี FEP ซึ่งเป็นการแบ่งแบนด์วิดท์อย่างละครึ่งด้วยวิธี GPS ให้กับหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส การที่มีปริมาณหน่วยข้อมูลคิวโอเอสจำนวนมากกว่าหน่วยข้อมูลนอนคิวโอเอสถึงสองเท่า ก็ไม่ได้ทำให้ได้ทำให้เกิดหน่วยข้อมูลเสียมากแต่อย่างใด ดังแสดงในภาพที่ 32 เนื่องจากวิธี FEP ใช้การจัดลำดับ EDF ในการจัดลำดับหน่วยข้อมูลคิวโอเอส จึงช่วยลดการเกิดหน่วยข้อมูลเสียลงได้ และการที่มีหน่วยข้อมูลเสียน้อย ก็แสดงว่ามีปริมาณหน่วยข้อมูลรวมที่สามารถส่งออกมาก ซึ่งก็น่าจะทำให้หน่วยข้อมูลนอนคิวโอเอสไม่ค่อยได้รับช่องสัญญาณ และทำให้มีค่าความหน่วงเวลาเฉลี่ยค่อนข้างมาก แต่จากภาพที่ 31 กลับไม่เป็นเช่นนั้น การที่ค่าความหน่วงเวลาเฉลี่ยมีค่าน้อยและสามารถส่งออกข้อมูล

ในปริมาณมากจะทำให้ทราบว่าการเรียงลำดับการส่งข้อมูลด้วยวิธี FEP มีความเหมาะสมในการจัดลำดับข้อมูลสองประเภทที่ต้องการการรับประกันคุณภาพที่แตกต่างกัน เนื่องจากใช้ช่องสัญญาณได้เต็มประสิทธิภาพ

ที่สัดส่วนของหน่วยข้อมูลนอนคิวโอเอสมากกว่าคิวโอเอสสองเท่า ($n_{NoS}/n_{oS} = 2$) วิธี FEP ก็ไม่ได้เสียเปรียบวิธีอื่น การที่วิธี FEP มีค่าหน่วยเวลาเฉลี่ยมากกว่าอีกสองวิธี ดังภาพที่ 31 ก็เป็นผลสืบเนื่องมาจากอัตราการเกิดหน่วยข้อมูลเสียมีค่าน้อยที่สุดดังในภาพที่ 32 จึงทำให้มีปริมาณหน่วยข้อมูลที่ส่งสำเร็จมากที่สุด

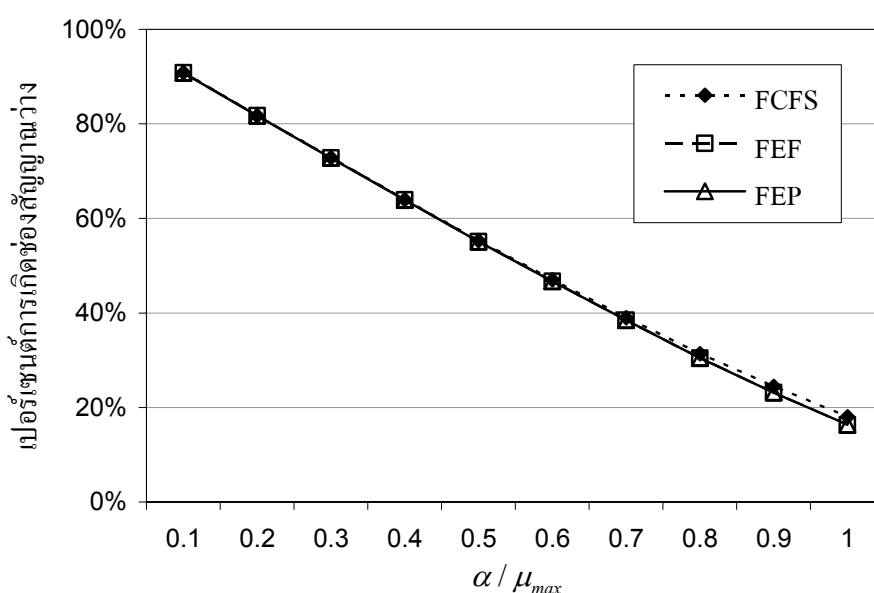
ดังนั้นวิธี FEP จึงน่าจะเป็นแนวคิดในการจัดลำดับที่เหมาะสมสำหรับการส่งข้อมูลแบบรวมช่องสัญญาณระหว่างหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอส เราจึงนำวิธีจัดลำดับ FEP มาพัฒนาต่อในส่วนของการประหยัดพลังงาน

การเปรียบเทียบกับงานวิจัยอื่น

เมื่อเปรียบเทียบวิธี FEP กับวิธีการจัดลำดับในงานวิจัยของ Chipalkatti *et al.* (1989) ซึ่งได้พยายามแก้ไขปัญหาด้านการจัดลำดับ ระหว่างหน่วยข้อมูลคิวโอเอสและหน่วยข้อมูลนอนคิวโอเอสเหมือนกัน แต่วิธีการจัดลำดับในงานวิจัยของ Chipalkatti *et al.* (1989) มีข้อจำกัดที่ว่าจำเป็นต้องกำหนดค่า (threshold) ให้กับระบบ ซึ่งเป็นค่าที่ใช้ตัดสินใจว่าควรส่งหน่วยข้อมูลทางเส้นทางใดก่อน เช่นว่า เมื่อมีหน่วยข้อมูลคิวโอเอสไม่สามารถส่งได้ทันเส้นตายเกินกว่าค่าที่ตั้งไว้ ก็ให้เพิ่มอัตราการส่งหน่วยข้อมูลคิวโอเอสให้สูงขึ้น (กำหนด priority ให้สูงขึ้น) หรือเมื่อมีปริมาณหน่วยข้อมูลนอนคิวโอเอส คั่งค้างในบัฟเฟอร์เกินกว่าค่าที่ตั้งไว้ ก็ให้เพิ่มอัตราการส่งหน่วยข้อมูลนอนคิวโอเอสให้สูงขึ้น แต่สำหรับวิธี FEP ไม่มีความจำเป็นต้องกำหนดค่า (threshold) การตัดสินใจว่าจะให้ช่องสัญญาณกับหน่วยข้อมูลใดนั้น พิจารณาจากค่าที่เราทราบจากหน่วยข้อมูลนั้นๆ โดย สำหรับหน่วยข้อมูลนอนคิวโอเอส หน่วยข้อมูลใดมาถึงก่อนก็สมควรให้ส่งก่อน สำหรับหน่วยข้อมูลคิวโอเอส หน่วยข้อมูลใดต้องรีบส่งให้ทันเส้นตาย (deadline ก่อน) ก็สมควรให้ส่งก่อน และการเลือกว่าจะส่งหน่วยข้อมูลนอนคิวโอเอสหรือคิวโอเอสก่อน ให้พิจารณาว่าหน่วยข้อมูลใดสามารถส่งสำเร็จด้วยวิธี GPS ก่อน ก็สมควรให้ส่งก่อน

จากเหตุผลดังกล่าว วิธี FEP จึงน่าจะมีความยืดหยุ่นและเป็นอิสระต่อการกำหนดค่าเริ่มต้นได้ดีกว่าวิธีการจัดลำดับในงานวิจัยของ Chipalkatti *et al.* (1989) อีกทั้งยังมีความเหมาะสมเพื่อการฝึกกับส่วนของการประหยัพลังงานอีกด้วย

ข้อสังเกตเพื่อการพัฒนาส่วนของการประหยัพลังงาน



ภาพที่ 33 เปอร์เซนต์การเกิดช่องสัญญาณว่าง โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{NQoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร

จากภาพที่ 33 เราสังเกตเห็นว่าช่องสัญญาณขาออกไม่ได้ส่งออกข้อมูลตลอดเวลา โดยในกรณีโหลดไม่คับคั่ง ช่องสัญญาณจะว่างเกินกว่าร้อยละ 50 และจะว่างมากขึ้นเมื่อโหลดเบาบางขึ้น จากการสังเกตนี้เราจึงนำวิธี DMS มาประยุกต์โดยลดอัตราบิตในการส่งบางหน่วยข้อมูลลงเพื่อประโยชน์ในการประหยัพลังงาน

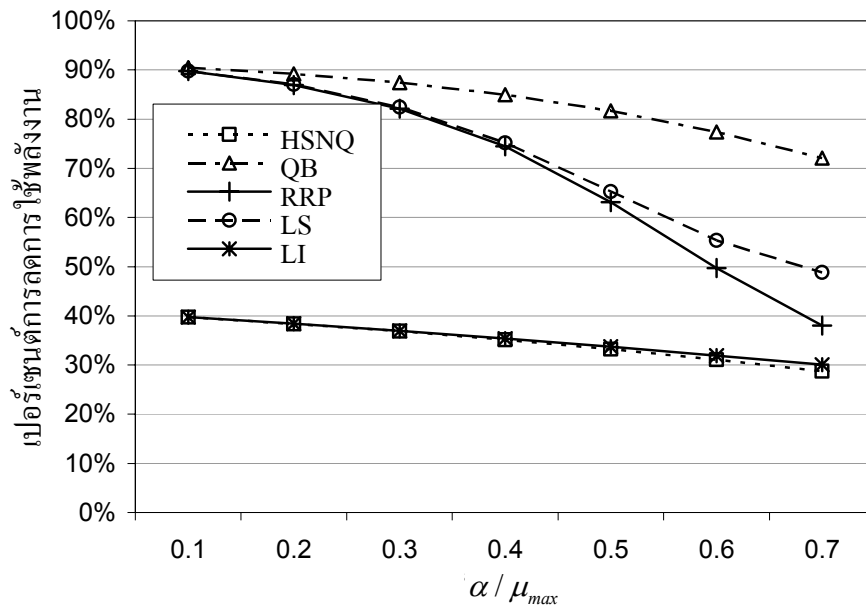
ขั้นตอนที่ 2 (กระบวนการประหยัดพลังงาน)

ความรวดเร็วในการส่งเป็นความต้องการพื้นฐานสำหรับข้อมูลทุกชนิด และการประหยัดพลังงานก็มีความจำเป็นอย่างยิ่งในระบบที่มีพลังงานจำกัดอย่างเช่น WSNs แต่การประหยัดพลังงานนั้นแปรผกผันกับความเร็ว ซึ่งแสดงว่า ในกรณีที่มีกำลังงานจำกัด เมื่อลดการใช้พลังงานลงย่อมมีผลกระทบต่อประสิทธิภาพในการส่งข้อมูล ไม่ว่าจะเป็นค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสที่นานขึ้น และอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสที่สูงขึ้นอย่างหลีกเลี่ยงไม่ได้ เมื่อเทียบกับตอนที่ยังไม่มีกรณีการประหยัดพลังงาน นั่นคือบิตข้อมูลที่ผ่านการจัดลำดับด้วยวิธี FEP แล้วส่งออกทุกบิตด้วยความเร็วสูงสุดตลอดเวลา ดังนั้นเราจึงได้คิดค้นแนวคิดในการประหยัดพลังงาน โดยประสิทธิภาพในการส่งข้อมูลอาจสูญเสียไปบ้าง แต่เห็นว่าคุ้มค่าเมื่อเทียบกับพลังงานที่ประหยัดได้

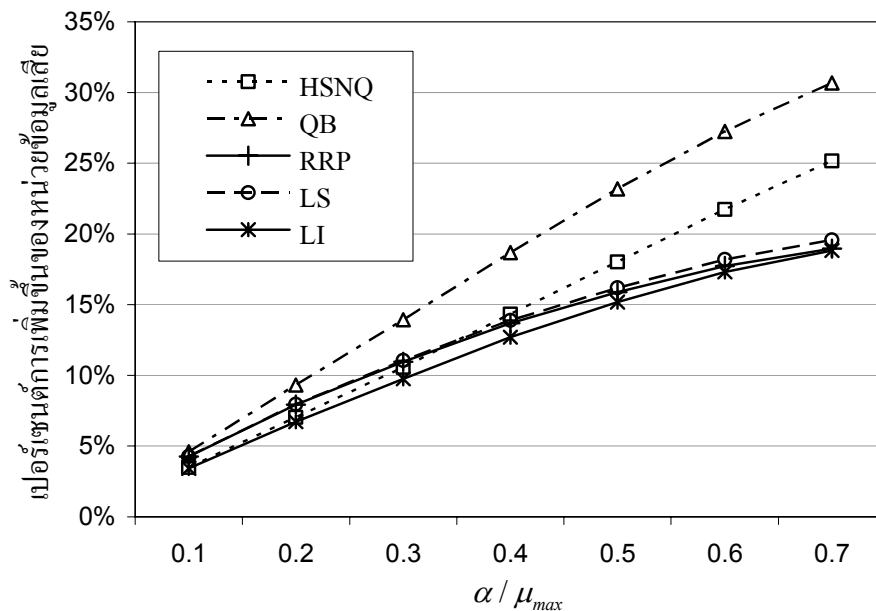
จากที่กล่าวในส่วนของวิธีการว่าหน่วยข้อมูลคิวโอเอสเป็นหน่วยข้อมูลที่ถูกจำกัดด้วยเส้นตาย จึงทำให้เราหาอัตราบิตที่เหมาะสมได้ นั่นคือทำให้หน่วยข้อมูลคิวโอเอสเหล่านั้นส่งได้ทันเส้นตาย แต่สำหรับหน่วยข้อมูลนอนคิวโอเอสนั้น เนื่องจากเป็นหน่วยข้อมูลที่ไม่ถูกจำกัดด้วยเส้นตาย ทำให้เราไม่ทราบว่าควรจะอัตราบิตเท่าไรจึงจะเหมาะสม ดังนั้นแนวคิดที่จะนำเสนอจึงแบ่งออกเป็น 2 ประเภทใหญ่ๆ แนวคิดแรกเป็นแนวคิดในการประหยัดพลังงานที่ต้องมีผลกระทบต่อค่าความหน่วงเวลาเฉลี่ยน้อยที่สุด นั่นคือให้ส่งทุกหน่วยข้อมูลนอนคิวโอเอสด้วยอัตราบิตสูงสุด เพื่อให้ผลจากการประหยัดพลังงานกระทบต่อค่าความหน่วงเวลาเฉลี่ยให้น้อยที่สุด แนวคิดนี้ได้แก่วิธี HSNQ และ LI แนวคิดที่สอง เป็นแนวคิดเป็นแนวคิดที่ยินยอมให้หน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดความเร็วในการส่งหน่วยข้อมูลนอนคิวโอเอส ซึ่งจะส่งผลต่อค่าความหน่วงเวลาเฉลี่ย โดยจะขึ้นอยู่กับความเร่งรีบของหน่วยข้อมูลคิวโอเอส นั่นคือหน่วยข้อมูลถูกกำกับด้วยเส้นตายที่สั้น ก็จะทำให้ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสมีค่าน้อยตาม แนวคิดนี้ได้แก่วิธี QB, RRP, LS, RRP w/ RPS และ LS w/ RPS

การวิเคราะห์ผลในเชิงการแปรค่าปริมาณโหลด (α / μ_{max})

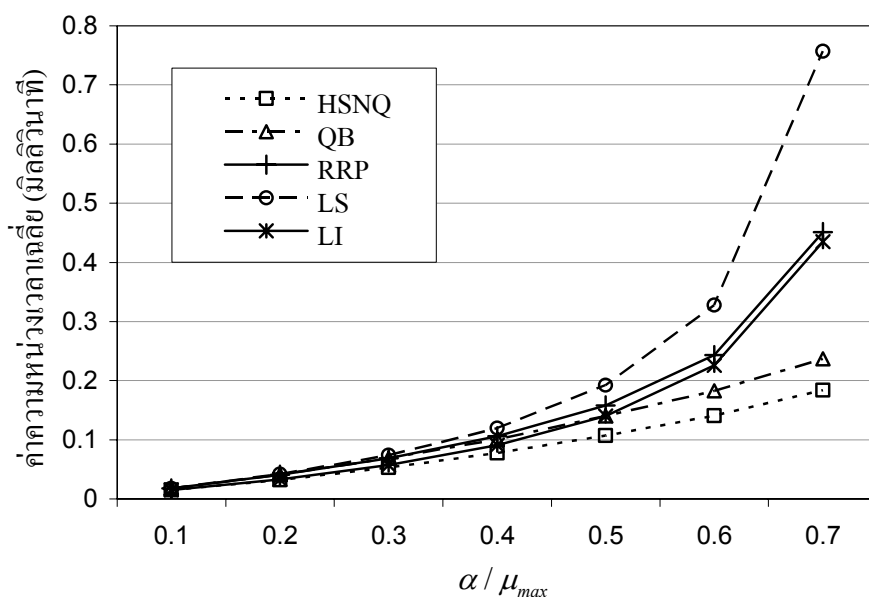
การประหยัดพลังงานด้วยวิธีการปรับเปลี่ยนอัตราบิตนั้น เมื่ออัตราบิตลดลงย่อมช่วยในการประหยัดพลังงานแต่ก็ต้องแลกกับระยะเวลาในการส่งข้อมูลที่นานขึ้น เวลาที่นานขึ้นนี้เป็นที่แน่นอนว่าต้องกระทบต่ออัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส เมื่อเทียบกับตอนที่ยังไม่มี การปรับลดอัตราบิต และมีผลกระทบต่อความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลบนคิวโอเอสอย่างหลีกเลี่ยงไม่ได้ วิธีการประหยัดพลังงานที่จะนำเสนอนี้ มีกระบวนการปรับอัตราบิตที่แตกต่างกัน ซึ่งก็ จะทำให้ผลที่ได้มีค่าแตกต่างกัน ดังภาพที่ 34, 35 และ 36 แต่ทุกวิธีมีแนวโน้มไปในทิศทางเดียวกันก็คือเมื่อ มีปริมาณโหลดมากขึ้นจะสามารถประหยัดพลังงานได้น้อยลง อัตราการเกิดหน่วยข้อมูลเสียของหน่วย ข้อมูลคิวโอเอสสูงขึ้น และความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลบนคิวโอเอสมีค่ามากขึ้น ที่เป็น เช่นนี้เพราะ การจะส่งข้อมูลปริมาณมากให้ทันเส้นตายของหน่วยข้อมูลคิวโอเอสนั้นจำเป็นต้องใช้ อัตราบิตในการส่งที่สูงจึงประหยัดพลังงานได้น้อยลง และแม้ว่าจะส่งด้วยอัตราบิตที่สูงขึ้นแต่ด้วย ปริมาณข้อมูลที่มากจึงมีข้อมูลค้างค้างอยู่ในบัฟเฟอร์เกือบตลอดเวลา จึงเป็นสาเหตุให้หน่วยข้อมูลบน คิวโอเอสมีค่าความหน่วงเวลาเฉลี่ยนานขึ้น และการที่หน่วยข้อมูลคิวโอเอสก็ต้องเข้าคิวในการรอส่งจึง ทำให้บางหน่วยข้อมูลไม่สามารถส่งได้ทันเส้นตายโดยยังมีปริมาณโหลดคั่งค้างขึ้น ยิ่งทำให้คิวยาวขึ้น ก็ จะยังเกิดหน่วยข้อมูลเสียมากขึ้น



ภาพที่ 34 เปอร์เซนต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) เมื่อกำหนดให้ $t_{min}/T = 0.2$, $n_{QoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 35 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

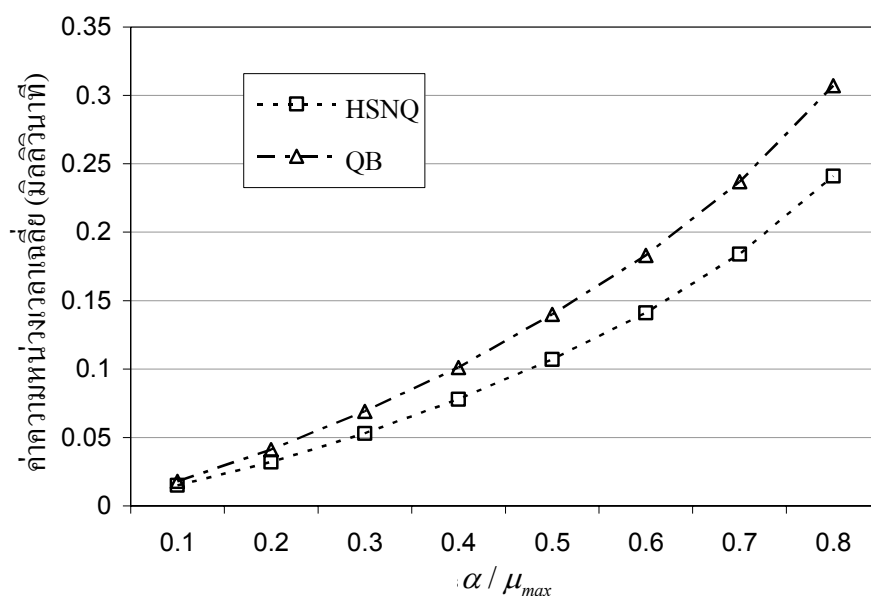


ภาพที่ 36 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

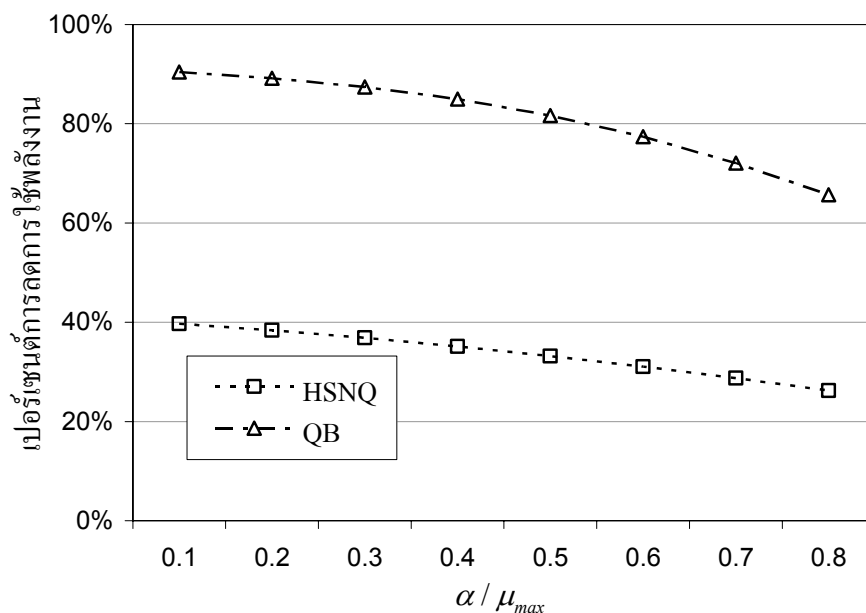
วิธีการประหยัดพลังงานที่นำเสนอนี้มีแนวโน้มไปในทางเดียวกัน แต่กระบวนการมีความแตกต่างกันผลในการประหยัดพลังงานจึงมีค่าแตกต่างกัน และผลกระทบจากการประหยัดพลังงานก็มีค่าแตกต่างกันด้วย

วิธี HSNQ เป็นวิธีที่มีความซับซ้อนน้อยที่สุดในแนวคิดการประหยัดประเภทแรก (แนวคิดในการประหยัดพลังงานที่ต้องมีผลกระทบต่อค่าความหน่วงเวลาน้อยที่สุด) และ QB ก็เป็นวิธีที่มีความซับซ้อนน้อยที่สุดในแนวคิดการประหยัดประเภทที่สอง (แนวคิดที่ยินยอมให้หน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดความเร็วในการส่งหน่วยข้อมูลนอนคิวโอเอส) โดยสองวิธีนี้จะเป็นการคำนวณโดยพิจารณาจากหน่วยข้อมูลที่กำลังหาอัตราบิตเพียงหน่วยเดียว ซึ่งไม่สนใจว่าหน่วยข้อมูลที่เหลือจะมีความเร่งรีบหรือมีปริมาณมากน้อยเพียงใด ความแตกต่างของสองวิธีก็คือการหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลนอนคิวโอเอส โดยวิธี HSNQ จะส่งด้วยอัตราบิตสูงสุดด้วยเหตุผลที่ว่าต้องการให้ผลกระทบจากการประหยัดพลังงาน มีผลต่อค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสน้อยที่สุด แต่วิธี QB อัตราบิตที่ใช้ในการส่งหน่วยข้อมูลนอนคิวโอเอสจะขึ้นอยู่กับอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลคิวโอเอสหรือก็ขึ้นอยู่กับความเร่งรีบของหน่วยข้อมูลนอนคิวโอเอสนั่นเอง

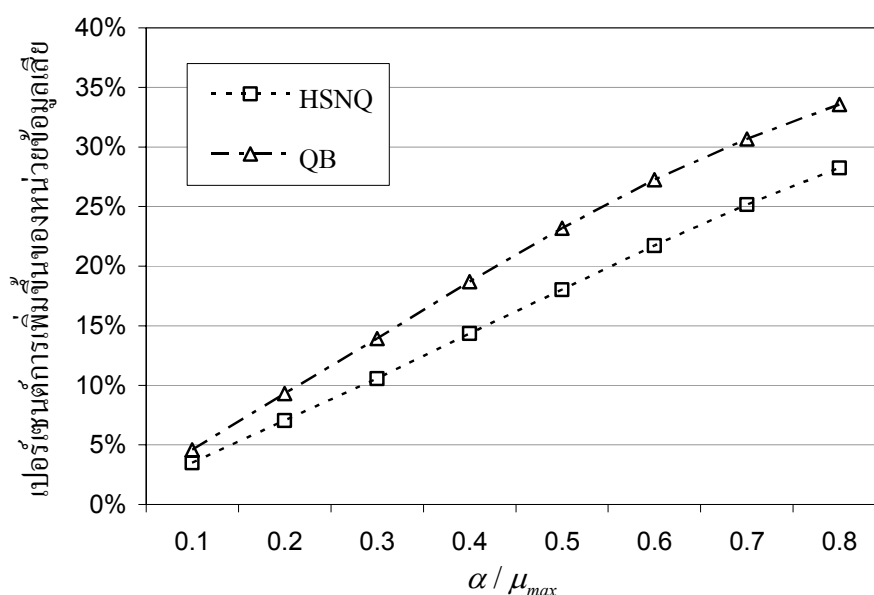
เพื่อแสดงการเปรียบเทียบให้ชัดเจนยิ่งขึ้นจึงแยกวิธี HSNQ และ QB ออกมาพิจารณาต่างหาก ในภาพที่ 37, 38 และ 39 ซึ่งเป็นผลการทดลองชุดเดิมที่ได้แสดงไว้ในภาพที่ 34, 35 และ 36 ตามลำดับ



ภาพที่ 37 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



ภาพที่ 38 เปอร์เซนต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



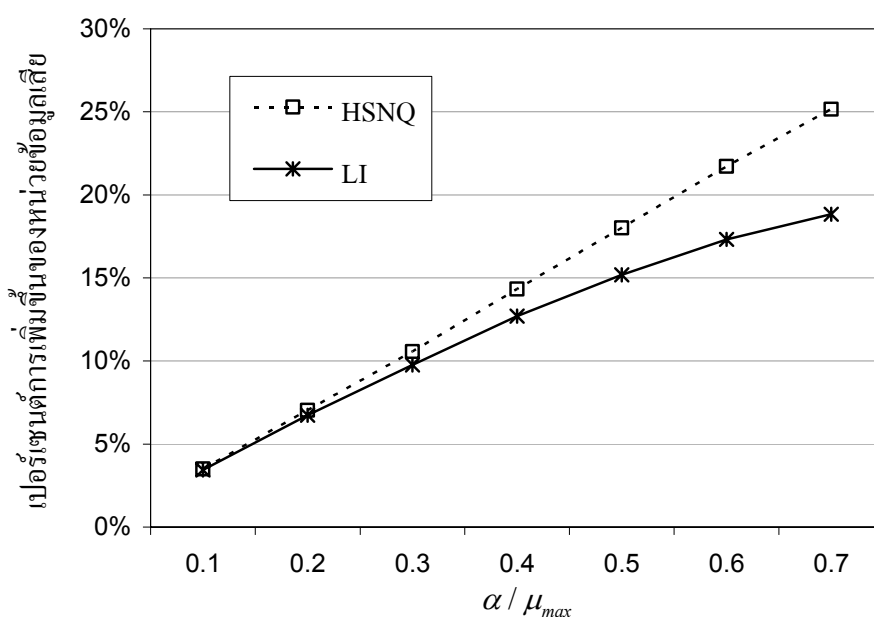
ภาพที่ 39 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

จากผลในภาพที่ 37, 38 และ 39 จะเห็นว่าการประหยัดพลังงานด้วยวิธี HSNQ จะทำให้ความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส (ภาพที่ 37) มีค่าน้อยกว่าวิธี QB ก็ด้วยเหตุผลดังที่กล่าวแล้วคือวิธี HSNQ จะใช้อัตราบิตสูงสุดในการส่งหน่วยข้อมูลนอนคิวโอเอส แต่การส่งหน่วยข้อมูลนอนคิวโอเอสด้วยอัตราบิตสูงสุดนี้จะทำให้สามารถประหยัดพลังงานลงได้น้อยกว่าวิธี QB ซึ่งพลังงานที่ประหยัดได้มีค่าแตกต่างกันประมาณ 40% โดยเฉพาะอย่างยิ่งในกรณีโหลดเบาบาง วิธี HSNQ ประหยัดพลังงานลงได้น้อยกว่าวิธี QB 50% ดังภาพที่ 38 แต่การบริโภคพลังงานที่แตกต่างกันนี้แทบจะไม่เห็นความแตกต่างของค่าความหน่วงเวลาเฉลี่ยที่เกิดขึ้นกับหน่วยข้อมูลนอนคิวโอเอส อีกทั้งความแตกต่างของอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส ก็มีค่าน้อยมากดังแสดงในภาพที่ 39

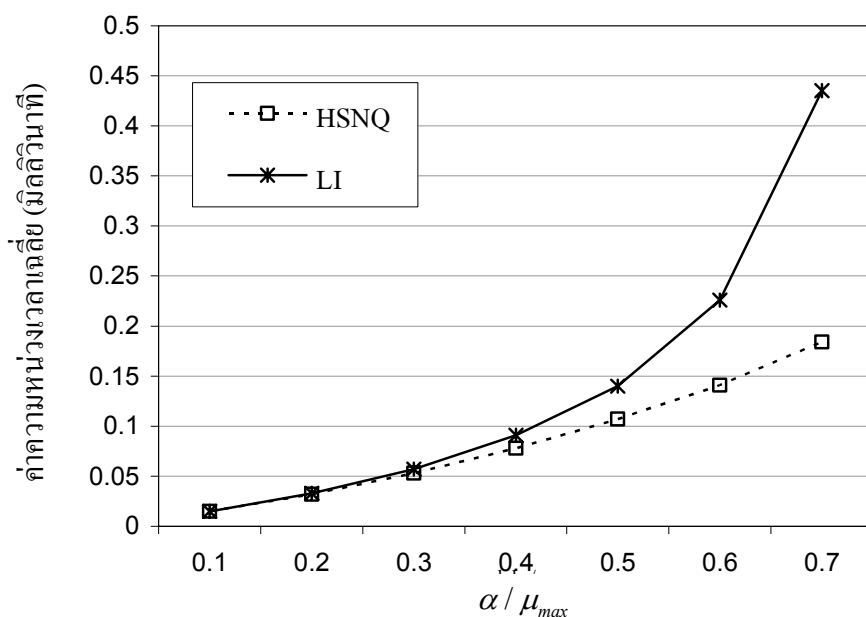
เนื่องจากวิธี HSNQ จะเป็นการคำนวณโดยพิจารณาจากหน่วยข้อมูลที่กำลังหาอัตราบิตเพียงหน่วยเดียว ซึ่งไม่สนใจว่าหน่วยข้อมูลที่เหลือจะมีความเร่งรีบหรือมีปริมาณมากน้อยเพียงใด จึงทำให้เป็นวิธีที่มีความซับซ้อนน้อยที่สุด แต่ถ้าระบบสามารถรองรับความซับซ้อนได้มากขึ้น เราจะสามารถ

ลดปริมาณหน่วยข้อมูลเสียลงได้ด้วยการคำนวณหาอัตราบิด โดยพิจารณาจากทุกหน่วยข้อมูลในบัฟเฟอร์

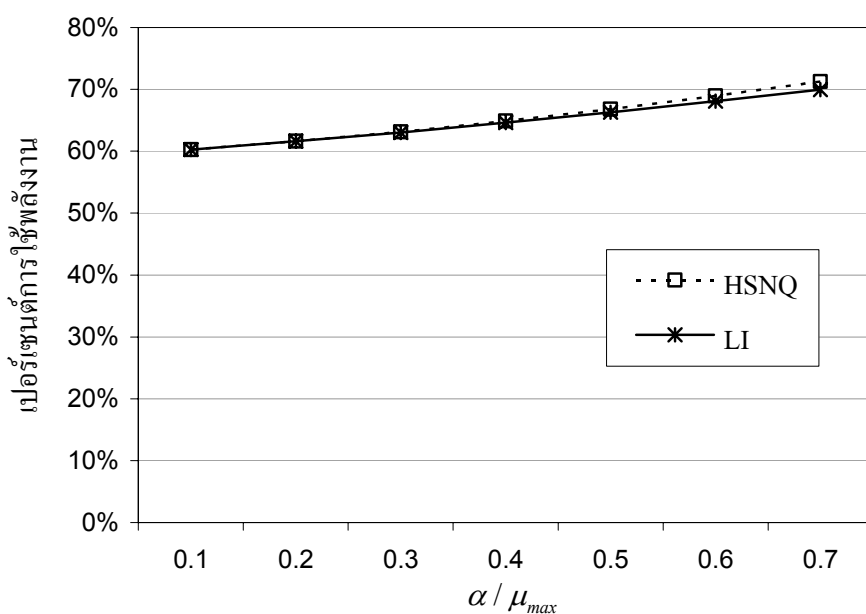
วิธี LI เป็นวิธีที่พัฒนามาจากวิธี HSNQ โดยเพิ่มส่วนของการคำนวณซ้ำดังที่กล่าวไว้ในส่วนของวิธีการ การคำนวณซ้ำนี้มีประโยชน์ในการช่วยลดอัตราการเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส ซึ่งผลที่ได้ก็เป็นไปตามนั้น ดังแสดงในภาพที่ 40 แต่การที่หน่วยข้อมูลเสียลดลงยอมทำให้มีปริมาณหน่วยข้อมูลรวมส่งผ่านช่องสัญญาณได้มากขึ้น จึงส่งผลให้หน่วยข้อมูลนอนคิวโอเอสต้องรอคิวนานขึ้นดังภาพที่ 41 ส่วนพลังงานที่ใช้ในการส่งข้อมูลดังแสดงในภาพที่ 42 มีค่าแทบไม่แตกต่างกันทั้งๆ วิธี LI ส่งออกหน่วยข้อมูลในปริมาณที่มากกว่าก็เพราะว่าการคำนวณซ้ำเพื่อช่วยลดอัตราการเกิดหน่วยข้อมูลเสีย เป็นการคำนวณหาอัตราบิดที่ต่ำที่สุดที่ทำให้ส่งทุกหน่วยข้อมูลในบัฟเฟอร์ได้ทัน



ภาพที่ 40 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



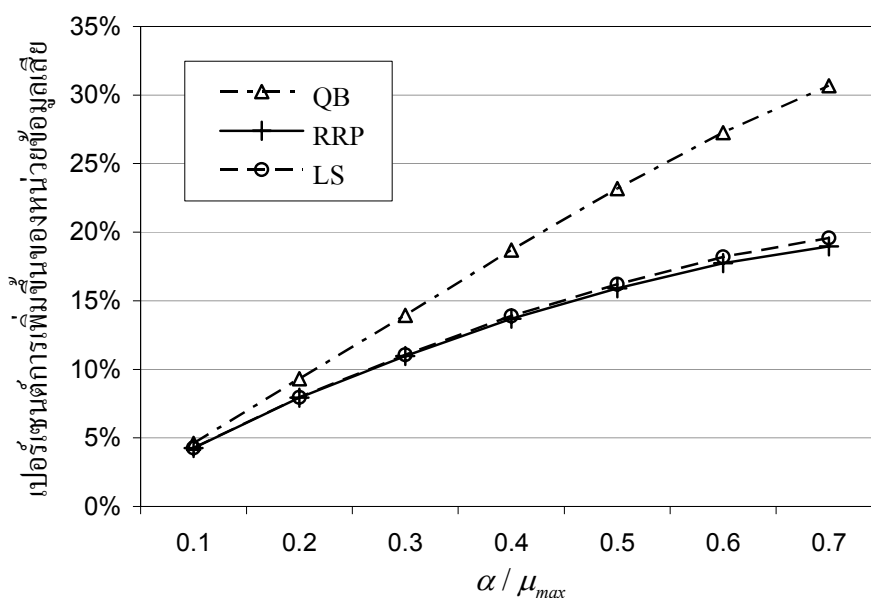
ภาพที่ 41 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



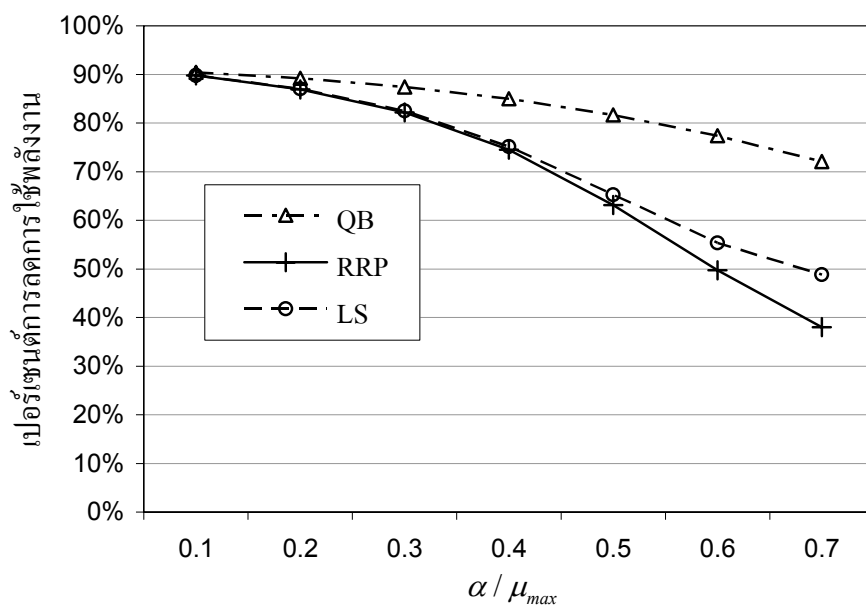
ภาพที่ 42 เปอร์เซนต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนวกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

ในการทำงานเดียวกันกับ HSNQ เนื่องจากวิธี QB เป็นการคำนวณโดยพิจารณาจากหน่วยข้อมูลที่กำลังหาอัตราบิดเพียงหน่วยเดียว ซึ่งไม่สนใจว่าหน่วยข้อมูลที่เหลือจะมีความเร่งรีบหรือมีปริมาณมากน้อยเพียงใด จึงทำให้เป็นวิธีที่มีความซับซ้อนน้อยที่สุด แต่ถ้าระบบสามารถรองรับความซับซ้อนได้มากขึ้น เราจะสามารถลดปริมาณหน่วยข้อมูลเสียลงได้ด้วยการคำนวณหาอัตราบิด โดยพิจารณาจากทุกหน่วยข้อมูลในบัพเฟอร์

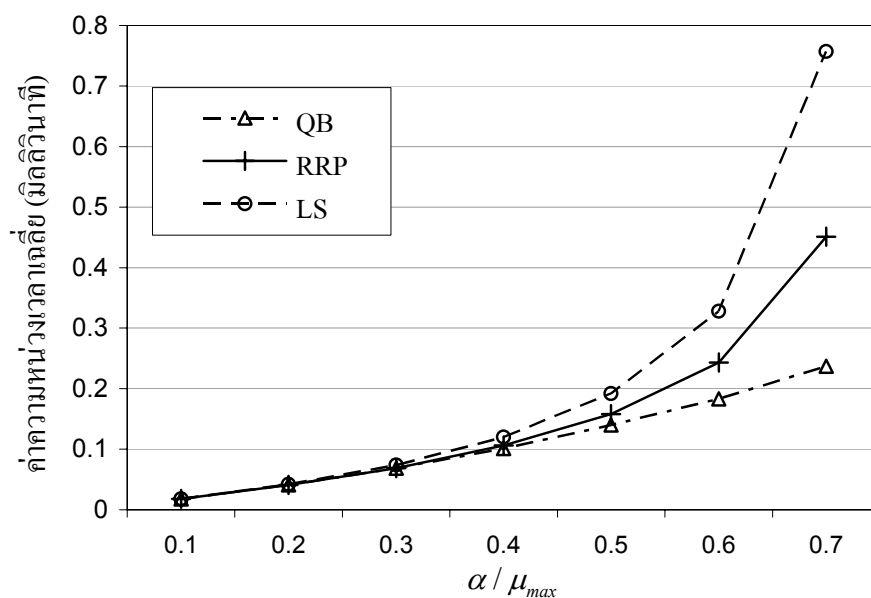
วิธี RRP และ LS เป็นวิธีที่พัฒนามาจากวิธี QB โดยเพิ่มส่วนของการคำนวณซ้ำดังที่กล่าวไว้ในส่วนของวิธีการ การคำนวณซ้ำนี้มีประโยชน์ในการช่วยลดอัตราการเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส ซึ่งผลที่ได้ก็เป็นไปตามนั้น ดังแสดงในภาพที่ 43



ภาพที่ 43 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



ภาพที่ 44 เปอร์เซนต์การลดการใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



ภาพที่ 45 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวไอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

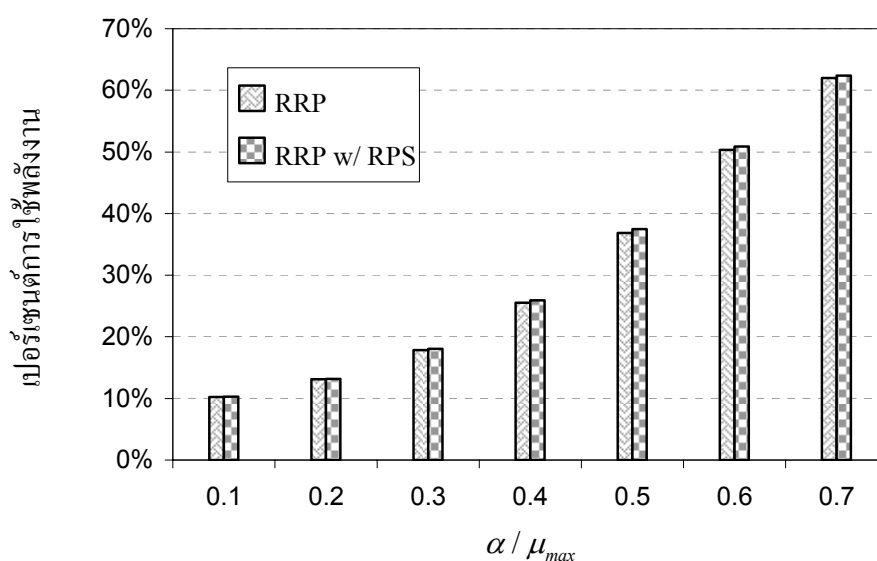
จากภาพที่ 44 ที่ปริมาณโหลดต่างๆ พลังงานที่ประหยัดได้ก็แตกต่างกัน ตามที่คาดไว้ว่า ช่วงเวลาที่โหลดไม่คับคั่งจะสามารถประหยัดพลังงานลงได้ และผลที่ได้ทั้ง 3 วิธี ก็เป็นไปได้คาด ที่ปริมาณโหลดบางวิธี QB, RRP และ LS สามารถลดการใช้พลังงานลงจากเดิมได้สูงกว่าร้อยละ 90 โดยมีผลกระทบต่ออัตราการเกิดหน่วยข้อมูลเสียและค่าความหน่วงเวลาเฉลี่ยสูงขึ้น จากตอนที่ไม่นึก ส่วนของการประหยัดพลังงานเพียงเล็กน้อยดังภาพที่ 43 และ 45

วิธี RRP มีการทำงานพื้นฐานคล้ายกับวิธี QB แต่มีกระบวนการคำนวณหาอัตราบิดใหม่ให้กับทุกหน่วยข้อมูลในบัฟเฟอร์ในกรณีที่คำนวณด้วยวิธี QB แล้วมีบางหน่วยข้อมูลคิวโอเอสส่งไม่ทันเส้นตาย การคำนวณหาอัตราบิดใหม่นี้พิจารณาจากทุกหน่วยข้อมูลในบัฟเฟอร์ว่าแต่ละหน่วยข้อมูลมีความเร่งรีบเพียงใด แล้วให้หน่วยข้อมูลที่เร่งรีบที่สุดเป็นตัวกำหนดอัตราบิดที่ใช้ส่งว่าควรสูงขึ้นเท่าใด จึงจะส่งข้อมูลทั้งหมดได้ทัน พร้อมทั้งยังคำนึงถึงการใช้พลังงาน แม้ว่าวิธี RRP จะประหยัดพลังงานได้น้อยที่สุด ดังรูป 34 แต่ก็ยังเป็นผลมาจากปริมาณข้อมูลที่ส่งสำเร็จที่มากขึ้น

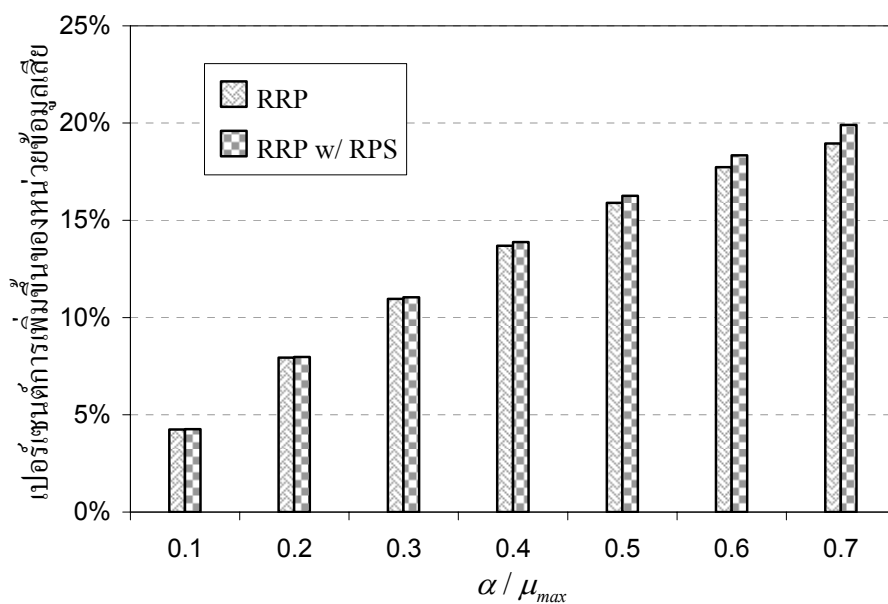
จากข้อดีของวิธี RRP เราจึงพัฒนามาเป็น LS ซึ่งมีคำนวณหาอัตราบิดโดยพิจารณาจากภาพรวมจากทุกหน่วยข้อมูลในบัฟเฟอร์ในทุกเวลาที่มีหน่วยข้อมูลใหม่เข้าหรือส่งหน่วยข้อมูลใดๆ สำเร็จ ซึ่งอัตราบิดเหล่านี้เป็นอัตราบิดที่ต่ำที่สุดสำหรับทุกหน่วยข้อมูล ที่ทำให้ทุกหน่วยข้อมูลส่งได้ทันเส้นตาย ซึ่งใช้เส้นตายของหน่วยข้อมูลคิวโอเอสที่เร่งรีบที่สุดเป็นตัวกำหนด ดังนั้นทุกหน่วยข้อมูลจึงได้ส่งด้วยอัตราบิดที่ต่ำที่สุดตลอดเวลา จากวิธี RRP ซึ่งได้พัฒนาวิธี QB เพื่อความต้องการในการลดอัตราการเกิดหน่วยข้อมูลเสีย แต่การพัฒนาวิธี RRP มาเป็นวิธี LS กลับทำให้อัตราการเกิดหน่วยข้อมูลเสียเพิ่มขึ้นดังแสดงในภาพที่ 43 เนื่องจากกรณีที่ทดลองเป็นแบบ Non-preemptive การที่หน่วยข้อมูลหนึ่งกำลังส่งที่อัตราบิดต่ำและมีหน่วยข้อมูลคิวโอเอสที่เร่งรีบเข้ามา เราไม่สามารถปรับอัตราบิดในการส่งหน่วยข้อมูลที่กำลังส่งได้ จึงทำให้เหลือเวลาไม่พอให้ส่งหน่วยข้อมูลคิวโอเอสที่เร่งรีบนั้นได้ทัน แต่การหาอัตราบิดที่ต่ำที่สุดตลอดเวลาก็มีข้อดีคือสามารถประหยัดพลังงานลงได้มากที่สุด

วิธี RRP w/ RPS และวิธี LS w/ RPS เป็นวิธีการที่พัฒนามาจากวิธี RRP และ LS ตามลำดับ โดยทั้งสองวิธีนี้จะแตกต่างกับวิธี RRP และ LS ตรงที่เมื่อจำเป็นต้องทิ้งบางหน่วยข้อมูลในบัฟเฟอร์ วิธี RRP และ LS จะเลือกหน่วยข้อมูลคิวโอเอสใหม่ แต่วิธี RRP w/ RPS และ LS w/ RPS จะเลือกทิ้งหน่วยข้อมูลคิวโอเอสที่ต้องใช้ความเร็วในการส่งสูงที่สุดเมื่อเปรียบเทียบกับหน่วยข้อมูลอื่นๆ (ความเร็วสูงสุดนี้มีค่ามากกว่า maximum bit rate เสมอ) เพื่อให้ส่งได้ทันเส้นตาย

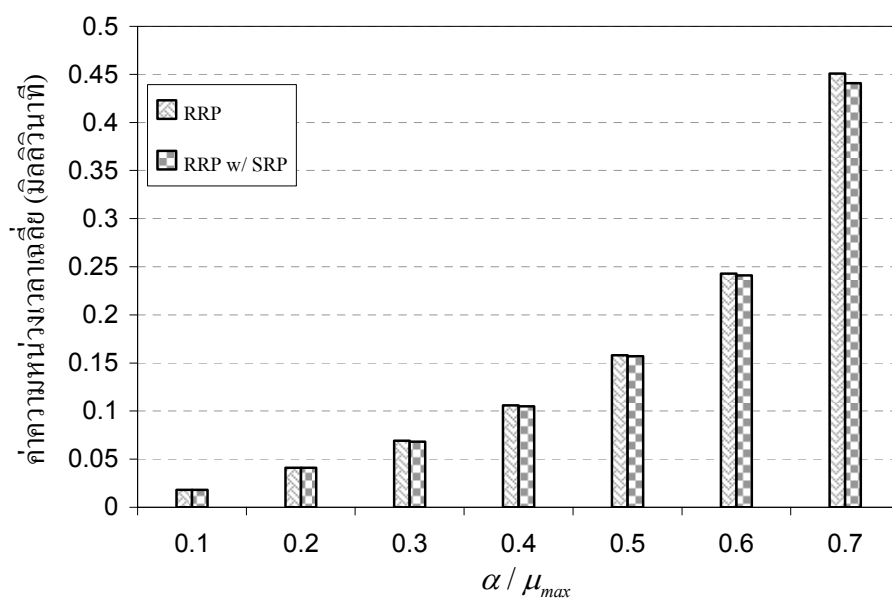
วิธี RRP และวิธี RRP w/ RPS มีกระบวนการที่คล้ายคลึงกันมากจึงทำให้ผลที่ได้มีค่าเกือบจะเท่ากันในทุกๆ กรณี ไม่ว่าจะเป็นอัตราการใช้พลังงาน อัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลบนคิวโอเอส ดังภาพที่ 46, 47 และ 48 โดยความแตกต่างในการเลือกทั้งหน่วยข้อมูลนี้ การเลือกของวิธี RRP จะรับประกันว่าจะทั้งหน่วยข้อมูลเพียงหน่วยข้อมูลเดียว แต่วิธี RRP w/ RPS อาจต้องทั้งหลายหน่วยข้อมูล (การทั้งหลายหน่วยข้อมูลจะเกิดขึ้นในกรณีที่การจัดลำดับไม่ได้เรียงตาม arrival time ซึ่งสามารถดูตัวอย่างการเกิดเหตุการณ์นี้ได้ในภาคผนวก) จึงทำให้วิธี RRP w/ RPS มีอัตราการเกิดหน่วยข้อมูลเสียที่สูงกว่าวิธี RRP ดังภาพที่ 47 แต่ก็ต่างกันเพียงเล็กน้อยเพราะเหตุการณ์ในการเลือกทั้งข้อมูลที่แตกต่างกันเกิดขึ้นไม่บ่อย การที่วิธี RRP w/ RPS มีหน่วยข้อมูลเสียมากกว่าวิธี RRP จะทำให้ค่าเฉลี่ยจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลคิวโอเอสมีค่าสูงกว่าตาม เนื่องจากทุกครั้งที่เกิดหน่วยข้อมูลเสีย เราจะเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลนั้นไว้ที่จำนวนบิตต่อสัญลักษณ์สูงสุด ดังนั้นหน่วยข้อมูลบนคิวโอเอสจึงส่งด้วยอัตราบิตที่สูงกว่า จึงทำให้ใช้พลังงานงานในการส่งมากกว่าและมีค่าความหน่วงเวลาน้อยกว่าวิธี RRP ดังภาพที่ 46 และ 48 ตามลำดับ



ภาพที่ 46 เปอร์เซนต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนวกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

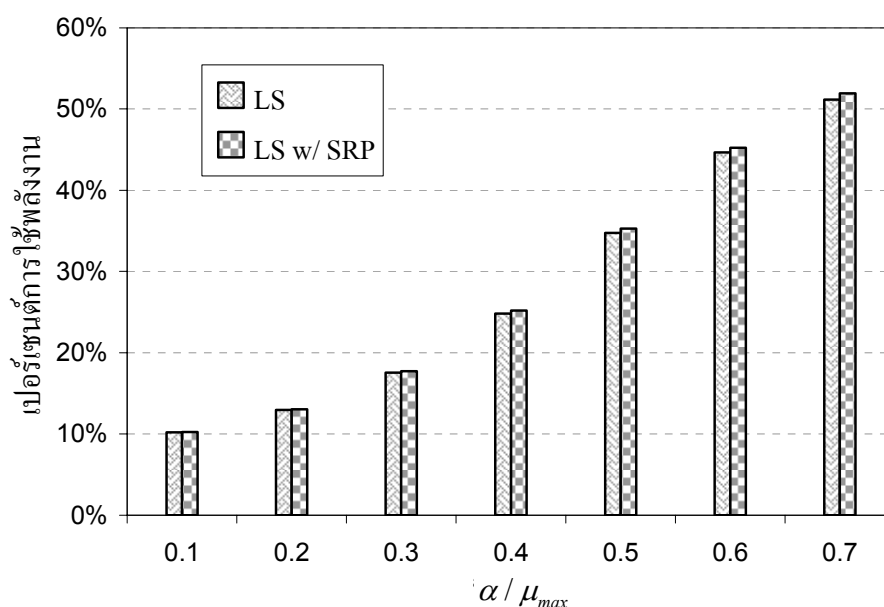


ภาพที่ 47 เปอร์เซ็นต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวไอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

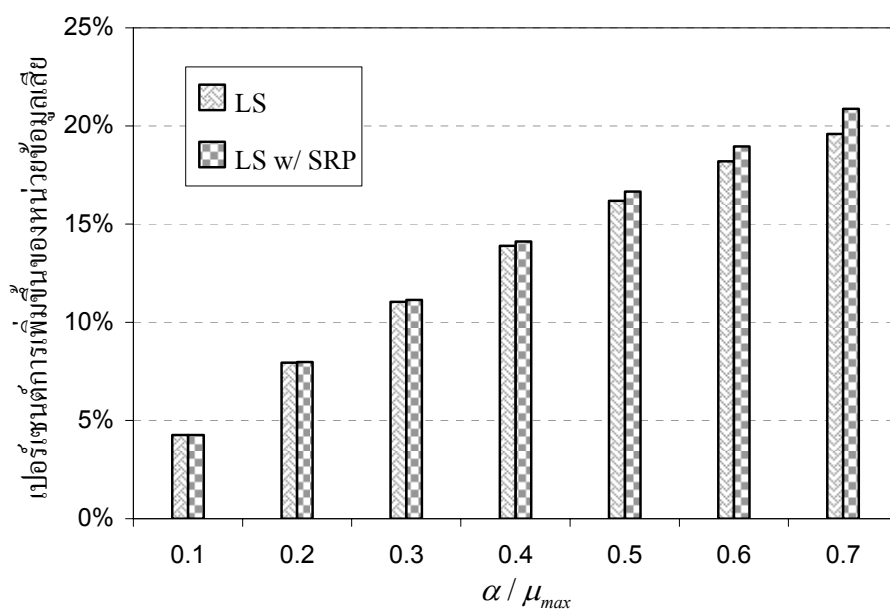


ภาพที่ 48 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลคิวไอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

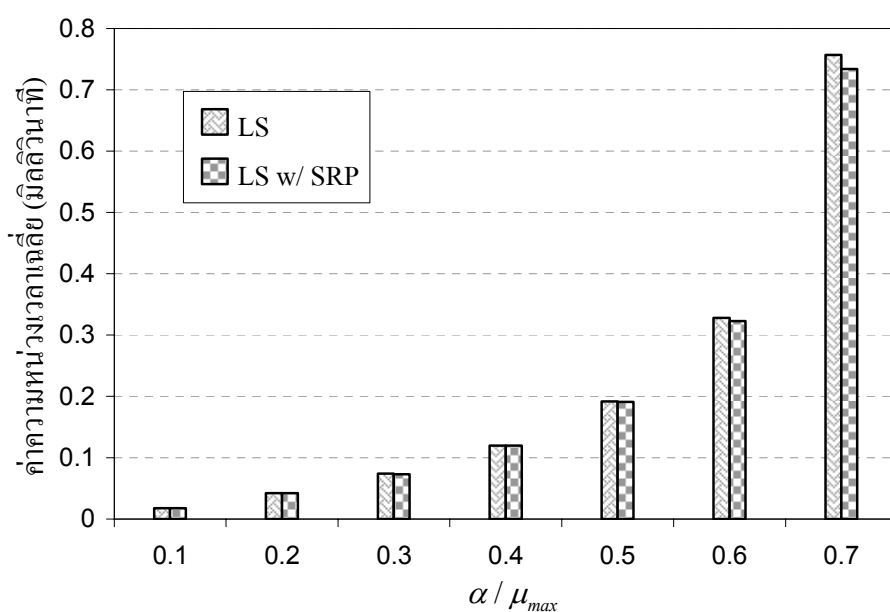
วิธี LS และวิธี LS w/ RPS มีกระบวนการที่คล้ายคลึงกันมากจึงทำให้ผลที่ได้มีค่าเกือบจะเท่ากันในทุกๆ กรณี ไม่ว่าจะเป็นอัตราการใช้พลังงาน อัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส ดังภาพที่ 49, 50 และ 51 โดยความแตกต่างในการเลือกทั้งหน่วยข้อมูลนี้ การเลือกของวิธี LS จะรับประกันว่าจะทั้งหน่วยข้อมูลเพียงหน่วยข้อมูลเดียว แต่วิธี LS w/ RPS อาจต้องทั้งหลายหน่วยข้อมูล (การทั้งหลายหน่วยข้อมูลจะเกิดขึ้นในกรณีที่การจัดลำดับไม่ได้เรียงตาม arrival time ซึ่งสามารถดูตัวอย่างการเกิดเหตุการณ์นี้ได้ในภาคผนวก) จึงทำให้วิธี LS w/ SRP มีอัตราการเกิดหน่วยข้อมูลเสียที่สูงกว่าวิธี LS ดังภาพที่ 50 แต่ก็ต่างกันเพียงเล็กน้อยเพราะเหตุการณ์ในการเลือกทั้งข้อมูลที่แตกต่างกันเกิดขึ้นไม่บ่อย การที่วิธี LS w/ RPS มีหน่วยข้อมูลเสียมากกว่าวิธี LS จะทำให้ค่าเฉลี่ยจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลคิวโอเอสมีค่าสูงกว่าตาม เนื่องจากทุกครั้งที่เกิดหน่วยข้อมูลเสีย เราจะเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลนั้นไว้ที่จำนวนบิตต่อสัญลักษณ์สูงสุด ดังนั้นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่สูงกว่า จึงทำให้ใช้พลังงานงานในการส่งมากกว่าและมีค่าความหน่วงเวลาน้อยกว่าวิธี LS ดังภาพที่ 49 และ 51 ตามลำดับ



ภาพที่ 49 เปอร์เซนต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนึกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนึกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34



ภาพที่ 50 เปรี่เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

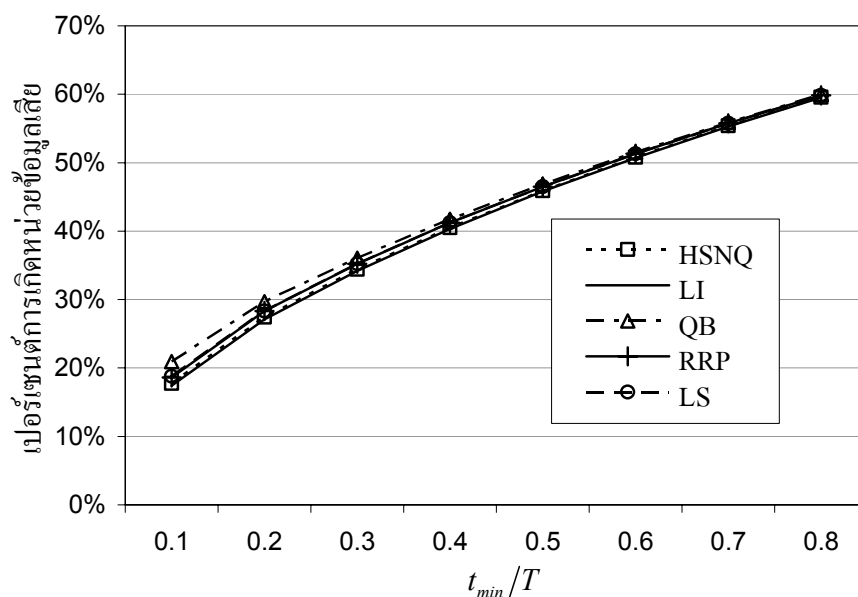


ภาพที่ 51 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลคิวโอเอส โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

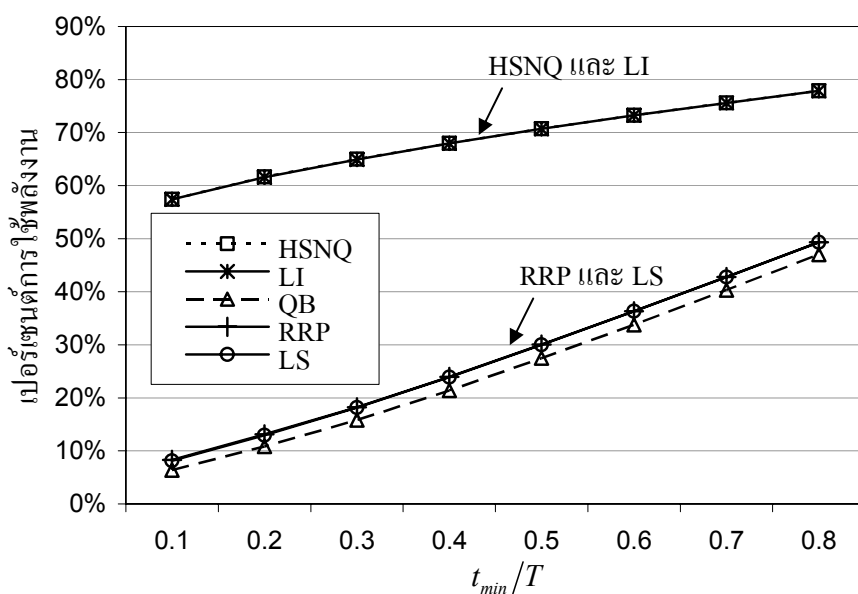
การวิเคราะห์ผลในเชิงการแปรค่าความเร่งรีบของหน่วยข้อมูลคิวโอเอส (t_{min}/T)

เนื่องจากวิธีการประหยัดพลังงานที่นำเสนอในงานวิจัยชิ้นนี้ เป็นการปรับเปลี่ยนอัตราบิตที่ใช้ในการส่งข้อมูลตามความเร่งรีบของหน่วยข้อมูลคิวโอเอส เมื่อหน่วยข้อมูลคิวโอเอสมีความเร่งรีบมากขึ้น (ค่า t_{min}/T สูงขึ้น) ย่อมทำให้เกิดหน่วยข้อมูลเสียได้ง่าย (ดังภาพที่ 52) แสดงว่าจำเป็นต้องส่งหน่วยข้อมูลด้วยอัตราบิตที่สูงขึ้น เพื่อให้ส่งหน่วยข้อมูลคิวโอเอสได้ทันเส้นตาย อัตราบิตที่สูงขึ้นนี้ ย่อมทำให้ต้องใช้พลังงานในการส่งที่มากขึ้น (ดังภาพที่ 53) และอัตราบิตที่สูงขึ้นยังทำให้ใช้ระยะเวลาในการส่งข้อมูลลดลงจึงทำให้ความหน่วงเวลาเฉลี่ยมีค่าน้อยลง (ดังภาพที่ 54)

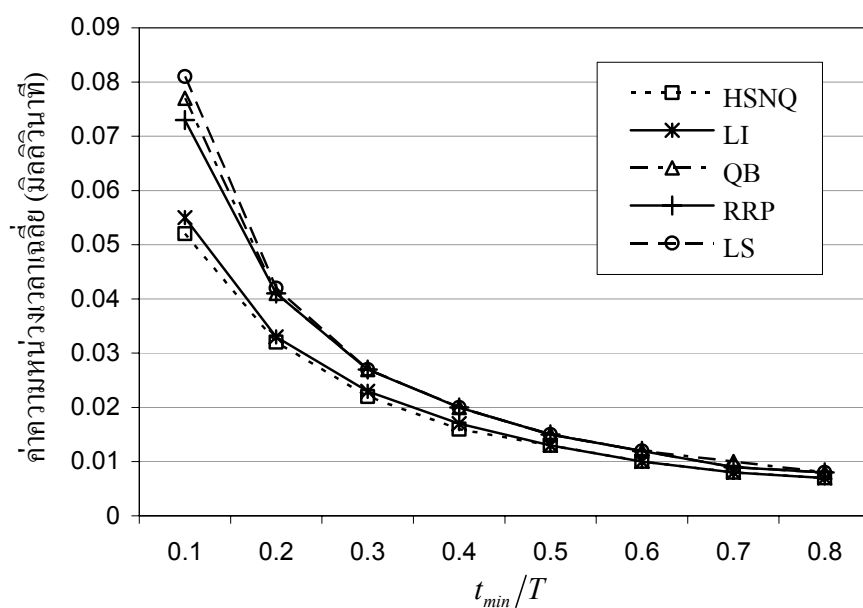
ตามที่กล่าวไว้ว่า การประหยัดพลังงานในการส่งข้อมูลที่มีความเร่งรีบ (ด้านขวามือของแกนนอน) ย่อมทำให้เกิดหน่วยข้อมูลเสียเพิ่มขึ้น (ดังภาพที่ 52) แต่เมื่อนำอัตราการเกิดหน่วยข้อมูลเสียนี้ไปเปรียบเทียบกับอัตราการเกิดหน่วยข้อมูลเสียก่อนที่จะผิ่กส่วนของการประหยัดพลังงาน เราจะเห็นว่าเมื่อหน่วยข้อมูลคิวโอเอสมีความเร่งรีบมากขึ้น การเพิ่มขึ้นของหน่วยข้อมูลเสียกลับมีค่าลดลง ดังภาพที่ 55 ที่เป็นเช่นนี้เพราะสำหรับวิธี HSNQ และ LI ความเร่งรีบของข้อมูลที่มากขึ้นจะทำให้หน่วยข้อมูลคิวโอเอสต้องส่งด้วยอัตราบิตที่สูงขึ้นเพื่อจะใช้เวลาในการส่งสำเร็จให้น้อยลงจนทันเส้นตาย การที่ใช้เวลาในการส่งน้อยลงทำให้หน่วยข้อมูลอื่นๆ ไม่ต้องรอคิวในการส่งนาน ซึ่งหน่วยข้อมูลจำนวนหนึ่งที่รอคิวก็จะเป็นหน่วยข้อมูลคิวโอเอส ดังนั้นหน่วยข้อมูลคิวโอเอสเหล่านี้ก็จะได้รับช่องสัญญาณเร็วขึ้น ส่วนสำหรับวิธี QB, RRP และ LS ความเร่งรีบของข้อมูลที่มากขึ้นจะทำให้หน่วยข้อมูลคิวโอเอสต้องส่งด้วยอัตราบิตที่สูงขึ้นเพื่อจะใช้เวลาในการส่งสำเร็จให้น้อยลงจนทันเส้นตาย การที่อัตราบิตในการส่งหน่วยข้อมูลคิวโอเอสสูงขึ้นก็จะทำให้อัตราบิตที่ใช้ในการส่งหน่วยข้อมูลนอนคิวโอเอสสูงขึ้นตาม (อัตราบิตของหน่วยข้อมูลนอนคิวโอเอสได้จากค่าเฉลี่ยอัตราบิตของหน่วยข้อมูลนอนคิวโอเอส) การที่ทั้งหน่วยข้อมูลคิวโอเอสและนอนคิวโอเอสใช้เวลาในการส่งน้อยลงทำให้หน่วยข้อมูลอื่นๆ ไม่ต้องรอคิวในการส่งนาน ซึ่งหน่วยข้อมูลจำนวนหนึ่งที่รอคิวก็จะเป็นหน่วยข้อมูลคิวโอเอส ดังนั้นหน่วยข้อมูลคิวโอเอสเหล่านี้ก็จะได้รับช่องสัญญาณเร็วขึ้นเช่นกัน



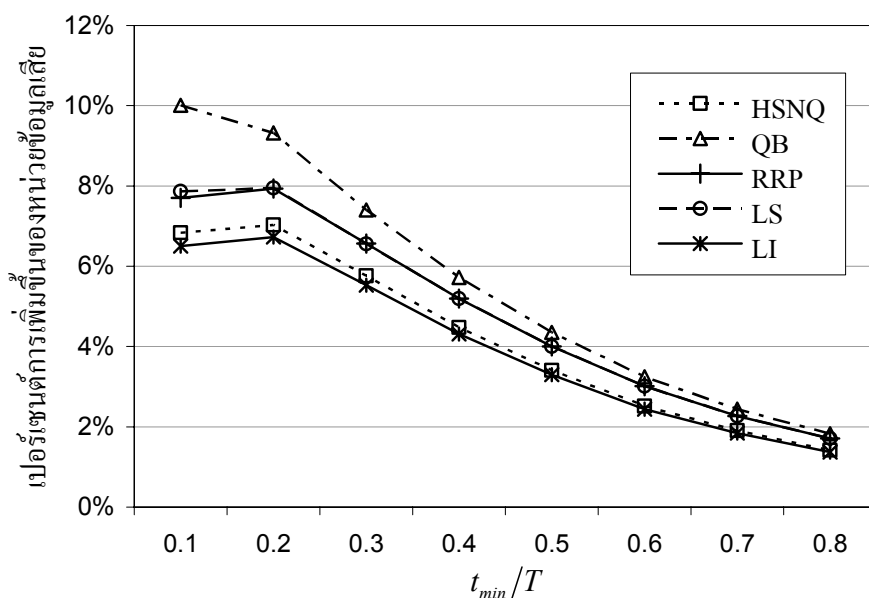
ภาพที่ 52 เปอร์เซนต์การเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสโดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) เมื่อกำหนดให้ $\alpha/\mu_{max} = 0.2$, $n_{QoS}/n_{QoS} = 1$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 53 เปอร์เซนต์การใช้พลังงานต่อบิต เมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงาน (ตอนที่ยังไม่ผนวกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52



ภาพที่ 54 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52

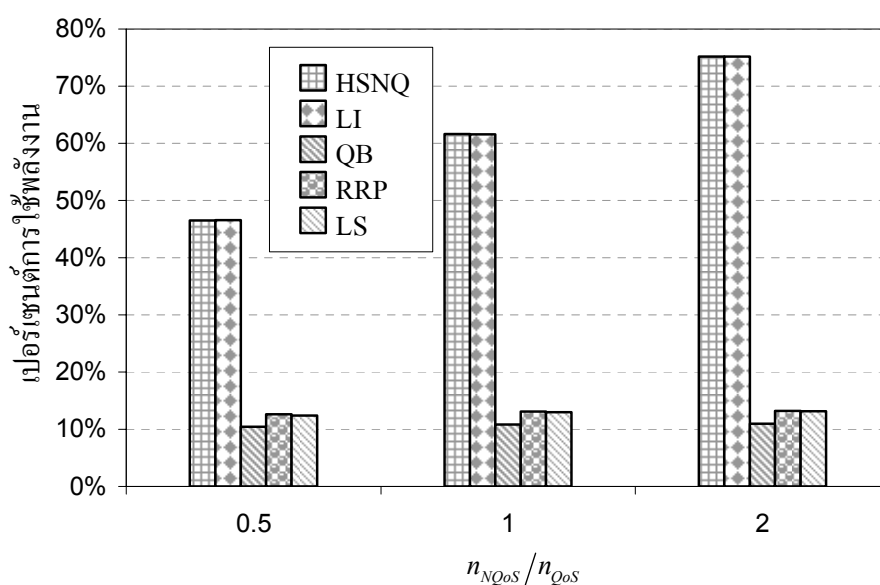


ภาพที่ 55 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่ความเร่งรีบของหน่วยข้อมูลคิวโอเอสต่างๆ (t_{min}/T) ตามข้อกำหนดเดียวกับภาพที่ 52

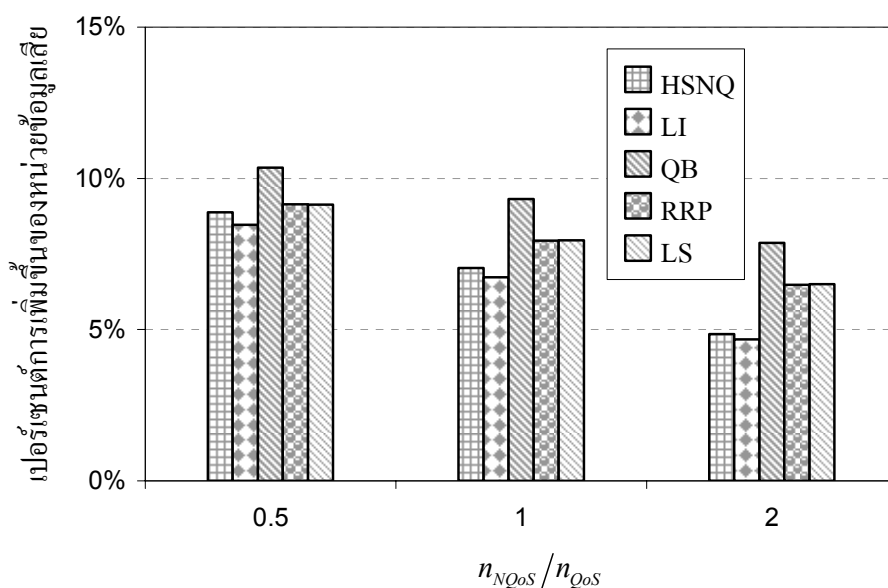
เมื่อข้อมูลมีความเร่งรีบมากขึ้น แนวคิดประเภทแรกคือแนวคิดในการประหยัดพลังงานที่ต้องมีผลกระทบต่อค่าความหน่วงเวลาเฉลี่ยน้อยที่สุด (วิธี HSNQ และ LI) จะใช้พลังงานมากกว่าแนวคิดที่สองคือ แนวคิดที่ยินยอมให้หน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดความเร็วในการส่งหน่วยข้อมูลคิวโอเอส (วิธี QB, RRP และ LS) โดยพลังงานที่ถูกใช้ในปริมาณมากกว่าเกือบเท่าตัวนี้ ไม่ก่อให้เกิดประโยชน์แต่อย่างใดดังภาพที่ 52 และ 54 เนื่องจากวิธี HSNQ และ LI ใช้อัตราบิตสูงสุดในการส่งทุกหน่วยข้อมูลคิวโอเอส การส่งด้วยอัตราบิตสูงสุดนี้ทำให้สูญเสียพลังงานมาก และเมื่อส่งออกหน่วยข้อมูลคิวโอเอสอย่างรวดเร็วทำให้เหลือช่องว่างมากขึ้น ซึ่งช่องว่างเหล่านี้ไม่ก่อให้เกิดประโยชน์แต่อย่างใด

ในกรณีการแปรค่าสัดส่วนต่างๆ ระหว่างหน่วยข้อมูลคิวโอเอสต่อคิวโอเอส (n_{NQoS}/n_{QoS})

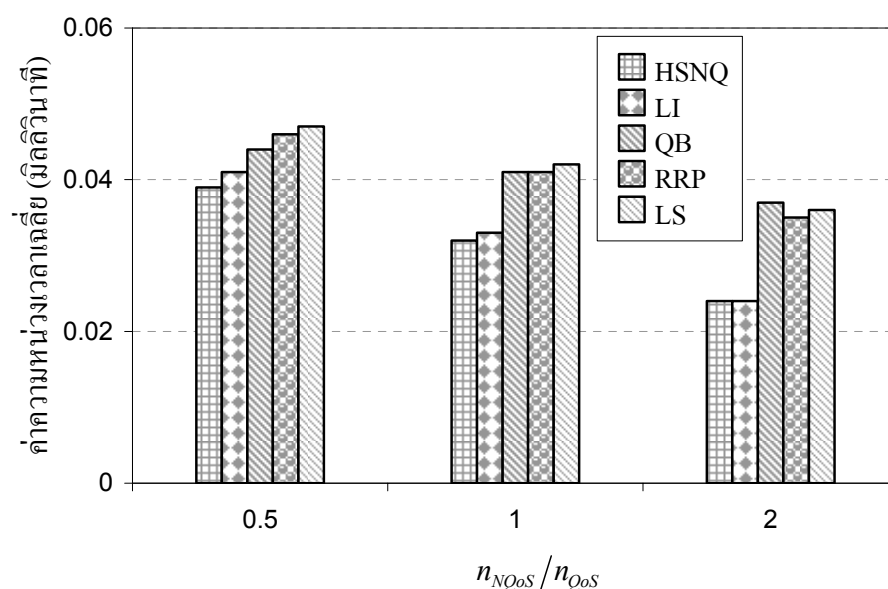
ที่สัดส่วนระหว่างหน่วยข้อมูลคิวโอเอสต่อหน่วยข้อมูลคิวโอเอสมีค่าเปลี่ยนไป จะส่งผลกระทบต่อการใช้พลังงาน อัตราการเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส และค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลคิวโอเอส ในกรณีโหนดแบบบางแสดงดังภาพที่ 56, 57 และ 58 ตามลำดับ และในกรณีโหนดคับคั่งแสดงดังภาพที่ 59, 60 และ 61 ตามลำดับ



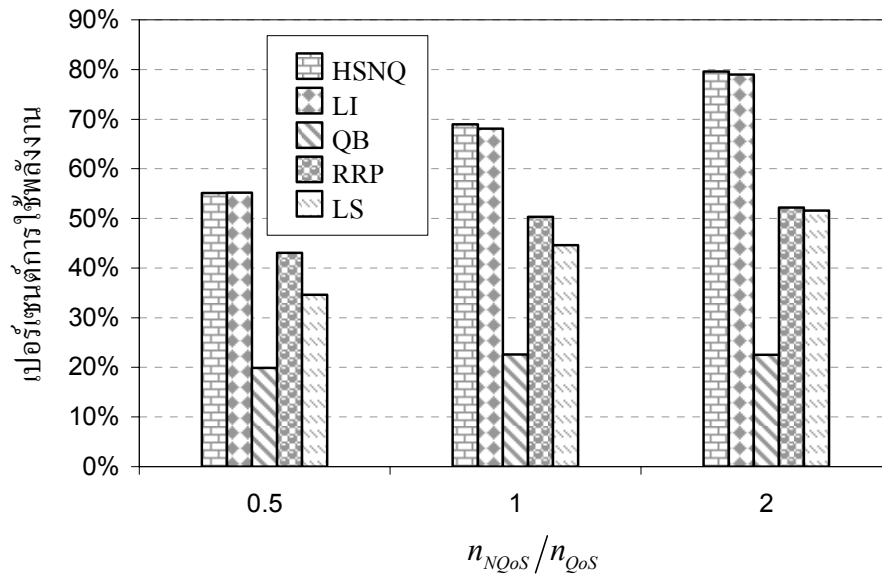
ภาพที่ 56 เปอร์เซนต์การใช้พลังงานเมื่อเปรียบเทียบกับตอนที่ยังไม่มีนักวิธีประหยัดพลังงาน (ตอนที่ยังไม่มีนักวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.2$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



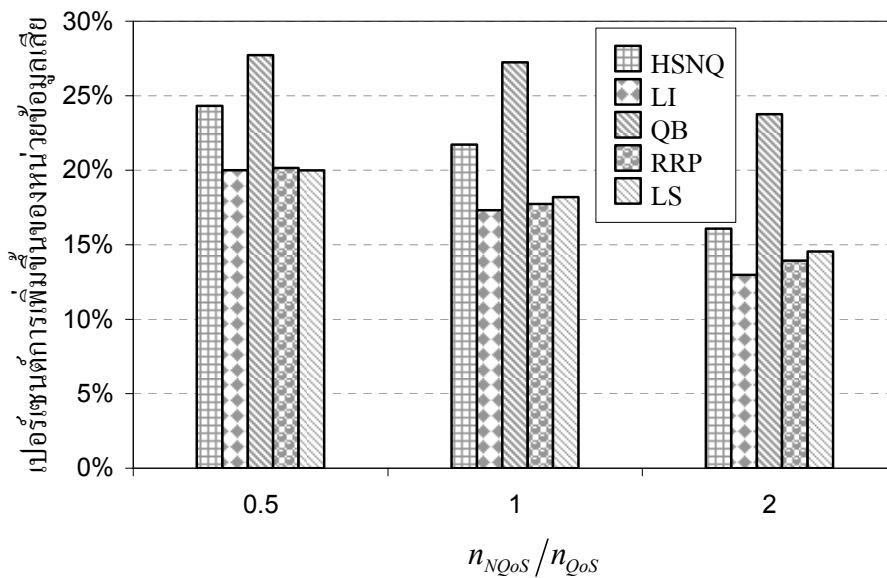
ภาพที่ 57 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคีย์ไอเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่ผนวกวิธีประหยัดพลังงานโดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคีย์ไอเอสต่อหน่วยข้อมูลคีย์ไอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 56



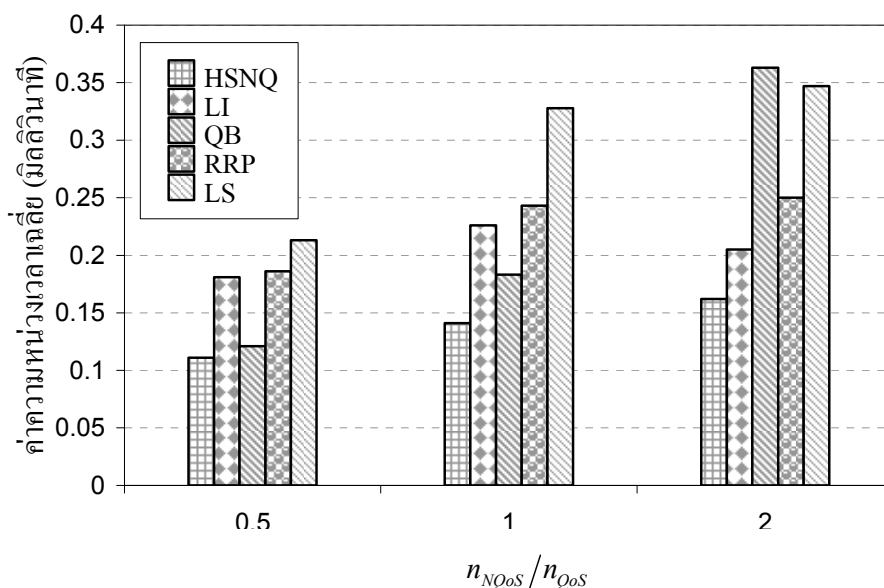
ภาพที่ 58 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคีย์ไอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคีย์ไอเอสต่อหน่วยข้อมูลคีย์ไอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 56



ภาพที่ 59 เปอร์เซนต์การใช้พลังงานเมื่อเปรียบเทียบกับตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงาน (ตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงานจะใช้พลังงานที่ 100%) โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลอนคิโวเอสต่อหน่วยข้อมูลคิโวเอส (n_{NQoS}/n_{QoS}) เมื่อกำหนดให้ $\alpha / \mu_{max} = 0.6$, $t_{min}/T = 0.2$ และจ่ายกำลังงานคงที่ 7.17 วัตต์ ที่รัศมีส่งข้อมูล 32 เมตร



ภาพที่ 60 เปอร์เซนต์การเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิโวเอสเมื่อเปรียบเทียบกับตอนที่ยังไม่มีผนวกวิธีประหยัดพลังงาน โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลอนคิโวเอสต่อหน่วยข้อมูลคิโวเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 59



ภาพที่ 61 ค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส โดยแสดงผลที่สัดส่วนต่างๆ ระหว่างหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอส (n_{NQoS}/n_{QoS}) ตามข้อกำหนดเดียวกับภาพที่ 59

ที่ปริมาณโหลดเบาบางและคับคั่ง เมื่อมีปริมาณหน่วยข้อมูลคิวโอเอสน้อยลงจะทำให้อัตราการเกิดหน่วยข้อมูลเสียมีค่าน้อยลง ดังภาพที่ 57 และ 60 เนื่องจากหน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดอัตราบิดโดยถ้ามีช่องเวลาว่างก็จะกำหนดอัตราบิดให้ช้าลง ดังนั้นการที่มีหน่วยข้อมูลคิวโอเอสจำนวนมาก ก็มักจะเกิดเหตุการณ์ที่ระบบพยายามส่งหน่วยข้อมูลด้วยอัตราบิดที่ต่ำที่สุดที่ทำให้หน่วยข้อมูลคิวโอเอสส่งได้ทันเส้นตาย แต่การส่งด้วยอัตราบิดที่ต่ำนี้จะส่งผลกระทบให้อัตราการเกิดหน่วยข้อมูลเสียเพิ่มขึ้น

การที่มีหน่วยข้อมูลนอนคิวโอเอสต้องการส่งจะแสดงถึงการรับประกันว่าจะต้องมีหน่วยข้อมูลที่สามารถส่งสำเร็จได้ไม่น้อยกว่าปริมาณหน่วยข้อมูลนอนคิวโอเอสเหล่านั้น ดังนั้นเมื่อมีหน่วยข้อมูลนอนคิวโอเอสมากขึ้นก็แสดงว่าจะต้องใช้พลังงานในการส่งข้อมูลเหล่านี้มากขึ้น ดังภาพที่ 56 และ 59

จากภาพที่ 58 และ 61 ค่าเฉลี่ยความหน่วงเวลาของหน่วยข้อมูลนอนคิวโอเอสมีแนวโน้มที่ต่างกันคือ ในกรณีโหลดเบาบาง เมื่อสัดส่วนของหน่วยข้อมูลนอนคิวโอเอสต่อหน่วยข้อมูลคิวโอเอสเพิ่มขึ้น ค่าความหน่วงเวลาเฉลี่ยมีแนวโน้มว่าน้อยลง แต่ในกรณีโหลดคับคั่ง ค่าความหน่วงเวลาเฉลี่ยมีแนวโน้มว่าเพิ่มขึ้น ด้วยเหตุผลที่ว่า กรณีโหลดคับคั่งมักมีหน่วยข้อมูลจำนวนมากรอคิวส่งจำนวนมาก จึงทำให้ค่าความหน่วงเวลาเฉลี่ยสูงขึ้น และยัง n_{NQoS}/n_{QoS} มากก็แสดงว่ามีข้อมูลที่จะส่งสำเร็จได้มาก

จึงยังเกิดค่าความหน่วงเวลาเฉลี่ยสูงขึ้นอีก แต่ในกรณีโหนดเบาบางส่วนใหญ่แล้วช่องสัญญาณจะว่าง หน่วยข้อมูลนอนคิวโอเอสจึงมักไม่ต้องรอคิว จึงทำให้หน่วยข้อมูลนอนคิวโอเอสมีค่าความหน่วงเวลาเฉลี่ยน้อย และจากเหตุผลข้างต้นหน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดอัตราบิตโดยถ้ามีช่องเวลาว่างก็จะกำหนดอัตราบิตให้ต่ำลง แสดงว่าที่หน่วยข้อมูลคิวโอเอสน้อยๆ หรือก็คือสัดส่วน n_{NQoS}/n_{QoS} มาก ก็มักจะไม่ว่างบิตให้ส่งด้วยความเร็วที่ช้าลง

สำหรับวิธี HSNQ และ LI เมื่อสัดส่วน n_{NQoS}/n_{QoS} มากขึ้นจะใช้พลังงานในการส่งที่มากขึ้นอย่างชัดเจน ดังภาพที่ 56 และ 59 เนื่องจากอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลนอนคิวโอเอสเป็นอัตราบิตสูงสุดจึงใช้พลังงานสูงสุดในการส่งทุกๆ หน่วยข้อมูลนอนคิวโอเอส ส่วนวิธีอื่นๆ ที่สัดส่วน n_{NQoS}/n_{QoS} เปลี่ยนไปจะใช้พลังงานต่างกันไม่มากเนื่องจากอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสจะขึ้นอยู่กับหน่วยข้อมูลคิวโอเอส

ที่โหนดกลับทั้งวิธี QB จะใช้พลังงานน้อยมากเมื่อเทียบกับวิธีอื่นๆ ดังภาพที่ 59 แต่การใช้พลังงานที่น้อยนี้ทำให้อัตราบิตที่ใช้ในการส่งมีค่าต่ำ ส่งผลให้เกิดหน่วยข้อมูลเสียจำนวนมากและความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอสก็มีค่ามากด้วย ดังภาพที่ 60 และ 61

การวิเคราะห์ผลด้วยตัววัดประสิทธิภาพ

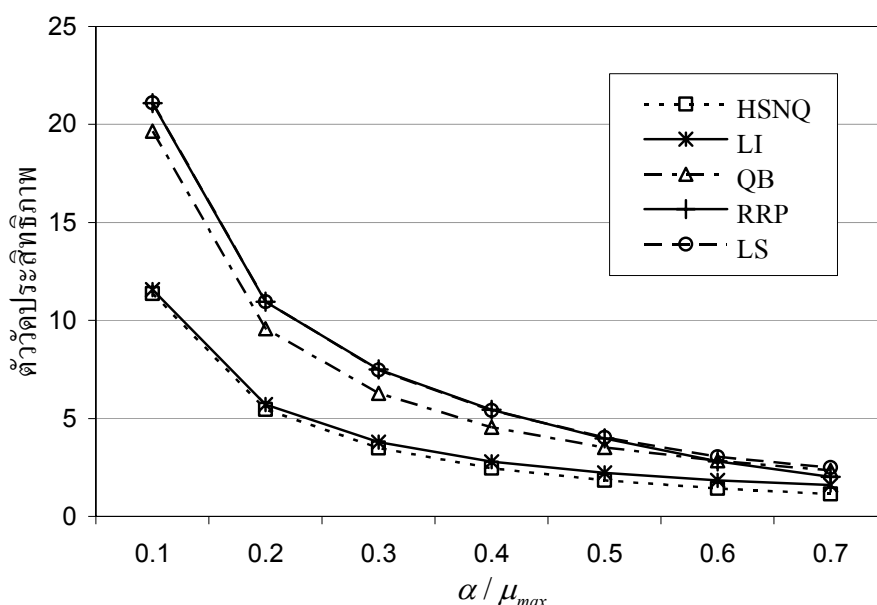
ตามที่ทราบกันดีว่า การประหยัดพลังงานย่อมสูญเสียประสิทธิภาพไปบางส่วน แต่เท่าไรคือคุ้มค่า เราจึงกำหนดตัววัดประสิทธิภาพ (η) ซึ่งเป็นอัตราส่วนระหว่างอัตราการลดการใช้พลังงานลงต่ออัตราการเพิ่มขึ้นของหน่วยข้อมูลเสีย (ใช้ค่าตอนไม่ผกผันวิธีประหยัดพลังงานเป็นฐาน)

$$\eta = \frac{\Delta \text{Energy per bit}}{\Delta \text{Rejected rate}} \quad (40)$$

โดย $\Delta \text{Energy per bit}$ คืออัตราส่วนของ ความแตกต่างของการใช้พลังงานต่อพลังงานในการส่งข้อมูลหนึ่งบิตขณะยังไม่ผกผันส่วนของการประหยัดพลังงาน (ความแตกต่างของการใช้พลังงาน คือความแตกต่างระหว่างพลังงานที่ใช้ในการส่งข้อมูลหนึ่งบิตขณะยังไม่ผกผันส่วนของการประหยัดพลังงาน กับพลังงานที่ใช้ในการส่งข้อมูลหนึ่งบิตเมื่อผกผันส่วนของการประหยัดพลังงาน)

$\Delta Rejected rate$ คือความแตกต่างระหว่างอัตราการเกิดหน่วยข้อมูลเสียเมื่อผนึกส่วนของการประหยัดพลังงาน กับอัตราการเกิดหน่วยข้อมูลเสียขณะยังไม่ผนึกส่วนของการประหยัดพลังงาน

ตัววัดประสิทธิภาพนี้ให้ถือว่า การประหยัดพลังงานกับอัตราการเกิดหน่วยข้อมูลเสียมีความสำคัญเท่ากัน โดยวิธีการประหยัดพลังงานจะเกิดความคุ้มค่าก็ต่อเมื่อ $\eta \gg 1$ และยังมีค่ามากยิ่งขึ้นแสดงถึงความคุ้มค่า แม้ว่า η จะวัดความคุ้มค่าจากอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส แต่ประสิทธิภาพในการส่งหน่วยข้อมูลบนคิวโอเอสก็ขึ้นอยู่กับอัตราบิตที่หน่วยข้อมูลคิวโอเอสเป็นตัวกำหนด เราจึงสามารถใช้ η เป็นตัววัดประสิทธิภาพในการส่งข้อมูล



ภาพที่ 62 ตัววัดประสิทธิภาพ (η) โดยแสดงผลที่ปริมาณโหลดต่างๆ (α / μ_{max}) ตามข้อกำหนดเดียวกับภาพที่ 34

ในกรณีโหลดไม่คับคั่ง เราจะเห็นว่าวิธีประหยัดพลังงานนี้มีความคุ้มค่าเป็นอย่างมาก และความคุ้มค่าจะน้อยลงเมื่อโหลดมีความคับคั่งมากขึ้น ดังแสดงในภาพที่ 62 เราจะเห็นว่าวิธี HSNQ และวิธี LI มีความคุ้มค่าที่น้อยที่สุดเพราะสามารถประหยัดพลังงานน้อย โดยเฉพาะอย่างยิ่งในกรณีโหลดคับคั่งขึ้น วิธี HSNQ สามารถประหยัดพลังงานได้น้อยที่สุดอีกทั้งยังทำให้เกิดหน่วยข้อมูลเสียมากกว่าวิธี RRP และ LS ดังภาพที่ 35 ทั่วๆ ไปที่ใช้พลังงานในการส่งมากกว่าเสียอีก ส่วนวิธี QB สามารถประหยัดพลังงานได้มาก แต่พลังงานที่ประหยัดได้นี้ก็เนื่องมาจากปริมาณหน่วยข้อมูลส่งสำเร็จมีจำนวนน้อยกว่า

วิธีอื่น เพราะหน่วยข้อมูลคิวโอเอสส่วนใหญ่ไม่สามารถส่งได้ทันเส้นตาย ส่วนวิธี RRP และวิธี LS มีความคุ้มค่าใกล้เคียงกันมาก และมีความคุ้มค่ามากที่สุดในกรณีที่ระบบสามารถยอมให้เกิดหน่วยข้อมูลเสียได้ (หน่วยข้อมูลคิวโอเอส) และทนต่อค่าความหน่วงเวลาเฉลี่ยที่เพิ่มขึ้นได้ (หน่วยข้อมูลนอนคิวโอเอส)

สรุป

แนวโน้มของการใช้งานระบบเครือข่ายในปัจจุบันและอนาคต จะมีทิศทางของการใช้งานร่วมกันของข้อมูลหลาย ๆ ประเภท ดังนั้นความต้องการการรับประกันคุณภาพด้านเวลาจึงเป็นสิ่งที่พึงปรารถนา ซึ่งหน่วยข้อมูลคิวโอเอสและนอนคิวโอเอสต่างก็ต้องการการส่งที่รวดเร็ว โดยเฉพาะอย่างยิ่งหน่วยข้อมูลคิวโอเอสมีเส้นตายเป็นตัวกำกับความเร่งรีบ อย่างไรก็ตามเมื่อหน่วยข้อมูลทั้งสองแบบนี้ต้องส่งผ่านช่องสัญญาณเดียวกันแล้ว การตอบสนองความต้องการดังกล่าวจะเป็นสิ่งที่ยากมาก เนื่องจากหน่วยข้อมูลคิวโอเอสก็ต้องการส่งข้อมูลให้ทันเส้นตาย ส่วนหน่วยข้อมูลนอนคิวโอเอสก็ต้องการให้เกิดค่าความหน่วงเวลาให้มีค่าน้อยที่สุด ซึ่งเมื่อฝ่ายใดฝ่ายหนึ่งได้รับช่องสัญญาณ แน่แน่นอนว่าต้องมีผลกระทบต่ออีกฝ่าย จากเหตุผลดังกล่าว งานวิจัยนี้จึงคิดค้นวิธีจัดลำดับ FEP ที่ยุติธรรมในการจัดลำดับการใช้ช่องสัญญาณและใช้ช่องสัญญาณได้อย่างมีคุณค่า รวมทั้งมีข้อดีที่แตกต่างจากงานวิจัยของ Chipalkatti et al. (1989) ตรงที่เป็นการจัดลำดับโดยปรับเปลี่ยนตามลักษณะของข้อมูลได้เอง โดยไม่จำเป็นต้องกำหนดค่าคงที่ใดๆ เพื่อเป็นตัวตัดสินใจว่าควรส่งข้อมูลประเภทใดก่อน จึงทำให้สามารถนำวิธีการจัดลำดับนี้ไปพัฒนาต่อในส่วนของการประหยัดพลังงาน เพื่อรองรับกับเครือข่ายที่มีพลังงานจำกัด ดังนั้นวิธีการประหยัดพลังงานจึงมีความจำเป็นอย่างยิ่ง แต่การประหยัดพลังงานก็จะต้องแลกด้วยประสิทธิภาพบางส่วน ซึ่งเห็นสมควรว่าคุ้มค่า

การประหยัดพลังงานด้วยการประยุกต์ใช้วิธี DMS โดยลดความเร็วในการส่งบางหน่วยข้อมูลลง ซึ่งแม้ว่าจะเลือกหน่วยข้อมูลที่คาดว่าจะเหมาะสมแล้ว แต่การลดความเร็วก็ย่อมส่งผลกระทบต่อประสิทธิภาพอย่างไม่สามารถหลีกเลี่ยงได้ ไม่ว่าจะเป็นการเพิ่มขึ้นของหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส และการเพิ่มขึ้นของค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส แต่สภาพข้อมูลที่ต่างกันผลกระทบต่อการทำงานและประสิทธิภาพก็ต่างกันด้วย

วิธีการประหยัดพลังงานในงานวิจัยชิ้นนี้แบ่งออกเป็น 2 กลุ่มใหญ่ๆ กลุ่มแรก เป็นแนวความคิดการประหยัดพลังงานที่ต้องการให้เกิดผลกระทบต่อหน่วยข้อมูลนอนคิวโอเอสน้อยที่สุด นั่นคือส่งทุกหน่วยข้อมูลนอนคิวโอเอสด้วยอัตราบิตสูงสุด โดยไม่สนใจว่าหน่วยข้อมูลคิวโอเอสมีความเร่งรีบหรือไม่ก็ตาม กลุ่มแรกได้แก่ วิธี HSNQ และวิธี LI กลุ่มที่สอง เป็นแนวความคิดการประหยัดพลังงานโดยยินยอมให้ความเร่งรีบของหน่วยข้อมูลคิวโอเอสเป็นตัวกำหนดความเร็วในการส่งให้กับหน่วยข้อมูลนอนคิวโอเอส กลุ่มที่สองได้แก่ วิธี QB, RRP, LS, RRP w/ RPS และ LS w/ RPS

วิธีการประหยัดพลังงานกลุ่มแรกจะประหยัดพลังงานได้น้อยกว่าวิธีการประหยัดพลังงานกลุ่มที่สองค่อนข้างมาก เนื่องจากพลังงานส่วนใหญ่สูญเสียไปเพื่อเร่งความเร็วสูงสุดในการส่งหน่วยข้อมูลนอนคิวโอเอสทุกหน่วย การประหยัดพลังงานที่แตกต่างกันนี้จะเด่นชัดมากในช่วงปริมาณโหลดไม่คับคั่ง โดยวิธีการประหยัดพลังงานในกลุ่มที่สองจะสามารถลดการใช้พลังงานลงได้ 90% ขณะที่วิธีการประหยัดพลังงานกลุ่มแรกสามารถลดการใช้พลังงานได้เพียง 40% ทั้งๆ ที่ผลกระทบต่อประสิทธิภาพมีค่าแทบจะไม่ต่างกัน ไม่ว่าจะเป็นค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิวโอเอส หรือจะเป็นอัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิวโอเอส ดังนั้นส่วนต่างของการประหยัดพลังงานที่น้อยกว่านี้ เป็นการใช้พลังงานที่เปล่าประโยชน์ เนื่องจากวิธีการประหยัดพลังงานกลุ่มแรกจะส่งออกหน่วยข้อมูลคิวโอเอสอย่างรวดเร็วตลอดเวลาทำให้เหลือช่วงเวลาวางจำนวนมาก ซึ่งช่วงเวลาวางเหล่านี้ไม่ก่อให้เกิดประโยชน์แต่อย่างใด ส่วนในกรณีโหลดคับคั่ง ไม่ว่าจะเป็นวิธีการประหยัดพลังงานกลุ่มแรกหรือกลุ่มที่สอง การประหยัดพลังงานด้วยการลดความเร็วก็จะส่งผลกระทบต่ออัตราการเกิดหน่วยข้อมูลเสียและค่าความหน่วงเวลาเฉลี่ยมากกว่ากรณีโหลดเบาบาง รวมทั้งสามารถประหยัดพลังงานได้น้อยลง เนื่องจากการที่มีปริมาณข้อมูลแออัด การปรับลดความเร็วจะทำให้ได้เพียงเล็กน้อย จึงประหยัดพลังงานได้ลดลง และแม้ว่าความเร็วจะลดลงเพียงเล็กน้อย แต่เนื่องจากหน่วยข้อมูลจำนวนหนึ่งกำลังรอคิวอยู่ การลดความเร็วนี้จึงส่งผลกระทบต่อประสิทธิภาพเพิ่มขึ้น

แม้ว่าวิธีการประหยัดพลังงานกลุ่มที่สอง จะกำหนดอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสจากความเร่งรีบของหน่วยข้อมูลคิวโอเอส แต่เมื่อหน่วยข้อมูลมีความเร่งรีบแปรเปลี่ยนไป ก็ไม่ได้ทำให้เกิดข้อเสียแต่อย่างใด โดยถ้าข้อมูลคิวโอเอสมีความเร่งรีบมากขึ้นก็จำเป็นต้องส่งด้วยอัตราบิตสูงขึ้น ซึ่งทำให้ต้องใช้พลังงานสูงขึ้นเพื่อรักษาระดับการเกิดหน่วยข้อมูลเสียให้ค่าน้อย และแม้ว่าความเร่งรีบที่เพิ่มขึ้นนี้ทำให้เกิดหน่วยข้อมูลเสียมากขึ้น แต่จากการสังเกตส่วนต่างของการเกิดหน่วยข้อมูลเสียระหว่างตอนที่ยังไม่ผนึกส่วนของการประหยัดพลังงานกับตอนที่ผนึกส่วนของการประหยัดพลังงาน จะเห็นว่าส่วนต่างนี้ก็กลับมีค่าลดลง เนื่องจากความเร่งรีบของหน่วยข้อมูลคิวโอเอส ทำให้ระบบต้องส่งหน่วยข้อมูลเหล่านี้เร็วขึ้น จึงเหลือช่วงเวลามากพอให้ส่งหน่วยข้อมูลคิวโอเอสอื่นๆ ได้ทันเส้นตาย ส่วนทางด้านหน่วยข้อมูลนอนคิวโอเอสก็จะได้รับการส่งเร็วขึ้น เพราะอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสได้มาจากค่าเฉลี่ยอัตราบิตของหน่วยข้อมูลคิวโอเอสในลำดับก่อนหน้า เมื่ออัตราบิตในการส่งหน่วยข้อมูลคิวโอเอสสูงขึ้น ทำให้อัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสสูงขึ้นตาม ส่งผลให้ใช้เวลาในการส่งหนึ่งหน่วยข้อมูลน้อยลง อีกทั้งหน่วยข้อมูลคิวโอเอสที่มีความเร่งรีบนี้จะใช้เวลาในการส่งน้อยเพื่อให้ทันเส้นตาย จึงทำให้หน่วยข้อมูลต่างๆ ที่กำลังรอคิว ได้รับช่องสัญญาณเร็วขึ้น จึงทำให้หน่วยข้อมูลนอนคิวโอเอสมีค่าความหน่วงเวลาเฉลี่ยลดลง ส่วนผลจากวิธีการประหยัดพลังงานกลุ่มแรกจะมีค่าใกล้เคียงกับกลุ่มที่สอง นั่นคืออัตราการเกิดหน่วยข้อมูลเสีย

และค่าความหน่วงเวลาเฉลี่ยมีค่าแทบจะไม่แตกต่างจากวิธีการประหยัดพลังงานกลุ่มที่สอง แต่กลับใช้พลังงานสูงกว่าเกือบเท่าตัว

ในกรณีที่สัดส่วนระหว่างปริมาณหน่วยข้อมูลนอนคิโวเอสต่อปริมาณหน่วยข้อมูลคิโวเอสมีการแปรเปลี่ยน ก็ไม่ได้ทำให้การทำงานแย่งแย่งแต่อย่างใด โดยแต่ละวิธีการประหยัดพลังงานของกลุ่มที่สองจะได้ผลที่ใกล้เคียงกันในแต่ละสัดส่วน แต่สำหรับวิธีการประหยัดพลังงานในกลุ่มแรกเมื่อสัดส่วนเปลี่ยนไป ผลที่ได้ก็เปลี่ยนอย่างชัดเจนกว่าวิธีการประหยัดพลังงานกลุ่มแรก คือถ้าสัดส่วนของหน่วยข้อมูลนอนคิโวเอสเพิ่มขึ้น ก็จะใช้พลังงานมากขึ้น เนื่องจากการที่มีหน่วยข้อมูลนอนคิโวเอสต้องการส่ง จะแสดงถึงการรับประกันว่าจะต้องมีหน่วยข้อมูลที่สามารถส่งสำเร็จได้ ไม่น้อยกว่าปริมาณหน่วยข้อมูลนอนคิโวเอสเหล่านั้น ดังนั้นเมื่อมีหน่วยข้อมูลนอนคิโวเอสมากขึ้นก็แสดงว่าต้องใช้พลังงานในการส่งข้อมูลเหล่านั้นมากขึ้น วิธีประหยัดพลังงานกลุ่มแรกจะส่งทุกหน่วยข้อมูลนอนคิโวเอสด้วยอัตราบิตสูงสุด จึงทำให้ประหยัดพลังงานได้น้อยลงอย่างชัดเจนเมื่อสัดส่วนของหน่วยข้อมูลนอนคิโวเอสเพิ่มขึ้น และการที่วิธีการประหยัดพลังงานกลุ่มแรกใช้พลังงานมากจึงเป็นผลทำให้เกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิโวเอส และค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลนอนคิโวเอสมีค่าน้อยกว่าวิธีประหยัดพลังงานกลุ่มที่สอง

จากที่กล่าวทั้งหมดข้างต้นจะแสดงให้เห็นว่า วิธีการประหยัดพลังงานในงานวิจัยชิ้นนี้สามารถประหยัดพลังงานได้อย่างมากในกรณีที่ปริมาณโหลดไม่คับคั่ง ด้วยวิธีการประหยัดพลังงานกลุ่มที่สองสามารถประหยัดพลังงานลงได้กว่าร้อยละ 90 หรือใช้พลังงานเพียง 10% จากเดิมที่เคยใช้ (ตอนที่ยังไม่ผนึกส่วนของการประหยัดพลังงาน) อีกทั้งยังผลกระทบต่ออัตราการเกิดหน่วยข้อมูลเสียของหน่วยข้อมูลคิโวเอสและค่าความหน่วงเวลาเฉลี่ยของหน่วยข้อมูลคิโวเอสเพียงเล็กน้อย โดยไม่ว่าลักษณะข้อมูลมีความเร่งรีบ หรือสัดส่วนระหว่างหน่วยข้อมูลนอนคิโวเอสต่อหน่วยข้อมูลคิโวเอสมีการเปลี่ยนแปลง ก็ไม่ได้ทำให้ระบบแย่งแย่งแต่อย่างใด

การตัดสินใจว่าวิธีใดมีความเหมาะสมมากกว่านั้น ก็ขึ้นอยู่กับว่าจะนำไปประยุกต์ใช้กับระบบใด นั่นคือต้องดูข้อจำกัดเช่น ความสำคัญของพลังงาน การรับประกันคุณภาพในการส่งข้อมูล ระบบสามารถรองรับความซับซ้อนได้แค่ไหน ปริมาณข้อมูลบนเครือข่าย เป็นต้น ซึ่งถ้าระบบสามารถรองรับความซับซ้อนได้มาก วิธี RRP และ LS จะมีความเหมาะสมเป็นอย่างยิ่ง เนื่องจากประหยัดพลังงานลงได้มาก อีกทั้งยังมีผลกระทบต่อประสิทธิภาพเพียงเล็กน้อย (สำหรับวิธี LI จากการทดลองแม้ว่าจะมีกระบวนการที่ซับซ้อนอีกทั้งยังประหยัดพลังงานได้น้อย แต่กลับมีค่าความหน่วงเวลาเฉลี่ยและอัตราการเกิดหน่วยข้อมูลเสียใกล้เคียงกับวิธี RRP) แต่ถ้าระบบไม่สามารถรองรับความซับซ้อนได้มาก วิธี

HSNQ และ QB น่าจะมีความเหมาะสม แต่สองวิธีนี้มีความแตกต่างกัน โดยถ้าระบบไม่สามารถยอมรับความหน่วงเวลาเฉลี่ยที่เพิ่มขึ้นได้ก็ควรใช้วิธี HSNQ แต่ถ้าระบบไม่สามารถยอมรับความหน่วงเวลาที่เพิ่มขึ้นจากกระบวนการประหยัดพลังงาน วิธี QB น่าจะเป็นอีกทางเลือกหนึ่ง

เอกสารและสิ่งอ้างอิง

- นฤมล ปฐมมาณิศ และ ศิริวัฒน์ พูนวสิน. 2548. การส่งข้อมูลต่างลักษณะผ่านระบบไร้สายที่มีพลังงานจำกัด, น. 781-784. รายงานการประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 28 (EECON 28). ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.
- Chipalkatti, R., J. F. Kurose and D. Towsley. 1989. Scheduling policies for real-time and non-real-time traffic in a statistical multiplexer. **Proceedings of IEEE Infocom'89**. 3: 774-783.
- Culler, D., D. Estrin and M. Srivastava. 2004. Guest Editors' Introduction: Overview of Sensor Networks. **The IEEE Computer Society**. 37(8): 41-49.
- Demers, A., S. Keshav and S. Shenker. 1990. Analysis and simulation of a fair queuing algorithm. **Journal of Internetworking Research and Experience**. 1(1): 3-26.
- Estrin, D., L. Girod, G. Pottie and M. Srivastava. 2001. Instrumenting the World with Wireless Sensor Networks. **International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)**. 4: 2033-2036.
- Haykin, S. 2001. **Communication system**. 4th ed. John Wiley & Sons, Inc., USA.
- Intanagonwiwat, C., R. Govindan, D. Estrin, J. Heidemann and F. Silva. 2003. Directed diffusion for wireless sensor networking. **IEEE/ACM Transactions on Networking**. 11(1): 2-16.
- Lewis, F. L. 2004. Wireless sensor networks. In D. J. Cook and S. K. Das, eds. **Smart Environments: Technologies, Protocols and Applications**. John Wiley & Sons, Inc., New York.
- Liu, C. L. and J. W. Layland. 1973. Scheduling algorithms for multiprogramming in hard real-time traffic environment. **Journal of the Association for Computing Machinery**. 20(1)

- Parekh, A. K. and R. G. Gallager. 1993. A generalized processor sharing approach to flow control in integrated services network: The single-node case. **IEEE/ACM Transactions on Networking**. 1(3): 344-357.
- Poonvasin, S. 2003. Speed scheduling of fixed-priority real-time jobs with maximum power awareness. **IEEE TENCON'03**.
- Raghunathan, V., C. Schurgers, S. Park and M. B. Srivastava. 2002. Energy aware wireless microsensor networks. **IEEE Signal Processing Magazine**. 19(2): 40-50.
- Raghunathan, V., S. Ganeriwal, C. Schurgers and M. B. Srivastava. 2002. E2WFQ: An energy efficient fair scheduling policy for wireless systems. **International Symposium of Low Power Electronics and Design (ISLPED'02)**. 1: 30-35.
- Schurgers, C., O. Aberthorne and M. B. Srivastava. 2001. Modulation scaling for energy aware communication systems. **Proc. ACM ISLPED'01**. 1: 96-99.
- Schurgers, C., V. Raghunathan and M. B. Srivastava. 2001. Modulation Scaling for Real-Time Energy Aware Packet Scheduling. **Global Communications Conference (GlobeCom'01)**. 6: 3653-3657
- Shah, R. C. and J. M. Rabaey. 2002. Energy aware routing for low energy ad hoc sensor networks. **IEEE Wireless Communication and Networks**. 1: 350-355.
- Towsley, D. 1993. Providing quality of service in packet switched networks, pp. 560-586. In D. Lorenzo and N. Randolph, eds. **Performance Evaluation of Computer and Communication Systems**. Springer-Verlag. Softcover.
- Yates, R. D. And D. J. Goodman. 1999. **Probability and stochastic processes**. John Wiley & Sons, Inc., USA.

- Younis, M., K. Akkaya, M. Eltoweissy, A. Wadaa. 2004. On handling QoS traffic in wireless sensor networks. **Proceedings of the 37th Annual Hawaii International Conference on 5-8 Jan. 2004**. 1: 292 – 301.
- Yu, Y. and V. K. Prasanna. 2003. Energy-balanced multi-hop packet transmission in wireless sensor networks. **IEEE GlobeCom**. 22: 480-486.
- Zhang, H. 1995. Service disciplines for guaranteed performance service in packet-switching networks. **Proceedings of IEEE**. 83(10): 1374-1396.

ภาคผนวก

ตัวอย่างวิธีการจัดลำดับ

ตัวอย่างการจัดลำดับหน่วยข้อมูลด้วยวิธี FEF

สมมุติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 17 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราบิตของช่องสัญญาณขาออก 1 kbps

ตารางผนวกที่ 1 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี FEF และ FEP

หน่วยข้อมูลที่	ชนิดของหน่วยข้อมูล	เวลาที่หน่วยข้อมูลมาถึง (วินาที)	จำนวนบิตของหน่วยข้อมูล (กิโลบิต)	เส้นตาย (วินาที)
1	NQoS	0	2	-
2	QoS	1	1	5
3	QoS	2	2	4
4	NQoS	6	3	-
5	QoS	6.5	1	20
6	NQoS	7	3	-
7	QoS	8	1	11
8	QoS	9	1	12
9	QoS	10	2	12.5
10	NQoS	16	1	-

การทำงานของวิธี FEF ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ จึงเริ่มส่งหน่วยข้อมูลนี้ทันที เนื่องจากหน่วยข้อมูลนี้ยาว 2 kbits และมีอัตราบิตที่ 1 kbps จึงใช้เวลาในการส่งออก 2 วินาที หน่วยข้อมูลนี้จะเริ่มส่งที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 2

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงทำการเก็บหน่วยข้อมูลที่ 2 ลงบัฟเฟอร์คิวโอเอส (QoS buffer)

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด และหน่วยข้อมูลที่ 1 ส่งสำเร็จพอดี ช่องสัญญาณจึงว่าง ณ เวลานี้ในบัฟเฟอร์คิวโอเอสมีอยู่สองหน่วยข้อมูล โดยหน่วยข้อมูลที่ 3 อยู่ในลำดับแรก และหน่วยข้อมูลที่ 2 อยู่ในลำดับต่อมา (เรียงลำดับตามเส้นตาย) ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 3 ที่วินาทีที่ 2 และจะเสร็จสิ้นที่วินาทีที่ 4

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 3 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 2 ต่อโดยเริ่มส่งที่วินาทีที่ 4 และจะเสร็จสิ้นที่วินาทีที่ 5

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 2 ส่งสำเร็จ และ ณ เวลานี้ช่องสัญญาณว่าง

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด ให้เริ่มส่งหน่วยข้อมูลที่ 4 ที่วินาทีที่ 6 และจะเสร็จสิ้นที่วินาทีที่ 9

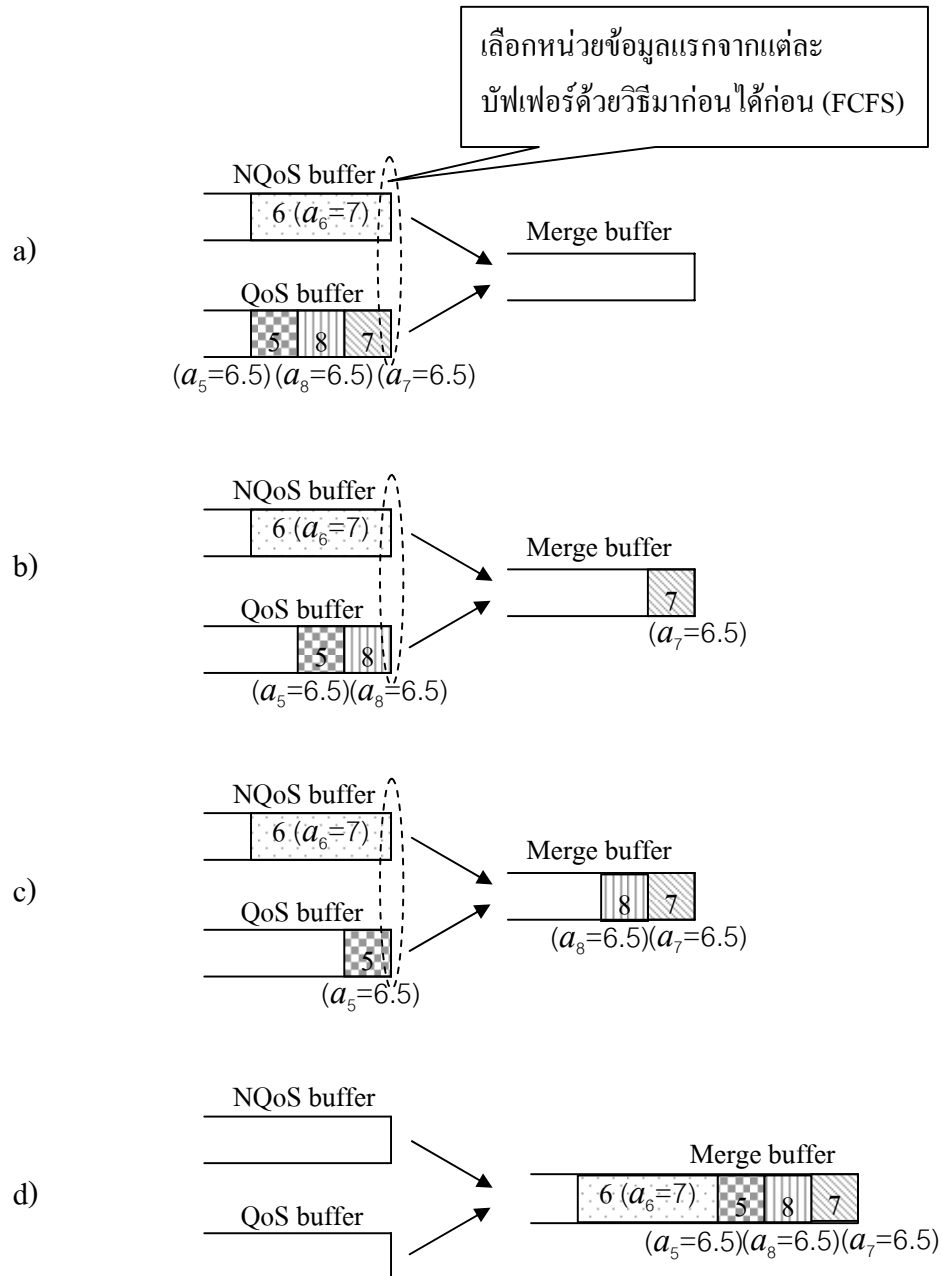
ณ เวลา 6.5 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 5 ลงในบัฟเฟอร์คิวโอเอส

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 6 ลงในบัฟเฟอร์คิวโอเอสคิวโอเอส

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 7 ลงในบัฟเฟอร์คิวโอเอส ณ เวลานี้ในบัฟเฟอร์คิวโอเอสมีหน่วยข้อมูลที่ 6 และในบัฟเฟอร์คิวโอเอสมีหน่วยข้อมูลที่ 7 และ 5 ตามลำดับการรอส่ง โดยหน่วยข้อมูลที่ 7 ทำการแทรกหน้าหน่วยข้อมูลที่ 5 ดังนั้นหน่วยข้อมูลที่ 7 จึงถูกเปลี่ยนค่าเวลาที่หน่วยข้อมูลมาถึง จากวินาทีที่ 8 เป็น 6.5

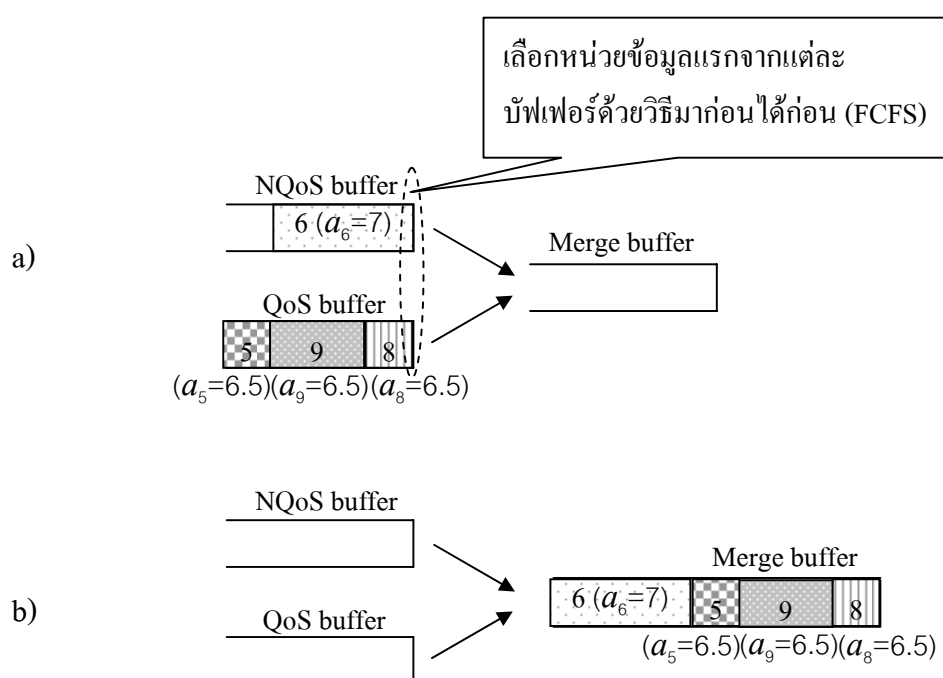
ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด และช่องสัญญาณว่างพอดี ณ เวลานี้ในบัฟเฟอร์คิวโอเอสมีหน่วยข้อมูลที่ 6 ซึ่งมีค่าเวลาการมาถึง (arrival time) ที่วินาทีที่ 7 และในบัฟเฟอร์คิวโอเอสมีหน่วยข้อมูลที่ 7, 8 และ 5 ตามลำดับการรอส่ง โดยหน่วยข้อมูลที่ 8 ทำการแทรกหน้าหน่วยข้อมูลที่ 5 ดังนั้นหน่วยข้อมูลที่ 8 จึงถูกเปลี่ยนค่าเวลาการมาถึง จากวินาทีที่ 9 เป็น 6.5 ต่อมาเข้าสู่กระบวนการ

เลือกหน่วยข้อมูลที่อยู่ในตำแหน่งหัวของบัฟเฟอร์ทั้งสองเพื่อใส่ลงบัฟเฟอร์รวม เป็นดังภาพผนวกที่ 1 โดยกำหนดให้ a_i คือค่าเวลาการมาถึงโหนด (arrival time) ของหน่วยข้อมูล i จากภาพผนวกที่ 1-d หน่วยข้อมูลที่ 7 เป็นหน่วยข้อมูลที่อยู่ลำดับแรกของบัฟเฟอร์รวม (merge buffer) และขณะนี้ช่องสัญญาณว่าง ให้เริ่มส่งหน่วยข้อมูลที่ 7 ที่วินาทีที่ 9 และจะเสร็จสิ้นที่วินาทีที่ 10



ภาพผนวกที่ 1 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEF ณ วินาทีที่ 9

ณ เวลา 10 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด และช่องสัญญาณว่างพอดี เนื่องจากหน่วยข้อมูลที่ 7 ส่งสำเร็จ ณ เวลานี้ในบัฟเฟอร์คิวโอเอสซึ่งได้จัดลำดับแล้ว มีหน่วยข้อมูลที่ 8, 9 และ 5 ตามลำดับการรอส่ง โดยหน่วยข้อมูลที่ 9 ทำการแทรกหน้าหน่วยข้อมูลที่ 5 ดังนั้นหน่วยข้อมูลที่ 9 จึงถูกเปลี่ยนค่าเวลาการมาถึง จากวินาทีที่ 10 เป็น 6.5 จากกระบวนการเลือกหน่วยข้อมูลที่อยู่ในตำแหน่งหัวของบัฟเฟอร์ทั้งสองเพื่อใส่ลงบัฟเฟอร์รวม จากภาพผนวกที่ 2 หน่วยข้อมูลที่ 8 เป็นหน่วยข้อมูลที่อยู่ลำดับแรกของบัฟเฟอร์รวม (merge buffer) และขณะนี้ช่องสัญญาณว่าง ให้เริ่มส่งหน่วยข้อมูลที่ 8 ที่วินาทีที่ 10 และจะเสร็จสิ้นที่วินาทีที่ 11



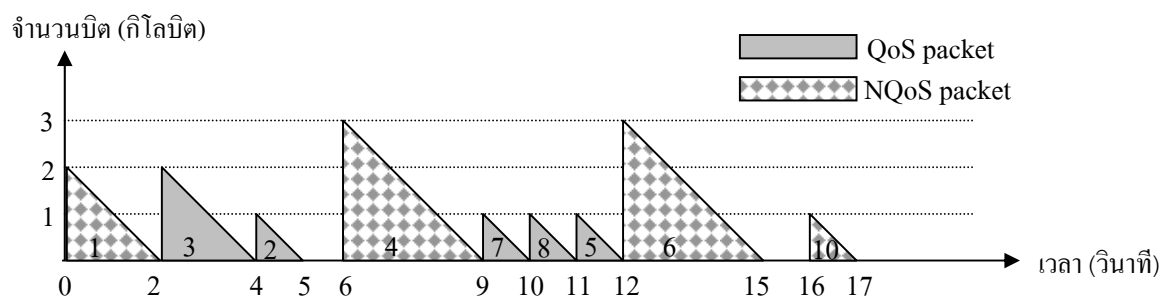
ภาพผนวกที่ 2 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEF ณ วินาทีที่ 10

ณ เวลา 11 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 9 ซึ่งอยู่ในลำดับแรกของบัฟเฟอร์รวม แต่ถ้าส่งหน่วยข้อมูลที่ 9 โดยเริ่มส่งที่วินาทีนี้ จะเสร็จสิ้นที่วินาทีที่ 13 ซึ่งเป็นเวลาที่เลยเส้นตาย (deadline) ที่ 12.5 วินาที ดังนั้นจึงทิ้งหน่วยข้อมูลนี้ไป (rejected packet) ทำให้ในบัฟเฟอร์รวมมีหน่วยข้อมูลที่ 5 อยู่ในลำดับแรกแทน เราจึงทำการส่งหน่วยข้อมูลที่ 5 โดยเริ่มส่งที่วินาทีที่ 11 และจะเสร็จสิ้นที่วินาทีที่ 12

ณ เวลา 12 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 6 ซึ่งอยู่ในลำดับแรกของบัฟเฟอร์รวม โดยเริ่มส่งที่วินาทีที่ 12 และจะเสร็จสิ้นที่วินาทีที่ 15

ณ เวลา 15 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จพอดี ขณะนี้บัฟเฟอร์ทั้งสามว่าง ช่องสัญญาณว่าง

ณ เวลา 16 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด เนื่องจากช่องสัญญาณว่าง ให้เริ่มส่งหน่วยข้อมูลที่ 10 ที่วินาทีที่ 16 และจะเสร็จสิ้นที่วินาทีที่ 17



ภาพผนวกที่ 3 แผนภาพการส่งออกหน่วยข้อมูลตัวอย่างที่ช่องสัญญาณขาออกเมื่อใช้วิธีจัดลำดับ FEF

ตัวอย่างการจัดลำดับหน่วยข้อมูลด้วยวิธี FEP

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่าน โหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 1 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราบิตของช่องสัญญาณขาออก 1 kbps

การทำงานของวิธี FEP ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 2

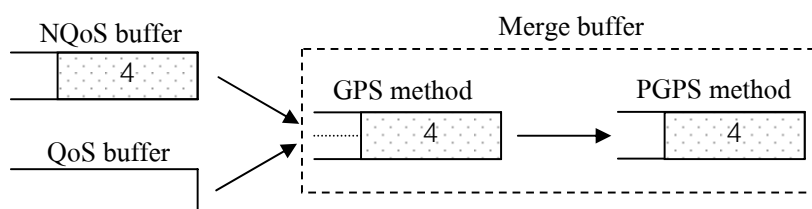
ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงทำการเก็บหน่วยข้อมูลที่ 2 ลงบัฟเฟอร์คิวโอเอส (QoS buffer)

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด และช่องสัญญาณว่างพอดี เนื่องจากหน่วยข้อมูลที่ 1 ส่งสำเร็จ ณ เวลานี้ในบัฟเฟอร์คิวโอเอสมียู่สองหน่วยข้อมูลโดยหน่วยข้อมูลที่ 3 อยู่ในลำดับแรก และหน่วยข้อมูลที่ 2 อยู่ในลำดับต่อมา (เรียงลำดับตามเส้นตาย) ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 3 ที่วินาทีที่ 2 และจะเสร็จสิ้นที่วินาทีที่ 4

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 3 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 2 ต่อโดยเริ่มส่งที่วินาทีที่ 4 และจะเสร็จสิ้นที่วินาทีที่ 5

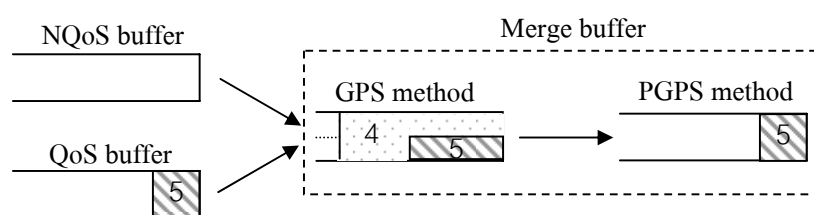
ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 2 ส่งสำเร็จ และ ณ เวลานี้ช่องสัญญาณว่าง

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด ให้เริ่มส่งหน่วยข้อมูลที่ 4 ที่วินาทีที่ 6 และจะเสร็จสิ้นที่วินาทีที่ 9



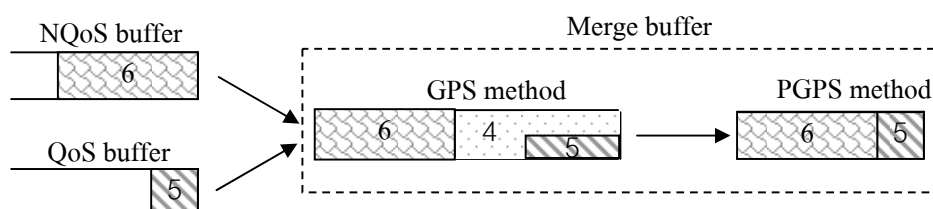
ภาพผนวกที่ 4 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 6

ณ เวลา 6.5 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 5 ลงในบัฟเฟอร์คิวโอเอส ดังภาพผนวกที่ 5 ถ้าสังเกตจะเห็นว่าในบัฟเฟอร์รวม (merge buffer) ที่การคำนวณ GPS จะดูเหมือนว่าหน่วยข้อมูลที่ 5 สามารถส่งสำเร็จได้ก่อนหน่วยข้อมูลที่ 4 แล้วในการส่งออกจริงจะต้องส่งหน่วยข้อมูลที่ 5 ก่อนหน่วยข้อมูลที่ 4 แต่เนื่องจากว่าหน่วยข้อมูลที่ 4 ได้เริ่มส่งไปแล้วตั้งแต่วันที่ 6 จึงไม่นำมาสลับลำดับการส่งออก



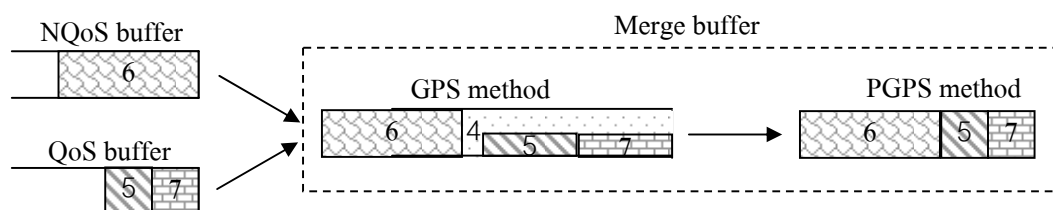
ภาพผนวกที่ 5 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 6.5

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 6 ลงในบัฟเฟอร์คิวโอเอส



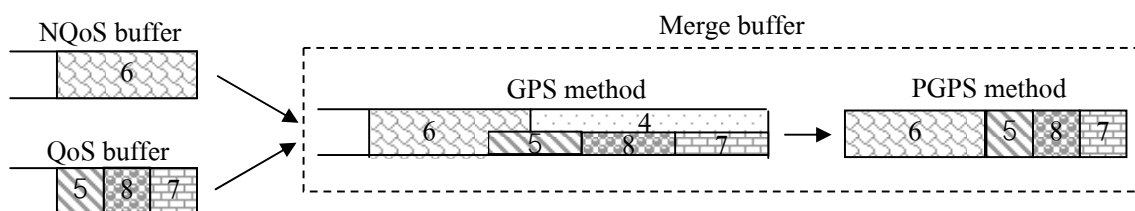
ภาพผนวกที่ 6 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 7

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด แต่เนื่องจากช่องสัญญาณยังไม่ว่าง จึงเก็บหน่วยข้อมูลที่ 7 ลงในบัฟเฟอร์คิวโอเอส และในบัฟเฟอร์คิวโอเอสมีหน่วยข้อมูลที่ 7 และ 5 ตามลำดับการรอส่ง (เรียงตามเส้นตาย)



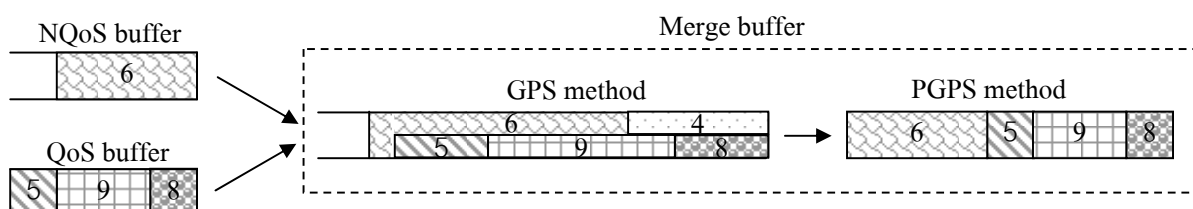
ภาพผนวกที่ 7 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด และส่งหน่วยข้อมูลที่ 4 สำเร็จพอดี ขณะนี้ช่องสัญญาณจึงว่าง ทำการจัดลำดับทั้งสามบัฟเฟอร์ จากภาพผนวกที่ 8 หน่วยข้อมูลที่ 7 เป็นหน่วยข้อมูลลำดับแรกในบัฟเฟอร์รวม (merge buffer) จึงทำการส่งหน่วยข้อมูลที่ 7 ต่อโดยเริ่มส่งที่วินาทีที่ 9 และจะเสร็จสิ้นที่วินาทีที่ 10



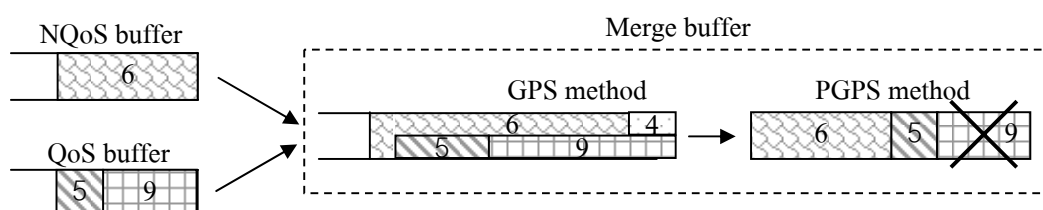
ภาพผนวกที่ 8 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 9

ณ เวลา 10 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด และส่งหน่วยข้อมูลที่ 7 สำเร็จพอดี ขณะนี้ช่องสัญญาณจึงว่าง ทำการจัดลำดับทั้งสามบัฟเฟอร์ จากภาพผนวกที่ 9 หน่วยข้อมูลที่ 8 เป็นหน่วยข้อมูลลำดับแรกในบัฟเฟอร์รวม (merge buffer) จึงทำการส่งหน่วยข้อมูลที่ 8 ต่อโดยเริ่มส่งที่วินาทีที่ 10 และจะเสร็จสิ้นที่วินาทีที่ 11



ภาพผนวกที่ 9 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 10

ณ เวลา 11 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 9 ซึ่งอยู่ในลำดับแรกของบัฟเฟอร์รวม แต่ถ้าส่งหน่วยข้อมูลที่ 9 โดยเริ่มส่งที่วินาทีนี้ จะเสร็จสิ้นที่วินาทีที่ 13 ซึ่งเป็นเวลาที่เลยเส้นตาย (deadline) ที่ 12.5 วินาที ดังนั้นจึงทิ้งหน่วยข้อมูลนี้ไป (rejected packet) ทำให้ในบัฟเฟอร์รวมมีหน่วยข้อมูลที่ 5 อยู่ในลำดับแรกแทน ดังภาพผนวกที่ 10 เราจึงทำการส่งหน่วยข้อมูลที่ 5 โดยเริ่มส่งที่วินาทีที่ 11 และจะเสร็จสิ้นที่วินาทีที่ 12

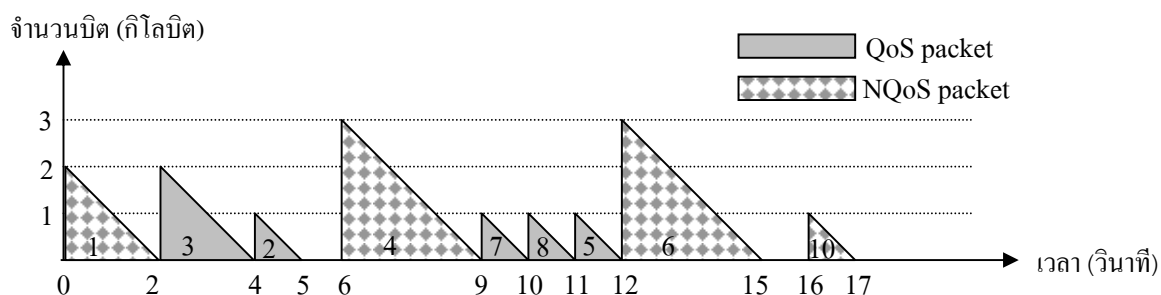


ภาพผนวกที่ 10 กระบวนการจัดลำดับทั้งสามบัฟเฟอร์ด้วยวิธี FEP ณ วินาทีที่ 11

ณ เวลา 12 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จพอดี จึงทำการส่งหน่วยข้อมูลที่ 6 ซึ่งอยู่ในลำดับแรกของบัฟเฟอร์รวม โดยเริ่มส่งที่วินาทีที่ 12 และจะเสร็จสิ้นที่วินาทีที่ 15

ณ เวลา 15 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จพอดี ขณะนี้บัฟเฟอร์ทั้งสามว่าง ช่องสัญญาณว่าง

ณ เวลา 16 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด เนื่องจากช่องสัญญาณว่าง ให้เริ่มส่งหน่วยข้อมูลที่ 10 ที่วินาทีที่ 16 และจะเสร็จสิ้นที่วินาทีที่ 17



ภาพผนวกที่ 11 แผนภาพการส่งออกหน่วยข้อมูลตัวอย่างที่ช่องสัญญาณขาออกเมื่อใช้วิธีจัดลำดับ FEP

ตัวอย่างวิธีการประหยัดพลังงาน

ตัวอย่างการประหยัดพลังงานด้วยวิธี HSNQ

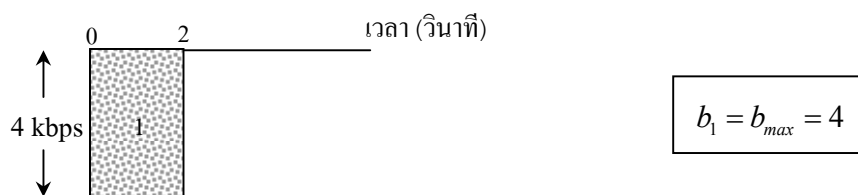
สมมุติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 2 ซึ่งข้อมูลเหล่านี้จะส่งออกจากช่องสัญญาณขาออกด้วยอัตราอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

ตารางผนวกที่ 2 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี HSNQ และ LI

หน่วยข้อมูลที่	ชนิดของหน่วยข้อมูล	เวลาที่หน่วยข้อมูลมาถึง (วินาที)	จำนวนบิตของหน่วยข้อมูล (กิโลบิต)	เส้นตาย (วินาที)
1	NQoS	0	8	-
2	QoS	1	6	6
3	QoS	2	2	4
4	NQoS	3	4	-
5	QoS	4	8	10
6	NQoS	5	2	-
7	QoS	6	4	13
8	QoS	7	6	14
9	QoS	8	11	16
10	NQoS	9	9	-

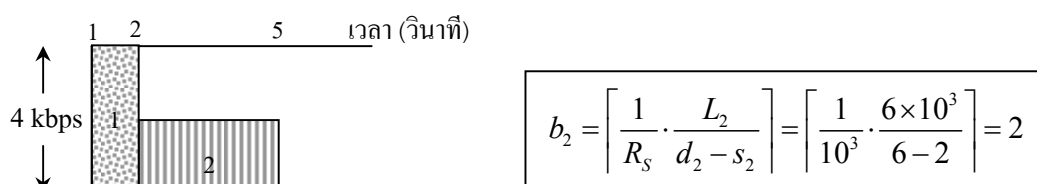
การทำงานของวิธี HSNQ ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิโอสจึงส่งด้วยอัตราบิตสูงสุด (4 kbps) และหน่วยข้อมูลนี้ยาว 8 kbits จึงใช้เวลาในการส่ง 2 วินาที ดังในภาพผนวกที่ 12 โดยเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 2



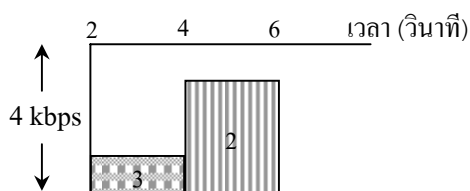
ภาพผนวกที่ 12 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิโอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 13 คือเมื่อส่งหน่วยข้อมูลที่ 2 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 2 และจะเสร็จสิ้นที่วินาทีที่ 5



ภาพผนวกที่ 13 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ จึงต้องมีการคำนวณหาอัตราบิตใหม่ให้กับหน่วยข้อมูลที่มีการเปลี่ยนลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 ดังภาพผนวกที่ 14 หน่วยข้อมูลที่ 3 จะส่งด้วยอัตราบิต 1 kbps จึงทำให้ส่งได้ทัน ซึ่งหน่วยข้อมูลที่ 3 เริ่มส่งในวินาทีที่ 2 และส่งสำเร็จได้ในวินาทีที่ 4 และหน่วยข้อมูลที่ 2 จะส่งด้วยอัตราบิต 3 kbps จึงทำให้ส่งหน่วยข้อมูลที่ 2 นี้ได้ทัน ซึ่งหน่วยข้อมูลที่ 2 เริ่มส่งในวินาทีที่ 4 และส่งสำเร็จได้ในวินาทีที่ 6

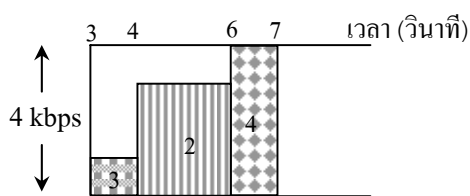


$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{2 \times 10^4}{4 - 2} \right\rceil = 1$$

$$b_2 = \left\lceil \frac{1}{R_s} \cdot \frac{L_2}{d_2 - s_2} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{6 \times 10^3}{6 - 4} \right\rceil = 3$$

ภาพผนวกที่ 14 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 2

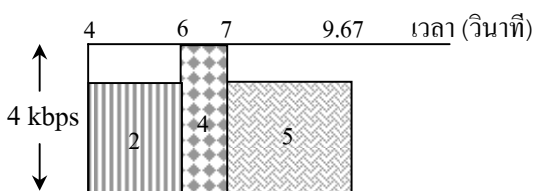
ณ เวลา 3 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 3 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 4 นี้ลงบัฟเฟอร์ หน่วยข้อมูลที่ 4 เป็นหน่วยข้อมูลทวนคิวโอเอสจึงใช้อัตราบิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 15 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และสามารถส่งสำเร็จได้ในวินาทีที่ 7



$$b_4 = b_{max} = 4$$

ภาพผนวกที่ 15 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 3

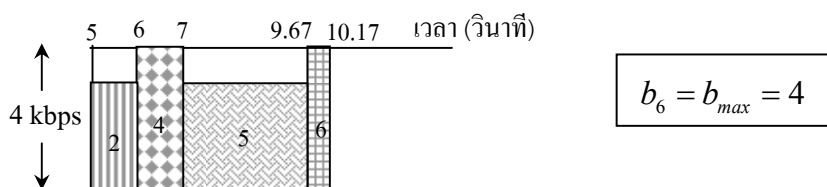
ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 5 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 16 คือเมื่อส่งหน่วยข้อมูลที่ 5 นี้ด้วยอัตราบิต 3 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 7 และจะเสร็จสิ้นที่วินาทีที่ 9.67 เนื่องจากหน่วยข้อมูลที่ 3 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 2 ต่อ



$$b_5 = \left\lceil \frac{1}{R_s} \cdot \frac{L_5}{d_5 - s_5} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{8 \times 10^3}{10 - 7} \right\rceil = 3$$

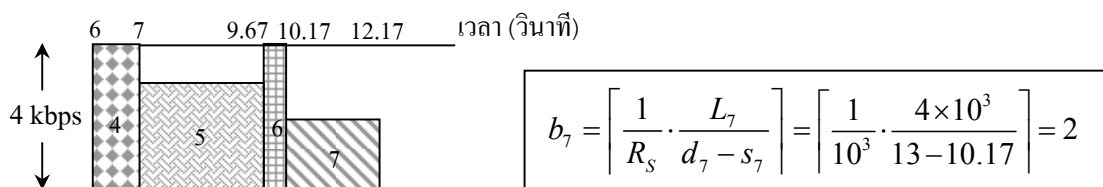
ภาพผนวกที่ 16 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 4

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ที่ 6 ลงบัฟเฟอร์ หน่วยข้อมูลที่ 6 เป็นหน่วยข้อมูลนอนคิวโอเอสจึงใช้อัตราบิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 17 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 9.67 และสามารถส่งสำเร็จได้ในวินาทีที่ 10.17



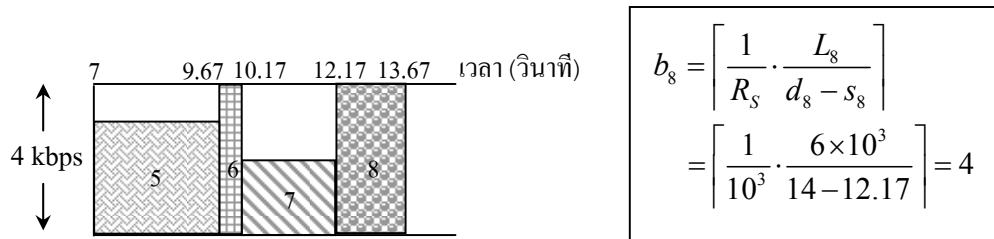
ภาพผนวกที่ 17 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 5

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 7 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 18 คือเมื่อส่งหน่วยข้อมูลที่ 7 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 10.17 และจะเสร็จสิ้นที่วินาทีที่ 12.17 เนื่องจากหน่วยข้อมูลที่ 2 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูล ที่ 4 ต่อ



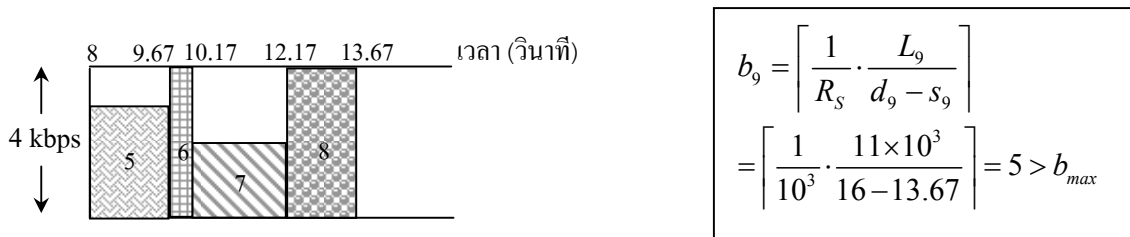
ภาพผนวกที่ 18 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 6

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 8 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 19 คือเมื่อส่งหน่วยข้อมูลที่ 8 นี้ด้วยอัตราบิต 4 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 12.17 และจะเสร็จสิ้นที่วินาทีที่ 13.67 เนื่องจากหน่วยข้อมูลที่ 4 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูล ที่ 5 ต่อ



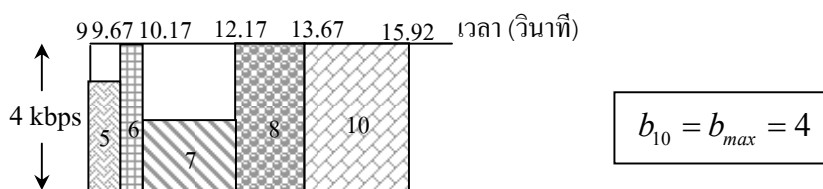
ภาพผนวกที่ 19 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 7

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 9 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 9 เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 จากการคำนวณทำให้ทราบว่าเราไม่สามารถส่งหน่วยข้อมูลที่ 9 ได้ทันเส้นตาย ดังแสดงในภาพผนวกที่ 20 จึงจำเป็นต้องตัดหน่วยข้อมูลนี้ทิ้งไป



ภาพผนวกที่ 20 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 10 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 10 เป็นหน่วยข้อมูลนอนคิวโอเอส จึงใช้อัตราบิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 21 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 10 ได้ในวินาทีที่ 13.67 และสามารถส่งสำเร็จได้ในวินาทีที่ 15.92



ภาพผนวกที่ 21 การคำนวณหาอัตราบิตด้วยวิธี HSNQ ณ วินาทีที่ 9

ณ เวลา 9.67 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 6 ต่อ

ณ เวลา 10.17 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 7 ต่อ

ณ เวลา 12.17 วินาที : หน่วยข้อมูลที่ 7 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 8 ต่อ

ณ เวลา 13.67 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 10 ต่อ

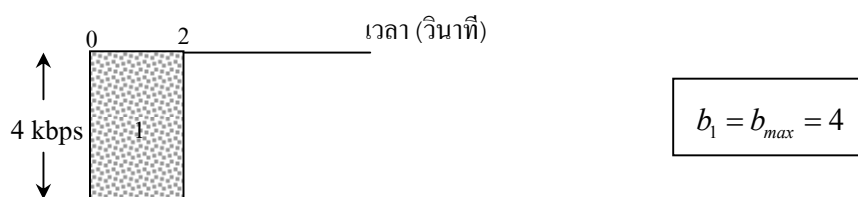
ณ เวลา 15.92 วินาที : หน่วยข้อมูลที่ 10 ส่งสำเร็จพอดี

ตัวอย่างการประหยัดพลังงานด้วยวิธี LI

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 2 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

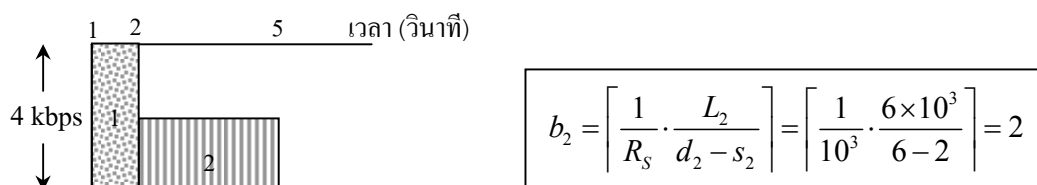
การทำงานของวิธี LI ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิโอสจึงส่งด้วยอัตราบิตสูงสุด (4 kbps) และหน่วยข้อมูลนี้ยาว 8 kbits จึงใช้เวลาในการส่ง 2 วินาที ดังในภาพผนวกที่ 22 โดยเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 2



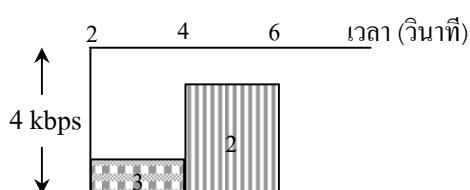
ภาพผนวกที่ 22 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิโอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 13 คือเมื่อส่งหน่วยข้อมูลที่ 2 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 2 และจะเสร็จสิ้นที่วินาทีที่ 5



ภาพผนวกที่ 23 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ จึงต้องมีการคำนวณหาอัตราบิตใหม่ให้กับหน่วยข้อมูลที่มีการเปลี่ยนลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 ดังภาพผนวกที่ 24 หน่วยข้อมูลที่ 3 จะส่งด้วยอัตราบิต 1 kbps จึงทำให้ส่งได้ทัน ซึ่งหน่วยข้อมูลที่ 3 เริ่มส่งในวินาทีที่ 2 และส่งสำเร็จได้ในวินาทีที่ 4 และหน่วยข้อมูลที่ 2 จะส่งด้วยอัตราบิต 3 kbps จึงทำให้ส่งหน่วยข้อมูลที่ 2 นี้ได้ทัน ซึ่งหน่วยข้อมูลที่ 2 เริ่มส่งในวินาทีที่ 4 และส่งสำเร็จได้ในวินาทีที่ 6

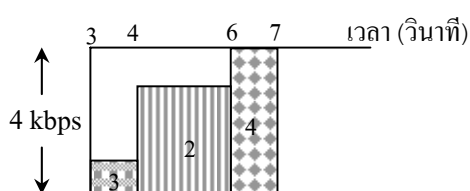


$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{2 \times 10^4}{4 - 2} \right\rceil = 1$$

$$b_2 = \left\lceil \frac{1}{R_s} \cdot \frac{L_2}{d_2 - s_2} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{6 \times 10^3}{6 - 4} \right\rceil = 3$$

ภาพผนวกที่ 24 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 2

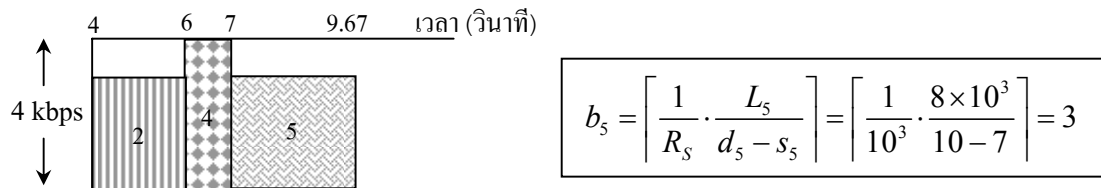
ณ เวลา 3 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 3 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 4 นี้ลงบัฟเฟอร์ หน่วยข้อมูลที่ 4 เป็นหน่วยข้อมูลนอนคิวโอเอสจึงใช้อัตราบิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 25 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และสามารถส่งสำเร็จได้ในวินาทีที่ 7



$$b_4 = b_{max} = 4$$

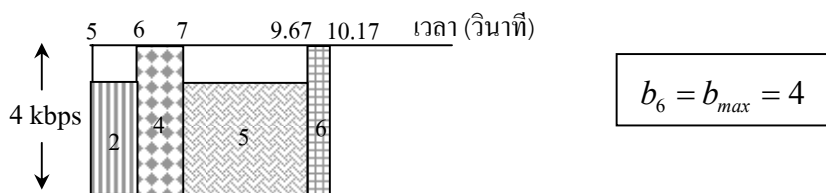
ภาพผนวกที่ 25 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 3

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 5 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 26 คือเมื่อส่งหน่วยข้อมูลที่ 5 นี้ด้วยอัตราบิต 3 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 7 และจะเสร็จสิ้นที่วินาทีที่ 9.67 เนื่องจากหน่วยข้อมูลที่ 3 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 2 ต่อ



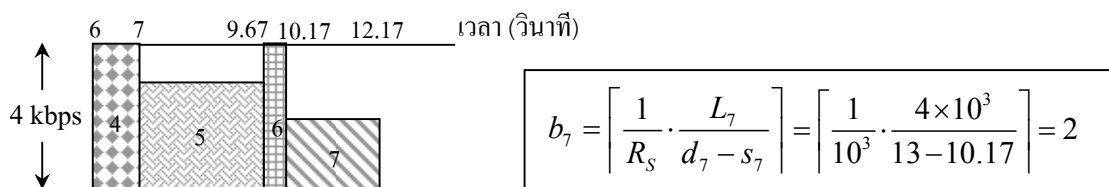
ภาพผนวกที่ 26 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 4

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 6 นี้ลงบัฟเฟอร์ หน่วยข้อมูลที่ 6 เป็นหน่วยข้อมูลนอนคิวโอเอสจึงใช้อัตราบิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 27 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 9.67 และสามารถส่งสำเร็จได้ในวินาทีที่ 10.17



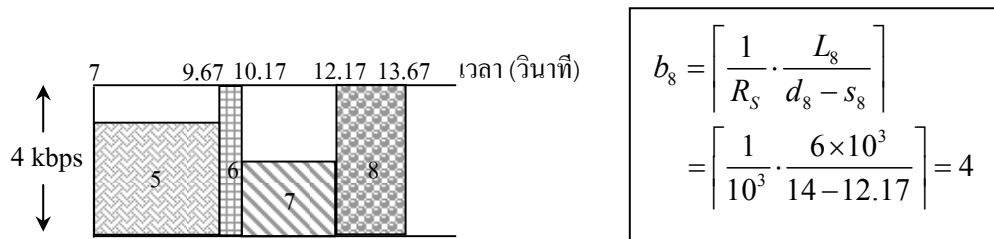
ภาพผนวกที่ 27 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 5

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 7 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 28 คือเมื่อส่งหน่วยข้อมูลที่ 7 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 10.17 และจะเสร็จสิ้นที่วินาทีที่ 12.17 เนื่องจากหน่วยข้อมูลที่ 2 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 4 ต่อ



ภาพผนวกที่ 28 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 6

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 8 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 ได้ค่าดังแสดงในภาพผนวกที่ 29 คือเมื่อส่งหน่วยข้อมูลที่ 8 นี้ด้วยอัตราบิต 4 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 12.17 และจะเสร็จสิ้นที่วินาทีที่ 13.67 เนื่องจากหน่วยข้อมูลที่ 4 ส่งสำเร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูล ที่ 5 ต่อ



ภาพผนวกที่ 29 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 7

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 9 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 9 เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 5 จากการคำนวณทำให้ทราบว่าเราไม่สามารถส่งหน่วยข้อมูลที่ 9 ได้ทันเส้นตาย

$$b_9 = \left\lceil \frac{1}{R_s} \cdot \frac{L_9}{d_9 - s_9} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{11 \times 10^3}{16 - 13.67} \right\rceil = 5 > b_{max}$$

จึงทำการคำนวณซ้ำด้วยภาพที่ 18 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 6, 7, 8 และ 9 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 4 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณซ้ำ เราจะให้หน่วยข้อมูลที่ 6, 7, 8 และ 9 แทนด้วยหน่วยข้อมูลลำดับที่ 1, 2, 3 และ 4 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^i L_j}{d_i - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} > b_{max} \cdot R_s \right)$$

หน่วยข้อมูลลำดับที่ 1 $\Rightarrow$ เป็นหน่วยข้อมูลอนคิวิโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(2+4) \times 10^3}{13 - 9.67} = 1.8 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 3} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(2+4+6) \times 10^3}{14 - 9.67} = 2.77 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(2+4+6+11) \times 10^3}{16 - 9.67} = 3.63 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัพเฟอร์ สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_S = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวิโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \operatorname{argmax}_{q \leq k \leq m, k \in Q} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q - \left(\frac{\sum_{r=q}^{k_0} L_r}{b_{\max} \cdot R_S} \right)} \right\}; \quad p = \left\lceil \frac{1}{R_S} \cdot \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q - \left(\frac{\sum_{r=q}^{k_0} L_r}{b_{\max} \cdot R_S} \right)} \right\rceil$$

เริ่มแรกให้ $q = 1$ (เริ่มจากหน่วยข้อมูลลำดับที่ 1) นั่นคือตรวจสอบทุกหน่วยข้อมูลว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต

ลำดับที่ $k = 1 \Rightarrow$ ไม่คิดเพราะหน่วยข้อมูลอนคิวิโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k = 2 \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1 - \left(\frac{\sum_{r=1}^2 L_r}{b_{\max} \cdot R_S} \right)} = \frac{(4) \times 10^3}{13 - 9.67 - \left(\frac{2 \times 10^3}{4 \times 10^3} \right)} = 1.41$$

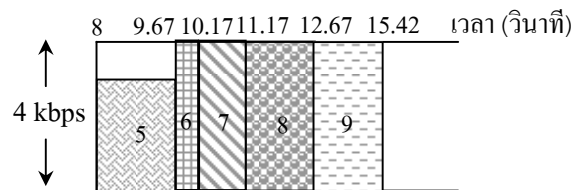
$$\text{ลำดับที่ } k = 3 \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1 - \left(\frac{\sum_{r=1}^3 L_r}{b_{\max} \cdot R_S} \right)} = \frac{(4+6) \times 10^3}{14 - 9.67 - \left(\frac{2 \times 10^3}{4 \times 10^3} \right)} = 2.61$$

$$\text{ลำดับที่ } k = 4 \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1 - \left(\frac{\sum_{r=1}^4 L_r}{b_{\max} \cdot R_S} \right)} = \frac{(4+6+11) \times 10^3}{16 - 9.67 - \left(\frac{2 \times 10^3}{4 \times 10^3} \right)} = 3.60$$

หน่วยข้อมูลลำดับที่ 4 มีความเร่งรีบมากที่สุด ดังนั้นจึงให้หน่วยข้อมูลลำดับที่ 4 เป็นตัวกำหนดอัตราบิต
จะได้ว่า $k_0 = 4$

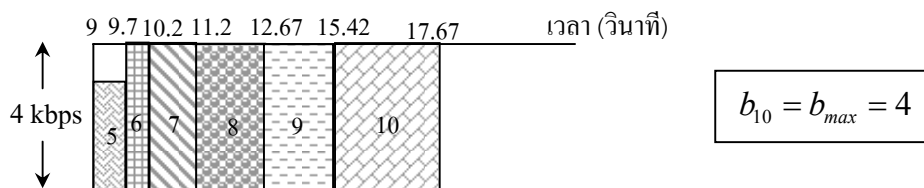
$$b_6 = b_7 = b_8 = b_9 = p = \left\lceil \frac{\sum_{j=1}^4 L_j}{d_4 - s_1 - \left(\frac{\sum_{r=1}^4 L_r}{b_{\max} \cdot R_S} \right)} \cdot \frac{1}{R_S} \right\rceil = \left\lceil \frac{(4+6+11) \times 10^3}{16 - 9.67 - \left(\frac{2 \times 10^3}{4 \times 10^3} \right)} \cdot \frac{1}{10^3} \right\rceil = 4$$

จึงเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 9.67 และส่งสำเร็จที่วินาทีที่ 10.17 และเริ่มส่งหน่วยข้อมูลที่ 7
ได้ในวินาทีที่ 10.17 และส่งสำเร็จที่วินาทีที่ 11.17 และเริ่มส่งหน่วยข้อมูลที่ 8 ได้ในวินาทีที่ 11.17 และ
ส่งสำเร็จที่วินาทีที่ 12.67 และเริ่มส่งหน่วยข้อมูลที่ 9 ได้ในวินาทีที่ 12.67 และส่งสำเร็จที่วินาทีที่ 15.42
ดังในภาพผนวกที่ 30



ภาพผนวกที่ 30 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บ
หน่วยข้อมูลที่ 10 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 10 เป็นหน่วยข้อมูลนอนคิวโอเอส จึงใช้อัตรา
บิตสูงสุดในการส่ง ดังแสดงในภาพผนวกที่ 31 ได้ว่าจะเริ่มส่งหน่วยข้อมูลที่ 10 ได้ในวินาทีที่ 15.42
และสามารถส่งสำเร็จได้ในวินาทีที่ 17.67



ภาพผนวกที่ 31 การคำนวณหาอัตราบิตด้วยวิธี LI ณ วินาทีที่ 9

ณ เวลา 9.67 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 6 ต่อ

ณ เวลา 10.17 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 7 ต่อ

ณ เวลา 12.17 วินาที : หน่วยข้อมูลที่ 7 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 8 ต่อ

ณ เวลา 12.67 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 9 ต่อ

ณ เวลา 15.42 วินาที : หน่วยข้อมูลที่ 9 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 10 ต่อ

ณ เวลา 17.67 วินาที : หน่วยข้อมูลที่ 10 ส่งสำเร็จพอดี

ตัวอย่างการประหยัดพลังงานด้วยวิธี QB

สมมุติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 3 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

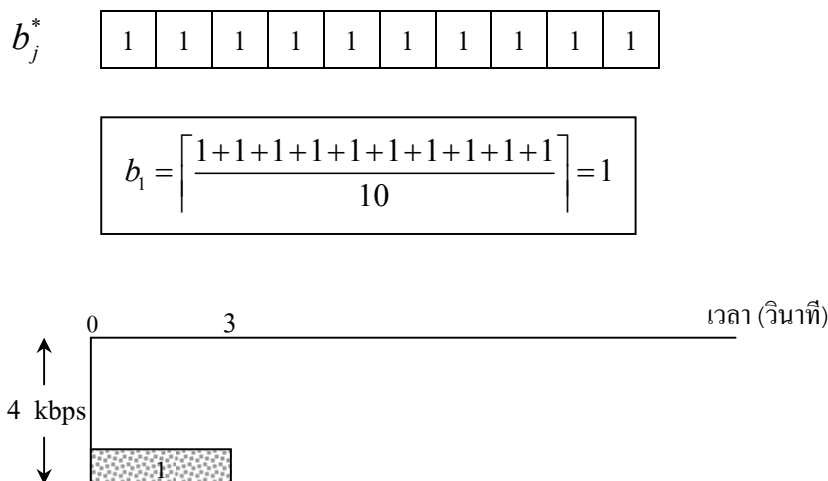
ตารางผนวกที่ 3 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี QB, RRP และ

LS

หน่วยข้อมูลที่	ชนิดของหน่วยข้อมูล	เวลาที่หน่วยข้อมูลมาถึง (วินาที)	จำนวนบิตของหน่วยข้อมูล (กิโลบิต)	เส้นตาย (วินาที)
1	NQoS	0	3	-
2	QoS	1	6	6
3	QoS	2	10	5
4	NQoS	3	4	-
5	QoS	4	8	10
6	NQoS	5	2	-
7	QoS	6	4	13
8	QoS	7	6	14
9	QoS	8	2	16
10	NQoS	9	9	-

การทำงานของวิธี QB ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลอนคิควโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิควโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 แต่เนื่องจากหน่วยข้อมูลนี้เป็นหน่วยข้อมูลแรก เราจึงกำหนดค่าอัตราบิตของหน่วยข้อมูลคิควโอเอสในลำดับเริ่มต้น 10 ลำดับไว้ที่อัตราบิตต่ำสุด (ในที่นี้คือ 1 kbps) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 1 นี้คือ 1 kbps จากภาพผนวกที่ 32 จึงเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 3



ภาพผนวกที่ 32 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิควโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 33 คือเมื่อส่งหน่วยข้อมูลที่ 2 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 6 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ไว้ ($b_2 = 2$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลอนคิควโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ \hline \end{array}$$

$$b_2 = \left\lceil \frac{1}{R_s} \cdot \frac{L_2}{d_2 - s_2} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{6 \times 10^3}{6 - 3} \right\rceil = 2$$



ภาพผนวกที่ 33 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 1

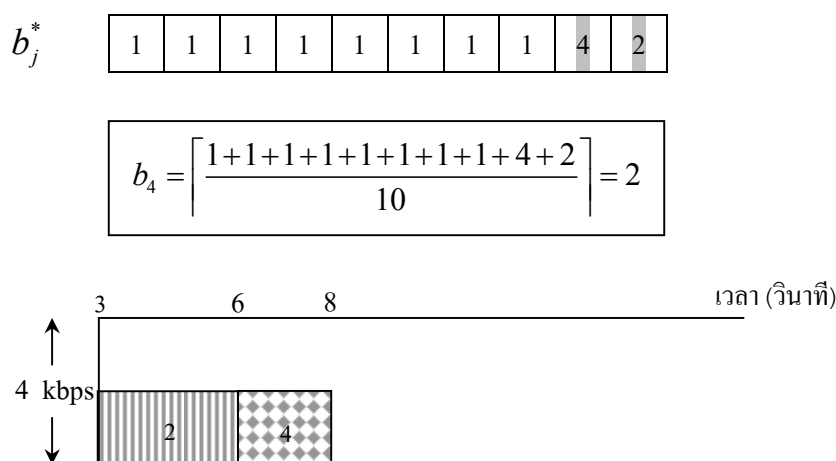
ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ จึงต้องมีการคำนวณหาอัตราบิตใหม่ให้กับหน่วยข้อมูลที่มีการเปลี่ยนลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 ดังภาพผนวกที่ 34 แต่เมื่อคำนวณหา b_3 ทำให้ทราบว่าไม่สามารถส่งหน่วยข้อมูลที่ 3 ได้ทันเส้นตาย จึงจำเป็นต้องทิ้งหน่วยข้อมูลที่ 3 นี้ไป จึงเหลือหน่วยข้อมูลที่ 2 อยู่ในบัฟเฟอร์เพียงหน่วยข้อมูลเดียว จะได้ค่า b_2 ตามเดิม นั่นคือจึงเริ่มส่งหน่วยข้อมูลที่ 2 ได้ที่วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 6 การที่หน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย เราจึงเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 3 ไว้ที่อัตราบิตสูงสุด ($b_{max} = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 \\ \hline \end{array}$$

$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{10^4}{5 - 3} \right\rceil = 5 > b_{max}$$

ภาพผนวกที่ 34 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 2

ณ เวลา 3 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนทิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลทิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 35 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นก็จะเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และสามารถส่งสำเร็จได้ในวินาทีที่ 8 เนื่องจากส่งหน่วยข้อมูลที่ 1 เสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 2 ต่อ

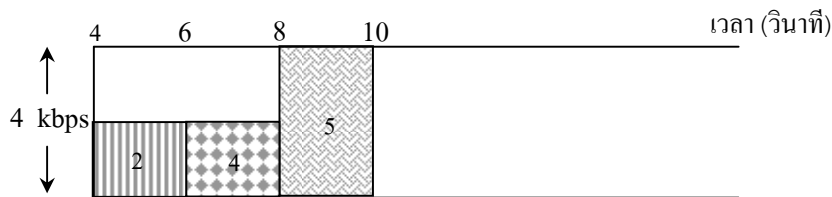


ภาพผนวกที่ 35 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 3

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 5 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 5 เนื่องจากหน่วยข้อมูลที่ 5 นี้เป็นหน่วยข้อมูลทิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 36 คือเมื่อส่งหน่วยข้อมูลที่ 5 นี้ด้วยอัตราบิต 4 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 5 ได้ในวินาทีที่ 8 และสามารถส่งสำเร็จได้ในวินาทีที่ 10 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 5 ไว้ ($b_5 = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนทิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 4 \\ \hline \end{array}$$

$$b_5 = \left\lceil \frac{1}{R_s} \cdot \frac{L_5}{d_5 - s_5} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{8 \times 10^3}{10 - 8} \right\rceil = 4$$

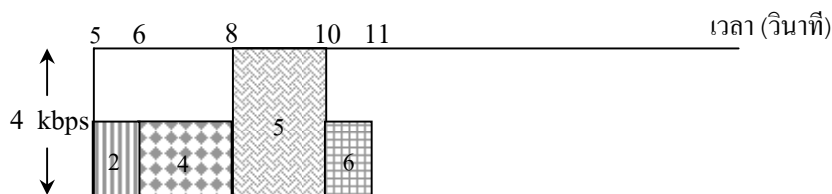


ภาพผนวกที่ 36 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 4

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 6 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 6 เป็นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 37 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และสามารถส่งสำเร็จได้ในวินาทีที่ 11

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 4 \\ \hline \end{array}$$

$$b_6 = \left\lceil \frac{1+1+1+1+1+1+1+2+4+4}{10} \right\rceil = 2$$

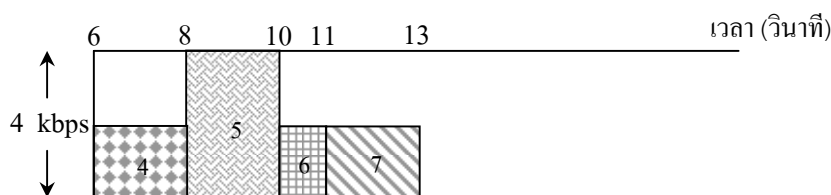


ภาพผนวกที่ 37 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 5

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 38 คือเมื่อส่งหน่วยข้อมูลที่ 7 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 11 และสามารถส่งสำเร็จได้ในวินาทีที่ 13 และเนื่องจาก ณ เวลานั้น หน่วยข้อมูลที่ 2 ส่งเสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 4 ต่อ และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 7 ไว้ ($b_7 = 2$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 4 & 2 \\ \hline \end{array}$$

$$b_7 = \left\lceil \frac{1}{R_s} \cdot \frac{L_7}{d_7 - s_7} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{4 \times 10^3}{13 - 11} \right\rceil = 2$$



ภาพผนวกที่ 38 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 6

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 4 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 8 เนื่องจากหน่วยข้อมูลที่ 8 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 39 แต่เมื่อคำนวณหา b_8 ทำให้ทราบว่าไม่สามารถส่งหน่วยข้อมูลที่ 8 ได้ทันเส้นตาย จึงจำเป็นต้องทิ้งหน่วยข้อมูลที่ 8 นี้ไป การที่หน่วยข้อมูลที่ 8 ไม่สามารถส่งได้ทันเส้นตาย เราจึงเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลนี้ ไว้ที่อัตราบิตสูงสุด ($b_{max} = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 \\ \hline \end{array}$$

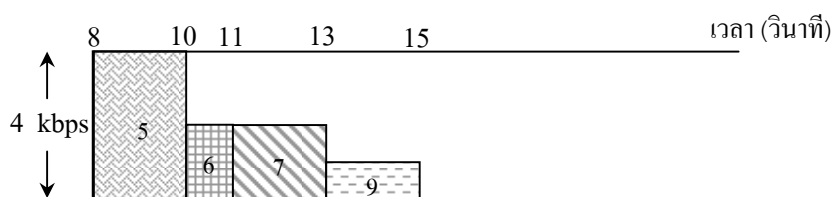
$$b_8 = \left\lceil \frac{1}{R_s} \cdot \frac{L_8}{d_8 - s_8} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{6 \times 10^3}{14 - 13} \right\rceil = 6 > b_{max}$$

ภาพผนวกที่ 39 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 7

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 40 คือเมื่อส่งหน่วยข้อมูลที่ 9 นี้ด้วยอัตราบิต 1 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 9 ได้ในวินาทีที่ 13 และสามารถส่งสำเร็จได้ในวินาทีที่ 15 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 9 ไว้ ($b_9 = 1$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป และเนื่องจาก ณ เวลานั้น หน่วยข้อมูลที่ 4 ส่งเสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 5 ต่อ

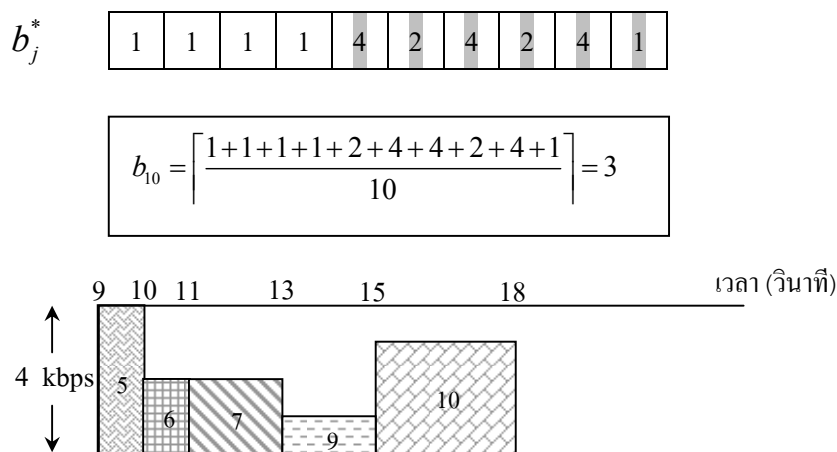
$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 4 & 2 & 4 & 2 & 4 & 1 \\ \hline \end{array}$$

$$b_9 = \left\lceil \frac{1}{R_s} \cdot \frac{L_9}{d_9 - s_9} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{2 \times 10^3}{16 - 13} \right\rceil = 1$$



ภาพผนวกที่ 40 การคำนวณหาอัตราบิตด้วยวิธี QB ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 10 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 10 เป็นหน่วยข้อมูลนอนควิโอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลควิโอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 41 ได้ค่าเฉลี่ย 3 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 10 ได้ในวินาทีที่ 15 และสามารถส่งสำเร็จได้ในวินาทีที่ 18



ภาพผนวกที่ 41 การคำนวณอัตราบิตด้วยวิธี QB ณ วินาทีที่ 9

ณ เวลา 10 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 6 ต่อ

ณ เวลา 11 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 7 ต่อ

ณ เวลา 13 วินาที : หน่วยข้อมูลที่ 7 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 9 ต่อ

ณ เวลา 15 วินาที : หน่วยข้อมูลที่ 9 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 10 ต่อ

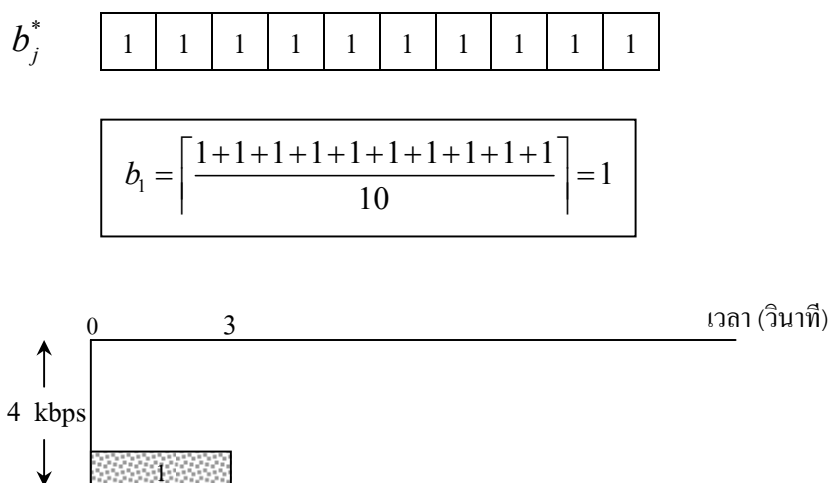
ณ เวลา 18 วินาที : หน่วยข้อมูลที่ 10 ส่งสำเร็จพอดี

ตัวอย่างการประหยัดพลังงานด้วยวิธี RRP

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 3 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

การทำงานของวิธี RRP ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 แต่เนื่องจากหน่วยข้อมูลนี้เป็นหน่วยข้อมูลแรก เราจึงกำหนดค่าอัตราบิตของหน่วยข้อมูลคิวโอเอสในลำดับเริ่มต้น 10 ลำดับไว้ที่อัตราบิตต่ำสุด (ในที่นี้คือ 1 kbps) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 1 นี้คือ 1 kbps จากภาพผนวกที่ 42 จึงเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 3

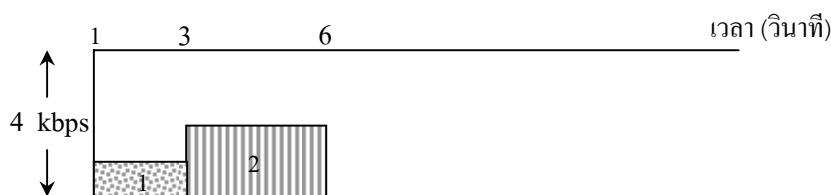


ภาพผนวกที่ 42 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 43 คือเมื่อส่งหน่วยข้อมูลที่ 2 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 6 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ไว้ ($b_2 = 2$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ \hline \end{array}$$

$$b_2 = \left\lceil \frac{1}{R_s} \cdot \frac{L_2}{d_2 - s_2} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{6 \times 10^3}{6 - 3} \right\rceil = 2$$



ภาพผนวกที่ 43 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ จึงต้องมีการคำนวณหาอัตราบิตใหม่ให้กับหน่วยข้อมูลที่มีการเปลี่ยนลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 ดังภาพผนวกที่ 44 แต่เมื่อคำนวณอัตราบิตตามตารางที่ 6 แล้วจะเห็นว่าหน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย จึงทำการคำนวณซ้ำตามภาพที่ 21 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 3 และ 2 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 2 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในส่วนการคำนวณซ้ำ เราจะให้หน่วยข้อมูลที่ 3 และ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1 และ 2 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(10) \times 10^3}{5 - 3} = 5 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(10 + 6) \times 10^3}{6 - 3} = 5.33 \times 10^3 \text{ bps}$$

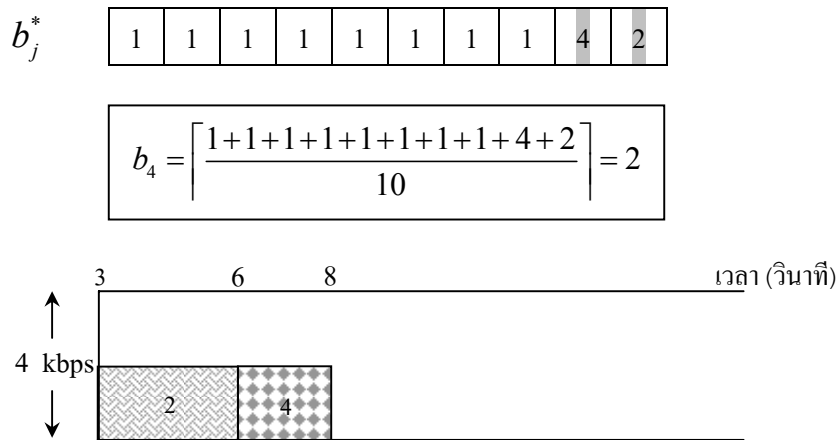
จะเห็นว่าต้องทิ้งบางหน่วยข้อมูล เนื่องจากไม่สามารถส่งหน่วยข้อมูลเหล่านั้นด้วยอัตราบิตที่สูงกว่าอัตราบิตสูงสุดได้ $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ จึงทิ้งหน่วยข้อมูลใหม่ที่เพิ่งเข้ามานั้นคือหน่วยข้อมูลที่ 3 (หรือหน่วยข้อมูลลำดับที่ 1 นั้นเอง) จึงเหลือหน่วยข้อมูลที่ 2 อยู่ในบัฟเฟอร์เพียงหน่วยข้อมูลเดียว และจะได้ค่า b_2 ตามเดิม นั่นคือจึงเริ่มส่งหน่วยข้อมูลที่ 2 ได้ที่วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 6 การที่หน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย เราจึงเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 3 ไว้ที่อัตราบิตสูงสุด ($b_{\max} = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 \\ \hline \end{array}$$

$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{10^4}{5 - 3} \right\rceil = 5$$

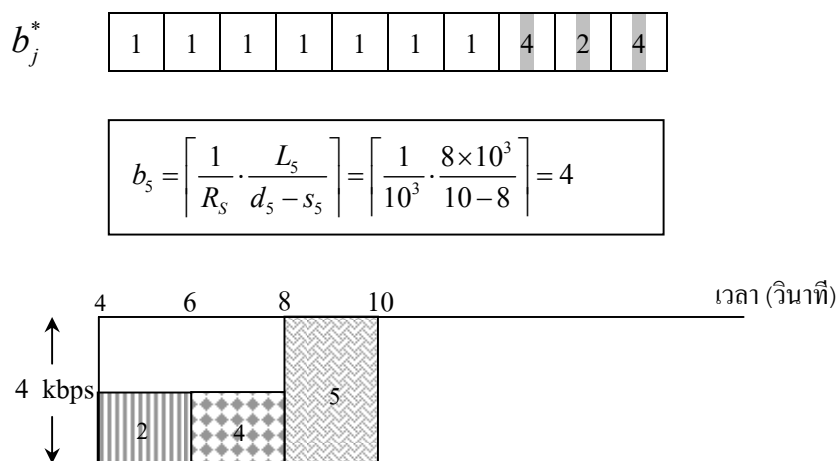
รูปที่ 44 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 2

ณ เวลา 3 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอกคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 45 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และสามารถส่งสำเร็จได้ในวินาทีที่ 8 เนื่องจาก ณ เวลานี้ ส่งหน่วยข้อมูลที่ 1 เสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 2 ต่อ



รูปที่ 45 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 3

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 5 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 5 เนื่องจากหน่วยข้อมูลที่ 5 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 46 คือเมื่อส่งหน่วยข้อมูลที่ 5 นี้ด้วยอัตราบิต 4 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 5 ได้ในวินาทีที่ 8 และสามารถส่งสำเร็จได้ในวินาทีที่ 10 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 5 ไว้ ($b_5 = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป

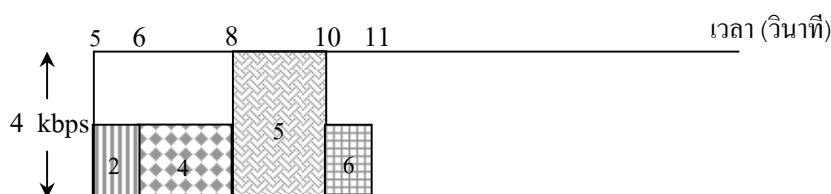


ภาพผนวกที่ 46 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 4

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 6 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 6 เป็นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 47 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และสามารถส่งสำเร็จได้ในวินาทีที่ 11

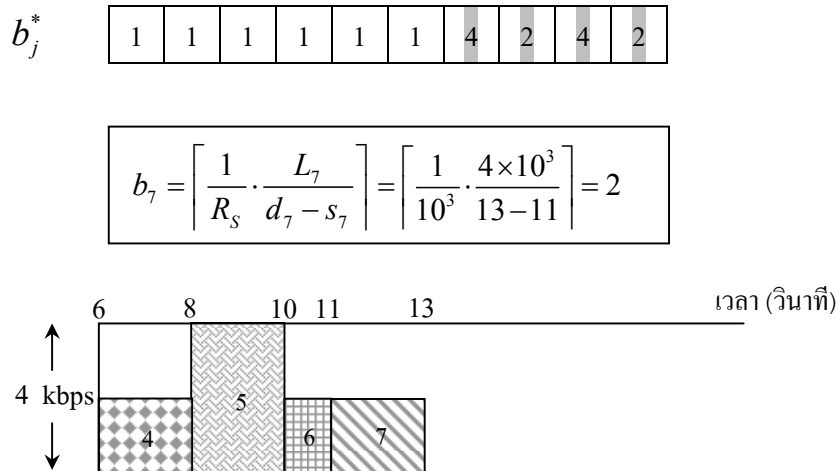
$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 4 \\ \hline \end{array}$$

$$b_6 = \left\lceil \frac{1+1+1+1+1+1+1+2+4+4}{10} \right\rceil = 2$$



ภาพผนวกที่ 47 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 5

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 48 คือเมื่อส่งหน่วยข้อมูลที่ 7 นี้ด้วยอัตราบิต 2 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 11 และสามารถส่งสำเร็จได้ในวินาทีที่ 13 และเนื่องจาก ณ เวลานั้น หน่วยข้อมูลที่ 2 ส่งเสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 4 ต่อ และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 7 ไว้ ($b_7 = 2$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป



ภาพผนวกที่ 48 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 6

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 4 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 8 เนื่องจากหน่วยข้อมูลที่ 8 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 49 แต่เมื่อคำนวณอัตราบิตตามตารางที่ 6 แล้วจะได้ว่าหน่วยข้อมูลที่ 8 ไม่สามารถส่งได้ทันเส้นตาย จึงทำการคำนวณซ้ำตามภาพที่ 21 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 5, 6, 7 และ 8 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 4 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในส่วนการคำนวณซ้ำ เราจะให้หน่วยข้อมูลที่ 5, 6, 7 และ 8 แทนด้วยหน่วยข้อมูลลำดับที่ 1, 2, 3 และ 4 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{10 - 8} = 4 \times 10^3 \text{ bps}$$

หน่วยข้อมูลลำดับที่ 2 $\Rightarrow$ เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 3} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8 + 2 + 4) \times 10^3}{13 - 8} = 2.8 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(8+2+4+6) \times 10^3}{14-8} = 3.33 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัพเฟอร์ สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \operatorname{argmax}_{q \leq k \leq m, k \in Q} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_s} \right\rceil$$

เริ่มแรกให้ $q = 1$ (เริ่มจากหน่วยข้อมูลลำดับที่ 1) นั่นคือตรวจสอบทุกหน่วยข้อมูลว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต

$$\text{ลำดับที่ } k=1 \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{8 \times 10^3}{10-8} = 4 \times 10^3$$

ลำดับที่ $k=2$ ไม่คิดเพราะหน่วยข้อมูลนอนคิวโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k=3 \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8+2+4) \times 10^3}{13-8} = 2.8 \times 10^3$$

$$\text{ลำดับที่ } k=4 \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(8+2+4+6) \times 10^3}{14-8} = 3.33 \times 10^3$$

ดังนั้น $k_0 = 1$ นั่นคือหน่วยข้อมูลลำดับที่ 1 (หน่วยข้อมูลที่ 5) เป็นตัวกำหนดอัตราบิตให้หน่วยข้อมูล

$$\text{ลำดับ } k=1 \text{ ได้ว่า } b_5 = p = \left\lceil \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} \cdot \frac{1}{R_s} \right\rceil = \frac{8 \times 10^3}{10-8} \cdot \frac{1}{10^3} = 4 \text{ จึงเริ่มส่งหน่วยข้อมูลที่ 5 ได้ใน}$$

วินาทีที่ 8 และส่งสำเร็จที่วินาทีที่ 10

ต่อมาในรอบที่สองให้ $q = 2$ นั่นคือตรวจสอบหน่วยข้อมูลลำดับที่ 2, 3 และ 4 ในบัพเฟอร์ (นั่นคือหน่วยข้อมูลที่ 6, 7 และ 8) ว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต

ลำดับที่ $k=2$ ไม่คิดเพราะหน่วยข้อมูลนอนคิวโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k=3 \Rightarrow \frac{\sum_{j=2}^3 L_j}{d_3 - s_2} = \frac{(2+4) \times 10^3}{13-10} = 2 \times 10^3$$

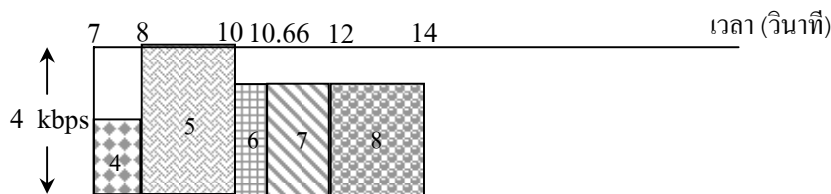
$$\text{ลำดับที่ } k=4 \Rightarrow \frac{\sum_{j=2}^4 L_j}{d_4 - s_2} = \frac{(2+4+6) \times 10^3}{14-10} = 3 \times 10^3$$

ดังนั้น $k_0 = 4$ นั่นคือหน่วยข้อมูลลำดับที่ 4 (หน่วยข้อมูลที่ 8) เป็นตัวกำหนดอัตราบิตให้หน่วยข้อมูลลำดับที่ $k=2, 3$ และ 4

$$b_6 = b_7 = b_8 = p = \left\lceil \frac{\sum_{j=2}^4 L_j}{d_4 - s_2} \cdot \frac{1}{R_s} \right\rceil = \frac{(2+4+6) \times 10^3}{14-10} \cdot \frac{1}{10^3} = 3$$

จึงเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และส่งสำเร็จที่วินาทีที่ 10.667 และเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 10.667 และส่งสำเร็จที่วินาทีที่ 12 และเริ่มส่งหน่วยข้อมูลที่ 8 ได้ในวินาทีที่ 12 และส่งสำเร็จที่วินาทีที่ 14 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 5 และ 7 ใหม่เป็น 4 และ 3 รวมทั้งเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 8 ไว้ ($b_8 = 3$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป

b_j^*	1	1	1	1	1	4	2	4	3	3
---------	---	---	---	---	---	---	---	---	---	---



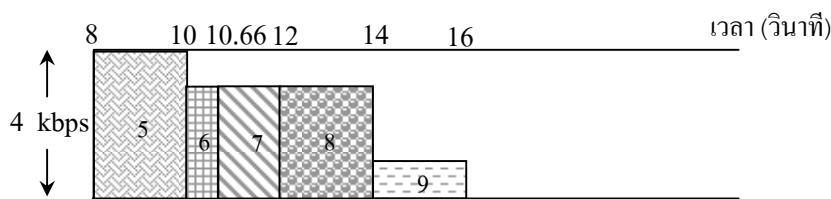
รูปที่ 49 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 7

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 50 คือเมื่อส่งหน่วยข้อมูลที่ 9 นี้ด้วยอัตราบิต 1 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลที่ 9 ได้ในวินาทีที่ 14 และสามารถส่งสำเร็จได้ในวินาทีที่ 16 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 9 ไว้ ($b_9 = 1$)

เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลอนคิวิโอเอสในลำดับถัดๆ ไป และเนื่องจาก ณ เวลานี้ หน่วยข้อมูลที่ 4 ส่งเสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 5 ต่อ

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 4 & 2 & 4 & 3 & 3 & 1 & \\ \hline \end{array}$$

$$b_9 = \left\lceil \frac{1}{R_s} \cdot \frac{L_9}{d_9 - s_9} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{2 \times 10^3}{16 - 14} \right\rceil = 1$$

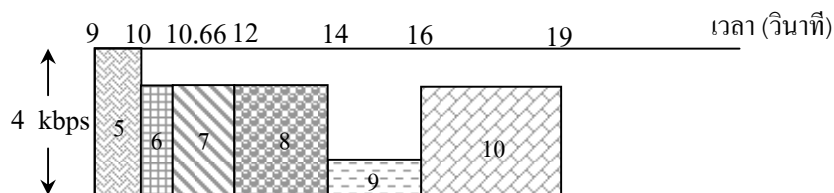


รูปที่ 50 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 10 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 10 เป็นหน่วยข้อมูลอนคิวิโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวิโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 51 ได้ค่าเฉลี่ย 3 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 10 ได้ในวินาทีที่ 16 และสามารถส่งสำเร็จได้ในวินาทีที่ 19

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 4 & 2 & 4 & 3 & 3 & 1 & \\ \hline \end{array}$$

$$b_{10} = \left\lceil \frac{1+1+1+1+2+4+4+3+3+1}{10} \right\rceil = 3$$



รูปที่ 51 การคำนวณหาอัตราบิตด้วยวิธี RRP ณ วินาทีที่ 9

ณ เวลา 10 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 6 ต่อ

ณ เวลา 10.667 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 7 ต่อ

ณ เวลา 12 วินาที : หน่วยข้อมูลที่ 7 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 8 ต่อ

ณ เวลา 14 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 9 ต่อ

ณ เวลา 16 วินาที : หน่วยข้อมูลที่ 9 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 10 ต่อ

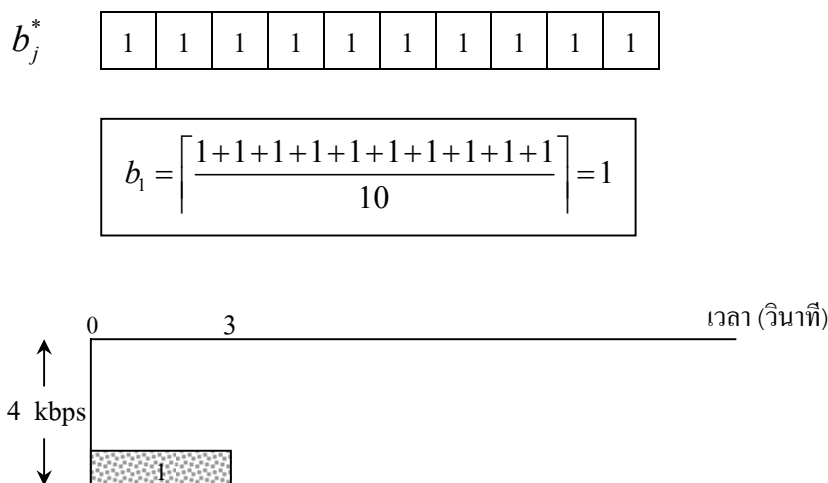
ณ เวลา 19 วินาที : หน่วยข้อมูลที่ 10 ส่งสำเร็จพอดี

ตัวอย่างการประหยัดพลังงานด้วยวิธี LS

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่าน โหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 3 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

การทำงานของวิธี LS ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 แต่เนื่องจากหน่วยข้อมูลนี้เป็นหน่วยข้อมูลแรก เราจึงกำหนดค่าอัตราบิตของหน่วยข้อมูลคิวโอเอสในลำดับเริ่มต้น 10 ลำดับไว้ที่อัตราบิตต่ำสุด (ในที่นี้คือ 1 kbps) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 1 นี้คือ 1 kbps จากภาพผนวกที่ 52 จึงเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 3



รูปที่ 52 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 53 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 2 อยู่ในบัฟเฟอร์เพียงหน่วยข้อมูลเดียว แล้วนำหน่วยข้อมูลนั้นมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1)

$$\left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} > b_{\max} \cdot R_s \right)$$

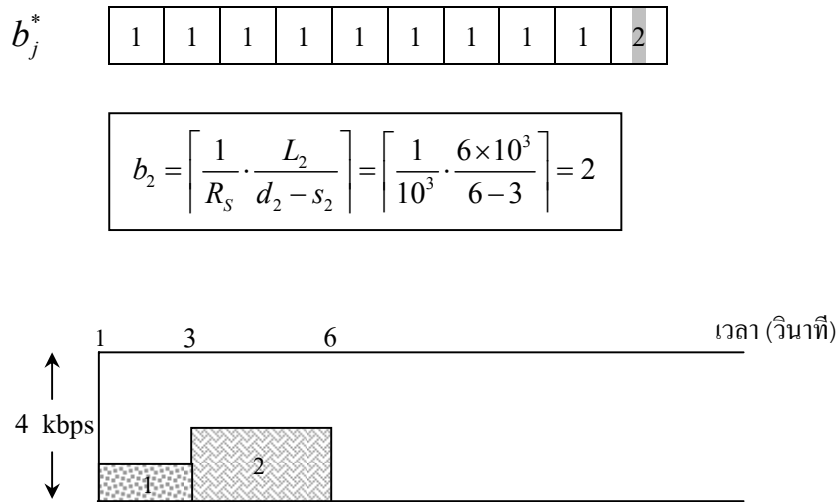
$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(6) \times 10^3}{6 - 3} = 2 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์ สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_s} \right\rceil$$

เนื่องจากมีหน่วยข้อมูลเดียว หน่วยข้อมูลนั้นจึงเป็นตัวกำหนดอัตราบิต ($k_0 = 1$) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 คือ $b_2 = p = \left\lceil \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} \cdot \frac{1}{R_s} \right\rceil = \frac{6 \times 10^3}{6 - 3} \cdot \frac{1}{10^3} = 2$ จึงเริ่มส่งหน่วยข้อมูลที่ 2 ได้ในวินาทีที่ 3 และส่งสำเร็จที่วินาทีที่ 6 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ไว้ ($b_2 = 2$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลอนคิวโอเอสในลำดับถัดๆ ไป



รูปที่ 53 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 และคำนวณด้วยกระบวนการตามภาพที่ 21 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 3 และ 2 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 2 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าแรงความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 3 และ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1 และ 2 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(10) \times 10^3}{5 - 3} = 5 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(10 + 6) \times 10^3}{6 - 3} = 5.33 \times 10^3 \text{ bps}$$

จะเห็นว่าต้องทิ้งบางหน่วยข้อมูล เนื่องจากไม่สามารถส่งหน่วยข้อมูลเหล่านั้นด้วยอัตราบิตที่สูงกว่าอัตราบิตสูงสุดได้ $\Rightarrow b_{max} \cdot R_s = 4 \times 10^3$ bps จึงทิ้งหน่วยข้อมูลใหม่ที่เพิ่งเข้ามานั้นคือหน่วยข้อมูลที่ 3 (หรือหน่วยข้อมูลลำดับที่ 1 นั้นเอง) จึงเหลือหน่วยข้อมูลที่ 2 อยู่ในบัฟเฟอร์เพียงหน่วยข้อมูลเดียว และจะได้ค่า b_2 ตามเดิม นั่นคือจึงเริ่มส่งหน่วยข้อมูลที่ 2 ได้ที่วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 6 การที่หน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย เราจึงเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 3 ไว้ที่อัตราบิตสูงสุด ($b_{max} = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลอนคิโอสในลำดับถัดๆ ไป ดังภาพผนวกที่ 54

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 \\ \hline \end{array}$$

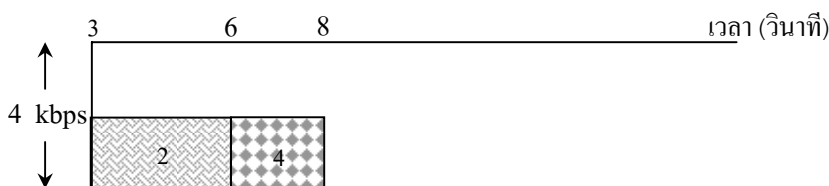
$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{10^4}{5 - 3} \right\rceil = 5$$

รูปที่ 54 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 2

ณ เวลา 3 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลอนคิโอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิโอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 55 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และสามารถส่งสำเร็จได้ในวินาทีที่ 8 เนื่องจาก ณ เวลานี้ ส่งหน่วยข้อมูลที่ 1 เสร็จพอดี จึงให้เริ่มส่งหน่วยข้อมูลที่ 2 ต่อ

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 \\ \hline \end{array}$$

$$b_4 = \left\lceil \frac{1+1+1+1+1+1+1+1+4+2}{10} \right\rceil = 2$$



รูปที่ 55 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 3

ณ เวลา 4 วินาที : หน่วยข้อมูลที่ 5 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 2 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 5 เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 54 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 4 และ 5 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 2 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 4 และ 5 แทนด้วยหน่วยข้อมูลลำดับที่ 1 และ 2 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

หน่วยข้อมูลลำดับที่ 1 => เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(4+8) \times 10^3}{10-6} = 3 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์ สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด => $b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

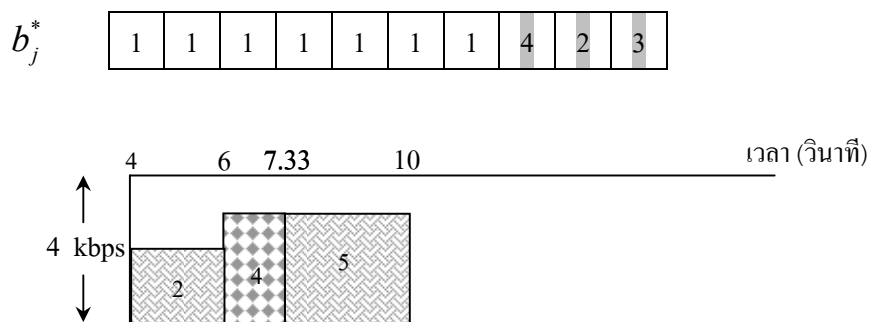
ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_s} \right\rceil$$

เนื่องจากมีหน่วยข้อมูลคิวโอเอสเพียงหน่วยเดียว หน่วยข้อมูลนั้นจึงเป็นตัวกำหนดอัตราบิต ($k_0 = 2$)

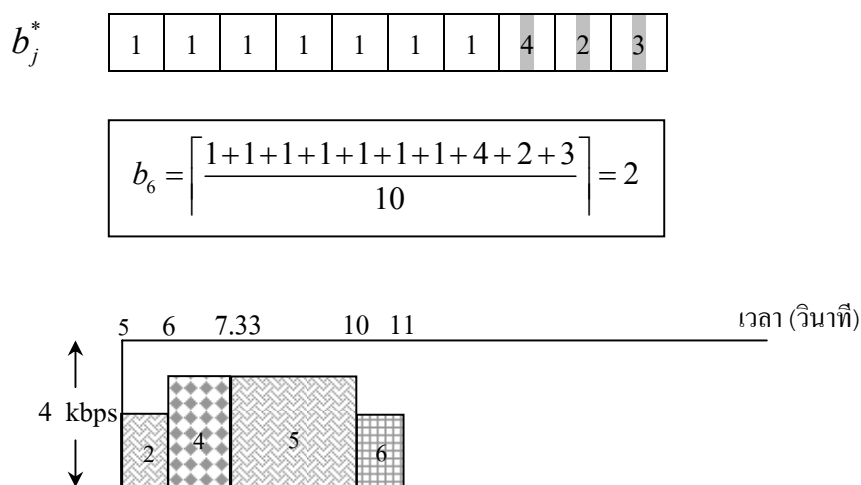
$$\text{ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 คือ } b_5 = p = \left\lceil \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} \cdot \frac{1}{R_s} \right\rceil = \frac{(4+8) \times 10^3}{10-6} \cdot \frac{1}{10^3} = 3$$

จึงเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และส่งสำเร็จที่วินาทีที่ 7.33 และเริ่มส่งหน่วยข้อมูลที่ 5 ได้ในวินาทีที่ 7.33 และส่งสำเร็จที่วินาทีที่ 10 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 5 ไว้ ($b_5 = 3$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป



รูปที่ 56 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 4

ณ เวลา 5 วินาที : หน่วยข้อมูลที่ 6 เข้ามายังโหนด หน่วยข้อมูลนี้เป็นหน่วยข้อมูลทวิโอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลทวิโอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 57 ได้ค่าเฉลี่ย 2 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และสามารถส่งสำเร็จได้ในวินาทีที่ 11



รูปที่ 57 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 5

ณ เวลา 6 วินาที : หน่วยข้อมูลที่ 7 เข้ามายังโหนด เนื่องจากหน่วยข้อมูลที่ 7 เป็นหน่วยข้อมูลทวิโอสจึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 58 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 4, 5, 6 และ 7 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 4 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 4, 5, 6 และ 7 แทนด้วยหน่วยข้อมูลลำดับที่ 1, 2, 3 และ 4 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} > b_{\max} \cdot R_S \right)$$

หน่วยข้อมูลลำดับที่ 1 $\Rightarrow$ เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(4+8) \times 10^3}{10-6} = 3 \times 10^3 \text{ bps}$$

หน่วยข้อมูลลำดับที่ 3 $\Rightarrow$ เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(4+8+2+4) \times 10^3}{13-6} = 2.57 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_S = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; \quad p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_S} \right\rceil$$

เริ่มแรกให้ $q = 1$ นั่นคือตรวจสอบทุกหน่วยข้อมูลว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต (ลำดับที่ 1, 2, 3 และ 4 ในบัฟเฟอร์คือหน่วยข้อมูลที่ 4, 5, 6 และ 7)

ลำดับที่ $k = 1$ ไม่คิดเพราะหน่วยข้อมูลนอนคิวโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k = 2 \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(4+8) \times 10^3}{10-6} = 3 \times 10^3$$

ลำดับที่ $k = 3$ ไม่คิดเพราะหน่วยข้อมูลนอนคิวโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k = 4 \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(4+8+2+4) \times 10^3}{13-6} = 2.57 \times 10^3$$

ได้ว่า $k_0 = 2$ นั่นคือหน่วยข้อมูลที่ 5 (หน่วยข้อมูลลำดับที่ 2) เป็นตัวกำหนดอัตราบิดให้หน่วยข้อมูลลำดับที่ $k = 1$ และ 2 โดย

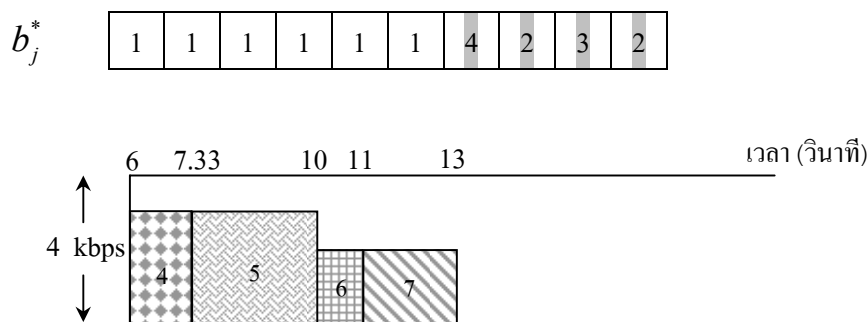
$$b_4 = b_5 = p = \left\lceil \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} \cdot \frac{1}{R_s} \right\rceil = \left\lceil \frac{(4+8) \times 10^3}{10-6} \cdot \frac{1}{10^3} \right\rceil = 3$$

ดังนั้นจึงสามารถเริ่มส่งหน่วยข้อมูลที่ 4 ได้ในวินาทีที่ 6 และส่งสำเร็จที่วินาทีที่ 7.33 และเริ่มส่งหน่วยข้อมูลที่ 5 ได้ในวินาทีที่ 7.33 และส่งสำเร็จที่วินาทีที่ 10

ต่อมาในรอบที่สองให้ $q = 2$ นั่นคือตรวจสอบหน่วยข้อมูลลำดับที่ 3 และ 4 ในบัฟเฟอร์ (นั่นคือหน่วยข้อมูลที่ 6 และ 7) ว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิด แต่เนื่องจากว่าในรอบนี้มีหน่วยข้อมูลคิวโอเอสเพียงหน่วยเดียวคือหน่วยข้อมูลลำดับที่ 4 (หน่วยข้อมูลที่ 7) หน่วยข้อมูลนี้จึงเป็นตัวกำหนดความเร็ว

$$b_6 = b_7 = p = \left\lceil \frac{\sum_{j=3}^4 L_j}{d_4 - s_3} \cdot \frac{1}{R_s} \right\rceil = \frac{(2+4) \times 10^3}{13-10} \cdot \frac{1}{10^3} = 2$$

จึงเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และส่งสำเร็จที่วินาทีที่ 11 และเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 11 และส่งสำเร็จที่วินาทีที่ 13 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 5 ใหม่เป็น 3 รวมทั้งเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 7 ไว้ ($b_7 = 2$) เพื่อนำไปหาอัตราบิดในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป



รูปที่ 58 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 6

ณ เวลา 7 วินาที : หน่วยข้อมูลที่ 8 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 4 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 8 เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 59 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 5, 6, 7 และ 8 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 4 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 5, 6, 7 และ 8 แทนด้วยหน่วยข้อมูลลำดับที่ 1, 2, 3 และ 4 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{10 - 7.33} = 3 \times 10^3 \text{ bps}$$

หน่วยข้อมูลลำดับที่ 2 $\Rightarrow$ เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 3} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8 + 2 + 4) \times 10^3}{13 - 7.33} = 2.47 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(8 + 2 + 4 + 6) \times 10^3}{14 - 7.33} = 3 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิต

สูงสุด $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_S} \right\rceil$$

เริ่มแรกให้ $q = 1$ นั่นคือตรวจสอบทุกหน่วยข้อมูลว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต (ลำดับที่ 1, 2, 3 และ 4 ในบัพเฟอร์คือหน่วยข้อมูลที่ 5, 6, 7 และ 8)

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{10 - 7.33} = 3 \times 10^3 \text{ bps}$$

หน่วยข้อมูลลำดับที่ 2 $\Rightarrow$ เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 3} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8+2+4) \times 10^3}{13 - 7.33} = 2.47 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(8+2+4+6) \times 10^3}{14 - 7.33} = 3 \times 10^3 \text{ bps}$$

$$\text{ลำดับที่ } k=1 \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{10 - 7.33} = 3 \times 10^3$$

ลำดับที่ $k=2$ ไม่คิดเพราะหน่วยข้อมูลนอนคิวโอเอสไม่มีเส้นตาย

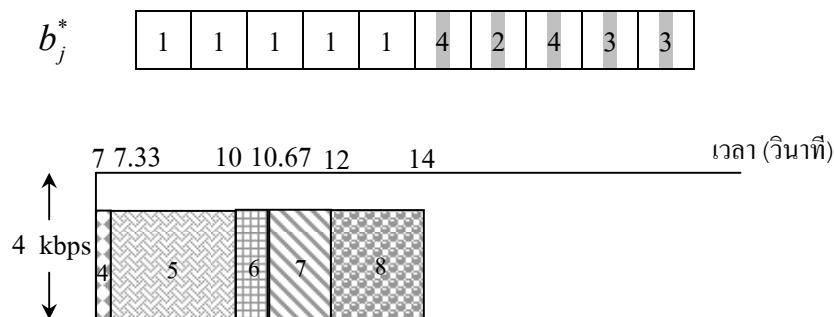
$$\text{ลำดับที่ } k=3 \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8+2+4) \times 10^3}{13 - 7.33} = 2.47 \times 10^3$$

$$\text{ลำดับที่ } k=4 \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(8+2+4+6) \times 10^3}{14 - 7.33} = 3 \times 10^3$$

ได้ว่า $k_0 = 4$ นั่นคือหน่วยข้อมูลที่ 8 (หน่วยข้อมูลลำดับที่ 4) เป็นตัวกำหนดอัตราบิตให้หน่วยข้อมูลลำดับที่ $k = 1, 2, 3$ และ 4 โดย

$$b_5 = b_6 = b_7 = b_8 = p = \left\lceil \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} \cdot \frac{1}{R_S} \right\rceil = \left\lceil \frac{(8+2+4+6) \times 10^3}{14 - 7.33} \cdot \frac{1}{10^3} \right\rceil = 3$$

ดังนั้นจึงสามารถเริ่มส่งหน่วยข้อมูลที่ 5 ได้ในวินาทีที่ 7.33 และส่งสำเร็จที่วินาทีที่ 10 และเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และส่งสำเร็จที่วินาทีที่ 10.67 และเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 10.67 และส่งสำเร็จที่วินาทีที่ 12 และเริ่มส่งหน่วยข้อมูลที่ 8 ได้ในวินาทีที่ 12 และส่งสำเร็จที่วินาทีที่ 14



รูปที่ 59 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 7

ณ เวลา 7.33 วินาที : หน่วยข้อมูลที่ 4 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 5 ต่อ

ณ เวลา 8 วินาที : หน่วยข้อมูลที่ 9 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลนี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 9 เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 60 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 6, 7, 8 และ 9 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 4 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณนี้ เราจะให้หน่วยข้อมูลที่ 6, 7, 8 และ 9 แทนด้วยหน่วยข้อมูลลำดับที่ 1, 2, 3 และ 4 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^4 L_j}{d_4 - s_1} > b_{\max} \cdot R_s \right)$$

หน่วยข้อมูลลำดับที่ 1 => เป็นหน่วยข้อมูลนอนคิวโอเอส

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(2+4) \times 10^3}{13-10} = 2 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 3} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(2+4+6) \times 10^3}{14-10} = 3 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 4} \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(2+4+6+2) \times 10^3}{16-10} = 2.33 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_S = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_S} \right\rceil$$

เริ่มแรกให้ $q = 1$ นั่นคือตรวจสอบทุกหน่วยข้อมูลว่าหน่วยข้อมูลใดควรเป็นตัวกำหนดอัตราบิต (ลำดับที่ 1, 2, 3 และ 4 ในบัฟเฟอร์คือหน่วยข้อมูลที่ 6, 7, 8 และ 9)

ลำดับที่ $k = 1$ ไม่คิดเพราะหน่วยข้อมูลนอกคิวโอเอสไม่มีเส้นตาย

$$\text{ลำดับที่ } k = 2 \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(2+4) \times 10^3}{13-10} = 2 \times 10^3$$

$$\text{ลำดับที่ } k = 3 \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(2+4+6) \times 10^3}{14-10} = 3 \times 10^3$$

$$\text{ลำดับที่ } k = 4 \Rightarrow \frac{\sum_{j=1}^4 L_j}{d_4 - s_1} = \frac{(2+4+6+2) \times 10^3}{16-10} = 2.33 \times 10^3$$

ได้ว่า $k_0 = 3$ นั่นคือหน่วยข้อมูลที่ 8 (หน่วยข้อมูลลำดับที่ 3) เป็นตัวกำหนดอัตราบิตให้หน่วยข้อมูลลำดับที่ $k = 1, 2$ และ 3 โดย

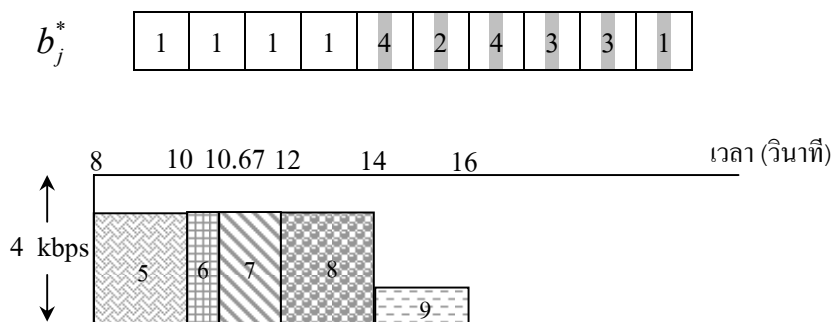
$$b_6 = b_7 = b_8 = p = \left\lceil \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} \cdot \frac{1}{R_S} \right\rceil = \left\lceil \frac{(2+4+6) \times 10^3}{14-10} \cdot \frac{1}{10^3} \right\rceil = 3$$

ดังนั้นจึงสามารถเริ่มส่งหน่วยข้อมูลที่ 6 ได้ในวินาทีที่ 10 และส่งสำเร็จที่วินาทีที่ 10.67 และเริ่มส่งหน่วยข้อมูลที่ 7 ได้ในวินาทีที่ 10.67 และส่งสำเร็จที่วินาทีที่ 12 และเริ่มส่งหน่วยข้อมูลที่ 8 ได้ในวินาทีที่ 12 และส่งสำเร็จที่วินาทีที่ 14

ต่อมาในรอบที่สองให้ $q = 4$ แต่ในรอบนี้เหลือหน่วยข้อมูลที่ 9 (หน่วยข้อมูลลำดับที่ 4) เพียงหน่วยเดียว ดังนั้น $k_0 = 4$

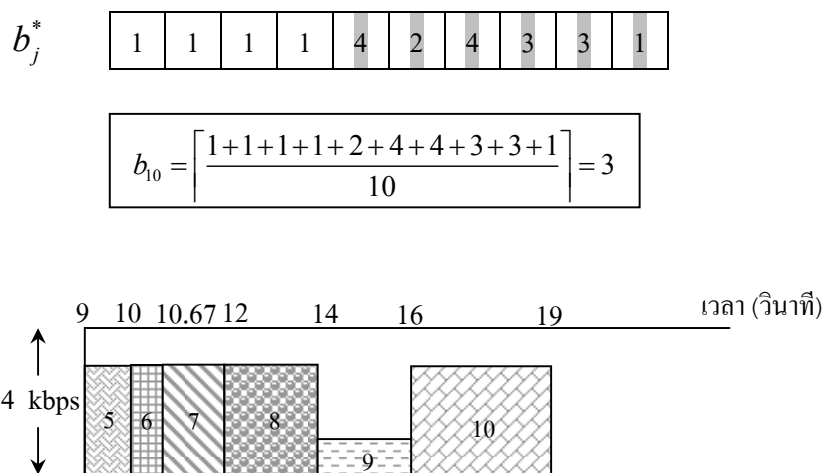
$$b_8 = p = \left[\frac{\sum_{j=4}^4 L_j}{d_4 - s_4} \cdot \frac{1}{R_s} \right] = \frac{(2) \times 10^3}{16 - 14} \cdot \frac{1}{10^3} = 1$$

เราสามารถเริ่มส่งหน่วยข้อมูลที่ 9 ได้ในวินาทีที่ 14 และส่งสำเร็จที่วินาทีที่ 16 และทำการเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 7 และ 8 ใหม่เป็น 3 และ 3 รวมทั้งเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 9 ไว้ ($b_9 = 1$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลอนคิโวเอสในลำดับถัดๆ ไป



รูปที่ 60 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 8

ณ เวลา 9 วินาที : หน่วยข้อมูลที่ 10 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 5 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 10 นี้ลงบัฟเฟอร์ เนื่องจากหน่วยข้อมูลที่ 10 เป็นหน่วยข้อมูลอนคิโวเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิโวเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 จากการคำนวณในภาพผนวกที่ 61 ได้ค่าเฉลี่ย 3 บิตต่อสัญลักษณ์ นั่นคือจะเริ่มส่งหน่วยข้อมูลที่ 10 ได้ในวินาทีที่ 16 และสามารถส่งสำเร็จได้ในวินาทีที่ 19



รูปที่ 61 การคำนวณหาอัตราบิตด้วยวิธี LS ณ วินาทีที่ 9

ณ เวลา 10 วินาที : หน่วยข้อมูลที่ 5 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 6 ต่อ

ณ เวลา 10.667 วินาที : หน่วยข้อมูลที่ 6 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 7 ต่อ

ณ เวลา 12 วินาที : หน่วยข้อมูลที่ 7 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 8 ต่อ

ณ เวลา 14 วินาที : หน่วยข้อมูลที่ 8 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 9 ต่อ

ณ เวลา 16 วินาที : หน่วยข้อมูลที่ 9 ส่งสำเร็จ ให้ส่งหน่วยข้อมูลที่ 10 ต่อ

ณ เวลา 19 วินาที : หน่วยข้อมูลที่ 10 ส่งสำเร็จพอดี

ตัวอย่างการประหยัดพลังงานด้วยวิธี RRP w/ RPS

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 4 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

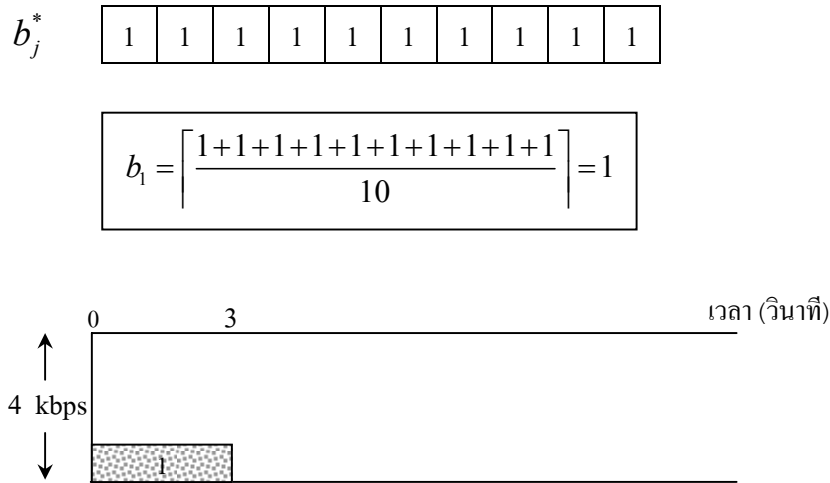
ตารางผนวกที่ 4 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี RRP w/ RPS และ LS w/ RPS

หน่วยข้อมูลที่	ชนิดของหน่วยข้อมูล	เวลาที่หน่วยข้อมูลมาถึง (วินาที)	จำนวนบิตของหน่วยข้อมูล (กิโลบิต)	เส้นตาย (วินาที)
1	NQoS	0	3	-
2	QoS	1	11	6
3	QoS	2	2	4

การทำงานของ RRP w/ RPS จะคล้ายกับ RRP เกือบทั้งหมดดังนั้นกระบวนการที่ซับซ้อนกว่านี้ให้ดูได้ที่ตัวอย่างการประหยัดพลังงานด้วยวิธี RRP เนื่องจากข้อมูลตัวอย่างในตารางผนวกที่ 3 ไม่ได้ทำให้การทำงานของ RRP และ RRP w/ RPS แตกต่างกัน

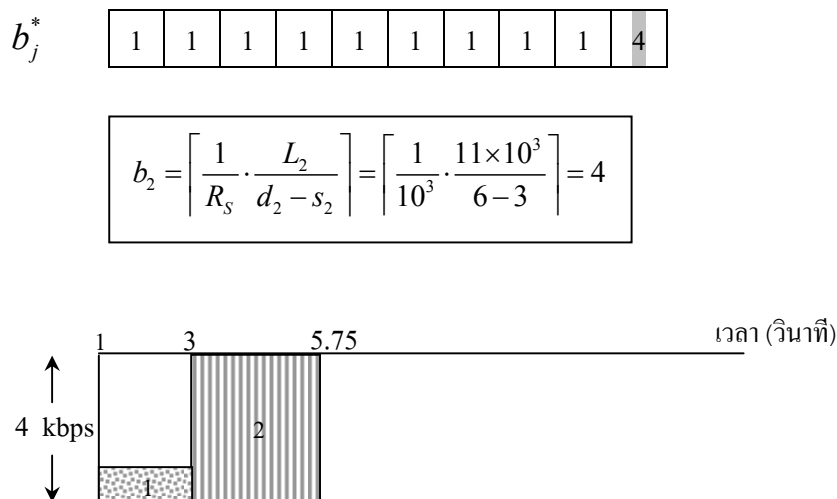
การทำงานของวิธี RRP w/ RPS ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิวโอเอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิวโอเอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 แต่เนื่องจากหน่วยข้อมูลนี้เป็นหน่วยข้อมูลแรก เราจึงกำหนดค่าอัตราบิตของหน่วยข้อมูลคิวโอเอสในลำดับเริ่มต้น 10 ลำดับไว้ที่อัตราบิตต่ำสุด (ในที่นี้คือ 1 kbps) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 1 นี้คือ 1 kbps จากภาพผนวกที่ 62 จึงเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 3



ภาพผนวกที่ 62 การคำนวณหาอัตราบิตด้วยวิธี RRP w/ RPS ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยสมการในตารางที่ 6 ได้ค่าดังแสดงในภาพผนวกที่ 63 คือเมื่อส่งหน่วยข้อมูลที่ 2 นี้ด้วยอัตราบิต 4 kbps จะทำให้ส่งได้ทันเส้นตาย ดังนั้นจึงเริ่มส่งหน่วยข้อมูลนี้ที่ วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 5.75 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ไว้ ($b_2 = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป



ภาพผนวกที่ 63 การคำนวณหาอัตราบิตด้วยวิธี RRP w/ RPS ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ จึงต้องมีการคำนวณหาอัตราบิตใหม่ให้กับหน่วยข้อมูลที่มีการเปลี่ยนลำดับ นั่นคือคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 แต่เมื่อคำนวณอัตราบิตตามตารางที่ 6 แล้วจะเห็นว่าหน่วยข้อมูลที่ 3 ไม่สามารถส่งได้ทันเส้นตาย จึงทำการคำนวณซ้ำตามภาพที่ 21 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 3 และ 2 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 2 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ใน ส่วนการคำนวณซ้ำ เราจะให้หน่วยข้อมูลที่ 3 และ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1 และ 2 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_S \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(2) \times 10^3}{4 - 3} = 2 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(2 + 11) \times 10^3}{6 - 3} = 4.33 \times 10^3 \text{ bps}$$

จะเห็นว่าต้องทิ้งบางหน่วยข้อมูล เนื่องจากไม่สามารถส่งหน่วยข้อมูลเหล่านั้นด้วยอัตราบิตที่สูงกว่าอัตราบิตสูงสุดได้ $\Rightarrow b_{\max} \cdot R_S = 4 \times 10^3 \text{ bps}$ ในกระบวนการเลือกทิ้งนี้ ถ้าเป็นวิธี RRP จะทิ้งหน่วยข้อมูลใหม่ที่เพิ่งเข้ามานั่นคือหน่วยข้อมูลที่ 3 (หรือหน่วยข้อมูลลำดับที่ 1 นั่นเอง) แต่สำหรับวิธี RRP w/ RPS จะเลือกทิ้งหน่วยข้อมูลที่เป็นต้องส่งด้วยความเร็วสูงกว่าอัตราบิตสูงสุดโดยเลือกจากหน่วยข้อมูลลำดับแรกของบัฟเฟอร์ นั่นคือในตัวอย่างนี้หน่วยข้อมูลลำดับที่ 2 ในบัฟเฟอร์ (หน่วยข้อมูลที่ 2) ต้องส่งด้วยอัตราบิต 4.33 kbps จึงทิ้งหน่วยข้อมูลนี้ แล้วจะได้ว่าเริ่มส่งหน่วยข้อมูลที่ 3 ได้ที่วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 5 และทำการเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ใหม่เป็นอัตราบิตสูงสุด ($b_{\max} = 4$) เนื่องจากถูกทิ้ง และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 3 ไว้ที่ 2 เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป ดังภาพผนวกที่ 64

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & \\ \hline \end{array}$$

$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{(2) \times 10^3}{4 - 3} \right\rceil = 2$$



รูปที่ 64 การคำนวณหาอัตราบิตด้วยวิธี RRP w/ RPS ณ วินาทีที่ 2

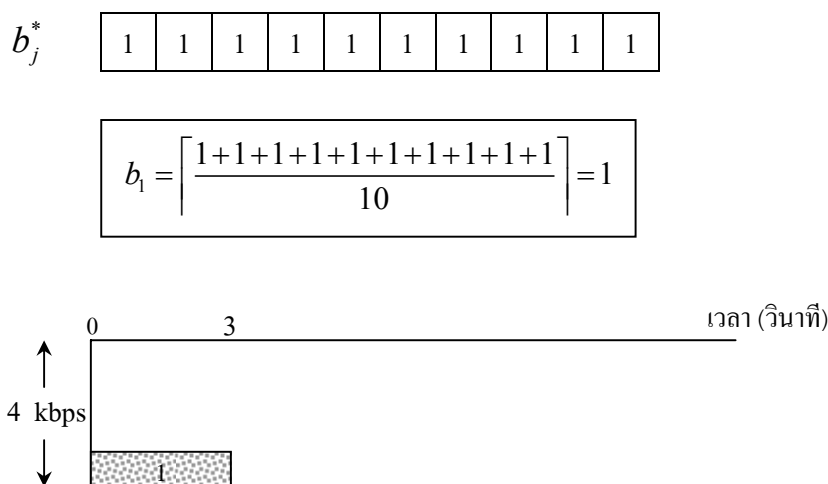
ตัวอย่างการประหยัดพลังงานด้วยวิธี LS w/ RPS

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่าน โหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 4 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็สามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

การทำงานของ LS w/ RPS จะคล้ายกับ LS เกือบทั้งหมดดังนั้นกระบวนการที่ซับซ้อนกว่านี้ให้ดูได้ที่ตัวอย่างการประหยัดพลังงานด้วยวิธี LS เนื่องจากข้อมูลตัวอย่างในตารางผนวกที่ 3 ไม่ได้ทำให้การทำงานของ LS และ LS w/ RPS แตกต่างกัน

การทำงานของวิธี LS w/ RPS ในแต่ละเวลาเป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนด และช่องสัญญาณว่างอยู่ หน่วยข้อมูลนี้เป็นหน่วยข้อมูลนอนคิโอสจึงส่งด้วยอัตราบิตที่ได้จากค่าเฉลี่ยของหน่วยข้อมูลคิโอส 10 หน่วยในอันดับก่อนหน้าดังสมการในตารางที่ 6 แต่เนื่องจากหน่วยข้อมูลนี้เป็นหน่วยข้อมูลแรก เราจึงกำหนดค่าอัตราบิตของหน่วยข้อมูลคิโอสในลำดับเริ่มต้น 10 ลำดับไว้ที่อัตราบิตต่ำสุด (ในที่นี้คือ 1 kbps) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 1 นี้คือ 1 kbps จากภาพผนวกที่ 65 จึงเริ่มส่งหน่วยข้อมูลนี้ที่วินาทีที่ 0 และจะเสร็จสิ้นที่วินาทีที่ 3



ภาพผนวกที่ 65 การคำนวณหาอัตราบิตด้วยวิธี LS w/ RPS ณ วินาทีที่ 0

ณ เวลา 1 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 2 นี้ลงบัฟเฟอร์ และหาอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 เนื่องจากหน่วยข้อมูลที่ 2 นี้เป็นหน่วยข้อมูลคิวโอเอส จึงต้องคำนวณหาอัตราบิตด้วยกระบวนการในภาพที่ 21 ได้ค่าดังแสดงในภาพผนวกที่ 66 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 2 อยู่ในบัฟเฟอร์เพียงหน่วยข้อมูลเดียว แล้วนำหน่วยข้อมูลนั้นมาตรวจสอบว่าถ้าเร่งความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในสถานการณ์นี้ เราจะให้หน่วยข้อมูลที่ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1)

$$\left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} > b_{\max} \cdot R_s \right)$$

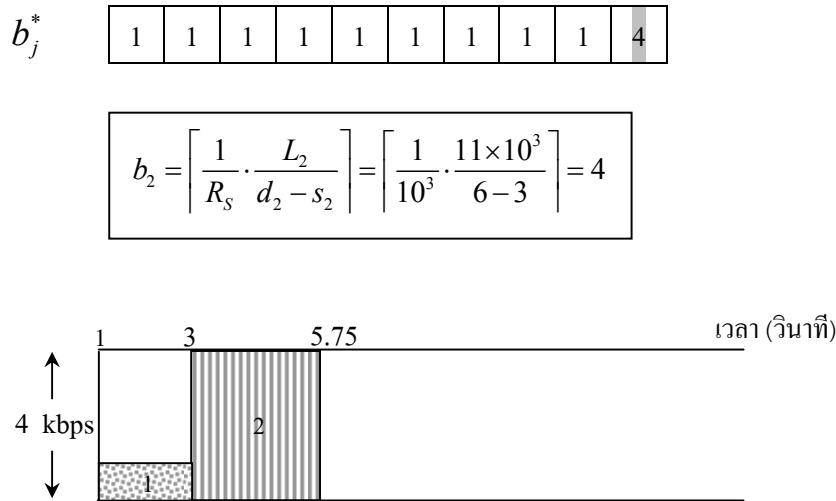
$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(11) \times 10^3}{6 - 3} = 4 \times 10^3 \text{ bps}$$

จะเห็นว่าทุกหน่วยข้อมูลในบัฟเฟอร์ สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตไม่เกินอัตราบิตสูงสุด $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ก็สามารถส่งได้ทันเส้นตาย

ต่อมาเป็นการหาอัตราบิตที่ต่ำที่สุดสำหรับหน่วยข้อมูลคิวโอเอสที่ทำให้ส่งทุกข้อมูลได้

$$k_0 = \underset{q \leq k \leq m, k \in Q}{\operatorname{argmax}} \left\{ \frac{\sum_{j=q}^k L_j}{d_k - s_q} \right\}; \quad p = \left\lceil \frac{\sum_{j=q}^{k_0} L_j}{d_{k_0} - s_q} \cdot \frac{1}{R_s} \right\rceil$$

เนื่องจากมีหน่วยข้อมูลเดียว หน่วยข้อมูลนั้นจึงเป็นตัวกำหนดอัตราบิต ($k_0 = 1$) ดังนั้นอัตราบิตที่ใช้ในการส่งหน่วยข้อมูลที่ 2 คือ $b_2 = p = \left\lceil \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} \cdot \frac{1}{R_s} \right\rceil = \left\lceil \frac{11 \times 10^3}{6 - 3} \cdot \frac{1}{10^3} \right\rceil = 4$ จึงเริ่มส่งหน่วยข้อมูลที่ 2 ได้ในวินาทีที่ 3 และส่งสำเร็จที่วินาทีที่ 5.75 และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ไว้ ($b_2 = 4$) เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอนคิวโอเอสในลำดับถัดๆ ไป



ภาพผนวกที่ 66 การคำนวณหาอัตราบิตด้วยวิธี LS w/ RPS ณ วินาทีที่ 1

ณ เวลา 2 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ระบบยังส่งหน่วยข้อมูลที่ 1 ไม่เสร็จ จึงเก็บหน่วยข้อมูลที่ 3 นี้ลงบัฟเฟอร์ สมมติว่าหลังจากการจัดลำดับทำให้ลำดับการส่งออกข้อมูลเป็นหน่วยข้อมูลที่ 3 และ 2 ตามลำดับ นั่นคือการคำนวณหาอัตราบิตสำหรับหน่วยข้อมูลที่ 3 แล้วตามด้วยอัตราบิตสำหรับหน่วยข้อมูลที่ 2 และคำนวณด้วยกระบวนการตามภาพที่ 21 ซึ่งสามารถอธิบายได้ดังนี้

ในขณะนี้หน่วยข้อมูลที่ 3 และ 2 เรียงอยู่ในบัฟเฟอร์ตามลำดับ จึงมีอยู่ 2 หน่วยข้อมูลที่ต้องนำมาตรวจสอบว่าถ้าแรงความเร็วแล้วจะสามารถส่งหน่วยข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ได้ทันหรือไม่ (ในส่วนการคำนวณซ้ำ เราจะให้หน่วยข้อมูลที่ 3 และ 2 แทนด้วยหน่วยข้อมูลลำดับที่ 1 และ 2 ตามลำดับ)

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(2) \times 10^3}{4 - 3} = 2 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(2 + 11) \times 10^3}{6 - 3} = 4.33 \times 10^3 \text{ bps}$$

จะเห็นว่าต้องทิ้งบางหน่วยข้อมูล เนื่องจากไม่สามารถส่งหน่วยข้อมูลเหล่านั้นด้วยอัตราบิตที่สูงกว่าอัตราบิตสูงสุดได้ $\Rightarrow b_{\max} \cdot R_s = 4 \times 10^3 \text{ bps}$ ในกระบวนการเลือกทิ้งนี้ ถ้าเป็นวิธี LS จะทิ้งหน่วย

ข้อมูลใหม่ที่เพิ่งเข้ามานั้นคือหน่วยข้อมูลที่ 3 (หรือหน่วยข้อมูลลำดับที่ 1 นั่นเอง) แต่สำหรับวิธี LS w/ RPS จะเลือกทั้งหน่วยข้อมูลที่จำเป็นต้องส่งด้วยความเร็วสูงกว่าอัตราบิตสูงสุดโดยเลือกจากหน่วยข้อมูลลำดับแรกของบัฟเฟอร์ นั่นคือในตัวอย่างนี้หน่วยข้อมูลลำดับที่ 2 ในบัฟเฟอร์ (หน่วยข้อมูลที่ 2) ต้องส่งด้วยอัตราบิต 4.33 kbps จึงทิ้งหน่วยข้อมูลนี้ แล้วจะได้ว่าเริ่มส่งหน่วยข้อมูลที่ 3 ได้ที่วินาทีที่ 3 และจะเสร็จสิ้นที่วินาทีที่ 5 และทำการเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 2 ใหม่เป็นอัตราบิตสูงสุด ($b_{max} = 4$) เนื่องจากถูกทิ้ง และเก็บค่าจำนวนบิตต่อสัญลักษณ์ของหน่วยข้อมูลที่ 3 ไว้ที่ 2 เพื่อนำไปหาอัตราบิตในการส่งหน่วยข้อมูลนอกคิวโอเอสในลำดับถัดๆ ไป ดังภาพผนวกที่ 67

$$b_j^* \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 \\ \hline \end{array}$$

$$b_3 = \left\lceil \frac{1}{R_s} \cdot \frac{L_3}{d_3 - s_3} \right\rceil = \left\lceil \frac{1}{10^3} \cdot \frac{(2) \times 10^3}{4 - 3} \right\rceil = 2$$



รูปที่ 67 การคำนวณหาอัตราบิตด้วยวิธี LS w/ RPS ณ วินาทีที่ 2

ตัวอย่างการเลือกทั้งหน่วยข้อมูลคิวโอเอสที่จำเป็นต้องทิ้งมากกว่าหนึ่งหน่วยของวิธี RRP w/ RPS และวิธี LS w/ RPS

สมมติให้ชุดข้อมูลที่ต้องการส่งผ่านโหนด ๆ หนึ่งมีค่าดังในตารางผนวกที่ 5 ซึ่งข้อมูลเหล่านี้จะส่งออกทางช่องสัญญาณขาออกด้วยอัตราสัญลักษณ์ (symbol rate) คงที่ที่ 1 kHz และแปรค่าจำนวนบิตต่อสัญลักษณ์ (b_i) ในแต่ละหน่วยข้อมูล i ได้ตั้งแต่ $b_{min}=1$ ถึง $b_{max}=4$ บิตต่อสัญลักษณ์ ซึ่งก็ยังสามารถแปรค่าอัตราบิต (bit rate) ได้ตั้งแต่ 1 ถึง 4 kbps

ตารางผนวกที่ 5 ตัวอย่างชุดของหน่วยข้อมูลที่ต้องการส่งผ่านโหนดๆ หนึ่ง สำหรับวิธี RRP w/ RPS และ LS w/ RPS ในกรณีที่ต้องเลือกทิ้งหลายหน่วยข้อมูล

หน่วยข้อมูลที่	ชนิดของหน่วยข้อมูล	เวลาที่หน่วยข้อมูลมาถึง (วินาที)	จำนวนบิตของหน่วยข้อมูล (กิโลบิต)	เส้นตาย (วินาที)
1	QoS	0	4	1
2	QoS	0.5	1	3.1
3	QoS	0.7	1	3.2
4	QoS	0.8	12	3

การทำงานของทั้งวิธี RRP w/ RPS และ LS w/ RPS เป็นดังนี้

ณ เวลา 0 วินาที : หน่วยข้อมูลที่ 1 เข้ามายังโหนดและขณะนี้ช่องสัญญาณว่าง จึงเริ่มส่งออกหน่วยข้อมูลนี้ด้วยอัตราบิต 4 kbps จะทำให้สามารถส่งได้ทันเส้นตายพอดี ดังนั้นหน่วยข้อมูลที่ 1 จะเริ่มส่ง ณ วินาทีที่ 0 และส่งสำเร็จ ณ วินาทีที่ 1

ณ เวลา 0.5 วินาที : หน่วยข้อมูลที่ 2 เข้ามายังโหนด แต่ขณะนี้ช่องสัญญาณยังไม่ว่าง จึงเก็บเข้าสู่บัฟเฟอร์ และเมื่อทำการคำนวณการส่งแล้วจะพบว่าระบบสามารถส่งหน่วยข้อมูลนี้ได้ทัน ดังนี้

$$\left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^1 L_j}{d_1 - s_1} > b_{\max} \cdot R_s \right)$$

หน่วยข้อมูลลำดับที่ 1 (หน่วยข้อมูลที่ 2) $\Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(1) \times 10^3}{3.1 - 1} = 0.48 \times 10^3$ bps จะเห็นว่าหน่วย

ข้อมูลที่ 2 สามารถส่งสำเร็จได้เนื่องจากใช้อัตราบิตในการส่งไม่เกินอัตราบิตสูงสุด (4 kbps) หน่วยข้อมูลนี้จึงทำการรอการส่ง

ณ เวลา 0.7 วินาที : หน่วยข้อมูลที่ 3 เข้ามายังโหนด แต่ขณะนี้ช่องสัญญาณยังไม่ว่าง จึงเก็บเข้าสู่บัฟเฟอร์ ณ เวลานี้ในบัฟเฟอร์มี 2 หน่วยข้อมูล สมมติให้ลำดับการส่งเป็นหน่วยข้อมูลที่ 2 แล้วตามด้วยหน่วยข้อมูลที่ 3 ตามลำดับ เมื่อทำการคำนวณ ทั้งสองหน่วยสามารถส่งได้ทันเส้นตาย ดังนี้

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

หน่วยข้อมูลลำดับที่ 1 (หน่วยข้อมูลที่ 2) $\Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(1) \times 10^3}{3.1 - 1} = 0.48 \times 10^3$ bps

หน่วยข้อมูลลำดับที่ 2 (หน่วยข้อมูลที่ 3) $\Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(1+1) \times 10^3}{3.2 - 1} = 0.91 \times 10^3$ bps

จะเห็นว่าหน่วยข้อมูลทั้งสองสามารถส่งสำเร็จได้ เนื่องจากทั้งสองหน่วยข้อมูลใช้อัตราบิตในการส่งไม่เกินอัตราบิตสูงสุด (4 kbps) หน่วยข้อมูลที่ 2 และ 3 จึงทำการรอการส่งตามลำดับ

ณ เวลา 0.8 วินาที : หน่วยข้อมูลที่ 4 เข้ามายังโหนด แต่ขณะนี้ช่องสัญญาณยังไม่ว่าง จึงเก็บเข้าสู่บัฟเฟอร์ ณ เวลานี้ในบัฟเฟอร์มี 3 หน่วยข้อมูล สมมติให้ลำดับการส่งเป็นหน่วยข้อมูลที่ 4, 2 และ 3 ตามลำดับ เมื่อทำการคำนวณ ทั้งสามหน่วยข้อมูล จะเห็นว่าจำเป็นต้องทิ้งบางหน่วยข้อมูลเพื่อให้หน่วยข้อมูลที่เหลือสามารถส่งได้ทันเส้นตาย การเลือกทิ้งบางหน่วยข้อมูลด้วยวิธี RRP w/ RPS และวิธี LS w/ RPS เป็นการทิ้งหน่วยข้อมูลที่ต้องส่งด้วยอัตราบิตสูงกว่าอัตราบิตสูงสุดจึงจะส่งได้ทันเส้นตาย โดยเลือกทิ้งตามลำดับการจัดเรียงในบัฟเฟอร์ ดังนี้

$$\left(\frac{\sum_{j=1}^3 L_j}{d_3 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^3 L_j}{d_3 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1 (หน่วยข้อมูลที่ 4)} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{3-1} = 4 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2 (หน่วยข้อมูลที่ 2)} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(8+1) \times 10^3}{3.1-1} = 4.29 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 3 (หน่วยข้อมูลที่ 3)} \Rightarrow \frac{\sum_{j=1}^3 L_j}{d_3 - s_1} = \frac{(8+1+1) \times 10^3}{3.2-1} = 4.55 \times 10^3 \text{ bps}$$

จะเห็นว่าจำเป็นต้องทิ้งบางหน่วยข้อมูล ถ้าเป็นวิธี RRP และ LS จะเลือกทิ้งหน่วยข้อมูลใหม่ในที่นี้คือ หน่วยข้อมูลลำดับที่ 4 (หน่วยข้อมูลลำดับแรกในบัฟเฟอร์) โดยเมื่อทิ้งหน่วยข้อมูลนี้แล้วจะทำให้อีกสอง หน่วยข้อมูลทำการส่งสำเร็จได้ แต่สำหรับวิธี RRP w/ RPS และ LS w/ RPS จะเลือกหน่วยข้อมูลที่มีความเร็วสูงกว่าอัตราบิตสูงสุด ซึ่งเลือกทิ้งตามลำดับการจัดเรียงในบัฟเฟอร์ ในที่นี้หน่วยข้อมูลลำดับที่ 2 (หน่วยข้อมูลลำดับที่ 2) เป็นหน่วยข้อมูลลำดับแรกที่ใช้ความเร็วเกินกว่าอัตราบิตสูงสุด จึงเลือกหน่วยข้อมูลลำดับที่ 2 ทิ้งไป ในขณะนี้เหลือหน่วยข้อมูลในบัฟเฟอร์สองหน่วย แล้วทำการตรวจสอบความเร็วใหม่

$$\left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} < 0 \right) \text{ หรือ } \left(\frac{\sum_{j=1}^2 L_j}{d_2 - s_1} > b_{\max} \cdot R_s \right)$$

$$\text{หน่วยข้อมูลลำดับที่ 1 (หน่วยข้อมูลที่ 4)} \Rightarrow \frac{\sum_{j=1}^1 L_j}{d_1 - s_1} = \frac{(8) \times 10^3}{3-1} = 4 \times 10^3 \text{ bps}$$

$$\text{หน่วยข้อมูลลำดับที่ 2 (หน่วยข้อมูลที่ 3)} \Rightarrow \frac{\sum_{j=1}^2 L_j}{d_2 - s_1} = \frac{(8+1) \times 10^3}{3.2-1} = 4.1 \times 10^3 \text{ bps}$$

จะเห็นว่าจำเป็นต้องทิ้งบางหน่วยข้อมูลอีกครั้ง ซึ่งถ้าเป็นวิธี RRP และ LS จะเลือกทิ้งหน่วยข้อมูลใหม่เพียงหน่วยข้อมูลเดียว สำหรับวิธี RRP w/ RPS และ LS w/ RPS การเลือกทิ้งในรอบนี้ก็จะทำเช่นเดิม จึงเลือกหน่วยข้อมูลลำดับที่ 3 ทิ้งไป ในขณะนี้เหลือหน่วยข้อมูลในบัฟเฟอร์เพียงหน่วยเดียว ซึ่งหน่วยข้อมูลลำดับที่ 4 นี้สามารถส่งสำเร็จได้ เนื่องจากใช้อัตราบิตในการส่งไม่เกินอัตราบิตสูงสุด (4 kbps) หน่วยข้อมูลนี้จึงทำการรอการส่งตามลำดับ