การควบคุมคุณภาพการให้บริการ สำหรับการประมวลผลแบบกริดด้วยเอเจนต์

GRID SERVICE MANAGEMENT USING AGENT BASE QUALITY OF SERVICE

มนตรี บุญเรืองเศษ
MONTREE BUNRUANGSES

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-15-1317-8

การควบคุมคุณภาพการให้บริการ
สำหรับการประมวลผลแบบกริดด้วยเอเจนต์

GRID SERVICE MANAGEMENT
USING AGENT BASE QUALITY OF SERVICE

มนตรี บุญเรืองเศษ
MONTREE BUNRUANGSES

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-15-1317-8

GRID SERVICE MANAGEMENT
USING AGENT BASE QUALITY OF SERVICE

MONTREE BUNRUANGSES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDENT
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2004

ISBN 974-15-1317-8

COPYRIGHT 2004
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การควบคุมคุณภาพการให้บริการสำหรับการประมวลผลแบบกริดด้วยเอเจนต์
นักศึกษา	นายมนตรี บุญเรืองเศษ
รหัสนักศึกษา	42061173
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ประทีป บัญญัตินพรัตน์
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	รศ. บรรจง ปิยธำรง

บทคัดย่อ

ระบบการประมวลผลแบบกริดพัฒนาขึ้นมาสำหรับรองรับการประมวลผลและใช้ทรัพยากรร่วมกันของระบบเครือข่ายคอมพิวเตอร์แบบกระจาย ปัจจุบันได้มีการนำเสนอสถาปัตยกรรมบริการของกริดแบบเปิด(Open Grid Service Architecture: OGSA) ซึ่งเป็นการทำงานของกริดบนเทคโนโลยีของเว็บเซอร์วิสซ์ที่ให้บริการข้อมูลแบบกระจายบนอินเตอร์เน็ต วิทยานิพนธ์นี้นำเสนอการควบคุมคุณภาพการให้บริการ (QoS) สำหรับการประมวลผลแบบกริดด้วยเอเจนต์เพื่อเพิ่มประสิทธิภาพการให้บริการและจัดการทรัพยากรของระบบในระดับงานประยุกต์หลังจากค้นพบที่อยู่แล้ว โดยประยุกต์ใช้โครงสร้างข้อมูลแบบฮีพ (Heap) และการเชื่อมต่อความจุแบบต่ำลง (Sink Link-Capacity) เข้ามาจัดการกับโครงสร้างของระบบเอเจนต์เพื่อกำหนดการควบคุมคุณภาพการให้บริการในรูปแบบของลำดับสำคัญการให้บริการ(Priority of Service) โดยพัฒนาอัลกอริทึมแบบมัลติโปรเซสส์ไพรออริตี้เป็นอัลกอริทึมในการจัดการแบนด์วิดท์สำหรับงานประยุกต์บนการประมวลผลแบบกริด

Thesis Title	Grid Service Management using Agent-base Quality of Service
Student	Mr. Montree Bunruangses
Student ID.	42061173
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2004
Thesis Advisor	Assoc.Prof. Prateep Bunyatnoparat
Thesis Advisor	Assoc.Prof. Banjong Piyatamrong

ABSTRACT

Grid computing technologies are adopted to enable wide-spread sharing and coordinated use of network resources. Grid Service is a part of Grid infrastructure that enables the integration of service across distributed resources based on Service-Oriented architecture. In particular, the Open Grid Service Architecture, the new Grid Service model based on Web Service technologies. In various fields of distributed computing, software agents are expected to play an increasing important role for improving performance such as scalability and availability. This thesis applied a heap structure and sink link capacity principle to manage agent structure system for the quality of service (QoS). The QoS concepts purpose the Multi-Process priority algorithm to manages the bandwidth for the Grid Applications.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ประทีบ บัญญัตินพรัตน์ และ รศ. บรรจง ปิยธำรง ที่คอยช่วยเหลือและให้คำแนะนำในหัวข้อการทำวิจัย ผู้เขียนขอขอบพระคุณอาจารย์ทั้งสองด้วยความเคารพยิ่ง

ขอขอบคุณกรรมการที่สอบหัวข้อและกรรมการสอบวิทยานิพนธ์ ที่ให้ข้อเสนอแนะและแก้ไขในการเขียนวิทยานิพนธ์เล่มนี้ให้สำเร็จลงด้วยดี

ขอขอบคุณ เจ้าหน้าที่บัณฑิตวิทยาลัยคณะวิศวกรรมศาสตร์ คอยอำนวยความสะดวกในเรื่องเอกสารและให้คำแนะนำเกี่ยวกับการสอบวิทยานิพนธ์

ขอขอบคุณ เพื่อนในกลุ่มนายวีระวัฒน์ ภูมิพัฒนพงษ์ ที่ให้คำแนะนำตลอดจนเอกสารตำราที่เป็นประโยชน์ที่เกี่ยวข้องกับวิทยานิพนธ์

ผลดีที่เกิดจากวิทยานิพนธ์เล่มนี้ขอมอบให้แก่บิดามารมารดา ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้อบรมสั่งสอนความรู้ที่ดีแก่ข้าพเจ้าตลอดมา

มนตรี บุญเรืองเศษ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของการศึกษา.....1

1.3 สมมติฐานของการศึกษา.....2

1.4 แนวความคิดที่ใช้ในการวิจัย.....3

1.5 ขอบเขตของการวิจัย.....3

1.6 ส่วนประกอบของวิทยานิพนธ์.....4

บทที่ 2 การประมวลผลแบบกริดและเวบเซอร์วิส.....6

2.1 แนวคิดการประมวลผลแบบกริด.....6

2.2 วิวัฒนาการของระบบการประมวลผลแบบกริด.....6

2.3 องค์ประกอบของการประมวลผลแบบกริด.....8

2.4 โครงสร้างสถาปัตยกรรมของระบบประมวลผลแบบกริด.....10

2.4.1 ชั้นฟาบริก.....12

2.4.2 ชั้นการติดต่อ.....12

2.4.3 ชั้นทรัพยากร.....13

2.4.4 ชั้นการรวม 14

2.4.5 ชั้นงานประยุกต์.....16

2.5 การให้บริการเวบเซอร์วิส.....16

2.6 ระบบกริดสำหรับการบริการบนเวบเซอร์วิส.....18

2.7 ระบบเอเจนต์ 21

2.7.1 ซอฟท์แวร์เอเจนต์.....22

2.7.2 ซิงเกิลเอเจนต์และมัลติเอเจนต์.....22

สารบัญ (ต่อ)

หน้า

2.7.3 เทคโนโลยีของเอเจนต์.....23

บทที่ 3 การควบคุมคุณภาพการให้บริการ(QoS) 25

3.1 บทนำ.....25

3.2 หลักในการควบคุมคุณภาพการให้บริการ.....25

3.2.1 รูปแบบในการควบคุมคุณภาพการให้บริการ 26

3.2.2 ข้อกำหนดของการควบคุมคุณภาพการให้บริการ.....27

3.2.3 กลไกในการของการควบคุมคุณภาพการให้บริการ.....27

3.2.3.1 การจัดหา.....27

3.2.3.2 การควบคุม.....28

3.2.3.2 การจัดการ.....28

3.2.4 ประเภทของข้อกำหนดในการควบคุมการให้บริการ 29

3.2.5 เมตริกสำหรับควบคุมคุณภาพการให้บริการ.....29

3.2.6 โครงสร้างสถาปัตยกรรมพื้นฐานของการทำQoS.....30

3.2.6.1 การควบคุมคุณภาพการให้บริการในเครือข่ายเดียว.....30

3.2.6.2 การสร้างและการทำเครื่องหมาย.....31

3.2.6.3 การใช้นโยบายและการจัดการ 32

3.2.4 การควบคุมคุณภาพการให้บริการแบบจุดต่อจุด.....32

3.3 ประเภทของการทำการควบคุมคุณภาพการให้บริการ.....33

3.3.1 การควบคุมคุณภาพการให้บริการระดับเนตเวิร์ค.....33

3.3.2 การควบคุมคุณภาพการให้บริการระดับงานประยุกต์.....34

3.4 การจัดการแบนด์วิดท์.....35

3.5 การควบคุมคุณภาพการให้บริการบนระบบกริด.....40

บทที่ 4 การควบคุมคุณภาพการให้บริการระบบกริดโดยระบบมัลติ-เอเจนต์.....45

4.1 บทนำ.....45

4.2 การประยุกต์ใช้โครงสร้างข้อมูลแบบฮีพ.....46

4.2.1 หลักพื้นฐานโครงสร้างข้อมูลแบบฮีพ.....47

สารบัญ (ต่อ)

หน้า

4.2.2 การประยุกต์ใช้งานฮีพสำหรับการบริการแบบลำดับความสำคัญ.....47

4.3 แนวคิดของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม.....48

4.3.1 แนวคิดการทำงานแบบขนานในโครงสร้างแบบฮีพ.....48

4.3.2 แนวคิดการฝากข้อมูลในโครงสร้างแบบฮีพ..... .49

4.4 ส่วนประกอบและการทำงานของระบบมัลติเอเจนต์.....52

4.4.1 องค์ประกอบของเอเจนต์ 53

4.4.2 หน้าที่การทำงานของเอเจนต์.....54

4.4.3 การประยุกต์ระบบมัลติ-เอเจนต์สำหรับระบบการบริการแบบกริด.....55

4.4.4 การจัดโครงข่ายของเอเจนต์.....57

4.4.5 ตัวควบคุมเอเจนต์ส่วนกลาง.....57

4.4.5.1 โมดูลตรวจสอบ.....58

4.4.5.2 โมดูลจัดลำดับ.....58

4.4.5.3 โมดูลควบคุม 59

4.5 การกำหนดลำดับความสำคัญให้กับเอเจนต์.....60

4.5.1 การกำหนดลำดับความสำคัญโดยแบนด์วิดท์.....60

4.5.2 การแก้ปัญหาเอเจนต์ที่มีแบนด์วิดท์เท่ากัน.....61

4.6 การควบคุมคุณภาพการให้บริการตามหลักของมัลติโปรเซสส์ไพรออริตี้.....63

4.6.1 หลักการการควบคุมคุณภาพการให้บริการของระบบ.....64

4.6.2 การทำงานในลักษณะมัลติโปรเซสส์.....65

4.6.3 การจัดการแบนด์วิดท์.....66

4.6.4 การวิเคราะห์เส้นทางในการส่งข้อมูล.....67

4.6.5 วิธีการฝากข้อมูล.....68

4.7 โฟลว์ชาร์ทแสดงการทำงานของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม.....69

บทที่ 5 การทดลองและผลการทดลอง.....71

5.1 บทนำ.....71

5.2 องค์ประกอบที่ใช้ในการทดลอง.....72

5.2.1 ขนาดของกริดและโครงสร้างของมัลติเอเจนต์.....72

5.2.2 ขนาดของแบนด์วิดท์.....73
5.3 หลักการทดลองและการวัดประสิทธิภาพ.....76
5.3.1 การวัดประสิทธิภาพ.....76
5.3.2 วิธีการในการจำลองเพื่อหาประสิทธิภาพ.....79
5.4 การทดลองกับงานประยุกต์แบบอิลาสติก.....80
5.5 การทดลองกับงานประยุกต์แบบเรียลไทม์.....87
5.6 การทดลองกับงานประยุกต์แบบอิลาสติกและแบบเรียวไทม์.....96

บทที่ 6 สรุปผลวิจัยและข้อเสนอแนะ.....99
6.1 สรุปงานวิจัย.....99
6.2 สรุปผลการทดลอง.....100
6.3 ข้อเสนอแนะและปัญหา.....101

เอกสารอ้างอิง.....102

ภาคผนวก.....106
ก. ตัวอย่างของเอเจนต์และเจ้าของผู้ผลิต.....106
ข. ตัวอย่างคำสั่งโปรแกรมของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม.....108
ค. ผลงานวิจัยในระหว่างการศึกษาที่ได้รับการตีพิมพ์เผยแพร่.....112

ประวัติผู้เขียน.....113

สารบัญตาราง

ตารางที่ หน้า

2.1 ชนิดของพอร์ทของการอินเตอร์เฟสสำหรับ OGSA..... 19

4.1 โอเปอร์เรชั่นที่ใช้จัดการโครงสร้างของฮีพ.....47

4.2 เส้นทาง(Path Flow) การส่งข้อมูลของแต่ละเอเจนต์.....50

4.3 ตัวอย่างแอตทริบิวของเอเจนต์ที่ใช้ในการควบคุมการให้บริการ.....54

4.4 ตัวอย่างงานประยุกต์และแบนด์วิดท์ที่ต้องการ.....55

5.1 แบนด์วิดท์ของกริดที่ใช้ในการทดลองชุดที่ 1 จาก 64 Kbps ถึง 4 Mbps.....74

5.2 แบนด์วิดท์ของกริดที่ใช้ในการทดลองชุดที่ 2 จาก 8 Mbps ถึง 512 Mbps.....74

5.3 ขนาดของข้อมูลแบบอิลาสติกที่ใช้ในการทดลอง.....80

5.4 ชนิดของงานประยุกต์แบบเรียลไทม์และแบนด์วิดท์ที่ต้องการที่ใช้ในการทดลอง.....87

ก. ตัวอย่างของเอเจนต์และเจ้าของผู้ผลิต.....106

สารบัญรูป

รูปที่ หน้า

2.1 แสดงลำดับตัวเลขของการประมวลผลแบบกริด.....7
2.2 องค์ประกอบพื้นฐานของระบบการประมวลผลแบบกริด.....8
2.3 โครงสร้างสถาปัตยกรรมกริดเปรียบเทียบกับโปรโตคอลอินเตอร์เน็ต.....11
2.4 ความสัมพันธ์ระหว่างชั้นต่างๆ ในการจัดการทรัพยากรโดยอาศัย SDK และ API.....14
2.5 สถาปัตยกรรมของกริดในมุมของการพัฒนา.....15
2.6 โมเดลการให้บริการของเวบเซอร์วิส.....17
2.7 สถาปัตยกรรมแบบลำดับชั้นของเวบเซอร์วิส.....17
2.8 โครงสร้างสถาปัตยกรรมของ OGSA.....20
2.9 ระบบการทำงานของเอเจนต์แบบซิงเกิล.....23
2.10 ระบบการทำงานของมัลติเอเจนต์.....23
3.1 โครงสร้างข้อกำหนดในการควบคุมคุณภาพการให้บริการ.....29
3.2 โครงสร้างสถาปัตยกรรมพื้นฐานของการทำ QoS.....30
3.3 รูปแบบของการควบคุมคุณภาพการให้บริการแบบจุดต่อจุด.....32
3.4 โครงสร้างของการควบคุมการให้บรbdkiที่เน็ตเวิร์คเลเยอร์.....33
3.5 โครงสร้างของงานประยุกต์ที่มีอยู่ในเครือข่าย.....35
3.6 รูปแบบการทำงานของ Custom queue.....36
3.7 รูปแบบการทำงานของ Weighted Fair Queue.....37
3.8 รูปแบบการทำงานของ General Traffic Shaping.....38
3.9 รูปแบบการทำงานของ Link Fragmentation and Interleaving.....39
3.10 รูปแบบการทำงานของ Real time Protocol Header Compression.....39
3.11 รูปแบบการทำงานของ SNA ToS.....40
3.12 โครงสร้างสถาปัตยกรรมของ G-QoSM.....42
3.13 โครงสร้างสถาปัตยกรรมของ A4 เอเจนต์.....42
3.14 โครงสร้างสถาปัตยกรรมของ PACE เอเจนต์.....44
4.1 โครงสร้างแบบทรีของระบบมัลติเอเจนต์.....46
4.2 โครงสร้างแบบฮีพของระบบมัลติเอเจนต์.....46
4.3 การสร้างไพออริตี้คิวจากโครงสร้างของฮีพ.....48
4.4 แนวคิดการทำงานแบบขนาน.....49

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.5 แนวคิดการทำงานการฝากข้อมูล....50

4.6 โฟลว์ชาร์ทการทำงานของมัลติโปรเซสส์อัลกอริทึม....51

4.7 กราฟเปรียบเทียบผลการทดสอบระหว่างคิวอัลกอริทึมและมัลติโปรเซสส์อัลกอริทึม....52

4.8 องค์ประกอบและฟังก์ชั่นการให้บริการของเอเจนต์....53

4.9 การใช้ระบบมัลติเอเจนต์สำหรับการบริการระบบกริด....56

4.10 รูปแบบการจัดโครงข่ายของเอเจนต์ 57

4.11 โครงสร้างของตัวควบคุมเอเจนต์ส่วนกลาง (ASM)....59

4.12 การจัดโครงสร้างของเอเจนต์โดยใช้การเชื่อมต่อความจุแบบต่ำลงตามแบนด์วิดท์....60

4.13 โครงสร้างของเอเจนต์เมื่อมีความจุเท่ากัน....62

4.14 ความแออัดของข้อมูลในกรณีที่ความจุเท่ากัน....62

4.15 โครงสร้างของระบบเมื่อใช้ดัมมี่เอเจนต์....63

4.16 การทำงานของระบบ....64

4.17 การทำงานในลักษณะมัลติโปรเซสส์....66

4.18 การจัดการแบนด์วิดท์ในกรณีที่เป็นงานแบบอิลาสติก....66

4.19 การจัดการแบนด์วิดท์ในกรณีที่เป็นงานแบบเรียลไทม์....67

4.20 แสดงเส้นทางการเดินของข้อมูลจากกริด 3 ไปกริด 6....67

4.21 หลักการฝากข้อมูลในกรณีที่ไม่สามารถส่งถึงโหนดปลายทาง....68

4.22 โฟลว์ชาร์ทแสดงลำดับการทำงานของมัลติโปรเซสส์อัลกอริทึม....70

5.1 โครงสร้างของระบบเอเจนต์ที่ใช้ในการทดลอง....73

5.2 ตัวอย่างในการกำหนดแบนด์วิดท์ตามหลักการของการของการเชื่อมต่อความจุแบบต่ำลง สำหรับข้อมูลชุดที่ 1 ที่ใช้ในการทดลอง....75

5.3 ตัวอย่างในการกำหนดแบนด์วิดท์ตามหลักการของการของการเชื่อมต่อความจุแบบต่ำลง สำหรับข้อมูลชุดที่ 2 ที่ใช้ในการทดลอง....75

5.4 ตัวอย่างการส่งข้อมูลสองชุดระหว่าง A2 ไป A3 และ A4 ไป A3....76

5.5 แสดงไดอะแกรมลำดับของการส่งข้อมูลตามหลักการบริการแบบลำดับความสำคัญ....77

5.6 แสดงไดอะแกรมลำดับของการส่งข้อมูลตามหลักของมัลติ-โปรเซสส์ไพรออริตี้....77

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.7 โฟล์วชาร์ทแสดงวิธีการที่ใช้ในการทดลองวัดประสิทธิภาพสำหรับมัลติ-โปรเซสส์ไพรออริตี้อัลกอริทึม.....79

5.8 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Kbytes กับแบนด์วิดท์ชุดที่ 1.....81

5.9 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Kbytes กับแบนด์วิดท์มูลชุดที่ 2.....81

5.10 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 10 Kbytes กับแบนด์วิดท์ชุดที่ 1.....82

5.11 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 10 Kbytes กับแบนด์วิดท์ชุดที่ 2.....82

5.12 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 50 Kbytes กับแบนด์วิดท์ชุดที่ 1.....83

5.13 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 50 Kbytes กับแบนด์วิดท์ชุดที่ 2.....83

5.14 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 100 Kbytes กับแบนด์วิดท์ชุดที่ 1.....84

5.15 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 100 Kbytes กับแบนด์วิดท์ชุดที่ 2.....84

5.16 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 500 Kbytes กับแบนด์วิดท์ชุดที่ 1.....85

5.17 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 500 Kbytes กับแบนด์วิดท์ชุดที่ 2.....85

5.18 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Mbytes กับแบนด์วิดท์ชุดที่ 1.....86

5.19 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Mbytes กับแบนด์วิดท์ชุดที่ 2.....86

5.20 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 100 Kbytes.....88

5.21 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 500 Kbytes.....88

5.22 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 1 Mbytes.....89

5.23 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 10 Mbytes.....89

5.24 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 100 Kbytes.....90

5.25 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 500 Kbytes.....90

5.26 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 1 Mbytes.....91

5.27 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 10 Mbytes.....91

5.28 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 100 Kbytes..92

5.29 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 500 Kbytes..92

5.30 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 1 Mbytes.....93

5.31 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 10 Mbytes...93

5.32 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 100 Kbytes.....94

5.33 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 500 Kbytes.....94

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.34 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 1 Mbytes.....95

5.35 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 10 Mbytes.....95

5.36 ผลการทดลองเปรียบเทียบระหว่าง Internet Voice PCM กับงานแบบอิลาสติก.....96

5.37 ผลการทดลองเปรียบเทียบระหว่าง Audio MP3 กับงานแบบอิลาสติก.....97

5.38 ผลการทดลองเปรียบเทียบระหว่าง Real time Internet CD Player กับงาน
แบบอิลาสติก.....97

5.39 ผลการทดลองเปรียบเทียบระหว่าง Video Conference กับงานแบบอิลาสติก.....98

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

การประยุกต์ใช้งานการประมวลผลแบบกริดในระยะแรกออกแบบมาสำหรับการประมวลผลทางด้านวิทยาศาสตร์และวิศวกรรม[5] เป็นส่วนใหญ่ ปัจจุบันได้นำมาประยุกต์ใช้กับระบบอินเตอร์เน็ตโดยใช้ร่วมกับเทคโนโลยีเว็บเซอร์วิส ซึ่งเว็บเซอร์วิสใช้มาตรฐานของ XML [7] เป็นตัวแลกเปลี่ยนข้อมูลต่างสถานที่โดยไม่ยึดติดกับแพลต์ฟอร์ม ภายใต้สถาปัตยกรรมบริการของกริดแบบเปิด [9,10] เป็นโครงสร้างพื้นฐาน การประยุกต์ใช้การประมวลผลแบบกริดบนเว็บเซอร์วิสนั้นตัวเว็บเซอร์วิสเองก็มีปัญหาอยู่เนื่องจากจะมีข้อมูลในส่วนที่เกินมามาก จึงไม่รับประกันเรื่องความเร็วและประสิทธิภาพในการใช้งาน

เมื่อมีการนำระบบของกริดและเว็บเซอร์วิสมาทำงานร่วมกัน ภายใต้สถาปัตยกรรมบริการของกริดแบบเปิด เนื่องจากการประมวลผลแบบกริดเป็นระบบที่ต้องประมวลผลด้วยความเร็วสูงและมีการส่งผ่านทรัพยากรในเครือข่ายเป็นจำนวนมาก แต่ระบบของเว็บเซอร์วิสเป็นระบบที่มีข้อจำกัดในเรื่องของความเร็วในการให้บริการ ทำให้เกิดปัญหาขึ้นเมื่อนำระบบที่ต้องการความเร็วและประสิทธิภาพที่สูงไปประมวลผลกับระบบที่มีความล่าช้าในการให้บริการ เพื่อเป็นการแก้ปัญหาที่กล่าวมา งานวิจัยนี้ได้นำเอาคุณสมบัติที่ดีของเอเจนต์มาช่วยในการเพิ่มประสิทธิภาพการทำงานของระบบ โดยจะให้ความสำคัญในเรื่องการควบคุมคุณภาพการให้บริการในระดับงานประยุกต์ (Application) เมื่อระบบค้นหาที่อยู่ของข้อมูลพบแล้วและพร้อมที่จะส่งข้อมูล

1.2 วัตถุประสงค์ของการศึกษา

จุดประสงค์หลักที่สำคัญในวิทยานิพนธ์นี้คือ

1. ศึกษารูปแบบการทำงานของการประมวลผลแบบกริด โครงสร้างสถาปัตยกรรมการจัดการ การประยุกต์ใช้งาน และเทคโนโลยีของกริดที่มีใช้ในปัจจุบัน
2. ศึกษาถึงวิธีการของการควบคุมคุณภาพการให้บริการในเครือข่าย ด้านรูปแบบ เทคนิค และสถาปัตยกรรมรวมถึงการประยุกต์ใช้กับการประมวลผลแบบกริด ในปัจจุบัน

3. นำเสนอหลักการควบคุมคุณภาพการให้บริการของการประมวลผลแบบกริด โดยปรับปรุงจากโครงสร้างเอเจนต์ของ [1] และตรวจสอบแบนด์วิดท์ของเส้นทางการส่งข้อมูลเปรียบเทียบกับความเร็วต่ำสุดที่ต้องการของงานประยุกต์
4. นำเสนอแนวความคิดการควบคุณภาพการให้บริการแบบใหม่ โดยการคำนวณในระดับงานประยุกต์เป็นหลัก ไม่ลงไปยุ่งเกี่ยวกับโปรโตคอลระดับล่าง สามารถจัดทำระบบได้ง่ายและไม่ซับซ้อน
5. ออกแบบอัลกอริทึมที่ใช้ในการควบคุมคุณภาพการให้บริการคือ มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมสำหรับการใช้งานบนโครงสร้างของฮีพ[2] และการเชื่อมต่อความจุแบบต่ำลง [3] โดยอัลกอริทึมนี้จะมองงานในภาพรวมหรือระบบที่มีการส่งข้อมูลจำนวนมากเป็นหลัก

1.3 สมมติฐานของการศึกษา

จากจุดประสงค์หลักข้างต้นที่กล่าวมา จะเห็นว่าการจัดโครงสร้างเอเจนต์ที่ใช้ในการควบคุมทรัพยากรของกริดนั้นจะจัดแบบลำดับชั้นแบบฮีพ เพื่อกำหนดลำดับความสำคัญสำหรับควบคุมคุณภาพการให้บริการตามระดับของเอเจนต์ ปกติเอเจนต์ระดับที่สูงนั้นจะมีความจุในการส่งมากจะจัดให้มีลำดับความสำคัญสูงกว่าเอเจนต์ที่มีระดับที่ต่ำ ในการส่งข้อมูลเอเจนต์ระดับต่ำต้องรอจนกว่าเอเจนต์ที่มีระดับสูงส่งข้อมูลเสร็จก่อนค่อยได้ใช้สิทธิ์ส่งข้อมูล แต่หลักการของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึมนั้นจะไม่เป็นแบบนั้น ระบบหรือศูนย์กลางจัดการทรัพยากรของระบบ จะรู้ตลอดเวลาว่าเอเจนต์แต่ละตัวนั้นจะส่งข้อมูลอะไรหรือมีความเหลือเท่าไร ถ้ามีทรัพยากรเพียงพอระบบจะส่งข้อมูลโดยไม่จำเป็นต้องรอให้เอเจนต์ระดับสูงส่งข้อมูลเสร็จ ตามหลักการแล้ววิธีการใช้มัลติโปรเซสส์ไพออริตี้อัลกอริทึมนี้จะให้ประสิทธิภาพที่ดีกว่าการให้บริการแบบลำดับความสำคัญแบบปกติไม่ว่าจะเป็นการส่งข้อมูลแบบอิลาสติกหรือเรียวไทม์

สิ่งที่คาดว่าจะได้จากการทดลองนี้คือ ประสิทธิภาพที่ดีในการส่งข้อมูลในการประมวลผลแบบกริด เมื่อมีการใช้มัลติโปรเซสส์ไพรออริตี้สำหรับการควบคุมคุณภาพการให้บริการ โดยเปรียบเทียบกับหลักการแบบเดิมคือทำงานตามโครงสร้างของฮีพ และการบริการแบบลำดับความสำคัญปกติ ซึ่งประสิทธิภาพในที่นี้คือการมีความจุเพียงพอในการส่งข้อมูล การกระจายทำงานพร้อม ๆ กันในโครงสร้างของฮีพ และความง่ายในการจัดการนั้นคือการจัดการจะคำนวณงานประยุกต์ง่ายเทียบกับแบนด์วิดท์ในขณะนั้น ๆ เหมือนกับเวลาเราดาวน์โหลดข้อมูลจากเว็บไซต์ในระบบอินเตอร์เน็ต

1.4 แนวความคิดที่ใช้ในการวิจัย

วิทยานิพนธ์นี้มีแนวคิดที่ใช้ในการวิจัยที่สำคัญอยู่สามอย่างคือ

1. การจัดลำดับความสำคัญในการให้บริการตามขนาดของความจุของกริด ในงานวิจัยนี้ได้ประยุกต์ใช้ระบบเอเจนต์เข้ามาควบคุม โดยที่เอเจนต์จะจัดโครงสร้างแบบฮีพตามหลักการของการเชื่อมต่อความจุแบบต่ำลง โดยพิจารณาจากความจุของกริดในโครงสร้าง

2. ทำการให้บริการแบบหลาย ๆ ลำดับความสำคัญ นั้นคือในการส่งข้อมูลนั้นไม่จำเป็นต้องรอให้ส่งงานใดงานหนึ่งเสร็จก่อนแต่สามารถส่งไปพร้อม ๆ กันได้ถ้าการทำงานนั้นไม่มีผลกระทบซึ่งกันและกัน

3. การจัดสรรแบนด์วิดท์ที่เพียงพอ คือการกำหนดแบนด์วิดท์ให้กับงานประยุกต์เท่าที่งานประยุกต์นั้นใช้เท่านั้นส่วนที่เหลือจะถูกจัดสรรให้กับงานอื่น ซึ่งแนวคิดนี้ได้ประยุกต์เอาโครงสร้างแบบการเชื่อมต่อความจุแบบต่ำลงมาเป็นการกำหนดเส้นทางการส่งข้อมูล

จากแนวคิดทั้งหมดที่กล่าวมาวิทยานิพนธ์นี้ได้ประยุกต์ใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมเป็นอัลกอริทึมหลักในการใช้งาน มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมนั้นจะสามารถส่งข้อมูลแบบขนานได้ คือเมื่อส่งข้อมูลที่มีลำดับสูงไปแล้ว สามารถส่งข้อมูลที่มีลำดับความสำคัญรองลงไปได้โดยไม่ต้องให้การส่งข้อมูลที่มีลำดับสูงสำเร็จ แต่มีข้อแม้ว่าจะต้องไม่มีผลกระทบกับงานของกริดที่มีลำดับสูงกำลังจะส่งอยู่ นั้นคือมีแบนด์วิดท์เพียงพอและนอกจากนี้การส่งข้อมูลครั้งแรกไม่จำเป็นต้องส่งให้ถึงกริดที่ร้องขอมาเสมอ สามารถส่งข้อมูลไปฝากไว้กับกริดที่อยู่ในเส้นทางการส่ง และรอส่งในรอบต่อไปของการทำงาน

1.5 ขอบเขตของการวิจัย

ขอบเขตการวิจัยในวิทยานิพนธ์เล่มนี้กำหนดไว้ดังนี้

1. การเปรียบเทียบประสิทธิภาพของการควบคุมคุณภาพการให้บริการจะเปรียบเทียบบนโครงสร้างการทำงานแบบฮีพเท่านั้น

2. ให้ความสำคัญในส่วนของการส่งข้อมูลในโครงสร้างเท่านั้น โดยจะไม่ให้ความสำคัญในเทคโนโลยีการค้นหาข้อมูล การจัดเก็บข้อมูลในระบบ เช่น เทคนิคการค้นหาแบบต่าง ๆ หรือระบบฐานข้อมูลในการจัดเก็บแบบต่าง ๆ ซึ่งในการทดลองนี้จะกำหนดพารามิเตอร์เหล่านี้เป็นค่าคงที่

3. ให้ความสำคัญกับประสิทธิภาพของการทำงานทั้งระบบหรือแบบส่วนรวมเป็นหลัก ดังนั้น พารามิเตอร์ที่ใช้วัดจะใช้เวลาทั้งหมดที่ส่งข้อมูลแต่ละครั้ง

4. ทำงานภายใต้หน่วยประมวลผลหน่วยเดียวเท่านั้น คือในการค้นหาข้อมูลแต่ละชุดในโครงสร้างของฮีพจะใช้โปรเซสส์เซอร์เพียงตัวเดียว

1.6 ส่วนประกอบของวิทยานิพนธ์

วิทยานิพนธ์เล่มนี้ประกอบด้วย 6 บท โดยแต่ละบทมีรายละเอียดดังนี้คือ

บทที่ 1 บทนำจะอธิบายถึง ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการศึกษา สมมติฐานของการศึกษา แนวความคิดที่ใช้ในการวิจัย และขอบเขตของการวิจัย

บทที่ 2 การประมวลผลแบบกริดและเทคโนโลยีที่เกี่ยวข้องกับงานวิจัยอธิบายถึงสถาปัตยกรรมของการประมวลผลแบบกริด การให้บริการเว็บเซอร์วิส การให้บริการของการประมวลผลแบบกริดบนเว็บเซอร์วิส และพื้นฐานการทำงานของเอเจนต์

บทที่ 3 การควบคุมคุณภาพการให้บริการ จะอธิบายถึง หลักการ กลไกล และสถาปัตยกรรมของการควบคุมคุณภาพการให้บริการ การควบคุมคุณภาพการให้บริการระดับเครือข่าย การควบคุมคุณภาพการให้บริการระดับแอพพลิเคชั่น และการควบคุมคุณภาพการให้บริการการกริดที่มีอยู่

บทที่ 4 หลักการควบคุมคุณภาพการให้บริการโดยใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึม เป็นงานวิจัยของวิทยานิพนธ์เล่มนี้ จะกล่าวถึงแนวคิดพื้นฐาน และหลักการในการควบคุมคุณภาพการให้บริการบนกริดโดยใช้มัลติเอเจนต์ และการออกแบบอัลกอริทึมที่ใช้

บทที่ 5 วิธีการทดลองและผลการทดลอง จะอธิบายถึงตัวแบบในการทดลอง การกำหนดค่าแบนด์วิดท์ การวัดประสิทธิภาพในการทดลอง หลักในการเขียนโปรแกรมจำลองเปรียบเทียบการให้การบริการแบบลำดับความสำคัญ ตามอัลกอริทึมไพรออริตี้ของฮีพและมัลติโปรเซสส์ไพรออริตี้สำหรับควบคุมคุณภาพการให้บริการของงานประยุกต์ แบบอิลาสติต แบบเรียลไทม์ และทั้งแบบอิลาสติกและเรียลไทม์อยู่รวมกัน

บทที่ 6 สรุปผลการวิจัยและปัญหาที่เกิดขึ้น เป็นการสรุปเนื้อหาทั้งหมดของการควบคุมคุณภาพการให้บริการโดยใช้มัลติเอเจนต์ สรุปผลการทดลองที่ได้ และบอกถึงปัญหาและข้อเสนอแนะ

บทที่ 2
การประมวลผลแบบกริดและเทคโนโลยีที่เกี่ยวข้อง

2.1 แนวคิดการประมวลผลแบบกริด

แนวคิดการประมวลผลแบบกริดมีแนวคิดมาจากการใช้งานของระบบไฟฟ้ากำลัง กล่าวคือผู้ใช้ไฟฟ้าไม่จำเป็นที่ต้องรู้แหล่งทรัพยากรที่จะใช้งาน สามารถต่ออุปกรณ์ไฟฟ้าเข้ากับระบบไฟฟ้าตามบ้านและสามารถใช้งานได้ เมื่อนำเอาแนวคิดดังกล่าวมาใช้งานทางคอมพิวเตอร์จะต้องมีการจัดการต่างกันเนื่องจากทรัพยากรในระบบไฟฟ้าและระบบคอมพิวเตอร์นั้นต่างกัน และทรัพยากรระบบคอมพิวเตอร์จะมีหลากหลาย

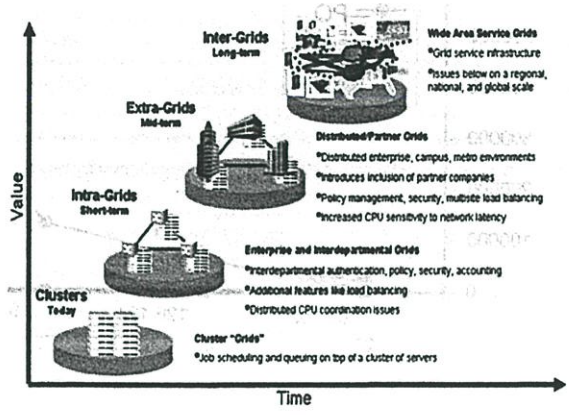
แนวคิดของการประมวลผลกริดเป็นแนวคิดของการประมวลผลแบบเสมือน หมายความว่าการสร้างระบบประมวลผลขนาดใหญ่ที่มีทรัพยากรแบบกระจายอยู่ทั่วไปในเครือข่าย ต่างสถานที่ ซึ่งทรัพยากรถ้าเป็นระบบคอมพิวเตอร์จะหมายถึง หน่วยประมวลผล หน่วยความจำ ระบบปฏิบัติการ ซอฟท์แวร์ งานบริการ หรือระบบเครือข่ายเป็นต้น การใช้งานนั้นผู้ใช้งานไม่จำเป็นต้องทราบแหล่งทรัพยากรในการใช้งาน การประมวลผลกริดจะมีส่วนในการจัดการทรัพยากร นั้นคือองค์กรเสมือน (Virtual Organization: VO) [6] เพื่อคอยจัดการทรัพยากรต่าง ๆ ของกริด

จุดประสงค์ของการใช้องค์กรเสมือนนั้นก็เพื่อใช้เป็นศูนย์กลางในการแลกเปลี่ยน ทรัพยากรที่มีอยู่ของสมาชิกแต่ละสถานที่ ซึ่งสมาชิกนั้นสามารถอยู่ในองค์เสมือนได้มากกว่าหนึ่งองค์กร และในองค์การเสมือนแต่ละองค์กรจะให้บริการทรัพยากรที่ต่างกัน ในขณะเดียวกันทรัพยากรในองค์กรสามารถเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นวิธีการใช้งานระบบกริดจึงมีความยุ่งยากมากกว่าระบบคอมพิวเตอร์ทั่ว ๆ ไป

2.2 วิวัฒนาการของระบบการประมวลผลแบบกริด

ถ้ามองการประมวลผลแบบกริดตามการประยุกต์ใช้งานแล้วจะแบ่งออกเป็น 2 แบบคือ Computing grid และ Data grid แต่ถ้ามองตามลำดับตัวเลข (Topology) ประกอบด้วย Cluster grid, Intra grid, Extra grid และ Inter grid สำหรับ Computing grid เป็นกริดที่ให้บริการทรัพยากรเกี่ยวกับการประมวลเช่น เป็นการรวมทรัพยากรของตัวประมวลผลของซูเปอร์คอมพิวเตอร์แบบเสมือนเป็นต้น Data grid เป็นการให้บริการหรือจัดการฐานข้อมูลหรือระบบแอพพลิชันแบบกระจายขนาดใหญ่ในต่างสถานที่ ซึ่งกริดแบบนี้จะมีการจัดการบริหารได้หลาย ๆ โด

เมนจะต่างจาก Computing grid กริดที่จัดการที่โดเมนเดียวและสามารถเข้าถึงได้โดยใช้นโยบาย (policy) ปัจจุบันมีการพัฒนา Data Grid ใช้บนอินเตอร์เน็ต

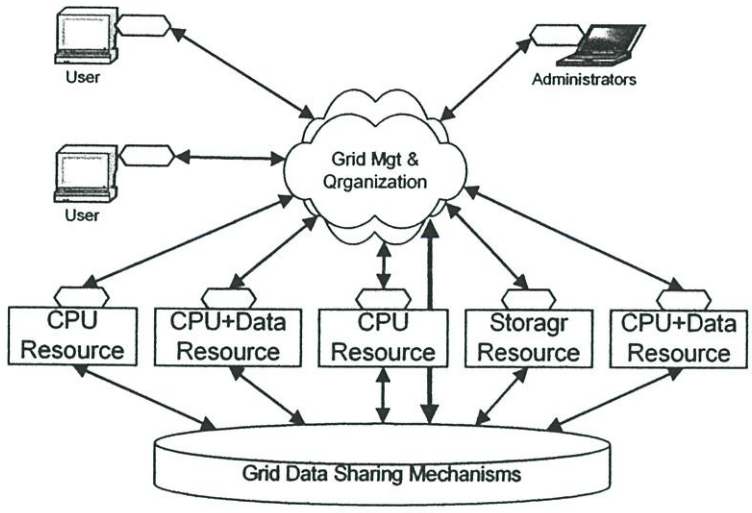


รูปที่ 2.1 แสดงลำดับตัวเลขของการประมวลผลแบบกริด

ถ้ามองทางด้านลำดับตัวเลขการประมวลผลแบบกริด มีทั้งหมด 5 ช่วงคือ Cluster grid ซึ่งเป็นการสร้างหน่วยประมวลผลขึ้นมาในองค์เพื่อรวมกลุ่มการทำงานอย่างใดอย่างหนึ่ง โดยที่ยังไม่มีการเชื่อมต่อไปสู่ระบบอื่นหรือถ้าเปรียบกับระบบการผลิตก็คือสร้างในระดับแผนก ในส่วนของ Intra-grid นั้นจะเริ่มมีการเชื่อมต่อกันระหว่าง Cluster grid อาจจะต่อภายในองค์กรเดียวกันหรือระหว่างแต่ล่ะส่วนงาน Extra-Grid เป็นกรณีของ Cluster หรือ Intra-grid ที่มีการเชื่อมต่อต่างพื้นที่ (Geography) บนระบบทำงานแบบกระจาย ข้อสำคัญของระบบการประมวลผลแบบกริดนี้คือเป็นการกระจายในต่างพื้นที่แต่จะเป็นส่วนของงานเดียวกัน Inter-grid นั้นเป็นการการบริการและการประมวลผลและบนระบบโครงสร้างพื้นฐานขนาดใหญ่ ซึ่งปัจจุบันยังไม่ประยุกต์ใช้ระบบธุรกิจมากนักแต่ส่วนใหญ่จะใช้กับงานวิจัยหรืองานทางด้านวิทยาศาสตร์

2.3 องค์ประกอบของการประมวลผลแบบกริด

จากแนวคิดการทำงานของการประมวลผลแบบกริดจะเห็นว่า ลักษณะการทำงานของระบบนั้นจะให้ความสำคัญกับทรัพยากรเป็นหลักโดยทรัพยากรนั้นอาจจะเป็น ระบบการสื่อสาร ข่าวสาร ข้อมูล ฐานข้อมูล ส่วนบริการ งานประยุกต์รวมถึงฮาร์ดแวร์และซอฟท์แวร์ โดยทรัพยากรเหล่านั้นจะถูกจัดการอยู่ภายใต้องค์การเสมือนซึ่งเป็นส่วนในการควบคุมและจัดการ องค์ประกอบของการประมวลผลแบบกริดแสดงดังรูปที่ 2.2



รูปที่ 2.2 องค์ประกอบพื้นฐานของระบบการประมวลผลแบบกริด

จากรูปที่ 2.2 แสดงส่วนประกอบของระบบกริดแบบพื้นฐานซึ่งประกอบด้วยส่วนของผู้ใช้ ผู้บริหารและองค์กรสำหรับบริหารจัดการระบบทรัพยากรของกริด ส่วนการจัดการของระบบกริดจะมีหน้าที่ในการจัดหาทรัพยากรของกริดรวมถึงการจัดการบริหารด้านอื่น ให้กับผู้ใช้งาน โดยใช้ส่วนบริการข้อมูลข่าวสารกริด [16,17] เป็นตัวจัดเก็บรายละเอียดข่าวสารข้อมูลของผู้ใช้และทรัพยากรในระบบกริด ผู้ใช้งานสามารถเป็นสมาชิกได้หลาย ๆ ผู้บริหารจัดการ ซึ่งผู้บริหารจัดการในที่นี้คือส่วนขององค์กรเสมือนนั้นเอง

ส่วนบริการข้อมูลข่าวสารกริดเป็นส่วนสำคัญในการจัดเก็บรายละเอียดข้อมูลข่าวสารทรัพยากรและบริการของระบบกริด ข้อมูลต่าง ๆ ของกริดจะให้บริการโดยส่วนนี้และข้อมูลข่าวสารจะปรับปรุงให้เป็นปัจจุบันแบบอัตโนมัติ โดยที่ลักษณะการให้บริการของส่วนการบริการข้อมูลข่าวสารจะอ้างอิงตามโมเดลสำหรับการบริการแบบเปิด ซึ่งประกอบด้วยการประกาศ(Publish) การลงทะเบียน (Registry) และการค้นหา (Discovery)

การบริหารจัดหารการทรัพยากรของระบบการประมวลผลแบบกริดจะถูกจัดการจัดการโดยองค์สำหรับจัดการบริหาร การบริหารระบบกริดมีการทำงานหลายรูปแบบดังต่อไปนี้คือ

- **บริการกริด** (Grid Service) ลักษณะการบริการนั้นจะเป็นลักษณะของส่วนประกอบซอฟท์แวร์ซึ่งจะเป็นการอธิบายเชิงความหมายโดยใช้ภาษาโปรแกรมเชิงความหมาย(Description Language) ซึ่งเป็นภาษาที่มีโครงสร้างไม่ซับซ้อนและง่ายต่อการเข้าใจ และมีการลงทะเบียน การประกาศใช้และการค้นหา ผู้ให้บริการโดยการทำลำดับเตรียมการ (Scheduling) ค้นหากลุ่มของ

ทรัพยากรและเลือกทรัพยากรจากกลุ่มที่ค้นหานี้มาทำงาน ซึ่งการเลือกทรัพยากรนั้นก็เป็นเรื่องสำคัญเพราะบางทีทรัพยากรนั้นอาจถูกใช้งานบางอย่างอยู่

● **การบริหารความปลอดภัยของกริด** (Security Service) หลักการควบคุมความปลอดภัยในระบบกริดนั้นเป็นเรื่องสำคัญเนื่องจากระบบนั้นเป็นระบบที่ซับซ้อนและมีทรัพยากรและการบริการอยู่มาก ระบบความปลอดภัยในกริดที่มีอยู่ประกอบด้วย

1. Authentication: เป็นการจัดความปลอดภัยในส่วนของผู้ใช้โดยการกำหนดสิทธิการใช้ทรัพยากรและการบริการให้กับผู้ใช้ระบบ
2. Authorization: เป็นการกำหนดสิทธิหลังจากที่ผ่านระบบรับรองการใช้งานมาแล้วซึ่งจะมีการสร้างนโยบายการใช้งานในระบบรวมถึงการกำหนดขอบเขต และขีดความสามารถในการใช้งาน
3. Encryption: เป็นระบบการเข้ารหัสเพื่อป้องการส่งข้อมูลข่าวสารระหว่างผู้ใช้
4. Nonrepudiation: เป็นระบบป้องกันข้อมูลข่าวสารเพื่อเป็นการยืนยันว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงในระหว่างการส่ง
5. Single Sign-on: เป็นระบบความปลอดภัยสำหรับรองรับ Authentication ของระบบทรัพยากรกริดขนาดใหญ่เมื่อผู้ใช้หรือทรัพยากรได้รับสิทธิในการกระทำซึ่งคล้ายการให้บริการแบบพรอกซี (Proxy)
6. Delegation: การมอบหมายสิทธิจากกริดเอนติตี้ (Entity) การรับมอบหมายจะกระทำอย่างระมัดระวังโดยการสร้างระบบลูกโซ่ขึ้นมา
7. Community authorization: เป็นกลไกสำหรับองค์เสมือนกำหนดนโยบายให้กับกลุ่มผู้ใช้งานซึ่งกลุ่มผู้ใช้นั้นจะสามารถเขาถึงทรัพยากรได้ทั้งกลุ่ม
8. Secure execution: เป็นระบบความปลอดภัยสำหรับกลุ่มผู้ใช้งาน (Group User) เป็นการกำหนดสภาวะแวดล้อมให้กับกลุมผู้ใช้เพื่อให้รองรับกับการประมวลผลงานที่มีขนาดใหญ่ ระบบนี้มีความสำคัญสำหรับกริดที่ใช้สภาวะแวดล้อมในการประมวลผลขนาดใหญ่เช่นศูนย์คอมพิวเตอร์ หรือ ระบบคลัสเตอร์

● **การบริหารระบบข่าวสารของกริด** คือระบบข่าวสารหรือรายละเอียดการบริการและทรัพยากรในระบบ การให้บริการส่วนนี้กริดจะเรียกว่าส่วนบริการข้อมูลข่าวสารกริดจะเป็นข่าวสารเกี่ยวกับผู้ใช้และข่าวสารของระบบถือเป็นส่วนสำคัญและวิกฤติ ข้อมูลเกี่ยวกับผู้ใช้จะช่วยในการทำงานร่วมกัน ส่วนข่าวสารของระบบจะช่วยในการเลือกทรพัยากรหรือแอพพลิเคชั่นที่เหมาะสมในการประมวลผล นอกจากนี้ระบบข้อมูลข่าวสารยังช่วยในการบำรุงรักษา การกำหนดค่าและการปรับเปลี่ยนระบบกริดพื้นฐาน (Infrastructure) โดยอัตโนมัติ

• **การบริหารข้อมูลของกริด** เนื่องจากขนาดของข้อมูลในระบบ กริดนั้นมีขนาดใหญ่และเป็นเรื่องยากในการจัดการ ระบบกริดนั้นมีข้อจำกัดในเรื่องของหน่วยความจำในแต่ละสถานที่ระบบกริดจะต้องกำหนดจำนวนข้อมูลที่เกิดขึ้นจริง ซึ่งในการคำนวณขนาดนั้นไม่สามารถกระทำได้ในขณะที่ข้อมูลนั้นถูกส่งไปในระบบ ดังนั้นต้องสามารถบันทึกทับข้อมูลเดิมได้ นอกจากนั้นยังต้อมีการลดจำนวนข้อมูลระหว่างการส่งลงได้โดยการทำงานกรอง (Filter)

• **การประมวลผลกริดและการจัดการทรัพยากร** ระบบที่ทำการคำนวณทรัพยากรของกริดนั้นคือส่วนการบริการประมวลผล (Execution Service) การทำงานระบบเดิมเช่น การคำนวณโดยระบบปฏิบัติการหรือการประมวลผลแบบแบทช์ (Batch) นั้นไม่เหมาะสมซึ่งในระบบกริดต้องสามารถจัดการทรัพยากรหลาย ๆ อินสแตน (Instance) คือต้องมีการใช้กลไกในการนำทรัพยากรมาใช้ใหม่อย่างรวดเร็วซึ่งจะต้องกำหนดกลุ่มทรัพยากรที่เหมาะสมบนพื้นฐานของการบริการข่าวสารของกริดจากนั้นใช้อัลกอริทึมตรวจสอบและเลือกทรัพยากรที่สามารถใช้ได้และทำการขอเพื่อใช้งาน

• **การบริหารซอฟท์แวร์ของกริด** การจัดการงานประยุกต์และคอมโมเนนท์และการบริการบนระบบกระจายที่ซับซ้อน เป็นปัญหาใหญ่ซึ่งต้องรับประกันการทำงานร่วมกันของซอฟท์แวร์และไลบารี (Libraries) ที่ต่างตระกูล การจัดการนั้นจะใช้โมเดลการบริการของกริดเป็นตัวรองรับซึ่งทุกแอพพลิเคชั่นและบริการที่นำมาใช้ในระบบกริดจะส่งข้อมูลไปเก็บที่ส่วนบริการข่าวสารของกริด

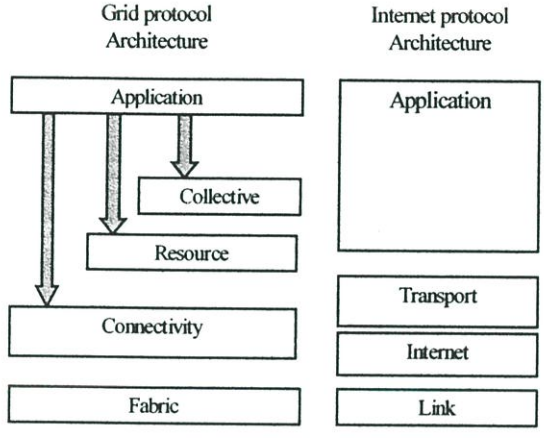
• **การบริหารฮาร์ดแวร์** ฮาร์ดแวร์ในระบบกริดนั้นจะเป็นส่วนที่จัดเตรียมทรัพยากรในระบบซึ่งระบบกริดจะใช้ส่วนบริการข่าวสารสำหรับเก็บรายละเอียดของการปรับปรุงหรือแก้ไขส่วนฮาร์ดแวร์ ซึ่งปกติจะจัดการโดยผู้ให้บริการ

2.4 โครงสร้างสถาปัตยกรรมของระบบประมวลผลแบบกริด

แนวคิดที่สำคัญที่สุดขององค์กรเสมือนก็คือการใช้รีซอร์สเพื่อทำงานร่วมกัน การทำงานพื้นฐานขององค์กรเสมือนจึงต้องระบุถึงการเชื่อมต่อข้อมูลระหว่างสมาชิกได้ สถาปัตยกรรมของกริดก็ถูกสร้างขึ้นมาเพื่อสนับสนุนแนวคิดขององค์กรเสมือน โดยระบุถึงวิธีการพื้นฐานที่สมาชิกขององค์กรเสมือนเข้าใช้ ต่อรอง จัดการ และอนุญาตให้ใช้ข้อมูลที่ตนมีอยู่ สำหรับมาตรฐานที่มีรากฐานมาจากสถาปัตยกรรมแบบเปิดอย่าง การแชร์ข้อมูล ความสามารถในการขยายขนาด ความเข้ากันได้ของข้อมูล ถือเป็นความสามารถของสถาปัตยกรรมของกริดที่เพิ่มเติมเข้ามาในระยะหลัง

ความสามารถในการใช้ทรัพยากรเพื่อทำงานร่วมกันเป็นประเด็นสำคัญที่ต้องพิจารณา สถาปัตยกรรมของกริดจะต้องมีความสามารถในการระบุการเชื่อมต่อระหว่างสมาชิกได้แบบอิสระ

ไม่ขึ้นอยู่กับแฟลตฟอร์ม ภาษา หรือ สภาพแวดล้อมของการพัฒนาของสมาชิก ดังนั้น ความสามารถในการค้นหาข้อมูล แสดงตัวตนของสมาชิก การรับรองสิทธิ และการระบุการแชร์ข้อมูลจะต้องเป็นวิธีการที่ยืดหยุ่นและไม่สิ้นเปลืองทรัพยากรมากนัก ซึ่งจะทำให้กลไกการเชื่อมต่อของระบบสามารถทำได้อย่างรวดเร็ว นอกจากนี้ โปรโตคอลที่สร้างขึ้นจะต้องไม่กระทบกับการทำงานพื้นฐานของข้อกำหนดในการใช้งานระบบปัจจุบันด้วย



รูปที่ 2.3 โครงสร้างสถาปัตยกรรมกริดเปรียบเทียบกับโปรโตคอลอินเตอร์เน็ต

โครงสร้างสถาปัตยกรรมของกริดประกอบด้วย 5 ชั้นแสดงดังรูป 2.3 คือ ชั้นงานประยุกต์ (Application later), ชั้นการรวม (Collective layer), ชั้นทรัพยากร (Resource layer), ชั้นการติดต่อ (Connectivity layer) และ ชั้นฟาบริก(Fabric layer) ในแต่ละชั้นจะมีหน้าที่การทำงานแตกต่างกัน ความสัมพันธ์ของเลเยอร์เหล่านี้ก็คือ การทำงานของเลเยอร์ที่อยู่บนจะถูกสร้างอยู่บนการทำงานของเลเยอร์ระดับล่าง เมื่อเปรียบเทียบกับโปรโตคอลอินเตอร์เน็ต สถาปัตยกรรมของกริดจะมีลักษณะคล้ายคลึงกัน กลไกการทำงานของชั้นฟาบริก ของสถาปัตยกรรมของกริดเทียบได้กับเชื่อม ของโปรโตคอลอินเตอร์เน็ต โดยมีหน้าที่เชื่อมต่อกันในระดับกายภาพ ส่วนชั้นการรวม ทำหน้าที่รับส่งข้อมูล และการอนุญาตการใช้บริการ ครอบคลุมการทำงานของเลเยอร์ Internet และ Transport ของอินเตอร์เน็ตโปรโตคอล ในส่วนชั้นทรัพยากร, ชั้นการติดต่อและชั้นงานประยุกต์ของกริด เทียบได้กับส่วนของชั้นงานประยุกต์ของโปรโตคอลอินเตอร์เน็ต จะเห็นว่าในสถาปัตยกรรมของกริดจะมีชั้นทรัพยากรและชั้นการรวมเพิ่มมาทำหน้าที่เกี่ยวข้องกับการบริหารจัดการและรวบรวมทรัพยากรที่ใช้ร่วมกัน อันเป็นจุดประสงค์หลักขององค์กรเสมือนของกริดนั่นเอง

2.4.1 ชั้นฟาบริก

การทำงานของชั้นฟาบริก จะเกี่ยวข้องกับส่วนประกอบพื้นฐานของสถาปัตยกรรมของ กริด มีหน้าที่ระบุการเชื่อมต่อทรัพยากร และทำโอเปอเรชันการแชร์ข้อมูล ทรัพยากรที่มีอยู่ในระบบจะต้องถูกตรวจสอบโครงสร้าง สถานะทำงาน และคุณสมบัติได้โดยระบบ นอกจากนี้ จะต้องมีส่วนทำงานจัดการทรัพยากร ที่มีหน้าที่ควบคุมคุณภาพการรับส่งข้อมูล สำหรับการจัดการทรัพยากรที่มีลักษณะต่างๆ จะมีวิธีการองค์ประกอบดังนี้

1. Computational Resources เป็นทรัพยากรที่เกี่ยวข้องกับการเริ่มต้นรัน ตรวจสอบและควบคุมการทำงานโปรแกรม โดยจองหน่วยประมวลผลเพื่อการทำงานโปรแกรมของสมาชิก ตรวจสถานะการทำงานของฮาร์ดแวร์ ซอฟท์แวร์
2. Storage Resources เป็นทรัพยากรที่เกี่ยวข้องกับแหล่งเก็บข้อมูล กลไกส่วนนี้มีหน้าที่จัดวาง และ เก็บไฟล์ข้อมูลในระบบกริด กลไกดังกล่าวจึงต้องสามารถควบคุมการทำงานของการจองเนื้อที่หน่วยความจำ
3. Network Resources เป็นทรัพยากรที่เกี่ยวข้องกับการสื่อสารในระบบเครือข่าย กลไกการทำงานต่อทรัพยากรแบบนี้มีหน้าที่จองเนื้อที่การรับส่งข้อมูลได้
4. Code repositories เป็นรูปแบบการเก็บข้อมูลแบบพิเศษ สำหรับการเก็บเวอร์ชันของข้อมูล และโค้ดของออบเจ็ก
5. Catalogs เป็นรูปแบบการเก็บข้อมูลแบบพิเศษ สำหรับการอิมพลีเมนต์การสอบถามและปรับปรุงการทำงานของชุดข้อมูล เช่น ฐานข้อมูลสัมพันธ์

2.4.2 ชั้นการติดต่อ

ชั้นการติดต่อระบุถึงกลไกการสื่อสารและโปรโตคอลการรับรองสิทธิ์ ซึ่งเป็นส่วนสำคัญของระบบเครือข่ายกริด การสื่อสารข้อมูลระบุถึงการแลกเปลี่ยนข้อมูลระหว่างรีซอร์สในชั้นฟาบริก ส่วนโปรโตคอลการเข้าใช้บริการจะจัดหาบริการที่เกี่ยวข้องกับการเข้ารหัสข้อมูลเพื่อยืนยันตัวตนของสมาชิกและทรัพยากร

ความต้องการด้านการสื่อสารข้อมูลจะเกี่ยวข้องกับการรับส่ง การค้นหาเส้นทาง และการอ้างอิงชื่อ กลไกที่ถูกนำมาใช้จะอ้างอิงจากเทคโนโลยีเดิมที่มีอยู่อันได้แก่ โปรโตคอลทีซีพีไอพี ยูดีพี ดีเอ็นเอส เป็นฐานในการพัฒนา ในอนาคตจะมีการขยายโปรโตคอลเหล่านี้ให้เหมาะสมกับลักษณะของระบบกริดต่อไป

ในส่วนการรับรองสิทธิ์จะพิจารณาต่อระบบความปลอดภัยของข้อมูล ในระยะแรกระบบกริดจะใช้งานโปรโตคอลการรับรองสิทธิ์ที่ใช้อยู่ในสถาปัตยกรรมของอินเตอร์เน็ต อย่างไรก็ดี

ระบบกริดมีลักษณะแตกต่างกับระบบอินเตอร์เน็ต ดังนั้น การพัฒนาโปรโตคอลการรับรองสิทธิ์จะให้เหมาะกับองค์กรเสมือน โดยมีลักษณะดังต่อไปนี้

1. Single sign on ผู้ใช้บริการต้องสามารถล็อกอินเข้าใช้บริการต่างๆ ที่ระบุไว้ในชั้นฟาบริกจากระบบเพียงครั้งเดียว
2. Delegation ผู้ใช้บริการต้องสามารถรันโปรแกรมของตนได้ตามสิทธิของตน ดังนั้นโปรแกรมดังกล่าวจึงต้องสามารถใช้ทรัพยากรที่ผู้ใช้บริการมีสิทธิใช้บริการ นอกจากนี้โปรแกรมควรจะทำงานแลกเปลี่ยนข้อมูลร่วมกับโปรแกรมอื่นในระบบได้ด้วย
3. Integration with various local security solutions ผู้ให้บริการและผู้ขอใช้บริการควรจะอิมพลีเมนต์ระบบความปลอดภัย ให้ทำงานร่วมกับระบบเดิมได้โดยไม่ต้องเปลี่ยนแปลงมากนัก
4. User-based trust relationships เพื่อที่จะทำให้ผู้ใช้บริการใช้งานทรัพยากรจากผู้ให้บริการแบบหลากหลาย ระบบความปลอดภัยที่สร้างขึ้นมาจะต้องไม่กระทบกับระบบความปลอดภัยของผู้ให้บริการแต่ละราย

2.4.3 **ชั้นทรัพยากร**

ชั้นทรัพยากรถูกสร้างอยู่บนชั้นการติดต่อ เพื่อระบุกลไกการสื่อสารอย่างปลอดภัย การเริ่มต้นการเชื่อมต่อ การตรวจตรา การควบคุม การเก็บรักษาข้อมูล และการใช้งานร่วมกันของทรัพยากรส่วนบุคคล ชั้นนี้อิมพลีเมนต์กลไกของชั้นฟาบริก ในการเข้าใช้ และควบคุมทรัพยากรในระดับโลคอล โดยไม่ได้เกี่ยวข้องกับการเชื่อมต่อข้อมูลในระดับโกลบอล ประเด็นดังกล่าวจะเกี่ยวข้องคอนชั้นการติดต่อซึ่งจะกล่าวในหัวข้อถัดไป

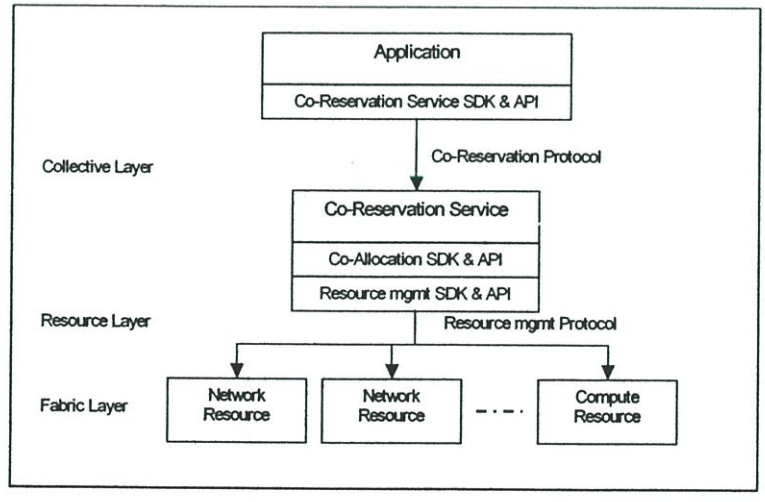
โปรโตคอลที่ใช้ในชั้นทรัพยากรแบ่งออกเป็น 2 รูปแบบใหญ่ดังนี้

1. Information Protocol ถูกนำมาใช้ในการเก็บข้อมูลเกี่ยวกับโครงสร้าง และสถานะของทรัพยากร ยกตัวอย่างเช่น ข้อกำหนดพื้นฐาน ภาระการใช้งานปัจจุบัน และ กฎการใช้ทรัพยากร
2. Management Protocol ถูกนำมาใช้ต่อรองการเข้าใช้ทรัพยากรที่แชร์ร่วมกันอยู่ โปรโตคอลนี้ทำหน้าที่ควบคุมการใช้ข้อมูลให้เป็นไปตามกฎเกณฑ์ที่วางไว้ร่วมกัน นอกจากนี้ โปรโตคอลนี้อาจจะสนับสนุนการตรวจตราสถานะการทำงาน และควบคุมการทำงานที่มีต่อทรัพยากรนั้นๆ

2.4.4 ชั้นการรวม

ชั้นการรวมเกี่ยวข้องกับโปรโตคอลและบริการในระดับโกลบอล ซึ่งแตกต่างจากชั้นทรัพยากร ที่เกี่ยวข้องกับโปรตคอลระดับโลคอลที่จัดการกับทรัพยากรตัวใดตัวหนึ่ง โดยชั้นการรวมจะจัดการกับทรัพยากรภายในระบบให้ทำงานร่วมกันเพื่อจุดประสงค์อย่างใดอย่างหนึ่ง อาทิเช่น เก็บ ค้นหา หรือควบคุมทรัพยากรเหล่านั้น เนื่องจากชั้นนี้สร้างอยู่บนรีชั้นทรัพยากร ทำให้โปรโตคอลที่สร้างขึ้นมาใหม่สามารถอิมพลีเมนต์โปรโตคอลที่มีอยู่แล้วได้ และสามารถทำงานได้ในหลายรูปแบบ ยกตัวอย่างเช่น

- Directory Service เป็นบริการที่ให้ผู้ใช้บริการสามารถค้นหาคุณสมบัติของทรัพยากรที่อยู่ในองค์กรเสมือนได้ ด้วยการใส่แอตทริบิว (Attribute) เช่น ประเภทข้อมูล ขนาด เป็นคำสำคัญในการค้นหา
- Co-allocation, scheduling and Broker Service เป็นบริการช่วยสมาชิกขององค์กรเสมือนจัดการจองทรัพยากรที่เหมาะสมเพื่อทำงานอย่างใดอย่างหนึ่ง
- Monitoring and diagnostics Services เป็นบริการที่ตรวจสอบสถานะของทรัพยากรที่ผิดปกติว่า สูญหาย ถูกโจมตี หรือ ไม่พร้อมใช้งาน
- Data Replication Services สนับสนุนการใช้งานแหล่งข้อมูลภายในระบบกริดให้มีประสิทธิภาพ ตามมาตรวัดที่อ้างอิง เช่น Response Time ความน่าเชื่อถือ และความสิ้นเปลือง

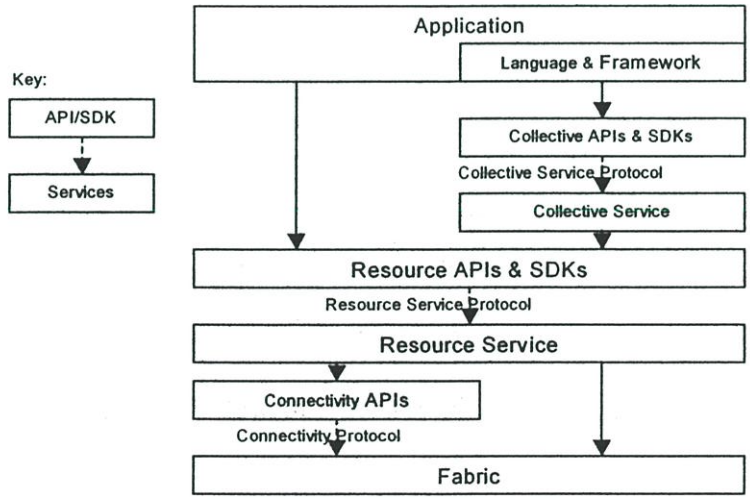


รูปที่ 2.4 ความสัมพันธ์ระหว่างชั้นต่างๆ ในการจัดการทรัพยากรโดยอาศัย SDK และ API

- Software Discovery Service เป็นบริการที่เลือกซอฟท์แวร์ที่มีอยู่ และแพลตฟอร์มที่ใช้ รันงาน โดยพิจารณาลักษณะปัญหาตามความเหมาะสม
- Collaboration services เป็นบริการที่สนับสนุนการแลกเปลี่ยนข้อมูลภายในระบบที่มีผู้ใช้บริการจำนวนมาก

ฟังก์ชันการทำงานของชั้นการรวมสามารถถูกนำมาสร้างเป็นบริการแบบหนึ่งในระบบได้ด้วยโปรโตคอลที่เกี่ยวข้อง หรือ SDK ที่ถูกออกแบบมาเพื่อเชื่อมต่องานประยุกต์ต่างๆ บริการที่สร้างขึ้นจะสร้างอยู่บนโปรโตคอลของรีซั้นทรัพยากร ยกตัวอย่างเช่นในรูปที่ 2.4 แสดงให้เห็นถึง co-allocation API และ SDK ที่ใช้โปรโตคอลการจัดการทรัพยากรในชั้นมาเป็นฐานในการพัฒนาโปรโตคอล co-reservation service ที่มีความสามารถเพิ่มขึ้นจากโปรโตคอลเดิม เช่น การรับรองสิทธิ์ และ การดูแลรักษาข้อมูล นอกจากนี้ ในส่วนชั้นงานประยุกต์ สามารถใช้ co-reservation service ในการร้องขอข้อมูลแบบจุดต่อจุดในระบบเครือข่ายได้

โปรโตคอลที่เป็นส่วนประกอบของชั้นการรวมอาจจะถูกสร้างขึ้นมาตามความต้องการของผู้ใช้รายใดรายหนึ่ง องค์กรเสมือน หรือ โดเมนงานประยุกต์ ก็ได้ ดังนั้นโปรโตคอลที่ใช้จึงมีจุดประสงค์ทั่วไปในการรองรับความต้องการของผู้ใช้บริการส่วนใหญ่ อย่างไรก็ตาม ก็ต้องมีลักษณะเฉพาะกับจุดประสงค์ของผู้ใช้บริการอื่นได้ด้วย



รูปที่ 2.5 สถาปัตยกรรมของกริดในมุมของการพัฒนา

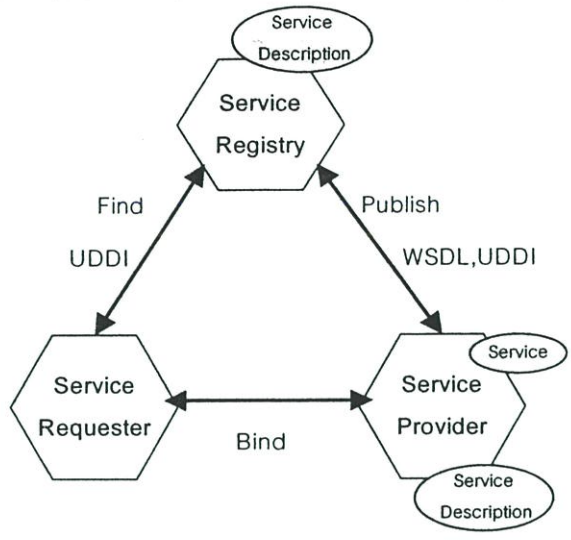
2.4.5 ชั้นงานประยุกต์

ส่วนประกอบสุดท้ายของสถาปัตยกรรมกริดก็คือชั้นงานประยุกต์ ประกอบด้วยงานประยุกต์ของผู้ใช้บริการที่ทำงานอยู่ในองค์กรเสมือน งานประยุกต์อยู่ในระบบกริดจะถูกสร้างขึ้นมาโดยอาศัยบริการ หรือโปรโตคอลการทำงานในชั้นระดับล่าง ในรูปที่ 2.5 แสดงสถาปัตยกรรมกริดในมุมของผู้พัฒนา งานประยุกต์จะถูกสร้างขึ้นมาโดยอิมพลิเมนต์บริการในระดับชั้นที่ต่ำกว่า ในการเข้าใช้และควบคุมทรัพยากรที่มีอยู่ในระบบ บริการที่กล่าวมาได้แก่ การจัดการทรัพยากร การเข้าใช้ข้อมูล การค้นหาข้อมูล และอื่นๆ ในแต่ละชั้นอาจจะระบุการอิมพลีเมนต์ที่มีการแลกเปลี่ยนข้อมูลระหว่างบริการเพื่อทำงานในจุดประสงค์เฉพาะอย่างได้

2.5 การให้บริการเว็บเซอร์วิส

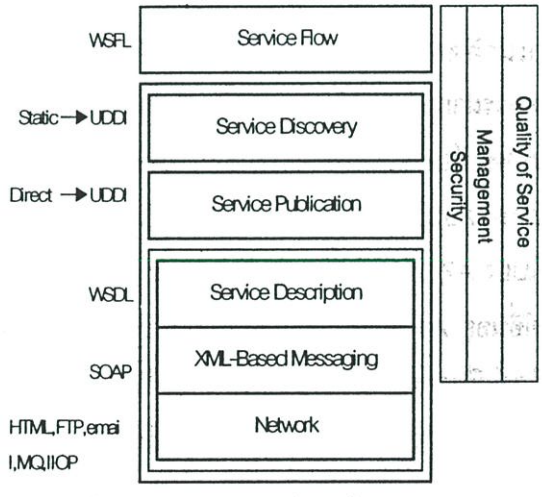
ระบบการแลกเปลี่ยนข้อมูลผ่านระบบเครือข่ายด้วยเทคโนโลยีอินเตอร์เน็ตนั้นประสบความสำเร็จมากในทศวรรษที่ผ่านมา กุญแจสำคัญต่อความสำเร็จนี้อยู่ความง่ายของรูปแบบข้อมูลที่ส่งระหว่างกัน การรับส่งข้อมูลรูปแบบ HTML ลักษณะที่สามารถอธิบายตัวเองได้ทำให้ระบบงานต่างๆ สามารถเข้าใจข้อมูลที่รับส่ง และแปลความหมายของข้อมูลนั้นด้วยเว็บบราวเซอร์เป็นเทคโนโลยีที่ดำเนินรอยตามกุญแจสำคัญดังกล่าว หากแต่ใช้ XML เป็นสื่อกลางรับส่งข้อมูล ด้วยความสามารถที่เหนือกว่าของ XML เว็บเซอร์วิสจึงใช้ XML เป็นตัวอธิบายข้อมูลสำหรับการติดต่อสื่อสารแบบเรียกการทำงานจากเครื่องที่อยู่บนเครือข่ายระยะไกลได้ (Remote Procedure Call: RPC) ผ่านโพรโตคอล SOAP [8] อันเป็น XML รูปแบบหนึ่ง ปัจจุบันมีหน่วยงานมากมายที่ใช้เทคโนยีเว็บเซอร์วิสกับระบบงานของตนในด้านการแลกเปลี่ยนข้อมูล ระบบประมวลแบบกริดก็เป็นเทคโนโลยีที่สามารถสร้างบนรากฐานของเว็บเซอร์วิสได้

เว็บเซอร์วิสเป็นลักษณะของงานบริการ ดังนั้นโมเดลของเว็บเซอร์วิสจะเป็นโมเดลสำหรับการให้บริการแสดงดังรูปที่ 2.6 ซึ่งจะประกอบด้วยส่วนให้บริการอยู่ 3 ส่วนคือ ผู้ให้บริการ, ส่วนบริการลงทะเบียน, ผู้ใช้บริการ โดยที่ผู้ให้บริการจะให้ข้อมูลที่เป็นรายละเอียดการให้บริการของตนและประกาศการให้ทราบที่ส่วนบริการลงทะเบียนเมื่อผู้ใช้บริการต้องการข้อมูลจะขอใช้บริการเมื่อต้องการใช้บริการงานอย่างใดอย่างหนึ่งก็ขอไปยังส่วนบริการลงทะเบียนซึ่งจะแจ้งรายละเอียดของข้อมูลซึ่งอาจจะเป็นที่อยู่หรือข้อมูลอย่างอื่นของผู้ให้บริการแก่ ผู้ใช้บริการเพื่อให้ ผู้ใช้บริการไปค้นหาข้อมูลนั้นตามต้องการ ในขั้นตอนที่ผู้ให้บริการประกาศ การให้บริการเราเรียกขั้นตอนนี้ว่า Publish และขั้นตอนที่ผู้ขอบริการขอใช้บริการเราเรียกว่า Find ส่วนขั้นตอนสุดท้ายกรณีที่ผู้ขอบริการนั้นค้นหาข้อมูลได้และไปขอใช้ข้อมูลเรียกว่า Bind



รูปที่ 2.6 โมเดลการให้บริการของเว็บเซอร์วิส

สถาปัตยกรรมของเว็บเซอร์วิส อธิบายได้โดยโครงสร้างลำดับขั้น (Stag) ดังรูปที่ 2.7 ซึ่งประกอบ 6 ลำดับชั้น



รูปที่ 2.7 สถาปัตยกรรมแบบลำดับชั้นของเว็บเซอร์วิส

การทำงานนั้นจะเป็นลักษณะแบบลำดับขั้นตอน (Stack) ซึ่งประกอบด้วย

- Network เป็นลำดับชั้นสำหรับการติดต่อสื่อสาร โดยเว็บเซอร์วิสนั้นจะติดต่อสื่อสารบนอินเตอร์เน็ต จึงใช้โปรโตคอลของ HTTP ร่วมกับโปรโตคอล อื่นเช่น FTP, SMTP, Message Queuing, RMI, IIOP เป็นต้น
- XML-base Messaging เป็นรูปแบบของภาษา XML โดยสื่อสารผ่านโปรโตคอล SOAP โดยใช้ SOAP Message
- Service Description เป็นชั้นสำหรับอธิบายรายละเอียดของเอกสารโดยใช้ โปรโตคอลของ WSDL โดยจะใช้รูปแบบ ภาษาของ XML
- Service Publication เป็นการประกาศการให้บริการโดยจะประกาศ ผ่านทางโปรโตคอลของ UDDI
- Service Discovery เป็นการค้นหาบริการของเว็บเซอร์วิสโดยใช้ UDDI ซึ่งเมื่ออธิบายการทำงานเบื้องต้นได้ดังนี้คือ เริ่มจากการสร้างเอกสารที่เป็น XML Message ซึ่งการใช้งานจะถูกเรียกผ่านทาง SOAP โปโตคอล หลังจากนั้นนำใบประกาศการใช้งานหรือลงทะเบียนโดยผ่านทาง UDDI โปโตคอล เมื่อผู้ใช้ต้องการรายละเอียดของเอกสารก็ค้นหาผ่าน UDDI ในชั้นของ Service Discovery นั้นเอง

ในการใช้งานระบบกริดในยุกต์แรกจะใช้ในด้านการวิจัยทางวิทยาศาสตร์และการคำนวน เป็นหลักปัจจุบันได้มีการนำระบบกริดมาใช้กับระบบธุรกิจมากขึ้นโดยการใช้กับระบบอินเตอร์เน็ต การนำระบบกริดมาใช้กับอินเตอร์เน็ตนั้น จะประยุกต์ใช้ร่วมกับเทคโนโลยีของเว็บเซอร์ซิส ในลักษณะของกริดมิดเดิลแวร์ (Middleware) คือ OGSA (Open Grid Service Architecture)

2.6 ระบบกริดสำหรับการบริการบนเว็บเซอร์วิส

โดยจุดเด่นของเทคโนโลยีเว็บเซอร์วิส ที่มีความง่ายในการแลกเปลี่ยนข้อมูลนี้เอง สามารถนำมาประยุกต์ใช้กับงานในรูปแบบบางอย่างภายในการประมวลแบบกริดได้ ยกตัวอย่างเช่น [18, 7, 8] เป็นการนำเว็บเซอร์วิสมาให้บริการข้อมูลสนเทศของกริดในระบบเปิดเป็นต้น ข้อมูลที่สามารถอธิบายตัวเองได้ของ XML เป็นคุณสมบัติที่เหมาะสมต่อสภาพแวดล้อมการใช้ข้อมูลที่หลากหลาย อันเป็นประโยชน์ต่อระบบ รวมทั้งโครงสร้างของระบบที่สามารถใช้ชิ้นส่วนของซอฟท์แวร์ร่วมกันก็เป็นประเด็นที่น่าสนใจต่อการพัฒนาระบบของกริด นอกจากนี้ ในส่วนของผู้ใช้ยังสามารถติดต่อกับระบบเพื่อเข้าใช้ข้อมูลสารสนเทศได้สะดวกขึ้นผ่านทางเว็บบราวเซอร์ จากเดิมที่

ต้องมีแอพพลิเคชั่นพิเศษที่ได้มาจากการเป็นสมาชิกของระบบ อย่างไรก็ตาม ประสิทธิภาพของระบบที่สร้างจากเทคโนโลยีเว็บเซอร์วิสนั้นเป็นประเด็นที่น่าสงสัยอยู่ เนื่องจากการรับส่งข้อมูลนั้นจะมีข้อมูลส่วนหัวที่ติดไปกับเนื้อข้อมูลด้วย อาจจะทำประสิทธิภาพของระบบด้อยลงไปได้ อีกประเด็นหนึ่งก็คือ ระบบการส่งข้อมูลบริการของเว็บเซอร์วิสในปัจจุบันนั้นยังมีความล่าช้าอยู่มาก การส่งข้อมูลแต่ละครั้งนั้นต้องอินเตอร์เฟสการติดต่อ WSDL [11] แปลเพื่อขอใช้ UDDI (Universal Description, Discovery & Integration) [11] สำหรับการส่งข้อมูล

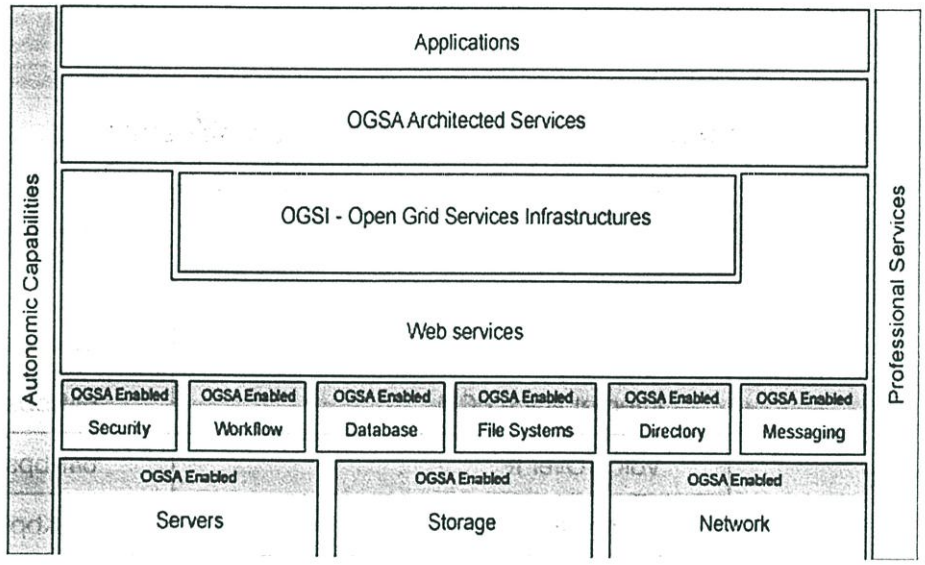
OGSA ร่วมกันพัฒนาโดย GLOBUS และ IBM มีจุดประสงค์เพื่อใช้งานร่วมกันระว่างกริดและเว็บเซอร์วิสในลักษณะการเรียกใช้งานผ่านทาง WSDL ซึ่งการทำงานของ OGSA ยังอาศัยพื้นฐานของการบริการ (Service Type) โดยจะอธิบายกำหนดการให้บริการในลักษณะของพอร์ท ซึ่งเป็นมาตรฐานการเชื่อมต่อแสดงดังตารางที่ 2.1

ตารางที่ 2.1 ชนิดของพอร์ทของการอินเตอร์เฟสสำหรับ OGSA

PortType	Operation	Description
GridService		Query a variety of information about the Grid service instance, including basic introspection Information(handle,reference,primary key, home handleMap: terms to be defined), richer per-interface Information, and service-specific information (e.g., service instances know to registry), Extensible support for various query languages.
	SetTerminationTime	Set(and get) termination time for Grid service instance
	Destory	Terminate Grid Service instance
Notification-Source	DeliverNotification	Subscribe to notifications of service-related events. Based on message type and interest statement. Allows for delivery via third party message services.
Notification-Sink	DeliverNotification	Carry out asynchronous delivery of notification message
Registry	RegistreService	Conduct soft-state registration of Grid service handles
	UnregistreService	Deregister a Grid service handle

ในการกำหนดการใช้งานของ OGSA นั้นจะต้องประกอบด้วย 3 ส่วนคือหนึ่งเป็นลักษณะงานชนิดของบริการ (Service Type) สองต้องเป็นการทำงานในลักษณะ Grid Service Handle (GSH) ซึ่ง GSH จะขอบริการไปยัง Grid Service Reference (GSR) โดยที่ GSR จะใช้เอกสารของ WSDL สำหรับการให้บริการ

จากหลักการข้างบนพอจะกล่าวได้ว่า OGSA รวมทั้งคุณลักษณะกำหนดการใช้งานรวมถึงการควบคุมการทำงานพร้อมกัน (Concurrence) จะนำคุณสมบัติของกริด คือแชร์ทรัพยากรมาใช้ร่วมกันและคุณสมบัติของเว็บเซอร์วิสมาใช้ในลักษณะการให้บริการเช่น การค้นหา, การประกาศ เป็นต้น โดยการทำงานนั้นกริดจะเชื่อมต่อกับโครงสร้างของเว็บเซอร์วิสและสื่อสารโดยใช้พอร์ทโดยโปรโตคอลของ WSDL โครงสร้างแสดงดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างสถาปัตยกรรมของ OGSA

จากรูปเป็นโครงสร้างสถาปัตยกรรมของ OGSA ซึ่งประกอบด้วย 4 ส่วนคือในส่วนของ Resource Layer, WEB Service layer, OGSA Architect Service และ Application layer

- Resource Layer เป็นส่วนของทรัพยากรของกริดซึ่ง ประกอบด้วยแบบฟิสิคอล และลอจิคอล (Physical and Logical) ที่ร่วมกันช่วยประมวลผลเช่น คอมพิวเตอร์, เซอร์เวอร์, หน่วยความจำ, เครือข่าย ในส่วนของลอจิคอล นั้นจะอยู่สูงกว่า ซึ่งจะเป็นฟังก์ชั่นต่าง ๆ ที่ร่วมกันใช้งานพิสิคอลอีกทีส่วนใหญ่จะมีลักษณะเป็นตัวกลาง (Middleware) ในการเชื่อมต่อเช่น File System, Database Manager, Directory และการจัดการเส้นทาง (Flow)
- WEB Service layer เป็นส่วนสำคัญของ OGSA ซึ่งจะเป็นโมเดลในการขอใช้ทรัพยากรของกริดทั้งหมด โดย OGSI จะกำหนดการใช้งานเว็บเซอร์วิส เหมือนกับ XML ในการกำหนดมาตรฐานการอินเตอร์เฟสเมื่อต้องการใช้งานทรัพยากรของกริด
- เว็บเซอร์วิสได้เพิ่มโครงสร้าง OGSI (Open grid service infrastructure) ขึ้นมาเพื่อเป็นส่วนใช้งานร่วมกับระบบกริด เช่น จัดการทรัพยากรและ เป็นตัวจัดการแอตทริบิว และกลุ่มของข้อ

มูลของกริดซึ่งจะใช้ซอฟท์แวร์ในการติดต่อกับการให้บริการของกริดสำหรับการทำ discovery, Life Cycle, State Management, Creation และ Destruction

- OGSA Service เป็นเลเยอร์ในการให้บริการของกริด ตามโครงสร้างของ SOA (Service Oriented Architecture) ซึ่งจะเป็นการให้บริการเพิ่มเติมจาก OGSI เช่น Grid Core Service, Grid program execution service, Grid data service และ Domain specific service

2.7 ระบบเอเจนต์

เอเจนต์เป็นการกระทำงานอย่างใดอย่างหนึ่งซึ่งเอเจนต์อาจจะเป็นบุคล เครื่องจักร ส่วนของซอฟท์แวร์ที่สามารถทำงานอย่างใดอย่างหนึ่ง ลักษณะคุณสมบัติของเอเจนต์ประกอบด้วย

- Autonomous: สามารถทำงานได้อย่างอิสระโดยปราศจากผลกระทบจากสภาวะแวดล้อมภายนอก
- Interactive: มีการสื่อสารกับสภาวะแวดล้อมและเอเจนต์อื่น
- Adaptive: สามารถตอบสนองต่อเอเจนต์อื่นหรือสภาวะแวดล้อมนอกจากนี้ยังสามารถปรับตัวเองให้เหมาะสมกับสิ่งที่เกิดขึ้น
- Sociable: มีการร่วมกระทำกับเอเจนต์ได้โดยร่วมกันทำงาน
- Mobile: สามารถเคลื่อนที่ (Transport) จากสภาวะแวดล้อมหนึ่งไปสู่อีกสภาวะแวดล้อมหนึ่ง หรือจากสถานที่หนึ่งไปอีกที่หนึ่ง
- Proxy: การทำงานในลักษณะตัวแทน โดยทำแทนคนใดคนหนึ่งหรือสิ่งใดสิ่งหนึ่ง
- Proactive: มีความมั่นคงเมื่อมีผลกระทบจากสภาวะแวดล้อมไม่ทำให้มีผลต่อการทำงาน
- Intelligent: สามารถเรียนรู้และติดต่อกับเอเจนต์อื่นโดยใช้เครื่องหมายภาษา (Symbolic Language)
- Rational: สามารถเลือกกระทำงานพิเศษบางอย่างที่จะทำให้ตรงตามจุดประสงค์และการเรียนรู้ของของตัวเอง
- Unpredictable: สามารถทำงานต่อได้ในกรณีที่สิ่งที่จะกระทำมีเงื่อนไขไม่สมบูรณ์
- Temporally continuous: สามารถทำงานต่อเนื่องเป็นเวลานานได้
- Coordinative:สามารถกระทำงานบางเหตุการณ์ร่วมกับเอเจนต์อื่น ที่อยู่ในสภาวะแวดล้อมเดียวกัน
- Cooperative: สามารถช่วยหรือร่วมกับเอเจนต์อื่นที่มีจุดประสงค์ร่วมกัน ซึ่งอาจจะกระทำการสำเร็จหรือล้มเหลวร่วมกัน

- Competitive: สามารถช่วยเอเจนต์อื่นกระทำงานต่อ เพื่อให้สำเร็จตามจุดประสงค์ของเอเจนต์ตัวนั้นกรณีที่เอเจนต์นั้นล้มเหลว

จากคุณสมบัติของเอเจนต์ที่กล่าวมาจะเห็นว่า เอเจนต์จะเป็นการทำงานในลักษณะที่ช่วยเพิ่มประสิทธิภาพ ตรวจสอบ ควบคุม หรือเรียนรู้การทำงานของหน่วยอื่น โดยคุณสมบัติเบื้องต้นที่สำคัญคือสามารถปรับตัวเองให้เข้ากับส่วนของงาน สามารถทำงานร่วมกับส่วนอื่นไม่ว่าจะเป็นเอเจนต์อื่นหรือสภาวะแวดล้อมอื่นได้อย่างอิสระ

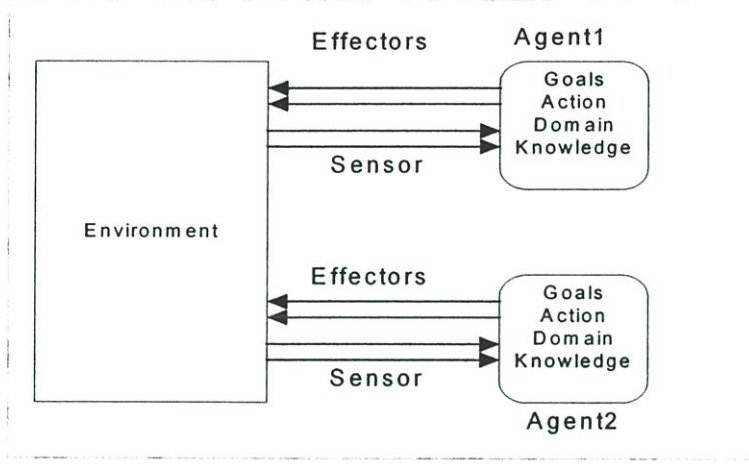
ปัจจุบันมีการเอาเอเจนต์มาประยุกต์ใช้กับระบบเทคโนโลยีทางด้านข่าวสารข้อมูล เพื่อร่วมทำงานกับซอฟท์แวร์ ฮาร์ดแวร์ หรือระบบเครือข่ายเพื่อจุดประสงค์ในการเพิ่มประสิทธิภาพการทำงาน ในงานวิจัยนี้จะใช้เอเจนต์เข้ามาเพื่อช่วยในการควบคุมคุณภาพการให้บริการทรัพยากรของระบบกริด

2.7.1 ซอฟท์แวร์เอเจนต์

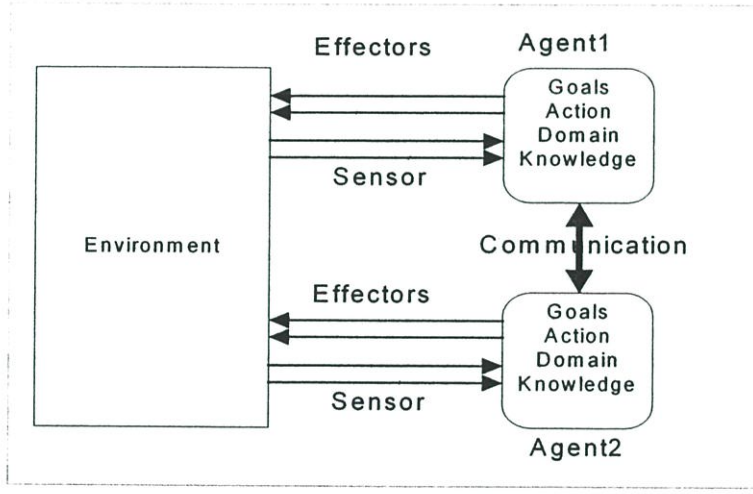
เป็นการทำงานในลักษณะใช้โปรแกรมประมวลผล (Execute Software) คุณสมบัติของซอฟท์แวร์เอเจนต์ต้องทำงานอิสระเพื่อเข้าร่วมทำงานสภาวะแวดล้อมอื่น ทั้งในส่วนของบุคคลหรือเอเจนต์ซอฟท์แวร์บนหลาย ๆ แพลตฟอร์ม (Platform) ส่วนหลักเบื้องต้นในการออกแบบซอฟท์แวร์เอเจนต์คือเป็นการใช้ซอฟท์แวร์สร้าง ภาษาที่ใช้สร้างเอเจนต์นั้นต้องสอดคล้องกับคุณสมบัติพื้นฐานของเอเจนต์ที่กล่าวมา ในปัจจุบันการออกแบบส่วนใหญ่จะใช้ภาษาที่เป็นโปรแกรมเชิงวัตถุ (Object-Oriented) หรือเครื่องมืออย่างอื่นที่สนับสนุนการทำงานของส่วนของซอฟท์แวร์ปัจจุบัน ภาษาที่นิยมใช้พัฒนาเอเจนต์เช่น Java, C^{++}, KAML, FIPACL, XML, KIFและ SLI[19]

2.7.2 ซิงเกิลเอเจนต์และมัลติเอเจนต์

ลักษณะการทำงานของเอเจนต์นั้นจะมีทั้งการทำงานและเอเจนต์เดียว และทำงานร่วมกันหลาย ๆ เอเจนต์ ในการทำงานแบบเอเจนต์เดียวนั้นจะทำงานในลักษณะแบบประมวลผลจากส่วนกลางโดยไม่มีการติดต่อกับเอเจนต์ตัวอื่น มัลติเอเจนต์นั้นนอกจากจะทำให้รูปแบบเอเจนต์เดียวแล้วระหว่างตัวเอเจนต์เองจะมีการติดต่อสื่อสารกัน เพื่อขอใช้ทรัพยากรร่วมกันหรือเพื่อการตัดสินใจในการกระทำบางกิจกรรมบางอย่าง โครงสร้างของเอเจนต์ทั้งสองแบบแสดงดังรูปที่ 2.9 และ 2.10



รูปที่ 2.9 ระบบการทำงานของเอเจนต์แบบซิงเกิล



รูปที่ 2.10 ระบบการทำงานของมัลติ-เอเจนต์

จากรูปจะเห็นว่าข้อแตกต่างหลักของระบบซิงเกิลเอเจนต์และระบบมัลติเอเจนต์คือ ซิงเกิลเอเจนต์จะมีการติดต่อกับสภาวะแวดล้อมเพียงอย่างเดียวส่วนระบบมัลติเอเจนต์นั้นจะมีการติดต่อกับสะภาวะแวดล้อมและระหว่างเอเจนต์ด้วยกัน

2.7.3 เทคโนโลยีของเอเจนต์

มีหลาย ๆ เทคโนโลยีที่รองรับการพัฒนาเอเจนต์เพื่อให้ได้คุณสมบัติการทำงานตามที่กล่าวมา ในที่นี้จะอธิบายในส่วนของ มาตรฐานของเอเจนต์ การสื่อสารระหว่างเอเจนต์ และเครื่องมือ (Toolkits) ที่นำมาพัฒนาเป็นต้น

- มาตรฐานของเอเจนต์: มีหลายมาตรฐานที่ต้องกล่าวถึง ในที่นี้จะให้ความสำคัญในด้านรูปร่าง (bodies) ของเอเจนต์ เพราะถ้าไม่พิจารณาในด้านรูปร่างของเอเจนต์ จะมีผลถึง ความเร็ว ประสิทธิภาพการทำงานและความคงทนของเอเจนต์ ในปัจจุบันมีการแบ่งเป็นมาตรฐานที่แตกต่างกันสองอย่างคือแบบคงที่ (Stationary) และแบบโมบาย โดยที่ Intelligent Physical Agent (FIPA) เป็นมาตรฐานของแบบคงที่และ Object Management Group's Mobile Agent System Interoperability Facility (MASIF) [20] เป็นมาตรฐานของแบบโมบาย FIPA เป็นซอฟท์แวร์เอเจนต์กำหนดมาตรฐานขึ้นในปี 1996 เพื่อให้ใช้ในด้านธุรกิจโดยที่ไม่หวังผลกำไร มาตรฐานหลักของ FIPA นั้นคือการสื่อสารถึงกันระหว่างเอเจนต์ ส่วน MASIF จะไม่มีการสื่อสารถึงกันระหว่างเอเจนต์ แต่จะกำหนดการติดต่อด้วย Remote Procedure Call(RPC) หรือ Remote Method Invocation (RMI)

- การสื่อสารระหว่างเอเจนต์: วิธีการสื่อสารระหว่างเอเจนต์จะภาษาที่เรียกว่า Agent Communication Language (ACL) ที่มีอยู่หลายภาษา เช่น KQML, Arcol and FIPA, KIF และ XML-base ซึ่ง ACLเหล่านี้จะเป็นตัวกลางในการติดต่อสื่อสารระหว่างข่าวสารซึ่งเทคโนโลยีที่ใช้ในการสร้างกลไกของข่าวสารเหล่านั้นอาจจะเป็น CORBA, OMG Message Service, Java message Service, RMI, DCOM และ Enterprise Java Beans Event ส่วนเทคโนโลยีที่จะรองรับการทำงานของเอเจนต์ในส่วนของความหลากหลายที่จะมำทำงานร่วมกัน ซึ่งเอเจนต์นั้นอาจจะมีหลายส่วนทำงานร่วมกันเทคโนโลยีที่มีอยู่อาจจะใช้ UML,MOF,OKBD,XML-Schema [19] เป็นต้น นอกจากนี้การสร้างเอเจนต์จะต้องคำนึงถึง โปรโตคอลที่ติดต่อสื่อสารกันภายในแต่ละส่วน นอกจากนี้เทคโนโลยีต่าง ๆ ที่ใช้พัฒนาเอเจนต์จะต้องคำนึงถึงโครงสร้างภายใน (Internal Agent) [19] การจัดการช่วงชีวิตของเอเจนต์ (Life Cycle management)[19] และสถานะของเอเจนต์เช่นอยู่กับที่หรือแบบโมบาย

- เอเจนต์ทูลคิท (Agent toolkits) ปัจจุบันมีกลุ่มผู้พัฒนาเอเจนต์จำนวนมาก ทีพัฒนาเอเจนต์ขึ้นมาสำหรับงานวิจัย และงานทางด้านธุรกิจ ซึ่งแสดงในตารางที่ภาคผนวก ก. ในส่วนท้ายของวิทยานิพนธ์เล่มนี้

ปัจจุบันมีการนำเอาเอเจนต์มาประยุกต์ใช้กับงาน ในระบบเครือข่ายแบบกระจายและระบบการจัดการข้อมูลทั้งขนาดใหญ่รวมทั้งในส่วนของงานบุคคล เช่น ในระบบเอกสารแบบฉลาด (Smart Document) ระบบ B2B [19] หรือระบบงานด้านควบคุมโปรเซสส์ (Process Control) การทำงานในระบบการจัดการส่วนบุคคล เช่น ระบบอีเมส์เอเจนต์ ระบบเลขาส่วนตัวอัตโนมัติ ในปัจจุบัน เอเจนต์ได้เข้ามามีประโยชน์อย่างมากในการบริหารจัดการกับระบบประมวลผลแบบกริด เช่น การจัดการทรัพยากร การควบคุมการให้บริการ การทำการค้นหา เป็นต้น

บทที่ 3
การควบคุมคุณภาพการให้บริการ

3.1 บทนำ

ระบบเครือข่ายคอมพิวเตอร์จะไม่เกิดปัญหาขึ้นเช่น ความคับคั่งของข้อมูล (Congestion) ปัญหาคอขวด (Bottlenecks) หรือการส่งข้อมูลช้า รวมถึงการสูญเสียข้อมูลบางส่วน (Data loss) เมื่อระบบเครือข่ายมีแบนด์วิดท์มากพอที่จะให้บริการกับทราฟฟิค (Traffic) ของงานประยุกต์หรือแอพพลิเคชั่นที่เกิดขึ้นในระบบเช่น งานประชุมทางไกล (Video Conference) งานด้านภาพ (Video) เสียง (Audio) หรืองานระบบออกอากาศโทรทัศน์บนอินเตอร์เน็ต เป็นต้น ซึ่งปัญหาที่เกิดขึ้นบางทีอาจจะแก้ได้โดยการเพิ่มแบนด์วิดท์ หรือทรัพยากรบางอย่างเช่น หน่วยความจำซึ่งส่วนใหญ่เป็นเครื่องมือทางด้านฮาร์ดแวร์ และเนื่องจากแบนด์วิดท์หรือทรัพยากรเหล่านั้นมีจำกัดไม่สามารถรองรับกับงานประยุกต์ที่พัฒนาอย่างรวดเร็วบนระบบเครือข่าย คอมพิวเตอร์แบบกระจายขนาดใหญ่ในปัจจุบัน ดังนั้นจึงต้องหาวิธีการที่จะมาบริหารหรือจัดการทรัพยากรของระบบเครือข่ายคอมพิวเตอร์เพื่อให้สามารถรองรับกับ งานประยุกต์ได้และทำให้ใช้งานประยุกต์นั้นได้อย่างมีประสิทธิภาพและเกิดความพึงพอใจกับผู้ใช้ระบบ ตัวอย่างระบบสามารถจัดหาแบนด์วิดท์กับงานที่กำลังฉุกเฉินหรือวิกฤตได้ และในทางตรงข้ามก็สามารถจำกัดแบนด์วิดท์ให้กับงานที่ไม่ฉุกเฉินหรือไม่วิกฤตได้ วิธีการที่ใช้อยู่ในปัจจุบันเรียกว่าการควบคุมคุณภาพการให้บริการ (Quality of Service: QoS)

เนื้อหาในบทนี้ ส่วนแรกจะกล่าวถึงกลไกและข้อกำหนดทั่วไปในการควบคุมคุณภาพการให้บริการ ส่วนที่สองจะกล่าวถึงโครงสร้างสถาปัตยกรรมในการการควบคุมคุณภาพการให้บริการรวมถึง รูปแบบที่ทำกันอยู่ทั่วไปและประเภทในการการควบคุมคุณภาพการให้บริการ ส่วนที่สามจะกล่าวถึงเทคนิคการจัดการและควบคุมแบนด์วิดท์ในเครือข่าย ซึ่งเป็นลักษณะการให้บริการในวิทยานิพนธ์นี้ และส่วนสุดท้ายจะกล่าวถึงการควบคุมคุณภาพการให้บริการ บนระบบการประมวลผลแบบกริดรวมถึงการประยุกต์ใช้เอเจนต์แต่ละประเภทพอประมาณ

3.2 หลักในการควบคุมคุณภาพการให้บริการ

การควบคุมคุณภาพการให้บริการ เป็นการหาวิธีการหรือเทคนิคที่ทำให้ระบบหรือเครือข่ายคอมพิวเตอร์มีความสามารถเพียงพอ ในการรองรับการให้บริการทราฟฟิคของงานประยุกต์บนเครือข่ายคอมพิวเตอร์ที่มีเทคโนโลยีของการจัดการข้อมูลที่แตกต่างกันหลากหลายรูปแบบ เช่น

อีเทอร์เน็ท , เฟรมรีเลย์และเอทีเอ็มเป็นต้น ซึ่งเทคโนโลยีการจัดการของเครือข่ายที่ต่างกันเหล่านี้ทำให้เกิดรูปแบบของทราฟฟิคที่ต่างกัน รูปแบบการการควบคุมคุณภาพการให้บริการจึงมีหลายวิธีการเพื่อรองรับทราฟฟิคแต่ละแบบเหล่านี้ให้เหมาะสม ประสิทธิภาพของการควบคุมคุณภาพการให้บริการ จะเป็นการวัดค่าพารามิเตอร์ที่เรียกว่า QoS เมตริก (QoS Metric) ให้ได้ค่าตามที่งานประยุกต์นั้นต้องการเช่น ดีเลย์ไทม์ (Delay time) ซึ่ง QoS เมตริกนี้จะขึ้นอยู่กับชนิดของงานหรืองานประยุกต์เช่น การส่งข้อมูล, (Data Transfer), FTP และมัลติมิเดีย (Multimedia) จะใช้พารามิเตอร์ต่างกัน

3.2.1 รูปแบบในการควบคุมคุณภาพการให้บริการ

เป็นแนวทางหรือหลักการต่าง ๆ ที่นำมาเป็นรูปแบบในการควบคุมคุณภาพการให้บริการซึ่งพอที่จะสรุปได้ทั้งหมด 5 หลักการด้วยกันคือ

- Integration: จัดการโดยส่วนควบคุมคุณภาพการให้บริการ โดยการจัดการ (Configurable, Predictable และ Maintainable) บนทุกโครงสร้างเลเยอร์จากด้านส่งถึงด้านรับ โมดูลสำหรับการควบคุมคุณภาพการให้บริการต้องจัดการทรัพยากร (CPU, memory, devices) ตามเส้นทางในเลเยอร์จากบนลงสู่เครือข่ายจนถึงปลายทาง โดยต้องจัดเตรียมการคอนฟิกกูเรชั่น (โดยกลไก QoS Specification) การรับประกันทรัพยากร (โดยกลไก QoS Control) และการบำรุงรักษา (โดยกลไก QoS monitoring) ตามเส้นทาง

- Separation: จัดการโดยตัวกลางในการส่ง การควบคุมและการจัดการ ซึ่งเป็นกิจกรรมโครงสร้างของฟังก์ชั่นการทำงานที่ต่างกัน กิจกรรมควรจะแยกในโครงสร้างของเฟรมเวิกร์ เช่นการแยกความแตกต่างระหว่างสัญญาณและตัวกลางในการส่ง ตัวกลางปกติต้องการแบนด์วิดท์สูง ลาเทนซีต่ำซึ่งจะให้ความไม่แน่นอนใน Jitter ในขณะที่สัญญาณ (แบบเต็มคลื่น (Full duplex) และเป็นอะซิงโครนัส) ปกติจะต้องการแบนด์วิดท์ต่ำเพื่อรับประกันความแน่นอนในเรื่อง Jitter

- Transparency: จัดการโดยส่วนงานประยุกต์ ควรจะลดความซับซ้อนของงานประยุกต์โดยฟังก์ชั่นของ QoS Specification และ QoS management ตัวอย่างที่สำคัญในการจัดการง่ายคือการใช้ API ประโยชน์ของทรานพาเรนซีมีสามอย่างคือลดการฝังตัวของฟังก์ชั่นการควบคุมคุณภาพการให้บริการในงานประยุกต์ ปิดบังรายละเอียดข้อกำหนดการบริการงานประยุกต์ และความซับซ้อนจะถูกทดแทนโดยกิจกรรมของ QoS management

- Asynchronous Resource Management: เป็นแนวทางในการแยกฟังก์ชั่นการทำงานระหว่าง โครงสร้างของโมดูล รูปแบบการควบคุมและกลไกการจัดการ ซึ่งจะมีผลสะท้อนโดยตรงต่อเวลาของการทำงานแบบขนานกับกิจกรรมในระบบการสื่อสารแบบกระจายเช่น กิจกรรมในการ Scheduling, flow control, routing, QoS Management

- Performance: เป็นข้อตกลงที่กว้างสำหรับการควบคุมคุณภาพการให้บริการ ในส่วนของออกแบบการสื่อสารและการใช้งาน ข้อตกลงเหล่านี้จะเกี่ยวกับการแบ่งฟังก์ชั่นในโปรโตคอลสำหรับสื่อสารระเพื่อให้ได้ประสิทธิภาพสูงตามหลักของ "Saltzer's system design principles" โดยจะหลีกเลี่ยงวิธีของการมัลติเพลก (Multiplexing) สนับสนุนโครงสร้างโปรโตคอลการสื่อสาร เช่นการรวมเฟร์มงานประยุกต์และเลเยอร์ของการประมวลผลหรือการใช้ฮาร์ทแวร์ช่วยสำหรับการประมวลผลของโปรโตคอล

3.2.2 **ข้อกำหนดของการควบคุมคุณภาพการให้บริการ** (QoS Specification)

เกี่ยวข้องกับการควบคุมคุณภาพการให้บริการระดับงานประยุกต์ ในส่วนของข้อกำหนดและการจัดการโพลิซีซึ่งจะต่างจากเลเยอร์อื่นที่จัดการในส่วนของคอนฟิกและเมนเทน (maintain) โดยปกติจะกำหนดความต้องการของผู้ใช้ก่อนที่จะใช้กลไกในการทำ ซึ่งประกอบด้วย

- Flow Synchronization: เกี่ยวข้องกับการกำหนดเส้นทางการส่งหลายเส้นทางพร้อมกัน
- Flow Performance: เกี่ยวข้องกับกำหนดประสิทธิภาพของเส้นทางการส่งของผู้ใช้
- Flow Commitment: เกี่ยวข้องกับการตรวจสอบความต้องการทรัพยากรแบบ end-to-end
- QoS Management Policy: เกี่ยวข้องกับการปรับปรุงเส้นทางเพื่อให้เหมาะสมและรองรับการกระทำต่าง ๆ ที่เกิดขึ้นกับเส้นทางได้
- Cost of Service: เกี่ยวข้องกับค่าใช้จ่ายที่จะเกิดขึ้นในแต่ละระดับการให้บริการ ซึ่งถือได้ว่าเป็นข้อกำหนดที่สำคัญที่สุดที่จะต้องพิจารณา

3.2.3 **กลไกในการควบคุมคุณภาพการให้บริการ** (QoS Mechanisms)

การควบคุมคุณภาพการให้บริการตามข้อกำหนดที่กล่าวมาข้างต้นนั้นจะมีเทคนิคหรือกลไกหลายรูปแบบขึ้นอยู่กับว่าเป็นข้อกำหนดแบบไหน กลไกแต่ละแบบจะมีข้อแตกต่างกันออกไป เช่น การจัดการทรัพยากรหรือการจัดการการไหลของข้อมูลซึ่งเทคนิคที่มีอยู่ประกอบด้วย

3.2.3.1 **การจัดหา** (Provision) เป็นกลไกที่ใช้ในการจัดเตรียมประกอบด้วย 3 ลักษณะคือ

- QoS mapping เป็นการจัดการฟังก์ชั่นแปลความหมายอัตโนมัติของการควบคุมคุณภาพการให้บริการระหว่างการแทนค่ากับระดับที่ต่างกันของระบบ เช่น ระบบปฏิบัติการ เครือข่าย หรือระดับของระดับ ISO เลเยอร์ ทำให้ไม่ต้องคำนึงถึงข้อกำหนดของระดับล่าง
- Admission testing เป็นกลไกในการเปรียบเทียบความต้องการทรัพยากร (Resource) ที่เพิ่มขึ้นเมื่อมีการควบคุมคุณภาพการให้บริการกับทรัพยากรของระบบ
- Resource Reservation: เป็นโปรโตคอลในการจัดการกำหนดระบบความเหมาะสมของระบบสุดท้าย (End System) และทรัพยากรของเครือข่ายตามข้อกำหนดของผู้ใช้

3.2.3.2 **การควบคุม** (Control) เป็นกลไกในการจัดการเวลาของความเร็วสำหรับตัวกลางในการส่ง ซึ่งจะเตรียมเส้นทางทราฟฟิคแบบเรียวไทม์ โดยการควบคุมโฟลว์บนพื้นฐานของระดับการร้องขอในช่วงเริ่มต้นระหว่างการเซ็ตอัพ ในส่วนของ QoS Control ประกอบด้วยส่วนประกอบย่อยดังนี้คือ

- Flow Shaping: เป็นการจัดแบนด์วิดท์ในการส่งซึ่งอาจจะเป็นการจัดแบบง่ายโดยใช้เต็มแบนวิดท์ หรือใช้การจัดการทางสถิติเข้าช่วย
- Flow Scheduling เป็นการจัดการเส้นทางล่วงหน้า (Forwarding) ส่วนใหญ่จะใช้จัดการในระดับงานประยุกต์แต่บางครั้งก็รวมถึงในส่วนของเลเยอร์ของเน็ตเวิร์ค
- Flow Policing อาจจะมองคล้ายแบบมอนิตอริง (monitoring) จะเกี่ยวกับ QoS Management จะเป็นการใช้กฎหรือนโยบายในการควบคุม โดยส่วนใหญ่จะเหมาะสมกับการจัดการงานบางจุดเช่น การอินเตอร์เฟสของ user-to-Network
- Flow control ใช้จัดการเส้นทางแบบปิดและแบบเปิด (open-loop and closed loop) ซึ่งส่วนใหญ่ใช้มากกับระบบโทรศัพท์
- Flow Synchronization ใช้ควบคุมลำดับของเหตุการณ์และเวลาที่แน่นอน ซึ่งส่วนมากจะใช้กับการจัดการของระบบมัลติมีเดีย

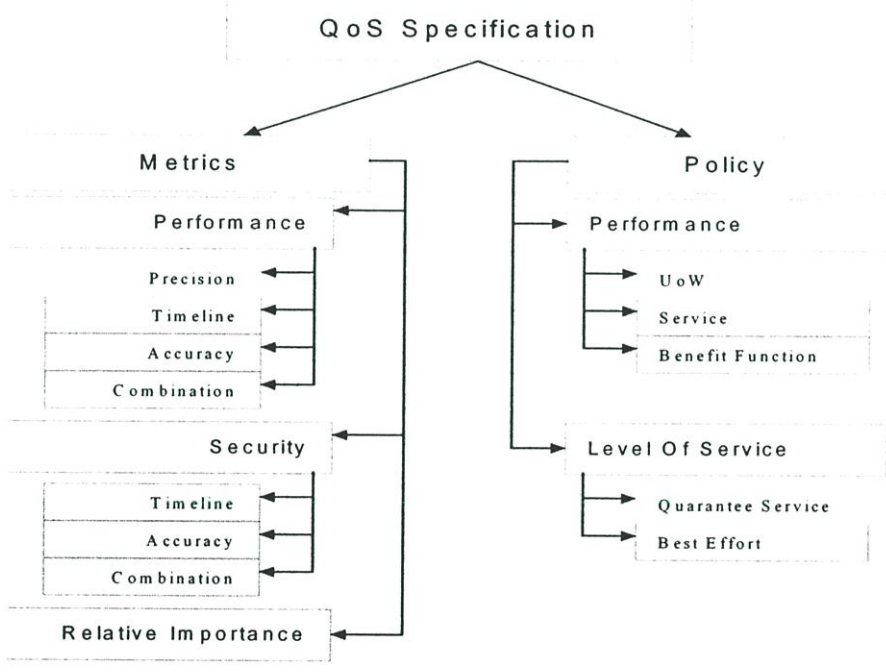
3.2.3.3 **การจัดการ**(Management) ใช้รักษาข้อตกลงของระดับการควบคุมคุณภาพการให้บริการ โดยปกติจะไม่เพียงพอต่อการใช้ทรัพยากรที่มีอยู่ ส่วนใหญ่แล้วกลไกของ QoS Management จะไม่ใช้ทำควบคุมคุณภาพการให้บริการโดยตรง แต่จะเป็นตัวช่วยสนับสนุนการทำควบคุมคุณภาพการให้บริการอีกที เทคนิคย่อยของ QoS Management ประกอบด้วย

- QoS Monitoring: เป็นการยอมให้แต่ละระดับของระบบสามารถจดจำได้โดยเลเยอร์ต่ำ
- QoS Maintenance: ใช้เปรียบเทียบคุณภาพของการบริการเพื่อประเมินประสิทธิภาพและหลังจากนั้นก็แก้ไขโดยการปรับปรุงการทำงาน ของทรัพยากรที่ช่วยสนับสนุนการควบคุมคุณภาพการให้บริการ เช่นปรับปรุงข้อผิดพลาดของ การจัดการบับเฟอร์ คิวดีเลย์
- QoS Degradation: เป็นตัวชี้วัดหรือตัวแสดงการทำงานของระบบ ซึ่งอาจเป็นการแสดงผลในระดับต่ำ (lower layer) ถ้ามีปัญหาก็จะทำการแก้ไขโดย QoS Maintenance
- QoS Signaling: เป็นเทคนิคที่ยอมให้ผู้ใช้กำหนดค่าช่วงของพารามิเตอร์และสามารถตรวจสอบประสิทธิภาพของระบบที่เกิดขึ้นที่ค่านั้นได้
- QoS Scaling จะเป็นการเปรียบเทียบ QoS Filtering ซึ่งเป็นเส้นทางในการส่งถึงระบบการสื่อสาร และ QoS Adaptation ซึ่งช่องทางของ end system อย่างเดียว

3.2.4 ประเภทของข้อกำหนดในการควบคุมการให้บริการ (QoS Specification)

เป็นข้อกำหนดเกี่ยวกับรูปแบบการควบคุมคุณภาพการให้บริการของงานประยุกต์ ซึ่งปกติแล้วจะมีรูปแบบการทำที่แตกต่างกันทั้งของงานประยุกต์และระบบ ตัวอย่างเช่น ระบบแบบกระจาย (Distributed system) การทำการควบคุมคุณภาพการให้บริการจะทำที่ User-oriented มากกว่า System-oriented

ข้อกำหนดในการทำการควบคุมคุณภาพการให้บริการสามารถแบ่งโครงสร้าง (Taxonomy) ได้เป็นสองรูปแบบใหญ่ ๆ ดังรูปที่ 3.1 คือกลุ่มของการทำการควบคุมคุณภาพการให้บริการ โดยการใช้เมตริกเป็นตัววัดประสิทธิภาพและการกำหนดโพลิซี (Policy) QoS เมตริก เป็นการทำการควบคุมคุณภาพการให้บริการโดยการวัดเมตริกพารามิเตอร์ให้ได้ค่าตามงานประยุกต์ หรือระบบนั้นต้องการ ส่วนโพลิซีนั้นเป็นการสร้างข้อกำหนดหรือเงื่อนไขของพารามิเตอร์ขึ้นมาบังคับการให้บริการ ส่วนใหญ่จะใช้กับการรับประกันการให้บริการ (Service Level Agreements) [22] ซึ่งเป็นการรับประกันการให้บริการระหว่างผู้ให้บริการ (Service Providers) และผู้ใช้บริการ (Users)



รูปที่ 3.1 โครงสร้างข้อกำหนดในการควบคุมคุณภาพการให้บริการ

3.2.5 เมตริกสำหรับควบคุมคุณภาพการให้บริการ (QoS Matrix)

การควบคุมคุณภาพการให้บริการตามข้อกำหนดของเมตริกนั้น จะใช้เมตริกเป็นค่าที่บอกหรือกำหนดเพื่อรองรับการควบคุมคุณภาพการให้บริการ เรียกว่าพารามิเตอร์ หรือ QoS เมตริก

โดยค่าเหล่านี้จะมีหลายลักษณะ เช่น ถ้าเป็นงานประยุกต์จะเป็นการวัดงานประยุกต์แต่ละแบบ เช่น งานประยุกต์ทางด้านฐานข้อมูลจะวัดเวลาตอบสนองที่เกิดขึ้นกับระบบเป็นต้น

โดยทั่วไปแล้วเมตริกที่ใช้ในการทำการควบคุมคุณภาพการให้บริการ จะแบ่งออกเป็น 3 ประเภท แสดงดังรูปที่ 3.1 ประกอบด้วย Performance, Security และ Relative Importance เมตริก โดยที่ Performance นั้นจะมีการวัดค่าประสิทธิภาพเกี่ยวกับเวลาเช่น delay, latency ความถูกต้องแน่นอนเกี่ยวกับข้อมูล Security เป็นเมตริกที่ใช้วัดค่าเกี่ยวกับความถูกต้องและมั่นคงของข้อมูล ในส่วนของ Relative Importance เป็นเมตริกที่ใช้วัดค่าใช้จ่ายที่จะเสียไปในการใช้งาน จะเห็นได้ว่าเมตริกนั้นมีอยู่มาก แล้วแต่จะทำการควบคุมคุณภาพการให้บริการตามประเภทของงาน ประเภทของระบบและระดับหรือเลเยอร์ เป็นที่น่าสังเกตว่าในการวัดค่าของการควบคุมคุณภาพการให้บริการนั้นไม่จำเป็นว่าต้องวัดประสิทธิภาพด้านเวลาเพียงอย่างเดียว

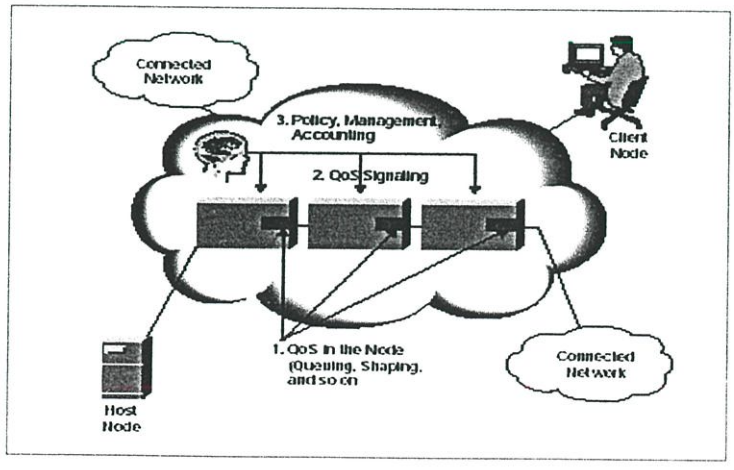
3.2.6 โครงสร้างสถาปัตยกรรมพื้นฐานของการทำQoS

โครงสร้างสถาปัตยกรรมพื้นฐานของการควบคุมคุณภาพการให้บริการ แบ่งได้สามประเภท ดังแสดงในรูป 3.2 ประกอบด้วย

3.2.4.1 **การควบคุมคุณภาพการให้บริการในเครือข่ายเดียว**(single network element)

การควบคุมคุณภาพการให้บริการภายในครือข่ายเดียวกัน (Single Network) เช่น การทำกับโฮสต์ (Host) แบบนี้จะต้องใช้เทคนิคต่าง ๆ เข้ามาจัดการซึ่งประกอบด้วย

- การควบคุมความคับคั่งของข้อมูล(Congestion) [4,23] เพื่อลดวามคับคั่งของข้อมูลที่เกิดขึ้นในเครือข่ายซึ่งเทคนิคต่าง ๆ ที่ประกอบไปด้วย การใช้ Priority Queue (PQ) [4] custom queuing (CQ), weighted fair queuing (WFQ) และ class-based weighted fair queuing (CBWFQ) เข้ามาช่วยจัดการ



รูปที่ 3.2 โครงสร้างสถาปัตยกรรมพื้นฐานของการทำ QoS

• การจัดการคิว (Queue management): ใช้เป็นตัวทำ QoS ในเครือข่ายอีกรูปแบบหนึ่ง โดยคิวนั้นจะไม่จำกัดขนาดดังนั้นบางครั้งจะทำให้เกิดโอเวอร์โฟลว์ (Overflow) จึงต้องหาวิธีการจัดการซึ่งปัจจุบันจะใช้ Weighted early random detection (WRED) [4] เป็นตัวจัดการโดย random early detection (RED)

• การจัดรูปแบบและนโยบายจราจร (Traffic Shaping and Policing): เป็นการสร้างโฟลว์ข้อมูลที่จำกัดตามขนาดของแบนวิดท์ ซึ่งป้องกันการเกิดโอเวอร์โฟลว์ ตัวอย่างของการจัดการ Traffic-shaping โดยบริษัท Cisco คือ generic traffic shaping (GTS) และ Frame Relay traffic shaping (FRTS) และในส่วนของ Policing tool คือ committed access rest (CAR)[4]

• การจัดการลิงค์ (Link Efficiency): เป็นการจัดการแบนวิดท์ของโฟลว์โดยการทำแฟลกเมนเตชั่น โดยการแบ่งแพคเก็ตใหญ่ ออกเป็นแพคเก็ตย่อย เพื่อให้เหมาะสมกับขนาดของแบนวิดท์ที่มีอยู่ ตัวอย่างของการจัดการลิงค์ Link Efficiency คือ Link-flagmentatian and interliaring (LEI) และ real-Time protocol header compression (RTP-HC) ในส่วนของ LFI [4]

3.2.4.2 **การสร้างและการทำเครื่องหมาย**(identification and marking)

เป็นการควบคุมคุณภาพการให้บริการแบบเป็นกลุ่มหรือคลาส (class) การให้บริการ โดยหลักการของวิธีนี้จะนำแพคเก็ตของข้อมูลมาจัดการแบ่งกลุ่ม ส่วนมากใช้กับการส่งข้อมูลที่ใช้ ไอพี (IP) เป็นหลักซึ่งจะใช้ส่วนหัวของแพคเก็ต (packet header) นำมาแบ่งกลุ่มโดยการจัดไพรออริตี้ การควบคุมคุณภาพการให้บริการแบบนี้ ทำให้เกิดการบริการหลายแบบเช่น ToS (Type of Service) หรือ Differential Service

วิธีการจัดการมีอยู่ 2 ขั้นตอนที่สำคัญ คือ การทำ Identification และการ Mark ในการทำ Identification นั้นเดิมจะใช้เทคนิของ ACL (access control list) [4] เป็นการเลือกทราฟฟิคสำหรับการจัดการความคับคั่งของข้อมูลและนอกจากนี้ยังใช้ NBAR (Network-base Application recognition) [4] จัดทำคลาสได้ละเอียดกว่าเทคนิคของ ACL โดย NBAR จะจัดการทราฟฟิคโฟล์วแบบไดนามิคและจะจัดระดับทราฟฟิคได้ลึกกว่า ACL โดยสามารถจัดได้ถึง URL หรือ HTTP แพคเก็ตซึ่งมีประโยชน์มากกับระบบอินเตอร์เน็ตในปัจจุบัน

สำหรับการทำ Precedence นั้นก็มีหลายวิธีการเช่นวิธีการของ PBR (policy-Base Routing)[4] ซึ่งวิธีการนี้จัดโดยการใช้เงื่อนไขกับข้อมูลที่เข้ามา (Access list criteria) และใช้บิตของ IP precedence อีกแบบคือ CAR (committed access list)[4] แบบนี้จะแบ่งคลาสของทราฟฟิค โดยการมองที่การอินเตอร์เฟส (Interface) วิธีการนี้จะยุ่งยากกว่าวิธีการของ PBR การ

ทำ Precedence อีกแบบจะใช้หลักการของ ToS [4] แบบนี้ จะทำให้เกิดการควบคุมคุณภาพการให้บริการที่สำคัญของ Differentiated คือ IPV4 หรือ IPV6 [4]

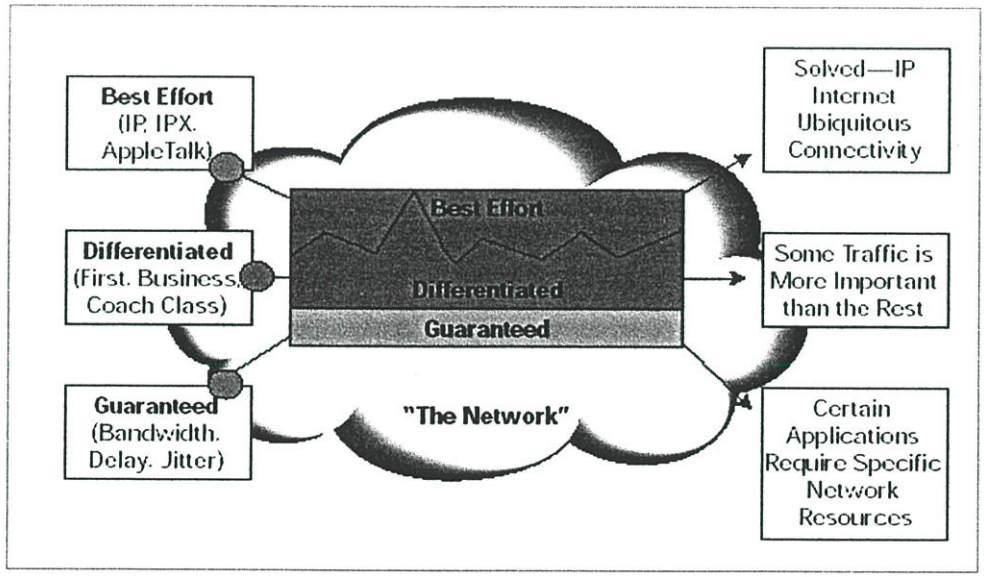
3.2.4.3 **การใช้นโยบายและการจัดการ** (Policy and Management)

นั้นเป็นการจัดการโดยใช้นโยบาย(Policy) และการบริหารจัดการทราฟฟิคผู้ใช้แบบ end-to-end [24,25] เป็นการควบคุมคุณภาพการให้บริการที่สำคัญในระบบแบบกระจายและระบบกริด

3.2.4 **การควบคุมคุณภาพการให้บริการแบบจุดต่อจุด** (End-to-End)

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์ได้พัฒนาไปอย่างรวดเร็ว โดยมีแนวโน้มเป็นระบบแบบกระจาย (Distributed System) ดังนั้นการควบคุมคุณภาพการให้บริการจึงต้องพัฒนารูปแบบและหลักการขึ้นให้ทัน ระบบอินเตอร์เน็ตเป็นรูปแบบของระบบการกระจายที่ใช้โปรโตคอล HTTP เป็นหลัก ซึ่งการควบคุมคุณภาพการให้บริการในระบบต้องทำในลักษณะจุดต่อจุด

การจัดการจุดต่อจุดเป็นการรับส่งข้อมูล โดยที่ผู้ส่งและผู้รับไม่สามารถระบุเส้นทางที่แน่นอนได้ เช่น ระบบอินเตอร์เน็ตเป็นต้น ปัจจุบันการควบคุมคุณภาพการให้บริการในระบบนี้ มีวิธีการในการจัดการอยู่สามวิธีดังแสดงในรูปที่ 3.3



รูปที่ 3.3 รูปแบบของการควบคุมคุณภาพการให้บริการแบบจุดต่อจุด

- Best-effort เป็นการส่งข้อมูล โดยไม่รับประกันเรื่องของประสิทธิภาพ ของพารามิเตอร์ในการควบคุมคุณภาพการให้บริการ ระบบจะเลือกเส้นทางที่คิดว่าดีที่สุด การควบคุมคุณภาพการให้บริการจะมีการนำเทคนิคของการลดความคับคั่งแบบพื้นฐานมาใช้ เช่น FIFO

- Differentiated service หรือ Soft QoS เป็นการควบคุมคุณภาพการให้บริการที่ไม่ซับซ้อนซึ่งแบบนี้จะเป็นแบบการจัดกลุ่มของทราฟฟิค (Classification of traffic) ตามที่กล่าวมา นอกจากนี้ยังมีการนำเครื่องมือที่ใช้ในการควบคุมคุณภาพการให้บริการบางตัวเช่น PQ, CQ, WFQ, and WRED มาช่วย ตัวอย่างคือ DiffSer

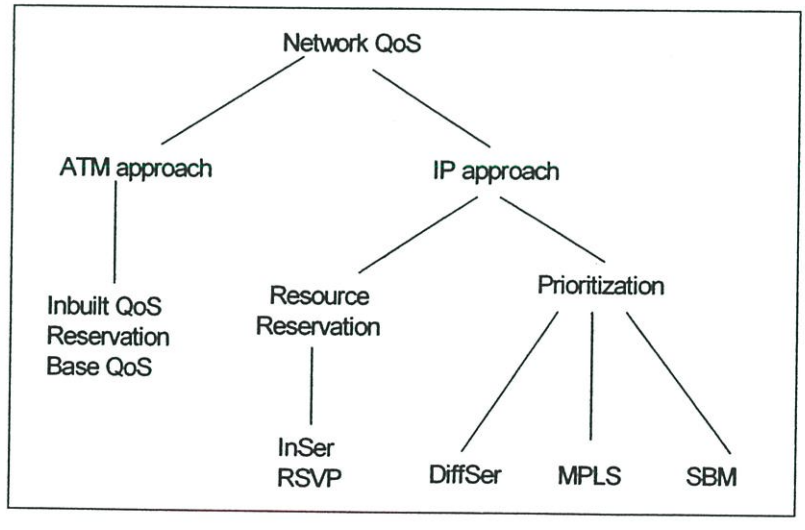
- Guaranteed service หรือ Hard QoS จะใช้นโยบายหรือโพลิซีหรือการจัดการเข้ามาจัดการ แบบนี้จะมีการจัดการที่ยุ่งยาก เพราะจะมีการจองทรัพยากรเพื่อที่จะนำมากำหนดเส้นทางของทราฟฟิค เครื่องมือที่นำมาใช้ช่วยจัดการคือ RSVP และ CBWFQ [4]

3.3 ประเภทของการทำการควบคุมคุณภาพการให้บริการ

การควบคุมคุณภาพการให้บริการที่มีใช้อยู่ในปัจจุบัน ส่วนใหญ่จะแบ่งออกเป็นสองกลุ่มคือ การควบคุมคุณภาพการให้บริการในระดับเน็ตเวิร์คและระดับงานประยุกต์

3.3.1 การควบคุมคุณภาพการให้บริการระดับเน็ตเวิร์ค

เป็นการควบคุมคุณภาพการให้บริการรูปแบบเก่า คือทำที่ระดับเน็ตเวิร์คเลเยอร์เมื่ออ้างอิงกับมาตรฐานของ ISO [23] วิธีการทำนั้นจะวัด QoS พารามิเตอร์ต่าง ๆ ในระดับเนตเวิร์ค ซึ่งอาจจะทำภายใต้เครือข่ายเดียวกันหรือข้ามเครือข่าย เช่น แบบจุดต่อจุด ในการควบคุมคุณภาพการให้บริการที่ระดับนี้จะแบ่งออกได้เป็นสองประเภทใหญ่ ๆ ตามรูปที่ 3.4 คือแบบกระทำที่ IPและระบบของ ATM (23) ในส่วนของการใช้ IP ยังแบ่งได้อีกเป็นสองกลุ่มคือ กลุ่มการจองทรัพยากร (Resource Reservation) ซึ่งจะใช้นโยบายในการจัดการและกลุ่มที่ใช้ไพรออริตี้



รูปที่ 3.4 โครงสร้างของการควบคุมคุณภาพการให้บริการที่เนตเวิร์คเลเยอร์

• Resource Reservation [26] เป็นวิธีการเกี่ยวกับการสำรองทรัพยากรซึ่งแบบนี้บางทีเรียกว่า InSer หรือ Integrate Service จะจัดการทรัพยากรตามแอพพลิชันที่ร้องขอมาตัวอย่างของโปรโตคอลแบบนี้ที่ใช้มากคือ RSVP (Resource Reservation Protocol) ซึ่งจะเป็นโปรโตคอลสำหรับการส่งข้อมูลแบบกระจายหลายจุดมีการใช้มากในการควบคุมคุณภาพการให้บริการแบบจุดต่อจุดหรือบนระบบกริดก็ยังมีการประยุกต์ใช้โปรโตคอลประเภทนี้

• Prioritization บางทีเรียก DiffSer [26] หรือ Difference จะเป็นการบริการในลักษณะ ToS (Type of Service) หรือแบบ CoS (Class of Service) จะทำการแบ่งกลุ่มของไพรออริตี้ของ Flow และจัดไว้เป็นกลุ่มเดียวกันซึ่งแบบนี้จะใช้สำหรับการควบคุมคุณภาพการให้บริการแบบ Per-hope

• MPLS (Multi Protocol Labeling) [26] จะรองรับการจัดการแบนด์วิดท์ โดยการควบคุมเส้นทางตามค่าของลาเบลของเฮดเดอร์แพคเก็ต

• SBM (Subnet Bandwidth Management) [26] สำหรับทำการควบคุมคุณภาพการให้บริการที่เลเยอร์ 2 หรือดาต้าลิงค์เลเยอร์

3.3.2 การควบคุมคุณภาพการให้บริการระดับงานประยุกต์

งานประยุกต์ที่ใช้งานอยู่บนเครือข่ายคอมพิวเตอร์แบบกระจาย จะมีผลเกี่ยวข้องกับ QoS พารามิเตอร์หลายแบบ โดยทั่วไปสามารถแบ่งแยกออกได้เป็น 2 ชนิดคือ แบบอิลาสติกและแบบเรียวไทม์ แสดงดังรูปที่ 3.5 การแบ่งจะยึดตามการตอบสนองต่อการส่งแพคเก็ตข้อมูล คือถ้างานประยุกต์มีผลตอบสนองต่อดีเลย์ในระหว่างการส่งและไม่สามารถทำงานต่อไปได้ หรือทำได้แต่ไม่สมบูรณ์และทำให้รูปแบบทางกายภาพของงานเปลี่ยนไปจะเป็นงานแบบเรียวไทม์ แต่ถ้างานนั้นสามารถรอข้อมูลที่จะส่งมาได้และไม่มีผลต่อการทำงานในขณะนั้น จะเป็นงานแบบอิลาสติก

สำหรับตัวอย่างของอิลาสติกแอพพลิเคชั่นเช่น งานแบบรีโมทเทอร์มินอล เช่น Telnet งานแบบการส่งแฟ้มข้อมูล เช่น FTP Naming Service เช่น DNS หรือแบ่งตามกลุ่มของงานคือ

- Interactive burse เช่น Telnet, X, NFS
- Interactive bulk transfer เช่น FTP
- Asynchronous bulk transfer (FTP)

สำหรับงานแบบเรียวไทม์ การทำงานจะทำเป็นแบบต่อเนื่องหรือส่งเป็นแบบสทรีม (Stream) เพื่อที่จะให้งานนั้นทำงานแบบต่อเนื่องได้ตัวอย่างของงานเช่น

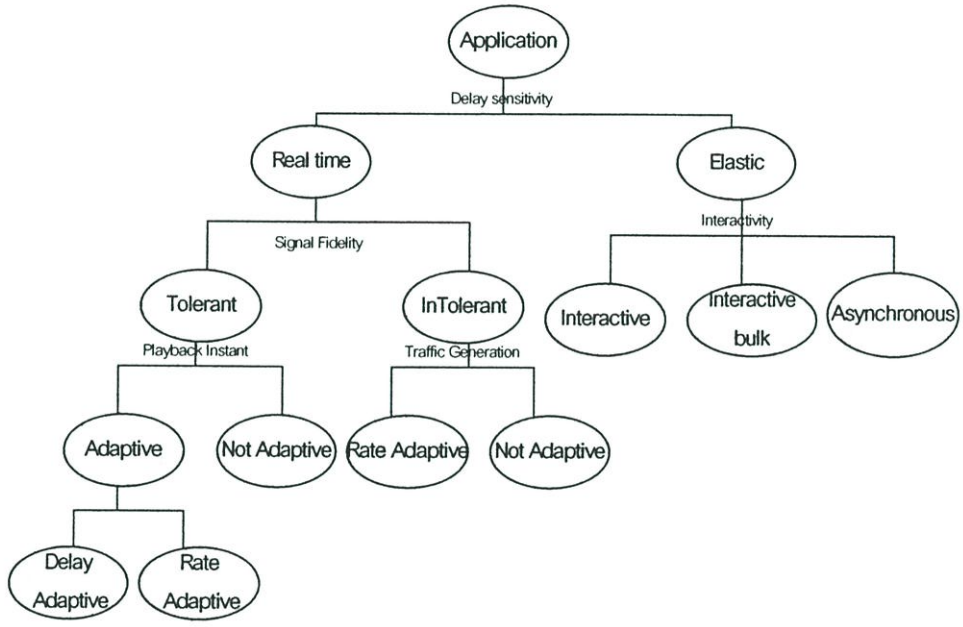
งานด้าน Audio

- Conversation audio
- High quality audio orchestration

- Professional quality audio Streaming

งานด้าน Video

- High quality audiovisual conference
- Video Conference
- Video broadcast
- High Definition TV



รูปที่ 3.5 โครงสร้างของงานประยุกต์ที่มีอยู่ในเครือข่าย

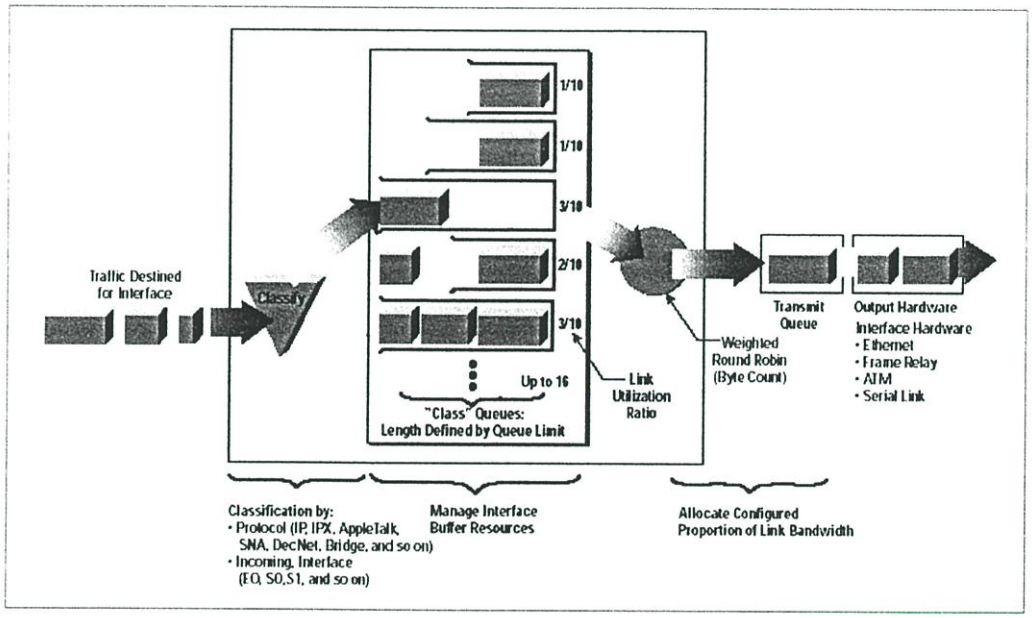
3.4 การจัดการแบนด์วิดท์

การส่งข้อมูลระหว่างคอมพิวเตอร์ในเครือข่ายนั้นต้องอาศัยเส้นทาง(Flow) สื่อสารในการส่งข้อมูลที่หลากหลายเส้นทางแต่ละเส้นทางจะมีขนาดความจุหรือแบนด์วิดท์ที่ต่างกัน จากการที่มีข้อจำกัดในส่วนของขนาดแบนด์วิดท์ ทำให้เกิดการจัดรูปแบบการส่งข้อมูลหรือการจัดลำดับของงานประยุกต์โดยการให้สิทธิในการส่งข้อมูล หรือการจัดลำดับของงานประยุกต์ด้วยวิธีการให้สิทธิในการส่งข้อมูลก่อนหลังหรือกำหนดช่องทางการส่งให้แก่งานประยุกต์ ตามขนาดความต้องการของแต่ละประเภทของงานประยุกต์นั้น ๆ ซึ่งวิธีการดังที่อธิบายมาคือเทคนิคการจัดการแบนด์วิดท์

การจัดการแบนด์วิดท์เป็นการบริหารทราฟฟิคของงานประยุกต์ ให้กับเส้นทางการส่งข้อมูล ซึ่งจะเห็นว่าก่อนที่จะจัดลักษณะของทราฟฟิคได้นั้นจะต้องมีเส้นทางการส่งก่อน ซึ่งเส้นทางการส่ง

ข้อมูลจะมีลักษณะเป็นบิลดิ้งบลอคส์(Building Block)[27,28] ในการหาเส้นทางจะมีกลไกในการเลือกเส้นทางที่ดีที่สุด ส่วนการจัดการแบนด์วิดท์นั้นต้องหาเส้นทางการส่งข้อมูลให้ได้ก่อนหลังจากนั้นจะใช้กลไกในการจัดการแบนด์วิดท์เพื่อจัดการเส้นทางให้เกิดประโยชน์สูงสุด เทคนิคต่าง ๆ ในการจัดการแบนด์วิดท์นั้นมีหลายรูปแบบ ตั้งแต่การจัดการที่เครือข่ายเดียว (Single Network) เช่น การใช้คิวในรูปแบบต่าง ๆ การจัดการที่ระดับสัญญาณ (Signaling) เช่นการทำ ToS และการให้โพลิซีเข้ามาจัดการ ตัวอย่างของการจัดการแบนด์วิดท์มีดังต่อไปนี้

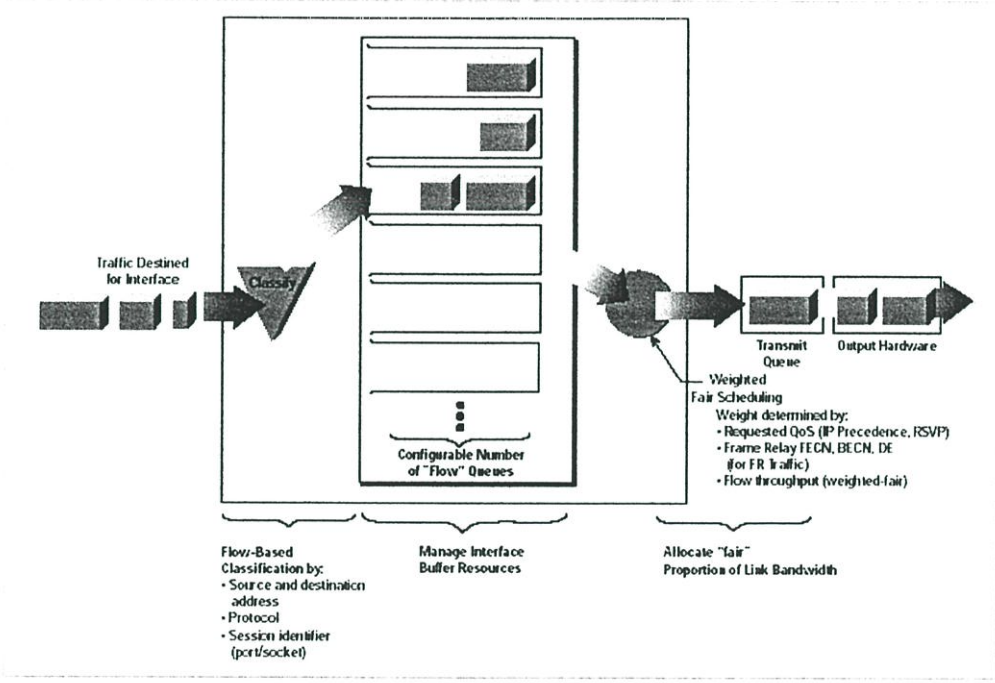
- Custom Queue (CQ) เป็นเครื่องมือในการรับประกันแบนด์วิดท์แบบง่าย โดยจะแชร์แบนด์วิดท์ให้กับงานประยุกต์หลาย ๆ งานประยุกต์ CQ จะให้บริการโดยการกำหนดจำนวนคิวให้แต่ละคลาสของแพกเก็ตและให้บริการแต่ละคิวแบบเราด์-โรบิน (round-robin) [4]



รูปที่ 3.6 รูปแบบการทำงานของ Custom queue

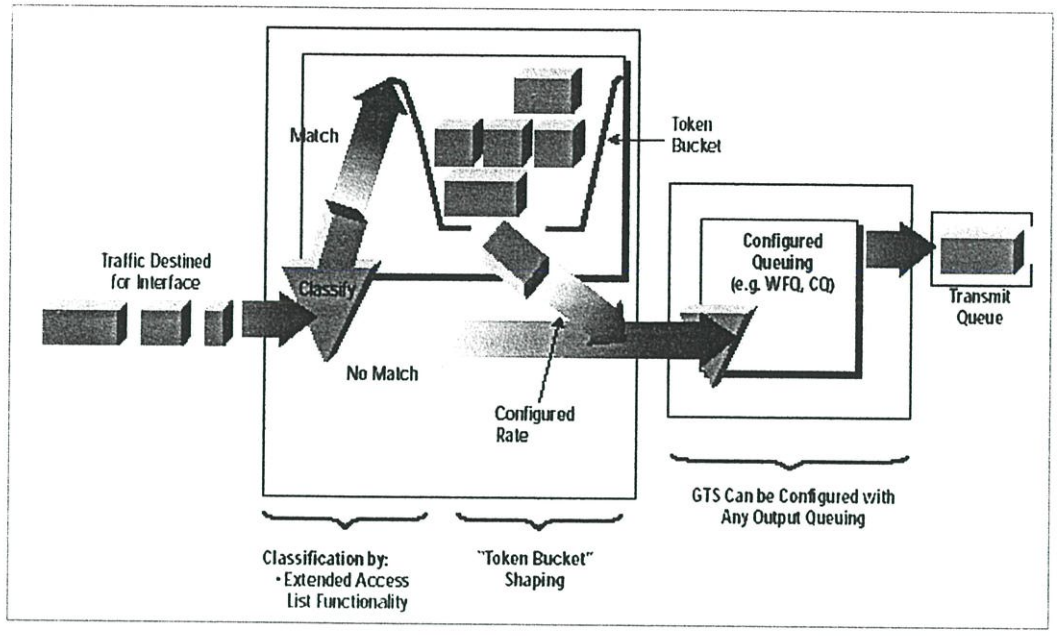
ตัวอย่างเช่นในการใช้งานกำหนดให้การใช้บริการ SNA (System Network Architecture) สามารถที่จะกำหนดแบนด์วิดท์ของเครือข่ายให้ SNA ครึ่งหนึ่งส่วนที่เหลือครึ่งหนึ่งให้บริการกับโปรโตคอลแบบอื่น จากรูป 3.6 อัลกอริทึมจะเลือกแพคเก็ตไว้ในคลาสคิวที่กำหนดสัดส่วนไว้ก่อนการเลือกบริการจะทำแบบพร้อม ๆ กันหมดทุกคิว

- Weighted Fair Queue (WFQ): เป็นการให้บริการแบบให้น้ำหนักกับแพ็คเก็ตอย่างเป็นธรรมตัวอย่างเช่นถ้าการให้บริการประกอบด้วย 2 คิว คิวแรก 100 ไบต์ คิวที่สอง 50 ไบต์ ถ้าใช้ WFQ อัลกอริทึม จะอ่าน 2 แพคเก็ตจากคิวที่สองและ 1 แพคเก็ตจากคิวที่หนึ่ง ในการอ่านแต่ละครั้ง รูที่ 3.7 แสดงการทำงานของ WFQ การประยุกต์ใช้งาน WFQ นั้นสามารถประยุกต์ใช้กับโครงสร้างของทรี (Tree) [27] และนอกจากนี้ยังมีการปรับ ปรุงอัลกอริทึมเป็น WF^2Q และ H-PFQ [27] ตามลำดับ นอกจากนี้จะมีการประยุกต์ใช้ WFQ อีกแบบโดยการสร้าง Class ให้กับงานที่เราต้องการ นั้นก็คือสามารถแยก WFQ ออกเป็นส่วนย่อยวิธีการนี้เรียกว่า Class-Base WFQ (CBWFQ)



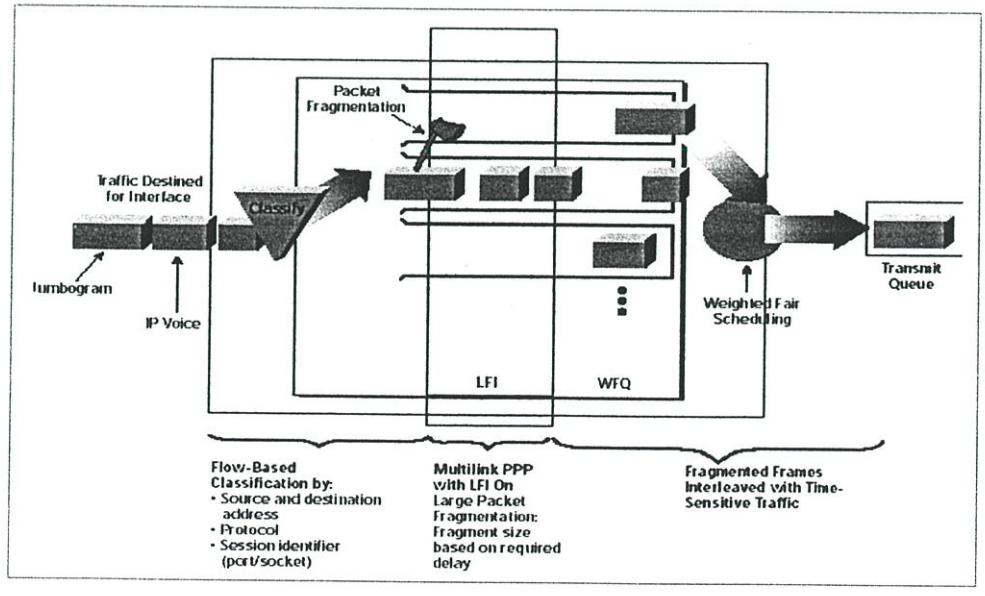
รูปที่ 3.7 รูปแบบการทำงานของ Weighted Fair Queue

- General Traffic Shaping (GTS): เป็นรูปแบบการจัดการแบนด์วิดท์โดยใช้การจัดการรูปแบบการจราจร (Traffic Shaping) จะควบคุมเส้นทางการส่งข้อมูลของทราฟฟิคโดยการอินเตอร์เฟสแบบพิเศษ GTS จะใช้รูปแบบของ Token Bucket [23] ซึ่งจะใช้สัญญาณโทเคนเป็นตัวเลือกแพคเก็ตตัวอย่างการทำงานแสดงดังรูปที่ 3.8 GTS ส่วนมากประยุกต์ใช้กับการจัดการในเลเยอร์ 2 เช่น Frame Relay, ATM, SMPS และ Ethernet



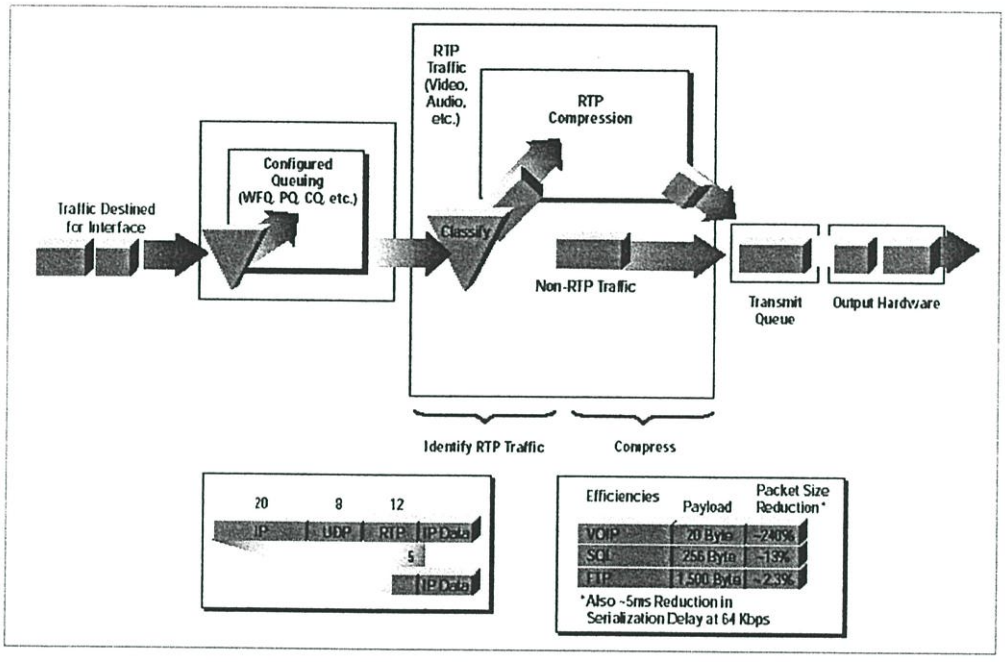
รูปที่ 3.8 รูปแบบการทำงานของ General Traffic Shaping

- Link Fragmentation and Interleaving (LFI): เป็นการควบคุมแบนด์วิดท์ในรูปแบบของการจัดการที่การลิงค์ (link) โดยที่ LFI จะแบ่งขนาดของแพคเก็ตของทราฟฟิคที่มีขนาดเล็กลง เนื่องจากขนาดของแพคเก็ตที่มีขนาดใหญ่เช่น Telnet, Voice on IP จะเกิดปัญหาเมื่อใช้กับเส้นทางการส่งข้อมูลที่มีความเร็วต่ำ เช่นอาจเกิด Jitter หรือ Latency ตัวอย่างการทำงานแสดงดังรูปที่ 3.9



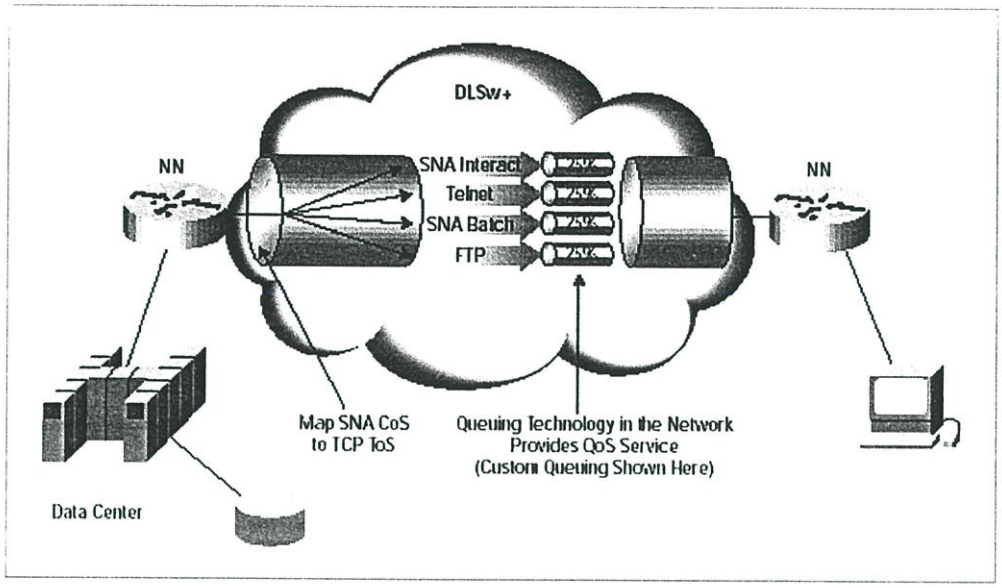
รูปที่ 3.9 รูปแบบการทำงานของ Link Fragmentation and Interleaving

- Real time Protocol Header Compression (RTP-HC): เป็นการควบคุมแบนด์วิดท์ในรูปแบบของการจัดการที่การลิงค์ (link) อีกแบบหนึ่ง ที่ใช้กับงานประยุกต์ในประเภทเรียลไทม์ เช่น Video, Audio โดยการจะบีบอัด (Compression) แพคเก็ตเดิมและเพิ่มหัว (Herder) เข้าไปก่อนที่จะมีการส่งไปในเส้นทางการส่งข้อมูล ปกติขนาดของหัวจะมีขนาด 40 ไบต์ การทำงานของ (RTP-HC) แสดงดังรูปที่ 3.10



รูปที่ 3.10 รูปแบบการทำงานของ Real time Protocol Header Compression

- SNA ToS: เป็นการควบคุมแบนด์วิดท์ในรูปแบบของการควบคุมสัญญาณโดยจะแบ่งกลุ่มของแพคเก็ตโดยใช้โพลิซี SNA ToS เป็นการทำงานร่วมกับ Data link Switching+(DLSw+) ซึ่งจะแปลง SNA ในรูปแบบการบริการ Class of Service (CoS) ให้อยู่ในรูปแบบของการบริการแบบ IP Difference โดยที่ DLSw+ จะใช้ TCP สี่คลาสสร้างเป็นการใช้งานแบบ SNA ToS สีกลุ่ม โดยที่แต่ละกลุ่มจะใช้หลักการของ IP Precedence เป็นตัวแบ่ง การทำงานแสดงดังรูปที่ 3.11



รูปที่ 3.11 รูปแบบการทำงานของ SNA ToS

นอกจากนี้ยังมีการจัดการแบนด์วิดท์โดยใช้หลักของ โพลิซีเช่น Policy Base Setting (PBR) [4] ซึ่งหลักการส่วนใหญ่จะจัดการที่ใช้หลักของ Difference Service โดยการใช้ IP Precedence จะเห็นว่าหลักการจัดการแบนด์วิดท์ต่าง ๆ ที่กล่าวมานั้น โดยทั่วไปเป็นเครื่องมือในการควบคุมคุณภาพการให้บริการในระดับสัญญาณ (Signaling) และใช้หลักการของไพรออริตี้ โดยไม่มีการจัดการแบบการจองทรัพยากร (RSVP) หรือใช้โพลีซีและจะเห็นว่าในการจัดการนั้นจะจัดการไปที่ระดับแพคเก็ตหรือเซล (cell) ของงานประยุกต์ไม่ได้จัดการที่ตัวงานประยุกต์

ในวิทยานิพนธ์นี้จะเสนอวิธีการควบคุมมาให้บริการรูปแบบใหม่ โดยไม่ได้มองลึกลงไปถึงระดับแพ็คเก็ตหรือเซลในการส่งข้อมูล แต่จะมองไปที่งานประยุกต์และความต้องการแบนด์วิดท์ของแต่ละงานว่าต้องการแบนด์วิดท์ในการส่งข้อมูลเท่าไหร่ การทำงานสามารถทำได้ง่าย ๆ โดยการคำนวณเหมือนการดาวน์โหลดข้อมูลในอินเตอร์เน็ตทั่ว ๆ ไป ซึ่งจะเห็นได้ว่าสามารถนำมาประยุกต์ใช้กับระบบกริด เพราะว่าระบบกริดนั้นมีหลากหลายเทคโนโลยีและหลายโปรโตคอลทำให้เป็นเรื่องยากในการทำการควบคุมการให้บริการในเลเยอร์ต่ำ แต่ในงานวิจัยนี้จะไม่มองไปที่โปรโตคอลเหล่านั้นจะมองไปที่ตัวงานประยุกต์และเส้นทางในการส่งข้อมูล

3.5 การควบคุมคุณภาพการให้บริการบนระบบกริด

ระบบกริดนั้นเป็นระบบที่มีการประมวลผลต่างกับระบบอื่นในเครือข่ายทั่วไป โดยที่ระบบกริดนั้นมีการใช้ทรัพยากรในต่างสถานที่กัน ดังนั้นจะต้องมีกลไกในการจัดการ เช่น ค้นหา การส่ง

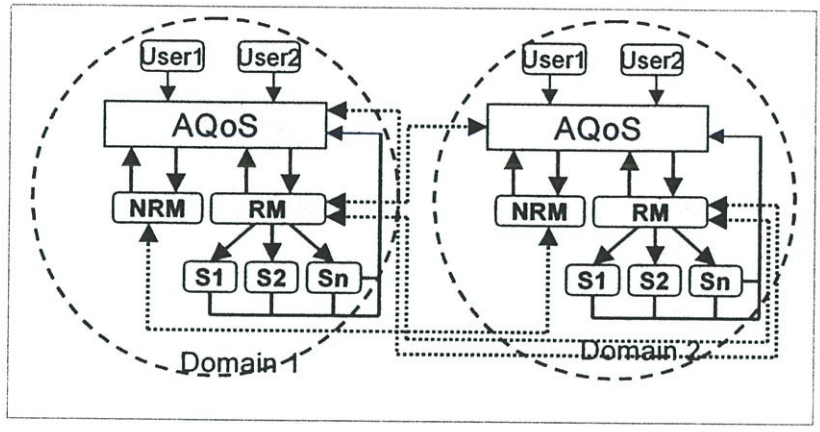
ข้อมูลที่ดี จึงสามารถที่จะใช้รองรับการทำงานระบบทีมีขนาดของข้อมูลมากและส่งผ่านในระบบตลอดเวลาได้ ในขณะเดียวกันกลไกในการควบคุมคุณภาพการให้บริการต้องมีคุณสมบัติที่ดีพอสำหรับการใช้กับระบบ คุณสมบัติโดยทั่วไปของการควบคุมคุณภาพการให้บริการไปที่ต้องนำมาพิจารณาเมื่อใช้กับการประมวลผลแบบกริดมีดังนี้

- Heterogeneous Flow เพื่อรองรับการทำงานของข้อมูล ที่มีจำนวนมหาศาลที่ส่งไปมาในระบบของกริด
- High bandwidth flow เพื่อรองรับกับงานประยุกต์ที่ต้องใช้แบนด์วิดท์สูงเช่นงาน Videoconference, การถ่ายทอดสดโทรทัศน์บนอินเตอร์เน็ตเป็นต้น
- เป็นการควบคุมคุณภาพการให้บริการแบบจุดต่อจุด(End-to-end) ซึ่งเหมาะสมในการทำการควบคุมคุณภาพการให้บริการที่ระดับงานประยุกต์
- เป็นการควบคุมการทำงานในระดับงานประยุกต์(Application level Control)
- เป็น Advance reservation เพื่อรองรับกับงานแอพพลิเคชั่นบางอย่างที่ต่องการเนื้อที่ในการเก็บข้อมูลสูงเช่นงานด้านการคำนวณทางวิทยาศาสตร์เป็ต้น

จากที่กล่าวมาทั้งหมดเป็นความต้องการที่จะใช้งานของทรัพยากรในระบบกริด ดังนั้นการควบคุมคุณภาพการให้บริการ สำหรับการจัดการบนระบบกริดต้องสามารถรองรับคุณสมบัติดังกล่าวได้ GARA (Globus Architecture for Reservation and Allocation) [29] เป็นงานวิจัยหนึ่งที่วิจัยเกี่ยวกับการควบคุมคุณภาพการให้บริการบนระบบกริดซึ่งสามารถรองรับคุณสมบัติที่กล่าวมาข้างบนได้ GARA ได้เสนอโครงสร้างในการควบคุมคุณภาพการให้บริการที่เรียกว่า G-QoSM ดังรูปที่ 3.12 ออกแบบมาเพื่อให้บริการในเรื่องของการสืบค้น (Discovery Services) บนพื้นฐานของการควบคุมการให้บริการโดยการรับประกันการให้บริการระดับแอพพลิเคชั่นมิดเดิลแวร์ (Middleware) และระดับเครือข่าย (Network) จัดการในเรื่องของ SLA และสุดท้ายคือการจัดการทรัพยากร โครงสร้างของ G-QoSM ประกอบไปด้วยสามส่วนคือ

- A-QoS เป็นตัวบริการโบรเกอร์ (Broker) เป็นตัวหลักของการ่าในการให้บริการระหว่างไคลอัลและการให้บริการเพื่อกำหนดรายละเอียดความต้องการไปให้ส่วนอื่น
- RM (Middleware resource Manager) เป็นส่วนในการจัดการทรัพยากรตามหลักการของ Globus Resource Allocation (GRAM) โดยใช้ UDDI เป็นตัวจัดการ RM จะให้รายละเอียดเกี่ยวกับคุณสมบัติของการให้บริการ

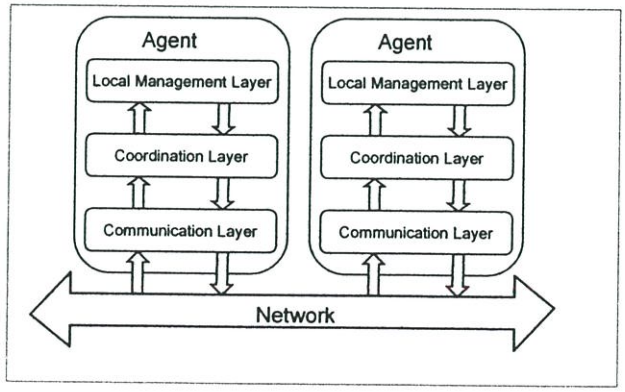
- NRM (Network Resource Manager) เป็นแบนด์วิดท์โบรเกอร์สำหรับจัดการพารามิเตอร์ที่กำหนดมาจาก RM อีกที



รูปที่ 3.12 โครงสร้างสถาปัตยกรรมของ G-QoSM

ในการควบคุมคุณภาพการให้บริการและการจัดการทรัพยากร ปัจจุบันได้มีการประยุกต์ใช้ระบบซอฟท์แวร์เอเจนต์ [30] เข้ามาช่วยซึ่งเอเจนต์มีทั้งแบบอยู่กับที่และแบบโมบาย มีหน้าที่ในการตรวจสอบและค้นหาทรัพยากรของระบบกริด [31,32] ที่เราต้องการและนอกจากนี้จะสามารถจัดการงานบางอย่างได้ด้วยตัวเองเอเจนต์ที่นำมาใช้ในระบบกริดประกอบด้วย

- Agile Architecture and Autonomous Agent System (A4): เป็นเอเจนต์ที่ให้บริการและการสืบค้นเอเจนต์จะเชื่อมต่อกันแบบลำดับชั้น (Hierarchy) และสามารถจะรับรู้สถานะของเอเจนต์ใกล้เคียงเพื่อประโยชน์ในการสืบค้นและบริการ โครงสร้างของเอเจนต์ประกอบด้วยสามส่วนดังรูปที่ 3.13



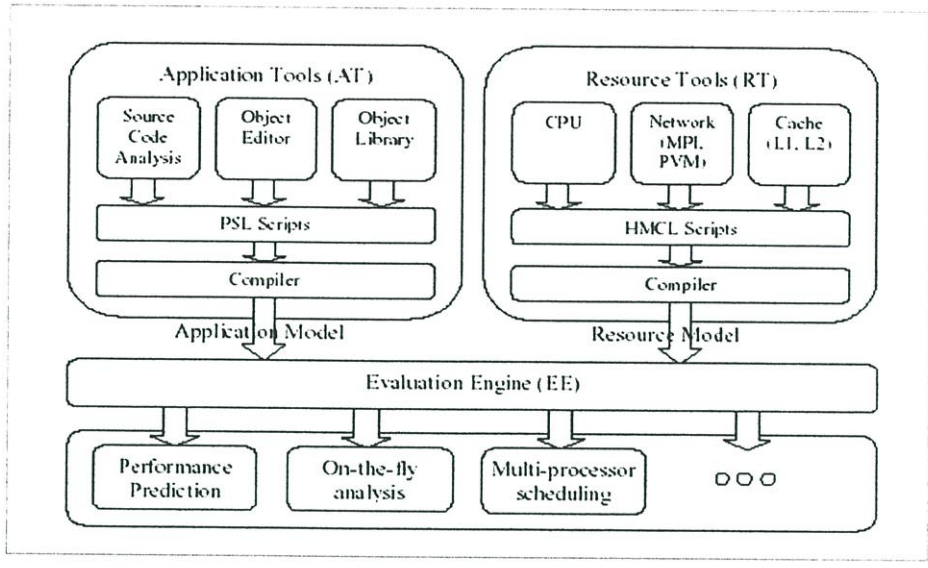
รูปที่ 3.13 โครงสร้างสถาปัตยกรรมของ A4 เอเจนต์

1. Communication layer: ติดต่อสื่อสารระหว่างเอเจนต์อื่นโดยใช้ Common data Model และโปรโตคอลการสื่อสาร ใช้ ACL ติดต่อแลกเปลี่ยนการเรียนรู้กับเอเจนต์อื่น
2. Coordinate layer: การทำงานหรือการกระทำของเอเจนต์หลังจากที่ได้เรียนรู้แล้ว
3. Local Management layer: เป็นฟังก์ชั่นและการบริการแบบโลคอลซึ่งจะให้บริการที่โลคอลตามความต้องการของ Coordinate layer

- Titan Local Resource Manager: เป็นเอเจนต์ที่ใช้จัดการงาน (work load) ในระดับโลคอล ซึ่งจะเลือกทรัพยากรที่เป็นไปได้โดยใช้การทำงานแบบไดนามิค Titan จะใช้ฮิวริสติกอัลกอริทึมเป็นพื้นฐานในการค้นหาข้อมูลในแต่ละโลคอลซึ่งจุดประสงค์ของอัลกอริทึมนี้ เพื่อหาค่าเวลาโปรเซสเซอร์ที่น้อยที่สุดเพื่อประมวลผล ในการใช้งานนั้น Titan เลือกและประมาณค่าของโปรเซสเซอร์และเลือกหรือรวมการทำงานที่ดีที่สุดเพื่อประมวลผล

- Performance Analysis and Characteristic Environment (PACE): เป็นส่วนประกอบของระบบเอเจนต์ที่ใช้ในการคาดคะเน (Prediction) ที่มีประสิทธิภาพ PACE จะประกอบด้วยโมเดลในการกำหนด (definition) สร้างโมเดลและวิเคราะห์ประเมินสำหรับซอฟแวร์ประยุกต์ทรัพยากรและการใช้งาน (Mapping) และตรวจสอบระบบฮาร์ดแวร์ ส่วนประกอบของ PACE ประกอบด้วยสามส่วนคือ

1. Application Tools: ใช้วิเคราะห์คุณสมบัติของแอพพลิเคชั่นแบบขนานโดย PSL [30]
2. Resource Tools: ใช้ HMCL [30] เป็นตัวกำหนดการประมวลผลสภาวะแวดล้อมในรูปแบบของส่วนประกอบของโมเดลที่มีประสิทธิภาพ
3. Evaluation Engine: จะเป็นส่วนที่ใช้เรียก (Core) ซึ่งจะใช้โมเดลที่มีประสิทธิภาพในการสร้างผลลัพธ์ PACE ใช้สำหรับการทำสเชดูลิ่ง (Scheduling) แบบไดนามิคมัลติโปรเซสเซอร์



รูปที่ 3.14 โครงสร้างสถาปัตยกรรมของ PACE เอเจนต์

ระบบมัลติ-เอเจนต์ในวิทยานิพนธ์เล่มนี้ จะทำงานในลักษณะของโลคอล คือจะตรวจสอบทรัพยากรต่าง ๆ ของระบบกริดที่โลคอล และรายงานไปยังศูนย์ควบคุมส่วนกลางเพื่อลงทะเบียนในระยะเริ่มแรกหรือกรณีที่มีการปรับปรุงสถานะ และเมื่อระบบกริดที่โลคอลต้องการส่งข้อมูลหรือจัดการทรัพยากรจะรายงานพารามิเตอร์ต่าง ๆ ไปยังส่วนกลาง และส่วนกลางจะนำเอาค่าพารามิเตอร์เหล่านั้นไปประกอบในการควบคุมคุณภาพการให้บริการ

บทที่ 4
การควบคุมคุณภาพการให้บริการระบบกริด โดยใช้ระบบมัลติเอเจนต์

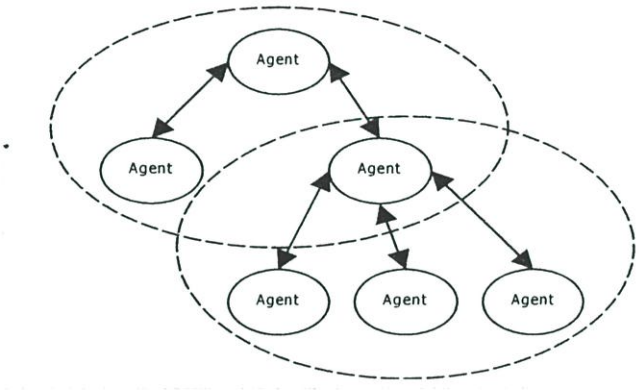
4.1 บทนำ

จากบทที่ผ่านมาจะเห็นว่ารูปแบบการให้บริการของการประมวลผลแบบกริด เป็นการทำงานแบบระบบกระจายขนาดใหญ่บนองค์กรเสมือนหลาย ๆ องค์กรเสมือน โดยจุดประสงค์ของระบบเพื่อใช้ทรัพยากรในการประมวลผลร่วมกัน ตามโครงสร้างของระบบเป็นเรื่องยากที่จะควบคุมการใช้ทรัพยากรต่าง ๆ ในระบบไม่ให้เกิดปัญหาในการใช้งาน และให้เกิดประโยชน์สูงสุดในการบริการทรัพยากรของกริด ในวิทยานิพนธ์นี้ได้นำเสนอการควบคุมคุณภาพการให้บริการของระบบ โดยประยุกต์เทคโนโลยีของระบบซอฟท์แวร์เอเจนต์ เข้ามาจัดการและควบคุมทรัพยากรของระบบกริด โดยรูปแบบของการควบคุมการบริการให้บริการ จะเป็นการควบคุมจากส่วนกลางเพื่อให้ง่ายต่อการจัดการระบบขนาดใหญ่ โดยการควบคุมคุณภาพการให้บริการจะเป็นการตรวจสอบแบนด์วิดท์ของเส้นทางการส่งของข้อมูล(Path Flow) เปรียบเทียบกับขนาดแบนด์วิดท์ของงานประยุกต์ที่จะส่งในขณะนั้นว่ามีเพียงพอหรือไม่ นอกจากนี้การส่งข้อมูลแต่ละครั้งนั้นจะยึดตามหลักการของ การให้บริการแบบลำดับความสำคัญโดยปกติจะเป็นการให้บริการกับงานที่มีความสำคัญสูงสุดก่อน และรอจนกว่างานประยุกต์นั้นถูกจะส่งเสร็จสมบูรณ์ก่อน ซึ่งจะไม่เป็นประโยชน์เมื่อนำมาประยุกต์ใช้กับโครงสร้างของเอเจนต์แบบอีพ เนื่องจากจะมีทรัพยากรบางส่วนในโครงสร้างนั้นจะไม่ได้ถูกนำมาใช้งานเพื่อให้เกิดประโยชน์สูงสุดในการใช้งาน วิทยานิพนธ์นี้ได้นำเสนอการทำงานแบบหลาย ๆ ลำดับความสำคัญพร้อม ๆ กันโดยได้ออกแบบอัลกอริทึมแบบมัลติโปรเซสส์ไพรออริตี้เข้ามาจัดการในโครงสร้างแบบฮีพ

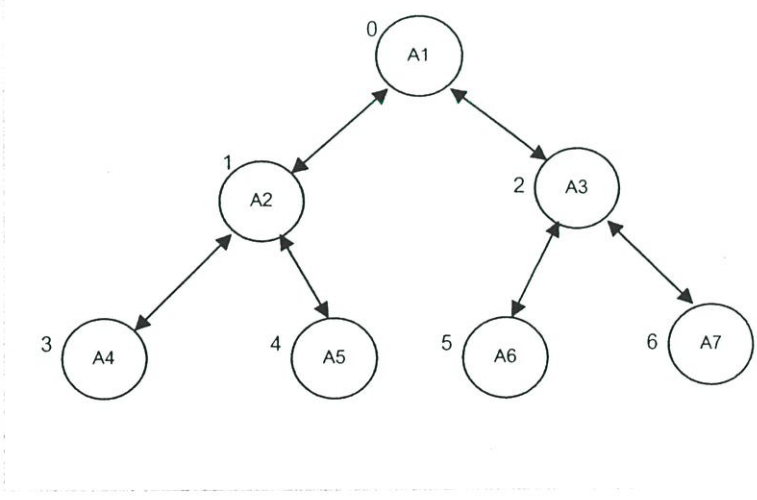
ในบทนี้ในส่วนแรกจะอธิบายถึงหลักการพื้นฐานของโครงสร้างแบบฮีพ และการประยุกต์ใช้งานโดยการนำมาใช้แทนโครงสร้างแบบทรี ในส่วนที่สองจะอธิบายถึงแนวคิดการทำงานเบื้องต้นของอัลกอริทึมมัลติโปรเซสส์ไพรออริตี้ที่ใช้ในการควบคุมคุณภาพการให้บริการ ในส่วนสุดท้ายจะเป็นการประยุกต์ใช้ระบบมัลติเอเจนต์ สำหรับการประยุกต์ใช้ในการควบคุมทรัพยากรของการประมวลผลแบบกริด จะอธิบายถึง ส่วนประกอบและหน้าที่การทำงานของเอเจนต์ การการจัดโครงสร้างของระบบมัลติเอเจนต์ การทำงานของระบบมัลติเอเจนต์ร่วมกับตัวควบคุมส่วนกลาง และการประยุกต์ใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมร่วมกับระบบมัลติเอเจนต์

4.2 การประยุกต์ใช้โครงสร้างข้อมูลแบบฮีพ

จากงานวิจัยที่ [1] เป็นการประยุกต์ใช้ระบบเอเจนต์เข้ามาช่วยในการให้บริการของระบบกริด ซึ่งเอเจนต์นั้นจัดโครงสร้างแบบทรีแสดงดังรูปที่ 4.1 โครงสร้างแบบทรีจะมีจำนวนโหนดหรือบัพลูกได้มากกว่าสอง โครงสร้างนี้สามารถพัฒนาไปสู่โครงสร้างการทำงานแบบฮีพได้เพื่อทำให้สะดวกในการประยุกต์ใช้งานและเหมาะสมกับการจัดการข้อมูลระบบขนาดใหญ่ โครงสร้างแบบฮีพเป็นการนำเอาโครงสร้างข้อมูลแบบไบนารี่ทรี (Binary tree) มาประยุกต์ใช้งานสำหรับการค้นหาข้อมูล ซึ่งจะเพิ่มคีย์ให้กับบัพและใช้ร่วมกับคิวลำดับเพื่อประโยชน์ในการเพิ่มประสิทธิภาพในการค้นหาข้อมูล



รูปที่ 4.1 โครงสร้างแบบทรีของระบบมัลติเอเจนต์



รูปที่ 4.2 โครงสร้างแบบฮีพของระบบมัลติเอเจนต์

4.2.1 หลักพื้นฐานของโครงสร้างข้อมูลแบบฮีพ

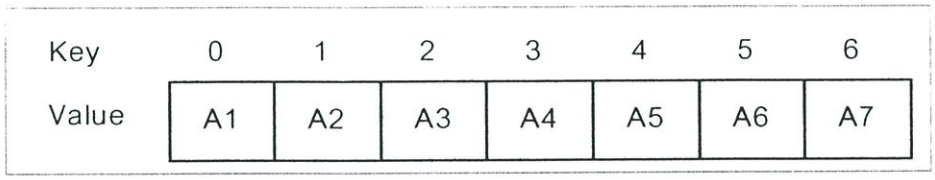
โครงสร้างข้อมูลแบบฮีพ[2, 33] พัฒนาขึ้นมาในปี 1964 โดยเจดับเบิ้ลยูเจวิลเลียม (J.W.J Williams) เป็นการจัดโครงสร้างแบบไบนารี่ทรีเกือบสมบูรณ์ (Almost Complete binary tree) โครงสร้างประกอบด้วยลำดับของคีย์ (Key) และโหนดหรือบัพซึ่งจะเป็นส่วนของข้อมูล ถ้า v คือโหนดในโครงสร้างของฮีพ จะได้ว่า key(v) >= Key(Parent(v)) คือค่าคีย์ของโหนดพ่อ (Parent Node) จะมีค่าน้อยกว่าค่าคีย์ของโหนดลูก (Children Node) เสมอแสดงดังรูปที่ 4.2 ซึ่งเป็นโครงสร้างของฮีพลำดับจากน้อยไปมาก (Min Heap) จำนวน 7 โหนดที่มีคีย์เริ่มจาก 0 ถึง 6 และมีระดับ (Depth) เท่ากับ 3 ในการสร้างโครงสร้างแบบฮีพจะมีตัวดำเนินงาน (Operation) เช่น การสร้างฮีพ การลบฮีพ การจัดเรียงข้อมูลในฮีพเป็นต้น ตัวดำเนินงานพื้นฐานแสดงตามตารางที่ 4.1

ตารางที่ 4.1 โอเปอร์เรชั่นที่ใช้จัดการโครงสร้างของฮีพ

ชื่อการทำงาน	รายละเอียดการทำงาน
Heapify	การปรับปรุงโครงสร้างของฮีพ
Shift-Down	การปรับปรุงข้อมูลของฮีพจากบนลงล่าง
Shift-Up	การปรับปรุงข้อมูลฮีพจากล่างขึ้นบน
Insert Heap	การเพิ่มโหนดใหม่เข้าไปในโครงสร้างของฮีพ
Remove Heap	การลบโหนดออกจากโครงสร้างของฮีพ
Heap Sort	การจัดเรียงข้อมูลภายในคิว

4.2.2 การประยุกต์ใช้งานฮีพสำหรับการบริการแบบลำดับความสำคัญ

การประยุกต์ใช้งานโครงสร้างแบบฮีพร่วมกับไพรออริตี้คิว(Priority Queue) มีหลักในการกำหนดลำดับความสำคัญอยู่สองหลักการคือ หนึ่งกำหนดจากบนลงล่าง (ในแนวดิ่ง) โหนดที่มีระดับ (Depth) สูงหรือค่าของคีย์ต่ำจะมีลำดับความสำคัญมากกว่าโหนดที่อยู่ในระดับต่ำหรือมีค่าของคีย์มาก สองกำหนดจากซ้ายไปขวา (ในแนวราบ) คือโหนดที่อยู่ด้านซ้ายจะมีลำดับความสำคัญสูงกว่าโหนดที่อยู่ด้านขวามือ จากรูปที่ 4.2 โหนด 0 จะมีลำดับความสำคัญสูงที่สุด โหนด 3 จะมีลำดับความสำคัญสูงกว่าโหนด 4, 5, 6 และโหนดที่มีลำดับความสำคัญต่ำสุดคือโหนด 6 สำหรับการประยุกต์โครงสร้างของฮีพกับไพรออริตี้คิวแสดงดังรูปที่ 4.3 ในการประยุกต์ใช้ในงานวิจัยนี้จะกำค่าคีย์ตามลำดับการสร้างเอเจนต์



รูปที่ 4.3 การสร้างไพรออริตี้คิวจากโครงสร้างของฮีพ

จากรูปแสดงการแทนไพรออริตี้คิวจากโครงสร้างของฮีพ จะประกอบด้วยข้อมูลสองส่วนคือ ส่วนที่เป็นคีย์ซึ่งอยู่ในส่วนบน (0,1,2,3,4,5,6) และส่วนที่เป็นข้อมูล (A1, A2, A3, A4, A5, A6, A7) การจัดลำดับความสำคัญจะจัดตามหลักการของฮีพคือจากบนลงล่างและจากซ้ายไปขวา

4.3 แนวคิดของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม

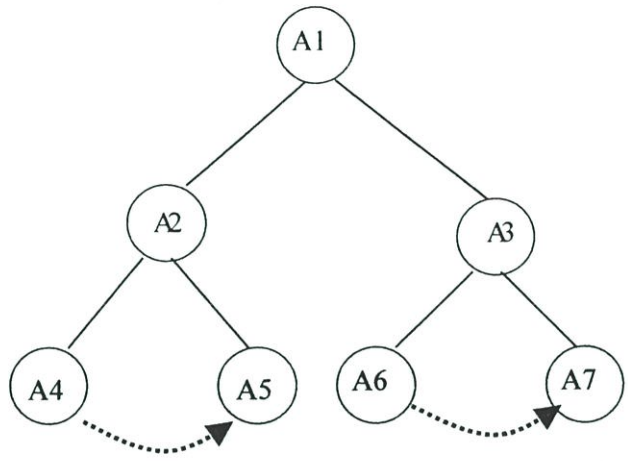
การทำงานของการให้บริการลำดับความสำคัญ เป็นการทำงานตามลำดับความสำคัญของข้อมูล โดยงานที่มีความสำคัญสูงสุดและมีอายุมากที่สุดจะได้รับการบริการหรือจัดการให้เสร็จก่อนแล้วค่อยกระทำกับงานที่มีความสำคัญรองลงมา แต่ในการประมวลผลหรือการจัดการงานในโครงสร้างของฮีพดูจะไม่เหมาะสมนัก เพราะระบบไม่จำเป็นต้องรอให้งานที่มีความสำคัญสูงสุดทำงานเสร็จก่อนค่อยเลือกงานต่อไป แต่ระบบสามารถเลือกจัดการได้เลยถ้ามีทรัพยากรที่เพียงพอ เมื่อนำหลักการของการให้บริการแบบลำดับความสำคัญมาประยุกต์ใช้กับโครงสร้างของฮีพ งานวิจัยนี้ได้เสนอหลักการทำงานแบบใหม่โดยเสนออัลกอริทึมมัลติโปรเซสส์ไพรออริตี้ เข้ามาจัดการแทนการให้บริการแบบลำดับความสำคัญฮีพตามหลักการที่กล่าวมา

มัลติโปรเซสส์ไพรออริตี้อัลกอริทึม (Multi-Process Priority Algorithm) เป็นหลักการที่พัฒนาเพิ่มเติมจากการประยุกต์ใช้งานของโครงสร้างแบบฮีพ และการให้บริการแบบลำดับความสำคัญโดยอัลกอริทึมของฮีพบนคิวลำดับ ซึ่งการจัดการเบื้องต้นจะใช้อัลกอริทึมพื้นฐานของฮีพทั้งหมด อัลกอริทึมนี้เหมาะที่จะนำมาประยุกต์กับการทำงานของระบบกริด เพราะระบบกริดนั้นสามารถแชร์ทรัพยากรได้ทุกส่วน เช่น ซีพียู หน่วยความจำ ซอฟท์แวร์รวมถึงเครือข่ายคอมพิวเตอร์ แนวคิดการทำงานของระบบนั้นแบ่งออกเป็นสองข้อดังนี้

4.3.1 แนวคิดการทำงานแบบขนานในโครงสร้างแบบฮีพ

แนวคิดการทำงานแบบขนานอธิบายได้ดังนี้ ถ้ามีข้อมูลหลาย ๆ ชุดภายใต้โครงสร้างของฮีพ โดยที่ข้อมูลแต่ละชุดนั้นอยู่ต่างเส้นทางการส่งข้อมูล หรือเส้นทางเดียวกันและมีทรัพยากรในการส่งเพียงพอระบบยอมให้ส่งข้อมูลได้ ตัวอย่างที่เห็นได้ชัด เช่น สมมุติกรมการปกครองต้องการนำ

ระบบกริดมาใช้กับระบบทะเบียนประวัติของประชากรทั่วประเทศโดยมีองค์กรเสมือนครอบคลุมศูนย์คอมพิวเตอร์จังหวัดทั่วประเทศ โดยมีศูนย์คอมพิวเตอร์กลางที่กรุงเทพคอยจัดการระบบกริดอยู่ ระบบสามารถส่งข้อมูลพร้อมกันระหว่างเชียงใหม่-เชียงรายและนครราชสีมา-อุบลราชธานีได้ หรือแสดงดังรูปที่ 4.4 โครงสร้างของเอเจนต์ที่จัดโครงสร้างแบบฮีพ และเอเจนต์แต่ละตัวจะเชื่อมต่อไปยังคอมพิวเตอร์ใช้ควบคุมทรัพยากรของกริด ซึ่งในรูปที่ 4.4 สามารถส่งข้อมูลจากเอเจนต์ A4 ไป A5 พร้อมกับ และ A6 ไป A7 ได้ในเวลาเดียวกัน

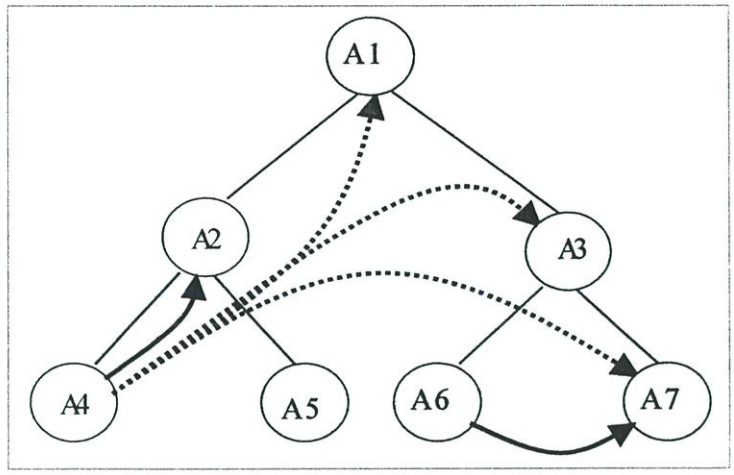


รูปที่ 4.4 แนวคิดการทำงานแบบขนาน

4.3.2 แนวคิดการฝากข้อมูลในโครงสร้างแบบฮีพ

แนวคิดการฝากข้อมูลมีหลักการคือระบบการจัดการยอมให้ส่งข้อมูลของกริดไปฝากไว้กับกริดที่มีทรัพยากรที่พอเพียงในเส้นทางการส่งข้อมูลเดียวกัน เมื่อได้สิทธิในการส่งในครั้งต่อไปและมีทรัพยากรเพียงพอค่อยส่งข้อมูลไปให้กริดที่ร้องขอมา จากรูปที่ 4.5 จะเห็นว่าระบบต้องการจะส่งข้อมูลจาก A4 ไปยัง A7 แต่ขณะนั้น A7 ยังมีช่องทางการส่งข้อมูลไม่พอซึ่งอาจจะทำการรับหรือส่งงานอยู่กับเอเจนต์อื่นอยู่ A4 จะพิจารณาเอเจนต์ตัวที่ใกล้ที่สุดในเส้นทางก่อนคือ A3 ในกรณีที่ A3 ว่างหรือมีแบนด์วิดท์เพียงพอ A4 จะส่งข้อมูลไปเก็บไว้ที่ A3 แต่ถ้าไม่สามารถส่งไปที่ A3 ได้ ระบบก็จะตรวจสอบเอเจนต์ถัดจาก A3 คือ A1 และ A2 ไปเรื่อย ๆ จนกว่าจะมีเอเจนต์ที่สามารถส่งข้อมูลได้ ถ้าไม่สามารถส่งข้อมูลได้คือในเส้นทางมีแบนด์วิดท์ไม่เพียงพอ การส่งข้อมูลในครั้งนั้นก็จะถูกยกเลิกไป และระบบก็จะไปจัดการเอเจนต์ที่ลำดับความสำคัญรองลงไป

จากแนวคิดที่กล่าวมาข้างบนอาศัยหลังการพื้นฐานสองหลักการคือ หลักการให้บริการแบบลำดับความสำคัญตามรูปแบบของฮีพและหลักการจัดแบนด์วิดท์ตามหลักการของ Sink Link Capacity การนำแนวคิดทั้งสองไปประยุกต์ใช้งานจะต้องพิจารณาร่วมกับจัดการแบนด์วิดท์



รูปที่ 4.5 แนวคิดการทำงานการฝากข้อมูล

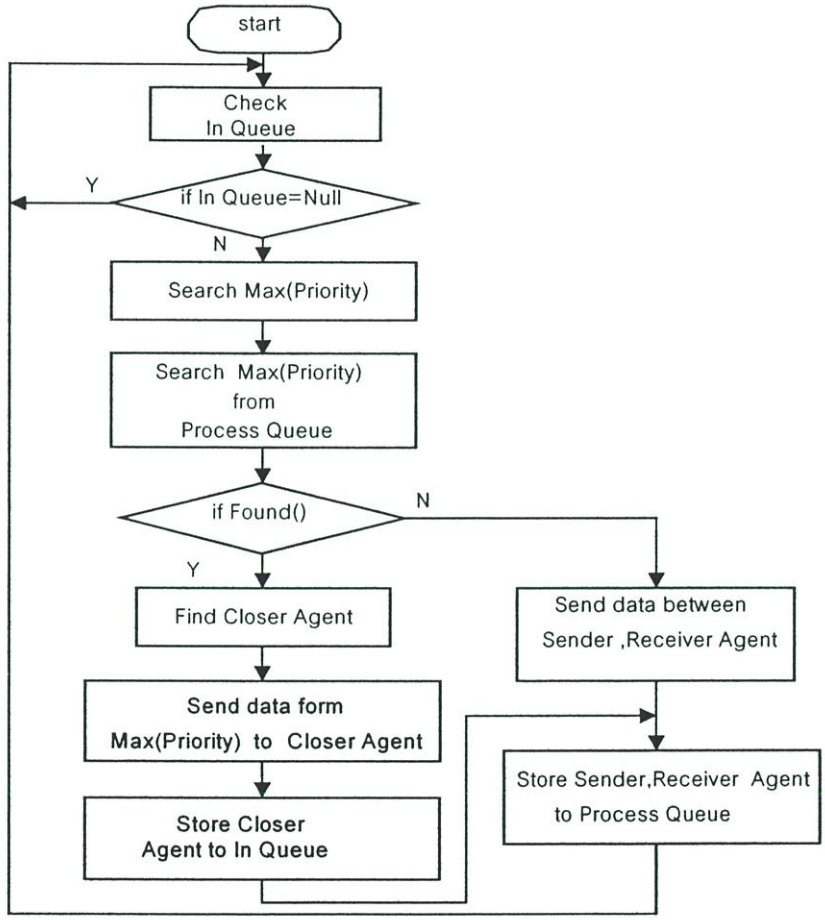
ตารางที่ 4.2 เส้นทาง(Path Flow) การส่งข้อมูลของแต่ละเอเจนต์

Flow	Path Agent Node						
12	A1	A2					
13	A1	A3					
14	A1	A2	A4				
15	A1	A2	A5				
16	A1	A3	A6				
17	A1	A3	A7				
● ●							
75	A7	A3	A1	A2	A4		
76	A7	A3	A6				

จากแนวคิดทั้งสองแนวคิดนี้เมื่อจะนำมาประยุกต์ใช้กับระบบใดระบบหนึ่งนั้นจะมีข้อจำกัดอยู่หลายอย่างซึ่งระบบนั้นต้องรองรับการทำงาน ที่สำคัญคือระบบต้องสามารถใช้ทรัพยากรร่วมกันได้ เช่นสามารถใช้หน่วยความจำทั้งหน่วยจำหลักและรองร่วมกันได้ สามารถใช้ซอฟท์แวร์ร่วมกันได้ ซึ่งจะเห็นว่าระบบการประมวลผลแบบกริดเป็นระบบที่สามารถรองรับแนวคิดการทำงานแบบนี้ได้ จาก

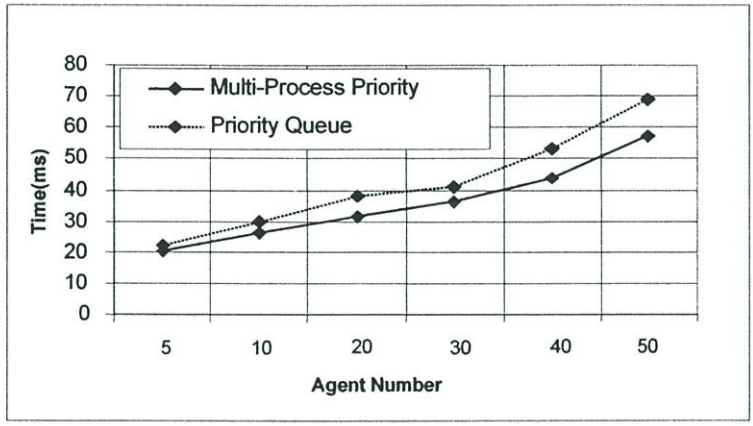
หลักการทั้งสองข้อนี้เมื่อนำมาใช้กับโครงสร้างของฮีพนั้นจะต้องกำหนดเส้นทางการส่งข้อมูล (Path Flow) ซึ่งเป็นเส้นทางการส่งข้อมูลของแต่ละกริดดังตารางที่ 4.2

จากหลักการของมัลติโปรเซสส์นี้ไดรับการตีพิมพ์โดย [34, 35] ซึ่งได้ทำการทดลองโดยการทดสอบเปรียบเทียบกับการให้บริการแบบลำดับความสำคัญ (Priority Queue: PQ) และอัลกอริทึมมัลติโปรเซสส์ไพรออริตี้(MPQ) โดยเขียนโปรแกรมทดสอบการทำงานตามโฟลว์ชาร์ท (Flow chart) ที่แสดงในรูปที่ 4.6



รูปที่ 4.6 โฟลว์ชาร์ทการทำงานของมัลติโปรเซสส์อัลกอริทึม

วิธีการทดสอบในเบื้องต้นจะเป็นการทดลองส่งข้อมูลขนาด 1 Kbytes ระหว่างเอเจนต์แบบพบกันหมด ในจำนวน 50 เอเจนต์ และใช้เทคนิคการเขียนโปรแกรมแบบเทรดจะได้ผลการทดลองตามรูปที่ 4.7 ซึ่งจะเห็นว่าเวลาที่ได้จะให้ผลไม่ต่างกันมากนักเพราะว่าไม่ได้วิธีการพิจารณาแบนด์วิดท์เข้ามาเกี่ยวข้อง



รูปที่ 4.7 กราฟเปรียบเทียบผลการทดสอบระหว่างคิวอัลกอริทึมและมัลติโปรเซสส์อัลกอริทึม

4.4 ส่วนประกอบและการทำงานของระบบมัลติเอเจนต์

จากหัวข้อที่ 4.3 เป็นการนำเสนอแนวคิดเบื้องต้นสำหรับการประยุกต์ใช้เท่านั้น หลักการของมัลติโปรเซสส์อัลกอริทึมที่อธิบายมาข้างต้นสามารถที่จะนำมาประยุกต์ใช้กับระบบกริดได้ เนื่องจากระบบกริดนั้นมีคุณสมบัติที่เหมาะสมคือ สามารถขอใช้ทรัพยากรของกริดที่สถานี (Node) อื่นได้ ซึ่งการขอใช้ทรัพยากรของกริดนี้ต่างจากการใช้ทรัพยากรของระบบอื่น โดยที่สามารถขอใช้ทรัพยากรได้จำนวนมากเพื่อให้เพียงพอกับข้อมูลขนาดใหญ่ ในขณะที่ระบบอื่นอาจจะขอใช้เพียงเล็กน้อยเพื่อใช้เป็นหน่วยความจำชั่วคราวในการเก็บแพกเก็ต (Packet) ของข้อมูล การจัดการทรัพยากรในส่วนของกริดที่นำเสนอ ระบบจะมีกลไกในการตรวจสอบทรัพยากรของกริดที่ขั้นตอนการลงทะเบียนเริ่มต้นและมีการปรับปรุงทรัพยากรอยู่เสมอแบบออนไลน์

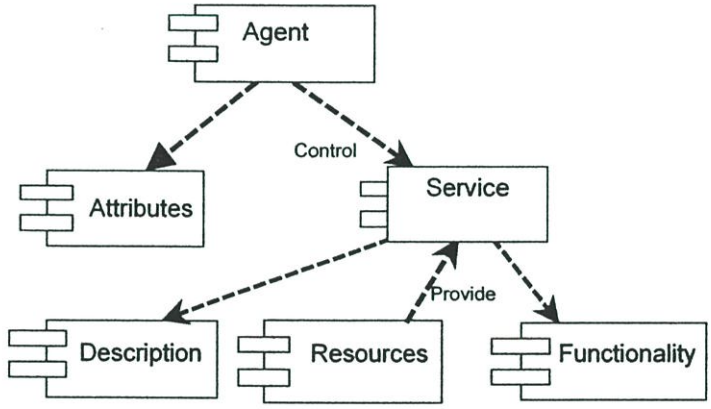
การควบคุมคุณภาพการให้บริการ โดยใช้มัลติโปรเซสส์อัลกอริทึมจะเน้นไปที่การจัดหาแบนด์วิดท์ที่มีอยู่จำกัดให้กับงานประยุกต์ในระบบของกริด โดยที่เป็นการจัดการข้อมูลในระดับงานประยุกต์คือมองไปที่ขนาดของงานประยุกต์ที่มีอยู่ โดยจะไม่ไปยุ่งกับโปรโตคอลในระดับล่าง แต่อย่างไรก็ตามถ้าจะนำไปประยุกต์ใช้กับงานจริงระบบก็จะต้องมีสิ่งที่สามารถบอกถึงความแตกต่างของแต่ละกริดโหนดได้บ้างเช่น เหมือนกับการแยกด้วยหมายเลข IP เป็นต้น

การทำงานของระบบเอเจนต์นั้นมีแนวคิดมาจากรูปแบบการทำงานของระกริด ซึ่งเป็นการทำงานตามสถาปัตยกรรมบริการแบบเปิด กริดจะใช้ GIS เป็นแหล่งข้อมูลของระบบเพื่อใช้ประโยชน์ในการสืบค้นหรือบริการอย่างอื่น โดยจะเก็บรายละเอียดของทรัพยากรตั้งแต่เริ่มลงทะเบียนหรือมีการเปลี่ยนแปลงบางอย่าง ในระบบมัลติเอเจนต์จะใช้ตัวควบคุมส่วนกลางเป็นตัวจัดเก็บข้อมูลของ

เอเจนต์และแอตทริบิวของกริด นอกจากจะเก็บรายละเอียดของเอเจนต์แล้วตัวควบคุมส่วนกลางยังใช้เป็นตัวสำคัญในการควบคุมคุณภาพการให้บริการ

4.4.1 องค์ประกอบของเอเจนต์

เอเจนต์จัดเป็นส่วนสำคัญหลักในการควบคุมคุณภาพการให้บริการ โดยเอเจนต์จะเป็นซอฟท์แวร์เล็ก ๆ ที่จะคอยควบคุมเซอร์วิส (Service) ซึ่งคือข้อมูลหรือการบริการของสถานีกริดที่ยอมให้สถานีอื่นใช้บริการได้ หน้าที่หลักของเอเจนต์คือจะรายงานค่าสถิติหรือพารามิเตอร์ของทรัพยากรไปยังส่วนกลาง ซึ่งอาจเป็นทรัพยากรในการใช้งานในตามรูปแบบของการค้นหาที่อยู่อย่างรวดเร็ว แต่ในที่นี้จะเน้นไปที่การควบคุมคุณภาพการให้บริการซึ่งเอเจนต์รายงานค่าพารามิเตอร์ที่ต้องใช้ในการทำการควบคุมไปยังส่วนกลาง ส่วนกลางจะนำค่าพารามิเตอร์เหล่านี้ไปตัดสินใจตามกรรมวิธีที่กำหนด องค์ประกอบหรือฟังก์ชั่นการทำงานของเอเจนต์แต่ละตัวแสดงดังรูปที่ 4.8



รูปที่ 4.8 องค์ประกอบและฟังก์ชั่นการให้บริการของเอเจนต์

การให้บริการของเอเจนต์ส่วนใหญ่ได้มาจาก [1] ซึ่งเอเจนต์ประกอบด้วยส่วนอธิบายการเชื่อมต่อ (Description) จะอธิบายวิธีการเชื่อมต่อของเอเจนต์ ส่วนอธิบายหน้าที่การทำงาน (Functionality) ในส่วนของแอตทริบิว (Attribute) จะเป็นส่วนที่อธิบายข้อมูลของเอเจนต์และส่วนที่ให้บริการเพื่อใช้ในการติดต่อสื่อสารในระบบและเป็นข้อมูลสำคัญในการควบคุมการให้บริการแอตทริบิวสำหรับการให้บริการแสดงดังตารางที่ 4.3

วิทยานิพนธ์นี้จะรับเอาการให้บริการของเอเจนต์บางส่วนจาก [1] เนื่องจากการควบคุมการให้บริการเป็นการจัดการจากส่วนกลาง ซึ่งไม่จำเป็นต้องมีการจัดการที่เอเจนต์มากนัก เช่น PST (Posting Service Table) [1] ซึ่งถือเป็นหัวใจหลักในการทำการค้นหาแบบรวดเร็วนั้นไม่ เหมาะที่จะ

นำมาใช้งานเนื่องจากจะเป็นการเพิ่มภาระให้เอเจนต์มากเกินไป ในที่นี้จะใช้แอตทริบิว (Attribute) เก็บค่าก่อนพารามิเตอร์ก่อนที่จะส่งต่อไปยังตัวควบคุมส่วนกลาง

ตารางที่ 4.3 ตัวอย่างแอตทริบิวของเอเจนต์ที่ใช้ในการควบคุมการให้บริการ

ลำดับ	แอตทริบิว	ความหมาย
1	Agent Name	ชื่อเอเจนต์
2	Agent key	เก็บค่า Key ของเอเจนต์
3	Parent Agent	เก็บค่า Key ของเอเจนต์ที่เป็นพ่อ
4	Left Agent	เก็บค่า Key ของเอเจนต์ที่เชื่อมต่อด้านซ้าย
5	Right Agent	เก็บค่า Key ของเอเจนต์ที่เชื่อมต่อด้านขวา
6	Resource list	เก็บรายละเอียดของทรัพยากรของกริด
7	Agent type	เก็บชนิดของเอเจนต์
8	Agent Dept	เก็บระดับของเอเจนต์
9	Link Capacity	เก็บแบนด์วิดท์ของกริด

จากตารางที่ 4.3 แอตทริบิวที่สำคัญที่ใช้สำหรับการควบคุมคุณภาพการให้บริการประกอบด้วย Agent Key จะเป็นค่าที่บอกลำดับความสำคัญของเอเจนต์แต่ละตัว Application list จะเก็บรายละเอียดของงานประยุกต์ โดยจะบอกถึงประเภทและขนาดของงาน พร้อมทั้งแบนด์วิดท์ต่ำสุดที่สามารถส่งข้อมูลได้อย่างไม่มีปัญหา โดยทุกครั้งที่กริดจะส่งข้อมูลกริดเว็บเซอร์วิสจะต้องบอกความต้องการมายังเอเจนต์ เอเจนต์มีโมดูลเล็กๆ ที่อยู่ในส่วนของ Functionality จะทำการวิเคราะห์งานที่กริดเว็บเซอร์วิสหรือกริดโหนดต้องการส่งและแจ้งรายละเอียดไปยังตัวควบคุมส่วนกลาง เพื่อเป็นข้อมูลสถิติในการควบคุมคุณภาพการให้บริการ

4.4.2 หน้าที่การทำงานของเอเจนต์

การทำงานพื้นฐานของเอเจนต์ในงานวิจัยนี้ยังต้องอ้างอิงตาม [1] คือการลงทะเบียนเอเจนต์การสื่อสารระหว่างเอเจนต์และการให้บริการงานส่วนอื่นที่สำคัญ แต่จะศึกษาเพิ่มเติมสำหรับการนำองค์ประกอบและส่วนบริการของเอเจนต์ที่กล่าวมาเพื่อการควบคุมคุณภาพการให้บริการ ซึ่งหน้าที่หลักของเอเจนต์ คือการตรวจสอบและกำหนดขนาดของแบนด์วิดท์ของงานประยุกต์แต่ละแบบโดยต้องกำหนดว่างานประยุกต์จะส่งข้อมูลด้วยความเร็วค่าใดจึงไม่มีผลต่อการทำงาน เมื่อประเมินผลได้

แล้ว ส่วนบริการของเอเจนต์จะส่งคำร้องขอส่งข้อมูลไปยังตัวควบคุมส่วนกลางโดยจะแจ้งประเภทของงานและขนาดของแบนด์วิดท์ที่ต้องการส่งให้ระบบ

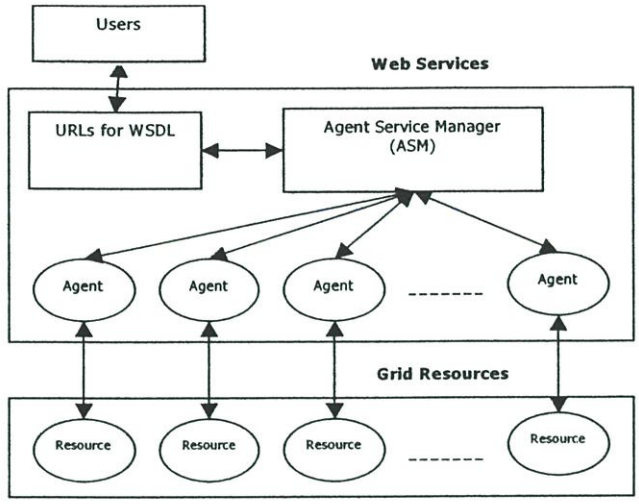
การแบ่งงานประยุกต์ของเอเจนต์จะแบ่งออกเป็น 2 ประเภทคืองานประยุกต์แบบอิลาสติกและงานประยุกต์แบบเรียลไทม์ ตามที่ได้อธิบายมาแล้ว เอเจนต์จะมีส่วนบริการอยู่ส่วนหนึ่งสำหรับเปรียบเทียบประเภทของงานประยุกต์และแบนด์วิดท์ที่งานประยุกต์นั้นต้องการ ซึ่งการกำหนดแบนด์วิดท์สำหรับงานประยุกต์นั้น ถ้าเป็นแบบอิลาสติจะกำหนดให้เท่ากับขนาดของงานและถ้าเป็นแบบเรียบไทม์จะกำหนดตามมาตราฐานตาม [34] ซึ่งแสดงดังตารางที่ 4.4

ตารางที่ 4.4 ตัวอย่างงานประยุกต์และแบนด์วิดท์ที่ต้องการ

ประเภทงานประยุกต์	แบนด์วิดท์ที่ต้องการ
Adaptive Differential PCM	32 Kbps
Internet Voice PCM	64Kbps
Voice Over IP	64Kbps
Audio MP3 Stream	128Kbps
Real time Internet CD	750Kbps
MPEG1	1Mbps
MPEG-2	6Mbps
MPEG-4	4Mbps
Broadcast quality HDTV	19.4Mbps
Video Uncompress	1.5Gbps

4.4.3 การประยุกต์ระบบมัลติเอเจนต์สำหรับระบบการบริการแบบกริด

การประยุกต์ระบบมัลติเอเจนต์สำหรับการบริการแบบกริดจะอ้างอิงตาม [1] ซึ่งเป็นการออกแบบโดยอ้างอิงการทำงานตามมาตรฐานการให้บริการระบบเปิด และออกแบบเพื่อใช้กับการทำงานของระบบกริดบนเว็บเซอร์วิสประกอบด้วยส่วนควบคุมการให้บริการ (Service Manager) ซึ่งเมื่อนำมาประยุกต์ใช้งานในระบบมัลติเอเจนต์ก็คือตัวควบคุมส่วนกลางนั้นเอง โดยที่เอเจนต์จะเป็นส่วนให้บริการที่กระจายอยู่ทั่วไปในระบบ



รูปที่ 4.9 การใช้ระบบมัลติเอเจนต์สำหรับการบริการระบบกริด

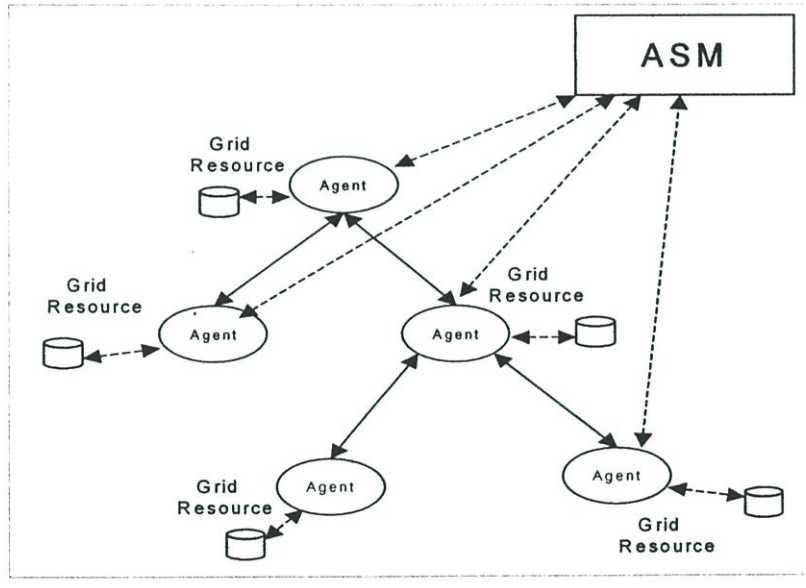
เนื่องจากระบบกริดอยู่บนพื้นฐานของเว็บเซอร์วิส ผู้ใช้ระบบจะติดต่อกับระบบโดยระบุผ่านทางยูอาร์แอล (URL) ของเว็บเซอร์วิส เว็บเซอร์วิสจะกำหนดการติดต่อผ่านทาง WSDL จากนั้นระบบจะทำการค้นหาที่อยู่ของข้อมูลตามขบวนการของระบบเว็บเซอร์วิส ตามที่ได้กล่าวถึงมาโดยใช้อัลกอริทึมการค้นหาแบบกระจายตาม [1] หรือวิธีการอื่นที่เหมาะสม การควบคุมคุณภาพการให้บริการทั้งหมดจะถูกควบคุมโดยตัวควบคุมส่วนกลาง ตัวควบคุมส่วนกลางจะสื่อสารกับเอเจนต์เพื่อการขอรับข้อมูลเพราะเอเจนต์จะเป็นส่วนที่ให้บริการข้อมูลเกี่ยวกับทรัพยากรของกริด ซึ่งจะกำหนดพารามิเตอร์ผ่านทางแอตทริบิว ขบวนการทำงานของการควบคุมคุณภาพการให้บริการนั้นจะเริ่มเมื่อระบบค้นหาที่อยู่เจอแล้วและพร้อมที่จะทำการส่งข้อมูลไปให้ผู้ร้องขอ โครงสร้างพื้นฐานของการประยุกต์ใช้ระบบมัลติเอเจนต์สำหรับการบริการแบบกริดแสดงดังรูป 4.9

จากรูปแสดงโครงสร้างการใช้งานของระบบ ประกอบด้วยผู้ใช้จะทำการร้องขอใช้ข้อมูลผ่านทาง URL และ WSDL ในระบบของเว็บเซอร์วิสและความต้องการจะถูกส่งต่อไปยังตัวควบคุมส่วนกลาง (Agent Service Manager: ASM) ซึ่งตัวควบคุมส่วนกลางจะรับรู้สถานะภาพของเอเจนต์ทั้งหมดสำหรับเอเจนต์นั้นจะมีหน้าที่ควบคุม ให้บริการและตรวจสอบการทำงานของสถานีกริดว่าต้องการที่จะส่งข้อมูลหรือไม่ เมื่อต้องการส่งก็จะส่งสัญญาณร้องขอและข้อมูลของงานประยุกต์เข้ามา เอเจนต์จะรายงานความต้องการส่งข้อมูลไปยังตัวควบคุมส่วนกลางและ หลังจากนั้นตัวควบคุมส่วนกลางจะกำหนดลำดับการส่งข้อมูลตามกลวิธีของการควบคุมคุณภาพการให้บริการ และจะส่งข้อมูลกลับไปยังเอเจนต์ที่ร้องขอบริการมาเพื่อบังคับการทำงาน

4.4.4 การจัดโครงข่ายของเอเจนต์

โดยปกติแล้วระบบกริดจะจัดโครงข่ายการเชื่อมต่อแบบกราฟ ซึ่งเมื่อนำมาประยุกต์กับระบบการประมวลผลแบบกระจายขนาดใหญ่แล้วจะมีความซับซ้อนและยุ่งยากสำหรับควบคุมการทำงาน ดังนั้นโครงสร้างของเอเจนต์ที่นำเสนอภายในระบบจะมีลักษณะเป็นแบบลำดับชั้นแบบฮีพ เพื่อทำให้การควบคุมการทำงานและจัดการได้ง่าย ซึ่งจะเกิดประสิทธิภาพสูงเมื่อระบบนั้นมีจำนวนสถานีของกริดมาก นั่นก็หมายความว่าโครงข่ายของกริดนั้นจะเชื่อมต่อแบบกราฟหรือรูปแบบใดก็ได้ แต่เอเจนต์จะจัดโครงข่ายในการทำงานแบบฮีพ สำหรับเอเจนต์ซึ่งมีหน้าที่ในการอำนวยความสะดวกในการให้บริการของสถานีกริดโดยการควบคุมการบริการและทรัพยากร ซึ่งการจัดโครงข่ายของการประยุกต์ใช้งานระหว่างเอเจนต์กับระบบกริดแสดงดังรูปที่ 4.10

ข้อแตกต่างของงานวิจัยนี้กับ[1] คือการจัดโครงข่ายของเอเจนต์ซึ่งจาก [1] นั้นจะเน้นไปการค้นหาสถานีกริดเพียงอย่างเดียวจึงไม่สนใจเรื่องความสมดุลย์ของเส้นทาง แต่ในวิทยานิพนธ์นี้เป็นการเสนอแนวคิดสำหรับการควบคุมคุณภาพการให้บริการ โดยต้องมีการพิจารณาถึงความจุหรือแบนด์วิดท์ของเส้นทางการส่ง จึงต้องจัดให้สมดุลย์ระหว่างเส้นทางแต่ละสถานีกริด โดยหลักการพิจารณาในการจัดโครงสร้างจะได้อธิบายในหัวข้อต่อไป



รูปที่ 4.10 รูปแบบการจัดโครงข่ายของเอเจนต์

4.4.5 ตัวควบคุมเอเจนต์ส่วนกลาง

ตัวควบคุมเอเจนต์ส่วนกลาง (Agent Service Manager: ASM) เป็นศูนย์รวมข้อมูลบริการของระบบมัลติเอเจนต์ มีลักษณะเหมือนส่วนบริการข่าวสารข้อมูลของระบบกริด (Grid Information

Service) ที่จะจัดเก็บรายละเอียดข่าวสารข้อมูลทุกอย่างของระบบ เนื่องจากเอเจนต์จะรายงานบริการและทรัพยากรของกริดมาเก็บไว้ที่ตัวควบคุมส่วนกลาง เพื่อเก็บไว้ในการประเมินผลในการควบคุมคุณภาพการบริการ ซึ่งจะต่างจาก [1] ซึ่งเก็บไว้เป็นข้อมูลสำหรับลงทะเบียนเท่านั้นไม่ได้ใช้ประโยชน์มากนักในการค้นหาเพราะการจัดการส่วนใหญ่จะให้ความสำคัญที่ PSTซึ่งกระจายอยู่ในเอเจนต์มากกว่า สำหรับในการควบคุมคุณภาพการบริการจะใช้เป็นตัววิเคราะห์และประมวลผลออกข้อกำหนดต่างๆ ในการทำควบคุมคุณภาพการให้บริการ ในขณะที่ตัวเอเจนต์เป็นส่วนวิเคราะห์ข้อมูลส่วนน้อยแต่จะเป็นตัวสำคัญในการรายงานการบริการและค่าข้อมูลสถิติของกริด ซึ่งในตัวควบคุมส่วนกลางจะไม่มีหน้าการทำงานในส่วนนี้ โครงสร้างของตัวควบคุมส่วนกลางแสดงดังรูปที่ 4.11 ซึ่งบริการและทรัพยากรของกริดนั้นจะถูกควบคุมโดยเอเจนต์ และเอเจนต์จะรายงานเมื่อมีข้อมูลบางอย่างของทรัพยากรเปลี่ยนแปลงเพื่อปรับปรุงทรัพยากรของกริดให้เป็นปัจจุบัน ส่วนประกอบของตัวควบคุมส่วนกลางประกอบไปด้วยการทำงานสามส่วนคือ Monitor Module Scheduler Module และ Control Module ดังรูปที่ 4.11 ซึ่งแต่ละส่วนนั้นจะทำงานแบบต่อเนื่องกัน

4.4.5.1 **โมดูลตรวจสอบ** (Monitor Modul)

เมื่อมีการลงทะเบียนโมดุลตรวจสอบจะเป็นตัวตรวจสอบ และจัดการสร้างโครงสร้างของเอเจนต์โดยยึดหลักโครงสร้างและอัลกอริทึมของฮีพ ซึ่งวิธีการจัดจะได้กล่าวถึงในบทต่อไป สำหรับหน้าที่หลัก ๆ ของมอนิเตอร์โมดูลมีดังต่อไปนี้

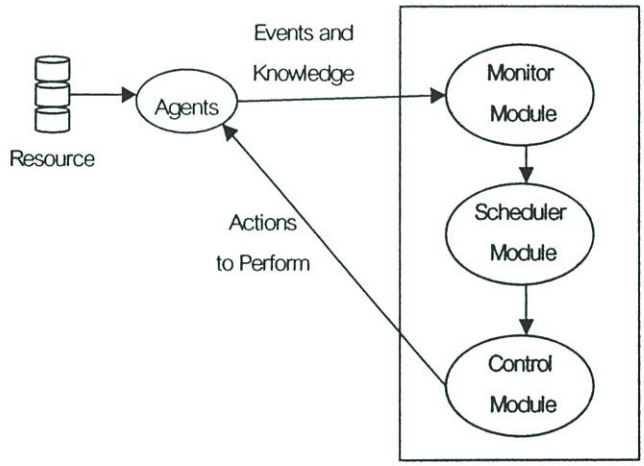
- จัดโครงสร้างของเอเจนต์ : การจัดโครงสร้างของเอเจนต์จะใช้อัลกอริทึมของฮีพตามที่กล่าวมาข้างต้นเช่น การเพิ่มเอเจนต์ การลบเอเจนต์ การปรับปรุงเอเจนต์ เป็นต้น โดยจะจัดการทุกครั้งที่มีการเพิ่ม ลบเอเจนต์และปรับปรุงรายละเอียดทรัพยากรของกริด
- จัดเก็บทะเบียนประวัติของเอเจนต์: เมื่อระบบมีการลงทะเบียนเพื่อขอเป็นสมาชิก Monitor Module จะจัดการเชื่อมต่อเอเจนต์ขึ้นใหม่มาเพื่อคอยให้บริการสถานีกริดที่ลงทะเบียนใหม่ และพร้อมกันนั้นก็จะสร้างทะเบียนของเอเจนต์เพื่อเป็นข้อมูลพื้นฐานในการใช้งาน
- สร้างเส้นทางข้อมูล: เป็นการสร้างเส้นทางเดินของข้อมูล (Path flow) ของแต่ละสถานีกริดและเก็บไว้ในฐานข้อมูลแบบออนไลน์ โดยการสร้างจะพิจารณาตามการเชื่อมต่อความจุแบบต่ำลง
- ตรวจสอบการร้องขอ : เป็นขั้นตอนการใช้งาน โดยที่โมดุลตรวจสอบจะรับการร้องขอการบริการจากเอเจนต์ หลังจากนั้นจะบันทึกลงเก็บไว้ในหน่วยความจำเพื่อให้โมดูลอื่นใช้งาน

4.4.5.2 **โมดูลจัดลำดับ** (Scheduler Module)

เป็นโมดูลหลักในการทำควบคุมคุณภาพการให้บริการหลังจากที่โมดูลตรวจสอบรับข้อมูลของเอเจนต์มาจะเก็บไว้ในหน่วยความจำ โดยการคุมคุณภาพการให้บริการนั้นจะใช้หลักการของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึมซึ่งจะได้อธิบายต่อไป

4.4.5.3 **โมดูลควบคุม** (Control Module)

มีหน้าที่ในการนำลำดับหรือผลจากการทำงานของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม นำไปปฏิบัติโดยการบังคับใช้โมดูลควบคุมจะส่งข้อมูลไปแจ้งให้เอเจนต์ทราบว่ายอมให้ส่งข้อมูลได้ ในจำนวนที่กำหนดจากมัลติโปรเซสส์ไพรออริตี้ หลังจากนั้นเอเจนต์จะส่งข้อมูลไปจนเสร็จ ในขณะที่ทำงาน ทั้งสามโมดูลจะทำงานสัมพันธ์กันตลอดเวลา คือมอร์นิเตอร์โมดูลจะคอยตรวจสอบข้อมูลจากเอเจนต์ตลอดเวลาและบันทึกข้อมูลไว้ในหน่วยความจำและโมดูลจัดลำดับจะคอยตรวจสอบข้อมูลในความจำและประมวลผลแล้ว ส่งข้อมูลไปให้โมดูลควบคุมเพื่อบังคับให้เอเจนต์ส่งข้อมูล ในขณะที่เอเจนต์ส่งข้อมูลนั้นระบบก็ต้องรู้ตลอดเวลาว่าข้อมูลนั้นส่งผ่านเส้นทางใดบ้างเพื่อจะนำไปเป็นค่าสถิติในการกำหนดการส่งข้อมูลชุดต่อไป นอกจากนี้โมดูลควบคุมจะมีหน่วยความจำส่วนหนึ่งไว้คอยจัดการประมวลผลข้อมูลของเอเจนต์และเก็บเส้นทางการเดินของข้อมูล โดยความจำส่วนนี้อาจเป็นฐานข้อมูลแบบออนไลน์คือฐานข้อมูลที่มีการเปลี่ยแปลงข้อมูลของเอเจนต์อยู่ตลอดเวลา



รูปที่ 4.11 โครงสร้างของตัวควบคุมเอเจนต์ส่วนกลาง (ASM)

จากหลักการทำงานที่อธิบายทั้งหมดจะเห็นได้ว่าเมื่อระบบมีขนาดใหญ่ จะเกิดการสื่อสารข้อมูลระหว่างระเอเจนต์และตัวควบคุมส่วนกลางเป็นจำนวนมากซึ่งเป็นข้อมูลที่เกิดขึ้นจากการควบคุมการทำงานของระบบเช่นข้อมูลในการปรับโครงสร้างของเอเจนต์ ข้อมูลในการร้องขอส่งที่มาจากเอเจนต์ และข้อมูลในการบังคับการทำงานที่เกิดจากตัวควบคุมส่วนกลาง ในกรณีจะไม่เกิดปัญหากับระบบเนื่องจากว่าข้อมูลที่เกิดขึ้นนั้นเป็นข้อมูลส่วนน้อย เมื่อเทียบกับข้อมูลที่เป็นงานประยุกต์ที่เกิดขึ้นในระบบ

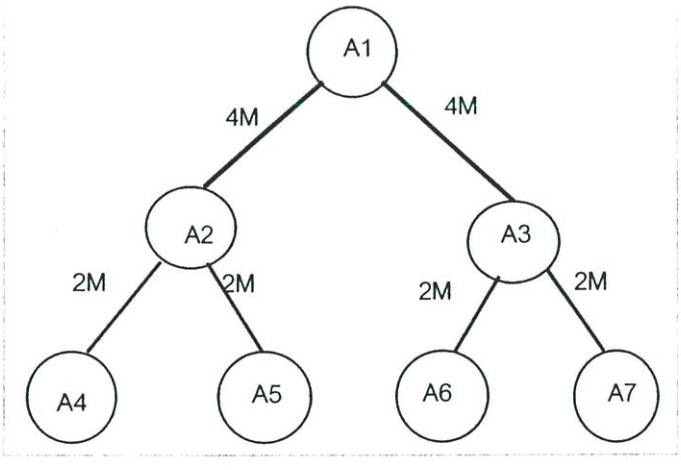
4.5 การกำหนดลำดับความสำคัญให้กับเอเจนต์

การจัดโครงข่ายจากหัวข้อที่ผ่านมาจะเห็นว่าเอเจนต์นั้นจะจัดโครงสร้างเป็นแบบฮีพ ซึ่งเป็นปัญหาว่าจะใช้ปัจจัยอะไรเพื่อมาเป็นเงื่อนไขในการจัดโครงข่ายของเอเจนต์ เพื่อให้สมนัยกับการให้บริการแบบลำดับความสำคัญ วิทยานิพนธ์นี้ได้นำเสนอหลักการจัดโครงสร้างแบบการเชื่อมต่อความจุแบบต่ำลงซึ่งเชื่อมแบนด์วิดท์หรือความจุของเอเจนต์ค่ามากมาน้อยจากบนลงล่าง โดยการจัดลำดับความสำคัญตามแบนด์วิดท์ของสถานีกริด

4.5.1 การกำหนดลำดับความสำคัญโดยแบนด์วิดท์

เริ่มต้นในการลงทะเบียนระบบจะสร้างเอเจนต์ขึ้นมา เพื่อเป็นตัวควบคุมบริการและทรัพยากรของกริด โดยเริ่มแรกในการขอเป็นสมาชิก เอเจนต์จะตรวจสอบขนาดแบนด์วิดท์ของสถานีกริดใหม่เพื่อหาค่าแบนด์วิดท์ จากนั้นตัวควบคุมส่วนกลางจะนำแบนด์วิดท์ของกริดที่สมัครสมาชิกใหม่นี้ไปปรับปรุงกับโครงข่ายของระบบเอเจนต์เดิมที่มีอยู่แล้ว โดยการทำงานนั้นจะจัดตามรูปแบบอัลกอริทึมของฮีพซึ่งเมื่อจัดแล้วจะได้โครงสร้างของระบบการเชื่อมต่อความจุแบบต่ำลง [3,4] ดังรูปที่ 4.12

ในกรณีที่มีการปรับปรุงแบนด์วิดท์ของกริด เอเจนต์จะรายงานแบนด์วิดท์ใหม่ไปให้ตัวควบคุมส่วนกลางทราบตัวควบคุมส่วนกลางก็จะใช้ฮีพอัลกอริทึมในการปรับปรุงโครงสร้างใหม่ ซึ่งก็คือการให้ลำดับความสำคัญใหม่แก่เอเจนต์ตามแบนด์วิดท์ใหม่นั่นเอง ในกรณีที่มีการบอกเลิกสมาชิกก็จะจัดการในกรณีเดียวกัน จากที่อธิบายมาจะแสดงให้เห็นว่าการจัดโครงข่ายของเอเจนต์นั้นจะมีการเปลี่ยนแปลงตลอดเวลาขึ้นอยู่กับการเปลี่ยนแปลงแบนด์วิดท์ของสถานีกริด



รูปที่ 4.12 การจัดโครงสร้างของเอเจนต์โดยใช้การเชื่อมต่อความจุแบบต่ำลงตามแบนด์วิดท์

มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมในหัวข้อที่ 4.2 เป็นโครงสร้างที่ยังไม่ได้กำหนดลำดับความสำคัญตามแบนด์วิดท์ เมื่อนำมาประสำหรับการควบคุมคุณภาพสำหรับการให้บริการแล้ว จะต้องปรับโครงสร้างของการเชื่อมต่อแต่ละเอเจนต์ ซึ่งเดิมนั้นการจัดโครงสร้างเป็นแบบฮีพโดยไม่มีการใช้เงื่อนไขในการเชื่อมต่อระหว่างเอเจนต์ จากโครงสร้างจะพิจารณาเห็นว่ากริดที่อยู่ในระดับต่ำเช่น ระดับ 0 หรือ 1 นั้นจะมีภาระการรองรับปริมาณของทราฟฟิคมากขึ้นเรื่อย ๆ ตามลำดับดังนั้นเพื่อลดภาระดังกล่าวจึงได้พิจารณาการจัดตามหลักการของการเชื่อมต่อความจุแบบต่ำลง ซึ่งเป็นการลิงค์แต่ละโหนดตามขนาดแบนด์วิดท์ในการส่งของข้อมูล

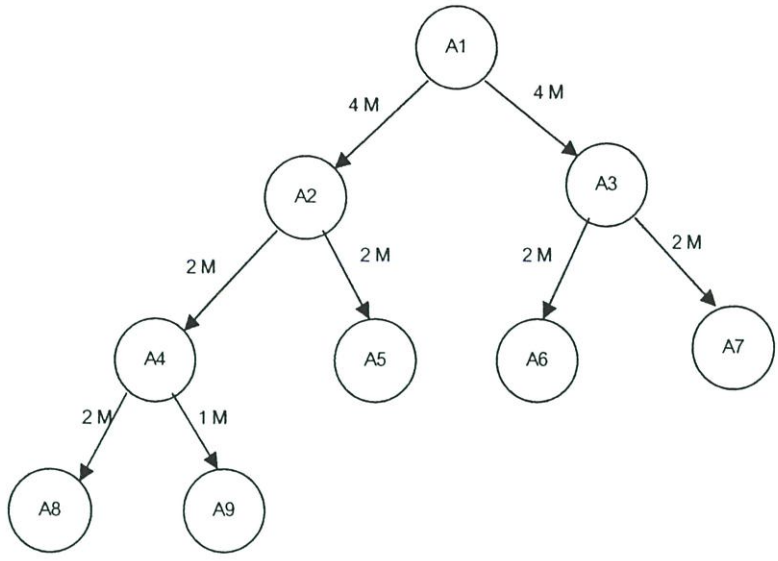
การจัดโครงสร้างตามหลักการของการเชื่อมต่อความจุแบบต่ำลงนั้นเป็นการจัดโครงสร้างตามขนาดแบนด์วิดท์ของกริด โดยกำหนดให้เอเจนต์ที่ควบคุมงานบริการของกริดที่มีแบนด์วิดท์สูงจะจัดให้มีลำดับความสำคัญสูงกว่าเอเจนต์ที่ควบคุมงานบริการของกริดที่มีแบนด์วิดท์ต่ำ ส่วนมากแล้วการประยุกต์ใช้อัลกอริทึมกับโครงสร้างแบบนี้นิยมใช้อัลกอริทึมแบบ WFQ [3,4] ซึ่งเป็นการพิจารณาเลือกแพกเก็ตของข้อมูลอย่างเป็นธรรม

เมื่อพิจารณาจากรูปที่ 4.12 จะเห็นได้ว่า เอเจนต์ A1 จะมีลำดับความสำคัญสูงกว่าเอเจนต์ A2 และเอเจนต์ A3 นั้นก็หมายความว่า A1 นั้นต้องมีความจุหรือแบนด์วิดท์สูงกว่าเอเจนต์ A2 และเอเจนต์ A3 คือ 4 Mbps จากการจัดโครงสร้างดังกล่าวสามารถบอกได้ว่าแบนด์วิดท์ของเอเจนต์พ่อจะมีมากกว่าหรือเท่ากับแบนด์วิดท์ของเอเจนต์ลูกเสมอ แต่ความจุในระดับเดียวกันอาจจะไม่เท่ากันก็ได้ ในกรณีที่แบนด์วิดท์ของเอเจนต์พ่อมีค่าเท่ากับแบนด์วิดท์ของเอเจนต์ลูก จะเกิดปัญหาขึ้นเมื่อนำมาใช้กับมัลติโปรเซสส์ไพรออริตีอัลกอริทึมเพราะเอเจนต์พ่อจะมีทราฟฟิคไหลผ่านมากเกินไป ซึ่งในวิทยานิพนธ์ได้แก้ปัญหาโดยการใช้ดัมมีเอเจนต์

4.5.2 การแก้ปัญหาเอเจนต์ที่มีแบนด์วิดท์เท่ากัน

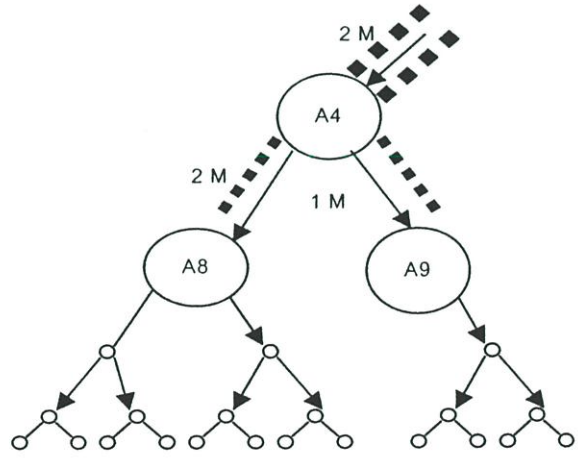
ในกรณีที่กริดมีแบนด์วิดท์เท่ากันดังรูปที่ 4.13 จะเห็นว่า A4 และ A8 นั้นมีค่าแบนด์วิดท์เท่ากันคือ 2 Mbps ถ้าจัดให้ A8 เป็นลูกของ A4 จะทำให้เกิดปัญหาคือมีทราฟฟิคไหลผ่านตัวที่เป็นพ่อคือ A4 มากเกินไปทั้งที่มีแบนด์วิดท์เท่ากันแสดงดังรูปที่ 4.14

จากปัญหาที่อธิบายมาสามารถแก้ปัญหานี้ได้โดยสร้างดัมมี่ (Dummy) เอเจนต์ขึ้นมาซึ่งดัมมี่เอเจนต์นี้เป็นเอเจนต์ที่สร้างขึ้นให้ครบตามโครงสร้างของระบบเท่านั้นเมื่อระบบสามารถหาค่าแบนด์วิดท์ที่เหมาะสมได้ก็จะสร้างการเชื่อมต่อของเอเจนต์ตามโครงสร้างปกติขึ้นมา โครงสร้างที่ใช้ดัมมี่แสดงดังรูปที่ 4.15



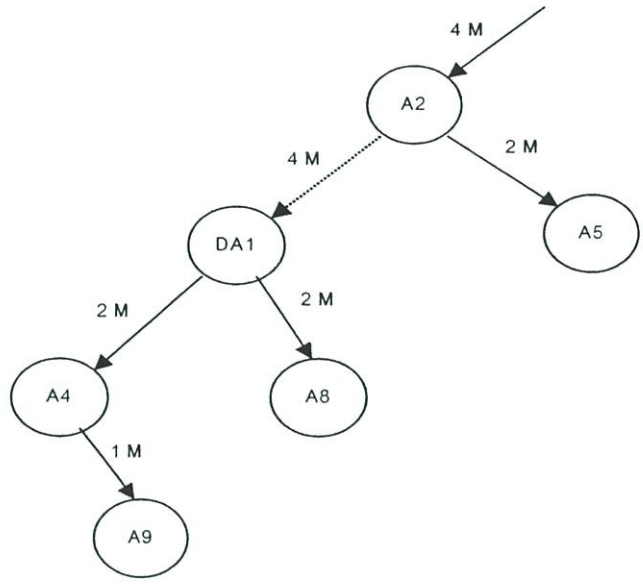
รูปที่ 4.13 โครงสร้างของเอเจนต์เมื่อมีความจุเท่ากัน

จากรูปที่ 4.13 จะเห็นว่าแบนด์วิดท์เอเจนต์ A2 เท่ากับเอเจนต์ A4 มีแบนด์วิดท์เท่ากันคือ 2 Mbps ถ้าหากว่าเอเจนต์ A4 มีเอเจนต์ลูกและหลานมาก ๆ ดังรูปที่ 4.14 จะทำให้เส้นทางการส่งของ A2 ไป A4 มีความเสี่ยงที่จะมีข้อมูลไหลผ่านมากและเราได้กำหนดดัมมี่เอเจนต์มาแก้ปัญหาจุดนี้ โดยที่ดัมมี่เอเจนต์เป็นเอเจนต์ไม่ได้เชื่อมต่อกับกริด แต่มีไว้เพื่อการจัดการเส้นทางการส่งให้เมาะสม เพื่อกระจายการรองรับทราฟฟิคในระบบเท่านั้น



รูปที่ 4.14 ความแออัดของข้อมูลในกรณีที่ความจุเท่ากัน

จากรูปที่ 4.14 จะเห็นว่าเอเจนต์ A4 นั้นมีแบนด์วิดท์ในการส่งข้อมูล 2 Mbps ในขณะที่ A8 นั้นก็มีแบนด์วิดท์ในการส่งข้อมูล 2 Mbps ซึ่งมีขนาดเท่ากันแต่รองรับข้อมูลไม่เท่ากัน



รูปที่ 4.15 โครงสร้างของระบบเมื่อใช้ดัมมี่เอเจนต์

จากรูปที่ 4.15 เอเจนต์ DA1 คือดัมมี่เอเจนต์ที่สร้างขึ้นมาเพื่อช่วยลดปัญหาของทราฟฟิคที่มีมากเกินไปของการขนส่งข้อมูล โดยที่ DA1 สร้างขึ้นมาเพื่อให้โครงสร้างข้อมูลครบสมบูรณ์และจัดการแบนด์วิดท์ให้ถูกต้องเท่านั้น สำหรับสถานที่อยู่ของเอเจนต์อาจจะอยู่ใกล้ ๆ หรืออยู่ที่เดียวกับเอเจนต์ A2 นั้นก็คือดัมมี่เอเจนต์จะมีแบนด์วิดท์เท่ากับ A2 แต่มีระดับความลึกต่างกัน

4.6 การควบคุมคุณภาพการให้บริการตามหลักของมัลติโปรเซสส์ไพรออริตี้

ลักษณะของการควบคุมคุณภาพการให้บริการของระบบมัลติเอเจนต์ ประยุกต์ใช้กลไกหลายหลักการเพื่อใช้ในการทำควบคุมคุณภาพการให้บริการ ซึ่งกลไกของแต่ละรูปแบบนั้นจะแตกต่างกันไป ขึ้นอยู่กับว่าจะทำเกี่ยวกับเรื่องใดหรือส่วนไหนของระบบเครือข่าย ในส่วนของการใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมใช้กลไกต่าง ๆ ดังนี้คือ

- QoS control เป็นการจัดรูปแบบการจัดการเส้นทางในการส่งข้อมูล การทำการควบคุมคุณภาพการให้บริการแบบนี้จะควบคุมเส้นทาง (Flow Control) ของการส่งข้อมูลของแต่ละกริดโหนดไว้ล่วงหน้าและเมื่อจะทำการส่งข้อมูลระบบจะบังคับส่งตามเส้นทางที่กำหนดไว้แล้วเท่านั้นซึ่งการจัดเส้นทางนั้นจะจัดตามเส้นทางการส่ง

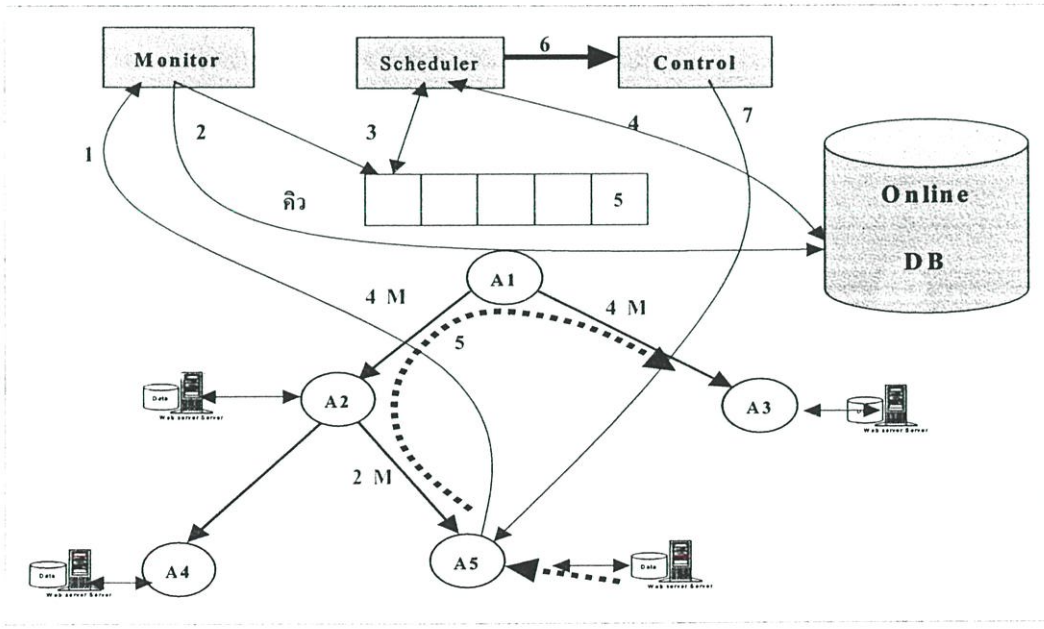
• QoS Management เป็นการจัดการเส้นทางโดย QoS Monitoring จะทำการตรวจสอบว่าเส้นทางปัจจุบันมีข้อมูลอะไรอยู่ในเส้นทางบ้าง กริดไหนกำลังส่งอยู่บ้าง ใช้แบนด์วิดท์กี่ Kbps และเหลือแบนด์วิดท์กี่ Kbps รวมทั้งการตรวจสอบทรัพยากรของแต่ละกริด เพื่อที่จะใช้ประกอบในการตัดสินใจในการส่งข้อมูลในแต่ละครั้ง

จากหลักการที่อธิบายมาข้างต้นจะเห็นว่า การจัดโครงสร้างจะกำหนดให้ตัวที่อยู่บนสุดจะมีลำดับความสำคัญมากกว่าตัวที่อยู่ในระดับที่รองลงมา โดยการจัดระดับหรือลำดับความสำคัญจะกำหนดตามแบนด์วิดท์ของสถานีกริด ซึ่งอาจกล่าวได้ว่าเอเจนต์ที่มีคีย์ต่ำสุดหรือมีค่าน้อยสุดจะควบคุมสถานีของกริดที่มีแบนด์วิดท์สูงสุด

การควบคุมคุณภาพการให้บริการนั้น จะเป็นลักษณะการกำหนดขนาดของแบนด์วิดท์ตามเส้นทางการส่งให้กับงานประยุกต์ของสถานีกริดที่ร้องขอ โดยการตรวจสอบว่าสามารถส่งทราฟฟิคของงานไปในเส้นทางได้หรือไม่และถ้าไม่ได้จะต้องทำการปรับขนาดของแบนด์วิดท์ใหม่ในกรณีงานแบบอิลาสติก แต่ถ้าเป็นงานแบบเรียลไทม์จะยกเลิกการส่งทันทีเพราะจะมีผลต่อการประมวลผลของงานที่กำลังประมวลผลอยู่ขณะนั้น

4.6.1 หลักการควบคุมคุณภาพการให้บริการของระบบ

หลักการควบคุมคุณภาพการให้บริการของระบบนั้น ปรับปรุงมาจากมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม การทำงานนั้นจะเริ่มต้นจากผู้ใช้ค้นหาข้อมูลพบแล้วและต้องการส่งข้อมูล กริดจะแจ้งความต้องการไปยังเอเจนต์หลังจากนั้นเอเจนต์จะทำการวิเคราะห์งานงานประยุกต์นั้นว่า เป็นงาน



รูปที่ 4.16 การทำงานของระบบ

ชนิดใดและต้องการแบนด์วิดท์ต่ำสุดในการส่งกี่ Kbps เมื่อวิเคราะห์ได้แล้วเอเจนต์จะส่งข้อมูลที่กล่าวมาไปยัง ASM โดย Monitor Module ใน ASM จะเก็บข้อมูลไว้ในหน่วยความจำซึ่งในที่นี้คือคิว ตาม โครงสร้างของฮีพ ในขณะเดียวกัน Scheduler Module ใช้อัลกอริทึมควบคุมคุณภาพการให้บริการ โดยใช้ข้อมูลในคิวและใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมทำ เมื่อประมวลผลเสร็จแล้วก็ ASM โดย Control Module จะส่งข้อมูลซึ่งเป็นข้อมูลของเอเจนต์ที่ผ่านการเลือกแล้วแจ้งให้เอเจนต์รู้ และเอเจนต์ก็จะเริ่มส่งข้อมูลตามเส้นทางต่อไป รูปที่ 4.16 แสดงการทำงานของระบบ

ลำดับการทำงานของระบบตามรูปที่ 4.16 อธิบายได้ดังนี้

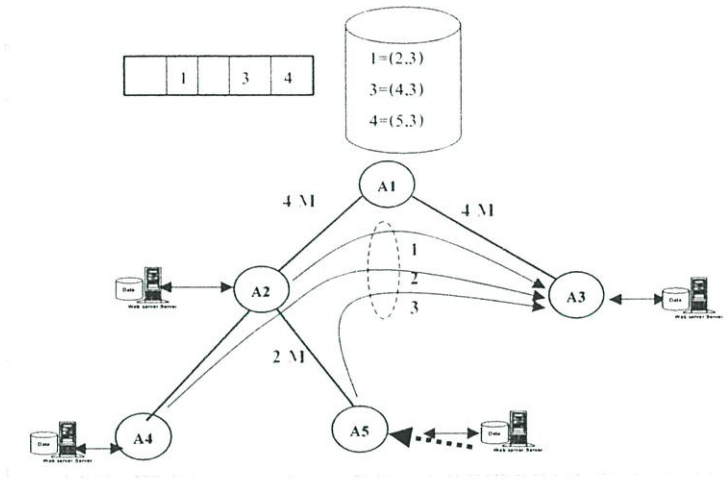
1. Agent รับข้อมูลจากกริดและทำการประมาณค่าของแบนด์วิดท์ของงาน
2. Monitor Module ตรวจสอบและเก็บความต้องการไว้ใน Queue และฐานข้อมูลออนไลน์
3. Scheduler Module ใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมจัดการข้อมูลในคิว
4. Scheduler Module ตรวจสอบข้อมูลฐานข้อมูลออนไลน์
5. มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมตรวจสอบแบนด์วิดท์ตามเส้นทางการส่ง
6. Scheduler Module ส่งผลลัพธ์ในการไปให้ Control Module
7. Control Module ส่งข้อมูลมาที่เอเจนต์เพื่อบังคับการส่งข้อมูล

จากลำดับการทำงานที่อธิบายมานั้นเป็นการทำงานเพียงลำดับเดียว แต่อัลกอริทึมแบบมัลติโปรเซสส์ไพรออริตี้นั้นสามารถยอมให้ส่งข้อมูลแบบพร้อม ๆ กันได้ ซึ่งเป็นการส่งข้อมูลแบบหลาย ๆ ลำดับความสำคัญไปพร้อม ๆ กัน เรียกว่าเป็นการทำงานแบบมัลติโปรเซสส์ไพรออริตี้

4.6.2 การทำงานในลักษณะมัลติโปรเซสส์

มัลติโปรเซสส์ไพออริตี้เป็นรูปแบบการส่งงานประยุกต์หลาย ๆ ลำดับความสำคัญหรือหลายงานพร้อมกันแต่เนื่องจากว่าในวิทยานิพนธ์นี้ยังใช้โปรเซสเซอร์ในการจัดการตัวเดียว ดังนั้นการทำงานจึงไม่สามารถทำงานแบบขนานได้ แต่จะทำแบบเรียงลำดับของลำดับความสำคัญโดยไม่ต้องรอให้งานที่มีลำดับความสำคัญสูงกว่าทำงานเสร็จก่อน ก็สามารถทำงานที่มีลำดับความสำคัญรองลงไปได้ ซึ่งการส่งข้อมูลจะส่งได้หรือไม่ได้นั้นก็ขึ้นอยู่กับเส้นทางการส่งว่ามีแบนด์วิดท์เพียงพอหรือไม่ลักษณะการทำงานแสดงดังรูปที่ 4.17

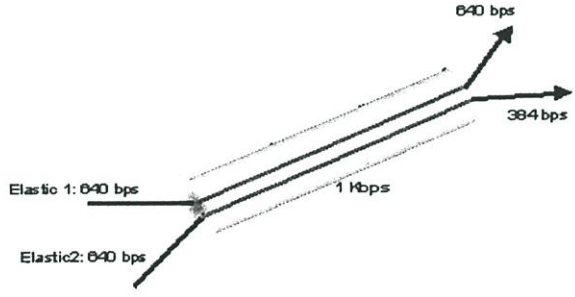
ดังนั้นจะเห็นว่าการส่งข้อมูลนั้นจะทำได้เร็วหรือช้านั้น ขึ้นกับเทคโนโลยีของการค้นหาข้อมูลในคิวลำดับเพื่อตรวจสอบลำดับความสำคัญ และเทคโนโลยีในการค้นข้อมูลเพื่อตรวจสอบเส้นทางที่เก็บอยู่ในฐานข้อมูลออนไลน์ซึ่งในวิทยานิพนธ์นี้จะไม่ให้ความสำคัญในส่วนการค้นข้อมูลในคิว แต่จะให้ความสำคัญในขั้นตอนหลังจากที่ค้นหาข้อมูลและพร้อมที่จะส่งข้อมูลไปในเครือข่าย



รูปที่ 4.17 การทำงานในลักษณะมัลติโปรเซสส์

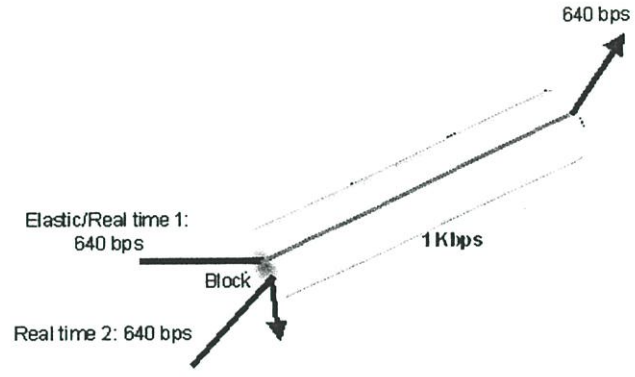
4.6.3 การจัดการแบนด์วิดท์

การจัดการแบนด์วิดท์ เป็นการเลือกความเร็วให้กับงานประยุกต์ที่แต่ละเอเจนต์ต้องการส่งว่าเมื่อเอเจนต์ต้องการส่งข้อมูลนั้นสามารถที่จะส่งได้หรือไม่ตาม Path Flow ของแต่ละเอเจนต์ โดยทั่วไปแล้วงานประยุกต์จะแบ่งออกเป็นสองชนิดคืออิลาสติกและเรียลไทม์ ถ้าเป็นแบบอิลาสติกแบนด์วิดท์ในการส่งข้อมูลนั้นปกติต้องส่งตามแบนด์วิดท์ทั้งหมดของแต่ละกริดที่มีอยู่ หรือตามแบนด์วิดท์กริดในเวลานั้น แต่ถ้าเป็นแบบเรียลไทม์อัลกอริทึมนี้จะยอมให้ส่งได้ก็ต่อเมื่อแต่กริดมีแบนด์วิดท์คงเหลือมากกว่าหรือเท่ากับความต้องการต่ำสุดของงานประยุกต์เท่านั้น เพื่อที่จะรับประกันการให้บริการของระบบ รูปที่ 4.18 การจัดการแบนด์วิดท์แบบอิลาสติก 4.19 แสดงการจัดการแบนด์วิดท์เรียลไทม์



รูปที่ 4.18 การจัดการแบนด์วิดท์ในกรณีที่เป็นงานแบบอิลาสติก

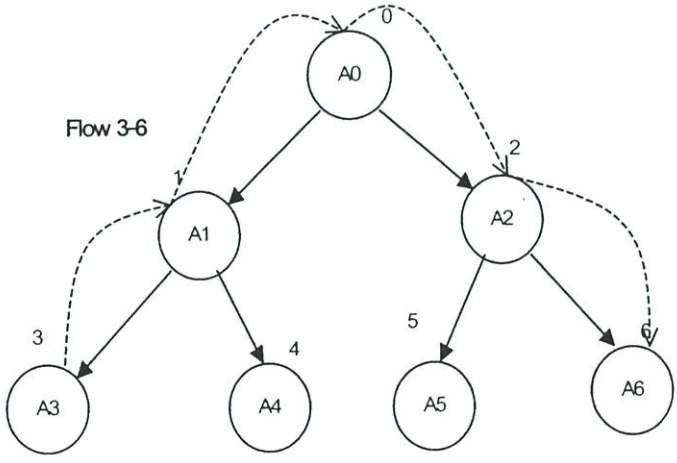
จากรูปเป็นการจัดการแบนด์วิดท์สำหรับงานแบบอิลาสติก โดยแบนด์วิดท์จากเอเจนต์ A1 และเอเจนต์ A2 คือ 1 Kbps และมีงาน Job1 นั้นส่งอยู่ก่อนแล้ว 640 bps เมื่องาน Job2 ต้องการส่งข้อมูลด้วยความต้องการส่งต่ำสุด 640 bps เหมือนกัน ในกรณีนี้ ASM จะต้องปรับความต้องการในการส่งลงมาเหลือ 384 bps ซึ่งต่างกับรูปที่ 4.19 ระบบไม่สามารถให้ส่งได้และงาน Job 2 จะต้องรอการส่งรอบต่อไป



รูปที่ 4.19 การจัดการแบนด์วิดท์ในกรณีที่เป็นงานแบบเรียลไทม์

4.6.4 การวิเคราะห์เส้นทางในการส่งข้อมูล

เมื่อ ASM จะส่งข้อมูลจากสถานีเริ่มต้นไปยังสถานีสุดท้ายหลังจากที่ได้เอเจนต์ที่มีลำดับความสำคัญสูงสุดจาก Queue แล้ว ASM จะต้องมีการวิเคราะห์เส้นทางในการส่งจากจุดเริ่มต้นจนถึงจุดสิ้นสุด โดยที่จะนำเอาค่าความต้องการที่จะส่ง (Minimum Bandwidth) ของแต่ละงานประยุกต์มาเป็นหลักในการพิจารณา ดังรูปที่ 4.20

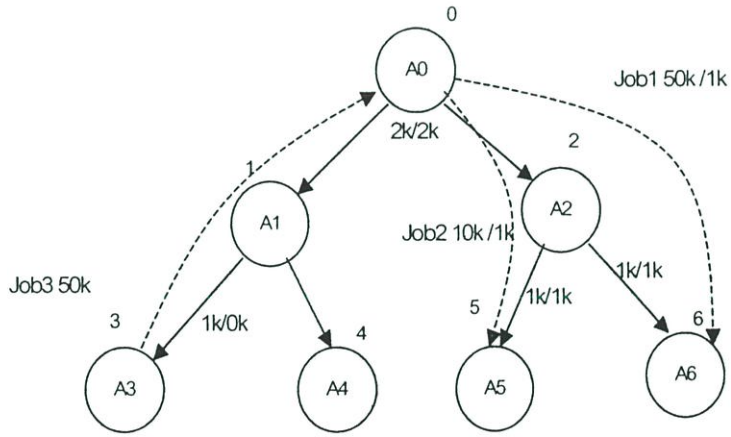


รูปที่ 4.20 แสดงเส้นทางการเดินของข้อมูลจากกริด 3 ไปกริด 6

การวิเคราะห์เส้นทางเป็นการคำนวณทราฟฟิคงานประยุกต์ที่กำลังส่งอยู่แต่ละสถานีกริด โดย Scheduler module ใน ASM จะทำการตรวจสอบกริดแต่ละตัวว่าปัจจุบันส่งงานอะไรบ้างพร้อมทั้งขนาด จากนั้นจะหาผลรวมของงานแต่ละสถานีกริดแล้วเปรียบเทียบกับแบนด์วิดท์ต่ำสุดที่ต้องการส่งของแต่ละงานประยุกต์ จากรูปที่ 4.20 ถ้าต้องการส่งงานจากสถานีกริดจาก 3 ไป 6 จะต้องวิเคราะห์ทราฟฟิคของสถานีกริด 3,1,0,2,6 ตามลำดับ

4.6.5 วิธีการฝากข้อมูล

ในระบบกริดนั้นมีการยอมให้สามารถใช้ทรัพยากรร่วมกันได้ ดังนั้นการควบคุมคุณภาพการให้บริการในที่นี้จึงนำคุณสมบัติข้อนี้มาใช้งาน โดยการฝากข้อมูลนั้นถ้าเป็นงานแบบอิลาสติกสามารถฝากได้ถ้าหากทรัพยากรของกริด ซึ่งก็คือเนื้อที่ในหน่วยความจำเช่น Hark disk หรืออุปกรณ์เก็บข้อมูลอื่นมีค่าเพียงพอ แต่ถ้าเป็นการฝากงานแบบเรียลไทม์นั้นจะสามารถฝากได้ก็ต่อเมื่อระบบตรวจสอบแล้วว่าสถานีกริดที่จะฝากนั้นมีซอฟท์แวร์หรือคอมโพเน้นท์อื่นที่จะใช้ประมวลผลหรือไม่ การฝากข้อมูลนั้นจะเกิดขึ้นได้ก็ต่อเมื่อ ขนาดของแบนดวิดท์ของงานประยุกต์ที่จะส่งนั้นมีค่ามากกว่าแบนดวิดท์ของสถานีกริดตาม Path Flow โดยจะฝากไว้กับสถานีกริดที่ใกล้ที่สุดที่มีทรัพยากรในการรับฝากเพียงพอ



รูปที่ 4.21 หลักการฝากข้อมูลในกรณีที่ไม่สามารถส่งถึงสถานีกริดปลายทาง

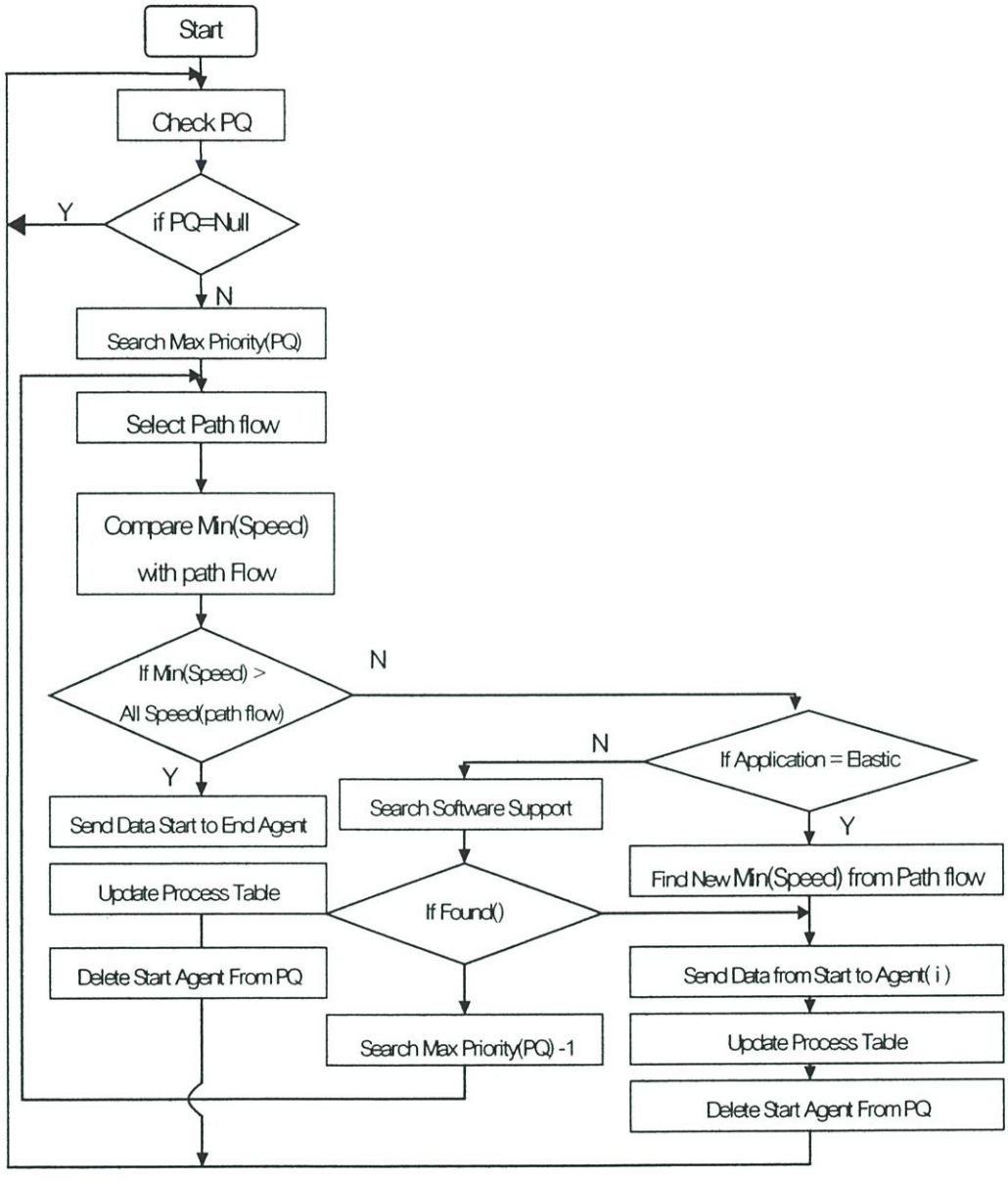
จากรูปแสดงหลักการฝากข้อมูลของอัลกอริทึม โดยที่มีงานประยุกต์ที่จะส่งอยู่ 3 งานคือ Job1 Job2 และ Job3 ในส่วนของ Job2 และ Job3 นั้นกำลังส่งข้อมูลอยู่และใช้แบนด์วิดท์เต็ม คือ Job1 ใช้ 1 Kbps และ Job2 ใช้ 1 Kbps ทำให้กริดที่เอเจนต์ที่ A0 ไม่มีแบนด์วิดท์พอที่จะรองรับงานต่อไป เมื่องานประยุกต์ Job3 ต้องการที่จะส่งงานไปที่สถานีกริดที่เอเจนต์ A0 จะต้องส่งข้อมูลไปฝากไว้ที่

สถานีกริดที่เอเจนต์ A1 ก่อน และปรับปรุงคิวลำดับเพื่อที่จะให้ข้อมูลส่งจากสถานีกริดที่เอเจนต์ A1 ไปยังสถานีกริดที่เอเจนต์ A0 ในรอบต่อไป

4.7 โฟลว์ชาร์ทแสดงการทำงานของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม

จากที่กล่าวมาทั้งหมดเป็นหลักการทำงานของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม ซึ่งสร้างขึ้นมาสำหรับควบคุมคุณภาพการให้บริการในงานวิจัยนี้ เมื่อนำมาเขียนเป็นโฟลว์ชาร์ทจะแสดงได้ดังรูปที่ 4.22 โดยการทำงานจะเริ่มต้นจากเอเจนต์ส่งข้อมูลในการร้องขอแจ้งไปที่ ASM เพื่อขอส่งข้อมูล ซึ่ง ASM จะเก็บข้อมูลส่วนนี้ไว้ในคิวลำดับจากนั้น ASM จะใช้อัลกอริทึมตรวจสอบคิวลำดับตามขั้นตอนคือ

1. ตรวจสอบคิวลำดับถ้าไม่มีข้อมูลให้รอไปเรื่อยจนกว่าเอเจนต์จะส่งข้อมูลเข้ามา
2. เลือกเอเจนต์ที่มีลำดับสูงสุด
3. เลือกเส้นทางในการส่งข้อมูลตาม Path Flow
4. หาค่าแบนด์วิดท์ต่ำสุดตามเส้นทางการส่งข้อมูล
5. ส่งข้อมูลได้ถ้ามีแบนด์วิดท์เพียงพอ
6. ถ้าแบนด์วิดท์ไม่พอต้องตรวจสอบว่าเป็นชนิดของงาน
7. ถ้าเป็นอิลาสติกจะต้องหาแบนด์วิดท์ต่ำสุดแล้วส่งข้อมูลไปไว้ที่เอเจนต์ตัวนั้น
8. ปรับปรุงคิวให้เป็นการส่งข้อมูลจากเอเจนต์ตัวที่รับฝาก
9. ถ้าเป็นเรียลไทม์จะต้องตรวจสอบเอเจนต์ที่จะส่งไปก่อนว่ามีทรัพยากรที่จะใช้หรือไม่
10. ถ้ามีให้ส่งไปและปรับปรุงคิวลำดับ
11. ถ้าไม่มีให้เริ่มต้นการทำงานโดยค้นหาเอเจนต์ตัวที่มีลำดับความสำคัญรองลงไป



รูปที่4.22 โฟลว์ชาร์ทแสดงลำดับการทำงานของมัลติโปรเซสส์อัลกอริทึม

บทที่ 5
การทดลองและผลการทดลอง

5.1 บทนำ

การควบคุมคุณภาพการให้บริการโดยใช้หลักการทำงานของ มัลติโปรเซสล์ไพรออริตี้อัลกอริทึมที่กล่าวมา เป็นแนวคิดพื้นฐานที่ที่นำมาประยุกต์ใช้กับระดับงานประยุกต์โดยไม่มองถึงโปรโตคอลระดับล่างของเครือข่าย หลักการทำงานของอัลกอริทึมนี้จะแตกต่างจากการทำงานของการให้บริการแบบลำดับความสำคัญของ ซึ่งจะให้บริการตามลำดับและเมื่อนำมาประยุกต์ใช้กับโครงสร้างของกริดที่จัดโครงสร้างการให้บริการแบบฮีพ ข้อมูลจะถูกส่งไปทีละชุดในแต่ละช่วงเวลาจากกริดที่ให้บริการไปยังกริดที่ร้องขอไปจนส่งสิ้นสุดการส่ง แล้วจึงเริ่มส่งงานที่มีลำดับความสำคัญลำดับต่อไป สำหรับการใช้มัลติโปรเซสล์ไพรออริตี้อัลกอริทึมกับโครงสร้างของฮีพนั้น ข้อมูลจะถูกส่งพร้อมกันหลายชุดในเวลาใดเวลาหนึ่ง ซึ่งการที่ข้อมูลจะส่งได้จากกริดที่ให้บริการและกริดที่ร้องขอนั้นขึ้นอยู่กับว่า ในเวลานั้นเส้นทางที่ใช้ในการส่งข้อมูลระหว่างตัวรับและตัวส่งมีแบนด์วิดท์เพียงพอหรือไม่ถ้าไม่เพียงพอก็ต้องให้บริการกับกริดอื่น ที่มีลำดับความสำคัญรองลงไปในลำดับซึ่งมีแบนด์วิดท์ในเส้นทางนั้นเพียงพอ

จะเห็นได้ว่าการทำงานของมัลติโปรเซสล์ไพรออริตี้อัลกอริทึมนั้น จะมีการทำงานที่ไม่เป็นไปตามลำดับการทำงานของการให้บริการแบบลำดับความสำคัญเสมอไป แต่อัลกอริทึมจะนำแบนด์วิดท์มาเป็นปัจจัยในการพิจารณาสำหรับการตัดสินใจส่งข้อมูล อีกอย่างในการส่งข้อมูลระหว่างกริดที่ให้บริการและกริดที่ร้องขอแต่ละครั้งนั้น ข้อมูลไม่จำเป็นที่จะต้องส่งให้ถึงกริดที่ร้องขอในทันทีทันใด อัลกอริทึมสามารถยอมให้ส่งข้อมูลจากกริดที่ให้บริการไปฝากที่กริดที่อยู่ในเส้นทางการส่งได้ โดยที่กริดที่รับฝากข้อมูลจะได้เป็นกริดที่ให้บริการแทนกริดเดิม

การประเมินผลประสิทธิภาพของการควบคุมคุณภาพการให้บริการในงานวิจัยนี้จะ ให้ความสำคัญกับการทำงานแบบองค์รวมคือมองประสิทธิภาพรวมทั้งหมดของระบบ ดังนั้นพารามิเตอร์ที่ใช้ในการทดสอบจะกำหนดไปที่เวลาที่ใช้ไปทั้งหมดในการส่งข้อมูล ในกลุ่มข้อมูลแต่ละกลุ่มที่ต้องการส่งข้อมูลพร้อม ๆ กันในเวลาใดเวลาหนึ่ง นั้นหมายความว่าถ้ากริดในโครงสร้างแบบทฮีพต้องการส่งข้อมูลพร้อมกันจำนวน 100 กริด การควบคุมคุณภาพการให้บริการแบบคิวหรือแบบการใช้มัลติโปรเซสล์ไพรออริตี้อัลกอริทึมเป็นตัวจัดการจะทำให้ผลของเวลาดีกว่า

การทดลองในงานวิจัยนี้จะแบ่งการทดสอบออกเป็นสามขั้นตอนคือ

1. การทดสอบการทำงานของระบบกริดระว่างการให้บริการแบบลำดับความสำคัญและมัลติโปรเซสล์ไพรอริตี้กับงานประยุกต์แบบอิลาสติกเพียงอย่างเดียวอยู่ในระบบ โดยจุดประสงค์ใน

การทดสอบคือจะศึกษาถึงผลของการทำงาน เมื่อขนาดของงานประยุกต์และ ขนาดของแบนด์วิดท์ในเครือข่ายที่ต่างกัน

2. การทดสอบการทำงานในระบบกริดระหว่างการให้บริการแบบลำดับความสำคัญและมัลติโปเซสล์ไพรอริตี้กับงานประยุกต์แบบเรียลไทม์เพียงอย่างเดียวอยู่ในระบบ โดยจุดประสงค์เพื่อศึกษาถึงผลของการทำงาน เมื่อขนาดของความต้องการแบนด์วิดท์ของงานประยุกต์แต่ละประเภทและขนาดของงานประยุกต์ที่ต่างกันในระบบ

3. การทดสอบการทำงานในระบบกริด ระหว่างการให้บริการแบบลำดับความสำคัญและมัลติโปรเซสล์ไพรอริตี้กับงานประยุกต์ที่มีทั้งแบบอิลาสติกและแบบเรียลไทม์รวมกันในระบบ โดยมีจุดประสงค์ เพื่อศึกษาถึงผลของการทำงานเมื่อระบบกริดนั้นประกอบไปด้วยงานทั้งสองแบบอยู่ร่วมกัน โดยที่งานแต่ล่ะแบบนั้นก็จะมีขนาดและความต้องการแบนด์วิดท์ของงานประยุกต์ต่างกัน ซึ่งจะเหมือนระบบงานจริงทั่ว ๆ ไปในปัจจุบัน

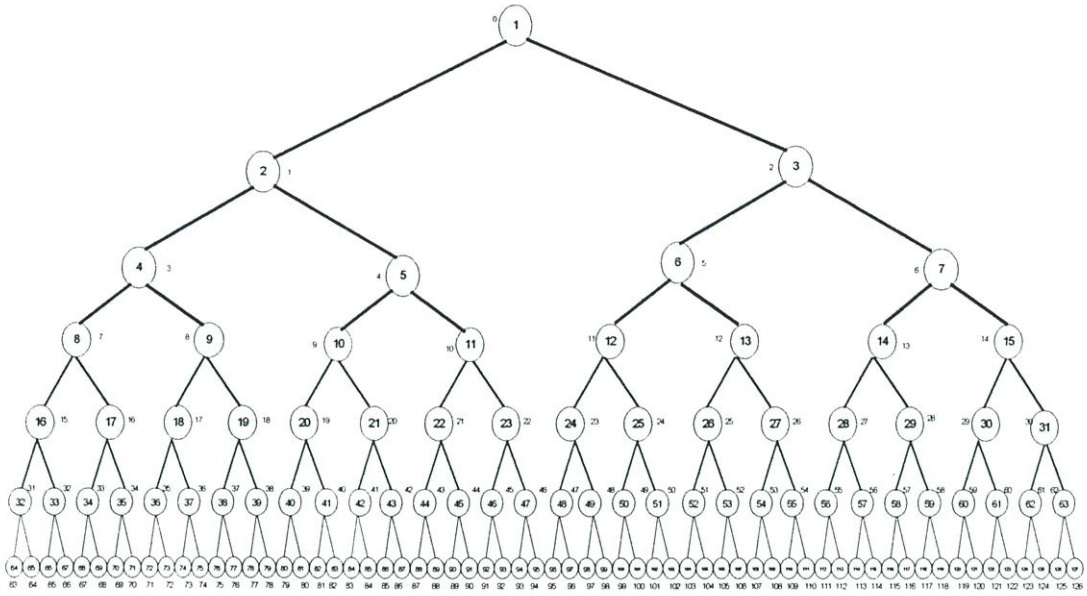
5.2 องค์ประกอบที่ใช้ในการทดลอง

การทดลองเป็นการจำลองระบบกริดที่มีโครงสร้างแบบฮีพ และมีการกำหนดแบนด์วิดท์แบบมีแบนด์วิดท์สูงมาต่ำจากระดับบนลงไปสู่ระดับล่าง ดั้งนั้นองค์ประกอบของระบบที่จะต้องทำการกำหนด ประกอบด้วยจำนวนกริด โครงสร้างของระบบมัลติเอเจนต์ ขนาดของแบนด์วิดท์ และเครื่องมืออย่างอื่นที่นำมาช่วยในการสร้างระบบจำลองการทำงาน เช่น ฐานข้อมูล ที่ใช้เก็บรายละเอียดของกริดและคิวลำดับซึ่งเป็นคิวที่ผ่านการจัดลำดับของลำดับความสำคัญแล้ว

5.2.1 ขนาดของกริดและโครงสร้างของมัลติเอเจนต์

การทดลองจะเริ่มจากการกำหนดการใช้ทรัพยากรของกริด ซึ่งจะกำหนดใช้ทรัพยากรในส่วนที่เกี่ยวข้องกับการรับหรือส่งข้อมูลคือ หน่วยความจำที่ใช้เก็บงานประยุกต์เท่านั้น รายละเอียดของกริดจะจัดถูกเก็บอยู่ในฐานข้อมูลสำหรับจัดเก็บรายละเอียด ซึ่งในการทำงานนั้นทรัพยากรต่างๆ จะถูกควบคุมโดยเอเจนต์อีกที่ การทดลองนี้จะใช้กริดทั้งหมด 127 กริด ซึ่งในที่นี้ไม่กำหนดความสำคัญว่า กริดแต่ละตัวนั้นจะเชื่อมต่อระหว่างกันด้วยเทคโนโลยีหรือโปรโตคอลอะไรแต่จะให้ความสำคัญไปที่การจัดโครงสร้างของเอเจนต์ซึ่งเป็นโครงสร้างที่นำเสนอ

ในการทำงานของเอเจนต์นั้นจะควบคุมการทำงานของกริด โดยเอเจนต์หนึ่งตัวจะควบคุมหนึ่งกริด ดั้งนั้นการทดลองนี้จะใช้เอเจนต์ทั้งหมด127 ตัว โครงสร้างของเอเจนต์แสดงดังรูปที่ 5.1



รูปที่ 5.1 โครงสร้างของระบบเอเจนต์ที่ใช้ในการทดลอง

จากรูปที่ 5.1 แสดงโครงสร้างพื้นฐานของเป็นระบบมัลติเอเจนต์แบบฮีพที่ใช้ในการทดลอง ซึ่งจะเป็นโครงสร้างที่ยังไม่ได้กำหนดการเชื่อต่อแบนด์วิดท์ ประกอบด้วยเอเจนต์ทั้งหมด 127 เอเจนต์ และมีความลึกทั้งหมด 7 ระดับ คือระดับ 0 ถึงระดับ 6 และมีสายเชื่อมโยงทั้งหมด 126 เส้น ภายในเอเจนต์แต่ละตัวจะกำหนดข้อมูลทางสถิติของกริดที่เอเจนต์จะรายงานไปยังส่วนกลาง เช่น ความจุของช่องสื่อสาร ระดับที่อยู่ และขนาดของงานที่มีอยู่ในเอเจนต์แต่ละตัว ซึ่งในการทดลองนี้จะใช้ฐานข้อมูลเป็นตัวจัดเก็บรายละเอียด

5.2.2 ขนาดของแบนด์วิดท์

แบนด์วิดท์เป็นขนาดความจุของกริดในการส่งข้อมูลซึ่ง ในงานวิจัยนี้จะกำหนดลำดับความสำคัญของเอเจนต์ตามความจุของกริดที่เอเจนต์นั้นควบคุมอยู่ โดยที่เอเจนต์ที่ควบคุมกริดที่มีแบนด์วิดท์สูงจะมีลำดับความสำคัญสูงกว่าเอเจนต์ที่ควบคุมกริดที่มีแบนด์วิดท์ต่ำกว่า ซึ่งการเชื่อมต่อจึงสามารถจัดให้อยู่ในรูปของ Sink Link Capacity ตามที่อธิบายมาแล้วในบทที่ผ่านมา

การกำหนดแบนด์วิดท์ในการทดลองเป็นสิ่งที่สำคัญที่ต้องคำนึงถึง ซึ่งในการทดลองนี้จะให้ความสำคัญกับการประยุกต์ใช้งานจริง โดยจะกำหนดให้ครอบคลุมกับแบนด์วิดท์ที่มีใช้ในปัจจุบันเป็นสำคัญ การกำหนดแบนด์วิดท์ในการทดลองจะกำหนดตั้งแต่ 64 Kbps ซึ่งเป็นความเร็วต่ำสุดที่มีให้บริการของระบบอินเตอร์เน็ต ถึง 512 Mbps ซึ่งแสดงดังตารางที่ 5.2 และ 5.3

ตารางที่ 5.1 แบนด์วิดท์ของกริดที่ใช้ในการทดลองชุดที่ 1 จาก 64 Kbps ถึง 4 Mbps

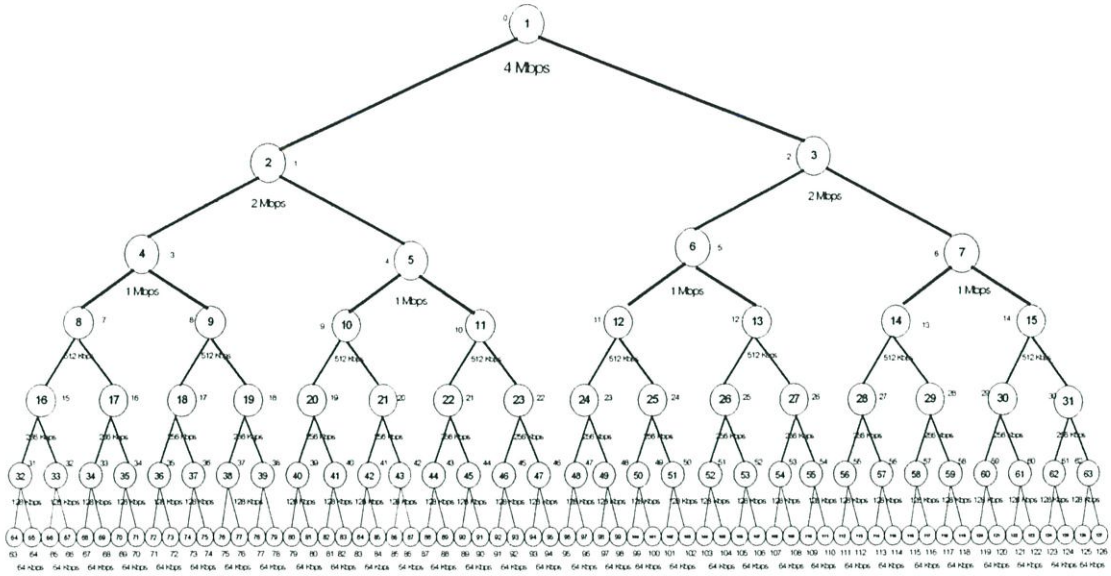
ระดับ	ขนาดของ แบนด์วิดท์	จำนวนกริด ที่กำหนด
0	4 Mbps	1
1	2 Mbps	2
2	1 Mbps	4
3	512 Kbps	8
4	256 Kbps	16
5	128 Kbps	32
6	64 Kbps	64

ตารางที่ 5.2 แบนด์วิดท์ของกริดที่ใช้ในการทดลองชุดที่ 2 จาก 8 Mbps ถึง 512 Mbps

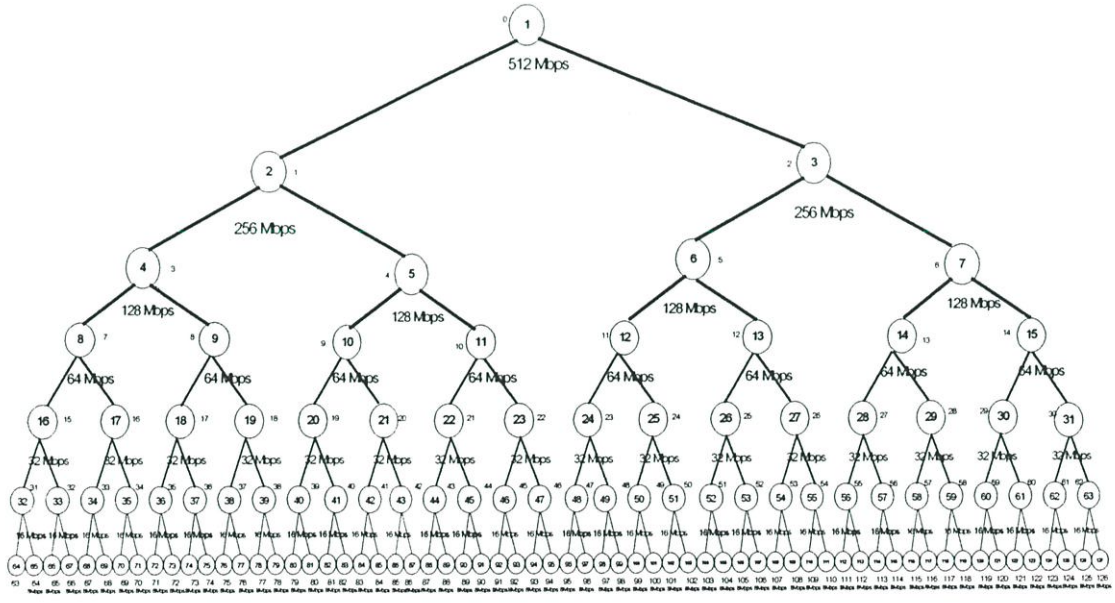
ระดับ	ขนาดของ แบนด์วิดท์	จำนวนกริด ที่กำหนด
0	512 Mbps	1
1	256 Mbps	2
2	128 Mbps	4
3	64 Mbps	8
4	32 Mbps	16
5	16 Mbps	32
6	8 Mbps	64

ในการทดลองจะแบ่งแบนด์วิดท์ของกริดออกเป็น 2 ชุด ชุดแรกเริ่มจาก 64 Kbps ถึง 4 Mbps และชุดที่สองเริ่มจาก 8 Mbps ถึง 512 Mbps เพื่อให้สอดคล้องกับหลักการของการเชื่อมต่อความจุแบบต่ำลง ในการกำหนดและการให้บริการแบบลำดับความสำคัญตามโครงสร้างของฮีพจึง

กำหนดให้กริดที่อยู่ระดับสูงกว่ามีแบนด์วิดท์สูงกว่ากริดที่อยู่ในระดับล่างซึ่งตัวอย่างแสดงดังรูปที่ 5.2 และ 5.3



รูปที่ 5.2 ตัวอย่างในการกำหนดแบนด์วิดท์ตามหลักการของการเชื่อมต่อความจุแบบต่ำลง สำหรับข้อมูลชุดที่ 1 ที่ใช้ในการทดลอง



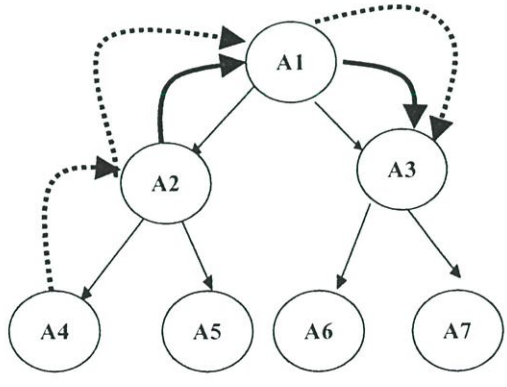
รูปที่ 5.3 ตัวอย่างในการกำหนดแบนด์วิดท์ตามหลักการของการเชื่อมต่อความจุแบบต่ำลง สำหรับข้อมูลชุดที่ 2 ที่ใช้ในการทดลอง

5.3 หลักการทดลองและการวัดประสิทธิภาพ

จุดประสงค์ในการทดลองนี้เพื่อเป็นการทดสอบประสิทธิภาพการทำงานระหว่าง การให้บริการแบบลำดับความสำคัญและมัลติโปรเซสส์ไพรอริตี้อัลกอริทึม เพื่อประยุกต์ใช้กับการควบคุมการส่งข้อมูลภายในเครือข่ายของกริด ซึ่งกริดมีการจัดโครงสร้างการส่งข้อมูลเป็นแบบฮีพ ดังนั้นในการทดลองนี้จะให้ความสำคัญในเรื่องของเวลาในการส่งข้อมูลในเวลาใดเวลาหนึ่งของกริดตามโครงสร้างที่ 5.1 นั้นก็หมายความว่ากริดทุกตัวในโครงสร้างนั้นต้องการสงข้อมูลในเวลาเดียวกัน และนอกจากนี้งานวิจัยนี้ไม่ให้ความสำคัญในส่วนของการค้นหาข้อมูลในคิว แต่จะเน้นไปที่การตรวจสอบเส้นทางการส่งข้อมูลและขนาดของข้อมูลที่จะส่งเป็นสำคัญ

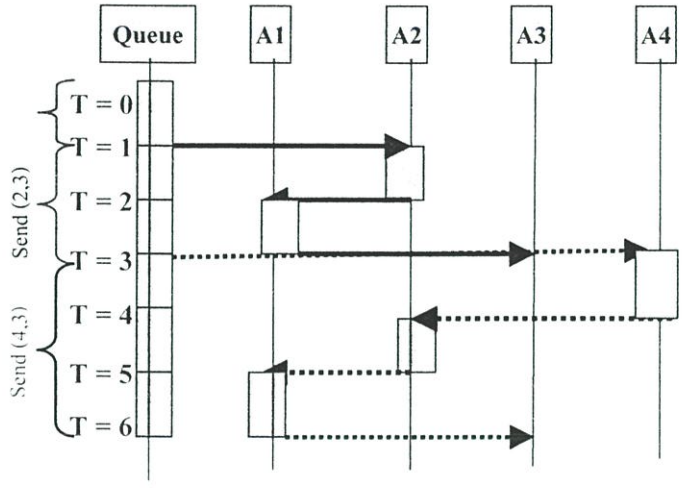
5.3.1 การวัดประสิทธิภาพ

ในการวัดค่าของการให้บริการแบบลำดับความสำคัญนั้น จะวัดจากผลรวมของระยะเวลาการส่งข้อมูลของเอเจนต์แต่ละตัวตั้งแต่ต้นจนจบตามที่มีอยู่ในคิวลำดับ แต่ในส่วนของมัลติโปรเซสส์ไพรออริตี้จะวัดแบบนั้นไม่ได้ เนื่องจากว่าการส่งนั้นจะไม่ส่งทีละตัวแต่จะขึ้นอยู่กับว่ามีความจุในการส่งขณะนั้นเพียงพอหรือไม่ ซึ่งในเวลาใดเวลาหนึ่งอาจจะมีงานจากหลายเอเจนต์ส่งอยู่พร้อม ๆ กัน ดังนั้นการวัดค่าจะวัดจากเวลาสูงสุดหรือเอเจนต์ที่ส่งช้าที่สุดของเอเจนต์ตัวใดตัวหนึ่งที่ส่งข้อมูลออกไปกับค่าเวลาที่ค้นหาข้อมูลของฮีพในคิวที่มีการร้องขอส่งข้อมูลจากเอเจนต์ เป็นพารามิเตอร์ในการทดสอบ จากรูปที่ 5.4 แสดงการส่งข้อมูลสองชุดระหว่าง A2 ไป A3 และ A4 ไป A3



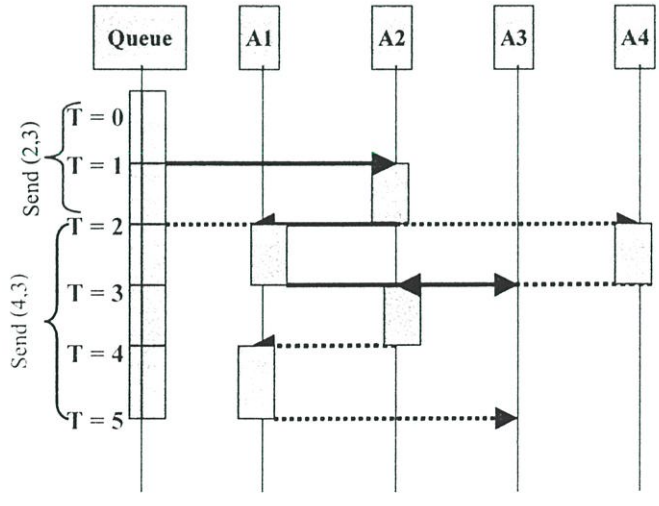
รูปที่ 5.4 ตัวอย่างการส่งข้อมูลสองชุดระหว่าง A2 ไป A3 และ A4 ไป A3

จากรูปที่ 5.4 การส่งข้อมูลจาก A2 ไป A3 (เส้นทึบ) จะได้รับสิทธิ์การส่งข้อมูลก่อนเนื่องจาก A2 มีลำดับความสำคัญสูงกว่า A4 ลำดับการส่งข้อมูลของการให้บริการแบบลำดับความสำคัญแสดงไดอะแกรมลำดับ (Sequence Diagram) ดังรูปที่ 5.5



รูปที่ 5.5 แสดงไดอะแกรมลำดับของการส่งข้อมูลตามหลักการบริการแบบลำดับความสำคัญ

ตามโครงสร้างที่ 5.4 เมื่อเราทำงานตามหลักการของมัลติโปรเซสลีไพรออริตี้อัลกอริทึมจะแสดงไดอะแกรมลำดับการทำงานได้ดังรูปที่ 5.6



รูปที่ 5.6 แสดงไดอะแกรมลำดับของการส่งข้อมูลตามหลักของมัลติโปรเซสลีไพรออริตี้

จากไดอะแกรมตามรูปที่ 5.5 การทำงานตามของหลักของการให้บริการแบบลำดับความสำคัญนั้น จะเริ่มจากการค้นหากริดที่มีลำดับความสำคัญสูงสุดในคิวก่อน ซึ่งวิธีการค้นหานั้นจะไม่ใช้ความสำคัญ แต่จะให้ความสำคัญหลังค้นหาข้อมูลตามลำดับความสำคัญ และพร้อมที่จะส่งไปตามโครงสร้างที่กำหนดแล้ว จากไดอะแกรมเมื่อกำหนดให้เวลาเป็น 1 หน่วย จะเห็นว่า T0 นั้นคือเวลาเริ่มต้นที่ต้องการค้นหาข้อมูล T1, T2 และ T3 คือเวลาในการส่งข้อมูลชุดแรกคือ A2 ไป A3 และ T3, T4,T 5 และ T6 คือเวลาส่งข้อมูลชุดที่สองดังนั้นเวลารวมคือ T0 + เวลาส่งข้อมูลชุดแรก + เวลาส่งข้อมูลชุดที่สอง ซึ่งจะเห็นว่าจะไม่รวมกับเวลาค้นหาข้อมูลในลำดับถัดไป โดยเวลานี้จะขึ้นอยู่กับเทคโนโลยีการค้นข้อมูลของฮีพ ในรูปที่ 5.6 นั้นจะจะแตกต่างจากจากรูปที่ 5.5 ตรงที่หลังจากส่งข้อมูลชุกแรกแล้วเราสมารถส่งข้อมูลชุดที่สองตามไปได้เลย แต่จะเสียเวลาตรงที่ค้นหาและตรวจสอบเส้นทางว่าสามารถส่งข้อมูลไปที่สถานีที่เรียกมาได้หรื่อไม่หรือส่งไปที่ใด ซึ่งจะเห็นว่า T0 คือเวลาในการค้นหาข้อมูลชุดแรกที่มีลำดับความสำคัญสูงสุดส่วน T1, T2, และ T3 คือเวลาในการส่งข้อมูลชุดแรกและ T2, T3, T4 และ T5 คือเวลาในการส่งข้อมูลชุดที่สองซึ่งจะเห็นได้ว่าจะเป็นการรวมเวลาในการค้นหาและเวลาสูงสุดในการส่งข้อมูลเสร็จสิ้นชุดสุดท้ายนั้นเอง

จากที่อธิบายมากำหนดให้

v = เวลาค้นหาเอเจนต์และตรวจสอบเส้นทางการส่งในคิวลำดับ
T = ผลรวมของเวลาทั้งหมดในการทำงาน
at = เวลาในการส่งข้อมูลของเอเจนต์แต่ละตัว
n = จำนวนการร้องขอส่งข้อมูลเอเจนต์ทั้งหมด

สูตรการคำนวณเวลาสูงสุดของการให้บริการแบบลำดับความสำคัญหาได้จาก

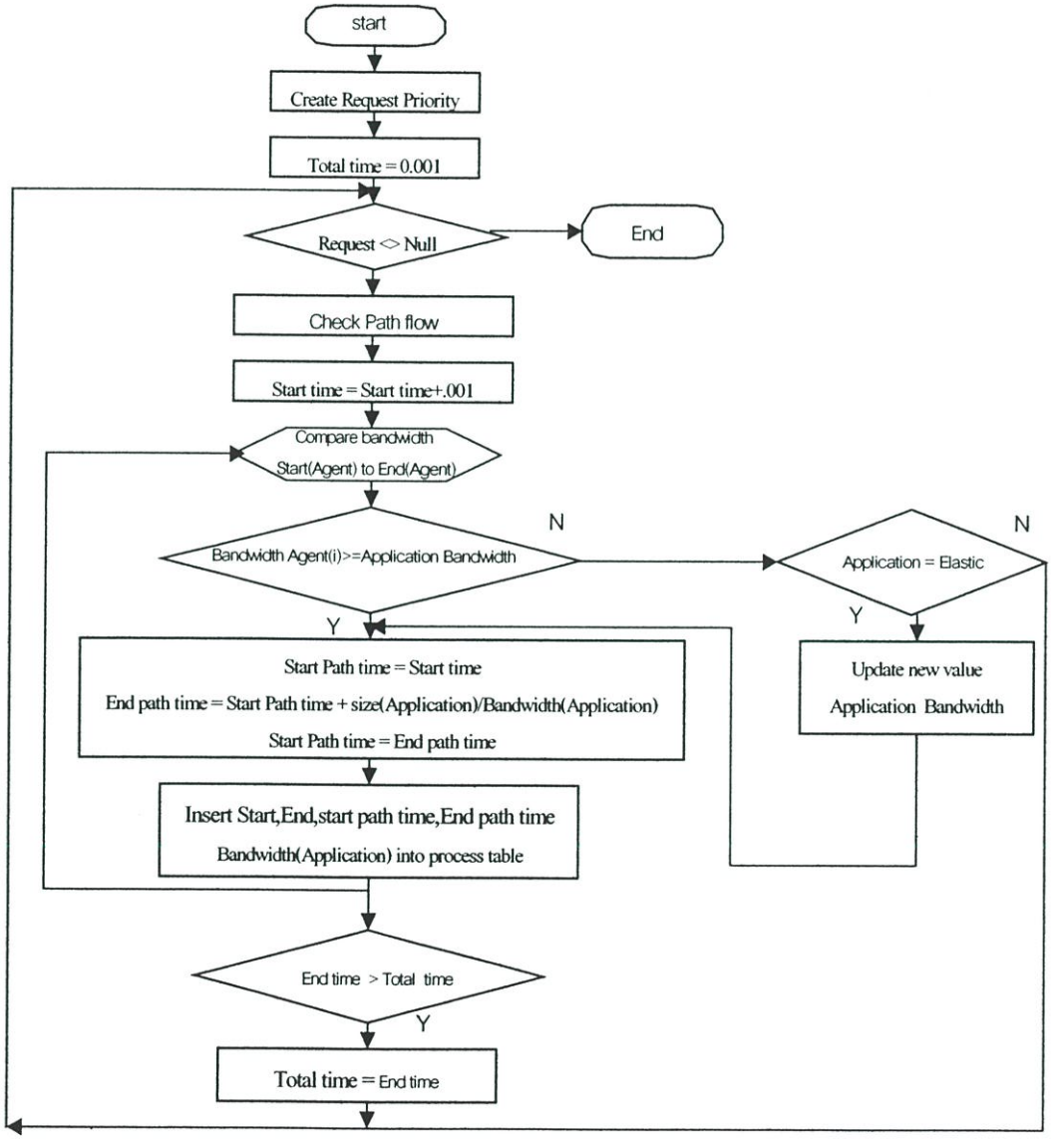
$$T = v_0 + \sum_{i=0}^{n-1} at\ (i) \qquad (5.1)$$

สูตรการคำนวณเวลาสูงสุดของมัลติโปรเซสส์ไพรออริตี้คิวหาได้จาก

$$T = \sum_{i=0}^{n-1} v\ (i) + at_{Max} \qquad (5.2)$$

5.3.2 วิธีการในการจำลองเพื่อหาประสิทธิภาพ

ในการทดลองนี้จะนำหลักการของฐานข้อมูลเข้ามาช่วยเพื่อให้ง่ายต่อการเขียนโปรแกรม เนื่องจากเราไม่ได้ให้ความสำคัญในการค้นหาเอเจนต์ในคิวลำดับมากนัก จะกำหนดเวลาในการค้นหาเป็นค่าคงที่คือ 0.001 วินาทีสำหรับวิธีการทดลอง การทำงานของ มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมแสดงตามรูปโฟว์ลชาร์ทดังรูปที่ 5.7



รูปที่ 5.7 โฟล์วชาร์ทแสดงวิธีการที่ใช้ในการทดลองวัดประสิทธิภาพสำหรับมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม

การทดลองเริ่มจากการสร้างข้อมูลที่ใช้ในการทดสอบโดยจะสร้างจากเอเจนต์ทั้งหมด 127 ตัวทำการส่งข้อมูลแบบผลคูณคาร์เทเซียน(Cartesian product) คือส่งแบบพบกันหมดในโครงสร้างยกเว้นตัวเอง โดยการสร้างข้อมูลจะค่อยเพิ่มขนาดให้มีมากขึ้นเรื่อย ๆ กระจายให้ครอบคลุมทั่วในโครงสร้าง จากนั้นจะทำการทดลองตามหลักการของอัลกอริทึมตามที่อธิบายไว้ในบทที่ 4

เมื่อได้ข้อมูลในการส่งจากคิวของฮีพเรียบร้อยแล้ว อัลกอริทึมจะตรวจสอบเส้นทางการส่งข้อมูล โดยจะตรวจสอบว่าเวลาปัจจุบันนั้นเอเจนต์มีข้อมูลอะไรบ้างส่งผ่านและมีขนาดแบนด์วิดท์เหลือเท่าใดในการทดลองนี้จะเก็บรายละเอียดต่าง ๆ เช่น เอเจนต์ที่ส่งและรับข้อมูล, ขนาดของแบนด์วิดท์ที่ส่งในขณะนั้นและเวลาที่ส่งข้อมูลผ่านเอเจนต์ คือเวลาเริ่มต้นและสิ้นสุดในการส่ง ไว้ที่ตารางการทำงาน (Process table) เมื่อต้องการทราบว่าเวลาใดเวลาหนึ่งเอเจต์มีแบนด์วิดท์เหลือเท่าไหรก็สามารถคำนวณเอาจากตารางนี้ได้

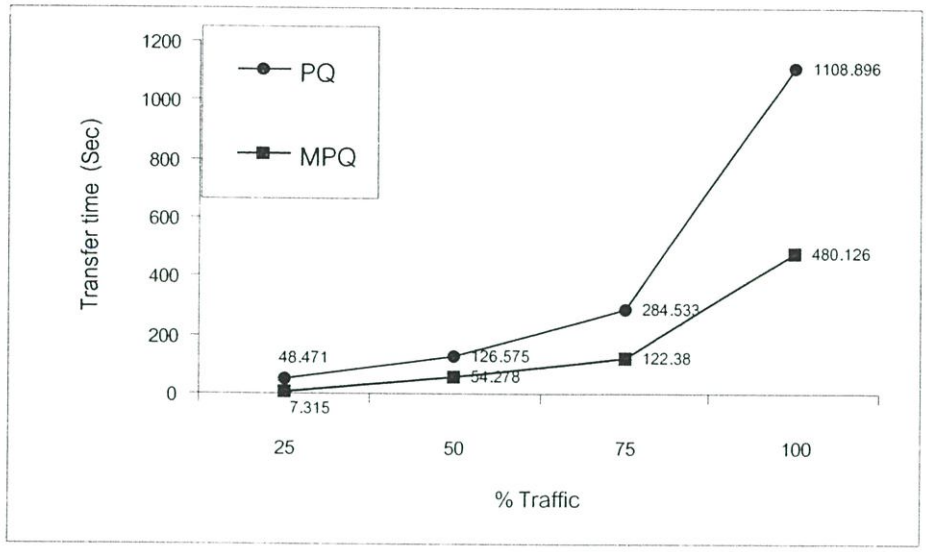
5.4 การทดลองกับงานประยุกต์แบบอิลาสติก (Elastic)

งานแบบอิลาสติกเป็นงานที่ต้องการแบนด์วิดท์ที่ไม่จำกัด คือยิ่งมากยิ่งดีเพราะจะไม่มีผลกระทบต่อการประมวลผลของงานแต่จะมีผลต่อการส่งเร็วหรือช้าลงเท่านั้น ดังนั้นแบนด์วิดท์ที่ใช้ในการทดลองสามารถยืดหยุ่นได้ ขนาดข้อมูลที่ใช้ทดลองกับงานอิลาสติกแสดงดังตารางที่ 5.3

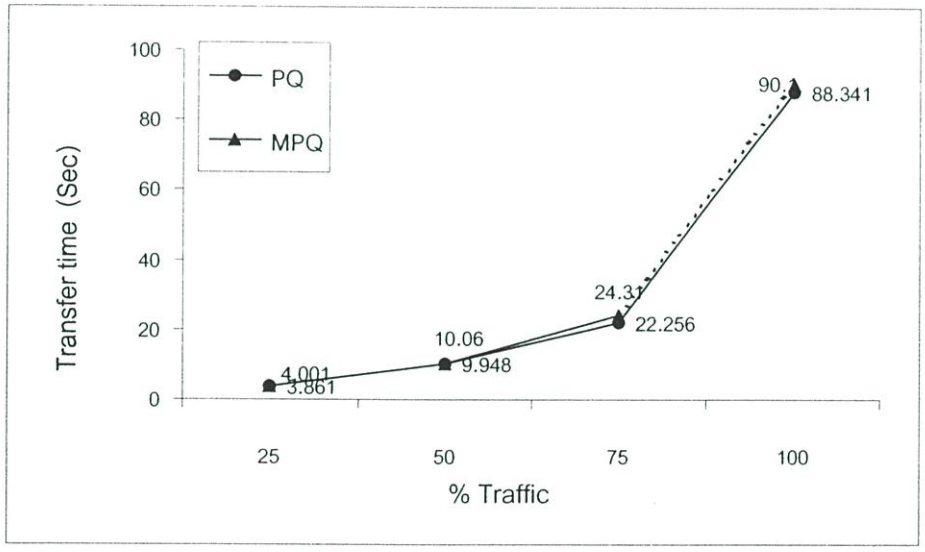
ตารางที่ 5.3 ขนาดของข้อมูลแบบอิลาสติกที่ใช้ในการทดลอง

ลำดับ	ขนาดของข้อมูล	แบนด์วิดท์ที่ต้องการ
1	1 Kbytes	1 Kbps
2	10 Kbytes	10 Kbps
3	50 Kbytes	50 Kbps
4	100 Kbytes	100 Kbps
5	500 Kbytes	500 Kbps
6	1 Mbytes	1 Mbps

และความจุในการทดลองทั้งสองแบบตามตารางที่ 5.1 และ 5.2 จากการทดลองเมื่อเทียบเวลาส่งข้อมูลตามหลักการของการบริการแบบลำดับความสำคัญปกติ (PQ) และมัลติโปรเซสส์ไพรออริตี้คิวอัลกอริทึม (MPQ) จะได้ผลตั้งแต่รูปที่ 5.8 ถึงรูปที่ 5.19

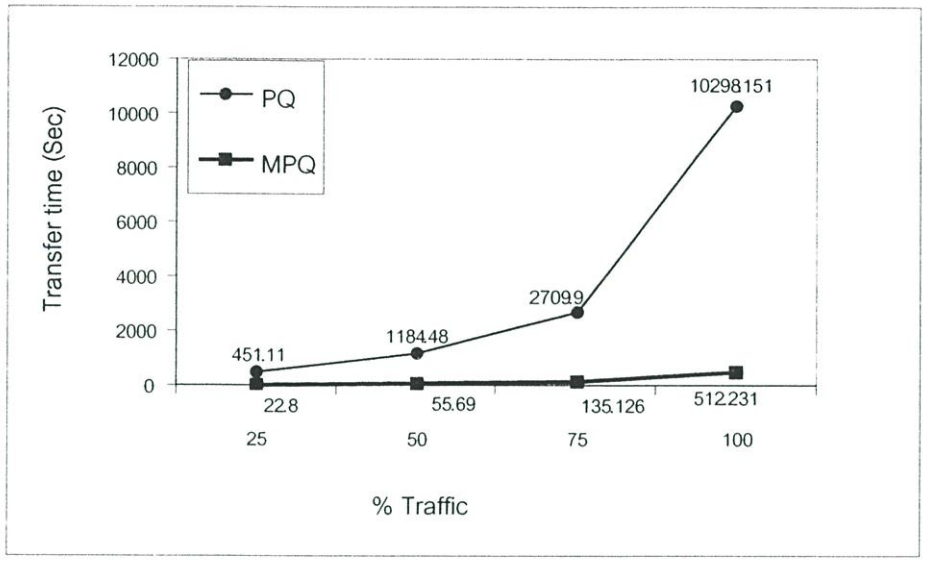


รูปที่ 5.8 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Kbytes กับแบนด์วิดท์ชุดที่ 1

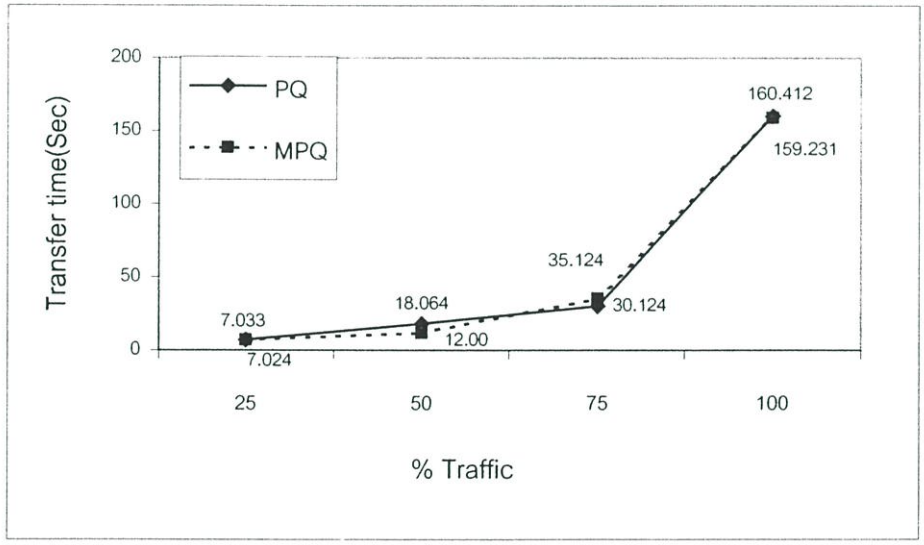


รูปที่ 5.9 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Kbytes กับแบนด์วิดท์มูลชุดที่ 2

จากรูปที่ 5.8 เป็นการทดลองข้อมูลขนาด 1 Kbytes กับแบนด์วิดท์ชุดที่ 1 จะเห็นความแตกต่างได้ชัดระหว่างการให้บริการแบบลำดับความสำคัญ และมัลติโปรเซสส์ไพรออริตี้คิวอัลกอริทึม (MPQ) ส่วนรูปที่ 5.9 เป็นการใช้ข้อมูล 1 Kbytes กับแบนด์วิดท์ชุดที่สองซึ่งสูงกว่าชุดแรก ผลของเวลาที่ได้จะไม่แตกต่างกันมากนักสำหรับการทดลองระหว่างการให้บริการแบบไพรออริตี้และมัลติโปรเซสส์ไพรออริตี้คิวอัลกอริทึม

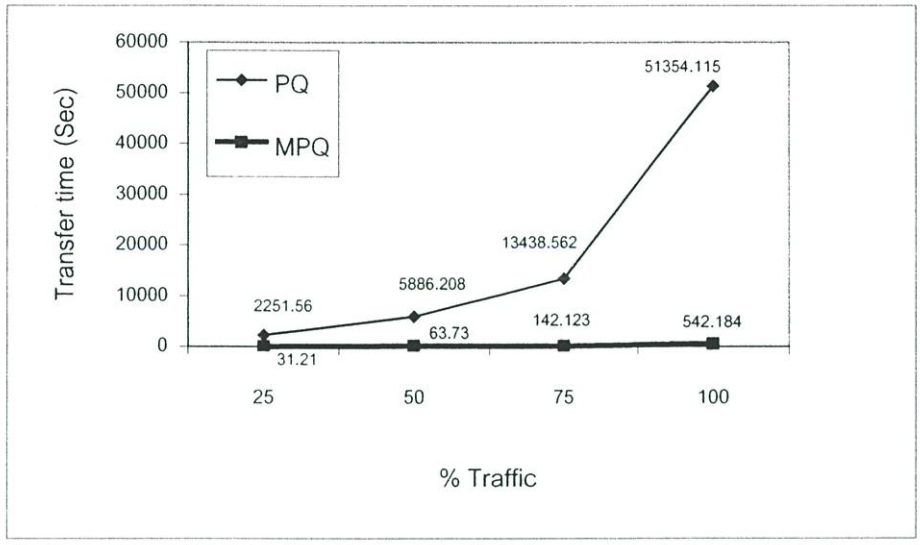


รูปที่ 5.10 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 10 Kbytes กับแบนด์วิดท์ชุดที่ 1

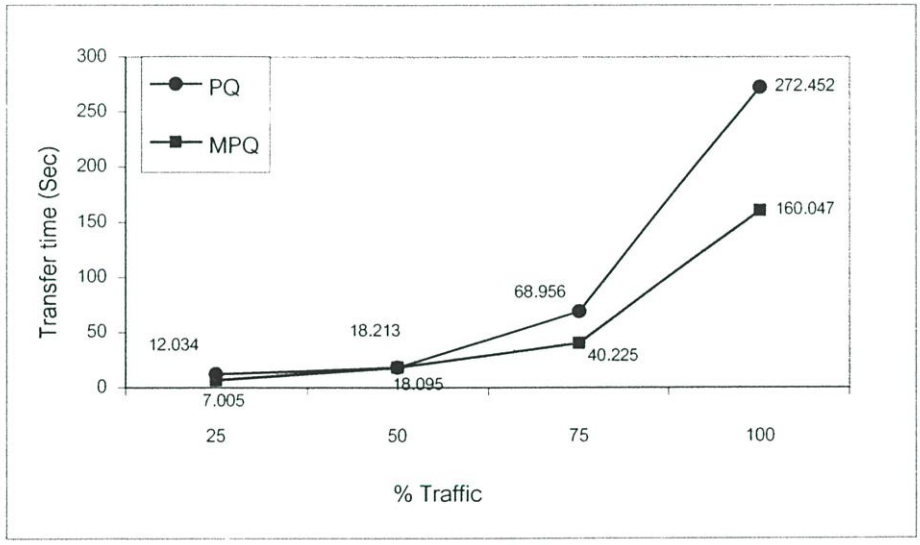


รูปที่ 5.11 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 10 Kbytes กับแบนด์วิดท์ชุดที่ 2

จากรูปที่ 5.10 เป็นการทดลองข้อมูลขนาด 10 Kbytes กับข้อมูลชุดแรกซึ่งมีแบนด์วิดท์ต่ำ จะเห็นความแตกต่างได้ชัดระหว่างการการให้บริการแบบลำดับความสำคัญ และมัลติโปรเซสส์ ไพรออริตี้อัลกอริทึม ส่วนรูปที่ 5.11 เป็นการใช้ข้อมูล10 Kbps ทดสอบกับข้อมูลชุดที่ 2 ผลของเวลาที่ได้จะไม่แตกต่างกันมาก เมื่อเปรียบเทียบการทดลองระหว่าง 1 Kbytes และ 10 Kbytes ผลที่ได้จะแตกต่างกันไม่มาก

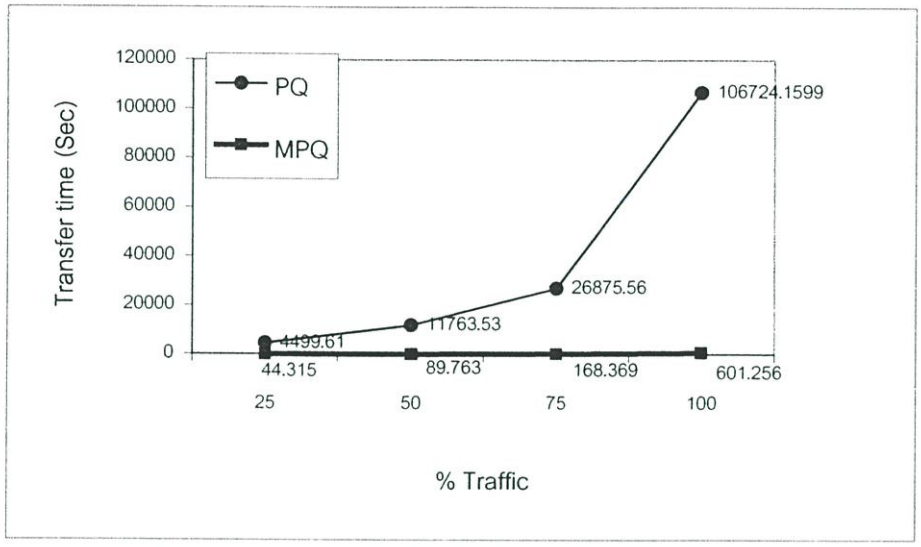


รูปที่ 5.12 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 50 Kbytes กับแบนด์วิดท์ชุดที่ 1

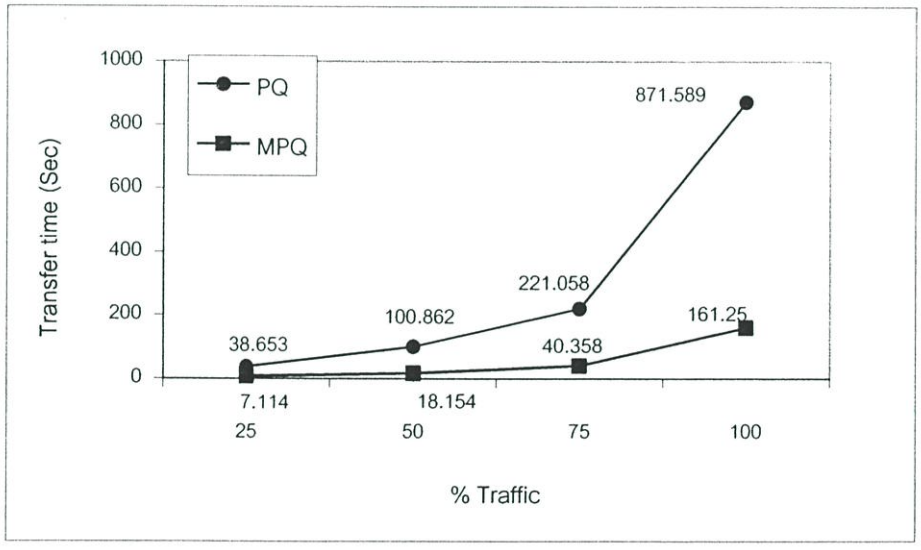


รูปที่ 5.13 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 50 Kbytes กับแบนด์วิดท์ชุดที่ 2

จากรูปที่ 5.12 เป็นการทดลองข้อมูลขนาด 50 Kbytes กับข้อมูลชุดแรกจะเห็นความแตกต่างจาก 10 Kbps ได้ชัดเจนมากขึ้น ส่วนรูปที่ 5.13 เป็นการใช้ข้อมูล 50 Kbytes ทดสอบกับข้อมูลชุดที่ 2 ผลของเวลาที่ได้จะเริ่มแตกต่างกันมาก ซึ่งจากรูปจะเห็นว่าผลการทดลองจะเห็นได้ชัดเจนเมื่อขนาดของข่อมูล 50 Kbytes และมีความหนาแน่น 50% ในระบบ

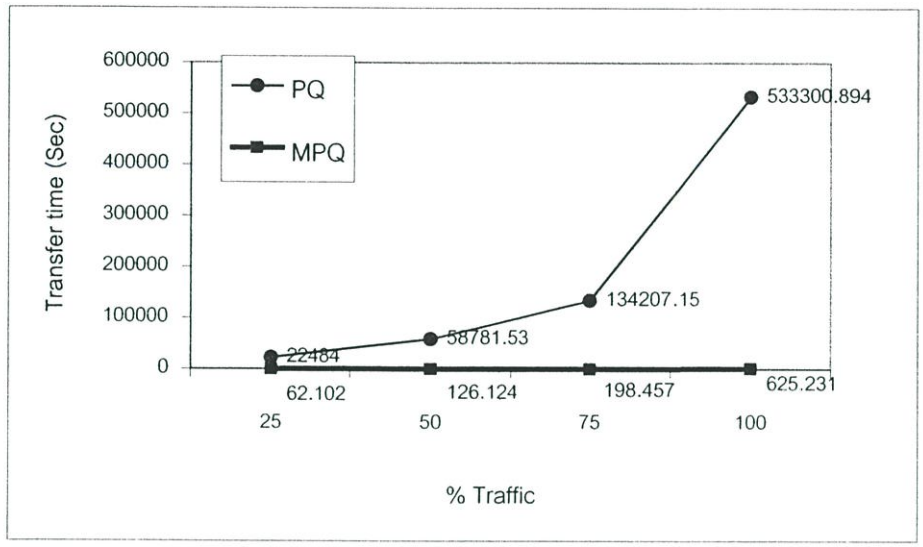


รูปที่ 5.14 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 100 Kbytes กับแบนด์วิดท์ชุดที่ 1

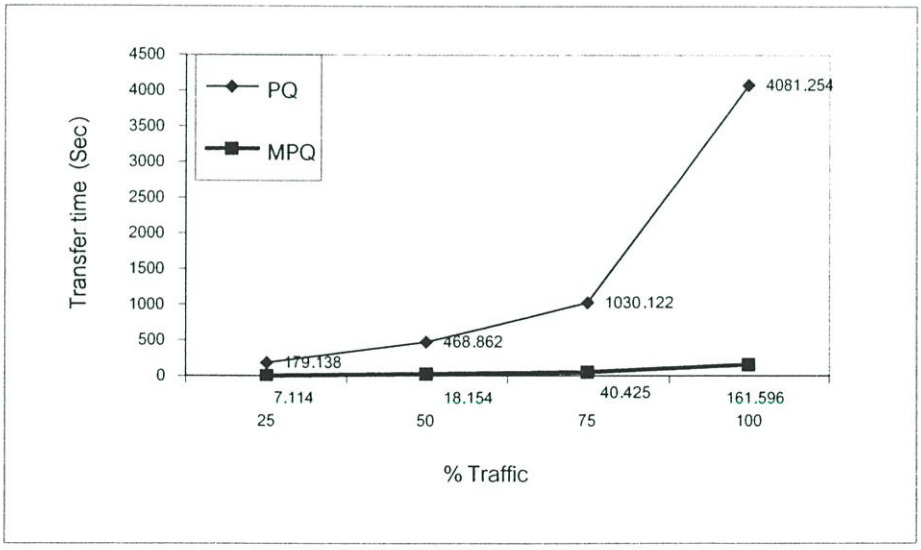


รูปที่ 5.15 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 100 Kbytes กับแบนด์วิดท์ชุดที่ 2

จากรูปที่ 5.14 เป็นการทดลองข้อมูลขนาด 100 Kbytes กับข้อมูลชุดแรกจากกราฟจะเห็นความแตกต่างจากได้ชัดเจนมากขึ้น ส่วนรูปที่ 5.15 เป็นการใช้ข้อมูล 100 Kbytes ทดสอบกับข้อมูลชุดที่ 2 ผลของเวลาที่ได้จะเริ่มแตกต่างกันมากกว่าผลของการทดลองกับข้อมูลขนาด 50 Kbytes

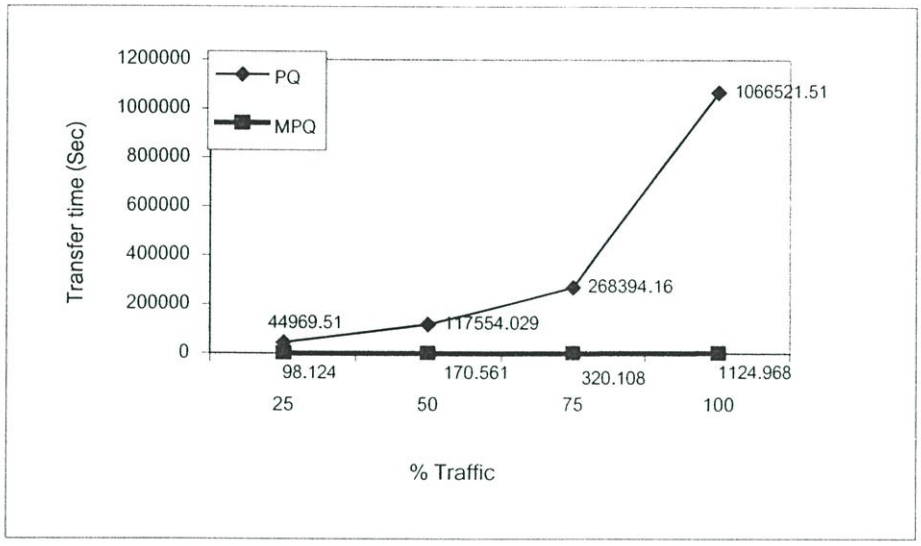


รูปที่ 5.16 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 500 Kbytes กับแบนด์วิดท์ชุดที่ 1

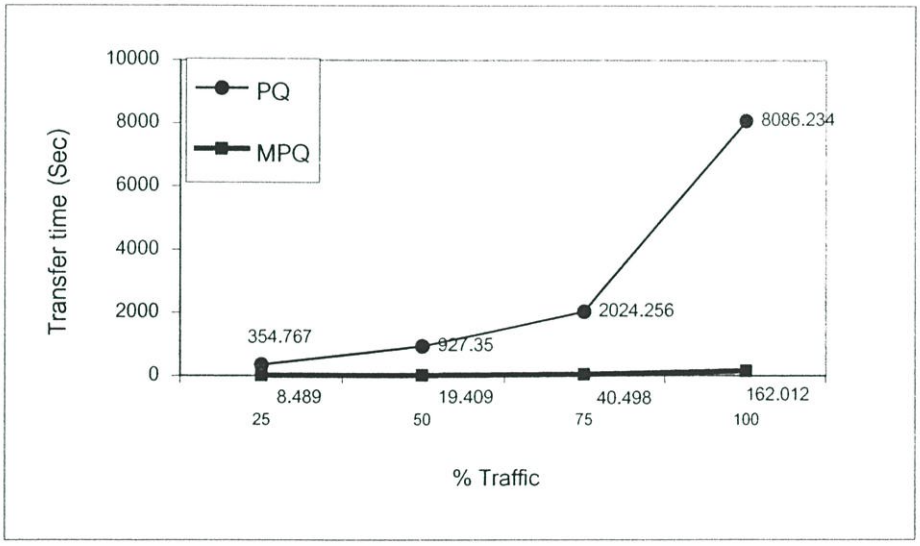


รูปที่ 5.17 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 500 Kbytes กับแบนด์วิดท์ชุดที่ 2

จากรูปที่ 5.16 เป็นการทดลองข้อมูลขนาด 500 Kbytes กับข้อมูลชุดแรกจากกราฟจะเห็นความแตกต่างจากข้อมูล 100 Kbytes ได้ชัดเจนมาก ส่วนรูปที่ 5.17 เป็นการใช้ข้อมูล 500 Kbytes ทดสอบกับข้อมูลชุดที่ 2 ผลของเวลาที่ได้จะแตกต่างกันมาก



รูปที่ 5.18 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Mbytes กับแบนด์วิดท์ชุดที่ 1



รูปที่ 5.19 ผลของการทดลองเปรียบเทียบของข้อมูลขนาด 1 Mbytes กับแบนด์วิดท์ชุดที่ 2

จากรูปที่ 5.8 ถึง 5.19 เมื่อเปรียบเทียบระหว่างแบนด์วิดท์จะเห็นว่าถ้าขนาดของแบนด์วิดต่ำแล้วการใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมจะให้ประสิทธิภาพที่ดีกว่า ซึ่งจะเห็นได้จากอัตราความแตกต่างของเวลาทั้งสองอัลกอริทึมระหว่างแบนด์วิดชุดที่ 1 และชุดที่ 2 เมื่อเปรียบเทียบกับขนาดของข้อมูลจะเห็นว่าอัตราการเพิ่มของเวลาของการให้บริการแบบลำดับความสำคัญจะมีมากในส่านของมัลติโปรเซสส์ไพรออริตี้จะมีอัตราการเพิ่มของเวลาน้อยกว่าเช่นในแบนด์วิดท์ชุดที่ 1 ที่ความหนาแน่นของทราฟฟิค 25 % ของข้อมูล 1 Kbytes,10 Kbytes , 50 Kbytes, 100 Kbytes,

500 Kbytes และ 1 Mbytes การเพิ่มของเวลาของการให้บริการแบบลำดับความสำคัญคือ 8.471, 451.11, 2251.56, 4499.61, 22484.00 และ 44969.51 วินาทีตามลำดับ ในส่วนของมัลติโปรเซสส์ไพรออริตี้คือ 7.351, 22.80, 31.21, 44.315, 62.102, 98.124 วินาทีตามลำดับ ซึ่งจะเห็นว่าผลต่างของเวลาจะน้อยกว่า

5.5 การทดลองกับงานประยุกต์แบบเรียลไทม์ (Real Time)

งานประยุกต์แบบเรียลไทม์นั้นต่างจากงานแบบอิลาสติก ซึ่งต้องมีแบนด์วิดท์ที่เพียงพอส่วนใหญ่ข้อมูลจะส่งแบบสทรีม (Stream) ถ้าขาดช่วงหรือส่งข้อมูลมาไม่ต่อเนื่อง จะมีผลต่อการใช้งานของระบบ เช่น ถ้าเป็น Audio ก็จะมีเสียงหายเป็นบางช่วงหรือถ้าเป็น Video ภาพหรือเสียงก็จะกระตุกหรือขาดหายไป

เมื่อเปรียบเทียบการทำงานระหว่างอัลกอริทึมทั้งสอง การให้บริการแบบลำดับความสำคัญจะมีผลกระทบด้านเวลาตอบสนองนั้นดีกว่า แต่เมื่อมองถึงประสิทธิภาพการทำงานของแต่ละเส้นทาง (Flow) จะไม่เกิดประโยชน์มากนักเพราะเป็นการส่งงานไปทีละงานซึ่งในความเป็นจริงแล้วงานแบบเรียลไทม์นั้นไม่จำเป็นต้องใช้แบนด์วิดท์มาก แต่ขอให้มีแบนด์วิดท์เพียงพอและต่อเนื่องในการส่งสทรีมข้อมูลของงานแบบเรียลไทม์ หลักการของมัลติโปรเซสส์อัลกอริทึมในการใช้งานกับระบบเรียลไทม์ จะกำหนดขนาดแบนด์วิดท์ของงานให้เพียงพอต่อการส่ง ในการทดลองนี้จะเลือกงานแบบเรียลไทม์ 4 ประเภทเพื่อนำมาใช้ในการทดสอบแสดงดังตารางที่ 5.4

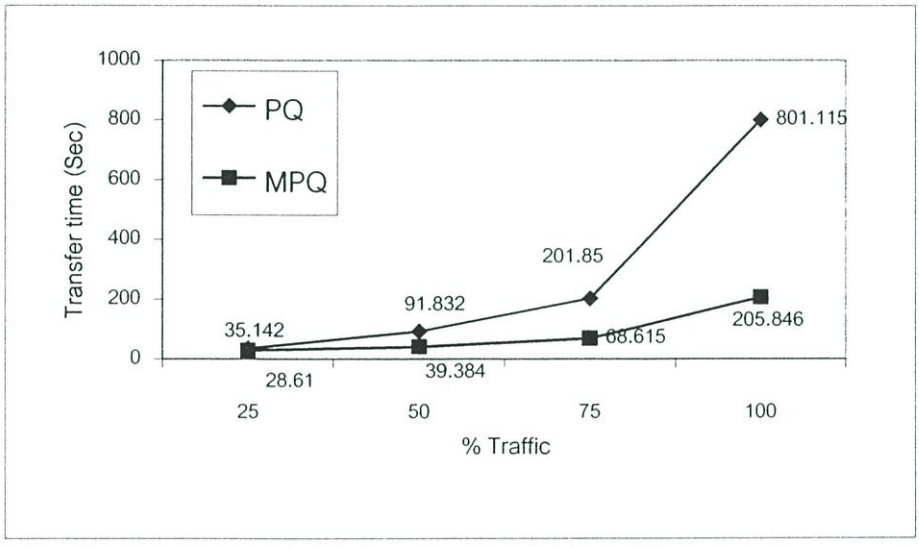
ตารางที่ 5.4 ชนิดของงานประยุกต์แบบเรียลไทม์และแบนด์วิดท์ที่ต้องการที่ใช้ในการทดลอง

งานประยุกต์	ขนาดข้อมูลที่ใช้ทดลอง	แบนด์วิดท์ที่ต้องการ
Internet Voice PCM	100 Kbytes,500 Kbytes,1 Mbytes, 10 Mbytes	64 Kbps
Audio MP3	100 Kbytes,500 Kbytes,1 Mbytes, 10 Mbytes	128 Kbps
Real time Internet CD Player	100 Kbytes,500 Kbytes,1 Mbytes, 10 Mbytes	750 Kbps
Video Conference	100 Kbytes,500 Kbytes,1 Mbytes, 10 Mbytes	1 Mbps

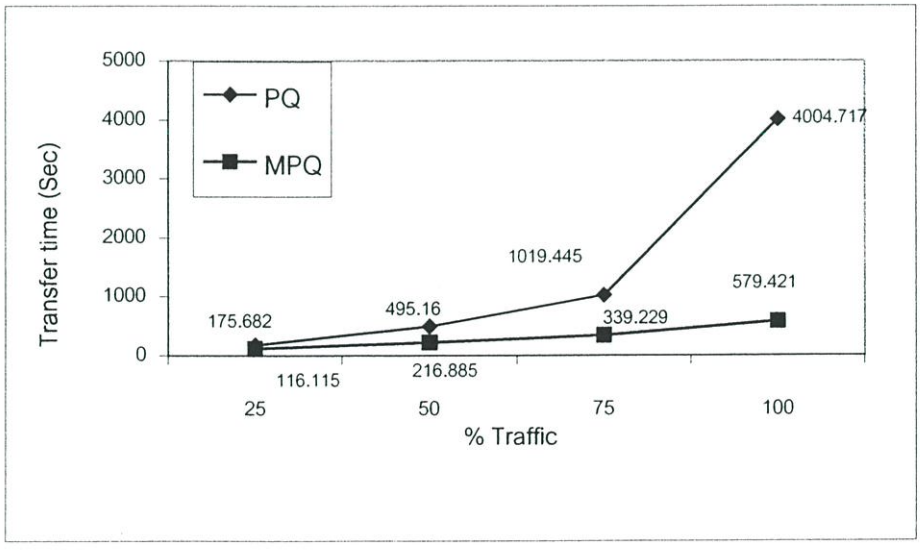
จากตารางข้างบนแสดงถึงงานแต่ละแบบที่นำมาใช้ในการทดลอง ซึ่งค่าแบนด์วิดท์ของแต่ละงานนั้นอ้างอิงตาม [33] จุดประสงค์ที่เลือกงานประยุกต์ตามตารางข้างบน เนื่องจากว่าความต้องการแบนด์วิดท์นั้นมีค่าเป็นลำดับและไม่ห่างกันมากนัก สำหรับการทดลองนั้นจะทดลองใช้งานแต่ละแบบกับขนาดของข้อมูลที่ต่างกัน 100 Kbytes, 500 Kbytes, 1 Mbytes, 10 Mbytes

ตามลำดับ ในการทดลองจะเลือกใช้แบนด์วิดท์ชุดที่ 2 และทำการทดลองกับความหนาแน่นของทราฟฟิคที่ต่างกัน ผลการทดลองแสดงตามรูป 5.20 ถึง 5.35

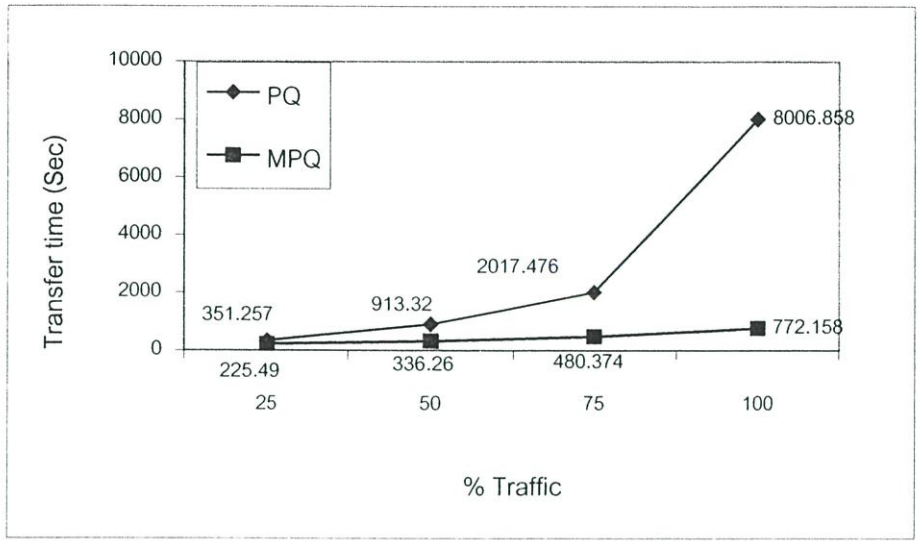
จากรูปที่ 5.20 ถึง 5.23 เป็นการทดลองงาน Internet Voice PCM โดยงานประยุกต์นี้ต้องใช้แบนด์วิดท์ 64 Kbps เป็นอย่างต่ำจึงจะสามารถส่งข้อมูลได้อย่างไม่เป็นปัญหา โดยรูปที่ 5.20 เป็นการทดลองกับ Internet Voice PCM ขนาด 100 Kbytes รูปที่ 5.21 ทดลองกับ Internet Voice PCM ขนาด 500 Kbytes รูปที่ 5.22 ทดลองกับ Internet Voice PCM ขนาด 1 Mbytes และรูปที่ 5.23 ทดลองกับ Internet Voice PCM ขนาด 10 Mbytes



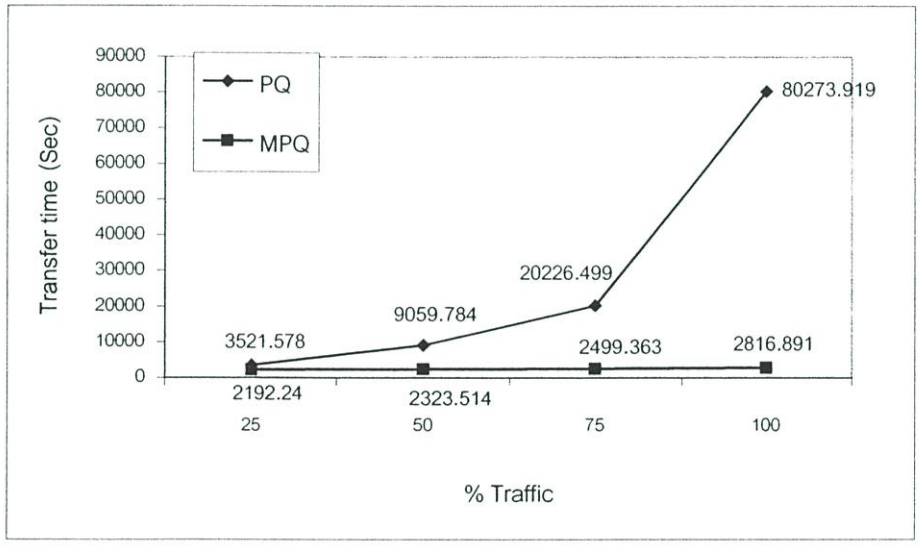
รูปที่ 5.20 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 100 Kbytes



รูปที่ 5.21 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 500 Kbytes



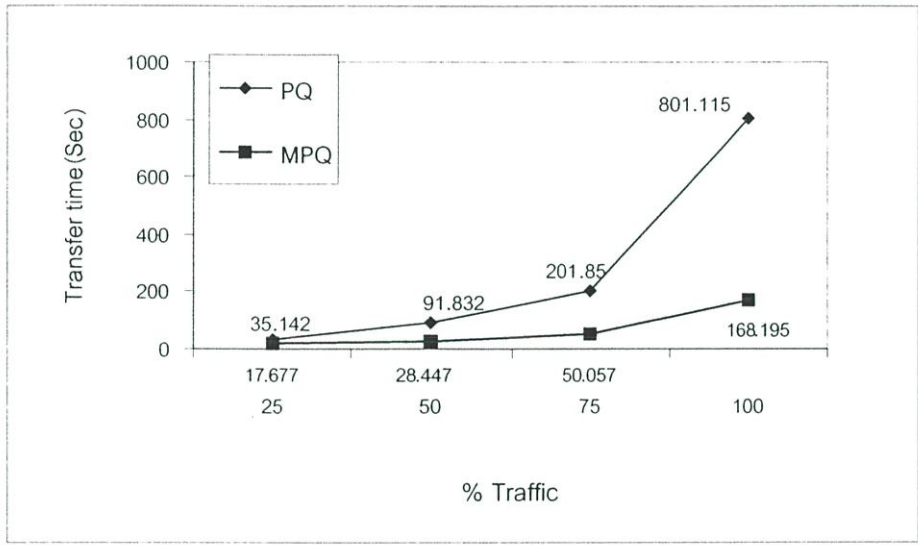
รูปที่ 5.22 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 1 Mbytes



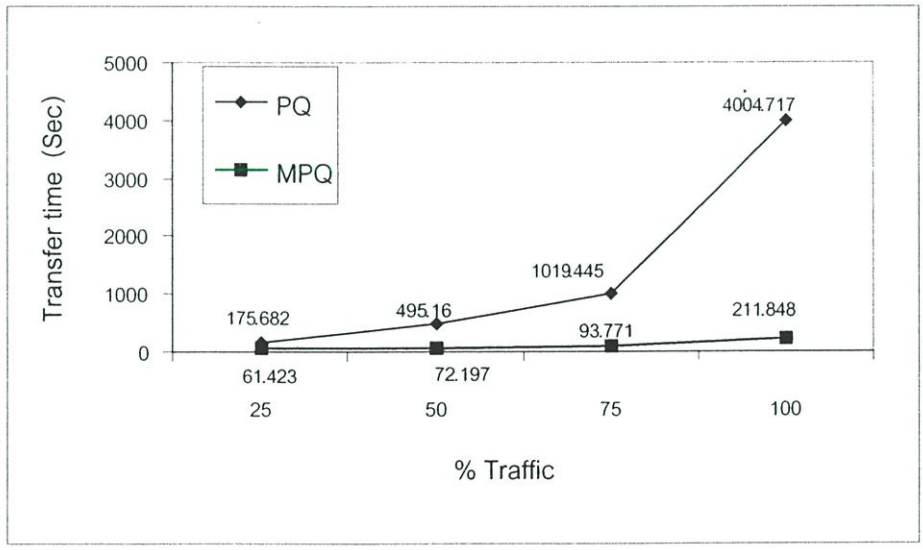
รูปที่ 5.23 ผลการทดลองเปรียบเทียบของงาน Internet Voice PCM ขนาด 10 Mbytes

จากผลการทดลองในแต่ละรูปจะเห็นว่าการให้บริการแบบลำดับความสำคัญนั้น จะให้ผลของเวลามากกว่าการใช้มัลติโปรเซสส์ไพรออริตี้ทุกกรณี และจะมีอัตราเพิ่มของเวลาของมากกว่าเช่นที่ความหนาแน่นของทราฟฟิค 50% ผลของเวลาที่ได้จากรูปที่ 5.20, 5.21, 5.22 และ 5.23 คือ 35.142, 175.682, 351.257 และ 3521.578 วินาที ในขณะที่มัลติโปรเซสส์ไพรออริตี้ จะได้ 28.61, 116.115, 226.49, และ 2192.240 วินาที ซึ่งจะเห็นว่าอัตราการเพิ่มจะของเวลาจะต่ำกว่า

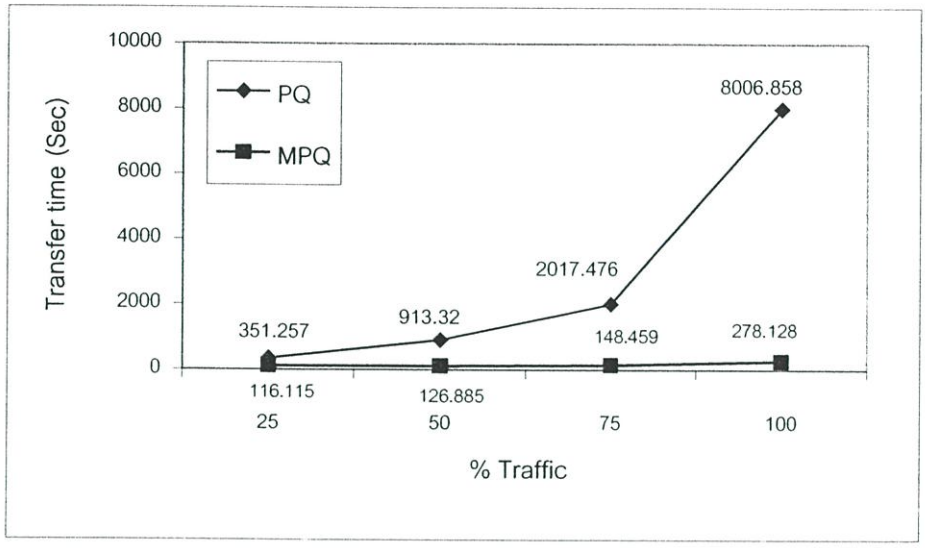
จากรูปที่ 5.24 ถึง 5.27 เป็นการทดลองงาน Audio MP3 โดยงานประยุกต์นี้ต้องใช้แบนด์วิดท์ 128 Kbps เป็นอย่างต่ำจึงจะสามารถส่งข้อมูลได้อย่างไม่เป็นปัญหา โดยรูปที่ 5.24 เป็นการทดลองกับ Audio MP3 ขนาด 100 Kbytes รูปที่ 5.25 ทดลองกับ Audio MP3 ขนาด 500 Kbytes รูปที่ 5.26 ทดลองกับ Audio MP3 ขนาด 1 Mbytes และรูปที่ 5.27 ทดลองกับ Audio MP3 ขนาด 10 Mbytes



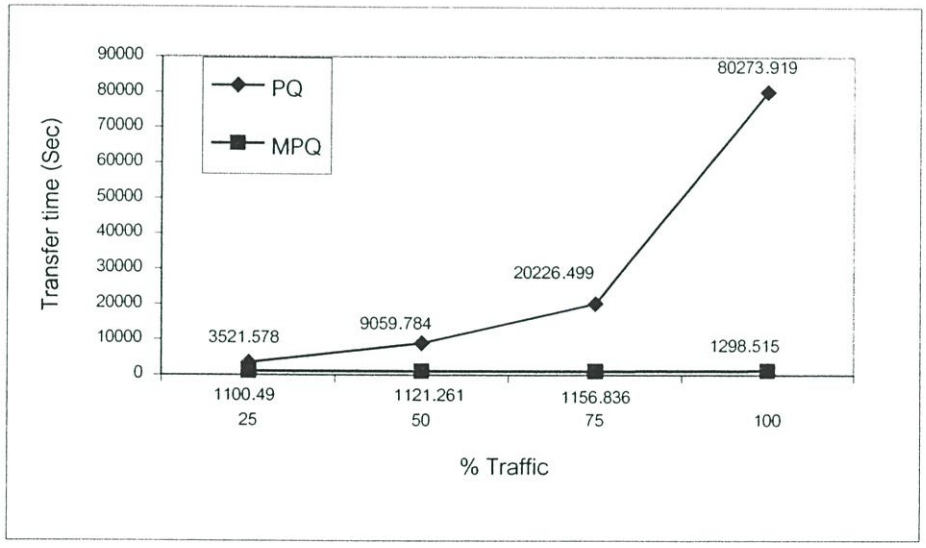
รูปที่ 5.24 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 100 Kbytes



รูปที่ 5.25 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 500 Kbytes



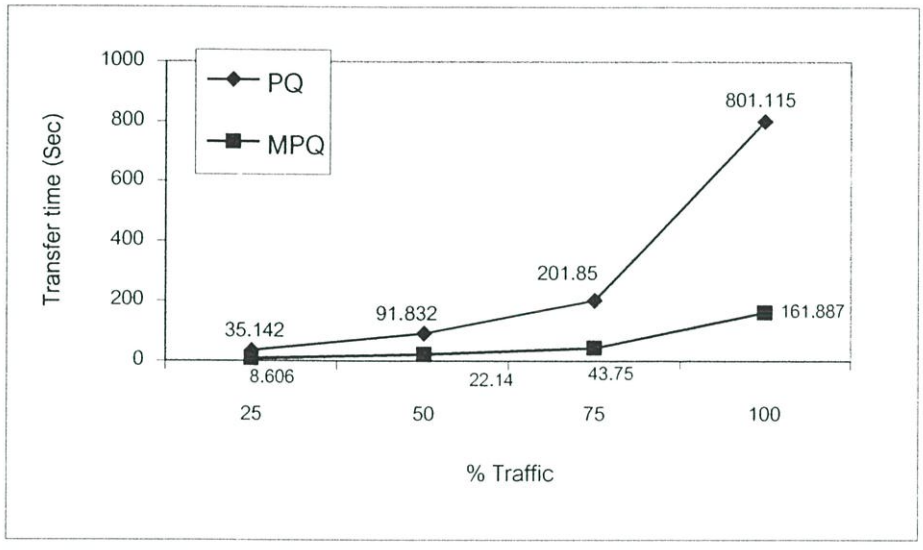
รูปที่ 5.26 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 1 Mbytes



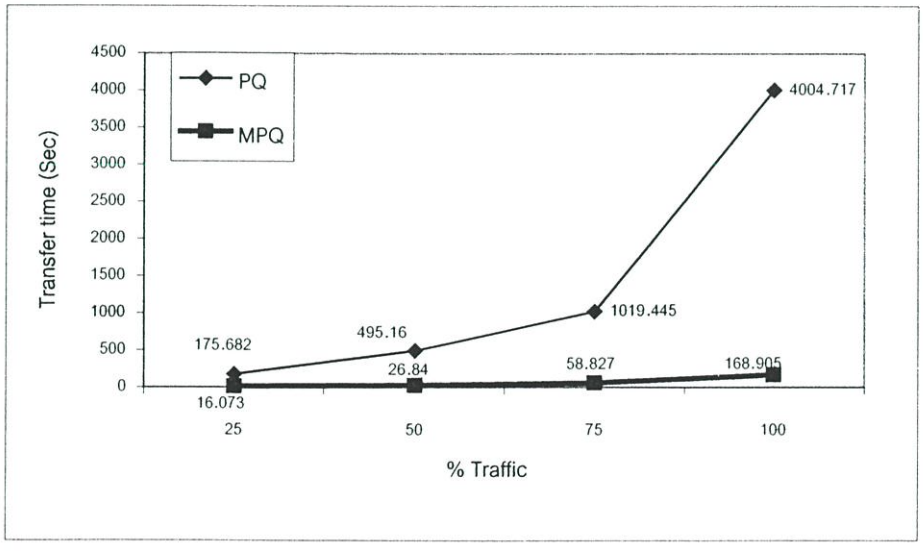
รูปที่ 5.27 ผลการทดลองเปรียบเทียบของงาน Audio MP3 ขนาด 10 Mbytes

จากผลการทดลองในแต่ละรูปจะเห็นว่าการให้บริการแบบลำดับความสำคัญ นั้นจะให้ผลของเวลามากกว่าการใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึม และจะมีอัตราเพิ่มของเวลาของมากกว่า โดยจะเห็นได้ชัดเจนที่ความหนาแน่นของทราฟฟิคนั้นมากขึ้นที่ 50%, 75% และ 100% ตามลำดับ เป็นที่น่าสังเกตว่าทุก ๆ ความหนาแน่นของทราฟฟิคจะมีค่าไม่ต่างกับ Internet Voice PCM มากนัก

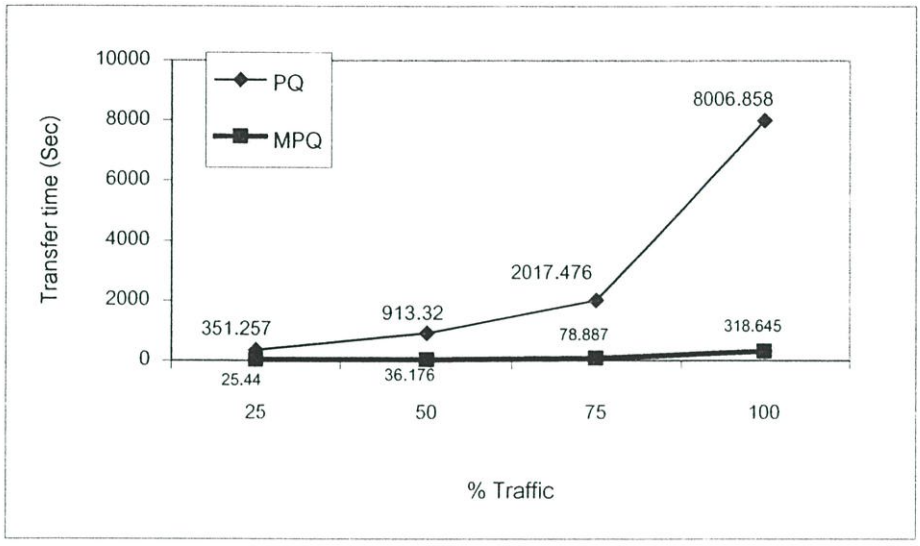
จากรูปที่ 5.28 ถึง 5.31 เป็นการทดลองงาน Real time Internet CD Player โดยงานประยุกต์นี้ต้องใช้แบนด์วิดท์ 750 Kbps เป็นอย่างต่ำจึงจะสามารถส่งข้อมูลได้อย่างไม่เป็นปัญหา โดยรูปที่ 5.28 เป็นการทดลองกับ Real time Internet CD Player ขนาด 100 Kbytes รูปที่ 5.29 ทดลองกับ Real time Internet CD Player ขนาด 500 Kbytes รูปที่ 5.30 ทดลองกับ Real time Internet CD Player ขนาด 1 Mbytes และรูปที่ 5.31 ทดลองกับ Real time Internet CD Player ขนาด 10 Mbytes ตามลำดับ



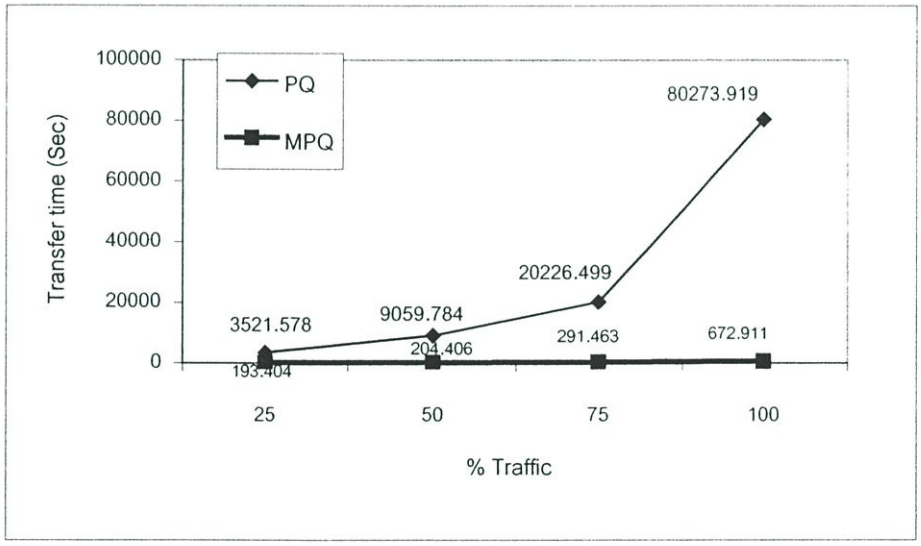
รูปที่ 5.28 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 100 Kbytes



รูปที่ 5.29 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 500 Kbytes



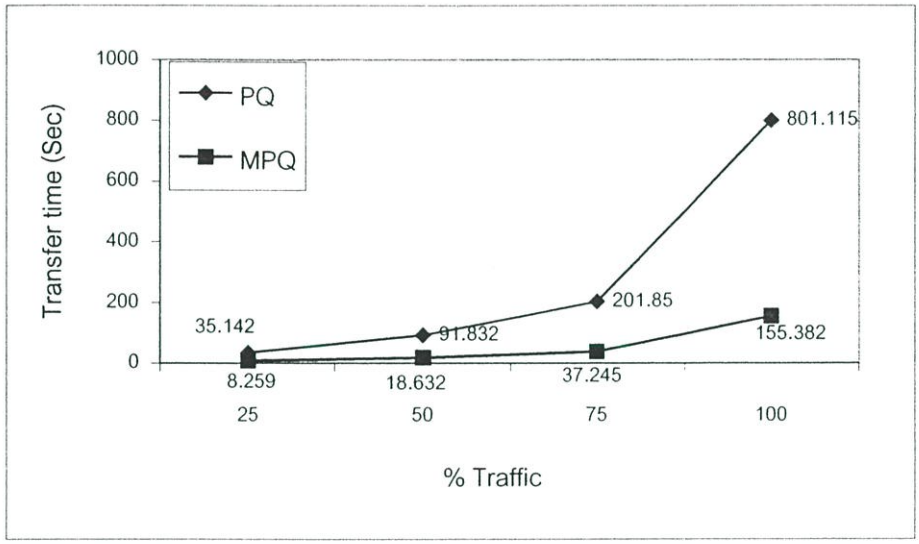
รูปที่ 5.30 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 1 Mbytes



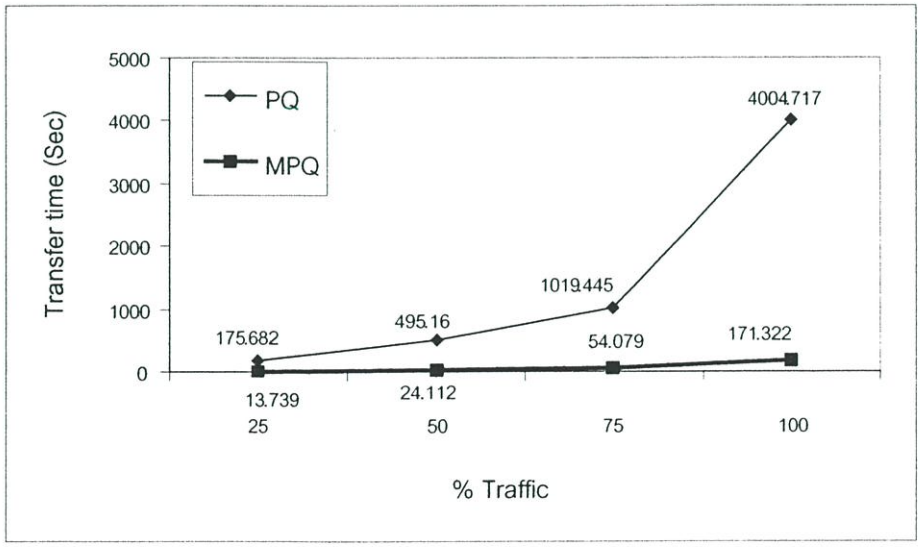
รูปที่ 5.31 ผลการทดลองเปรียบเทียบของงาน Real time Internet CD Player ขนาด 10 Mbytes

จากผลการทดลองในแต่ละรูปจะเห็นว่าการให้บริการแบบลำดับความสำคัญนั้นจะให้ผลของเวลามากกว่าการใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมในทุกขนาดของงานและจะมีอัตราเพิ่มของเวลาของมากกว่า โดยจะเห็นได้ชัดเจนที่ความหนาแน่นของทราฟฟิคนั้นมากขึ้นที่ซึ่งจะเหมือนกันกับกรณีของ Audio MP3 และ Internet Voice PCM

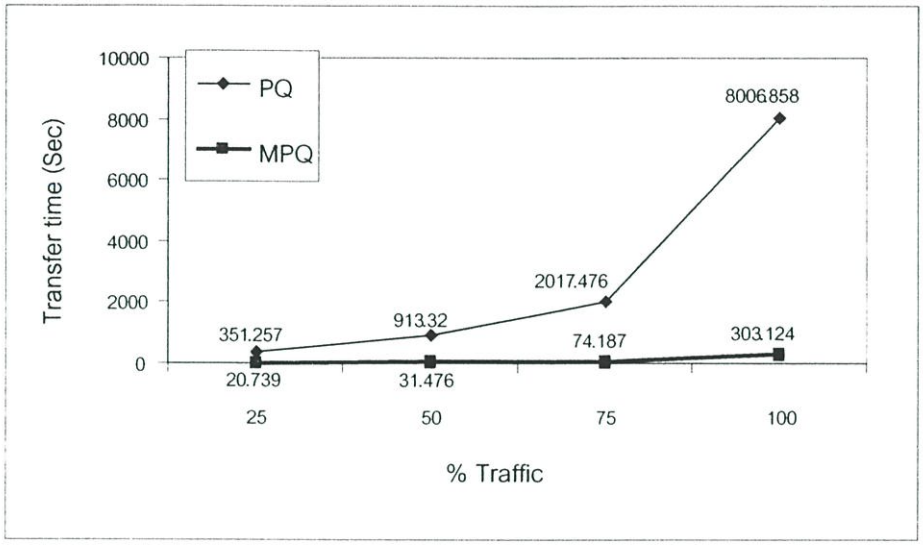
จากรูปที่ 5.32 ถึง 5.35 เป็นการทดลองงาน Video Conference โดยงานประยุกต์นี้ต้องใช้แบนด์วิดท์ 1 Mbps เป็นอย่างต่ำจึงจะสามารถส่งข้อมูลได้อย่างไม่เป็นปัญหา โดยรูปที่ 5.32 เป็นการทดลองกับ Video Conference ขนาด 100 Kbytes รูปที่ 5.33 ทดลองกับ Video Conference ขนาด 500 Kbytes รูปที่ 5.34 ทดลองกับ Video Conference ขนาด 1 Mbytes และรูปที่ 5.35 ทดลองกับ Video Conference ขนาด 10 Mbytes ตามลำดับ



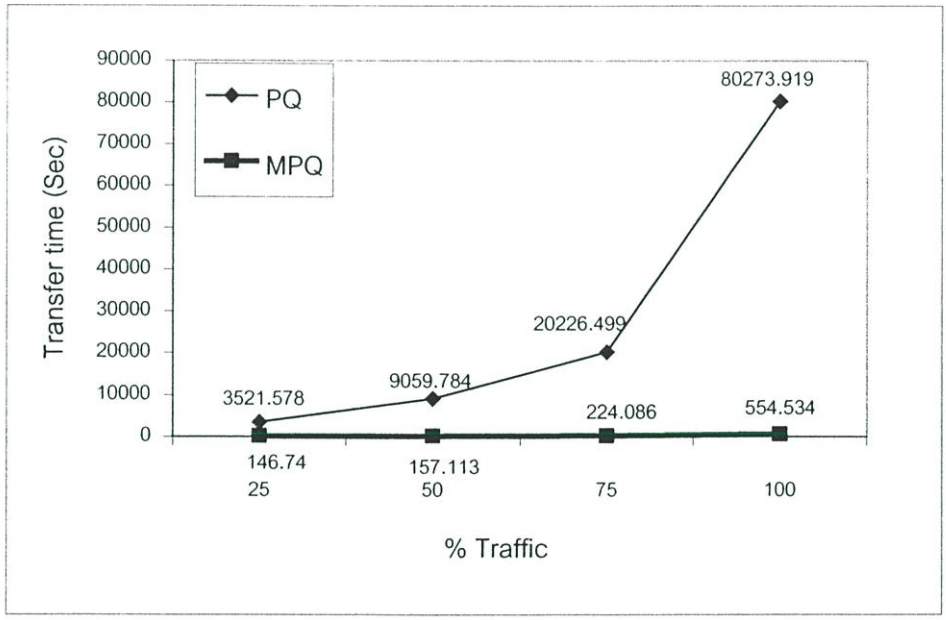
รูปที่ 5.32 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 100 Kbytes



รูปที่ 5.33 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 500 Kbytes



รูปที่ 5.34 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 1 Mbytes



รูปที่ 5.35 ผลการทดลองเปรียบเทียบของงาน Video Conference ขนาด 10 Mbytes

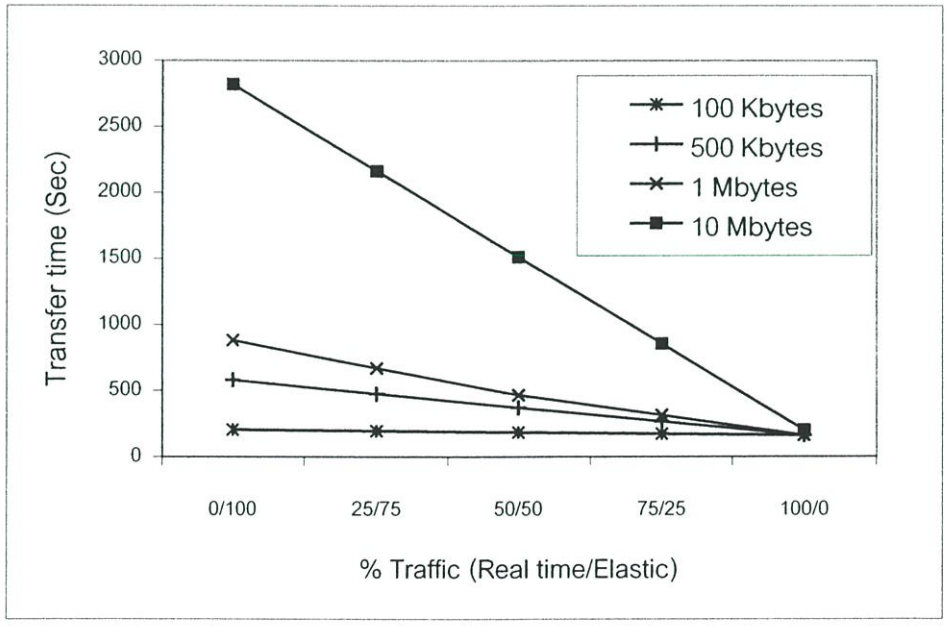
จากการทดลองกับงานแบบเรียลไทม์ของงานประยุกต์ทั้ง 4 งานจะเห็นได้ว่า ในส่วนของการให้บริการแบบลำดับความสำคัญนั้น ถ้าขนาดของข้อมูลมีค่าเท่ากันจะผลของเวลาที่ได้จะเท่ากันหมดทุกงานประยุกต์ ในส่วนการใช้มัลติโปรเซสส์ไพรออริตี้จะให้ผลต่างจากการให้บริการลำดับ

ความสำคัญคือ ถ้างานประยุกต์นั้นมีค่าความต้องการแบนด์วิดท์ในการส่งข้อมูลมากขึ้นจะให้ผลของเวลาดีกว่างานประยุกต์ที่มีความต้องการแบนด์วิดท์ที่ต่ำกว่าซึ่งจากการทดลองจะเห็นได้ว่าที่ขนาดของข้อมูล 100 Kbytes ผลของเวลาในการส่งของงานประยุกต์ Video Conference จะดีที่สุดและผลเวลาของงานประยุกต์ของ Internet Voice PCM จะให้ค่าต่ำสุด

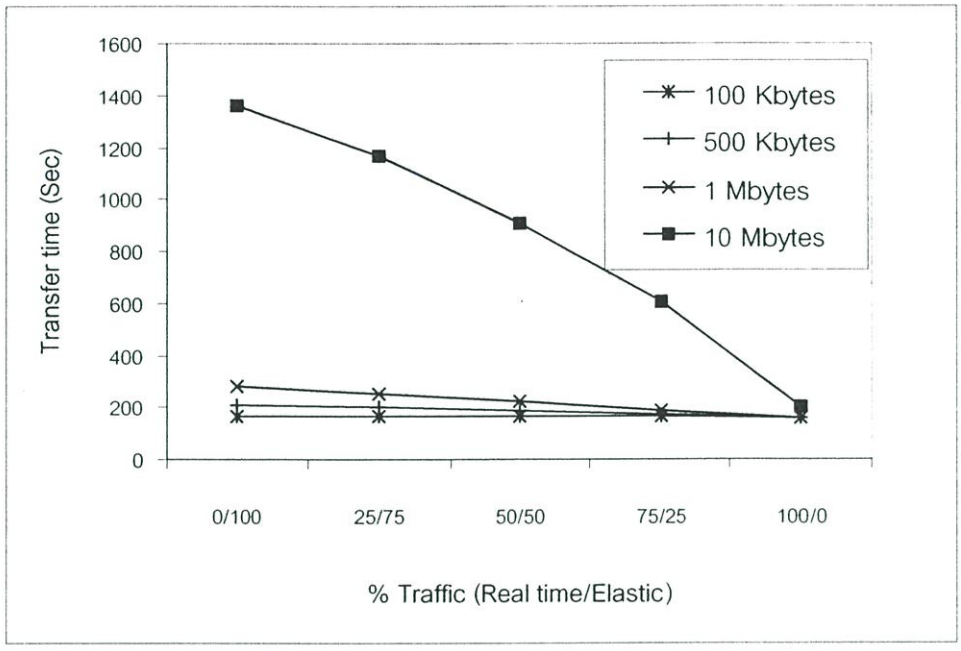
5.6 การทดลองกับงานประยุกต์แบบอิลาสติกและแบบเรียวไทม์

ในการประยุกต์ใช้งานจริงนั้นงานประยุกต์ที่ใช้อยู่ในเครือข่ายจะมีทั้งแบบเรียลไทม์และแบบอีลาสติกซึ่งโดยเฉพาะระบบกริดงานแต่ละชนิด จะเป็นงานประยุกต์ที่ต้องใช้ทรัพยากรร่วมกันสูงและหลากหลายกว่าระบบทั่ว ๆ ไป ในหัวข้อนี้จะเป็นการทดสอบระบบงานที่มีทั้งเรียลไทม์และอีลาสติกเพื่อจะทดสอบว่าเมื่อระบบกริดมีอัตราส่วนของงานงานประยุกต์ในระบบ รวมกันอยู่ทั้งสองแบบจะมีผลอย่างไรเมื่อใช้มัลติโปรเซสส์อัลกอริทึมสำหรับควบคุมการให้บริการ

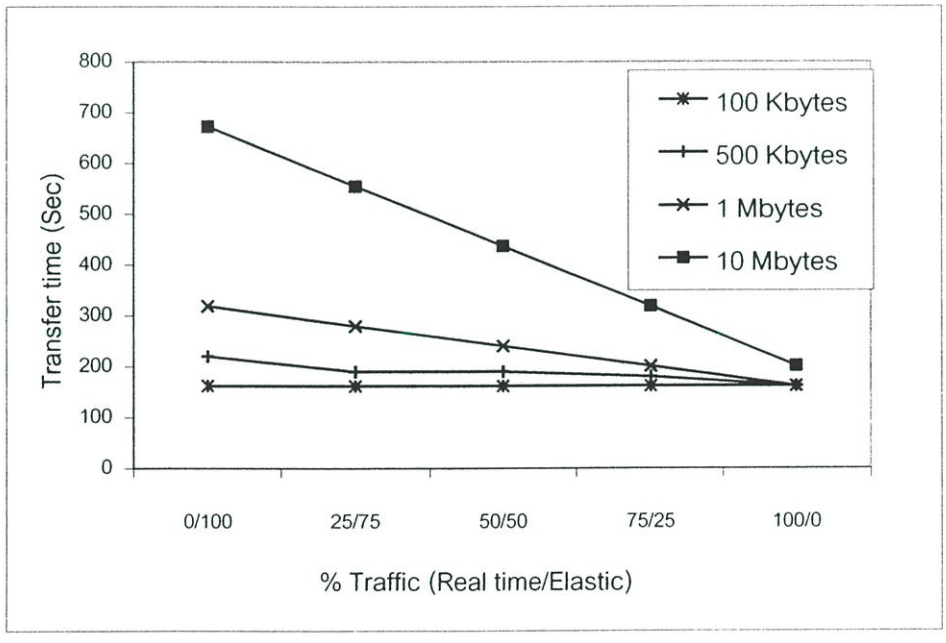
ในการทดลองนั้นจะกำหนดอัตราส่วนของข้อมูล ระหว่างงานประยุกต์แบบเรียลไทม์และงานประยุกต์แบบอิลาสติกอยู่ในระบบ โดยทดสอบกับขนาดของงานตั้งแต่ 100 Kbytes, 500 Kbytes, 1 Mbytes และ 10 Mbytes ตามลำดับ และกำหนดอัตราส่วนในการทดลอง อิลาสติก/เรียลไทม์ เป็น 0/100, 25/75, 50/50 และ 75/25, 100/0 จากการทดลองกับระบบที่มีข้อมูลทั้งสองแบบจะได้ผลแสดงดังรูปที่ 5.36 ถึง รูปที่ 5.39



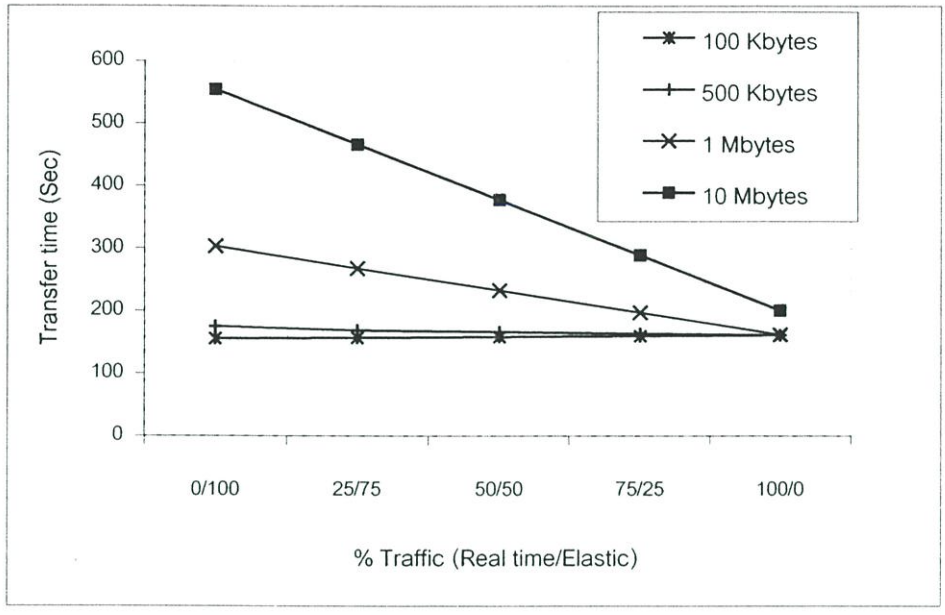
รูปที่ 5.36 ผลการทดลองเปรียบเทียบระหว่าง Internet Voice PCM กับงานแบบอิลาสติก



รูปที่ 5.37 ผลการทดลองเปรียบเทียบระหว่าง Audio MP3 กับงานแบบอิลาสติก



รูปที่ 5.38 ผลการทดลองเปรียบเทียบระหว่าง Real time Internet CD Player กับงานแบบอิลาสติก



รูปที่ 5.39 ผลการทดลองเปรียบเทียบระหว่าง Video Conference กับงานแบบอิลาสติก

อัตราส่วน 0/100 นั้นหมายความว่าในความหนาแน่นของเครือข่าย 100% นั้นมีงานแบบเรียลไทม์อยู่ 100 %และไม่มีงานแบบอิลาสติก ถ้า 25/75 หมายความว่ามีงานแบบอิลาสติก 25% และงานเรียลไทม์อยู่ 75% กราฟในการทดลองการ 100 Kbytes หมายความว่าในการทดลองแต่ละครั้งจะกำหนดงานแบบเรียลไทม์ 100 Kbytes และงานแบบอิลาสติก 100 Kbytes โดยการทดลองแต่ละครั้งจะกำหนดความหนาแน่นของทราฟฟิคเป็น 100%

สำหรับผลของเวลาในการทดลองจากกราฟจะเห็นได้ว่า เวลาจะลดลงเมื่ออัตราส่วนของงานแบบเรียลไทม์ลดลงในระบบลดลงเช่นที่ข้อมูลขนาด 100 Kbytes ถ้ามีงานแบบเรียลไทม์อยู่ 100% จะใช้เวลา 205.486 และจะลดลงเป็น 194.679, 183.548 และ 172.399 วินาทีเมื่ออัตราส่วนเป็น75%, 50%, และ 25% ตามลำดับ และถ้าในระบบมีงานประยุกต์แบบเรียลไทม์ที่ใช้แบนด์วิดท์ในการส่งมากขึ้นเวลาจะลดลงได้อีก เช่นงาน Internet Voice PCM (64 Kbps) จะใช้เวลามากกว่างาน Video Conference (1 Mbps) เมื่อขนาดข้อมูลมากขึ้น

บทที่ 6
สรุปผลวิจัยและข้อเสนอแนะ

6.1 สรุปงานวิจัย

งานวิจัยนี้ได้นำเสนอแนะแนวคิดเบื้องต้นในการควบคุมคุณภาพการให้บริการ โดยใช้แนวคิดการทำงานตามมาตรฐานสถาปัตยกรรมการให้บริการแบบเปิด โดยประยุกต์ใช้ในระดับงานประยุกต์ จะเป็นการจัดการแบนด์วิดท์ให้กับงานประยุกต์ จากการประเมินค่าตามขนาดของงานประยุกต์ที่กำลังส่งอยู่ เพื่อให้สามารถใช้ได้กับระบบที่มีเทคโนโลยีหลากหลายจึงไม่ตรวจสอบลึกลงไปในระดับแพคเก็ตหรือเลเยอร์ต่ำเมื่ออ้างถึงตามมาตรฐานของ OSI โดยการวิจัยจะเน้นไปที่การใช้งานบนระการประมวลผลแบบกริด เมื่อระบบนั้นมีจุดมุ่งหมายที่แน่นนอนในการส่งข้อมูลไปที่ใดที่หนึ่งหรือระบบนั้นค้นหาข้อมูลเจอแล้วและพร้อมที่จะส่ง

โครงสร้างที่เสนอจะปรับปรุงจากลำดับชั้นแบบทรีให้เป็นโครงสร้างแบบฮีพ และการให้บริการแบบลำดับความสำคัญเป็นตัวจัดการโครงสร้างนี้ โดยทั่วไปการประยุกต์ใช้งานของโครงสร้างแบบฮีพจะใช้คิวลำดับ (Priority Queue) เป็นตัวจัดการสำหรับการบริการแบบลำดับความสำคัญแต่จะใช้วิธีการจัดการของฮีพ วิยานิพนธ์นี้จึงปรับปรุงเพิ่มเติมในส่วนการทำงานโดยที่ไม่ต้องเสียเวลาส่วนหนึ่งในการรอการส่งข้อมูลของแต่ละงานแต่จะส่งไปพร้อม ๆ กันถ้าหากมีทรัพยากรเพียงพอ ซึ่งทรัพยากรที่ใช้ในงานวิจัยนี้จะใช้สองอย่างเป็นหลัก คือแบนด์วิดท์ของเครือข่ายและเนื้อที่ในหน่วยความจำถาวรขนาดใหญ่ซึ่งในระบบกริดนั้นยอมให้ใช้ทรัพยากรเหล่านี้ร่วมกันได้จำนวนมาก

หน่วยที่สำคัญในการควบคุมการให้บริการโดยใช้อัลกอริทึมนี้คือตัวเอเจนต์ และหน่วยควบคุมส่วนกลาง เอเจนต์เป็นส่วนควบคุมทรัพยากรของกริดและคอยรายงานค่าของทรัพยากรต่างๆ ของกริดไปยังส่วนกลางโดยหน้าที่สำคัญที่สุดของเอเจนต์คือ การวิเคราะห์แบนด์วิดท์ที่ต้องใช้ของงานประยุกต์ที่ต้องการส่ง ต้องหาให้ได้ว่างานประยุกต์ที่กริดต้องการจะส่งนั้นต้องการแบนด์วิดท์เท่าไหร่และรายงานไปให้ตัวควบคุมส่วนกลาง (ASM) ตัวควบคุมส่วนกลางจะทำหน้าที่เหมือนส่วนข่าวสารข้อมูลของกริด ในการเก็บข้อมูลข่าวสารของเอเจนต์ และใช้มัลติโปรเซสส์ไพรออริตี้อัลกอริทึมวิเคราะห์การควบคุมคุณภาพการให้บริการว่า สมควรจะส่งงานประยุกต์นั้นได้หรือไม่ตามที่เอเจนต์แจ้งมาและถ้าส่งได้ต้องใช้แบนด์วิดท์ค่าใด

ส่วนของการวิเคราะห์เส้นทางการส่งนั้นตัวควบคุมส่วนกลางจะจัดเส้นทางการส่งข้อมูลของเอเจนต์แต่ละตัวไว้แล้วในขั้นตอนการลงทะเบียนเป็นสมาชิกของกริดเริ่มแรก โดยจะเก็บไว้ในหน่วยความจำ การจัดเส้นทางจะจัดตามหลักการของการเชื่อมต่อความจุแบบต่ำลง การส่งข้อมูลต้องส่งตามเส้นทางนี้เท่านั้นเพราะถึงแม้ว่าเอเจนต์กริดโหนดจะอยู่ใกล้กัน ถ้าเราส่งข้อมูลโดยไม่ใช้เส้นทาง

นี้อาจจะเป็นปัญหาต่องานของกริดอื่นก็ได้ ถึงแม้ระยะทางจะไกลกว่า แต่มีแบนด์วิดท์เพียงพอต่อการใช้งานนั้นก็คืออัลกอริทึมนี้จะมองที่การทำงานกับจำนวนข้อมูลมาก ๆ หรือมองประสิทธิภาพทั้งระบบเป็นสำคัญ

6.2 สรุปผลการทดลอง

จากการทดลองเปรียบการควบคุมคุณภาพการให้บริการ โดยใช้อัลกอริทึมแบบการให้บริการแบบลำดับความสำคัญและมัลติโปรเซสส์ไพรออริตี้ นั้นเมื่อทดลองกับงานแบบอิลาสติกจะเห็นได้ว่าที่แบนด์วิดท์ต่ำจำนวนความหนาแน่นของข้อมูลสูงจะเห็นผลต่างของเวลาระหว่างสองอัลกอริทึมนี้มาก เนื่องจากว่าการให้บริการแบบลำดับความสำคัญนั้นไม่ได้มีการส่งข้อมูลที่เส้นทางอื่นด้วย จึงเสียเวลารอให้งานปัจจุบันส่งเสร็จก่อน แต่ในมัลติโปรเซสส์นั้นถึงแม้ว่าจะมีแบนด์วิดท์ในการส่งข้อมูลน้อยแต่สามารถส่งข้อมูลในเส้นทางอื่นได้ ถ้าเส้นทางนั้นไม่ซ้ำซ้อนกับเส้นทางที่กำลังส่งอยู่แต่ถ้าหรือมีแบนด์วิดท์เพียงพอ ถ้าแบนด์วิดท์ที่ใช้ในการส่งข้อมูลเพิ่มขึ้นการให้บริการแบบลำดับความสำคัญจะมีผลของเวลาดีขึ้นเรื่อย ๆ จนดีกว่าแบบมัลติโปรเซสส์ไพรออริตี้ ซึ่งจะเห็นได้จากรูปที่ 5.11 เพราะว่าแบบมัลติโปรเซสส์นั้นต้องเสียเวลาช่วงหนึ่งไปกับการจัดการ แต่การบริการแบบลำดับความสำคัญไม่ต้องเสียเวลาในส่วนนี้ นั้นคือจะสามารถค้นหาข้อมูลในคิวได้เลยในระหว่างที่รอจึงไม่เสียเวลาในส่วนนี้แต่อย่างไรก็ตาม เมื่อขนาดของข้อมูลมากขึ้นประสิทธิการส่งข้อมูลแบบการให้บริการแบบลำดับความสำคัญก็เริ่มลดลงเรื่อย ๆ หรือสามารถกล่าวได้ว่าเมื่อมีแบนด์วิดท์ในการส่งข้อมูลที่มากพอและขนาดของข้อมูลน้อยควรใช้อัลกอริทึมแบบการให้บริการแบบลำดับความสำคัญเป็นตัวจัดการระบบ เนื่องจากสามารถจัดการได้ง่ายและให้ประสิทธิภาพที่ดีกว่า ส่วนการทำงานแบบเรียวไทม์นั้นแบบการให้บริการแบบลำดับความสำคัญ จะไม่สนใจว่าข้อมูลนั้นเป็นข้อมูลของงานประเภทอะไรจะส่งเท่าที่มีแบนด์วิดท์ที่ส่งได้ในขณะนั้น แต่เป็นไปได้ว่าสามารถส่งไปเก็บไว้ที่หน่วยความจำของกริดนั้น ๆ ก่อนที่จะประมวลผลงานซึ่งจะไม่ให้ความสำคัญในส่วนนี้ แต่แบบมัลติโปรเซสส์จะกำหนดขนาดแบนด์วิดท์ให้มีปริมาณเพียงพอในการส่งงานแบบเรียวไทม์แต่ละครั้ง จึงสามารถส่งงานได้หลายงานในเวลาเดียวกัน ซึ่งผลที่ได้คือเวลารวมในการทำงานก็จะดีกว่าเดิมซึ่งสามารถดูได้จากการรูปที่ 5.20 ถึง 5.35

ถ้าเป็นการประยุกต์ใช้งานกับระบบทั่วไป ถ้ามีงานแบบอิลาสติกมากกว่า ระบบที่ใช้มัลติโปรเซสส์อัลกอริทึม จะให้เวลาดีกว่าในระบบที่มีงานประยุกต์แบบที่มีอัตราส่วนของงานแบบเรียวไทม์มากว่า สำหรับงานแบบเรียวไทม์ที่มีขนาดของข้อมูลเท่ากันแต่มีความต้องการแบนด์วิดท์ในการส่งมากกว่าจะให้ผลของเวลาดีกว่า นั้นก็เพราะว่าสามารถส่งข้อมูลได้มากกว่าจากการกำหนดแบนด์วิดท์เริ่มต้นที่มากกกว่า และในการส่งแบบอิลาสติกนั้นเป็นไปได้ว่าการกำหนดแบนด์วิดท์เริ่มต้นนั้น

บางทีแบบมัลติโปรเซสส์อัลกอริทึมส่งงานได้น้อยกว่า เนื่องจากว่ากำหนดค่าเริ่มต้นในการส่งน้อยกว่านั้นเอง

อย่างไรก็ตามอัลกอริทึมแบบมัลติโปรเซสส์ นั้นมีความเหมาะสมที่จะใช้กับโครงสร้างแบบแบบฮีพมากกว่าที่จะใช้อัลกอริทึมแบบการให้บริการแบบลำดับความสำคัญ เพราะเมื่อมีความหนาแน่นของงานประยุกต์มากขึ้นอัลกอริทึมนี้ สามารถจัดการงานได้หลายทางหรืออาจกล่าวได้ว่าสามารถจัดเส้นทางได้แบบขนานหลายเส้นทางนั้นเอง ถึงแม้ว่าบางครั้งจะมีเวลาส่งด้อยกว่าการให้บริการแบบลำดับความสำคัญแต่นั้นก็หมายความว่าต้องมีความจุของเส้นทางสูงมาก ๆ เท่านั้น

6.3 ข้อเสนอแนะและปัญหา

เนื่องจากงานวิจัยนี้เป็นการเสนอแค่แนวคิดพื้นฐาน ของการควบคุมคุณภาพการให้บริการโดยจะมองและเลือกงานทั้งก้อนหรือทั้งไฟล์ ซึ่งต่างจากการทำการควบคุมคุมคุณภาพการให้บริการแบบทั่วไปที่มองเป็นแพคเก็ต ดังนั้นจึงมีตัวแปรอีกมากที่ยังไม่ได้รับการทดสอบเช่น ในส่วนของดีเลย์แบบต่าง ๆ ซึ่งเป็นเรื่องยากต่อการเขียนโปรแกรมจำลองการทำงาน ดังนั้นจึงเน้นไปที่การวิเคราะห์เส้นทางการส่งข้อมูลเป็นสำคัญ

อีกส่วนที่สำคัญคือการค้นหาลำดับลำดับความสำคัญของข้อมูลในคิวลำดับโดยใช้อัลกอริทึมของฮีพนั้นถ้าเป็นแบบลำดับความสำคัญ แน่นอนว่าการค้นต้องทำตามลำดับการส่งข้อมูล แต่ถ้าเป็นแบบมัลติโปรเซสส์แล้ว สามารถที่จะค้นหาแบบขนานโดยใช้โปรเซสส์เซอร์หลาย ๆ ตัวได้ คือสามารถค้นหาลำดับความสำคัญสูงสุดกับลำดับความสำคัญที่มีอันดับรองลงไปในเวลาพร้อมกันได้ตามหลักการทำงานที่เรียกว่า อัลกอริทึมแบบการโปรเซสส์ไพรออริตี้แบบขนานสำหรับโครงสร้างแบบฮีพ (Parallel Process Priority for Heap) ซึ่งน่าจะใช้เป็นหัวข้อต่อไปในการวิจัย

เอกสารอ้างอิง

[1] W. Poompattanapong, B. Piyatamrong; "A Multi-Agent for Grid Service Management", In the 3rd International Symposium on Communication and Information Technologies (ISCIT'03)., Songkla,Thailand, pp. 735-739, September 2003.

[2] R. Bhagwan, B. Lin; "Fast and Scalable Priority Queue Architecture for High-Speed Network Switches", Center for Wireless Communication, University of California San Diego, http://cwc.ucsd.edu/~billlin/papers/application/infocom2000.pdf

[3] K.Pyun. J. Song, H. Lee," A generalized hierarchical service curve algorithm for high network utilization and link-sharing", Dept. of computer engineering and computer science, KAIST,Soult Korea,2003

[4] Quality of Service Networking, Chapter 49,[Online] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm

[5] I. Foster, and C. Kesselman, (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.

[6] I. Foster, C. Kesselman, and S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". International Journal of High Performance Computing Applications, Vol. 15 No. 3, pp. 200-222, 2001.

[7] World Wide Web Consortium. Extensible Markup Language (XML) 1.1. W3C Recommendation, http://www.w3.org/TR/xml11/, October, 2002.

[8] World Wide Web Consortium. Simple Object Access Protocol (SOAP) 1.2. W3C Recommendation,[Online].Available:http://www.w3.org/TR/soap12-part1/, May,2003.

[9] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", http://www.globus.org/research/papers/ogsa.pdf, June, 2002.

[10] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. "Grid Information Services for Distributed Resource Sharing". In Tenth IEEE Int'l. Symposium on High-Performance Distributed Computing (HPDC-10), San Francisco,2001

[11] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1. W3C Note 15, 2001,[Online] http://www.w3.org/TR/wsdl

[12] C. Li, L. Li, "Integrate software agents and CORBA in computational grid", Journal of Computer Standards & Interfaces, Vol. 25, No. 4, pp. 357-371, Elsevier, August, 2003.

[13] N. R. Jennings and M. J. Wooldridge (eds). Agent Technology: Foundations, Application and Market, Springer-Verlag, 1998.

[14] Y. Labrou, T. Finin, Yun Peng; "Agent communication languages: the current landscape", IEEE Intelligent Systems, Vol 14, No. 2 , pp. 45-52, Mar/Apr, 1999.

[15] The DARPA CoABS Project: "Control of Agent based System". http://coabs.globalinfotek.com/, 2000.

[16] IBM, "Introduction to Grid Computing with Globus", International Business Machine Corp, 2003, [Online] Http://www.readbook.ibm.com/readbook/pdf/sg246895.pdf

[17] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Service for Distributed Resource Sharing", 10th IEEE International Symposium on High Performance Distributed Computing,2001,[Online], Http://www.globus.org/research/paper/mds-hppc.pdf

[18] J. Marin. "Web Services: The Next Big Thing". XML-Journal 2,[Online], http://www.sys-con.com/xml/

[19] Object Management Group, "Agent Technology", OMG Document, 1 September,2001,[Online],http://www.omg.org/docs/agent/00-09-01.pdf

[20] J. Cao,S. A. Jarvis, D. P. Spooner, "Performance Prediction Technology for Agent-Base Resource Management in Grid Environment", Proceeding of 11th IEEE Heterogeneous Computing Workshop,Florida,USA,2002

[21] B. S. S. Chatterjee, T. F. Lawrence; "Taxonomy for QoS Specification", proceeding of Words '97,Newport beach,California,1997

[22] J. Houtari, M. Makela, "Service Level Agreements", Kelsinky University of Technology, November,2001

[23] C. Luckchonlatee,"Quality control of packet on network layer switchs", Master Thesis of King mongkut's institute of technology ladklabang,2002

[24] M. Albrecht, M. Koster, P. Martini, M. Frank,"End-to_End QoS Management for Delay-sensitive Scalable Multimedia Stream over DiffServ", Proceeding of the 25th Annual Conference on Local Computer Network (LCN'00), Tampa, FL, USA, 2002

[25] W. Zhao, D. Olshefski, H. Schulzrinne, "Internet Quality of Service: an Overview", Columbia University, [Online], Http://www.cs.columbia.edu/~hgs/papers/Zhao0002_Internet.pdf

[26] A. Pual, "QoS in Data Network: Protocols and Standard",[Online], Http://www.ghostship.com/infosyssec/netprot1.htm

[27] S. Belenki, "Traffic Management in QoS network: Overview and Suggested Improvements", Department of Computer Engineering, Chalmers University of Technology, Sweden,2000, [Online], Http://fibba.ce.chalmer.se/staff/betenki/tr005.pdf

[28] P. A. Buhler, J. M. Vida," Towards Adaptive Workflow Enactment Using Multiagent Systems",[Online], jmridal.sce.se.edu/paper/buhler03b.pdf

[29] I. Foster, A. Roy, V. Sander, L. Winkler, "End-to-End Quality of Service for High-End Applications, Mathematic and Computer Science Division, Aragon National Laboratory, Aragon, U.S.A

[30] H.N. Lim Choi Keung, S.A. Jarvis, J. Cao, D.P. Spooner,G.R. Nudd,"Grid Information Service using Software Agents", 18th Anuual UK Performance Engineering Workshop, University of Glasgow, UK,2002,pp 178-198

[31] J. Cao, D. J. Kerbyson, G. R. Nudd, "Performance Evaluation of an Agent-Based Resource Management Infrastructure for Grid Computing", Proceedings of 1st IEEE/ACM International Symposium on Cluster Computing and the Grid. Brisbane, Australia, pp.311–318, May, 2001.

[32] J. Cao, D. P. Spooner, J. D. Turner, S. A. Jarvis, D. J. Kerbyson, S. Saini, and G. R. Nudd, "Agent-based Resource Management for Grid Computing", Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid. Berlin, Germany, pp. 323-324, May, 2002.

[33] A. Loannou, M. Katevenis, "Pipeline Heap (Priority Queue) Management for Advance Scheduling in High-Speed Networks", Science and Technology Park of Crete, Greece,
[Online], http://archvlsi.ics.forth.gr/muqpro/heapMgt.html

[34] M. Bunruangses, P. Buayatnoparat, "QoS Multi-Agent Systems by Multi-Process Algorithm", 3rd Asian International Mobile Computing Conference (AMOC 2004), 2004, Thailand

[35] M. Bunruangses, W. Poompattanapong, P. Banyatnoparat, B. Piyayamrong, "QoS Multi-Agent Applied for Grid Service Management", Proceeding of the 3rd Internation Symposium on Information and Communication Technologies, Las Vegas, Nevada, June 16th-18th, 2004

ภาคผนวก ก.

ตัวอย่างของเอเจนต์และเจ้าของผู้ผลิต

ตารางที่ ก. ตัวอย่างของเอเจนต์และเจ้าของผู้ผลิต

Product Name (Company) URL	Mobile/ Stationary	Language	Standards	Availability Licensing	Example application
AgentBuilder(Reticular Systems) http://www.agentbuilder.com	Stationary Mobile	Java		Evaluation, Academic	E-commerce,auctions, job-finder,private e-mail,etic.
Aglets (IBM Japan) http://www.trl.ibm.co.jp/aglets	Mobile	Java	MASIF	Evaluation	Remote Monitoring (e.g., Web site),meeting Scheduling, auction Etc.
APRIL (Fujitsu,USA) http://www.nar.fujitsulabs.com		APRIL	FIPA	Open Source	Schedule Mgt etc.
Comtee Agent platform (Japan) http://www.fipa.org/glointe.htm	Stationary	Java	FIPA	Open Source	Enterprise Information Mgt.
Concordia (Mitsubishi) http://www.meitca.com/HSL/ Projects/Concordia	Mobile	Java		Evaluation	Interprise Information, ect.
DMARS(AAII) http://www.aaii.oz.au/proj/ dMARS-prod-brief.html	Stationary	C/C++		Superseded by Dcimos	Air traffic control, TNM, simulation,fault diagnosis,etc.
FIPA-OS (Nortel Networks) http://www.nortelnetworks.com/ fipa-os	Stationary	JAVA	FIPA	Open Source	Personalised services, VPN,VHE,meeting scheduler,etc.
Grasshopper (ikv++) http://www.ikv.de/products/ grasshopper/grasshopper.html	Mobile	JAVA	MASIF/ FIPA	Evaluation	E-commerce, Information retrieval, etc.
Jackal(UMBC) http://www.alphaworks.ibm.com/ tech	Stationary	JAVA	KQML	Evaluation	Enterprise information Mgt, manufacturing planning, etc.

Product Name (Company) URL	Mobile/ Stationary	Language	Standards	Availability Licensing	Example application
JADE(CSELT) Http;//Sharon.cselt.it/projects/jade	Stationary	JAVA	FIPA	Open Source	Travel assistant, audio-Visual entertainment, meeting scheduler, etc,
JAFMAS(Unit. Cincinnati) Http://www.ececs.uc.edu/~abaker/ JAFMAS	Stationary	JAVA	KQML	Unrestricted Licensing	Supply chain mgt, etc.
JATLite(Stanford Uni.)	Stationary	JAVA	KQML	Open Source	Design decision Tracking, constraint mgt, Enterprise control.
Jess(Sandia Nat.Labs) http://herzberg.ca.sandia.gov/jess		JAVA		Academic	Various
MOLE(Uni-stuttgart) http://mole.informatik.uni-Stuttgart.de/	Mobile	JAVA		Evaluation, Academic	Information retrieval, groupware, active documents, etc.
Open Agent Architecture(SRI) http://www.ai.sri.com/~oaa/main.Html	Stationary	JAVA,C, VB,Lisp, Prolog,etc.	KQML	Academic	Automated office, robot Control, collaborative Fridge, etc.
Voyager (ObjectSpace) http://www.objectspace.com/	Mobile	JAVA		Academic	Various
ZEUS(bt uk) http://www.labs.bt.com/projeccts/ agents/zeus/	Stationary	JAVA	FIPA KQML	Open Source	Supply chain mgt, Service provisioning, Network resource mgt.

ภาคผนวก ข.

ตัวอย่างตัวอย่างคำสั่งโปรแกรมของมัลติโปรเซสส์ไพรออริตี้อัลกอริทึม

```
Private Sub Process_Mpriority_Algorithn()
'/** โมดูลการประมวลผลแบบ มัล ติโปรเซส ไพออริ ตี้
Dim rs As rdoResultset
Dim rs1 As rdoResultset
Dim search_time As Double
Dim start_time As Double
Dim flow_time  As Double
Dim AppSize, Req_Minsize  As Double
Dim start_kbps As Double
Dim i, j, k, App_type As Integer
Dim Max_pri As Integer
Dim Start_A As Integer, End_A As Integer, A_no As
Integer
Grand_tot_time = 0
Bar1.Visible = True
Bar1.Min = 0
Set rs1 = SpvCountPiQ  ' ค้นหา Agent ที่มี Priority มากที่สุด
Bar1.Max = rs1!tot_rec
Gtot_rec = rs1!tot_rec
Max_flow_time = 0
k = 0
For i = 1 To Gtot_rec
flow_time = 0
Bar1.Value = i
Set rs = SpvRetrieveMaxQ 'ค้นหา Agent ที่มี Priority มากที่สุด
   If rs!H_Order > 0 Then
    A_no = rs!H_Order
    Set rs = SpvRetrieveFlow(Val(A_no)) ' ค้นหา Agent เริ่มต้น
และ Agent สุดท้าย จากเลข Order ที่ต่ำสุด
     If rs.RowCount > 0 Then
     Start_A = rs!from_a   ' Agent เริ่มต้นที่ส่ง
     End_A = rs!to_a       ' Agent สุดท้ายที่ส่ง
     AppSize = rs!app_size
      Req_Minsize = rs!Min_size
     App_type = rs!App_type
     End If
```

```
start_time = start_time + 0.01
      Call Assigment_Flow(Start_A, End_A) ' หา Flow การทำงาน
      If Val(App_type) = 0 Then ' ถ้าเป็นงานElastic ต้องหาค่าเริ่มต้น
       start_kbps = Caculate_Start_Kbps(Val(Start_A), Val(start_time), Val(Req_Minsize),
Val(App_type)) ' คำนวณหาKbps ของตัวแรกที่จะส่ง
      Else
        start_kbps = Val(Req_Minsize)
      End If
      If Val(App_type) = 0 Then
       'If start_kbps > 64 Then  ' Brandwidth ที่เหลือต้องมากกว่าBrandwidth ต่ำสุดของระบบ(กรณีของ
Elastic เท่านั้น)
          Call Analyst_Multiflow_Elastic(Val(start_time), Val(start_kbps), Val(AppSize /
start_kbps))
          flow_time = Insert_Data_Process_Temp(Val(Start_A), Val(End_A),
Val(start_time), Val(start_kbps), Val(AppSize / start_kbps), Val(AppSize), Val(App_type))
          SpvDeleteAFlow (Val(A_no)) ' ลบ Agent ออกจากPriority Queue
       ' End If
       Else ' งานแบบReat time
        'If start_kbps > Req_Minsize Then ' ถ้าจำนวนKbps ไม่เพียงพอส่งไม่ได้ออกจากloop
          Call Analyst_Multiflow_Realtime(Val(start_time), Val(start_kbps), Val(AppSize /
start_kbps))
          flow_time = Insert_Data_Process_Temp(Val(Start_A), Val(End_A),
Val(start_time), Val(start_kbps), Val(AppSize / start_kbps), Val(AppSize), Val(App_type))
          SpvDeleteAFlow (Val(A_no)) ' ลบ Agent ออกจากPriority Queue
        'End If
      End If
   End If
 Next i
Grand_tot_time = start_time + Max_flow_time
Bar1.Visible = False
'MsgBox Str(k)
MsgBox Str(Grand_tot_time) + "      flow Timr = " + Str(Max_flow_time)
End Sub
```

```
Private Sub Analyst_Multiflow_Elastic(start_time As Double, start_kbps As Double, Send_time As
Double)
Dim rs As rdoResultset
Dim i As Integer
Dim Process_time As Double
Process_time = start_time
For i = 0 To 50
If Aflow(i) > 0 Then
  Set rs = Check_Bandwith_requirement_Mul(Aflow(i), Val(Process_time))  ' ความต้องการแต่ละ   Kbps
  If IsNull(rs!curent_Kbps) Then
     End_node = Aflow(i) ' เก็บ Node ก่อนหน้านี้
     Process_time = start_time + Send_time
   Else
  If rs!curent_Kbps >= Val(start_kbps) Then ' ค่า Bandwidth ของ Agent(i) ต้องมากกว่า  Start_Kbps หรือค่า
เริ่มต้น
       End_node = Aflow(i) ' เก็บ Node ก่อนหน้านี้
       Process_time = start_time + Send_time
  Else
    Exit For  ' ถ้าเจอตัวที่น้อยกว่าให้ยกเลิกและส่งไปตัวใหม่ที่ดีกว่า
   End If
 End If
Else
  Exit For
End If
Next i
End Sub ' หลังออกจาก  Function นี้จะทำให้ได้  Agent ที่จะส่งและจำนวน  Kbps ที่ถูก
```

```
Private Sub Analyst_Multiflow_Elastic(start_time As Double, start_kbps As Double, Send_time As
Double)
Dim rs As rdoResultset
Dim i As Integer
Dim Process_time As Double
Process_time = start_time
For i = 0 To 50
If Aflow(i) > 0 Then
  Set rs = Check_Bandwith_requirement_Mul(Aflow(i), Val(Process_time)) ' ความต้องการล่าสุด  Kbps
  If IsNull(rs!curent_Kbps) Then
     End_node = Aflow(i) ' เก็บ Node ก่อนหน้านี้
     Process_time = start_time + Send_time
   Else
   If rs!curent_Kbps >= Val(start_kbps) Then ' ค่า Bandwidth ของ Agent(i) ต้องมากกว่า  Start_Kbps หรือค่า
เริ่มต้น
       End_node = Aflow(i) ' เก็บ Node ก่อนหน้านี้
       Process_time = start_time + Send_time
  Else
    Exit For ' ถ้าเจอตัวที่น้อยกว่าให้ยกเลิกและส่งไปตัวใกล้เคียง
   End If
  End If
 Else
  Exit For
 End If
Next i
End Sub ' หลังออกจาก  Function นี้จะทำให้ได้  Agent ที่จะส่งและจำนวน  Kbps ที่รถือ
```

ภาคผนวก ค.

ผลงานวิจัยในระหว่างการศึกษาที่ได้รับการตีพิมพ์เผยแพร่

[1] M. Bunruangses, P. Bunyatnoparat,"QoS Multi-Agent Systems by Multi-Process Algorithm", 3rd Asian International Mobile Computing Conference (AMOC2004), Bangkok, Thailand, May 2004, pp. 46-49.

[2] M. Bunruangses, W. Poompattanapong, P. Bunyatnoparat, B. Piyatamrong, "QoS Multi-Agent Applied for Grid Service Management", 3rd International Symposium On Information and Communication Technology (ISICT'04), Las Vegas, Nevada, USA, June 2004, pp. 74-79

ประวัติผู้เขียน

นายมนตรี บุญเรืองเศษ เกิดเมื่อวันที่ 2 มกราคม 2512 ที่อำเภอคูเมือง จังหวัดบุรีรัมย์ สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพสาขาไฟฟ้ากำลัง จากวิทยาลัยเทคนิคบุรีรัมย์ ปีการศึกษา 2530 ปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ จากมหาวิทยาลัยรามคำแหง ปีการศึกษา 2535

ประวัติการทำงาน

Programmer ปี พ. ศ. 2533 บริษัท Data Product Systems

Program Analyst ปี พ.ศ.2535 บริษัท International Software Factory (ISOFAC)

System Analyst ปี พ. ศ. 2537 บริษัท Advance Information System (AIT)

Service engineer ปี พ. ศ. 2538 บริษัท Transtel Private Network

ในระหว่างการศึกษาร่วมงานกับอาจารย์บรรจง ปิยธำรง พัฒนาระบบงานคอมพิวเตอร์ให้กับหน่วยงานทั้งรัฐบาลและเอกชนหลายแห่ง รวมทั้งเป็นอาจารย์พิเศษที่มหาวิทยาลัยมหิดลวิทยาเขตไทรโยคและวิทยาลัยรัตนบัณฑิต

ผู้เขียนมีความสามารถออกแบบระบบฐานข้อมูลแบบ Relational โดยใช้ Microsoft Access, SQL Server และ Oracle Database ออกแบบและติดตั้งระบบเครือข่ายคอมพิวเตอร์และระบบเครือข่ายโทรศัพท์