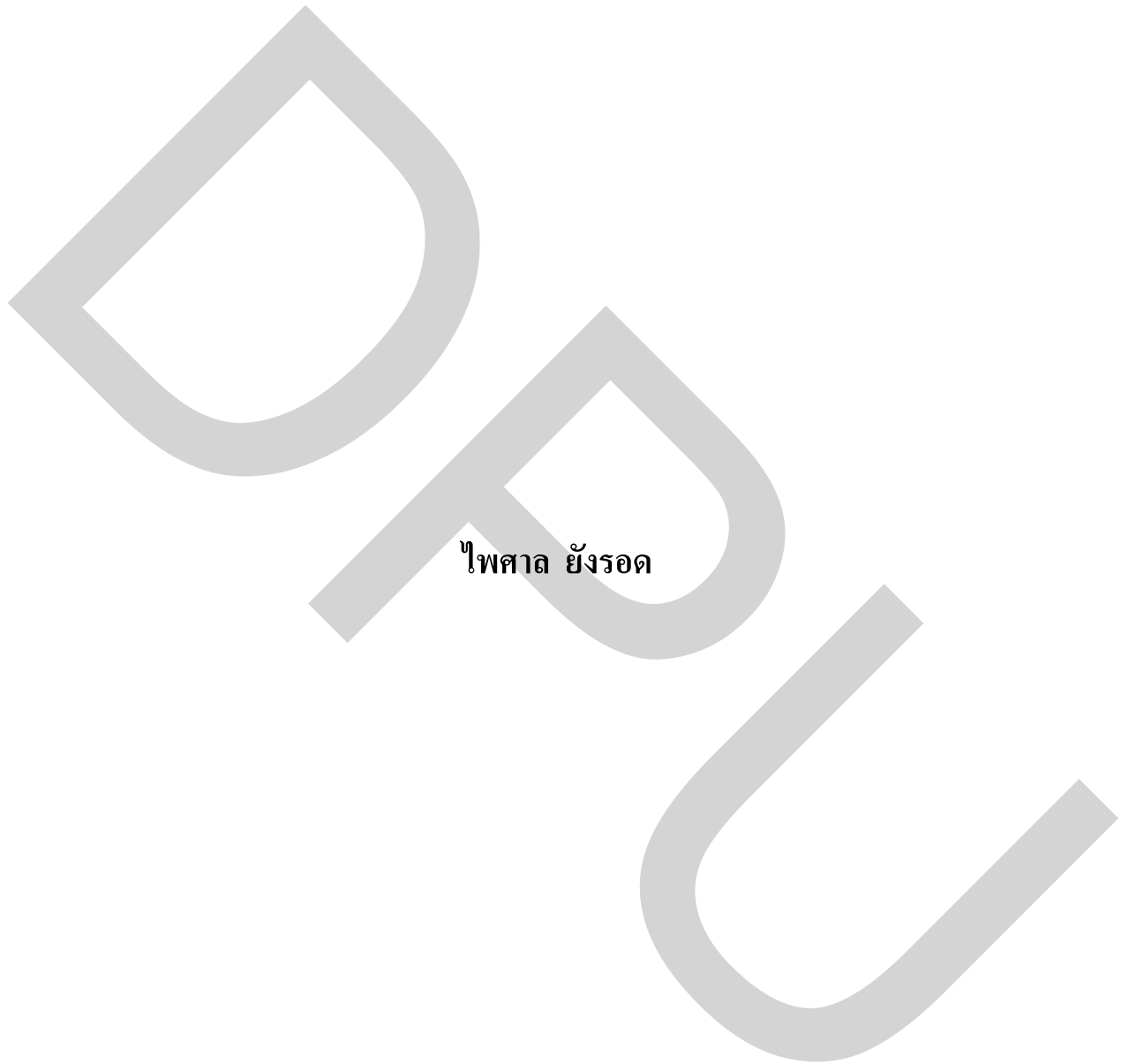


การพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม บัณฑิตวิทยาลัย มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2554

**Development of a Quality of Service Measurement Program
for Voice over IP usage**



Phaisan Youngrod

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering**

Department of Computer and Telecommunication Engineering

Graduate School, Dhurakij Pundit University

2011

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยความกรุณาจาก อาจารย์ ดร.ธัญ จารุวิทย์โกวิท อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่คอยให้คำแนะนำให้คำปรึกษา ตลอดจนแนวทางในการแก้ไขปัญหาต่างๆ ขอขอบคุณ อาจารย์ ดร.ชัยพร เขมะภักตะพันธ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้ข้อคิดเห็นและคำแนะนำที่เป็นประโยชน์ต่องานวิจัย และเอาใจใส่นักศึกษาเสมอมา

ขอขอบคุณ รศ.ดร.ไพบูลย์ พฤกษ์สุนันท์ และ อาจารย์ ดร.ประศาสน์ จันทราทิพย์ กรรมการสอบวิทยานิพนธ์ ที่สละเวลามาเป็นกรรมการสอบวิทยานิพนธ์และให้คำแนะนำที่เป็นประโยชน์ต่องานวิจัย และขอบคุณเจ้าหน้าที่ทุกท่านที่ช่วยดำเนินเรื่องต่างๆ ให้เป็นอย่างดี

ขอขอบคุณเพื่อนๆ ร่วมรุ่นทุกๆ คนที่คอยช่วยเหลือมาตลอด

ท้ายสุดนี้ขอขอบคุณ คุณพ่อ คุณแม่ ตลอดจนคนในครอบครัวของผู้วิจัย ที่คอยให้กำลังใจ และสนับสนุนผู้วิจัยในทุกๆ ด้าน ตลอดระยะเวลาการศึกษาจนสำเร็จการศึกษา

ไพศาล ชัยรอด

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	๗
บทคัดย่อภาษาอังกฤษ.....	๘
กิตติกรรมประกาศ.....	๑
สารบัญตาราง.....	๕
สารบัญรูป.....	๗
บทที่	
1. บทนำ.....	1
1.1 ที่มาของงานวิจัย.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ความรู้เบื้องต้นเกี่ยวกับโทรศัพท์ไอพี.....	4
2.2 โพรโตคอลที่เกี่ยวข้องกับ VoIP.....	7
2.3 ความรู้เบื้องต้นเกี่ยวกับ Voice Codec.....	16
2.4 คุณภาพการให้บริการ (QoS) ที่มีผลกระทบต่อคุณภาพเสียง.....	16
2.5 วิธีการวัดคุณภาพเสียงจากเครือข่ายไอพี.....	19
2.6 ประเภทการใช้ VoIP ตาม OSI Layer.....	22
2.7 วิธีการปรับปรุงคุณภาพเสียงบนเครือข่ายไอพี.....	26
2.8 Visual Basic.....	30
2.9 โปรแกรม iperf.....	31
2.10 คำสั่ง Ping.....	32

สารบัญ (ต่อ)

บทที่	หน้า
2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง(ต่อ).....	
2.11 การตรวจสอบและการยอมรับของโปรแกรมที่พัฒนาขึ้น.....	33
2.12 งานวิจัยที่เกี่ยวข้อง.....	36
3. ระเบียบวิธีวิจัย.....	38
3.1 แนวทางการวิจัยและพัฒนา.....	38
3.2 เครื่องมือที่ใช้ในงานวิจัย.....	39
3.3 แผนการดำเนินงาน.....	40
3.4 ขั้นตอนและวิธีดำเนินงาน.....	41
4. การทดสอบระบบ.....	55
4.1 วิธีการทดสอบโปรแกรมโดยการวัดค่า MOS.....	55
4.2 การทดสอบสมมติฐาน.....	63
4.3 วิธีการทดสอบโปรแกรมโดยการวัดค่า Concurrent User.....	71
5. สรุปผลและข้อเสนอแนะ.....	76
5.1 สรุปผลการวิจัย.....	76
5.2 ข้อจำกัดของระบบ.....	77
5.3 ข้อเสนอแนะ.....	77
บรรณานุกรม.....	78
ประวัติผู้เขียน.....	83

สารบัญตาราง

ตารางที่	หน้า
2.1	15
2.2	16
2.3	22
2.4	25
2.5	34
2.6	37
3.1	40
3.2	49
3.3	53
4.1	57
4.2	58
4.3	60
4.4	61
4.5	61
4.6	62
4.7	62
4.8	62
4.9	62
4.10	70
4.11	72
4.12	74

สารบัญรูป

รูปที่	หน้า
2.1 การเชื่อมต่อของอุปกรณ์ในเครือข่ายการส่งสัญญาณเสียงบนไอพี.....	4
2.2 การสื่อสารระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์และโทรศัพท์.....	6
2.3 การสื่อสารแบบโทรศัพท์กับโทรศัพท์.....	7
2.4 แสดงอุปกรณ์ต่าง ๆ ในโครงข่าย H.323.....	8
2.5 แสดงอุปกรณ์ต่าง ๆ ในโครงข่ายโปรโตคอล SIP.....	10
2.6 สถาปัตยกรรมของ SIP.....	11
2.7 รูปตัวอย่างการเกิด Jitter delay.....	18
2.8 ระดับคุณภาพของสัญญาณเสียงเปรียบเทียบในเชิงปริมาณ ทราฟฟิก ที่สูญหายกับความล่าช้าของสัญญาณในการส่งผ่านเครือข่ายไอพี.....	18
2.9 แสดงให้เห็นความสัมพันธ์ระหว่าง R- Factor กับ MOS Score.....	22
2.10 รูปแบบของแพ็กเก็ต VoIP.....	23
2.11 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC G.711.....	24
2.12 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC G.729.....	24
2.13 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC GSM.....	24
2.14 การจัดแถวคอยแบบ First in First Out.....	28
2.15 การจัดแถวคอยแบบ Weighted Fair Queuing.....	28
2.16 การจัดแถวคอยแบบ Priority Queuing.....	29
2.17 การจัดแถวคอยแบบ Custom Queuing.....	29
3.1 แผนภาพการทำงานของโปรแกรมในการคำนวณหาค่า MOS.....	41
3.2 แผนภาพการทำงานของโปรแกรมในการคำนวณหาค่า Concurrent User.....	43
3.3 ลักษณะจอภาพของโปรแกรม.....	44
3.4 ตัวอย่างรูป Property ของ Calculate Mode.....	46

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 ลักษณะของรูปแบบคำสั่งบน Object ที่ชื่อว่า Server.....	47
3.6 ตัวอย่างของคำสั่งในการคำนวณหา Concurrent User.....	48
3.7 ลักษณะของเครือข่ายที่ใช้ในการทดสอบ.....	54
4.1 ลักษณะการใช้โปรแกรมที่พัฒนาวัดหาค่า MOS จากเครือข่ายจำลอง.....	56
4.2 เครื่องมือวัดของ JDSU รุ่น HST 3000.....	56
4.3 ลักษณะการใช้โปรแกรมที่พัฒนาวัดหาค่า Concurrent User จากเครือข่ายจำลอง....	71
4.4 โปรแกรมแสดงจำนวน Concurrent User ที่วัดได้ในเครือข่าย.....	72
4.5 ค่า MOS เมื่อมีจำนวน Session Call เพิ่มขึ้นกรณีที่ไม่มี background traffic.....	73
4.6 ค่า MOS เมื่อมีจำนวน Session Call เพิ่มขึ้นกรณีที่มี background traffic 100%.....	74
4.7 ลักษณะการทดสอบ Concurrent User จากเครือข่ายที่จำลอง.....	75

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP
ชื่อผู้เขียน	ไพศาล ย้งรอด
อาจารย์ที่ปรึกษา	ดร.ชัยพร เชมะภาคะพันธ์
อาจารย์ที่ปรึกษาร่วม	ดร.ธนัญ จารุวิทย์โกวิท
สาขาวิชา	วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา	2553

บทคัดย่อ

ระบบโทรศัพท์ไอพี (Voice over IP: VoIP) เป็นรูปแบบที่มีการใช้งานกันอย่างแพร่หลาย และสามารถประหยัดค่าใช้จ่ายมากกว่ารูปแบบอื่น แต่เนื่องจากเครือข่ายไอพีถูกออกแบบมาสำหรับการใช้งานกับข้อมูลที่หลากหลายรูปแบบทั้งภาพ เสียง และข้อมูลค้ำ จึงอาจมีปัญหาด้านคุณภาพการให้บริการ (Quality of Service: QoS) ขณะใช้งาน VoIP โดยปัจจุบันเครื่องมือที่ใช้วัดคุณภาพการให้บริการ VoIP นั้นมีราคาสูง ทำให้ผู้ให้บริการ VoIP รายเล็ก ผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider) หรือผู้สนใจไม่สามารถซื้อมาใช้งานได้ งานวิจัยนี้จึงมีแนวคิดที่จะออกแบบและพัฒนาโปรแกรมที่สามารถตรวจวัดคุณภาพการให้บริการของ VoIP ที่สามารถใช้งานได้จริงขึ้นมา

โปรแกรมที่พัฒนาขึ้นมาสามารถวัดค่าความล่าช้าทางเวลา (delay) ค่าความแปรผันของความล่าช้าทางเวลา (jitter) และข้อมูลที่สูญหายในเครือข่าย (packet loss) โดยใช้ Visual Basic 6 ในการพัฒนา เมื่อได้ข้อมูลจากการวัดดังกล่าวครบถ้วน โปรแกรมที่พัฒนาจะคำนวณหาคุณภาพการให้บริการให้โดยอัตโนมัติ โดยคุณภาพการให้บริการจะวัดในรูปแบบของ MOS (Mean Opinion Score) งานวิจัยนี้ได้เปรียบเทียบผลของ MOS ที่วัดได้กับเครื่องมือวัดของ JDSU รุ่น HST 3000 ตามหลักสถิติด้วยวิธีการทดสอบสมมุติฐาน ภายใต้สภาพแวดล้อมเครือข่ายการทดสอบที่จำลองขึ้นมา ผลการเปรียบเทียบค่าทั้งสองส่วนใหญ่มักจะไม่แตกต่างกันที่ระดับนัยสำคัญ 95% นอกจากนี้โปรแกรมที่พัฒนายังสามารถคำนวณหาจำนวนผู้ใช้งาน VoIP ที่ระบบสามารถรองรับได้โดยมีคุณภาพการให้บริการอยู่ในเกณฑ์ดี

Thesis Title	Development of a Quality of Service Measurement Program for Voice over IP usage
Author	Phaisan Youngrod
Thesis Advisor	Chiyaporn Khemapatapan, Ph.D
Co-Thesis Advisor	Tanun Jaruvitayakovit, Ph.D
Department	Computer and Telecommunication Engineering
Academic Year	2010

ABSTRACT

VoIP technology is currently popular due to its cost effective solution. Normally, Internet Protocol (IP) is designed for delivery of multimedia traffic i.e. video, voice and data. As the result, the user VoIP may face the Quality of Service (QoS) problem. Moreover, VoIP QoS measurement tool is very costly. So, it quite difficult for small scale VoIP operators or Internet Service Providers (ISP) or interested people to own the measurement tool. This research is aim to design and develop a program that can measure VoIP QoS in the real environment.

The developed program can measure time delay, delay jitter and packet loss in the computer network using Visual Basic 6 as the development tool. The measured variables are used to calculate the QoS parameter. This paper uses Mean Opinion Score (MOS) as the QoS parameter. The calculated MOS is statistically compared with the test tool model HST3000 from JDSU. The proof was done by Hypothesis test under a simulated network scenario. Comparison results show that both values are not significantly different at 95% confident interval. In addition, the developed program can calculate the concurrent VoIP users that a system can supported with good quality of service.

บทที่ 1

บทนำ

1.1 ที่มาของงานวิจัย

ในปัจจุบันความต้องการเชื่อมต่อคอมพิวเตอร์ต่างๆ เข้ากับเครือข่ายมีจำนวนมากขึ้น และเมื่อเครือข่ายคอมพิวเตอร์ต่างๆ มาเชื่อมต่อเข้าด้วยกันจนเป็นเครือข่ายที่มีขนาดใหญ่ก็มีมากขึ้น ด้วยเช่นกัน ตัวอย่างที่สำคัญของเครือข่ายคอมพิวเตอร์ที่เชื่อมต่อเข้าด้วยกัน โดยใช้เครือข่ายไอพีคือ เครือข่ายอินเทอร์เน็ต (Internet) ที่เป็นเครือข่ายคอมพิวเตอร์ที่ใหญ่ที่สุดในโลก

เครือข่ายไอพีมีการเติบโตขึ้นอย่างรวดเร็วพร้อมๆ กับการเชื่อมต่อของคอมพิวเตอร์กับเครือข่าย มีการนำเครือข่ายอินเทอร์เน็ตไปใช้ทั้งในแง่เพื่อการศึกษาและด้านพาณิชย์ อันส่งผลให้มีการพัฒนาเทคนิคต่างๆ หรือบริการต่างๆ ขึ้นเพื่อใช้ประโยชน์จากเครือข่ายไอพี และหนึ่งในการพัฒนาที่สำคัญบนเครือข่ายไอพีคือ การพัฒนาให้เครือข่ายไอพีสามารถส่งผ่านสัญญาณเสียงแทนเครือข่ายโทรศัพท์แบบเดิม ทั้งนี้ก็ด้วยเหตุผลหลักที่ว่า การส่งสัญญาณเสียงผ่านเครือข่ายไอพีจะช่วยประหยัดค่าใช้จ่ายในการโทรศัพท์ขึ้นอย่างมาก เพราะไม่จำเป็นต้องเสียค่าใช้จ่ายในราคาของการโทรศัพท์ทางไกล แต่หากเป็นเพียงแค่การเสียค่าใช้จ่ายในราคาท้องถิ่นเท่านั้น สิ่งนี้เองที่เป็นตัวผลักดันให้เทคโนโลยีของการพยายามส่งสัญญาณเสียงผ่านบนเครือข่ายไอพีมีการพัฒนาอย่างรวดเร็ว

ระบบโทรศัพท์ไอพี (Voice over IP: VoIP) เป็นรูปแบบที่มีการใช้งานกันอย่างแพร่หลาย และสามารถประหยัดค่าใช้จ่ายมากกว่ารูปแบบอื่น แต่เนื่องจากเครือข่ายไอพีถูกออกแบบมาสำหรับการใช้งานกราฟฟิกที่หลากหลายรูปแบบทั้งภาพ เสียงและข้อมูล จึงอาจจะมีปัญหาด้านคุณภาพการให้บริการ (Quality of Service: QoS) ขณะใช้งาน VoIP อาจทำให้เกิดความล่าช้าหรือมีแพ็กเก็ตสูญหายเมื่อส่งข้อมูลเสียงได้ สิ่งที่สำคัญคือระบบจะต้องสามารถรับประกันคุณภาพในการให้บริการ ในระดับที่ยอมรับได้

แต่หากมองย้อนกลับไปที่เทคโนโลยีที่ใช้สร้างเครือข่ายไอพี จะพบว่าโดยพื้นฐานของเครือข่ายไอพีนั้น ไม่เหมาะสมที่จะใช้ส่งสัญญาณเสียงโดยเฉพาะอย่างยิ่งในกรณีที่ไม่มีการจัดการเครือข่ายที่ดี เพราะเครือข่ายไอพีอาจจะไม่สามารถรับประกันว่าข้อมูลเสียงนั้นจะเดินทางไปถึงปลายทางตามเวลาที่กำหนดไว้ อันส่งผลให้คุณภาพของสัญญาณเสียงไม่เป็นที่ยอมรับ โดยเฉพาะเมื่อมีการส่งสัญญาณเสียงไปบนเครือข่ายไอพีแบบที่เป็นเครือข่ายอินเทอร์เน็ตสาธารณะ (Public Internet) เนื่องจากความที่ไม่มีใครคนใดคนหนึ่งเป็นเจ้าของเครือข่ายสาธารณะนี้ ทำให้การ

ออกแบบเครือข่ายเพื่อควบคุมคุณภาพเสียงตลอดเส้นทางทำได้ลำบาก แต่หากเป็นเครือข่ายของผู้ให้บริการที่สร้างเครือข่ายไอพีขึ้นมาโดยเฉพาะ จะทำให้สามารถควบคุมคุณภาพการส่งข้อมูลบนเครือข่ายไอพีได้

ด้วยเหตุนี้ผู้วิจัยจึงมีแนวคิดที่จะพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP ที่สามารถใช้งานได้จริงขึ้นมา สำหรับผู้สนใจทุกๆ ไป ที่ต้องการใช้งานโทรศัพท์ไอพี เพราะปัจจุบันเครื่องมือที่ใช้วัดคุณภาพการให้บริการ VoIP นั้นมีราคาสูง ทำให้ผู้ให้บริการ VoIP รายเล็กหรือผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider) หรือผู้สนใจไม่สามารถซื้อมาใช้งานได้ โดยศึกษารูปแบบและปัจจัยต่างๆ ที่จำเป็นต่อการคำนวณและเพื่อข้อมูลเบื้องต้นในการออกแบบเครือข่ายไอพี ในการใช้ส่งสัญญาณเสียงเป็นหลัก

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อออกแบบและพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP
2. เพื่อออกแบบและพัฒนาโปรแกรมที่สามารถวัดจำนวนผู้ใช้งาน VoIP พร้อมกัน (Concurrent user) โดยที่ยังมีคุณภาพให้บริการในระดับที่ยอมรับได้

1.3 ขอบเขตของงานวิจัย

ขอบเขตของวิทยานิพนธ์ฉบับนี้ มีขอบเขตการวิจัยดังนี้

1. โปรแกรมที่พัฒนาสามารถวัด ข้อมูลสูญหายในเครือข่าย (packet loss) ค่าความล่าช้าทางเวลา (delay) ค่าความแปรผันของความล่าช้าทางเวลา (jitter) โดยใช้โปรแกรมที่เป็นโปรแกรมฟรี (Freeware) มาช่วย
2. โปรแกรมที่พัฒนาสามารถคำนวณค่าคุณภาพการให้บริการ (MOS) ตามมาตรฐาน ITU-T recommendation P.800
3. เปรียบเทียบค่าคุณภาพการให้บริการที่คำนวณได้กับค่าที่วัดได้จากเครื่องมือวัด JDSU รุ่น HST 3000
4. โปรแกรมที่พัฒนาสามารถคำนวณผู้ใช้งานพร้อมกัน ที่เครือข่ายสามารถรองรับการใช้งาน VoIP ได้โดยมีคุณภาพการให้บริการ (QoS) ที่ดี โดยโปรแกรมที่พัฒนาจะรองรับเฉพาะการใช้งาน VoIP ภายในองค์กร (LAN) เท่านั้น โดยโปรแกรมที่พัฒนาจะรองรับเฉพาะ CODEC แบบ G.711, G.729 และ GSM เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องมือต้นแบบที่สามารถวัดคุณภาพ การให้บริการสำหรับการใช้งาน VoIP
2. ได้ทราบถึงปัจจัยที่มีผลต่อการคำนวณหาคุณภาพของเครือข่าย ในการใช้งาน VoIP
3. ได้ทราบถึงขนาดความจุของเครือข่าย ในแต่ละเส้นทาง เพื่อนำไปเป็นข้อมูลเบื้องต้น ออกแบบเครือข่ายได้
4. สามารถนำความรู้ที่ได้และค้นคว้าจากแหล่งข้อมูลต่างๆ มาใช้งานจริงในการทำงาน

บทที่ 2

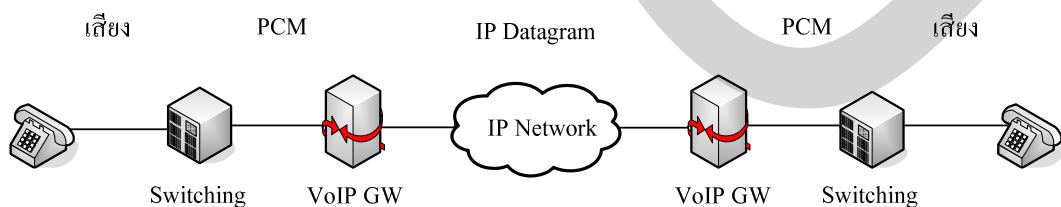
แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

2.1 ความรู้เบื้องต้นเกี่ยวกับโทรศัพท์ไอพี (Minoli Daniel, 1998)

เนื่องจากเครือข่ายไอพีไม่ได้ถูกออกแบบมาให้สามารถรองรับการส่งข้อมูลในรูปแบบของงานประยุกต์ที่ไวต่อเวลาหน่วง (Delay Sensitive Application) ดังนั้นหากมีการออกแบบเครือข่ายไอพีที่ไม่ถูกต้องจะทำให้การส่งผ่านสัญญาณเสียงบนเครือข่ายไอพีไม่มีคุณภาพดีพอหรืออาจไม่สามารถใช้งานได้เลย

การส่งสัญญาณเสียงผ่านเครือข่ายไอพี เป็นระบบที่นำสัญญาณเสียงที่ผ่านการดิจิไทซ์ (Digitize) มาบีบอัดแล้วบรรจุลงในแพ็คเกจไอพี (IP Packet) เพื่อทำการส่งผ่านทางเครือข่ายไอพี โดยสามารถใช้งานร่วมกับทราฟฟิกไอพี (IP Traffic) อื่นๆ ซึ่งการบีบอัดสัญญาณเสียงให้เล็กลง ทำให้ลดปริมาณการส่งสัญญาณ ลดความล่าช้าของสัญญาณ และลดช่องสัญญาณในการส่งสัญญาณ แต่คุณภาพเสียงที่ได้ก็จะลดลงเช่นกัน

การประยุกต์ส่งสัญญาณเสียงผ่านเครือข่ายไอพีสามารถกระทำได้โดยอาศัยอุปกรณ์ VoIP Gateway ทำหน้าที่แปลงข้อมูลสัญญาณเสียงที่อยู่ในรูปของ Pulse Code Modulation (PCM) ให้อยู่ในรูปของข้อมูลที่พร้อมจะบรรจุใน IP Datagram เพื่อให้สามารถส่งต่อไปในเครือข่ายไอพีดังแสดงในรูปที่ 2.1 ซึ่งเป็นรูปการเชื่อมต่อระหว่างอุปกรณ์ VoIP Gateway, PSTN Switching และเครือข่ายไอพีเข้าด้วยกัน



รูปที่ 2.1 การเชื่อมต่อของอุปกรณ์ในเครือข่ายการส่งสัญญาณเสียงบนไอพี (Minoli Daniel, 1998)

เครือข่ายไอพีมีรูปแบบการส่งข้อมูลที่เรียกว่า Connectionless ซึ่งเป็นรูปแบบการส่งที่ไม่ต้องมีการสร้างการเชื่อมต่อ (Connection) ระหว่างผู้ส่งและผู้รับกล่าว คือ ข้อมูล 2 ชุดที่ส่งจากผู้

ส่งไปยังผู้รับอาจเดินทางไปคนละเส้นทางจากผู้ส่งไปยังผู้รับก็ได้ และในขณะเดียวกันก็ไม่มี การควบคุมการไหล (Flow Control) เพื่อควบคุมความน่าเชื่อถือของการส่งข้อมูลอันส่งผลให้การส่งข้อมูลแบบ Connectionless ไม่มีการรับประกันว่าข้อมูลสามารถส่งได้ถึงปลายทางหรือไม่ อาจมีการสูญหายของข้อมูล ส่งข้อมูลซ้ำซ้อนหรือส่งข้อมูลผิดพลาด ซึ่งหากข้อมูลที่ถูกส่งต้องขึ้นอยู่กับเวลา (Delay Sensitive) เช่นเสียงพูด วิดีโอ การส่งแบบ Connectionless จะมีโอกาสทำให้เสียงหรือภาพที่ส่งขาดหายไปเป็นบางช่วงได้

รายละเอียดของเนื้อหาในบทนี้จะเกี่ยวข้องกับปัจจัยต่างๆ ที่จะส่งผลต่อคุณภาพการส่งสัญญาณเสียงบนเครือข่ายไอพี และแนวทางการออกแบบเครือข่ายไอพี เพื่อให้ส่งสัญญาณเสียงบนเครือข่ายไอพีได้อย่างมีคุณภาพ

2.1.1 องค์ประกอบของ VoIP (บงการ หอมานัน, 2547)

2.1.1.1 Software IP Telephony (Soft phone) หรือ โทรศัพท์ไอพี อาจจะเป็นเครื่องคอมพิวเตอร์ที่ได้รับการติดตั้งโปรแกรม หรืออุปกรณ์ที่ได้รับการออกแบบขึ้นมาสำหรับการใช้งานโทรศัพท์ผ่านอินเทอร์เน็ตโดยเฉพาะ

2.1.1.2 VoIP Gateway เป็นเครื่องเซิร์ฟเวอร์ที่ใช้งานสำหรับให้บริการโทรศัพท์ผ่านระบบอินเทอร์เน็ต เพื่อเป็นตัวกลางในการเชื่อมต่อเข้ากับเครื่องโทรศัพท์ผู้ชุมสายโทรศัพท์สาธารณะ PSTN (Public Switched Telephone Network) กับระบบเครือข่ายอินเทอร์เน็ตอย่างเครือข่ายไอพี ซึ่งการจะใช้งานโทรศัพท์ไอพีต้องอาศัยอุปกรณ์ตัวนี้เป็นตัวกลาง VoIP Gateway เป็นอุปกรณ์ในรูปแบบเราเตอร์ที่มีคุณสมบัติเช่นเดียวกับเราเตอร์ที่ใช้งานกันอยู่ แต่มีคุณสมบัติที่ ถูกเพิ่มเติมให้รองรับโปรโตคอลการสื่อสารของ VoIP นั่นก็คือ โปรโตคอล H.323, SIP เป็นต้น

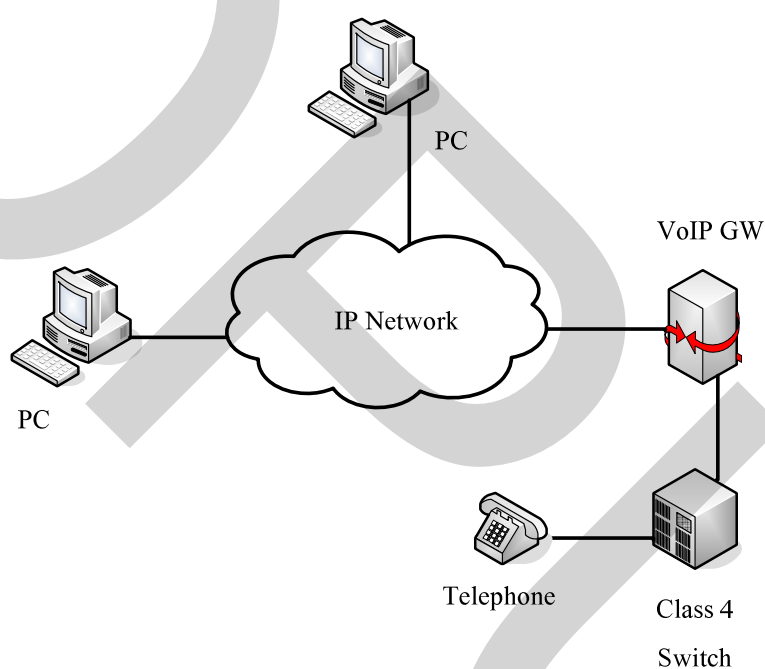
2.1.1.3 SIP Server/Gatekeeper เป็นเครื่องเซิร์ฟเวอร์ที่ถูกเชื่อมต่อเข้ากับระบบอินเทอร์เน็ต เป็นตัวกลางที่ใช้บริหารจัดการและควบคุมการให้บริการของ VoIP Gateway กับเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมสำหรับใช้งานสื่อสาร VoIP หรือเครื่องโทรศัพท์แบบไอพี แบบดั้งเดิมที่ต้องการในการให้บริการ VoIP โดยทั่วไป แบนด์วิดท์ที่ต้องการขึ้นอยู่กับชนิดของการเข้ารหัสและบีบอัดระบบเสียง (Voice Codec) ซึ่ง VoIP Packet มีขนาดเล็กมากแต่แบนด์วิดท์ส่วนใหญ่จะถูกใช้ไปกับเฮดเดอร์ของ IP และ UDP ซึ่งมีขนาดใหญ่กว่ามาก

2.1.2 ลักษณะการให้บริการโดยทั่วไปของ VoIP

2.1.2.1 แบบเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ (PC-to-PC) รูปแบบการใช้งาน VoIP แบบนี้ส่วนใหญ่จะเป็นการพัฒนาต่อจากโปรแกรมประเภท Chat หรือที่เรียกกันว่า Instant messaging ที่แต่เดิมเป็นการส่งข้อความเป็นตัวหนังสือ โดยเพิ่มฟังก์ชันในการพูดคุยด้วยเสียงเข้าไปเช่น Google Talk, MSN กล่าวง่ายๆ คือวิธีนี้จำเป็นต้องอาศัยเครื่องคอมพิวเตอร์

ทั้งที่ต้นทางและปลายทาง ซึ่งรูปแบบนี้เป็นวิธีการสื่อสารที่ไม่ต้องเสียค่าบริการโทรศัพท์แต่อย่างใด และต้องนัดแนะเวลาในการใช้อินเตอร์เน็ตในเวลาเดียวกัน ดังแสดงในรูปที่ 2.2

2.1.2.2 แบบเครื่องคอมพิวเตอร์สู่โทรศัพท์ (PC-to-Phone) การใช้งาน VoIP ประเภทนี้ เครื่องคอมพิวเตอร์ต้องติดตั้งโปรแกรมที่เรียกว่า Soft phone ซึ่งเป็นโปรแกรมที่จำลองเครื่องโทรศัพท์บนคอมพิวเตอร์เพื่อให้ส่งข้อมูลเสียงไปยังปลายทางที่ไม่ใช่คอมพิวเตอร์ได้ ปัจจุบันมีผู้ให้บริการด้วย Soft phone หลายรายเช่น Skype, TRUE, CAT และ TOT โดยผู้ใช้บริการต้องเสียค่าบริการตามเวลาใช้งานจริง ดังแสดงในรูปที่ 2.2 (Franklin D. Ohrtman, Jr)

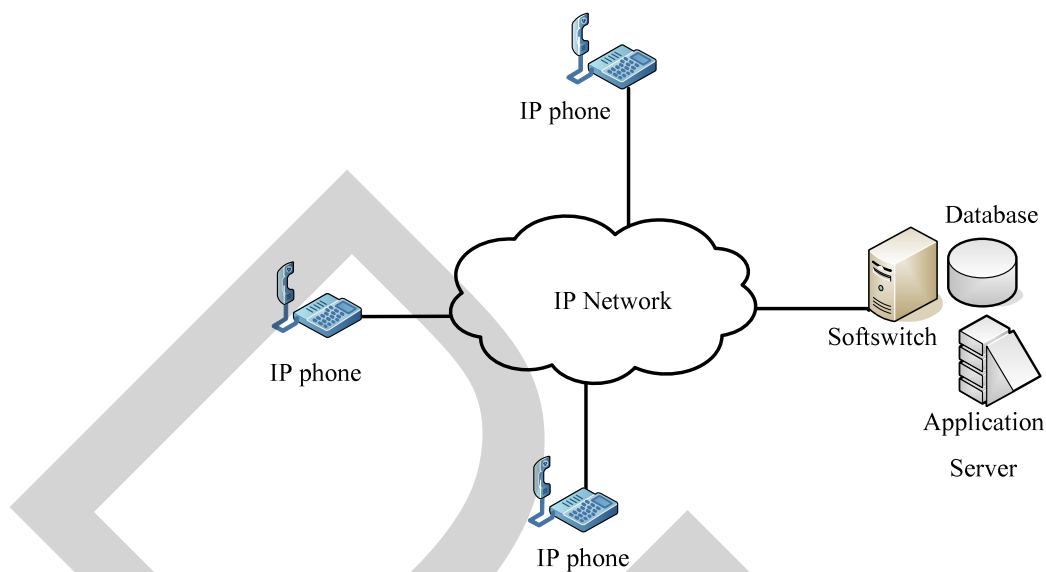


รูปที่ 2.2 การสื่อสารระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์และโทรศัพท์

2.1.2.3 แบบเครื่องโทรศัพท์กับเครื่องโทรศัพท์ (Phone-to-Phone) คือการใช้งาน VoIP แบบเครื่องโทรศัพท์หาเครื่องโทรศัพท์กันเองโดยโทรศัพท์ VoIP มีอยู่สองประเภทคือ

1) โทรศัพท์ปกติที่ต้องเชื่อมต่อกับ ATA (Analog Telephone Adapter) เนื่องจากโทรศัพท์ปกติที่ใช้งานในปัจจุบันไม่สามารถเชื่อมต่อกับโครงข่ายไอพีได้โดยตรง จึงต้องแปลงข้อมูลของเสียงเป็นข้อมูลดิจิทัลแพ็กเก็ตเสียก่อนโดยอาศัยอุปกรณ์ ATA (FXS Gateway) ซึ่งสามารถเชื่อมต่อ VoIP ทั้งโปรโตคอล SIP และ H.323 ผ่านหัวต่อ RJ-11 และ RJ-45 ได้

2) โทรศัพท์ที่สามารถใช้ติดต่อผ่านโครงข่ายไอพีได้เลย (IP Phone) เป็นวิธีที่สะดวกที่สุดในการใช้งาน ดังแสดงในรูปที่ 2.3 (Franklin D. Ohrtman, Jr)



รูปที่ 2.3 การสื่อสารแบบโทรศัพท์กับโทรศัพท์

2.1.3 ขั้นตอนการทำงานของ VoIP

เมื่อผู้พูดโทรศัพท์จากเครื่องโทรศัพท์ธรรมดา หรือพูดผ่านไมโครโฟนที่ถูกต่อเข้ากับการ์ดเสียงของเครื่องคอมพิวเตอร์คลื่นสัญญาณเสียงแบบอนาล็อกก็จะได้รับการแปลงเป็นสัญญาณดิจิทัลจากนั้นจะถูกบีบอัดด้วยตัวอุปกรณ์ VoIP Gateway และเมื่อผ่าน VoIP Gateway แล้วก็จะถูกส่งต่อไปยัง Gatekeeper เพื่อค้นหาเครื่องปลายทางที่จะติดต่อแล้วจะแปลงเป็นแพ็กเกจข้อมูลส่งออกไปบนเครือข่ายอินเทอร์เน็ตเมื่อถึงปลายทางก็จะถูก VoIP Gateway ปลายทางก็จะทำการย้อนกระบวนการทั้งหมดเพื่อให้ฝั่งรับปลายทางต่อไป

2.2 โปรโตคอลที่เกี่ยวข้องกับ VoIP (สาริตพงษ์ พุทธิประเสริฐและคณะ, 2544)

2.2.1 โปรโตคอล H.323 เป็น VoIP โปรโตคอลตัวแรกๆ ที่ได้รับความนิยมกันอย่างแพร่หลาย โดยเริ่มแรกได้รับการพัฒนามาจาก ITU-T's (International Telecommunications Union) standard เพื่อให้บริการ Multimedia conferencing บนโครงข่าย LAN และได้พัฒนาต่อมาเพื่อให้รองรับบริการ VoIP โดย H.323 เวอร์ชันแรกได้ประกาศในปี 1996 และตลอดเวลามากกว่า 10 ปี H.323 โปรโตคอลได้มีการแก้ไขเพิ่มเติมเพื่อให้มีบริการเสริมหลากหลายมากขึ้น มีความเสถียรสูงและรองรับการขยายโครงข่ายในอนาคตได้ดีขึ้น ซึ่งในปัจจุบัน H.323 โปรโตคอลได้พัฒนาอยู่ที่ Version 5.

ลักษณะของโปรโตคอล H.323

1) เป็น Peer-to-Peer โปรโตคอล ซึ่งหมายถึงอุปกรณ์ปลายทางทั้งสองฝั่งมีความเท่าเทียมกัน(ซึ่งจะแตกต่างจาก MGCP/H.248 โปรโตคอลซึ่งเป็น Master / Slave โปรโตคอล) ดังแสดงในรูปที่ 2.4 (สำนักงานคณะกรรมการกิจการ โทรคมนาคมแห่งชาติ, 2550)

2) โปรโตคอล H.323 เป็น VoIP protocol suite, หมายถึงภายใต้ H.323 จะมีโปรโตคอลย่อยอื่นๆ เพื่อระบุรายละเอียดการทำงานในแต่ละส่วนเพื่อให้สามารถให้บริการได้ตัวอย่าง โปรโตคอลย่อยที่สำคัญของ H.323 ได้แก่ H.225 RAS signaling, H.225.0 Call signaling, H.245 Control signaling, Media coding, H.246 for Interworking issues and etc.

3) มีการ Coding แบบ ASN.1 (binary form)

4) อุปกรณ์หลักใน โปรโตคอล H.323 ประกอบด้วย

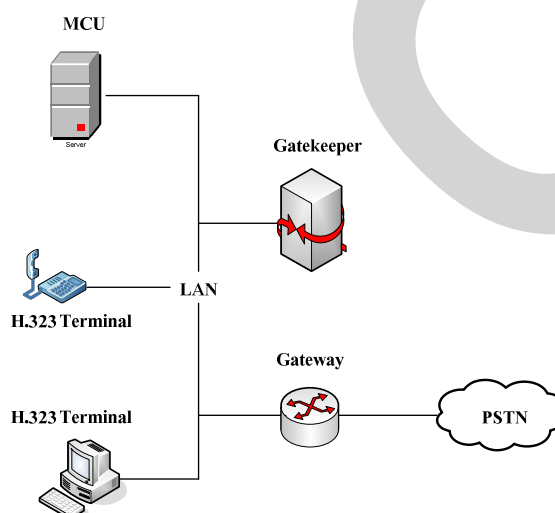
4.1) Terminal เป็นอุปกรณ์ปลายทางของผู้ใช้งาน

4.2) Gatekeeper เป็นอุปกรณ์ควบคุมการเชื่อมต่อของอุปกรณ์ต่างๆ ภายในโครงข่าย H.323

4.3) Gateway เป็นอุปกรณ์ใช้เชื่อมต่อไปยังโครงข่ายอื่นๆ เช่น โครงข่ายโทรศัพท์พื้นฐาน (Fixed Network)

4.4) MCU (Multi Point Control Units) เป็นอุปกรณ์เชื่อมต่อระหว่างผู้ใช้บริการหลายๆ คนให้สามารถ Conferencing ร่วมกันได้

4.5) PSTN (Public Switch Telephone Network) หรือ เครือข่ายโทรศัพท์พื้นฐาน หรือเรียกง่ายๆว่าเครือข่ายโทรศัพท์บ้าน



รูปที่ 2.4 แสดงอุปกรณ์ต่างๆ ในโครงข่าย H.323

2.2.2 โพรโทคอล SIP (Session Initial Protocol) เป็นโพรโทคอลที่ใช้งานสำหรับ IP Telephony แบบ Peer-to-Peer เช่นเดียวกับ H.323 พัฒนาโดย IETF's standard (Internet Engineering Task Force) โดย SIP เป็นโพรโทคอลในชั้นแอปพลิเคชันซึ่งทำหน้าที่ในการสร้าง ลีนสุดและเปลี่ยนแปลงแก้ไขเซสชันของพหุสื่อ (Multimedia session) หรือการเรียก ซึ่งรวมถึง Internet Telephony การประชุมแบบพหุสื่อ (Multimedia conference) และแอปพลิเคชันอื่นที่คล้ายคลึงกัน SIP เป็นโพรโทคอลไคลเอนท์-เซิร์ฟเวอร์ (client-server) โดยการใช้ส่งข้อมูลในรูปแบบของตัวอักษร (text based) เช่นเดียวกับ โพรโทคอล HTTP (Hypertext Transfer Protocol) รวมทั้งยังมีกลไกที่คล้ายคลึงกัน ทำให้สามารถใช้เซิร์ฟเวอร์และกลไกที่มีอยู่บางอย่างของ HTTP ได้โดยรองรับได้มากกว่าหนึ่งอุปกรณ์ในคราวเดียวกัน โดยบริการที่รองรับได้ไม่จำกัดเฉพาะ VoIP session เท่านั้น แต่ SIP โพรโทคอลยังสามารถรองรับบริการอื่นได้อีกเช่น Instant Messaging , Presence และอื่นๆ

โดย IETF ประกาศมาตรฐาน SIP ครั้งแรกในปี 1999 และได้มีการพัฒนาเพิ่มเติมมาเรื่อยๆ ดังตัวอย่างต่อไปนี้

- 1) ในปี 1999 ได้ออกแบบ มาตรฐาน SIP RFC 2543, 153 ASCII pages
- 2) ในปี 2000 3GPP(Third generation mobile) ได้เลือก SIP โพรโทคอลเป็นมาตรฐานในการสื่อสารบนโครงข่ายโมบาย ซึ่งในอนาคตจะส่งผลให้ โพรโทคอล SIP เป็นที่นิยมอย่างแพร่หลายมากขึ้นและมีข้อกำหนดเกี่ยวกับโพรโทคอล SIP อื่นๆเพิ่มเติมอีกหลายตัวเพื่อให้โพรโทคอล SIP สามารถใช้ในโครงข่ายโมบายได้อย่างมีประสิทธิภาพ
- 3) ในปี 2002 ได้ออก มาตรฐาน SIP ตัวใหม่ RFC 3261, 270 ASCII pages
- 4) จนถึงปัจจุบันก็ยังคงมีข้อกำหนดอื่นๆเกี่ยวกับโพรโทคอล SIP ออกมาอยู่เรื่อยๆ

2.2.2.1 ลักษณะของโพรโทคอล SIP

- 1) เป็นโพรโทคอล Peer-to-Peer เช่นเดียวกับโพรโทคอล H.323 และดูจะเหมือนว่าจะมาทดแทน โพรโทคอล H.323 ในอนาคตอันใกล้
- 2) มีการ Coding แบบ ISO UTF-8 (Text based) สามารถอ่านข้อมูลในโพรโทคอลได้ง่าย
- 3) รูปแบบข้อมูลมีลักษณะใกล้เคียงที่ใช้ใน Internet protocol (HTTP syntax) มีความยืดหยุ่นสูง
- 4) การระบุปลายทาง (addressing scheme) ใช้ ULR ซึ่งทำให้สามารถรองรับได้ทั้ง phone number, IP address และ e-mail address
- 5) อุปกรณ์หลักใน SIP โพรโทคอลดังแสดงในรูปที่ 2.5 (สำนักงานคณะกรรมการกิจการโทรคมนาคมแห่งชาติ, 2550) ซึ่งจะประกอบด้วย

2.2.2.1.1 User Agents เป็นอุปกรณ์ปลายทางสำหรับผู้ใช้ (SIP user) สามารถแบ่งได้ 2 ประเภท

1) User Agent Client (UAC) โดย UAC จะเป็นผู้เริ่มต้นร้องขอ SIP request

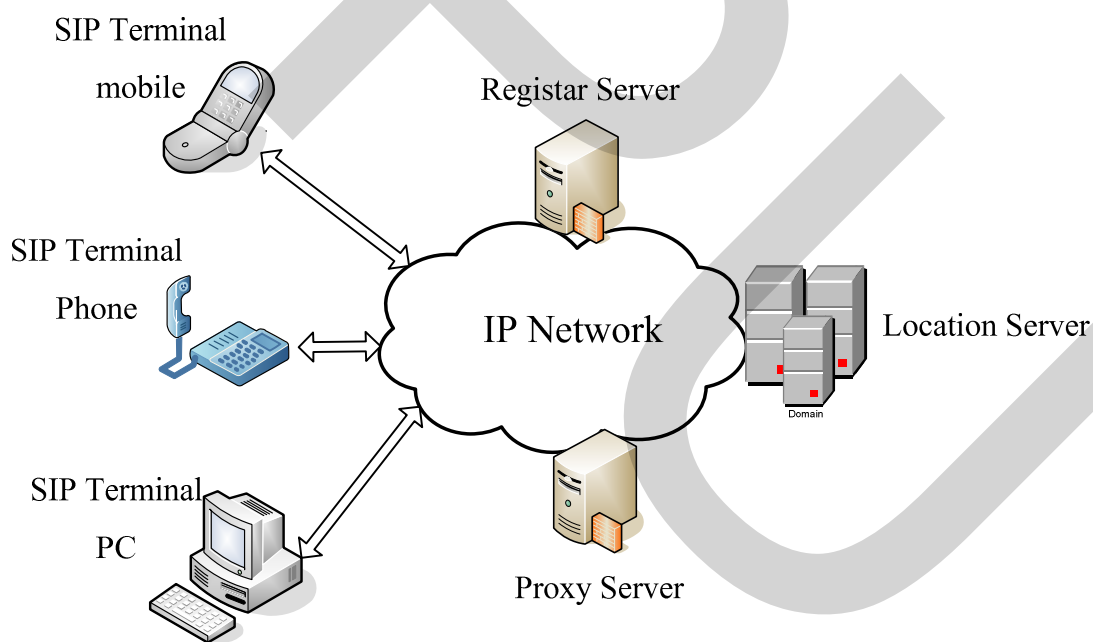
2) User Agent Server (UAS) โดย UAS จะเป็นผู้รับ SIP request และตอบกลับ

2.2.2.1.2 Network Servers เป็นอุปกรณ์ในโครงข่ายมีอยู่ 3 ประเภทด้วยกันคือ

1) Registrar server ทำหน้าที่รองรับการลงทะเบียนของผู้ใช้บริการ (SIP user)

2) proxy server ทำหน้าที่รองรับการส่งต่อข้อมูลไปยัง server ถัดไป

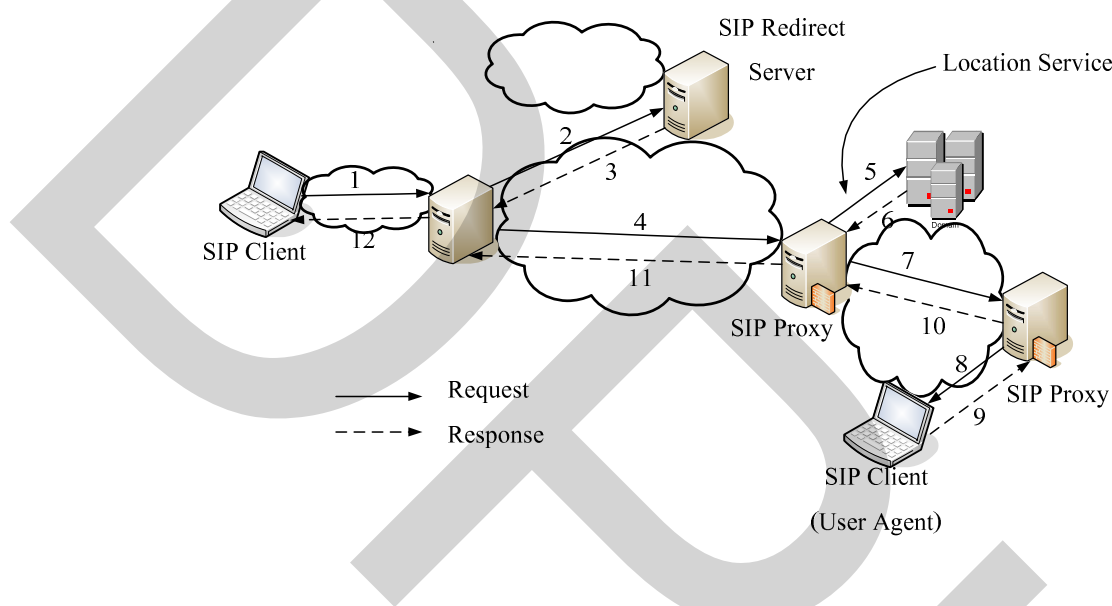
3) redirection server ทำหน้าที่รองรับ ตัดสินใจระบุทิศทางและที่อยู่ของ server ถัดไป



รูปที่ 2.5 แสดงอุปกรณ์ต่างๆ ในโครงข่ายโพรโตคอล SIP

2.2.2.2 สถาปัตยกรรมและองค์ประกอบของโพรโตคอล SIP (SIP Architecture & Components) โดย SIP เป็นโพรโตคอลไคลเอนต์-เซิร์ฟเวอร์ ไคลเอนต์จะทำหน้าที่ส่งคำร้องขอให้กับเซิร์ฟเวอร์เพื่อทำการประมวลผลแล้วจึงตอบสนองมายังไคลเอนต์ ในการส่งข้อมูลร้องขอเมสเสจ

อาจจะถูกส่งผ่านเซิร์ฟเวอร์หลายตัว จนกระทั่งถึงเซิร์ฟเวอร์ที่สามารถตอบสนองคำร้องของไคลเอนต์ได้ในระบบ SIP จะมีองค์ประกอบที่ทำหน้าที่ของไคลเอนต์และเซิร์ฟเวอร์ องค์ประกอบเหล่านี้จะทำการติดต่อสื่อสารกันโดยตรงโดยใช้แมสเสจ SIP ซึ่งมีสถาปัตยกรรมดังรูปที่ 2.6 (สาธิตพงษ์ พุทธิประเสริฐและคณะ, 2544)



รูปที่ 2.6 สถาปัตยกรรมของ SIP

ใน SIP จะแบ่งองค์ประกอบเป็น 2 ชนิดหลักคือ user agent และ network agent ดังรายละเอียดดังต่อไปนี้

2.2.2.2.1 User Agent เป็น endpoint ที่ทำหน้าที่แทนผู้ใช้ในการติดต่อสื่อสาร เนื่องจากว่าผู้ใช้ต้องสามารถเริ่ม การเรียก หรือตอบสนองต่อการเรียกที่เข้ามา ดังนั้น user agent ควรจะสามารถทำหน้าที่เป็น ได้ทั้ง ไคลเอนต์และเซิร์ฟเวอร์ในกรณีที่มีการเริ่ม การเรียก ผู้ใช้จะทำหน้าที่เป็นไคลเอนต์เพื่อทำการร้องขอการสื่อสาร ไปยังผู้ถูกเรียกซึ่งจะทำหน้าที่เป็นเซิร์ฟเวอร์ในการตอบสนองการร้องขอ โดยทั่วไป user agent จึงประกอบด้วยส่วนที่ทำหน้าที่เป็นไคลเอนต์และเซิร์ฟเวอร์ดังนี้

- 1) ผู้ใช้ฝั่งไคลเอนต์ (User Agent Client: UAC) จะทำหน้าที่ในการเริ่ม การเรียก โดยการส่งแมสเสจร้องขอไปยังผู้ถูกเรียกโดยผ่านทางเซิร์ฟเวอร์เครือข่าย
- 2) ผู้ใช้ฝั่งเซิร์ฟเวอร์ (User Agent Server: UAS) จะทำหน้าที่ในการรับคำร้องขอและตอบสนองต่อคำร้องขอโดยจะรอการตอบสนองจากผู้ใช้ ซึ่งการตอบสนองอาจจะ

เป็นการยอมรับหรือปฏิเสธ การเรียก ในกรณีที่ผู้ใช้มีการใช้เทอร์มินัลหลายตัว ผู้ใช้ยังอาจจะกำหนดให้ UAS ทำการส่งต่อการเรียก (redirect) ไปยังที่ UAS อื่น ที่ผู้ใช้ใช้งานอยู่จริง

2.2.2.2.2 Network server เป็นเซิร์ฟเวอร์ภายในเครือข่ายซึ่งจะทำหน้าที่ในการจัดการกับแอสเสกที่ได้รับ โดยอาจจะได้รับจาก user agent หรือ network server อื่นๆ การจัดการกับแอสเสกจะขึ้นกับชนิดของเซิร์ฟเวอร์ ซึ่งมี 2 ชนิด

1) พร็อกซีเซิร์ฟเวอร์ (Proxy server) จะทำการกำหนดเอนทิตีที่ได้รับข้อมูลต่อไป โดยอาจจะเป็น UAS หรือ network server ก็ได้ จากนั้นเซิร์ฟเวอร์จะเป็นผู้ทำการร้องขอไปยังเอนทิตีนั้น พร้อมกับข้อมูลตอบสนองให้กับ UAC (หรืออาจจะเป็น network server อื่นที่ส่งข้อมูลร้องขอมา) เพื่อระบุว่ากำลังรอการตอบสนองจากผู้ถูกเรียก เมื่อเซิร์ฟเวอร์ได้รับการตอบสนองจากผู้ถูกเรียกหรือ UAS เซิร์ฟเวอร์จึงจะส่งแอสเสกตอบสนองต่อกลับไปให้กับ UAC ดังรูปที่ 2.6 เซิร์ฟเวอร์ชนิดนี้จะทำหน้าที่เป็นทั้งไคลเอนต์และเซิร์ฟเวอร์ ในกรณีที่ส่งแอสเสกร้องขอจะเป็นไคลเอนต์ส่วนในกรณีที่ส่งข้อมูลตอบสนองจะเป็นเซิร์ฟเวอร์

2) รีไดเร็กต์ เซิร์ฟเวอร์ (Redirect server) เมื่อเซิร์ฟเวอร์ได้รับแอสเสกร้องขอแล้วจะกำหนดเอนทิตีที่จะรับข้อมูลต่อไป จากนั้นเซิร์ฟเวอร์ก็จะส่งแอสเสกของเอนทิตีนั้นไปให้กับ UAC หรือ network server ที่ส่งข้อมูลร้องขอมา เมื่อ UAC (network server) ได้รับแอสเสกแล้วจึงจะส่งคำร้องไปยังเซิร์ฟเวอร์นั้นด้วยตนเองดังรูปที่ 2.6

เนื่องจากว่าผู้ใช้ อาจจะมีการเปลี่ยนแปลงเทอร์มินัลที่ใช้งานได้ ดังนั้น network server จึงจะต้องสามารถกำหนดเอนทิตีที่รับข้อมูลเพื่อให้สามารถส่งแอสเสกให้กับผู้ถูกเรียกได้ โดย network server จะทำการติดต่อกับ location server เพื่อทำการกำหนดเอนทิตีต่อไปที่จะรับแอสเสก location server จะทำหน้าที่ในการหาตำแหน่งปัจจุบันของผู้ถูกเรียกโดยการกำหนดเอนทิตีที่จะรับแอสเสกต่อไป แล้วส่งแอสเสกของเอนทิตีให้กับ network server ข้อมูลของ location server จะได้รับจาก registrar ซึ่งทำหน้าที่ในการรับข้อมูลเกี่ยวกับตำแหน่งของผู้ใช้ แล้วส่งข้อมูลนี้ให้กับ location server ในการให้ข้อมูลของผู้ใช้กับ registrar จะทำได้โดยการใช้แอสเสก REGISTER เพื่อบอกตำแหน่งที่อยู่ของผู้ใช้ โดยทั่วไปแล้ว registrar จะถูกรวมเข้ากับ network server

2.2.2.3 ชื่อและแอสเสก (Addressing & Naming) ในระบบ SIP การส่งแอสเสกระหว่างเอนทิตีจะต้องระบุ SIP URL เพื่อใช้อ้างอิงถึงผู้ใช้ SIP URL จะประกอบด้วย SIP Address รูปแบบของแอสเสกจะอยู่ในรูปของ name@domain โดยอาจจะเป็น user@domain user@address phone-number@gateway และ user@host แอสเสกนี้จะถูกใช้อ้างอิงถึงผู้ใช้ทั้งผู้เรียกและผู้ถูกเรียก ในการส่งแอสเสก ตัวอย่างของ SIP URL เช่น SIP ://j.doe@example.com โดยที่ URL นี้จะอยู่ในส่วนของแอสเสกของแอสเสก ในการส่งแอสเสกไปยัง SIP URL ที่ระบุไว้จะต้องมีการแปลง SIP

แอดเดรสให้อยู่ในรูปของ user@host โดยอาจจะผ่านการแปลงมากกว่าหนึ่งครั้งจนกระทั่งได้ตำแหน่งที่อยู่ของผู้ใช้ ในการแปลงแอดเดรสอาจจะใช้ DNS (Domain Name Server) หรือ LDAP (Lightweight Directory Access Protocol)

2.2.2.4 Locating Server ในการส่งแอดเดรสจะใช้ SIP URL อ้างอิงถึงในการส่ง โดยจะต้องมีการแปลงส่วนโดเมนของ SIP แอดเดรสไปเป็นหมายเลขไอพี ซึ่งเป็นแอดเดรสของ SIP Server ที่สามารถค้นหาตำแหน่งของผู้ใช้ต่อไปได้ การแปลง SIP แอดเดรสอาจทำโดย UAC หรือ UAC จะส่งแอดเดรสให้กับเซิร์ฟเวอร์ที่กำหนดซึ่งเซิร์ฟเวอร์จะเป็นผู้ที่ทำหน้าที่ในการแปลง SIP แอดเดรสแทน ในการแปลง SIP แอดเดรสสามารถใช้ DNS เข้ามาช่วยได้

2.2.2.5 Locate User จากข้างต้น เมื่อได้ตำแหน่งของเซิร์ฟเวอร์ที่ส่งข้อมูลมาให้กับผู้ถูกเรียกแล้วต่อไปจะเป็นการหาตำแหน่งของผู้ถูกเรียก เมื่อ SIP Server ได้รับแอดเดรสร้องขอแล้ว เซิร์ฟเวอร์จะต้องการค้นหาผู้ใช้ที่อ้างอิงถึงใน SIP แอดเดรส โดยการร้องขอข้อมูลไปยัง Location server ซึ่งจะตอบกลับด้วยรายการตำแหน่งที่เป็นไปได้ของผู้ถูกเรียก เมื่อ SIP server ได้ข้อมูลเกี่ยวกับตำแหน่งของผู้ถูกเรียกแล้ว ถ้าเป็น proxy server จะทำการส่งแอดเดสร้องขอต่อไปยังตำแหน่งต่างๆ ตามรายการที่ได้รับการ location server ไว้ โดยอาจจะส่งแบบ sequential หรือ parallel ส่วนถ้าเป็น redirect server จะส่งรายการตำแหน่งของผู้ถูกเรียกไปให้ผู้เรียกผ่าน โดยการใช้เฮดเดอร์ contact เพื่อให้ผู้เรียกส่งแอดเดสร้องขอไปเอง สำหรับตำแหน่งของผู้ใช้จะต้องทำการลงทะเบียนกับ registrar โดยใช้เฮดเดอร์ REGISTER รวมทั้งยังอาจจะอัปเดต script ของผู้ใช้งานเองเพื่อเก็บไว้ที่เซิร์ฟเวอร์สำหรับจัดการกับการเรียกตามความต้องการของผู้ใช้

2.2.2.6 ความน่าเชื่อถือ (Reliability) ในระบบ SIP จะมีกลไกเรื่องความเชื่อถือได้ไม่ว่าจะใช้โปรโตคอล UDP หรือ TCP โดยการใช้เมชอด Ack ไคล์เอนท์จะส่งแอดเดสร้องขอใหม่ตามช่วงเวลาที่กำหนดจนกระทั่งได้รับแอดเดรสตอบจากเซิร์ฟเวอร์ ทางด้านเซิร์ฟเวอร์ก็จะส่งแอดเดรสตอบจนกระทั่งได้รับแอดเดรส Ack จากไคล์เอนท์จึงทำให้การร้องขอที่สมบูรณ์ต้องใช้ในการแลกเปลี่ยนแอดเดรส 3 แอดเดรส เซิร์ฟเวอร์อาจจะตอบสนองต่อ Ack ในการส่งแอดเดรสตอบสุดท้ายให้กับไคล์เอนท์ซึ่งอาจจะไม่จำเป็นต้องมีก็ได้ สำหรับการส่งมีเดียสตรีมเซิร์ฟเวอร์จะยอมจะมีการส่งเมื่อได้รับ Ack จากไคล์เอนท์เท่านั้นด้วยกลไกนี้จึงทำให้เกิดความน่าเชื่อถือได้ในการแลกเปลี่ยนแอดเดรสโดยไม่จำเป็นต้องอาศัยกลไกของโปรโตคอลในชั้นต่ำกว่า เช่น TCP

2.2.2.7 ความสามารถในการขยาย (Protocol extension) SIP สามารถรองรับคุณลักษณะใหม่ที่เพิ่มเติมขึ้นสำหรับเมชอด เฮดเดอร์ และ status code ดังนี้

1) เมชอด เซิร์ฟเวอร์จะส่งแอดเดรสแสดงความผิดพลาด (Error message) กลับมาให้ไคล์เอนท์ถ้าเมชอดที่ร้องขอมาเซิร์ฟเวอร์ไม่เข้าใจและจะบอกเมชอดที่เซิร์ฟเวอร์เข้าใจโดยใช้

เฮดเดอร์ Public และ Allow โคล์เอ็นท์อาจจะส่งเมสเสจร้องขอเพื่อขอทราบเมธอดที่เซิร์ฟเวอร์สนับสนุนโดยใช้ตัวเลือกที่เฮดเดอร์ (header option)

2) เฮดเดอร์ เมื่อเอนทิตีได้รับเฮดเดอร์ที่ไม่เข้าใจก็จะละทิ้งเฮดเดอร์นั้นในกรณีทีโคล์เอ็นท์จำเป็นต้องการใช้เฮดเดอร์บางเฮดเดอร์ โคล์เอ็นท์จะส่งเมสเสจเพื่อร้องขอเฮดเดอร์ที่จำเป็นต้องใช้ไปโดยระบุในเฮดเดอร์ Require หากมีเฮดเดอร์ที่เซิร์ฟเวอร์ไม่สามารถให้การสนับสนุนได้เซิร์ฟเวอร์จะตอบปฏิเสธกลับมา

3) Status code ได้แบ่งเป็นคลาสต่างๆ เช่นเดียวกับ Response code ของโปรโตคอล HTTP ซึ่งโคล์เอ็นท์ต้องเข้าใจในความหมายในแต่ละคลาสเพื่อที่จะได้ทราบผลของการร้องขอว่าสำเร็จหรือไม่ สำหรับ status code ในเมสเสจตอบจะมีข้อความต่อหลังซึ่งจะเป็นความหมายของ code ซึ่งสามารถอ่านเข้าใจได้ โดยถ้าโคล์เอ็นท์ไม่ใจในรายละเอียดของ code ทั้งหมด โคล์เอ็นท์จะตีความหมายเป็น XO0 เมื่อ X เป็นตัวเลขตัวแรกของ status code และนอกจากนี้อาจจะนำ PEP (Protocol extension protocol) มาปรับปรุงใช้งานกับ SIP ได้

ในกรณีมีการส่งเมสเสจผ่านเซิร์ฟเวอร์หลายตัวจะใช้เฮดเดอร์ Via เพื่อระบุเซิร์ฟเวอร์ที่เป็นทางผ่านของเมสเสจทั้งหมด สำหรับใช้ในการส่งเมสเสจตอบสนองจะมีการตอบตกลงเกี่ยวกับพารามิเตอร์ของเซสชันด้วย ซึ่งรายละเอียดจะอยู่ในส่วนของ message body เช่นในกรณีของการสื่อสารโดยใช้เสียง พารามิเตอร์จะเป็น IP Address พอร์ตสำหรับ RTP และการเข้า/ถอดรหัสเสียงหลังจากการสร้าง การเรียก เสร็จสมบูรณ์ ช่องสัญญาณสำหรับ RTP จะถูกสร้างขึ้นทำให้ทั้งสองฝ่ายสามารถสื่อสารกันได้รวมทั้งยังอาจจะเชิญผู้อื่นมาเข้าร่วมในเซสชันนี้ได้ ในกรณีที่ต้องการเปลี่ยนพารามิเตอร์ของเซสชันสามารถทำได้โดยส่งเมสเสจร้องขอใหม่อีกครั้งโดยใช้วิธีการ invite ซึ่งมี call-id เดิม ไปยังผู้ร่วมเซสชันพร้อมทั้งค่าพารามิเตอร์ของเซสชันใหม่ที่ต้องการใช้รายละเอียดในส่วนนี้จะอยู่ในส่วนของ message body ซึ่งโดยทั่วไปจะใช้โปรโตคอล SDP (Session Description Protocol) ในการอธิบายความหมาย

2.2.3 เปรียบเทียบ H.323 กับ SIP โปรโตคอล เนื่องจากการพัฒนา H.323 โปรโตคอลได้ถูกพัฒนาโดยมีพื้นฐานมาจาก ATM และ ISDN จึงทำให้ H.323 โปรโตคอลยังมีข้อจำกัดอยู่หลายประการในการให้บริการ VoIP โดยเฉพาะเมื่อเปรียบเทียบกับ SIP โปรโตคอล ซึ่งถูกพัฒนาโดยพื้นฐานของ Internet โดยตรง H.323 โปรโตคอลจึงมีความยืดหยุ่นต่ำ รองรับกายยาวตัวของโครงข่ายต่ำ และมีความซับซ้อนสูงเมื่อเทียบกับ SIP โปรโตคอล เช่น H.323 โปรโตคอลได้มีการกำหนด element ต่างๆ นับร้อยในขณะที่ SIP โปรโตคอลมีแค่ 37 headers โดยจะแสดงข้อแตกต่างในตารางที่ 2.1 (สำนักงานคณะกรรมการกิจการโทรคมนาคมแห่งชาติ, 2550)

ตารางที่ 2.1 ข้อเปรียบเทียบระหว่าง H.323 โพรโตคอลกับ SIP โพรโตคอล

หัวข้อ	H.323	SIP
ความซับซ้อน	สูง	ต่ำ
จำนวน Message set	มาก	น้อย
การ Debug	ต้องการอุปกรณ์เสริมที่มี ความซับซ้อน	ต้องการอุปกรณ์เสริมไม่ซับซ้อน
การรองรับการขยายตัวในอนาคต	ต่ำ	สูง
ความยืดหยุ่นเพื่อรองรับบริการอื่น	ต่ำ	สูง
รองรับ Telephone service	เชื่อถือได้	เชื่อถือได้
ความสิ้นเปลืองใน Processor/Memory usage	สูง	ต่ำ
ความแพร่หลาย	สูง แต่คาดว่าจะถูกแทนที่โดย SIP ในอนาคตอันใกล้	สูง และคาดว่าจะ เป็น โพรโตคอล หลัก แทนที่ H.323

ฉะนั้นจึงพอสรุปได้ว่า SIP จะเป็นโพรโตคอลที่จะใช้ทดแทน H.323 โพรโตคอลในอนาคตอันใกล้ทั้งในส่วนโครงข่าย Fixed และ Mobile

2.2.3 เรียลไทม์โพรโตคอล (Real Time Protocol : RTP) (Schlzinne H, 1996)

Real Time Protocol (RTP) ถูกออกแบบมาเพื่อใช้สนับสนุนการส่งข้อมูลแบบเวลาจริง (Real Time Traffic) ที่ต้องการส่งข้อมูลและรับข้อมูลในช่วงเวลาที่สั้นมากๆ ตัวอย่างการส่งข้อมูลแบบเวลาจริงเช่น การส่งข้อมูลเสียงของการสนทนาระหว่างคู่สนทนา การส่งภาพของภาพยนตร์จากเครื่องส่งไปยังผู้รับหรือผู้ชมปลายทาง

RTP เป็นโพรโตคอลที่ทำงานในระดับ Transport Layer โดยการทำงานร่วมกับ UDP กล่าวคือ ข้อมูลสัญญาณเสียงที่ถูกส่งมาจากระดับชั้นบนจะถูกใส่ข้อมูลส่วนหัวของ RTP ก่อนที่จะส่งต่อไปยัง UDP ซึ่งอยู่ในระดับ Transport Layer เช่นเดียวกับ RTP จากนั้นข้อมูลจะถูกส่งต่อไปยัง Network Layer (IP) ต่อไป RTP ถูกออกแบบมาให้มีส่วนของการกำกับเวลา (Time Stamp) ในส่วนหัวของ RTP เพื่อใช้เป็นส่วนข้อมูลในการเข้าจังหวะข้อมูลสำหรับผู้รับปลายทาง เพื่อให้มีความมั่นใจ ได้ว่าลำดับของข้อมูลที่ได้รับมาแล้วที่ถูกส่งต่อไปให้ชั้นที่สูงกว่ามีความถูกต้อง และถูกลำดับแน่นอน ดังนั้นรูปแบบการทำงานของ RTP จึงมี 2 หน้าหลักคือ

1) RTP ทำหน้าที่เป็น Translator แปลง Syntax ของข้อมูลที่ได้รับเข้ามา ให้เป็นข้อมูลส่งออกที่มี Syntax อีกแบบที่เหมาะสมกับเครื่องคอมพิวเตอร์ปลายทาง

2) RTP ทำหน้าที่เป็น Mixer รวมข้อมูลที่มาจากต้นทางหลายๆ แหล่ง ให้เป็นข้อมูลส่งออกที่รวมกันเป็นข้อมูลสายเดียวกัน

2.3 ความรู้เบื้องต้นเกี่ยวกับ Voice Codec (วาริน เล้าสกุล, 2544)

VoIP จะมีการแปลงสัญญาณเสียงที่อยู่ในรูปแบบของ PCM ให้เป็นรูปแบบใหม่ เพื่อใช้ส่งไปในเครือข่ายไอน์นั้น จะต้องแปลงสัญญาณเสียงให้อยู่ในรูปแบบของดิจิทัล (CODEC) โดยรูปแบบของ CODEC ที่นิยมใช้กันในปัจจุบันคือ G.711, G.729 และ GSM

หน้าที่ของตัวแปลงสัญญาณเสียงที่ต้นทางคือ การเข้ารหัสข้อมูล PCM ของสัญญาณเสียงที่ได้รับมาให้อยู่ในรูปแบบของข้อมูลแบบใหม่ที่มีขนาดเล็กลง และที่ปลายทางอุปกรณ์ตัวแปลงสัญญาณเสียง ก็จะทำหน้าที่ในการถอดรหัสข้อมูลที่ได้รับมาให้กลับมาอยู่ในรูปแบบของข้อมูล PCM ของสัญญาณเสียงเหมือนเดิม จากนั้นจึงจะเป็นหน้าที่ของอุปกรณ์อื่นต่อไปที่จะทำหน้าที่แปลงข้อมูล PCM ให้เป็นสัญญาณเสียงแล้วส่งต่อไปยังผู้รับปลายทาง

รายละเอียดการเปรียบเทียบคุณสมบัติด้านต่างๆ ของการแปลงสัญญาณเสียงในมาตรฐานต่างๆ สามารถแสดงดังในตารางที่ 2.2

ตารางที่ 2.2 เปรียบเทียบคุณสมบัติของการแปลงสัญญาณเสียงของมาตรฐานต่างๆ

ชนิดของ CODEC	Data Rate	MOS
G.711	64 kbps	4.20
GSM	13.2 kbps	3.57
G.729	8 kbps	3.91

ที่มา: วิน โด ไอทีโพร (Windows IT Pro), 2542.

2.4 คุณลักษณะของเครือข่ายที่มีผลกระทบต่อคุณภาพเสียง (พรภัทร ภัทรจารี, 2548)

QoS (Quality of Service) เป็นการจัดลำดับความสำคัญของข้อมูลในระดับ Application โดยที่การทำงานของเทคโนโลยี Quality of Service (QoS) นั้น จะเป็นการจัดแบ่งประเภทของข้อมูล Application ออกเป็นหมวดหมู่ และมีการจัดลำดับความสำคัญของข้อมูล Application ในแต่ละหมวดหมู่นั้นๆ ซึ่งจะทำให้เราสามารถที่จะควบคุม Bandwidth ในระบบเครือข่ายของเราให้ใช้ประโยชน์ได้สูงสุดตาม Application ต่างๆ ที่เราต้องการ

QoS (Quality of Service) เข้ามาช่วยเป็นเกณฑ์มาตรฐานที่ใช้ในการวัดการรับส่งข้อมูล และการยอมรับในเรื่องคุณภาพของการให้บริการต่างๆ ซึ่งการยอมรับในเรื่องคุณภาพเป็นระดับที่ยากแก่การกำหนดว่า มาตรฐานคุณภาพในเครือข่ายสื่อสารเป็นอย่างไร ซึ่งผลกระทบที่มีต่อคุณภาพของเสียงบนเครือข่ายไอพีมีดังนี้

2.4.1 ความล่าช้า (Delay) ในการส่งสัญญาณข้อมูลเสียงแบบแพ็คเก็ตเกิดจากการรวบรวมสัญญาณที่สุ่มตัวอย่างจากสัญญาณเสียง เวลาในการเข้ารหัส/ถอดรหัส เวลาในการเข้าแพ็คเก็ต Jitter buffer delay และความล่าช้าของเน็ตเวิร์ค ปัญหาที่เกิดจากความล่าช้าของสัญญาณจากปลายทางหนึ่งถึงอีกปลายทางหนึ่งในเครือข่ายสัญญาณเสียงคือ เสียงสะท้อน และผู้พูดพูดซ้อนกัน จะเกิดเสียงสะท้อน(echo)ขึ้นเมื่อความล่าช้าของเสียงที่เดินทาง 1 รอบมีค่ามากกว่า 50 ms ระบบ VoIP จำเป็นจะต้องมีการควบคุมเสียงสะท้อนและโปรแกรมกำจัดเสียงสะท้อน การที่ผู้พูดพูดซ้อนกัน (talker overlap) จะเป็นปัญหาสำคัญเมื่อความล่าช้าของสัญญาณเสียงทิศทางเดียว (one way delay) มีค่ามากกว่า 150 ms

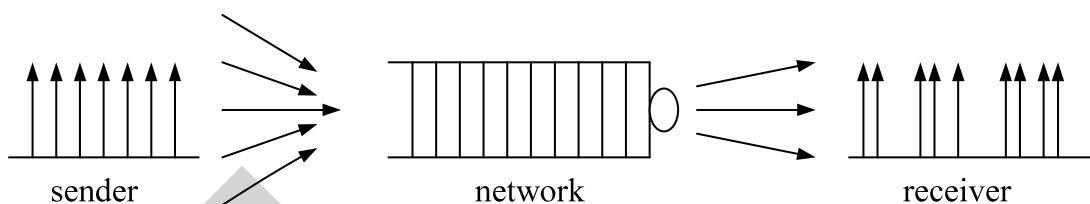
ค่าของดีเลย์ที่วัดจากต้นทางกับปลายทางที่เกิดขึ้นจริงจะประกอบด้วย 3 อย่างด้วยกัน คือ

1) Propagation delay : เป็นเวลาที่ข้อมูลเดินทางผ่านเครือข่ายจากต้นทางไปยังปลายทาง เช่น ดีเลย์ที่เกิดจากต้นทางอยู่กรุงเทพฯ ปลายทางอยู่มาเลเซีย ย่อมมีดีเลย์น้อยกว่า ที่ต้นทางอยู่กรุงเทพฯ แต่ปลายทางอยู่อเมริกา มีหน่วยเป็นมิลลิวินาที (ms)

2) Transport delay : เป็นดีเลย์ที่เกิดจากอัตราการส่งข้อมูล ค่านี้จะมีความสัมพันธ์กับแบนด์วิดท์ คือ ถ้าค่าแบนด์วิดท์กว้าง ดีเลย์ก็จะน้อย มีหน่วยเป็นมิลลิวินาที (ms)

3) Packetization delay เป็นเวลาที่ Codec แปลงสัญญาณอนาล็อกไปสู่การสร้างเฟรม และแปลงกลับเมื่อถึงปลายทาง เช่น Codec G.729 จะใช้เวลาในการแปลงเป็นแพ็คเก็ตที่สูงกว่า Codec G.711 เพราะจะต้องใช้เวลาในการบีบอัดข้อมูลที่มากกว่า

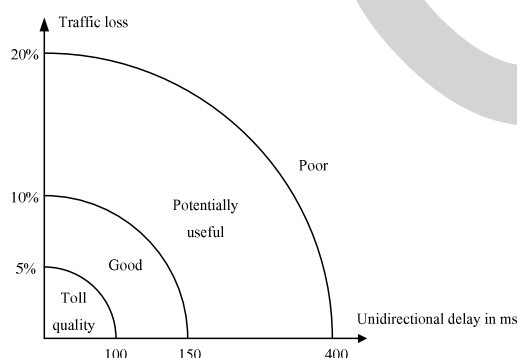
2.4.2 ความผันแปรเฉลี่ยของระยะเวลาดีเลย์ที่เกิดขึ้นกับแต่ละแพ็คเก็ต (Jitter) ในการส่งข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ต ข้อมูลเสียงแต่ละแพ็คเก็ตจะใช้เวลาในการเดินทางจากต้นทางไปยังปลายทางไม่เท่ากัน ซึ่งทำให้ข้อมูลที่ปลายทางได้รับนั้นมี Jitter เกิดขึ้น และถ้านำข้อมูลเสียงมาถอดรหัสทันทีจะทำให้เกิดความไม่ต่อเนื่องของเสียงได้ การลด Jitter สามารถทำได้โดยการนำข้อมูลเสียงที่ได้รับจากต้นทางมาเก็บไว้ในบัฟเฟอร์ก่อนช่วงเวลาหนึ่งแล้วค่อยนำข้อมูลเสียงนั้นไปถอดรหัสดังแสดงในรูป 2.7 โดยถ้าบัฟเฟอร์ข้อมูลเสียงมากก็จะลด Jitter ได้มาก แต่ข้อมูลเสียงจะเกิดการประวิงทางเวลามากเช่นกัน ดังนั้นการเลือกขนาดของบัฟเฟอร์จึงเป็นสิ่งสำคัญต่อคุณภาพเสียงที่ผู้รับจะได้ยิน



รูปที่ 2.7 รูปตัวอย่างการเกิด Jitter delay

ที่มา: เกษมศักดิ์, 2547.

2.4.3 การสูญหายของแพ็กเก็ต (Packet Loss) เครือข่ายไอพีไม่สามารถรับประกันได้ว่าแพ็กเก็ตที่ส่งไปจะครบและถูกต้องตามลำดับแพ็กเก็ตส่วนใหญ่จะสูญหายในช่วงที่มีการใช้งานมาก (peak load) ซึ่งเกิดจากความจุไม่เพียงพอและการถ่ายทอดเสียงต้องการความต่อเนื่องของเวลา วิธีการส่งข้อมูลซ้ำใหม่บนพื้นฐานของ TCP เดิมจึงไม่เหมาะสมกับสัญญาณเสียงต้องใช้วิธีแทรกคำพูดโดยขยี้เล่นแพ็กเก็ตก่อนหน้าและส่งซ้ำเดิมซ้ำกันหลายๆ ครั้ง เพื่อชดเชยแพ็กเก็ตที่สูญหาย โดยทั่วไประบบสามารถทดแทนแพ็กเก็ตที่สูญหายได้ไม่เกิน 10% เพื่อให้เครื่องปลายทางสามารถถอดสัญญาณได้อย่างถูกต้องหากมีความล่าช้าหรือแพ็กเก็ตสูญหายมากจะทำให้การสนทนาไม่ต่อเนื่องอันจะทำให้เกิดเสียงขาดๆ หายๆ ไป ทำให้ผู้ใช้รู้สึกได้ จึงต้องมีการกำหนดคุณภาพการให้บริการของการส่งสัญญาณเสียงดังรูปที่ 2.8 (อรศรี ศรีระยา, 2545) แสดงให้เห็นถึงคุณภาพของสัญญาณเสียงจัดอยู่ในระดับดีมาก ถ้ามีปริมาณทราฟฟิกสูญหายไม่เกิน 5% และความล่าช้าของสัญญาณไม่เกิน 100 มิลลิวินาที และถ้าปริมาณทราฟฟิกสูญหายไม่เกิน 10% ความล่าช้าของสัญญาณไม่เกิน 150 มิลลิวินาทีที่สามารถรองรับการสนทนาได้ดี เป็นต้น



รูปที่ 2.8 ระดับคุณภาพของสัญญาณเสียงเปรียบเทียบในเชิงปริมาณ ทราฟฟิกที่สูญหายกับความล่าช้าของสัญญาณในการส่งผ่าน โครงข่ายไอพี

การรักษาระดับของคุณภาพเสียงในฝั่งที่กระทำได้โดยใช้เทคนิคการบีบอัดสัญญาณ การห้ามรับเสียง Voice Activity Detection (VAD) หรือ Silence Suppression ซึ่งเป็นการตรวจหาช่องว่างในเสียงพูดและระงับการส่งสัญญาณในจังหวะเงียบนี้ และสามารถใส่ความสะดวกในการให้บริการบนชั้นทรานสปอร์ตเลเยอร์ (QoS-enabled transport network)

2.4.4 การเลือก CODEC จะมีผลต่อคุณภาพของเสียงเช่นกันเพราะว่า CODEC จะใช้แปลงสัญญาณเสียงที่เป็นอนาล็อกไปเป็นแบบดิจิทัล โดย CODEC G.711 จะให้คุณภาพเสียงที่ดีที่สุดเพราะว่าไม่มีการบีบอัด แต่มีดีเลย์เล็กน้อยและมีความไวกับข้อมูลที่จะสูญหายน้อยกว่า CODEC อื่น ส่วน CODEC G.729 จะกินแบนด์วิดท์น้อยกว่าเพราะมีการบีบอัดข้อมูลซึ่งนั่นก็ถือเป็นสิ่งที่ดีเพราะว่าจะสามารถใช้งานพร้อมๆ กันได้หลายๆ concurrent แต่การบีบอัดทำให้ความชัดเจนของเสียงลดลง ดีเลย์เพิ่มขึ้นและทำให้เสียงอาจขาดหาย

2.5 วิธีการวัดคุณภาพเสียงจากเครือข่ายไอพี (สมาคมวิศวกรรมศาสตร์ไทย, 2551, 18 สิงหาคม)

คุณภาพเสียงในที่นี้ก็คือคุณภาพของข้อมูลเสียงบนเครือข่ายไอพี ดังนั้นคุณภาพเสียงที่ดีก็คือเสียงที่ออกมาจะต้องใกล้เคียง หรือเหมือนกับต้นฉบับให้มากที่สุด ซึ่งการวัดคุณภาพเสียงผ่านเครือข่ายมีอยู่ 2 แบบด้วยกันคือ

2.5.1 แบบ Active Testing เป็นวิธีการแบบที่ใช้โดยการใช้เปรียบเทียบกับต้นฉบับ โดยการนำเอาออดิโอไฟล์ต้นฉบับมาแบ่งเป็นบล็อกเล็กๆ ที่เหลื่อมกันอยู่จากนั้นก็ทำการคำนวณค่า Fourier Transform Coefficient ของแต่ละบล็อกเก็บไว้ และเมื่อเสียงนั้นผ่านเครือข่ายแล้วก็ทำวิธีเดียวกันเพื่อให้ได้ชุดของค่า Fourier Transform Coefficient มาเพื่อเปรียบเทียบและทำการให้คะแนน ถึงแม้ว่าคะแนนที่ได้จะไม่ตรงกับคะแนน MOS เสียงที่เดียวแต่ก็มีความน่าเชื่อถือโดยจะมีค่าของ Correlation สูงถึง 0.95 ที่เดียว (ค่าสูงสุดเท่ากับ 1) จึงนับว่าเป็นวิธีการที่น่าเชื่อถือมากที่สุด อย่างไรก็ตามวิธีนี้ยังมีข้อเสียคือ เนื่องจากการเปรียบเทียบกับไฟล์ออดิโออ้างอิง จึงจำเป็นที่จะต้องมีการสร้างไฟล์เสียงอ้างอิงนั้นแล้วส่งเข้าไปในเครือข่าย ซึ่งจะเป็นการเพิ่มโหลดหรือทราฟฟิกให้กับเครือข่ายจะต้องมีกาวัดที่ต้นทางและปลายทาง ซึ่งส่วนใหญ่จะอยู่คนละที่ จึงทำให้ลำบากต่อการติดตั้งอุปกรณ์วัดและทำการวัดจริงๆ และยากที่จะทำการวัดนอกเครือข่ายของตนเองได้ จึงทำให้ขอบเขตของการวัดค่อนข้างจำกัด จึงทำให้ไม่เป็นที่นิยมมากนัก ยกเว้นกรณีทดสอบในห้องทดลอง

2.5.2 แบบ Passive Monitoring วิธีการนี้จะตรงข้ามกับแบบแรกคือจะไม่มีการเปรียบเทียบแต่อย่างใด โดยจะเริ่มจากการนำเอาเสียงที่ผ่านเครือข่ายมาแล้ว ผ่านกระบวนการที่เรียกว่า Pre-Processing ซึ่งได้แก่การทำ Filtering ,ปรับระดับเสียงและแยกเสียง Voice กับ Non-Voice ออกจากกัน (Voice Activity Detection) จากนั้นก็นำมาประมวลผลเพื่อแยกเอาค่าพารามิเตอร์ของเสียงและ

ค่าความเพี้ยนต่างๆ ออกมาเช่น Unnatural Speech เป็นเสียงประหลาดที่ไม่ใช่เสียงที่พบเห็นโดยทั่วไปเช่น เสียงป๊อป, Noise เป็นเสียงรบกวนต่างๆรวมทั้ง Background Noise, Interruptions หรือ Mute ซึ่งเป็นเสียงที่หายไปหรือเสียงเงียบไป เป็นต้น ซึ่งเมื่อได้ค่าพารามิเตอร์ของเสียงและค่าความเพี้ยนต่างๆ แล้ว ก็จะนำมาสร้างเป็นค่า MOS ที่มีค่าตั้งแต่ 1 ถึง 5 ซึ่งจะมีค่า Correlation เท่ากับ 0.89 – 0.9 ซึ่งวิธีนี้อาจจะไม่ดีเท่ากับวิธีก่อนหน้านี้ แต่ก็นับว่าใช้ได้ทีเดียวเพราะเหมาะสมกับการวัดจากเครือข่ายที่ใช้งานอยู่จริงมากกว่า ซึ่ง MOS เป็นวิธีที่ยอมรับกันอย่างแพร่หลายซึ่งมีต้นกำเนิดจากเครือข่ายโทรศัพท์ ตามวิธีการอย่างเป็นทางการที่กำหนดโดย ITU ซึ่งจะเป็นการประเมินโดยการพิจารณาเชิงคุณภาพ (Subjective) ด้วยการให้คะแนนจากผู้ทดสอบซึ่งแน่นอนว่าต้องเอาคนที่มีความรู้ในการฟังพอสมควร และความน่าเชื่อถือจะมากหรือน้อยขึ้นอยู่กับจำนวนคนที่นำมาทดสอบ และได้มีการนำเอาวิธีการในการวางแผนเครือข่าย หรือ E-Model มาปรับแต่งในการวัดคุณภาพเสียง และมีต้นทุนที่ต่ำกว่าโดยจะเกี่ยวข้องกับ Transmission Rating Factor (R-Factor) คือระดับคุณภาพการส่งสัญญาณเสียงโดยรวมซึ่งพิจารณาจากแวลลุ่มต่างๆ จากผู้พูดต้นทางไปยังผู้ฟังปลายทาง (Mouth to Ear) ซึ่งกำหนดโดยองค์กรสหภาพโทรคมนาคมระหว่างประเทศ (ITU) โดยค่า R-Factor จะอยู่ระหว่าง 0 ถึง 100 ซึ่ง 100 เป็นค่าดีเยี่ยมและ 0 เป็นค่าที่แย่มาก สูตรในการคำนวณ R-Factor (Chris Bajorek, 2003) มีดังนี้

$$R \text{ Factor} = R_0 - I_s - I_d - I_e + A \quad (1)$$

- โดย R_0 : อัตราส่วนของสัญญาณต่อเสียงรบกวน (Signal to noise ratio)
 I_s : การสูญเสียโดยรวมทั้งหมด (A combination of all impairments simultaneously)
 I_d : การสูญเสียที่เกิดจากดีเลย์ (Impairment caused by delay)
 I_e : การสูญเสียที่เกิดจากอุปกรณ์ (The packet-loss dependent Effective Equipment Impairment factor)
 A : ปัจจัยอื่น (Advantage factor) ซึ่งกำหนดไว้ดังนี้
- | | |
|--|--------|
| ใช้งานแบบสาย Conventional (wirebound) | A = 0 |
| ใช้งานเซลลูล่าภายในอาคาร (Mobility by cellular networks in a building) | A = 5 |
| ใช้งานเซลลูล่าภายในรถยนต์ (Mobility in a geographic area or moving in a vehicle) | A = 10 |

ใช้งานในตำแหน่งที่ยากต่อการเข้าถึง (Access to hard-to-reach locations)

$$A = 20$$

ค่า MOS Score และ R- Factor สามารถสรุปได้ดังนี้

$$MOS = 1 + 0.035R + 7 \cdot 10^{-6} R(R - 60)(100 - R) \quad (2)$$

แต่ในความเป็นจริงแล้วในสมการที่ (1) จะไม่สามารถวัดค่าเหล่านั้นจากเครือข่ายจริง นั่นคือค่าอัตราส่วนของสัญญาณต่อเสียงรบกวน (R_0) การสูญเสียโดยรวมทั้งหมด (I_0) การสูญเสียที่เกิดจากดีเลย์ (I_1) การสูญเสียที่เกิดจากอุปกรณ์ (I_2) และการสูญเสียจากปัจจัยอื่นๆ (A) ผู้วิจัยจึงได้ศึกษาหาวิธีการคำนวณใหม่ เพราะจากการทำการวัดจากเครือข่ายจริง จะสามารถวัดได้เฉพาะค่า Delay, Loss และ Jitter จากเครือข่ายเท่านั้น โดยการศึกษาค้นคว้าซึ่งใช้ในเครื่องมือวัดที่มีชื่อว่า PingPlotter Pro ซึ่งผู้ใช้สามารถเปลี่ยนแปลงค่าคงที่ตามสมการที่ (4), (5) ได้ตามความเหมาะสม ซึ่งวิธีการคำนวณจะใช้สมการดังนี้

$$\text{Effective Latency} = (\text{Average Latency} + \text{Jitter} * 2 + 10) \quad (3)$$

จากสมการที่ (3) ถ้าได้ค่าของ Effective Latency มีค่าน้อยกว่า 160 จะได้

$$R_1 = 93.2 - (\text{Effective Latency} / 40) \quad (4)$$

แต่ถ้าค่าที่ได้จากสมการที่ (3) มากกว่า 160 ก็จะได้ค่า R_1 ว่า

$$R_1 = 93.2 - (\text{Effective Latency} - 120) / 10 \quad (5)$$

แล้วนำค่า R_1 ซึ่งเกิดจากผลกระทบของดีเลย์กับจิสเตอร์ นำมาหาค่า R_0 ซึ่งเป็นผลกระทบทั้งหมดที่เกิดขึ้นรวมทั้งแพ็กเก็ตที่สูญหายด้วย

$$R_0 = R_1 - (\text{Packet Loss} * 2.5) \quad (6)$$

จากสมการที่ (6) นั้นก็นำมาหาค่า MOS ได้โดยใช้สมการ ดังนี้

$$MOS = 1 + (0.035) * R_0 + (.000007) * R_0 * (R_0 - 60) * (100 - R_0) \quad (7)$$

โดยค่า MOS Score จะอยู่ระหว่าง 1 ถึง 5 โดย 1 เป็นค่าที่มีคุณภาพเสียงแย่สุดแต่ 5 เป็นค่าที่มีคุณภาพเสียงดีสุด แต่ในความเป็นจริงค่าของ R- Factor สูงสุดจะไม่เกิน 93.2 และค่าของ MOS Score สูงสุดจะไม่เกิน 4.41 โดยรูปที่ 2.9 (กิติ ภัคคีวัฒนะกุล และ จำลอง ทรูอุตสาหะ, 2542) จะแสดงให้เห็นความสัมพันธ์ระหว่าง R- Factor กับ MOS Score

R	พอใจมากที่สุด	MOS
100	พอใจมาก	4.3
90	พอใจ	
80	ส่วนหนึ่งไม่พอใจ	4.0
70	ไม่พอใจจำนวนมาก	3.6
60	เกือบทั้งหมดไม่พอใจ	3.1
50	ไม่แนะนำให้ใช้	2.6
0		1.0

รูปที่ 2.9 แสดงให้เห็นความสัมพันธ์ระหว่าง R- Factor กับ MOS Score

2.6 ประเภทการใช้ VoIP ตาม OSI Layer

ในเครือข่าย VoIP ใช้ User Datagram Protocol (UDP) ร่วมกับ Real Time Protocol (RTP) ในการขนส่งข้อมูลจากต้นทางถึงปลายทางโดยชั้นที่ดูแลตาม OSI Layer ในชั้นที่ 4 คือ ชั้นทรานสปอร์ตเลเยอร์ โดยจะแสดงในตารางที่ 2.3 (อรศรี ศรีระษา, 2545)

ตารางที่ 2.3 ชั้นของ OSI Layer กับ โปรโตคอลที่เกี่ยวข้องกับ VoIP

OSI Layer	โปรโตคอล IP	VoIP Stack
Layer 7 Application	Application	Call Center
Layer 6 Presentation		G.723.1, G.711, G.729
Layer 5 Session		H.323, SIP, MGCP
Layer 4 Transport	Transport	RTP/RTCP, UDP

ตารางที่ 2.3 ชั้นของ OSI Layer กับโปรโตคอลที่เกี่ยวข้องกับ VoIP (ต่อ)

OSI Layer	โปรโตคอล IP	VoIP Stack
Layer 3 Network	Network	IP
Layer 2 Data Link	Data Link	Ethernet, FR, ATM, PPP
Layer 1 Physical	Physical	Copper, Fiber

โดย UDP มีขนาดของแพ็กเก็ตเล็กและไม่มีการกลไกที่ช่วยควบคุมการส่งข้อมูลทำให้สามารถส่งข้อมูลได้เร็วเหมาะสำหรับกราฟิกประเภทสัญญาณเสียงที่ยอมให้เกิดความล่าช้าของข้อมูลการส่งได้น้อย แต่ยอมให้เกิดการสูญหายของข้อมูลได้ระดับหนึ่ง โดย UDP จะมีเฮดเดอร์ขนาด 8 ไบต์

RTP ถูกออกแบบมาเพื่อใช้ในการส่งข้อมูลกราฟิกประเภทเรียลไทม์ที่ต้องการส่งข้อมูลในช่วงเวลาที่สั้นมากๆ โดยจะทำหน้าที่รวมข้อมูลที่เข้ามาจากหลายๆ แหล่งให้ออกเป็นข้อมูลที่ไหลรวมกันเป็นข้อมูลเดียวกันเหมาะสำหรับใช้กับการส่งข้อมูลประเภทเสียง เนื่องจากช่วยให้อุณหภูมิของสัญญาณเสียงดีขึ้น โดย RTP มีเฮดเดอร์ขนาด 12 ไบต์

IP เป็นโปรโตคอลในชั้นที่ 3 สำหรับทำหน้าที่ในการหาเส้นทางที่ใช้ในการส่งข้อมูลมีขนาดเฮดเดอร์ 20 ไบต์ โดยรูปแบบของแพ็กเก็ตจะแสดงในรูปที่ 2.10

IP Hdr (20 Bytes)	UDP Hdr (8 Bytes)	RTP Hdr (12 Bytes)	Voice sample
----------------------	----------------------	-----------------------	--------------

รูปที่ 2.10 รูปแบบของแพ็กเก็ต VoIP

โดยสัญญาณเสียงเมื่อแปลงให้อยู่ในรูปของสัญญาณข้อมูลซึ่งอยู่ในรูปของแพ็กเก็ตไอพี ขนาดของแพ็กเก็ตที่แตกต่างกันจะทำให้เกิดความล่าช้าในการส่งข้อมูลที่แตกต่าง ฉะนั้นการเลือกมาตรฐานในการเข้ารหัสก็เป็นสิ่งสำคัญเพื่อจะได้กำหนดขนาดของแบนด์วิดท์ที่ใช้ได้

การเชื่อมต่อแบบ Ethernet เป็นเทคโนโลยีสำหรับเครือข่ายแบบแลน (LAN) ที่ได้รับความนิยมสูงสุดในปัจจุบัน คิดค้นโดยบริษัท Xerox ตามมาตรฐาน IEEE 802.3 การเชื่อมต่อเครือข่ายแบบ Ethernet สามารถใช้สายเชื่อมต่อได้ทั้งแบบ Co-Axial และ UTP (Unshielded Twisted Pair) โดยสายสัญญาณที่ได้รับความนิยม คือ UTP 10Base-T ซึ่งสามารถส่งข้อมูลได้เร็วถึง 10 Mbps โดยถ้า

ใช้ CODEC G.711 แบบตัววิตซ์ 64 Kbps ก็จะมีรูปแบบของแพ็กเก็ตดังรูป 2.11 ถ้าจะคำนวณแบนด์วิธทั้งหมดที่ใช้ใน 1 แพ็กเก็ตก็จะได้ว่า

$$220 \text{ bytes(headers+payload)} / 160 \text{ bytes(payload only)} * 64,000 \text{ bit per second} = 88,000 \text{ bps}$$

Ethernet Hdr (14 Bytes)	IP Hdr (20 Bytes)	UDP Hdr (8 Bytes)	RTP Hdr (12 Bytes)	Voice Data (160 Bytes)	FEC/Filler (6 Bytes)
----------------------------	----------------------	----------------------	-----------------------	---------------------------	-------------------------

รูปที่ 2.11 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC G.711

แต่ถ้าเลือกการเข้ารหัสแบบ G.729 ซึ่งมีแบนด์วิธ 8 Kbps ก็จะมีรูปแบบของแพ็กเก็ตดังรูป 2.12 ถ้าจะคำนวณแบนด์วิธทั้งหมดที่ใช้ใน 1 แพ็กเก็ตก็จะได้ว่า

$$80 \text{ bytes(headers+payload)} / 20 \text{ bytes(payload only)} * 8,000 \text{ bit per second} = 32,000 \text{ bps}$$

Ethernet Hdr (14 Bytes)	IP Hdr (20 Bytes)	UDP Hdr (8 Bytes)	RTP Hdr (12 Bytes)	Voice Data (20 Bytes)	FEC/Filler (6 Bytes)
----------------------------	----------------------	----------------------	-----------------------	--------------------------	-------------------------

รูปที่ 2.12 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC G.729

และถ้าเลือกการเข้ารหัสแบบ GSM ซึ่งมีแบนด์วิธ 13.2 Kbps ก็จะมีรูปแบบของแพ็กเก็ตดังรูป 2.13 ถ้าจะคำนวณแบนด์วิธทั้งหมดที่ใช้ใน 1 แพ็กเก็ตก็จะได้ว่า

$$93 \text{ bytes(headers+payload)} / 33 \text{ bytes(payload only)} * 13,200 \text{ bit per second} = 37,200 \text{ bps}$$

Ethernet Hdr (14 Bytes)	IP Hdr (20 Bytes)	UDP Hdr (8 Bytes)	RTP Hdr (12 Bytes)	Voice Data (33 Bytes)	FEC/Filler (6 Bytes)
----------------------------	----------------------	----------------------	-----------------------	--------------------------	-------------------------

รูปที่ 2.13 รูปแบบของแพ็กเก็ต VoIP ที่ใช้ CODEC GSM

จากข้อมูลทั้งหมดพอจะสรุปรวมเป็นตารางที่ 2.4 เพื่อให้เปรียบเทียบข้อมูลของแต่ละวิธีในการเข้ารหัส ซึ่งจากข้อมูลแบนด์วิดท์ที่ต้องการในการสื่อสารด้วยเสียงนั้นจะเห็นได้ว่าแบนด์วิดท์ที่ต้องการบนเน็ตเวิร์กอย่างเครือข่ายแลน นับว่าค่อนข้างน้อยถ้าเทียบกับแอปพลิเคชันที่มีอยู่ การที่โทรศัพท์หนึ่งสายจะใช้แบนด์วิดท์ 64 กิโลบิตต่อวินาที หรือคิดเป็น 0.0625 เปอร์เซ็นต์ ของแบนด์วิดท์ที่เครือข่ายแลนแบบ Full Duplex 100 เมกะบิตต่อวินาทีที่มีเท่านั้น เมื่อคำนวณอย่างละเอียดแล้ว บนเครือข่ายอีเทอร์เน็ตความเร็ว 100 เมกะบิตต่อวินาทีนั้น โทรศัพท์หนึ่งสายจะใช้แบนด์วิดท์ประมาณ 88 กิโลบิตต่อวินาที (64 กิโลบิตต่อวินาที+ไอพีเฮดเดอร์+อีเทอร์เน็ตเฮดเดอร์) ถ้าเป็นการสื่อสารทางเดียวจะรองรับโทรศัพท์ได้ทั้งหมด 1,160 สายพร้อมๆ กันบนเครือข่ายแบบ Full Duplex แต่ถ้าใช้แบ็กโบนในระดับเป็นกิกะบิตอีเทอร์เน็ตด้วยแล้ว ก็จะรองรับโทรศัพท์ได้ถึง 11,600 สายเลยทีเดียว

ตารางที่ 2.4 แสดงแบนด์วิดท์สำหรับตัวเข้า/ถอดรหัสแต่ละชนิดบน Ethernet

ชนิดของการเข้ารหัส	แบนด์วิดท์ของเสียง (กิโลบิตต่อวินาที)	MOS	ขนาดแพ็กเก็ต (ไบต์)	เฮดเดอร์ L3-4 IP/UDP/RTP (ไบต์)	เฮดเดอร์ L2 (ไบต์)	แบนด์วิดท์ทั้งหมด (กิโลบิตต่อวินาที)
G.711	64	4.10	160	40	14	88
G.729	8	3.91	20	40	14	32
GSM	13.2	3.57	33	40	14	37.2

อย่างไรก็ตามถ้าแบนด์วิดท์เป็นปัจจัยเพียงอย่างเดียว เทคโนโลยีของระบบโทรศัพท์ผ่านไอพี คงจะได้รับความนิยมมาเป็นเวลานานแล้ว แต่ปัญหาที่แท้จริงนั้นอยู่เวลาในการตอบสนอง (Respons Time) เพราะในระบบเครือข่ายจะต้องให้บริการแอปพลิเคชันทางธุรกิจ ที่ต้องใช้แบนด์วิดท์จำนวนมาก ซึ่งก่อให้เกิดปัญหาความล่าช้า (Delay) ในการส่งข้อมูล โดยเฉพาะการสื่อสารด้วยเสียงที่ข้อมูลต้องได้รับการถ่ายโอนโดยใช้เวลาให้น้อยที่สุดได้อย่างเพียงพอ ดังนั้นเทคโนโลยีที่ถูกนำมาใช้เพื่อแก้ปัญหานี้ก็คือ QoS ที่มี Bandwidth Manager หรือ Complex Queuing Scheme คอยทำหน้าที่จัดลำดับความสำคัญให้กับข้อมูลบนเครือข่ายแลนและแวน

2.7 วิธีการปรับปรุงคุณภาพเสียงบนเครือข่ายไอพี (สมมติษัยและรังสิมา, 2550), (เอกพลชัยและนาโอพาร, 2548)

ระบบโทรศัพท์ผ่านไอพี (IP Telephone) เป็นแอปพลิเคชันที่ใช้กันอย่างแพร่หลายในแผนกไอทีขององค์กรต่างๆ โดยผู้ใช้อย่างมีความคาดหวังในระบบโทรศัพท์ผ่านไอพีไว้สูง ด้วยหนึ่งในสิ่งที่คนส่วนใหญ่ต้องการก็คืออย่างน้อยก็คือระบบโทรศัพท์ผ่านไอพี จะมีความเสถียรเทียบเท่ากับระบบ PBX เดิมที่เคยใช้มาก่อน แต่ระบบโทรศัพท์ผ่านไอพีนั้น ต้องพึ่งพาโครงสร้างพื้นฐานทางการสื่อสารบนด้านข้อมูล ดังนั้น ความเสถียรของระบบการสื่อสารข้อมูลจึงมักเป็นจุดอ่อนที่สุดในระบบโทรศัพท์ผ่านไอพีซึ่งต่างจาก PBX ตรงที่โทรศัพท์ผ่านไอพี มีจุดเชื่อมต่อจำนวนมาก อย่างเช่น เซิร์ฟเวอร์ โทรศัพท์ เกตเวย์ เราเตอร์ สวิตช์ และอื่นๆ อีกมากมาย หากเกิดปัญหาที่จุดใดจุดหนึ่ง ก็สามารถก่อให้เกิดปัญหาติดต่อกันขึ้นทั้งระบบเลยทีเดียว ความเสถียรนั้นจำเป็นต้องเริ่มต้นกันตั้งแต่การออกแบบ ซึ่งหมายความถึงมีแนวทางและระเบียบปฏิบัติที่เหมาะสม ไม่ใช่แค่เพียงมีฮาร์ดแวร์ จำนวนมากเท่านั้น ฮาร์ดแวร์ที่มีระบบสำรองทุกตัวในโลกนี้จะไม่สามารถแก้ไขปัญหาที่เกิดจากการออกแบบ โดยการส่งข้อมูลเสียงไปบนเครือข่ายไอพีนั้นจำเป็นต้องส่งข้อมูลแบบเวลาจริง แต่สำหรับ TCP/IP นั้น ไม่ได้ถูกออกแบบมาให้ทำเช่นนั้นได้ เราทำได้เพียงกำหนดนโยบายเพื่อให้ แพ็กเก็ต ของเสียง ผ่านเราเตอร์แต่ละตัวไปให้เร็วที่สุด โดยจะขึ้นอยู่กับโครงสร้างของเครือข่ายซึ่งสามารถทำได้ดังนี้

2.7.1 การแบ่งชั้นตามความสำคัญของข้อมูล (Traffic classification)

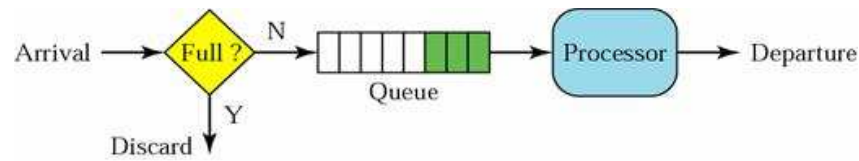
1) การทำ Virtual LAN (VLAN) ซึ่งจะเป็นการแยกชั้นเน็ตของเสียงและข้อมูลออกจากกันเพราะเมื่อเราแยกอุปกรณ์เสียง (โทรศัพท์, เกตเวย์) ออกจากอุปกรณ์ข้อมูล (เวิร์กสเตชัน, เซิร์ฟเวอร์) เราสามารถแยกการติดต่อสื่อสารของเสียงและข้อมูลออกจากกันได้ ซึ่งจะช่วยเพิ่มความเสถียรและความปลอดภัยได้อีกมาก โดยการจัดวางอุปกรณ์เสียงและอุปกรณ์ข้อมูลไว้บน VLAN คนละวง แล้วกำหนดไอพีแอดเดรสโดยแยกแอดเดรสเป็นคนละวงกัน นอกจากนั้นการแยก VLAN ยังทำให้สามารถกำหนดระดับการให้บริการ (Qos) และนโยบายความปลอดภัยให้แก่เสียงกับข้อมูลแตกต่างกันได้ เพราะ ไม่มีความจำเป็นที่โทรศัพท์ต้องติดต่อสื่อสารไปยังพีซี หรือในทางตรงกันข้าม เมื่อได้จัดการแยกการสื่อสารชั้นเน็ตของเสียงและข้อมูลจากกันแล้ว จะส่งผลให้ช่องโหว่ของระบบความปลอดภัย การกำหนดค่าผิดพลาด และข้อผิดพลาดในการปฏิบัติงานหมดไป แต่มีข้อยกเว้นคือ สะเตชันของฝ่ายบริหารยังคงสามารถเข้ามาดูระบบได้ โดยใช้หลักการเดียวกันคือ จัดวางเวิร์กสเตชันเหล่านั้นไว้ใน VLAN แยกวงกัน แล้วให้ VLAN วงนั้นเข้าถึงได้เฉพาะชั้นเน็ตของเสียงเท่านั้น โดยการกำหนดรายการควบคุมการเข้าถึง ACL (Access Control List) เพื่อแยกเสียงออกจากข้อมูล ซึ่งถ้าหากใช้ไอพีแอดเดรสจริง (Public IP) สำหรับข้อมูล ก็ควรให้ระบบ

โทรศัพท์ผ่านไอพี ใช้ไอพีแอดเดรสส่วนตัว (Private IP) เนื่องจากไม่มีความจำเป็นต้องให้โทรศัพท์ใช้ไอพีแอดเดรสที่สามารถสื่อสารเป็นไอพีไปโลกภายนอก เซิร์ฟเวอร์สำหรับระบบโทรศัพท์ควรจะใช้ VLAN แยกจากวงอื่นด้วย ซึ่งจะช่วยให้สามารถกั้นกรองข้อมูลที่ส่งไปยัง (และกลับจาก) เซิร์ฟเวอร์ เนื่องจากเซิร์ฟเวอร์สำหรับระบบโทรศัพท์เป็นหัวใจของระบบโทรศัพท์ผ่านไอพี เราต้องปกป้องไว้ไม่ให้เกิดเหตุอันไม่พึงประสงค์ โดยการกำหนด ACL ให้อนุญาตเฉพาะกราฟฟิกที่จำเป็นจริงๆ เท่านั้น (ปกติก็คือกราฟฟิกที่เกี่ยวกับการตั้งค่าและบริหารจัดการโทรศัพท์) ที่สามารถเข้าถึงเซิร์ฟเวอร์ได้ ถ้าจะให้ดียิ่งขึ้นไปอีก (หากมีงบประมาณพอ) ควรติดตั้งไฟร์วอลล์กั้นไว้ระหว่างเซิร์ฟเวอร์สำหรับระบบโทรศัพท์กับส่วนอื่นๆ ในเครือข่ายเพื่อสกัดกั้นกราฟฟิกที่ไม่พึงประสงค์

2) การกำหนดคุณภาพการให้บริการ (Quality of Service : QoS) ซึ่งอยู่ใน header ของ IP Protocol จะถูกกำหนดให้เป็น high เพื่อระบุให้ แพ็กเก็ต นั้นเป็น แพ็กเก็ต ที่มีความสำคัญสูง ยิ่งให้ความสำคัญสูง แพ็กเก็ต ยิ่งใกล้ถูกส่งออกไปใกล้เวลาจริงยิ่งขึ้น เพราะ QoS เป็นสิ่งที่จำเป็นอย่างยิ่งสำหรับโทรศัพท์ผ่านไอพีเพราะต้องการคุณภาพของการให้บริการที่คงที่นั่นคือความสามารถในการควบคุมความเร็วและแบนวิดท์ของเครือข่ายการรับส่งข้อมูลระหว่างต้นทางและปลายทางได้ เนื่องจากเสียงการสนทนาถ้าไม่ต่อเนื่องจะฟังไม่รู้เรื่อง ดังนั้นข้อมูลเสียงจะต้องถึงปลายทางตามกำหนดเวลา และให้รูปแบบที่ต่อเนื่องตลอดเวลา เพราะว่าเครือข่ายนั้นเปลี่ยนแปลงอยู่ตลอดเวลา ลิงค์ที่มีการใช้งานน้อยอาจจะกลายเป็นลิงค์ที่มีการใช้งานมากไปก็ได้ หากไม่มีการทำ QoS ซึ่งมักจะก่อให้เกิดปัญหาคุณภาพเสียง (ขาดๆ หายๆ) เนื่องจาก การใช้งานเครือข่ายไม่เท่ากันในแต่ละช่วงของวัน หากมีการติดตั้ง QoS ให้คอยพิจารณาแอปพลิเคชันอื่นๆ ที่ไวต่อความล่าช้าของ สัญญาณด้วย อย่างเช่น วิดีโอคอนเฟอร์เรนซ์ หรือสตรีมมิ่งมีเดีย ที่สำคัญควรวางแผนไว้เพื่อการเติบโตในอนาคต

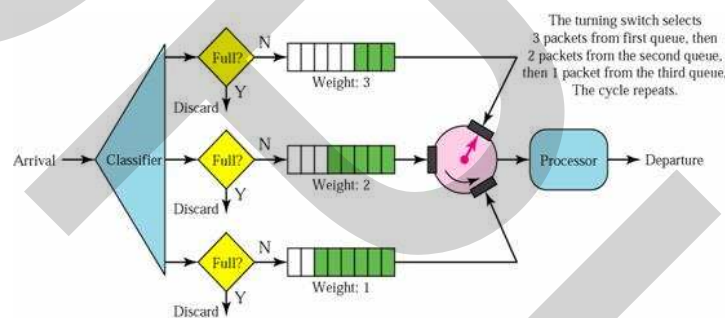
2.7.2 วิธีการจัดแถวคอยของ แพ็กเก็ต มีอยู่ด้วยกันหลายวิธีคือ

1) FIFO (First in First Out) ถือเป็นพื้นฐานที่สุดของเครือข่ายทุกชนิด โดยเป็นการเลือกแพ็กเก็ตที่มาถึงเราเตอร์เป็นตัวแรกให้ทำการส่งออกไปก่อน ซึ่ง FIFO นี้จะมีคิวเพียงคิวเดียว โดยไม่สนใจความสำคัญ (priority) และเนื่องจากพื้นที่ว่างของเราเตอร์แต่ละตัวมีขีด จำกัด เมื่อแพ็กเก็ตมาถึง แต่ไม่มีพื้นที่ว่างเพียงพอ เราเตอร์จะทำการละทิ้งแพ็กเก็ตนั้น โดยการละทิ้งนี้ จะไม่สนใจว่าแพ็กเก็ตมีความสำคัญมากน้อยเพียงใด จะมีลักษณะการทำงานตามรูปที่ 2.14 (กิต ภัคดี วัฒนะกุล และจำลอง ครูอุตสาหะ, 2542)



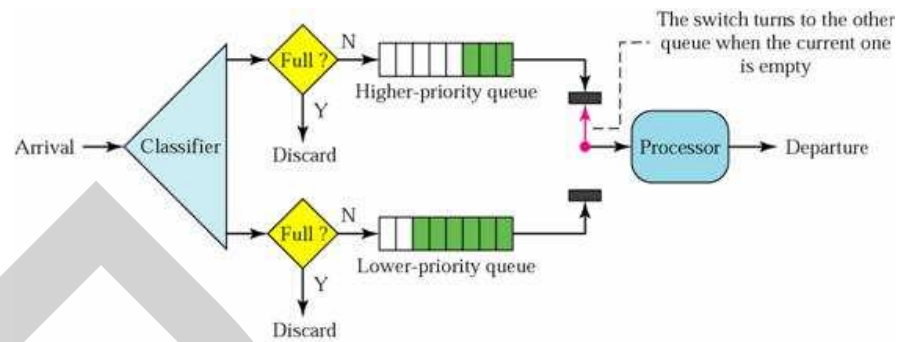
รูปที่ 2.14 การจัดแถวคอยแบบ First in First Out

2) WFQ (Weighted Fair Queuing) วิธีนี้จะกำหนดให้มีความเสมอภาคกันของแต่ละคิว เพื่อไม่ให้มีเซอร์วิสตัวใดตัวหนึ่งใช้ช่องสัญญาณมากเกินไป แต่ในส่วนของการให้น้ำหนักในคิวแต่ละคิวจะแตกต่างกัน สำหรับคิวที่มีความสำคัญมาก จะได้รับจำนวนบิตที่ส่งไปมากกว่าคิวที่มีความสำคัญน้อยลงมา แต่ยังคงมีการวนรอบให้เซอร์วิสกับทุกๆ คิวเช่นเดิม โดยจะมีลักษณะการทำงานตามรูปที่ 2.15 (กิติ ภัคดีวัฒน์กุล และจำลอง ทรูอุตสาหะ, 2542)



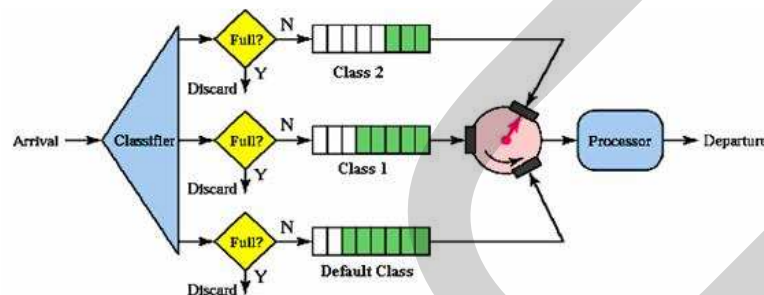
รูปที่ 2.15 การจัดแถวคอยแบบ Weighted Fair Queuing

3) PQ (Priority Queuing) เป็นรูปแบบหนึ่ง queuing ที่ปรับเปลี่ยนมาจาก FIFO คือ เราเตอร์จะสามารถเลือกแพ็คเก็ตได้จากคิวหลายคิว และ จะมีการกำหนดความสำคัญให้กับแต่ละคิว ซึ่งจะแตกต่างกันไป เราเตอร์จะส่งแพ็คเก็ตโดยเลือกจากคิวที่มีความสำคัญมากที่สุดเป็นอันดับแรก จากนั้นจะเลือกคิวที่มีความสำคัญรองลงไปตามลำดับ และในแต่ละคิวจะมีการจัดการกับแพ็คเก็ตในคิวนั้นแบบ FIFO จะมีลักษณะการทำงานตามรูปที่ 2.16 (กิติ ภัคดีวัฒน์กุล และจำลอง ทรูอุตสาหะ, 2542)



รูปที่ 2.16 การจัดแถวคอยแบบ Priority Queuing

4) CQ (Custom Queuing) หรืออีกอย่างเรียกว่า Class-Based Queue (CBQ) เป็นวิธีการจัดคิวที่มีการกำหนดความสำคัญเหมือนกับแบบ PQ แต่จะเลือกให้บริการแพ็กเก็ตในคิวต่างๆ แบบหมุนวน (Round Robin) ตามข้อกำหนดที่ตั้งไว้เพื่อแก้ปัญหาการรอโดยไม่มีกำหนดของแพ็กเก็ตที่มีความสำคัญระดับต่ำ ซึ่งผู้ดูแลระบบสามารถกำหนดค่าสำหรับการให้บริการแพ็กเก็ตในแต่ละคิวเพื่อประกันขนาดของช่องทางต่ำสุดที่แต่ละคิวจะได้รับ ดังนั้นในแต่ละรอบของคิวต่างๆ จะได้รับบริการอย่างน้อยที่สุดตามค่าที่ระบุไว้ โดยจะมีลักษณะการทำงานตามรูปที่ 2.17 (กิติ ภัคคีวัฒนะกุล และจำลอง คุรุอุตสาหะ, 2542)



รูปที่ 2.17 การจัดแถวคอยแบบ Custom Queuing

5) CB-WFQ (Class Based Weighted Fair Queuing) วิธีการนี้มีความคล้ายคลึงกับ WFQ แต่ต่างกันที่ ได้มีการเพิ่มคุณสมบัติของ class เข้าไป โดยให้ค่าของแบนด์วิดท์เป็นคุณสมบัติของแต่ละ class

2.8 Visual Basic (ixiacom, 1998-2011)

ปัจจุบันระบบปฏิบัติการ (Operating System) ในลักษณะของ Windows ได้เข้ามาแทนที่ในระบบปฏิบัติการในลักษณะเดิม ซึ่งส่วนใหญ่ที่นิยมใช้กันอยู่คือ MS-DOS เพราะรูปแบบของจอภาพที่ใช้ติดต่อระหว่างคอมพิวเตอร์กับผู้ใช้อยู่ในรูปของคำสั่งซึ่งอยู่ในรูปแบบของตัวอักษรเป็นแบบป้อนทีละบรรทัดหรือเรียกว่า Command Line ซึ่งผู้ใช้จะต้องเรียนรู้และจดจำรูปแบบของคำสั่งให้ถูกต้องและแม่นยำจึงจะใช้งานโปรแกรมนั้นได้เป็นอย่างดี ซึ่งต่างจาก Visual Basic ที่รูปแบบของคำสั่งจะอยู่ในรูปแบบของ Graphic User Interface (GUI) ที่ใช้รูปภาพแทนคำสั่งต่างๆ แทน ซึ่งแต่เดิมการพัฒนาโปรแกรมจะอยู่บน MS-DOS จึงต้องเปลี่ยนแปลงรูปแบบและแนวความคิด และหันมาพัฒนาโปรแกรมบนวินโดวส์แทน

ภาษา BASIC ถูกสร้างขึ้นมาในปี 1963 โดย John Keneney และ Thomas Kurtz ที่วิทยาลัย Dartmouth ในเบื้องต้นพวกเขามีจุดมุ่งหมายในการพัฒนาภาษา BASIC ขึ้นเพื่อใช้ในการสอนแนวเขียนโปรแกรม (Programming Concept) โดยเน้นให้รูปแบบของภาษานั้นง่ายต่อการเข้าใจและใช้งาน รวมทั้งทำงานในลักษณะ Interpreter ซึ่งแตกต่างจากภาษาคอมพิวเตอร์อื่นๆ ในยุคนั้นที่จะอาศัย Job Control Language (JCL) และขั้นตอนในการ Compile และ Link ผลก็คือภาษา BASIC ได้กลายมาเป็นที่นิยมกันอย่างกว้างขวาง โดยเฉพาะในหมู่ผู้ใช้คอมพิวเตอร์ส่วนบุคคล จึงอาจกล่าวได้ว่าภาษา BASIC ได้รับการพัฒนาควบคู่ไปกับการพัฒนาคอมพิวเตอร์ส่วนบุคคล ในปี 1970 Microsoft ได้เริ่มผลิตตัวแปรภาษา BASIC ใน ROM ซึ่งเรียกว่า ROM-Based BASIC ขึ้น เช่น ชิป Radio Sheek TRS-80 เป็นต้น ต่อมาได้พัฒนาเป็น GW-BASIC ซึ่งเป็น Interpreter ภาษาที่ใช้กับ MS-DOS และในปี 1982 Microsoft QuickBasic ได้รับการพัฒนาขึ้น โดยการเพิ่มความสามารถในการ Compile ให้เป็น Executed Program รวมทั้งทำให้ BASIC มีความเป็น “Structured Programming” มากขึ้นโดยการตัด Line Number ทิ้งไป เพื่อลบข้อกล่าวหาว่าเป็นภาษาคอมพิวเตอร์ที่มีโครงสร้างในลักษณะ Spaghetti Code (Logical Flow ของภาษาขาดโครงสร้าง) มาใช้รูปแบบของ Subprogram และ User Defined รวมทั้งการใช้ Structured Data Type และการพัฒนาการใช้งานด้านกราฟฟิกให้มีการใช้งานในระดับที่สูงขึ้น รวมทั้งมีการใช้เสียงประกอบได้เหมือนกับภาษาคอมพิวเตอร์อื่นๆ เช่น C หรือ Pascal

Visual Basic เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้พัฒนาโปรแกรมบน Windows เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแค่เลือก Control ที่เหมาะสมแล้ววาดลงบน Form ก็สามารถสร้างจอภาพที่ใช้สำหรับติดต่อกับผู้ใช้รวมถึงการใช้เทคนิคการเขียนโปรแกรมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับ Control ต่างๆ ที่สร้างตามเหตุการณ์ (Event) ต่างๆ ที่เกิดขึ้น เช่น การเลื่อนเมาท์

หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น ประกอบกับภาษาที่ใช้เขียนโปรแกรมเป็นภาษา BASIC ซึ่งเป็นภาษาคอมพิวเตอร์ที่ผู้ใช้นคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่คุ้นเคยจึงส่งผลทำให้การพัฒนาโปรแกรมบน Windows ด้วย Visual Basic มีขั้นตอนน้อยกระทำได้ง่ายและสะดวกต่อการใช้งานจึงทำให้ผู้ใช้สามารถเรียนรู้ได้ภายในเวลา 2-3 ชั่วโมง ก็สามารถพัฒนาโปรแกรม Windows ขึ้นเป็นโปรแกรมแรก

Visual Basic นี้เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมขึ้นใช้งาน ที่ใช้ได้ตั้งแต่ผู้ใช้ระดับต้นเพื่อใช้สร้างโปรแกรมง่ายๆ บน Windows หรือโปรแกรมเมอร์ระดับกลาง ที่เรียกใช้ฟังก์ชันการทำงานต่างๆ ของ Visual Basic ได้อย่างมีประสิทธิภาพตลอดจนโปรแกรมเมอร์ในระดับอาชีพที่จะพัฒนาในโปรแกรมระดับสูงโดยการใช้ Object Linking and Embedding (OLE) และ Application Programming Programming Interface (API) ของ Windows มาประกอบในการเขียนโปรแกรม

2.9 โปรแกรม iperf (Mitchkutzko Jdugan, 2008)

โปรแกรม iperf เป็น โปรแกรมฟรี (freeware) ใช้สำหรับตรวจสอบระบบ network เป็นสิ่งสำคัญอย่างยิ่งในการใช้งานระบบ VoIP เนื่องจากระบบ VoIP เป็นการทำงานแบบ Realtime ซึ่งเป็นการทำงานที่มีความอ่อนไหวเป็นอย่างมากกับระบบ Network ซึ่งหากมีปัญหาเกี่ยวกับระบบ Network จะมีผลกระทบโดยตรงต่อคุณภาพเสียง ซึ่งในการใช้งานจริงผู้ใช้งานบางท่านอาจคิดว่าในปัจจุบันก็สามารถใช้งาน internet, mail ได้คืออยู่แล้วทำไมต้องทำการตรวจสอบอีก จึงขออธิบายว่าในการใช้งานระบบเครือข่ายนั้น บาง Protocol เช่น Web (http) หรือ Mail (POP3) จะเป็นการทำงานโดยใช้ Protocol TCP เป็นหลัก โดยหากมีปัญหาเกี่ยวกับระบบ network เช่น delay หรือ package loss จะไม่ส่งผลกระทบต่อ Application นั้นมาก เนื่องจาก Application นั้นๆ สามารถจะรอหรือร้องขอ data ชุดนั้นๆใหม่ได้ ส่วน ระบบ VoIP ส่วนใหญ่ทำงานในลักษณะ Realtime และใช้ UDP เป็น Protocol หลักในการส่งเสียง ซึ่งจะไม่สามารถรอ data ใต้นาน และ ถ้า data lost ก็จะไม่สามารถส่ง data นั้นได้ใหม่

Iperf เป็นโปรแกรมที่ใช้ในการสร้างแพ็กเก็ตและทดสอบเครือข่ายเบื้องต้น โดยจะทำการวัดขนาดของการส่งข้อมูล (Bandwidth) บนเครือข่ายระหว่างจุดที่สนใจ ประกอบด้วยส่วนที่เป็นเซิร์ฟเวอร์และไคล์แอนต์ ในตัวเดียวกัน โดยสามารถเลือกได้ว่าจะใช้แบบใด หลักการทำงานคือทำการรันที่เครื่องเซิร์ฟเวอร์ก่อนแล้วจากนั้นจึงทำการรันตัวไคล์แอนต์เพื่อทำการส่งแพ็กเก็ตไปที่เครื่องเซิร์ฟเวอร์ แล้วจะคำนวณค่าต่างๆส่งกลับไปยังตัวไคล์แอนต์ โดยโปรแกรมสามารถกำหนดโปรโตคอลของการส่งข้อมูลได้หลายชนิด ได้แก่ TCP Protocol, UDP Protocol เป็นต้น

คุณสมบัติของ Iperf

- 1) ใช้วัดความจุของเครือข่าย (แบนด์วิดท์)
- 2) โคล์เอ็นต์สามารถสร้างสตรีมแบบ UDP ส่งเข้าไปยังเครือข่ายเพื่อทำการตรวจวัด
- 3) สามารถวัดความขาดหายของแพ็กเก็ต
- 4) สามารถวัดค่าเฉลี่ยจิสเตอร์

2.10 คำสั่ง Ping (Microsoft, 2552, 1 กรกฎาคม), (วัชรพงษ์. 2547)

คำสั่ง Ping เป็นคำสั่งที่ใช้ในการตรวจสอบการเชื่อมต่อกับเครือข่ายระหว่างคอมพิวเตอร์แต่ละเครื่องที่อยู่ในเครือข่าย โดยคำสั่ง Ping จะส่งข้อมูลที่เป็นแพ็กเกจ 4 ชุดๆละ 32 Byte ไปยังคอมพิวเตอร์ปลายทางที่ต้องการตรวจสอบ หากมีการตอบรับกลับมาจากคอมพิวเตอร์เป้าหมายก็แสดงว่าการเชื่อมต่อเครือข่ายยังเป็นปกติ แต่หากไม่มีการตอบรับกลับมาก็แสดงว่าคอมพิวเตอร์ปลายทางหรือเครือข่ายอยู่ในช่วงหนาแน่น ดังนั้นจะเห็นว่าคำสั่ง Ping มีประโยชน์อย่างมากในการตรวจสอบสถานการณ์เชื่อมต่อเครือข่ายเบื้องต้นได้เป็นอย่างดีโดยคำสั่ง ping จะใช้โปรโตคอล ที่มีชื่อว่า ICMP

2.10.1 รูปแบบการใช้คำสั่ง Ping

Ping [ไอพีแอดเดรส,ชื่อเครื่องคอมพิวเตอร์] เช่น ping 10.40.0.1 จะหมายถึงการตรวจสอบเครื่องที่มี IP Address 10.40.0.1 ว่ามีการทำงานติดต่อบริเวณเครือข่ายหรือไม่

2.10.2 การรายงานข้อผิดพลาดของคำสั่ง ping

เมื่อคำสั่ง ping ทำงานแล้วไม่ว่าระบบเครือข่ายจะสามารถเชื่อมต่อได้หรือไม่ คำสั่ง ping ก็รายงานผลออกมาโดยการแจ้งผลเมื่อไม่สามารถติดต่อกับเครือข่ายปลายทางได้ก็คือ

- 1) คำร้องหมดเวลา (Request time out)
- 2) หา Host ปลายทางไม่พบ (Host unknown)

ข้อผิดพลาด แต่ละอย่างที่โปรแกรม ping รายงานให้ทราบนั้นจะระบุรายละเอียดให้ด้วยว่าข้อผิดพลาดนั้นเกิดจากอะไร ซึ่งข้อผิดพลาดแต่ละอย่างมีความหมายดังนี้

คำร้องหมดเวลา นี้หมายความว่า เมื่อคำสั่ง Ping ส่งแพ็กเกจ ไปยังเครื่องปลายทางแล้วเครื่องปลายทางไม่ตอบกลับมาภายในเวลาที่กำหนด ปัญหาที่พบส่วนใหญ่จะมีดังนี้

- 1) เครื่องปลายทางไม่ได้เปิดอยู่
- 2) ระบบเครือข่ายมีปัญหาไม่สามารถติดต่อกับเครื่องปลายทางได้
- 3) ระบบเครือข่ายช้ามากไม่สามารถตอบรับการ ping ได้ภายในเวลาที่กำหนดที่กำหนด
- 4) เครื่องปลายทางทำงานหนักมากไม่สามารถรองรับข้อมูลการ ping ไม่ทัน

Host unknown ข้อผิดพลาด Host unknown นี้เกิดจากการที่เราสั่งให้ ping เป็นชื่อเครื่องคอมพิวเตอร์แล้ว เครื่องคอมพิวเตอร์ไม่สามารถเปลี่ยนชื่อเครื่องคอมพิวเตอร์ให้เป็นไอพีแอดเดรสได้ ping ได้ เพราะว่าการ ping นั้นจะต้อง ping ไปยังไอพีแอดเดรส

2.10.3 Options ของคำสั่ง ping

-t คือ Ping ไปยัง Host ตามที่ระบุเรื่อยๆ จนกว่าจะสั่งยกเลิกโดยกดเป็น Ctrl-C และหากต้องการยุติให้กดเป็น Ctrl-Break

-a คือ เปลี่ยนหมายเลข IP Address ของ Host เป็นชื่อแบบตัวอักษร

-n คือ count Ping แบบระบุจำนวน echo ที่จะส่ง

-l คือ size กำหนดขนาด buffer

-f คือ ตั้งค่าไม่ให้แยก flag ใน packet.

-i คือ TTL Ping แบบกำหนด Time To Live โดยกำหนดค่าตั้งแต่ 1-255

-v คือ TOS กำหนดประเภทของบริการ (Type of service)

-r คือ count Ping แบบให้มีการบันทึกเส้นทางและนับจำนวนครั้งในการ hops จนกว่าจะถึงปลายทาง

-s คือ count Ping แบบนับเวลาในการ hop แต่ละครั้ง

-j คือ host-list Loose source route along host-list.

-k คือ host-list Strict source route along host-list.

-w คือ timeout Ping แบบกำหนดเวลารอคอยการตอบรับ

2.11 การตรวจสอบความถูกต้อง และการวิจัยยอมรับของโปรแกรมที่พัฒนาขึ้น

การตรวจสอบและการยอมรับของโปรแกรมที่พัฒนาขึ้นมา จะใช้หลักสถิติ วิธีการแบบ Hypothesis (สมมติฐาน) ในการตรวจสอบความถูกต้องหรือไม่ ซึ่ง Hypothesis คือ คำตอบที่คาดการณ์ความสัมพันธ์ในเชิงเหตุผลระหว่างตัวแปรต่างๆ ที่มีผลต่อปัญหาที่ศึกษา เป็นการคาดคะเนคำตอบไว้ล่วงหน้า เพื่อทดสอบด้วยข้อเท็จจริง (empirical data) ว่ามีความถูกต้องหรือไม่ สมมติฐานมี 2 ประเภทคือ

1) สมมติฐานของการวิจัย (Research Hypothesis) เป็นการสมมติที่เขียนอยู่ในรูปของข้อความที่อธิบายความสัมพันธ์ของตัวแปรที่ศึกษา แบ่งออกเป็น 2 ประเภท คือ สมมติฐานแบบมีทิศทาง (Direction Hypothesis) และ สมมติฐานแบบไม่มีทิศทาง (Non-direction Hypothesis)

2) สมมติฐานในทางสถิติ (Statistical Hypothesis) เป็นสมมติฐานที่เขียนขึ้นในรูปโครงสร้างทางคณิตศาสตร์เพื่ออธิบายความสัมพันธ์ของตัวแปร ประกอบด้วยสมมติฐานที่เป็นกลางหรือ

สมมติฐานหลัก (Null Hypothesis : H_0) เช่น เท่ากับ มากกว่าเท่ากับ หรือน้อยกว่าเท่ากับ สมมติฐานไม่เป็นกลาง หรือสมมติฐานรองหรือสมมติฐานทางเลือก (Alternative Hypothesis : H_a หรือ H_1) เช่น มากกว่า น้อยกว่า หรือไม่เท่ากับ

ประเภทของความคลาดเคลื่อน

1. ความคลาดเคลื่อนประเภท I (Type I error) เป็นความคลาดเคลื่อนที่เกิดขึ้นจากการปฏิเสธสมมติฐานกลาง (H_0) เมื่อสมมติฐานกลางเป็นจริง และเรียกความผิดพลาดชนิดนี้ว่า “ระดับนัยสำคัญ” (Level of Significance)

2. ความคลาดเคลื่อนประเภท II (Type II error) เป็นความคลาดเคลื่อนที่เกิดขึ้นจากการยอมรับสมมติฐานกลาง (H_0) เมื่อสมมติฐานกลางไม่เป็นจริง ตามตารางที่ 2.5

ตารางที่ 2.5 ข้อเท็จจริงกับการตัดสินใจ

การตัดสินใจ	ข้อเท็จจริงของ H_0	
	H_0 เป็นจริง	H_0 ไม่เป็นจริง
ปฏิเสธ H_0	ความคลาดเคลื่อนประเภท I เขียนแทนด้วย α	การตัดสินใจถูกต้อง
ยอมรับ H_0	การตัดสินใจถูกต้อง	ความคลาดเคลื่อนประเภท II เขียนแทนด้วย β

ซึ่ง α และ β เป็นความน่าจะเป็นที่จะตัดสินใจผิด $1 - \alpha$ และ $1 - \beta$ เป็นความน่าจะเป็นที่จะตัดสินใจถูกต้อง ไม่ว่าจะตัดสินใจอย่างไรก็มีโอกาสคลาดเคลื่อนได้ทั้ง 2 ชนิด ดังนั้นการตัดสินใจในแต่ละครั้งจึงต้องการให้ความน่าจะเป็นที่จะเกิดจากความคลาดเคลื่อนทั้ง 2 ชนิดมีค่าน้อย ซึ่งพบว่า α และ β จะมีค่าลดลงถ้าขนาดของตัวอย่างเพิ่มขึ้น แต่การเพิ่มของขนาดตัวอย่างก็จะเป็นการเพิ่มค่าใช้จ่ายและเวลามากยิ่งขึ้น นอกจากนั้นแล้วยังพบว่า α และ β มีความสัมพันธ์กัน โดยถ้า α มีค่าลดลง β จะเพิ่มขึ้นหรือถ้า α มีค่าเพิ่มขึ้น β ก็จะมีค่าลดลง แต่ผลบวกของ α และ β ไม่เท่ากับ 1 และโดยเหตุที่ความน่าจะเป็นของการเกิดความคลาดเคลื่อนชนิดที่ 2 หรือ β มีผลกระทบต่อ การตัดสินใจมากกว่าความน่าจะเป็นของการเกิดความคลาดเคลื่อนชนิดที่ 1 หรือ α ด้วยเหตุนี้ ในการทดสอบสมมติฐานแต่ละครั้งผู้ทดสอบต้องทำการเลือกขนาดของ α และ n ไว้ก่อน แล้วค่าของ β ก็จะถูกกำหนดโดย α และ n ข้างต้น ทั่วไปการทดสอบสมมติฐานนิยมกำหนดระดับความมีนัยสำคัญ $\alpha = 0.05$ หรือ 0.01

ประโยชน์ของสมมุติฐาน

- 1) ช่วยบอกขอบเขตของปัญหา
- 2) ช่วยชี้แนวทางในการวางแผนการวิจัย
- 3) ช่วยให้นักวิจัยมีความคิดแจ่มแจ้งในเรื่องที่ทำการวิจัย
- 4) เป็นแนวทางในการลงสรุป

ลักษณะของสมมุติฐานที่ดี

- 1) สอดคล้องกับจุดมุ่งหมายของการวิจัย
- 2) อธิบายหรือตอบคำถามได้หมด
- 3) แต่ละข้อควรตอบคำถามเพียงข้อเดียว
- 4) สอดคล้องกับสภาพความเป็นจริง
- 5) สมเหตุสมผลตามทฤษฎี
- 6) เข้าใจง่าย ชัดเจน
- 7) สามารถตรวจสอบได้
- 8) มีขอบเขตพอเหมาะ
- 9) มีอำนาจในการพยากรณ์สูง

การทดสอบสมมุติฐาน (Tests of Hypothesis)

จากผลการทดสอบโดยการวัดของโปรแกรมที่พัฒนาขึ้นมา เปรียบเทียบกับเครื่องมือวัด JDSU ว่าจะสามารถยอมรับได้หรือไม่ แลไหนนั้น ก็โดยการใช้วิธีการทดสอบสมมุติฐาน ซึ่งมีวิธีการดังนี้

โดยจากการวัดทดสอบจำนวน 10 ครั้ง ที่ CODEC ต่างๆ ซึ่งจะนำค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานมาคำนวณโดยใช้สูตร

$$\bar{X}_1 = \frac{\{\sum_{i=1}^{n_1} X_{1i}\}}{n_1}$$

$$\bar{X}_2 = \frac{\{\sum_{i=1}^{n_2} X_{2i}\}}{n_2}$$

$$Z = \frac{(\bar{X}_1 - \bar{X}_2)}{\sqrt{(\sigma_1^2/n_1) + (\sigma_2^2/n_2)}}$$

เมื่อ \bar{x} คือ ค่าเฉลี่ยที่วัดได้

σ คือ ค่าเบี่ยงเบนมาตรฐาน

n คือ จำนวนที่ทดสอบ

2.12 งานวิจัยที่เกี่ยวข้อง

2.11.1 งานวิจัยเรื่องโปรแกรมช่วยออกแบบเครือข่าย VoIP (อรศรี ศรีระษา, 2545) ได้สรุปไว้ว่า โปรแกรมช่วยออกแบบเครือข่าย VoIP เป็นเครื่องมือที่ช่วยออกแบบโดยมุ่งเน้นการคำนวณหาขนาดของอุปกรณ์หลักต่างๆ ในโครงข่าย VoIP เช่น ขนาดของ Gateway จำนวนของ Gatekeeper ขนาดของสื่อสัญญาณที่ใช้เชื่อมโยงกันในโครงข่ายไอพี โดยคำนึงคุณภาพการให้บริการของสัญญาณเสียงคือ เรื่องข้อจำกัดของความล่าช้าของสัญญาณและประสิทธิภาพการใช้งานของวงจรเพื่อหลีกเลี่ยงการเกิดสภาวะปริมาณทราฟฟิกคับคั่งในเครือข่าย โดยจะแสดงผลการออกแบบโครงสร้างของโครงข่าย VoIP และสามารถนำไปเป็นพื้นฐานประกอบการออกแบบเครือข่าย VoIP จริงเพื่อใช้เป็นข้อมูลในการวิเคราะห์และตัดสินใจหารูปแบบเครือข่ายที่เหมาะสมและสอดคล้องตามปริมาณความต้องการใช้งานและค่าใช้จ่ายเพื่อเป็นข้อมูลในการวิเคราะห์และตัดสินใจเพื่อหารูปแบบเครือข่ายที่เหมาะสมและสอดคล้องกับปริมาณการใช้งานภายในเครือข่าย

2.11.2 งานวิจัยเรื่อง ระบบการวิเคราะห์แบนด์วิดท์ของเครือข่ายสำหรับการส่งข้อมูลเสียงบนไอพี (วาริน เล้าสกุล, 2544) ได้สรุปไว้ว่าการพัฒนาโปรแกรมวิเคราะห์แบนด์วิดท์จะใช้เป็นเครื่องมือสนับสนุนการตัดสินใจของผู้ออกแบบเครือข่ายในการออกแบบขนาดแบนด์วิดท์แต่ละเส้นทางที่เชื่อมโยงโหนดต่างๆ เข้าหากันว่าควรจะมีค่าความเร็วของการส่งข้อมูลสายเท่าไรเพราะว่าค่าเช่าวงจรจะเพิ่มราคาตามขนาดความเร็วการส่งข้อมูลสาย ดังนั้นผู้ออกแบบเครือข่ายจึงต้องออกแบบให้เส้นทางเชื่อมต่อโหนดภายในเครือข่ายมีขนาดที่เหมาะสม มิฉะนั้นจะทำให้ต้นทุนการสร้างเครือข่ายสูงจนเกินจำเป็น โดยโปรแกรมวิเคราะห์แบนด์วิดท์ มีจุดประสงค์หลักเพื่อนำเสนอผลการคำนวณที่แสดงให้เห็นถึงขนาดความเร็วการส่งข้อมูลของสายบนเส้นทางเชื่อมโยงโหนดต่างๆ ที่มีต้นทุนเครือข่ายต่ำที่สุดในขณะที่ค่า Round Trip Time (RTT) ของทุกเส้นทางในเครือข่ายจะมีค่าไม่เกินค่าที่กำหนดไว้ที่ 200 มิลลิวินาที

จากการศึกษางานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์ที่นำเสนอสามารถเปรียบเทียบคุณสมบัติได้ดังตารางที่ 2.6

ตารางที่ 2.6 การเปรียบเทียบคุณลักษณะของงานวิจัยที่เกี่ยวข้อง

ความสามารถของโปรแกรม งานวิจัย	VNDAT (1)	BASVN (2)	QSMP (3)
หาขนาดของเกตเวย์	✓	✗	✗
หาจำนวน Gatekeeper	✓	✗	✗
หาขนาดของสื่อสัญญาณ	✓	✓	✗
วัดแบนด์วิดท์ที่เหลือของสื่อสัญญาณ	✗	✗	✓
วัดการสูญหายขอแพ็คเก็ต	✗	✗	✓
วัดความผันแปรเฉลี่ยของระยะเวลาดีเลย์ (Jitter)	✗	✗	✓
หา Concurrent Users ในแต่ละ CODEC	✗	✗	✓

หมายเหตุ 1. VNDAT = VoIP Network Design Assistant Tool

2. BASVN = Bandwidth Analysis System for VoIP Network

3. QSMP = Development of a Quality of Service Measurement Program for Voice over IP usage (นำเสนอในงานวิทยานิพนธ์ฉบับนี้)

บทที่ 3

ระเบียบวิธีวิจัย

3.1 แนวทางการวิจัยและพัฒนา

งานวิจัยนี้มีวัตถุประสงค์พัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP โดยโปรแกรมที่พัฒนาจะเป็นการสร้างโปรแกรมประยุกต์จากโปรแกรมวิซวลเบสิก ซึ่งจะมี Freeware เป็นเครื่องมือในการช่วยพัฒนาคือ iperf และ ping บนปฏิบัติการ windows มาทำงานร่วมกัน โดย iperf สามารถวัดค่าความจุของโครงข่าย แพ็คเก็ตสูญหาย และความแปรผันของความล่าช้าทางเวลา ส่วน ping บนปฏิบัติการ windows ก็จะสามารถวัดค่า ความล่าช้าของแพ็คเก็ต ซึ่งสามารถนำเสนอเพื่อเป็นแนวทางในการปรับปรุงแบนด์วิดท์ของเส้นทางได้ โดยแบ่งขั้นตอนการวิจัยออกเป็น 2 ส่วน ๆ ดังต่อไปนี้

3.1.1 ศึกษาและรวบรวมข้อมูล

- 1) จัดหา วิเคราะห์และศึกษาโปรแกรมที่เป็น Freeware ที่จะนำไปวัดค่าความจุของโครงข่าย การล่าช้าทางเวลาและความแปรผันของความล่าช้าทางเวลา
- 2) ศึกษาการส่งสัญญาณเสียงผ่านโครงข่ายไอพี การทำงานของโปรโตคอลและปัจจัยที่มีผลกระทบต่อคุณภาพเสียง เพื่อใช้เป็นแนวทางในการพัฒนาระบบ
- 3) ศึกษาวิธีการบีบอัดหรือเข้ารหัสสัญญาณเสียง แบบต่างๆ เพื่อใช้เป็นข้อมูลในการพัฒนาระบบ
- 4) ศึกษาเครื่องมือที่ใช้ในการพัฒนาโปรแกรมโดยจะใช้โปรแกรมวิซวลเบสิก เพื่อใช้สร้างโปรแกรมสำหรับคำนวณค่า MOS และ จำนวนผู้ใช้งานในโครงข่ายไอพีบนปฏิบัติการ Windows โดยอาศัยหลักการออกแบบโปรแกรมลักษณะ Visualize ซึ่งเพียงแต่เลือก Control ที่เหมาะสมแล้ววาดลงบน Form ก็สามารถสร้างจอภาพที่ติดต่อกับผู้ใช้ได้

3.1.2 การออกแบบโปรแกรม

การออกแบบลักษณะรูปแบบการทำงานของโปรแกรมจะมี ส่วนที่ติดต่อกับผู้ใช้ ที่สามารถพัฒนาได้อย่างสะดวกเนื่องจากการออกแบบหน้าจอของโปรแกรมวิซวลเบสิกเป็นแบบกราฟฟิกโดยใช้เทคโนโลยีการลากแล้ววาง (Drag and Drop) ก็จะได้ส่วนประกอบที่ต้องการและหน้าจอที่ออกมาในขณะรันแอปพลิเคชันจะมีลักษณะเหมือนกับหน้าจอตอนที่ออกแบบ ซึ่งในส่วนนี้จะเป็นส่วนที่ให้ผู้ใช้งานกำหนดค่าความต้องการของผู้ใช้งาน รวมถึงขนาดความจุของโครงข่าย เพื่อ

ส่งค่าดังกล่าวทั้งหมดเข้าไปในโครงข่ายเพื่อให้โปรแกรมที่พัฒนาขึ้น หาค่าความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา การสูญหายของแพ็กเก็ต และ ค่าความจุที่เหลือของเครือข่าย เพื่อนำไปคำนวณหาค่า MOS และ จำนวนผู้ใช้งานที่สามารถใช้ VoIP ได้พร้อมกันได้ในเวลาเดียวกัน โดยสามารถแบ่งส่วนการติดต่อกับผู้ใช้งานออกเป็น 2 ส่วนๆ ดังนี้

1) ส่วนกำหนดความต้องการจากผู้ใช้งาน ซึ่งส่วนนี้จะป้อนอินพุต (Input) จากผู้ใช้งาน กำหนดค่าความต้องการเช่นการเลือกใช้งานว่าจะเลือกวัดอะไร เช่น วัด MOS หรือ Concurrent user การเลือก Mode จะเป็นแบบไหน Server หรือ Client กำหนดข้อมูลทางด้านปริมาณกราฟฟิก เช่น แบนด์วิดท์, IP Address และ CODEC

2) ส่วนแสดงผลคุณภาพของโครงข่าย VoIP โดยจะนำข้อมูลที่รับมาจากส่วนติดต่อกับผู้ใช้งานและ ข้อมูลที่วัดได้จากโครงข่าย มาประมวลผล

3.2 เครื่องมือที่ใช้ในงานวิจัย

3.2.1 ฮาร์ดแวร์

1) คอมพิวเตอร์ 5 เครื่อง ใช้เป็น User Agent Server (UAS) และ User Agent Client (UAC) อย่างละ 2 เครื่อง และใช้เป็น SIP Server (Asterisk) 1 เครื่อง

2) เครื่องมือวัด JDSU รุ่น HST-3000 Handheld Services Tester เพื่อใช้วัดเปรียบเทียบกับโปรแกรมที่พัฒนาขึ้นมา

3) Modem Router 1 ตัว ยี่ห้อ D-Link รุ่น DSL- 2640T (ตาม Spec จะรองรับแบนด์วิดท์สูงสุดที่ 100 Mbps) เพื่อนำมาทำเป็นเครือข่ายจำลองสำหรับการทดสอบ

4) Switch 1 ตัว ยี่ห้อ TP-Link รุ่น TL-SF100D (ตาม Spec จะรองรับแบนด์วิดท์สูงสุดที่ 100 Mbps) เพื่อนำมาทำเป็นเครือข่ายจำลองสำหรับการทดสอบ

5) Hub 1 ตัว ยี่ห้อ Compex รุ่น TP1008C (ตาม Spec จะรองรับแบนด์วิดท์สูงสุดที่ 10 Mbps) เพื่อนำมาทำเป็นเครือข่ายจำลองสำหรับการทดสอบ

3.2.2 ซอฟต์แวร์

1) โปรแกรม iperf (Freeware) ที่นำมาใช้พัฒนาในงานวิจัยนี้ เพื่อนำมาวัดค่าความจุของเครือข่าย ความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลาและการสูญหายของแพ็กเก็ต

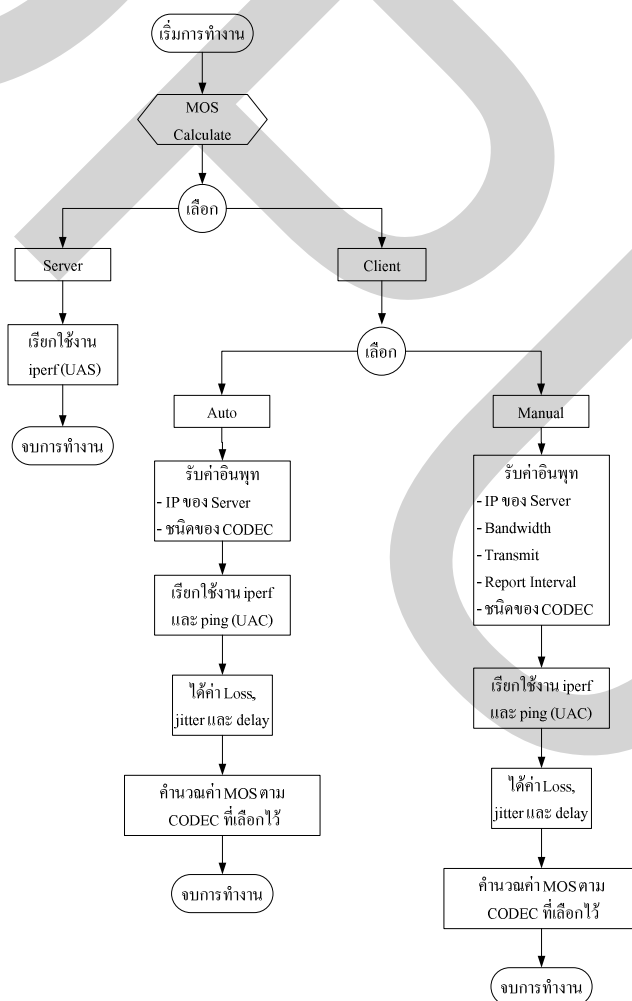
2) Ping บนปฏิบัติการ Windows เพื่อนำมาวัดค่าความล่าช้าทางเวลา

3) โปรแกรม Visual Basic เพื่อใช้ในการออกแบบ แอปพลิเคชันบนวินโดวส์

3.4 ขั้นตอนและวิธีดำเนินงาน

3.4.1 แนวคิดการทำงานของโปรแกรม

ก่อนที่จะทำการออกแบบผู้วิจัยมีแนวคิดที่จะนำเอาโปรแกรมที่เป็น Freeware ที่สามารถวัดค่าความจุของเครือข่าย ความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ต ซึ่งจากการที่ได้ศึกษาค้นคว้าก็ได้พบว่าโปรแกรมที่เป็น Freeware ตัวหนึ่งที่ชื่อ iperf ซึ่งสามารถวัดค่าดังกล่าวข้างต้นได้ แต่ไม่สามารถวัดได้ค่าความล่าช้าทางเวลา จึงได้นำเอา Command ping ที่ใช้งานบนปฏิบัติการ Windows มาทำงานร่วมกัน โดยใช้โปรแกรมวิซวลเบสิกเป็นตัวเชื่อมโยง ซึ่งทั้ง 2 โปรแกรมให้ทำงานร่วมกัน ซึ่งมีลักษณะการออกแบบให้ทำงานเป็น 2 ลักษณะคือ การวัดค่า MOS และการวัดค่า Concurrent User ซึ่งขั้นตอนทำงานตามแผนภาพ (Flowchart) ดังแสดงดังต่อไปนี้



รูปที่ 3.1 แผนภาพการทำงานของโปรแกรมในการคำนวณค่า MOS

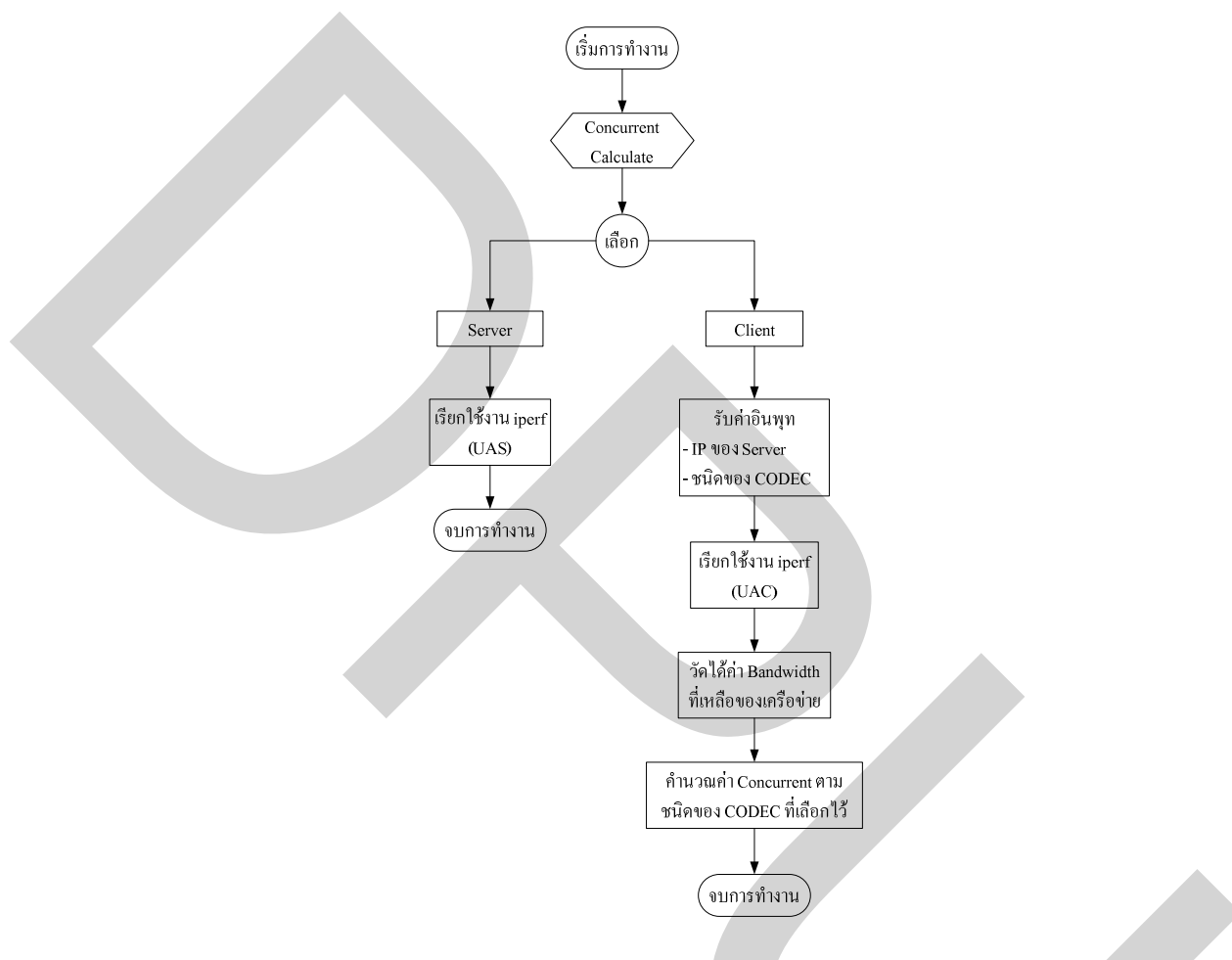
3.4.1.1 จากรูปที่ 3.1 เป็นแผนภาพแสดงการทำงานของโปรแกรมในการคำนวณหาค่า MOS ซึ่งจะสามารถอธิบายการทำงานได้ดังนี้

เมื่อต้องการให้โปรแกรมที่พัฒนาขึ้นทำการวัดค่า MOS สิ่งที่จะต้องทำเป็นอันดับแรกคือจะต้องให้โปรแกรมที่เครื่องแม่ข่ายทำงานก่อน โดยโปรแกรมที่ทำหน้าที่เป็นเครื่องแม่ข่ายไม่ต้องการข้อมูลอะไรมา เพียงแต่แค่เลือกว่าต้องการจะวัดค่า MOS กับการเลือกให้โปรแกรมทำงานเป็น Server เท่านั้น ก็สามารถให้โปรแกรมที่เป็นเครื่องแม่ข่ายทำงานได้เลย โดยโปรแกรมที่เป็นเครื่องแม่ข่ายเบื้องหลังจะมีการทำงานของ Iperf เพียงอย่างเดียวเท่านั้น โดยที่อีกฝั่งหนึ่งของเครื่องแม่ข่ายก็ต้องเลือกให้โปรแกรมทำงานเป็นเครื่องลูกข่าย ซึ่งเครื่องลูกข่ายจะมีการทำงานเป็น 2 ลักษณะคือ แบบ Auto และ Manual โดยการทำงานของแต่ละชนิดพอที่จะอธิบายตามขั้นตอนได้ดังต่อไปนี้

1) การให้โปรแกรมเครื่องลูกข่ายทำงานแบบ Auto โดยผู้ใช้งานจะต้องป้อนค่าลงไป ในหน้าต่างของโปรแกรมตามที่กำหนด ซึ่งก็คือ IP Address ของเครื่องแม่ข่าย และชนิดของ CODEC ที่ต้องการจะทดสอบ แล้วทำการให้โปรแกรมที่เป็นเครื่องลูกข่ายทำงาน โดยการทำงานของโปรแกรมที่พัฒนาขึ้นนี้ เบื้องหลังจะมีการทำงานของโปรแกรม Iperf และ Ping โดยเครื่องลูกข่ายจะทำการส่ง UDP Protocol ซึ่งมีแบนด์วิดท์เท่ากับ CODEC ที่ผู้ใช้งานเลือกไว้ก่อนหน้านี้ ตัวอย่างเช่น ถ้าผู้ใช้งานเลือก CODEC G.711, G.729 และ GSM โปรแกรมที่ทำตัวเป็นเครื่องลูกข่ายก็จะทำการส่งแบนด์วิดท์ที่มีค่าเท่ากับ 88, 32, 37.2 Kbps ตามลำดับ (รวมเฮดเดอร์) ออกไปยังโปรแกรมตัวที่เป็นเครื่องแม่ข่าย และระยะเวลาที่กำหนดไว้ในการส่งไปหาเครื่องแม่ข่ายเท่ากับ 30 วินาที เมื่อครบระยะกำหนดเวลา เครื่องแม่ข่ายก็จะรายงานค่า ความล่าช้าทางเวลา ค่าความแปรผันของความล่าช้าทางเวลา และค่าการสูญหายของแพ็กเก็ต ไปยังเครื่องลูกข่าย ต่อจากนั้นโปรแกรมที่ทำหน้าที่เป็นเครื่องลูกข่ายก็จะทำการคำนวณหาค่า MOS ให้อัตโนมัติ โดยการนำค่าที่วัดได้จากเครื่องแม่ข่ายมาคำนวณค่า MOS ก็คือค่าความล่าช้าทางเวลา ค่าความแปรผันของความล่าช้าทางเวลา ค่าการสูญหายของแพ็กเก็ต และชนิดของ CODEC ที่ผู้ใช้งานเลือก

2) การให้โปรแกรมเครื่องลูกข่ายทำงานแบบ Manual จุดประสงค์ที่ต้องการให้เครื่องลูกข่ายทำงานลักษณะนี้ก็เพราะว่า เมื่อผู้ใช้งานต้องการวัดค่า MOS ในกรณีที่ต้องการ Make call จำนวนหลาย ๆ call พร้อม ๆ กัน ก็สามารถทำได้โดยการใส่ค่าลงในช่องของ Bandwidth ตามที่เราต้องการตัวอย่างเช่น เราต้องการหาค่า MOS เมื่อเราใช้งานโดยการโทรพร้อมกัน 3 สาย เมื่อใช้ CODEC G.711 เราก็ใส่ค่าลงในช่องของ Bandwidth เท่ากับ $88*3 = 264$ kbps เป็นต้น ส่วนค่าอื่น ๆ ที่จำเป็นจะต้องใส่ เมื่อเลือกการทำงานแบบ Manual ก็คือ ค่า Transmit ซึ่งเป็นค่าระยะเวลาที่จะทำการวัดว่าต้องการระยะเวลาในการวัดนานแค่ไหน มีหน่วยเป็นวินาที สุดท้ายเป็นช่อง

ใส่ค่า Report Interval ซึ่งเป็นค่าระยะเวลาที่โปรแกรมจะรายงานออกมาให้ผู้ทำการวัดทราบว่าจะให้รายงานให้ผู้วัดทราบทุก ๆ กี่วินาที



รูปที่ 3.2 แผนภาพการทำงานของโปรแกรมในการคำนวณหาค่า Concurrent Use

3.4.1.2 ส่วนในรูปที่ 3.2 เป็นแผนภาพแสดงการทำงานของโปรแกรมในการคำนวณหาค่า Concurrent User ซึ่งจะมีการทำงานพอสรุปได้ดังนี้

เมื่อต้องการให้โปรแกรมที่พัฒนาขึ้นทำการวัดหาค่า Concurrent Use โดยตัวโปรแกรมที่เลือกเป็นเครื่องแม่ข่าย ก็ทำเช่นเดียวกับการวัดค่า MOS เพียงต่างกันที่ต้องเลือกวัดค่า Concurrent Use เท่านั้น แล้วให้เครื่องแม่ข่ายทำงานก่อน โดยโปรแกรมที่เป็นเครื่องแม่ข่ายจะมีโปรแกรม Iperf อยู่เบื้องหลังเพียงโปรแกรมเดียว โดยที่อีกฝั่งหนึ่งของเครื่องแม่ข่าย ก็ต้องเลือกให้โปรแกรมทำงานเป็นเครื่องลูกข่าย โดยผู้ใช้จะต้องป้อนค่าลงในโปรแกรมตามที่กำหนด ซึ่งก็คือ IP Address ของเครื่องแม่ข่าย และเลือกชนิดของ CODEC ที่จะใช้งาน แล้วก็ทำการ Run โปรแกรม เมื่อโปรแกรม

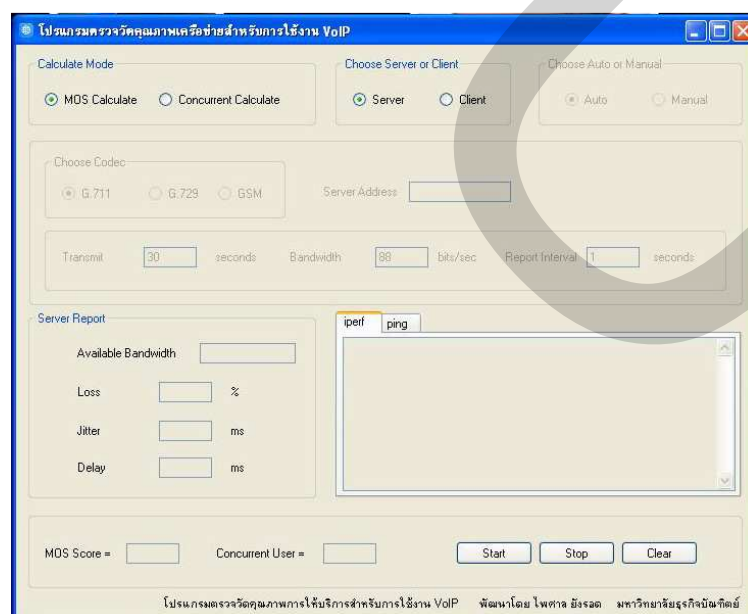
ทำการวัดค่า Bandwidth ที่เหลือของเครือข่ายออกมาได้ ก็จะคำนวณค่า ผู้ใช้งาน VoIP ในเครือข่ายที่สามารถใช้งานพร้อม ๆ กัน โดยค่าของ Concurrent Use ที่ได้ก็คือ ค่าของแบนด์วิดท์ที่เหลือที่วัดได้จากเครือข่าย นำมาหารด้วย แบนด์วิดท์ของ CODEC แต่ละชนิดที่ผู้วัดทำการเลือกไว้ก่อนนี้ โดย CODEC G.711 G.729, GSM จะมีค่าแบนด์วิดท์เท่ากับ 88, 32, 37.2 Kbps ตามลำดับ (รวมเฮดเดอร์) ตัวอย่างเช่น เมื่อวัดแบนด์วิดท์ในเครือข่ายได้เท่ากับ 8 Mbps และได้เลือก CODEC ที่ใช้เป็น G.711 ก็จะได้ค่า Concurrent Use เท่ากับ 39.09 User พร้อมกัน แต่โปรแกรมจะไม่แสดงทศนิยมจะแสดงเฉพาะจำนวนเต็มเท่านั้นคือ 39 User พร้อมกันเป็นต้น

3.4.2 การออกแบบโปรแกรม

ขั้นตอนในการพัฒนาโปรแกรม ประกอบด้วยขั้นตอนหลัก 2 ขั้นตอน ดังนี้

3.4.2.1 การสร้างจอภาพของโปรแกรม

จากที่ได้ออกแบบลักษณะการทำงานของโปรแกรมตามโฟร์ซาร์ต เป็นที่เรียบร้อยแล้ว ถึงตรงนี้ โปรแกรมที่ช่วยในการออกแบบการพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP จะถูกพัฒนาขึ้นด้วยโปรแกรมวิซวล เบสิก เนื่องจากโปรแกรมวิซวล เบสิก เป็นโปรแกรมภาษาที่สามารถพัฒนาแอปพลิเคชันบนวินโดวส์ได้อย่างสะดวกและรวดเร็ว มีเครื่องมือสนับสนุนมากมายทั้งจากไมโครซอฟท์และบริษัทอื่นๆ และเป็นโปรแกรมแอปพลิเคชันที่สนับสนุนการคำนวณและมีใช้งานกันอย่างแพร่หลาย ซึ่งในขั้นต้นจะเป็นการสร้างจอภาพของโปรแกรมเพื่อใช้ในการติดต่อกับผู้ใช้ จะมีลักษณะตามรูปที่ 3.3



รูปที่ 3.3 ลักษณะจอภาพของโปรแกรม

จากรูปแบบจอภาพของโปรแกรม สามารถอธิบายการทำงานของโปรแกรมออกเป็น 2 ส่วนใหญ่ ๆ ส่วนที่ผู้ใช้ต้องป้อนข้อมูล และ ส่วนแสดงผลของโปรแกรม ดังต่อไปนี้

1) ส่วนที่ผู้ใช้ต้องป้อนข้อมูล

(1) Calculate Mode เป็นการเลือกให้โปรแกรมคำนวณหา MOS หรือ คำนวณ Concurrent use จากเครือข่ายที่ต้องการจะวัด

(2) Server or client เป็นส่วนที่จะให้โปรแกรมทำงานเป็น Server หรือ Client

(3) Auto and Manual เป็นการเลือกที่จะให้โปรแกรมทำงานแบบ อัตโนมัตหรือผู้ใช้งานต้องการป้อนข้อมูลของส่วนต่าง ๆ เอง

(4) CODEC เป็นการเลือกความต้องการของผู้ใช้ว่าต้องการใช้การเข้ารหัสเสียงแบบใดในเครือข่าย

(5) Server Address เป็นส่วนที่ผู้ใช้งานต้องป้อนข้อมูลของ IP Address Server เมื่อผู้ใช้งานเลือกการทำงานเป็น Client

(6) Bandwidth เป็นส่วนที่ผู้ใช้งานต้องป้อนค่า แบนด์วิดท์ ที่จะให้โปรแกรมส่งค่าแบนด์วิดท์นี้เข้าไปในเครือข่ายที่ต้องการวัด เมื่อผู้ใช้งานเลือกการทำงานแบบ Manual ในการใช้งานเมื่อวัด MOS

(7) Transmit เป็นส่วนที่ผู้ใช้งานต้องป้อนค่าซึ่งมีค่าเป็นวินาที ซึ่งเป็นระยะเวลาที่ผู้ใช้งานต้องการจะวัดว่าต้องการใช้ระยะเวลาในการวัดนานเท่าไร เมื่อผู้ใช้งานเลือกการทำงานแบบ Manual

(8) Report Interval เป็นส่วนที่ผู้ใช้งานต้องป้อนค่าซึ่งมีค่าเป็นวินาที ซึ่งเป็นระยะเวลาที่ผู้ใช้งานต้องการให้โปรแกรมรายงานทุก ๆ กี่วินาที เมื่อผู้ใช้งานเลือกการทำงานแบบ Manual

2) ส่วนแสดงผลของโปรแกรม

(1) Available Bandwidth เป็นแบนด์วิดท์ที่เหลือที่โปรแกรมสามารถวัดได้จากเครือข่าย

(2) Loss เป็นการสูญเสียของข้อมูลในเครือข่ายที่สามารถวัดได้ซึ่งจะมีหน่วยเป็นเปอร์เซ็นต์

(3) Jitter เป็นการความผันแปรเฉลี่ยของระยะเวลาเฉลี่ย ในเครือข่ายที่สามารถวัดได้โดยจะมีหน่วย ms

(4) Delay เป็นความล่าช้าทางเวลาที่สามารถวัดได้จากเครือข่าย โดยจะมีหน่วยเป็น ms

(5) MOS Score เป็นค่า MOS ที่โปรแกรมสามารถคำนวณได้จากเครือข่ายที่ทำการวัด

(6) Concurrent Use เป็นค่าที่ผู้ใช้งานสามารถใช้งานโทรศัพท์ที่ไอพีได้พร้อมกันในเวลาขณะนั้น

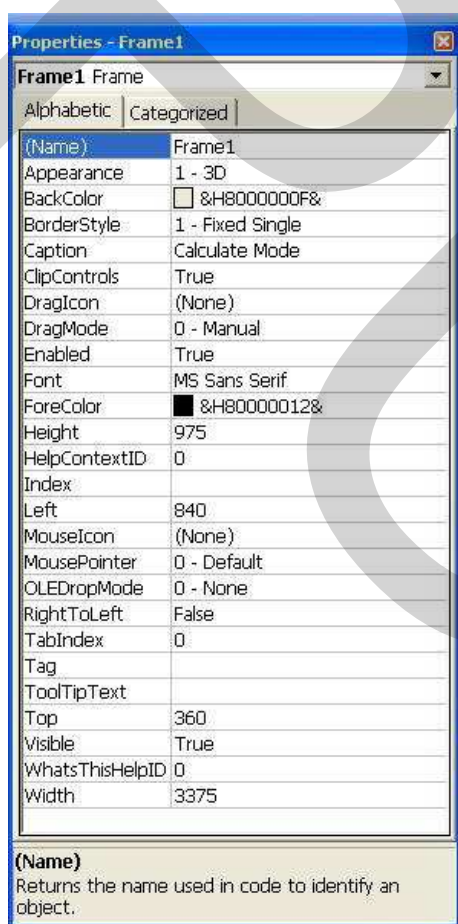
(7) Start เป็นปุ่มให้โปรแกรมเริ่มทำงานเมื่อป้อนค่าต่าง ๆ เรียบร้อย

(8) Stop เป็นปุ่มให้โปรแกรมหยุดการทำงานทันที เมื่อต้องการหยุดการทำงานของโปรแกรม

(9) Clear เป็นปุ่มที่ผู้ใช้งานต้องการล้างข้อมูลทั้งหมดที่ผู้ใช้งานป้อนเข้าไปในช่องต่าง ๆ

3.4.2.2 การเขียนคำสั่ง

เมื่อขั้นตอนการออกแบบสร้างจอภาพของโปรแกรมเพื่อใช้ในการติดต่อกับผู้ใช้เสร็จสิ้น ขั้นตอนต่อมา ก็จะเป็นการกำหนดคุณสมบัติ Property ให้กับแต่ละ Object ที่ปรากฏอยู่บน Form ซึ่งรูปที่จะแสดงให้เห็นเป็นการกำหนดค่าให้กับ Property ของ Calculate Mode ดังรูปที่ 3.4



รูปที่ 3.4 ตัวอย่างรูป Property ของ Calculate Mode

การเขียนรูปแบบของคำสั่งเพื่อกำหนดการทำงานให้แต่ละ Object ภายใต้อุเหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับจอภาพนั้นๆ โดยจะแสดงรายละเอียดของคำสั่งเป็นตัวอย่าง ซึ่งเป็นคำสั่งบน Object บนปุ่มที่ชื่อว่า Server ดังแสดงในรูปที่ 3.5

```
Private Sub OptServer_Click()
If StartStatus Then
cmdStartServer.Enabled = False
Else
cmdStartServer.Enabled = True
End If
If StartStatus Then
cmdGetData.Enabled = True
Else
cmdGetData.Enabled = False
End If

cmdGetData.Visible = True
cmdProcess.Visible = False
txtListenPort.Enabled = True
txtListenPort.BackColor = &H80000005
txtServerAddress.Text = ""
txtServerAddress.Enabled = False
txtServerAddress.BackColor = &H8000000F
txtBandWidth.Enabled = False
txtPort.Enabled = False
txtReInterval.Enabled = False
txtTransmit.Enabled = False
txtBandWidth.BackColor = &H8000000F
txtPort.BackColor = &H8000000F
txtReInterval.BackColor = &H8000000F
txtTransmit.BackColor = &H8000000F
End Sub
```

รูปที่ 3.5 ลักษณะของรูปแบบคำสั่งบน Object ที่ชื่อว่า Server

ในรูปที่ 3.5 จะแสดงตัวอย่างรูปแบบของคำสั่งเพื่อให้คำนวณการหา Concurrent User ที่เกิดขึ้นบน Object ที่ชื่อว่า Start เมื่อทำการเลือกให้โปรแกรมทำงานเป็นคิว Client

```

Private Sub calConcurrentUser()
    If txtServerAddress.Text = "" Or txtBandWidth.Text = "" Or txtPort.Text = "" Or txtReInterval.Text = "" Or txtTransmit.Text = "" Then
        Exit Sub
    End If
    Dim BandwidthValue As Single
    Dim ConcurrentUserValue As Integer

    If optPPP.Value = True Then
        If optG711.Value = True Then
            BandwidthValue = 84.8
        End If
        If optG729.Value = True Then
            BandwidthValue = 28.8
        End If
        If optGSM.Value = True Then
            BandwidthValue = 34
        End If
    End If

    If optEthernet.Value = True Then
        If optG711.Value = True Then
            BandwidthValue = 88
        End If
        If optG729.Value = True Then
            BandwidthValue = 32
        End If
        If optGSM.Value = True Then
            BandwidthValue = 37.2
        End If
    End If

    If optFrameRelay.Value = True Then
        If optG711.Value = True Then
            BandwidthValue = 84
        End If
        If optG729.Value = True Then
            BandwidthValue = 28
        End If
        If optGSM.Value = True Then
            BandwidthValue = 33.2
        End If
    End If

    ConcurrentUserValue = Fix((lblBanRemain.Caption * 1024) / BandwidthValue)
    lblConcurrentUser.Caption = ConcurrentUserValue
End Sub

```

รูปที่ 3.6 ตัวอย่างของคำสั่งในการคำนวณหา Concurrent User

จากการศึกษา วิธีการคำนวณหาคุณภาพของเครือข่ายจากโปรแกรมที่พัฒนาขึ้น ซึ่งสามารถวัดค่าความล่าช้าทางเวลา, การสูญหายของแพ็กเก็ต และ ความแปรผันของความล่าช้าทางเวลาแล้ว จากนั้นนำมาคำนวณตามสมการที่ได้ศึกษาค้นคว้ามา ซึ่งใช้ในเครื่องวัดที่มีชื่อว่า PingPlotter (Nessoft, LLC, 2005) ค่า R-factor ที่นำมาใช้ในสมการ มาจากตารางที่ 3.2 ซึ่งแสดงการเปรียบเทียบค่า MOS ต่างๆ กับค่า R-factor

ตารางที่ 3.2 การเปรียบเทียบค่า MOS Score กับค่า R-factor

User Opinion	R Factor	MOS Score
Very satisfied	90 - 100	4.3 – 5.0
Satisfied	80 - 90	4.0 – 4.3
Some users satisfied	70 - 80	3.6 – 4.0
Many users dissatisfied	60 - 70	3.1 – 3.6
Nearly all users dissatisfied	50 -60	2.1 – 3.1
Not recommended	0 - 50	1.0 – 2.6

อ้างอิงจากบทที่ 2 จะเห็นว่าค่า MOS Score ของแต่ละ CODEC มีค่าดังนี้

CODEC	G.711	จะมี MOS สูงสุด	4.20
CODEC	G.729	จะมี MOS สูงสุด	3.91
CODEC	GSM	จะมี MOS สูงสุด	3.57

ตัวอย่างวิธีการคำนวณหาค่า MOS ของโปรแกรมที่พัฒนาโดยเริ่มจาก CODEC G.711 (ค่า MOS ที่ดีที่สุดจะมีค่าเท่ากับ 4.2 ซึ่งอ้างอิงจากเครื่องมือวัด JDSU) ซึ่งจะได้ค่าคงที่เท่ากับ 85.3 ซึ่งค่าคงที่นี้ได้จากการแทนค่าจากสมการที่ (3) ถึงสมการที่ (7) ในบทที่ 2 เพื่อให้ได้ผลลัพธ์ค่า MOS เท่ากับ 4.2 โดยการคำนวณนี้ค่าคงที่นี้ยังไม่รวมกับค่าผลกระทบที่เกิดจากค่าความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ตในเครือข่ายรวมเข้ามา (คือให้ค่าเหล่านั้นเป็น 0) แล้วแทนค่าแล้วให้ได้ผลลัพธ์ของค่า MOS เท่ากับ 4.20

จากนั้นเมื่อใช้โปรแกรมที่เราได้พัฒนาขึ้นมาตรวจวัดในเครือข่ายก็จะมีค่า ความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ต แสดงออกมาให้เห็นก็จะนำค่าเหล่านั้นมาคำนวณตามสมการที่ (3) ถึง (7) ในบทที่ 2 ดังจะแสดงให้เห็นต่อไปนี้

ตัวอย่าง สมมติค่าที่วัดได้จากโปรแกรมที่พัฒนาขึ้นมาสามารถวัดค่าได้ค่าดังนี้

$$\text{Loss} = 2\%$$

$$\text{Jitter} = 3 \text{ ms}$$

$$\text{Delay} = 8 \text{ ms}$$

$$\begin{aligned} \text{จากสมการ Effective Latency} &= [\text{Average Latency} + (\text{Jitter} * 2) + 10] \\ &= [8 + (3*2) + 10] \\ &= 24 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_1 &= 85.3 - (\text{Effective Latency} / 40) \\ &= 85.3 - (24 / 40) \\ &= 84.7 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_r &= R_1 - (\text{Packet Loss} * 2.5) \\ &= 84.7 - (2 * 2.5) \\ &= 79.7 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า MOS} &= 1 + (0.035) * R_r + (.000007) * R_r * (R_r - 60) * (100 - R_r) \\ &= 1 + (0.035) * 79.7 + (.000007) * 79.7 * (79.7 - 60) * \\ &\quad (100 - 79.7) \\ &= 1 + 2.79 + 0.22 \end{aligned}$$

$$\text{ฉะนั้นจะได้ค่า MOS} = 4.01 \quad (\text{เป็นค่า MOS ที่วัดได้จากโปรแกรมที่พัฒนาขึ้นมา})$$

เช่นเดียวกันที่ CODEC G.729 (ค่า MOS ที่ดีที่สุดจะมีค่าเท่ากับ 3.91 ซึ่งอ้างอิงจากเครื่องมือวัด JDSU) ซึ่งจะได้ค่าคงที่เท่ากับ 77.3 ซึ่งค่าคงที่นี้ได้จากการแทนค่าจากสมการที่ (3) ถึงสมการที่ (7) ในบทที่ 2 เพื่อให้ได้ผลลัพธ์ค่า MOS เท่ากับ 3.91 โดยการคำนวณนี้ค่าคงที่นี้ยังไม่รวมกับค่าผลกระทบที่เกิดจากค่าความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ตเกิดในเครือข่ายรวมเข้ามา (คือให้ค่าเหล่านั้นเป็น 0) แล้วแทนค่าแล้วได้ผลลัพธ์ของค่า MOS เท่ากับ 3.91

จากนั้นเมื่อใช้โปรแกรมที่เราได้พัฒนาขึ้นมาตรวจวัดในเครือข่ายก็จะมีค่า ความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ตเกิด แสดงออกมาให้เห็นก็จะนำค่าเหล่านั้นมาคำนวณตามสมการที่ (3) ถึง (7) ในบทที่ 2 ดังจะแสดงให้เห็นต่อไป

ตัวอย่าง สมมติค่าที่วัดได้จากโปรแกรมที่พัฒนาขึ้นมาสามารถวัดค่าได้ค่าดังนี้

$$\text{Loss} = 1 \%$$

$$\text{Jitter} = 2 \text{ ms}$$

$$\text{Delay} = 8 \text{ ms}$$

$$\begin{aligned} \text{จากสมการ Effective Latency} &= [\text{Average Latency} + (\text{Jitter} * 2) + 10] \\ &= [8 + (2*2) + 10] \\ &= 22 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_1 &= 77.3 - (\text{Effective Latency} / 40) \\ &= 77.3 - (22 / 40) \\ &= 76.75 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_r &= R_1 - (\text{Packet Loss} * 2.5) \\ &= 76.75 - (1 * 2.5) \\ &= 74.25 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า MOS} &= 1 + (0.035) * R_r + (.000007) * R_r * (R_r - 60) * (100 - R_r) \\ &= 1 + (0.035) * 74.25 + (.000007) * 74.25 * (74.25 - 60) * \\ &\quad (100 - 74.25) \\ &= 1 + 2.59 + 0.19 \end{aligned}$$

$$\text{ฉะนั้นจะได้ค่า MOS} = 3.78 \quad (\text{เป็นค่า MOS ที่วัดได้จากโปรแกรมที่พัฒนาขึ้น})$$

ส่วนที่ CODEC GSM (ค่า MOS ที่ดีที่สุดจะมีค่าเท่ากับ 3.57 ซึ่งอ้างอิงจากเครื่องมือวัด JDSU) ซึ่งจะได้ค่าที่เท่ากับ 69.7 ซึ่งค่าที่นี้ได้จากการแทนค่าจากสมการที่ (3) ถึงสมการที่ (7) ในบทที่ 2 เพื่อให้ได้ผลลัพธ์ค่า MOS เท่ากับ 3.57 โดยการคำนวณนี้ค่าที่นี้ยังไม่รวมกับค่าผลกระทบที่เกิดจากค่าความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ตเกิดในเครือข่ายรวมเข้ามา (คือให้ค่าเหล่านั้นเป็น 0) แล้วแทนค่าแล้วให้ผลลัพธ์ของค่า MOS เท่ากับ 3.57

จากนั้นเมื่อใช้โปรแกรมที่เราได้พัฒนาขึ้นมาตรวจวัดในเครือข่ายก็จะมีค่า ความล่าช้าทางเวลา ความแปรผันของความล่าช้าทางเวลา และการสูญหายของแพ็กเก็ต แสดงออกมาให้เห็นก็จะนำค่าเหล่านั้นมาคำนวณตามสมการที่ (3) ถึง (7) ในบทที่ 2 ดังจะแสดงให้เห็นต่อไปนี้

ตัวอย่าง สมมติค่าที่วัดได้จากโปรแกรมที่พัฒนาขึ้นมาสามารถวัดค่าได้ค่าดังนี้

$$\text{Loss} = 0 \%$$

$$\text{Jitter} = 0 \text{ ms}$$

$$\text{Delay} = 4 \text{ ms}$$

$$\begin{aligned} \text{จากสมการ Effective Latency} &= [\text{Average Latency} + (\text{Jitter} * 2) + 10] \\ &= [4 + (0*2) + 10] \\ &= 14 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_1 &= 69.7 - (\text{Effective Latency} / 40) \\ &= 69.7 - (14 / 40) \\ &= 69.35 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า } R_r &= R_1 - (\text{Packet Loss} * 2.5) \\ &= 69.35 - (0 * 2.5) \\ &= 69.35 \end{aligned}$$

$$\begin{aligned} \text{จากสมการหาค่า MOS} &= 1 + (0.035) * R_r + (.000007) * R_r * (R_r - 60) * (100 - R_r) \\ &= 1 + (0.035) * 69.35 + (.000007) * 69.35 * (69.35 - 60) * \\ &\quad (100 - 69.35) \\ &= 1 + 2.43 + 0.14 \end{aligned}$$

$$\text{ฉะนั้นจะได้ค่า MOS} = 3.57 \quad (\text{เป็นค่า MOS ที่วัดได้จากโปรแกรมที่พัฒนาขึ้นมา})$$

ส่วนในการคำนวณหา Concurrent Use นั้น ก็ได้ใช้โปรแกรม Iperf ในการหาแบนด์วิดท์ที่เหลือของเครือข่าย โดยการป้อนแบนด์วิดท์โดยคร่าวๆ ของเครือข่าย ควรจะป้อนค่าแบนด์วิดท์ที่สูงกว่าแบนด์วิดท์จริงในโครงข่ายเล็กน้อยแล้วคอยสังเกตค่าดีเลย์ และจิสเตอร์ แล้วค่อยๆ ลดค่าแบนด์วิดท์ลงมาจนค่าดีเลย์ และจิสเตอร์ ไม่มีในเครือข่าย นั่นก็คือค่าแบนด์วิดท์ที่เหลือในเครือข่ายที่สามารถใช้งานได้จริง จากนั้นโปรแกรมก็จะทำการคำนวณ ตาม CODEC และ ลักษณะการเชื่อมต่อ ที่ผู้ใช้งานเลือกไว้ ก็จะสามารถคำนวณออกมาเป็นจำนวนผู้ใช้งานจริง ที่สามารถใช้งาน VoIP ได้ในเครือข่าย ณ ขณะนั้น ซึ่ง CODEC ต่างๆ ที่มาคำนวณ จะมีค่าตามตารางที่ 3.3

ตารางที่ 3.3 ค่าแบนด์วิดท์โดยรวมของ CODEC ต่างๆ ในการเชื่อมต่อแบบ Ethernet

TYPE OF CODEC	แบนด์วิดท์รวม (Kbps)
G.711	88
G.729	32
GSM	37.2

ตัวอย่างการคำนวณของโปรแกรมในการหาจำนวนผู้ใช้งานพร้อมกัน อย่างเช่นเมื่อผู้ใช้งานต้องการใช้ CODEC แบบ G.711 ซึ่งมีแบนด์วิดท์รวมเท่ากับ 88 Kbps แล้วทำการ Start โปรแกรม ให้ทำการวัด โปรแกรมวัดค่าแบนด์วิดท์ออกมาได้เท่ากับ 2 Mbps ก็จะมีวิธีการคำนวณค่า Concurrent User ดังนี้

$$\begin{aligned} \text{ค่า Concurrent Use} &= 2000 \text{ Kbps} / 88 \text{ Kbps} \\ &= 22.72 \quad \text{Concurrent User} \end{aligned}$$

แต่โปรแกรมที่พัฒนานี้จะตัดเศษหลังจุดทศนิยมทิ้งไป โดยจะแสดงค่าที่เป็นจำนวนเต็มเท่านั้นคือจะแสดงแค่ 22 Concurrent Use สาเหตุที่ให้แสดงเฉพาะจำนวนเต็มโดยการปัดทศนิยมทิ้งก็เพราะว่า ถ้าเศษปัดทศนิยมขึ้นจะมีผลกับคุณภาพของเสียงเมื่อใช้งานเต็มจำนวน

เมื่อผู้ใช้งานต้องการใช้ CODEC แบบ G.729 ซึ่งมีแบนด์วิดท์รวม เท่ากับ 32 Kbps แล้วทำการ Start โปรแกรม ให้ทำการวัด โปรแกรมวัดค่าแบนด์วิดท์ออกมาได้เท่ากับ 2 Mbps ก็จะมีวิธีการคำนวณค่า Concurrent User ดังนี้

$$\begin{aligned} \text{ค่า Concurrent Use} &= 2000 \text{ Kbps} / 32 \text{ Kbps} \\ &= 62.5 \quad \text{Concurrent User} \end{aligned}$$

โปรแกรมจะแสดงค่าที่เป็นจำนวนเต็มเท่านั้นคือจะแสดงแค่ 62 Concurrent Use เท่านั้น

ตัวอย่างสุดท้ายเมื่อผู้ใช้งานต้องการใช้ CODEC แบบ GSM ซึ่งมีแบนด์วิดท์รวม เท่ากับ 37.2 Kbps แล้วทำการ Start โปรแกรม ให้ทำการวัด โปรแกรมวัดค่าแบนด์วิดท์ออกมาได้เท่ากับ 2 Mbps ก็จะมีวิธีการคำนวณค่า Concurrent User ดังนี้

$$\begin{aligned} \text{ค่า Concurrent Use} &= 2000 \text{ Kbps} / 37.2 \text{ Kbps} \\ &= 53.76 \quad \text{Concurrent User} \end{aligned}$$

โปรแกรมจะแสดงค่าที่เป็นจำนวนเต็มเท่านั้นคือจะแสดงแค่ 53 Concurrent Use เท่านั้น

เมื่อได้ทำการออกแบบทุกอย่างเรียบร้อยแล้ว ไม่ว่าจะเป็นหน้าตาลักษณะการทำงานของโปรแกรมและการเขียนคำสั่ง ของในแต่ละ Object รวมถึงวิธีการคำนวณค่าต่าง ๆ ของโปรแกรมที่พัฒนาขึ้น ขั้นตอนต่อไปก็จะเป็นวัดหาความถูกต้องและวิธีทดสอบโปรแกรมที่ได้พัฒนาขึ้น โดยจะทำสร้างเครือข่ายจำลองอย่างง่าย ๆ เพื่อจะนำมาทดสอบความถูกต้องของโปรแกรมที่พัฒนาขึ้น ซึ่งเบื้องต้นซึ่งจะใช้ Router , Switch อย่างละ 1 ตัว และจะมี PC อีก 2 ตัว เป็นอย่างน้อย โดยจะมีลักษณะดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ลักษณะของเครือข่ายที่ใช้ในการทดสอบ

จากรูปที่ 3.7 สามารถอธิบายวิธีการทำงานของโปรแกรมที่พัฒนาขึ้นโดยการจำลองเครือข่ายขึ้นมาเพื่อการทดสอบโดยจะให้ PC 1 เป็นคอมพิวเตอร์ที่จะให้โปรแกรมที่พัฒนาขึ้นทำงานเป็นเครื่องแม่ข่าย ส่วน PC 2 เป็นคอมพิวเตอร์ที่จะให้โปรแกรมที่พัฒนาขึ้นทำงานเป็นเครื่องลูกข่าย โดยจะต้องให้ เครื่องแม่ข่ายให้ทำงานก่อนแล้วจากนั้นก็ทำการใส่ข้อมูลที่จำเป็นบนเครื่องลูกข่าย แล้วทำการ Run โปรแกรมที่พัฒนาขึ้นก็จะทำการวัดค่า MOS หรือ Concurrent Use ระหว่างเครื่องที่เป็นแม่ข่าย กับเครื่องที่เป็นลูกข่าย ตามจุดประสงค์ของผู้ใช้งานต่อไป

บทที่ 4

การทดสอบระบบ

เนื้อหาในบทนี้จะกล่าวถึงการทดสอบ โปรแกรมที่ได้พัฒนาขึ้น เพื่อประเมินผลการใช้งานและปรับปรุงข้อผิดพลาดที่เกิดขึ้น โดยได้ออกแบบการทดสอบด้วยการจำลองเครือข่ายของระบบ ในการทดสอบระบบซึ่งมีระบบเครือข่ายภายในดังแสดงในรูปที่ 4.1 โดยการทดสอบกำหนดให้มี 2 ลักษณะคือ การทดสอบหาค่า MOS และการทดสอบหา Concurrent User โดยได้ทำการทดสอบระบบตามฟังก์ชันการทำงานต่างๆของระบบ

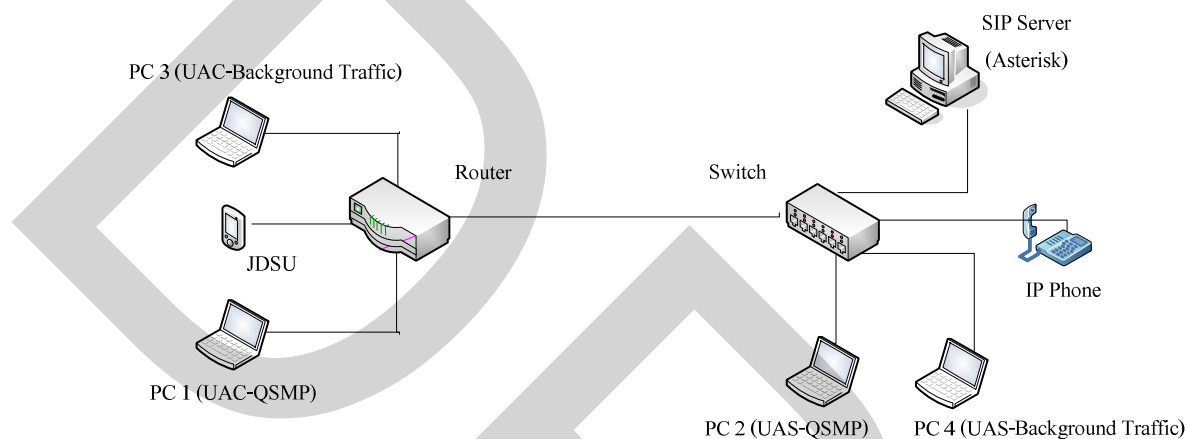
4.1 วิธีการทดสอบโปรแกรมโดยการวัดค่า MOS

งานวิจัยนี้ได้จำลองเครือข่ายขึ้นมาตามรูปที่ 4.1 เพื่อทดสอบโปรแกรมที่ผู้วิจัยได้พัฒนาขึ้น ซึ่งจะมีการพิสูจน์ความถูกต้องของผลลัพธ์จากการตรวจวัด โดยการเปรียบเทียบผลลัพธ์ที่ได้กับเครื่องมือวัดที่ใช้งานอยู่จริงในเครือข่ายของผู้ให้บริการโทรคมนาคม โดยในที่นี้จะใช้เครื่องมือวัดของ JDSU รุ่น HST 3000 ซึ่งจะมีลักษณะการวัดที่แตกต่างกับ โปรแกรมที่ผู้วิจัยได้พัฒนาขึ้น ดังนี้

4.1.1 ลักษณะวิธีการวัดของ JDSU รุ่น HST 3000 ซึ่งจะมีรูปแบบการวัดโดยการใช้เครื่องมือวัด JDSU เป็นตัว SIP Terminal Phone ซึ่งจะต้องทำการ Configuration เครื่องมือวัด JDSU ให้ Register กับ SIP Server (Asterisk) แล้วทำการเรียกไปยัง IP Phone ตามรูปที่ 4.1 เมื่อเครื่องมือวัด JDSU กับ IP Phone สามารถเชื่อมต่อกันได้ ที่เครื่องมือวัด JDSU ก็จะสามารถเรียกดูค่าการสูญหายของแพ็กเก็ต ความล่าช้าของข้อมูล ความผันแปรเฉลี่ยของระยะเวลาดีเลย์และ ค่า MOS บนเครื่องมือวัด JDSU ได้ ซึ่งจะเป็นการวัดคุณภาพของ Voice บน link ระหว่าง Router กับ Switch ซึ่ง Router กับ Switch นี้จะรองรับแบนด์วิดท์ 100 Mbps (ตาม Spec) โดยในระหว่างการวัดนี้ก็จะมีการสร้าง background traffic ให้กับเครือข่ายที่ทำการวัด โดยจะมี PC1 เป็นตัวสร้าง background traffic ที่จะส่งไปยัง PC3 ซึ่งเป็นเครื่องแม่ข่ายที่ทำหน้าที่รับ background traffic

4.1.2 ลักษณะวิธีการวัดของ โปรแกรมที่ผู้วิจัยได้พัฒนาขึ้นนั้น จะเป็นลักษณะ Server กับ Client โดยจากรูปที่ 4.1 จะให้ PC 2 เป็นเครื่องลูกข่าย ที่จะทำการ Run โปรแกรมที่พัฒนาขึ้นมา และใน PC 1 ก็ทำการสร้าง background traffic เพื่อจำลองสภาวะการทำงานในระบบจริง ส่วน PC 4 จะทำหน้าที่เป็นเครื่องแม่ข่ายของ โปรแกรมที่พัฒนาขึ้นมา และใน PC3 ก็ทำหน้าที่เป็นตัวรับ background

traffic (เครื่องแม่ข่าย) ซึ่งเมื่อ PC 2 ส่งข้อมูลเข้าไปในเครือข่ายตามที่ผู้ใช้กำหนด PC 4 ก็จะทำหน้าที่อ่านข้อมูลนั้นที่ แล้วทำการรายงานผลนั้นไปยัง PC 2 ซึ่งจะมีทั้งค่าการสูญหายของแพ็กเก็ต ความล่าช้าของข้อมูล ความผันแปรเฉลี่ยของระยะเวลาดีเลย์ และจะคำนวณค่า MOS ออกมาโดยอัตโนมัติให้ผู้ใช้งานได้ทราบ เพื่อจะได้ทราบคุณภาพของเครือข่ายในขณะนั้น



รูปที่ 4.1 ลักษณะการใช้โปรแกรมที่พัฒนาวัดค่า MOS จากเครือข่ายจำลอง



รูปที่ 4.2 เครื่องมือวัดของ JDSU รุ่น HST 3000

จากการทดสอบเบื้องต้นโดยการสร้างเครือข่ายจำลองและสร้าง background traffic ปริมาณน้อยไปหามากโดยจะเริ่มตั้งแต่ 30, 50, 80, 90, 100 Mbps ตามลำดับ โดยในแต่ละกรณีจะทำการตรวจวัดค่าความล่าช้าทางเวลา ค่าความแปรผันของความล่าช้าทางเวลา ค่าการสูญหายของข้อมูลและค่า MOS จำนวน 10 ครั้ง แล้วนำผลที่ได้มาทำการเปรียบเทียบกับเครื่องมือวัด JDSU ในแต่ละกรณีโดยเลือกใช้ CODEC G.711, G729 และ GSM ตามลำดับตามที่ได้ระบุไว้ในขอบเขตการวิจัยดังแสดงในตารางที่ 4.1 ถึง 4.3

ตารางที่ 4.1 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC G.711

จำนวน Background traffic	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 30 Mbps/1	0	1	6	4.06	0	0	8	4.09
Gen 30 Mbps/2	0	3	6	4.06	0	0	2	4.10
Gen 30 Mbps/3	0	3	6	4.04	0	0	4	4.10
Gen 30 Mbps/4	0	3	6	4.01	0	0	6	4.09
Gen 30 Mbps/5	0	2	2	4.14	0	0.152	8	4.09
Gen 30 Mbps/6	0	2	5	4.06	0	0	4	4.10
Gen 30 Mbps/7	0	2	5	4.04	0	0	4	4.10
Gen 30 Mbps/8	0	2	5	4.04	0	0	6	4.09
Gen 30 Mbps/9	0	3	3	4.11	0	0	4	4.10
Gen 30 Mbps/10	0	3	3	4.09	0	0.001	4	4.10
Gen 50 Mbps/1	0	2	9	4.14	0	0	6	4.09
Gen 50 Mbps/2	0	2	9	4.11	0	0.002	8	4.09
Gen 50 Mbps/3	0	2	9	4.04	0	0	6	4.09
Gen 50 Mbps/4	0	3	9	4.04	0	0	6	4.09
Gen 50 Mbps/5	0	2	5	4.2	0	0	7	4.09
Gen 50 Mbps/6	0	2	5	4.14	0	0	6	4.09
Gen 50 Mbps/7	0	2	5	4.09	0	0	6	4.09
Gen 50 Mbps/8	0	2	5	4.06	0	1.624	5	4.09
Gen 50 Mbps/9	0	2	7	4.06	0	0	6	4.09
Gen 50 Mbps/10	0	2	7	4.14	0	0.005	6	4.09
Gen 80 Mbps/1	0	3	6	4.11	0.69	0.36	5	4.14
Gen 80 Mbps/2	0	3	6	4.11	0.69	1.389	12	4.13
Gen 80 Mbps/3	0	3	12	4.06	0.69	1.762	6	4.14
Gen 80 Mbps/4	0	2	15	4.18	0.69	1.474	8	4.13
Gen 80 Mbps/5	0	2	15	4.04	0	1.179	5	4.19
Gen 80 Mbps/6	0	3	15	3.98	0	1.888	10	4.19
Gen 80 Mbps/7	0	3	15	3.95	0	1.438	12	4.19
Gen 80 Mbps/8	0	2	5	4.2	0.69	1.474	10	4.13
Gen 80 Mbps/9	0	3	5	4.09	0.69	1.156	8	4.13

ตารางที่ 4.1 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC G.711
(ต่อ)

จำนวน Background traffic	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 80 Mbps/10	0	3	13	4.06	0.69	0.457	6	4.14
Gen 90 Mbps/1	0	3	7	4.18	0.69	0.938	10	4.13
Gen 90 Mbps/2	0	4	16	4.16	0.69	3.693	14	4.12
Gen 90 Mbps/3	0	4	16	4.11	0.69	2.414	12	4.13
Gen 90 Mbps/4	0	4	16	4.09	1.4	0.746	10	4.07
Gen 90 Mbps/5	0	2	4	4.18	1.4	0.878	11	4.07
Gen 90 Mbps/6	0	2	8	4.14	1.4	2.318	7	4.07
Gen 90 Mbps/7	0	3	9	4.04	0	3.353	10	4.19
Gen 90 Mbps/8	0	3	10	4.18	0	2.689	6	4.19
Gen 90 Mbps/9	0	3	10	4.11	0.69	2.625	9	4.13
Gen 90 Mbps/10	0	3	10	4.06	0.69	3.658	13	4.13
Gen 100 Mbps/1	1	3	10	4.04	1.4	2.886	11	3.97
Gen 100 Mbps/2	1	3	10	3.91	1.4	1.193	12	3.97
Gen 100 Mbps/3	1	3	10	3.84	1.9	1.863	10	3.92
Gen 100 Mbps/4	2	3	10	3.81	1.4	1.231	9	3.97
Gen 100 Mbps/5	2	3	8	4.01	1.9	1.41	11	3.92
Gen 100 Mbps/6	1	3	9	3.95	1.4	2.207	12	3.97
Gen 100 Mbps/7	1	3	9	3.91	1.4	2.494	11	3.97
Gen 100 Mbps/8	3	2	7	3.98	1.9	3.737	10	3.92
Gen 100 Mbps/9	3	2	7	3.84	2.3	1.581	11	3.88
Gen 100 Mbps/10	2	2	15	3.77	2.1	3.769	12	3.90

ตารางที่ 4.2 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC G.729

จำนวน Background traffic	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 30 Mbps/1	0	1	3	3.91	0	0	3	3.90
Gen 30 Mbps/2	0	1	3	3.91	0	0	2	3.91
Gen 30 Mbps/3	0	1	4	3.91	0	0	2	3.91
Gen 30 Mbps/4	0	1	4	3.91	0	0	3	3.90
Gen 30 Mbps/5	0	1	4	3.91	0	0	2	3.91
Gen 30 Mbps/6	0	2	4	3.91	0	0	2	3.91
Gen 30 Mbps/7	0	1	3	3.91	0	0	3	3.90
Gen 30 Mbps/8	0	1	4	3.91	0	0	2	3.91
Gen 30 Mbps/9	0	2	2	3.91	0	0	2	3.91
Gen 30 Mbps/10	0	2	3	3.91	0	1.058	3	3.90
Gen 50 Mbps/1	0	1	7	3.91	0	0	3	3.90

ตารางที่ 4.2 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC G.729

(ต่อ)

จำนวน Background traffic	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 50 Mbps/2	0	3	7	3.91	0	0.441	2	3.90
Gen 50 Mbps/3	0	2	4	3.91	0	0.03	3	3.90
Gen 50 Mbps/4	0	2	4	3.91	0	1.536	4	3.90
Gen 50 Mbps/5	0	2	5	3.91	0	0.802	4	3.90
Gen 50 Mbps/6	0	2	5	3.91	0	1.793	4	3.90
Gen 50 Mbps/7	0	1	4	3.91	0	0.817	4	3.90
Gen 50 Mbps/8	0	1	5	3.91	0	1.022	4	3.90
Gen 50 Mbps/9	0	1	5	3.91	0	3.045	3	3.90
Gen 50 Mbps/10	0	1	5	3.91	0	0.81	5	3.90
Gen 80 Mbps/1	0	2	5	3.91	0	3.258	5	3.85
Gen 80 Mbps/2	0	2	8	3.91	0	3.473	6	3.97
Gen 80 Mbps/3	0	2	6	3.91	0	2.976	5	3.85
Gen 80 Mbps/4	0	1	7	3.91	0	3.298	7	3.97
Gen 80 Mbps/5	0	2	6	3.91	0	3.024	6	3.97
Gen 80 Mbps/6	0	2	8	3.91	0	1.719	5	3.85
Gen 80 Mbps/7	0	1	7	3.91	0	1.99	7	3.97
Gen 80 Mbps/8	0	2	7	3.91	0	1.017	6	3.97
Gen 80 Mbps/9	0	2	5	3.91	0	1.35	6	3.97
Gen 80 Mbps/10	0	2	7	3.91	0	1.816	7	3.97
Gen 90 Mbps/1	1	2	8	3.91	0.4	0.49	7	3.86
Gen 90 Mbps/2	0	2	8	3.91	0.6	0.209	7	3.86
Gen 90 Mbps/3	0	2	5	3.91	0.6	0.36	8	3.90
Gen 90 Mbps/4	0	2	7	3.91	0.3	1.488	9	3.95
Gen 90 Mbps/5	0	2	5	3.91	0	1.513	8	3.90
Gen 90 Mbps/6	0	2	5	3.91	0.4	2.086	6	3.86
Gen 90 Mbps/7	0	2	7	3.91	0.4	2.548	8	3.95
Gen 90 Mbps/8	0	2	8	3.91	0.4	1.748	9	3.95
Gen 90 Mbps/9	0	2	8	3.91	0	2.025	8	3.90
Gen 90 Mbps/10	0	2	8	3.91	0.2	2.06	7	3.86
Gen 100 Mbps/1	2	2	8	3.91	1.6	3.578	9	3.82
Gen 100 Mbps/2	2	2	11	3.91	1.8	3.35	8	3.95
Gen 100 Mbps/3	2	2	7	3.91	1.4	1.317	9	3.82
Gen 100 Mbps/4	2	1	11	3.91	1.2	1.993	8	3.95
Gen 100 Mbps/5	2	2	11	3.91	1.6	1.859	10	3.82
Gen 100 Mbps/6	1	2	8	3.91	1.4	3.276	8	3.95
Gen 100 Mbps/7	0	2	8	3.91	1.2	1.719	8	3.95
Gen 100 Mbps/8	0	2	8	3.91	1.4	2.975	9	3.84
Gen 100 Mbps/9	2	2	8	3.91	0.8	3.44	8	3.95
Gen 100 Mbps/10	1	2	8	3.91	1.4	1.719	8	3.95

ตารางที่ 4.3 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC GSM

จำนวน Background traffic	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 30 Mbps/1	0	3	3	3.57	0	0	3	3.57
Gen 30 Mbps/2	0	3	3	3.57	0	0	2	3.57
Gen 30 Mbps/3	0	3	2	3.57	0	0	3	3.57
Gen 30 Mbps/4	0	3	3	3.57	0	0	3	3.57
Gen 30 Mbps/5	0	3	4	3.57	0	0	4	3.57
Gen 30 Mbps/6	0	3	5	3.57	0	0	4	3.57
Gen 30 Mbps/7	0	5	3	3.57	0	0	4	3.57
Gen 30 Mbps/8	0	5	3	3.57	0	0	3	3.57
Gen 30 Mbps/9	0	5	2	3.57	0	0	4	3.57
Gen 30 Mbps/10	0	4	4	3.57	0	0	4	3.57
Gen 50 Mbps/1	0	3	6	3.57	0	0.916	5	3.56
Gen 50 Mbps/2	0	3	6	3.57	0	0.377	6	3.56
Gen 50 Mbps/3	0	4	5	3.57	0	0.299	5	3.56
Gen 50 Mbps/4	0	4	5	3.57	0	0.858	6	3.56
Gen 50 Mbps/5	0	3	8	3.75	0	0.409	5	3.56
Gen 50 Mbps/6	0	4	8	3.57	0	0.831	6	3.56
Gen 50 Mbps/7	0	3	5	3.57	0	0.277	5	3.56
Gen 50 Mbps/8	0	3	8	3.57	0	0.409	6	3.56
Gen 50 Mbps/9	0	4	8	3.57	0	0.858	5	3.56
Gen 50 Mbps/10	0	4	8	3.57	0	0.916	7	3.56
Gen 80 Mbps/1	0	3	6	3.53	0	2.398	5	3.56
Gen 80 Mbps/2	0	3	6	3.53	0	1.623	6	3.56
Gen 80 Mbps/3	0	3	6	3.54	0	1.224	7	3.56
Gen 80 Mbps/4	0	4	7	3.54	0	1.623	6	3.56
Gen 80 Mbps/5	0	4	5	3.53	0.2	2.188	6	3.54
Gen 80 Mbps/6	0	4	6	3.53	0	2.188	6	3.56
Gen 80 Mbps/7	1	3	7	3.53	0.4	2.253	6	3.51
Gen 80 Mbps/8	0	3	6	3.53	0	2.656	5	3.56
Gen 80 Mbps/9	0	3	7	3.53	0	1.988	7	3.56
Gen 80 Mbps/10	0	3	7	3.53	0.2	1.224	6	3.54
Gen 90 Mbps/1	0	4	8	3.53	0.1	2.231	9	3.54
Gen 90 Mbps/2	0	4	11	3.44	0.4	2.275	9	3.51
Gen 90 Mbps/3	0	4	11	3.43	0.4	2.761	10	3.5
Gen 90 Mbps/4	0	4	11	3.44	0.2	3.351	11	3.52
Gen 90 Mbps/5	0	4	8	3.63	0.3	3.586	8	3.52
Gen 90 Mbps/6	0	5	11	3.44	0.2	3.338	7	3.53
Gen 90 Mbps/7	0	3	7	3.53	0.2	2.519	10	3.53
Gen 90 Mbps/8	0	4	8	3.53	0.4	2.761	8	3.51
Gen 90 Mbps/9	0	3	8	3.43	0.2	3.825	8	3.53

ตารางที่ 4.3 เปรียบเทียบโปรแกรมที่พัฒนาขึ้นมากับเครื่องมือวัด JDSU โดยใช้ CODEC GSM (ต่อ)

จำนวน	JDSU - Handheld (HST-3000)				DQSMP			
	loss(%)	jitter(ms)	delay(ms)	MOS	loss(%)	jitter(ms)	delay(ms)	MOS
Gen 90 Mbps/10	0	3	11	3.53	0.1	2.687	9	3.54
Gen 100 Mbps/1	1	3	9	3.21	0.72	2.158	8	3.37
Gen 100 Mbps/2	1	3	9	3.21	0.53	3.948	12	3.38
Gen 100 Mbps/3	1	4	6	3.25	0.68	3.253	11	3.37
Gen 100 Mbps/4	2	4	7	3.25	0.42	2.68	8	3.41
Gen 100 Mbps/5	2	3	10	3.16	0.6	3.524	8	3.38
Gen 100 Mbps/6	2	3	10	3.16	0.98	4.21	10	3.33
Gen 100 Mbps/7	1	4	8	3.44	0.88	4.12	10	3.34
Gen 100 Mbps/8	1	4	12	3.44	0.98	4.12	8	3.24
Gen 100 Mbps/9	2	3	6	3.44	0.58	3.532	8	3.38
Gen 100 Mbps/10	2	3	12	3.35	0.68	3.332	11	3.37

จากนั้นนำค่าที่วัดในตาราง 4.1 ถึง 4.3 ได้มาหาค่าเฉลี่ยและค่าของส่วนเบี่ยงเบนมาตรฐานซึ่งจะได้แสดงในตารางที่ 4.4 ถึง 4.9 ตาม CODEC G.711, G.729 และ GSM ตามลำดับ

ตารางที่ 4.4 เปรียบเทียบค่าเฉลี่ย (Average) ของตัวแปรต่าง ๆ โดยใช้ CODEC G.711

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Gen 30 Mbps/1	0	0	4.7	5	2.4	0.0153	4.065	4.097
Gen 50 Mbps/2	0	0	7	6.2	2.1	0.1631	4.102	4.09
Gen 80 Mbps/3	0	0.483	10.7	8.2	2.7	1.2577	4.078	4.151
Gen 90 Mbps/4	0	0.765	10.6	10.2	3.1	2.3312	4.125	4.123
Gen 100 Mbps/5	1.7	1.71	9.5	10.9	2.7	2.2371	3.906	3.939

ตารางที่ 4.5 เปรียบเทียบค่าเบี่ยงเบนมาตรฐาน (Stdev) ของตัวแปรต่าง ๆ โดยใช้ CODEC G.711

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Gen 30 Mbps/1	0	0	1.4944	1.943650	0.69920	0.048032	0.038369	0.004830
Gen 50 Mbps/2	0	0	1.8856	0.788810	0.316227	0.513310	0.053082	0
Gen 80 Mbps/3	0	0.333301	4.5958	2.699794	0.483045	0.500936	0.078570	0.027264
Gen 90 Mbps/4	0	0.517284	4.1419	2.485513	0.737864	1.127731	0.051261	0.044234
Gen 100 Mbps/5	0.82327	0.347850	2.2730	0.994428	0.483045	0.96846	0.089839	0.034785

ตารางที่ 4.6 เปรียบเทียบค่าเฉลี่ย (Average) ของตัวแปรต่าง ๆ โดยใช้ CODEC G.729

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Background traffic								
Gen 30 Mbps/1	0	0	3.4	2.4	1.3	0.1058	3.91	3.906
Gen 50 Mbps/2	0	0	5.1	3.6	1.6	1.0296	3.91	3.9
Gen 80 Mbps/3	0	0	6.6	6	1.8	2.3921	3.91	3.934
Gen 90 Mbps/4	0.1	0.33	6.9	7.7	2	1.4527	3.91	3.899
Gen 100 Mbps/5	1.4	1.38	8.8	8.5	1.9	2.5226	3.91	3.9

ตารางที่ 4.7 เปรียบเทียบค่าเบี่ยงเบนมาตรฐาน (Stdev) ของตัวแปรต่าง ๆ โดยใช้ CODEC G.729

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Background traffic								
Gen 30 Mbps/1	0	0	0.699205	0.516397	0.483045	0.334568	0	0.007163
Gen 50 Mbps/2	0	0	1.100504	0.843274	0.69920	0.909949	0	0
Gen 80 Mbps/3	0	0	1.074967	0.816496	0.421637	0.906919	0	0.057965
Gen 90 Mbps/4	0.316227	0.211081	1.370320	0.948683	0	0.820348	0	0.039285
Gen 100 Mbps/5	0.843274	0.274064	1.549193	0.707106	0.316227	0.874211	0	0.064807

ตารางที่ 4.8 เปรียบเทียบค่าเฉลี่ย (Average) ของตัวแปรต่าง ๆ โดยใช้ CODEC GSM

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Background traffic								
Gen 30 Mbps/1	0	0	3.2	3.4	3.7	0	3.57	3.57
Gen 50 Mbps/2	0	0	6.7	5.6	3.5	0.615	3.57	3.56
Gen 80 Mbps/3	0.1	0.08	6.3	6	3.3	1.9365	3.532	3.551
Gen 90 Mbps/4	0	0.25	9.4	8.9	3.8	2.9334	3.493	3.523
Gen 100 Mbps/5	1.5	0.705	8.9	9.4	3.4	3.4877	3.291	3.357

ตารางที่ 4.9 เปรียบเทียบค่าเบี่ยงเบนมาตรฐาน (Stdev) ของตัวแปรต่าง ๆ โดยใช้ CODEC GSM

จำนวน	loss(%)		delay(ms)		jitter(ms)		MOS	
	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP	JDSU	DQSMP
Background traffic								
Gen 30 Mbps/1	0	0	0.918936	0.699205	0.948683	0	0	0
Gen 50 Mbps/2	0	0	1.418136	0.699205	0.527046	0.279199	0	0
Gen 80 Mbps/3	0.316227	0.139841	0.674948	0.666666	0.483045	0.491242	0.004216	0.016633
Gen 90 Mbps/4	0	0.117851	1.712697	1.197219	0.632455	0.555544	0.067173	0.013374
Gen 100 Mbps/5	0.527046	0.189106	2.183269	1.577621	0.516397	0.669683	0.115897	0.046678

จากการทดลองผลของการวัดของเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นมานั้น จะเห็นว่าผลที่ได้มีค่าใกล้เคียงกัน แต่เพื่อให้การทดสอบโปรแกรมที่พัฒนาขึ้นนี้ มีความถูกต้องทางสถิติมากขึ้น ผู้วิจัยจึงใช้การทดสอบสมมติฐานเพื่อทดสอบโปรแกรมที่พัฒนาขึ้นมา เพื่อให้ได้เป็นไปตามมาตรฐานงานวิจัย

4.2 การทดสอบสมมติฐาน (Tests of Hypothesis)

จากผลการทดสอบโดยการวัดของโปรแกรมที่พัฒนาขึ้นมา เปรียบเทียบกับเครื่องมือวัด JDSU ว่าจะสามารถยอมรับได้หรือไม่ แล้ไหนนั้น ก็โดยการใช้วิธีการทดสอบสมมติฐาน ซึ่งจะมีวิธีการคำนวณโดยนำผลที่ได้จากการวัดของโปรแกรมที่พัฒนาขึ้นและเครื่องมือวัด JDSU มาเปรียบเทียบกัน ซึ่งจะแสดงวิธีการคำนวณโดยจะอ้างอิงทฤษฎีจากบทที่ 2 พอเป็นตัวอย่างดังนี้

ตัวอย่าง 1 ทดสอบสมมติฐานที่ CODEC G.711 ที่ Generate Traffic 90 mbps จากการทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.4 กับ 4.5 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

DQSMP

$n_1 = 10$

$n_2 = 10$

$\bar{x}_1 = 4.125$

$\bar{x}_2 = 4.123$

$\sigma_1 = 0.051$

$\sigma_2 = 0.44$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$H_0 : \mu_1 - \mu_2 = 0$

VS

$H_1 : \mu_1 - \mu_2 \neq 0$

หรือ

$H_0 : \mu_1 = \mu_2$

VS

$H_1 : \mu_1 \neq \mu_2$

2) คำนวณค่าสถิติที่ใช้ทดสอบ

$$\begin{aligned} \bar{x}_1 &= \sum_{i=1}^{n_1} X_{li} \\ &= 4.125 \end{aligned}$$

$$\begin{aligned}\bar{X}_2 &= \sum_{i=1}^{n_2} X_{2i} \\ &= 4.123\end{aligned}$$

$$\begin{aligned}\text{หาค่า } Z &= \frac{(x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)}}{0.002 / 0.02} \\ &= 0.1\end{aligned}$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = 0.1 < -1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

ตัวอย่าง 2 ทดสอบสมมติฐานที่ CODEC G.711 ที่ Generate Traffic 100 mbps จากการทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.4 กับ 4.5 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

$$n_1 = 10$$

$$\bar{x}_1 = 3.906$$

$$\sigma_1 = 0.089$$

Developed Program

$$n_2 = 10$$

$$\bar{x}_2 = 3.939$$

$$\sigma_2 = 0.034$$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$$H_0 : \mu_1 - \mu_2 = 0$$

VS

$$H_1 : \mu_1 - \mu_2 \neq 0$$

หรือ

$$H_1 : \mu_1 = \mu_2$$

VS

$$H_1 : \mu_1 \neq \mu_2$$

2) คำนวณค่าสถิติที่ใช้ทดสอบ

$$\begin{aligned}\bar{x}_1 &= \sum_{i=1}^{n_1} X_{1i} \\ &= 3.906\end{aligned}$$

$$\begin{aligned}\bar{x}_2 &= \sum_{i=1}^{n_2} X_{2i} \\ &= 3.939\end{aligned}$$

$$\begin{aligned}\text{หาค่า } Z &= (x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)} \\ &= -0.033 / 0.03 \\ &= -1.09\end{aligned}$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = -1.09 > -1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

ตัวอย่าง 3 ทดสอบสมมติฐานที่ CODEC G.729 ที่ Generate Traffic 30 mbps จากการศึกษาทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.6 กับ 4.7 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

$$n_1 = 10$$

$$\bar{x}_1 = 3.91$$

$$\sigma_1 = 0$$

Developed Program

$$n_2 = 10$$

$$\bar{x}_2 = 3.906$$

$$\sigma_2 = 0.007$$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$$H_0 : \mu_1 - \mu_2 = 0 \quad \text{VS} \quad H_1 : \mu_1 - \mu_2 \neq 0$$

หรือ $H_1 : \mu_1 = \mu_2 \quad \text{VS} \quad H_1 : \mu_1 \neq \mu_2$

2) คำนวณค่าสถิติที่ใช้ทดสอบ

$$\bar{x}_1 = \sum_{i=1}^{n_1} X_{1i}$$

$$= 3.91$$

$$\bar{x}_2 = \sum_{i=1}^{n_2} X_{2i}$$

$$= 3.906$$

$$\text{หาค่า } Z = (x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)}$$

$$= 0.004 / 0.0022$$

$$= 1.8$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = 1.8 < 1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

ตัวอย่าง 4 ทดสอบสมมติฐานที่ CODEC G.729 ที่ Generate Traffic 80 mbps จากการทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.6 กับ 4.7 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

$$n_1 = 10$$

$$\bar{x}_1 = 3.91$$

$$\sigma_1 = 0$$

Developed Program

$$n_2 = 10$$

$$\bar{x}_2 = 3.934$$

$$\sigma_2 = 0.046$$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$$H_0 : \mu_1 - \mu_2 = 0 \quad \text{VS} \quad H_1 : \mu_1 - \mu_2 \neq 0$$

หรือ $H_1 : \mu_1 = \mu_2 \quad \text{VS} \quad H_1 : \mu_1 \neq \mu_2$

2) กำหนดค่าสถิติที่ใช้ทดสอบ

$$\begin{aligned} \bar{X}_1 &= \sum_{i=1}^{n_1} X_{1i} \\ &= 3.91 \end{aligned}$$

$$\begin{aligned} \bar{X}_2 &= \sum_{i=1}^{n_2} X_{2i} \\ &= 3.934 \end{aligned}$$

$$\begin{aligned} \text{หาค่า } Z &= (x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)} \\ &= -0.024 / 0.018 \\ &= -1.331 \end{aligned}$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = -1.33 > -1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

ตัวอย่าง 5 ทดสอบสมมติฐานที่ CODEC GSM ที่ Generate Traffic 90 mbps จากการศึกษาทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.8 กับ 4.9 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

Developed Program

$$n_1 = 10$$

$$n_2 = 10$$

$$\bar{x}_1 = 3.493$$

$$\bar{x}_2 = 3.523$$

$$\sigma_1 = 0.067$$

$$\sigma_2 = 0.013$$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$$H_0 : \mu_1 - \mu_2 = 0$$

VS

$$H_1 : \mu_1 - \mu_2 \neq 0$$

หรือ

$$H_1 : \mu_1 = \mu_2$$

VS

$$H_1 : \mu_1 \neq \mu_2$$

2) คำนวณค่าสถิติที่ใช้ทดสอบ

$$\begin{aligned} \bar{x}_1 &= \sum_{i=1}^{n_1} X_{1i} \\ &= 3.493 \end{aligned}$$

$$\begin{aligned} \bar{x}_2 &= \sum_{i=1}^{n_2} X_{2i} \\ &= 3.523 \end{aligned}$$

$$\begin{aligned} \text{หาค่า } Z &= (x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)} \\ &= -0.03 / 0.021 \\ &= -1.39 \end{aligned}$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = -1.39 > -1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

ตัวอย่าง 6 ทดสอบสมมติฐานที่ CODEC GSM ที่ Generate Traffic 100 mbps จากการศึกษาทดสอบวัดเครื่องมือวัด JDSU กับโปรแกรมที่พัฒนาขึ้นที่แสดงในตารางที่ 4.8 กับ 4.9 โดยเราเชื่อมั่นว่าค่าที่วัดได้จากเครื่องมือวัดทั้ง 2 ไม่แตกต่างกันที่ระดับนัยสำคัญ 0.05

$H_0 : \mu_1 = \mu_2$ และมีค่า $\alpha = 0.05$ เมื่อได้ค่าจากตารางดังนี้

JDSU

$$n_1 = 10$$

$$\bar{x}_1 = 3.291$$

$$\sigma_1 = 0.115$$

Developed Program

$$n_2 = 10$$

$$\bar{x}_2 = 3.537$$

$$\sigma_2 = 0.046$$

ให้ μ_1 แทน ค่าเฉลี่ยที่วัดได้ของ JDSU

ให้ μ_2 แทน ค่าเฉลี่ยที่วัดได้ของ Developed Program

1) ตั้งสมมติฐานการวิจัย

$$H_0 : \mu_1 - \mu_2 = 0 \quad \text{VS} \quad H_1 : \mu_1 - \mu_2 \neq 0$$

หรือ $H_1 : \mu_1 = \mu_2 \quad \text{VS} \quad H_1 : \mu_1 \neq \mu_2$

2) คำนวณค่าสถิติที่ใช้ทดสอบ

$$\bar{x}_1 = \sum_{i=1}^{n_1} x_{1i}$$

$$= 3.29$$

$$\bar{x}_2 = \sum_{i=1}^{n_2} x_{2i}$$

$$= 3.35$$

$$\text{หาค่า } Z = (x_1 - x_2) / \sqrt{(\sigma_1^2 / n_1) + (\sigma_2^2 / n_2)}$$

$$= -0.066 / 0.039$$

$$= -1.68$$

3) กำหนดระดับนัยสำคัญ $\alpha = 0.05$

4) กำหนดอาณาเขตวิกฤต

จะปฏิเสธ H_0 ถ้า $Z > Z_{0.025} = 1.96$ หรือ $Z < -Z_{0.025} = -1.96$

5) การสรุปผล

เนื่องจาก $Z = -1.68 > -1.96$ ไม่ตกอยู่ในอาณาเขตวิกฤต จึงยอมรับ H_0

จากผลการคำนวณข้างต้นนั้นเป็นยกตัวอย่างการคำนวณบางค่าของการ Generate background traffic ซึ่งถ้าเป็นทั้งหมดของทุกการ Generate background traffic ทุกค่าตั้งแต่ 30, 50, 80, 90, 100 Mbps จะแสดงให้เห็นในตารางที่ 4.10 ซึ่งจะเป็นตารางผลการคำนวณในการหาค่าการทดสอบสมมติฐานของโปรแกรมที่พัฒนาขึ้น

ตารางที่ 4.10 สรุปผลการคำนวณในการหาค่าการทดสอบสมมติฐานของโปรแกรมที่พัฒนา

	G.711	G.729	GSM
ที่ 30 Mbps	-2.64 (ปฏิเสธ H_0)	1.8 (ยอมรับ H_0)	หาค่าไม่ได้
ที่ 50 Mbps	0.715 (ยอมรับ H_0)	หาค่าไม่ได้	หาค่าไม่ได้
ที่ 80 Mbps	-2.79 (ปฏิเสธ H_0)	-1.33 (ยอมรับ H_0)	-3.64 (ปฏิเสธ H_0)
ที่ 90 Mbps	0.1 (ยอมรับ H_0)	0.08 (ยอมรับ H_0)	-1.39 (ยอมรับ H_0)
ที่ 100 Mbps	-1.09 (ยอมรับ H_0)	0.494 (ยอมรับ H_0)	-1.68 (ยอมรับ H_0)

จากตารางที่ 4.10 ถ้าเรามองภาพรวมจะเห็นว่าส่วนใหญ่จะยอมรับกับการทดสอบสมมติฐานของโปรแกรมที่พัฒนาขึ้น ที่ระดับนัยสำคัญ 0.05 โดยที่ CODEC G.729 จะสังเกตได้ว่าการทดสอบสมมติฐาน สามารถยอมรับได้เกือบทุกการ Generate background traffic ยกเว้นที่ Generate background traffic 50 Mbps ที่ไม่สามารถหาค่าได้ก็เพราะว่ามีค่าเบี่ยงเบนมาตรฐานเท่ากับศูนย์ ซึ่งค่านี้จะเป็นตัวหารในสูตรทำให้ไม่สามารถหาค่าออกมาได้

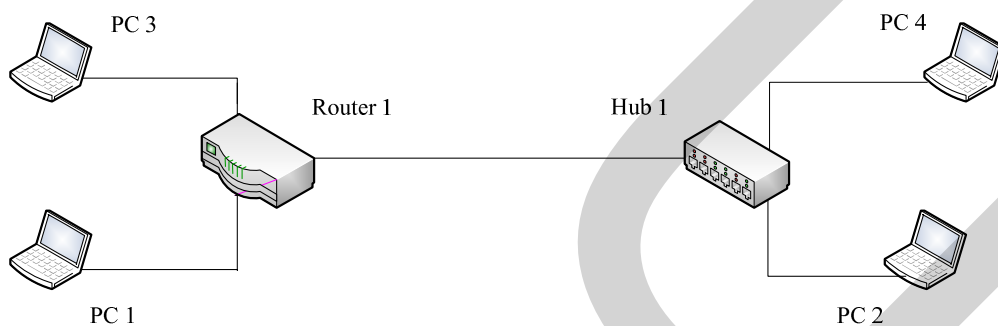
โดยที่ CODEC G.711 การทดสอบสมมติฐานจะมีการปฏิเสธ H_0 ที่ระดับนัยสำคัญ 95% อยู่สองกรณีคือที่การ Generate background traffic 30 Mbps และ 80 Mbps แต่จะสังเกตว่าค่าที่คำนวณได้ก็ไม่ห่างจากค่าความเชื่อมั่นที่นัยสำคัญ 95% (-1.96 ถึง 1.96) มากนัก คือ ที่การ Generate background traffic 30 Mbps จะคำนวณได้ค่า -2.64 ในขณะที่เวลาที่การ Generate background traffic 80 Mbps ก็จะคำนวณได้ -2.79 นอกนั้นก็ค่า Generate background traffic อื่นๆ ก็สามารถยอมรับ H_0 ได้ทั้งหมด

ส่วนที่ CODEC GSM ก็จะเป็นลักษณะเช่นเดียวกันซึ่งส่วนใหญ่ก็สามารถยอมรับการทดสอบสมมติฐานของโปรแกรมที่พัฒนาขึ้น ที่ระดับนัยสำคัญ 95% ยกเว้นที่ Generate background traffic 30 Mbps กับ 50 Mbps ที่ไม่สามารถหาค่าได้ก็เพราะว่ามีค่าเบี่ยงเบนมาตรฐานเท่ากับศูนย์ ซึ่งค่านี้จะเป็นตัวหารในสูตรทำให้ไม่สามารถหาค่าออกมาได้

ซึ่งจะถ้าคิดเป็นเปอร์เซ็นต์แล้วค่าการยอมรับ H_0 จะมีค่า 60 เปอร์เซ็นต์ ในขณะที่การปฏิเสธ H_0 มีค่าเพียง 20 เปอร์เซ็นต์ ส่วนอีก 20 เปอร์เซ็นต์จะเป็นส่วนของการหาค่าไม่ได้ ซึ่งในส่วนนี้เกิดจากค่าข้อมูลที่วัดได้ไม่มีการเบี่ยงเบน

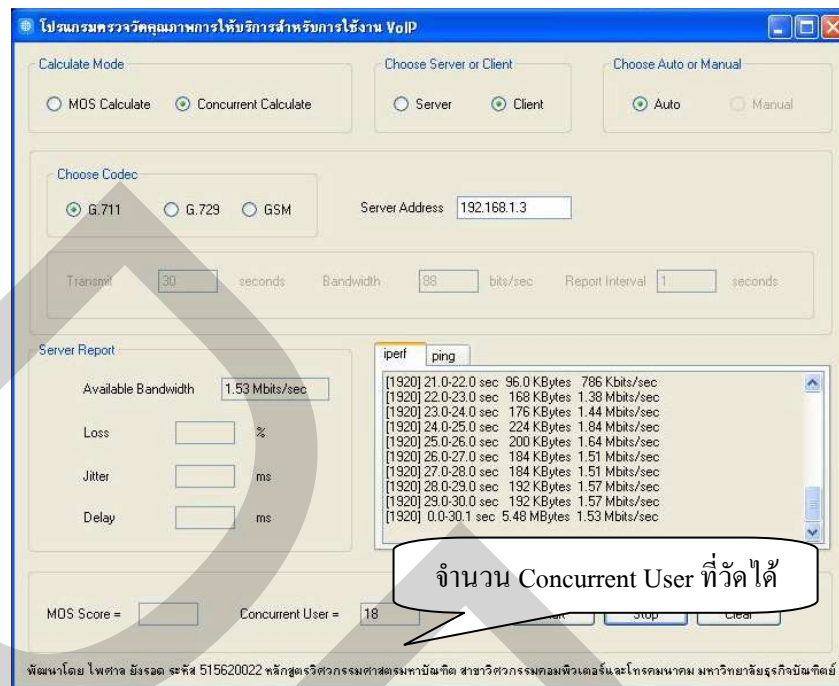
4.3 วิธีการทดสอบโปรแกรมโดยการวัดค่า Concurrent User

การทดสอบหาค่า Concurrent Use ก็เช่นเดียวกันกับการทดสอบการหาค่า MOS โดยได้ทำการจำลองเครือข่ายขึ้นมาตามรูปที่ 4.3 แล้วใช้โปรแกรมที่พัฒนาขึ้นวัดหาค่า Concurrent Use ก็จะสามารถทราบแบนด์วิดท์ที่เหลือ และ จำนวนผู้ใช้งานพร้อมกันในเครือข่ายนั้นแล้ว ก็จะใช้ PC 3 และ PC 4 ในการสร้าง background traffic โดยใช้โปรแกรม Iperf ซึ่งเป็นโปรแกรมฟรี เข้าไปในเครือข่าย โดยที่ปริมาณ traffic ที่สร้างขึ้นในเครือข่ายนั้นจะทำการสร้างแบนด์วิดท์ให้เต็มเครือข่ายแล้วใช้โปรแกรมที่พัฒนาขึ้นวัดค่า MOS ที่เกิดขึ้นของ CODEC G.711 เพราะมีแบนด์วิดท์มากกว่า CODEC อื่นๆ เพื่อจะได้ทราบคุณภาพของ call ที่ใช้งานในขณะนั้น โดย PC 1 และ PC 2 ใช้ในการทดสอบโปรแกรมที่พัฒนาขึ้น โดยจะทำหน้าที่เป็น Server และ Client ตามลำดับ ส่วน PC 3 และ PC 4 ใช้ในการสร้าง background traffic ซึ่งจะมีทั้ง Server และ Client เช่นกันแต่จะใช้ Hub (ตาม spec จะรองรับแบนด์วิดท์ได้ 10 Mbps) แทน Switch



รูปที่ 4.3 ลักษณะการใช้โปรแกรมที่พัฒนาวัดค่า Concurrent User จากเครือข่ายจำลอง

จากการทดสอบตามรูปที่ 4.3 จะมีการทดสอบใน 2 ลักษณะคือ การทดสอบในลักษณะที่ไม่มี background traffic และ การทดสอบแบบมี background traffic เต็มเครือข่ายตามที่อุปกรณ์สามารถรองรับได้ (ตาม spec) ซึ่งจากการวัดค่า Concurrent Use ที่ใช้งานระหว่าง Router กับ Hub จะได้เท่ากับ 18 Concurrent User ตามรูปที่ 4.4 ที่จะแสดงต่อไป



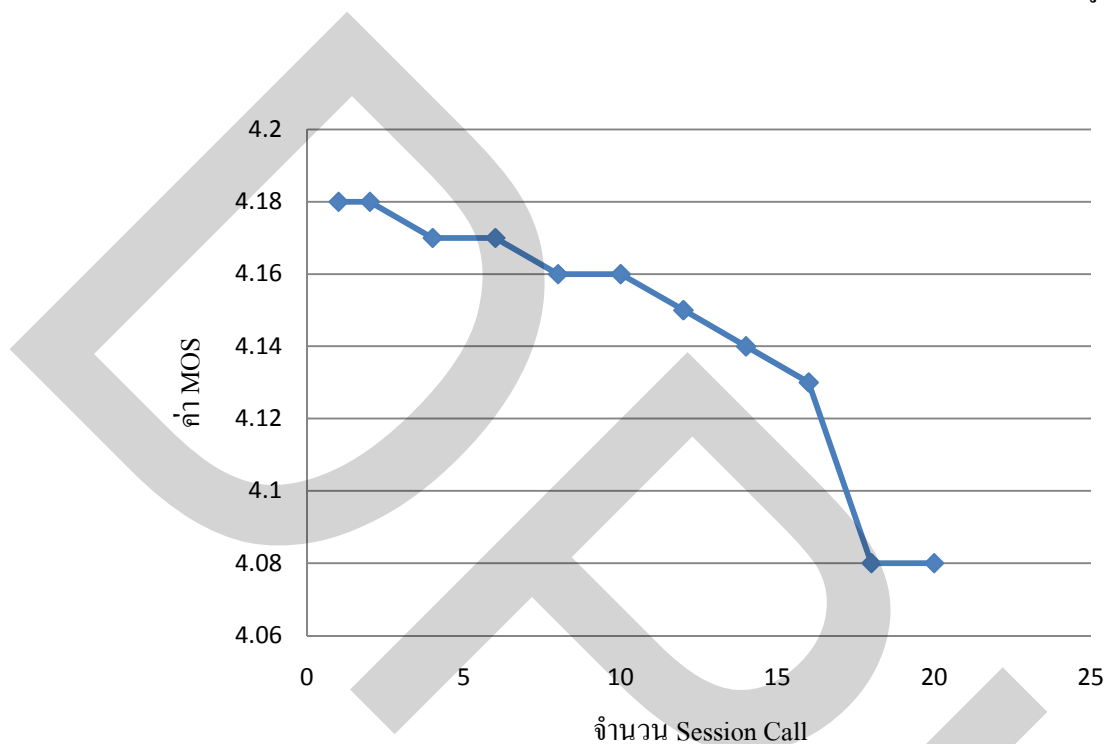
รูปที่ 4.4 โปรแกรมแสดงจำนวน Concurrent User ที่วัดได้ในเครือข่าย

4.2.1 การทดสอบในลักษณะที่ไม่มี background traffic โดยได้นำโปรแกรมที่พัฒนาขึ้นมาสร้างทำการสร้าง Session Call ตั้งแต่ 1, 2, 4, 8 ถึง 20 Call ตามลำดับ โดยแต่ละ Session Call จะทำการทดสอบจำนวน 10 ครั้ง แล้วนำมาหาค่าเฉลี่ย ก่อนที่จะบันทึกข้อมูลลงในแต่ละช่อง ซึ่งจะได้เห็นค่า MOS ที่เปลี่ยนแปลงไปในตารางที่ 4.11

ตารางที่ 4.11 ค่า MOS เมื่อจำนวน Call Session เพิ่มขึ้นในเครือข่ายที่ไม่มี background traffic

	Loss (%)	Jitter (ms)	Delay (ms)	MOS
1 Session Call (88 Kbps)	0.074	3.481	10	4.18
2 Session Call (176 Kbps)	0.22	0.808	3	4.18
4 Session Call (352 Kbps)	0	15157	1	4.17
6 Session Call (528 Kbps)	0.075	3.028	24	4.17
8 Session Call (704 Kbps)	0.11	11.117	17	4.16
10 Session Call (880 Kbps)	0.44	1.436	7	4.16
12 Session Call (1056 Kbps)	0.13	15.513	13	4.15
14 Session Call (1232 Kbps)	0.22	15.152	18	4.14
16 Session Call (1408 Kbps)	0.23	13.464	38	4.13
18 Session Call (1584 Kbps)	0.65	18.515	34	4.08
20 Session Call (1760 Kbps)	0.73	12.889	38	4.08

จากตารางที่ 4.11 ผลที่ได้จากการวัดในกรณีที่ไม่มี background traffic สามารถนำมาเขียนกราฟ เพื่อที่จะให้เห็นภาพได้ดีขึ้น โดยจะเห็นว่าเมื่อจำนวน Session Call ช่วงที่เครือข่ายไม่สามารถรองรับได้คือที่ 18 Calls ขึ้นไป ค่า MOS ที่วัดได้ จะลดลงอย่างมาก ดังแสดงในรูปที่ 4.5



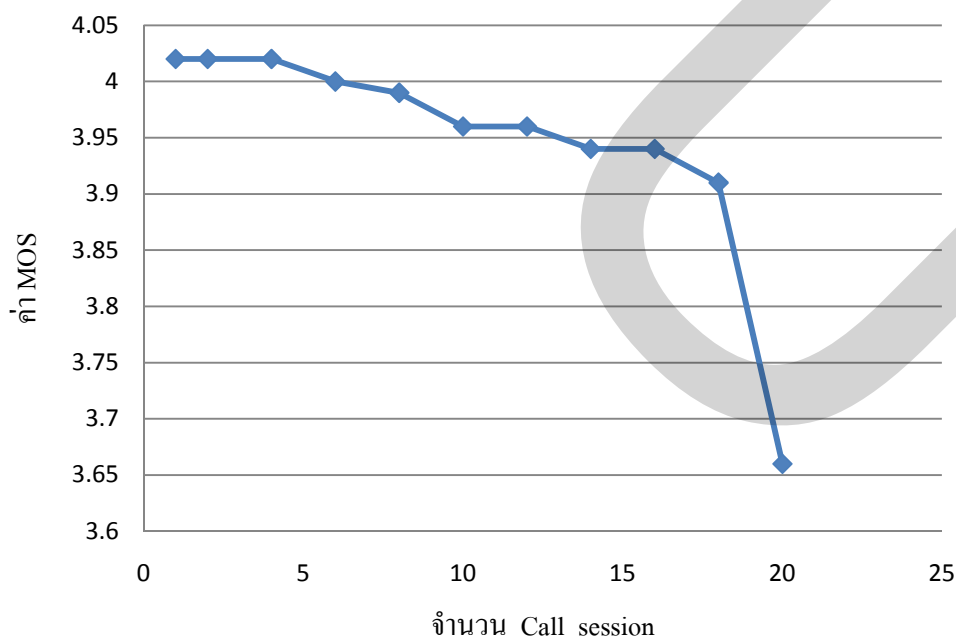
รูปที่ 4.5 ค่า MOS เมื่อมีจำนวน Session Call เพิ่มขึ้นกรณีที่ไม่มี background traffic

4.2.2 การทดสอบในลักษณะที่มี background traffic จากการทดสอบตามรูปที่ 4.3 โดยจะทำการสร้าง background traffic ระหว่าง Router กับ Hub ให้เท่ากับแบนด์วิธของเครือข่ายที่อุปกรณ์สามารถรองรับได้คือ 10 Mbps และได้้นำโปรแกรมที่พัฒนาขึ้นมาสร้าง Session Call โดยแต่ละ Session Call จะทำการทดสอบจำนวน 10 ครั้ง แล้วนำมาหาค่าเฉลี่ย ก่อนที่จะบันทึกข้อมูลลงในแต่ละช่อง ตั้งแต่ 1, 2, 4, 8 ถึง 20 Calls ตามลำดับ ซึ่งจะได้เห็นค่า MOS ที่เปลี่ยนแปลงไปในตารางที่ 4.12

ตารางที่ 4.12 ค่า MOS เมื่อจำนวน Call Session เพิ่มขึ้นในเครือข่ายที่มี background traffic 100%

	Loss (%)	Jitter (ms)	Delay (ms)	MOS
1 Session Call (88 Kbps)	0.12	19.036	157	4.02
2 Session Call (176 Kbps)	0.21	25.036	136	4.02
4 Session Call (352 Kbps)	0.34	16.386	137	4.02
6 Session Call (528 Kbps)	0.46	39.353	100	4.00
8 Session Call (704 Kbps)	0.4	19.245	160	3.99
10 Session Call (880 Kbps)	0.24	45.273	158	3.96
12 Session Call (1056 Kbps)	1.2	2.826	138	3.96
14 Session Call (1232 Kbps)	0.44	74.513	91	3.94
16 Session Call (1408 Kbps)	1	16.514	154	3.94
18 Session Call (1584 Kbps)	1.5	9.388	145	3.91
20 Session Call (1760 Kbps)	3.3	41.240	132	3.66

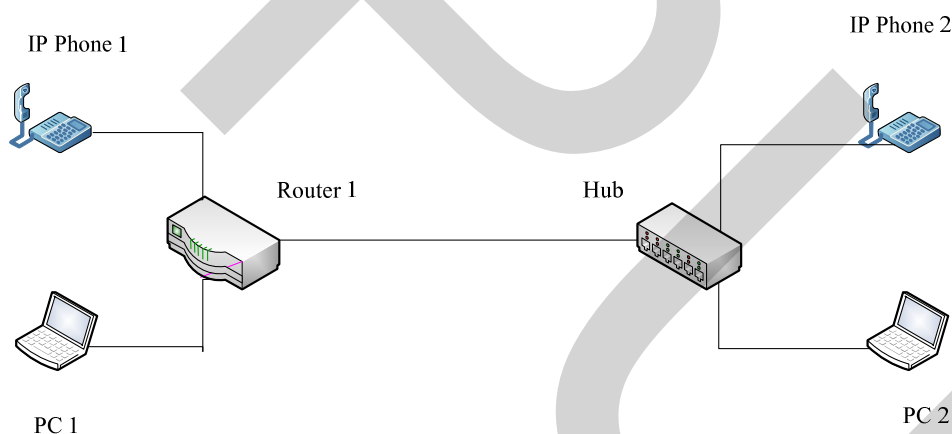
จากตารางที่ 4.12 ผลที่ได้จากการวัดในกรณีที่มี background traffic สามารถนำมาเขียนกราฟ เพื่อที่จะให้เห็นภาพได้ดีขึ้น โดยค่า MOS ที่เริ่มต้นที่ Session Call มีจำนวนน้อย จะสังเกตว่าค่า MOS จะมีค่าน้อยกว่า ผลที่ได้จากการวัดช่วงที่ไม่มี background traffic ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 ค่า MOS เมื่อมีจำนวน Session Call เพิ่มขึ้นกรณีที่มี background traffic 100%

จากการทดสอบ จะเห็นว่า แม้จะมี background traffic จนเต็มในเครือข่าย ก็มีได้หมายความว่า เมื่อเราใช้งาน VoIP ในเครือข่ายนั้นแล้วจะใช้งานไม่ได้ แต่พบว่ายังสามารถใช้งานได้ดี ในกรณีที่มี จำนวน Session call น้อย ๆ เหตุก็เพราะว่าในอุปกรณ์ Router และ Hub จะมี Buffer อยู่จำนวนหนึ่ง ซึ่งสามารถรองรับ Bandwidth ที่เกินจากที่วัดได้จากโปรแกรมที่ผู้วิจัยพัฒนาขึ้นตรงนี้ได้ แต่เมื่อเพิ่มจำนวน Session call มากขึ้นจะสังเกตเห็นว่าค่า Loss, Delay, Jitter ก็จะเพิ่มขึ้นตามก็ทำให้ค่า MOS ที่ได้ค่อย ๆ ลดลง ทำให้คุณภาพของเสียงที่ได้ค่อย ๆ ลดลงด้วย

การทดสอบอีกลักษณะหนึ่งที่สามารถทำได้ก็คือ การทำทดสอบการใช้งาน VoIP จริงๆ เข้าไปในเครือข่าย ตามรูปที่ 4.3 โดยจะใช้ IP Phone 1 และ IP Phone 2 ทำการซึ่งใน IP Phone ทั้ง 2 ตัวจะมี ฟังก์ชัน Peer to Peer เมื่อทำการเชื่อมต่อใน IP Phone ทั้ง 2 เรียบร้อยแล้วทำการโทรหากัน โดยจะมี PC 1 และ PC 2 เป็นตัว Server และ Client ในการสร้าง background traffic เข้าไปในเครือข่ายจากน้อยไปหามากจนเต็ม แล้วให้ผู้ฟังสังเกตคุณภาพของเสียงที่โทรหากันระหว่าง IP Phone 1 กับ IP Phone 2 ตามรูปที่ 4.7



รูปที่ 4.7 ลักษณะการทดสอบ Concurrent User จากเครือข่ายที่จำลอง

บทที่ 5

สรุปผลและข้อเสนอแนะ

ในบทนี้จะเป็นการอภิปรายเพื่อสรุปผลที่ได้จากการทดสอบงานวิจัย รวมทั้งข้อจำกัดของระบบที่พบจากการทดสอบระบบ และข้อเสนอแนะสำหรับแนวทางในการพัฒนางานวิจัยนี้ต่อไปเพื่อแก้ไขข้อบกพร่องของระบบให้มีประสิทธิภาพมากขึ้น

5.1 สรุปผลการวิจัย

5.1.1 สรุปผลตามวัตถุประสงค์ของงานวิจัย

5.1.1.1 ในการพัฒนาโปรแกรมตรวจวัดคุณภาพการให้บริการสำหรับการใช้งาน VoIP ตามวัตถุประสงค์ของงานวิจัย สรุปได้จากผลการทดสอบระบบ โปรแกรมสามารถใช้คำนวณหาค่า MOS ได้ใกล้เคียงกับค่าที่วัดได้โดยเครื่องมือวัด JDSU โดยหลักการพิสูจน์ทางสถิติ ทำให้โปรแกรมที่พัฒนาสามารถนำไปใช้ในองค์กรที่มีระบบเครือข่ายอยู่แล้ว แล้วต้องการใช้งาน VoIP เพิ่มเติม โดยไม่ต้องลงทุนที่สูงมากนักก็สามารถได้เครื่องมือที่สามารถวัดคุณภาพการให้บริการ VoIP ที่สามารถทำงานได้ดี

5.1.1.2 ในการทดสอบหา Concurrent user ก็สามารวัดจำนวนผู้ใช้งานพร้อมกันตามลักษณะของ CODEC แต่ละชนิดได้ จะเห็นได้จากการทดสอบเมื่อเรา Generate background traffic เข้าในเครือข่ายเต็มจำนวนของผู้ใช้งานในขณะนั้น เมื่อทำการสร้าง call เพิ่มเข้าไปที่ละ call ก็จะสังเกตได้ว่า คุณภาพของ call จะมีคุณภาพลดลง นอกจากนั้น โปรแกรมยังสามารถช่วยคำนวณหาจำนวนผู้ใช้งาน VoIP ที่ระบบสามารถรองรับได้

5.1.2 สรุปผลตามขอบเขตของงานวิจัย จากการทดสอบการทำงานต่างๆ ตามขอบเขตของระบบสามารถสรุปผลได้ดังนี้

5.1.2.1 โปรแกรมที่พัฒนาสามารถวัด ข้อมูลสูญหายในเครือข่าย (packet loss) ค่าความแปรผันของความล่าช้าทางเวลา (jitter) โดยใช้โปรแกรม Iperf ที่เป็นโปรแกรมฟรี (Freeware) มาช่วย และวัดค่าความล่าช้าทางเวลา (delay) โดยใช้ Command ping บน Windows

5.1.2.2 โปรแกรมที่พัฒนาสามารถคำนวณค่าคุณภาพการให้บริการ (MOS) ตามมาตรฐานได้ โดยใช้ผลจากการวัดในข้อ 5.1.2.1 มาคำนวณ โดยข้อมูลส่วนใหญ่มีความถูกต้องที่ระดับ 95 %

5.1.2.3 จากผลการทดสอบเบื้องต้นจากการวัดค่าความล่าช้าทางเวลา ค่าความแปรผันของความล่าช้าทางเวลา และ ค่าสูญหายของข้อมูล จะเห็นว่าค่าสูญหายของข้อมูล และค่าความแปรผันของความล่าช้าทางเวลา ที่วัดได้จากเครื่องมือวัด JDSU กับ โปรแกรมที่พัฒนา มีค่าเฉลี่ยที่วัดได้ใกล้เคียงกันมาก แต่สำหรับค่าความแปรผันของความล่าช้าทางเวลา จะเห็นว่าในช่วงที่ทำการสร้าง background traffic ระหว่าง 30 Mbps ถึง 80 Mbps จะมีค่าห่างกันพอสมควร แต่ช่วงที่มี background traffic ระหว่าง 90 Mbps ถึง 100 Mbps จะมีค่าใกล้เคียงกัน แต่เมื่อนำมาคำนวณค่า MOS ก็จะไม่ต่างกันมากนัก

5.1.2.4 โปรแกรมที่พัฒนา สามารถคำนวณผู้ใช้งานพร้อมกัน ที่เครือข่ายสามารถรองรับการใช้งาน VoIP ได้โดยโปรแกรมที่พัฒนาจะรองรับเฉพาะการใช้งาน VoIP ภายในองค์กร (LAN) เท่านั้น ซึ่งจะสามารถรองรับเฉพาะ CODEC แบบ G.711, G.729 และ GSM ได้จริง

ผลสรุปของงานวิจัยในส่วนของขอบเขตต่างๆที่ออกแบบไว้ จากการทดสอบตามขอบเขตของระบบ ระบบสามารถทำงานได้ตามขอบเขตที่ออกแบบทุกข้อ

5.2 ข้อจำกัดของระบบ

5.2.1 จากการทดสอบโปรแกรมที่พัฒนาขึ้นมากับ เครื่องมือวัด JDSU ผลที่ได้ถือว่าใกล้เคียง เพราะมีทดสอบที่ระดับความเชื่อมั่น 95 % แต่ถ้าจะให้ดีกว่านี้ควรจะมีการปรับปรุงให้ได้ค่าระดับความเชื่อมั่นที่สูงกว่า 95 %

5.2.2 การใช้งานโปรแกรมที่พัฒนาขึ้นนี้ ผู้ที่จะนำไปใช้งานควรจะมีพื้นฐานความรู้ทางด้าน VoIP อยู่บ้าง

5.2.3 โปรแกรมที่พัฒนาขึ้น ยังไม่สามารถตรวจวัดคุณภาพการให้บริการผ่านเครือข่ายที่มีการใช้งานแบบ NAT (Network Address Translation) ได้

5.3 ข้อเสนอแนะ

5.3.1 ถ้าจะมีการพัฒนาต่อน่าจะมีการพัฒนาโปรแกรมให้มีความสามารถเพิ่มขึ้น ตัวอย่างเช่น สามารถวัดคุณภาพของสื่อประเภทมีเดียที่ใช้งานบนอินเทอร์เน็ตให้ได้หลากหลายยิ่งขึ้น เช่น IP TV เป็นต้น

5.3.2 ควรจะมีการพัฒนาโปรแกรมให้สามารถวัดเครือข่ายที่มีการใช้งาน NAT (Network Address Translation) ได้ เพราะจะสามารถใช้ประโยชน์ได้มากยิ่งขึ้นในกรณีที่ถูกค่าที่เป็น Home use เมื่อเกิดปัญหาคุณภาพเสียงไม่เป็นที่พอใจ ก็จะสามารถตรวจสอบคุณภาพของเครือข่ายได้โดยตรงกับผู้ให้บริการ VoIP เพื่อจะได้ทราบปัญหาของต้นตอที่แท้จริง



บรรณานุกรม

บรรณานุกรม

ภาษาไทย

หนังสือ

กิติ ภัคดีวัฒน์กุล และจำลอง กระจูตสาหะ. (2542). **Visual Basic 6 ฉบับโปรแกรมเมอร์**. ไทยเจริญ การพิมพ์.

บทความ

คณะอนุกรรมการเฉพาะกิจจัดทำมาตรฐานทางเทคนิคสำหรับ NGN. (2550, ตุลาคม). “รายงานผลการศึกษาเกี่ยวกับมาตรฐานทางเทคนิคสำหรับ Next Generation Network (NGN).”

สำนักงานคณะกรรมการกิจการโทรคมนาคมแห่งชาติ, หน้า 2-7.

บงการ หอมนาน. (2547, เมษายน). “VoIP เทคโนโลยีสื่อสารไร้พรมแดน.” **ไมโครคอมพิวเตอร์** (22, ฉบับที่ 225). กรุงเทพฯ: ไฮเทค ฟรันทิ่ง. หน้า 121-124.

สาธิตพงษ์ พุทธิประเสริฐ, สิ้นชัย กมลภิวส์ และสัญญากร วุฒิสัทติกุล. (2544) “โปรโตคอลมาตรฐานสำหรับอินเทอร์เน็ตเทเลโฟนนี่” **NECTEC Technical Journal** (Vol.3, No.10). หน้า 69-84.

วิทยานิพนธ์

พรภัทร ภัทรจारी. (2548). **ระบบรับส่งสัญญาณเสียงผ่านเครือข่ายไอพี (บีบอัดสัญญาณเสียงหลายแบบ)**. วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ. กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

วาริน เล้าสกุล. (2544). **ระบบวิเคราะห์แบนด์วิดท์ของเครือข่ายสำหรับการส่งข้อมูลเสียงบนเครือข่ายไอพี**. วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ. กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.

อรศรี ศรีระษา. (2545). โปรแกรมช่วยออกแบบโครงข่าย VoIP. วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ. กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.

สารสนเทศจากสื่ออิเล็กทรอนิกส์

เกษมศักดิ์.(2547). โทรศัพท์บนเครือข่ายอินเทอร์เน็ต. สืบค้นเมื่อ 20 มีนาคม 2553. จาก www2.cs.science.cmu.ac.th/seminar/2547/voip/.

แฟร์รี่ (Fairy). (2552). Queuing Mechanism. สืบค้นเมื่อ 20 พฤษภาคม 2553. จาก <http://www.siamnetworker.com/board/index.php?topic=455.0>.

วัชรพงษ์. (2547). เทคนิคการใช้คำสั่ง Ping ในการตรวจสอบระบบเครือข่าย. สืบค้นเมื่อ 15 เมษายน 2553. จาก http://www.pyo.nu.ac.th/di-tec/Manuals/Ping_command.pdf.

Microsoft. (1 กรกฎาคม. 2552). ความหมายของ IP Address, IP Config และการใช้งานคำสั่ง Ping. สืบค้นเมื่อ 15 เมษายน 2553. จาก <http://www.varietypc.net/main/archives/692> .

วินโด อีที โพร (Windows IT Pro). (2542). การวัดคุณภาพเสียงจากเครือข่ายสื่อสาร. สืบค้นเมื่อ 20 มีนาคม 2553. จาก http://www.tmi.or.th/index.php?option=com_content&task=view&id=293&Itemid=64 .

สมิทวิชัย ไชยวงศ์และรังสิมา เกียรติยุทธชาติ (2550). Network management. สืบค้นเมื่อ 20 พฤษภาคม 2553. จาก <http://www.vcharkarn.com/varticle/17875>.

เอกพลชัย โพธิจักร และ โอฟาร ปาลีโพธิศักดิ์ (2548). Quality-of-Service (QoS). สืบค้นเมื่อ 20 เมษายน 2553. จาก <http://cpe.kmutt.ac.th/previousproject/2005/23/a/p2.htm>.

ภาษาต่างประเทศ

BOOKS

Franklin D. Ohrtman, Jr. (2003). **Softswitch Architecture for VoIP**. London, GB: McGraw-Hill Education - Europe.

Minoli Daniel. (1998). **Delivery voice over IP Network** (pp.79-114). United States of America: John Wiley & Sons, Inc.

Schulzrinne Henning (1996). **RFC 1889 : RTP A transport Protocol for Real-Time Applications**. United States, NY: Department of Computer Science, Columbia University

ELECTRONIC SOURCES

Chris Bajorek. (2003). R-Value Vs. MOS. Retrieved 20 March 2010, form <http://www.callcentermagazine.com/shared/printableArticle.jhtml?articleID=8701338>

Ixia.(1998-2011). Assessing VoIP Call Quality Using the E-mode. Retrieved 20 March 2010, form http://www.ixiacom.com/library/white_papers/display?skey=voip_quality

Jdugan, Mitchkutzko (2008). Iperf . Retrieved 20 March 2010, form <http://sourceforge.net/projects/iperf/>

Nessoft, LLC (2005). How is MOS calculated in PingPlotter Pro?. Retrieved 20 May 2010, form <http://www.nessoft.com/kb/50>

