การออกแบบและพัฒนาเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธ

นางสาว อมรรัตน์ จิระพรกุล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2024-2

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN AND IMPLEMENTATION OF AN INTENTIONAL-NAMING COMMUNICATION FRAMEWORK FOR BLUETOOTH NETWORKS

Miss Amornrut Jirapornkul

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2005

ISBN 974-53-2024-2

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธ
โดย	นางสาว อมรรัตน์ จิระพรกุล
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร. เฉลิมเอก อินทนากรวิวัฒน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์)

..... กรรมการ
(อาจารย์ ธงชัย โรจน์กังสดาล)

..... กรรมการ
(อาจารย์ ดร.วิษณุ โคตรจรัส)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อมรรัตน์ จิระพรกุล : การออกแบบและพัฒนาเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธ. (DESIGN AND IMPLEMENTATION OF AN INTENTIONAL-NAMING COMMUNICATION FRAMEWORK FOR BLUETOOTH NETWORKS) อ. ที่ปรึกษา: อ.ดร.เฉลิมเอก อินทนากรวิวัฒน์, 106 หน้า. ISBN 974-53-2024-2

งานวิจัยชิ้นนี้ได้ทำการออกแบบ และพัฒนาเฟรมเวิร์คการติดต่อสื่อสารสำหรับอุปกรณ์ในเครือข่ายบลูทูธ โดยใช้การระบุชื่อในการติดต่อสื่อสารแทนการติดต่อโดยใช้เลขที่อยู่บลูทูธของอุปกรณ์ ทั้งนี้ผู้ใช้สามารถติดต่อไปยังบริการที่ต้องการได้โดยไม่ต้องสนใจตำแหน่งของอุปกรณ์ที่ต้องการติดต่อ เฟรมเวิร์คสามารถจัดการกับการเปลี่ยนแปลงที่อาจเกิดขึ้นระหว่างการติดต่อสื่อสาร อาทิเช่น การเปลี่ยนตำแหน่งของอุปกรณ์หรือการเปลี่ยนแปลงบริการ โดยมีจุดประสงค์เพื่อลดความซับซ้อนของขั้นตอนการติดต่อสื่อสารในเครือข่ายบลูทูธ และสามารถนำเฟรมเวิร์คไปประยุกต์ใช้ในเครือข่ายบลูทูธรูปแบบต่างๆ ได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต อมรรัตน์ จิระพรกุล
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา
ปีการศึกษา 2548

##4670590021: MAJOR COMPUTER SCIENCE
KEY WORD: BLUETOOTH / PICONET / SCATTERNET / INTENTIONAL NAMING

AMORNRUT JIRAPORNKUL: DESIGN AND IMPLEMENTATION OF AN INTENTIONAL-NAMING COMMUNICATION FRAMEWORK FOR BLUETOOTH NETWORKS. THESIS ADVISOR: CHALERMEK INTANAGONWIWAT, Ph.D. 106 pp. ISBN 974-53-2024-2

This research is aimed for design and development of an Intentional-Naming Communication Framework for Bluetooth Networks. The concept of this framework is to provide communication services among bluetooth devices using intentional names to specify target devices instead of using bluetooth addresses. Therefore, users can communicate with devices of interest transparently from the actual locations or addresses of the devices. This framework also provides a mechanism to manage the change of locations or services that may occur during communications. This framework can reduce the complexity in developing applications that run on bluetooth networks.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering Student's signature Amornrut Jirapornkul
Field of study Computer Science Advisor's signature Chalermek
Academic year 2005

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์ ที่ให้ความช่วยเหลือเป็นอย่างมากในการทำงานวิจัยนี้ และสละเวลาให้คำปรึกษาและคำแนะนำที่ดีเสมอมา

ขอขอบคุณ ผศ. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา อาจารย์ ธงชัย โรจน์กังสดาล และอาจารย์ ดร.วิษณุ โคตรจรัส กรรมการวิทยานิพนธ์ ที่กรุณาให้คำแนะนำและตรวจสอบแก้ไขต้นฉบับวิทยานิพนธ์นี้

ขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ที่ให้คำปรึกษา รวมทั้งความสนุกสนาน และความบันเทิงมาโดยตลอด

สุดท้ายนี้ ขอขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้การสนับสนุนในด้านต่างๆ และให้กำลังใจที่ดีเสมอมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อภาษาไทย.....ง
บทคัดย่อภาษาอังกฤษ.....จ
กิตติกรรมประกาศ.....ฉ
สารบัญ.....ช
สารบัญตาราง.....ญ
สารบัญรูปภาพ.....ฎ

บทที่ 1 บทนำ.....1
1.1 ความเป็นมาและความสำคัญของปัญหา.....1
1.2 วัตถุประสงค์ของการวิจัย.....2
1.3 ขอบเขตการวิจัย.....3
1.4 ขั้นตอนการวิจัย.....3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....4
บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง.....5
2.1 งานวิจัยที่เกี่ยวข้อง.....5
2.1.1 Intentional Naming System.....5
2.1.2 Directed Diffusion.....5
2.2 ทฤษฎีที่เกี่ยวข้อง.....5
2.2.1 ระบบของการระบุชื่อ (Naming System).....5
2.2.2 พิโคเน็ต (Piconet) และสแคทเทอร์เน็ต (Scatternet).....6
2.2.3 มัลติคาสโพรโทคอล (Multicast Protocols).....7
บทที่ 3 การออกแบบเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อ.....11
3.1 รูปแบบการระบุชื่อ.....11
3.2 ลักษณะการติดต่อสื่อสารที่พิจารณา.....12
3.3 โครงสร้างของระบบ.....13
3.3.1 ส่วนการจัดการภายในพิโคเน็ต.....13

3.3.2 ส่วนการจัดการเพื่อการติดต่อสื่อสารระหว่างกลุ่มของพิโคเน็ต 16
3.3.3 สรุประบบโดยรวม 19
3.3.4 สรุปบทบาทและหน้าที่ของอุปกรณ์บลูทูธ 20
3.4 การออกแบบเฟรมเวิร์ค 20
3.4.1 สภาวะแวดล้อมการทำงานของเฟรมเวิร์ค 20
3.4.2 องค์ประกอบของอุปกรณ์บลูทูธ 21
3.4.3 องค์ประกอบการสร้างเครือข่ายการติดต่อสื่อสาร 25
3.5 การออกแบบการจัดการแผนภาพต้นไม้ของชื่อ 28
3.5.1 โครงสร้างชื่อ 28
3.5.2 ขั้นตอนวิธีการจัดการแผนภาพต้นไม้ของชื่อ 29
3.6 โครงสร้างของข่าวสารต่างๆ 32
3.6.1 ข่าวสารของระบบ (System Message) 32
3.6.2 ข่าวสารจากการประมวลผลคำสั่งของผู้ใช้ (User Message) 34
3.6.3 ข่าวสารสำหรับอุปกรณ์บลูทูธ (Destination Message) 36
บทที่ 4 การพัฒนาเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อ 43
4.1 แพคเกจ intentionalnaming.bluetooth.connection 43
4.2 แพคเกจ intentionalnaming.bluetooth.connection.impl 64
4.3 แพคเกจ intentionalnaming.nametree 83
บทที่ 5 ทดสอบการใช้งานเฟรมเวิร์ค 86
5.1 สภาวะที่ใช้ในการทดสอบ 86
5.2 ทดสอบการทำงานของเฟรมเวิร์ค 86
5.2.1 ทดสอบการจัดการในพิโคเน็ต 86
5.2.2 ทดสอบการประยุกต์ใช้กับสแคทเทอร์เน็ตรูปแบบต่างๆ 89
5.2.3 ทดสอบการจัดเส้นทางและสภาวะที่มีการเปลี่ยนแปลง 91
5.3 ตัวอย่างการทดสอบประสิทธิภาพ 93
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 99
6.1 สรุปผลการวิจัย 99
6.2 ปัญหาและข้อจำกัดของงานวิจัย 100
6.3 ข้อเสนอแนะ 100

รายการอ้างอิง..... 101
ภาคผนวก..... 102
ประวัติผู้เขียนวิทยานิพนธ์ 106

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 3.1 อักขระพิเศษและความหมาย 11
ตารางที่ 3.2 ตัวอย่างการตั้งชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ 12
ตารางที่ 3.3 ตัวอย่างการระบุชื่อเพื่อการส่งไปยังปลายทาง 12
ตารางที่ 3.4 บทบาท และหน้าที่ของอุปกรณ์บลูทูธในเครือข่าย 20
ตารางที่ 3.5 โครงสร้างข่าวสารสำหรับระบบ 32
ตารางที่ 3.6 โครงสร้างข่าวสารเพื่อประมวลผลจากคำสั่งของผู้ใช้ 34
ตารางที่ 3.7 โครงสร้างข่าวสารสำหรับอุปกรณ์บลูทูธ 36

ตารางที่ 4.1 คลาสต่อประสานของเฟรมเวิร์ค 44
ตารางที่ 4.2 คลาสสำหรับผู้ใช้ 45
ตารางที่ 4.3 ตัวกระทำการของคลาส BTDeviceFactory โดยสรุป 45
ตารางที่ 4.4 เขตข้อมูลของคลาส BTDevice 46
ตารางที่ 4.5 ตัวกระทำการของคลาส BTDevice 49
ตารางที่ 4.6 ตัวกระทำการของคลาส BTDeviceListener 57
ตารางที่ 4.7 ตัวกระทำการของคลาส BTDeviceConnector 58
ตารางที่ 4.8 ตัวกระทำการของคลาส BTDeviceNotifier 60
ตารางที่ 4.9 ตัวกระทำการของคลาส BTDeviceRemoteDevice 62
ตารางที่ 4.10 ตัวอย่างตารางเส้นทาง 80
ตารางที่ 4.11 ตัวอย่างรูปแบบข้อมูลที่เก็บในตาราง hashIndexs 84

ตารางที่ 5.1 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลจำแนกตามประเภทข้อมูล 96
ตารางที่ 5.2 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปจำแจกตามชนิดข้อมูล 97

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญรูปภาพ

รูปที่ 2.1 รูปแบบการระบุคุณลักษณะ 6
รูปที่ 2.2 พิโคเน็ต [2] 6
รูปที่ 2.3 สแคทเทอร์เน็ต [2] 7
รูปที่ 2.4 การส่งผ่านข้อมูลแบบเอกซ์พลิซิทมัลติคาส (Explicit Multicast) [6] 8
รูปที่ 2.5 ไอจีเอ็มพี และการจัดเส้นทางสำหรับมัลติคาส [6] 8

รูปที่ 3.1 ตัวอย่างแสดงมุมมองของผู้ใช้ในพิโคเน็ต 15
รูปที่ 3.2 ตัวอย่างแผนภาพต้นไม้ของชื่อ (Name-Tree) 15
รูปที่ 3.3 ตัวอย่างการส่งผ่านข้อมูลระหว่างพิโคเน็ต 16
รูปที่ 3.4 กระบวนการกระจายข้อมูล 17
รูปที่ 3.5 กระบวนการตัดเส้นทาง 18
รูปที่ 3.6 กระบวนการต่อเส้นทาง 18
รูปที่ 3.7 สรุประบบโดยรวม 19
รูปที่ 3.8 แสดงส่วนประกอบสภาวะแวดล้อมของเฟรมเวิร์ค 21
รูปที่ 3.9 องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์ 22
รูปที่ 3.10 องค์ประกอบของอุปกรณ์บลูทูธสเลฟ 23
รูปที่ 3.11 องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์/สเลฟบริดจ์ 25
รูปที่ 3.12 ตัวอย่างการสร้างพิโคเน็ต 26
รูปที่ 3.13 ขั้นตอนการเพิ่ม/ลบชื่อ 27
รูปที่ 3.14 ตัวอย่างการสร้างสแคทเทอร์เน็ตแบบสเลฟ/สเลฟบริดจ์ 27
รูปที่ 3.15 ตัวอย่างการสร้างสแคทเทอร์เน็ตแบบมาสเตอร์/สเลฟบริดจ์ 28
รูปที่ 3.16 แสดงค่าจากการระบุชื่อ 28
รูปที่ 3.17 แสดงค่าในแผนภาพต้นไม้ของชื่อ 29
รูปที่ 3.18 ขั้นตอนวิธีการเพิ่มชื่อเข้าแผนภาพต้นไม้ (Add Name-Tree Algorithm) 30
รูปที่ 3.19 ขั้นตอนวิธีการลบชื่อออกจากแผนภาพต้นไม้ (Delete Name-Tree Algorithm) 30
รูปที่ 3.20 ขั้นตอนวิธีการค้นหาปลายทางตามชื่อที่ระบุจากแผนภาพต้นไม้ (Search Name-Tree Algorithm) 31

รูปที่ 4.1 แบบจำลองแพคเกจ intentionalnaming.bluetooth.connection 43
รูปที่ 4.2 แบบจำลองแพคเกจ intentionalnaming.bluetooth.connection.impl 64
รูปที่ 4.3 แบบจำลองคลาส BTDeviceFactoryImpl 65
รูปที่ 4.4 แบบจำลองคลาส BTDeviceImpl 66
รูปที่ 4.5 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createBTDeviceConnectorImpl() 67
รูปที่ 4.6 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createBTDeviceNotifierImpl() 68
รูปที่ 4.7 แบบจำลองของคลาส BTDviceNotifierImpl 71
รูปที่ 4.8 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ openConnection() 72
รูปที่ 4.9 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createIntentionalName() 73
รูปที่ 4.10 แบบจำลองคลาส BTDeviceConnectorImpl 74
รูปที่ 4.11 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createConnection() 75
รูปที่ 4.12 แบบจำลองคลาส BTDeviceHandler 77
รูปที่ 4.13 แบบจำลองคลาส BTDeviceRouter 79
รูปที่ 4.14 แบบจำลองคลาสของแพคเกจ intentionalnaming.nametree 84

รูปที่ 5.1 ตัวอย่างสถานะที่สเลฟได้รับจากการประกาศชื่อ 87
รูปที่ 5.2 ตัวอย่างสถานะที่มาสเตอร์ได้รับจากการประกาศชื่อ 88
รูปที่ 5.3 ตัวอย่างการค้นหาอุปกรณ์ของมาสเตอร์ 88
รูปที่ 5.4 ตัวอย่างสถานะที่มาสเตอร์ได้รับจากการสร้างการติดต่อสื่อสารไปยังสเลฟ 89
รูปที่ 5.5 ตัวอย่างสถานะที่สเลฟได้รับเมื่อได้รับการติดต่อจากมาสเตอร์ 89
รูปที่ 5.6 ตัวอย่างอุปกรณ์ในเครือข่ายบลูทูธแบบ Slave/Slave Mesh 89
รูปที่ 5.7 ตัวอย่างอุปกรณ์ในเครือข่ายบลูทูธแบบ Tree Hierarchy 90
รูปที่ 5.8 แสดงข้อมูลการรับข้อมูลของอุปกรณ์ 91
รูปที่ 5.9 ตัวอย่างการส่งข้อมูลและการตัดเส้นทาง 92
รูปที่ 5.10 ตัวอย่างเปลี่ยนแปลงในเครือข่ายและการต่อเส้นทาง 92
รูปที่ 5.11 เวลาที่ใช้ค้นหาชื่อจากแผนภาพต้นไม้ 94

รูปที่ 5.12 ทดสอบในเครือข่ายบลูทูธแบบ Slave/Slave Mesh 95
รูปที่ 5.13 กราฟแสดงจำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูล 96
รูปที่ 5.14 กราฟแสดงจำนวนข้อมูลในเครือข่ายหลังจากรู้เส้นทางส่งข้อมูล 97

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

บลูทูธ (Bluetooth) [1] เป็นรูปแบบหนึ่งของเทคโนโลยีการติดต่อสื่อสารไร้สายโดยใช้คลื่นวิทยุ มีกลุ่มสมาชิกบลูทูธเอสไอจี (Bluetooth SIG) ที่กำหนดมาตรฐานสำหรับบลูทูธร่วมกัน บลูทูธถูกออกแบบมาสำหรับอุปกรณ์แบบพกพาที่ใช้พลังงานจากแบตเตอรีเป็นหลัก เพื่อช่วยอำนวยความสะดวกในการติดต่อสื่อสาร อุปกรณ์ในเครือข่ายสามารถเคลื่อนที่ได้ สำหรับการติดต่อไปยังอุปกรณ์ปลายทางที่พบในปัจจุบันทำได้โดยการระบุเลขที่อยู่บลูทูธ (Bluetooth Address) ของอุปกรณ์ปลายทางที่ต้องการติดต่อ (เลขที่อยู่ บลูทูธ มี 48 บิท ซึ่งไม่ซ้ำกันในแต่ละอุปกรณ์ เปรียบเสมือนเลขที่อยู่สัมบูรณ์ของอุปกรณ์ (MAC address)) ลักษณะการติดต่อโดยการระบุเลขที่อยู่บลูทูธดังกล่าวทำให้เกิดเครือข่ายบลูทูธขึ้นเพื่อให้อุปกรณ์มีการติดต่อสื่อสารแลกเปลี่ยนข้อมูลระหว่างกันได้ทั้งการติดต่อไปยังผู้รับคนเดียว และการติดต่อไปยังกลุ่มของผู้รับ ตัวอย่างการติดต่อสื่อสารที่พบโดยทั่วไป เช่น การเล่นเกมส์ผ่านบลูทูธ การประชุมผ่าน บลูทูธ การส่งไฟล์ เป็นต้น

เมื่อพิจารณารูปแบบการติดต่อที่ใช้ในปัจจุบันพบว่าการใช้เลขที่อยู่บลูทูธสำหรับอ้างอิงถึงปลายทางยากต่อการจดจำและไม่สื่อความหมาย จึงมีแนวคิดในการนำชื่อมาใช้ในการระบุปลายทางแทนการใช้เลขที่อยู่บลูทูธ เพื่อสร้างรูปแบบในการอ้างอิงถึงปลายทางที่มีความยืดหยุ่นมากขึ้นให้สามารถรองรับการติดต่อสื่อสารในเครือข่ายบลูทูธ รวมถึงปัจจัยด้านการเปลี่ยนแปลงที่เกิดขึ้นที่อุปกรณ์ปลายทาง ทั้งการเคลื่อนที่ของอุปกรณ์ระหว่างการสื่อสารหรือการเปลี่ยนแปลงบริการ ซึ่งเป็นลักษณะที่พบในการติดต่อสื่อสารผ่านเครือข่ายบลูทูธ การออกแบบรูปแบบในการติดต่อสื่อสารที่เหมาะสมจึงเป็นสิ่งสำคัญในการพัฒนาประสิทธิภาพของการติดต่อสื่อสารด้วยเครือข่ายบลูทูธ อีกทั้งยังมีความเป็นไปได้ในทางปฏิบัติเพราะว่าไม่จำเป็นต้องเปลี่ยนแปลงโมเดลมาตรฐานของบลูทูธที่ใช้กันในปัจจุบัน

จากการศึกษาพบว่า ขั้นตอนโดยทั่วไปในการที่จะติดต่อสื่อสารในเครือข่ายบลูทูธนั้น อุปกรณ์ต้องทราบเลขที่อยู่บลูทูธของอุปกรณ์รอบข้าง เพื่อระบุไปยังเลขที่อยู่บลูทูธของอุปกรณ์ปลายทางที่ต้องการติดต่อ เมื่อติดต่อไปยังอุปกรณ์ปลายทางได้จึงจะสามารถสอบถามบริการที่ปลายทางรองรับ จากนั้นจึงจะติดต่อไปยังบริการที่ต้องการได้ [2] เมื่อพิจารณาขั้นตอนการติดต่อสื่อสารที่เกิดขึ้นสรุปได้ว่า สิ่งที่ผู้ใช้สนใจอยู่ที่บริการที่ต้องการ (Service) และชื่อเรียกไปยังปลายทางตามที่ผู้ใช้สนใจ (Subject) ทั้งนี้ขึ้นอยู่กับว่าผู้ใช้ต้องการอะไร ส่วนเลขที่อยู่ของอุปกรณ์

ปลายทาง รวมทั้งตำแหน่งที่ตั้งของอุปกรณ์ปลายทางนั้นไม่ใช่เป็นประเด็นที่ผู้ใช้สนใจ ดังนั้น รูปแบบที่ใช้ในการติดต่อ และอ้างอิงถึงปลายทางที่เหมาะสมจึงควรอยู่ในรูปแบบของคุณลักษณะและค่า (Attribute-based) ของสิ่งที่สนใจ ซึ่งนำเสนอไว้ใน [3, 4] ผู้วิจัยได้นำรูปแบบการระบุชื่อของสิ่งที่ผู้ใช้สนใจมาประยุกต์ใช้ในการสื่อสาร และออกแบบกลไกการทำงานตลอดจนการหาเส้นทางและการส่งต่อข้อมูลของเฟรมเวิร์คให้เหมาะสมกับลักษณะการติดต่อสื่อสารสำหรับเครือข่ายบลูทูธ รวมทั้งเพิ่มประสิทธิภาพ ลดความซับซ้อนของการติดต่อสื่อสาร แทนมุมมองแบบเดิมที่ใช้เลขที่อยู่บลูทูธในการอ้างอิงปลายทาง

วิทยานิพนธ์นี้จึงมีเป้าหมายในการออกแบบและสร้างกลไกการทำงาน การติดต่อสื่อสารอยู่ในรูปแบบการระบุชื่อตามจุดมุ่งหมาย (Intentional Naming) สำหรับการติดต่อสื่อสารในเครือข่ายบลูทูธโดยการระบุคุณลักษณะของสิ่งที่สนใจ จากบริการและชื่อเรียกอุปกรณ์ปลายทางที่สื่อความหมายต่อผู้ใช้แทนการติดต่อโดยการระบุเลขที่อยู่ของอุปกรณ์ การทำงานในส่วนของการระบุชื่อตามจุดมุ่งหมายแยกได้เป็น 2 ส่วนหลัก ส่วนแรกเป็นส่วนที่ผู้ใช้กำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ (Listen) ประกอบด้วยการระบุบริการและชื่อเรียกสำหรับอุปกรณ์ของตนเองเพื่อรอรับการติดต่อจากอุปกรณ์อื่น ส่วนที่สองเป็นส่วนการติดต่อเพื่อส่งไปยังปลายทาง (Send) โดยการระบุบริการและชื่อเรียกอุปกรณ์ปลายทางที่ต้องการ จุดประสงค์ของงานวิจัยนี้คือการนำรูปแบบการระบุชื่อของสิ่งที่ผู้ใช้สนใจ มาประยุกต์ใช้ในการสื่อสาร และออกแบบกลไกการทำงานตลอดจนการหาเส้นทางและการส่งต่อข้อมูล ซึ่งประยุกต์มาจากหลักการหาเส้นทางของพิม-เดนซ์โหมด (PIM-Dense Mode) โดยใช้ชื่อแทนมัลติคาสไอพี เพื่อให้เฟรมเวิร์คมีความเหมาะสมกับลักษณะการติดต่อสื่อสารสำหรับเครือข่ายบลูทูธ รวมทั้งเพิ่มประสิทธิภาพ ลดความซับซ้อนของการติดต่อสื่อสาร แทนมุมมองแบบเดิมที่ใช้เลขที่อยู่บลูทูธในการอ้างอิงปลายทาง ผู้ส่งข้อมูลไม่ต้องสนใจว่าตำแหน่งของบริการนั้นอยู่ที่ไหน อุปกรณ์บลูทูธซึ่งทำหน้าที่เป็นตัวกลางจะส่งข้อมูลไปยังตำแหน่งที่เหมาะสมต่อไปทั้งการส่งไปยังปลายทางโดยตรงหรือกลุ่มของปลายทางที่สนใจได้ และสามารถรองรับการติดต่อสื่อสารในเครือข่ายบลูทูธรูปแบบต่างๆ โดยพัฒนาเป็นเอพีไอ (API) เพื่อให้โปรแกรมเมอร์สามารถนำเฟรมเวิร์คนี้ไปพัฒนาโปรแกรมประยุกต์ด้วยเจทูเอ็มอี (J2ME) ตามสถาปัตยกรรมของเฟรมเวิร์คต่อไปได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาเฟรมเวิร์คสำหรับการติดต่อสื่อสารภายในเครือข่ายบลูทูธพิโคเน็ตและ สแคทเทอร์เน็ต โดยการใช้ชื่อที่สื่อความหมายในการอธิบายคุณลักษณะของสิ่งที่สนใจตามจุดมุ่งหมายของผู้ใช้

1.3 ขอบเขตการวิจัย

รูปแบบการติดต่อสื่อสารที่พิจารณามีด้วยกัน 3 ประเภท คือ

- ยูนิคาส (Unicast): ติดต่อไปยังปลายทางเพียงโหนดเดียว
- มัลติคาส (Multicast): ติดต่อไปกลุ่มสมาชิกในเครือข่าย
- บรอดคาส (Broadcast): ติดต่อไปยังทุกโหนดในเครือข่าย

2. สร้างส่วนต่อประสานสำหรับการรับ-ส่งข้อมูลด้วยการระบุชื่อตามจุดมุ่งหมาย ตลอดจนการหาเส้นทางและการส่งข้อมูลสำหรับเครือข่ายบลูทูธ
3. จัดทำคำอธิบายของคลาสและฟังก์ชันภาษาจาวาของเฟรมเวิร์คการติดต่อด้วยการระบุชื่อตามจุดมุ่งหมาย
4. พัฒนาโปรแกรมประยุกต์ J2ME เพื่อทดสอบการทำงานของเฟรมเวิร์ค ด้วยตัวจำลองการทำงานบลูทูธ (Emulator) J2ME Wireless Toolkit, Version 2.2
5. ทดสอบการทำงานของเฟรมเวิร์คโดยการจำลองสภาวะที่มีการเปลี่ยนแปลงเกิดขึ้นในเครือข่ายอันเนื่องมาจาก
   - โหนดมีการเคลื่อนที่
   - การเปลี่ยนแปลงคุณลักษณะของบริการ
6. ทดสอบการทำงานของเฟรมเวิร์คโดยการนำไปประยุกต์ใช้ในสแคทเทอร์เน็ตรูปแบบต่างๆ ได้แก่ แบบ Slave/Slave Mesh และ Tree Hierarchy
7. เฟรมเวิร์คที่พัฒนาขึ้น ไม่พิจารณาเกี่ยวกับด้านความปลอดภัย (Security) และความขัดแย้งของชื่อ (Name Conflict)

1.4 ขั้นตอนการวิจัย

1. ศึกษาทฤษฎีและการทำงานของบลูทูธ รวมทั้งลักษณะการติดต่อสื่อสารในเครือข่ายบลูทูธ
2. ศึกษาทฤษฎีและรูปแบบในการติดต่อสื่อสารโดยการระบุชื่อ
3. ศึกษาทฤษฎีและการทำงานของมัลติคาสโพรโทคอล (Multicast Protocol)
4. ศึกษาทฤษฎีและการทำงานของการจัดเส้นทางในเครือข่ายเฉพาะกิจ (Ad-hoc Networks Routing Protocol)

5. วิเคราะห์และออกแบบระบบการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธ และศึกษาทดลองเครื่องมือต่างๆ ที่จะนำมาใช้งาน
6. พัฒนาโปรแกรมตามที่ได้ออกแบบไว้
7. ทดสอบและแก้ไข ให้เป็นไปตามขอบเขตที่กำหนดไว้
8. วิเคราะห์และสรุปผลการวิจัย พร้อมข้อเสนอแนะ
9. จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธ ซึ่งสามารถนำมาใช้แทนมุมมองการติดต่อสื่อสารแบบเดิมที่ใช้เลขที่อยู่บลูทูธ
2. สามารถนำเฟรมเวิร์คไปใช้กับเครือข่ายบลูทูธสแคทเทอร์เน็ตรูปแบบต่างๆ ได้
3. สามารถส่งผ่านข้อมูลแบบมัลติคาสได้โดยไม่ต้องเปลี่ยนแปลงโมเดลมาตรฐานของบลูทูธ
4. เพิ่มประสิทธิภาพการติดต่อสื่อสารในเครือข่ายบลูทูธ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

2.1.1 Intentional Naming System [4]

งานวิจัยนี้งานวิจัยนี้เป็นการติดต่อบริการโดยอธิบายคุณลักษณะของบริการที่ต้องการในรูปของชื่อแบบลำดับชั้น (Hierarchical) ในการระบุบริการซึ่งประกอบด้วยคุณลักษณะ (attribute) และค่า (value) จากนั้นให้อุปกรณ์ใดๆ ในเครือข่ายทำหน้าที่เป็นตัวกลาง เพื่อพิจารณาส่งต่อข้อมูลไปยังผู้ให้บริการ โดยใช้ข้อมูลในแผนภาพต้นไม้ของชื่อ (Name-Tree) อย่างไรก็ตามงานวิจัยชิ้นนี้มีวิธีการส่งผ่านข้อมูลซึ่งมีพื้นฐานบนการใช้เลขที่อยู่ไอพี ผู้วิจัยได้นำเสนอกลไกการส่งข้อมูลเพื่อให้เหมาะกับเครือข่ายบลูทูธสแคทเทอร์เน็ตซึ่งมีลักษณะของอุปกรณ์ในเครือข่ายได้แก่ มาสเตอร์ สเลฟ และบริดจ์ซึ่งแตกต่างจากอุปกรณ์ในเครือข่ายทั่วไป และมีพื้นฐานการส่งต่อข้อมูลแบบเครื่องต่อเครื่อง (hop-by-hop) โดยใช้เลขที่อยู่ของอุปกรณ์บลูทูธ

2.1.2 Directed Diffusion [3] [5]

งานวิจัยชิ้นนี้ได้นำหลักการของการระบุชื่อ (Naming) มาใช้กับเครือข่ายของอุปกรณ์รับรู้ (Sensor Networks) งานวิจัยนี้เป็นการติดต่อสื่อสารเพื่อขอข้อมูลจากแหล่งข้อมูล (source) โดยผู้ขอข้อมูล (sink) บรรยายรายละเอียดของข้อมูลที่ต้องการในรูปแบบของคุณลักษณะ (data naming) เมื่ออุปกรณ์ที่ได้รับคุณลักษณะตรวจสอบข้อมูลตนเองแล้วว่าสอดคล้องกับคุณลักษณะที่ผู้ร้องขอระบุจึงจะส่งข้อมูลตอบกลับไป

ผู้วิจัยได้ใช้กลไกการส่งต่อข้อมูลโดยการกระจายข้อมูลในครั้งแรกของการส่งต่อข้อมูลเพื่อเรียนรู้และเลือกเส้นทางที่เหมาะสมในการส่งข้อมูลครั้งต่อไปโดยมีพื้นฐานบนการส่งบนเลขที่อยู่ของอุปกรณ์ เช่นเดียวกับที่งานวิจัยชิ้นนี้ โดยนำหลักการดังกล่าวมาประยุกต์ใช้สำหรับเครือข่ายบลูทูธ

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 ระบบของการระบุชื่อ (Naming System)[6]

ระบบของการระบุชื่อ (Naming System) เปรียบเสมือนฐานข้อมูลชนิดหนึ่ง เนื่องจาก ในการหาวัตถุที่เกี่ยวข้องกับชื่อที่ระบุ มีหลักการเช่นเดียวกับหลักการคิวรีข้อมูลจากฐานข้อมูล ระบบของการระบุชื่อ จึงเป็นกลไกในการหาคำตอบของสิ่งที่สอดคล้องกับชื่อที่ถูกระบุมา คำตอบที่ได้รับ

กลับมานั้น จะขึ้นอยู่กับข้อมูลที่เก็บเอาไว้ และกระบวนการในการวินิจฉัยผลลัพธ์ ตัวอย่างของระบบของการระบุชื่อแบบง่าย เช่น ระบบไฟล์ ซึ่งประกอบด้วยเซตของชื่อ/ค่า (name/value pair) เมื่อได้รับ “ชื่อ” ก็จะคืน “ค่า” ที่สอดคล้องกับชื่อกลับไป ระบบของการระบุชื่อที่ซับซ้อนยิ่งขึ้น เช่น บริการสมุดหน้าเหลือง ที่ใช้สำหรับค้นหาทรัพยากรที่มีในเครือข่าย โดยใช้กลุ่มของคุณลักษณะที่ประกอบด้วย คุณลักษณะ (attribute) และค่า (value) ในการอธิบายวัตถุที่สนใจ เช่น ลูกข่ายต้องการทราบเลขที่อยู่ของเครื่องแม่ข่าย ก็จะสร้างคิวรี (query) โดยการระบุคุณลักษณะของเครื่องแม่ข่ายที่ต้องการ ดังรูปที่ 2.1

```
“Service = cycle, arch = mips, load < 2, mips = max”
```

รูปที่ 2.1 รูปแบบการระบุคุณลักษณะ

เรียกระบบในลักษณะนี้ว่าระบบของการระบุชื่อโดยการอธิบาย หรือการระบุชื่อตามคุณลักษณะ (Descriptive หรือ Attribute-based Naming Systems)

จากหลักการทำงานระบบของการระบุชื่อที่กล่าวมาข้างต้น ผู้วิจัยได้นำการระบุชื่อตามคุณลักษณะไปประยุกต์ใช้เป็นรูปแบบของการติดต่อสื่อสารบนเครือข่ายบลูทูธสำหรับผู้ใช้ ทำให้การติดต่อสื่อสารที่ผู้ใช้มองเห็นแตกต่างออกไปจากลักษณะเดิมที่ใช้การติดต่อสื่อสารด้วยการระบุเลขที่อยู่ของอุปกรณ์ปลายทางที่ต้องการติดต่อ

2.2.2 **พิโคเน็ต** (Piconet) **และสแคทเทอร์เน็ต** (Scatternet)

เครือข่ายการติดต่อสื่อสารของบลูทูธมีชื่อเรียกว่า พิโคเน็ต และสแคทเทอร์เน็ต ในแต่ละพิโคเน็ต ดังรูปที่ 2.2 มีมาสเตอร์ (Master) ทำหน้าที่จัดการส่งผ่านข้อมูลไปยังสเลฟ (Slave) ที่อยู่ใกล้เคียง โดยมีอุปกรณ์ตัวหนึ่งซึ่งเริ่มต้นการติดต่อจะทำหน้าที่เป็นมาสเตอร์ ส่วนอุปกรณ์ที่รับการติดต่อจากมาสเตอร์จะกลายเป็นสเลฟ ในพิโคเน็ตหนึ่งๆ มีสเลฟได้ไม่เกิน 7 ตัวและสเลฟในพิโคเน็ตเดียวกันไม่สามารถติดต่อกับสเลฟด้วยกันได้ การติดต่อจะกระทำได้โดยผ่านมาสเตอร์ในพิโคเน็ตของตนเองเท่านั้น

รูปที่ 2.2 พิโคเน็ต [2]

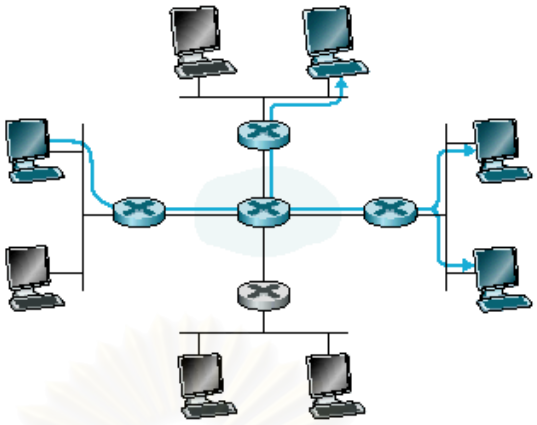
กรณีที่ มีอุปกรณ์ต้องการเข้ามาติดต่อสื่อสารกับพิโคเน็ต ในขณะที่มาสเตอร์รองรับสเลฟไว้ถึง 7 ตัวแล้ว หรือกรณีที่มาสเตอร์ต้องการส่งผ่านข้อมูลไปยังอุปกรณ์ตัวอื่นที่อยู่นอกรัศมีการส่งสัญญาณของตนเอง ก็จะให้สเลฟตัวหนึ่งที่สามารถเชื่อมต่อไปยังพิโคเน็ตใหม่ได้ รับหน้าที่เป็นบริดจ์ (Bridge) โดยที่บริดจ์อาจมีบทบาทเป็นสเลฟในพิโคเน็ตเดิม และเป็นมาสเตอร์ในพิโคเน็ตใหม่ (สเลฟ/มาสเตอร์) หรือ เป็นสเลฟของทั้งสองพิโคเน็ต (สเลฟ/สเลฟ) ณ ขณะใดขณะหนึ่ง บริดจ์จะทำหน้าที่ได้เพียงบทบาทเดียวเท่านั้น เรียกเครือข่ายที่ประกอบด้วยพิโคเน็ตตั้งแต่ 2 พิโคเน็ตขึ้นไปซึ่งมีการเชื่อมต่อระหว่างกันด้วยบริดจ์ว่า สแคทเทอร์เน็ต ดังรูปที่ 2.3

รูปที่ 2.3 สแคทเทอร์เน็ต [2]

2.2.3 **มัลติคาสโพรโทคอล** (Multicast Protocols) [7-9]

มัลติคาส (Multicast) เป็นการส่งผ่านข้อมูลจากผู้ส่งคนหนึ่งไปยังกลุ่มของผู้รับได้ด้วยการส่งเพียงครั้งเดียว กลุ่มของผู้รับระบุได้โดยใช้ไอพีมัลติคาส (IP Multicast) มีเลขที่อยู่ระหว่าง 224.0.0.0 ถึง 239.255.255.255 ทั้งนี้จำนวนสมาชิกในกลุ่มอาจประกอบด้วยสมาชิกตั้งแต่ 0 ขึ้นไปและไม่จำกัดจำนวน สมาชิกสามารถเข้าร่วมหรือออกจากกลุ่มได้โดยไม่ต้องแจ้งให้สมาชิกในกลุ่มตัวอื่น ผู้ส่งไม่จำเป็นต้องรู้ว่าผู้รับเป็นใคร และผู้ส่งไม่จำเป็นต้องอยู่ในกลุ่มเดียวกับผู้รับ มัลติคาส จึงอยู่กึ่งกลางระหว่างการส่งผ่านข้อมูลแบบยูนิคาส และบรอดคาส

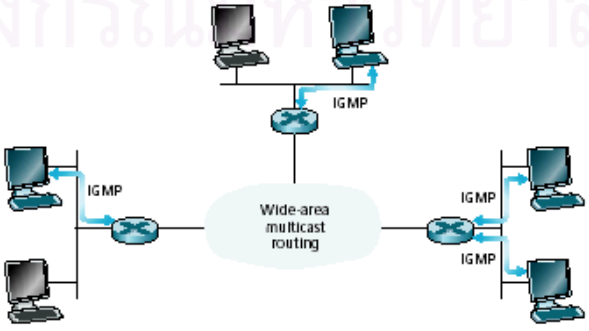
การส่งผ่านข้อมูลแบบเอกซ์พลิซิทมัลติคาส (Explicit Multicast) เป็นวิธีการที่มีประสิทธิภาพมากที่สุดสำหรับการส่งผ่านข้อมูลแบบมัลติคาส ทำงานในระดับชั้นเครือข่าย (Network Layer) ผู้ส่งทำการส่งข้อมูลเพียงครั้งเดียว จากนั้นอุปกรณ์จัดเส้นทาง (router) ทำหน้าที่ส่งผ่านข้อมูลไปยังผู้รับ ดังรูปที่ 2.4 จึงเป็นการประหยัดแบนด์วิดท์ (Bandwidth) และพลังงานในการส่งข้อมูล ทำให้การติดต่อสื่อสารมีประสิทธิภาพ



รูปที่ 2.4 การส่งผ่านข้อมูลแบบเอกซ์พลิซิทมัลติคาส (Explicit Multicast) [7]

ไอจีเอ็มพี (Internet Group Management Protocol – IGMP) เป็นโพรโทคอลระหว่างโฮสต์กับอุปกรณ์จัดเส้นทางที่เชื่อมต่อกับโฮสต์โดยตรง โฮสต์ทำการแจ้งกลุ่มมัลติคาส (Multicast Group) ที่โปรแกรมประยุกต์ของโฮสต์ต้องการเข้าร่วม ไปยังอุปกรณ์จัดเส้นทางที่เชื่อมต่อกับโฮสต์ เมื่ออุปกรณ์จัดเส้นทางได้รับข้อมูลที่ส่งมายังกลุ่มมัลติคาส ก็จะสามารถส่งข้อมูลดังกล่าวไปยังโฮสต์ได้ ดังรูปที่ 2.5 ไอจีเอ็มพี ประกอบด้วยข่าวสาร (message) 3 ชนิด ดังต่อไปนี้

1) Membership query message เป็นข่าวสารที่อุปกรณ์จัดเส้นทางสอบถามไปยังโฮสต์ทุกตัวที่เชื่อมต่อกับมันว่าโฮสต์อยู่ในกลุ่มมัลติคาสใดบ้าง
2) Membership report เมื่อโปรแกรมประยุกต์ของโฮสต์เริ่มต้นเข้าร่วมในกลุ่มมัลติคาส โฮสต์จะรายงานไปยังอุปกรณ์จัดเส้นทางที่มันเชื่อมต่อ ว่าโฮสต์ต้องการเข้าร่วมกลุ่มมัลติคาสใด โดยไม่ต้องรอให้อุปกรณ์จัดเส้นทางสอบถามมายังโฮสต์
3) Leave group อุปกรณ์จัดเส้นทางส่ง Membership query message พร้อมทั้งเลขที่อยู่ของกลุ่ม (group address) ถ้าไม่มีโฮสต์ตัวใดตอบกลับแสดงว่าไม่มีโฮสต์ใดเข้าร่วม กลุ่มมัลติคาส นั้น เช่นเดียวกับ Soft state protocol ที่จะยกเลิกสถานะที่เก็บไว้เพื่อไม่ได้รับการรีเฟรชข้อมูลภายในระยะเวลาที่กำหนด



รูปที่ 2.5 ไอจีเอ็มพี และการจัดเส้นทางสำหรับมัลติคาส [7]

การจัดเส้นทางสำหรับมัลติคาส (Multicast Routing) จุดประสงค์ของการจัดเส้นทางสำหรับมัลติคาสคือการหาต้นไม้เชื่อมโยงไปยังอุปกรณ์จัดเส้นทางทุกตัวที่มีโฮสต์เป็นสมาชิกของกลุ่มมัลติคาส แบ่งต้นไม้การจัดเส้นทางสำหรับมัลติคาสออกได้ 2 แบบ ดังต่อไปนี้

1) ต้นไม้ร่วมสำหรับกลุ่ม (Group-Shared Tree): ต้นทาง (source) ทุกตัวใช้ต้นไม้เชื่อมโยง เดียวกันในการส่งข้อมูลไปยังกลุ่มมัลติคาส ข้อเสียของวิธีนี้คือต้นไม้เชื่อมโยงที่ได้อาจทำให้เส้นทางใช้ในการส่งข้อมูลจากต้นทางไปยังปลายทางไม่ใช่เส้นทางที่ดีที่สุดสำหรับต้นทางทุกตัว แทนต้นไม้ร่วมสำหรับกลุ่มได้ด้วยสัญลักษณ์ *(*, g)*
2) ต้นไม้เฉพาะสำหรับต้นทาง (Source-Based Tree): โดยสร้าง multicast routing tree เฉพาะของต้นทางแต่ละตัว เพื่อหาเส้นทางที่ดีที่สุดสำหรับต้นทางด้วยขั้นตอนวิธีอาร์พีเอฟ (Reverse Path Forwarding – RPF) มีขั้นตอนการทำงานดังนี้ เมื่ออุปกรณ์จัดเส้นทางได้รับข้อมูลมัลติคาสมาจากต้นทางอุปกรณ์จัดเส้นทางตัวนั้นก็จะส่งข้อมูลออกไปทุกๆ ช่องทาง ยกเว้นช่องทางที่มันได้รับข้อมูลมา และอุปกรณ์จัดเส้นทางจะรับเฉพาะข้อมูลที่ส่งมาจากเส้นทางที่เป็นเส้นทางกลับไปยังต้นทางที่สั้นที่สุดสำหรับตัวมันเอง และจะไม่รับข้อมูลจากเส้นทางที่ไม่ใช่สั้นที่สุดเมื่อกลับไปยังต้นทาง ถ้าอุปกรณ์จัดเส้นทางตัวใดไม่ต้องการรับข้อมูลมัลติคาส ก็จะส่ง การตัดเส้นทาง (pruning) ขึ้นไปยังเส้นทางที่ไม่ต้องการ สำหรับขั้นตอนวิธีอาร์พีเอฟอุปกรณ์จัดเส้นทางต้องการรู้เพียงแค่อุปกรณ์จัดเส้นทางตัวถัดไปเพื่อกลับไปยังต้นทาง โดยไม่ต้องรู้เส้นทางทั้งหมดเพื่อกลับไปยังต้นทาง สามารถแทนต้นไม้เฉพาะสำหรับต้นทางได้ด้วยสัญลักษณ์ *(s, g)*

การจัดเส้นทางสำหรับมัลติคาสในอินเทอร์เน็ต (Multicast Routing in Internet) ได้แก่ ดีวีเอ็มอาร์พี (Distance Vector Multicast Routing Protocol - DVMRP), เอ็มโอเอสพีเอฟ (Multicast Open Shortest Path First – MOSPF), ซีบีที (Core Based Trees - CBT), บีจีเอ็มพี (Border Gateway Multicast Protocol), พิม (Protocol Independent Multicast - PIM)[10] แต่ในที่นี้จะกล่าวถึงพิม เนื่องจากเป็นโพรโทคอลที่ได้รับการพัฒนาขึ้นมาโดยหลีกเลี่ยงข้อบกพร่องที่เกิดจากการใช้โพรโทคอลแบบอื่น และสามารถปรับใช้ได้ตามลักษณะของเครือข่าย โดยใช้พิม-เดนซ์โหมด (PIM - Dense Mode)

- **พิม-เดนซ์โหมด (PIM-Dense Mode)** มีลักษณะการทำงาน ดังนี้

ลักษณะเครือข่าย: สมาชิกของกลุ่มกระจายตัวอยู่ทั่วทั้งเน็ตเวิร์ค จึงมีอุปกรณ์จัดเส้นทางที่เกี่ยวข้องกับเดทาแกรมของการจัดเส้นทางสำหรับมัลติคาส (Routing multicast datagrams) อยู่มาก

การทำงาน: เริ่มจากต้นทางทำการกระจายข้อมูลทุกทิศทาง (flooding) ไปให้อุปกรณ์จัดเส้นทางทุกตัวเช่นเดียวกับขั้นตอนวิธีอาร์พีเอฟ ถ้าอุปกรณ์จัดเส้นทางตัวไหนไม่มีโฮสต์ที่เป็นสมาชิกของ กลุ่มมัลติคาสอยู่เลย อุปกรณ์จัดเส้นทางตัวนั้นก็จะส่งข้อความตัดเส้นทาง (prune message) กลับไปยังต้นทาง แต่ถ้าภายหลังอุปกรณ์จัดเส้นทางมีโฮสต์ที่เป็นสมาชิกของกลุ่มมัลติคาส อุปกรณ์จัดเส้นทางสามารถส่งข้อความเพื่อต่อเส้นทาง (graft message) ไปยังต้นทาง เป็นการขอรับข้อมูลที่มาจากกลุ่มมัลติคาสนี้ใหม่ และเมื่อผ่านไปช่วงระยะเวลาหนึ่งต้นทางก็จะเริ่มการกระจายทุกทิศทางอีกครั้ง

สรุปลักษณะโพรโทคอล: เป็น data-driven ต้นทางจะส่งข้อมูลส่งออกไปทุกที่จนกว่าอุปกรณ์จัดเส้นทางจะส่งข้อความตัดเส้นทาง ไปยังต้นทางเพื่อยกเลิกการรับข้อมูล

ข้อควรพิจารณา:

- อุปกรณ์จัดเส้นทางแต่ละตัวต้องเก็บข้อมูล (s, g) การหาเส้นทางของต้นทางและกลุ่มมัลติคาส ซึ่งอาจส่งผลต่อเนื้อที่สำหรับเก็บข้อมูล

จากทฤษฎีการทำงานของมัลติคาสโพรโทคอล ผู้วิจัยได้นำไปใช้เป็นแนวความคิดในการออกแบบ และปรับให้เหมาะสมสำหรับเครือข่ายบลูทูธ โดยนำหลักการไอจีเอ็มพีไปประยุกต์สำหรับวิธีการจัดการระหว่างมาสเตอร์/สเลฟภายในพิโคเน็ต ตลอดจนการส่งผ่านข้อมูลในสแคทเทอร์เน็ตซึ่งประยุกต์จากหลักการกระจายข้อมูลไปทุกเส้นทาง การตัดเส้นทาง และการต่อเส้นทางของพิม-เดนซ์โหมด เพื่อรองรับการเปลี่ยนแปลงที่เกิดขึ้นในเครือข่าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อ

ในบทนี้กล่าวถึงรายละเอียดการออกแบบเฟรมเวิร์คการติดต่อสื่อสารสำหรับเครือข่ายบลูทูธ โดยมีรูปแบบการติดต่อสื่อสารเป็นการระบุชื่อแทนการติดต่อสื่อสารแบบเดิมที่ใช้เลขที่อยู่บลูทูธในการระบุอุปกรณ์ เนื้อหาประกอบด้วย รูปแบบการระบุชื่อ ลักษณะการติดต่อสื่อสารที่พิจารณา โครงสร้างระบบ ขั้นตอนวิธีการหาเส้นทาง และโครงสร้างของข้อมูลต่างๆ เช่น ข้อมูลสำหรับผู้ใช้ ข้อมูลแจ้งสถานะของระบบแก่ผู้ใช้ ข้อมูลของระบบ เป็นต้น

3.1 รูปแบบการระบุชื่อ

วิทยานิพนธ์นี้สร้างการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมาย ประกอบด้วย 2 ส่วน คือ ส่วนที่ใช้สำหรับระบุชื่อบริการ และระบุชื่อของอุปกรณ์

- ชื่อบริการ (service) เป็นชื่อที่สื่อถึงบริการที่ผู้ใช้รองรับ เช่น ชื่อโปรแกรมประยุกต์ของผู้ใช้
- ชื่อของอุปกรณ์ (subject) เป็นชื่อที่สื่อความหมายเฉพาะแทนการอ้างอิงผู้ใช้หรืออุปกรณ์

ทั้งชื่อบริการและชื่ออุปกรณ์เป็นชื่อที่ผู้ใช้กำหนดเอง ประกอบด้วยสายอักขระที่สามารถแยกย่อยเพื่อความเฉพาะเจาะจงของชื่อให้มากขึ้นโดยใช้เครื่องหมายจุด " . " เช่น “Level1.Sublevel2” สำหรับการส่งผู้ส่งสามารถกำหนดขอบเขตของผู้รับได้โดยใช้เครื่องหมาย “ * ” เช่น “Level1.*” หมายความว่า ผู้รับคือผู้ที่ประกาศชื่อใดๆ ที่ขึ้นต้นด้วย “Level1”

หมายเหตุ ชื่อที่ระบุเป็นชื่อที่ไวต่ออักษรใหญ่เล็ก (case sensitive) เช่น COM และ Com ถือว่าเป็นชื่อที่ต่างกัน เป็นต้น

ตารางที่ 3.1 อักขระพิเศษและความหมาย

อักขระพิเศษ	ความหมาย
.	ใช้แบ่งชื่อออกเป็นส่วนย่อย
*	ใช้กำหนดขอบเขตของชื่อ
~ ` * _ | \ / { } [ ] : ; " ' ,	อักขระพิเศษเหล่านี้ไม่อนุญาตให้ปรากฏอยู่ในส่วนใดๆ ของการระบุชื่อ ยกเว้นการระบุชื่อเพื่อการส่งไปยังปลายทาง สามารถใช้เครื่องหมาย * ในตำแหน่งสุดท้ายของชื่อเพื่อระบุกลุ่มของผู้รับปลายทางได้

ตารางที่ 3.2 ตัวอย่างการตั้งชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ

	ตัวอย่างการระบุชื่อ
ตัวอย่างการระบุชื่อบริการ	Chat
	Printer.Cannon
ตัวอย่างการระบุชื่ออุปกรณ์	NET.BLUENET.C
	GroupName1.MyName
ตัวอย่างการระบุชื่อที่ผิด	this.is.wrong.name.* (เนื่องจาก ใช้อักขระพิเศษในการตั้งชื่อ)
	also.another./:reserved special chars

ตารางที่ 3.3 ตัวอย่างการระบุชื่อเพื่อการส่งไปยังปลายทาง

	ตัวอย่างการระบุชื่อ
ตัวอย่างการระบุชื่อบริการ	Chat
	Printer.*
ตัวอย่างการระบุชื่ออุปกรณ์	NET.BLUENET.C
	GroupName1.*
ตัวอย่างการระบุชื่อที่ผิด	this.is.*.wrong.wildcard.position (เนื่องจาก ในการกำหนดกลุ่มของผู้รับ ให้ใช้เครื่องหมาย * ที่ตำแหน่งสุดท้าย)
	also.another./:reserved special chars

<u>ข้อแนะนำในการตั้งชื่อ</u>

ผู้ใช้ควรตั้งชื่อที่สื่อความหมายเข้าใจง่าย เพื่อความชัดเจนในการใช้อ้างอิงอุปกรณ์บลูทูธ

ไม่ควรใช้อักขระว่างในการระบุชื่อและ/หรือส่วนย่อยของชื่อ

3.2 ลักษณะการติดต่อสื่อสารที่พิจารณา

ลักษณะการติดต่อสื่อสารที่เกิดขึ้นในเครือข่ายโดยทั่วไป รวมทั้งเครือข่ายบลูทูธ มีด้วยกัน 3 ลักษณะ ดังต่อไปนี้

- ยูนิคาส (Unicast): เป็นการติดต่อไปยังโหนดปลายทางตามที่กำหนดไว้เพียงโหนดเดียว เช่น Service = “chat”, Subject = “Group1.A”

- มัลติคาส (Multicast): เป็นการติดต่อไปยังโหนดปลายทางที่เป็นกลุ่มสมาชิกภายในเครือข่าย เช่น Service = “chat”, Subject = “Group1.*”
- บรอดคาส (Broadcast): เป็นการติดต่อไปยังโหนดปลายทางทุกโหนดภายในเครือข่าย เช่น Service = “ * ”, Subject = “ * ”

ในวิทยานิพนธ์นี้ได้พิจารณาลักษณะการติดต่อสื่อสารแบบมัลติคาส เนื่องจากรูปแบบการติดต่อสื่อสารด้วยการระบุชื่อตามจุดมุ่งหมายทำให้สามารถระบุไปยังผู้รับที่ต้องการติดต่อ หรือระบุกลุ่มของผู้รับได้ (ด้วยเครื่องหมาย *) และยังทำให้ลักษณะของการติดต่อสื่อสารครอบคลุมทั้งแบบยูนิคาส และบรอดคาส โดยใช้หลักการมัลติคาสโพรโทคอล สำหรับควบคุมและจัดการในการติดต่อสื่อสารเพื่อทำให้เกิดการใช้แบนด์วิดท์อย่างมีประสิทธิภาพ ทั้งนี้ยังช่วยเพิ่มประสิทธิภาพการใช้พลังงานของอุปกรณ์ได้ดีกว่าแบบยูนิคาสที่การส่งข้อมูลไปยังกลุ่มของผู้รับต้องส่งไปยังผู้รับทีละคนทำให้ต้องส่งข้อมูลหลายๆ ครั้ง และดีกว่าการบรอดคาสเนื่องจากการบรอดคาสทำให้อุปกรณ์ทุกตัวในเครือข่ายได้รับข้อมูลทั้งหมดแม้ว่าอุปกรณ์บางตัวอาจไม่ต้องการรับข้อมูลดังกล่าว

3.3 โครงสร้างของระบบ

แนวคิดในการออกแบบระบบสำหรับวิทยานิพนธ์นี้ได้อาศัยหลักการโดยรวมของมัลติคาสโพรโทคอลมาเป็นแนวทางเพื่อพัฒนากลไกการทำงานของเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมาย รวมทั้งปรับหลักการเพื่อให้เหมาะสมกับเครือข่ายบลูทูธสำหรับรองรับการติดต่อสื่อสารที่ครอบคลุมในเครือข่ายสแคทเทอร์เน็ตอันประกอบด้วยส่วนการจัดการภายในพิโคเน็ตและส่วนการจัดการเพื่อการติดต่อสื่อสารระหว่างสแคทเทอร์เน็ต รายละเอียดการทำงานในแต่ละส่วนเป็นดังนี้

3.3.1 ส่วนการจัดการภายในพิโคเน็ต

ส่วนการจัดการภายในพิโคเน็ตอยู่ภายในขอบเขตระหว่างสเลฟกับมาสเตอร์ภายในพิโคเน็ต วัตถุประสงค์ในการออกแบบส่วนนี้เพื่อให้มีการระบุชื่อสำหรับเรียกอุปกรณ์ที่สื่อความหมายตามที่ผู้ใช้ต้องการ แทนการใช้เลขที่อยู่บลูทูธ และเนื่องจากวิทยานิพนธ์นี้ใช้ชื่อในการอ้างอิงทำให้ผู้ใช้สามารถกำหนดขอบเขตหรือกลุ่มข้อมูลที่ต้องการรับได้ อีกทั้งยังทำให้ผู้ส่งสามารถส่งข้อมูลถึงกลุ่มของผู้รับที่ใช้ชื่อเหมือนกันได้ ดังนั้นการใช้ชื่อจึงเปรียบเสมือนการกำหนดไอพีมัลติคาสให้กับอุปกรณ์ (ซึ่งอยู่ในรูปของชื่อ) เมื่ออุปกรณ์ได้กำหนดชื่อแล้วหลังจากนั้นจะเป็นหน้าที่ของตัวจัดการชื่อซึ่งจะทำหน้าที่เป็นตัวกลางในการตรวจสอบและส่งผ่านข้อมูลไปยังอุปกรณ์ที่มีชื่อสอดคล้องกับ

ส่วนการระบุชื่อของข้อมูลที่ได้รับมา เนื่องจากถ้าให้อุปกรณ์บลูทูธตรวจสอบชื่อเองว่ามีคุณสมบัติสอดคล้องตามชื่อที่ผู้ส่งระบุหรือไม่ จะต้องส่งข้อมูลไปยังอุปกรณ์ทุกตัวเพื่อให้ตรวจสอบตนเอง ซึ่งจะส่งผลต่อความหนาแน่นของข้อมูลในเครือข่าย การออกแบบระบบสำหรับเฟรมเวิร์คประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

1. **<u>บทบาทและหน้าที่ของอุปกรณ์บลูทูธ</u>**

- ตัวจัดการชื่อ (Intentional Name Resolver – INR)
  - หมายถึง มาสเตอร์ เนื่องจากมาสเตอร์แต่ละตัวทำหน้าที่เป็นตัวกลางในการจัดการชื่อเพื่อส่งผ่านข้อมูลไปยังสเลฟ
  - รับฟังการประกาศคุณลักษณะจากสเลฟเพื่อสร้างแผนภาพต้นไม้ของชื่อ (Name-Tree) ดังนั้น ตัวจัดการชื่อแต่ละตัวมีแผนภาพต้นไม้ของชื่อต่างกันขึ้นอยู่กับชื่อที่ได้รับการประกาศจากโหนดปลายทาง
  - ปรับปรุง แผนภาพต้นไม้ของชื่อ ให้เป็นปัจจุบันอยู่เสมอ
- โหนดปลายทาง (Destination Node)
  - หมายถึง ทั้งมาสเตอร์และ/หรือสเลฟ เป็นอุปกรณ์บลูทูธที่รอรับการติดต่อจากโหนดต้นทาง
  - ระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ
  - ประกาศคุณลักษณะของบริการและชื่อเรียกอุปกรณ์ไปยังตัวจัดการชื่อ (การที่โหนดปลายทางประกาศคุณลักษณะไปยังตัวจัดการชื่อแต่ละครั้งหมายถึงการกำหนดชื่อสำหรับโหนดปลายทาง ดังนั้นโหนดปลายทางสามารถมีชื่อได้หลายชื่อและชื่อของโหนดปลายทางสามารถซ้ำกันได้ (เช่นเดียวกับการใช้ไอพีมัลติคาส))
  - เมื่อมีการเปลี่ยนแปลงแก้ไขเกี่ยวกับคุณลักษณะต้องแจ้งให้ตัวจัดการชื่อทราบ
  - ส่งการปรับปรุงไปยังตัวจัดการชื่อภายในระยะเวลาที่กำหนด (Soft State Protocol)

<u>ขั้นตอนการจัดการภายในพิโคเน็ต</u>

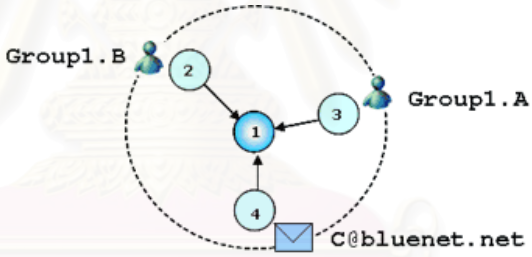
การจัดการที่เกิดขึ้นภายในพิโคเน็ตให้พิจารณาที่มาสเตอร์กับสเลฟ เริ่มจากสเลฟส่งการระบุชื่อสำหรับเป็นชื่อที่ใช้เรียกอุปกรณ์ตามที่สเลฟต้องการไปยังมาสเตอร์ เมื่อมาสเตอร์ได้รับแล้วจะทำการสร้าง แผนภาพต้นไม้ของชื่อ เพื่อใช้เป็นฐานข้อมูลสำหรับใช้ในการพิจารณาอุปกรณ์ปลายทางที่สอดคล้องกับชื่อที่โหนดต้นทางระบุ มาสเตอร์มีหน้าที่จัดการ และปรับปรุงแผนภาพ

ต้นไม้ของชื่อให้มีความถูกต้องอยู่เสมอ สเลฟต้องแจ้งให้มาสเตอร์ทราบเมื่อมีการเปลี่ยนแปลงแก้ไขเกี่ยวกับคุณลักษณะของโหนดปลายทางเอง รวมทั้งต้องส่งการปรับปรุงไปยังมาสเตอร์ภายในระยะเวลาที่กำหนด มิเช่นนั้นจะถือว่าสเลฟไม่ต้องการใช้ชื่อนั้นอีกต่อไป (ซึ่งเป็นลักษณะเช่นเดียวกับ “Soft State Protocol” ที่ต้องมีการปรับปรุงข้อมูลภายในระยะเวลาที่กำหนดไม่เช่นนั้นข้อมูลจะถูกลบออกไป)

3. **<u>การออกแบบในส่วนของการระบุชื่อตามจุดมุ่งหมาย</u>**

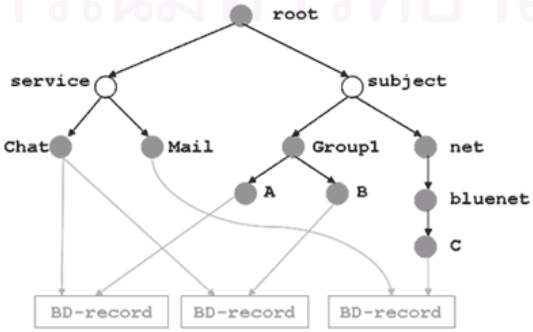
การออกแบบในส่วนนี้พิจารณาจากขั้นตอนการติดต่อสื่อสารในเครือข่ายบลูทูธ และประยุกต์จากหลักการของ [4] สำหรับวิทยานิพนธ์นี้ ในการระบุชื่อที่ใช้เรียกอุปกรณ์ตามที่ต้องการนั้นอุปกรณ์ต้องส่งการระบุชื่อไปยังมาสเตอร์ผ่านทางส่วนต่อประสานของเฟรมเวิร์ค

<u>ตัวอย่าง</u> จากรูปที่ 3.1 อุปกรณ์ที่ 3 เปิดโปรแกรมสนทนาชื่อว่า “Chat” อยู่ภายในกลุ่มที่ชื่อ “Group1” และตั้งชื่อ “A” สิ่งที่สเลฟประกาศไปยังมาสเตอร์เพื่อให้เป็นชื่อสำหรับรอรับการติดต่อจากอุปกรณ์อื่นคือ *Service* = “Chat”, *Subject* = “Group1.A”



รูปที่ 3.1 ตัวอย่างแสดงมุมมองของผู้ใช้ในพิโคเน็ต

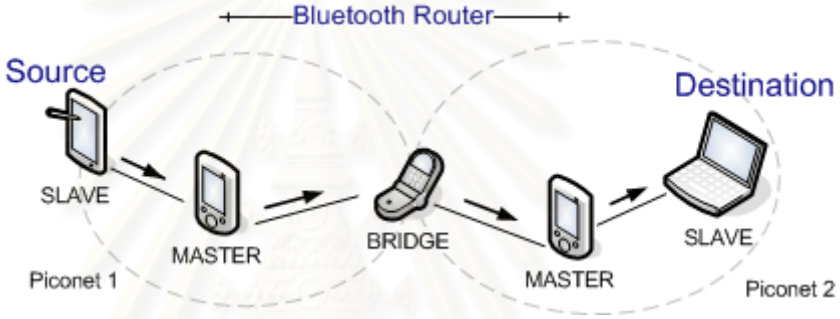
จากรูปที่ 3.1 เมื่อมาสเตอร์ (อุปกรณ์ 1) ได้รับการประกาศชื่อจากสเลฟ (อุปกรณ์ 2 3 และ 4) จะทำการสร้างเป็นแผนภาพต้นไม้ของชื่อโดยจัดเก็บไว้ในลักษณะของลำดับชั้น ดังรูปที่ 3.2 ซึ่งโหนดใบของแผนภาพต้นไม้จะอ้างอิงไปยังระเบียนเลขที่อยู่บลูทูธ (BD-record หมายถึงเลขที่อยู่บลูทูธของอุปกรณ์ที่สอดคล้องกับชื่อ)



รูปที่ 3.2 ตัวอย่างแผนภาพต้นไม้ของชื่อ (Name-Tree)

3.3.2 ส่วนการจัดการเพื่อการติดต่อสื่อสารระหว่างกลุ่มของพิโคเน็ต

การติดต่อสื่อสารระหว่างกลุ่มของพิโคเน็ตทำให้เกิดเครือข่ายที่เรียกว่าสแคทเทอร์เน็ตขึ้นได้นั้นต้องอาศัยมาสเตอร์และบริดจ์ เนื่องจากการรับ-ส่งข้อมูลภายในพิโคเน็ตต้องส่งผ่านมาสเตอร์ ส่วนบริดจ์ทำหน้าที่เชื่อมต่อเพื่อส่งผ่านข้อมูลระหว่างพิโคเน็ต (โดยที่บริดจ์อาจเป็นมาสเตอร์/สเลฟ หรือสเลฟ/สเลฟ) รูปที่ 3.3 ทั้งมาสเตอร์และบริดจ์จึงมีหน้าที่รับผิดชอบในการส่งต่อข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง เช่นเดียวกับอุปกรณ์จัดเส้นทาง (Router) ที่พบในเครือข่ายอินเตอร์เน็ต ดังนั้นการออกแบบขั้นตอนวิธีการจัดเส้นทาง (Routing Algorithm) รวมทั้งการประสานงานระหว่างอุปกรณ์เหล่านี้จึงเป็นส่วนสำคัญอีกส่วนหนึ่งของเฟรมเวิร์ค



รูปที่ 3.3 ตัวอย่างการส่งผ่านข้อมูลระหว่างพิโคเน็ต

วัตถุประสงค์ในการออกแบบส่วนนี้เพื่อใช้การระบุชื่อตามจุดมุ่งหมายในการรองรับให้ผู้ใช้สามารถส่งข้อมูลไปยังปลายทางตามที่ระบุได้โดยขึ้นอยู่กับชื่อที่ผู้ใช้ระบุมา ทั้งนี้ยังสามารถกำหนดขอบเขตของผู้รับปลายทางได้ (ด้วยเครื่องหมาย *) โดยที่ผู้ใช้ไม่ต้องทราบตำแหน่งของอุปกรณ์ปลายทางว่าอยู่ที่ใด อุปกรณ์จัดเส้นทางบลูทูธจะทำหน้าที่เป็นตัวกลางในการส่งข้อมูลไปยังปลายทางที่ตรงตามที่ผู้ใช้ต้องการ

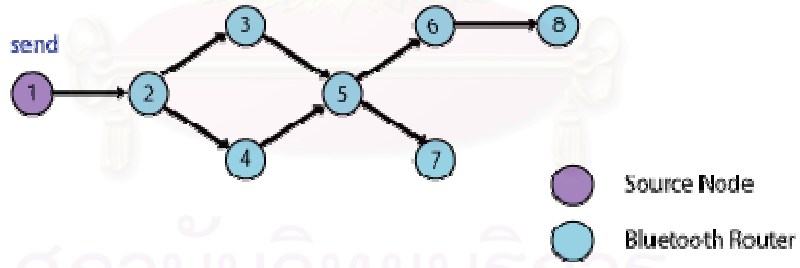
1. **<u>บทบาทและหน้าที่ของอุปกรณ์บลูทูธ</u>**

- โหนดต้นทาง (Source Node)
  - หมายถึง อุปกรณ์บลูทูธใดๆ ไม่ว่าจะเป็นมาสเตอร์ สเลฟ บริดจ์ ที่ต้องการติดต่อไปยังโหนดปลายทาง
  - ระบุชื่อบริการ และ/หรือชื่ออุปกรณ์ของโหนดปลายทาง ที่ต้องการติดต่อ
- อุปกรณ์จัดเส้นทางบลูทูธ (Bluetooth Router)
  - หมายถึง มาสเตอร์ และบริดจ์ ซึ่งเป็นอุปกรณ์ทำหน้าที่ในการจัดเส้นทาง

- อุปกรณ์จัดเส้นทางแต่ละตัวเก็บข้อมูลเกี่ยวกับเส้นทาง (s, n) (สัญลักษณ์ (s, n) แทนเส้นทางที่อุปกรณ์จัดเส้นทางแต่ละตัวใช้ในการส่งผ่านข้อมูลจากโหนดต้นทาง (s) ไปยังอุปกรณ์ที่มีชื่อสอดคล้องกับชื่อที่โหนดต้นทางระบุหรือโหนดปลายทาง (n) ประกอบด้วยเส้นทางที่ได้รับข้อมูลเข้ามาและเส้นทางที่ใช้ส่งข้อมูลต่อไปยังอุปกรณ์จัดเส้นทางข้างเคียง และสัญลักษณ์ [s, n] แทนข้อมูลใดๆ ที่โหนดต้นทาง (s) ระบุชื่อโหนดปลายทางเป็น n)
- ฟังก์ชันการทำงานประกอบด้วย กระบวนการกระจายข้อมูล/กระบวนการตัดเส้นทาง/กระบวนการต่อเส้นทาง (flood/prune/graft)
- เรียนรู้อุปกรณ์จัดเส้นทางบลูทูธอื่นๆ ที่อยู่ใกล้เคียง
- จัดเก็บเส้นทางที่ใช้ส่งข้อมูลไว้ในตารางการจัดเส้นทาง (Routing Table)

2. **ขั้นตอนวิธีการหาเส้นทาง (Routing Algorithm)** ประกอบด้วยฟังก์ชันการทำงานเพื่อให้เกิดการเชื่อมต่อเส้นทาง (s, n) สำหรับการส่งผ่านข้อมูล ซึ่งเป็นการเรียนรู้เส้นทางส่งข้อมูลที่มาจากโหนดต้นทาง (s) ไปยังโหนดปลายทาง (n) สำหรับอุปกรณ์จัดเส้นทางบลูทูธแต่ละตัวดังต่อไปนี้

- **กระบวนการกระจายข้อมูล** (Flood)



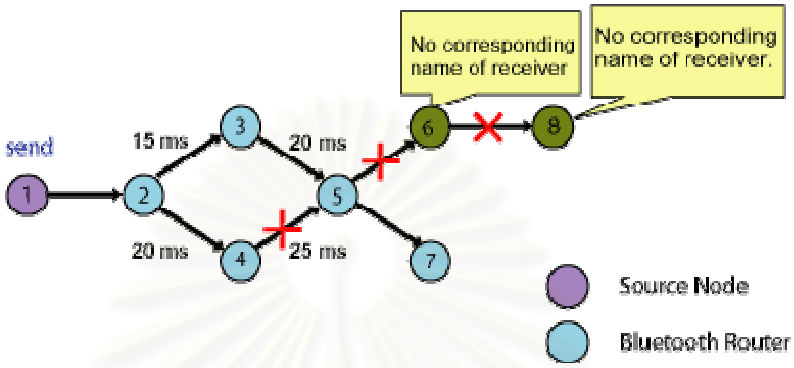
รูปที่ 3.4 กระบวนการกระจายข้อมูล

1) เมื่ออุปกรณ์จัดเส้นทางทั้งมาสเตอร์และบริดจ์ได้รับข้อมูลจากโหนดต้นทาง (s) จะส่งข้อมูลต่อไปยังอุปกรณ์จัดเส้นทางทุกตัวที่อยู่ใกล้เคียง โดยการกระจายข้อมูลออกไปทุกเส้นทาง (flooding) ยกเว้นเส้นทางเดิมที่มันได้รับข้อมูลมา ดังรูปที่ 3.4

2) ถ้าอุปกรณ์จัดเส้นทางนั้นเป็นมาสเตอร์ เมื่อได้รับข้อมูลเข้ามา จะทำการตรวจสอบว่าโหนดปลายทาง (n) ที่ระบุไว้ตรงกับอุปกรณ์ใดที่อยู่ภายในพิโคเน็ตของตนหรือไม่ถ้ามีก็จะส่งข้อมูลไปยังโหนดปลายทางตัวนั้น แล้วจึงส่งต่อข้อมูลที่ได้รับมาไปยังอุปกรณ์จัดเส้นทางทุกตัวที่อยู่ข้างเคียง

3) ถ้าอุปกรณ์จัดเส้นทางที่ได้รับข้อมูลเข้ามานั้นเป็นบริดจ์ ก็จะส่งต่อข้อมูล ไปยังอุปกรณ์จัดเส้นทางทุกตัวที่อยู่ข้างเคียง แต่ไม่มีหน้าที่ในการตรวจสอบปลายทางดังเช่นอุปกรณ์จัดเส้นทางที่เป็น มาสเตอร์

- **กระบวนการตัดเส้นทาง** (Prune) แบ่งได้เป็น 2 กรณี

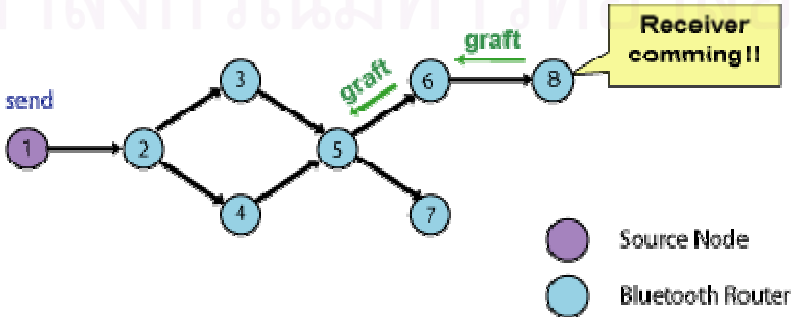


รูปที่ 3.5 กระบวนการตัดเส้นทาง

กรณีที่1) อุปกรณ์จัดเส้นทางทั้งที่เป็นมาสเตอร์และบริดจ์ (จากรูปที่ 3.5 พิจารณาอุปกรณ์ 5) ซึ่งได้รับข้อมูลเดิมซ้ำจากหลายเส้นทาง ก็จะทำการส่งข้อความตัดเส้นทาง (prune message) ไปยังเส้นทางที่ไม่ต้องการ ยกเว้นเส้นทางที่ทำให้ได้รับข้อมูลมาถึงอุปกรณ์จัดเส้นทางเร็วที่สุด ดังนั้นในครั้งต่อๆ ไปอุปกรณ์จัดเส้นทางที่ได้รับข้อความตัดเส้นทาง (อุปกรณ์ 4) จะไม่ส่งข้อมูล [s, n] ต่อไปยังอุปกรณ์จัดเส้นทางที่ได้ตัดเส้นทางออกไป (อุปกรณ์ 5)

กรณีที่2) อุปกรณ์จัดเส้นทางที่เป็นมาสเตอร์ (จากรูปที่ 3.5 พิจารณาอุปกรณ์ 8) เมื่อไม่ต้องการรับข้อมูล [s, n] อันเนื่องมาจากไม่มีอุปกรณ์ใดภายใต้พิโคเน็ตของมันเอง รวมทั้งในพิโคเน็ตต่อๆ ไปที่ใช้ชื่อตามที่โหนดต้นทางระบุ จะส่งข้อความเพื่อตัดเส้นทางที่ส่งข้อมูลดังกล่าวเข้ามา ส่วนอุปกรณ์จัดเส้นทางที่เป็นบริดจ์ เมื่อไม่ต้องการรับข้อมูล [s, n] (จากรูปที่ 3.5 เช่นอุปกรณ์ 6) อันเนื่องมาจากไม่มีอุปกรณ์จัดเส้นทางบลูทูธใดในพิโคเน็ตต่อๆ ต้องการข้อมูลนั้น ก็จะส่งข้อความเพื่อตัดเส้นทางเช่นเดียวกัน

- **กระบวนการต่อเส้นทาง**(Graft) แบ่งได้เป็น 2 กรณี



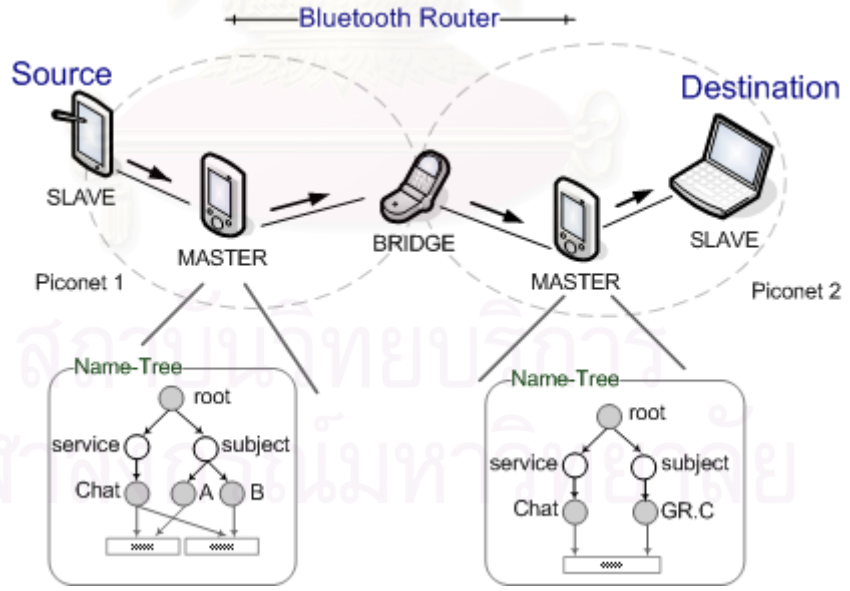
รูปที่ 3.6 กระบวนการต่อเส้นทาง

กรณีที่1) อุปกรณ์จัดเส้นทางที่เป็นมาสเตอร์ต้องการรับข้อมูลจากที่ส่งมาจาก [s, n] อีกครั้ง (รูปที่ 3.6 อุปกรณ์ 8) เนื่องจากมีโหนดที่เป็นสมาชิกของกลุ่มดังกล่าวอยู่ภายใต้พิโคเน็ตของมันเอง และ/หรือในพิโคเน็ตต่อๆ ไป จะส่งข้อความต่อเส้นทางไปยังเส้นทางเดิมที่ส่ง [s, n] เข้ามา

กรณีที่2) อุปกรณ์จัดเส้นทางที่เป็นบริดจ์ต้องการรับข้อมูลจากที่ส่งมาจาก [s, n] อีกครั้ง (รูปที่ 3.6อุปกรณ์ 6) เนื่องจากอุปกรณ์จัดเส้นทางอื่นๆ ที่อยู่ถัดไปต้องการรับข้อมูลดังกล่าว จะส่งข้อความต่อเส้นทางไปยังเส้นทางเดิมที่เคยส่ง [s, n] เข้ามา

- **กระบวนการกระจายข้อมูลอีกครั้ง** (Reflooding) เมื่อผ่านไปช่วงเวลาหนึ่งก็จะเริ่มกระบวนการกระจายข้อมูลอีกครั้ง อุปกรณ์จัดเส้นทางจะส่งข้อมูลไปยังอุปกรณ์จัดเส้นทางข้างเคียงทุกตัว เพื่อรองรับการเปลี่ยนแปลงที่เกิดขึ้นในเครือข่าย เส้นทางเพื่อใช้ในการส่งผ่านข้อมูลระหว่างอุปกรณ์จัดเส้นทางบลูทูธจะได้รับการพิจารณาใหม่

3.3.3 **สรุประบบโดยรวม** เมื่อรวมส่วนการจัดการภายในพิโคเน็ต (รายละเอียดในหัวข้อ 3.3.1) และส่วนการจัดการเพื่อการติดต่อสื่อสารระหว่างกลุ่มของพิโคเน็ต (รายละเอียดในหัวข้อ 3.3.2) จะได้ภาพรวมระบบการทำงานของเฟรมเวิร์คทั้งหมด



รูปที่ 3.7 สรุประบบโดยรวม

รูปที่ 3.7 โหนดปลายทาง (Destination) ประกาศชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ (เช่น service = “Chat”, subject = “GR.C”) ไปยังตัวจัดการชื่อ (มาสเตอร์) จากนั้นตัวจัดการชื่อก็จะสร้างแผนภาพต้นไม้ของชื่อ ซึ่งประกอบด้วยชื่อและเลขที่อยู่บลูทูธของ

อุปกรณ์ของโหนดปลายทาง เมื่อโหนดต้นทาง (Source) ต้องการติดต่อไปยังโหนดปลายทางก็จะระบุชื่อของโหนดปลายทางที่ต้องการติดต่อ จากตัวอย่างในรูปที่ 3.7 โหนดต้นทางเป็นสเลฟ ดังนั้นโหนดต้นทางส่งข้อมูลไปยังมาสเตอร์ในพิโคเน็ตของตนเอง เมื่อมาสเตอร์ได้รับข้อมูลก็จะทำหน้าที่เป็นตัวจัดการชื่อ เพื่อตรวจสอบว่ามีอุปกรณ์ใดในพิโคเน็ตตนเองมีชื่อสอดคล้องกับชื่อที่โหนดต้นทางระบุไว้หรือไม่ ถ้ามีก็จะส่งข้อมูลไปยังอุปกรณ์ตัวนั้น พร้อมทั้งรับหน้าที่เป็นอุปกรณ์จัดเส้นทางโดยส่งต่อข้อมูลไปยังอุปกรณ์จัดเส้นทางอื่นที่อยู่ข้างเคียง (ซึ่งหมายถึงบริดจ์ จากนั้นบริดจ์จะทำการส่งข้อมูลต่อไปยังมาสเตอร์ในพิโคเน็ตถัดไป) ส่วนกระบวนการจัดการเส้นทางที่ใช้ในการส่งผ่านข้อมูลประกอบด้วย การกระจายข้อมูล การตัดเส้นทาง การต่อเส้นทางและกระบวนการกระจายข้อมูลอีกครั้งของอุปกรณ์จัดเส้นทาง เมื่อรวมการทำงานในส่วนการจัดการในพิโคเน็ตและการจัดการเพื่อการติดต่อสื่อสารระหว่างพิโคเน็ต ทำให้สามารถส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางทุกตัวในเครือข่ายบลูทูธที่มีชื่อสอดคล้องกับชื่อที่ระบุโดยโหนดต้นทางได้

3.3.4 **สรุปบทบาทและหน้าที่ของอุปกรณ์บลูทูธ** ตามที่ได้กล่าวมาในส่วนการจัดการพิโคเน็ต และสแคทเทอร์เน็ตสามารถสรุปบทบาท และหน้าที่ของอุปกรณ์บลูทูธ ดังแสดงในตารางที่ 3.4

ตารางที่ 3.4 บทบาท และหน้าที่ของอุปกรณ์บลูทูธในเครือข่าย

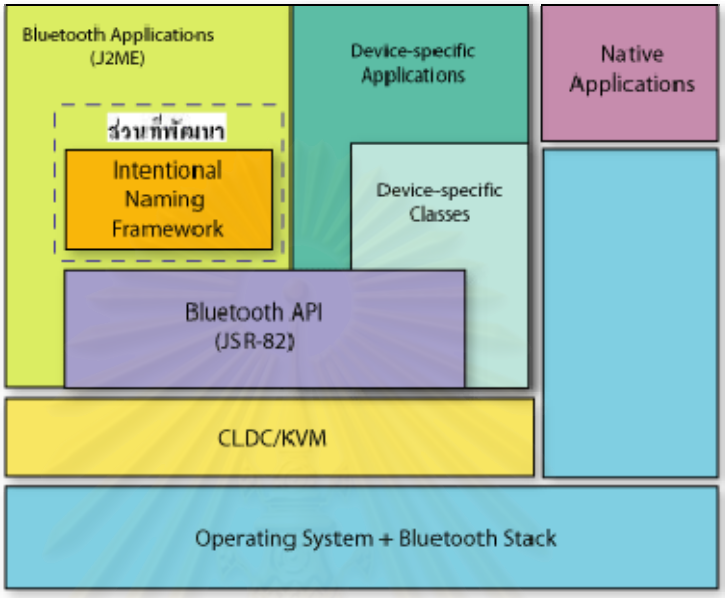
	มาสเตอร์	**บริดจ์**	**สเลฟ**
โหนดต้นทาง (Source Node)	*	*	*
โหนดปลายทาง (Destination Node)	*	*	*
อุปกรณ์จัดเส้นทางบลูทูธ (Bluetooth Router)	*	*	
ตัวจัดการชื่อ (Intentional Name Resolver)	*		

3.4 การออกแบบเฟรมเวิร์ค

3.4.1 สภาวะแวดล้อมการทำงานของเฟรมเวิร์ค

เฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อตามจุดมุ่งหมายสำหรับเครือข่ายบลูทูธภายในวิทยานิพนธ์นี้ได้รับการพัฒนาขึ้นภายใต้สภาวะแวดล้อมของเจเอสอาร์-82 (Java Specification Request 82 – JSR-82) ซึ่งเป็นจาวาเอพีไอ (JavaAPI) สำหรับบลูทูธและเป็นมาตรฐานเปิดสำหรับพัฒนาโปรแกรมประยุกต์ผ่านบลูทูธรุ่นที่ 1.1 และใช้ภาษาจาวา (Java) รุ่นที่ 1.4 ในการ

พัฒนา เฟรมเวิร์คที่พัฒนาขึ้นรองรับโปรแกรมประยุกต์ที่พัฒนาขึ้นด้วยเจทูเอ็มอี (J2ME) ดังรูปที่ 3.8 ได้รับการทดสอบการทำงานผ่านตัวจำลองการทำงาน (Emulator) เจทูเอ็มอี ไวร์เลส ทูลคิท (J2ME Wireless Toolkit) รุ่นที่ 2.2 [11]



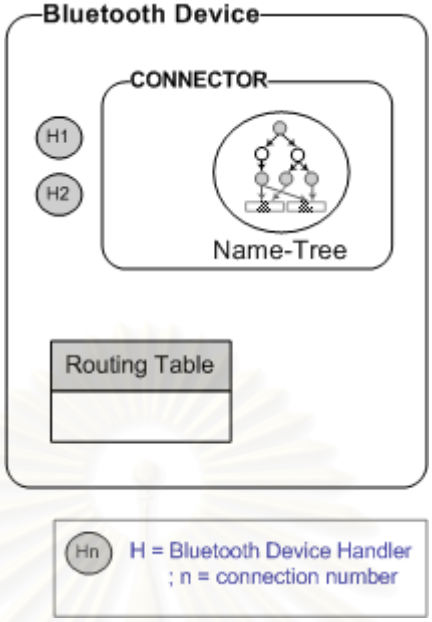
รูปที่ 3.8 แสดงส่วนประกอบสภาวะแวดล้อมของเฟรมเวิร์ค

3.4.2 **องค์ประกอบของอุปกรณ์บลูทูธ**

ในวิทยานิพนธ์นี้ได้ออกแบบองค์ประกอบต่างๆ ของอุปกรณ์บลูทูธทั้งอุปกรณ์ที่เป็นมาสเตอร์ สเลฟ สเลฟ/สเลฟบริดจ์ และมาสเตอร์/สเลฟบริดจ์ ขึ้นตามข้อกำหนดของบลูทูธร่วมกับรูปแบบการติดต่อสื่อสารโดยการระบุชื่อตามที่ได้ออกแบบไว้ มีรายละเอียดดังต่อไปนี้

- **องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์**

การกำหนดบทบาทอุปกรณ์บลูทูธเป็นมาสเตอร์จะถูกกำหนดขึ้นโดยผู้ใช้ องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์จึงมีความเกี่ยวข้องกับหน้าที่รับผิดชอบ อันประกอบด้วย การเป็นตัวจัดการชื่อ อุปกรณ์จัดเส้นทางบลูทูธ รวมทั้งการเป็นโหนดต้นทางและโหนดปลายทาง



รูปที่ 3.9 องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์

จากรูปที่ 3.9 องค์ประกอบแต่ละส่วนอธิบายหน้าที่การทำงานได้ดังต่อไปนี้

1) **อุปกรณ์บลูทูธของเฟรมเวิร์ค** (Bluetooth Device) เป็นวัตถุที่ใช้เป็นตัวแทนอุปกรณ์บลูทูธจริง ภายในวัตถุนี้มีการกำหนดค่าพื้นฐานของอุปกรณ์มาจาก LocalDevice ของเจเอสอาร์-82 ได้แก่ เลขที่อยู่บลูทูธ รวมทั้งมีฟังก์ชันการทำงานพื้นฐานของอุปกรณ์บลูทูธ อาทิเช่น การสร้างมาสเตอร์และ/หรือสเลฟ การส่งข้อมูล การรับข้อมูล เป็นต้น

2) **มาสเตอร์ หรือ อุปกรณ์สร้างการติดต่อ** (Bluetooth Device Connector) เนื่องจากอุปกรณ์บลูทูธมาสเตอร์เป็นผู้สร้างการติดต่อสื่อสารขึ้นเป็นเครือข่ายพิโคเน็ต โดยค้นหาอุปกรณ์สเลฟและสร้างการติดต่อสื่อสารไปยังสเลฟ โดยใช้ตัวแทนการค้นหา (Discovery Agent) จากเจเอสอาร์-82 สำหรับการค้นหาอุปกรณ์และบริการจากสเลฟที่พบ

3) **ตัวจัดการการเชื่อมต่อ** (Bluetooth Device Handler) อุปกรณ์บลูทูธมาสเตอร์ที่สร้างการติดต่อสื่อสารกับสเลฟได้แล้วจะทำการสร้างตัวจัดการการเชื่อมต่อ หน้าที่หลักของตัวจัดการการเชื่อมต่อคือ การจัดการเกี่ยวกับการรับ-ส่งข้อมูลกับอุปกรณ์ข้างเคียง (สเลฟ) ที่มาสเตอร์เชื่อมต่ออยู่ด้วย ตลอดจนตรวจสอบอุปกรณ์ข้างเคียงว่ายังคงอยู่หรือไม่ จากรูปที่ 3.9 ตัวอย่าง H1, H2 หมายถึงตัวจัดการการเชื่อมต่อไปยังสเลฟตัวที่ 1 และ 2 ตามลำดับ โดยที่ H1 และ H2 ต่างทำหน้าที่จัดการการเชื่อมต่อของตนเองและไม่เกี่ยวข้องกัน

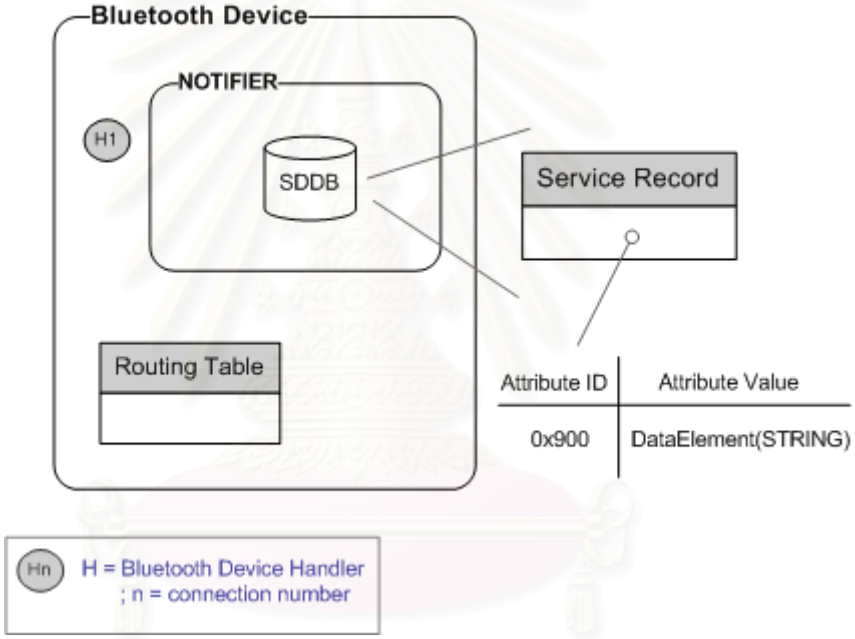
4) **ตารางการจัดเส้นทาง** (Routing Table) ในการส่งข้อมูลจะใช้ตารางการจัดเส้นทาง เพื่อตรวจสอบเส้นทางที่ใช้ส่งข้อมูล มีการตรวจสอบเส้นทางจากตารางจัดเส้นทางทั้งในขณะที่อุปกรณ์

เป็นผู้ส่งข้อมูลเอง (รายละเอียดโหนดต้นทาง ในหัวข้อ 3.3.2) และขณะทำหน้าที่เป็นอุปกรณ์จัดเส้นทางบลูทูธ (รายละเอียดการทำงานของอุปกรณ์จัดเส้นทางบลูทูธ กล่าวไว้ในหัวข้อ 3.3.2)

5) **แผนภาพต้นไม้ของชื่อ** (Name-Tree) เป็นองค์ประกอบที่สำคัญสำหรับอุปกรณ์บลูทูธมาสเตอร์ สำหรับจัดเก็บชื่อและค้นหาเลขที่อยู่บลูทูธของอุปกรณ์ปลายทางที่สอดคล้องกับชื่อ (รายละเอียดตัวจัดการชื่อ ในหัวข้อ 3.3.1)

- **องค์ประกอบของอุปกรณ์บลูทูธสเลฟ**

อุปกรณ์บลูทูธที่เป็นสเลฟถูกกำหนดบทบาทขึ้นโดยผู้ใช้ หน้าที่ของอุปกรณ์คือเป็นโหนดต้นทางและโหนดปลายทาง



รูปที่ 3.10 องค์ประกอบของอุปกรณ์บลูทูธสเลฟ

องค์ประกอบของอุปกรณ์บลูทูธสเลฟ แสดงในรูปที่ 3.10 อธิบายหน้าที่ของแต่ละส่วนมีดังต่อไปนี้

1) **อุปกรณ์บลูทูธของเฟรมเวิร์ค** (Bluetooth Device) เป็นวัตถุที่ใช้เป็นตัวแทนอุปกรณ์บลูทูธ รวมทั้งมีฟังก์ชันการทำงานพื้นฐานของอุปกรณ์บลูทูธ อาทิเช่น การสร้างมาสเตอร์และ/หรือสเลฟ การส่งข้อมูล การรับข้อมูล เป็นต้น

2) **สเลฟ หรือ อุปกรณ์รอรับการติดต่อ** (Bluetooth Device Notifier) สเลฟอาจทำการประกาศชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการไว้ เพื่อรอรับการติดต่อสื่อสารจากมาสเตอร์

3) **ตัวจัดการการเชื่อมต่อ** (Bluetooth Device Handler) เมื่อสเลฟได้รับการเชื่อมต่อจากมาสเตอร์แล้วก็จะสร้างวัตถุตัวจัดการการเชื่อมต่อขึ้นเพื่อใช้รับ-ส่งข้อมูลกับอุปกรณ์ข้างเคียง (มาสเตอร์) ที่เชื่อมต่ออยู่ด้วย ตลอดจนตรวจสอบอุปกรณ์ข้างเคียงว่ายังคงอยู่หรือไม่ จากตัวอย่างรูปที่ 3.10 วัตถุ H1 หมายถึงตัวจัดการการติดต่อสื่อสารกับมาสเตอร์ เนื่องจากอุปกรณ์ที่มีสถานะเป็นสเลฟจะมีการเชื่อมต่อกับมาสเตอร์เพียงตัวเดียวเท่านั้น ดังนั้นตัวจัดการการเชื่อมต่อของอุปกรณ์บลูทูธสเลฟจึงมีเพียงหนึ่งตัว

4) **ตารางการจัดเส้นทาง** (Routing Table) ในการส่งข้อมูลจะใช้ตารางการจัดเส้นทาง (Routing Table) เพื่อตรวจสอบเส้นทางที่ใช้ส่งข้อมูล

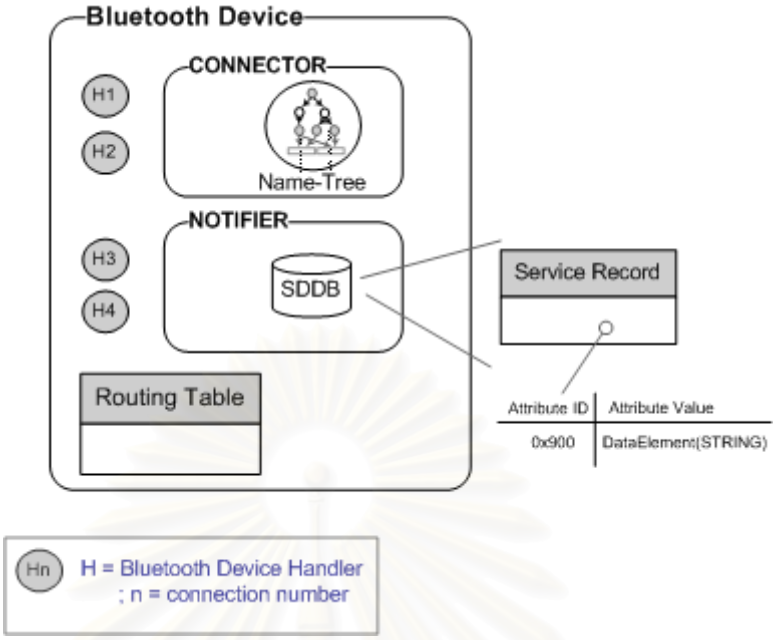
5) **ฐานข้อมูลบริการ** (SDDB – Service Discovery Database) เพื่อให้มาสเตอร์สามารถตรวจสอบชื่อกำหนดขอบเขตของสิ่งที่สเลฟจะรอให้บริการได้ก่อนที่จะสร้างการเชื่อมต่อมายังสเลฟ ในวิทยานิพนธ์นี้จึงเพิ่มฐานข้อมูลบริการ จากเจเอสอาร์-82 เข้าไปในสเลฟ หน้าที่หลักของฐานข้อมูลบริการคือ การจัดเก็บชื่อ (เช่น service = “Chat”, subject = “Group1.A”) โดยแต่ละชื่อที่ผู้ใช้กำหนดจะถูกเก็บเป็นสายอักขระไว้ในระเบียนบริการ (Service Record) ทำให้มาสเตอร์สามารถตรวจสอบข้อมูลเกี่ยวกับบริการของสเลฟได้ วัตถุประสงค์หลักในการนำฐานข้อมูลบริการมาใช้เพื่อให้มาสเตอร์สามารถพิจารณาเลือกอุปกรณ์ที่มาเข้าร่วมอยู่ในพิโคเน็ตจากกลุ่มของอุปกรณ์ที่มีบริการประเภทเดียวกัน เนื่องจากจะช่วยเพิ่มประสิทธิภาพการใช้แบนด์วิดท์ของเครือข่ายบลูทูธได้

- **องค์ประกอบของอุปกรณ์บลูทูธสเลฟ/สเลฟบริดจ์**

อุปกรณ์ที่เป็นสเลฟจะถูกเปลี่ยนสถานะใหม่เป็นสเลฟ/สเลฟบริดจ์ ก็ต่อเมื่อมีมาสเตอร์ที่สร้างการติดต่อเข้ามายังสเลฟมากกว่าหนึ่งตัว โดยจำนวนของตัวจัดการการเชื่อมต่อของสเลฟจะเพิ่มขึ้นตามจำนวนของมาสเตอร์ และสเลฟที่มีการเปลี่ยนแปลงสถานะต้องส่งสถานะใหม่ไปแจ้งให้มาสเตอร์ทุกครั้ง ดังนั้น สเลฟ/สเลฟบริดจ์มีองค์ประกอบของอุปกรณ์เช่นเดียวกับสเลฟ เพียงแต่สเลฟ/สเลฟบริดจ์จะมีจำนวนตัวจัดการการเชื่อมต่อมากกว่าหนึ่งตัว ตัวจัดการการเชื่อมต่อแต่ละตัวต่างทำหน้าที่จัดการการติดต่อสื่อสารของตนเองและไม่เกี่ยวข้องกัน

- **องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์/สเลฟบริดจ์**

เพื่อให้อุปกรณ์บลูทูธสามารถเป็นได้ทั้งมาสเตอร์และสเลฟ จึงได้ออกแบบให้ภายในอุปกรณ์มีทั้งวัตถุมาสเตอร์และสเลฟ ประกอบด้วย



รูปที่ 3.11 องค์ประกอบของอุปกรณ์บลูทูธมาสเตอร์/สเลฟบริดจ์

1) **มาสเตอร์ หรือ ตัวสร้างการติดต่อ** เช่นเดียวกับการทำงานของอุปกรณ์มาสเตอร์ คือ สามารถค้นหาและติดต่อไปยังอุปกรณ์สเลฟตัวอื่นได้ รวมทั้งมีหน้าที่ในการจัดการแผนภาพต้นไม้ของชื่อ มุมมองของสเลฟอื่นๆ ที่เชื่อมต่ออยู่ด้วยจะรับว่าอุปกรณ์นี้เป็นมาสเตอร์

2) **สเลฟ หรือ อุปกรณ์รอรับการติดต่อ** รอรับการติดต่อสื่อสารจากมาสเตอร์ตัวอื่น และทุกครั้งที่มีการเปลี่ยนแปลงสถานะต้องส่งสถานะไปแจ้งมาสเตอร์ที่เชื่อมต่อด้วย รวมทั้งมีฐานข้อมูลบริการ กล่าวได้ว่าการทำงานในส่วนนี้เหมือนกับอุปกรณ์ที่เป็นสเลฟ

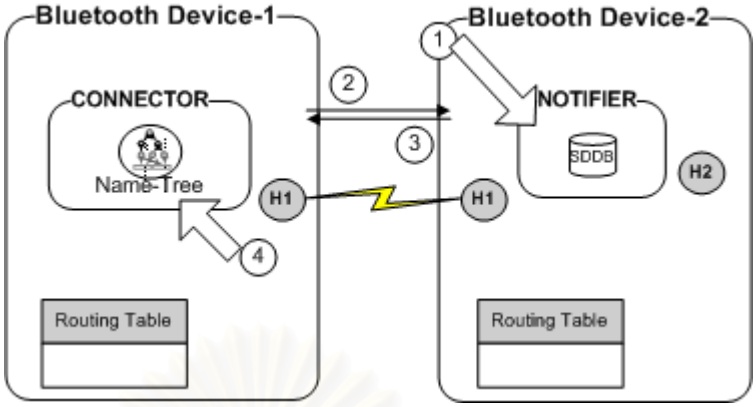
3) **ตัวจัดการการเชื่อมต่อ** (Bluetooth Device Handler) จากรูปที่ 3.11 H1, H2 เป็นตัวจัดการการเชื่อมต่อของวัตถุมาสเตอร์ ทำหน้าที่จัดการการเชื่อมต่อกับอุปกรณ์สเลฟรอบข้าง ส่วน H3, H4 เป็นตัวจัดการการเชื่อมต่อของวัตถุสเลฟ ทำหน้าที่จัดการการเชื่อมต่อกับอุปกรณ์มาสเตอร์ที่เข้ามาสร้างการติดต่อสื่อสารด้วย

4) **ตารางการจัดเส้นทาง** (Routing Table) การตรวจสอบเส้นทางสำหรับส่งข้อมูล ทั้งวัตถุมาสเตอร์และสเลฟใช้ตารางจัดเส้นทางเดียวกัน

3.4.3 องค์ประกอบการสร้างเครือข่ายการติดต่อสื่อสาร

- **การสร้างเครือข่ายพิโคเน็ต**

อธิบายขั้นตอนการติดต่อสื่อสารและองค์ประกอบของอุปกรณ์บลูทูธที่เกี่ยวข้องในการสร้างการติดต่อสื่อสารระหว่างมาสเตอร์กับสเลฟ จากตัวอย่างรูปที่ 3.12 ดังต่อไปนี้



รูปที่ 3.12 ตัวอย่างการสร้างพิโคเน็ต

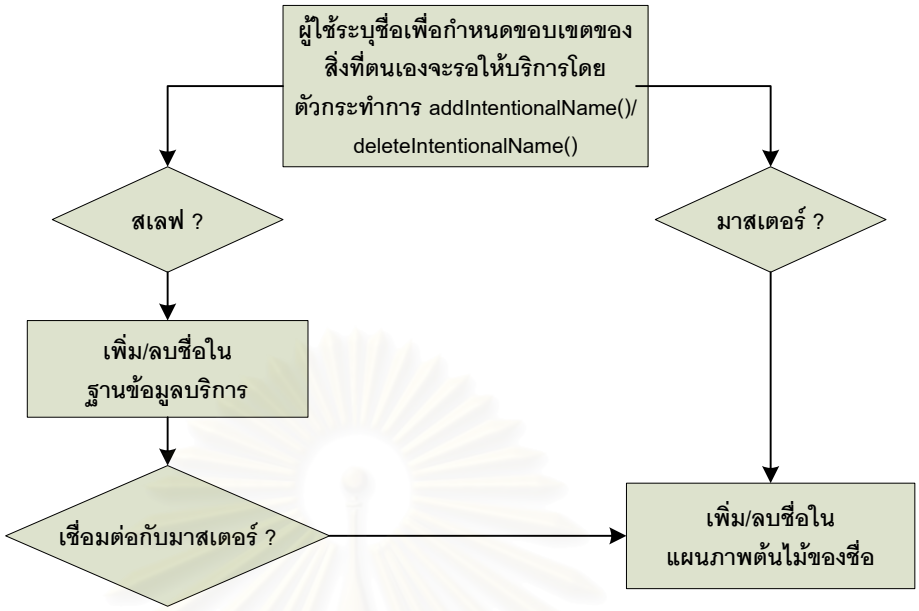
ขั้นตอนที่ 1 อุปกรณ์สเลฟ (Bluetooth Device-2 จากรูปที่ 3.12) เปิดการเชื่อมต่อไว้เพื่อให้มาสเตอร์สามารถค้นหาและติดต่อเข้ามาได้ และในระหว่างรอการติดต่อจากมาสเตอร์สเลฟสามารถประกาศชื่อได้ โดยชื่อจะถูกเก็บไว้ในฐานข้อมูลบริการ

ขั้นตอนที่ 2 มาสเตอร์ (Bluetooth Device-1 จากรูปที่ 3.12) ทำการค้นหาอุปกรณ์รวมทั้งค้นหาบริการจากฐานข้อมูลบริการของฝั่งสเลฟ

ขั้นตอนที่ 3 ถ้าสเลฟมีข้อมูลเกี่ยวกับการระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการของสเลฟจะถูกส่งไปยังมาสเตอร์

ขั้นตอนที่ 4 หลังจากที่มาสเตอร์เชื่อมต่อไปยังสเลฟได้ ก็จะนำชื่อที่พบจากฐานข้อมูลบริการมาจัดเก็บเพิ่มเข้าไปในแผนภาพต้นไม้ จากนั้นทั้งมาสเตอร์และสเลฟ สร้างตัวจัดการการสื่อสาร (H1) ขึ้นเพื่อทำหน้าที่ดูแลการเชื่อมต่อ โดยที่ H1 ทางฝั่งมาสเตอร์จะรวบรวมข้อมูล (ได้แก่ สถานะ เลขที่อยู่บลูทูธ) ของสเลฟเก็บไว้ ในทางกลับกัน H1 ทางฝั่งสเลฟก็เก็บข้อมูลเกี่ยวกับมาสเตอร์ไว้เช่นกัน

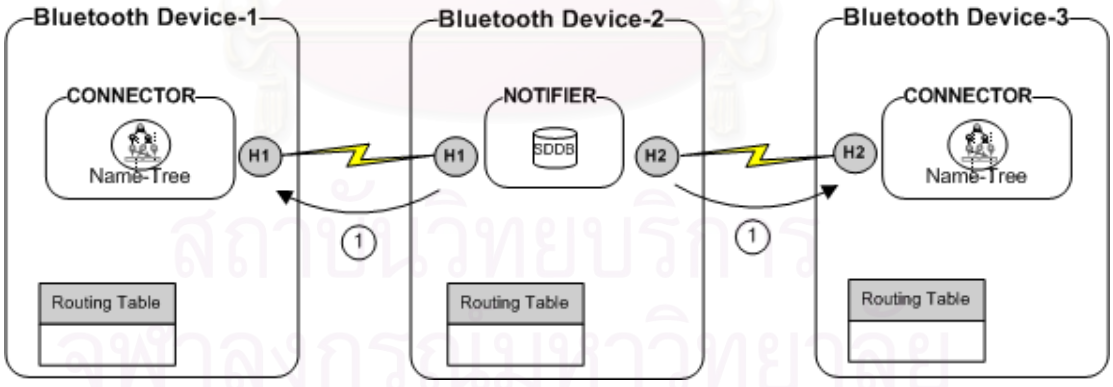
หมายเหตุ การประกาศชื่อของสเลฟ ถูกเก็บไว้ในฐานข้อมูลบริการของสเลฟเองแม้ว่าในขณะนั้นสเลฟไม่ได้เชื่อมต่อกับมาสเตอร์ เมื่อใดที่สเลฟมีการเชื่อมต่อกับมาสเตอร์ ชื่อที่สเลฟประกาศไว้จะถูกส่งไปเพิ่มในแผนภาพต้นไม้ของชื่อของทางฝั่งมาสเตอร์ด้วย ส่วนชื่อที่ประกาศโดยมาสเตอร์จะถูกจัดเก็บในแผนภาพต้นไม้ของชื่อได้โดยตรง ดังรูปที่ 3.13 ชื่อที่ได้รับการเพิ่มเข้าไปในแผนภาพต้นไม้ของชื่อจะทำให้อุปกรณ์สามารถรอรับข้อมูลจากอุปกรณ์ใดๆ ในเครือข่ายบลูทูธตามชื่อที่ระบุได้



รูปที่ 3.13 ขั้นตอนการเพิ่ม/ลบชื่อ

- **การสร้างเครือข่ายสแคทเทอร์เน็ตแบบสเลฟ/สเลฟบริดจ์**

จากรูปที่ 3.14 (หมายเลข 1) เนื่องจากสเลฟที่เชื่อมต่อกับมาสเตอร์มากกว่าหนึ่งตัวจะมีสถานะเป็นสเลฟ/สเลฟบริดจ์ ดังนั้นเมื่อสเลฟ (Bluetooth Device-2) ได้เชื่อมต่อกับมาสเตอร์-1 (Bluetooth Device-1) และมาสเตอร์-2 (Bluetooth Device-3) สเลฟทำการแจ้งสถานะไปยังมาสเตอร์ทุกครั้งที่มีการเปลี่ยนแปลงสถานะ

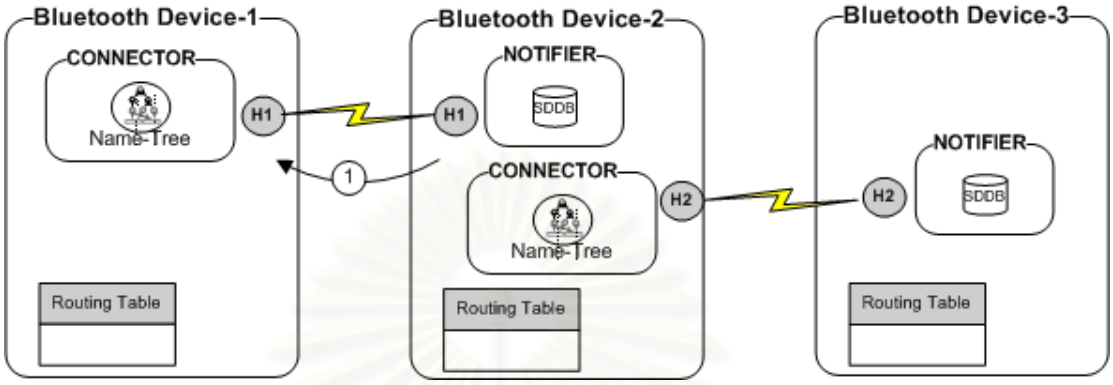


รูปที่ 3.14 ตัวอย่างการสร้างสแคทเทอร์เน็ตแบบสเลฟ/สเลฟบริดจ์

- **การสร้างเครือข่ายสแคทเทอร์เน็ตแบบมาสเตอร์/สเลฟบริดจ์**

อุปกรณ์บลูทูธ-2 (Bluetooth Device-2) สามารถรอรับการติดต่อจากมาสเตอร์และสามารถติดต่อไปยังสเลฟตัวอื่นได้ ทั้งนี้เมื่อเกิดการเปลี่ยนแปลงสถานะ สเลฟของอุปกรณ์บลูทูธ-2 ต้องแจ้งสถานะที่เปลี่ยนแปลงไปยังมาสเตอร์ (Bluetooth Device-1) (รูปที่ 3.15 หมายเลข 1)

การที่สเลฟต้องแจ้งสถานะไปยังมาสเตอร์ที่เชื่อมต่ออยู่ด้วยเพื่อให้มาสเตอร์ตรวจสอบได้ว่าอุปกรณ์บลูทูธในพิโคเน็ตตัวใดที่เป็นบริดจ์

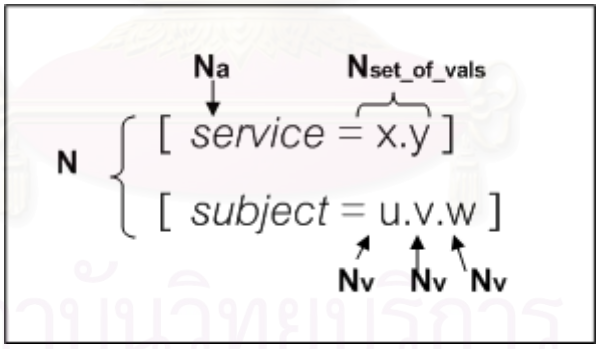


รูปที่ 3.15 ตัวอย่างการสร้างสแคทเทอร์เน็ตแบบมาสเตอร์/สเลฟบริดจ์

3.5 การออกแบบการจัดการแผนภาพต้นไม้ของชื่อ

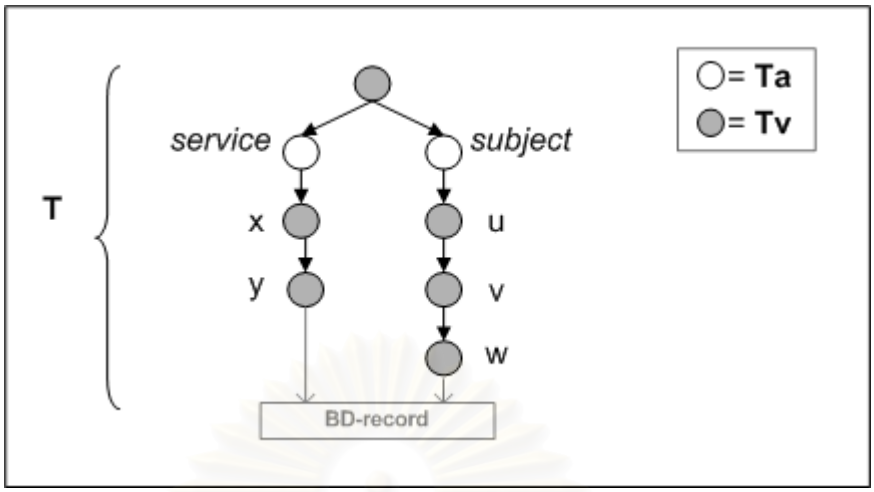
เนื้อหาในส่วนนี้ประกอบด้วยโครงสร้างของชื่อและขั้นตอนวิธีการจัดการแผนภาพต้นไม้ของชื่อ ได้แก่ ขั้นตอนวิธีการเพิ่ม การลบ และการค้นหา

3.5.1 โครงสร้างชื่อ



รูปที่ 3.16 แสดงค่าจากการระบุชื่อ

จากรูปที่ 3.16 แสดงการระบุชื่อ (แทนด้วยสัญลักษณ์ N) ที่กำหนดโดยผู้ใช้ ประกอบด้วยคุณสมบัติ (attribute) *service, subject* (แทนด้วยสัญลักษณ์ Na) และค่าของคุณสมบัติ (value) เช่น “x.y”, “u.v.w” (แทนด้วยสัญลักษณ์ Nset_of_vals) สามารถแบ่งค่าของคุณสมบัติออกเป็นชื่อย่อยได้จากเครื่องหมายจุด เมื่อแบ่งแล้วจะได้ชื่อที่ไม่สามารถแบ่งย่อยต่อไปได้อีก (แทนด้วยสัญลักษณ์ Nv)



รูปที่ 3.17 แสดงค่าในแผนภาพต้นไม้ของชื่อ

จากรูปที่ 3.17 เป็นตัวอย่างแผนภาพต้นไม้ (แทนด้วยสัญลักษณ์ T) ที่ได้จากการแปลงชื่อในรูปที่ 3.16 ประกอบด้วยโหนดคุณลักษณะ (แทนด้วยสัญลักษณ์ Ta ใช้เก็บค่า Na) และโหนดสำหรับเก็บค่า (แทนด้วยสัญลักษณ์ Tv ใช้เก็บค่า Nv) ค่าของแต่ละโหนดเป็นที่เก็บชื่อที่อยู่ในหน่วยย่อยที่สุดไม่สามารถแบ่งต่อไปได้อีก เช่น “x”, “u”, “Chat”, “Group1” เป็นต้น และโหนดใบจะอ้างอิงไปยังเลขที่อยู่บลูทูธของอุปกรณ์ที่สอดคล้องกับชื่อ

3.5.2 ขั้นตอนวิธีการจัดการแผนภาพต้นไม้ของชื่อ

ขั้นตอนวิธีการจัดการแผนภาพต้นไม้ของชื่อประกอบด้วย การเพิ่มชื่อเข้าแผนภาพต้นไม้ (Add Name-Tree Algorithm) การลบชื่อออกจากแผนภาพต้นไม้ (Delete Name-Tree Algorithm) การค้นหาปลายทางตามชื่อที่ระบุจากแผนภาพต้นไม้ (Search Name-Tree Algorithm)

- **ขั้นตอนวิธีการเพิ่มชื่อเข้าแผนภาพต้นไม้** (Add Name-Tree Algorithm)

```
1   ADD(T, N, BD-record)
2
3    for p ← each av-pair in N
4           Ta ← the attribute-node such that Ta attribute = p.attribute
5
6           for each Nv in Nset_of_values
7                 Tv ← the value-node such that Tv .value = Nv.value
8
9                 if Tv = null
10                      Tv ← a new value-node with Nv.value
11                      append Tv to Tv.parent
```

```
12
13                    //add the BD-record
14                    if Nv is an end-value
15                              append r to Tv.records
16                    else
17                              continue
```

รูปที่ 3.18 ขั้นตอนวิธีการเพิ่มชื่อเข้าแผนภาพต้นไม้ (Add Name-Tree Algorithm) [4]

จากรูปที่ 3.18 แสดงขั้นตอนวิธีการเพิ่มชื่อเข้าแผนภาพต้นไม้ วิธีการเพิ่มชื่อออธิบายได้ดังนี้ ในการเพิ่มชื่อจะทำการสร้างโหนดใหม่ก็ต่อเมื่อไม่พบโหนดที่ต้องการในแผนภาพต้นไม้ (นั่นคือ Tv = null) โหนดแต่ละโหนดถูกจัดเก็บไว้ในลักษณะเป็นลำดับชั้นจากโหนดแม่ไปยังโหนดลูก (บรรทัด 10-11) จนกระทั่งถึงโหนดใบเป็นโหนดที่เก็บเลขที่อยู่บลูทูธเอาไว้เพื่อใช้ในการอ้างอิงอุปกรณ์ต่อไป (บรรทัด 13-15)

- **ขั้นตอนวิธีการลบชื่อออกจากแผนภาพต้นไม้** (Delete Name-Tree Algorithm)

```
1   DELETE ( T, N, bd )
2
3    for each av-pair p:= (Na, Nset_of_vals) in N
4           Ta ← the child of T such that Ta's attribute = Na's attribute
5
6              // goto leaf node; the last Tv
7              for each Nv in Nset_of_vals
8                     Tv ← such that Tv's value = Nv's value
9
10  // at the leaf node
11  Delete bd in the set of bds's Tv
12
13  for each Tv such that Tv's value = Nv's value
14  // start from the last Nv backward to the first Nv
15
16          if Tv has no child nodes or the set of  bds's Tv
17                  Delete Tv
```

รูปที่ 3.19 ขั้นตอนวิธีการลบชื่อออกจากแผนภาพต้นไม้ (Delete Name-Tree Algorithm)

จากรูปที่ 3.19 แสดงขั้นตอนวิธีการลบชื่อออกจากแผนภาพต้นไม้ การลบจะทำการลบชื่อออกจากแผนภาพต้นไม้ได้ก็ต่อเมื่อได้รับการตรวจสอบแล้วว่าชื่อและเลขที่อยู่บลูทูธที่ระบุมานั้นถูกต้อง การลบชื่อเริ่มจากการค้นหาโหนดที่เชื่อมต่อกันตามลำดับของชื่อเพื่อไปยังโหนดใบ (บรรทัด 3-8) เมื่อถึงโหนดใบจะทำการลบเลขที่อยู่บลูทูธตามที่ระบุออกไปก่อน (บรรทัด 11) ต่อจากนั้นในกระบวนการลบโหนดจะเริ่มลบจากโหนดใบย้อนกลับไปสู่โหนดเริ่มต้น (บรรทัด 13-

17) แต่ก่อนที่จะลบโหนดใดๆ ออกไปจากแผนภาพต้นไม้จะมีการตรวจสอบก่อนว่าโหนดที่สามารถลบได้ ต้องเป็นโหนดที่ไม่มีโหนดลูกและเป็นโหนดที่ไม่มีระเบียนเลขที่อยู่บลูทูธเหลืออยู่ จึงจะสามารถลบโหนดได้ (บรรทัด 16)

- **ขั้นตอนวิธีการค้นหาปลายทางตามชื่อที่ระบุจากแผนภาพต้นไม้** (Search Name-Tree Algorithm)

```
1   SEARCH (T, N)

2    // search BD_Records from the Name-Specifier (N) in Name-Tree (T)
3
4    S ← the set of all possible BD-records
5
6    for each av-pair p:= (Na, Nset_of_vals) in n
7            Ta ← the child of T such that Ta's attribute = na's attribute
8
9          for each nv in Nset_of_vals
10                 if nv = *
11                         S' ← ∅
12                         for each Tv which is a child of Tv .parent
                                   S' ← S' ∪ all of the BD-records in the subtree rooted at
13  Tv
14                         S ← S ∩ S'
15
16                 else
17                         Tv ← such that Tv's value = Nv's value
18
19                         if Tv is a leaf node or Nv is a leaf node
20                                 S ← S ∩ the BD-records of Tv
21                         else
22                                 continue
23
24         Return S
```

รูปที่ 3.20 ขั้นตอนวิธีการค้นหาปลายทางตามชื่อที่ระบุจากแผนภาพต้นไม้ (Search Name-Tree Algorithm) [4]

จากรูปที่ 3.20 แสดงขั้นตอนวิธีการค้นหาปลายทางตามชื่อที่ระบุจากแผนภาพต้นไม้ หลักการของการค้นหาคือการนำเซตผลลัพธ์ที่ได้จากการค้นหาตามคุณสมบัติแต่ละตัวมาอินเตอร์เซกชันเพื่อให้ได้ผลลัพธ์เป็นค่าเลขที่อยู่บลูทูธของอุปกรณ์ปลายทางที่สอดคล้องกับชื่อที่ระบุ ส่วนรายละเอียดเกี่ยวกับวิธีการค้นหาเซตผลลัพธ์มีความแตกต่างกันระหว่างการค้นหาจากการระบุปลายทางแบบมัลติคาส บรอดคาส (ระบุโดยเครื่องหมาย *) และการค้นหาจากการระบุปลายทาง

แบบยูนิคาส ดังต่อไปนี้ วิธีการในการค้นหาปลายทางแบบมัลติคาสหรือบรอดคาสจะค้นหาไปจนกระทั่งถึงโหนดสุดท้าย (หมายถึง โหนดที่พบก่อนเครื่องหมาย * เช่น " u.* " โหนดสุดท้ายคือ Tv = "u") จากนั้นรวบรวมระเบียนเลขที่อยู่บลูทูธทุกระเบียนที่พบในโหนดสุดท้ายและโหนดที่อยู่ถัดจากโหนดสุดท้ายทุกโหนดจนกระทั่งถึงโหนดใบ (บรรทัด 10-14) ส่วนการค้นหาปลายทางแบบยูนิคาสจะสิ้นสุดการค้นหาที่โหนดใบแล้วเก็บเซตของระเบียนเลขที่อยู่บลูทูธจากโหนดใบไว้ (บรรทัด 17-22)

3.6 โครงสร้างของข่าวสารต่าง ๆ

ในส่วนนี้กล่าวถึงรายละเอียดโครงสร้างของข่าวสาร (message) ที่ใช้ในเฟรมเวิร์ค แบ่งได้ 3 ประเภท ข่าวสารของระบบ ข่าวสารจากการประมวลผลคำสั่งของผู้ใช้ และข่าวสารสำหรับอุปกรณ์บลูทูธ

3.6.1 ข่าวสารของระบบ (System Message) เป็นข่าวสารที่เฟรมเวิร์คสร้างขึ้นเอง ซึ่งเกี่ยวข้องกับระบบการทำงานภายในของเฟรมเวิร์ค ผู้ใช้ไม่ต้องจัดการในส่วนนี้เอง ข่าวสารประเภทนี้ประกอบด้วย ข่าวสารการปรับปรุงสถานะของบริดจ์ ข่าวสารเพื่อการตัดเส้นทาง ข่าวสารเพื่อการต่อเส้นทาง ข่าวสารแจ้งการกระจายข้อมูลอีกครั้ง ข่าวสารปรับปรุงชื่อ แสดงรายละเอียดในตารางที่ 3.5

ตารางที่ 3.5 โครงสร้างข่าวสารสำหรับระบบ

ข่าวสาร	รายละเอียด
ข่าวสารการปรับปรุงสถานะของบริดจ์	**รูปแบบข่าวสาร** _SYS: BR: _ *[SLAVE, SS_BRIDGE, MS_BRIDGE]* **คำอธิบาย** เป็นข่าวสารที่ส่งไปยังมาสเตอร์ เมื่อสเลฟมีการเปลี่ยนแปลงสถานะ โดยกำหนดสถานะใดสถานะหนึ่ง ดังต่อไปนี้ *SLAVE* – สถานะสเลฟ *(1)* *SS_BRIDGE* – สถานะสเลฟ/สเลฟบริดจ์ *(2)* *MS_BRIDGE* – สถานะมาสเตอร์/สเลฟบริดจ์ *(3)*
ข่าวสารเพื่อการตัดเส้นทาง	**รูปแบบข่าวสาร**

(Prune)	_SYS: PR: [bdSource: service, subject] **คำอธิบาย** เป็นข่าวสารที่อุปกรณ์จัดเส้นทางบลูทูธส่งไปเพื่อยกเลิกการรับข้อมูลจาก [s, n] ประกอบด้วย bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง
ข่าวสารเพื่อการต่อเส้นทาง (Graft)	**รูปแบบข่าวสาร** _SYS: GR: [bdSource: service, subject] **คำอธิบาย** เป็นข่าวสารที่อุปกรณ์จัดเส้นทางบลูทูธส่งไปเพื่อขอรับข้อมูลจาก [s, n] อีกครั้ง bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง
ข่าวสารแจ้งการกระจายข้อมูลอีกครั้ง (Reflood)	**รูปแบบข่าวสาร** _SYS: RF: [bdSource: service, subject]_passedRtr, … **คำอธิบาย** เป็นข่าวสารที่ส่งโดยผู้ส่งไปยังอุปกรณ์จัดเส้นทางบลูทูธทุกตัวในเครือข่ายตามระยะเวลาที่โหนดต้นทางกำหนด เพื่อให้อุปกรณ์จัดเส้นทางสามารถรับข้อมูลและพิจารณาเส้นทางที่ใช้ส่งผ่านข้อมูลใหม่ bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง passedRtr – เลขที่อยู่บลูทูธ ของอุปกรณ์จัดเส้นทางบลูทูธที่เคยได้รับข้อความนี้มาแล้ว อุปกรณ์จัดเส้นทางบลูทูธที่

	ได้รับข่าวสารนี้จะไม่ส่งข่าวสารไปอุปกรณ์บลูทูธข้างเคียงที่มีเลขที่อยู่ในกลุ่ม passedRtr เพื่อป้องกันการส่งแบบวนซ้ำในเครือข่าย
ข่าวสารปรับปรุงชื่อ (Update Name)	**รูปแบบข่าวสาร** _SYS:NOTIFY_UPDATENAME: **คำอธิบาย** เป็นข่าวสารที่อุปกรณ์สเลฟส่งไปยังมาสเตอร์เพื่อทำการปรับปรุงชื่อภายในระยะเวลาที่กำหนด เนื่องจากถ้าหากว่ามาสเตอร์รอสเลฟส่งการปรับปรุงชื่อเกินกว่า 3 ช่วงเวลา จะถือว่าสเลฟไม่ต้องการใช้ชื่อที่เคยประกาศไว้อีกต่อไป ซึ่งชื่อจะถูกลบออกจากแผนภาพต้นไม้ของมาสเตอร์

3.6.2 **ข่าวสารจากการประมวลผลคำสั่งของผู้ใช้ (User Message)** เป็นข่าวสารที่แปลงจากคำสั่งของผู้ใช้ให้อยู่ในรูปแบบที่อุปกรณ์แปลความหมายและสามารถนำมาประมวลผลตามคำสั่งต่อไปได้ แสดงในตารางที่ 3.6

ตารางที่ 3.6 โครงสร้างข่าวสารเพื่อประมวลผลจากคำสั่งของผู้ใช้

ข่าวสาร	รายละเอียด
ข่าวสารการเพิ่มชื่อ (Add Name)	**รูปแบบข่าวสาร** *_USR_ADDNAME_*: service, subject **คำอธิบาย** เป็นข่าวสารที่ส่งจากสเลฟไปยังมาสเตอร์เพื่อทำการเพิ่มชื่อเข้าไปในแผนภาพต้นไม้ของชื่อ *_USR_ADDNAME* – แสดงข่าวสารประเภทการเพิ่มชื่อ *(8)* service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง
ข่าวสารการลบชื่อ (Delete Name)	**รูปแบบข่าวสาร** *_USR_DELNAME_*: service, subject

	คำอธิบาย เป็นข่าวสารที่ส่งจากสเลฟไปยังมาสเตอร์เพื่อทำการลบชื่อออกจากแผนภาพต้นไม้ของชื่อ *_USR_DELNAME* – แสดงข่าวสารประเภทการลบชื่อ *(9)* service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง
ข่าวสารการส่ง (Send)	**รูปแบบข่าวสาร** *_USR_SEND*_bdSource: service, subject: data **คำอธิบาย** เป็นข่าวสารประเภทการส่ง ข่าวสารนี้ถูกใช้โดยอุปกรณ์จัดเส้นทางบลูทูธ ในการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง *_USR_SEND* – แสดงข่าวสารประเภทการส่ง *(5)* bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง data - ข้อมูล
ข่าวสารการส่งแบบขอการตอบกลับ (Send Request)	**รูปแบบข่าวสาร** *_USR_SEND_REQUEST*_bdSource: service, subject: data **คำอธิบาย** เป็นข่าวสารประเภทการส่งแบบขอการตอบกลับ ข่าวสารนี้ถูกใช้โดยอุปกรณ์จัดเส้นทางบลูทูธ ในการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง *_USR_ SEND_REQUEST* – แสดงข่าวสารประเภทการส่งแบบขอการตอบกลับ *(6)* bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง service - การระบุชื่อบริการเพื่อส่งไปปลายทาง subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง data - ข้อมูล

ข่าวสารการส่งแบบตอบกลับ (Send Reply)	**รูปแบบข่าวสาร** *_USR_SEND_REPLY_*key: data **คำอธิบาย** เป็นข่าวสารประเภทการส่งแบบตอบกลับ ข่าวสารนี้ถูกใช้โดยอุปกรณ์จัดเส้นทางบลูทูธ ในการส่งข้อมูลจากโหนดปลายทางกลับไปยังโหนดต้นทาง *_USR_ SEND_REPLY* – แสดงข่าวสารประเภทการส่งแบบตอบกลับ *(7)* key – ประกอบด้วย bdSource, service, subject data - ข้อมูล

3.6.3 **ข่าวสารสำหรับอุปกรณ์บลูทูธ** (Destination Message) เป็นข่าวสารที่ถูกแสดงจากเฟรมเวิร์คไปยังผู้ใช้ เพื่อรายงานข้อมูลและสถานะต่างๆ เมื่อผู้ใช้ได้รับข่าวสารสามารถตรวจสอบความหมายของข้อมูลต่างๆ ที่อยู่ในโครงสร้างข่าวสารได้ตามรูปแบบที่แสดงในตารางที่ 3.7

หมายเหตุ ข่าวสารสำหรับอุปกรณ์บลูทูธที่เฟรมเวิร์คได้รับมีรูปแบบเป็น "_DEST: *msg*" แบ่งได้เป็นส่วน "_DEST" เป็นส่วนที่เฟรมเวิร์คใช้ตรวจสอบว่าเป็นชนิดข่าวสารสำหรับอุปกรณ์ ส่วน *msg* เป็นส่วนที่ส่งไปยังผู้ใช้ตามที่แสดงในตารางที่ 3.7

ตารางที่ 3.7 โครงสร้างข่าวสารสำหรับอุปกรณ์บลูทูธ

รายละเอียดโครงสร้างข่าวสาร
ชนิดข่าวสาร*_USR_SEND* (5) **รูปแบบข่าวสาร** *_USR_SEND_*bdSource: **service, subject: data** **คำอธิบาย** เป็นข่าวสารประเภทการส่ง ข่าวสารนี้ถูกแสดงที่โหนดปลายทางซึ่งมีชื่อสอดคล้องกับชื่อที่โหนดต้นทางระบุ
ชนิดข่าวสาร*_USR_SEND_REQUEST* (6) **รูปแบบข่าวสาร**

_USR_SEND_REQUEST **_bdSource: service, subject: data**

คำอธิบาย

เป็นข่าวสารประเภทการส่งแบบขอการตอบกลับ (หมายความว่าโหนดต้นทางต้องการให้โหนดปลายทางส่งข้อมูลตอบกลับ) ข่าวสารนี้ถูกแสดงที่โหนดปลายทาง เช่น การคิวรี (query) ข้อมูลจากฐานข้อมูลของโหนดปลายทาง เป็นต้น

ชนิดข่าวสาร *_USR_SEND_REPLY* (7)

รูปแบบข่าวสาร

_USR_SEND_REPLY **_key: data**

คำอธิบาย

ข่าวสารนี้ถูกแสดงที่โหนดต้นทาง เป็นข่าวสารประเภทการส่งแบบตอบกลับที่ได้รับจากโหนดปลายทาง

ชนิดข่าวสาร*_STATUS_SEND_REQUEST_TIMEOUT* (11)

รูปแบบข่าวสาร

*_STATUS_SEND_REQUEST_TIMEOUT_*: **service, subject:**

คำอธิบาย

เป็นสถานะแจ้งให้โหนดต้นทาง แสดงการสิ้นสุดระยะเวลารอของการส่งข้อมูลแบบขอการตอบกลับ และหลังจากนี้โหนดต้นทางสามารถส่งข้อมูลแบบขอการตอบกลับไปยังชื่อดังกล่าวได้ เช่น เมื่อหมดระยะเวลารอการตอบกลับของ “chat”, “chula.a” จึงจะเริ่มส่งข้อมูลแบบขอการตอบกลับไปยังชื่อ “chat”, “chula.a” ใหม่ได้

<u>หมายเหตุ</u> โหนดต้นทางจะไม่สามารถรับข่าวสารตอบกลับของข้อมูลที่ส่งไปก่อนหน้านี้จากโหนดปลายทาง เนื่องจากได้สิ้นสุดระยะเวลารอของโหนดต้นทาง

ชนิดข่าวสาร*_ERR_DESTINATION_NAME_ISTIMING* (12)

รูปแบบข่าวสาร

*_ERR_DESTINATION_NAME_ISTIMING_*: **service, subject:**

คำอธิบาย

เป็นสถานะแจ้งความผิดพลาด เมื่อโหนดต้นทางส่งข้อมูลแบบขอการตอบกลับขณะที่ยังไม่หมด

ระยะเวลาที่กำหนดไว้ ดังนั้นผู้ใช้จึงไม่สามารถส่งข้อมูลแบบขอการตอบกลับด้วยชื่อดังกล่าว เนื่องจากโหนดต้นทางต้องรอให้สิ้นสุดระยะเวลาของการรอรับข้อมูลตอบกลับก่อนจึงจะสามารถส่งใหม่ได้

ชนิดข่าวสาร*_ERR_KEY_SEND_REPLY* (13)

รูปแบบข่าวสาร

*_ERR_KEY_SEND_REPLY_*

คำอธิบาย

เป็นสถานะแจ้งโหนดปลายทาง แสดงความผิดพลาดเมื่อผู้ใช้ระบุคีย์สำหรับส่งข้อมูลแบบตอบกลับไม่ถูกต้อง เฉพาะโหนดปลายทางที่ได้รับข้อมูลแบบขอการตอบกลับเท่านั้น จึงจะสามารถส่งข้อมูลตอบกลับไปยังโหนดต้นทางได้

ชนิดข่าวสาร *_STATUS_RECEIVER_EXIST* (14)

รูปแบบข่าวสาร

*_STATUS_RECEIVER_EXIST_*: service, subject:

คำอธิบาย

เป็นการแจ้งสถานะให้โหนดต้นทาง ว่ามีโหนดปลายทางในเครือข่ายที่สามารถรับข้อมูลที่ส่งไปได้

<u>หมายเหตุ</u>

โหนดต้นทางจะได้รับสถานะนี้จากเฟรมเวิร์คทุกครั้ง (ยกเว้น ครั้งแรกที่ส่งไปยังโหนดปลายทางตามชื่อดังกล่าว) จนกระทั่งถึงเวลาที่เฟรมเวิร์คทำการกระจายข้อมูลอีกครั้งก็จะเข้าสู่สถานะเริ่มต้นใหม่ทั้งหมด

ชนิดข่าวสาร*_STATUS_RECEIVER_NOT_EXIST* (15)

รูปแบบข่าวสาร

*_STATUS_RECEIVER_NOT_EXIST_*: service, subject:

คำอธิบาย

เป็นการแจ้งสถานะให้โหนดต้นทาง ว่าไม่มีโหนดปลายทางตามชื่อที่ระบุอยู่ในเครือข่าย

<u>หมายเหตุ</u>

โหนดต้นทางจะได้รับการแจ้งสถานะนี้จากเฟรมเวิร์คเมื่อทำการส่งทุกครั้ง ยกเว้น เมื่อส่งไปยัง

โหนดปลายทางตามชื่อดังกล่าวครั้งแรก จนกระทั่งถึงเวลาที่เฟรมเวิร์คทำการกระจายข้อมูลอีกครั้ง จะเข้าสู่สถานะเริ่มต้นใหม่ทั้งหมด

ชนิดข่าวสาร *_STATUS_CREATE_CONNECTION_SUCCESS* (21)

รูปแบบข่าวสาร

*_STATUS_CREATE_CONNECTION_SUCCESS_*: connectionName:

คำอธิบาย

เป็นการแจ้งสถานะให้มาสเตอร์ เมื่อสร้างการติดต่อสื่อสารไปยังสเลฟได้

ชนิดข่าวสาร *_STATUS_CREATE_CONNECTION_FAIL* (22)

รูปแบบข่าวสาร

*_STATUS_CREATE_CONNECTION_FAIL_*: connectionName:

คำอธิบาย

เป็นการแจ้งสถานะให้มาสเตอร์ เมื่อไม่สามารถสร้างการติดต่อสื่อสารไปยังสเลฟได้ สาเหตุอาจเกิดจาก ขณะนั้นมาสเตอร์สร้างการติดต่อกับสเลฟที่ระบุไว้อยู่แล้ว หรือเนื่องจากการเคลื่อนที่ของอุปกรณ์

ชนิดข่าวสาร *_STATUS_SERVICE_RECORD_ADDNAME_SUCCESS* (31)

รูปแบบข่าวสาร

*_STATUS_SERVICE_RECORD_ADDNAME_SUCCESS_*: service, subject:

คำอธิบาย

เป็นสถานะที่แจ้งไปยังสเลฟว่าชื่อที่ระบุเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ได้รับการเพิ่มเข้าไปในฐานข้อมูลบริการของอุปกรณ์บลูทูธเรียบร้อยแล้ว (ชื่อดังกล่าวต้องมีรูปแบบถูกต้อง และเป็นชื่อที่ไม่ซ้ำกับชื่อเดิมที่เคยเก็บไว้) โดยชื่อที่เก็บไว้ในฐานข้อมูลบริการ ซึ่งมาสเตอร์สามารถมองเห็นได้ขณะค้นหาอุปกรณ์เพื่อสร้างพิโคเน็ต

ชนิดข่าวสาร *_STATUS_SERVICE_RECORD_ADDNAME_FAIL* (32)

รูปแบบข่าวสาร

*_STATUS_SERVICE_RECORD_ADDNAME_FAIL_*: service, subject:

คำอธิบาย
เป็นสถานะที่แจ้งไปยังสเลฟว่าชื่อที่ระบุเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ไม่สามารถเพิ่มเข้าไปในฐานข้อมูลบริการของอุปกรณ์บลูทูธ สาเหตุอาจเกิดจาก รูปแบบของชื่อไม่ถูกต้อง หรือเป็นชื่อที่ซ้ำกับชื่อเดิมที่พบในฐานข้อมูลบริการ

ชนิดข่าวสาร *_STATUS_SERVICE_RECORD_DELNAME_SUCCESS* (33)
รูปแบบข่าวสาร
*_STATUS_SERVICE_RECORD_DELNAME_SUCCESS_*: service, subject:

คำอธิบาย
เป็นการแจ้งสถานะให้สเลฟ เมื่อลบชื่อออกจากฐานข้อมูลบริการของสเลฟได้

ชนิดข่าวสาร *_STATUS_SERVICE_RECORD_DELNAME_FAIL* (34)
รูปแบบข่าวสาร
*_STATUS_SERVICE_RECORD_DELNAME_FAIL_*: service, subject:

คำอธิบาย
เป็นการแจ้งสถานะให้สเลฟ เมื่อไม่สามารถลบชื่อออกจากฐานข้อมูลบริการของสเลฟ อาจเนื่องมาจาก รูปแบบของชื่อไม่ถูกต้อง หรือเป็นชื่อที่ไม่ปรากฏในฐานข้อมูลบริการ

ชนิดข่าวสาร *_STATUS_DEVICE_ACCEPTED_CONNECTION* (35)
รูปแบบข่าวสาร
_STATUS_DEVICE_ACCEPTED_CONNECTION _

คำอธิบาย
เป็นการแจ้งสถานะให้สเลฟ เมื่อสเลฟได้รับการเชื่อมต่อจากมาสเตอร์

ชนิดข่าวสาร *_STATUS_DEVICE_WAITINGCONNECTION_FAIL* (36)
รูปแบบข่าวสาร
_STATUS_DEVICE_WAITINGCONNECTION_FAIL _

คำอธิบาย
เป็นการแจ้งสถานะให้สเลฟ เมื่อมีความผิดพลาดที่สเลฟขณะรอการเชื่อมต่อจากมาสเตอร์

ชนิดข่าวสาร *_STATUS_CONNECTED_DEVICE_CLOSED* (41)
รูปแบบข่าวสาร
*_STATUS_CONNECTED_DEVICE_CLOSED_*: connectionName:

คำอธิบาย
เป็นการแจ้งสถานะให้มาสเตอร์ สเลฟ เมื่ออุปกรณ์บลูทูธที่เชื่อมต่อกันอยู่ปิดการติดต่อสื่อสารออกไป

ชนิดข่าวสาร *_STATUS_NAMETREE_ADDNAME_SUCCESS* (42)
รูปแบบข่าวสาร
*_STATUS_NAMETREE_ADDNAME_SUCCESS_*: service, subject:

คำอธิบาย
เป็นการแจ้งสถานะให้ทั้งมาสเตอร์ สเลฟ เมื่อเพิ่มชื่อเข้าในแผนภาพต้นไม้ของชื่อได้

ชนิดข่าวสาร *_STATUS_NAMETREE_ADDNAME_FAIL* (43)
รูปแบบข่าวสาร
*_STATUS_NAMETREE_ADDNAME_FAIL_*: service, subject:

คำอธิบาย
เป็นการแจ้งสถานะให้มาสเตอร์ สเลฟ เมื่อไม่สามารถเพิ่มชื่อที่ระบุเข้าในแผนภาพต้นไม้ของชื่อ สาเหตุอาจเกิดจากการระบุชื่อซ้ำกับชื่อที่ตนเองเคยประกาศไว้หรือรูปแบบของชื่อไม่ถูกต้อง

ชนิดข่าวสาร *_STATUS_NAMETREE_DELNAME_SUCCESS* (44)
รูปแบบข่าวสาร
*_STATUS_NAMETREE_DELNAME_SUCCESS_*: service, subject:

คำอธิบาย

เป็นการแจ้งสถานะให้มาสเตอร์ สเลฟ เมื่อลบชื่อที่ระบุออกจากแผนภาพต้นไม้ของชื่อได้

ชนิดข่าวสาร *_STATUS_NAMETREE_DELNAME_FAIL* (45)

รูปแบบข่าวสาร

*_STATUS_NAMETREE_DELNAME_FAIL_*: **service**, **subject**:

คำอธิบาย

เป็นการแจ้งสถานะให้มาสเตอร์ สเลฟ เมื่อไม่สามารถลบชื่อที่ระบุออกจากแผนภาพต้นไม้ของชื่อ

หมายเหตุ ความหมายของคำต่างๆ จากตารางที่ 3.7

bdSource - เลขที่อยู่บลูทูธของโหนดต้นทาง

service - การระบุชื่อบริการเพื่อส่งไปปลายทาง

subject - การระบุชื่ออุปกรณ์เพื่อส่งไปปลายทาง

data – ข้อมูล

key – ประกอบด้วย bdSource, service, subject

connectionName – ชื่อการติดต่อสื่อสารระหว่างมาสเตอร์กับสเลฟ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การพัฒนาเฟรมเวิร์คการติดต่อสื่อสารโดยการระบุชื่อ

ผู้วิจัยได้พัฒนาเฟรมเวิร์คขึ้นเพื่อให้โปรแกรมเมอร์สามารถนำเฟรมเวิร์คนี้ไปพัฒนาโปรแกรมประยุกต์เจทูเอ็มอี ตามสถาปัตยกรรมของเฟรมเวิร์คต่อไปได้ ประกอบด้วย แพคเกจ intentionalnaming.bluetooth.connection เป็นแพคเกจที่ติดต่อกับผู้ใช้โดยตรงเพื่อนำไปใช้พัฒนาโปรแกรมประยุกต์ และส่วนต่อมาเป็นแพคแกจที่พัฒนาขึ้นเพื่อซ่อนการทำงานของเฟรมเวิร์คออกจากส่วนติดต่อกับผู้ใช้ ประกอบด้วย แพคเกจ intentionalnaming.bluetooth.connection.impl และแพคเกจ intentionalnaming.nametree

4.1 แพคเกจ intentionalnaming.bluetooth.connection

แพคเกจ intentionalnaming.bluetooth.connection เป็นแพคเกจสำหรับผู้ใช้ในการติดต่อกับเฟรมเวิร์คเพื่อการพัฒนาโปรแกรมประยุกต์ คลาสต่างๆ ในแบบจำลองแพคเกจ intentionalnaming.bluetooth.connection แสดงได้ดังรูปที่ 4.1

รูปที่ 4.1 แบบจำลองแพคเกจ intentionalnaming.bluetooth.connection

ขั้นตอนโดยทั่วไปของผู้ใช้ในการสร้างระบบการติดต่อสื่อสารโดยการระบุชื่อจากเฟรมเวิร์คดังต่อไปนี้

- การสร้างอุปกรณ์บลูทูธ

  ในการพัฒนาโปรแกรมประยุกต์จากเฟรมเวิร์คเริ่มจากผู้ใช้ทำการสร้างอุปกรณ์บลูทูธ อุปกรณ์บลูทูธที่สร้างขึ้นในวิทยานิพนธ์นี้หมายถึงวัตถุ BTDevice เพื่อใช้เป็นตัวแทนการทำงานของอุปกรณ์บลูทูธจริงของระบบ และการปิดอุปกรณ์บลูทูธของเฟรมเวิร์คเมื่อต้องการสิ้นสุดการทำงานโดยคลาส BTDeviceFactory

- การกำหนดบทบาทมาสเตอร์และ/หรือสเลฟให้อุปกรณ์บลูทูธ

การกำหนดบทบาทของอุปกรณ์แบ่งได้ 2 แบบ คือให้เฟรมเวิร์คกำหนดให้อัตโนมัติ และแบบที่ผู้ใช้เป็นผู้กำหนดเอง ซึ่งผู้ใช้สามารถกำหนดให้อุปกรณ์บลูทูธเป็นมาสเตอร์หรือสเลฟอย่างใดอย่างหนึ่ง หรือเป็นทั้งมาสเตอร์และสเลฟภายในอุปกรณ์เดียวกันได้ ด้วยตัวกระทำการของคลาส BTDevice ผู้ใช้ที่กำหนดบทบาทให้อุปกรณ์บลูทูธเป็นมาสเตอร์จะได้รับวัตถุ BTDeviceConnector ส่วนผู้ใช้ที่กำหนดบทบาทให้อุปกรณ์บลูทูธเป็นสเลฟจะได้รับวัตถุ BTDeviceNotifier

- การสร้างการติดต่อสื่อสาร

  วิธีสร้างการติดต่อสื่อสารจะมีความแตกต่างระหว่างมาสเตอร์และสเลฟ กล่าวคือสเลฟเปิดการติดต่อสื่อสารไว้ (Inquiry scan และ Page scan) เพื่อรอรับการติดต่อจากมาสเตอร์ ส่วนมาสเตอร์ทำการค้นหาสเลฟที่อยู่ภายในบริเวณและติดต่อไปยังสเลฟที่ต้องการ (Inquiry และ Page)

- การระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ

  การระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ในวิทยานิพนธ์นี้ประกอบด้วยชื่อบริการและชื่ออุปกรณ์

- การส่งข้อมูล

  ผู้ใช้สามารถส่งข้อมูลได้ 3 รูปแบบ ประกอบด้วย การส่งข้อมูล (Send) การส่งข้อมูลแบบขอการตอบกลับ (Send Request) เป็นการระบุชื่อเพื่อการส่งไปยังปลายทาง และการส่งข้อมูลแบบตอบกลับ (Send Reply)

- การรับข่าวสารและสถานะจากเฟรมเวิร์ค

  ผู้ใช้รับการแจ้งข่าวสารและสถานะต่างๆ ของการทำงานจากเฟรมเวิร์คได้ด้วยคลาส BTDeviceListener

ตารางที่ 4.1 คลาสต่อประสานของเฟรมเวิร์ค

คลาสต่อประสาน (Interface Summary)	
BTDevice	เป็นคลาสสำหรับจัดการอุปกรณ์บลูทูธ
BTDeviceConnector	เป็นคลาสสำหรับอุปกรณ์บลูทูธที่มีบทบาทเป็นมาสเตอร์
BTDeviceNotifier	เป็นคลาสสำหรับอุปกรณ์บลูทูธที่มีบทบาทเป็นสเลฟ
BTDeviceListener	เป็นคลาสที่ทำให้โปรแกรมประยุกต์สามารถรับข่าวสารและสถานะต่างๆ จากเฟรมเวิร์คได้

ตารางที่ 4.2 คลาสของเฟรมเวิร์ค

คลาส (Class Summary)	
BTDeviceFactory	เป็นคลาสที่ใช้สำหรับสร้างอุปกรณ์บลูทูธ
BTRemoteDevice	เป็นคลาสสำหรับจัดเก็บข้อมูลเกี่ยวกับอุปกรณ์ที่ค้นพบ

ในตารางที่ 4.1 และตารางที่ 4.2 สรุปหน้าที่ของคลาสต่างๆ ที่พบในแพคเกจ intentionalnaming.bluetooth.connection ส่วนรายละเอียดของแต่ละคลาสอธิบายได้ดังต่อไปนี้

คลาส BTDeviceFactory

คลาส BTDeviceFactory เป็นคลาสสำหรับผู้ใช้ในการเริ่มต้นสร้างอุปกรณ์บลูทูธ และสิ้นสุดการทำงานของอุปกรณ์บลูทูธที่ได้สร้างขึ้น ตารางที่ 4.3 แสดงตัวกระทำการของคลาส BTDeviceFactory

ตารางที่ 4.3 ตัวกระทำการของคลาส BTDeviceFactory โดยสรุป

ตัวกระทำการ (Method Summary)	
static BTDevice	createBTDevice() ใช้สำหรับสร้างอุปกรณ์บลูทูธ
static void	closeBTDevice() เพื่อปิดการทำงานของอุปกรณ์บลูทูธ

รายละเอียดของตัวกระทำการ

- **createBTDevice**

public static BTDevice **createBTDevice** ()

ใช้สำหรับสร้างอุปกรณ์บลูทูธ ซึ่งให้ผู้ใช้สร้างอุปกรณ์บลูทูธได้เพียงตัวเดียวเท่านั้น (singleton) อุปกรณ์บลูทูธที่สร้างขึ้นโดยตัวกระทำการนี้เพื่อเป็นตัวแทนอุปกรณ์บลูทูธจริงของผู้ใช้ ดังนั้นค่าเริ่มต้นต่างๆ จึงขึ้นอยู่กับพื้นฐานอุปกรณ์บลูทูธของผู้ใช้เป็นหลัก

ค่าคืนกลับของตัวกระทำการ:

BTDevice ซึ่งใช้เป็นตัวแทนของอุปกรณ์บลูทูธ

- **closeBTDevice**

public static void **closeBTDevice** ()

เพื่อการปิดการทำงานของอุปกรณ์บลูทูธ รวมทั้งปิดการทำงานของวัตถุทุกตัวที่ได้สร้างขึ้นและคืนทรัพยากรแก่ระบบทั้งหมด

คลาส BTDevice

BTDevice เป็นคลาสต่อประสานสำหรับการจัดการเกี่ยวกับอุปกรณ์บลูทูธโดยทั่วไป ทำหน้าที่ในการกำหนดบทบาทให้กับอุปกรณ์บลูทูธว่าให้ทำหน้าที่เป็นมาสเตอร์และ/หรือสเลฟ การรับข่าวสารและสถานะ การส่งข้อมูลในรูปแบบต่างๆ รวมถึงการกำหนดระยะเวลาที่จะให้อุปกรณ์ส่งข้อมูลแบบกระจาย ภายในคลาส BTDevice ประกอบด้วยเขตข้อมูล แสดงไว้ในตารางที่ 4.4 (รายละเอียดโครงสร้างข่าวสาร แสดงใน 3.6.3 หัวข้อ ข่าวสารสำหรับอุปกรณ์บลูทูธ) และตัวกระทำการของคลาส BTDevice แสดงในตารางที่ 4.5

ตารางที่ 4.4 เขตข้อมูลของคลาส BTDevice

เขตข้อมูล (Field Summary)	
static int	_USR_SEND ข่าวสารประเภทการส่ง ที่โหนดปลายทางได้รับมาจากโหนดต้นทาง **See Also:** onData ()
static int	_USR_SEND_REQUEST ข่าวสารประเภทการส่งข้อมูลแบบขอการตอบกลับ (ในกรณีที่โหนดต้นทางต้องการข้อมูลตอบกลับจากโหนดปลายทาง) **See Also:** onData ()
static int	_USR_SEND_REPLY ข่าวสารประเภทการส่งข้อมูลตอบกลับจากโหนดปลายทางไปยังโหนดต้นทาง **See Also:** onData ()
static int	_STATUS_SEND_REQUEST_TIMEOUT สถานะบ่งบอกการสิ้นสุดระยะเวลารอ ในการส่งข้อมูลแบบขอการตอบกลับ **See Also:** onStatus (), sendRequest ()

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

static int	_ERR_DESTINATION_NAME_ISTIMING สถานะแสดงข้อผิดพลาด กรณีที่โหนดต้นทางส่งข้อมูลแบบขอการตอบกลับ ไปยังโหนดปลายทางขณะที่ระยะเวลาของการรอการตอบกลับยังไม่สิ้นสุด **See Also:** onStatus (), sendRequest ()
static int	_ERR_KEY_SEND_REPLY สถานะแสดงข้อผิดพลาด เนื่องจากโหนดปลายทางระบุคีย์ (key) สำหรับการส่งข้อมูลแบบตอบกลับ ไม่ถูกต้อง **See Also:** onStatus (), getRequestMsgKey (), sendReply ()
static int	_STATUS_RECEIVER_EXIST สถานะแจ้งโหนดต้นทางเมื่อมีการส่งข้อมูล เพื่อแสดงว่าในเครือข่ายบลูทูธมีโหนดปลายทางตามชื่อที่โหนดต้นทางโหนดต้นทางระบุ **See Also:** onStatus (), send (), sendRequest ()
static int	_STATUS_RECEIVER_NOT_EXIST สถานะแจ้งโหนดต้นทางเมื่อมีการส่งข้อมูล เพื่อแสดงว่าในเครือข่ายบลูทูธไม่มีโหนดปลายทางตามชื่อที่โหนดต้นทางระบุ **See Also:** onStatus (), send (), sendRequest ()
static int	_STATUS_CREATE_CONNECTION_SUCCESS สถานะแจ้งอุปกรณ์บลูทูธมาสเตอร์เมื่อสามารถสร้างการเชื่อมต่อไปยังสเลฟได้ **See Also:** onStatus (), createConnection ()
static int	_STATUS_CREATE_CONNECTION_FAIL สถานะแจ้งอุปกรณ์บลูทูธมาสเตอร์เมื่อไม่สามารถสร้างการเชื่อมต่อไปยังสเลฟได้ **See Also:** onStatus (), createConnection ()

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

static int	_STATUS_SERVICE_RECORD_ADDNAME_SUCCESS สถานะแจ้งอุปกรณ์บลูทูธที่เป็นสเลฟ เมื่อชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ได้รับการเพิ่มเข้าไปในฐานข้อมูลบริการของอุปกรณ์บลูทูธเรียบร้อยแล้ว **See Also:** onStatus (), createIntentionalName () ของคลาส BTDeviceNotifier
static int	_STATUS_SERVICE_RECORD_ADDNAME_FAIL สถานะแจ้งอุปกรณ์บลูทูธที่เป็นสเลฟ เมื่อไม่สามารถเพิ่มชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการเข้าไปในฐานข้อมูลบริการของอุปกรณ์บลูทูธ **See Also:** onStatus (), createIntentionalName () ของคลาส BTDeviceNotifier
static int	_STATUS_SERVICE_RECORD_DELNAME_SUCCESS สถานะแจ้งอุปกรณ์บลูทูธที่เป็นสเลฟ เมื่อสามารถลบชื่อตามที่ระบุออกจากฐานข้อมูลบริการของอุปกรณ์บลูทูธเรียบร้อยแล้ว **See Also:** onStatus (), deleteIntentionalName () ของคลาส BTDeviceNotifier
static int	_STATUS_SERVICE_RECORD_DELNAME_FAIL สถานะแจ้งอุปกรณ์บลูทูธที่เป็นสเลฟ เมื่อไม่สามารถลบชื่อตามที่ระบุออกจากฐานข้อมูลบริการของอุปกรณ์บลูทูธ **See Also:** onStatus (), deleteIntentionalName () ของคลาส BTDeviceNotifier
static int	_STATUS_DEVICE_ACCEPTED_CONNECTION สถานะที่แจ้งไปยังสเลฟ ขณะที่ได้รับการเชื่อมต่อจากมาสเตอร์ **See Also:** onStatus (), openConnection ()
static int	_STATUS_DEVICE_WAITINGCONNECTION_FAIL สถานะที่แจ้งไปยังสเลฟ เมื่อเกิดความผิดพลาดที่สเลฟ ขณะรอรับการติดต่อจากมาสเตอร์ **See Also:** onStatus (), openConnection ()

static int	_STATUS_CONNECTED_DEVICE_CLOSED สถานะ เมื่อการเชื่อมต่อกับอุปกรณ์บลูทูธข้างเคียงตัวใดตัวหนึ่งปิด **See Also:** onStatus ()
static int	_STATUS_NAMETREE_ADDNAME_SUCCESS สถานะ เมื่อชื่อที่ประกาศเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการถูกเก็บไว้ในแผนภาพต้นไม้ของชื่อ นั่นคือ อุปกรณ์สามารถรอรับข่าวสารด้วยชื่อนี้ได้ **See Also:** onStatus (), createIntentionalName()
static int	_STATUS_NAMETREE_ADDNAME_FAIL สถานะ เมื่อไม่สามารถเก็บชื่อที่ประกาศเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการไว้ในแผนภาพต้นไม้ของชื่อ **See Also:** onStatus (), createIntentionalName()
static int	_STATUS_NAMETREE_DELNAME_SUCCESS สถานะ เมื่อชื่อที่ประกาศเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการถูกลบออกจากแผนภาพต้นไม้ของชื่อเรียบร้อยแล้ว **See Also:** onStatus (), deleteIntentionalName()
static int	_STATUS_NAMETREE_DELNAME_FAIL สถานะ เมื่อไม่สามารถลบชื่อที่ระบุออกจากแผนภาพต้นไม้ของชื่อ **See Also:** onStatus (), deleteIntentionalName()

ตารางที่ 4.5 ตัวกระทำการของคลาส BTDevice

ตัวกระทำการ (Method Summary)	
BTDeviceConnector	createBTDeviceConnector () ใช้สร้างอุปกรณ์บลูทูธมาสเตอร์
BTDeviceNotifier	createBTDeviceNotifier () ใช้สร้างอุปกรณ์บลูทูธสเลฟ
void	addBTDeviceListener (BTDeviceListener listener) เพื่อให้อุปกรณ์บลูทูธสามารถรับข่าวสารและการแจ้งสถานะจากเฟรมเวิร์คได้

Object	createAutoConnection () เพื่อให้เฟรมเวิร์คทำการติดต่อกับอุปกรณ์บลูทูธรอบข้างให้อัตโนมัติ
Object	createAutoConnection (int x, int y) เพื่อให้เฟรมเวิร์คทำการติดต่อกับอุปกรณ์บลูทูธรอบข้างให้อัตโนมัติ โดยผู้ใช้สามารถจำลองตำแหน่งของอุปกรณ์ได้
void	send (String serviceName, String subjectName, String data) เรียกใช้เมื่อต้องการส่งข้อมูลโดยการระบุชื่อเพื่อการส่งไปยังปลายทาง
void	sendRequest (String serviceName, String subjectName, String data, long timeoutMSec) เรียกใช้เมื่อต้องการส่งข้อมูลแบบขอการตอบกลับโดยการระบุชื่อเพื่อการส่งไปยังปลายทาง กำหนดระยะเวลาการรอการตอบกลับ (มิลลิวินาที)
void	sendReply (String requestMsgKey, String data) เรียกใช้เมื่อโหนดปลายทางต้องการส่งข้อมูลตอบกลับไปยังโหนดต้นทาง
java.lang.String	getRequestMsgKey (String requestMsg) เป็นตัวกระทำการสำหรับเรียกใช้เมื่อโหนดปลายทางได้รับข่าวสารชนิด _USR_SEND_REQUEST เพื่อดึงค่าคีย์จากข่าวสารที่ได้รับ ไว้เพื่อใช้เป็นพารามิเตอร์ในการส่งข้อมูลตอบกลับ
int	getUserMsgType (String msg) คืนค่าชนิดของข่าวสารที่ผู้ใช้ได้รับ
java.lang.String	getUserMsgServiceName (String msg) คืนค่าเป็นชื่อบริการที่โหนดต้นทางระบุ
java.lang.String	getUserMsgSubjectName (String msg) คืนค่าเป็นชื่อของอุปกรณ์ที่โหนดต้นทางระบุ
java.lang.String	getUserMsgName (String msg) คืนค่าชื่อบริการและชื่อของอุปกรณ์ที่โหนดต้นทางระบุ รวมทั้งชื่อ

	แทนการเชื่อมต่อ
java.lang.String	getUserMsgData (String msg) คืนค่าข้อมูล จากข่าวสารที่ผู้ใช้ได้รับ
void	setTimeReflood (long sec) ใช้กำหนดระยะเวลาที่จะให้อุปกรณ์ ทำการกระจายข้อมูลอีกครั้ง เพื่อรองรับการเปลี่ยนแปลงของเครือข่ายบลูทูธที่อาจเกิดขึ้นระหว่างการติดต่อสื่อสาร
void	setDebugMode (Boolean isDebug) ให้พิมพ์ค่าแสดงการทำงานต่างๆ ภายในเฟรมเวิร์ค

รายละเอียดของตัวกระทำการ

- createBTDeviceConnector

public BTDeviceConnector createBTDeviceConnector ()

เป็นตัวกระทำการสำหรับสร้างอุปกรณ์บลูทูธมาสเตอร์

ค่าคืนกลับของตัวกระทำการ:

BTDeviceConnector วัตถุที่มีบทบาทและทำหน้าที่เป็นมาสเตอร์

- createBTDeviceNotifier

public BTDeviceNotifier createBTDeviceNotifier ()

เป็นตัวกระทำการสำหรับสร้างอุปกรณ์บลูทูธสเลฟ

ค่าคืนกลับของตัวกระทำการ:

BTDeviceNotifier วัตถุที่มีบทบาทและทำหน้าที่เป็นสเลฟ

- createAutoConnection

public Object createAutoConnection ()

เป็นตัวกระทำการสำหรับให้เฟรมเวิร์คทำการติดต่อกับอุปกรณ์บลูทูธรอบข้างให้อัตโนมัติ โดยเฟรมเวิร์คจะเริ่มกำหนดให้อุปกรณ์เป็นมาสเตอร์ เพื่อค้นหาและติดต่ออุปกรณ์ข้างเคียง ถ้าสามารถติดต่อกับอุปกรณ์บลูทูธอื่นได้ ก็จะส่งวัตถุ BTDeviceConnector กลับไปยังผู้ใช้ ถ้าไม่พบก็จะเปลี่ยนสถานะอุปกรณ์ให้เป็นสเลฟเพื่อรอรับการติดต่อจากมาสเตอร์ตัวอื่น ถ้าได้รับการติดต่อจากอุปกรณ์บลูทูธอื่น ก็จะส่งวัตถุ BTDeviceNotifier

กลับไปยังผู้ใช้ แต่ถ้าไม่ได้รับการติดต่อก็จะเปลี่ยนสถานะไปเป็นมาสเตอร์อีกครั้ง ทำเช่นนี้ทั้งหมด 2 รอบ

ค่าคืนกลับของตัวกระทำการ:

Object วัตถุที่คืนกลับมาอาจเป็นมาสเตอร์ หรือสเลฟ ถ้าสามารถสร้างการติดต่อกับอุปกรณ์บลูทูธข้างเคียงได้ หรือค่าว่าง (null) ถ้าไม่สามารถสร้างการติดต่อกับอุปกรณ์บลูทูธอื่นๆ

- **createAutoConnection**

public Object createAutoConnection (int x, int y)

เป็นตัวกระทำการสำหรับให้เฟรมเวิร์คทำการติดต่อกับอุปกรณ์บลูทูธรอบข้างให้อัตโนมัติ โดยเฟรมเวิร์คเช่นเดียวกับตัวกระทำการ createAutoConnection() แต่สามารถให้ผู้ใช้สามารถจำลองตำแหน่งของอุปกรณ์ในเครือข่ายได้ โดยกำหนดให้อุปกรณ์สามารถสามารถติดต่อสื่อสารกันได้เมื่อระยะห่างระหว่างอุปกรณ์ไม่เกิน 10 เมตร

พารามิเตอร์ของตัวกระทำการ:

x - พิกัดแกนเอกซ์ (X)

y – พิกัดแกนวาย (Y)

ค่าคืนกลับของตัวกระทำการ:

Object วัตถุที่คืนกลับมาอาจเป็นมาสเตอร์ หรือสเลฟ ถ้าสามารถสร้างการติดต่อกับอุปกรณ์บลูทูธข้างเคียงได้ หรือค่าว่าง (null) ถ้าไม่สามารถสร้างการติดต่อกับอุปกรณ์บลูทูธอื่นๆ

- **addBTDeviceListener**

public void addBTDeviceListener (BTDeviceListener listener)

เพื่อให้อุปกรณ์บลูทูธสามารถรับข่าวสารและการแจ้งสถานะจากเฟรมเวิร์คได้โดยตรง ผู้ใช้สามารถตรวจสอบชนิดของสถานะที่ได้รับโดยใช้ตัวกระทำการ getUserMsgType() ส่วนความหมายของสถานะต่างๆ แสดงในตารางที่ 4.4

พารามิเตอร์ของตัวกระทำการ:

listener – ค่าอ้างอิงวัตถุ BTDeviceListener

- **send**

public void send (String serviceName,

String subjectName,

String data)

เป็นตัวกระทำการสำหรับโหนดต้นทางส่งข้อมูลไปยังโหนดปลายทาง โดยการระบุชื่อเพื่อส่งไปยังปลายทาง

พารามิเตอร์ของตัวกระทำการ:

serviceName - ระบุชื่อบริการเพื่อการส่งไปยังปลายทาง

subjectName – ระบุชื่ออุปกรณ์เพื่อการส่งไปยังปลายทาง

data - ข้อมูลที่โหนดต้นทางต้องการส่ง

- **sendRequest**

public void **sendRequest** (String serviceName,

String subjectName,

String data,

long timeoutMSec)

เป็นตัวกระทำการสำหรับโหนดต้นทางส่งข้อมูลไปยังโหนดปลายทาง โดยโหนดต้นทางกำหนดระยะเวลาในการรอรับข้อมูลตอบกลับจากโหนดปลายทาง ถ้ายังไม่หมดระยะเวลาที่กำหนดไว้จะไม่สามารถส่งข้อมูลชนิดนี้โดยการระบุชื่อเพื่อส่งไปยังปลายทางเดิม

พารามิเตอร์ของตัวกระทำการ:

serviceName - ระบุชื่อบริการเพื่อการส่งไปยังปลายทาง

subjectName – ระบุชื่ออุปกรณ์เพื่อการส่งไปยังปลายทาง

data – ข้อมูลที่โหนดต้นทางต้องการส่ง

timeoutMSec - เวลาที่รอรับข้อมูลตอบกลับจากโหนดปลายทาง

- **sendReply**

public void **sendReply** (String requestMsgKey,

String data)

เป็นตัวกระทำการสำหรับโหนดปลายทางส่งข้อมูลตอบกลับไปยังโหนดต้นทาง เมื่อโหนดปลายทางได้รับข่าวสารชนิด _USR_SEND_REQUEST จากโหนดต้นทาง โหนดปลายทางใช้ตัวกระทำการ getRequestMsgKey() เพื่อดึงค่าคีย์จากข่าวสารมาใช้เป็นพารามิเตอร์ของตัวกระทำการในการส่งข้อมูลตอบกลับไปยังโหนดต้นทาง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พารามิเตอร์ของตัวกระทำการ:

requestMsgKey - คีย์ที่ใช้สำหรับการตอบกลับไปยังโหนดต้นทาง

data - ข้อมูลที่ต้องการส่ง

- **getRequestMsgKey**

public String getRequestMsgKey (String requestMsg)

โหนดปลายทางที่ได้รับข่าวสารชนิด _USR_SEND_REQUEST จากโหนดต้นทาง สามารถเรียกใช้ตัวกระทำการนี้ดึงค่าที่ใช้เป็นคีย์ (key) เพื่อเป็นพารามิเตอร์ในการตอบกลับไปยังโหนดต้นทาง ทั้งนี้ผู้ที่ได้รับข่าวสารจากโหนดต้นทางเท่านั้นจึงสามารถส่งข้อมูลตอบกลับไปยังโหนดต้นทางได้

พารามิเตอร์ของตัวกระทำการ:

requestMsg - คือ ข่าวสารที่ได้รับจากเฟรมเวิร์ค มีรูปแบบข่าวสารดังนี้ *_USR_SENDREQ*_bdSource:service,subject:data

ค่าคืนกลับของตัวกระทำการ:

“bdSource:service,subject” ที่ได้จากข่าวสาร ซึ่งใช้เป็นคีย์สำหรับส่งข่าวสารชนิด _USR_SEND_REPLY

- **getUserMsgType**

public int getUserMsgType (String msg)

ใช้ตัวกระทำการนี้ในการตรวจสอบว่าข่าวสารที่ได้รับมานั้นเป็นข่าวสารชนิดใด เมื่อโหนดปลายทางที่ได้รับข่าวสารและสถานะจากตัวกระทำการ onData() และ onStatus() ตามลำดับ

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารหรือสถานะที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

ชนิดของข่าวสาร ตามที่แสดงไว้ในตารางที่ 4.4

- **getUserMsgSource**

public String getUserMsgSource (String msg)

โหนดปลายทางสามารถใช้ตัวกระทำการนี้ในการตรวจสอบโหนดต้นทาง เมื่อได้รับข่าวสารที่มีชนิดของข่าวสารเป็น _USR_SEND, _USR_SENDREQ, _USR_SENDREP

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

เลขที่อยู่บลูทูธของโหนดต้นทาง

- **getUserMsgServiceName**

public String getUserMsgServiceName (String msg)

ตัวกระทำการสำหรับโหนดปลายทางในการตรวจสอบชื่อบริการที่โหนดต้นทางระบุเพื่อการส่งไปยังปลายทาง สำหรับข่าวสารและสถานะที่มีชนิดข่าวสาร ดังต่อไปนี้

_USR_SEND, _USR_SENDREQ, _USR_SENDREP,

_STATUS_SERVICE_RECORD_ADDNAME_SUCCESS,

_STATUS_SERVICE_RECORD_ADDNAME_FAIL,

_STATUS_SERVICE_RECORD_DELNAME_SUCCESS,

_STATUS_SERVICE_RECORD_DELNAME_FAIL,

_ERR_DESTINATION_NAME_ISTIMING,

_STATUS_SEND_REQUEST_TIMEOUT,

_STATUS_RECEIVER_EXIST,

_STATUS_RECEIVER_NOT_EXIST,

_STATUS_NAMETREE_ADDNAME_SUCCESS,

_STATUS_NAMETREE_ADDNAME_FAIL

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารหรือสถานะ ที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

ชื่อบริการ

- **getUserMsgSubjectName**

public String getUserMsgSubjectName (String msg)

ตัวกระทำการสำหรับโหนดปลายทางในการตรวจสอบชื่ออุปกรณ์ที่โหนดต้นทางระบุเพื่อการส่งไปยังปลายทาง สำหรับข่าวสารและสถานะที่มีชนิดเดียวกับที่ตัวกระทำการ getUserMsgServiceName () ได้รับ

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารหรือสถานะ ที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

ชื่ออุปกรณ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- **getUserMsgName**

public String **getUserMsgName** (String msg)

ตัวกระทำการสำหรับโหนดปลายทางในการตรวจสอบชื่อทั้งชื่อที่โหนดต้นทางระบุเพื่อการส่งไปยังปลายทาง สำหรับข่าวสารและสถานะเป็นชนิดเดียวกับที่ตัวกระทำการ getUserMsgServiceName () ได้รับ รวมทั้งสถานะ _STATUS_CREATE_CONNECTION_SUCCESS

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารหรือสถานะ (เฉพาะบางสถานะ) ที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

ชื่อบริการและชื่ออุปกรณ์ที่โหนดต้นทางระบุเพื่อการส่งไปยังปลายทาง มีรูปแบบดังนี้ "*serviceName,subjectName*" หรือ "*connectionName*"

- **getUserMsgData**

public String **getUserMsgData** (String msg)

เพื่อดึงข้อมูลที่ส่งมาจากโหนดต้นทาง เมื่อโหนดปลายทางที่ได้รับข่าวสารชนิด _USR_SEND, _USR_SENDREQ, _USR_SENDREP

พารามิเตอร์ของตัวกระทำการ:

msg - คือ ข่าวสารที่ได้รับจากเฟรมเวิร์ค

ค่าคืนกลับของตัวกระทำการ:

ข้อมูลที่ส่งมาจากโหนดต้นทาง

- **setTimeReflood**

public void **setTimeReflood** (long msec)

เป็นตัวกระทำการในการกำหนดช่วงเวลาที่จะให้อุปกรณ์บลูทูธ เข้าสู่กระบวนการกระจายข้อมูลอีกครั้ง (reflood) เพื่อให้เส้นทางใช้ในการส่งผ่านข้อมูลได้รับการพิจารณาใหม่ เพื่อรองรับการเปลี่ยนแปลงรูปแบบเครือข่ายที่อาจเกิดขึ้นระหว่างติดต่อสื่อสาร

พารามิเตอร์ของตัวกระทำการ:

msec - เวลา (หน่วยมิลลิวินาที)

- **setDebugMode**

public void **setDebugMode** (boolean isDebug)

ให้พิมพ์ค่าแสดงการทำงานต่างๆ ภายในเฟรมเวิร์ค เมื่อผู้ใช้ต้องการตรวจสอบรายละเอียดการทำงานของเฟรมเวิร์ค

พารามิเตอร์ของตัวกระทำการ:

isDebug - จริง (true) ถ้าต้องการให้พิมพ์การทำงานภายในเฟรมเวิร์ค

คลาสต่อประสาน BTDeviceListener

BTDeviceListener เป็นคลาสต่อประสานสำหรับผู้ใช้ในการรับข่าวสารโดยตัวกระทำการ onData() และรับการแจ้งสถานะจากเฟรมเวิร์คโดยตัวกระทำการ onStatus() ตารางที่ 4.6 แสดงตัวกระทำการของคลาส BTDeviceListener

ตารางที่ 4.6 ตัวกระทำการของคลาส BTDeviceListener

ตัวกระทำการ (Method Summary)	
void	onData(String dataMsg) เพื่อรับข้อมูลข่าวสารสำหรับอุปกรณ์บลูทูธ
void	onStatus(String statusData) เพื่อรับการแจ้งสถานะสำหรับอุปกรณ์บลูทูธ

รายละเอียดของตัวกระทำการ

- **onData**

public void **onData** (String dataMsg)

เพื่อให้โหนดปลายทางสามารถรับข้อมูลข่าวสารสำหรับอุปกรณ์บลูทูธได้

พารามิเตอร์ของตัวกระทำการ:

dataMsg – ข้อมูลข่าวสารที่ได้รับ ชนิดของข่าวสารที่ได้รับจากตัวกระทำการนี้ประกอบด้วย _USR_SEND,_USR_SEND_REQUEST หรือ _USR_SEND_REPLY

- **onStatus**

public void **onStatus** (String statusData)

เพื่อให้อุปกรณ์บลูทูธสามารถรับการแจ้งสถานะต่างๆ จากเฟรมเวิร์คได้

พารามิเตอร์ของตัวกระทำการ:

statusData – ข้อมูลสถานะที่ผู้ใช้ได้รับ (ความหมายของสถานะต่างๆ แสดงในตารางที่ 4.4)

คลาสต่อประสาน BTDeviceConnector

BTDeviceConnector เป็นคลาสต่อประสานสำหรับให้อุปกรณ์บลูทูธมีบทบาทเป็นมาสเตอร์ ตัวกระทำการของคลาส BTDeviceConnector แสดงในตารางที่ 4.7

ตารางที่ 4.7 ตัวกระทำการของคลาส BTDeviceConnector

ตัวกระทำการ (Method Summary)	
BTRemoteDevice []	searchBTRemoteDevices () คืนค่าอาร์เรย์ของอุปกรณ์บลูทูธที่ค้นพบ
void	stopSearchBTRemoteDevices () หยุดการค้นหาอุปกรณ์บลูทูธ
boolean	createConnection (BTRemoteDevice[] selectedBTRemotes) สร้างการติดต่อสื่อสารไปยังกลุ่มของอุปกรณ์บลูทูธที่ต้องการ
boolean	closeConnection (BTRemoteDevice[] selectedBTRemotes) เลือกปิดการติดต่อสื่อสารกับกลุ่มของอุปกรณ์บลูทูธที่ต้องการ
void	createIntentionalName (String serviceName, String subjectName) มาสเตอร์ระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ
void	deleteIntentionalName (String serviceName, String subjectName) มาสเตอร์ลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ
void	setTimeUpdateName (long msec) กำหนดระยะเวลาการลบชื่อของสเลฟออกจากแผนภาพต้นไม้ของชื่อเมื่อสเลฟไม่ส่งการปรับปรุงในระยะเวลาที่กำหนด

รายละเอียดของตัวกระทำการ

- **searchBTRemoteDevices**

public BTRemoteDevice[] searchBTRemoteDevices ( )

มาสเตอร์ค้นหาอุปกรณ์บลูทูธที่เป็นสเลฟ

ค่าคืนกลับของตัวกระทำการ:

อาร์เรย์ของอุปกรณ์บลูทูธ (สเลฟ) ที่ค้นพบ รวมทั้งข้อมูลแสดงรายละเอียดต่างๆ ของอุปกรณ์แต่ละตัวที่ค้นพบ (อธิบายรายละเอียดที่คลาส BTRemoteDevice)

- **stopSearchBTRemoteDevices**

public void stopSearchBTRemoteDevices ( )

หยุดการค้นหาอุปกรณ์บลูทูธระหว่างที่รอการค้นหาอุปกรณ์จากตัวกระทำการ searchBTRemoteDevices ( )

- **createConnection**

public boolean createConnection (BTRemoteDevice[] selectedBTRemotes)

สร้างการติดต่อสื่อสารไปยังกลุ่มของอุปกรณ์บลูทูธที่ต้องการ

พารามิเตอร์ของตัวกระทำการ:

selectedBTRemotes - อาร์เรย์ของอุปกรณ์บลูทูธที่ต้องการสร้างการติดต่อ

ค่าคืนกลับของตัวกระทำการ:

เท็จ (false) ถ้าอาร์เรย์ของอุปกรณ์บลูทูธที่เลือกไว้เป็นค่าว่าง (null) นอกจากนั้นจะคืนค่าจริง (true) และรายงานสถานะการติดต่อจากเฟรมเวิร์ค

_STATUS_CREATE_CONNECTION_SUCCESS หรือ

_STATUS_CREATE_CONNECTION_FAIL

- **closeConnection**

public boolean closeConnection (BTRemoteDevice[] selectedBTRemotes)

ปิดการติดต่อสื่อสารโดยการระบุกลุ่มของอุปกรณ์บลูทูธ

พารามิเตอร์ของตัวกระทำการ:

selectedBTRemotes - อาร์เรย์ของอุปกรณ์บลูทูธที่ต้องการปิดการติดต่อ

ค่าคืนกลับของตัวกระทำการ:

เท็จ (false) ถ้าอาร์เรย์ของอุปกรณ์บลูทูธที่เลือกไว้เป็นค่าว่าง (null) นอกจากนั้นจะคืนค่าจริง (true) ถ้าสามารถปิดการติดต่อสื่อสารของอุปกรณ์ตัวใดได้จะแสดงสถานะ _STATUS_CONNECTED_DEVICE_CLOSED

- **createIntentionalName**

public void createIntentionalName (String serviceName, String subjectName)

เพื่อประกาศชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ประกอบด้วยการระบุชื่อบริการและชื่ออุปกรณ์ ผู้ใช้ตรวจสอบสถานะการเพิ่มได้ ดังนี้

_STATUS_NAMETREE_ADDNAME_SUCCESS,

_STATUS_NAMETREE_ADDNAME_FAIL

พารามิเตอร์ของตัวกระทำการ:

serviceName – ชื่อบริการ

subjectName - ชื่ออุปกรณ์

- **deleteIntentionalName**

public void deleteIntentionalName (String serviceName, String subjectName)

เพื่อลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ด้วยการระบุชื่อบริการและชื่ออุปกรณ์ที่ไม่ต้องการ ผู้ใช้ตรวจสอบสถานะการลบได้ ดังนี้

_STATUS_NAMETREE_DELNAME_SUCCESS,

_STATUS_NAMETREE_DELNAME_ FAIL

พารามิเตอร์ของตัวกระทำการ:

serviceName – ชื่อบริการ

subjectName - ชื่ออุปกรณ์

- **setTimeUpdateName**

public void setTimeUpdateName (long msec)

กำหนดระยะเวลารอสเลฟส่งการปรับปรุงชื่อ ถ้ามาสเตอร์ไม่ได้รับการปรับปรุงจากสเลฟมากกว่า 3 ช่วงเวลา ก็จะถือว่าสเลฟได้ปิดการติดต่อออกไป จากนั้นมาสเตอร์จะทำการลบชื่อออกจากแผนภาพต้นไม้ของชื่อ เพื่อปรับปรุงให้แผนภาพต้นไม้มีความถูกต้องอยู่เสมอ

พารามิเตอร์ของตัวกระทำการ:

msec – ช่วงเวลา (หน่วยมิลลิวินาที)

<u>คลาสต่อประสาน</u> <u>BTDeviceNotifier</u>

BTDeviceNotifier เป็นคลาสต่อประสานสำหรับอุปกรณ์บลูทูธที่เป็นสเลฟ ตารางที่ 4.8 แสดงตัวกระทำการของคลาส BTDeviceNotifier

ตารางที่ 4.8 ตัวกระทำการของคลาส BTDeviceNotifier

ตัวกระทำการ (Method Summary)	
void	openConnection () เปิดการติดต่อไว้เพื่อรอรับการติดต่อที่มาจากมาสเตอร์
void	createIntentionalName (String serviceName, String subjectName) สเลฟประกาศชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ
void	deleteIntentionalName (String serviceName, String subjectName) สเลฟลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ
void	setTimeUpdateName (long msec) กำหนดระยะเวลาส่งการปรับปรุงชื่อไปยังมาสเตอร์

รายละเอียดของตัวกระทำการ

- **openConnection**

public void openConnection ( )

สเลฟเปิดการติดต่อไว้เพื่อรอรับการติดต่อที่มาจากมาสเตอร์ ในระหว่างที่สเลฟรอการติดต่อ สเลฟสามารถเพิ่มและลบชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการได้ด้วยตัวกระทำการ createIntentionalName() และตัวกระทำการ deleteIntentionalName() ตามลำดับ

- **createIntentionalName**

public void createIntentionalName (String serviceName, String subjectName)

เพื่อประกาศชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ประกอบด้วยการระบุชื่อบริการและชื่ออุปกรณ์ ผู้ใช้ตรวจสอบสถานะ ดังต่อไปนี้

_STATUS_SERVICE_RECORD_ADDNAME_SUCCESS,

_STATUS_SERVICE_RECORD_ADDNAME_FAIL สำหรับสเลฟที่มีการติดต่อสื่อสารกับมาสเตอร์ จะได้รับสถานะต่อไปนี้เพิ่มมาจากมาสเตอร์

_STATUS_NAMETREE_ADDNAME_SUCCESS,

_STATUS_NAMETREE_ADDNAME_FAIL

พารามิเตอร์ของตัวกระทำการ:

serviceName – ชื่อบริการ

subjectName - ชื่ออุปกรณ์

- **deleteIntentionalName**

public void deleteIntentionalName (String serviceName, String subjectName)

เพื่อลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ด้วยการระบุชื่อบริการและชื่ออุปกรณ์ที่ไม่ต้องการ ผู้ใช้ตรวจสอบสถานะจากการลบชื่อได้จากสถานะดังต่อไปนี้

_STATUS_SERVICE_RECORD_DELNAME_SUCCESS,

_STATUS_SERVICE_RECORD_DELNAME_FAIL

สำหรับสเลฟที่มีการติดต่อสื่อสารกับมาสเตอร์ จะได้รับสถานะต่อไปนี้เพิ่มมาจากมาสเตอร์

_STATUS_NAMETREE_DELNAME_SUCCESS,

_STATUS_NAMETREE_DELNAME_ FAIL

พารามิเตอร์ของตัวกระทำการ:

serviceName – ชื่อบริการ

subjectName – ชื่ออุปกรณ์

- **setTimeUpdateName**

public void setTimeUpdateName (long msec)

สเลฟกำหนดระยะเวลาส่งการปรับปรุงชื่อไปยังมาสเตอร์ เพื่อไม่ให้ชื่อที่สเลฟเคยประกาศไว้ถูกลบออกจากแผนภาพต้นไม้ของชื่อ

พารามิเตอร์ของตัวกระทำการ:

msec – ช่วงเวลา (หน่วยมิลลิวินาที)

คลาส BTRemoteDevice

คลาส BTDeviceRemoteDevice เป็นคลาสที่เก็บข้อมูลต่างๆ ของอุปกรณ์ที่มาสเตอร์ค้นพบ ตารางที่ 4.9 แสดงตัวกระทำการของคลาส BTDeviceRemoteDevice

ตารางที่ 4.9 ตัวกระทำการของคลาส BTDeviceRemoteDevice

ตัวกระทำการ (Method Summary)	
java.lang.String	getConnectionName () คืนชื่อแทนการติดต่อสื่อสารกับอุปกรณ์ที่พบ
int	getRemoteNamesSize ()

	คืนจำนวนชื่อทั้งหมดที่อุปกรณ์ประกาศไว้
java.lang.String	getRemoteServiceName (int i) คืนชื่อบริการ
java.lang.String	getRemoteSubjectName (int i) คืนชื่ออุปกรณ์
void	setConnectionName (String newConnectionName) เปลี่ยนแปลงชื่อแทนการติดต่อสื่อสาร

รายละเอียดของตัวกระทำการ

- **getConnectionName**

public String getConnectionName ( )

คืนค่าเป็นชื่อที่ใช้เรียกแทนการติดต่อสื่อสารไปยังอุปกรณ์ที่พบนี้ ชื่อที่ได้เป็นชื่อที่กำหนดโดยเฟรมเวิร์ค ผู้ใช้สามารถเปลี่ยนชื่อนี้ได้โดยตัวกระทำการ setConnectionName()

ค่าคืนกลับของตัวกระทำการ:

ชื่อแทนการติดต่อสื่อสารที่กำหนดโดยเฟรมเวิร์ค มีรูปแบบดังนี้ "*Connection n*" โดยที่ n = 1, 2,…, n

- **getRemoteNamesSize**

public int getRemoteNamesSize ( )

คืนจำนวนชื่อที่ประกาศไว้ทั้งหมด

ค่าคืนกลับของตัวกระทำการ:

จำนวนชื่อที่ประกาศไว้ทั้งหมด

- **getRemoteServiceName**

public String getRemoteServiceName (int i)

คืนชื่อบริการของอุปกรณ์ที่พบ

พารามิเตอร์ของตัวกระทำการ:

i – คือ ลำดับของชื่อที่ประกาศไว้ มีค่าตั้งแต่ 0 และไม่เกินจำนวนชื่อที่ประกาศไว้ทั้งหมด

ค่าคืนกลับของตัวกระทำการ:

ชื่อบริการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- getRemoteSubjectName

public String getRemoteSubjectName (int i)

คืนชื่ออุปกรณ์ที่พบ

พารามิเตอร์ของตัวกระทำการ:

i – คือ ลำดับของชื่อที่ประกาศไว้ มีค่าตั้งแต่ 0 และไม่เกินจำนวนชื่อที่ประกาศไว้ทั้งหมด

ค่าคืนกลับของตัวกระทำการ:

ชื่ออุปกรณ์

- setConnectionName

public void setConnectionName (String newConnectionName)

มาสเตอร์สามารถปรับเปลี่ยนชื่อแทนการติดต่อสื่อสารที่เฟรมเวิร์คกำหนดไว้ได้

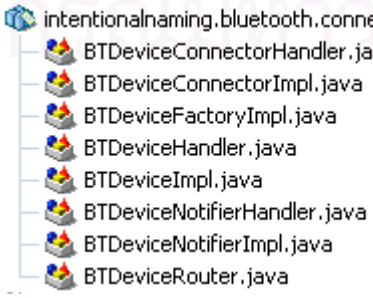
พารามิเตอร์ของตัวกระทำการ:

newConnectionName – ชื่อแทนการติดต่อสื่อสารกับสเลฟ

หมายเหตุ: ชื่อแทนการติดต่อสื่อสารที่เฟรมเวิร์คจะแจ้งไปยังมาสเตอร์รวมอยู่ในรายละเอียดของสถานะต่อไปนี้ _STATUS_CREATE_CONNECTION_SUCCESS, _STATUS_CREATE_CONNECTION_FAIL, _STATUS_CONNECTED_DEVICE_CLOSED

4.2 แพคเกจ intentionalnaming.bluetooth.connection.impl

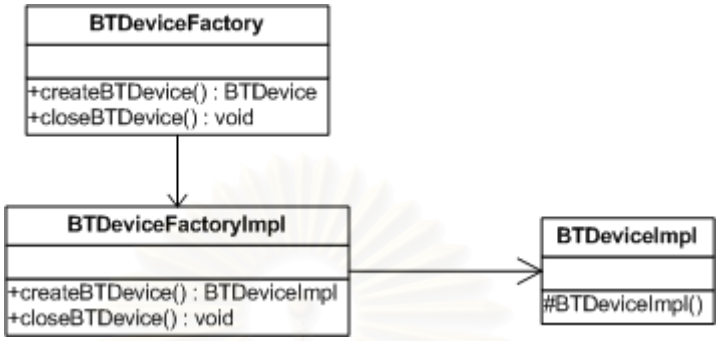
แพคเกจ intentionalnaming.bluetooth.connection.impl แสดงในรูปที่ 4.2 ประกอบด้วยคลาสที่ถูกพัฒนาขึ้นเพื่อซ่อนการทำงานของเฟรมเวิร์คออกจากส่วนติดต่อกับผู้ใช้ที่อยู่ในแพคเกจ intentionalnaming.bluetooth.connection



รูปที่ 4.2 แบบจำลองแพคเกจ intentionalnaming.bluetooth.connection.impl

คลาสต่างๆ ของแพคเกจ intentionalnaming.bluetooth.connection.impl มีรายละเอียดดังนี้

1. <u>คลาส BTDeviceFactoryImpl</u>



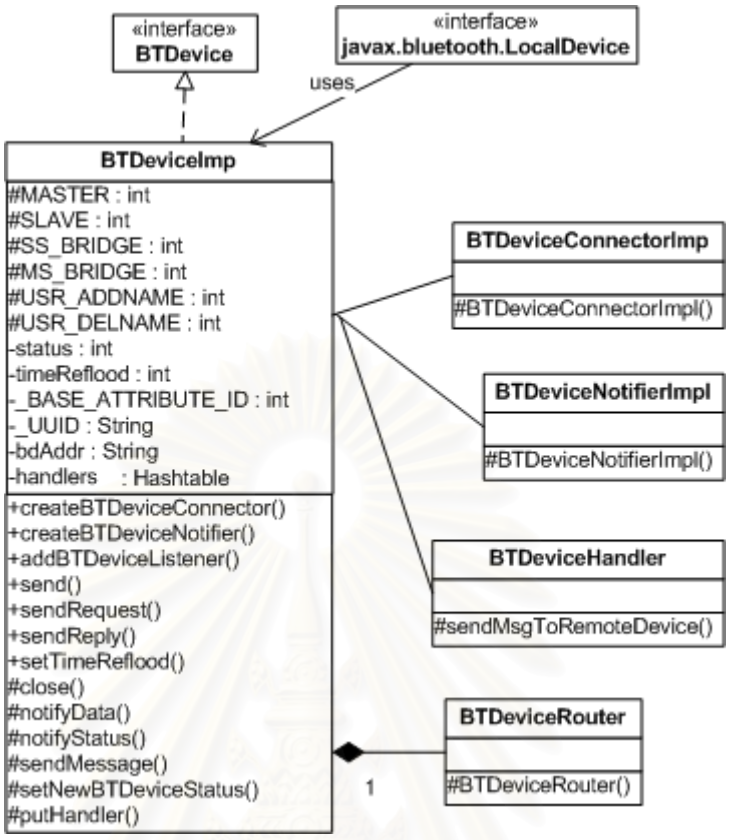
รูปที่ 4.3 แบบจำลองคลาส BTDeviceFactoryImpl

จากรูปที่ 4.3 คลาส BTDeviceFactoryImpl เป็นคลาสซึ่งถูกเรียกใช้โดยคลาส BTDeviceFactory ทำหน้าที่สร้างและปิดวัตถุ BTDeviceImpl ตัวกระทำการต่างๆ ของ คลาส BTDeviceFactoryImpl สามารถอธิบายได้ดังนี้

- ตัวกระทำการ createBTDevice() เป็นตัวกระทำการสำหรับสร้างวัตถุ BTDeviceImpl เพื่อจัดการอุปกรณ์บลูทูธ
- ตัวกระทำการ closeBTDevice() เป็นตัวกระทำการสำหรับปิดอุปกรณ์บลูทูธของเฟรมเวิร์ค โดยเรียกตัวกระทำการ close() ของวัตถุ BTDeviceImpl

<u>คลาส BTDeviceImpl</u>

คลาส BTDeviceImpl จากรูปที่ 4.4 เป็นคลาสที่ได้รับการพัฒนาตามตัวกระทำการต่อประสานของคลาส BTDevice เพื่อให้ผู้ใช้สามารถกำหนดบทบาทให้อุปกรณ์ (มาสเตอร์ และ/หรือสเลฟ) สามารถเป็นโหนดต้นทาง (ผู้ส่ง) และโหนดปลายทาง (ผู้รับ) โดยภายในคลาส BTDeviceImpl มีการกำหนดค่าพื้นฐานที่ใช้ในการทำงานและการติดต่อสื่อสารของเฟรมเวิร์ค ซึ่งคลาส BTDeviceImpl ได้รับการพัฒนาตามตัวกระทำการต่อประสานของคลาส javax.bluetooth.LocalDevice ของเจเอสอาร์-82 เพื่อเป็นตัวแทนอุปกรณ์ ส่วนคลาส BTDeviceHandler มีหน้าที่หลักในการจัดการการติดต่อสื่อสารและรับ-ส่งข้อมูลให้กับอุปกรณ์ และเมื่อต้องการส่งต่อข้อมูลจะมีการตรวจสอบเส้นทางให้กับอุปกรณ์โดยคลาส BTDeviceRouter



รูปที่ 4.4 แบบจำลองคลาส BTDeviceImpl

ลักษณะประจำและตัวกระทำการที่สำคัญของคลาส BTDeviceImpl สามารถอธิบายได้ดังนี้

- ลักษณะประจำ
  - status

    สถานะของอุปกรณ์บลูทูธ มีสถานะเป็น MASTER, SLAVE, SS_BRIDGE, MS_BRIDGE
  - _USR_ADDNAME

    ชนิดข่าวสารที่หมายถึงให้เพิ่มชื่อเข้าไปในแผนภาพต้นไม้ของชื่อ
  - _USR_DELNAME

    ชนิดข่าวสารที่หมายถึงให้ลบชื่อออกจากแผนภาพต้นไม้ของชื่อ
  - timeReflood

    ระยะเวลาของการกระจายข้อมูลอีกครั้ง (หน่วยมิลลิวินาที) ทำการกระจายข้อมูลอีกครั้ง เพื่อรองรับการเคลื่อนที่และการเปลี่ยนแปลงรูปแบบเครือข่าย
  - _BASE_ATTRIBUTE_ID

    ระบุค่าเริ่มต้นสำหรับค้นหาชื่อที่เก็บในฐานข้อมูลบริการ กำหนดให้มีค่าเท่ากับ 0x900 โดยที่จำนวนของชื่อที่เก็บได้มากสุดไม่เกิน จำนวนข้อมูลมากที่สุดที่

สามารถค้นหาได้จากฐานข้อมูลบริการ (จากคุณสมบัติ "bluetooth.sd.attr.retrievable.max" ของคลาส javax.bluetooth.LocalDevice)

- _UUID

  เป็นการระบุค่า ยู ยู ไอ ดี (Universal Unique Identifier-UUID) ของเจเอสอาร์-82 กำหนดค่าเป็น "4670590021AABBCCDDEEFFAAA2110811" เพื่อให้มาสเตอร์และสเลฟสามารถติดต่อกันได้

- bdAddr

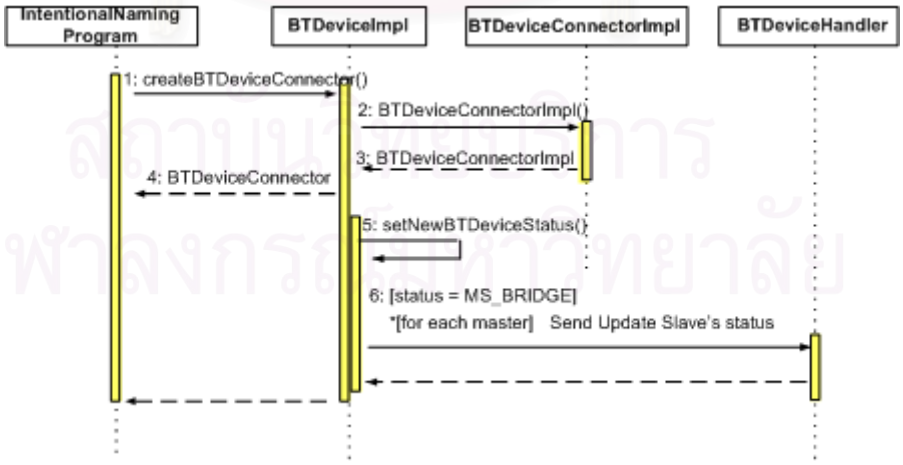
  เลขที่อยู่บลูทูธของอุปกรณ์ เพื่อใช้อ้างอิงอุปกรณ์ (โดยนำค่ามาจากคลาส LocalDevice ซึ่งเป็นเลขที่อยู่บลูทูธจริงของอุปกรณ์)

- handlers

  เก็บตัวจัดการการเชื่อมต่อเพื่อใช้อ้างอิงในการส่งข้อมูล

หมายเหตุ: เมื่อมีการเปลี่ยนแปลงสถานะของอุปกรณ์ที่เป็นสเลฟ (SLAVE, SS_BRIDGE, MS_BRIDGE) สเลฟจะทำการส่งข่าวสารแจ้งสถานะไปยังมาสเตอร์ทุกครั้ง ทั้งนี้เพื่อให้มาสเตอร์ตรวจสอบได้ว่าสามรถใช้สเลฟตัวใดสามารถทำหน้าที่ส่งผ่านข้อมูลระหว่างพิโคเน็ตได้

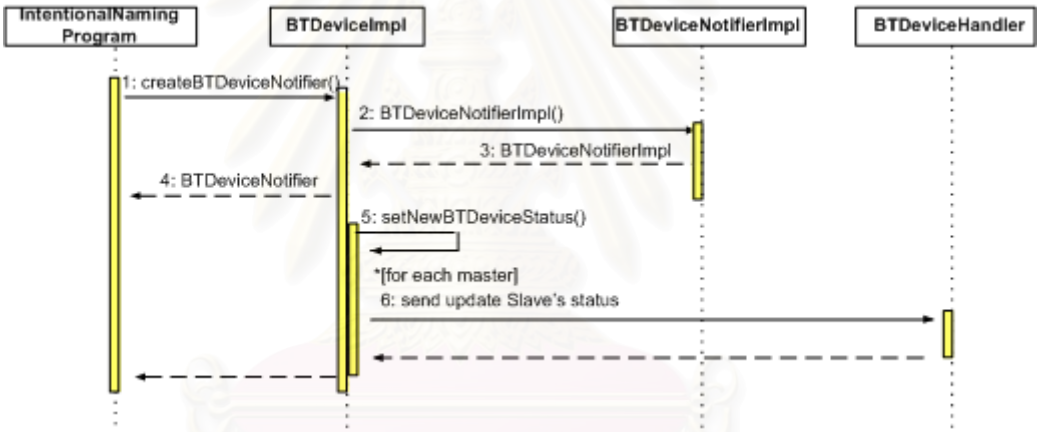
- ตัวกระทำการ createBTDeviceConnector() ทำหน้าที่สร้างวัตถุ BTDeviceConnectorImpl



รูปที่ 4.5 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createBTDeviceConnectorImpl()

จากรูปที่ 4.5 แสดงลำดับเหตุการณ์เมื่อผู้ใช้มีการเรียกใช้งานตัวกระทำการ createBTDeviceConnector() วัตถุ BTDeviceConnectorImpl จะถูกสร้างโดยตัวกระทำการ BTDeviceConnectorImpl() จากนั้นคืนวัตถุ BTDeviceConnector ไปยังผู้ใช้ เพื่อเป็นตัวแทนอุปกรณ์มาสเตอร์ (ขั้นตอน 2-4) จากนั้นกำหนดสถานะให้กับอุปกรณ์ด้วยตัวกระทำการ setNewBTDeviceStatus() เพื่อให้สถานะเป็น MASTER แต่ถ้าอุปกรณ์มีสถานะเดิมเป็น SLAVE หรือ SS_BRIDGE สถานะของอุปกรณ์จะถูกกำหนดเป็น MS_BRIDGE (5) กรณีที่อุปกรณ์มีสถานะเป็น MS_BRIDGE วัตถุ BTDeviceHandler ส่งข่าวสารรายงานสถานะใหม่ไปยังมาสเตอร์ที่เชื่อมต่ออยู่ด้วย (ขั้นตอน 6)

- ตัวกระทำการ createBTDeviceNotifier() ทำหน้าที่สร้างวัตถุ BTDeviceNotifierImpl



รูปที่ 4.6 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createBTDeviceNotifierImpl()

จากรูปที่ 4.6 เมื่อผู้ใช้เรียกตัวกระทำการ createBTDeviceNotifier() ทำการสร้างวัตถุ BTDeviceNotifierImpl (ขั้นตอน 1-4) จากนั้นตัวกระทำการ setNewBTDeviceStatus() จะกำหนดสถานะให้กับอุปกรณ์เป็น SLAVE (แต่ถ้าอุปกรณ์มีสถานะเดิมเป็น MASTER ก็จะกำหนดสถานะของอุปกรณ์ใหม่เป็น MS_BRIDGE) (5) และให้วัตถุ BTDeviceHandler ที่เชื่อมต่อกับมาสเตอร์ ส่งข่าวสารรายงานสถานะใหม่ไปยังมาสเตอร์ที่เชื่อมต่ออยู่ด้วย (ขั้นตอน 6)

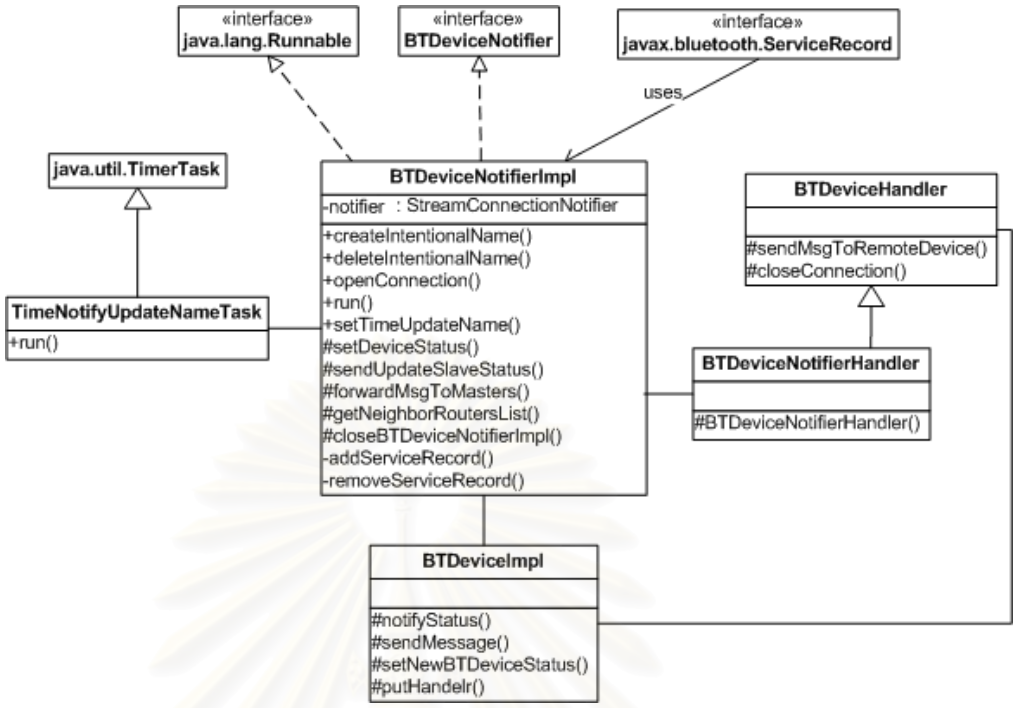
- ตัวกระทำการ addBTDeviceListener() เป็นตัวกระทำการสำหรับให้ผู้ใช้ลงทะเบียนตัวต่อประสาน BTDeviceListener เพื่อรับข่าวสารและสถานะการทำงานต่างๆ จากเฟรมเวิร์ค
- ตัวกระทำการ send() ทำหน้าที่นำข้อมูลจากผู้ใช้ (โหนดต้นทาง) เพื่อสร้างข่าวสารประเภท _USR_SEND (รูปแบบโครงสร้างข่าวสาร หัวข้อ 3.6.3) แล้วส่งข่าวสารต่อให้วัตถุ BTDeviceRouter ทำการจัดส่งข้อมูลต่อไป
- ตัวกระทำการ sendRequest() ทำหน้าที่นำข้อมูลจากผู้ใช้ (โหนดต้นทาง) ประเภท _USR_SEND_REQUEST (รูปแบบโครงสร้างข่าวสาร หัวข้อ 3.6.3) โดยตรวจสอบชื่อของปลายทางก่อนว่าชื่อดังกล่าวยังอยู่ในช่วงเวลารอการตอบกลับหรือไม่ ถ้ายังอยู่ในช่วงรอการตอบกลับจะเรียกตัวกระทำการ notifyStatus() เพื่อแจ้งสถานะ _ERR_DESTINATION_NAME_ISTIMING กลับไปยังผู้ใช้ นั่นคือในขณะนั้นผู้ใช้ไม่สามารถส่งข่าวสารได้ แต่ถ้าตรวจสอบแล้วไม่พบจึงจะสร้างและส่งข่าวสารต่อให้วัตถุ BTDeviceRouter เป็นตัวจัดการส่งข้อมูลต่อไป
- ตัวกระทำการ sendReply() เป็นตัวกระทำการสำหรับนำข้อมูลจากผู้ใช้ (โหนดปลายทาง) เพื่อสร้างข่าวสารประเภท _USR_SEND_REPLY (รูปแบบโครงสร้างข่าวสาร หัวข้อ 3.6.3) จากนั้นจะส่งข่าวสารต่อให้วัตถุ BTDeviceRouter จัดการส่งข้อมูลกลับไปยังโหนดต้นทางต่อไป แต่ถ้าอุปกรณ์เป็นสเลฟข่าวสารจะถูกส่งไปที่มาสเตอร์ เพื่อให้มาสเตอร์รับหน้าที่การส่งข้อมูลต่อไปยังโหนดต้นทาง
- ตัวกระทำการ setTimeReflood() เป็นตัวกระทำการกำหนดระยะเวลาของการเริ่มกระบวนการกระจายข้อมูลอีกครั้ง (หน่วยมิลลิวินาที) ทั้งนี้ผู้ใช้สามารถกำหนดระยะเวลาตามสภาวะแวดล้อมของเครือข่ายในขณะนั้น
- ตัวกระทำการ close() เป็นตัวกระทำการสำหรับปิดวัตถุ BTDeviceImpl รวมทั้งปิดการทำงานของวัตถุทั้งหมดที่ถูกสร้างขึ้น กำหนดค่าของวัตถุให้เป็นค่าเช่นเดียวกับตอนเริ่มต้นและคืนทรัพยากรให้แก่ระบบ
- ตัวกระทำการ notifyData() เป็นตัวกระทำการในการนำข่าวสารจากเฟรมเวิร์คแจ้งไปยังผู้ใช้ ผู้ใช้สามารถรับข่าวสารโดยผ่านตัวกระทำการ onData() ของวัตถุ BTDeviceListener

- ตัวกระทำการ notifyStatus() เป็นตัวกระทำการในการนำสถานะต่างๆ ที่เกิดขึ้นจากเฟรมเวิร์คไปแจ้งให้ผู้ใช้ ผู้ใช้สามารถรับสถานะโดยผ่านตัวกระทำการ onStatus() ของวัตถุ BTDeviceListener
- ตัวกระทำการ sendMessage() เป็นตัวกระทำการเพื่อการส่งข่าวสารจากอุปกรณ์บลูทูธของผู้ใช้ไปยังอุปกรณ์บลูทูธที่ต้องการ โดยใช้วัตถุ BTDeviceHandler ส่งข้อมูล
- ตัวกระทำการ setNewBTDeviceStatus() เป็นตัวกระทำการสำหรับการกำหนดค่าสถานะของอุปกรณ์บลูทูธ
- ตัวกระทำการ putHandler() เป็นตัวกระทำการสำหรับเก็บตัวจัดการการเชื่อมต่อ (วัตถุ BTDeviceHandler) ของอุปกรณ์

3. <u>คลาส BTDeviceNotifierImpl</u>

คลาส BTDeviceNotifierImpl ได้พัฒนาตามส่วนต่อประสานของคลาส BTDeviceNotifier ซึ่งเป็นคลาสสำหรับจัดการวัตถุสเลฟ ฟังก์ชันหลักของคลาส BTDeviceNotifierImpl มีดังต่อไปนี้ ประกาศชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ส่งการปรับปรุงไปยังมาสเตอร์ภายในระยะเวลาที่กำหนด การทำงานภายในของคลาส BTDeviceNotifierImpl ใช้คลาส javax.bluetooth.LocalDevice เพื่อให้อุปกรณ์สามารถค้นพบได้โดยอุปกรณ์อื่น (สเลฟรอการติดต่อสื่อสารจากมาสเตอร์) นำคลาส javax.bluetooth.ServiceRecord มาใช้จัดการฐานข้อมูลบริการ และคลาส BTDeviceNotifierHandler จัดการการรับ-ส่งข้อมูลกับมาสเตอร์ และใช้คลาส TimeNotifyUpdateNameTask เพื่อส่งการปรับปรุงชื่อไปยังมาสเตอร์ตามเวลาที่กำหนด รูปที่ 4.7

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

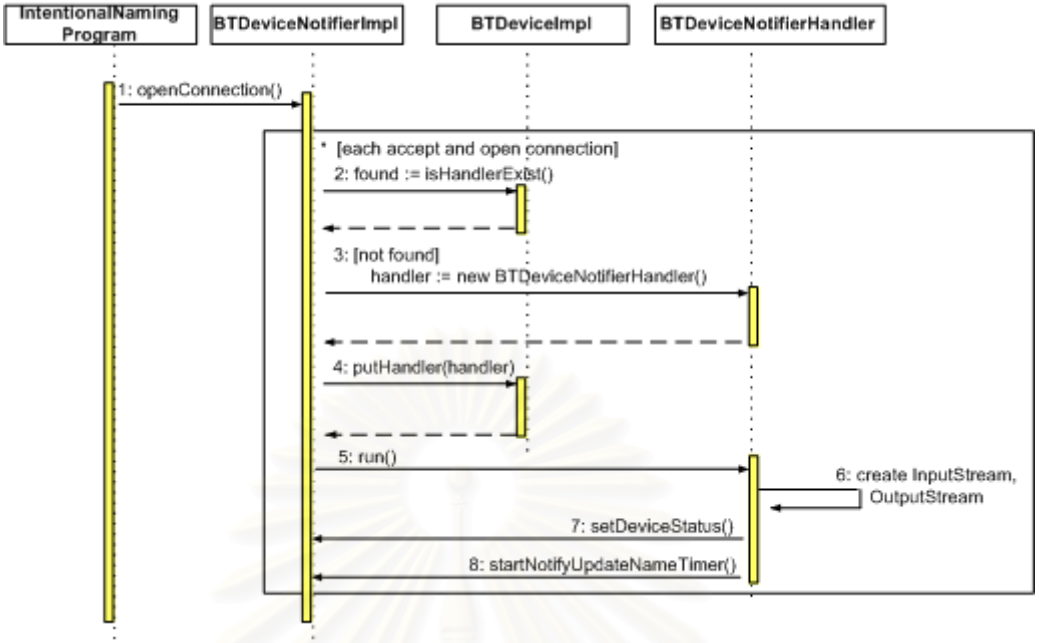


รูปที่ 4.7 แบบจำลองของคลาส BTDviceNotifierImpl

คลาส BTDeviceNotifierImpl ประกอบด้วยตัวกระทำการดังต่อไปนี้

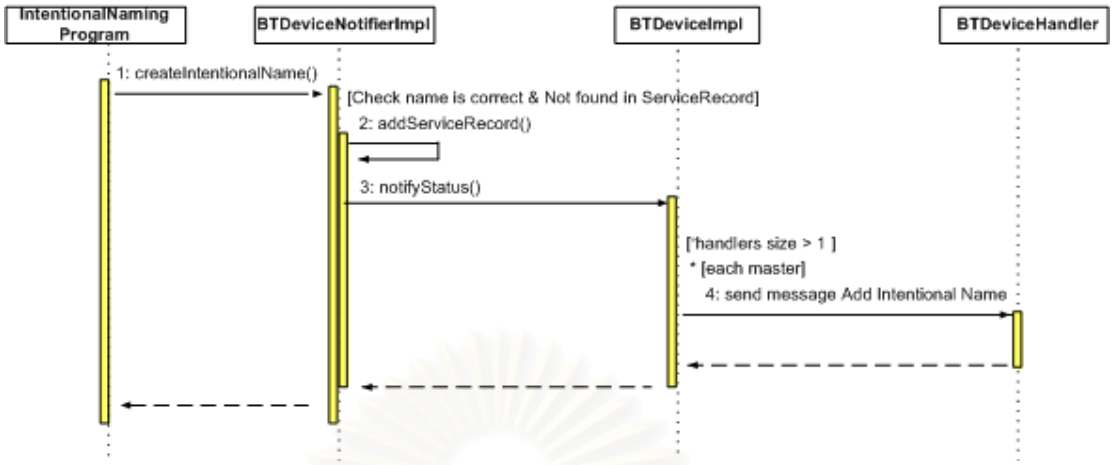
- ตัวกระทำการ openConnection() เป็นตัวกระทำการสำหรับกำหนดค่าเริ่มต้นเพื่อเปิดการติดต่อสื่อสาร โดยเรียกตัวกระทำการ run() เพื่อรอรับการติดต่อจากมาสเตอร์
- ตัวกระทำการ run() ทำหน้าที่รอรับการติดต่อจากมาสเตอร์และสร้างวัตถุ BTDeviceNotifierHandler เมื่อได้รับการเชื่อมต่อจากมาสเตอร์

รูปที่ 4.8 แสดงลำดับขั้นการทำงานเมื่อมีการเรียกใช้งานตัวกระทำการ openConnection() โดยตรวจสอบว่าอุปกรณ์ที่สร้างการติดต่อเข้ามายังสเลฟเป็นอุปกรณ์ที่ได้การสร้างการติดต่อไว้แล้วหรือไม่ (ขั้นตอน 2) ถ้ายังไม่มี จะทำการสร้างวัตถุ BTDeviceNotifierHandler เพื่อคอยจัดการการติดต่อสื่อสารให้กับสเลฟ จากนั้นเก็บค่าอ้างอิงวัตถุไว้ (ขั้นตอน 3-4) และเริ่มต้นการทำงานของวัตถุ BTDeviceNotifierHandler เพื่อทำการสร้างช่องทางการรับ/ส่งข้อมูล ถ้าไม่มีข้อผิดพลาดใดๆ ในการสร้างการเชื่อมต่อก็จะทำการปรับปรุงสถานะสเลฟต่อไป สเลฟที่ได้รับการเชื่อมต่อเข้ามามากกว่า 1 จะมีสถานะเป็นสเลฟ/สเลฟบริดจ์ (ขั้นตอน 5-7) ต่อจากนั้นเริ่มระยะเวลาในการส่งการปรับปรุงไปยังอุปกรณ์ (นั่นคือ มาสเตอร์) ที่เข้ามาเชื่อมต่อ (ขั้นตอน 8)



รูปที่ 4.8 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ openConnection()

- ตัวกระทำการ setDeviceStatus() เป็นตัวกระทำการสำหรับกำหนดสถานะของอุปกรณ์ใหม่ หลังจากที่สเลฟได้รับการเชื่อมต่อจากมาสเตอร์ และหลังจากวัตถุ BTDeviceNotifierHandler พบว่ามาสเตอร์ที่เชื่อมต่อปิดการติดต่อออกไป ทั้งนี้เพื่อปรับปรุงสถานะของอุปกรณ์ให้ถูกต้อง
- ตัวกระทำการ sendUpdateSlaveStatus() เมื่อมีการกำหนดสถานะทุกครั้งวัตถุ BTDeviceNotifierImpl ต้องส่งข่าวสารไปที่มาสเตอร์เพื่อปรับปรุงสถานะ
- ตัวกระทำการ forwardMsgToMasters() เป็นตัวกระทำการที่ถูกเรียกใช้เพื่อให้วัตถุ BTDeviceImpl ส่งข่าวสารไปยังมาสเตอร์ทุกตัว
- ตัวกระทำการ getNeighborRoutersList() ได้ลิสต์ของมาสเตอร์ทุกตัวที่เชื่อมต่ออยู่ด้วย
- ตัวกระทำการ createIntentionalName() ใช้ในการระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ รูปที่ 4.9 ชื่อจะถูกเก็บไว้ทั้งในฐานข้อมูลบริการของวัตถุ BTDeviceNotifierImpl (ขั้นตอน 1-3) และถ้าสเลฟได้รับการติดต่อกับมาสเตอร์ เฟรมเวิร์คจะให้วัตถุ BTDeviceNotifierHandler ส่งชื่อเข้าไปเพิ่มในแผนภาพต้นไม้ของชื่อทางฝั่งมาสเตอร์ด้วย (ขั้นตอน 4)

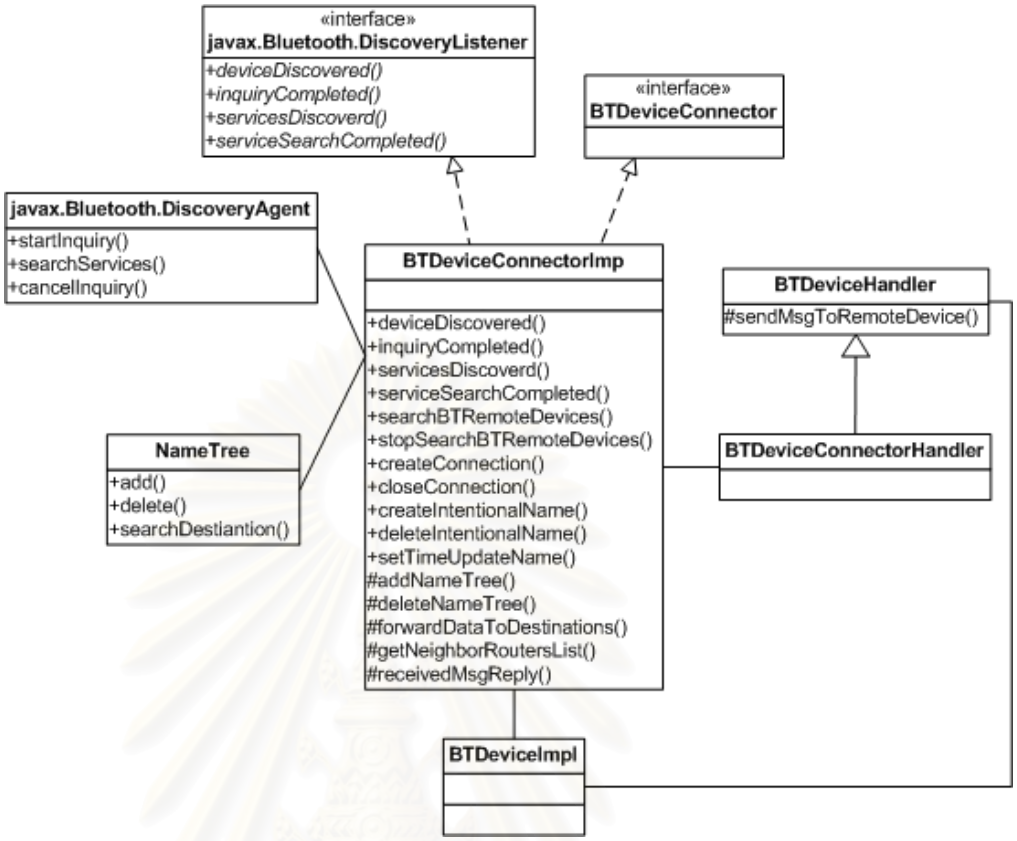


รูปที่ 4.9 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createIntentionalName()

- ตัวกระทำการ deleteIntentionalName() ใช้ในการลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ชื่อที่ผู้ใช้ระบุจะถูกลบออกจากฐานข้อมูลบริการของวัตถุ BTDeviceNotifierImpl และถ้าสเลฟติดต่อกับมาสเตอร์ ก็จะให้วัตถุ BTDeviceNotifierHandler ส่งชื่อไปลบในแผนภาพต้นไม้ของชื่อทางฝั่งมาสเตอร์
- ตัวกระทำการ addServiceRecord() เป็นตัวกระทำการในการเพิ่มชื่อระบุขอบเขตของสิ่งที่ตนเองจะรอให้บริการลงในฐานข้อมูลบริการ
- ตัวกระทำการ removeServiceRecord() เป็นตัวกระทำการในการลบชื่อระบุขอบเขตของสิ่งที่ตนเองจะรอให้บริการออกจากฐานข้อมูลบริการ

4. <u>คลาส BTDeviceConnectorImpl</u>

คลาส BTDeviceConnectorImpl รูปที่ 4.10 พัฒนาขึ้นตามส่วนต่อประสานของคลาส BTDeviceConnector เพื่อทำหน้าที่เป็นอุปกรณ์บลูทูธมาสเตอร์ และพัฒนาขึ้นตามส่วนต่อประสานของคลาส javax.bluetooth.DiscoveryListener โดยใช้คลาส javax.bluetooth.DiscoveryAgent ในการค้นหาอุปกรณ์และบริการจากอุปกรณ์ที่พบ ฟังก์ชันหลักของคลาส BTDeviceConnectorImpl คือประกาศชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ลบชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ รับฟังการประกาศชื่อจากสเลฟและรอสเลฟส่งการปรับปรุงชื่อในระยะเวลาที่กำหนด โดยติดต่อกับคลาส NameTree จากแพคเกจ intentionalnaming.nametree ทำหน้าที่จัดการแผนภาพต้นไม้ของชื่อ ส่วนคลาส BTDeviceConnectorHandler ทำหน้าที่ดูแลจัดการการเชื่อมต่อรับ-ส่งข่าวสารกับสเลฟ

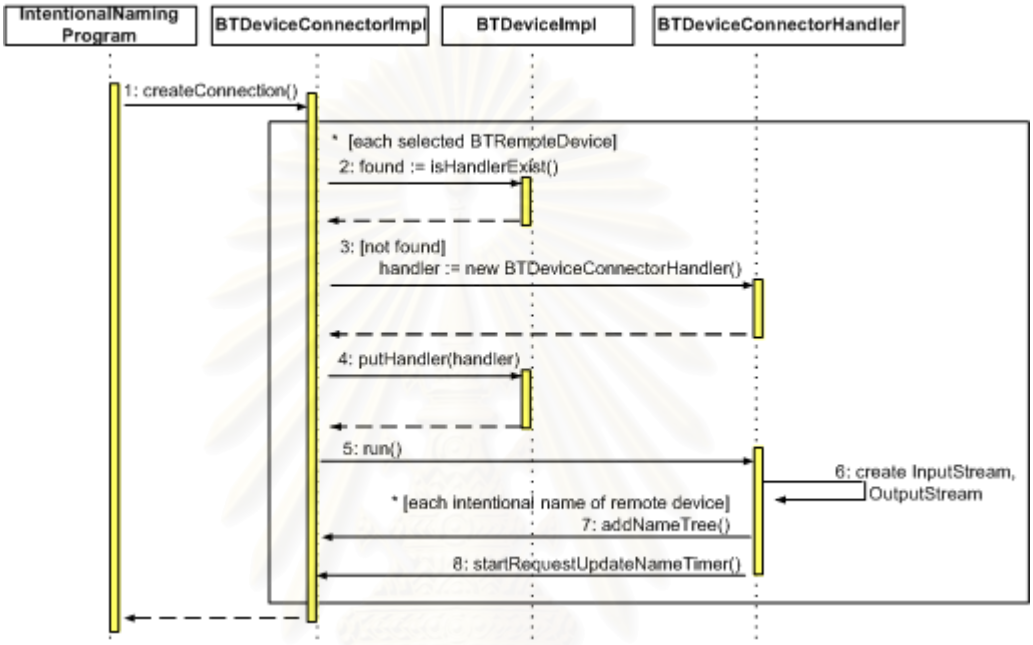


รูปที่ 4.10 แบบจำลองคลาส BTDeviceConnectorImpl

ตัวกระทำการของคลาส BTDeviceConnectorImpl มีรายละเอียดดังต่อไปนี้

- ตัวกระทำการ searchBTRemoteDevices() ทำการค้นหาสเลฟ ซึ่งดำเนินการค้นหาโดยวัตถุ DiscoveryAgent ของเจเอสอาร์-82 เมื่อพบอุปกรณ์แล้วจะจัดเก็บข้อมูลของสเลฟแต่ละตัวที่พบไว้ในวัตถุ BTRemoteDevice แล้วส่งเซตของวัตถุ BTRemoteDevice คืนกลับไปยังผู้ใช้ เพื่อให้ผู้ใช้ตรวจสอบรายละเอียด (ชื่อระบุสิ่งที่รอให้บริการ) ของอุปกรณ์ที่ค้นพบได้
- ตัวกระทำการ stopSearchBTRemoteDevices() เป็นตัวกระทำการยกเลิกการค้นหาอุปกรณ์บลูทูธ ระหว่างที่ตัวดำเนินการ searchBTRemoteDevices() กำลังค้นหาอุปกรณ์
- ตัวกระทำการ createConnection() ทำหน้าที่สร้างการเชื่อมต่อไปยังเซตของสเลฟตามที่ผู้ใช้เลือก รูปที่ 4.11 ถ้าไม่พบวัตถุ BTDeviceConnectorHandler ที่จัดการการเชื่อมต่อกับสเลฟ (ตรวจสอบได้จากเลขที่อยู่บลูทูธของวัตถุ BTRemoteDevice ที่ผู้ใช้เลือก) เฟรมเวิร์คจะสร้างวัตถุ BTDeviceConnectorHandler มาทำหน้าที่ดูแลการเชื่อมต่อ เพื่อป้องกันการสร้างการเชื่อมต่อซ้ำไปยังอุปกรณ์ที่มีการเชื่อมต่อกันอยู่แล้ว

(ขั้นตอน 2-4) และเริ่มต้นการทำงานของวัตถุ BTDeviceConnectorHandler เพื่อทำการสร้างช่องทางการรับ/ส่งข้อมูล ถ้าไม่มีข้อผิดพลาดใดๆ ในการสร้างการเชื่อมต่อก็จะทำการเพิ่มชื่อพบสเลฟประกาศไว้เข้าไปในแผนภาพต้นไม้ของชื่อ (ขั้นตอน 5-7) จากนั้นเริ่มระยะเวลารอสเลฟส่งการปรับปรุงชื่อ (ขั้นตอน 8)



รูปที่ 4.11 แผนภาพลำดับเหตุการณ์เมื่อมีการเรียกใช้งานตัวกระทำการ createConnection()

- ตัวกระทำการ closeConnection() เพื่อปิดการติดต่อสื่อสารกับเซตของสเลฟที่ผู้ใช้เลือก
- ตัวกระทำการ createIntentionalName() ใช้ในการระบุชื่อเพื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ ชื่อที่ประกาศไว้จะถูกนำไปจัดเก็บในวัตถุ NameTree
- ตัวกระทำการ deleteIntentionalName() ใช้ในการลบชื่อที่เคยประกาศไว้ออกจากวัตถุ NameTree
- ตัวกระทำการ setTimeUpdateName() ทำหน้าที่กำหนดระยะเวลารอรับการปรับปรุงชื่อจากสเลฟ ถ้าวัตถุ BTDeviceConnector ไม่ได้รับการปรับปรุงจากสเลฟมากกว่า 3 ช่วงเวลา จะถือว่าสเลฟปิดการติดต่อไปแล้ว จากนั้นวัตถุ BTDeviceConnector จะทำการลบชื่อของสเลฟออกจากวัตถุ NameTree เพื่อปรับปรุงข้อมูลที่เก็บในวัตถุ NameTree ให้มีความถูกต้องอยู่เสมอ
- ตัวกระทำการ addNameTree() แบ่งออกเป็น 2 แบบ ดังนี้

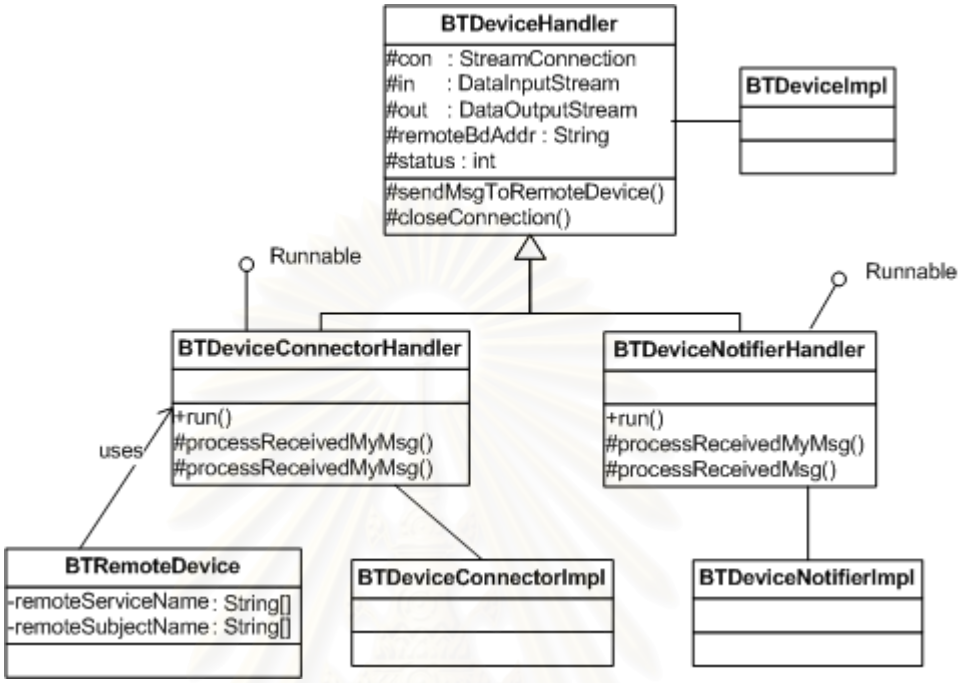
i. addNameTree(BTRemoteDevice btRemoteDevice) เป็นตัวกระทำการที่ถูกเรียกใช้ขณะมาสเตอร์สร้างการติดต่อไปยังสเลฟได้ เพื่อนำชื่อที่พบทั้งหมดจากวัตถุ BTRemoteDevice (ชื่อระบุสิ่งที่รอให้บริการ) เพิ่มเข้าไปในวัตถุ NameTree

ii. addNameTree(String service, String subject, String bdAddr) เป็นตัวกระทำการเพิ่มชื่อที่ได้จากการระบุ ชื่อบริการ ชื่ออุปกรณ์และเลขที่อยู่บลูทูธของอุปกรณ์ที่ประกาศชื่อ เข้าไปเก็บไว้วัตถุ NameTree

วัตถุ BTDeviceConnector จะส่งข่าวสารแสดงผลการเพิ่มชื่อที่ได้รับจากวัตถุ NameTree ไปให้อุปกรณ์ที่ทำการประกาศชื่อ ด้วยสถานะดังต่อไปนี้ _STATUS_NAMETREE_ADDNAME_SUCCESS เมื่อสามารถเพิ่มชื่อได้ หรือ _STATUS_NAMETREE_ADDNAME_FAIL เมื่อไม่สามารถเพิ่มชื่อได้

- ตัวกระทำการ deleteNameTree() แบ่งได้ 2 กรณี ดังต่อไปนี้
  i. deleteNameTree(String bd) เป็นตัวกระทำการที่ทำการลบชื่อทั้งหมดตามเลขที่อยู่บลูทูธของอุปกรณ์ กรณีที่มีการเรียกใช้คือ เมื่อพบว่าสเลฟปิดการติดต่อไป และกรณีที่ไม่ได้รับการปรับปรุงชื่อจากสเลฟเกินกว่าช่วงเวลาที่กำหนด
  ii. deleteNameTree(String servName, String subjName, String bd) ทำการลบชื่อ โดยระบุ ชื่อบริการ ชื่ออุปกรณ์และเลขที่อยู่บลูทูธของอุปกรณ์ที่ประกาศชื่อออกจากวัตถุ NameTree
- ตัวกระทำการ getIntentDestination() เป็นตัวกระทำการค้นหาเซตของเลขที่อยู่บลูทูธของอุปกรณ์ที่สอดคล้องกับชื่อที่ระบุ จากวัตถุ NameTree
- ตัวกระทำการ forwardDataToDestinations() ส่งข้อมูลไปยังโหนดปลายทางตามเลขที่อยู่บลูทูธที่ได้จากตัวกระทำการ getIntentDestination()
- ตัวกระทำการ getNeighborRoutersList() ทำหน้าที่ค้นหาอุปกรณ์ข้างเคียงที่เป็นอุปกรณ์จัดเส้นทางบลูทูธ โดยการตรวจสอบได้จากสถานะของวัตถุ BTDeviceHandler ที่มีสถานะเป็น SS_BRIDGE, MS_BRIDGE หรือ MASTER
- ตัวกระทำการ receivedMsgReply(String msg) เป็นตัวกระทำการที่ถูกเรียกใช้เมื่อได้รับข่าวสารชนิด _USR_SEND_REPLY มาจากสเลฟ ถ้าหากว่าเลขที่อยู่บลูทูธของโหนดต้นทางตรงกับเลขที่อยู่บลูทูธของตนเองก็จะแจ้งข่าวสารไปให้ผู้ใช้ทราบด้วยตัวกระทำการ notifyData() แต่ถ้าไม่ใช่ก็จะให้วัตถุ BTDeviceRouter จัดการส่งข้อมูลต่อไป

5. <u>คลาส BTDeviceHandler</u>



รูปที่ 4.12 แบบจำลองคลาส BTDeviceHandler

อธิบายจากรูปที่ 4.12 คลาส BTDeviceHandler ทำหน้าที่หลักในการจัดเก็บข้อมูลที่เกี่ยวข้องกับอุปกรณ์ข้างเคียงที่มีการเชื่อมต่อกันอยู่ อาทิเช่น เลขที่อยู่บลูทูธ สถานะของอุปกรณ์ข้างเคียง วัตถุสำหรับจัดการเกี่ยวกับการเชื่อมต่อ ประกอบด้วย วัตถุ StreamConnection วัตถุ DataInputStream และวัตถุ DataOutputStream เป็นต้น ฟังก์ชันหลักคือจัดการการติดต่อสื่อสารและส่งข้อมูล ตัวกระทำการของคลาส BTDeviceHandler มีดังนี้

- ตัวกระทำการ sendMsgToRemoteDevice() เพื่อส่งข้อมูลออกไปยังอุปกรณ์ข้างเคียงที่ต้องการ
- ตัวกระทำการ closeConnection() เมื่อตรวจสอบพบว่าอุปกรณ์ข้างเคียงปิดไปก็จะแจ้งให้ผู้ใช้ทราบ แล้วปิดการทำงานพร้อมทั้งลบวัตถุ BTDeviceHandler ออกจากอุปกรณ์
- ตัวกระทำการ processReceivedMyMsg() เป็นตัวกระทำการที่ให้ซับคลาสนำไปประยุกต์ใช้ตามความเหมาะสม เมื่อได้รับข่าวสารสำหรับอุปกรณ์บลูทูธ

- ตัวกระทำการ processReceivedMsg() เป็นตัวกระทำการที่ให้ซับคลาสนำไปประยุกต์ใช้ตามความเหมาะสม เมื่อได้รับข่าวสารสำหรับระบบและข่าวสารเพื่อประมวลผลจากคำสั่งของผู้ใช้

แบบจำลองคลาส รูปที่ 4.12 ประกอบด้วย คลาส BTDeviceNotifierHandler และคลาส BTDeviceConnectorHandler เป็นคลาสที่สืบทอดมาจากคลาส BTDeviceHandler กล่าวคือ วัตถุที่สร้างจากคลาส BTDeviceNotifierHandler และคลาส BTDeviceConnectorHandler มีพฤติกรรมที่แตกต่างกัน ทั้งขั้นตอนการสร้างวัตถุ และข้อมูลที่รับเข้ามาบางส่วนมีความแตกต่างกันทำให้กระบวนการทำงานต่างกันออกไป

6. <u>คลาส BTDeviceNotifierHandler</u>

ฟังก์ชันหลักของคลาส BTDeviceNotifierHandler ประกอบด้วย การจัดการการติดต่อสื่อสารให้กับวัตถุ BTDeviceNotifierImpl โดยที่วัตถุ BTDeviceNotifierHandler มีการติดต่อสื่อสารกับวัตถุ BTDeviceConnectorHandler ของฝั่งมาสเตอร์ และรับข้อมูล (input) ที่ถูกส่งเข้ามายังสเลฟ

จากแบบจำลองคลาส BTDeviceNotifierHandler แสดงในรูปที่ 4.12 ประกอบด้วย ตัวกระทำการ ดังนี้

- ตัวกระทำการ processReceivedMyMsg() เป็นตัวกระทำการที่สำหรับประมวลผลข่าวสารสำหรับอุปกรณ์บลูทูธ ประเภทข่าวสารที่ส่งเข้ามาโดยเฉพาะคือ ข่าวสารการแจ้งสถานะการเพิ่มหรือลบข้อมูลในแผนภาพต้นไม้ของชื่อที่ส่งมาจากมาสเตอร์ ดังนี้
  _STATUS_NAMETREE_ADDNAME_SUCCESS,
  _STATUS_NAMETREE_ADDNAME_FAIL,
  _STATUS_NAMETREE_DELNAME_SUCCESS,
  _STATUS_NAMETREE_DELNAME_FAIL
- ตัวกระทำการ processReceivedMsg() เป็นตัวกระทำการสำหรับรับข่าวสารเพื่อส่งต่อ

7. <u>คลาส BTDeviceConnectorHandler</u>

ฟังก์ชันหลักของคลาส BTDeviceConnectorHandler ประกอบด้วย การจัดการการติดต่อสื่อสารให้กับวัตถุ BTDeviceConnectorImpl โดยที่วัตถุ BTDeviceConnectorHandler มีการติดต่อสื่อสารกับวัตถุ BTDeviceNotifierHandler ของฝั่งสเลฟ และรับข้อมูล (input) ที่ถูก

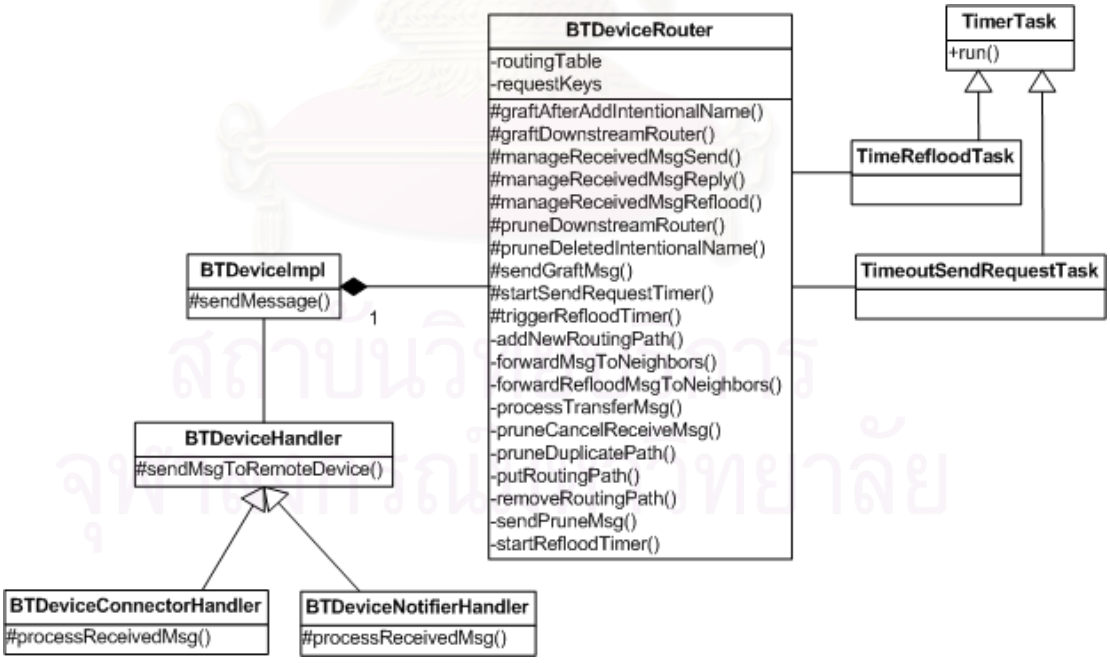
ส่งเข้ามายังมาสเตอร์ วัตถุ BTDeviceConnectorHandler เก็บข้อมูลของสเลฟที่เชื่อมต่ออยู่ด้วยโดยนำข้อมูลมาจากวัตถุ BTRemoteDevice

แบบจำลองคลาส BTDeviceConnectorHandler แสดงในรูปที่ 4.12 ประกอบด้วย ตัวกระทำการ ดังนี้

- ตัวกระทำการ processReceivedMyMsg() รับข่าวสารชนิด _USR_SEND_REPLY ที่ส่งมาจากอุปกรณ์จัดเส้นทางบลูทูธ
- ตัวกระทำการ processReceivedMsg() เป็นตัวกระทำการสำหรับรับข้อมูลเพื่อส่งต่อและข่าวสารที่ส่งโดยสเลฟ (สำหรับประเภทข่าวสารที่ส่งเข้ามายังวัตถุ BTDeviceConnectorHandler โดยเฉพาะ มีดังต่อไปนี้ BR, NOTIFY_UPDATENAME, _USR_ADDNAME, _USR_DELNAME)

8. <u>คลาส BTDeviceRouter</u>

ฟังก์ชันหลักของคลาส BTDeviceRouter ประกอบด้วย การกระจายข้อมูล การตัดเส้นทาง การต่อเส้นทาง การกระจายข้อมูลอีกครั้ง



รูปที่ 4.13 แบบจำลองคลาส BTDeviceRouter

อธิบายจากรูปที่ 4.13 แบบจำลองคลาส BTDeviceRouter คลาส BTDeviceRouter ทำหน้าที่จัดการเส้นทางการส่งข้อมูลให้กับอุปกรณ์ ในการส่งข้อมูลจะเรียกใช้คลาส

BTDeviceHandler เพื่อจัดส่งข้อมูลไปยังอุปกรณ์ที่ต้องการ เมื่อโหนดต้นทางระบุชื่อปลายทางในการส่งข้อมูล คลาส TimeRefloodTask จะถูกใช้สำหรับกระบวนการกระจายข้อมูลอีกครั้ง และในกรณีที่โหนดต้นทางส่งข้อมูลแบบขอการตอบกลับจะใช้คลาส TimeoutSendRequestTask สำหรับจัดการเวลาเมื่อหมดระยะเวลารอการตอบกลับจากโหนดปลายทาง รวมทั้งจัดเส้นทางการส่งต่อข้อมูลที่ได้รับเข้ามาจากคลาส BTDeviceConnectorHandler หรือคลาส BTDeviceNotifierHandler

คุณลักษณะของคลาส BTDeviceRouter มีดังนี้

- requestKeys โครงสร้างข้อมูลเป็น java.Util.Vector ทำหน้าที่เก็บคีย์ในกรณีที่อุปกรณ์เป็นโหนดปลายทางที่ได้รับข่าวสารชนิด _USR_SEND_REQUEST ทั้งนี้ถ้าโหนดปลายทางไม่ได้รับข่าวสารดังกล่าวจะไม่สามารถส่งข้อมูลตอบกลับไปยังโหนดต้นทางได้
- routingTable มีโครงสร้างข้อมูลเป็น java.Util.Hashtable ทำหน้าที่เป็นตารางเส้นทาง ถูกสร้างขึ้นในวัตถุ BTDeviceRouter ตัวอย่างข้อมูลที่เก็บในตารางเส้นทาง ตารางที่ 4.10

ตารางที่ 4.10 ตัวอย่างตารางเส้นทาง

คีย์ (Key)	**ข้อมูลเส้นทาง** (Routing Path)
SourceBDAddr1:serviceName1,subjectName1	{UpRouterBDAddr}{DownRouterBDAddr,…,}
SourceBDAddr2:serviceName2,subjectName2	{UpRouterBDAddr}{DownRouterBDAddr,…,}
.	.
.	.

คำอธิบายค่าในตารางที่ 4.10

ค่าคีย์ของตารางเส้นทาง เป็นค่าที่ไม่ซ้ำกัน ประกอบด้วย

- SourceBDAddr หมายถึง เลขที่อยู่บลูทูธของโหนดต้นทาง
- serviceName, subjectName เป็นการระบุชื่อเพื่อการส่งไปยังปลายทาง

ข้อมูลเกี่ยวกับเส้นทาง เป็นเส้นทางรับ-ส่งข้อมูล ประกอบด้วย

- UpRouterBDAddr เป็นเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา
- DownRouterBDAddr เป็นเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อ

หมายเหตุ แต่ละเส้นทางในตาราง เลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามามีได้เพียงตัวเดียวเท่านั้น ส่วนเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อมีจำนวนขึ้นอยู่กับอุปกรณ์บลูทูธข้างเคียงในขณะนั้น หรืออาจไม่มีถ้าไม่มีอุปกรณ์ใดในเครือข่ายระบุชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการที่สอดคล้องกับชื่อที่โหนดต้นทางระบุ

ตัวกระทำการของคลาส BTDeviceRouter มีดังต่อไปนี้

- ตัวกระทำการ addNewRoutingPath() ถูกเรียกใช้เพื่อเตรียมข้อมูลสำหรับเพิ่มเส้นทางใหม่ (พิจารณาจากค่าคีย์ของตารางเส้นทาง ที่ยังไม่ปรากฏในตารางเส้นทาง) จากนั้นเรียกตัวกระทำการ putRoutingPath() เพื่อจัดเก็บเส้นทางใหม่ลงในตารางเส้นทาง
- ตัวกระทำการ putRoutingPath() เพื่อเก็บเส้นทางลงในตารางเส้นทาง ทั้งการเพิ่มเส้นทางใหม่ หรือการปรับปรุงแก้ไขเส้นทางเดิม
- ตัวกระทำการ removeRoutingPath() ทำหน้าที่ลบเส้นทางเมื่อเส้นทางเก่าที่ใช้การไม่ได้ หรือเมื่อได้รับข่าวสารของระบบแจ้งการกระจายข้อมูลอีกครั้งถ้าระบุตรงกับคีย์ที่มีในตาราง ก็จะลบเส้นทางที่ได้รับแจ้งออกจากตารางเส้นทาง
- ตัวกระทำการ manageReceivedMsgSend() ถูกเรียกใช้เมื่ออุปกรณ์ได้รับข่าวสารชนิด _USR_SEND หรือ _USR_SEND_REQUEST ทำการหาเส้นทางที่ใช้ส่งต่อข้อมูล โดยตรวจสอบจากคีย์ของเส้นทางในตารางเส้นทางก่อน แบ่งได้ 2 กรณี

  กรณี 1 ถ้าไม่พบเส้นทาง ทำการเพิ่มเส้นทางใหม่ และถ้าอุปกรณ์เป็นโหนดต้นทางจะเรียกตัวกระทำการ triggerRefloodTimer() เพื่อเริ่มระยะเวลาการดำเนินการกระจายข้อมูลอีกครั้ง

  กรณี 2 ถ้ามีเส้นทางในตารางเส้นทาง จะตรวจสอบเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา (UpRouterBDAddr) ถ้าต่างกันแสดงว่าได้รับข้อมูลเดิมซ้ำมาจากหลายเส้นทางจะส่งข่าวสารเพื่อตัดเส้นทาง แต่ถ้าเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามาตรงกับที่พบในข้อมูลเส้นทาง ก็จะใช้ข้อมูลเส้นทางเดิม

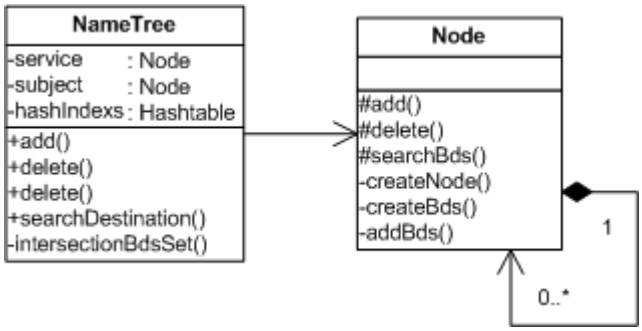
  เมื่อได้เส้นทางที่จะใช้ส่งแล้ว จากนั้นส่งต่อหน้าที่ให้ตัวกระทำการ processTransferMsg() เพื่อจัดส่งข่าวสารต่อไป
- ตัวกระทำการ manageReceivedMsgReply() ถูกเรียกใช้เมื่ออุปกรณ์ได้รับข่าวสารชนิด _USR_SEND_REPLY
- ตัวกระทำการ manageReceivedMsgReflood() ถูกเรียกใช้เมื่ออุปกรณ์ได้รับข่าวสารแจ้งการกระจายข้อมูลอีกครั้ง ถ้าพบคีย์ที่ระบุในตารางเส้นทาง ก็จะลบเส้นทางนั้นออก

จากตาราง แล้วส่งต่อไปยังอุปกรณ์ข้างเคียงด้วยตัวกระทำการ forwardMsgRefloodToNeighbors()

- ตัวกระทำการ processTransferMsg() เป็นตัวกระทำการเพื่อจัดการส่งต่อข้อมูลโดยพิจารณาจากสถานะของอุปกรณ์
- ตัวกระทำการ pruneDuplicatePath() ส่งข่าวสารแจ้งการตัดเส้นทางไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา กรณีที่ได้รับข้อมูลเดิมซ้ำจากหลายเส้นทาง
- ตัวกระทำการ pruneCancelReceivedMsg() ส่งข่าวสารแจ้งการตัดเส้นทางไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา กรณีไม่ต้องการรับข้อมูล และทำเครื่องหมาย “P” ไว้หน้าข้อมูลเส้นทางนั้น เพื่อให้สามารถตรวจสอบสำหรับใช้ในการต่อเส้นทางได้ภายหลัง
- ตัวกระทำการ pruneDownStreamRouter() เมื่อได้รับข่าวสารตัดเส้นทางจากอุปกรณ์จัดเส้นทาง จะทำการลบอุปกรณ์จัดเส้นทางนั้นออกเซตของอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อของเส้นทางตามที่ได้รับแจ้งมา และถ้าไม่มีอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อเหลืออยู่ในเส้นทาง จะทำการส่งข่าวสารตัดเส้นทางไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา โดยเรียกใช้ตัวกระทำการ sendPruneMsg()
- ตัวกระทำการ pruneAfterDeletedIntentionalName() ถูกเรียกโดยมาสเตอร์เพื่อตรวจสอบชื่อที่ถูกลบไปจากส่วนระบุปลายทางของเส้นทาง ถ้าไม่มีอุปกรณ์จัดเส้นทางตัวถัดไปที่ต้องการข้อมูลก็จะเรียกตัวกระทำการ pruneCancelReceivedMsg()
- ตัวกระทำการ sendPruneMsg() ส่งข่าวสารเพื่อการตัดเส้นทางส่งไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา ถ้าตัวจัดการการเชื่อมต่อของอุปกรณ์จัดเส้นทางยังอยู่ แต่ถ้าตรวจสอบไม่พบตัวจัดการการเชื่อมต่อ ก็จะลบเส้นทางออกจากตารางเส้นทาง เนื่องจากเป็นเส้นทางเก่าที่ข้อมูลไม่ถูกต้อง
- ตัวกระทำการ graftDownStreamRouter() เมื่อได้รับข่าวสารต่อเส้นทางจากอุปกรณ์จัดเส้นทาง จะทำการเพิ่มเลขที่อยู่บลูทูธของอุปกรณ์จัดเส้นทางที่จะรับข้อมูลต่อ เข้าไปในเส้นทางตามที่ได้รับแจ้งมา และตรวจสอบต่อไปถ้าหากว่าเส้นทางมีเครื่องหมาย “P” หมายถึงถูกตัดเส้นทางไว้ก็ให้ส่งข่าวสารต่อเส้นทางไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา โดยเรียกตัวกระทำการ sendGraftMsg()
- ตัวกระทำการ graftAfterAddIntentionalName() ถูกเรียกโดยมาสเตอร์เพื่อตรวจสอบชื่อที่เพิ่มใหม่สอดคล้องกับชื่อที่พบในคีย์และมีเครื่องหมาย “P” หรือไม่ ถ้ามีก็จะส่งข้อความต่อเส้นทางด้วยตัวกระทำการ sendGraftMsg()

- ตัวกระทำการ sendGraftMsg() ส่งข่าวสารต่อเส้นทาง โดยทำการตรวจสอบตัวจัดการการเชื่อมต่อของอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามายังคงอยู่หรือไม่ ถ้าไม่มีก็จะลบเส้นทางออกจากตารางเส้นทาง เนื่องจากเป็นเส้นทางเก่าที่มีข้อมูลไม่ถูกต้อง แต่ถ้าตัวจัดการการเชื่อมต่อของ อุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามายังอยู่ ก็จะสร้างข่าวสารเพื่อการต่อเส้นทางส่งไปยังอุปกรณ์จัดเส้นทางที่ส่งข้อมูลเข้ามา
- ตัวกระทำการ forwardMsgToNeighbors() ทำการตรวจสอบเส้นทางเพื่อส่งข้อมูลไปยังกลุ่มของอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อ พร้อมทั้งตรวจสอบสถานะของอุปกรณ์จัดเส้นทางตัวถัดไปที่จะรับข้อมูลต่อ หากว่าสถานะของอุปกรณ์เปลี่ยนไปเป็น SLAVE ก็จะลบออกจากกลุ่มของอุปกรณ์จัดเส้นทาง
- ตัวกระทำการ forwardMsgRefloodToNeighbors() เมื่ออุปกรณ์จัดเส้นทางบลูทูธได้รับข่าวสารการกระจายข้อมูลอีกครั้ง จะทำการส่งข่าวสารต่อไปยังอุปกรณ์บลูทูธข้างเคียงทุกตัวที่ไม่พบในเซตของอุปกรณ์จัดเส้นทางที่เคยได้รับข่าวสารนี้ (อุปกรณ์จัดเส้นทางบลูทูธที่เคยได้รับข่าวสารนี้แล้ว จะถูกเพิ่มเข้าไปเป็น passRtr ในข่าวสาร) ทั้งนี้มีจุดประสงค์เพื่อป้องกันการส่งข่าวสารดังกล่าววนซ้ำในเครือข่ายบลูทูธ
- ตัวกระทำการ triggerRefloodTimer() ทำหน้าที่ตรวจสอบจากชื่อระบุการส่งไปยังปลายทางของข่าวสาร เมื่ออุปกรณ์ของตนเองเป็นโหนดต้นทางในการส่งข้อมูล (_USR_SEND, _USR_SEND_REQUEST) โดยตรวจสอบก่อนว่ายังอยู่ในช่วงเวลารอการกระจายข้อมูลอีกครั้งหรือไม่ ถ้าไม่พบจึงเรียกตัวกระทำการ startRefloodTimer()
- ตัวกระทำการ startRefloodTimer() เพื่อเริ่มต้นระยะเวลาในการส่งข่าวสารกระจายข้อมูลอีกครั้ง
- ตัวกระทำการ startSendRequestTimer() เริ่มระยะเวลารอข่าวสารตอบจากโหนดปลายทาง เมื่ออุปกรณ์ของตนเองเป็นโหนดต้นทางในการส่งข้อมูลประเภท _USR_SEND_REQUEST

4.3 **แพคเกจ** intentionalnaming.nametree



รูปที่ 4.14 แบบจำลองคลาสของแพคเกจ intentionalnaming.nametree

แพคเกจ intentionalnaming.nametree (ถูกเรียกใช้โดยคลาส BTDeviceConnectorImpl จากแพคเกจ intentionalnaming.bluetooth.connection.impl) รูปที่ 4.14 ทำหน้าที่จัดการแผนภาพต้นไม้ของชื่อและโหนดภายในแผนภาพต้นไม้ โหนดแต่ละโหนดเชื่อมต่อกันในลักษณะของโหนดแม่ไปยังโหนดลูกจนกระทั่งถึงโหนดใบ ซึ่งโหนดใบเป็นโหนดที่เก็บเลขที่อยู่บลูทูธของอุปกรณ์ที่สอดคล้องกับชื่อ คลาสต่างๆ ในแพคเกจ intentionalnaming.nametree อธิบายได้ดังต่อไปนี้

1. <u>คลาส NameTree</u>

ฟังก์ชันหลักของคลาส NameTree ประกอบด้วย การเพิ่มชื่อเข้าไปในแผนภาพต้นไม้ การลบชื่อออกจากแผนภาพต้นไม้ และการค้นหาอุปกรณ์ปลายทางจากแผนภาพต้นไม้ของชื่อ ส่วนประกอบของคลาส NameTree อธิบายได้ดังต่อไปนี้

<u>ลักษณะประจำของคลาส NameTree</u>

- Node service ใช้เป็นโหนดเริ่มต้นคุณลักษณะบริการ
- Node subject ใช้เป็นโหนดเริ่มต้นคุณลักษณะชื่ออุปกรณ์
- Hashtable hashIndexs ใช้เป็นตารางในการเก็บชื่อและเลขที่อยู่บลูทูธของอุปกรณ์ที่สอดคล้องกับชื่อ เนื่องจากอุปกรณ์หนึ่งสามารถมีได้หลายชื่อ ดังนั้นเพื่อความถูกต้องของข้อมูลก่อนทำการลบ จึงต้องมีการตรวจสอบจากตาราง hashIndexs ก่อน

ตารางที่ 4.11 ตัวอย่างรูปแบบข้อมูลที่เก็บในตาราง hashIndexs

ค่าคีย์ (key)	**ข้อมูล** (String)
Bd1	“ Service1, Subject1 | Service2, Subject2 | ”
Bd5	“ mail, group1.a | ”
…	…

หมายเหตุ ค่าที่พบจากตารางที่ 4.11 อธิบายได้ดังนี้

- คีย์ เป็น Bd (เลขที่อยู่บลูทูธของอุปกรณ์ที่ประกาศชื่อ)
- ใช้ , เป็นตัวแบ่งชื่อซึ่งประกอบด้วย Service และ Subject
- ใช้ | เป็นตัวแบ่งชุดข้อมูลของแต่ละชื่อ เนื่องจากอุปกรณ์เดียวอาจมีหลายชื่อ

ตัวกระทำการของคลาส NameTree

- ตัวกระทำการ add() เพื่อชื่อเข้าไปตามโหนด service และโหนด subject
- ตัวกระทำการ delete() เพื่อลบชื่อออกจากโหนดลูกของโหนด service และโหนด subject
- ตัวกระทำการ searchDestination() เพื่อค้นหาเลขที่อยู่บลูทูธของโหนดปลายทางตามชื่อที่ระบุจากโหนด service และโหนด subject
- ตัวกระทำการ intersectionBdsSet() นำเซตของเลขที่อยู่บลูทูธของการค้นหาจากโหนด service และโหนด subject มาทำการอินเตอร์เซกชัน สุดท้ายจะได้เซตของเลขที่อยู่บลูทูธที่สอดคล้องกับทั้งโหนด service และโหนด subject

2. คลาส Node

ฟังก์ชันหลักของคลาส Node ประกอบด้วย การสร้างโหนด การลบโหนด และการค้นหาเลขที่อยู่บลูทูธของอุปกรณ์จากโหนด ตัวกระทำการของคลาส Node อธิบายได้ดังนี้

- ตัวกระทำการ add() สร้างโหนดตามชื่อที่ระบุ เพิ่มต่อเข้าไปในโหนด service หรือ subject (ขั้นตอนการเพิ่ม แสดงในบทที่ 3) และเก็บข้อมูลเพิ่มเข้าในตาราง hashIndexs
- ตัวกระทำการ delete() ทำหน้าที่ลบโหนดตามชื่อและเลขที่อยู่บลูทูธตามที่ระบุ โดยตรวจสอบจากตาราง hashIndexs ก่อนว่าชื่อและเลขที่อยู่บลูทูธถูกต้อง จึงจะทำการลบได้ (ขั้นตอนการลบ แสดงในบทที่ 3)
- ตัวกระทำการ searchBds() ค้นหาเซตของเลขที่อยู่บลูทูธของโหนด (โหนดเริ่มต้นคือโหนด service และโหนด subject) (ขั้นตอนการค้นหา แสดงในบทที่ 3)
- ตัวกระทำการ createNode() สร้างโหนดใหม่
- ตัวกระทำการ createBds() สร้างเซตของเลขที่อยู่บลูทูธเก็บไว้ในโหนดใบ
- ตัวกระทำการ addBDs() เพิ่มเลขที่อยู่บลูทูธเข้าไปในเซตของเลขที่อยู่บลูทูธของโหนดใบ (ในกรณีที่โหนดใบได้สร้างเซตของเลขที่อยู่บลูทูธไว้ก่อนแล้ว)

บทที่ 5

ทดสอบการใช้งานเฟรมเวิร์ค

5.1 สภาวะที่ใช้ในการทดสอบ

1. เครื่อง Intel Celeron 2.20 กิกะเฮิร์ต หน่วยความจำขนาด 512 เมกะไบต์ ระบบปฏิบัติการวินโดว์ เอ็กซ์พี เวอร์ชัน 2002
2. เครื่องมือพัฒนาภาษาจาวา เจดีเค 1.4
3. เจเอสอาร์-82 (Java Specification Request 82 – JSR-82) เป็นจาวาเอพีไอสำหรับบลูทูธและเป็นมาตรฐานเปิดสำหรับพัฒนาโปรแกรมประยุกต์ผ่านบลูทูธรุ่นที่ 1.1
4. ตัวจำลองการทำงาน เจทูเอ็มอี ไวร์เลส ทูลคิท (J2ME Wireless Toolkit) รุ่นที่ 2.2

5.2 ทดสอบการทำงานของเฟรมเวิร์ค

5.2.1 ทดสอบการจัดการในพิโคเน็ต

เริ่มการทดสอบโดยกำหนดอุปกรณ์ให้เป็นสเลฟ หรือมาสเตอร์เพื่อการติดต่อสื่อสารภายในพิโคเน็ต จากนั้นทดสอบการประกาศชื่อกำหนดขอบเขตของสิ่งที่ตนเองจะรอให้บริการ

ขั้นตอนการสร้างอุปกรณ์สเลฟ

1. เรียกใช้งานตัวกระทำการ createBTDevice() ของส่วนต่อประสาน BTDeviceFactory เพื่อสร้างวัตถุ BTDevice สำหรับทำหน้าที่เป็นอุปกรณ์บลูทูธ
2. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ createBTDeviceNotifier() เพื่อกำหนดบทบาทให้อุปกรณ์เป็นสเลฟ จะได้วัตถุ BTDeviceNotifier
3. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ addBTDeviceListener() เพื่อรับข่าวสาร
4. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ setTimeReflood() เพื่อกำหนดเวลาเริ่มกระบวนการกระจายข้อมูลอีกครั้ง
5. วัตถุ BTDeviceNotifier เรียกใช้งานตัวกระทำการ setTimeUpdateName() เพื่อกำหนดระยะเวลาส่งการปรับปรุงไปยังตัวจัดการชื่อ (มาสเตอร์) ถ้าหากไม่ส่งภายในระยะเวลาที่กำหนดชื่อจะถูกลบออกจากแผนภาพต้นไม้
6. วัตถุ BTDeviceNotifier เรียกใช้งานตัวกระทำการ openConnection() เพื่อรอรับการติดต่อจากมาสเตอร์

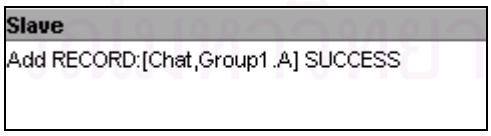
ขั้นตอนการสร้างอุปกรณ์มาสเตอร์

1. เรียกใช้งานตัวกระทำการ createBTDevice() ของส่วนต่อประสาน BTDeviceFactory เพื่อสร้างวัตถุ BTDevice สำหรับทำหน้าที่เป็นอุปกรณ์บลูทูธ
2. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ createBTDeviceConnector() เพื่อกำหนดบทบาทให้อุปกรณ์เป็นมาสเตอร์ จะได้วัตถุ BTDeviceConnector
3. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ addBTDeviceListener() เพื่อรับฟังประกาศและข่าวสาร
4. วัตถุ BTDevice เรียกใช้งานตัวกระทำการ setTimeReflood() เพื่อกำหนดระยะเวลาการส่งแบบกระจายข้อมูลอีกครั้ง
5. วัตถุ BTDeviceConnector เรียกใช้งานตัวกระทำการ setTimeUpdateName() เพื่อกำหนดระยะเวลาให้สเลฟส่งการปรับปรุงชื่อ ภายในระยะเวลาที่กำหนด

ทดสอบการเพิ่ม-ลบชื่อ

เมื่อสร้างวัตถุ BTDeviceNotifier หรือ BTDeviceConnector แล้วสามารถเรียกตัวกระทำการ createIntentionalName() สำหรับเพิ่มชื่อ และตัวกระทำการ deleteIntentionalName() สำหรับลบชื่อที่ไม่ต้องการ

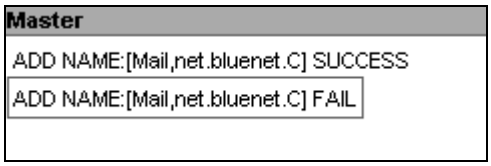
ตัวอย่าง สเลฟประกาศชื่อ โดยเรียกตัวกระทำการ createIntentionalName() ของส่วนต่อประสาน BTDeviceNotifier รูปที่ 5.1 เป็นตัวอย่างที่สเลฟไม่มีการติดต่อสื่อสารกับมาสเตอร์จึงได้รับเพียงสถานะการเพิ่มชื่อในฐานข้อมูลของชื่อ โดยมีสถานะเป็น _STATUS_SERVICE_RECORD_ADDNAME_SUCCESS สำหรับการประกาศชื่อ Service = “Chat”, Subject = “Group1.A” (ถ้าสเลฟติดต่อกับมาสเตอร์จึงจะได้รับสถานะการเพิ่มชื่อในแผนภาพต้นไม้ของชื่ออีกด้วย)



รูปที่ 5.1 ตัวอย่างสถานะที่สเลฟได้รับจากการประกาศชื่อ

ตัวอย่าง มาสเตอร์ประกาศชื่อ โดยเรียกตัวกระทำการ createIntentionalName() ของส่วนต่อประสาน BTDeviceConnector รูปที่ 5.2 มาสเตอร์ได้รับสถานะ _STATUS_NAMETREE_ADDNAME_SUCCESS แสดงว่าชื่อ Service = “Mail”, Subject = “net.bluenet.C” ถูกเพิ่มชื่อเข้าไปในแผนภาพต้นไม้ ส่วนสถานะ

_STATUS_NAMETREE_ADDNAME_FAIL แสดงว่าไม่สามารถเพิ่มชื่อในแผนภาพต้นไม้เนื่องจากเป็นชื่อซ้ำกับที่ได้ประกาศไปแล้ว



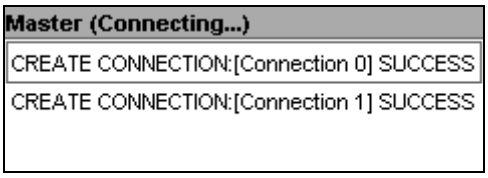
รูปที่ 5.2 ตัวอย่างสถานะที่มาสเตอร์ได้รับจากการประกาศชื่อ

<u>ทดสอบการสร้างเครือข่ายพิโคเน็ต</u>

1. วัตถุ BTDeviceConnector เรียกใช้งานตัวกระทำการ searchBTRemoteDevices() เพื่อค้นหาสเลฟ ข้อมูลของสเลฟถูกเก็บไว้ในวัตถุ BTRemoteDevice ผู้ใช้สามารถตรวจสอบชื่อแทนการติดต่อสื่อสารได้จากตัวกระทำการ getConnectionName() โดยมีค่าเป็น Connection n; n = 0, 1,... และตรวจสอบชื่อบริการและชื่ออุปกรณ์ทั้งหมดของอุปกรณ์ที่พบ ได้จากตัวกระทำการ getRemoteServiceName() และ getRemoteSubjectName() ตามลำดับ แสดงในรูปที่ 5.3

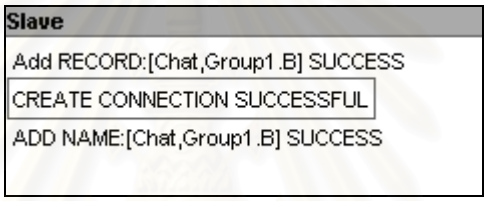
รูปที่ 5.3 ตัวอย่างการค้นหาอุปกรณ์ของมาสเตอร์

2. จากรูปที่ 5.3 ได้เลือกสร้างการติดต่อสื่อสารไปยัง Connection0 และ Connection1 โดยวัตถุ BTDeviceConnector เรียกใช้งานตัวกระทำการ createConnection() เพื่อสร้างการติดต่อไปยังอุปกรณ์สเลฟที่ต้องการ จากนั้นจะได้รับสถานะแจ้งผลการเชื่อมต่อ แสดงในรูปที่ 5.4



รูปที่ 5.4 ตัวอย่างสถานะที่มาสเตอร์ได้รับจากการสร้างการติดต่อสื่อสารไปยังสเลฟ

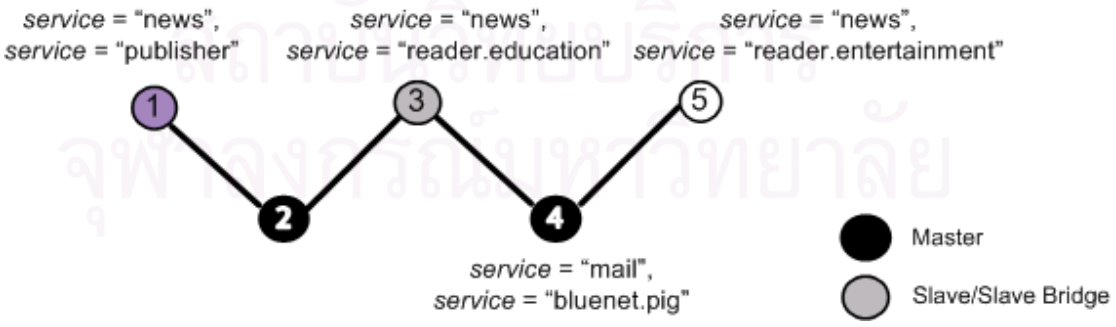
3. เมื่อสเลฟได้รับการเชื่อมต่อจากมาสเตอร์ สเลฟจะได้รับสถานะแจ้งเข้ามาเป็น _STATUS_DEVICE_ACCEPTED_CONNECTION พร้อมทั้งสถานะของชื่อที่ถูกนำไปจัดเก็บในแผนภาพต้นไม้ของมาสเตอร์ _STATUS_NAMETREE_ADDNAME_SUCCESS แสดงในรูปที่ 5.5 เป็น CREATE CONNECTION SUCCESSFUL และ ADD NAME:[Chat,Group1.B] SUCCESS ตามลำดับ



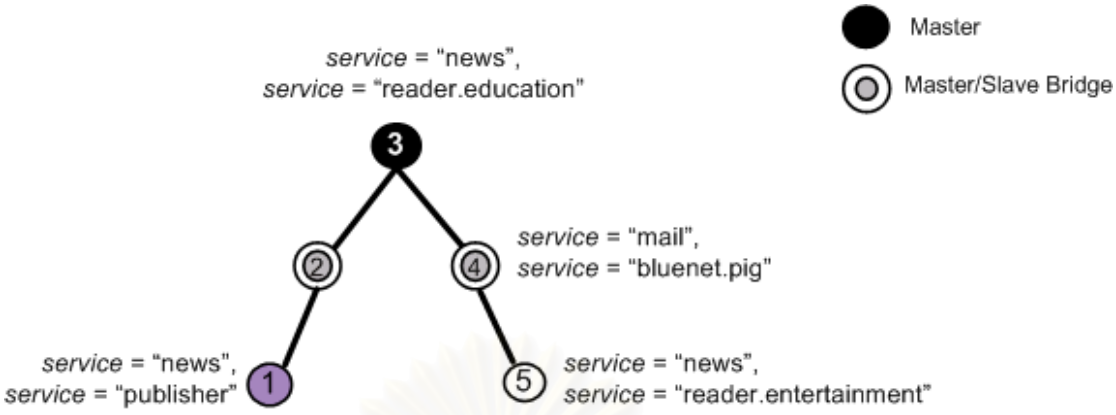
รูปที่ 5.5 ตัวอย่างสถานะที่สเลฟได้รับเมื่อได้รับการติดต่อจากมาสเตอร์

5.2.2 ทดสอบการประยุกต์ใช้กับสแคทเทอร์เน็ตรูปแบบต่าง ๆ

ทดสอบการรับ-ส่งข้อมูลระหว่างโหนดต้นทางและโหนดปลายทาง ในเครือข่ายสแคทเทอร์เน็ตแบบ Slave/Slave Mesh และ Tree Hierarchy



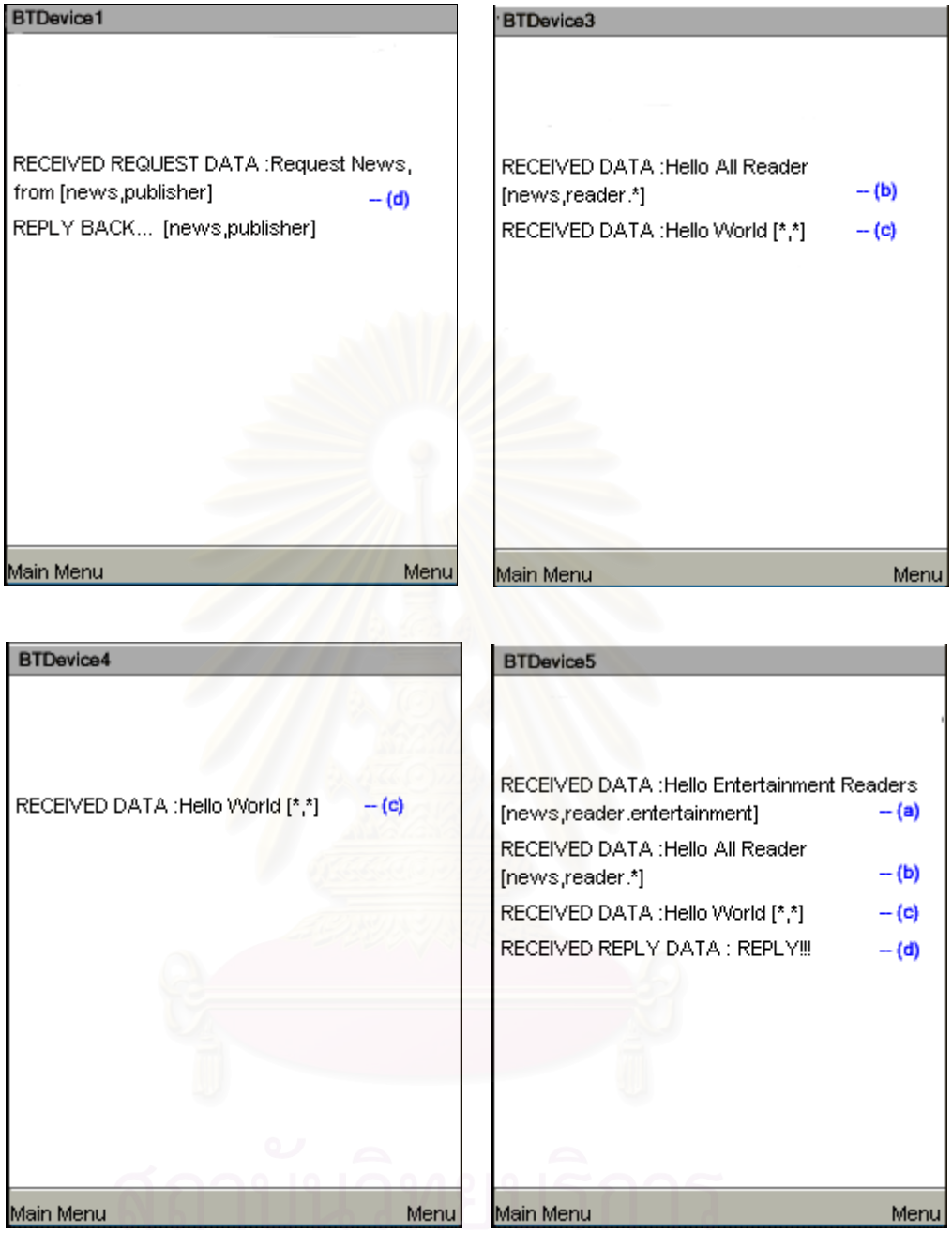
รูปที่ 5.6 ตัวอย่างอุปกรณ์ในเครือข่ายบลูทูธแบบ Slave/Slave Mesh



รูปที่ 5.7 ตัวอย่างอุปกรณ์ในเครือข่ายบลูทูธแบบ Tree Hierarchy

ตัวอย่าง การติดต่อสื่อสารในเครือข่ายสแคทเทอร์เน็ต สเลฟ/สเลฟบริดจ์ และมาสเตอร์/สเลฟบริดจ์ ตามรูปที่ 5.6 และรูปที่ 5.7 ตามลำดับ ทดสอบการรับ-ส่งข้อมูลโดยการระบุชื่อ กำหนดให้อุปกรณ์1 ทำหน้าที่เป็นโหนดต้นทางในการส่งข้อมูล แบ่งการทดสอบเป็นการส่งข้อมูลแบบยูนิคาส มัลติคาส บรอดคาส ตลอดจนการส่งข้อมูลแบบขอการตอบกลับ และการส่งข้อมูลตอบกลับ เมื่อโหนดต้นทางระบุชื่อเพื่อส่งข้อมูลไปยังปลายทาง จะเริ่มระยะเวลาในการกระจายข้อมูลอีกครั้ง ได้ผลการทดสอบดังต่อไปนี้

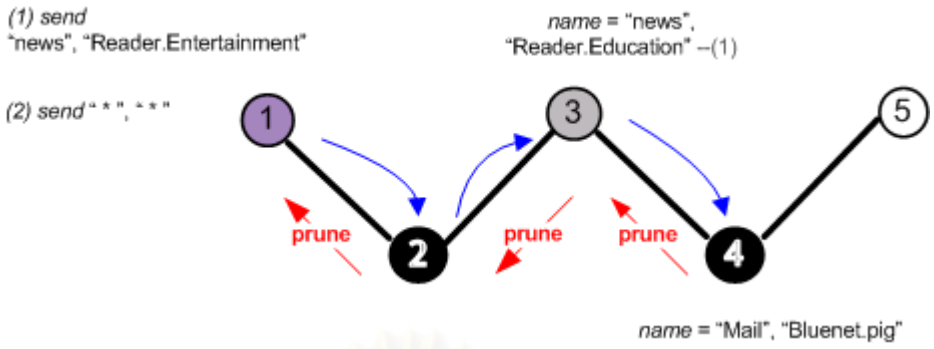
- ทดสอบการส่งข้อมูลแบบยูนิคาส
  อุปกรณ์1 ส่งข้อมูล โดยเรียกใช้ตัวกระทำการ send (“ news ” , “ reader.entertainment ”) จากนั้นอุปกรณ์5 ได้รับข้อมูลอย่างถูกต้อง (รูปที่ 5.8 a)
- ทดสอบการส่งข้อมูลแบบมัลติคาส
  อุปกรณ์1 ส่งข้อมูล โดยเรียกใช้ตัวกระทำการ send (“ news ” , “ reader.* ”) จากนั้นอุปกรณ์3 และ 5 ได้รับข้อมูลอย่างถูกต้อง (รูปที่ 5.8 b)
- ทดสอบการส่งข้อมุลแบบบรอดคาส
  อุปกรณ์1 ส่งข้อมูล โดยเรียกใช้ตัวกระทำการ send (“ * ” , “ * ”) จากนั้นอุปกรณ์3 4 และ 5 ได้รับข้อมูลอย่างถูกต้อง (รูปที่ 5.8 c)
- ทดสอบการส่งข้อมูลแบบขอการตอบกลับ และการส่งข้อมูลตอบกลับ
  ทำการทดสอบโดยให้อุปกรณ์5 ส่งข้อมูลแบบขอการตอบกลับโดยเรียกใช้ตัวกระทำการ sendRequest (“news” , “publisher”) เมื่ออุปกรณ์1 ได้รับข้อมูลจะส่งข้อมูลตอบกลับโดยใช้ตัวกระทำการ sendReply () ข้อมูลตอบกลับจากโหนดปลายทางถูกส่งไปยังโหนดต้นทางโดยใช้เส้นทางเดิม (รูปที่ 5.8 d)



รูปที่ 5.8 แสดงข้อมูลการรับข้อมูลของอุปกรณ์

5.2.3 ทดสอบการจัดเส้นทางและสภาวะที่มีการเปลี่ยนแปลง

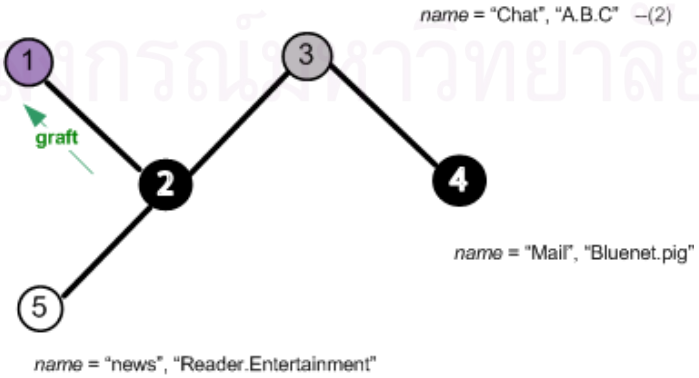
ทำการทดสอบโดยจำลองเครือข่ายสแคทเทอร์เน็ต ประกอบด้วยการทดสอบขั้นตอนวิธีการหาเส้นทาง และการจำลองสภาวะที่มีการเปลี่ยนแปลงเกิดขึ้นในเครือข่าย อันเนื่องมาจากโหนดมีการเคลื่อนที่ และการเปลี่ยนแปลงคุณลักษณะของบริการ



รูปที่ 5.9 ตัวอย่างการส่งข้อมูลและการตัดเส้นทาง

อธิบายจากรูปที่ 5.9 อุปกรณ์1 ทำหน้าที่เป็นโหนดต้นทางในการส่งข้อมูลไปยังโหนดปลายทาง กำหนดระยะเวลาในการกระจายข้อมูลอีกครั้งเท่ากับ 420 วินาที กำหนดค่าโดยตัวกระทำการ setTimeReflood() ของคลาส BTDevice เมื่อส่งข้อมูลโดยระบุชื่อปลายทาง "*", "*" (ที่เวลา 00.00.00; hh.mm.ss) อุปกรณ์3 และ 4 ได้รับข้อมูล

จากนั้นให้อุปกรณ์ 1 ส่งข้อมูลโดยระบุปลายทางเป็น "news", "Reader.Entertainment" (ที่เวลา 00.02.00) ข้อมูลจะถูกส่งแบบกระจายผ่านไปยังอุปกรณ์2 3 และ 4 เนื่องจากอุปกรณ์ที่4 เป็นมาสเตอร์เมื่อตรวจสอบแผนภาพต้นไม้ของชื่อแล้วไม่พบอุปกรณ์ใดในพิโคเน็ตที่มีชื่อสอดคล้องกับชื่อที่อุปกรณ์1 ระบุและไม่มีอุปกรณ์ถัดไปที่สามารถรับข้อมูลเพื่อส่งต่อ จึงส่งข้อความตัดเส้นทางไปยังอุปกรณ์3 เมื่ออุปกรณ์ 3 ได้รับข้อความตัดเส้นทางและพบว่าไม่มีอุปกรณ์ถัดไปที่ต้องการรับข้อมูล จึงส่งข้อความตัดเส้นทางไปยังอุปกรณ์ 2 โดยอุปกรณ์ 2 มีวิธีการตรวจสอบก่อนการส่งข่าวสารตัดเส้นทางเช่นเดียวกับอุปกรณ์4 (เนื่องจากเป็นมาสเตอร์เช่นเดียวกัน) จึงส่งข้อความตัดเส้นทางไปยังอุปกรณ์1 ซึ่งเป็นโหนดต้นทางในการส่งข้อมูล ในการส่งครั้งถัดไปสามารถแจ้งให้อุปกรณ์ 1 ทราบได้ว่ามีอุปกรณ์ใดในเครือข่ายที่สามารถรับข้อมูลได้หรือไม่ (สถานะ _STATUS_RECEIVER_EXIST หรือ _STATUS_RECEIVER_NOT_EXIST) เมื่อไม่มีอุปกรณ์ใดในเครือข่ายที่ต้องการรับข้อมูลก็จะไม่ส่งข้อมูลออกไปในเครือข่าย



รูปที่ 5.10 ตัวอย่างเปลี่ยนแปลงในเครือข่ายและการต่อเส้นทาง

ต่อมาที่เวลา 00.03.00 อุปกรณ์5 ได้มีการเคลื่อนที่เข้ามาในพิโคเน็ตที่มีอุปกรณ์2 เป็นมาสเตอร์ ตามรูปที่ 5.10 และได้สร้างการติดต่อสื่อสารกับอุปกรณ์2 ที่เวลา 00.04.00 (ดังนั้นในระหว่างเวลา 00.03.00 ถึง 00.04.00 อุปกรณ์5 จะไม่สามารถรับข้อมูลได้) ต่อจากนั้นอุปกรณ์5 ได้ประกาศชื่อเป็น “news”, “Reader.Entertainment” โดยชื่อจะถูกส่งไปยังอุปกรณ์2 เพื่อจัดเก็บในแผนภาพต้นไม้ เมื่อชื่อถูกเพิ่มเข้าในแผนภาพต้นไม้ของมาสเตอร์ (อุปกรณ์2) และชื่อที่ตรวจพบในตารางเส้นทางมีการระบุปลายทางสอดคล้องกับชื่อใหม่ที่เพิ่มเข้าไปในแผนภาพต้นไม้ อุปกรณ์2 จึงส่งข่าวสารต่อเส้นทางไปยังอุปกรณ์1

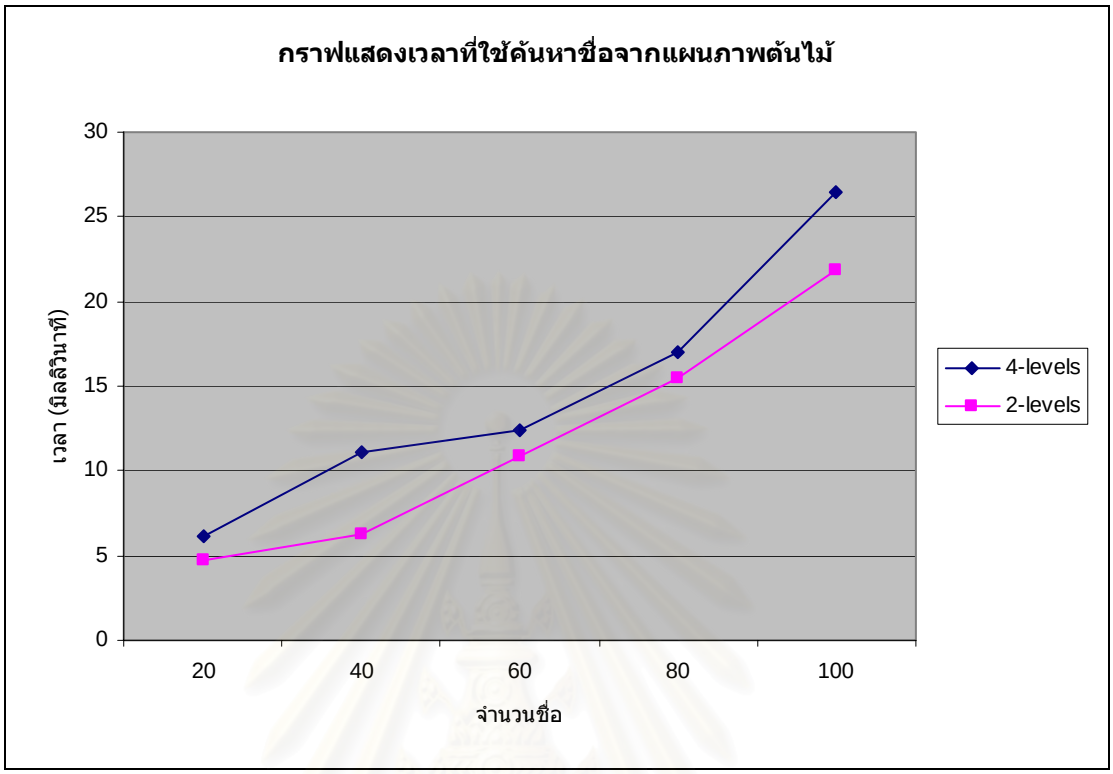
จากรูปที่ 5.10 อุปกรณ์3 ยกเลิกการใช้ชื่อ “news”, “Reader.Education” แล้วทำการประกาศชื่อใหม่เป็น “Chat”, “A.B.C” เมื่ออุปกรณ์1 ระบุชื่อเพื่อส่งไปยังปลายทางเป็น “*”, “*” ทั้งอุปกรณ์ 3 4 และ 5 ได้รับข้อมูลได้อย่างถูกต้อง

เมื่อเวลาผ่านไป 420 วินาทีเป็นเวลาของการเริ่มส่งแบบกระจายข้อมูลอีกครั้ง ของชื่อที่ระบุปลายทางด้วย “*”, “*” นั่นคือที่เวลา 00.07.00 อุปกรณ์1 ทำการส่งข่าวสารเพื่อกระจายข้อมูลอีกครั้งไปยังอุปกรณ์จัดเส้นทางบลูทูธทุกตัวในเครือข่าย หลังจากนั้นข้อมูลที่ส่งโดยระบุปลายทางด้วยชื่อดังกล่าวจะถูกส่งแบบกระจายไปยังอุปกรณ์จัดเส้นทางทุกตัว (อุปกรณ์2 3 และ 4) เพื่อรองรับการเปลี่ยนแปลงที่เกิดขึ้นระหว่างการติดต่อสื่อสาร เช่นเดียวกับชื่อที่ระบุปลายทางเป็น “news, Reader.Entertainment” กระจายข้อมูลอีกครั้งที่เวลา 00.09.00 หลังจากนั้นจะเริ่มระยะเวลากระจายข้อมูลอีกครั้งก็ต่อเมื่อเมื่อผู้ใช้ระบุชื่อปลายทางในการส่งข้อมูลใหม่

5.3 ตัวอย่างการทดสอบประสิทธิภาพ

ในส่วนนี้เป็นตัวอย่างการทดสอบประสิทธิภาพในการค้นหาของแผนภาพต้นไม้ สำหรับการค้นหาเลขที่อยู่บลูทูธจากชื่อที่กำหนด ทำการทดสอบดังต่อไปนี้

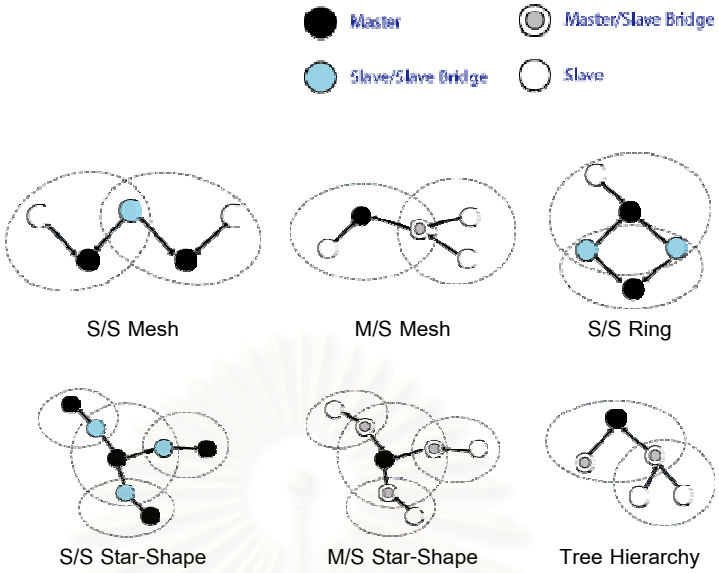
การทดสอบกระทำโดย จำลองจำนวนชื่อที่จัดเก็บไว้ในแผนภาพต้นไม้ ในจำนวนที่แตกต่างกัน คือ จำนวน 20, 40, 60, 80 และ 100 ชื่อ แบ่งออกเป็น 2 ชุดทดสอบ โดยชุดทดสอบแรกกำหนดให้ชื่อมี 2 ลำดับชั้นของชื่อ (เช่น “level1.level2”) และอีกชุดการทดสอบกำหนดให้ชื่อมี 4 ลำดับชั้นของชื่อ (เช่น “level1.level2.level3.level4”) แล้วทำการระบุชื่อเพื่อใช้ค้นหาเลขที่อยู่บลูทูธจากแผนภาพต้นไม้ในรูปแบบต่างๆ กันทั้งหมด 10 ตัวอย่างการทดสอบ เพื่อหาเวลาที่ใช้ค้นหาจากแผนภาพต้นไม้ โดยนำเวลาที่สิ้นสุดการค้นหาลบด้วยเวลาเริ่มต้นค้นหา จากนั้นคำนวณเวลาที่ใช้ในการค้นหาโดยเฉลี่ย (= ผลรวมของเวลาที่ใช้ค้นหาทั้งหมด หารด้วย10) และหาความสัมพันธ์ระหว่างเวลาที่ใช้ในการค้นหา (มีหน่วยเป็น มิลลิวินาที) กับจำนวนชื่อทั้งหมดที่เก็บในแผนภาพต้นไม้ ผลการทดสอบแสดงได้ดังกราฟในรูปที่ 5.11



รูปที่ 5.11 เวลาที่ใช้ค้นหาชื่อจากแผนภาพต้นไม้

จากกราฟในรูปที่ 5.11 พบว่า เมื่อจำนวนชื่อที่เก็บในแผนภาพต้นไม้เพิ่มขึ้น เวลาที่ใช้ในการค้นหาเลขที่อยู่บลูทูธที่สอดคล้องกับชื่อจะมีค่าเพิ่มมากขึ้นด้วย ซึ่งการเพิ่มขึ้นนั้นมีลักษณะเพิ่มขึ้นเป็นแบบลำดับชั้น สำหรับชื่อที่มีอยู่ในแผนภาพต้นไม้ทั้งหมด 100 ชื่อนั้น เวลาที่แผนภาพต้นไม้ใช้ในการค้นหาประมาณ 21.8 มิลลิวินาทีสำหรับชื่อที่มี 2 ลำดับชั้นของชื่อ ส่วนชื่อที่มี 4 ลำดับชั้น ใช้เวลาค้นหาประมาณ 26.5 มิลลิวินาที ซึ่งเป็นผลที่ยอมรับได้ และจากการทดสอบเมื่อชื่อมีลำดับชั้นเพิ่มมากขึ้นเช่นจากตัวอย่างทดสอบเพิ่มลำดับชั้นขึ้นจาก 2 เป็น 4 ลำดับชั้น พบได้ว่าเวลาที่ใช้ในการค้นหาเพิ่มมากขึ้นด้วย ดังนั้น ผู้ใช้ควรตั้งชื่อที่มีความกะทัดรัดและสื่อความหมายชัดเจน และตั้งชื่อเฉพาะชื่อที่ต้องการใช้จริงๆ เท่านั้น จะช่วยลดเวลาที่ใช้ในการค้นหาได้มากขึ้น

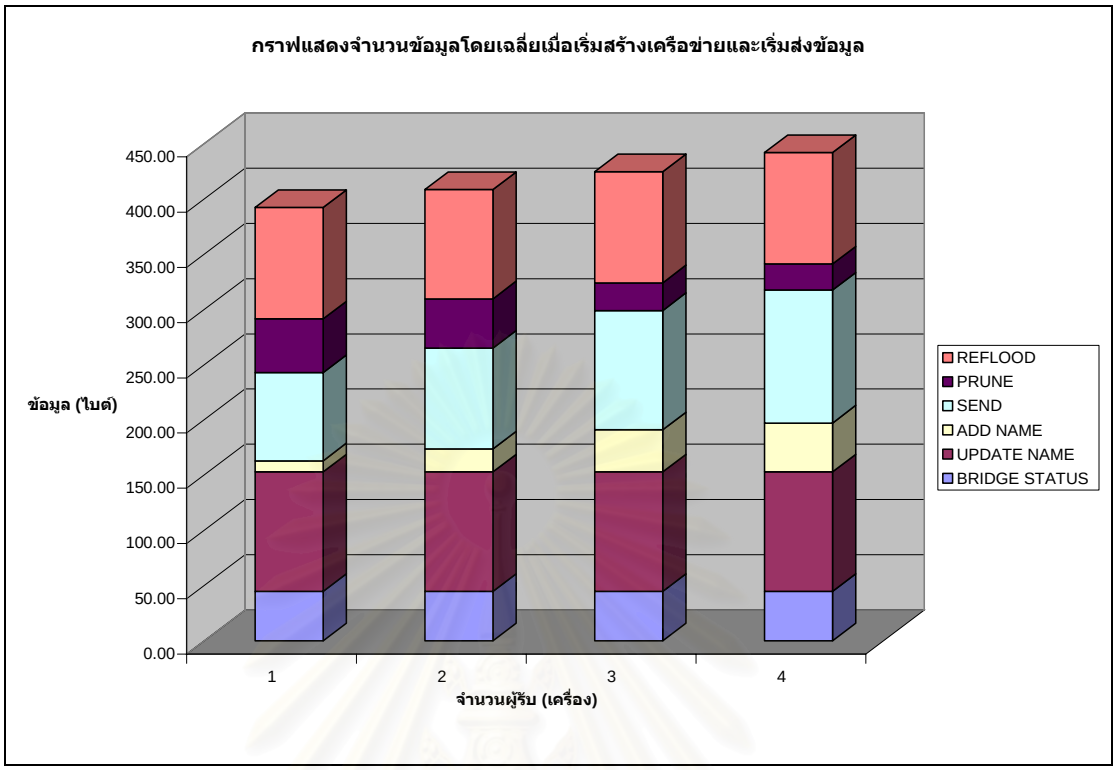
ส่วนต่อมาทำการทดสอบเพื่อหาจำนวนข้อมูลที่เฟรมเวิร์คทำการส่งในเครือข่าย ผลที่ได้จากการทดสอบเกิดจากรูปแบบในการติดต่อสื่อสารที่เฟรมเวิร์คสร้างขึ้นเพื่อให้ข้อมูลสามารถไปถึงอุปกรณ์ปลายทางที่ต้องการได้ จึงมีข้อมูลส่วนที่เพิ่มขึ้นจากข้อมูลจริงของผู้ใช้ ทำให้แบนด์วิดท์ที่ใช้ในการส่งข้อมูลผ่านเครือข่ายที่เพิ่มขึ้น (Overhead) ในส่วนต่อไปนี้จะแสดงจำนวนข้อมูลเฉลี่ยในเครือข่ายรวมทั้งได้จำแนกออกตามชนิดของข้อมูล



รูปที่ 5.12 ทดสอบในเครือข่ายบลูทูธรูปแบบต่างๆ

ทำการทดสอบในเครือข่ายบลูทูธสแคทเทอร์เน็ตทั้งหมด 6 รูปแบบตามรูปที่ 5.12 โดยหาจำนวนข้อมูลเฉลี่ยของเครือข่ายแต่ละแบบเมื่อมีอุปกรณ์บลูทูธทั้งหมด 5 ตัว กำหนดช่วงระยะเวลาการทดสอบเป็น 120 วินาที ระยะเวลาในการกระจายข้อมูลอีกครั้ง (reflood) ที่ 60 วินาที และกำหนดให้สเลฟส่งการปรับปรุงชื่อไปยังมาสเตอร์ทุกๆ 100 วินาที และอุปกรณ์แต่ละตัวอยู่ที่ตำแหน่งเดิมตลอดการทดสอบ จากนั้นแบ่งชุดทดสอบโดยให้ภายในเครือข่ายมีจำนวนผู้รับ 1 2 3 และ 4 อุปกรณ์ เพื่อหาค่าเฉลี่ยของจำนวนข้อมูลในเครือข่าย (= ผลรวมของข้อมูลทั้งหมดที่พบในเครือข่ายลบด้วยส่วนของข้อมูลจริงของผู้ใช้หารด้วย 5 (เนื่องจากแต่ละชุดทดสอบจะกำหนดอุปกรณ์ส่งและรับแตกต่างกันไปทั้งหมด 5 แบบ) ค่าที่เฉลี่ยที่ได้มีหน่วยเป็นไบต์) รายละเอียดผลการทดสอบหาจำนวนข้อมูลเฉลี่ยในเครือข่ายรูปแบบต่างๆ แสดงไว้ที่ภาคผนวก ก

จำนวนข้อมูลเฉลี่ยในเครือข่ายบลูทูธที่เกิดจากการใช้งานเฟรมเวิร์ค ดังที่แสดงไว้ในกราฟรูปที่ 5.13 พบว่าจำนวนข้อมูลโดยเฉลี่ยทั้งหมดที่ส่งผ่านในเครือข่ายเมื่อเริ่มต้นสร้างเครือข่ายและทำการส่งข้อมูลครั้งแรกมีค่าเป็นเป็น 392.29, 408.56, 424.78 และ 441.99 ไบต์ ตามลำดับ และจากกราฟในรูปที่ 5.13 แบ่งตามจำนวนข้อมูลแต่ละประเภทได้ดังตารางที่ 5.1 ประกอบด้วย SEND (การส่งข้อมูล), REFLOOD (การกระจายข้อมูล), UPDATE NAME (การปรับปรุงชื่อ) รวมทั้งข้อมูลที่เกี่ยวข้องกับการสร้างเครือข่าย ได้แก่ BRIDGE STATUS (การแจ้งสถานะของบริดจ์), ADD NAME (การเพิ่มชื่อในแผนภาพต้นไม้) เพิ่มขึ้นมาด้วย และข้อมูลประเภท PRUNE (การตัดเส้นทาง) เนื่องจากการส่งข้อมูลครั้งแรกจะมีการตัดเส้นทางที่ไม่ต้องการข้อมูลออกไป



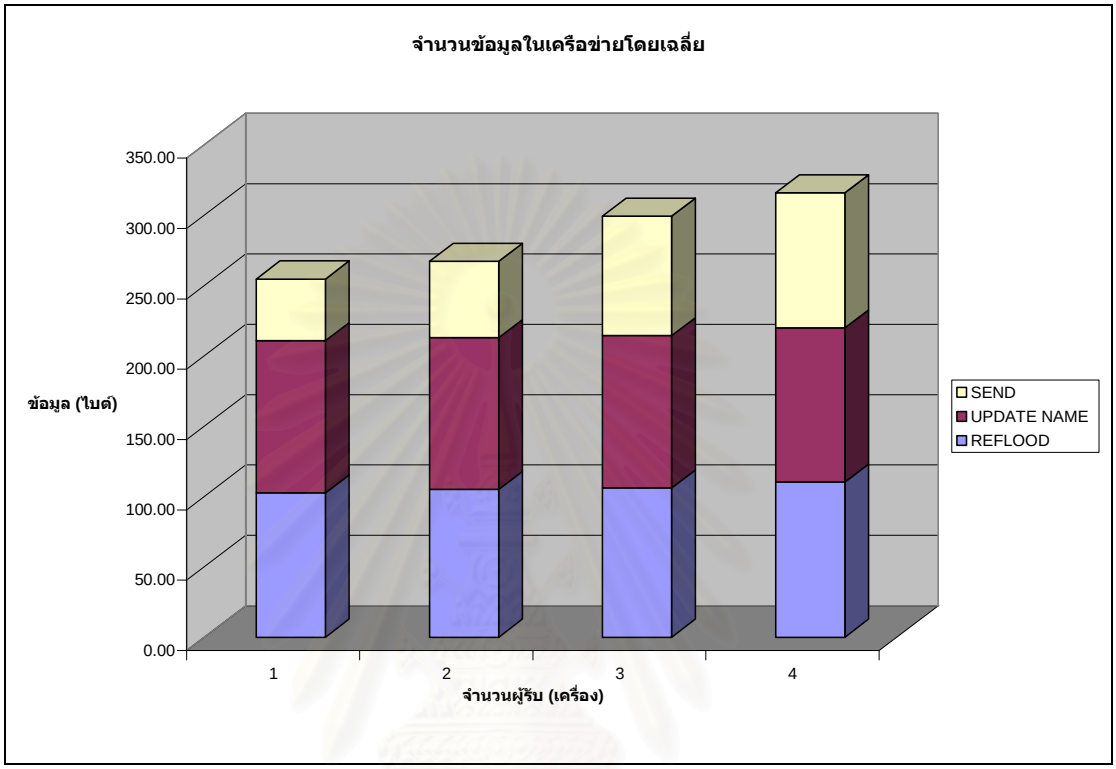
รูปที่ 5.13 กราฟแสดงจำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูล

ตารางที่ 5.1 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลจำแนกตามประเภทข้อมูล

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
REFLOOD	100.72	98.91	100.72	100.72
PRUNE	48.75	44.50	25.00	23.75
SEND	79.85	91.33	108.02	120.34
ADD NAME	10.14	21.00	38.22	44.36
UPDATE NAME	108.04	108.04	108.04	108.04
BRIDGE STATUS	44.79	44.79	44.79	44.79
Total	392.29	408.56	424.78	441.99

ต่อมาทำการหาจำนวนข้อมูลเฉลี่ยในเครือข่ายหลังจากช่วงการทดสอบแรกต่อไปอีก 120 วินาที ซึ่งช่วงเวลานี้เป็นการส่งข้อมูลที่รู้เส้นทางสำหรับส่งข้อมูลแล้ว พบว่าจำนวนข้อมูลที่ส่งผ่านในเครือข่ายลดลง แสดงกราฟดังรูปที่ 5.14 ซึ่งจำนวนข้อมูลโดยเฉลี่ยในเครือข่ายมีค่าเป็น 254.55, 267.13, 299.18 และ 315.82 ไบต์ เมื่อมีผู้รับ 1, 2, 3 และ 4 อุปกรณ์ตามลำดับ และพบเฉพาะ

ข้อมูลประเภท SEND, UPDATE NAME และ REFLOOD โดยได้แสดงขนาดของข้อมูลแต่ละประเภทไว้ในตารางที่ 5.2



รูปที่ 5.14 กราฟแสดงจำนวนข้อมูลในเครือข่ายหลังจากรู้เส้นทางส่งข้อมูล

ตารางที่ 5.2 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปจำแนกตามชนิดข้อมูล

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
SEND	43.97	54.18	84.98	96.06
UPDATE NAME	108.04	108.04	108.04	109.60
REFLOOD	102.54	104.91	106.16	110.16
Total	254.55	267.13	299.18	315.82

สรุปได้ว่าในการส่งข้อมูลครั้งแรกจะแบนด์วิดท์โดยรวมทั้งเครือข่ายมากที่สุดเนื่องจากต้องกระจายข้อมูลไปยังอุปกรณ์จัดเส้นทางบลูทูธทุกตัว หลังจากนั้นในการส่งข้อมูลครั้งถัดไปการใช้แบนด์วิดท์โดยรวมลดลง เนื่องจากเฟรมเวิร์คได้มีการเรียนรู้เส้นทางสำหรับส่งข้อมูลแล้วว่าจะต้องใช้เส้นทางใดเพื่อส่งข้อมูลไปยังผู้รับ โดยที่จำนวนแบนด์วิดท์ที่ใช้สำหรับการส่งข้อมูลในเครือข่ายโดยเฉลี่ยลดลง คิดเป็น 31.96 เปอร์เซ็นต์ (แบนด์วิดท์ที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทาง

(SEND) จากตารางที่ 5.1 มีค่าเป็น 79.85, 91.33, 108.02 และ 120.34 ไบต์ และจากตารางที่ 5.2 ใช้แบนด์วิดท์ของเครือข่ายในการส่งข้อมูล 43.97, 54.18, 84.98 และ 96.06 ไบต์ พบว่าจำนวนข้อมูลในเครือข่ายลดลง 35.11%, 34.62%, 29.57% และ 28.55% เมื่อมีผู้รับ 1, 2, 3 และ 4 อุปกรณ์ตามลำดับ คิดเป็นเปอร์เซ็นต์การใช้แบนด์วิดท์ของเครือข่ายโดยรวมลดลง 31.96%)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลสรุปของงานวิจัย ปัญหาและข้อจำกัดต่างๆ ของระบบ รวมทั้งข้อเสนอแนะที่สามารถนำไปปรับปรุงและพัฒนาได้ต่อไปในอนาคต โดยมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอการติดต่อสื่อสารโดยการระบุชื่อสำหรับเครือข่ายบลูทูธ ผู้วิจัยได้พัฒนาได้พัฒนาเฟรมเวิร์คขึ้นภายใต้สภาวะแวดล้อมของเจเอสอาร์-82 (Java Specification Request 82 – JSR-82) เพื่อรองรับโปรแกรมประยุกต์เจทูเอ็มอี และได้ทดสอบการทำงานของเฟรมเวิร์คผ่านตัวจำลองการทำงาน เจทูเอ็มอี ไวร์เลส ทูลคิท รุ่นที่ 2.2 โดยผลที่ได้รับคือทำให้การติดต่อสื่อสารในเครือข่ายบลูทูธทำได้ง่ายขึ้นและมีประสิทธิภาพมากขึ้น สามารถส่งข้อมูลแบบยูนิคาส มัลติคาสและบรอดคาสได้โดยไม่เปลี่ยนแปลงโมเดลพื้นฐานบลูทูธ สามารถนำไปใช้ได้กับสแคทเทอร์เน็ตได้ทุกรูปแบบ ทั้งที่เป็นสเลฟ/สเลฟ และมาสเตอร์/สเลฟบริดจ์ มีการจัดเส้นทางเพื่อส่งข้อมูลไปยังปลายทางโดยการส่งข้อมูลไปยังผู้รับหลายคนได้ด้วยการส่งเพียงครั้งเดียวเป็นการเพิ่มประสิทธิภาพการติดต่อสื่อสารในเครือข่ายบลูทูธ เฟรมเวิร์คที่พัฒนาขึ้นเหมาะสำหรับการติดต่อสื่อสารส่งผ่านข้อมูลแบบต่อเนื่อง เนื่องจากการส่งแต่ละครั้งอุปกรณ์จัดเส้นทางจะมีการเรียนรู้เส้นทางที่เหมาะสมที่สุดสำหรับการส่งผ่านข้อมูลในครั้งต่อไป

จากการทดสอบการทำงานของเฟรมเวิร์คในส่วนต่างๆ ที่ประกอบด้วย การทดสอบการจัดการในพิโคเน็ต อันประกอบด้วย การทดสอบสร้างสเลฟและ/หรือมาสเตอร์ การทดสอบเพิ่ม-ลบชื่อ การทดสอบสร้างเครือข่ายพิโคเน็ต และการทดสอบการประยุกต์ใช้กับเครือข่ายสแคทเทอร์เน็ตแบบ Slave/Slave Mesh และ Tree Hierarchy เพื่อทดสอบการส่งข้อมูลในรูปแบบต่างๆ ประกอบด้วยการส่งแบบยูนิคาส มัลติคาส บรอดคาส รวมถึงการทดสอบการส่งข้อมูลแบบขอการตอบกลับและการส่งข้อมูลตอบกลับ การทดสอบการทำงานในกระบวนการจัดเส้นทางสำหรับส่งข้อมูล รวมทั้งตรวจสอบสถานะต่างๆ ที่ได้รับจากเฟรมเวิร์ค พบได้ว่าเฟรมเวิร์คสามารถทำงานได้อย่างถูกต้องตามที่กำหนด สำหรับการทดสอบประสิทธิภาพโดยทำการวัดเวลาที่ใช้ในการค้นหาเลขที่อยู่บลูทูธจากแผนภาพต้นไม้โดยการระบุชื่อ พบว่าจำนวนชื่อที่จัดเก็บไว้ทั้งหมด 100 ชื่อใช้เวลาในการค้นหาประมาณ 21.8 มิลลิวินาที สำหรับชื่อที่มี 2 ลำดับชั้น และ 26.5 มิลลิวินาที สำหรับชื่อที่มี 4 ลำดับชั้น ซึ่งเป็นค่าที่สามารถยอมรับได้ กล่าวได้ว่าประสิทธิภาพในการค้นหาของแผนภาพต้นไม้ขึ้นอยู่กับจำนวนและขนาดของชื่อที่เก็บไว้ในแผนภาพต้นไม้ ดังนั้นชื่อที่ไม่ต้องการ

ใช้แล้วควรทำการลบออกจากแผนภาพต้นไม้ทันที เพื่อเพิ่มประสิทธิภาพการในค้นหาชื่อ ทั้งยังช่วยลดระยะเวลาการส่งข้อมูลไปยังปลายทางได้

6.2 ปัญหาและข้อจำกัดของงานวิจัย

1. ข้อจำกัดการใช้สัญลักษณ์พิเศษในการตั้งชื่อ อาทิเช่น { ~ ` * _ | \ / [ ] : ; " ‘ , }
2. ข้อจำกัดในการทดสอบการใช้งานเฟรมเวิร์คกับเครือข่ายบลูทูธขนาดใหญ่
3. ปัญหาด้านความปลอดภัย เนื่องจากรูปแบบการส่งข้อมูลโดยการระบุชื่อ โดยที่ผู้รับสามารถตั้งชื่อใดก็ได้

6.3 ข้อเสนอแนะ

1. เพิ่มตัวกระทำการสำหรับสร้างเครือข่ายการติดต่อสื่อสารแบบอัตโนมัติ ที่เป็นบริดจ์ชนิดมาสเตอร์/สเลฟ
2. ให้ตำแหน่งของเครื่องหมาย * ที่ใช้ในการระบุชื่อเพื่อการติดต่อไปยังปลายทาง สามารถอยู่ที่ตำแหน่งของการระบุชื่อได้ เช่น “bluetooth.*.device”
3. สร้างสารบบของชื่อ (Name Directory) เพื่อให้สามารถตรวจสอบได้ว่าอุปกรณ์ในเครือข่ายมีการประกาศชื่อใดบ้าง
4. กำหนดการหมดเวลารอ (Timeout) ในตัวกระทำการ sendRequest() แบบทีซีพี (TCP) โดยคำนวณเป็นช่วงเวลาระหว่าง 1 Round Trip Time ซึ่งเป็นระยะเวลาที่ใช้ในการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง และจากโหนดปลายทางกลับมายังโหนดต้นทาง
5. ในด้านความปลอดภัย (Security) เนื่องจากเฟรมเวิร์คสามารถทำการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางครั้งละหลายตัวได้ซึ่งเป็นลักษณะการส่งแบบมัลติคาส สำหรับเครือข่ายที่พิจารณาด้านความปลอดภัยควรมีการตรวจสอบผู้ใช้เพิ่มเติม เช่น การพิสูจน์ตัวจริง (authenticate) ของอุปกรณ์ที่ทำการตั้งชื่อ เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Bluetooth Specification 1.2. Bluetooth Special Interest Group (SIG). Available form: http://www.Bluetooth.org.
2. Hopkins, B., and Antony, R. Bluetooth for Java. 2003.
3. Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., and Ganesan, D. Building efficient wireless sensor networks with low-level naming. In Proceedings of the ACM Symposium on Operating Systems Principles (October 2001): 146–159.
4. Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., and Lilley, J. The design and implementation of an intentional naming system. In Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP '99) 34 (December 1999): 186–201.
5. Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'2000) (August 2000): 56–67.
6. Bowman, M., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. ACM Transactions on Programming Languages and Systems 15(1993): 795–825.
7. Kurose, J. F. and Ross, K. W. Computer Networking A Top-Down Approach Featuring the Internet. 2nd ed. Addison Wesley Longman, 1999.
8. Comer, D. E. Internetworking with TCP/IP Vol I principles, protocols,and architectures. 4th ed. Prentice Hall, 1996.
9. Deering, S. Multicast routing in internetworks and extended lans. In Proceedings of the ACM SIGCOMM (August 1988): 55–64.
10. Deering, S. E., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., and Wei, L. The PIM architecture for wide-area multicast routing. IEEE Transactions on Networking 4 (April 1996): 153–162.
11. Sun Microsystems. J2ME Wireless Toolkit 2.2. 2004.

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

จำนวนข้อมูลเฉลี่ยจำแนกตามเครือข่ายบลูทูธสแคทเทอร์เน็ตรูปแบบต่างๆ

- **จำนวนข้อมูลเฉลี่ยในเครือข่ายบลูทูธแบบ Mesh**

ตารางที่ 1 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลในเครือข่ายบลูทูธแบบ Mesh

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
REFLOOD	58	50.75	58	58
PRUNE	14	21	0	0
SEND	54	76	84.5	104.25
ADD NAME	16	24	34	42
UPDATE NAME	100	100	100	100
BRIDGE STATUS	33	33	33	33
Total	275	304.75	309.5	337.25

ตารางที่ 2 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปในเครือข่ายบลูทูธแบบ Mesh

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
SEND	40	46.75	60.75	65
UPDATE NAME	100	100	100	106.25
REFLOOD	65.25	74.75	79.75	95.75
Total	205.25	221.5	240.5	267

- **จำนวนข้อมูลเฉลี่ยในเครือข่ายบลูทูธแบบ Ring**

ตารางที่ 3 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลในเครือข่ายบลูทูธแบบ Ring

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
REFLOOD	188.5	188.5	188.5	188.5
PRUNE	112	98	70	70
SEND	129	145.5	196.5	199.5
ADD NAME	8	24	72	72
UPDATE NAME	125	125	125	125
BRIDGE STATUS	77	77	77	77

Total	639.5	658	729	732

ตารางที่ 4 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปในเครือข่ายบลูทูธแบบ Ring

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
SEND	45	72	136.5	147
UPDATE NAME	125	125	125	125
REFLOOD	188.5	188.5	188.5	188.5
Total	358.5	385.5	450	460.5

- **จำนวนข้อมูลเฉลี่ยในเครือข่ายบลูทูธแบบ Star**

ตารางที่ 5 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลในเครือข่ายบลูทูธแบบ Star

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
REFLOOD	98.40	98.40	98.40	98.40
PRUNE	55.00	45.00	30.00	25.00
SEND	80.89	92.32	93.57	104.11
ADD NAME	8.57	20.00	22.86	31.43
UPDATE NAME	107.14	107.14	107.14	107.14
BRIDGE STATUS	47.14	47.14	47.14	47.14
Total	397.14	410.00	399.11	413.21

ตารางที่ 6 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปในเครือข่ายบลูทูธแบบ Star

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
SEND	35.89	36.97	70.18	86.25
UPDATE NAME	107.14	107.14	107.14	107.14
REFLOOD	98.40	98.40	98.40	98.40
Total	241.43	242.50	275.72	291.79

- **จำนวนข้อมูลเฉลี่ยในเครือข่ายบลูทูธแบบ Tree**

ตารางที่ 7 จำนวนข้อมูลเมื่อเริ่มสร้างเครือข่ายและเริ่มส่งข้อมูลในเครือข่ายบลูทูธแบบ Tree

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
REFLOOD	58	58	58	58

PRUNE	14	14	0	0
SEND	55.5	51.5	57.5	73.5
ADD NAME	8	16	24	32
UPDATE NAME	100	100	100	100
BRIDGE STATUS	22	22	22	22
Total	257.5	261.5	261.5	285.5

ตารางที่ 8 ข้อมูลที่ถูกส่งในเครือข่ายโดยทั่วไปในเครือข่ายบลูทูธแบบ Tree

ข้อมูล(ไบต์)\จำนวนผู้รับ	1	2	3	4
SEND	55	61	72.5	86
UPDATE NAME	100	100	100	100
REFLOOD	58	58	58	58
Total	213	219	230.5	244

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวอมรรัตน์ จิระพรกุล เกิดเมื่อวันที่ 16 พฤษภาคม 2525 ที่จังหวัดอุดรธานี สำเร็จการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) เกียรตินิยมอันดับหนึ่ง สาขาวิชาวิทยาการคอมพิวเตอร์ จากคณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปีการศึกษา 2546 และเข้าศึกษาต่อหลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย