

โปรแกรมจำลองเหตุการณ์เพื่อศึกษาโปรโตคอล
การพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอค
SIMULATION PROGRAM FOR ROUTING PROTOCOL
IN AD HOC WIRELESS NETWORKS



นายสรารุท ทองสันต์
นายสันติสุข สุตสว่าง

เลขหมู่.....
เลขทะเบียน..... 61467
วัน,เดือน,ปี..... 18 ก.ค. 2549

b.....
.....

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจำลองเหตุการณ์เพื่อศึกษาโพรโตคอล
การพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอค
SIMULATION PROGRAM FOR ROUTING PROTOCOL
IN AD HOC WIRELESS NETWORKS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมจำลองเหตุการณ์เพื่อศึกษาโพรโทคอลการพิจารณาเส้นทางในเครือข่ายไร้สายแบบ
แอดฮอค

SIMULATION PROGRAM FOR ROUTING PROTOCOL IN AD HOC WIRELESS
NETWORKS

คณะผู้จัดทำ นายสรวิศ ทองสันต์ รหัส 44010514

นายสันติสุข สุดสวาท รหัส 44010519



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SIMULATION PROGRAM FOR ROUTING PROTOCOL IN AD HOC WIRELESS
NETWORKS**

Mr. Sarawuth Tongsant

Mr. Santisuk Sudsawat

Dr. Sakchai Thipchaksurat Advisor

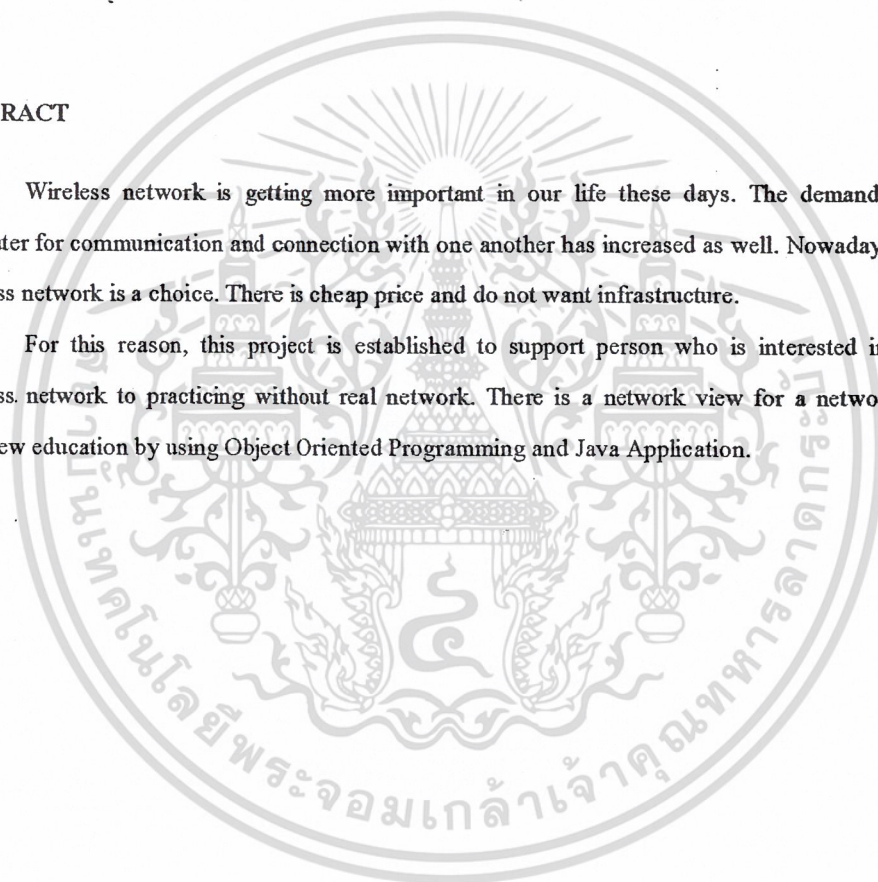
Mr. Thana Hongsuwan Advisor

Academic Year 2004

ABSTRACT

Wireless network is getting more important in our life these days. The demand of using computer for communication and connection with one another has increased as well. Nowadays, Ad hoc wireless network is a choice. There is cheap price and do not want infrastructure.

For this reason, this project is established to support person who is interested in Ad hoc wireless network to practicing without real network. There is a network view for a network system overview education by using Object Oriented Programming and Java Application.



กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษาและคอยดูแลจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาทั้งสองท่านที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญาานิพนธ์ฉบับนี้ คอยให้ความเอาใจใส่ แนะนำและความช่วยเหลือเสมอมา คือ ดร.ศักดิ์ชัย ทิพย์จักรนุรัตน์และอาจารย์ธนา หงษ์สุวรรณซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและพัฒนา โปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็ว รวมทั้งยังมีอินเทอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าหาความรู้ต่างๆ ซึ่งท้ายที่สุดแล้วก็ประกอบกันเป็นส่วนหนึ่งของโครงการนี้ ขอขอบคุณเพื่อนๆในห้องเน็คเวิร์กที่คอยสร้างความคิดครุ่นยามอยู่ในห้อง เป็นกำลังใจเสมอมา และที่ขาดไม่ได้ต้องขบขบคุณห้องเน็คเวิร์กที่ให้พี่เสี่ย ทักสอน มีแรงยก เข็มในกรรพิง เบนและสุดท้ายต้องขบขบคุณ

บุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ นั่นคือ บิดา มารดาและบุคคลในครอบครัว อันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดู คอยสั่งสอนข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบพระคุณมา ณ ที่นี้ด้วย

นายสรวิช ทองสันต์
นายสันติสุข สุตสวาท

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพประกอบ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 หลักการและเหตุผล	1
1.3 วัตถุประสงค์ของปริญญาโท	1
1.4 ขอบเขตของปริญญาโท	2
1.5 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ความรู้พื้นฐาน	3
2.1 เครือข่ายคอมพิวเตอร์	3
2.2 โปรแกรมสำหรับเครือข่าย	3
2.2.1 ลำดับชั้นของโพรโตคอล	3
2.2.2 ระบบมาตรฐาน OSI	3
บทที่ 3 เครือข่ายไร้สายแบบแอดฮอค	9
3.1 ทำไมจึงต้องเป็นเครือข่ายแบบแอดฮอค	9
3.2 ลักษณะความแตกต่างกันของอุปกรณ์โมบาย	10
3.3 ชนิดของการสื่อสารบนแอดฮอค โมบาย	11
3.4 รูปแบบของการเคลื่อนไหวของตัวโมบายโฮสต์	11
3.5 ประเด็นที่ควรให้ความสำคัญในเครือข่ายไร้สายแบบแอดฮอค	12
3.6 โพรโตคอลการเข้าใช้งานเครือข่ายของเครือข่ายไร้สายแบบแอดฮอค	15
3.7 ปัญหาในการเข้าใช้งานช่องสัญญาณ	15
3.8 การเริ่มต้นแมกโพรโตคอลของผู้รับ	21
3.9 การเริ่มต้นแมกโพรโตคอลของผู้ส่ง	21
บทที่ 4 มาตรฐาน IEEE 802.11	23
4.1 กล่าวนำ	23
4.2 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน IEEE 802.11	23
4.3 วิวัฒนาการของมาตรฐาน IEEE 802.11	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
4.3.1 IEEE 802.11b	24
4.3.2 IEEE 802.11a	25
4.3.3 IEEE 802.11g	25
4.3.4 IEEE 802.11e	25
4.3.5 IEEE 802.11i	26
4.4 ลักษณะการเชื่อมต่อของอุปกรณ์ IEEE 802.11 WLAN	26
4.4.1 โหมด Infrastructure	26
4.4.2 โหมด Ad-Hoc หรือ Peer-to-Peer	27
4.5 การเข้าใช้ช่องสัญญาณด้วยกลไก CSMA/CA	28
บทที่ 5 แอดฮอกรูทีนงที่ดิงโทรโตคอลล	31
5.1 ตัวอย่างแอดฮอกรูทีนงที่ดิงโทรโตคอลล	31
5.2 วิธีแบบ Table-Driven	31
5.2.1 Destination Sequenced Distance Vector	32
5.2.2 Wireless Routing Protocol	32
5.2.3 Cluster Switch Gateway Routing	33
5.3 วิธีแบบ Source-Initiated On-Demand	35
5.3.1 Ad Hoc On-Demand Distance Vector Routing	35
5.3.2 Dynamic Source Routing	37
5.3.3 Temporally Ordered Routing Algorithm	39
5.3.4 Signal Stability Routing	41
5.3.5 Ad hoc Virtual Switching Routing Protocol	42
5.4 ปัจจัยที่ต้องพิจารณาในการเลือกรูทีนงที่ดิงโทรโตคอลล	42
บทที่ 6 หลักการพื้นฐานของการสร้างโปรแกรมจำลองระบบ	44
6.1 ความเป็นมาของโปรแกรมจำลองระบบ	44
6.2 System Model and Simulation	44
6.3 Discrete-event simulation	46
6.4 Time-Advance Mechanisms	47
6.5 Components and Organization of a Discrete-Event Simulation Model	47

สารบัญ(ต่อ)

	หน้าที่
บทที่ 7 หลักการสร้างและออกแบบโปรแกรม	49
7.1 โครงสร้างของ โปรแกรม	49
7.2 การจัดการส่วนติดต่อกับผู้ใช้	51
7.3 การจัดการในส่วนของการรันแบบจำลอง	53
7.4 การจัดการตำแหน่งของเครื่องคอมพิวเตอร์	55
7.5 จัดการเกี่ยวกับโพรโทคอล	57
7.6 การจัดการเกี่ยวกับการรengel	58
บทที่ 8 การทดสอบการทำงานของโปรแกรม	61
บทที่ 9 บทวิจารณ์และสรุป	69
9.1 ขอบเขตและข้อจำกัดของ โปรแกรม	69
9.2 กลุ่มผู้ใช้โปรแกรม	69
9.3 ปัญหาและอุปสรรค	69
9.4 แนวทางการประยุกต์และพัฒนา	70
ภาคผนวก ก	71
บรรณานุกรม	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ

	หน้าที่
รูปที่ 2-1 แสดงลำดับชั้นของมาตรฐาน OSI	4
รูปที่ 3-1 ปัญหาของเทอร์มินัลถูกปิดบัง	16
รูปที่ 3-2 แสดงการใช้ RTS-CTS แชนด์เช็กแก้ปัญหา โหนดถูกปิดบัง	17
รูปที่ 3-3 ความไม่สมบูรณ์ของวิธี RTS-CTS	18
รูปที่ 3-4 ปัญหาที่เกิดขึ้นอีกอย่างหนึ่งจาก RTS-CTS	19
รูปที่ 3-5 ปัญหาของ Exposed node	19
รูปที่ 3-6 ปัญหาที่เกิดจากการใช้อากาศแบบทุกทิศทุกทาง	20
รูปที่ 3-7 เป็นกรแก้ปัญหาของ exposed node โดยใช้สเปซเทคแบบมีทิศทาง	21
รูปที่ 3-8 การเริ่มต้นแมคโปรโตคอลฝั่งผู้รับ	22
รูปที่ 3-9 การเริ่มต้นแมคโปรโตคอลฝั่งผู้ส่ง	22
รูปที่ 4-1 แสดง BSS และ ESS	27
รูปที่ 4-2 แสดงการทำงานใน โหมด Ad hoc หรือ Peer-to-Peer Mode	28
รูปที่ 4-3 แสดงHidden Node Problem และ กลไก RTS/CTS Handshake	30
รูปที่ 5-1 ประเภทของแอตตริบิวต์ที่ติดโปรโตคอล	31
รูปที่ 5-2 เส้นทาง CSGR ซึ่งจำเป็นต้องผ่านกลัสเตอร์เฮดในเส้นทาง	34
รูปที่ 5-3 AODV route request	36
รูปที่ 5-4 AODV route reply	37
รูปที่ 5-5 DSR route request	38
รูปที่ 5-6 DSR route reply	38
รูปที่ 5-7 TORA route creation	40
รูปที่ 5-8 TORA route maintenance	40
รูปที่ 6-1 แนวทางการศึกษาระบบ	45
รูปที่ 6-2 Flow of control for next-event time-advance approach	48
รูปที่ 7-1 การออกแบบโปรแกรม	49
รูปที่ 7-2 Class Diagram	50
รูปที่ 7-3 หน้าจอหลักของโปรแกรม	52
รูปที่ 7-4 ขั้นตอนการทำงานของโปรแกรมในส่วนของการรันแบบจำลอง	54
รูปที่ 7-5 หน้าจอแสดงการรายงานผล	59
รูปที่ 8-1 ภาพแสดงเครือข่ายที่จะเริ่มทำการทดสอบที่ 1	61
รูปที่ 8-2 ภาพแสดงเครือข่ายหลังจากการทดสอบแล้ว	62
รูปที่ 8-3 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 1	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพประกอบ(ต่อ)

	หน้าที่
รูปที่ 8-4 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 2	63
รูปที่ 8-5 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 3	63
รูปที่ 8-6 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 4	64
รูปที่ 8-7 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 5	64
รูปที่ 8-8 ภาพแสดงเครือข่ายที่จะเริ่มทำการทดลองที่ 2	65
รูปที่ 8-9 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 1	66
รูปที่ 8-10 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 2	66
รูปที่ 8-11 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 3	67
รูปที่ 8-12 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 4	67
รูปที่ 8-13 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 5	68
รูปที่ 10-1 แสดงขั้นตอนการทำงานของโปรแกรมจำลองระบบ Single Queue Model	72
รูปที่ 10-2 แสดงขั้นตอนการ Simulation ของ โปรแกรม	73
รูปที่ 10-3 แสดงขั้นตอนการ Generate Random Number	74

สารบัญตาราง

ตารางที่ 3-1 แสดงคุณสมบัติของอุปกรณ์โมบายแต่ละตัว

หน้าที่

10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ปัจจุบันระบบเครือข่ายคอมพิวเตอร์มีความสำคัญมากขึ้น ความต้องการในการนำคอมพิวเตอร์ไปใช้ในการติดต่อสื่อสารหรือเชื่อมต่อเพื่อให้บริการแก่กันก็ย่อมมากขึ้นตามไปด้วย เครือข่ายแอดฮอคเป็นทางเลือกหนึ่งในการเชื่อมต่อที่ต้องการการลงทุนต่ำและเหมาะสำหรับเครือข่ายที่มีอายุการใช้งานในระยะสั้น แต่เครือข่ายแอดฮอกลก็ยังมีข้อจำกัดอยู่มาก โพรโตคอลต้องจัดการกับข้อจำกัดต่างๆของเครือข่ายแบบนี้ ซึ่งรวมถึงการบริโภคพลังงานที่สูง แบนด์วิดท์ต่ำ และมีอัตราความผิดพลาดที่สูง มีโพรโตคอลจำนวนมากมาถูกพัฒนาขึ้นสำหรับเครือข่ายแอดฮอค การเลือกโพรโตคอลให้เหมาะสมกับเครือข่ายจึงเป็นสิ่งจำเป็น

ดังนั้นการจัดทำโปรแกรมจำลองเหตุการณ์เพื่อศึกษาโพรโตคอลการพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอค จึงเป็นแนวคิดที่ต้องการเปิดโอกาสทางการศึกษาให้กับบุคคลทั่วไปที่มีความสนใจทางด้านเราต์ติง โพรโตคอลแบบแอดฮอคแต่ยังไม่มีความรู้และความชำนาญพอที่จะทำความเข้าใจกับการทำงานของโพรโตคอล หรือไม่มีเครือข่ายจริงให้ทำการศึกษา ได้มีโอกาสในการศึกษาและวิเคราะห์การทำงานของโพรโตคอล เพื่อเป็นแนวทางในการเลือกใช้โพรโตคอลให้เหมาะกับเครือข่ายหรือพัฒนาเราต์ติง โพรโตคอลต่อไป

1.2 หลักการและเหตุผล

โปรแกรมจำลองเหตุการณ์เพื่อศึกษาโพรโตคอลการพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอค เป็นโปรแกรมที่พัฒนาขึ้นเพื่อจำลองการทำงานของเครือข่ายไร้สายแบบแอดฮอคและการทำงานของแอดฮอคเราต์ติงโพรโตคอล โดยสามารถกำหนดคุณสมบัติของเครือข่าย ทำการติดต่อกับผู้ใช้ผ่านหน้าจอกกราฟฟิก และสามารถแสดงผลภาพจำลองเครือข่ายในลักษณะสองมิติได้ โปรแกรมนี้เขียนด้วยภาษาจาวา จึงสามารถทำงานได้กับทุกแพลตฟอร์ม

1.3 วัตถุประสงค์ของปริญญาานิพนธ์

วัตถุประสงค์ของการจัดทำโปรแกรมจำลองการทำงานของเราต์ติงโพรโตคอลแบบแอดฮอคได้แก่

1. เพื่อจัดทำโปรแกรมจำลองการทำงานของเราต์ติงโพรโตคอลแบบแอดฮอค ทำให้สามารถทดสอบประสิทธิภาพของเราต์ติงโพรโตคอลได้
2. เพื่อศึกษาการทำงานของเราต์ติงโพรโตคอลที่ใช้กับเครือข่ายแบบแอดฮอค
3. เป็นเครื่องมือสำหรับจำลองเหตุการณ์และผลกระทบที่จะเกิดขึ้นของการให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของปริญญาโท

- โปรแกรมที่สามารถจำลองการทำงานของเรตติ้ง โปรโตคอลได้
- สามารถแสดงประสิทธิภาพการทำงานของเรตติ้งโปรโตคอลได้
- โปรแกรมสามารถแสดงผลในรูปแบบของกราฟฟิกส์
- สามารถตั้งค่าคุณสมบัติของเครือข่ายได้

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาหลักการของการสร้างโปรแกรมจำลองระบบ
2. ทำการสร้างโปรแกรมจำลองอย่างง่ายเพื่อศึกษาการทำงานและฝึกฝนการสร้างโปรแกรม
3. ศึกษาหลักการและการทำงานของเครือข่ายแบบแอดฮอค
4. ศึกษาหลักการของเรตติ้งโปรโตคอลแบบต่างๆ
5. ออกแบบโปรแกรมจำลองการทำงานของเรตติ้งโปรโตคอล โดยมีโปรโตคอลอย่างน้อยหนึ่งโปรโตคอล
6. เขียนโปรแกรมจำลองการทำงานของโปรโตคอล
7. ทดสอบโปรแกรมและตรวจสอบข้อผิดพลาด

ทำบทเรียนสอนการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ความรู้พื้นฐาน

2.1 เครือข่ายคอมพิวเตอร์ (Computer Network)

เครือข่ายคอมพิวเตอร์ หมายถึง เครื่องคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปที่เป็นอิสระต่อกัน นำมาเชื่อมต่อกันได้โดยไม่คำนึงถึงระยะทางระหว่างเครื่องทั้งสอง โดยมีจุดประสงค์เพื่อให้เครื่องคอมพิวเตอร์สามารถติดต่อสื่อสาร และแลกเปลี่ยนข้อมูลระหว่างกันได้ เครือข่ายคอมพิวเตอร์เริ่มจากเครือข่ายขนาดเล็กภายในองค์กรที่เชื่อมโยงกันภายใต้สภาพพื้นที่จำกัดซึ่งเรียกว่า [14] เครือข่ายท้องถิ่น (LAN: Local Area Network) เมื่อเชื่อมโยงเครือข่ายเข้าด้วยกัน และขยายขอบเขตครอบคลุมพื้นที่ระหว่างเมือง หรือระหว่างประเทศ ก็จะเรียกเครือข่าวนั้นว่า เครือข่ายวงกว้าง (WAN: Wide Area Network)

2.2 โปรแกรมสำหรับเครือข่าย (Network Software)

2.2.1 ลำดับชั้นของโพรโตคอล (Protocol Hierarchies)

เพื่อเป็นการลดความซับซ้อนของการออกแบบโปรแกรมทั้งระบบในคราวเดียวกัน ระบบโปรแกรมเครือข่ายส่วนมากจะแบ่งเป็นการทำงานออกเป็นหลายระดับ หรือหลายชั้น แต่ละชั้นจะสร้างฟังก์ชันการทำงานขึ้น โดยอาศัยการทำงานของฟังก์ชันต่าง ๆ ที่สร้างไว้ในชั้นระดับล่างลงมา จำนวนชั้น, ชื่อที่เรียก และฟังก์ชันการทำงานของเครือข่ายต่าง ๆ จะแตกต่างกันออกไป อย่างไรก็ตาม ทุกระบบจะมีแนวความคิดอย่างเดียวกัน ก็คือการเรียกใช้บริการจากชั้นล่าง และการให้บริการแก่ชั้นบน โดยซ่อนรายละเอียดและความซับซ้อนของฟังก์ชันในแต่ละชั้นไว้ภายใน

ในการสื่อสารที่เกิดขึ้นระหว่างผู้ส่งข้อมูลกับผู้รับข้อมูลนั้น จะเสมือนว่าเป็นการติดต่อในชั้นเดียวกัน โดยในแต่ละชั้นจะมีโพรโตคอลของตนเอง ซึ่งจะช่วยให้ผู้ส่งข้อมูลและผู้รับข้อมูลสามารถติดต่อกันได้ แต่ในความจริงแล้ว การสื่อสารจะเกิดขึ้นจริง โดยผ่านสายสื่อสารที่อยู่ชั้นหนึ่งเท่านั้น ข้อมูลที่สื่อสารในระหว่างชั้นต่าง ๆ จะถูกส่งต่อกันเป็นลำดับจากชั้นบนสู่ชั้นล่าง จากฝ่ายของทางผู้ส่งข้อมูล และทางฝ่ายของผู้รับข้อมูลจะทำการรับข้อมูลจากชั้นหนึ่ง แล้วจึงถูกแยกข้อมูลออกไปทีละชั้น จนถึงชั้นบนสุดเพื่อใช้ในการประมวลผลต่อไป

2.2.2 ระบบมาตรฐาน OSI (The OSI Reference Model)

เมื่อคอมพิวเตอร์ของเรามีการรับส่งข้อมูลกับคอมพิวเตอร์เครื่องอื่น ๆ การเชื่อมต่อคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกันเป็นระบบเครือข่ายก็เกิดขึ้น อย่างไรก็ตาม การเชื่อมต่อคอมพิวเตอร์คนละระบบ หรือ คนละยี่ห้อเป็นสิ่งที่ทำได้ยากในชุดแรก ๆ ของการสื่อสารข้อมูล เนื่องจากขาดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล ส่วนมากแต่ละยี่ห้อก็จะมีมาตรฐานของตนเอง ซึ่งเข้ากันไม่ได้กับของ

ที่ห่ออื่น ทำให้ผู้ใช้ต้องผูกติดอยู่กับผู้ผลิตแต่ละยี่ห้อ และเป็นขีดจำกัดในการเชื่อมต่อคอมพิวเตอร์คนละชนิด ไม่ให้รับส่งข้อมูลกันได้ ระบบคอมพิวเตอร์ในยุคนั้น จึงเป็นระบบปิด (Closed System) นั่นเอง

ปัญหานี้ ทำให้หน่วยงานกำหนดมาตรฐานสากล คือ International Standards Organization หรือ ISO จัดการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลจากคอมพิวเตอร์ระบบหนึ่งไปยังคอมพิวเตอร์อีกระบบหนึ่งขึ้น จุดมุ่งหมายก็เพื่อเปิดช่องทางให้ข้อมูลที่เก็บอยู่ในระบบคอมพิวเตอร์หนึ่ง ๆ รับส่งไปยังคอมพิวเตอร์ที่เป็นระบบเดียวกัน หรือต่างระบบได้อย่างอิสระ โดยไม่ขึ้นกับผู้ผลิตอย่างที่เป็นอย่างอยู่ในอดีต ซึ่งเป็นการทำงานแบบที่เรียกว่า ระบบเปิด (Open systems) เราเรียกโครงสร้างของมาตรฐานการรับส่งข้อมูลนี้ว่า (Open System Interconnection) หรือ OSI ซึ่งจัดทำขึ้นราวกลางปี ค.ศ. 1970 โดยองค์การ International Standards Organization (ISO) และใช้อ้างอิงกันมาจนถึงในศตวรรษปัจจุบัน

โมเดล OSI มีทั้ง 7 ชั้นด้วยกัน หรือที่เราเรียกว่า OSI 7-Layers Model โดยตัวโครงสร้างจะเน้นความสำคัญของรูปแบบการติดต่อสื่อสาร ระหว่างระบบเปิด (open systems) กับระบบเปิด จึงสามารถนำไปใช้อ้างอิงได้ในระดับสากลอย่างแท้จริง

แนวความคิดของการกำหนดมาตรฐานเป็นแบบชั้นสื่อสาร (Layers) คือ

1. ชั้นสื่อสารแต่ละชั้นถูกกำหนดขึ้นมาตามบทบาทที่แตกต่างกัน
2. แต่ละฟังก์ชันในชั้นใด ๆ จะต้องกำหนดขึ้นมาโดยใช้แนวความคิดในระดับสากลเป็นวัตถุประสงค์หลัก
3. ขอบเขตความรับผิดชอบของแต่ละชั้น จะต้องกำหนดขึ้นมาเพื่อจำกัดปริมาณการแลกเปลี่ยนข้อมูลและ ผลกระทบข้างเคียงระหว่างการติดต่อให้มีน้อยที่สุด
4. ในแต่ละชั้น จะเสมือนเชื่อมต่ออยู่กับชั้นเดียวกันของอีกฝั่งหนึ่ง โดยทำงานเสมือนกับว่า มีการติดต่อรับส่งข้อมูลกับกลไกในชั้นเดียวกัน แต่จะมีเพียงชั้นหนึ่ง ซึ่งเป็นชั้นล่างสุดเท่านั้น ที่มีการเชื่อมต่อ และรับส่งข้อมูลผ่านสายส่งข้อมูลระหว่างคอมพิวเตอร์ทั้งสองระบบ
5. ในแต่ละชั้น ที่ทำหน้าที่รับส่งข้อมูลจะมีการติดต่อรับส่งข้อมูลกับชั้นที่อยู่ติดกับตนเองเท่านั้น ไม่สามารถติดต่อรับส่งข้อมูลข้ามชั้นได้

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

รูปที่ 2-1 แสดงลำดับชั้นของมาตรฐาน OSI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นสื่อสารกายภาพ (The Physical Layer)

ชั้นนี้ คือชั้นที่ 1 หรือชั้นล่างสุด จะทำงานเกี่ยวข้องกับอุปกรณ์สื่อสารต่าง ๆ ทำหน้าที่ในการกำหนดวิธีการควบคุมการรับ และ การส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ในระดับบิต ได้แก่ การส่งบิต 0 และ 1 จะแทนด้วยกระแสไฟฟ้าที่โวลต์ แต่ละบิตในระยะเวลาในการส่งนานเท่าไร การส่งเป็นแบบทางเดียวหรือสองทาง เป็นต้น จะเห็นได้ว่า การทำงานในชั้นนี้จะเกี่ยวข้องกับการทำงานของอุปกรณ์สัญญาณไฟฟ้า (หรือสัญญาณใด ๆ) ชั้นก่อนในการใช้อุปกรณ์เหล่านั้น และความสัมพันธ์กับสื่อที่รับ – ส่งสัญญาณ

ชั้นสื่อสารการเชื่อมต่อข้อมูล (The Data Link Layer)

หน้าที่หลักของชั้นนี้คือ ทำการรวบรวมข้อมูลจากชั้นที่ 1 ตรวจสอบความถูกต้องของข้อมูล แล้วส่งข้อมูลที่ปราศจากข้อผิดพลาดนี้ให้กับชั้นที่ 3 ต่อไป โดยปกติผู้ส่งข้อมูลจะแบ่งข้อมูลที่มีความยาวมากออกเป็นกลุ่มข้อมูลย่อย ๆ แต่ละส่วนย่อยเรียกว่า คาต้าเฟรม (Data Frame) ซึ่งจะมีขนาดคงที่ ชุดของคาต้าเฟรมสำหรับข้อมูลที่ต้องการส่งไปให้ผู้รับก็จะถูกส่งไปที่ละเฟรม ตั้งแต่เฟรมแรกไปจนครบทุกเฟรม ข้างฝ่ายผู้รับจะตอบสนองโดยการส่งคาต้าเฟรมพิเศษ เรียกว่า เฟรมตอบรับ (Acknowledgement frame) ไปถึงผู้ส่งเพื่อเป็นการบอกให้ทราบว่า ได้รับข้อมูลแล้ว กระบวนการรับ-ส่งข้อมูลนี้ก็จะเสร็จสิ้นสมบูรณ์

การรับ-ส่งข้อมูลในชั้นที่ 1 นั้นจะไม่รับรู้ในเรื่องโครงสร้างข้อมูล ก็จะมองเห็นข้อมูลว่าเป็นบิต 0 หรือบิต 1 กลุ่มหรือชุดหนึ่งที่เรียงตามลำดับ เรียกว่า กระแสบิต (Bit Stream) จึงเป็นหน้าที่ของชั้นที่ 2 ในการทำการตรวจสอบความถูกต้อง ซึ่งทำได้โดยการเพิ่มข้อมูลสำหรับการตรวจสอบคิ่วกับข้อมูลทุกเฟรม

การส่งข้อมูลผ่านระบบเครือข่ายใด ๆ ก็ตาม ข้อมูลที่ส่งนั้นมีโอกาสที่จะเสียหายหรือสูญหายไปเลยก็ได้ โปรแกรมในชั้นที่ 2 จะต้องสามารถตรวจสอบความผิดปกตินี้ได้เอง หรืออาจตอบสนองต่อการตรวจพบโดยโปรแกรมในชั้นที่ 1 เมื่อพบความผิดปกติแล้วก็ต้องมีวิธีการแก้ไข เช่น แจ้งให้ผู้ส่งข้อมูลได้ส่งข้อมูลชุดเดิมกลับมาใหม่ (เรียกเฟรมนี้ว่า Duplicate Frame) อย่างไรก็ตาม การส่งข้อมูลซ้ำทำให้เกิดปัญหาตามมาในกรณีที่ชุดข้อมูลไม่ได้สูญหายไปไหน เพียงแต่ใช้เวลาเดินทางมากกว่าปกติ ดังนั้นข้อมูลชุดเดียวกันก็จะมาถึงผู้ใช้ทั้งสองเฟรม โปรแกรมในชั้นนี้จึงต้องหาวิธีการตรวจสอบและกำจัดเฟรมที่ซ้ำออกไป

ชั้นสื่อสารเครือข่าย (The Network Layer)

ชั้นนี้มีหน้าที่รับผิดชอบในการควบคุมการติดต่อรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ (โหนด) ต่าง ๆ ในระบบเครือข่ายให้เป็นไปด้วยความเรียบร้อย สิ่งที่สำคัญที่สุด คือการกำหนดเส้นทางเดินของข้อมูลจากโหนดผู้ส่งข้อมูลไปตามโหนดต่าง ๆ จนถึงโหนดผู้รับข้อมูลในที่สุด โสสต์บางกลุ่มจะกำหนดเส้นทางเดินข้อมูลโดยศึกษาาระบบเครือข่ายแล้วสร้างตารางเส้นทางเดินข้อมูลแบบถาวร โสสต์บาง

กลุ่มจะกำหนดเส้นทางเดินข้อมูลในตอนเริ่มต้นของการสื่อสาร ดังนั้นการสื่อสารในครั้งต่อไป (ติดต่อกับ โหนดเดิม) อาจจะไปใช้เส้นทางอื่นได้ โหนดในกลุ่มที่มีวิธีการซับซ้อนมาก จะกำหนดเส้นทางเดินข้อมูลในระดับแพ็กเก็ต กรณีที่มีผู้ส่งข้อมูลพร้อม ๆ กันหลาย ๆ จุดจะทำให้เกิดความคับคั่งของข้อมูล คล้ายกับสภาวะจราจรในชั่วโมงเร่งด่วน ซึ่งมีปริมาณรถยนต์มากจนทำให้การจราจรติดขัด โหนดในกลุ่มนี้ ก็จะปรับเส้นทางเดินข้อมูลของแต่ละแพ็กเก็ตให้เหมาะสมกับสภาวะของระบบเครือข่ายอยู่ตลอดเวลา

การส่งผ่านข้อมูลในระบบเครือข่ายอาจมีการบันทึก ผู้ส่ง ผู้รับ และปริมาณข้อมูลที่ไหลผ่าน โหนดหรือ เราเตอร์ต่าง ๆ เพื่อประโยชน์ทางการคิดค่าบริการซึ่งจะมีความซับซ้อนมากขึ้นถ้าข้อมูล ไหลผ่านระบบเครือข่ายย่อยที่มีการคิดอัตราค่าบริการต่างกัน

ชั้นจัดการนำส่งข้อมูล (The Transport Layer)

โปรแกรมในชั้นนี้ มีหน้าที่หลักในการรับข้อมูลมาจากชั้นที่ 5 ซึ่งอาจต้องแบ่งข้อมูลออกเป็น แพ็กเก็ตขนาดย่อม (ในกรณีที่มีข้อมูลมีปริมาณมาก) หลาย ๆ แพ็กเก็ต แล้วจึงส่งข้อมูลทั้งหมดต่อไปให้ โปรแกรมในชั้นที่ 3 ทางด้าน โปรแกรมในชั้นที่ 4 ก็จะทำหน้าที่ประกอบแพ็กเก็ตชุดนี้ให้กลับมารวมกัน เป็นข้อมูลชุดเดิม

ในภาวะปกติ การเชื่อมต่อการสื่อสารจะเป็นการจัดตั้งหน้าต่างสื่อสาร (Session) ระหว่างผู้ส่ง และผู้รับตามที่เกิดขึ้น ถ้าต้องการเพิ่มประสิทธิภาพก็อาจสร้างโพลีเซสของโปรแกรมในชั้นนี้ขึ้นมาหลาย ๆ โพลีเซสเพื่อช่วยกันจัดส่งข้อมูลให้เร็วขึ้น แต่แน่นอนในด้านความประหยัดก็อาจทำในทางตรงกันข้าม นั่นคือ การรวบรวมโพลีเซสหลาย ๆ โพลีเซสให้เหลือจำนวนน้อยลงแล้วจึงจัดการให้โพลีเซสที่เหลืออยู่ทำการส่งข้อมูลทั้งหมด โดยการใช้ช่องสื่อสารร่วมกัน

โปรแกรมในชั้นนี้เป็นผู้กำหนดประเภทของการให้บริการต่าง ๆ รวมไปถึงการอำนวยความสะดวกในการใช้ระบบเครือข่ายซึ่งแบ่งออกได้เป็น 3 ประเภท ประเภทแรกเป็นการให้บริการแบบจุด-ต่อ-จุด โดยเน้นการรับประกันความถูกต้องของข้อมูลเป็นสำคัญ ประเภทที่สองเน้นการให้บริการข้อมูล ข้อมูลในระดับแพ็กเก็ตซึ่งแม้ว่าจะไม่รับประกันการสูญหายของข้อมูลแต่ก็ให้ความคล่องตัวสูงกว่าแบบแรก (การรับประกันความถูกต้องของข้อมูลสามารถทำในชั้นอื่นได้) ประเภทที่สามเป็นการส่งข้อมูลแบบกระจายข่าวเพื่อประโยชน์ในการส่งข้อมูลชุดเดียวกันไปยังผู้ใช้หลายจุดพร้อมกัน

โปรแกรมในชั้นนี้ติดต่อกันผ่านช่องสัญญาณเสมือน (Virtual Channel) ระหว่างผู้ส่งและผู้รับ โดยตรงเรียกว่า เป็นการติดต่อกับ end-to-end connection ในขณะที่โปรแกรมในสามชั้นแรกนั้นเป็นการติดต่อแบบจุด-ต่อ-จุด ซึ่งผู้รับอาจไม่ใช่ผู้รับข้อมูลแต่เป็นเพียง โหนดตัวกลางในการรับแล้วส่งข้อมูลต่อไปตามเส้นทางเดินข้อมูลที่กำหนดไว้

นอกจากการใช้ช่องสื่อสารร่วมกันแล้ว โปรแกรมในชั้นนี้ จะต้องมีความสามารถในการจัดตั้ง หน้าต่างสื่อสารกับโหนดอื่น ๆ ในระบบเครือข่ายและจัดการยกเลิกเมื่อการสื่อสารสิ้นสุดลง โปรแกรมในชั้นนี้ยังต้องมีวิธีการกำหนดการตั้งชื่อให้แก่ตนเองและแนะนำให้ผู้อื่นในระบบได้รู้จัก รวมทั้งการควบคุม

การไหลของข้อมูล ซึ่งมีทั้งในระดับโหนดและระดับเราเตอร์โดยมีวัตถุประสงค์ในการควบคุมการรับและส่งข้อมูล โดยเฉพาะ ในกรณีที่ผู้ส่งจัดการส่งข้อมูลเร็วเกินกว่าทางผู้รับจะทำงานได้ทัน

ชั้นหน้าต่างสื่อสาร (The Session Layer)

ชั้นนี้ ทำหน้าที่เป็นผู้กำหนดวิธีการควบคุมการเชื่อมต่อระหว่างผู้รับข้อมูลและผู้ส่งข้อมูลตั้งแต่เริ่มต้นการสื่อสารไปจนยุติการสื่อสาร เช่น การติดต่อขอใช้โหนดจากเครื่องคอมพิวเตอร์ที่อยู่ไกลออกไป (Remote Login) โดยภาพรวมแล้ว การให้บริการในชั้นนี้จะคล้ายกับบริการที่มีอยู่ในชั้นที่ 4 แต่ในชั้นนี้จะให้บริการหลายอย่างที่ประ โยชน์มากกว่าสำหรับการประยุกต์ใช้งานบางประเภท

หน้าที่สำคัญอย่างหนึ่งคือ บริหารการแลกเปลี่ยนข่าวสาร อันได้แก่การกำหนดให้การแลกเปลี่ยนข่าวสารเป็นไปแบบสองทางในเวลาเดียวกัน (Full duplex) หรือถ้าเป็นการสื่อสารแบบทางเดียวแต่สลับทิศทางได้ (Half duplex) ก็จะต้องเป็นผู้จัดลำดับให้ทั้งผู้ใช้และผู้ส่งทำการส่งข้อมูลได้คล้ายกับการควบคุมสลับหลักการไฟ

สำหรับการสื่อสารประเภทที่ต้องใช้โทเคน (Token) โปรแกรมในชั้นนี้ จะเป็นผู้บริหารการใช้โทเคนเพื่อให้โหนดต่าง ๆ ในระบบนี้ผลัดเปลี่ยนการครอบครองโทเคนอย่างเป็นธรรมชาติ หรือถูกต้องตามลำดับความสำคัญ

หน้าที่อีกประการหนึ่งได้แก่การแก้ปัญหาความล้มเหลวในการส่งข้อมูลขนาดใหญ่มากระหว่างโหนดต่าง ๆ ในกรณีที่การส่งข้อมูลเกิดล้มเหลวกลางคันโดยไม่มีอาการแก้ไขใด ๆ โหนดทั้งสองก็จะต้องเริ่มต้นใหม่หมด ถ้าเกิดการล้มเหลวขึ้นอีกก็จะต้องเริ่มต้นใหม่อีก วิธีการแก้ไขวิธีหนึ่งก็คือการแทรกจุดตรวจสอบความถูกต้อง (Checkpoints) เข้าไปจำนวนหนึ่ง โดยจำนวนขึ้นกับปริมาณข้อมูล ในระหว่างการส่งข้อมูล จุดตรวจสอบทั้งหมดจะต้องถูกแทรกเข้าไปในข้อมูล ที่ตำแหน่งเดียวกันของทั้งผู้ส่งและผู้รับซึ่งเรียกว่าการ Synchronization หากเกิดการล้มเหลวขึ้น โปรแกรมในชั้นนี้ของผู้รับก็จะค้นหาจุดตรวจสอบจุดสุดท้ายก่อนการล้มเหลวเพื่อลบข้อมูลส่วนที่อยู่หลังจุดตรวจสอบนั้นทิ้งไป แล้วแจ้งให้ผู้ส่งเริ่มต้นการส่งข้อมูลใหม่จากจุดตรวจสอบนั้นแทนที่จะต้องเริ่มต้นใหม่ทั้งหมด

ชั้นนำเสนอข้อมูล (The Presentation Layer)

โปรแกรมที่ทำงานในระดับชั้นต้น ๆ นั้นที่ได้กล่าวมาแล้วนั้น จะให้ความสนใจในประสิทธิภาพของการรับ-ส่งข้อมูลและมองเห็นว่าข้อมูลคือกระแสบิต (Bit Stream) หรือกระแสไบนารี (Byte Stream) เท่านั้น โปรแกรมในชั้นนี้จะมองข้อมูลว่าเป็นสิ่งที่มีรูปแบบ (Syntax) และความหมาย (Semantics) มากกว่ากระแสบิตหรือไบนารี ความแตกต่างของการให้ความหมายข้อมูลของเครื่องคอมพิวเตอร์ในระบบต่าง ๆ เป็นปัญหาที่จะต้องได้รับการแก้ไขในระดับส่วนรวมไม่ใช่ให้แต่ละฝ่ายแก้ปัญหาโดยลำพัง การควบคุมรูปแบบและความหมายของข้อมูล การใช้รหัสแทนข้อมูล หรือการแทนข้อมูลด้วยระบบต่าง ๆ รวมไปถึงการเข้ารหัสและถอดรหัส สิ่งต่าง ๆ ที่กล่าวมานี้ล้วนแต่เป็นความรับผิดชอบของโปรแกรมในชั้นนี้

ชั้นสื่อสารการประยุกต์ (The Application Layer)

ในปัจจุบัน มีจอภาพเทอร์มินัล (Terminals) อยู่หลายร้อยชนิดทั่วโลกซึ่งส่วนใหญ่จะไม่สามารถใช้ทดแทนหรือใช้งานร่วมกันได้ การติดต่อระหว่างเครื่องคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายยังงังไม่อาจสื่อสารกันได้โดยสมบูรณ์ โปรแกรมในชั้นประยุกต์จึงเข้ามามีบทบาทสำคัญสองด้านคือ การเป็นตัวกลาง หรือส่วนติดต่อระหว่างโปรแกรมประยุกต์ (Application Programs) กับโปรแกรมใน 6 ชั้นที่เหลือและ การกำหนดแบบมาตรฐานของจอ (Terminal Type)

การกำหนดแบบมาตรฐานของจอ นั้น ไม่ได้เป็นการกำหนดวิธีสร้างจอเทอร์มินัลให้เหมือนกัน แต่จะคล้ายกับการสร้างจอเทอร์มินัลเสมือน (Virtual Terminal) ขึ้นบนจอเทอร์มินัลจริง ทั้งนี้เพื่อให้จอเทอร์มินัลทุกชนิดในโลกมีความเข้าใจตรงกัน เช่น ขนาดบริเวณที่ในการการแสดงผล การเคลื่อนย้ายตำแหน่งเคอร์เซอร์ และการแสดงตัวอักษร ณ ตำแหน่งต่าง ๆ บนจอภาพ เป็นต้น จึงทำให้การใช้อจอเทอร์มินัลเพื่อการสื่อสารบนระบบเครือข่ายเกิดขึ้นได้ แม้ว่า จะใช้อจอเทอร์มินัลต่างแบบกันก็ตาม



บทที่ 3

เครือข่ายไร้สายแบบแอดฮอค

3.1 ทำไมจึงต้องเป็นเครือข่ายแบบแอดฮอค

เครือข่ายไร้สายแบบแอดฮอค (Ad hoc wireless) [15] เป็นการรวมตัวกันของอุปกรณ์ตั้งแต่ 2 ตัวขึ้นไปทำการติดต่อสื่อสารแบบไร้สาย และมีความสามารถทางด้านเครือข่ายรวมอยู่ด้วย ดังเช่นการที่อุปกรณ์สามารถติดต่อสื่อสารกับโหนดอื่นๆ ที่อยู่ใกล้เคียงกันในระยะทางที่เคลื่อนความถี่วิทยุครอบคลุมถึงหรือแม้แต่อยู่นอกระยะก็ตาม ในที่นี้จะมีโหนดระหว่างกลาง (intermediate node) ที่ทำหน้าที่คอยส่งทอดต่อ หรือฟอร์เวิร์ดแพ็คเก็ตจากต้นทางไปยังปลายทาง

เครือข่ายไร้สายแบบแอดฮอคมักจะมีการจัดการตัวเองและมีการดัดแปลง (self-organization and adaptive) ซึ่งหมายความว่ารูปแบบของเครือข่ายไม่มีรูปแบบที่ตายตัวสามารถเคลื่อนที่เปลี่ยนแปลงได้ตลอดเวลาโดยที่ไม่จำเป็นต้องมีตัวที่คอยดูแลการดำเนินการกลาง [17] คำว่าแอดฮอค (ad hoc) มีความหมายในที่นี้คือ การที่มีรูปแบบของอุปกรณ์ที่ต่างชนิดกัน แต่ละอันสามารถพบกันได้ สามารถทำงานได้ทั้งแบบเครื่องเดียว (standalone) หรือทำงานแบบเครือข่ายได้ ตัวแอดฮอคโหนดหรืออุปกรณ์ควรที่จะสามารถทำการตรวจจับอุปกรณ์อื่นๆ ที่อยู่ใกล้เคียงอาณาบริเวณเดียวกันได้รวมทั้งสามารถทำการแฮนด์เช็ก (Handshaking) ที่จำเป็นเพื่อที่จะทำให้การสื่อสารสามารถทำการแลกเปลี่ยนข้อมูลแลบริการต่างๆ ระหว่างกันได้

ตัวอุปกรณ์ไร้สายแอดฮอคสามารถมีรูปแบบชนิดต่างๆ กันได้ เช่น palmtop, laptop, internet mobilephone เป็นต้น ความสามารถในการประมวลผล การเก็บข้อมูลและการติดต่อสื่อสารในอุปกรณ์แต่ละตัวมีความแตกต่างกันอย่างมากมาย อุปกรณ์แอดฮอคไม่ควรจะมีเฉพาะการตรวจจับสัญญาณการเชื่อมต่อกับอุปกรณ์อื่นๆ ที่อยู่ใกล้เคียงเท่านั้นแต่ควรที่จะจำแนกชนิดของอุปกรณ์นั้นๆ และคุณสมบัติต่างๆ ที่ใกล้เคียงกันกับตัวมันเอง เพราะว่าเครือข่ายไร้สายแบบแอดฮอคไม่สามารถที่จะไว้ใจระบบเครือข่ายแบบคงที่ที่มีอยู่ขณะปัจจุบันระบบเครือข่ายของมันจำเป็นที่จะต้องเป็นแบบ infrastructureless หรือไม่มีรูปแบบโครงสร้างของเครือข่ายที่ตายตัว พวกมันไม่จำเป็นที่จะต้องมีสถานีวิทยุฐาน (radio base station) ไม่มีสายหรือตัวเราเตอร์ที่อยู่กับที่อย่างไรก็ตามอุปกรณ์โมบายที่อยู่ในเครือข่ายไร้สายแบบแอดฮอคจะต้องมีการเปลี่ยนแปลงข้อมูลเร้าติง (routing) อยู่เสมอ เมื่อมีการเปลี่ยนแปลงลักษณะของการเชื่อมต่อเครือข่าย

อุปกรณ์แอดฮอคโมบายต่างๆ นานา มีความจุของแบตเตอรี่แตกต่างกันตามแต่ละอุปกรณ์ แต่เครือข่ายแบบแอดฮอคจะต้องอาศัยการฟอร์เวิร์ดข้อมูลแพ็คเก็ตโดยโหนดต่างๆ ดังนั้นเรื่องของการใช้พลังงานในอุปกรณ์แต่ละตัวจึงเป็นประเด็นที่ต้องระมัดระวัง

3.2 ลักษณะความแตกต่างกันของอุปกรณ์โมบาย

อุปกรณ์โมบายในปัจจุบันมีอยู่หลายชนิดหลายรูปแบบด้วยกัน มีความแตกต่างกันอย่างมากมานักซึ่งส่งผลต่อการออกแบบโทร โคคอลของและประสิทธิภาพของการติดต่อสื่อสาร

จากตารางที่ 3-1 แสดงถึงคุณลักษณะของระบบของอุปกรณ์โมบายต่างที่มีอยู่ซึ่งแสดงให้เห็นถึงความแตกต่างทางด้านขนาด ความสามารถในการคำนวณ หน่วยความจำ ดิสก์ และความจุของแบตเตอรี่ อุปกรณ์โมบายมีความสามารถที่จะกระทำคล้ายเซิร์ฟเวอร์หรือผู้ให้บริการ โดยขึ้นอยู่กับความสามารถในการคำนวณของมัน ขนาดของหน่วยความจำ ขนาดของตัวเก็บข้อมูล และระยะเวลาในการใช้งานของแบตเตอรี่ ความแตกต่างที่มีอยู่นี้แสดงว่าอุปกรณ์บางตัวมีความสามารถที่สูงกว่าอุปกรณ์ตัวอื่นๆ และบางอุปกรณ์สามารถเป็นเซิร์ฟเวอร์ได้ในขณะที่ตัวอื่นๆ เป็นได้เพียงแค่อุปกรณ์ นอกจากนั้นการส่งทอดต่อกันของแท่งที่เครื่องอื่นๆ เป็นผลที่ทำให้อุปกรณ์มีการใช้พลังงานภายในเครื่องของตนเอง เนื่องจากเหตุนี้โมบายโหนดควรทำการตรวจพลังงานของตนเองว่าเพียงพอที่จะใช้งานได้เป็นอย่างดีก่อนที่จะทำการฟอร์เวิร์ดแท่งที่เครื่องอื่นๆ

ตารางที่ 3-1 แสดงให้เห็นว่าถึงแม้ว่าจะมีความแตกต่างในอุปกรณ์แอสคโมบาย พวกมันก็ยังคงสามารถที่จะทำการเชื่อมต่อเครือข่ายเข้าด้วยกันแบบไร้สาย อย่างไรก็ตามข้อมูลการประมวลผลไม่จำเป็นที่จะต้องเหมือนกัน Palm pilot จะมีหน้าจอแสดงผลที่เล็กไม่ควรแสดงผลแอนิเมชันที่เป็นเวิร์ดโปรเซสซึ่งขนาดใหญ่ Active badge มีส่วนที่ติดต่อกับผู้ใช้สำหรับเวิร์ดโปรเซสซึ่งอย่างจำกัด มันจึงกระทำสิ่งอื่นที่นอกเหนือไปจากนี้เช่นเป็นเซ็นเซอร์ และ actuator มากกว่าที่จะต้องเป็นยูสเซอร์เทอร์มินัล

Device Type	Form Factor	CPU (MIPS)	Memory (MB)	Disk	Battery (Watt-hr)
Palm Pilot	3.5x4.7 cm	2.7	2-8	None	3-5.5
Active Badge	3.5x3.5 cm	Philips 87C571	64 bytes	None	5V
Cellular Phone	2.5x5.5 cm	16	1 Mbit	None	10-20 mA (3.6V)
Pocket PC	13x7.8 cm	130-224	32-64	16 MB Flash ROM	3-5
Laptop Computer	12x9 cm	1306-2123	32-128	5-20 GB	37.44-66.60

ตารางที่ 3-1 แสดงคุณสมบัติของอุปกรณ์โมบายแต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ชนิดของการสื่อสารบนแอดฮอคโมบาย

ตัวโมบายโฮสต์ในระบบสื่อสารไร้สายแบบแอดฮอคนั้นสามารถทำการสื่อสารกับตัว Immediate peers นั่นคือเป็นการสื่อสารแบบเพียร์ทูเพียร์(peer-to-peer) ที่อยู่ในระยะสัญญาณวิทยุสอปเดียวกัน อย่างไรก็ตามหากมีสามโหนดขึ้นไปโดยที่โหนดอยู่ภายในระยะเรนจ์ของโหนดอื่นๆ แต่ละโหนด โดยที่ไม่จำเป็นต้องจะตั้งอยู่ในสอปเดียวกันกับโหนดอื่นๆ จากนั้นการติดต่อสื่อสารของโมบายโหนดแบบรีโมทรีโมท (remote-to-remote) จึงจะใช้ได้ การสื่อสารแบบรีโมทรีโมทก็มีความเกี่ยวข้องกับกลุ่มอุปกรณ์ที่ต้องมีการเคลื่อนย้ายตลอดเวลาเช่นเดียวกัน ความแตกต่างของชนิดอุปกรณ์สื่อสารแบบแอดฮอคนี้ส่งผลให้ลักษณะของทราฟฟิกมีความแตกต่างกันด้วย

3.4 รูปแบบของการเคลื่อนไหวของตัวโมบายโฮสต์

ในส่วนนี้จะทำการพิจารณารูปแบบของการเคลื่อนไหวของตัวโมบายโฮสต์ที่ส่งผลต่อความถูกต้องของการเร้าท์หาเส้นทางโดยตรง

1. การเคลื่อนไหวจาก โหนดในเส้นทาง

ในการสื่อสารแบบแอดฮอคจะต้องประกอบไปด้วยโหนดต้นทาง (Source node) โหนดปลายทาง (Destination node) และ โหนดระหว่างกลาง (intermediate node) อื่นๆ โหนด การเคลื่อนไหวของโหนดต่างเหล่านี้จะส่งผลกระทบต่อการทำงานทั้งเส้น ตัวโหนดต้นทางในเส้นทางจะมีดาวน์สตรีมลิงค์ (Downstream link) เมื่อมีการเคลื่อนที่ออกจากระยะทางครอบคลุมสัญญาณวิทยุคว้านสตรีมของโหนดข้างเคียงของตัวเอง เส้นทางต่างๆ ที่มีอยู่จะใช้งานไม่ได้ เนื่องจากเหตุนี้เองตัวดาวน์สตรีมโหนดทั้งหมดจึงควรที่จะต้องทำการลบข้อมูลในการเร้าท์หาเส้นทางที่ไม่ถูกต้องออกไปเช่นเดียวกันหากว่าโหนดปลายทางมีการเคลื่อนไหวออกจากระยะครอบคลุมของสัญญาณวิทยุอัพสตรีม (Upstream) ของโหนดข้างเคียงของมันเองข้อมูลของการเร้าท์หาเส้นทางเดิมที่มีอยู่ก็จะไม่สามารถใช้ได้ และเช่นเดียวกันกับโหนดต้นทางและ โหนดปลายทาง การเคลื่อนไหวต่างๆที่เกิดขึ้นในตัว โหนดระหว่างกลางก็จะส่งผลกระทบต่อการทำงานได้เช่นเดียวกัน

การเคลื่อนไหวต่างๆ ที่เกิดขึ้นทั้งหมดนี้ จึงมีแบบแผนของเราที่ดึงโรรโทคอล (Routing protocol) กระจายตัวอย่างมากมายเพื่อที่จะให้มีการสนองที่เหมือนกันในการเปลี่ยนการเชื่อมต่อ การอัปเดตโหนดต่างๆ ที่มีเหลืออยู่ทั้งหมดในระบบเครือข่ายต้องมีข้อมูลเร้าท์ที่ตรงกันและสามารถที่จะทำการปรับปรุงเปลี่ยนแปลงได้ อย่างไรก็ตามกระบวนการอัปเดตนี้ต้องอาศัยการบรอดคาสต์ (Bandwidth) และจะทำให้ทราฟฟิกทั้งหมดของเครือข่ายเพิ่มขึ้น เนื่องจากเหตุนี้เองจึงจำเป็นต้องมีตัวเร้าท์ติ้งโรรโทคอลใหม่ๆ

2. การเคลื่อนไหวจาก โหนดที่เชื่อมข้ามเน็ตเข้าด้วยกัน (Subnet-Bridging Node)

นอกจากการเคลื่อนไหวของโหนดต่างๆ ดังที่ได้กล่าวไปแล้ว ยังมีการเคลื่อนไหวของโหนดที่ทำหน้าที่ในการเชื่อมต่อข้ามเน็ตระหว่าง โมบายสองข้ามเน็ตเข้าด้วยกันสามารถที่จะทำการแบ่งส่วน โมบายข้ามเน็ตให้มีขนาดของข้ามเน็ตที่เล็กลงได้ โหนดต้นทางและปลายทางจะต้องเป็นส่วนที่มีเส้นทางเร้าท์อยู่

ในซบ้นต์เท่านั้นจึงจะเป็นคุณลักษณะของโมบายซบ้นต์ ยกเว้นซบ้นต์ที่ถูกแบ่ง โดยตัว โหนดที่ทำการเชื่อมต้อซบ้นต์เข้าด้วยกันการเคลื่อนไหว โหนดต่างๆ สามารถส่งผลต่อการเชื่อมต้อซบ้นต์เข้าด้วยกันได้ เช่นกันทำให้เกิดการเปลี่ยนรูปแบบเป็นซบ้นต์ที่ใหญ่ขึ้นได้

เมื่อมีการเชื่อมต้อซบ้นต์ให้มีรูปแบบซบ้นต์ที่ใหญ่ขึ้น ตัวเรที่คิงอัลกอริทึมจะยอมรับซบ้นต์ใหม่นี้โดยการอัปเดตตารางเรที่คิงของ โหนดทั้งหมดซึ่งทำให้ขาดประสิทธิภาพ การที่จะทำให้การเรที่คิงมีประสิทธิภาพควรทำการละทิ้งการดำเนินการเช่นนี้ และควรเลือกทำการอัปเดตเฉพาะ โหนดที่ได้รับผลกระทบจากการรวมกันของตาราง ซึ่งเป็นหน้าที่ของคาล์คูลเลเตอร์ที่เป็นส่วนประจําตัวใน โหนดอยู่แล้ว เรที่คิงใช้หลักการนี้ในการแบ่งซบ้นต์เช่นเดียวกัน

การที่จะทำประโยชน์ให้ปรากฏได้นั้น โมบายซบ้นต์จะต้องสามารถถูกใช้งานให้ร่วมมือในการประมวลผลเพื่อการเดินทางของข้อมูล และจะต้องมีผู้ร่วมเพื่อทำการร่วมมือให้การทำจะขยายขนาดเมื่อมีการรวมต้อเข้าด้วยกันของทั้งสองกลุ่ม หรือเมื่อมีโมบายของผู้ใช้เข้ามาในระยะ สัญญาวิทยุเพื่อต้อเชื่อมเข้าด้วยกัน

3. การเคลื่อนไหวของ โหนดที่พร้อมกัน

ในความเป็นจริงแล้วมีการเคลื่อนไหวของ โหนดต่างๆ เกิดขึ้นพร้อมกัน โดยโหนดค้นทางโหนดปลายทาง หรือ โหนดระหว่างกลางที่มีอยู่ เนื่องจากเหตุนี้เองจึงจำเป็นต้องมีกฎเกณฑ์เพื่อที่จะสร้างความมั่นใจได้ว่ามีความตรงกันของ โหนดต่างๆ เหล่านี้เมื่อมีการคอนฟิกเส้นทางต่างๆ หลายเส้นทางใหม่หรือทำการแก้ไขการดำเนินการที่ถูกอ้างถึง แต่ละการดำเนินการเมื่อกระทำการคอนฟิกเส้นทางใหม่ควรให้มีความเหมาะสมมากที่สุด โดยที่ท้ายสุดแล้วจะมีการบรรจบกันของเส้นทาง

3.5 ประเด็นที่ควรให้ความสำคัญในเครือข่ายไร้สายแบบแอดฮอค

1. การกำหนดระยะเวลาคลื่นความถี่ที่ใช้งานและการเช่าซื้อ

โดยทั่วไปแล้วการพิจารณาการใช้คลื่นความถี่ในปัจจุบันนี้อยู่ภายใต้ข้อบังคับของ FCC การทดลองระบบเครือข่ายแบบแอดฮอคส่วนใหญ่แล้วอยู่บนแถบคลื่นวิทยุ ISM เพื่อที่จะทำการป้องกันการรบกวนกันของคลื่นวิทยุแล้วระบบเครือข่ายแบบแอดฮอคจะต้องมีการใช้งานภายใต้ระยะเวลาคลื่นความถี่ที่ได้รับอนุญาตเท่านั้น ส่วนใหญ่แล้วคลื่น ไมโครเวฟจะใช้งานที่แถบความถี่ 2.4 GHz. ซึ่งสามารถที่จะเกิดการรบกวนกับระบบเครือข่ายแบบไร้สายได้ ความถี่ไม่ได้มีเฉพาะที่ถูกจำกัดการควบคุมและแบ่งปันจัดสรรเท่านั้น แต่ยังมีแถบความถี่ที่ถูกซื้อด้วย และด้วยความสามารถในการติดต่อสื่อสารของระบบเครือข่ายแอดฮอคที่สามารถทำการจัดรูปแบบและยกเลิกรูปแบบการสื่อสารได้อย่างอิสระและไม่หยุดนิ่งอยู่กับที่จึงทำให้เกิดความไม่กระจ่างว่าควรจะใช้ค่าใช้จําสำหรับการใช้แถบความถี่นี้ให้กับผู้ใด

2. การเข้าใช้งานสื่อ (Media Access)

เครือข่ายไร้สายแบบแอดฮอคปราศจากการควบคุมการซิงโครไนซ์ (Synchronization) ทั้งหมดจากศูนย์กลางซึ่งแตกต่างกับระบบเครือข่ายแบบเซลลูลาร์ (Cellular) เนื่องจากเหตุนี้การใช้หลักการ TDMA และ FDMA จึงไม่เหมาะสม รวมทั้งโพรโทคอล MAC (Media Access Control) อีก

หลายๆ โพรโตคอลไม่ได้ทำการจัดการกับโหนดที่มีการเคลื่อนไหวอยู่เสมอ ดังนั้นการที่จะทำรายการมารับส่งเฟรมข้อมูลให้ถูกเวลาเพื่อให้รองรับกับ QoS ทำได้ยาก

ในเครือข่ายไร้สายแบบแอดฮอคมีสื่อเดียวกันสำหรับการแบ่งปันให้กับโหนดแอดฮอคโมบายที่มีจำนวนหลายโหนดนั้น การเข้าใช้งานช่องสัญญาณโดยทั่วไปแล้วจะต้องทำการแพร่สัญญาณผ่านโพรโตคอล MAC ซึ่งยอมให้เกิดความจริงที่ว่าที่สายไม่มีตัวโหนดใดๆ อยู่คงที่สาย ตัวโหนดต่างๆ ไม่สามารถอาศัยการประสานงานจากศูนย์กลางได้ ดังนั้นตัวโพรโตคอล MAC ต้องทำการแข่งขันเพื่อที่จะเข้าใช้งานแชนแนลที่เกิดขึ้นในเวลาเดียวกัน โดยหลีกเลี่ยงการชนกันของข้อมูลกับโหนดข้างเคียง การเคลื่อนไหวที่เกิดขึ้นอยู่อย่างเสมอในขณะนั้นทำให้เกิดปัญหาของปลายทางถูกปิดและปัญหาโหนด exposed จึงเป็นเรื่องที่จะต้องถูกแก้ไขเมื่อจะทำการออกแบบโพรโตคอล MAC สำหรับระบบเครือข่ายไร้สายแบบแอดฮอค

3. การหาเส้นทาง (Routing)

การเคลื่อนที่อยู่เสมอของตัวอุปกรณ์จะต้องมีการสร้างและทำลายการเชื่อมต่ออยู่บ่อยๆ ดังนั้นจึงมีการใช้อัลกอริทึมในการหาเส้นทาง Bellman-Ford ซึ่งได้รับการยอมรับว่าดีถูกใช้ในการดูแลและอัปเดตข้อมูลเราต์ติ้งในเครือข่ายแพ็กเกตเรดิโอ ในขณะที่การหาเส้นทางแบบดิสแทนซ์เวกเตอร์ (Distancevector) ไม่ได้รับการออกแบบมาสำหรับเครือข่ายแบบไร้สาย มันยังคงใช้ได้กับเครือข่ายแพ็กเกตเรดิโอเพราะว่าการเคลื่อนที่มีอัตราไม่สูงมาก คลื่นวิทยุที่มีขนาดใหญ่เป็นก้อนจะทำการแตกแยกออกเป็นขนาดเล็กๆ ความก้าวหน้าของเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์สามารถทำให้อุปกรณ์มีขนาดรูปร่างเล็ก ถือกักได้เคลื่อนที่ได้สะดวก ดังนั้นเครือข่ายโมบายแบบแอดฮอคนั้นแตกต่างกับเครือข่ายแพ็กเกตเรดิโอเพราะว่าโหนดสามารถที่จะทำการเคลื่อนที่ได้อิสระกว่าจึงทำให้มีการเปลี่ยนโทโพโลยีที่ไม่หยุดนิ่ง โพรโตคอลการหาเส้นทางแบบดิสแทนซ์เวกเตอร์และเล็งส์เตจไม่สามารถตามจับการเปลี่ยนแปลงลักษณะการเชื่อมต่อที่เกิดขึ้นบ่อยครั้งในเครือข่ายไร้สายแบบแอดฮอคได้ทำให้ผลลัพธ์ที่ได้มีเส้นทางบรรจบกันที่ไม่ดีและมี Throughput น้อย จึงจำเป็นที่ต้องการวิธีการหาเส้นทางใหม่ที่ดีและเหมาะสมกว่านี้

4. ประสิทธิภาพของพลังงาน

โพรโตคอลของระบบเครือข่ายที่อยู่ส่วนใหญ่ไม่ได้พิจารณาถึงเรื่องการผลิตพลังงานประเด็นอยู่ที่ว่าพวกมันสมมุติโหนดที่มีอยู่ว่าเป็นโหนดที่คงที่และตัวเราท์ติ้งหาเส้นทาง ซึ่งเป็นอุปกรณ์ที่ใช้พลังงานเป็นหลัก อย่างไรก็ตามอุปกรณ์โมบายในทุกวันนี้ส่วนใหญ่จะใช้พลังงานจากแบตเตอรี่เทคโนโลยีของแบตเตอรี่ยังถือว่าตามหลังเทคโนโลยีของไมโครโพรเซสเซอร์ ระยะเวลาที่สามารถใช้งานของแบตเตอรี่แบบ Li-ion ในปัจจุบันนี้สามารถใช้ได้เพียง 2-3 ชั่วโมง ซึ่งเป็นการจำกัดระยะเวลาการใช้งาน ดังนั้นอุปกรณ์จึงจำเป็นต้องมีระบบประหยัดพลังงานรวมอยู่ด้วย ในรายละเอียดสำหรับระบบเครือข่ายโมบายแบบแอดฮอคนี้ อุปกรณ์แต่ละตัวต้องกระทำตามหน้าที่สองหน้าที่คือ ทำหน้าที่ในการติดต่อกับผู้ใช้ซึ่งควรจะทราบว่ามีประสิทธิภาพโต้ตอบกับผู้ใช้ เมื่อไหร่ควรจะทำการเอ็กซ์คิวิต์แอฟฟิเคชันของผู้ใช้ และอีกหน้าที่หนึ่งก็คือการส่งข้อมูลในระบบเครื่อง(ส่งต่อแพ็กเกต) เนื่องจากเหตุนี้เองการส่งต่อแพ็กเกต

เพื่อให้โหนดอื่นๆ สามารถส่งข้อมูลผ่านโหนดตัวเองได้จะทำให้เกิดการผลาญพลังงานขึ้น ซึ่งเป็นเรื่องที่ต้องคำนึงถึงค่าความถี่ที่เคียวสำหรับโหนดแต่ละ โหนดในระบบเครือข่ายไร้สายแบบแอดฮอค

5. คุณสมบัตินี้ของ TCP

ทีซีพี (Transmission Control Protocol) เป็น โพรโตคอลที่ได้รับการออกแบบเพื่อทำการควบคุมการไหลของข้อมูลและควบคุมความแออัดของเครือข่ายจากต้นทางสู่ปลายทาง ทีซีพีเป็นโพรโตคอลที่มีการสถาปนาการเชื่อมต่อ (Connection-oriented) การเชื่อมต่อจะถูกกำจัดเมื่อมีการส่งข้อมูลเสร็จสมบูรณ์แล้ว อินเทอร์เน็ตในปัจจุบันใช้เน็ตเวิร์กโพรโตคอลแบบ Connectionless ซึ่งโพรโตคอลแบบนี้ อาจจะมีชั้นระดับทรานสปอร์ตโพรโตคอลที่เชื่อถือได้ผู้ดำเนินการระดับเน็ตเวิร์กโพรโตคอลที่เชื่อถือไม่ได้ ถ้าต้องการการเชื่อมต่อแบบ Connection-oriented อย่างไรก็ตามตัวทีซีพีจะสมมุติโหนดในเส้นทางว่าเป็นโหนดคงที่และจะทำการควบคุมการไหลและความแออัดของข้อมูลที่โหนดต้นทางและ โหนดปลายทาง

ทีซีพีอาศัยการวัดค่า Round-trip time (RTT) และแพ็กเก็ตจะหายไปถ้าหากว่ามีความแออัดเกิดขึ้นในระบบเครือข่าย โครข่ายที่ทีซีพีไม่สามารถแยกแยะ โหนดที่เปลี่ยนแปลงอยู่ตลอดเวลาและความแออัดในเครือข่าย การที่โหนดต่างๆ มีการเคลื่อนไหวอยู่ตลอดเวลาในขณะที่การเชื่อมต่ออาจจะทำให้แพ็กเก็ตสูญหายและมีค่า RTT ที่สูงได้ เนื่องจากเหตุนี้เองจึงมีการเพิ่มเติมหรือเปลี่ยนแปลงคุณสมบัติในระดับชั้นทรานสปอร์ตโพรโตคอลที่จำเป็นเพื่อที่จะสร้างความแน่ใจได้ว่าจะกระทำสิ่งที่จะต้อง โดยปราศจากผลกระทบต่อ throughput จากต้นทางไปยังปลายทาง

6. ความลับและความปลอดภัย (Security & Privacy)

เครือข่ายแอดฮอคเป็นเครือข่ายแบบอินทราเน็ต (Intranets) และพวกมันยังคงเป็นอินทราเน็ตจนกว่าจะมีการเชื่อมต่อกับอินเทอร์เน็ต ดังนั้นผู้ที่ไม่ได้อยู่ในพื้นที่โลคอลเดียวกันก็ไม่สามารถเข้าถึงได้เครือข่ายจะมีการตรวจสอบตัวตนและทำการรับรองโหนดข้างเคียงซึ่งทำให้ผู้ใช้ทราบได้ว่าโหนดข้างเคียงนั้นเป็นมิตรหรือว่าผู้ไม่ประสงค์ดี ข้อมูลที่ส่งออกไปตามเส้นทางนั้นสามารถถูกป้องกันได้ ถ้ามีจำนวนโหนดห้อมล้อมกันเป็นจำนวนมากในการส่งทอดต่อของแพ็กเก็ตจะมีการรับรอง โดยการจดจำผู้เริ่มต้นรับส่งแพ็กเก็ตและเส้นทางตามไอดีหรือลายเบล

7. ข้อกำหนดและการแอดเซส

การสื่อสารแบบแอดฮอคทุกโหนดจำเป็นที่จะต้องสามารถแอดเซสได้อย่างเท่าเทียมกัน แต่ถ้าหากเราต้องการสมมติการติดต่อให้เป็นแบบไคลเอนท์-เซิร์ฟเวอร์ RPC (remote procedure call) ระบบเครือข่ายแอดฮอคนั้นประกอบไปด้วยอุปกรณ์ต่างๆที่แตกต่างกันมากมายซึ่งจะมีอุปกรณ์บางชนิดเท่านั้นที่สามารถเป็นเซิร์ฟเวอร์ได้ แนวคิดของไคลเอนท์คือทำการร้องขอไปที่เซิร์ฟเวอร์สำหรับการดำเนินงานและรอคอยผลลัพธ์ที่ตอบกลับมาจากเซิร์ฟเวอร์ อาจจะไม่ดึงดูดด้วยข้อกำหนดทางด้านแบนด์วิดท์และพลังงาน บางทีแนวคิดของรีโมทโปรแกรมมิ่ง (remote programming) คล้ายกับรูปแบบที่ถูกใช้โมบายเอเจนท์ซึ่งใช้ประโยชน์ได้มากกว่าเพราะว่าวิธีนี้สามารถลดจำนวนครั้งการแลกเปลี่ยนข้อมูลระหว่างไคลเอนท์กับเซิร์ฟเวอร์ผ่านทางสื่อแบบไร้สาย แล้วเราจะทำอย่างไรให้อุปกรณ์โมบายสามารถที่จะแอดเซสการบริการแบบรีโมทในเครือข่ายแอดฮอค จะทำอย่างไรให้อุปกรณ์ที่มีขีดความสามารถที่ต่ำกว่าให้เครื่อง

อื่นๆที่เป็สมาชิกอื่นในระบบเครือข่ายที่ต้องการใช้งานบริการที่จัดไว้ให้ทราบได้เรื่องเหล่านี้เป็นประเด็นที่ต้องทำการวิจัย

3.6 โพรโทคอลการเข้าใช้งานสื่อของการสื่อสารไร้สายแบบแอดฮอค (Media Access Protocols)

สื่อแบบไร้สายสามารถทำกรแบ่งปันและ โหนดใดๆ สามารถทำการรับ-ส่งได้ตำแหน่งใดๆ ภายในเวลาหนึ่งสิ่งนี้จึงทำให้ผลลัพธ์ที่มีความเป็นไปได้ที่จะต้องมีการแย่งชิงการใช้งานช่องสัญญาณซึ่งเป็นผลทำให้มี Throughput ที่ต่ำในการสื่อสารไร้สายแบบแอดฮอคนั้นทุกๆ โหนดสามารถที่จะเคลื่อนย้ายคั้งนั้นจึงไม่มีตัวเน็ตเวิร์กโหนดที่ทำหน้าที่เป็นตัวควบคุมกลาง

โพรโทคอล MAC คือกฎที่กำหนดหรือการปฏิบัติที่ทำให้การใช้สื่อกลางที่ถูกแบ่งปันอยู่ให้มีประสิทธิภาพ เรามิยามให้โหนดซึ่งเป็นโฮสต์ที่ทดลองเข้าใช้งานสื่อคั้งนี้คือ ผู้ส่ง คือโหนดที่พยายามที่จะส่งถ่ายการติดต่อสื่อสารผ่านทางสื่อ ผู้รับ คือ โหนดที่รอรับการติดต่อจากการสื่อสาร ณ ปัจจุบัน โพรโทคอล MAC เกี่ยวข้องกับ ต่อลิงค์การสื่อสาร (per-link communication) ไม่ใช่จากปลายถึงปลาย (end-to-end)

Synchronous MAC Protocols

ในการสื่อสารแบบแมคโพรโทคอลแบบซิงโครนัสนั้น ทุกโหนดจะซิงโครไนซ์ที่เวลาเดียวกันซึ่งจะได้รับจากตัวนับเวลาที่ทำการบรอดคาสต์ ทุกโหนดจะคอยฟังสัญญาณนาฬิกาและทำการซิงโครไนซ์กับสัญญาณนาฬิกาเหล่านั้น

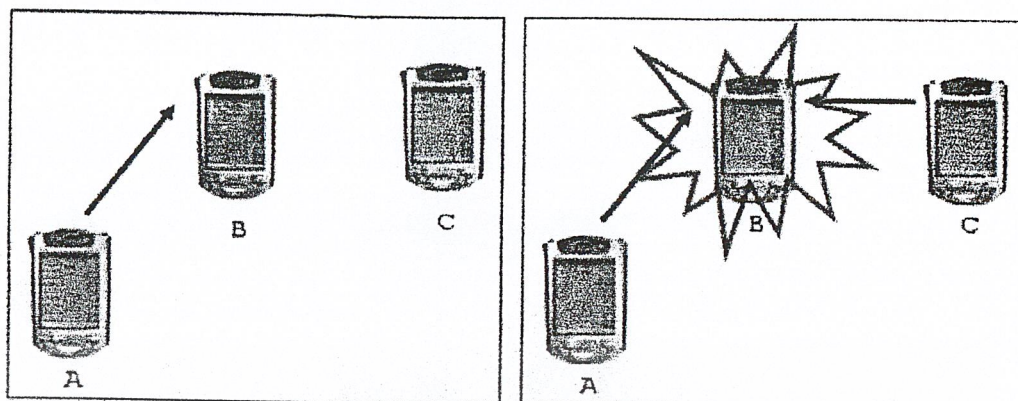
Asynchronous MAC Protocols

ในแมคโพรโทคอลแบบอะซิงโครนัสโหนดไม่จำเป็นต้องทำตามเวลาเดียวกัน

3.7 ปัญหาในการเข้าใช้งานช่องสัญญาณ

1. ปัญหาเทอร์มินัลถูกปิดบัง (Hidden Terminal)

เป็นที่รู้กันโดยทั่วไปว่าปัญหาลักของโพรโทคอลที่ใช้การแย่งชิงเป็นหลัก (contention-based protocols) ดังเช่นใน pure ALOHA, slotted ALOHA, CSMA, IEEE 802.11 เป็นต้น จะมีสองโหนดที่ต้องการทำการติดต่อสื่อสารเมื่อโหนดทั้งสองพยายามที่จะส่งข้อมูลลงในโหนดผู้รับเดียวกัน ผลลัพธ์จะเกิดการชนกันของข้อมูลเกิดขึ้นที่โหนดฝั่งผู้รับ

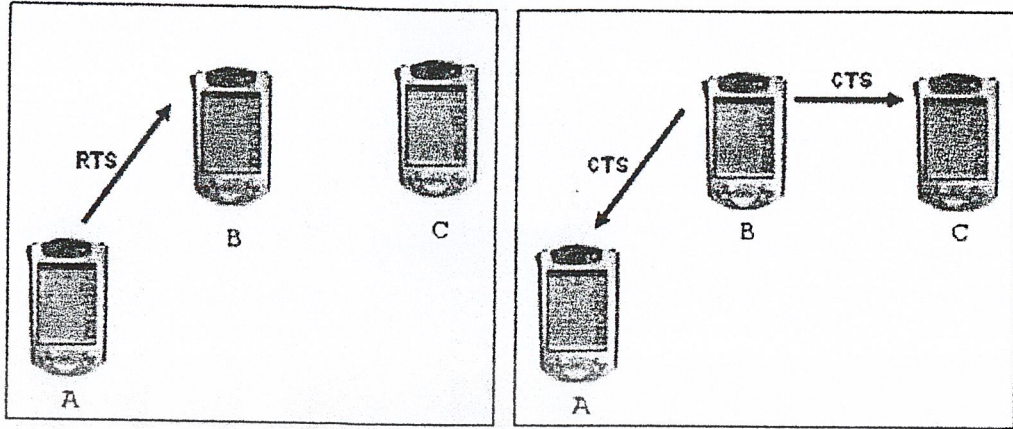


1. A ติดต่อสื่อสารกับ B (C ไม่ได้มีการติดต่อ)

2. C ติดต่อสื่อสารกับ B ซึ่งเกิดภาวะชนกันเกิดขึ้น

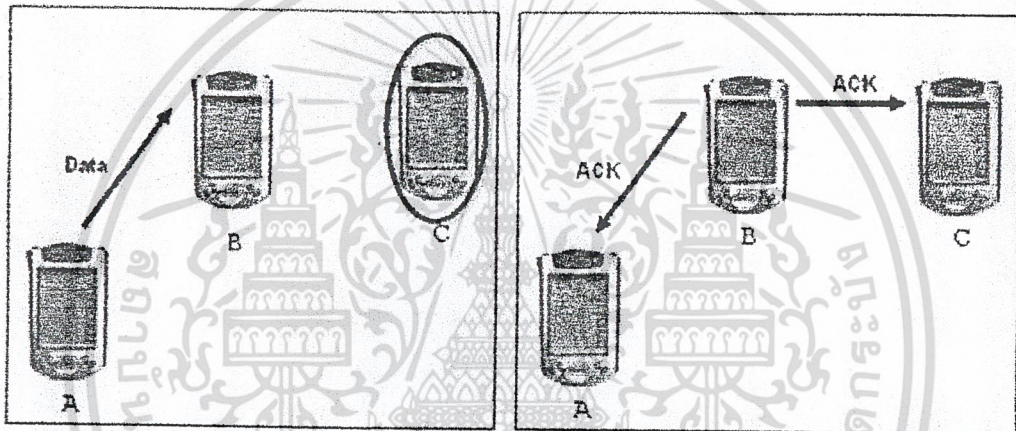
รูปที่ 3-1 ปัญหาของเทอร์มินัลถูกปิดบัง

เพื่อที่จะทำการหลีกเลี่ยงการชนกันนั้น ทุกๆ โหนดข้างเคียงของโหนดผู้รับจำเป็นที่จะต้องทราบว่าช่องสัญญาณมีผู้ครอบครองแล้ว โดยมีการทำแฮนด์เช็ก โพรโตคอล RTS แมสเสจ (Request to Send) จะใช้ในการบ่งบอกว่ามีความต้องการที่จะส่งข้อมูล โหนดผู้รับสามารถที่จะยอมรับการติดต่อสื่อสารนี้โดยการส่ง CTS แมสเสจ (Clear to Send) เพื่อทำการยอมรับ เพราะโดยธรรมชาติของการส่งแมสเสจเหล่านี้ ทุกๆ โหนดข้างเคียงของโหนดผู้ส่งและผู้รับจะสามารถรับรู้ว่ามีสื่ออาจจะไม่ว่าง ด้วยเหตุนี้เองจึงสามารถที่จะทำการป้องกันการติดต่อสื่อสารและหลีกเลี่ยงการชนกัน



1. A ส่งขอการติดต่อสื่อสารกับ B (RTS)

2. B ตอบกลับว่าช่องสื่อสารนี้ว่าง (CTS) ทั้ง A และ C จะได้เป็นการ broadcast สัญญาณ



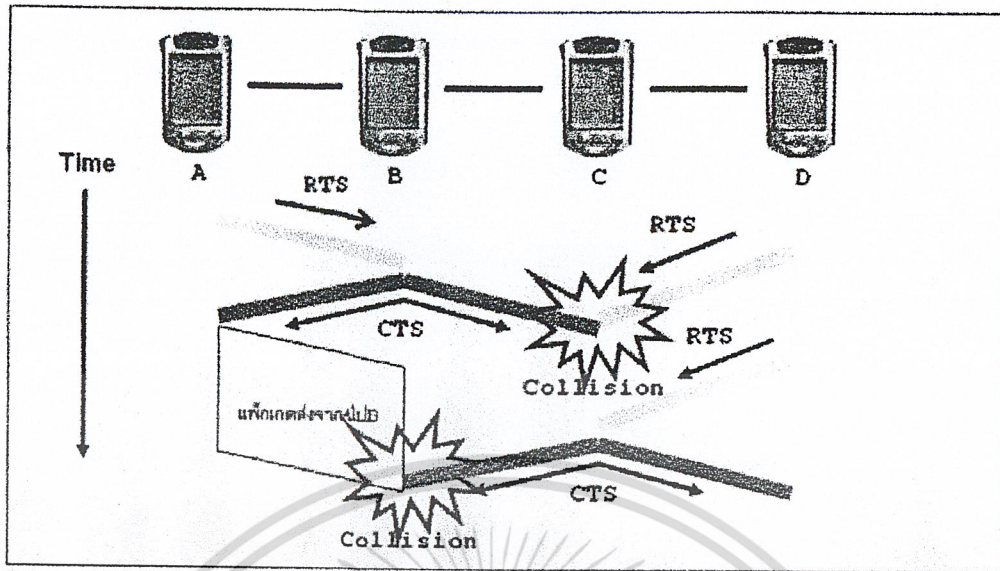
3. A ทำการส่งข้อมูลให้กับ B
C ถูกบดบังจากการสื่อสาร

4. B ตอบ acknowledges ข้อมูลที่ทำการรับ-ส่ง

รูปที่ 3-2 แสดงการใช้ RTS-CTS แชนด์เซ็คแก้ปัญหาโหนดถูกปิดบัง

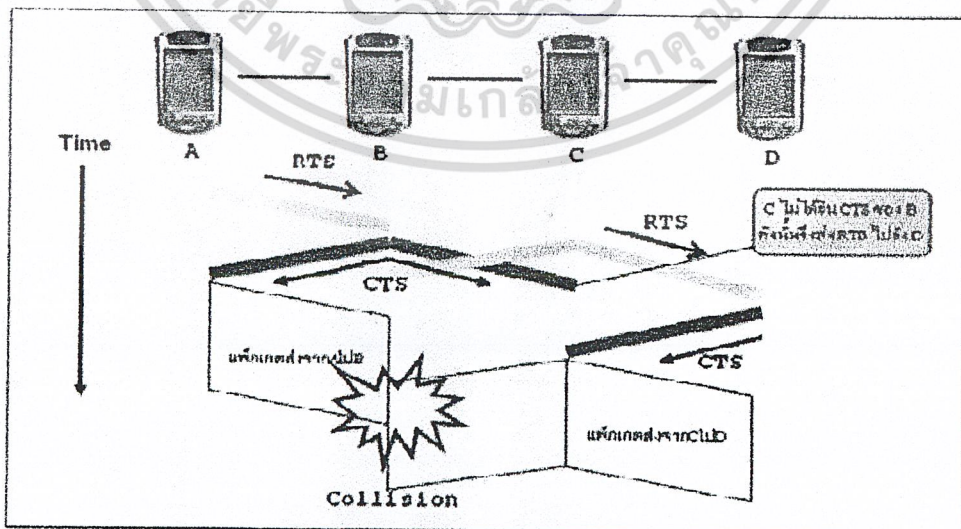
2. ปัญหา Shortcoming ของ RTS-CTS

วิธีของ RTS-CTS ไม่ใช่แนวทางแก้ไขปัญหาที่สมบูรณ์ของปัญหาเทอร์มินัลถูกปิดบัง เช่น อาจเกิดกรณีที่มีเกิดการชนกันเกิดขึ้น ตัวแม่สแตจควบคุม RTS และ CTS จะถูกส่งโดยโหนดต่างๆ กันดังแสดงในรูปที่ 3-3 โหนด B ตอบตอบ CTS ไปที่โหนดที่ส่ง RTS อย่างไรก็ตามเกิดการชนกันกับ RTS ที่ถูกส่งโดยโหนด D ที่จุดโหนด C ดังนั้นโหนด D คือเทอร์มินัลที่ถูกปิดบังจากโหนด B เพราะว่าโหนด D ไม่ได้รับสัญญาณ CTS ที่ควรจะ ได้รับจากโหนด C มันจะทำการส่งการติดต่อสื่อสาร RTS มาอีกครั้งเมื่อโหนด A ได้รับ CTS ก็จะ ไม่ทราบว่ามีเกิดการชนกันเกิดขึ้นที่โหนด C เนื่องจากเหตุนี้เองจึงดำเนินการต่อไปโดยส่งข้อมูลไปให้กับโหนด B แต่โชคไม่ดีที่เหตุการณ์นี้เกิดการชนกันกับ CTS ที่ถูกส่งมาจากโหนด C เพื่อตอบสนองสัญญาณ RTS ของโหนด D



รูปที่ 3-3 ความไม่สมบูรณ์ของวิธี RTS-CTS

เหตุการณ์ปัญหาอื่นๆ เกิดขึ้นเมื่อมีแมสเสจ CTS หลายๆแมสเสจที่ถูกยอมรับกับโหนดข้างเคียงต่างๆ ซึ่งเป็นเหตุให้เกิดการชนกันขึ้น ดังรูปที่ 3-4 มี 2 โหนดที่ทำการส่งแมสเสจ RTS ไปที่โหนดต่างๆ ในเวลาที่แตกต่างกัน โหนด A ส่ง RTS ไปหาโหนด B เมื่อโหนด B ทำการส่งแมสเสจ CTS กลับมายังโหนด A โหนด C ก็ทำการส่งแมสเสจ RTS มายังโหนด B เพราะที่โหนด C เองไม่ได้ยิน CTS ที่ถูกส่งมาจากโหนด B ในขณะที่มันทำการส่ง RTS ไปที่โหนด D โหนด C ซึ่งไม่ทราบว่ามี การติดต่อสื่อสารระหว่างโหนด A กับ โหนด B โหนด D จึงดำเนินกาต่อไปโดยการยอมรับโดยการส่งแมสเสจ CTS กลับมาให้ยังโหนด C เนื่องจากโหนด A และ โหนด C จึงยอมรับให้สามารถทำการรับ-ส่งข้อมูลได้จึงเกิดการชนกันขึ้นเมื่อทั้งสองเริ่มทำการส่งข้อมูล

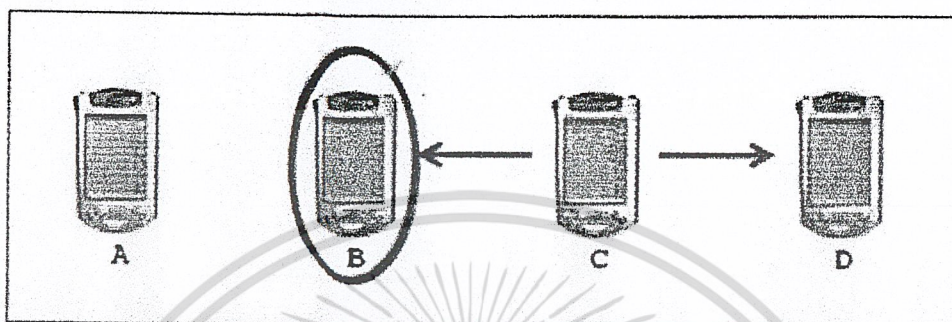


รูปที่ 3-4 ปัญหาที่เกิดขึ้นอีกอย่างหนึ่งจาก RTS-CTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปัญหา Exposed Node

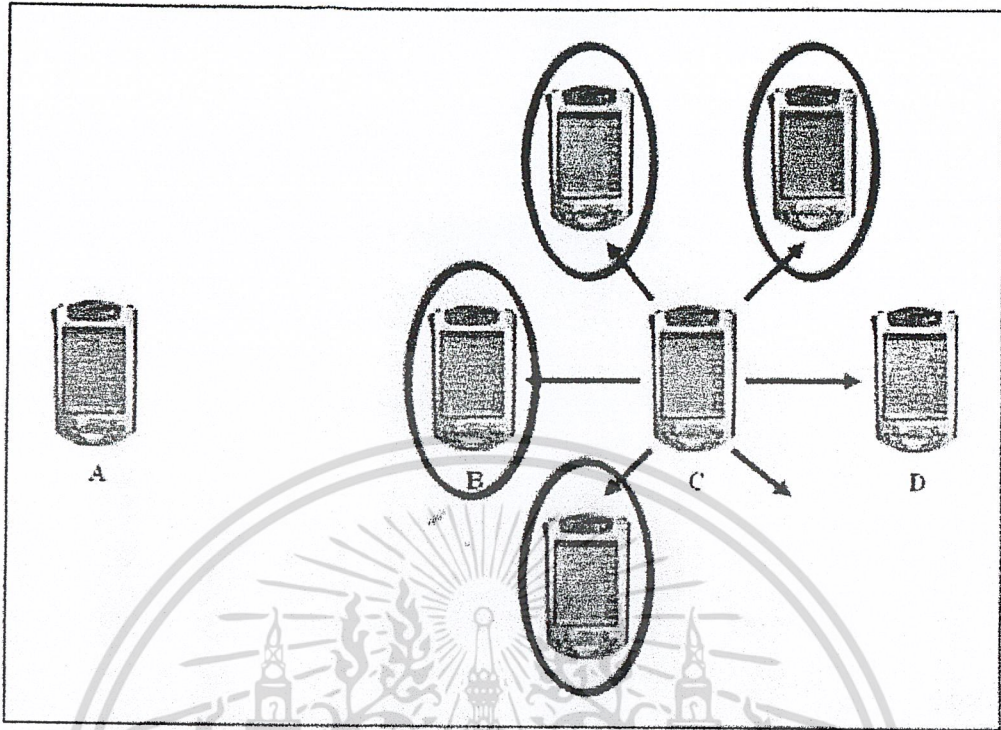
การที่ได้ยินการรับ-ส่งข้อมูลจากโหนดข้างเคียงซึ่งสามารถยับยั้งโหนดใดโหนดหนึ่งจากการรับส่งข้อมูลกับโหนดอื่นๆ ซึ่งเป็นหาของ Exposed node ซึ่งเป็นโหนดที่อยู่ในระยะของการสื่อสารข้อมูล แต่โหนดผู้รับที่ต้องการติดต่อด้วยอยู่นอกระยะการติดต่อสื่อสารดังแสดงในรูปที่ 3-5



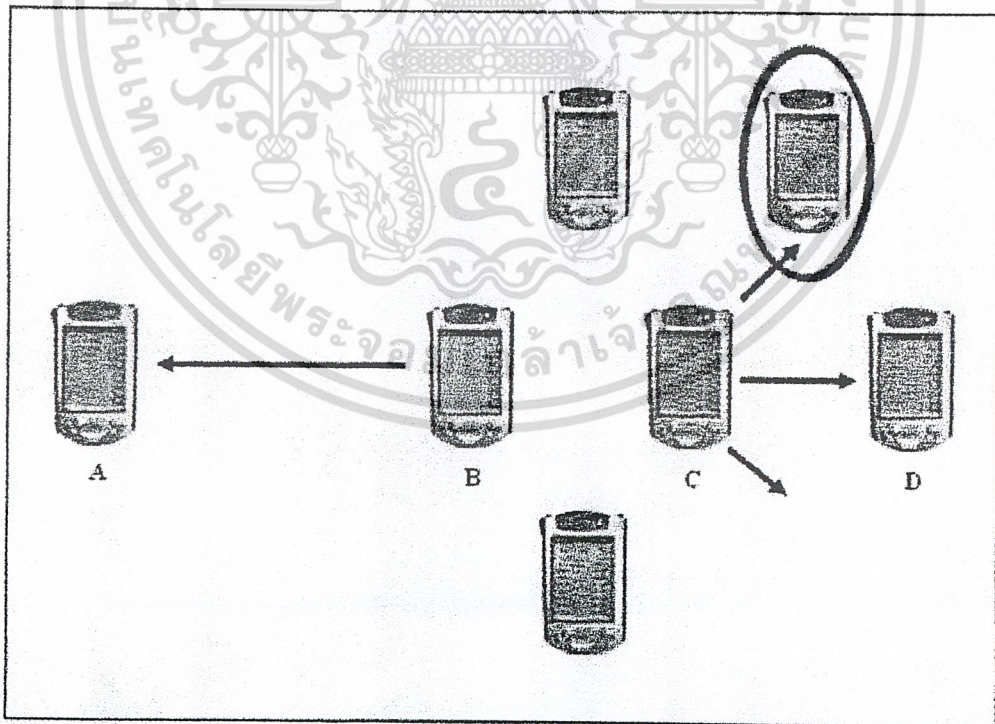
- C ทำการส่งข้อมูลให้กับ D
- B ได้ยินจึงถูกบล็อก
- B ต้องการส่งข้อมูลให้กับ A แต่ยังคงถูกบล็อกโดย C
- เป็นการสลับเปลี่ยนแบนด์วิธ

รูปที่ 3-5 ปัญหาของ Exposed node

ทางแก้ปัญหของปัญหา exposed node ซึ่งใช้การแยกการควบคุมและช่องสัญญาณข้อมูลออกจากกัน หรือใช้สายอากาศที่มีทิศทาง รูปที่ 3-6 จะแสดงถึงโมบายโหนดที่ใช้สายอากาศแบบทุกทิศทางทุกทางทำให้ผลลัพธ์ที่ได้ทำให้หลายๆ โหนดรอบข้างอยู่ในลักษณะ “exposed” ด้วยเหตุนี้เองจึงเป็นการห้ามไม่ให้พวกมันทำการติดต่อสื่อสารกับโหนดอื่นๆ ซึ่งเป็นการใช้ประโยชน์ของระบบเครือข่ายที่ต่ำและมี throughput ของระบบที่น้อยอีกทางหนึ่งหากมีการใช้สายอากาศแบบมีทิศทางปัญหานี้สามารถบรรเทาลงได้ ดังแสดงในรูปที่ 3-7 โหนด C สามารถทำการติดต่อสื่อสารกับตัวรับที่โหนด D ได้ต่อไป โดยปราศจากผลกระทบกับการติดต่อสื่อสารระหว่างโหนด A และ โหนด B การที่มีการจำกัดทิศทางการติดต่อสื่อสารเปิดช่องให้มีการจำกัดระยะทางและการแยกตัวขอความเกี่ยวข้องกัน ซึ่งจะ ไม่พบในระบบที่ใช้สายอากาศแบบทุกทิศทาง



เมื่อใช้สายอากาศแบบทุกทิศทางทุกทาง โหนดข้างเคียงทุก โหนดจะ exposed
รูปที่ 3-6 ปัญหาที่เกิดจากการใช้สายอากาศแบบทุกทิศทาง



เมื่อใช้สายอากาศแบบมีทิศทางสามารถช่วยแก้ปัญหานี้ได้ โดยที่ โหนด B ไม่ถูกล็อกสามารถส่งข้อมูลให้ A ได้

รูปที่ 3-7 เป็นการแก้ปัญหาของ exposed node โดยใช้สายอากาศแบบมีทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การเริ่มต้นแมคโปรโตคอลของฝั่งผู้รับ

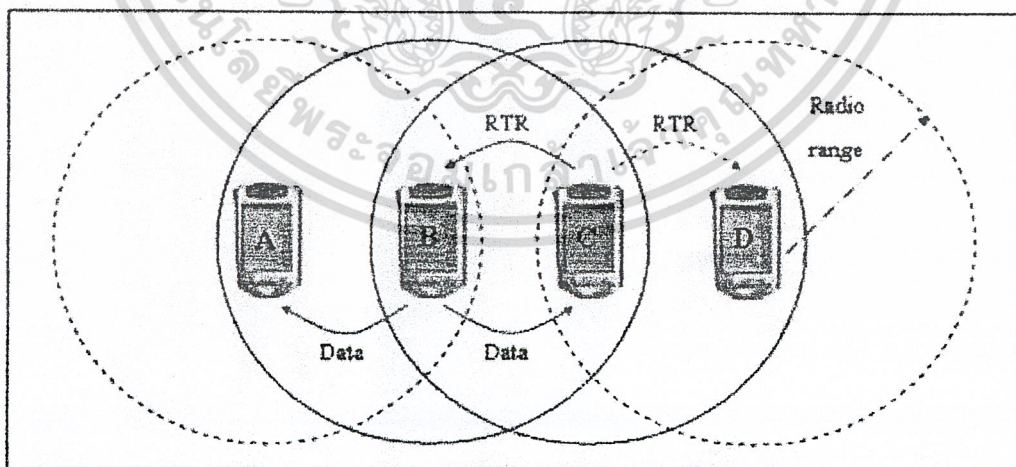
ในขณะที่แมคโปรโตคอลสามารถแยกออกเป็นประเภทๆ ได้แก่ แบบชิงโครนัสหรือแบบอะชิงโครนัส ในการใช้งานมันสามารถแบ่งแยกได้โดยผู้ที่ทำการเริ่มต้นทำการร้องขอติดต่อสื่อสารดังเช่นในรูปที่ 3-8 ฝั่งผู้รับ (โหนด C) อันดับแรกมีการติดต่อกับผู้ส่ง (โหนด B) โหนดทำการแจ้งให้ผู้ส่งได้ทราบว่าพร้อมที่จะทากรับข้อมูลแล้ว รูปแบบของการ โพลนี้จะทำให้ผู้รับไม่มีทางที่จะทราบหรือแน่ใจได้ว่าฝั่งผู้ส่งจะมีข้อมูลส่งจริงๆ

สิ่งนี้ยังรวมถึงรูปแบบพาสซีฟของการเริ่มต้นเพราะว่าผู้ส่งไม่มีการเริ่มต้นทำการร้องขอ นอกจากนี้ยังมีเพียงแค่แมสเสจควบคุมเพียงตัวเดียวที่ถูกใช้งาน โดยดูเปรียบเทียบกับวิธี RTS-CTS

3.9 การเริ่มต้นแมคโปรโตคอลของฝั่งผู้ส่ง

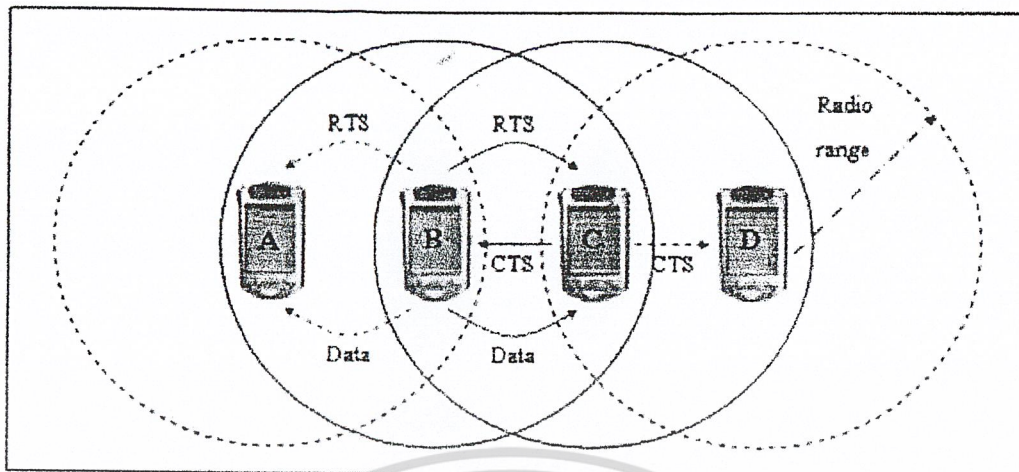
ตรงกันข้ามกับการเริ่มต้นแมคโปรโตคอลฝั่งผู้รับ ซึ่งต้องการให้ฝั่งผู้ส่งจะทำการเริ่มต้นแมคโปรโตคอล ดังนั้นฝั่งผู้ส่งต้องทำการเริ่มต้นการเชื่อมต่อการติดต่อสื่อสาร โดยทำการแจ้งไปที่ฝั่งผู้รับทราบว่าพร้อมที่จะส่งตัวอย่างของการเริ่มต้น โพรโตคอลในฝั่งผู้ส่งนี้ประกอบด้วย MACA (Multiple Access with Collision Avoidance), MACAW (MACA with Acknowledgment) และ FAMA (Floor Acquisition Multiple Access)

ดังรูปที่ 3-8 โหนด B ทำการส่งแมสเสจ RTS ที่แน่นอนไปที่โหนด C (ฝั่งผู้รับ) เพื่อที่จะแสดงว่ามีความต้องการติดต่อสื่อสาร โหนด C สามารถทำการตอบกลับหากต้องการจะรับข้อมูลจากโหนด B โดยส่งแมสเสจ CTS ตอบกลับไปที่โหนด B จากนั้นโหนด B จึงจะสามารถทำการส่งข้อมูลต่อไปได้



รูปที่ 3-8 การเริ่มต้นแมคโปรโตคอลฝั่งผู้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-9 การเริ่มต้นแมกโปรโตคอลฟังผู้ส่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

มาตรฐาน IEEE 802.11

4.1 กล่าวนำ

ปัจจุบันเทคโนโลยีเครือข่าย LAN แบบไร้สาย หรือ WLAN (Wireless LAN) กำลังได้รับความนิยมเป็นอย่างมาก เนื่องจากประโยชน์ของ WLAN มีอยู่มากมายโดยเฉพาะอย่างยิ่ง WLAN สร้างความสะดวกและอิสระในการใช้งานและติดตั้งเครือข่าย เทคโนโลยี WLAN ทำให้การเชื่อมต่ออุปกรณ์คอมพิวเตอร์ในบ้านหรือสำนักงานเข้าด้วยกันหรือต่อเข้ากับเครือข่ายไม่จำเป็นต้องใช้สายนำสัญญาณให้ยุ่งยากและดูแลรักษาอีกต่อไป อุปกรณ์คอมพิวเตอร์ทั้งแบบตั้งโต๊ะและพกพาสามารถเชื่อมต่อถึงกันหรือเชื่อมต่อเข้ากับเครือข่ายจากตำแหน่งต่างๆ ที่อยู่ในรัศมีของสัญญาณได้อย่างอิสระ

เทคโนโลยีสำหรับการเชื่อมต่ออุปกรณ์ต่างๆ ผ่านสื่อไร้สายที่รู้จักกันมีอยู่หลายเทคโนโลยีเช่น Bluetooth, IEEE 802.11, IrDA, HyperLAN, HomeRF, และ GPRS เป็นต้น แต่เทคโนโลยีที่นิยมใช้กันอย่างแพร่หลายมากที่สุดสำหรับ WLAN คือเทคโนโลยีตามมาตรฐาน IEEE 802.11 [10] เนื่องจากอุปกรณ์ IEEE 802.11 WLAN มีราคาไม่แพงนักและดูแลรักษาอีกทั้งมีสมรรถนะในการรับส่งข้อมูลค่อนข้างสูงง่ายต่อการติดตั้งและใช้งาน IEEE 802.11 WLAN ได้รับความนิยมอย่างแพร่หลายมากขึ้นเรื่อยๆ และมีแนวโน้มว่าในอนาคตอุปกรณ์คอมพิวเตอร์ต่างๆ จะมีอุปกรณ์ IEEE 802.11 WLAN ติดตั้งจากโรงงานหรือ Built-in มาด้วย

แต่อย่างไรก็ตาม ความง่ายและสะดวกต่อการติดตั้งและใช้งานของอุปกรณ์ IEEE 802.11 WLAN ก็นำมาซึ่งความไม่ปลอดภัยของเครือข่ายด้วยเช่นกัน อีกทั้งเทคโนโลยี IEEE 802.11 WLAN อยู่ในช่วงเริ่มต้นเท่านั้น (ยังไม่ถึงจุดสมบูรณ์และอิมตัว) ทำให้ยังมีช่องโหว่ด้านความปลอดภัยอีกมาก ดังนั้นผู้ที่เลือกใช้ IEEE 802.11 WLAN ควรมีความรู้เกี่ยวกับเทคโนโลยีและตระหนักถึงช่องโหว่ต่างๆ รวมถึงการรักษาความปลอดภัยอย่างเหมาะสม ซึ่งบทความนี้จะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับมาตรฐาน IEEE 802.11 รวมถึงช่องโหว่และการรักษาความปลอดภัยสำหรับเครือข่าย IEEE 802.11 WLAN

4.2 ความรู้เบื้องต้นเกี่ยวกับมาตรฐาน IEEE 802.11

มาตรฐาน IEEE 802.11 ซึ่งได้รับการตีพิมพ์ครั้งแรกเมื่อปี พ.ศ. 2540 โดย IEEE (The Institute of Electronics and Electrical Engineers) และเป็นเทคโนโลยีสำหรับ WLAN ที่นิยมใช้กันอย่างแพร่หลายมากที่สุด คือข้อกำหนด (Specification) สำหรับอุปกรณ์ WLAN ในส่วนของ Physical (PHY) Layer และ Media Access Control (MAC) Layer โดยในส่วนของ PHY Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้อุปกรณ์มีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1, 2, 5.5, 11 และ 54 Mbps โดยมีสื่อ 3 ประเภทให้เลือกใช้ได้แก่ คลื่นวิทยุที่ความถี่สาธารณะ 2.4 และ 5 GHz, และ อินฟราเรด (Infrared) (1 และ 2 Mbps เท่านั้น) สำหรับในส่วนของ MAC Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้มีกลไกการทำงานที่

เรียกว่า CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ซึ่งมีความคล้ายคลึงกับหลักการ CSMA/CD (Collision Detection) ของมาตรฐาน IEEE 802.3 Ethernet ซึ่งเป็นที่นิยมใช้กันทั่วไปในเครือข่าย LAN แบบใช้สายนำสัญญาณ นอกจากนี้ในมาตรฐาน IEEE802.11 ยังกำหนดให้มีทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย IEEE 802.11 WLAN โดยกลไกการเข้ารหัสข้อมูล (Encryption) และการตรวจสอบผู้ใช้ (Authentication) ที่มีชื่อเรียกว่า WEP (Wired Equivalent Privacy) ด้วย

4.3 วิวัฒนาการของมาตรฐาน IEEE 802.11

มาตรฐาน IEEE 802.11 ได้รับการตีพิมพ์ครั้งแรกในปี พ.ศ. 2540 ซึ่งอุปกรณ์ตามมาตรฐานดังกล่าวจะมีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1 และ 2 Mbps ด้วยคลื่นอินฟราเรด (Infrared) หรือคลื่นวิทยุที่ความถี่ 2.4 GHz และมีกลไก WEP ซึ่งเป็นทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย WLAN ได้ในระดับหนึ่ง เนื่องจากมาตรฐาน IEEE 802.11 เวอร์ชันแรกเริ่มมีประสิทธิภาพค่อนข้างต่ำและไม่มีการรองรับหลักการ Quality of Service (QoS) ซึ่งเป็นที่ต้องการของตลาด อีกทั้งกลไกรักษาความปลอดภัยที่ใช้ยังมีช่องโหว่อยู่มาก IEEE จึงได้จัดตั้งคณะทำงาน (Task Group) ขึ้นมาหลายชุดด้วยกันเพื่อทำการปรับปรุงเพิ่มเติมมาตรฐานให้มีศักยภาพสูงขึ้น โดยคณะทำงานกลุ่มที่มีผลงานที่น่าสนใจและเป็นที่ยอมรับกันดีได้แก่ IEEE 802.11a, IEEE 802.11b, IEEE 802.11e, IEEE 802.11g, และ IEEE 802.11i

4.3.1 IEEE 802.11b

คณะทำงานชุด IEEE 802.11b ได้ตีพิมพ์มาตรฐานเพิ่มเติมเมื่อปี พ.ศ. 2542 ซึ่งเป็นที่ยอมรับกันดีและใช้งานกันอย่างแพร่หลายมากที่สุด มาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ส่นวกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุความถี่ 2.4 GHz (เป็นย่านความถี่ที่เรียกว่า ISM (Industrial Scientific and Medical) ซึ่งถูกจัดสรรไว้อย่างสากลสำหรับการใช้งานอย่างสาธารณะด้านวิทยาศาสตร์ อุตสาหกรรม และการแพทย์ โดยอุปกรณ์ที่ใช้ความถี่ย่านนี้ก็เช่น IEEE 802.11, Bluetooth, โทรศัพท์ไร้สาย, และเตาไมโครเวฟ) ส่วนใหญ่แล้วอุปกรณ์ IEEE 802.11 WLAN ที่ใช้กันอยู่ในปัจจุบันจะเป็นอุปกรณ์ตามมาตรฐาน IEEE 802.11b นี้และใช้เครื่องหมายการค้าที่รู้จักกันดีในนาม Wi-Fi ซึ่งเครื่องหมายการค้าดังกล่าวถูกกำหนดขึ้นโดยสมาคม WECA (Wireless Ethernet Compatibility Alliance) โดยอุปกรณ์ที่ได้รับเครื่องหมายการค้าดังกล่าวได้ผ่านการตรวจสอบแล้วว่าเป็นไปตามมาตรฐาน IEEE 802.11b และสามารถนำไปใช้งานร่วมกับอุปกรณ์ยี่ห้ออื่นๆที่ได้รับเครื่องหมาย Wi-Fi ได้

4.3.2 IEEE 802.11a

คณะกรรมการมาตรฐาน IEEE 802.11a ได้ตีพิมพ์มาตรฐานเพิ่มเติมนี้เมื่อปี พ.ศ. 2542 มาตรฐาน IEEE 802.11a ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 54 Mbps แต่จะใช้คลื่นวิทยุที่มีความถี่ 5 GHz ซึ่งเป็นย่านความถี่สาธารณะสำหรับใช้งานในประเทศสหรัฐอเมริกาที่มีสัญญาณรบกวนจากอุปกรณ์อื่นน้อยกว่าในย่านความถี่ 2.4 GHz อย่างไรก็ตามข้อเสียหนึ่งของมาตรฐาน IEEE 802.11a ที่ใช้คลื่นวิทยุที่มีความถี่ 5 GHz ก็คือในบางประเทศย่านความถี่ดังกล่าวไม่สามารถนำมาใช้งานได้อย่างสาธารณะ ตัวอย่างเช่น ประเทศไทยไม่อนุญาตให้มีการใช้งานอุปกรณ์ IEEE 802.11a เนื่องจากความถี่ 5 GHz ได้ถูกจัดสรรสำหรับกิจการอื่นอยู่ก่อนแล้ว นอกจากนี้ข้อเสียอีกอย่างหนึ่งของอุปกรณ์ IEEE 802.11a WLAN ก็คือรัศมีของสัญญาณมีขนาดเล็กค่อนข้างสั้น (ประมาณ 30 เมตร ซึ่งสั้นกว่ารัศมีสัญญาณของอุปกรณ์ IEEE 802.11b WLAN ที่มีขนาดประมาณ 100 เมตร สำหรับการใช้งานภายในอาคาร) อีกทั้งอุปกรณ์ IEEE 802.11a WLAN ยังมีราคาสูงกว่า IEEE 802.11b WLAN ด้วย ดังนั้นอุปกรณ์ IEEE 802.11a WLAN จึงได้รับความนิยมน้อยกว่า IEEE 802.11b WLAN มาก

4.3.3 IEEE 802.11g

คณะกรรมการมาตรฐาน IEEE 802.11g ได้ใช้นวัตกรรมเทคโนโลยี OFDM มาประยุกต์ใช้ในช่องสัญญาณวิทยุความถี่ 2.4 GHz ซึ่งอุปกรณ์ IEEE 802.11g WLAN มีความสามารถในการรับส่งข้อมูลด้วยความเร็วสูงสุดที่ 54 Mbps ส่วนรัศมีสัญญาณของอุปกรณ์ IEEE 802.11g WLAN จะอยู่ระหว่างรัศมีสัญญาณของอุปกรณ์ IEEE 802.11a และ IEEE 802.11b เนื่องจากความถี่ 2.4 GHz เป็นย่านความถี่สาธารณะสากล อีกทั้งอุปกรณ์ IEEE 802.11g WLAN สามารถทำงานร่วมกับอุปกรณ์ IEEE 802.11b WLAN ได้ (backward-compatible) ดังนั้นจึงมีแนวโน้มสูงกว่าอุปกรณ์ IEEE 802.11g WLAN จะได้รับความนิยมอย่างแพร่หลายหากมีราคาไม่แพงจนเกินไปและน่าจะมาแทนที่ IEEE 802.11b ในที่สุด ตามแผนการแล้วมาตรฐาน IEEE 802.11g จะได้รับการตีพิมพ์ประมาณช่วงกลางปี พ.ศ. 2546

4.3.4 IEEE 802.11e

คณะกรรมการนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 เพื่อให้สามารถรองรับการใช้งานหลักการ Quality of Service สำหรับ application เกี่ยวกับมัลติมีเดีย (Multimedia) เนื่องจาก IEEE 802.11e เป็นการปรับปรุง MAC Layer ดังนั้นมาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชันได้ แต่อย่างไรก็ตามการทำงานของคณะกรรมการนี้ยังไม่แล้วเสร็จในขณะนี้ (พฤษภาคม พ.ศ. 2546)

4.3.5 IEEE 802.11i

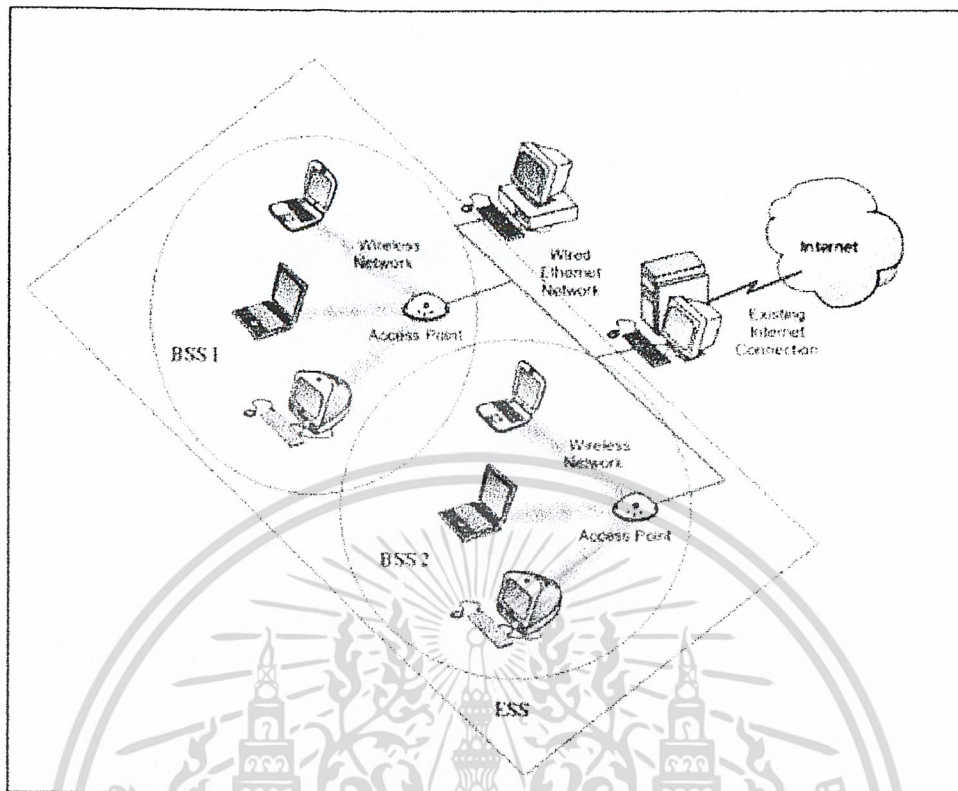
คณะกรรมการชุดนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 ในด้านความปลอดภัย เนื่องจากเครือข่าย IEEE 802.11 WLAN มีช่องโหว่อยู่มากโดยเฉพาะอย่างยิ่งการเข้ารหัสข้อมูล (Encryption) ด้วย key ที่ไม่มีการเปลี่ยนแปลง คณะทำงานชุด IEEE 802.11i จะนำเอาเทคนิคขั้นสูงมาใช้ในการเข้ารหัสข้อมูลด้วย key ที่มีการเปลี่ยนค่าอยู่เสมอและการตรวจสอบผู้ใช้ที่มีความปลอดภัยสูง มาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชันได้ แต่อย่างไรก็ตาม การทำงานของคณะกรรมการชุดนี้ยังไม่แล้วเสร็จในขณะนี้ (พฤษภาคม พ.ศ. 2546)

4.4 ลักษณะการเชื่อมต่อของอุปกรณ์ IEEE 802.11 WLAN

มาตรฐาน IEEE 802.11 ได้กำหนดลักษณะการเชื่อมต่อของอุปกรณ์ภายในเครือข่าย WLAN ไว้ 2 ลักษณะคือ โหมด Infrastructure และ โหมด Ad-Hoc หรือ Peer-to-Peer

4.4.1 โหมด Infrastructure

โดยทั่วไปแล้วอุปกรณ์ในเครือข่าย IEEE 802.11 WLAN จะเชื่อมต่อกันในลักษณะของโหมด Infrastructure ซึ่งเป็นโหมดที่อนุญาตให้อุปกรณ์ภายใน WLAN สามารถเชื่อมต่อกับเครือข่ายอื่นได้ ในโหมด Infrastructure นี้เครือข่าย IEEE 802.11 WLAN จะประกอบไปด้วยอุปกรณ์ 2 ประเภทได้แก่ สถานีผู้ใช้ (Client Station) ซึ่งก็คืออุปกรณ์คอมพิวเตอร์ (Desktop, Laptop, หรือ PDA ต่างๆ) ที่มีอุปกรณ์ Client Adapter เพื่อใช้รับส่งข้อมูลผ่าน IEEE 802.11 WLAN และสถานีแม่ข่าย (Access Point) ซึ่งทำหน้าที่ต่อเชื่อมสถานีผู้ใช้เข้ากับเครือข่ายอื่น (ซึ่งโดยปกติจะเป็นเครือข่าย IEEE 802.3 Ethernet LAN) การทำงานในโหมด Infrastructure มีพื้นฐานมาจากระบบเครือข่ายโทรศัพท์มือถือ กล่าวคือสถานีผู้ใช้จะสามารถรับส่งข้อมูลโดยตรงกับสถานีแม่ข่ายที่ให้บริการแก่สถานีผู้ใช้นั้นอยู่นั้น ส่วนสถานีแม่ข่ายจะทำหน้าที่ส่งต่อ (forward) ข้อมูลที่ได้รับจากสถานีผู้ใช้ไปยังจุดหมายปลายทางหรือส่งต่อข้อมูลที่ได้รับจากเครือข่ายอื่นมายังสถานีผู้ใช้



รูปที่ 4-1 แสดง BSS และ ESS

Basic Service Set (BSS)

Basic Service Set (BSS) หมายถึงบริเวณของเครือข่าย IEEE 802.11 WLAN ที่มีสถานีแม่ข่าย 1 สถานี ซึ่งสถานีผู้ใช้ภายในขอบเขตของ BSS นี้ทุกสถานีจะต้องสื่อสารข้อมูลผ่านสถานีแม่ข่ายดังกล่าวเท่านั้น

Extended Service Set (ESS)

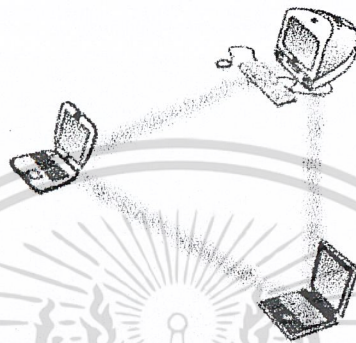
Extended Service Set (ESS) หมายถึงบริเวณของเครือข่าย IEEE 802.11 WLAN ที่ประกอบด้วย BSS มากกว่า 1 BSS ซึ่งได้รับการเชื่อมต่อเข้าด้วยกัน สถานีผู้ใช้สามารถเคลื่อนย้ายจาก BSS หนึ่งไปอยู่ในอีก BSS หนึ่งได้โดย BSS เหล่านี้จะทำการ Roaming หรือติดต่อสื่อสารกันเพื่อทำการ โอนย้ายการให้บริการสำหรับสถานีผู้ใช้ดังกล่าว

4.4.2 โหมด Ad-Hoc หรือ Peer-to-Peer

เครือข่าย IEEE 802.11 WLAN ใน โหมด Ad-Hoc หรือ Peer-to-Peer เป็นเครือข่ายที่ปิดคือไม่มี สถานีแม่ข่ายและ ไม่มีการเชื่อมต่อกับเครือข่ายอื่น บริเวณของเครือข่าย IEEE 802.11 WLAN ในโหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ad-Hoc จะถูกเรียกว่า Independent Basic Service Set (IBSS) ซึ่งสถานีผู้ใช้หนึ่งสามารถติดต่อสื่อสาร ข้อมูลกับสถานีผู้ใช้อื่นๆในเขต IBSS เดียวกันได้โดยตรงโดยไม่ต้องผ่านสถานีแม่ข่าย แต่สถานีผู้ใช้จะ ไม่สามารถรับส่งข้อมูลกับเครือข่ายอื่นๆได้



รูปที่ 4-2 แสดงการทำงานในโหมด Ad hoc หรือ Peer-to-Peer Mode

4.5 การเข้าใช้ช่องสัญญาณด้วยกลไก CSMA/CA

บทบาทหนึ่งของ MAC Layer ในมาตรฐาน IEEE 802.11 คือการจัดการเข้าใช้ช่องสัญญาณ ซึ่งแต่ละสถานีใน BSS หรือ IBSS จะต้องแบ่งกันใช้ช่องสัญญาณที่ถูกกำหนดมาสำหรับใช้งานร่วมกัน อย่างเป็นธรรมชาติ มาตรฐาน IEEE 802.11 ได้กำหนดให้ใช้กลไก CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) เพื่อจัดการการเข้าใช้ช่องสัญญาณร่วมกันดังกล่าว

CSMA with Random Back-Off

กลไก CSMA (Carrier Sense Multiple Access) with Random Back-Off เป็นเทคนิคอย่างง่าย สำหรับจัดการการเข้าใช้ช่องสัญญาณของผู้ใช้แต่ละคน (ซึ่งต้องแบ่งกันใช้ช่องสัญญาณร่วมกัน) อย่าง ยุติธรรม กลไกนี้เป็นที่ยอมรับและนิยมใช้กันอย่างแพร่หลาย เช่น ในมาตรฐาน IEEE 802.3 Ethernet LAN หลักการทำงานของกลไก CSMA คือ เมื่อสถานีหนึ่งต้องการเข้าใช้ช่องสัญญาณ สถานีดังกล่าวจะต้อง ตรวจสอบช่องสัญญาณก่อนว่ามีสถานีอื่นทำการรับส่งสัญญาณข้อมูลอยู่หรือไม่และรอนจนกว่า ช่องสัญญาณจะว่าง เมื่อช่องสัญญาณว่างแล้วสถานีที่ต้องการเข้าใช้ช่องสัญญาณจะต้องรอต่อไปอีกระยะ หนึ่ง (Random Back-Off) ซึ่งแต่ละสถานีได้กำหนดระยะเวลาในการรอนดังกล่าวไว้แล้วด้วยการสุ่มค่า หลังจากเสร็จการเข้าใช้ช่องสัญญาณครั้งก่อน สถานีที่สุ่มได้ค่าระยะเวลาในการรอนน้อยกว่าก็จะมีสิทธิในการ เข้าใช้ช่องสัญญาณก่อน แต่อย่างไรก็ตามในบางกรณีกลไกดังกล่าวอาจจะกำหนดให้สถานีมากกว่าหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

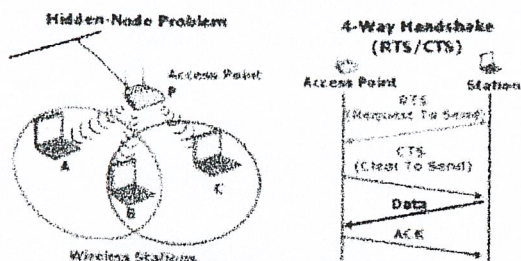
สถานีส่งข้อมูลในเวลาพร้อมๆ กันซึ่งจะทำให้เกิดการชนกันของสัญญาณได้ ซึ่งหากเกิดการชนกันของสัญญาณขึ้นจะต้องมีการส่งสัญญาณข้อมูลเดิมซ้ำอีกครั้งด้วยกลไกที่กล่าวมาแล้วข้างต้น

CSMA/CD

กลไก CSMA/CD (Collision Detection) เป็นเทคนิคที่รู้จักกันดีซึ่งถูกนำมาใช้ในมาตรฐาน IEEE 802.3 Ethernet LAN ซึ่งการทำงานกลไก CSMA/CD โดยหลักแล้วเป็นเช่นเดียวกับที่กล่าวไว้ในส่วนของ CSMA with Random Back-Off แต่จะมีรายละเอียดเพิ่มเติมเกี่ยวกับการตรวจสอบว่าเกิดการชนกันของสัญญาณหรือไม่ ในกรณีนี้สถานีที่กำลังทำการส่งสัญญาณข้อมูลอยู่จะต้องคอยตรวจสอบด้วยว่ามีการชนกันของสัญญาณเกิดขึ้นหรือไม่ (ในขณะที่ชนกันกับที่ทำการส่งสัญญาณข้อมูล) โดยการตรวจวัดระดับ voltage ของสัญญาณในสายสัญญาณว่ามีค่าสูงกว่าปกติหรือไม่ ซึ่งหากระดับ voltage ของสัญญาณในสายสัญญาณในสายสัญญาณมีค่าสูงกว่าค่าที่กำหนดแสดงว่าเกิดการชนกันของสัญญาณขึ้น ในกรณีดังกล่าวสถานีที่กำลังส่งสัญญาณข้อมูลอยู่จะต้องยกเลิกการส่งสัญญาณทันทีและปฏิบัติตามกลไกที่กล่าวมาแล้วข้างต้นเพื่อทำการส่งข้อมูลเดิมซ้ำอีกต่อไป

CSMA/CA with Acknowledgement

เป็นที่ควรสังเกตว่าเทคนิค CSMA/CD ไม่สามารถนำมาใช้กับ WLAN ซึ่งใช้การสื่อสารแบบไร้สายได้ สาเหตุหลักๆ ก็คือการตรวจสอบการชนกันของสัญญาณในระหว่างที่ทำการส่งสัญญาณจะต้องใช้อุปกรณ์รับส่งคลื่นวิทยุที่เป็น Full Duplex (สามารถรับและส่งสัญญาณในเวลาเดียวกันได้) ซึ่งจะมีราคาแพงกว่าอุปกรณ์รับส่งคลื่นวิทยุที่ไม่สามารถรับและส่งสัญญาณในเวลาเดียวกัน นอกจากนี้แต่ละสถานีใน BSS หรือ IBSS อาจไม่ได้ยินสัญญาณจากสถานีอื่นทุกสถานีหรือปัญหาที่เรียกว่า Hidden Node Problem (ดังในรูปที่ 6-3 : สถานี A ได้ยินสัญญาณจากสถานีแม่ข่าย (Access Point) แต่ไม่ได้ยินสัญญาณจากสถานี C และในทางกลับกันสถานี C ไม่ได้ยินสัญญาณจากสถานี A แต่ได้ยินสัญญาณจากสถานีแม่ข่าย ซึ่งสถานการณ์ดังกล่าวนี้เป็นสถานการณ์เกิดขึ้นใน WLAN โดยทั่วไป) ดังนั้นการตรวจสอบการชนกันของสัญญาณโดยตรงเป็นไปได้ยากหรือเป็นไปได้ไม่สะดวก มาตรฐาน IEEE 802.11 จึงได้กำหนดให้ใช้เทคนิค CSMA/CA with Acknowledgement สำหรับการจัดการสรรกรการเข้าใช้ช่องสัญญาณของแต่ละสถานีเพื่อแก้ไขปัญหาเหล่านี้ ซึ่งการทำงานของกลไก CSMA/CA โดยหลักแล้วเป็นเช่นเดียวกับที่กล่าวไว้ในส่วนของ CSMA with Random Back-Off แต่จะมีรายละเอียดเพิ่มเติมเกี่ยวกับการหลีกเลี่ยงไม่ให้เกิดการชนกันของสัญญาณและเทคนิคสำหรับการตรวจสอบว่าเกิดการชนของสัญญาณหรือไม่แบบเป็นนัย โดยสถานีผู้ส่งสัญญาณข้อมูลจะต้องรอรับ Acknowledgement จากสถานีที่ส่งข้อมูลไปให้ หากไม่ได้รับ Acknowledgement กลับมาภายในเวลาที่กำหนดจะถือว่าเกิดการชนของสัญญาณขึ้นและต้องทำการส่งข้อมูลเดิมซ้ำอีกต่อไป



รูปที่ 4-3 แสดง Hidden Node Problem และ กลไก RTS/CTS Handshake

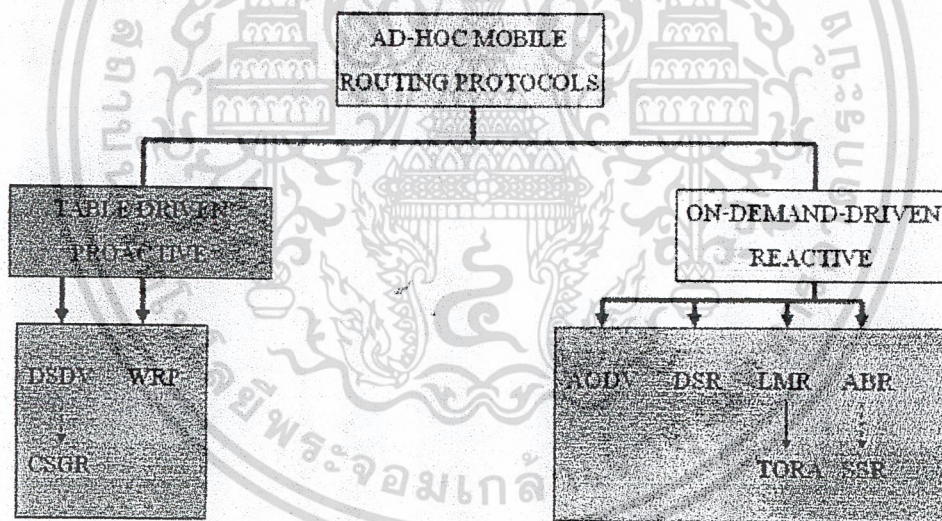
สำหรับการหลีกเลี่ยงไม่ให้เกิดการชนกันของสัญญาณนั้น มาตรฐาน IEEE 802.11 ได้ใช้กลไกที่เรียกว่า Virtual Carrier Sense เพื่อแก้ไขปัญหาที่แต่ละสถานีใน BSS หรือ IBSS อาจไม่ได้ยินสัญญาณจากสถานีอื่นบางสถานี (Hidden Node Problem) กลไกดังกล่าวมีการทำงานดังนี้ เมื่อสถานีที่ต้องการจะส่งแพ็กเก็ตเกิดข้อมูลได้รับสิทธิในการเข้าใช้ช่องสัญญาณแล้วจะทำการส่งแพ็กเก็ตเกิดขึ้นๆ ที่เรียกว่า RTS (Request To Send) เพื่อเป็นการจองช่องสัญญาณ ก่อนที่จะส่งแพ็กเก็ตข้อมูลจริง ซึ่งแพ็กเก็ต RTS ประกอบไปด้วยระยะเวลาที่คาดว่าใช้ช่องสัญญาณจนแล้วเสร็จ (Duration ID) รวมถึง Address ของสถานีผู้ส่งและผู้รับ เมื่อสถานีผู้รับได้ยินสัญญาณ RTS ก็จะตอบรับกลับมาด้วยการส่งสัญญาณ CTS (Clear To Send) ซึ่งจะบ่งบอกข้อมูลระยะเวลาที่คาดว่าสถานีที่กำลังจะทำการส่งข้อมูลนั้นจะใช้ช่องสัญญาณจนแล้วเสร็จ หลักการก็คือทุกๆสถานีใน BSS หรือ IBSS ควรจะได้ยินสัญญาณ RTS หรือไม่ก็ CTS อย่างใดอย่างหนึ่งหรือทั้งสองอย่าง เมื่อได้รับ RTS หรือ CTS ทุกๆสถานีจะทราบถึงว่าช่วงเวลาที่จะบุไว้ใน Duration ID ซึ่งช่องสัญญาณจะถูกใช้และทุกสถานีที่ยังไม่ได้รับสิทธิในการเข้าใช้ช่องสัญญาณจะตั้งค่า NAV (Network Allocation Vector) ให้เท่ากับ Duration ID ซึ่งแสดงถึงช่วงเวลาที่ยังไม่สามารถเข้าใช้ช่องสัญญาณได้ ทุกๆสถานีจะใช้กลไก Virtual Carrier Sense ดังกล่าวผนวกกับการฟังสัญญาณในช่องสัญญาณจริงๆ ในการตรวจสอบว่าช่องสัญญาณว่างอยู่หรือไม่

บทที่ 5

แอดฮอคเราท์ติ้งโพรโตคอล

5.1 ตัวอย่างแอดฮอคเราท์ติ้งโพรโตคอล (Ad hoc Routing Protocols)

ตั้งแต่ DARPA ได้นำเทคโนโลยีเครือข่ายเข้ามาถึงในช่วงต้นปี 1970 มีโพรโตคอลจำนวนมากมาถูกพัฒนาขึ้นสำหรับเครือข่ายแอดฮอค โมบาย [5] ดังนั้น โพรโตคอลต้องจัดการกับข้อจำกัดต่างๆของเครือข่ายแบบนี้ ซึ่งรวมถึงการบริโภคพลังงานที่สูง แบนด์วิดท์ต่ำ และมีอัตราความผิดพลาดที่สูง รูปที่ 5-1 แสดงตัวเราท์ติ้งโพรโตคอลซึ่งถูกจัดกลุ่มโดยทั่วไปดังนี้คือ 1. Table - driven 2. Source initiated on demand-driven เส้นถูกครีทึบในภาพนี้แสดงถึงการสืบต่อโดยตรง ในขณะที่เส้นถูกครีทึบประแสดงถึงการสืบต่อตามความสมควร แม้จะยังคงมีการออกแบบสำหรับเครือข่ายทั้งหมดเป็นชนิดเดียวกัน คุณลักษณะของแต่ละเราท์ติ้งโพรโตคอลค่อนข้างจะแตกต่างกัน ในส่วนนี้จะอธิบายถึง โพรโตคอลและประเภทของโพรโตคอลที่มีคุณลักษณะเข้ากัน



รูปที่ 5-1 ประเภทของแอดฮอคเราท์ติ้งโพรโตคอล

5.2 วิธีแบบ Table-Driven

เราท์ติ้งโพรโตคอลแบบ Table-Driven พยายามที่จะดูแลรักษาความตรงกัน การอัปเดตข้อมูลเราท์ติ้งจากโหนดแต่ละตัวไปยัง โหนดอื่นๆทุกๆ โหนดในเครือข่าย โพรโตคอลเหล่านี้ต้องการให้แต่ละโหนดทำการดูแลรักษาหนึ่งตารางหรือมากกว่าเพื่อที่จะทำการเก็บข้อมูลเราท์ติ้ง และทำการตอบสนองการเปลี่ยนแปลงโท โทโลยีของเครือข่ายโดยการแพร่การอัปเดตเส้นทางตลอดทั้งหมดยของเครือข่าย เพื่อที่จะทำการดูแลรักษาความตรงกันของมุมมองของเครือข่าย พื้นที่โหนดที่มีการติดกันของจำนวนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขที่จำเป็นต้องมีตารางเรทติ้งและแบบแผนที่สัมพันธ์กัน ก็จะทำการเปลี่ยน โครงสร้างของเครือข่าย โดยการบรอดคาสต์

5.2.1 Destination Sequenced Distance Vector (DSDV)

DSDV เป็นเรทติ้ง โพรโตคอลแบบ table-driven ซึ่งตั้งอยู่บนแบบหลักของเรทติ้งเบลแมนฟอร์ด อัลกอริทึม (Bellman-Ford routing algorithm) ซึ่งถูกปรับปรุงให้ดีขึ้น โดยการหลีกเลี่ยงการเกิดลูปบนเรทติ้งเตอร์ของเครือข่ายโมบาย แต่ละ โหนดในเครือข่ายโมบายจะทำการดูแลตารางเรทติ้งในปลายทางทั้งหมดเท่าที่เป็นไปได้ภายในเครือข่ายที่ไม่มีการแบ่งย่อย และจำนวนของเรทติ้งฮอป (ในกรณีนี้เป็นจำนวนของเรดิโอฮอป) ไปจนถึงแต่ละปลายทางที่ถูกบันทึก เนื่องจากเหตุนี้ข้อมูลเรทติ้งจึงพร้อมที่จะนำใช้งานได้อยู่เสมอ โดยไม่คำนึงถึงโหนดต้นทางว่าต้องการเส้นทางหรือไม่

ลำดับของระบบจำนวนตัวเลขที่ถูกนำมาใช้ใน โมบายโฮสต์เพื่อที่จะทำการแบ่งแยกเส้นทางเก่า ออกจากเส้นทางใหม่ การอัปเดตตารางเรทติ้งจะทำการส่งแบบเป็นระยะ ๆ ตลอดทั้งเครือข่ายเพื่อที่จะดูแลตารางให้มีความตรงกัน ดังนั้นจึงก่อให้เกิดกราฟฟิคในการควบคุมเป็นจำนวนมาก ทำให้ปราศจากประสิทธิภาพจากการใช้ประโยชน์ของทรัพยากรบนเครือข่าย เพื่อที่จะระงับปัญหานี้ DSDV จึงใช้การแพ็กเกตอัปเดตเส้นทาง 2 แบบด้วยกัน อันดับแรกคือรู้จักการนำมาเก็บไว้ที่เดียวกัน โดยสมบูรณ์ (full dump) แบบนี้แพ็กเกตจะบรรจุทุกเอาข้อมูลเรทติ้งที่ใช้งานได้ทั้งหมด และยังสามารถกำหนดหน่วยข้อมูล โพรโตคอลของเครือข่ายได้จำนวนมาก (NPDUs) ตลอดระยะเวลาที่เกิดการเคลื่อนไหวเป็นบางครั้งบางคราวแพ็กเกตเหล่านี้มาจครั้งจึงจะสื่อสารส่งถ่ายข้อมูลกัน ทำให้มีส่วนของแพ็กเกตที่เพิ่มขึ้นน้อยกว่าจึงถูกใช้งานการส่งทอดต่อเฉพาะ ข้อมูลที่มีการเปลี่ยนแปลงเท่านั้น

เส้นทางใหม่ที่บรอดคาสต์จะบรรจุแอดเดรสของโหนดปลายทาง จำนวนของฮอปที่ไปจนถึงปลายทาง ลำดับจำนวนของข้อมูลที่ได้รับเข้ามาพิจารณาที่ปลายทาง จำนวนลำดับใหม่จะมีอยู่อันเดียวไม่ซ้ำกับใครเพื่อที่จะทำการบรอดคาสต์ เส้นทางที่ถูกลาเบลโดยลำดับล่าสุดที่เข้ามาจะถูกใช้งานเสมอ ในเหตุการณ์การอัปเดตของทั้ง 2 แบบนั้นจะมีลำดับจำนวนเดียวกัน เส้นทางที่มีจำนวนฮอปเวลาน้อยที่สุดจะถูกนำมาใช้งาน

5.2.2 Wireless Routing Protocol (WRP)

ส่วนที่ใหม่ ของ WRP จากการทำสำเร็จจากดูปเรเนดคอม ในเรทติ้งแบบ WRP ตัวเรทติ้งโหนดจะสื่อสารระยะทางตั้งแต่ฮอปที่สองจนถึงฮอปสุดท้ายภายในข้อมูลสำหรับปลายทางแต่ละตัวภายในเครือข่ายไร้สาย WRP จะเกี่ยวกับคลาสของอัลกอริทึมการหาเส้นทาง นอกเหนือความสำคัญ มันจะทำการหลีกเลี่ยงการเกิดปัญหา count-to-infinity โดยการบังคับให้แต่ละ โหนดทำการตรวจสอบความตรงกันของข้อมูลก่อนหน้าซึ่งจะถูกรายงาน โดย โหนดข้างเคียงของมันเองทั้งหมด สุดท้ายจะกำจัดสถานการณ์การเกิดลูปและจัดเตรียมการบรรจุกันของเส้นทางในกรณีที่เกิดเหตุการณ์ถึงขีดเสียขึ้นด้วยความเร็วที่มีมากกว่า

ใน WRP ตัวโหนดจะทำการเรียนรู้เกี่ยวกับการมีอยู่ของโหนดข้างเคียงจาก Acknowledgments และแมสเสจอื่นๆที่ได้รับมา ถ้าหาก โหนดไม่ได้กำลังทำการส่งแพ็กเกตมันจะต้องทำการส่งแมสเสจ

HELLO อยู่ภายในช่วงระยะเวลาที่ระบุเพื่อที่จะทำให้แน่ใจว่ามีการเชื่อมต่อโดยคำนึงถึงความถูกต้องตรงกันข้ามหากขาดแมสเสจจาก โหนดแสดงว่ามีความล้มเหลวเกิดขึ้นของลิงค์แบบไร้สายและสิ่งนี้เองจะมีการเตือนสิ่งผิดปกติขึ้น เมื่อตัวโมบายได้รับแมสเสจ HELLO จากโหนดใหม่ ข้อมูลของโหนดใหม่นั้นจะถูกเพิ่มเข้าไปในตารางเราท์ติงของ โมบายตัวนั้นๆ และ โมบายจะทำการส่งสำเนาตัวข้อมูลตารางเราท์ติงของตนเองไปให้กับโหนดใหม่นั้นๆ

WRP ต้องทำการดูแลตาราง 4 ตารางด้วยกันคือ 1 ตารางระยะทาง 2 ตารางเราท์ติง 3 ตารางค่า โสหุ่ยของลิงค์ (link-cost) 4 ตารางรายการแมสเสจที่ทำการส่งใหม่ (MRL) ตารางระยะทางนั้นแสดงถึงจำนวนของสอพระหว่างโหนดนั้นๆจนถึงปลายทางของมัน ตารางเราท์ติงแสดงถึงตัวสอพโหนดต่อไป ตาราง โสหุ่ยของลิงค์จะคำนึงถึงค่าตัวเลขที่ส่วนร่วมกับลิงค์โดยเฉพาะ MRL จะบรรจุลำดับจำนวนของแมสเสจการอัปเดตจำนวนนับการส่งข้อมูลใหม่ แฟล็กเวกเตอร์ ACK และรายการของการอัปเดตที่ส่งมาในแมสเสจอัปเดต MRL จะทำการบันทึกการอัปเดตในแมสเสจอัปเดตซึ่งจำเป็นที่จะต้องใช้ในการส่งข้อมูลใหม่และ โหนดข้างเคียงควรตอบว่าได้รับข้อมูลที่ส่งใหม่แล้ว

เพื่อที่จะสร้างความแน่ใจว่าข้อมูลเราท์ติงมีความเที่ยงตรง ตัวโมบายจะทำการส่งแมสเสจอัปเดตเป็นบางครั้งบางคราวให้กับ โหนดข้างเคียงของมัน แมสเสจอัปเดตจะ บรรจุรายการของการอัปเดต (ปลายทาง, ระยะทางจนถึงปลายทาง, ผู้มาก่อนของปลายทาง) เช่นเดียวกับรายการตอบสนองที่แสดงถึง โมบายโหนดว่าควรจะตอบรับการอัปเดต โมบายจะทำการส่งแมสเสจอัปเดตหลังจากดำเนินการ อัปเดตจากโหนดเพื่อนบ้านหรือเมื่อตรวจพบว่าลิงค์มีการเปลี่ยนแปลง ในเหตุการณ์ของลิงค์เกิดล้มเหลวขึ้นมา โหนดจะทำการตรวจจับความล้มเหลวนี้และจะทำการส่งแมสเสจอัปเดต ไปให้กับ โหนดเพื่อนบ้านของมัน และ โหนดเพื่อนบ้านเหล่านั้นจะเข้ามาแก้ไขตารางระยะทางของตนเองและตรวจสอบสำหรับเส้นทางใหม่ที่เป็นไปได้ในการผ่านไปยัง โหนดอื่นๆ

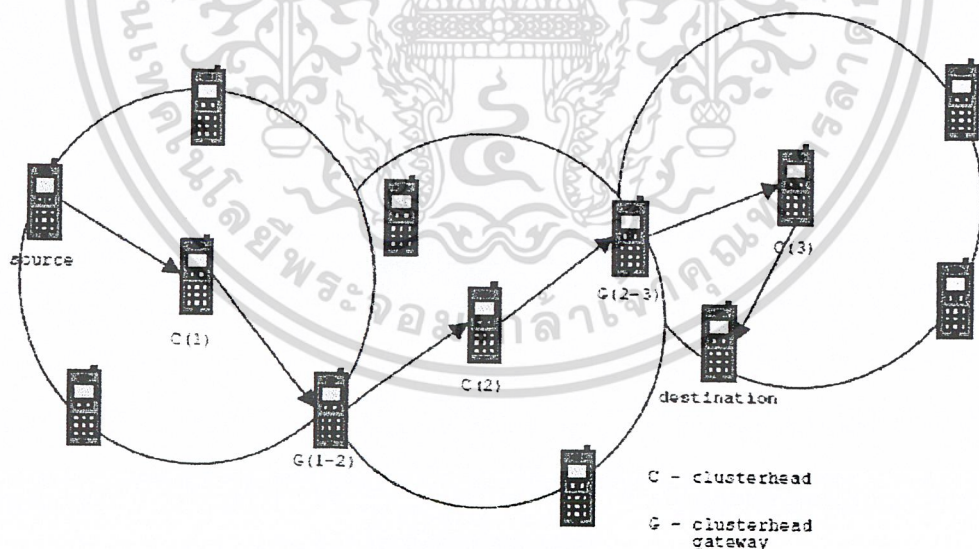
5.2.3 Cluster Switch Gateway Routing (CSGR)

CSGR เป็นเราท์ติง โพร โทคอลที่ตัว โหนด โมบายนั้นจะถูกรวมกันเป็นกลุ่มก้อนและแต่ละก้อนนั้นจะมีหัวหน้าหรือคลัสเตอร์เฮด การรวมกลุ่มนี้ยังแนะนำรูปแบบลำดับชั้นด้วย คลัสเตอร์เฮดสามารถควบคุมกลุ่มของแอดฮอค โสสต์และกลุ่มก้อนนั้นๆมีการจัดเตรียม โครงร่างสำหรับการประมวลที่แยกจากกันตามจำนวนคลัสเตอร์ ช่องสัญญาณการเข้าถึงเราท์ติง และการจองแบนด์วิดท์ การเลือกตัวคลัสเตอร์เฮด จะใช้อัลกอริทึมในการเลือกคลัสเตอร์เฮดซึ่งถูกใช้อย่างแพร่หลาย แม้ว่าการใช้คลัสเตอร์เฮดอนุญาตให้ใช้การควบคุมรูปแบบและการร่วมมือกัน มันทำการกำหนดให้โหนดอื่นๆในกลุ่มก้อนนั้นๆไว้วางใจเมื่อคลัสเตอร์เฮดมีการเคลื่อนไหวนออกไปจะมีการเลือกตัวคลัสเตอร์เฮดใหม่ขึ้นมาปัญหาอาจเกิดขึ้นถ้าหากคลัสเตอร์เฮดมีการเปลี่ยนแปลงบ่อยครั้ง และ โหนดจะใช้เวลาส่วนใหญ่ในการทำให้เกิดการบรรจบที่คลัสเตอร์เฮดมาแทนของ โดยการฟอร์เวิร์ดข้อมูลไปทางปลายทางที่มุ่งหมาย เพื่อที่จะหลีกเลี่ยงการอ้างถึงคลัสเตอร์เฮดการเลือกซ้ำทุกๆครั้งวงสมาชิกของกลุ่มก้อนนั้นๆจะต้องมีการเปลี่ยนแปลง ซึ่งจะมีอัลกอริทึมในการเปลี่ยนกลุ่มก้อน (LLC) ไม่มากนัก การใช้อัลกอริทึม LLC คลัสเตอร์เฮดจะมีการเปลี่ยนแปลง

เฉพาะเมื่อมีคลัสเตอร์เฮด 2 ตัวมาติดต่อกันเท่านั้น หรือเมื่อ โหนดมีการเคลื่อนที่ออกจากคลัสเตอร์เฮดอื่นๆ ทั้งหมด

CSGR จะ ใช้ DSDV ตามแบบแผนเราต์ติง อย่างไรก็ตามมันได้ทำการแก้ไข DSDV โดยการใช้วิธีเราต์ติงลำดับชั้นจากคลัสเตอร์เฮดถึงเกตเวย์ ในการวางแผนทางทราฟฟิกจากต้นทางไปยังปลายทาง โหนดเกตเวย์คือ โหนดที่อยู่ภายในระยะการสื่อสารตั้งแต่ 2 คลัสเตอร์เฮดขึ้นไปดังแสดงในรูปที่ 5-2 แท็กเกตส่งโดยโหนดแรกเราเดินทางไปที่คลัสเตอร์เฮดของตนเอง จากนั้นแแท็กเกตจะถูกเราต์ติงเส้นทางจากคลัสเตอร์เฮดไปหาเกตเวย์เพื่อไปหาคลัสเตอร์เฮดอีกตัวหนึ่งไปจนกระทั่งถึงคลัสเตอร์เฮดของโหนดปลายทาง ดังนั้นแแท็กเกตจึงถูกส่งไปยังปลายทาง

ใน CSGR แต่ละ โหนดต้องทำการเก็บตารางกลุ่มสมาชิก (cluster member table) ซึ่งเป็นที่ใช้เก็บปลายทางของคลัสเตอร์เฮดสำหรับแต่ละ โมบายโฮสต์บนเครือข่าย ตารางกลุ่มสมาชิกเหล่านี้จะทำการบรอดคาสต์เป็นระยะๆผ่านไปตามแต่ละ โหนด โดยใช้โพรโตคอล DSDV โหนดที่ได้รับการอัปเดตนี้จะทำการรีเฟรชตารางกลุ่มสมาชิกของพวกมันเองเพื่อที่จะแน่ใจได้ว่ามีข้อมูลในตารางที่ถูกต้อง นอกจากนี้แล้วที่ตารางกลุ่มสมาชิกนั้น แต่ละ โหนดต้องทำการดูแลตารางเราต์ติงด้วยซึ่งจะถูกใช้ในการกำหนดคอสตต่อไปเพื่อที่จะ ไปให้ถึงปลายทาง ในการรับแแท็กเกตนั้น โหนดจะพิจารณาสมาชิกกลุ่มของตนเองและตารางเราต์ติงเพื่อที่จะกำหนดเส้นทางผ่านคลัสเตอร์เฮดที่ใกล้ที่สุด ไปยังปลายทาง จากนั้น โหนดจะทำการตรวจสอบตารางเราต์ติงของตนเองเพื่อที่จะกำหนดคอสต โหนดต่อไปเพื่อ ไปให้ถึงคลัสเตอร์เฮด โดยรวมแล้วการอัปเดตจำเป็นสำหรับทั้งเราต์ติงและตารางสมาชิกกลุ่มใน CSGR



รูปที่ 5-2 เส้นทาง CSGR ซึ่งจำเป็นต้องผ่านคลัสเตอร์เฮดในเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

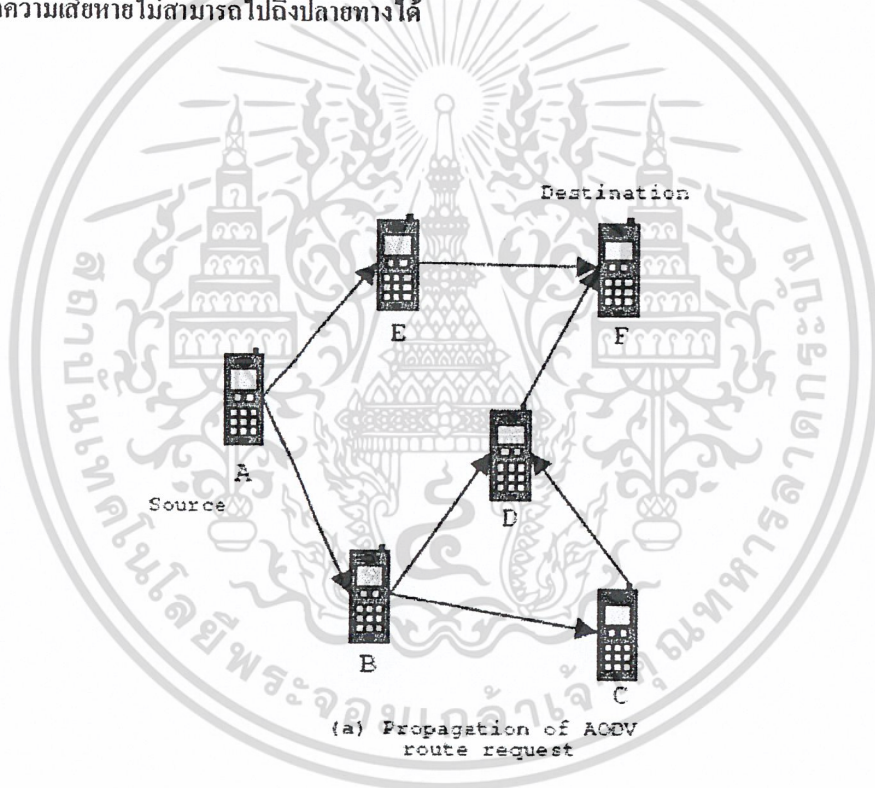
5.3 วิธีแบบ Source-Initiated On-Demand

เป็นความแตกต่างระหว่าง table-driven กับ source-initiated on-demand Routing ซึ่ง routing ชนิดนี้จะสร้างเส้นทางอย่างเดียวนั้นเมื่อ โหนดต้นทางต้องการที่จะติดต่อ โหนดปลายทางหรือต้องการเส้นทางไปยัง โหนดปลายทางชั้นแรกมันจะเริ่มต้นในการค้นหาเส้นทางภายในเครือข่ายจากต้นทางไปยังปลายทางซึ่งการค้นหาเส้นทางจะเสร็จสมบูรณ์ได้ต้องมีเส้นทางที่ค้นหาเจอ 1 เส้นทางหรือค้นหาเส้นทางเจอหลายๆเส้นทางที่เป็นไปได้และมีการตรวจสอบเส้นทางนั้น เส้นทางหนึ่งเส้นทางที่มีการค้นหานั้นจะมีการเชื่อมต่อเส้นทางและจะมีการรักษาเส้นทางนั้นไว้จนกระทั่งต้นทางต้องการที่จะเลิกทำการคิดปลายทางหรือต้นทางไม่ต้องการเส้นทางนั้นแล้ว

5.3.1 Ad Hoc On-Demand Distance Vector Routing (AODV)

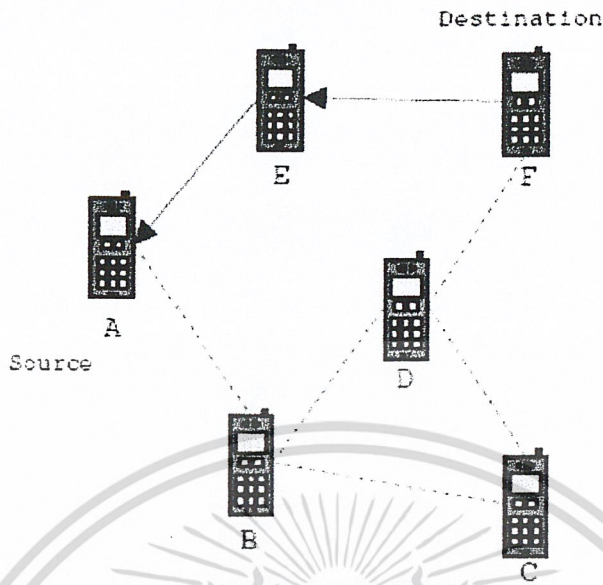
Ad Hoc On-Demand Distance Vector Routing protocol (AODV) ซึ่งก่อนหน้าคือริบาช DSDV algorithm ไว้แล้ว AODV [16][18] จะเป็นการปรับปรุงแก้ไข DSDV เพราะโดยทั่วไปจำนวนน้อยที่สุดของความต้องการที่จะบรอดคาสต์จะอยู่บนพื้นฐานของความต้องการที่จะสร้างเส้นทางซึ่ง DSDV algorithm จะคัดค้านการที่จะให้มีการรักษาเส้นทางนั้นไว้ ผู้สร้าง AODV จะแบ่งตามลักษณะของเส้นทางที่ได้รับมาจากระบบถ้า โหนดที่ไม่ได้รับการเลือกเส้นทางก็จะไม่มีการแลกเปลี่ยน information ในเราที่ดิงเทเบิล เมื่อ source node ต้องการที่จะส่งแพคเกจไปยัง destination node และยังไม่มีความรู้ที่จะไป destination node นั้น จะนั่นเริ่มต้น source node จะทำการค้นหาเส้นทางที่ไป โหนดต่างๆที่ติดกับมันอยู่ซึ่ง source node มันก็จะ broadcasts โดยการส่ง route request (RREQ) packet ไปยังทุก โหนดที่อยู่ข้างเคียงเมื่อ โหนดข้างเคียงได้รับ RREQ packet ก็จะ forward RREQ packet นั้นให้กับ โหนดอื่นๆต่อไปที่อยู่ในเครือข่ายจนกระทั่งถึง โหนดปลายทาง AODV จะใช้ destination sequence number เพื่อให้แน่ใจว่าทุกๆเส้นทางเป็น loop-free และก็จะได้ route information ด้วย แต่ละ โหนดก็จะมี sequence number ของตัวเอง เช่นเดียวกับ broadcast ID broadcast ID จะเพิ่มขึ้นทีละหนึ่งสำหรับทุกๆ RREQ โหนดที่เริ่มต้นและ โหนดก็จะแสดง IP Address ที่ไม่ซ้ำกันพร้อมกันนี้ source node ก็จะรวมเอา sequence number, broadcast ID และ sequence number ปลายทางใส่เข้าไปใน RREQ packet โหนดปลายทางสามารถตอบกลับ RREQ ได้ ถ้ามีเส้นทางที่จะไปถึงปลายทางซึ่งจะต้องสอดคล้องกับ sequence number ปลายทางโดยจะต้องมี sequence number ที่มากกว่าหรือเท่ากับ sequence number ปัจจุบันที่เก็บไว้ใน RREQ ระหว่างที่มีการ forward RREQ นั้น โหนดก็จะบันทึกเส้นทาง address ของ โหนดข้างเคียงไว้ใน route table จากนั้นก็จะมี การ copy packet นั้นและ broadcast packet นั้นไปให้ โหนดข้างเคียง ถ้า โหนดใดได้รับแพคเกจ RREQ ที่เหมือนกันคือได้รับแพคเกจมากกว่า 1 แพคเกจ AODV ก็จะพิจารณาว่าแพคเกจใดมี information route ที่ดีกว่าก็จะเลือกแพคเกจนั้นแล้ว forward ต่อไปส่วนแพคเกจที่เหลือก็จะ discarded ในครั้งหนึ่งๆ RREQ จะมีระยะทางที่ไปถึงปลายทางหรือ โหนดระหว่างกลางที่มีการปรับเปลี่ยนเส้นทาง โหนดปลายทางหรือ โหนดที่ทำการปรับเปลี่ยนเส้นทางนั้นก็จะมีการตอบกลับ โดยใช้ unicasting route reply (RREP) packet ส่งกลับมายัง โหนดข้างเคียง โดยใช้เส้นทางเดียวกับเส้นทางที่ได้รับแพคเกจ RREQ นั้นมา และจะ reverse

packet นั้นตามเส้นทางซึ่งทำการ set up ไว้แล้ว โดยเส้นทางนี้จะอยู่ใน route table ซึ่งโหนดแต่ละจุดก็จะ forward RREP packet ไปตามเส้นทางที่แอกที่พบความสัมพันธ์แต่ละ entry ของเส้นทางคือ route timer ซึ่งสาเหตุที่จะทำการลบ entry ที่ตั้งถ้า RREP packet นั้นไม่มาตามกำหนดเวลาเพราะว่า RREP จะ forward ตามเส้นทางที่สร้างไว้แล้วโดย RREQ จะนั้น AODV จะรองรับการใช้ลิงค์ที่เรียกว่า symmetric link ใน AODV จะมีการรักษาเส้นทางตามเงื่อนไขนี้ ถ้า source node มีการเคลื่อนที่จะมีการเริ่มต้นในการค้นหาเส้นทางใหม่เพื่อไปหาปลายทางถ้าโหนดตามเส้นทางที่จะไปหาปลายทางเคลื่อนที่มันจะแจ้งให้โหนดข้างเคียงทราบว่าจะมีการเคลื่อนที่และส่ง message ไปบอกโหนดข้างเคียงแต่ละโหนดที่แอกที่พอยู่ว่า link failure และจะทำการลบส่วนของเส้นทางนั้นทิ้ง เมื่อโหนดที่แจ้งว่า link failure แก่โหนดข้างเคียงแล้ว โหนดข้างเคียงก็จะประกาศบอกกับ โหนดอื่นๆที่ต่ออยู่ว่าเกิด link failure ที่โหนดนั้นๆจนกระทั่งส่งไปถึง source node เมื่อ source node รั้งบางครั้งก็อาจจะทำการเลือกเส้นทางใหม่สำหรับไปปลายทางถ้าเส้นทางนั้นเกิดความเสียหายไม่สามารถไปถึงปลายทางได้



รูปที่ 5-3 AODV route request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b) Path of AODV route reply

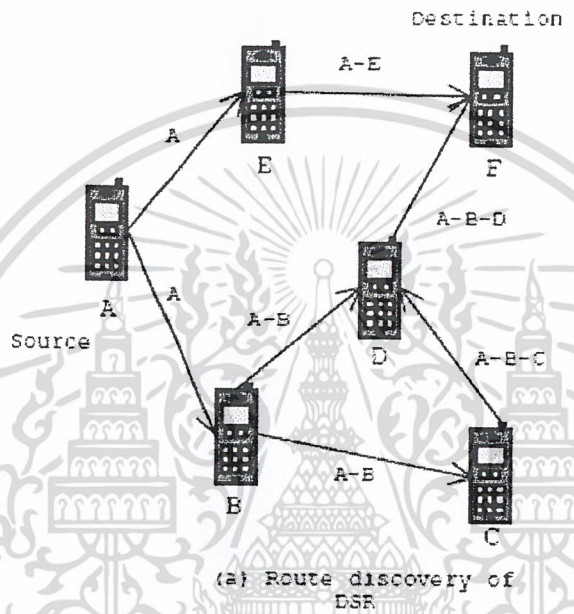
รูปที่ 5-4 AODV route reply

5.3.2 Dynamic Source Routing (DSR)

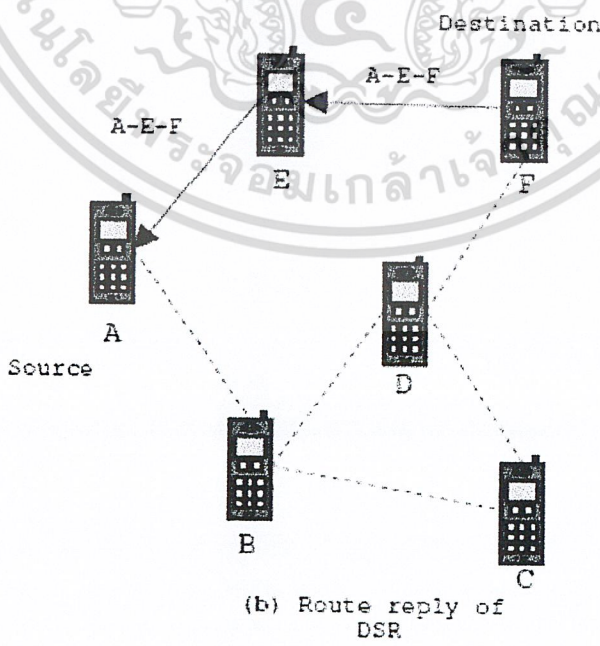
Dynamic Source Routing (DSR) protocol ซึ่งเราที่ตงขนิคนี้เป็รเรที่ตงขแบบ on-demand routing protocol ซึ่งจะเป็นพื้นฐาน concept ของ source routing Mobile โหนดเป็รความต้องการที่จะมีการรักษา route caches ซึ่งจะบรรจุไว้ใน source route ของ mobile ที่ทราบ entry ใน route cache บอขครั้งจะมีการอัปเดตเส้นทางใหม่ที่ได้จากการ learned โพรโตคอลหลักๆที่ใช้ใน phase นี้ประกอบด้วย a) ค้นหาเส้นทาง b) เก็บรักษาเส้นทางนั้น เมื่อ mobile node มีแพ็กเกตที่จะส่งไปปลายทางมันจะปรึกษากับ route cache ก่อนว่ามีเส้นทางไปที่ปลายทางหรือไม่ ถ้ามีเส้นทางที่ไปปลายทางมันก็จะใช้เส้นทางนี้ที่จะส่งแพ็กเกต ถ้าโหนดนั้นยังไม่มีเส้นทางมันก็จะเริ่มต้นในการค้นหาเส้นทางโดยการ broadcast route request packet ไปเพื่อค้นหาเส้นทาง message PREQ นี้จะบรรจุที่อยู่ของปลายทางตามด้วยที่อยู่ของต้นทางและจะมีหมายเลขที่ไม่ซ้ำกัน แต่ละโหนดเมื่อได้รับแพ็กเกตก็จะตรวจสอบว่ารู้จักเส้นทางที่ไปปลายทางหรือไม่ ถ้าไม่รู้มันก็จะเพิ่มที่อยู่ (address) ของตัวเองและเส้นทางนั้นเข้าไปในแพ็กเกตและทำการ forward packet นั้นออกไปตามลิงค์ จำนวนของ route request (RREQ) ที่จะส่งออกไปตามลิงค์ของโหนดนั้นจะมีจำนวนที่จำกัด mobile จะ forward route request อย่างเดียวเท่านั้นถ้ามีการร้องขอจาก mobile และถ้าที่อยู่ (address) ของ mobile นั้นยังไม่ปรากฏใน route record Route reply จะ generate ได้เมื่อเกิดอย่างใดอย่างหนึ่งขึ้นคือเมื่อ route request (RREQ) ส่งไปถึงปลายทางด้วยตัวมันเองหรือเมื่อมันส่งถึงโหนดระหว่างกลาง (intermediate node) ที่จะบรรจุ (RREQ) ไว้ใน route cache ที่มีเส้นทางที่ยังไม่หมดอายุไปถึงปลายทางได้โดยเวลาที่แพ็กเกตส่งถึงปลายทางหรือโหนดระหว่างกลาง (intermediate node) มันก็จะบรรจุเส้นทางที่บันทึกไว้ (route record) และยอมให้มี sequence ของสอพด้วย การสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

route record ขณะที่ route request message ก็จะถูกแพร่ไปยังเครือข่าย ถ้าโหนดที่ทำการผลิต route reply เป็นโหนดปลายทางมันก็จะให้ route record (เส้นทางที่บันทึกไว้) ที่บรรจุใน route request เข้ามาเก็บไว้ใน route reply ถ้าโหนดตอบรับเป็นโหนดระหว่างกลางมันก็จะมีการเพิ่มเติม cache route ให้กับ route record และในเวลานั้นก็จะมีการ generate route reply กลับมา เรื่อง route reply โหนดที่มีการตอบรับจะต้องมีเส้นทางที่เริ่มต้นแล้ว ถ้ามีเส้นทางแล้วใน route cache มันก็อาจใช้เส้นทางในนั้น อีกอย่างหนึ่งก็คือถ้าลิงก์ที่ใช้รองรับแบบ symmetric link โหนดก็อาจจะใช้เส้นทางใน route record ได้ในการใช้เส้นทาง



รูปที่ 5-5 DSR route request

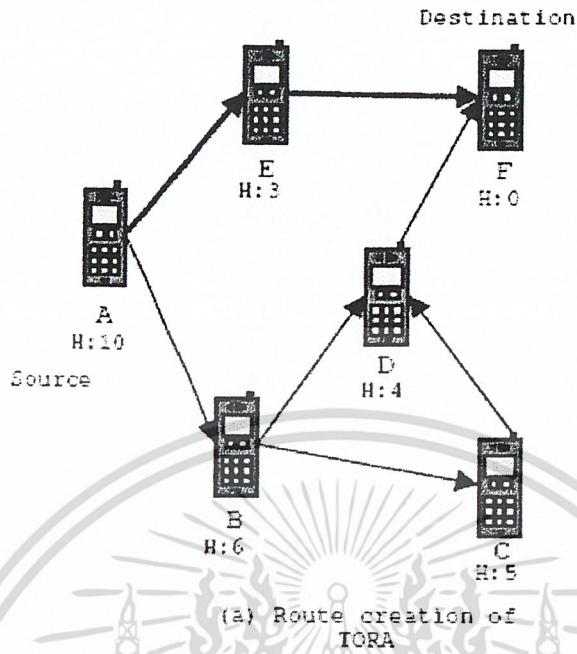


รูปที่ 5-6 DSR route reply

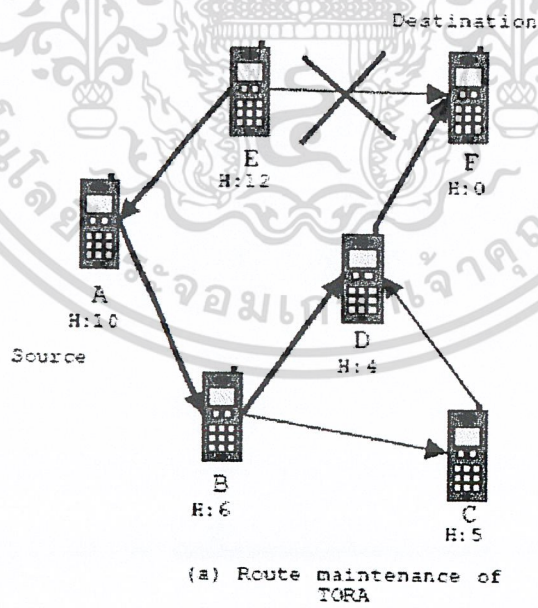
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 Temporally Ordered Routing Algorithm (TORA)

TORA เป็นการตัดแปลงที่ง่ายอย่างยิ่ง loop-free distributed routing algorithm จะอยู่บนพื้นฐานของ link reversal TORA เสนอการปฏิบัติการในสภาพแวดล้อมที่มีเครือข่ายของ mobile มันคือ source-initiated และจัดหาเส้นทางหลายๆเส้นทางสำหรับความต้องการใดๆของ source/destination ที่จับคู่กัน หัวใจสำคัญในการออกแบบ concept ของ TORA คือการที่ควบคุมการกำจัดพื้นที่ในวงที่เล็กๆซึ่งจัดให้มี โหนดอยู่ใกล้ๆกันสิ่งที่เกิดขึ้นคือการเปลี่ยนแปลง ความสามารถนี้ โหนดมีความจำเป็นที่จะต้องรักษา information routing เกี่ยวกับ โหนดที่อยู่ใกล้ๆกัน (one loop) โพรโทคอลจะทำหน้าที่พื้นฐาน 3 อย่างคือ a) สร้างเส้นทาง b) รักษาเส้นทางที่สร้าง c) ลบเส้นทางนั้น ระหว่างที่มีการสร้างเส้นทางและรักษาเส้นทางนั้น โหนดจะวัดความสูงในการสร้าง DAG(Directed Acyclic Graph) จาก root node ไปยัง destination หลังจากนั้นลิงก์ที่กำหนดทิศทางก็จะมีความสัมพันธ์กับความสูงของ โหนดข้างเคียง ซึ่งการสร้างเส้นทางแบบ DAG นี้จะคล้ายกับ query/reply process ที่เสนอใน LMR (Lightweight Mobile Routing) ในเวลาที่ โหนดเคลื่อนที่อยู่ตามเส้นทางที่เป็น DAG ก็จะต้องมีการสร้าง DAG จาก root ไปที่ปลายทางที่เหมือนกันใหม่ การเกิดการ failure ของ link เมื่อ link เกิดความเสียหายขึ้นมาจะทำให้ โหนดมีการกำหนดระดับในการอ้างอิงใหม่ซึ่งผลจากการที่ โหนดกำหนดระดับในการอ้างอิงใหม่นี้ก็จะแพร่ไปยัง โหนดข้างเคียงเป็นผลให้โครงสร้างเกิดการเปลี่ยนแปลงร่วมกัน ลิงก์ก็จะเกิดการเปลี่ยนแปลงที่ reverse สะท้อนกลับเพื่อให้มีความเหมาะสมกับการอ้างอิง level ใหม่และจะมีผลที่เหมือนกันในการกลับทิศทางของลิงก์หนึ่งลิงก์หรือหลายๆลิงก์ การกะเวลาเป็นปัจจัยที่สำคัญของ TORA เพราะว่า การวัดความสูง (height metric) จะขึ้นอยู่กับเวลาของ logical time ของลิงก์ที่เกิดความเสียหาย TORA จะสมมุติฐานว่า ทุกๆ โหนดมีการ synchronized เวลากัน (ซึ่งเป็นความสามารถผ่าน external time เช่น ระบบ GPS) TORA จะมีการวัด 5 ส่วนและ 5 ส่วนนั้นประกอบด้วย a) logical time ของลิงก์ที่เสีย b) หมายเลข ID ของ โหนดที่ประกาศเป็นการระดับอ้างอิงใหม่ (new reference level) ต้องไม่ซ้ำกัน c) bit ที่แสดงการสะท้อนกลับ d) การแพร่จะเป็นไปตามลำดับพหามิเตอร์และ e) ID ของ โหนดต้องไม่ซ้ำ โดยสามส่วนแรกจะเป็นการแสดงถึงระดับการอ้างอิง (reference level) ระดับการอ้างอิงใหม่ (new reference level) จะมีความชัดเจนเมื่อแต่ละเวลาของ โหนดใดๆเกิดการสูญเสียลิงก์อย่างหลีกเลี่ยงไม่ได้ TORA มีการลบเส้นทาง ฉะนั้นก็จะมีจุดสำคัญคือก่อให้เกิดการ flooding แพ็กเก็ตที่ชื่อว่า "clear packet" (CLR) ไปยังตลอดทั้งเครือข่ายเพื่อ บอกว่ามีเส้นทางที่จะลบ



รูปที่ 5-7 TORA route creation



รูปที่ 5-8 TORA route maintenance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 Signal Stability Routing (SSR)

โพรโทคอลแบบ on-demand อีกตัวหนึ่งคือ โพรโทคอล Signal Stability-Base Adaptive Routing (SSR) SSR เป็นโพรโทคอลที่พัฒนามาต่อจาก Associative-Base Routing (ABR) และ ABR เป็นต้นกำเนิดของ SSR ด้วยความที่เหมือนกันกับ ABR ตัว SSR จะทำการเลือกเส้นทางบนสัญญาณที่มีความแรงระหว่างโหนดและตำแหน่งที่ตั้งของโหนดที่เสถียรภาพเนื่องจากเหตุนี้ SSR จึงเสนอความแปลกใหม่เล็กๆน้อยๆเกณฑ์การเลือกเส้นทาง SSR มีผลต่อการเลือกเส้นทางที่มีความแข็งแกร่งของการเชื่อมต่อ SSR สามารถแบ่งออกเป็น 2 โพรโทคอลที่ร่วมมือกันทำงานคือ Dynamic Routing Protocol (DRP) และ Static Routing Protocol (SRP)

DRP รับผิดชอบในการดูแลตารางความเสถียรภาพของสัญญาณ (SST) และตารางเร้าที่ถึง (RT) ตัว SST จะทำการบันทึกความแรงของสัญญาณของโหนดข้างเคียง ซึ่งถูกใช้ได้โดยการส่งสัญญาณเตือนบอกเป็นระยะๆจากลิงค์เดสอรัของโหนดข้างเคียงแต่ละ โหนด ความแรงของสัญญาณอาจถูกบันทึกว่าเป็นช่องสัญญาณแรงหรืออ่อนอย่างใดอย่างหนึ่ง การสื่อสารทั้งหมดจะถูกรับและถูกดำเนินการโดย DRP จะทำการส่งแพ็กเก็ตที่ได้รับไปให้กับ SRP

SRP จะดำเนินการกับแพ็กเก็ตโดยการส่งแพ็กเก็ตขึ้นไปไว้บนเสต็กหากแพ็กเก็ตเหล่านั้นเป็นที่ตั้งใจของผู้รับ หรือถ้าไม่ก็ให้ปลายทางของแพ็กเก็ตเหล่านั้นใน RT จากนั้นจึงทำการฟอร์เวิร์ดแพ็กเก็ตเหล่านั้น หากไม่มีข้อมูลที่พบใน RT สำหรับปลายทาง การดำเนินการหาเส้นทางจะถูกเริ่มต้นขึ้นเพื่อหาเส้นทาง การร้องขอเส้นทางเป็นการแพร่ผ่านเครือข่ายทั้งหมดและไม่ถูกดำเนินการก่อนหน้าเพื่อป้องกันการเกิดลูป ปลายทางจะทำการเลือกแพ็กเก็ตที่มาถึงทั้งหมดเป็นเส้นทางที่ดีที่สุดหรือเป็นเส้นทางที่มีความคับคั่งน้อยที่สุด DRP จะย้อนกลับเส้นทางที่ถูกเลือกและส่งเมสเสจตอบรับเส้นทาง (Route-reply) กลับมาที่ผู้ริเริ่ม DRP ของ โหนดจะมาตามเส้นทางที่อัปเดตตาม RTs ของโหนดนั้นๆ

สันนิษฐานว่า SSR ทำให้การมาถึงของแพ็กเก็ตค้นหาเส้นทาง (route search) ที่ปลายทางจะต้องมีการเลือกเส้นทางที่มีความแรงของสัญญาณสูงที่สุดอย่างสม่ำเสมอ ขณะที่แพ็กเก็ตจะถูกครอบที่โหนดถ้าหากว่าแพ็กเก็ตนั้นมาถึงจากช่องสัญญาณทั้งหมดที่อ่อน หากว่าไม่ได้รับเมสเสจเส้นทางตอบกลับ (route reply) ที่ต้นทางภายในช่วงระยะเวลาที่ระบุที่ต้นทางจะทำการเปลี่ยนเป็นขอบเขต PREF ในเฮลเดอร์ที่บ่งบอกว่ามีช่องสัญญาณอ่อนให้สามารถยอมรับได้ซึ่งอาจจะทำได้เฉพาะถึงค้ทั้งหมดที่แพ็กเก็ตสามารถแพร่ออกไปได้

เมื่อลิงค์เกิดล้มเหลวขึ้นมาจะถูกตรวจจับได้ภายในเครือข่าย โหนดระหว่างกลางสามารถส่งเมสเสจแจ้งความผิดพลาดไปยังฝั่งต้นทางเพื่อบ่งบอกว่าช่องสัญญาณนั้นมีความล้มเหลวเกิดขึ้น ที่ฝั่งต้นทาง จะทำการเริ่มต้นดำเนินการค้นหาเส้นทางอื่นเพื่อหาเส้นทางใหม่เพื่อที่จะ ไปให้ถึงปลายทาง หลังจากนั้น ฝั่งต้นทางจะทำการส่งเมสเสจกลับไปเพื่อแจ้งให้ทุกโหนดว่ามีลิงค์ที่ชำรุด เนื่องจากเหตุนี้เอง DSR และ AODV จึงมีความคล้ายคลึงกัน ที่ฝั่งต้นทางจะถูกขัดจังหวะอยู่เสมอเมื่อมีเส้นทางเกิดการชำรุดซึ่งเป็นเรื่องที่ไม่ดีเลยไม่ได้สำหรับการที่สามารถเคลื่อนที่ได้อยู่ตลอดเวลา ซึ่งไม่เป็นที่พึงปรารถนา

5.3.5 Ad hoc Virtual Switching Routing Protocol (AVSR)

AVSR ประกอบด้วยกระบวนการ route discovery process ซึ่งสามารถลดกราฟที่สร้างขึ้นโดย routing algorithm ซึ่ง AVSR พัฒนามาจาก DSR เพื่อช่วยให้ออกแบบกระบวนการทำงานให้ง่ายขึ้น การจัดการ switching map ในแต่ละ โหนดและกระบวนการพื้นฐาน ในการ routing ซึ่งจะ ได้อธิบายดังต่อไปนี้

Route Discovery Process กระบวนการนี้ประกอบด้วย 2 กระบวนการ

1. ทำการกระจาย Request packet จาก โหนดต้นทาง ไปยัง โหนดปลายทาง
 2. ส่ง Reply packet จาก โหนดปลายทาง ไปยัง โหนดต้นทาง
- เมื่อ โหนดต้นทาง Broadcast request packet ไปแล้ว โหนดที่ได้รับ route request จะมี

กระบวนการตอบสนองได้ 3 กระบวนการ

- 1) ถ้า โหนดนั้นเป็น โหนดปลายทางให้ตรวจสอบเวลาก่อนหน้า ว่าเคยได้รับ Packet นี้มาก่อนหรือไม่ถ้าใช่ก็จะ ไม่สนใจ packet นั้นถ้าไม่เคยได้รับก็จะ ส่ง reply packet เพื่อตอบสนอง route request
- 2) ถ้า โหนดนั้นไม่ใช่โหนดปลายทางและ Packet ที่ได้รับไม่เคยผ่านโหนดนี้มาก่อนก็จะ ทำการส่งต่อ packet
- 3) โหนดนั้น ไม่ใช่โหนดปลายทางและ Packet ที่ได้รับเคยผ่านโหนดนี้มาแล้วก็จะ ไม่สนใจ packet นี้

ในการส่ง Route reply packet ไปยังต้นทาง DSR อนุญาตให้โหนดปลายทางส่งผ่าน reversal route ซึ่งอยู่ใน request header

Incoming Data Packet Processing หลังจาก โหนดได้รับ data packet ก็จะทำ การ update timestamp ของ route ที่ถูกใช้กราฟฟิคในทางตรงกันข้าม โหนดจะตรวจสอบว่ามันเป็นโหนดปลายทางหรือไม่ ถ้ามันจะยอมรับแพ็คเกจ ถ้าไม่ใช่จะตรวจสอบว่า route ที่เกี่ยวข้องกับ packet ทั้งหมดยังคงใช้ได้หรือไม่ถ้าใช่ก็จะ ส่งต่อแพ็คเกจ ถ้าไม่ใช่ก็จะ ไม่สนใจแพ็คเกจนั้น

5.4 ปัจจัยที่ต้องพิจารณาในการเลือกเราติงอัลกอริทึม

1. ช่องสัญญาณและ โมเดลของสัญญาณวิทยุ (Channel and Model)
2. แมค โพร โคคอด (Medium access control protocol)
3. รูปแบบของกราฟฟิค (Traffic pattern)
4. รูปแบบของการเคลื่อนที่ (Mobility pattern)
5. ค่าเมตริกซ์ (Metrics)
 - อัตราส่วนของการส่งแพ็คเกจ
 - จำนวนสอเพเลท
 - จำนวนแพ็คเกจข้อมูลที่ได้ติดต่อสื่อสารต่อจำนวนแพ็คเกจข้อมูลที่ถูกส่ง
 - จำนวนของไบต์ควบคุมต่อที่ได้ติดต่อสื่อสารต่อ ไบต์ข้อมูลที่ถูกส่ง
 - จำนวนของแพ็คเกจควบคุมและ ข้อมูลที่ได้ติดต่อสื่อสารต่อแพ็คเกจข้อมูลที่ถูกส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโตคอลแต่ละตัวจะมีความสามารถที่แตกต่างกันตามแต่สถานการณ์ ซึ่งโพรโตคอลทั้งหมด จะทำงานได้ดีหากมีโมเดลการเคลื่อนไหวเป็นกลุ่ม มากกว่าโมเดลการเคลื่อนไหวแบบมีเส้นทางที่เรคคอม โพรโตคอลแบบ On-Demand จะมีประสิทธิภาพที่สูงที่สุดคือจะมีทราฟฟิกควบคุมที่น้อยกว่าแบบ Table-Driven เราว่าตารางเราที่ถึงไม่ต้องแลกเปลี่ยนเป็นระยะๆซึ่งจะสร้างแพ็กเก็ตควบคุมก็ต่อเมื่อมีความจำเป็นเท่านั้นซึ่งนำไปใช้ในเหตุการณ์ที่มีการเคลื่อนไหวสูงได้แต่ว่าสจะมีดีเลย์เพิ่มขึ้นซึ่งไม่เหมาะกับทราฟฟิกที่ต้องส่งแบบเรียลไทม์ แต่โพรโตคอลดิสแทนซ์เวกเตอร์แบบ Table-driven จะเป็นทางเลือกที่ดีเมื่อมีการส่งข้อมูลแบบเรียลไทม์และทราฟฟิกที่ค่อนข้างมากอย่างไรก็ตามหากมีสเกลขนาดใหญ่ขึ้น และมีการเคลื่อนไหวค่อนข้างสูงจะมีปัญหาในเรื่องโอเวอร์เฮดควบคุมที่มากเกินไป การบรรจบของเส้นทางที่ช้า การเกิดปัญหา count-to-infinity ซึ่งเราที่ถึงโพรโตคอล WRP ก็ได้ทำการแก้ไขปัญหาดังกล่าว ซึ่งจะทำงานได้ดีเมื่อ โหนดอยู่ใน โหนดอยู่ในรูปแบบและมีการเคลื่อนไหวเป็นกลุ่ม

โดยสรุปแล้ว ไม่มีเราที่ถึงโพรโตคอลตัวใดตัวหนึ่งที่ดีที่สุดสำหรับทุกๆสถานการณ์ในระบบเครือข่ายทุกๆ โพรโตคอลมีข้อดีและข้อเสียในสถานการณ์ที่แตกต่างกันไป ดังนั้นการเลือกรเราที่ถึงโพรโตคอลนั้นควรพิจารณาด้วยความรอบคอบ โดยอาศัยปัจจัยต่างๆที่กล่าวในข้างต้นเป็นตัวตัดสินใจ



บทที่ 6

หลักการพื้นฐานของการสร้างโปรแกรมจำลองระบบ

6.1 ความเป็นมาของโปรแกรมจำลองระบบ (Simulation)

เมื่อเราได้ตั้งสมมุติฐานขึ้นมาแล้วเราต้องการทราบผลที่เกิดขึ้นเราและการทำงานเราจำเป็นต้องสร้างแบบจำลองขึ้นมาเพื่อพิสูจน์สมมุติฐานดังกล่าว ซึ่งวิธีหนึ่งที่ทำให้ได้มาซึ่งผลการพิสูจน์สมมุติฐานคือ การทำ simulation คือการใช้คอมพิวเตอร์ในการประเมินแบบจำลองและข้อมูลที่รวบรวมไว้ในแบบจำลองเพื่อนำมาวิเคราะห์คุณสมบัติที่แท้จริงของแบบจำลอง การสร้างแบบจำลองคือกระบวนการในการออกแบบ สร้างตัวแทนของระบบจริงขึ้นมา แล้วทำการทดลองต่างๆ ต่อแบบจำลองนี้ โดยมีวัตถุประสงค์เพื่อทำความเข้าใจต่อพฤติกรรมของระบบ ทำการวิเคราะห์ ค้นหา แนวทางหรือทางเลือกแบบต่างๆ สำหรับการดำเนินการกับระบบนั้นๆ

การสร้างแบบจำลองเป็นเครื่องมือวิเคราะห์ที่มีประสิทธิภาพที่สุดตัวหนึ่ง ใช้สำหรับการออกแบบ วางแผนและดำเนินการควบคุมระบบหรือกระบวนการที่มีความซับซ้อน ในปัจจุบันการสร้างแบบจำลองถูกมองว่าเป็นวิธีการแก้ปัญหาแบบหนึ่งที่สำคัญสำหรับวิศวกร นักออกแบบ และผู้บริหารสำหรับการผลิต [4][7] การสร้างแบบจำลองสามารถทำนายพฤติกรรมของระบบการผลิตที่ซับซ้อนได้โดยคำนวณการเคลื่อนที่และความสัมพันธ์ต่างๆ ที่มีต่อกันของแต่ละองค์ประกอบในระบบ

การนำ simulation ไปใช้นั้นมีได้หลากหลาย ดังเช่นตัวอย่างข้างล่างเป็นการนำ simulation มาใช้เพื่อการแก้ปัญหาต่าง เพื่อให้ได้ประโยชน์และมีประสิทธิภาพ

- การออกแบบและวิเคราะห์ระบบโรงงาน
- การประเมินความต้องการในด้าน hard และ software ของระบบคอมพิวเตอร์
- การประเมินอาวุธทางการทหาร หรือแทลติก
- การตัดสินใจใช้เงิน โฆษณาในการจัดการกับระบบจัดเก็บสินค้า
- ออกแบบระบบติดต่อสื่อสารและการใช้โปร โทคอล
- ออกแบบการขนส่งสินค้า
- ประเมินการออกแบบในการให้บริการในหน่วยงานต่างๆ
- วิเคราะห์ระบบการเงินและเศรษฐกิจ
-

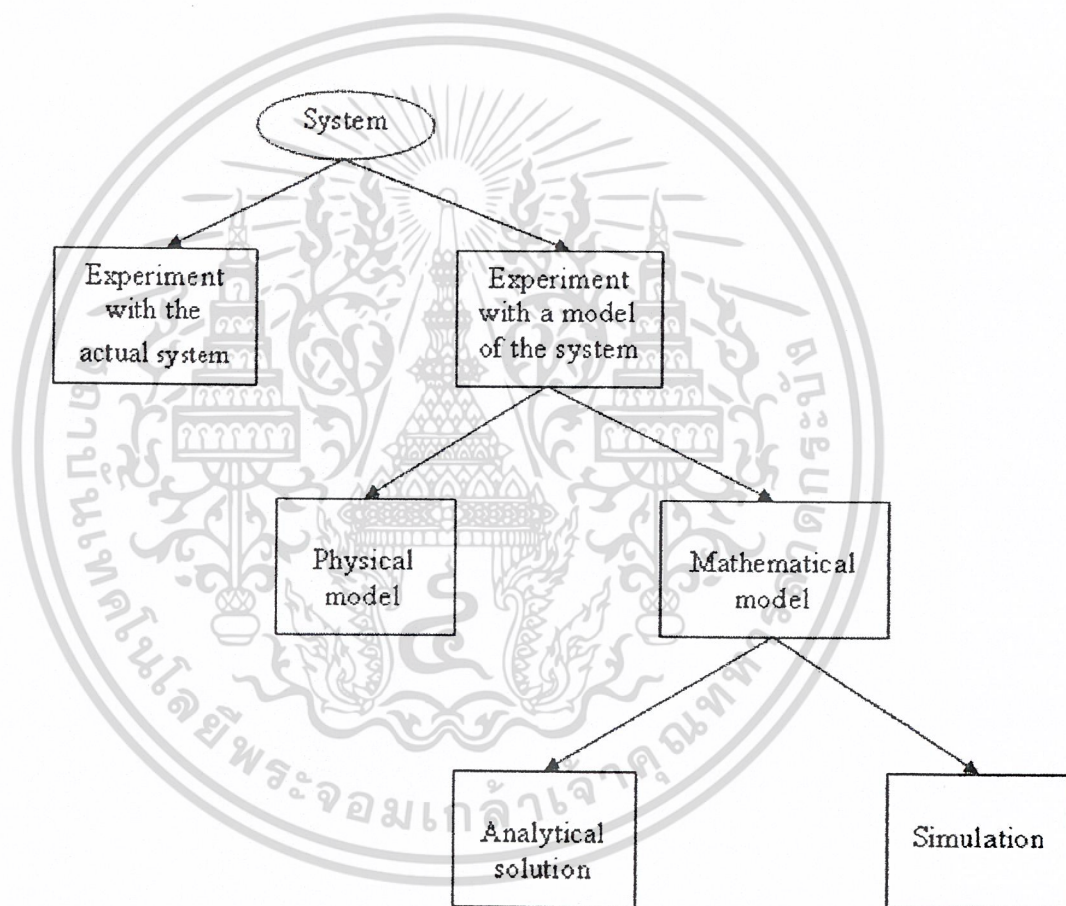
6.2 System Model and Simulation

System คือ สิ่งที่มีอยู่จริงและเราสนใจที่จะศึกษา เช่น คน หรือเครื่องจักรที่กระทำและคิดต่อเพื่อทำการอย่างใดอย่างหนึ่ง ในทางปฏิบัติสิ่งที่เกี่ยวข้องกับระบบขึ้นอยู่กับจุดมุ่งหมายและสิ่งที่เราสนใจศึกษา ตัวอย่างเช่น ถ้าเราต้องการจะศึกษาเกี่ยวกับธนาคารเพื่อตัดสินใจเกี่ยวกับความจำเป็นและจำนวนใน

การใช้พนักงานธนาคารในการช่วยบริการลูกค้าให้พอเพียง ซึ่งในระบบนี้จะประกอบด้วยพนักงานธนาคารและลูกค้าที่รอคิวบริการอยู่หรือลูกค้าที่กำลังเข้ารับการบริการอยู่

สถานะของระบบคือ ช่วงเวลาที่บอกว่าในช่วงนั้นๆ ระบบกำลังทำอะไรอยู่ ความสัมพันธ์กับจุดมุ่งหมายในการศึกษา ในการศึกษาระบบธนาคาร ตัวอย่างที่เป็นไปได้ของสถานะของระบบคือ จำนวนของพนักงานที่ให้บริการอยู่, จำนวนของลูกค้าในธนาคาร, เวลาในการมาถึงของลูกค้าในธนาคาร

เราแบ่งระบบออกเป็นสองชนิด คือ discrete และ continuous ซึ่งรายละเอียดจะได้กล่าวถึงในภายหลัง และบางจุดในระบบส่วนใหญ่ เรามักมีความจำเป็นที่จะต้องศึกษามันถึงความสัมพันธ์ระหว่างส่วนประกอบต่างๆ หรือทำนายการทำงานภายใต้เงื่อนไขใหม่



รูปที่ 6-1 แนวทางการศึกษาระบบ

Experiment with the Actual System VS. Experiment with a Model of the System

เป็นการเปรียบเทียบระหว่างการทดลองด้วยการศึกษาระบบจริงกับการทดลองด้วยแบบจำลองของระบบ ในการที่เราจะทดลองทำตามเงื่อนไขที่เราต้องการนั้นในความเป็นจริงมันเป็นไปได้ยาก เพราะว่าการทดลองกับระบบจริงนั้นจะทำให้สิ้นเปลืองและมีผลกระทบต่อระบบที่เราต้องการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ธนาคารต้องการที่จะลดพนักงานเพื่อลดความสิ้นเปลืองแต่การที่เราทำเช่นนี้กับระบบจริงจะทำให้ลูกค้าต้องรอนานทำให้กระทบกระเทือนกับระบบจริงได้ ด้วยเหตุนี้เราจึงมีการทำแบบจำลองขึ้นมาเพื่อทำการศึกษาความเป็นไปของระบบดูก่อนเพื่อตรวจสอบจุดประสงค์และความถูกต้อง

Physical Model VS. Mathematical Model

ในด้านของ physical model นั้นขนาดของแบบจำลองจะจำลองมาจากของจริงที่ใช้การไม่ได้จริงหรือการต่อชิ้นส่วนที่ไม่ครบวงจร แต่จะมีขนาดเท่าของจริงส่วนทางด้าน mathematical model จะทำการจำลองขึ้นมาไม่ใช่นำของจริงมาทดลอง ซึ่งเป็นหมายเหตุทดสอบด้านหลักการและเหตุผล การได้ตอบ และการทดลองที่ต้องใช้ปริมาณมากๆ ซึ่งของจริงไม่สามารถทำได้หรือถ้าทำได้ก็จะเป็นการสิ้นเปลืองมาก

Analytical Solution VS. Simulation

ในการที่เราจะทำ mathematical model นั้น เราต้องคิดก่อนว่าเราต้องการแบบไหนถ้าต้องการแบบง่าย หรือพิชิตสูตรพื้นฐานอย่างง่ายไม่ซับซ้อนนั้นเราสามารถใช่ แบบ analytical solution ได้เลย แต่ถ้าเราต้องการทดลองที่ยุ่งยากซับซ้อนมีสูตรการคำนวณที่หลากหลาย เราต้องใช้ทรัพยากรทางด้านคอมพิวเตอร์เข้ามาช่วยซึ่งวิธีการนี้ก็คือการทำ simulation นั่นเอง

เราได้แบ่ง simulation ตามจุดประสงค์การใช้งานออกเป็น 3 แบบ

Static VS. Dynamic Simulation Models

Static Simulation Model เป็นการแสดงระบบในช่วงเวลาที่เฉพาะเจาะจง ในทางตรงกันข้าม Dynamic Simulation Models เป็นการแสดงระบบที่เวลาเดินไปเรื่อยๆ

Deterministic VS. Stochastic Simulation Models

Deterministic Simulation Models เป็นแบบจำลองที่ประกอบด้วยชิ้นส่วนที่ไม่สามารถคาดการณ์ล่วงหน้าได้ ซึ่งจะให้ผลลัพธ์ออกมาตามเซตของ input ซึ่งได้ค่าที่เฉพาะเจาะจง ส่วน Stochastic Simulation Models นั้นเป็นแบบจำลองที่มีการจัดลำดับของ input แต่จะ ให้ผลลัพธ์ออกมาแบบสุ่ม

Continuous VS. Discrete Simulation Models

ดังจะ ได้กล่าวในส่วนต่อไป

6.3 Discrete-event simulation

Discrete-event simulation เกี่ยวกับการแสดงระบบในลักษณะของการเปลี่ยนแปลงด้วยการแสดงแทนด้วย state variable ซึ่งจะเปลี่ยนแปลงแบบฉับพลัน ณ เวลาที่แตกต่างกัน (หรืออาจกล่าวได้ว่าระบบจะเปลี่ยนแปลง ณ เวลาใดๆ ซึ่งจำนวนเวลานี้สามารถนับได้) ณ เวลาใดๆซึ่งเป็นเวลาเดียวกับที่เกิดเหตุการณ์ขึ้น เหตุการณ์จะถูกกำหนดให้เป็นการเกิดขึ้นอย่างฉับพลันที่จะเปลี่ยนแปลง state ของระบบ และถึงแม้ว่า Discrete-event simulation สามารถที่จะคำนวณเหตุการณ์ลงบนกระดาษได้ แต่ด้วยจำนวนข้อมูลและการจัดการที่มีมากมายในระบบที่ถูกใช้งานอยู่จริง จึงเป็นการสรุปได้ว่า Discrete-event simulation ควรที่จะดำเนินการด้วยคอมพิวเตอร์จะเป็นการดีกว่า

6.4 Time-Advance Mechanisms

เพราะ Discrete-event simulation มีการเปลี่ยนแปลงแบบ Dynamic จึงต้องมีการกำหนดค่าของเวลาที่ simulation ได้ดำเนินการไปแล้วและเราก็ต้องการวิธีการเพื่อเพิ่มค่าของเวลาจากค่าหนึ่งไปเป็นค่าอื่น เราเรียกค่าของตัวแปรใน simulation model ที่เก็บค่าปัจจุบันของเวลาของการทำ simulation ว่า simulation clock หน่วยของ simulation clock ไม่มีความแตกต่างกันแต่อย่างใดไม่ว่าจะเขียนด้วยภาษา FORTRAN, Pascal, C หรือภาษาอื่นๆ เพราะมันถูกมองว่าเป็นค่าของตัวแปรที่มีหน่วยเหมือนกัน ในทำนองเดียวกันเวลาใน simulation และเวลาที่ใช้ในการ run simulation ไม่มีความเกี่ยวข้องกันแต่อย่างใด

ในความคิดที่ผ่านมามีวิธีการสองวิธีที่ได้รับการแนะนำให้ใช้ในการเพิ่ม simulation clock คือ next-event time advance และ fixed-increment time advance โดยวิธีการแรกถูกใช้งานอย่างแพร่หลายในภาษาที่ใช้ในการเขียน simulation และนักพัฒนาที่ทำการสร้าง simulation ของตัวเองด้วยภาษาคอมพิวเตอร์ต่างๆไป ส่วนวิธีการที่สองเป็น special case ของวิธีการแรก

วิธีการ next-event time advance จะกำหนดค่าเริ่มต้นของ simulation clock เป็นศูนย์และทำการกำหนดเวลาที่将会เกิดเหตุการณ์ในอนาคต simulation clock จะเพิ่มเวลาเมื่อเหตุการณ์ในอนาคตที่กำหนดไว้เกิดขึ้นและระบบจะทำการเปลี่ยน state ของตัวเองให้ตรงกับเหตุการณ์ที่เกิดขึ้นแล้วทำการกำหนดเวลาของเหตุการณ์ที่จะเกิดขึ้นในอนาคตใหม่ เช่นนี้เวลาของ simulation clock ก็จะได้รับปรับปรุงให้เพิ่มขึ้น state ของระบบก็จะได้รับการเปลี่ยนแปลงให้เหมาะสม และเหตุการณ์ที่กำลังจะเกิดในอนาคตก็จะถูกกำหนด กระบวนการนี้จะดำเนินไปจนกว่าจะมีเงื่อนไขให้จบการทำงาน

6.5 Components and Organization of a Discrete-Event Simulation Model

แม้ว่า Simulation จะถูกใช้งานในระบบที่หลากหลาย แต่ Discrete-event simulation มีส่วนประกอบต่างๆที่เหมือนกันในแทบทุก Model ซึ่ง โครงสร้างทางตรรกะของส่วนประกอบเหล่านี้จะเป็นแนวทางในการเขียน โปรแกรมต่อไป ส่วนประกอบดังต่อไปนี้พบได้ทั่วไปใน Discrete-event simulation model ที่ใช้ next-event time advance

System state: เป็นกลุ่มของ state variable ที่ใช้อธิบายความเป็นไปของระบบ ณ เวลาต่างๆ

Simulation clock: เป็นตัวแปรที่ใช้เก็บค่าเวลาของ simulation

Event list: เป็นลิสต์ที่ใช้เก็บค่าเวลาและชนิดของเหตุการณ์ที่จะเกิดในอนาคต

Statistical counter: เป็นตัวแปรที่ใช้เก็บค่าที่จำเป็นต่อการคำนวณประสิทธิภาพของระบบ

Initialization routine: เป็น โปรแกรมย่อยที่จะกำหนดค่าเริ่มต้นของ simulation

Timing routine: เป็น โปรแกรมย่อยที่ทำการกำหนดเวลาของเหตุการณ์และเพิ่มค่า simulation clock

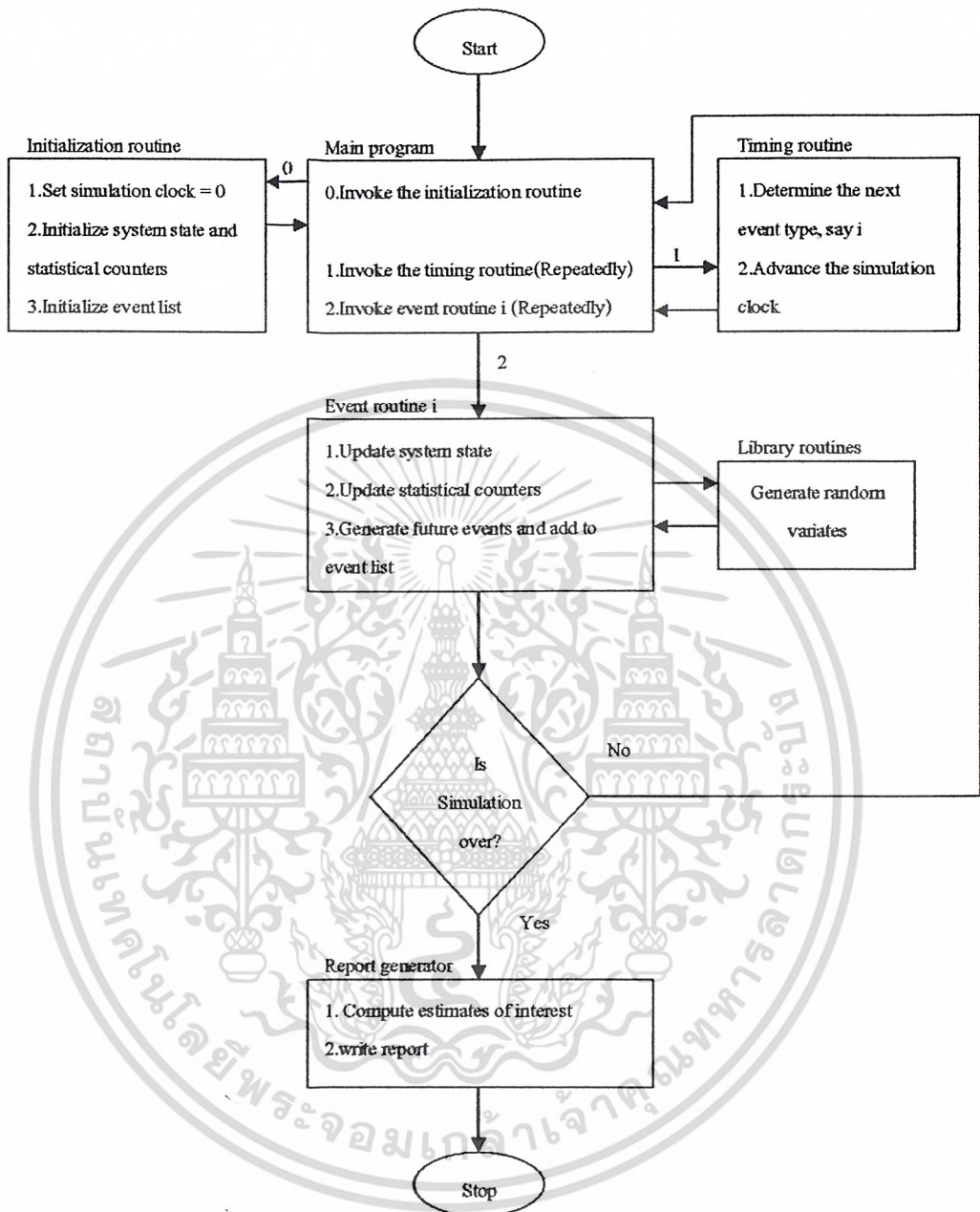
Event routine: เป็น โปรแกรมย่อยที่ทำการเปลี่ยนแปลง state ของระบบเมื่อเกิดเหตุการณ์ขึ้น

Library routines: เป็นกลุ่มของ โปรแกรมย่อยที่ทำการสร้างค่าสุ่มต่างๆตาม Probability distributions

Report generator: เป็น โปรแกรมย่อยที่ทำการคำนวณประสิทธิภาพของระบบและสร้างรายงาน

Main program: เป็น โปรแกรมย่อยที่ทำการควบคุมการทำงานของ simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 Flow of control for next-event time-advance approach

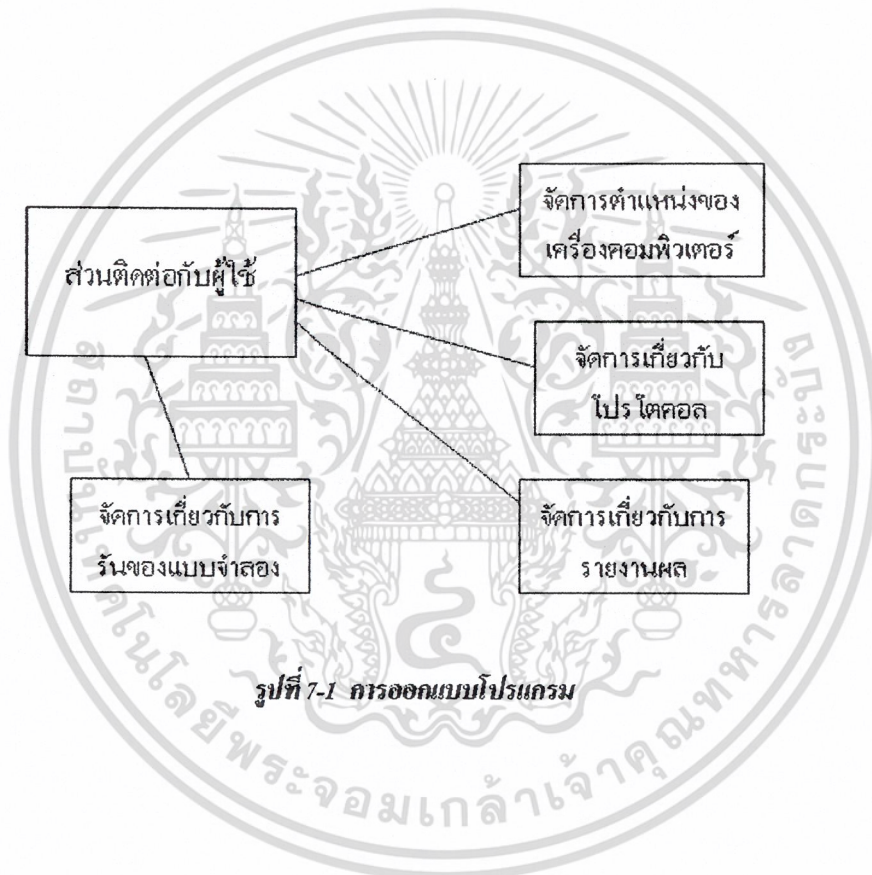
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

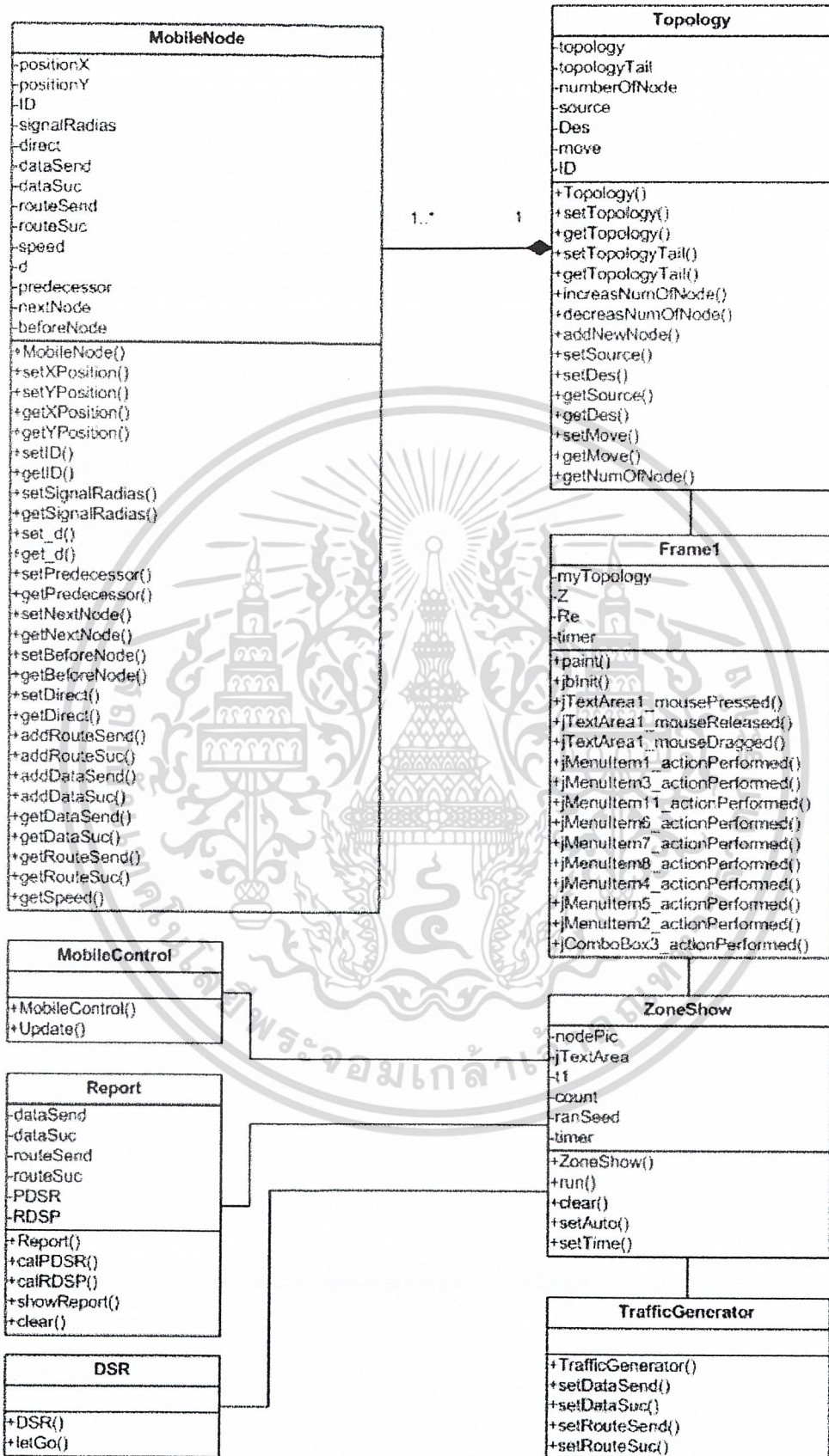
หลักการสร้างและออกแบบโปรแกรม

7.1 โครงสร้างของโปรแกรม (Design)

โปรแกรมนี้ถูกออกแบบและสร้างโดยใช้ภาษา JAVA [1] [2] [3] โดยแบ่งการทำงานของโปรแกรมออกเป็นส่วนย่อยๆ เป็นคลาสต่างๆ แต่ละคลาสจะมีหน้าที่การทำงานที่แตกต่างกันโดยคลาสและฟังก์ชันต่างๆ สามารถแบ่งตามการจัดการได้ดังรูปที่ 7-1 และสามารถแบ่งเป็นคลาสได้ดังรูปที่ 7-2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-2 Class Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การจัดการส่วนติดต่อผู้ใช้ (Graphic User Interface: GUI)

แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

1. หน้าจอการทำงานหลัก ในส่วนนี้ประกอบด้วย ทูลบาร์ ส่วนของหน้าจอแสดงแผนภาพของเครื่องข่าย และส่วนของการสร้างเครื่องข่าย โดยที่ผู้ใช้สามารถที่จะใช้งานได้ดังต่อไปนี้

ส่วนของทูลบาร์ ประกอบด้วย

File เป็นส่วนที่ใช้จัดการเกี่ยวกับไฟล์ เช่น New, Run Auto, Run Manual, Pause, Stop, Resume, Exit เป็นต้น

Edit เป็นส่วนที่ใช้ในการแก้ไข เช่น Set Radius เป็นต้น

Help เป็นส่วนที่ใช้ในการเปิดคู่มือการใช้งานของโปรแกรม

ส่วนของแถบเครื่องมือที่ใช้ในการสร้างแบบจำลอง

สัญลักษณ์รูปลูกศร ใช้ในการเคลื่อนย้ายโหนดที่ได้ทำการวางไปแล้ว

สัญลักษณ์รูปตัว i ใช้ในการวางโหนดซึ่งแทนเครื่องคอมพิวเตอร์ในแบบจำลอง

สัญลักษณ์รูป— ใช้ในการกำหนดการเชื่อมต่อของโหนดที่ต้องการศึกษา ซึ่งเราสามารถที่จะ

กำหนดการเชื่อมต่อได้พร้อมกันที่ละหลายโหนด

Protocol หมายถึง การเลือกโปรโตคอลที่เราสนใจจะศึกษาการทำงาน ซึ่งในที่นี้จะมีด้วยการ 3

โปรโตคอล คือ ดีเอสอา, ดีเอสดีวี, เอ ไอซีวี

Area หมายถึง ขนาดของพื้นที่ที่เราสนใจจะศึกษาการทำงานของแบบจำลอง

Time หมายถึง เวลาที่เราจะทำการรันแบบจำลอง

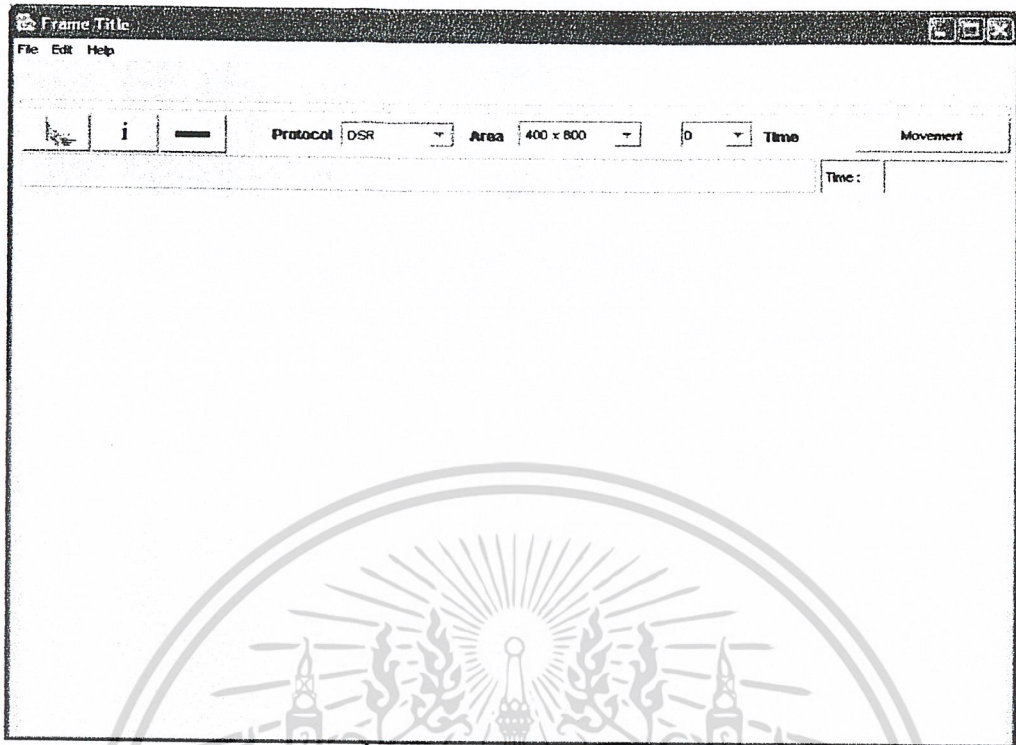
Movement หมายถึง การที่เราจะกำหนดให้แต่ละโหนดทำการเคลื่อนที่ในระหว่างที่ทำการรันแบบจำลอง

2. ส่วนหน้าจอแสดงผลการรันแบบจำลอง ซึ่งได้แสดงค่าการประมวลผลต่างๆ ของโปรแกรมโหนดการทำงาน

แบ่งเป็น 2 โหมด คือ

1. โหมด Manual จะทำการรันโปรแกรมและทำการกำหนดจุดส่งและจุดรับด้วยตัวผู้ใช้เอง และยังสามารถกำหนดการเคลื่อนไหวของโหนดได้ด้วยตนเอง

2. โหมด Auto โปรแกรมจะทำการรันโดยอัตโนมัติและทำการสุ่มโหนดที่ไว้รับและส่งข้อมูลตามระยะเวลาที่กำหนด



รูปที่ 7-3 หน้าจอหลักของโปรแกรม

คลาสและฟังก์ชันในการจัดการส่วนติดต่อกับผู้ใช้ดังนี้

Frame1 เป็นคลาสสำหรับสร้างหน้าต่างเทอร์มินัลเพื่อติดต่อกับผู้ใช้ โดยหน้าต่างเทอร์มินัลจะประกอบไปด้วยเมนูบาร์ ทูลบาร์ ส่วนที่ใช้แสดงภาพเครือข่ายจำลองในรูปแบบกราฟฟิคส์ข้อมูลของอินเทอร์เน็ตแต่ละตัว เช่น ชื่ออินเทอร์เน็ต เฟส ประเภทอินเทอร์เน็ต เฟส และสถานะของอินเทอร์เน็ต ดังรูปที่ 7-3 ซึ่งประกอบด้วยส่วนย่อยๆ ดังต่อไปนี้

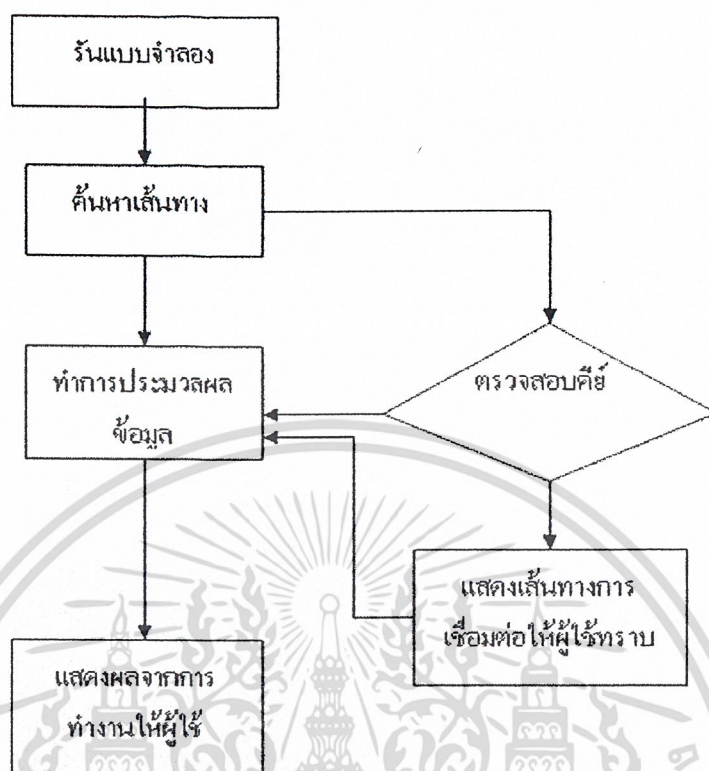
- Topology myTopology เป็นคุณสมบัติเพื่อการอ้างอิงถึงเครือข่ายจำลอง
- ZoneShow Z เป็นคุณสมบัติเพื่อการอ้างอิงถึงส่วนที่ทำการแสดงเครือข่ายในรูปแบบกราฟฟิค
- Report Re เป็นคุณสมบัติเพื่อการอ้างอิงถึงการแสดงผลประสิทธิภาพของเครือข่าย
- long timer เป็นคุณสมบัติเพื่อการแสดงเวลาที่ใช้ในการรัน โปรแกรม
- public void paint(Graphics g) เป็นฟังก์ชันที่ใช้ในการแสดงกราฟฟิค ในขณะที่รัน โปรแกรม
- private void jbInit() ในการกำหนดค่าให้กับองค์ประกอบต่างๆ ของโปรแกรม เช่น ขนาดของปุ่ม ขนาดของหน้าจอแสดงผล
- void JTextArea1_mousePressed(MouseEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการกดเมาส์ที่ส่วนแสดงผลเครือข่ายในรูปแบบกราฟฟิค
- void JTextArea1_mouseReleased(MouseEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการปล่อยเมาส์ที่ส่วนแสดงผลเครือข่ายในรูปแบบกราฟฟิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- void JTextArea1_mouseDragged(MouseEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการเคลื่อนย้ายที่ส่วนแสดงผลเครือข่ายในรูปแบบกราฟฟิก
- void jMenuItem1_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อต้องการสร้างเครือข่ายใหม่
- void jMenuItem3_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการเลือกการทำงานในโหมด manual
- void jMenuItem11_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการกำหนดครีสมี่ของ โหนด
- void jMenuItem6_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการหยุดการทำงานของเครือข่ายที่ทำการจำลองขึ้น
- void jMenuItem7_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อต้องการการรายงานผลการทำงานของเครือข่าย
- void jMenuItem8_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการออกจาก โปรแกรม
- void jMenuItem4_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการ suspend
- void jMenuItem5_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการ resume
- void jMenuItem2_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการเลือกให้รันในโหมด Auto
- void JComboBox3_actionPerformed(ActionEvent e) เป็นฟังก์ชันที่จัดการเหตุการณ์เมื่อเกิดการเลือกเวลาที่ใช้ในการรัน โปรแกรม

7.3 การจัดการในส่วนของการรันแบบจำลอง

เมื่อได้เลือกคำสั่ง Run Simulator แล้วโปรแกรมจะเริ่มทำการรันแบบจำลอง โดยในโหมดนี้ โปรแกรมจะทำการหาเส้นทางตามโปรโตคอลที่ได้ทำการเลือกไว้ แล้วเลือกเส้นทางที่เหมาะสมที่สุดในขณะนั้นทำการเชื่อมต่อแล้วจึงส่งข้อมูล เมื่อเลือกคำสั่ง Run Simulator คำสั่งต่างๆ จะไม่สามารถใช้ได้ ในช่วงนี้จนกว่าจะรันแบบจำลองเสร็จ ยกเว้นคำสั่ง Movement ซึ่งจะเป็นการสั่งให้โหนดเคลื่อนที่หรือหยุดได้ เมื่อโหนดหยุดเคลื่อนที่จะทำให้เราให้เห็นเส้นทางการเชื่อมต่อของโหนดต่างๆ ที่เราได้กำหนดไว้ และในช่วงที่ทำการรันแบบจำลองนี้เราสามารถนำเมาส์ไปชี้ที่โหนดเหล่านั้นเพื่อดูการประมวลผลและรายละเอียดของโหนดต่างๆ ได้ดังรูปที่ 7-4



รูปที่ 7-4 ขั้นตอนการทำงานของโปรแกรมในส่วนของการรับแบบจำลอง

คลาสและฟังก์ชันในการจัดการการรับแบบจำลองมีดังนี้

ZoneShow เป็นคลาสที่ทำหน้าที่ในการแสดงผลในรูปแบบของกราฟฟิกและทำการดำเนินการให้เครือข่ายจำลองทำงานประกอบด้วยคุณสมบัติและฟังก์ชันดังต่อไปนี้

- Icon nodePic เป็นคุณสมบัติที่ใช้ในการแสดงรูปโหนดในส่วนแสดงผลกราฟฟิก
- JTextArea JTextArea เป็นคุณสมบัติที่เป็นพื้นที่ในการสร้างเครือข่าย
- Topology t1 เป็นคุณสมบัติที่อ้างอิงถึงเครือข่าย
- int count เป็นคุณสมบัติที่ใช้เพื่อการตรวจสอบ Time Out ของการส่งแพ็กเก็ต
- int ranSeed เป็นคุณสมบัติเพื่อการสุ่มข้อมูลให้กับแต่ละ โหนด
- boolean Auto เป็นคุณสมบัติที่แสดงถึง โหมดการทำงานของ โปรแกรม
- long timer เป็นคุณสมบัติที่แสดงถึงเวลาในขณะที่โปรแกรมทำงาน
- void ZoneShow(JTextArea JTextArea1, Topology t, boolean mode, JTextField j) เป็นคอนสตรัคเตอร์ของคลาสเพื่อกำหนดค่าเริ่มต้นให้กับคุณสมบัติต่างๆของคลาส
- void run() เป็นส่วนที่ทำการรันเครือข่ายจำลอง
- void clear(JTextArea JTextArea 1, Topology t, boolean mode) เป็นส่วนที่ทำการล้างข้อมูลเดิมที่มีอยู่เพื่อทำการสร้างเครือข่ายใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- void setAuto(boolean m) เป็นส่วนที่ใช้ในการกำหนดโหมดของการทำงาน
- void setTime(long l) เป็นส่วนที่ใช้ในการกำหนดเวลาที่ใช้ในการรันแบบจำลอง

7.4 การจัดการตำแหน่งของเครื่องคอมพิวเตอร์

คลาสและฟังก์ชันในการจัดการตำแหน่งของเครื่องคอมพิวเตอร์มีดังนี้

Topology เป็นคลาสที่ทำหน้าที่เป็นตัวแทนของเครือข่ายที่เราทำการสร้างขึ้น [11] ประกอบด้วยคุณสมบัติและฟังก์ชันดังต่อไปนี้

- private MobileNode topology ใช้เป็นจุดเริ่มต้นของลิงก์ที่ทำการเก็บโหนดทั้งหมดในเครือข่าย
- private MobileNode topologyTail เป็นจุดที่อ้างอิงถึงโหนดล่าสุดในเครือข่าย
- private int numberOfNode เป็นจำนวน โหนดทั้งหมดที่อยู่ในเครือข่าย
- private MobileNode source เป็นค่าที่อ้างอิงถึงโหนดที่เป็นจุดเริ่มต้นการส่ง
- private MobileNode Des เป็นค่าที่อ้างอิงถึงโหนดที่เป็นปลายทางในการส่ง
- private boolean move เป็นค่าที่กำหนดว่า โหนดในเครือข่ายมีการเคลื่อนไหวหรือไม่
- private static int ID เป็นหมายเลขของเครือข่าย
- public Topology() เป็นคอนสตรัคเตอร์ของคลาส
- public void setTopology(MobileNode n) เป็นการกำหนด โหนดแรกของเครือข่าย
- public MobileNode getTopology() เป็นการส่งค่าที่อ้างอิงถึงเครือข่าย
- public void setTopologyTail(MobileNode n) เป็นการกำหนดค่าที่อ้างอิงถึงโหนดตัวสุดท้ายในเครือข่าย
- public MobileNode getTopologyTail() เป็นฟังก์ชันที่คืนค่าที่อ้างอิงถึงโหนดตัวสุดท้ายในเครือข่าย
- public void increasNumOfNode() เพิ่มจำนวน โหนดในเครือข่าย
- public void decreasNumOfNode() ลดจำนวน โหนดในเครือข่าย
- public void addNewNode() เพิ่ม โหนดใหม่เข้าไปในเครือข่าย
- public void setSource(MobileNode s) กำหนดโหนดที่เป็นจุดเริ่มของการส่งข้อมูล
- public void setDes(MobileNode d) กำหนด โหนดที่เป็นจุดปลายทางของการส่งข้อมูล
- public MobileNode getSource() คืนค่าที่อ้างอิงถึง โหนดที่เป็นจุดเริ่มของการส่งข้อมูล
- public MobileNode getDes() คืนค่าที่อ้างอิงถึง โหนดที่เป็นจุดปลายทางของการส่งข้อมูล
- public void setMove(boolean m) กำหนดว่า โหนดมีการเคลื่อนไหวหรือไม่
- public boolean getMove() คืนค่าที่บอกว่า โหนดมีการเคลื่อนไหวหรือไม่
- public int getNumOfNode() คืนค่าจำนวน โหนดทั้งหมดที่อยู่ในเครือข่ายในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MobileNode เป็นคลาสที่เป็นตัวแทนของ โหนดแต่ละ โหนด[6]ซึ่ง โหนดจะทำหน้าที่เป็นผู้รับ และส่งแพ็คเกจภายใน โหนดจะมีการเก็บค่าคุณสมบัติต่างๆของ โหนด โดยจะแตกต่างกันไปในแต่ละ โหนด เช่นความเร็วในการเคลื่อนที่ ทิศทางการเคลื่อนที่ ตำแหน่งปัจจุบัน หมายเลขประจำตัวของ โหนด เป็นต้น ใน โหนดจะประกอบด้วยส่วนต่างๆดังต่อไปนี้

- private int positionX ตำแหน่งของ โหนดในแนวนอน
- private int positionY ตำแหน่งของ โหนดในแนวตั้ง
- private int ID หมายเลขประจำ โหนด
- private int signalRadias รัศมีการส่งสัญญาณของ โหนด
- private int direct ทิศทางการเคลื่อนที่ของ โหนด
- private int dataSend ข้อมูลเกี่ยวกับแพ็คเกจข้อมูลที่ส่งไป
- private int dataSuc ข้อมูลเกี่ยวกับแพ็คเกจข้อมูลที่ส่ง ไปถึงจุดหมาย
- private int routeSend ข้อมูลเกี่ยวกับแพ็คเกจหาเส้นทางที่ส่งไป
- private int routeSuc ข้อมูลเกี่ยวกับแพ็คเกจหาเส้นทางที่ส่งไปแล้วสามารถหาเส้นทางได้สำเร็จ
- private int speed ความเร็วของ โหนด
- private double d ระยะทางจากจุดเริ่มต้นมายัง โหนดนี้ที่เป็นระยะทางที่สั้นที่สุด
- private MobileNode predeccessor โหนดก่อนหน้าในการค้นหาเส้นทาง ได้สำเร็จ
- MobileNode nextNode โหนดต่อไปในเครือข่าย
- MobileNode beforeNode โหนดก่อนหน้าในเครือข่าย
- public MobileNode() เป็นคอนสตรัคเตอร์ของคลาส
- public void setXPosition(int x) กำหนดตำแหน่งของ โหนดในแนวนอน
- public void setYPosition(int y) กำหนดตำแหน่งของ โหนดในแนวตั้ง
- public int getXPosition() คืนค่าตำแหน่งของ โหนดในแนวนอน
- public int getYPosition()คืนค่าตำแหน่งของ โหนด ในแนวตั้ง
- public void setID(int id) กำหนดหมายเลขประจำ โหนด
- public int getID() คืนค่าหมายเลขประจำ โหนด
- public void setSignalRadias(int r) กำหนดค่ารัศมีการส่งสัญญาณของ โหนด
- public int getSignalRadias() คืนค่ารัศมีการส่งสัญญาณของ โหนด
- public void set_d(double dd) กำหนดระยะทางจากจุดเริ่มต้นมายัง โหนดนี้ที่เป็นระยะทางที่สั้นที่สุด
- public double get_d() คืนค่าระยะทางจากจุดเริ่มต้นมายัง โหนดนี้ที่เป็นระยะทางที่สั้นที่สุด
- public void setPredeccessor(MobileNode m) กำหนด โหนดก่อนหน้าในการค้นหาเส้นทาง ได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `public MobileNode getPredecessor()` คืนค่า โหนดก่อนหน้าในการค้นหาเส้นทางได้สำเร็จ
- `public void setNextNode(MobileNode m)` กำหนด โหนดต่อไปในเครือข่าย
- `public MobileNode getNextNode()` คืนค่า โหนดต่อไปในเครือข่าย
- `public void setBeforeNode(MobileNode m)` กำหนด โหนดก่อนหน้าในเครือข่าย
- `public MobileNode getBeforeNode()` คืนค่า โหนดก่อนหน้าในเครือข่าย
- `public void setDirect(int a)` กำหนดทิศทาง
- `public int getDirect()` คืนค่าทิศทาง
- `public void addRouteSend()` เพิ่มข้อมูลเกี่ยวกับแพ็กเก็ตข้อมูลที่ส่งไป
- `public void addRouteSuc()` เพิ่มข้อมูลเกี่ยวกับแพ็กเก็ตข้อมูลที่ส่งไปถึงจุดหมาย
- `public void addDataSend()` เพิ่มข้อมูลเกี่ยวกับแพ็กเก็ตหาเส้นทางที่ส่งไป
- `public void addDataSuc()` เพิ่มข้อมูลเกี่ยวกับแพ็กเก็ตหาเส้นทางที่ส่งไปแล้วสามารถหาเส้นทางได้สำเร็จ
- `public int getDataSend()` คืนค่าข้อมูลเกี่ยวกับแพ็กเก็ตข้อมูลที่ส่งไป
- `public int getDataSuc()` คืนค่าข้อมูลเกี่ยวกับแพ็กเก็ตข้อมูลที่ส่งไปถึงจุดหมาย
- `public int getRouteSend()` คืนค่าข้อมูลเกี่ยวกับแพ็กเก็ตหาเส้นทางที่ส่งไป
- `public int getRouteSuc()` คืนค่าข้อมูลเกี่ยวกับแพ็กเก็ตหาเส้นทางที่ส่งไปแล้วสามารถหาเส้นทางได้สำเร็จ
- `public int getSpeed()` คืนค่าความเร็วของ โหนด

`MobileControl` เป็นคลาสที่ทำหน้าที่ปรับปรุงข้อมูลเกี่ยวกับการเคลื่อนที่ของ โหนด[8][9]

ประกอบด้วยคุณสมบัติและฟังก์ชันดังต่อไปนี้

- `public MobileControl()` เป็นคอนสตรัคเตอร์ของคลาส
- `public static void Update(Topology t1)` เป็นฟังก์ชันที่ใช้ในการจัดการการเคลื่อนไหวของ โหนด

7.5 จัดการเกี่ยวกับโปรโตคอล

ในการจัดการเกี่ยวกับโปรโตคอลนั้นคลาสที่เกี่ยวข้องจะขึ้นอยู่กับโปรโตคอลที่ใช้ด้วยดังนั้นหากต้องการโปรโตคอลใหม่ๆก็สามารถเขียนคลาสใหม่ขึ้นมารองรับได้ ในส่วนของส่วนประกอบเกี่ยวกับการจัดการโปรโตคอลมีดังนี้

`TrafficGenerator` เป็นคลาสที่อินสแตนซ์ของคลาสโทโพโลยีใช้ในการจัดการด้านการสร้างทราฟฟิกในเครือข่าย ก็คือการส่งสตรีมของแพ็กเก็ตไปในเครือข่ายโดยคลาสนี้จะประกอบด้วยส่วนประกอบดังนี้

- `public TrafficGenerator()` เป็นคอนสตรัคเตอร์ของคลาส
- `public static void setDataSend(Topology t1)` เป็นฟังก์ชันที่ใช้ในการปรับปรุงข้อมูลเกี่ยวกับการส่งแพ็กเก็ตข้อมูล
- `public static void setDataSuc(Topology t1)` เป็นฟังก์ชันที่ใช้ในการปรับปรุงข้อมูลเกี่ยวกับการส่งแพ็กเก็ตข้อมูลที่สามารถส่งได้สำเร็จ
- `public static void setRouteSend(Topology t1)` เป็นฟังก์ชันที่ใช้ในการปรับปรุงข้อมูลเกี่ยวกับการส่งแพ็กเก็ตหาเส้นทาง
- `public static void setRouteSuc(Topology t1)` เป็นฟังก์ชันที่ใช้ในการปรับปรุงข้อมูลเกี่ยวกับการส่งแพ็กเก็ตหาเส้นทางที่สามารถหาเส้นทางได้สำเร็จ

DSR เป็นคลาสที่ทำหน้าที่หาเส้นทางในการส่งข้อมูลในรูปแบบของ DSR [13] ประกอบด้วยคุณสมบัติและฟังก์ชันดังต่อไปนี้

- `public DSR()` เป็นคอนสตรัคเตอร์ของคลาส
- `public static void letGo(Topology t1)` เป็นฟังก์ชันที่ทำหน้าที่หาเส้นทางจากจุดเริ่มต้นไปยังจุดหมายปลายทาง

7.6 การจัดการเกี่ยวกับการรายงานผล

ในส่วนของจัดการเกี่ยวกับการจัดการเกี่ยวกับการรายงานผลจะทำหน้าที่นำข้อมูลจากอินสแตนซ์ของ โหนดและ โทโทโลยีมาทำการประมวลผลออกมาเป็นภาพรวมของการทำงานของเครือข่ายที่เราได้ทำการจำลองขึ้นซึ่งหลังจากได้ทำการประมวลผลแล้วจะทำการแสดงในหน้าจอพาเนลและสามารถบันทึกลงเป็นไฟล์ได้โดยส่วนประกอบต่างๆของส่วนการจัดการเกี่ยวกับการรายงานผลมีดังนี้

```

Reported
Number Of Node 0
DataSend 0
DataSuccess 0
RouteOccur 0
RouteSuccess 0
Packet Delivery Success Ratio 0.0
Route Discovery Success Probability 0.0

```

รูปที่ 7-5 หน้าจอแสดงการรายงานผล

Report เป็นคลาสที่ทำหน้าที่เป็นตัวแทนของรายงานที่เราทำการคำนวณดังแสดงในรูปที่ 7-5 ประกอบด้วยคุณสมบัติและฟังก์ชันดังต่อไปนี้

- int dataSend ข้อมูลเกี่ยวกับแพ็กเกตข้อมูลที่ส่งไป
- int dataSuc ข้อมูลเกี่ยวกับแพ็กเกตข้อมูลที่ส่งไปถึงจุดหมาย
- int routeSend ข้อมูลเกี่ยวกับแพ็กเกตหาเส้นทางที่ส่งไป
- int routeSuc ข้อมูลเกี่ยวกับแพ็กเกตหาเส้นทางที่ส่งไปแล้วสามารถหาเส้นทางได้สำเร็จ
- double PDSR เป็นอัตราส่วนระหว่างข้อมูลที่ส่งสำเร็จกับข้อมูลทั้งหมดที่ส่งไป
- double RDSP เป็นอัตราส่วนระหว่างแพ็กเกตหาเส้นทางที่สามารถหาเส้นทางได้สำเร็จกับแพ็กเกตหาเส้นทางทั้งหมดที่ส่งไป
- public Report() เป็นคอนสตรัคเตอร์ของคลาส
- public void calPDSR() ใช้ในการคำนวณค่าอัตราส่วนระหว่างข้อมูลที่ส่งสำเร็จกับข้อมูลทั้งหมดที่ส่งไป
- public void calRDSP() ใช้ในการคำนวณค่าอัตราส่วนระหว่างแพ็กเกตหาเส้นทางที่สามารถหาเส้นทางได้สำเร็จกับแพ็กเกตหาเส้นทางทั้งหมดที่ส่งไป
- public void showReport(JTextArea reportArea, JFrame report, Topology myTopology) ใช้ในการแสดงข้อมูลรายงานออกหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- public void clear() ใช้ในการล้างข้อมูลรายงาน

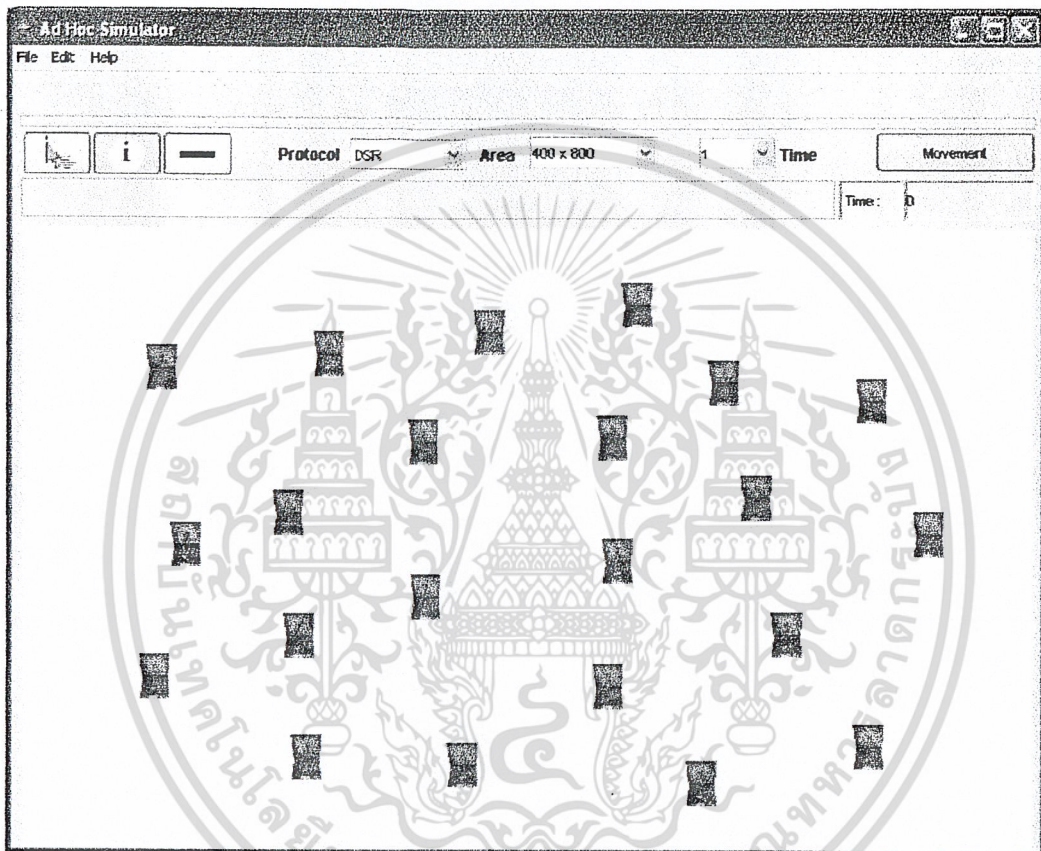


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

ตัวอย่างและการทดสอบการทำงานของโปรแกรม

ตัวอย่างนี้เป็นการทดสอบการทำงานของโปรโตคอลดีเอสอาร์เมื่อสร้างเครือข่ายจำลองให้เป็นดังรูป 8-1



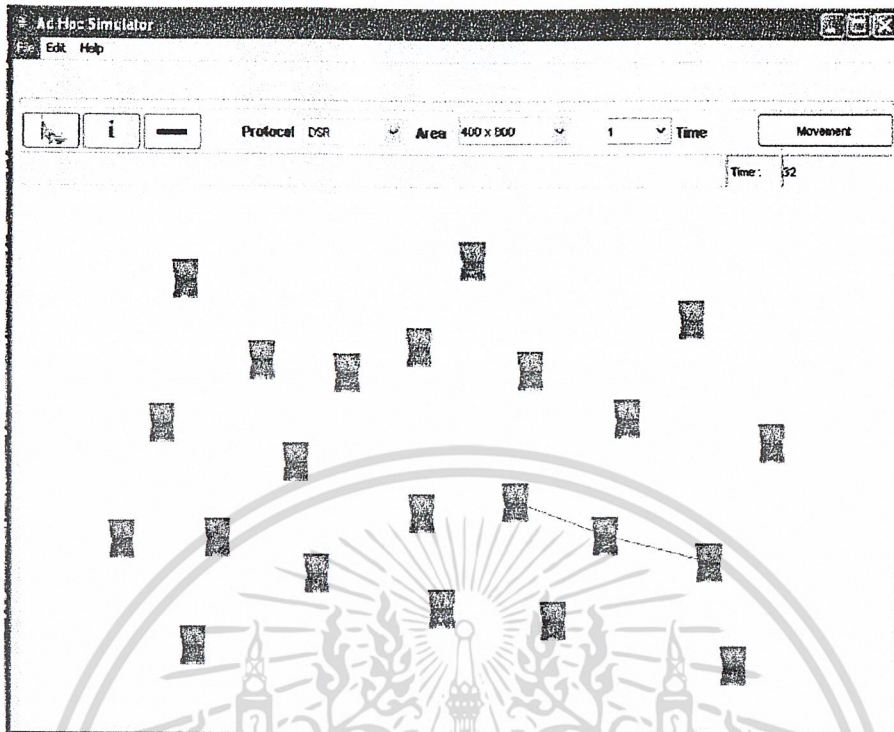
รูปที่ 8-1 ภาพแสดงเครือข่ายที่จะเริ่มทำการทดสอบที่ 1

การกำหนดค่าต่างๆ ที่ใช้ในการทดสอบที่ 1 เป็นการทดสอบเกี่ยวกับรัศมีที่เหมาะสมที่ใช้ในการส่งข้อมูลในขนาดของพื้นที่หนึ่ง

1. รัศมีของพื้นที่จำนวน 22 โหนด
2. กำหนดให้ใช้โปรโตคอล DSR
3. ขนาดพื้นที่ 400 x 800 หน่วย
4. เวลาที่ใช้ในการรันแบบจำลอง 1 นาที

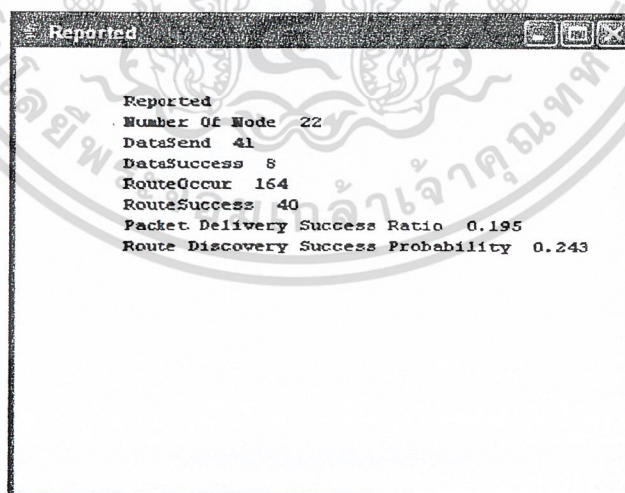
โดยที่หน่วยในพื้นที่สามารถนำไปเทียบอัตราส่วนกับขนาดพื้นที่จริงได้ เช่น 1 พิกเซล ต่อ 1 เซนติเมตร หรือ 1 พิกเซล ต่อ 1 เมตร ซึ่งจะเปลี่ยนไปตามชนิดของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-2 ภาพแสดงเครือข่ายหลังจากการทดสอบแล้ว

การทดลองครั้งที่ 1 ทำการตั้งค่ารัศมีของสัญญาณเป็น 100 หน่วย เมื่อผ่านการทดสอบแล้วจะได้ผลดังรูป 8-2 และ ได้นำจอการรายงานผลตามรูป 8-3



รูปที่ 8-3 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 2 ได้เพิ่มค่ารัศมีเป็น 110 หน่วย เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-4

```

Reported
Number Of Node 22
DataSend 22
DataSuccess 7
RouteOccur 47
RouteSuccess 23
Packet Delivery Success Ratio 0.318
Route Discovery Success Probability 0.489
  
```

รูปที่ 8-4 หน้าจอแสดงผลการรายงานผลการทดลองครั้งที่ 2

การทดลองครั้งที่ 3 ได้เพิ่มค่ารัศมีเป็น 120 หน่วย เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-5

```

Reported
Number Of Node 22
DataSend 237
DataSuccess 139
RouteOccur 332
RouteSuccess 235
Packet Delivery Success Ratio 0.586
Route Discovery Success Probability 0.707
  
```

รูปที่ 8-5 หน้าจอแสดงผลการรายงานผลการทดลองครั้งที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 4 ได้เพิ่มค่ารัศมีเป็น 130 หน่วย เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-6

```

Reported
Number Of Node 22
DataSend 913
DataSuccess 750
RouteOccur 949
RouteSuccess 909
Packet Delivery Success Ratio 0.821
Route Discovery Success Probability 0.957
  
```

รูปที่ 8-6 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 4

การทดลองครั้งที่ 4 ได้เพิ่มค่ารัศมีเป็น 140 หน่วย เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-7

```

Reported
Number Of Node 22
DataSend 1460
DataSuccess 1427
RouteOccur 1493
RouteSuccess 1456
Packet Delivery Success Ratio 0.977
Route Discovery Success Probability 0.975
  
```

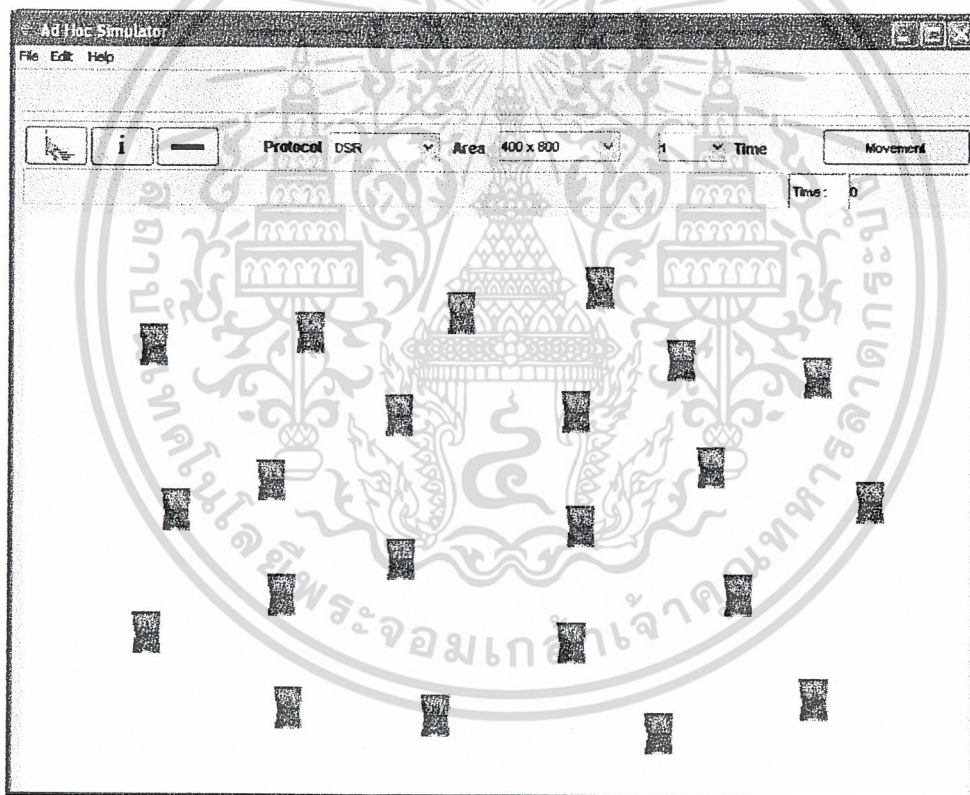
รูปที่ 8-7 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบจะเห็นได้ว่าเมื่อรัศมีของสัญญาณมีความไกลมากขึ้นประสิทธิภาพที่ได้จะมีค่าเพิ่มมากขึ้นตามไปด้วย โดยที่ค่ารัศมีที่ใช้สำหรับในพื้นที่ 400 x 800 หน่วย และมีโหนดทั้งหมด 22 โหนดจะต้องมีค่ารัศมีตั้งแต่ 140 หน่วย ขึ้นไป จึงจะได้ค่าประสิทธิภาพที่ดี

การกำหนดค่าต่างๆ ที่ใช้ในการทดสอบที่ 2 เป็นการทดสอบจำนวนโหนดที่เหมาะสมที่ใช้ในการส่งข้อมูลในขนาดของพื้นที่หนึ่ง

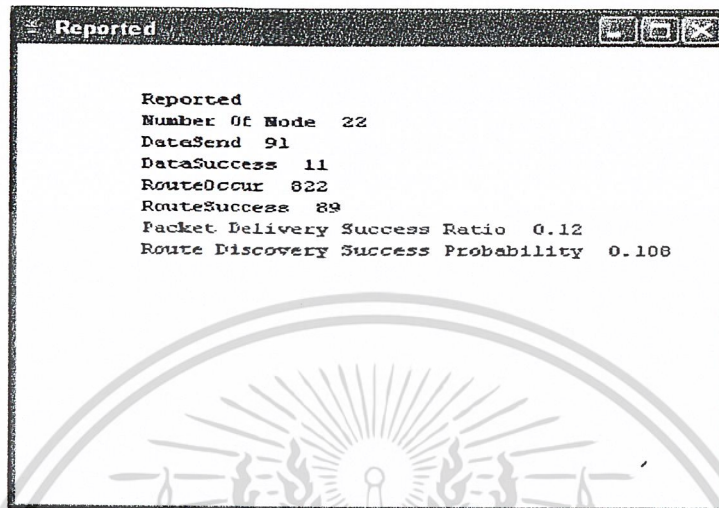
1. ใส่โหนดเข้าไปจำนวน 22 โหนด
2. กำหนดให้ใช้โปรโตคอล DSR
3. ขนาดพื้นที่ 400 x 800 หน่วย
4. เวลาที่ใช้ในการรันแบบจำลอง 1 นาที



รูปที่ 8-8 ภาพแสดงเครือข่ายที่จะเริ่มทำการทดสอบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 1 ได้ตั้งค่ารัศมีเป็น 100 หน่วย และมีจำนวน โหนด 22 โหนด เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-9

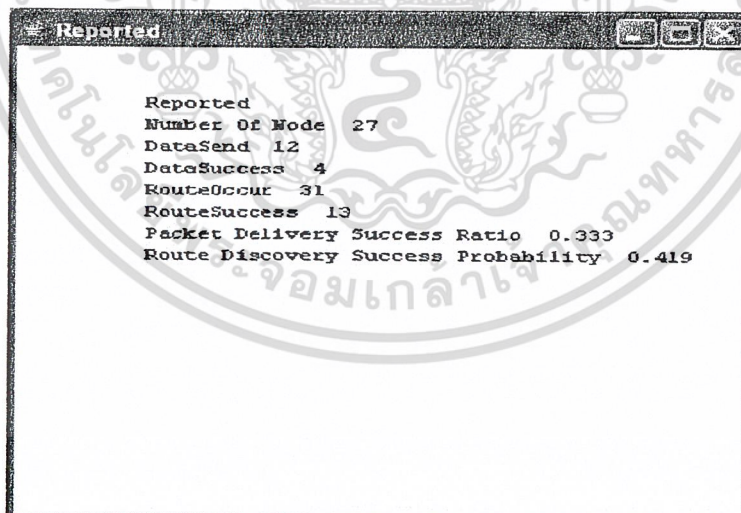


```

Reported
Number Of Node 22
DataSend 91
DataSuccess 11
RouteOccur 822
RouteSuccess 89
Packet Delivery Success Ratio 0.12
Route Discovery Success Probability 0.108
  
```

รูปที่ 8-9 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 1

การทดลองครั้งที่ 2 ได้ตั้งค่ารัศมีเป็น 100 หน่วย และเพิ่มจำนวน โหนดเป็น 27 โหนด เมื่อผ่านการทดสอบแล้วได้ผลดังรูป 8-10



```

Reported
Number Of Node 27
DataSend 12
DataSuccess 4
RouteOccur 31
RouteSuccess 13
Packet Delivery Success Ratio 0.333
Route Discovery Success Probability 0.419
  
```

รูปที่ 8-10 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 3 ได้ตั้งค่ารัศมีเป็น 100 หน่วย และเพิ่มจำนวน โหนดเป็น 32 โหนด เมื่อผ่าน การทดสอบแล้ว ได้ผลดังรูป 8-11

```

Reported
Number Of Node 32
DataSend 2943
DataSuccess 1290
RouteOccur 5903
RouteSuccess 2872
Packet Delivery Success Ratio 0.438
Route Discovery Success Probability 0.406
  
```

รูปที่ 8-11 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 3

การทดลองครั้งที่ 4 ได้ตั้งค่ารัศมีเป็น 100 หน่วย และเพิ่มจำนวน โหนดเป็น 37 โหนด เมื่อผ่าน การทดสอบแล้ว ได้ผลดังรูป 8-12

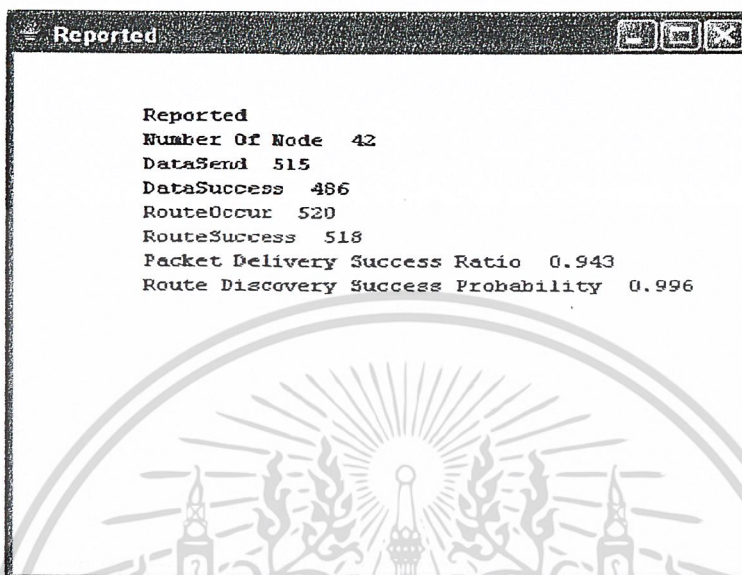
```

Reported
Number Of Node 37
DataSend 456
DataSuccess 312
RouteOccur 600
RouteSuccess 457
Packet Delivery Success Ratio 0.684
Route Discovery Success Probability 0.761
  
```

รูปที่ 8-12 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองครั้งที่ 5 ได้ตั้งค่ารัศมีเป็น 100 หน่วย และเพิ่มจำนวน โหนดเป็น 42 โหนด เมื่อผ่าน การทดสอบแล้วได้ผลดังรูป 8-13



```

Reported
Number Of Node 42
DataSend 515
DataSuccess 486
RouteOccur 520
RouteSuccess 518
Packet Delivery Success Ratio 0.943
Route Discovery Success Probability 0.996
  
```

รูปที่ 8-13 หน้าจอแสดงการรายงานผลการทดลองครั้งที่ 5

จากการทดลองจะเห็นได้ว่าเมื่อเพิ่มจำนวน โหนดให้มากขึ้นจะทำให้มีประสิทธิภาพเพิ่มมากขึ้น ด้วยเนื่องจากมีโหนดที่ใช้ในการเชื่อมต่อมากขึ้น และระหว่างโหนดก็ลดลงด้วย

บทที่ 9

บทวิจารณ์และสรุป

โครงการที่จัดทำขึ้นนี้ เป็นโครงการที่มีจุดประสงค์เพื่อเปิด โอกาสทางการศึกษาให้กับบุคคลทั่วไปที่สนใจทางด้านเครือข่ายไร้สายแบบแอดฮอค รวมไปถึงการศึกษางานของโพรโตคอลการพิจารณาเส้นทางในเครือข่ายแบบนี้ โดยโครงการนี้คือซอฟต์แวร์ที่สามารถจำลองการทำงานของเครือข่ายแบบแอดฮอค และจำลองการทำงานของโพรโตคอลที่ใช้ในเครือข่ายแบบนี้ อีกทั้งยังสามารถแสดงเครือข่ายจำลองออกมาเป็นภาพ 2 มิติเพื่อให้เห็นภาพรวมของเครือข่ายได้อย่างชัดเจนอีกด้วย

ในทุกวันนี้ เครือข่ายไร้สายแบบแอดฮอคกำลังเป็นทางเลือกใหม่ในการเชื่อมต่อที่ลงทุนต่ำและเหมาะสำหรับเครือข่ายที่มีอายุการใช้งานในระยะสั้น การเข้าใจในพื้นฐานของเครือข่ายไร้สายแบบแอดฮอค และหลักการพื้นฐานในการพิจารณาเส้นทางของโพรโตคอล จะทำให้สามารถประยุกต์ใช้งานเครือข่ายได้ดียิ่งขึ้น

9.1 ขอบเขตและข้อจำกัดของโปรแกรม

- สามารถรองรับการสร้างโหนดได้ในจำนวนน้อย
- รองรับโพรโตคอลการพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอคได้ในจำนวนน้อย
- แสดงข้อมูลภาพรวมของเครือข่ายได้ไม่ละเอียด

9.2 กลุ่มผู้ใช้โปรแกรม

บุคคลโดยทั่วไปที่มีความต้องการศึกษาและทำความเข้าใจการทำงานของโพรโตคอลการพิจารณาเส้นทางในเครือข่ายไร้สายแบบแอดฮอค หรือศึกษางานของเครือข่ายไร้สายแบบแอดฮอค เหมาะกับผู้ที่ต้องการศึกษาแต่ไม่มีเครือข่ายจริงให้ศึกษา หรือมีแต่ยังไม่เข้าใจการทำงานและข้อจำกัดของเครือข่ายแบบนี้เป็นคือ หรือไม่ต้องการใช้เครือข่ายจริงในการทดสอบเนื่องจากเหตุผลทางด้านค่าใช้จ่าย

9.3 ปัญหาและอุปสรรค

- เนื่องจาก โหนด ในเครือข่ายต้องมีการเคลื่อนที่อยู่ตลอดเวลา ทำให้ต้องใช้เวลาในการออกแบบ และมีความยุ่งยากในการเขียน โปรแกรมเพิ่มขึ้น
- เนื่องจากมีหลายปัจจัยที่ส่งผลต่อการทำงานของเครือข่ายไร้สายแบบแอดฮอค ทำให้ยากต่อการจำลอง เช่น ทิศทางในการเคลื่อนที่ของโหนด ความเร็วของโหนด หรือรัศมีหรือขอบเขตการสื่อสารของแต่ละ โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4 แนวทางการประยุกต์และพัฒนา

โครงการนี้ จัดทำขึ้นแบบ OOP (Object Oriented Programming) ซึ่งทำให้สามารถเพิ่มเติมและ พัฒนาต่อได้โดยง่าย โดยสามารถเพิ่มโพรโทคอลใหม่ๆ ได้ด้วยการกำหนดให้เป็นคลาสใหม่ขึ้นมา รวมทั้งยังสามารถเพิ่มความสามารถต่าง ๆ ให้กับตัวโปรแกรม เช่น ความสามารถในการแสดงผล ปรับปรุงในส่วนติดต่อกับผู้ใช้ หรือเพิ่มความสามารถของตัวโปรแกรมในการรองรับการจำลองเครือข่ายขนาดใหญ่และมีความซับซ้อนยิ่งกว่าเดิม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

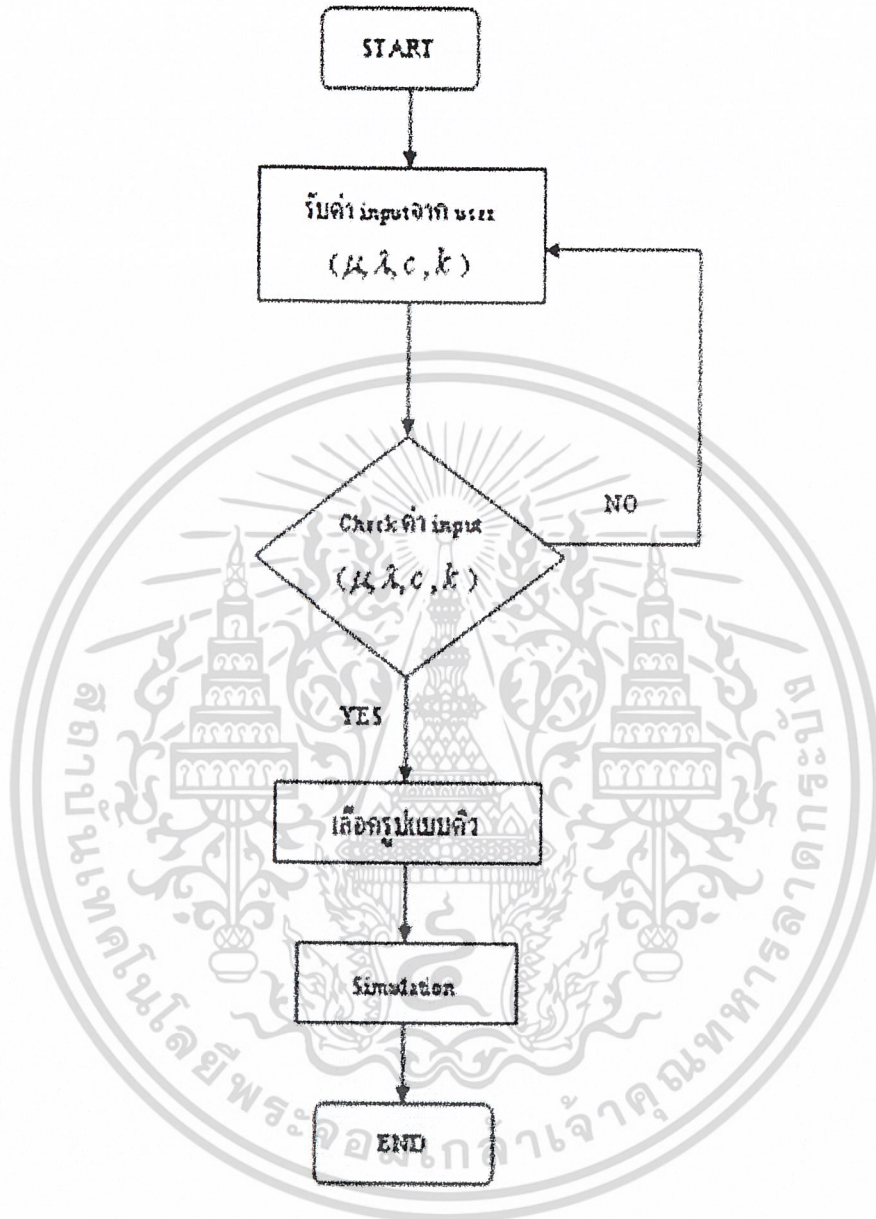
แบบจำลองของ Single-Server Queuing System

เป็นการจำลองระบบคิวโดยเน้นที่ Single Queue Model ซึ่งมีอัตราการเข้าใช้บริการและอัตราการให้บริการเป็นแบบ Markovian property หรือ Memoryless property โดยมี 5 model ดังนี้ $M/M/1$, $M/M/1/K$, $M/M/c$, $M/M/c/K$ และ $M/M/c/c$

ในการออกแบบโปรแกรมจำลองระบบ Single Queue Model จะแสดงขั้นตอนการทำงานของโปรแกรมได้ดัง รูปที่ 10-1

จากรูปที่ 10-1 จะสามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1. รับค่า input จาก user โดยค่า input ที่รับเข้ามาจะมี 4 ค่า คือ อัตราการเข้าใช้บริการ อัตราการให้บริการ จำนวนเครื่องให้บริการ และ ขนาดของระบบ
2. ตรวจสอบค่า input ที่ user ป้อนเข้ามาค่าที่ไม่ถูกต้อง เช่น user ป้อนค่า input มีค่าใดค่าหนึ่งเป็น 0 หรือติดลบ user ป้อนค่า c และ K ที่ไม่ใช่จำนวนเต็ม และ user ป้อนค่า c เป็นค่าที่เรากำหนดไว้ถ้าค่า input ไม่ถูกต้อง โปรแกรมจะกลับไปรับค่า input ใหม่ โดยถ้าค่า input ถูกต้อง โปรแกรมจะทำงานขั้นตอนต่อไป โดยการนำค่าที่ได้ไปทำการตรวจสอบว่าตรงกับคิวในรูปแบบใด ซึ่งในคิวแต่ละแบบจะใช้สูตรในการคำนวณไม่เหมือนกัน โดยแบ่งออกเป็น 5 กรณีดังนี้
 - ถ้า $c=1$ และ $K > 200$ จะเป็นคิวแบบ $M/M/1$
 - ถ้า $c=1$ และ $K \leq 200$ จะเป็นคิวแบบ $M/M/1/K$
 - ถ้า $c > 1$ และ $K > 200$ จะเป็นคิวแบบ $M/M/c$
 - ถ้า $c > 1$ และ $K \leq 200$ จะเป็นคิวแบบ $M/M/c/K$
 - ถ้า $c > 1$ และ $K = c$ จะเป็นคิวแบบ $M/M/c/c$



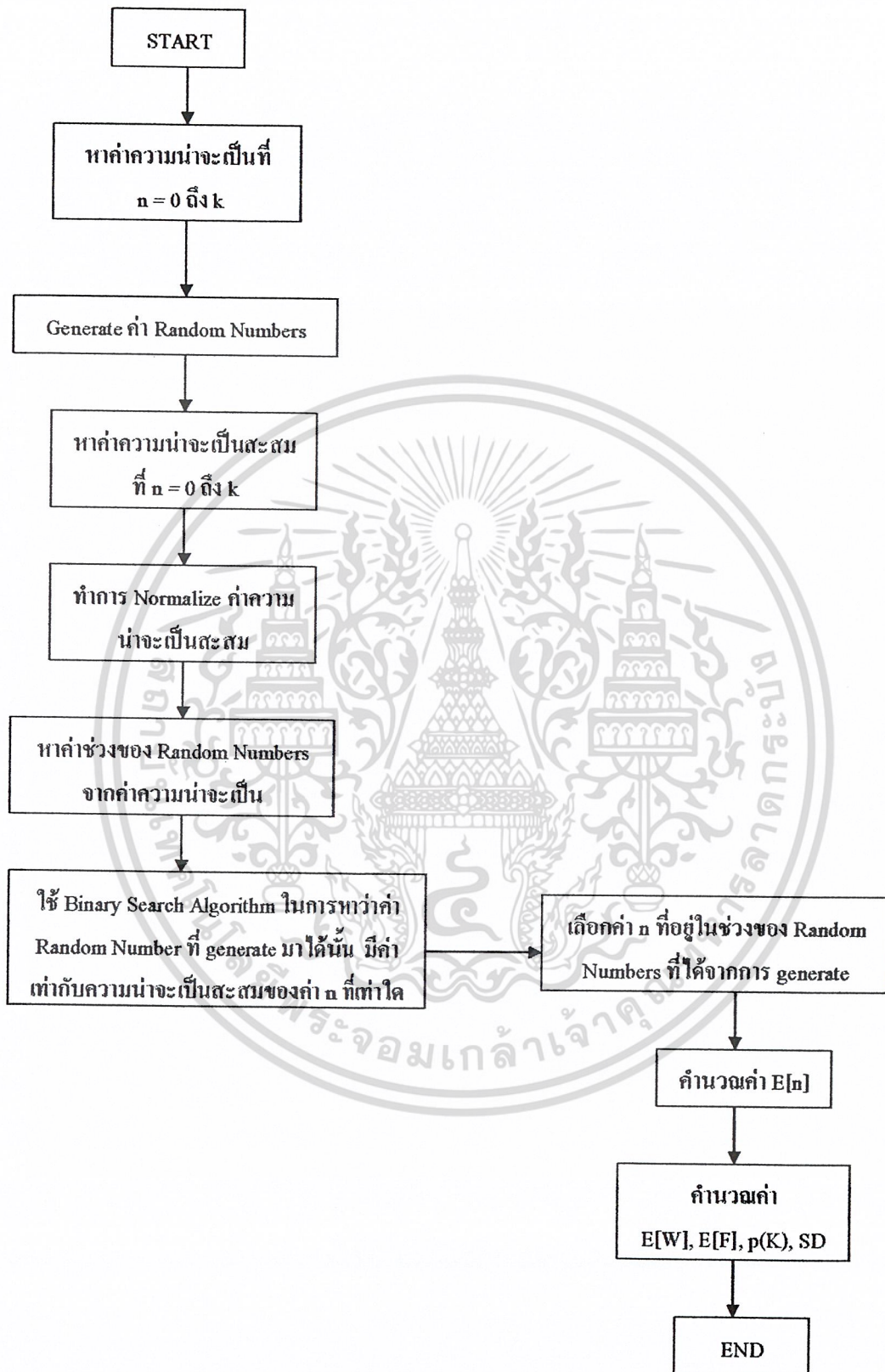
รูปที่ 10-1 แสดงขั้นตอนการทำงานของโปรแกรมจำลองระบบ Single Queue Model

จะทำการ simulation โดยในขั้นตอนนี้จะแบ่งออกเป็น 3 ส่วน ดังนี้

1. ส่วน Generate Random Number
2. ส่วน Binary search
3. ส่วนการคำนวณ

ขั้นตอนการ Simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10-2 แสดงขั้นตอนการ Simulation ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

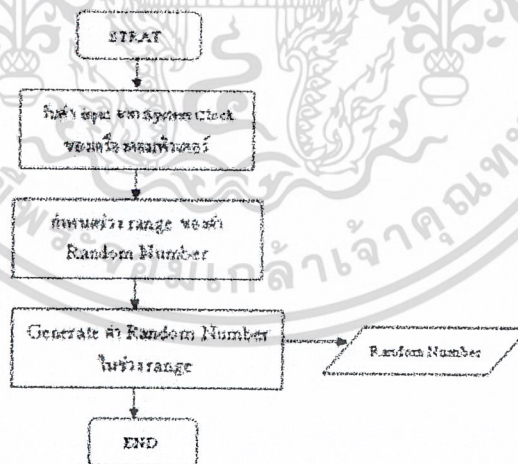
จากรูปที่ 10-2 เราจะสามารถอธิบายขั้นตอนการ Simulation ได้ดังนี้

1. เมื่อได้รูปแบบคิวมาแล้ว ทำการคำนวณหาค่าความน่าจะเป็นที่มี n งานในระบบโดยใช้สูตรของคิวในรูปแบบนั้นๆ โดยจะคำนวณตั้งแต่ค่าที่ $n=0$ ถึง K
2. หาค่าความน่าจะเป็นสะสมที่มี n งานในระบบจากค่าความน่าจะเป็นที่ได้จากข้อ 1
3. ทำการ Normalization ค่าความน่าจะเป็นสะสมที่มี n งานในระบบ
4. หาช่วงของ Random Number จากค่าความน่าจะเป็น ยกตัวอย่างเช่น กรณีที่ $K=2$ หาค่าความน่าจะเป็นที่มี n งานในระบบได้ ดังนี้

$$\begin{array}{llll} n = 0, & p_0 = 0.3000 & p_{(0)} = 0.3000 & RN = 0000 - 0002 \\ n = 1, & p_1 = 0.6000 & p_{(1)} = 0.9000 & RN = 0003 - 0008 \\ n = 2, & p_2 = 0.1000 & p_{(2)} = 1.2000 & RN = 0009 - 0011 \end{array}$$

ซึ่งจากตัวอย่างจะเห็นว่าช่วงของ Random Number ที่ได้โดยดูจากค่าความน่าจะเป็น โดยที่ $n=0$ ค่าความน่าจะเป็น = 0.3000 ซึ่งเราใช้ตัวเลข Random Number จำนวน 4 หลักในการแทนค่าความน่าจะเป็น ซึ่งตัวเลข Random Number ที่ใช้ทั้งหมดจะมีค่าตั้งแต่ 0000-9999 ซึ่งมีจำนวน 10,000 ตัว แล้วนำค่าความน่าจะเป็นที่หาได้มาคูณกับ 10000 จะได้เท่ากับ 3 แสดงว่าต้องใช้ Random Number จำนวน 3 ค่าในการแทนค่าความน่าจะเป็นที่ $n=0$ ซึ่งจะได้ช่วงค่า Random Number เท่ากับ 0000-0002

5. การ Generate ค่า Random Number โดยใช้ System Clock ของเครื่องคอมพิวเตอร์เพื่อให้ค่า Random Number แต่ละชุดออกมาไม่ซ้ำกัน ดังรูปที่ 10-3



รูปที่ 10-3 แสดงขั้นตอนการ Generate Random Number

6. Binary Search Algorithmหาค่า Random Number ที่ได้แต่ละตัวมาทำการเปรียบเทียบกับช่วงของ Random Number ที่ได้จากการคำนวณในข้อ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลือกค่า n ที่อยู่ในช่วงของ Random Number ที่ได้จากข้อ 6
8. นำค่า n ที่ได้ในแต่ละครั้งมาหาค่าเฉลี่ยของจำนวนงานในระบบ
9. หาค่าเฉลี่ยของจำนวนงานที่อยู่ในคิว ค่าเฉลี่ยของเวลาที่รอคอยในคิว ค่าเฉลี่ยของเวลาทั้งหมดที่ใช้ในระบบ ความน่าจะเป็นของการเกิด Blocking ค่าการกระจายของข้อมูล และค่าเฉลี่ยของข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] วีระศักดิ์ ชิงฉาว, “Java Programming Volume 1”, ซีเอ็ดยูเคชั่น, 432 หน้า. 2543.
- [2] วีระศักดิ์ ชิงฉาว, “Java Programming Volume 2”, ซีเอ็ดยูเคชั่น, 432 หน้า. 2545.
- [3] H. M. Deitel and P.J. Deitel (2003): “Java How to Program Fifth Edition”, Prentice Hall, 1447 p. 2003.
- [4] Averill M. Law and W. David Kelton (2000): “Simulation Modeling and Analysis Third Edition”, Mc Graw Hill, 745 p. 2000.
- [5] Toh, C.-K: “Ad hoc mobile wireless networks: protocols and systems”, Prentice Hall PTR, 302p. 2002.
- [6] <http://www.isi.edu/nsnam/ns/>
- [7] <http://www.mhhe.com/engcs/industrial/lawkelton/>
- [8] <http://www.j-sim.org/>
- [9] <http://www.cs.pitt.edu/~melhem/Mobility/>
- [10] <http://www.thaicert.nectec.or.th/>
- [11] <http://www.topology.org/soft/sim.html>
- [12] <http://www.sims.berkeley.edu/~christin/index.html>
- [13] <http://renaud.waldura.com>
- [14] <http://www.ensc.sfu.ca/cnl>
- [15] <http://www.antd.nist.gov/>
- [16] <http://moment.cs.ucsb.edu/>
- [17] <http://www.staffs.ac.uk/airnet/>
- [18] <http://www.ietf.org/rfc/rfc3561.txt>