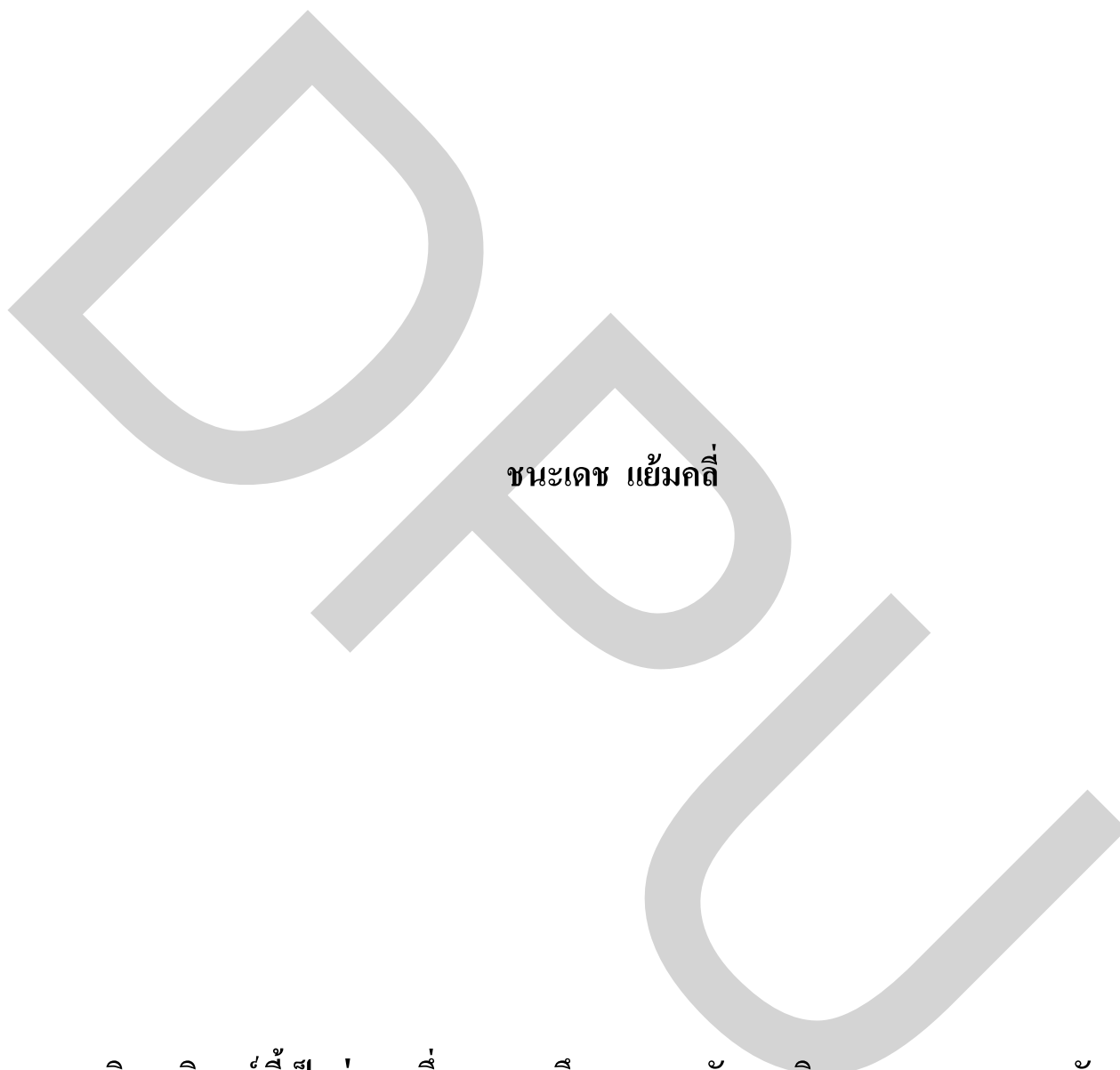


เครือข่ายเซนเซอร์ไร้สาย IEEE802.15.4 แบบหลายเกตเวย์



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์

มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2556

Multiple Gateways System for Wireless Sensor Network IEEE802.15.4



Chanadej Yamclee

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Computer and Telecommunication Engineering

Faculty of Engineering, Dhurakij Pundit University

2013

หัวข้อวิทยานิพนธ์	เครือข่ายเซนเซอร์ไร้สาย IEEE802.15.4 แบบหลายเกตเวย์
ชื่อผู้เขียน	ชนะเดช แยมกล้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์	อาจารย์ ดร.ชัยพร เขมะภาคะพันธ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา	2556

บทคัดย่อ

วิทยานิพนธ์นี้เสนอการเพิ่ม โหนด โคออดิเนเตอร์ที่ทำหน้าที่เป็นเกตเวย์สำรองในเครือข่ายตรวจจับ ไร้สายเพื่อสร้างช่องทางสำรองในกรณีที่เกิดเกตเวย์หลักไม่สามารถทำงานได้ โดยปรับปรุงการทำงานของโพรโทคอลการค้นหาเส้นทางแบบ AODV ให้สามารถทำงานดังกล่าวได้ โดยทำการจำลองการทำงานด้วยโปรแกรม NS2

ผลการทดสอบ พบว่าวิธีการที่เสนอทำให้ประสิทธิภาพของเครือข่ายไร้สายดีขึ้นดังนี้ ค่า Throughput มีค่าสูงขึ้นเฉลี่ย 6.44 kbps ค่า Packet Delivery Ratio (PDR) มีค่าสูงขึ้นเฉลี่ย 43.80% ค่า Packet Drop มีค่าต่ำลงเฉลี่ย 43.80% และค่า Packet Delay มีค่าไม่ต่างกันเมื่อเทียบกับเครือข่ายเซนเซอร์ไร้สายแบบดั้งเดิม อย่างไรก็ตามการปรับเปลี่ยนการทำงานของโพรโทคอล AODV ทำให้มีระยะเวลาในการเปลี่ยนเกตเวย์เพิ่มขึ้นมากขึ้นเล็กน้อยและมีค่าไม่แตกต่างกันมากนัก เมื่อ Packet Rate มีอัตราต่างๆ

Thesis Title	Multiple Gateways System for Wireless Sensor Network IEEE802.15.4
Author	Chanadej Yamclee
Thesis Advisor	Chaiyaporn Khemapatapan, Ph.D.
Department	Computer and Telecommunication Engineering
Academic Year	2013

ABSTRACT

This thesis proposes adding a coordinator node which acts as a redundant gateway into wireless sensor network or WSN. The redundant gateway will immediately serve WSN when the main gateway cannot operate. The propose method modifies the operation of routing protocol AODV. This study simulated the testing scenarios using program NS2.

The testing results found that the proposed method will improve the performance of WSN as following: Throughput of WSN increases about 6.44 kbps, Packet Delivery Ratio (PDR) increases about 43.80% and Packet Drop decreases about 43.80%. In addition, Packet Delay of the proposed method is not different in comparison with conventional WSN. However, the modification of protocol AODV will add a slightly duration of changing gateway which is not different for various Packet Rates.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาของ อาจารย์ ดร.ชัยพร เขมะภาคะพันธ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำปรึกษา เสนอแนะ และความช่วยเหลือหลายสิ่ง ทำให้ทุกอย่างสำเร็จได้ด้วยดี ขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ด้วย

ขอกราบขอบคุณกรรมการสอบวิทยานิพนธ์ อาจารย์ ดร.ประศาสน์ จันทราทิพย์ อาจารย์ ดร.ลัญจนกร วุฒิสัทติกุลกิจ และ อาจารย์ ดร.วรพล พงษ์เพชร ที่สละเวลามาเป็นกรรมการสอบวิทยานิพนธ์ และได้ให้ข้อเสนอแนะ ให้ข้อคิดเห็น รวมทั้งแนวทางการปรับปรุงแก้ไขงานวิจัยนี้ให้มีความสมบูรณ์มากที่สุด และขอขอบคุณเพื่อนนักศึกษาช่วยตรวจทานการเขียนวิทยานิพนธ์

ท้ายสุดนี้ประโยชน์และความดีของวิทยานิพนธ์ฉบับนี้ ขอมอบแต่ผู้มีพระคุณทุกท่านที่ให้ความช่วยเหลือและเสริมสร้างกำลังใจ กำลังใจให้ จนการทำวิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี

ชนะเดช เข้มกล้า

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ฉ
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญตาราง.....	ช
สารบัญภาพ.....	ญ
ประมวลศัพท์และคำย่อ.....	ฒ
บทที่	
1. บทนำ.....	1
1.1 ที่มาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง.....	4
2.1 เครือข่ายไร้สาย.....	4
2.2 การประยุกต์ใช้งาน WSN.....	6
2.3 คุณลักษณะของ WSN.....	7
2.4 มาตรฐานและคุณสมบัติของ WSN.....	8
2.5 โครงสร้างเครือข่ายเซนเซอร์ไร้สาย ZigBee.....	9
2.6 โครงสร้าง LR-WPAN	11
2.7 พลังงานที่ใช้ใน WSN.....	12
2.8 กระบวนการเข้าถึงช่องสัญญาณสื่อสารใน WSN.....	13
2.9 MAC ใน WSN.....	15
2.10 ยานความถี่ที่ใช้งานใน IEEE802.15.4.....	17
2.11 โพรโทคอลการค้นหาเส้นทางใน WSN.....	19
2.12 โพรโทคอล AODV.....	21
2.13 โปรแกรมจำลองการทำงานของเครือข่าย NS2.....	26
2.14 งานวิจัยที่เกี่ยวข้อง.....	27

สารบัญ (ต่อ)

บทที่	หน้า
3. ระเบียบวิธีวิจัย.....	37
3.1 แนวทางการวิจัยและพัฒนา.....	37
3.2 เครื่องมือที่ใช้ในงานวิจัย.....	39
3.3 ขั้นตอนการดำเนินงานวิจัย.....	39
4. การทดลองและผลการทดลอง.....	70
4.1 หาค่าพลังงานของเกตเวย์สำรองที่ใช้ไปและเวลาที่ใช้ในการเปลี่ยนเกตเวย์....	70
4.2 หาประสิทธิภาพของเครือข่าย WSN แบบ 2 เกตเวย์เทียบกับ 1 เกตเวย์.....	80
5. สรุปผลการวิจัยและข้อเสนอแนะ.....	94
5.1 สรุปผลการวิจัย.....	94
5.2 ปัญหาและข้อเสนอแนะ.....	94
บรรณานุกรม.....	96
ภาคผนวก.....	99
ประวัติผู้เขียน.....	107

สารบัญตาราง

ตารางที่	หน้า
2.1 ค่าการตั้งค่าพารามิเตอร์ในโปรแกรม NS2.....	31
3.1 แผนการดำเนินงาน.....	38
3.2 การตั้งค่าต่างๆ บน Topology แบบ 1 Coordinator/Gateway.....	41
3.3 การตั้งค่าต่างๆ บน Topology แบบ 2 Coordinator/Gateway.....	42
3.4 ชื่อและค่าตัวแปรที่กำหนด.....	45
3.5 ค่าปริมาณความหนาแน่นของข้อมูลต่อเวลา.....	55
3.6 ค่าที่ใช้ในการกำหนดค่า Packet Interval.....	55
3.7 ผลการทดลองเบื้องต้นใน WSN แบบ 1 เกตเวย์.....	60
3.8 ผลการทดลองเบื้องต้นใน WSN แบบ 2 เกตเวย์.....	61
3.9 จำนวน packets ที่ใช้ในการทดลอง.....	63
4.1 จำนวน packets ที่ใช้ในการทดลอง.....	70
4.2 ผลการทดลองเมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้.....	76
4.3 ผลการทดลองเมื่อเกตเวย์ 1 เคลื่อนที่ออกจากเครือข่าย.....	79
4.4 เปรียบเทียบค่าเวลาการตอบสนอง Error ของเครือข่าย WSN แบบ 2 เกตเวย์....	80
4.5 ค่าการหน่วงเวลา packets delay ของแบบเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย	81
4.6 ค่าการหน่วงเวลา packets delay ของแบบเกตเวย์ 1 เสียหาย.....	82
4.7 ค่าความสำเร็จในการส่งข้อมูล %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ โดย ที่มีเกตเวย์ 1 เกิดการเสียหาย.....	83
4.8 ค่าความสำเร็จในการส่งข้อมูล %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ โดย ที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย.....	84
4.9 ค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เสียหาย.....	86
4.10 ค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย.....	87
4.11 ปริมาณ Throughput (kbps) กรณีเกตเวย์ 1 เสียหาย.....	89
4.12 ปริมาณ Throughput (kbps) กรณีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย.....	90

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.13 ปริมาณข้อมูลของเครือข่ายแบบ 2 เกตเวย์ ที่รับได้เทียบกับเวลา (Throughput kbps) โดยเปรียบเทียบกรณีเกตเวย์ 1 เสียหาย และกรณีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย.....	92

สารบัญภาพ

ภาพที่	หน้า
2.1 แผนผังแสดงมาตรฐาน IEEE802.....	4
2.2 โครงสร้างเครือข่ายของ ZigBee	10
2.3 กลไกการทำงาน CSMA/CA.....	11
2.4 โครงสร้างของ LR-WPAN.....	12
2.5 การเข้าถึงช่องสัญญาณของ CSMA/CA.....	12
2.6 โครงสร้างเฟรมของ TDMA.....	14
2.7 การทำงานของ S-MAC.....	16
2.8 การเปรียบเทียบการทำงานของ S-MAC และ T-MAC.....	17
2.9 ค่าย่านความถี่ที่ใช้ใน IEEE802.15.4-2006.....	17
2.10 ค่าความถี่มูลฐานของแต่ละช่องสัญญาณ.....	18
2.11 ค่าความถี่ 2.4GHz ที่ทับซ้อนกันของ IEEE802.15.4 กับ IEEE802.11.....	18
2.12 โพรโทคอลการค้นหาเส้นทางที่ใช้ในเครือข่ายเฉพาะกิจ.....	20
2.13 กระบวนการค้นหาเส้นทางไปยังโหนดปลายทาง (RREQ)	23
2.14 การตอบข้อความกลับ (RREP) ไปยังโหนดต้นทาง.....	24
2.15 การทำงานของผู้ใช้งานผ่าน โปรแกรม NS2.....	27
2.16 โครงสร้างของ Multiple Gateway System.....	27
2.17 โครงสร้างของ Terminal Gateway.....	28
2.18 (a) Circular Topology	
(b) An example of the multipath ruting in subnet ZigBee cluster-tree.....	30
2.19 (a) Packet delivery fraction	
(b) Number of intra-path and inter-path interference in the different routing strategies.....	32
2.20 (a) Number of collisions at the sink	
(b) Number of intra-path collisions in the different routing strategies.....	33
2.21 (a) Average end to end delay in the different routing at time: 0.05s to 1s	
(b) at time: 0.005s to 2s.....	34

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.22 (a) Network lifetime	
(b) Throughput in the different routing strategies.....	35
2.23 การทำงานของ IEEE802.15.4 MAC super frame.....	36
3.1 โครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบมี 1 Gateway.....	40
3.2 โครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบมี 2 Gateways	42
3.3 คำสั่งการใช้โปรแกรม Network Simulator 2.35 เพื่อสั่งงาน Otcl Script.....	43
3.4 แผนผังการกำหนดค่าในโปรแกรมภาษา Otcl Script สำหรับโมเดลจำลอง.....	44
3.5 โครงสร้าง WSN และระยะห่างระหว่างโหนดต้นทาง กับโหนดปลายทาง.....	46
3.6 การส่งข้อมูลระหว่างโหนด Source กับ โหนด Destination ใน NS2.....	47
3.7 ช่วงเวลาในส่ง CBR Packet กรณีที่มี 1 Gateway.....	47
3.8 ลำดับขั้นตอนการส่งข้อมูลของตัวโพรโทคอล AODV.....	49
3.9 โปรแกรม Network Animation จำลองการทำงานของ WSN 1 Gateway.....	50
3.10 โปรแกรม Network Animation จำลองการทำงานของ WSN 2 Gateways.....	50
3.11 โครงสร้างของ Trace ไฟล์ ที่สร้างขึ้น.....	51
3.12 ข้อมูลใน trace file หลังจากแบ่งข้อมูลแต่ละ column แล้ว.....	53
3.13 กราฟของค่าเฉลี่ยการหน่วงเวลาแบบจุดต่อจุด.....	54
3.14 ตัวอย่างการกำหนดค่าของ CBR Packet.....	55
3.15 รหัสเทียม (Pseudo Code) การหาจำนวน CBR Packet ที่โหนดต้นทาง.....	56
3.16 รหัสเทียม (Pseudo Code) การหาจำนวน CBR Packet ที่โหนดปลายทาง.....	56
3.17 รหัสเทียม (Pseudo Code) การหาค่าเฉลี่ย CBR Packet ต่อเวลา ที่โหนดต้น ทาง.....	57
3.18 รหัสเทียม (Pseudo Code) การหาค่าเฉลี่ย CBR Packet ต่อเวลา ที่โหนด ปลายทาง.....	57
3.19 รหัสเทียม (Pseudo Code) การหาค่าค่าสัดส่วนของค่าเฉลี่ย CBR Packet ต่อ เวลาระหว่างโหนดต้นทาง กับโหนดปลายทาง.....	58
3.20 ช่วงเวลาในส่ง CBR Packet กรณีที่มี 2 Gateways.....	59

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.21 การปรับปรุงโปรแกรมกรณีที่มี 2 Gateways.....	59
3.22 ค่าเปรียบเทียบ Throughput ระหว่าง WSN แบบ 1 เกตเวย์ กับ 2 เกตเวย์.....	61
3.23 ค่า PDR ระหว่าง WSN แบบ 1 เกตเวย์ กับ 2 เกตเวย์.....	62
3.24 เครื่องข่ายเซนเซอร์ไร้สาย แบบ 2 เกตเวย์ ที่ใช้ในการทดลอง.....	63
3.25 เครื่องข่ายที่ใช้ทดลองแบบเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากรับความเสียหาย.....	64
3.26 เครื่องข่ายที่ใช้ทดลองแบบเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากเคลื่อนที่ออกจากเครื่องข่าย.....	64
3.27 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC.....	65
3.28 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC เพื่อเชื่อมโยงข้อมูล.....	66
3.29 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC.....	66
3.30 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC เพื่อเชื่อมโยงข้อมูล.....	67
3.31 ส่วนของโปรแกรมที่พัฒนาโดยภาษา TCL เพื่อสร้าง packets CBR.....	67
3.32 ส่วนของโปรแกรมที่พัฒนาโดยภาษา TCL เพื่อกำหนดเงื่อนไขในการเปลี่ยนเกตเวย์.....	68
3.33 การส่ง Message RERR ออกไป เมื่อเกตเวย์หลักไม่สามารถทำงานได้.....	68
3.34 การส่ง Message RERR ออกไป เมื่อเกตเวย์หลักไม่สามารถทำงานได้.....	69
3.35 ขั้นตอนการรอ Message RERR เพื่อเปลี่ยนเกตเวย์.....	69
4.1 ค่าพลังงานของเกตเวย์สำรองที่ใช้ในขณะรอทำงาน.....	71
4.2 เวลาที่ใช้ในทดสอบ WSN กรณีเกตเวย์หลักหยุดทำงานแบบทันที ทันใด.....	72
4.3 เวลาที่ใช้ในทดสอบ WSN กรณีเกตเวย์หลักเคลื่อนที่ออกนอกเครื่องข่าย.....	72
4.4 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 3000.....	73
4.5 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 4000.....	73

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.6 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 5000.....	74
4.7 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 6000.....	74
4.8 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 7000.....	75
4.9 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 8000.....	75
4.10 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 3000.....	76
4.11 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 4000.....	77
4.12 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 5000.....	77
4.13 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 6000.....	78
4.14 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 7000.....	78
4.15 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 8000.....	79
4.16 เปรียบเทียบ Packets delay (ms) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เคลื่อนที่.....	81
4.17 เปรียบเทียบ Packets delay (ms) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เสียหาย.....	82
4.18 เปรียบเทียบ % PDR ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เสียหาย.....	84

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.19 เปรียบเทียบ % PDR ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เคลื่อนที่.....	85
4.20 เปรียบเทียบ % Packets Drop ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เสียหาย.....	87
4.21 เปรียบเทียบ % Packets Drop ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 มีการเคลื่อนที่.....	88
4.22 เปรียบเทียบ Throughput (kbps) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เสียหาย.....	90
4.23 เปรียบเทียบ Throughput (kbps) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เคลื่อนที่.....	91
4.24 เปรียบเทียบ Throughput (kbps) ของระบบ 2 เกตเวย์ ในกรณีที่เกิดเกตเวย์ 1 เสียหาย กับ เกตเวย์ 1 เคลื่อนที่.....	92

ประมวลศัพท์และคำย่อ

ABR	Associativity Based Routing
AODV	Ad-Hoc Distance Vector
ARPAM	Aeronautical Mobile Ad hoc Networks
BEMS	Building Energy Management System
B-MAC	Beamformed MAC
CBR	Constant Bit Rate
CLH	Cluster Head
CLI	Cluster Identifier
C-MAC	Cluster Based MAC
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Request to Send
DARPA	Defense Advanced Research Projects Agency
D-MAC	Directional MAC
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
FFD	Full-function device
FIFO	First In First Out
GHz	Giga Hertz
GSR	Global State Routing
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISI	Information Sciences Institute
Kbps	Kilobit per second
LLC	Logical Link Control
LR-WPAN	Low-Rate Wireless Personal Area Networks
MAC	Media Access Control

ประมวลศัพท์และคำย่อ (ต่อ)

MANET	Mobile Ad Hoc Network
Mbps	Megabit per second
MHz	Mega Hertz
NAM	Network Animation
NS2.35	Network Simulation 2.35
OLSR	Optimized Link State Routing
OORP	Order One MANET Routing Protocol
OSI model	Open Systems Interconnection model
PAN	Personal Area Network
PAN ID	Personal Area Network Identification
PDR	Packet Delivery Ratio
PCS-MAC	Physical Carrier Sense MAC
PHY	Physical
P-MAC	Parallel MAC
PPS	Packets Per Second
RERR	Route Error
RF	Radio Frequencies
RFC3561	Request for Comments 3561
RFD	Reduced-function device
RREP	Route Reply
RREP-ACK	Route Reply Acknowledgment
RREQ	Route Request
RTP	Real-time Transport Protocol
RTS	Clear to Send
S-MAC	Sensor Media Access Control
SPOF	Single Point of Failure
SRM	Scalable Reliable Multicast
SSCS	Service-specific Convergence Sub layer

ประมวลศัพท์และคำย่อ (ต่อ)

TA	Adaptive Timeout
TBRPF	Topology Dissemination Based on Reverse-Path Forwarding
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
T-MAC	Timeout Media Access Control
TTL	Time to live
UDP	User Datagram Protocol
U-MAC	Universal Media Access Control
Wise-MAC	Wireless Sensor MAC
WRP	Wireless Routing Protocol
WSN	Wireless Sensor Network
Z-MHTR	ZigBee Multi-Path Hierarchical Tree Routing
ZRP	Zone Routing Protocol

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

ปัจจุบันระบบเครือข่ายเซ็นเซอร์ไร้สายหรือ Wireless Sensor Network (WSN) ได้มีการกล่าวถึงในวงการวิจัยมากขึ้น อีกทั้งยังมีการนำเอา WSN มาประยุกต์ใช้งานจริงในภาคส่วนต่างๆ เช่น ทางด้านการทหาร ได้มีการนำใช้ตรวจจับการสั่นสะเทือนเมื่อมีการโจมตีด้วยอาวุธหนักภายในอาณาเขตที่ได้ติดตั้งเซ็นเซอร์ไว้ หรือตรวจจับการเคลื่อนพลของฝ่ายตรงข้ามที่ผ่านเข้ามาในบริเวณที่กำหนดไว้ ทางด้านธรณีวิทยา ได้มีการนำมาใช้ในการติดตั้งตัวเซ็นเซอร์ไว้ที่ปากปล่องภูเขาไฟ เพื่อตรวจสอบการสั่นสะเทือน ก่อนที่จะเกิดการระเบิดของภูเขาไฟ แม้กระทั่งทางการสำรวจมหาสมุทร ก็ได้มีการติดตั้งเซ็นเซอร์ เพื่อตรวจสอบตัวแปรทางด้านฟิสิกส์ต่างๆ เช่น อุณหภูมิ ความดัน ที่ระดับความลึกต่างๆ ซึ่งทั้งหมดนี้เพื่อการนำเอาข้อมูลทางฟิสิกส์ต่างๆ ที่ได้จากการตรวจจับหรือตรวจวัด จากเครือข่ายเซ็นเซอร์มาประมวลผลวิเคราะห์ที่ส่วนกลาง หรือใช้เป็นข้อมูลทางสถิติเพื่อแบ่งปันให้กับผู้ที่สนใจ ดังนั้น จากตัวอย่างที่ยกมาจะเห็นได้ว่า บริเวณต่างๆ ที่มีการติดตั้งเครือข่ายเซ็นเซอร์นั้น จะเป็นพื้นที่ ที่มีสภาพภูมิศาสตร์ไม่เหมาะกับการเดินสายไฟเพื่อสื่อสารกันระหว่างตัวเซ็นเซอร์ หรือตัวเซ็นเซอร์กับศูนย์กลางการเก็บข้อมูล ดังนั้น WSN จึงเป็นตัวเลือกที่ดีสำหรับการประยุกต์ใช้งานเหล่านี้

ใน WSN ที่ถูกติดตั้งนั้น โดยปกติจะประกอบไปด้วยโหนดอยู่ 3 โหนด โหนดที่ 1 คือ เซ็นเซอร์โหนด เพื่อใช้เป็นตัวรับอินพุตจากตัวเซ็นเซอร์หรือทรานสดิวเซอร์ชนิดต่างๆ เช่น เทอร์โมคัปเปิ้ล อาร์ทีดี สำหรับการตรวจวัดอุณหภูมิ สเตรนเกจ สำหรับการตรวจจับความเครียด ความเค้นในวัตถุ และสามารถเป็นตัวส่งเอาต์พุต เพื่อไปควบคุมอุปกรณ์ทางไฟฟ้า เช่น รีเลย์ โหนดที่ 2 คือเรทเตอร์โหนด มีหน้าที่การทำงานเหมือนกับเซ็นเซอร์โหนด แต่มีสามารถในการค้นหาเส้นทางเพื่อส่งผ่านข้อมูลที่ได้จากเซ็นเซอร์โหนด ไปยังโคออดิเนเตอร์โหนด และโหนดสุดท้ายคือโคออดิเนเตอร์โหนดหรือเกตเวย์โหนด ทำหน้าที่คอยรวบรวมข้อมูลที่ได้จากโหนดเรทเตอร์

ทั้งหมดใน WSN เพื่อนำไปประมวลผลที่ส่วนกลาง นอกจากนี้ยังทำหน้าที่คอยควบคุม จัดการและ แจกจ่ายหมายเลขประจำตัวให้กับโหนดต่างๆ ภายใน WSN ดังนั้นจะเห็นได้ว่าถ้าเกตเวย์โหนดใน เครือข่ายมีปัญหาไม่สามารถทำงานได้ตามฟังก์ชันแล้ว ก็จะทำให้ใน WSN ไม่สามารถติดต่อกับ ส่วนกลางการประมวลผลได้ จึงหมายความว่าใน WSN ที่ถูกติดตั้งนั้นจะไม่มีประโยชน์ใดๆ เลย เนื่องจากไม่สามารถนำข้อมูลที่ต้องการออกมาได้ หรืออาจพูดได้ว่าตัวเกตเวย์โหนด เป็น SPOF ของเครือข่าย

ในงานวิจัยนี้จึงนำเสนอการเพิ่มจำนวนตัวเกตเวย์โหนดเข้าไปใน WSN ซึ่งแต่เดิม สามารถมีได้แค่ 1 ตัว ในเครือข่าย ดังนั้นการเพิ่มตัวเกตเวย์โหนดเข้าไปใน WSN นั้น สามารถที่จะ ช่วยแก้ปัญหาการเกิด SPOF ที่ตัวเกตเวย์โหนดใน WSN ได้

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อสร้างเส้นทางสำรองให้กับข้อมูลที่ต้องการออกจากเครือข่ายเซนเซอร์ไร้สายตาม มาตรฐาน IEEE802.15.4
2. เพื่อหาแนวทางในการเพิ่มตัวเกตเวย์จาก 1 ตัว เป็น 2 ตัว ภายใน WSN
3. ศึกษาโปรโตคอลเส้นทาง AODV เพื่อใช้งานใน WSN ที่มีเกตเวย์ 2 ตัว
4. เพื่อหาประสิทธิภาพของการส่งข้อมูลประเภท CBR ที่ส่งไปยังเกตเวย์โหนด โดยใช้โปร- โทคอล AODV ที่ได้ปรับปรุงขึ้น ภายใน WSN

1.3 ขอบเขตของงานวิจัย

1. มีจำนวนเกตเวย์โหนดใน WSN ทั้งหมด 2 โหนด
2. ใช้โปรแกรมจำลองเครือข่าย NS2.35 ในการทดสอบโปรโตคอล AODV ใน WSN
3. โหนดใน WSN ทั้งหมด เป็นแบบอยู่กับที่ ไม่มีการเคลื่อนที่
4. การหาประสิทธิภาพของการส่งข้อมูลเป็นแบบ End to End โดยใช้ข้อมูลประเภท CBR

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. แก้ปัญหาการเกิด SPOF ที่ตัวเกตเวย์โหนด
2. เมื่อมีข้อมูลต้องการออกจาก WSN สามารถส่งออกได้ในกรณีที่มีเกตเวย์โหนดล้ม 1 โหนด
3. สามารถพัฒนา หรือ ปรับปรุง โพรโทคอลเส้นทางใน WSN เพื่อเพิ่มประสิทธิภาพในการส่งข้อมูล

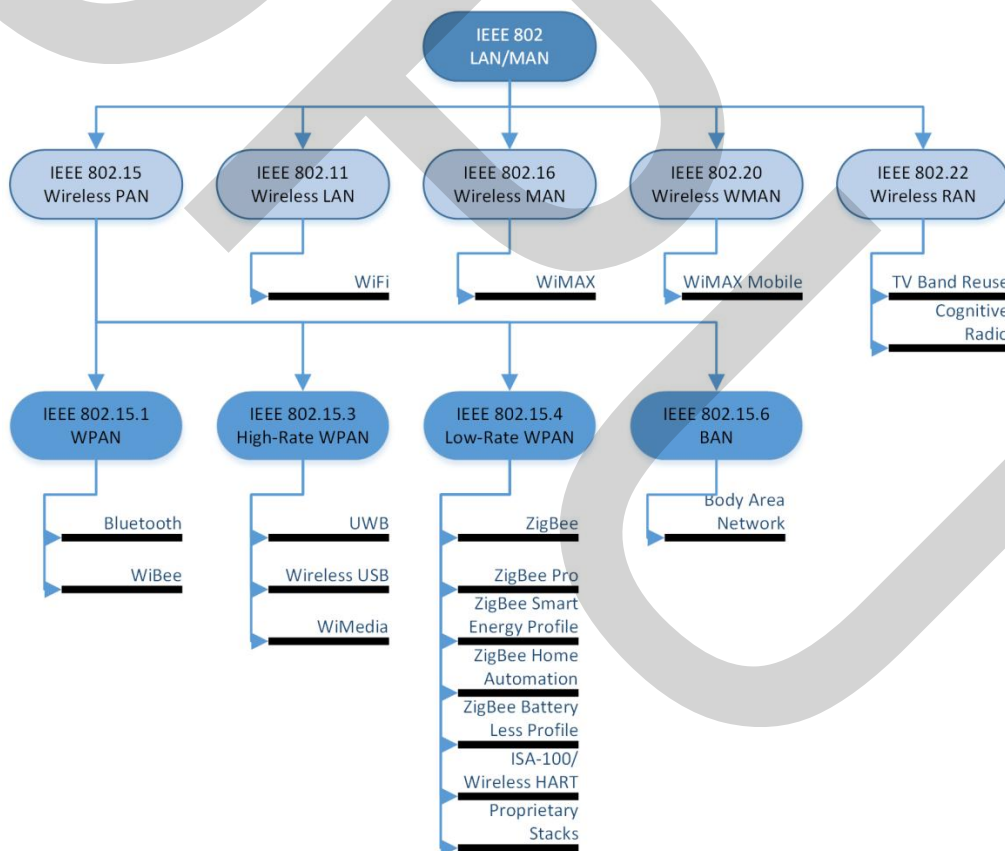
บทที่ 2

ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้อง

2.1 เครือข่ายไร้สาย

ปัจจุบันเครือข่ายไร้สายมีการประยุกต์ใช้งานกันมากขึ้น และมีการแบ่งแยกกลุ่มงานวิจัยและพัฒนาออกไปเป็นแต่ละด้านโดยอ้างอิงตามมาตรฐาน IEEE802 แล้ว สามารถแบ่งออกเป็นส่วนต่างๆ ได้ดังนี้



ภาพที่ 2.1 แผนผังแสดงมาตรฐาน IEEE802

ที่มา: http://www.eetasia.com/ART_8800590160_590626_NT_3e1932a3.HTM

งานวิจัยนี้จะมุ่งเน้นไปในส่วนของ IEEE802.15.4 LR-WPAN ซึ่งเป็นกลุ่มที่เน้นเรื่องของการส่งข้อมูลที่ต่ำ และการประหยัดพลังงาน ซึ่งเจาะจงไปที่ส่วนที่เป็นมาตรฐาน IEEE802.15.4 ซึ่งเป็นมาตรฐานที่มีการนำไปใช้งานใน WSN มากที่สุด ซึ่ง WSN คือเครือข่ายที่ประกอบไปด้วยโหนดเซนเซอร์ต่างๆ ที่กระจายตัวอยู่กันอย่างอิสระ เพื่อใช้ในการเฝ้าติดตามตัวแปรทางด้านฟิสิกส์ต่างๆ ในบริเวณหรือพื้นที่ที่สนใจ เช่น ความดัน ระดับเสียง อุณหภูมิ ความชื้น ความเข้มแสง แรง ความเค้น ความเครียดที่วัตถุ ฯลฯ โดยอาศัยเซนเซอร์ที่ตรงกับความต้องการในการวัด เช่น วัดอุณหภูมิ อาจจะใช้ อาร์ทีดี วัดแรงอาจจะใช้ โหลดเซลล์ และตัว WSN จะต้องสามารถส่งผ่านข้อมูลที่ไ่มายังส่วนกลางได้ ปัจจุบัน WSN สมัยใหม่ได้มีการเพิ่มความสามารถในการทำงาน โดยสามารถที่จะควบคุมอุปกรณ์ปลายทางที่ต่ออยู่กับโหนดของเครือข่ายได้ ในการพัฒนาเครือข่ายเซนเซอร์ไร้สายตั้งแต่อดีตนั้น ได้แรงจูงใจมาจากความต้องการใช้งานในทางการทหาร เช่น สนามรบ ตรวจจับการบุกรุก การเคลื่อนพล การใช้อาวุธเคมี จนถึงในปัจจุบัน ได้มีการนำมาประยุกต์ใช้ในหลากหลายอุตสาหกรรม เช่น การเฝ้าติดตามกระบวนการผลิต การติดตามและควบคุมการทำงานของเครื่องจักร การติดตามสุขภาพของผู้ป่วย

ใน WSN สามารถมีโหนดจากไม่กี่ร้อย จนกระทั่งเป็นหลักพันได้ โดยแต่ละโหนดสามารถต่อเข้ากับเซนเซอร์แต่ละชนิด ในแต่ละโหนดจะประกอบด้วยส่วนหลัก 3 ส่วน ดังนี้ ส่วนแรกภาคส่งสัญญาณวิทยุรวมถึงสายอากาศ (RF Transceiver And Antenna) ส่วนของสายอากาศนั้นมีทั้งแบบอยู่ที่ตัวโหนดหรือสายอากาศภายใน (Internal Antenna) และแบบที่ใช้ขั้วต่อสายจากภาคส่งสัญญาณวิทยุไปยังสายอากาศด้านนอก (External Antenna) ส่วนที่สอง เป็นส่วนของคอนโทรล ซึ่งโดยปกติจะใช้ไมโครคอนโทรลเลอร์ที่มีขนาดเล็กและใช้พลังงานต่ำ รวมถึงวงจรอินเตอร์เฟสที่เป็นตัวรับสัญญาณจากตัวเซนเซอร์มายังตัวไมโครคอนโทรลเลอร์ โดยปกติแล้วสัญญาณจากตัวตัวเซนเซอร์มีอยู่ 2 แบบคือสัญญาณที่เป็นแบบดิจิทัล กับสัญญาณที่เป็นแบบอนาล็อก ส่วนที่สามเป็นภาคจ่ายไฟ ซึ่งโดยปกติจะใช้แบตเตอรี่เนื่องจากโหนดต่างๆ จะถูกกระจายติดตั้งในพื้นที่ต่างๆ จึงยากที่จะใช้พลังงานจากแหล่งจ่ายไฟปกติ หรือจะใช้พลังงานไฟฟ้าจากการผลิตขึ้นที่โหนดเลยก็ได้เช่น จากไดนาโมขนาดเล็ก หรือแผงเซลล์พลังงานแสงอาทิตย์ โดยระดับแรงดันที่ใช้ในแต่ละโหนดนั้นจะขึ้นอยู่กับส่วนควบคุมว่าใช้แรงดันเท่าไร โดยปกติจะใช้อยู่ 3.3 โวลต์ดีซี

ตัวเซนเซอร์โหนดนั้นอาจจะมีขนาดที่แตกต่างกัน ซึ่งขึ้นอยู่กับงานที่ใช้ โดยต้นทุนของแต่ละโหนดมีตั้งแต่ราคาถูกจนกระทั่งราคาแพง ซึ่งตัวแปรด้านต้นทุนที่สำคัญนั้นก็คือ ความซับซ้อนของโหนด ฟังก์ชันในการทำงานของแต่ละโหนด ซึ่งปัจจัยต่างๆ เหล่านี้ไม่ว่าจะเป็น ต้นทุน หรือขนาด เป็นตัวกำหนดคุณสมบัติทางกายภาพของโหนดเซนเซอร์ เช่น อายุการใช้งาน

ของโหนดซึ่งโหนดที่ต้องการอายุการทำงานที่นาน ก็จะต้องการพลังงานในการทำงานที่สูงกว่า โหนดที่ไม่ต้องการอายุการทำงานที่ยาว ขนาดของหน่วยความจำ ความเร็วในการประมวลผล และ ความเร็วในการสื่อสารข้อมูล โดยโครงสร้าง WSN นั้น สามารถเป็นได้ทั้งแบบอย่างง่ายเช่น เครือข่ายแบบดาว (Star Network) จนกระทั่งถึงเป็นแบบที่ซับซ้อนมากๆ หรือเครือข่ายเมช (Mesh Network) โดยทั้งหมดนี้สามารถเริ่มการเชื่อมต่อระหว่างโหนดได้โดยการใช้เส้นทาง (Routing) หรือการกระจายข้อมูลออกไปทั้งหมด (Flooding)

2.2 การประยุกต์ใช้งาน WSN

การประยุกต์ใช้งานเครือข่าย WSN นั้น สามารถใช้งานได้หลากหลาย โดยขึ้นอยู่กับวัตถุประสงค์ในการใช้งาน และความต้องการพื้นฐานของระบบ ไม่ว่าจะเป็นการตรวจวัดเพื่อเก็บข้อมูลเพียงการเดียว หรือมีการควบคุมสั่งการจากระยะไกล หรือต้องการทั้งการตรวจวัดเพื่อเก็บข้อมูลและการควบคุมสั่งการจากระยะไกล ซึ่งในการตรวจวัดนั้นจะต้องหาเซนเซอร์ที่เหมาะสมกับงานที่ต้องการตรวจวัด หรือการควบคุมก็จะต้องการอุปกรณ์ควบคุมที่เหมาะสมกับงานมาใช้ โดยตัวอย่างการใช้งาน WSN ที่ได้ศึกษามามีอยู่หลากหลายซึ่งได้แสดงรายละเอียดไว้ดังนี้

2.2.1 การเฝ้าระวังพื้นที่ (Area Monitoring)

เป็นการประยุกต์ใช้งาน WSN โดยการกระจายตัวโหนดเซนเซอร์ไปยังพื้นที่ ที่เราสนใจ ตัวอย่างเช่นในงานด้านการทหาร จะมีการเฝ้าระวังพื้นที่จากการรุกรานของฝ่ายตรงข้าม เมื่อมีการรุกรานเข้ามา ก็ส่งข้อมูลกลับมาที่ส่วนกลางเพื่อรอคำสั่งในการจุดฉนวนระเบิดที่ต่ออยู่กับตัวโหนดเซนเซอร์ แต่ในการใช้งานด้านการทหารจะต้องคำนึงถึงการรักษาความปลอดภัยข้อมูลที่ถูกส่งภายในเครือข่ายเซนเซอร์ไร้สาย เพื่อป้องกันไม่ให้มีการปลอมแปลงข้อมูลที่ใช้ในการทำงานได้ ในงานด้านการสำรวจแหล่งทรัพยากรธรรมชาติ เช่นก๊าซ หรือน้ำมัน หรือทรัพยากรธรรมชาติอื่นๆ

2.2.2 การเฝ้าระวังสิ่งแวดล้อม และการเปลี่ยนแปลงของโลก

การเฝ้าระวังสิ่งแวดล้อม และการเปลี่ยนแปลงของโลก เช่น การตรวจจับการเกิดภูเขาไฟระเบิด ตรวจจับกระแส น้ำ อุณหภูมิ น้ำในมหาสมุทร การติดตามการเจริญเติบโตของสัตว์หรือ ต้นไม้ในป่า การเคลื่อนที่ของธารน้ำแข็ง ในการเฝ้าระวังสิ่งแวดล้อม และการเปลี่ยนแปลงของโลก นั้น ทำให้นักวิทยาศาสตร์ สามารถเก็บข้อมูลตัวอย่างที่ต้องการใช้ในงานวิจัยโดยข้อมูลที่ได้เป็นข้อมูลแบบต่อเนื่องทางเวลา ทำให้สามารถนำข้อมูลเหล่านั้นมาใช้ในการประกอบการตัดสินใจต่างๆ ได้ อีกทั้งยังประหยัดงบประมาณในการเดินทางเพื่อเข้าไปเก็บข้อมูลในพื้นที่ห่างไกล และมีความยากลำบากในการเข้าถึง

2.2.3 การเฝ้าระวังคุณภาพของอากาศ

เนื่องจากมลพิษทางอากาศเริ่มมีปริมาณมากขึ้นในปัจจุบัน จึงต้องมีการตรวจวัดกันเป็นประจำ เพื่อปกป้องประชากรและสิ่งแวดล้อมอื่นๆ อันเนื่องมาจากมลพิษทางอากาศ ซึ่งเป็นภัยอันตรายที่อยู่รอบๆ ตัวเรา ดังนั้นการเฝ้าระวังแบบตลอด (Real time monitoring) นั้นมีความสำคัญมาก เนื่องจากสภาวะอากาศมีการเปลี่ยนแปลงที่รวดเร็วมาก

2.2.4 การเฝ้าระวังมลพิษทางอากาศ

ปัจจุบัน WSN มีการติดตั้งใช้งานในหลายๆ เมืองเช่น ลอนดอน เพื่อเฝ้าระวังแก๊สที่เป็นอันตรายต่อพลเมืองของเขา โหนดเซนเซอร์เหล่านี้ใช้ประโยชน์จากเครือข่ายไร้สาย แทนที่จะเป็นแบบสาย เนื่องจากสามารถเคลื่อนที่ไปเฝ้าระวังในจุดต่างๆ ได้

นอกจากการประยุกต์ใช้งาน WSN ที่ได้กล่าวและยกตัวอย่างมาแล้ว ยังมีการใช้งานอื่นๆ อีกเช่น การเฝ้าระวังการเคลื่อนที่ของหน้าดิน (Landslide detection) การเฝ้าระวังคุณภาพของน้ำ (Water quality monitoring) การเฝ้าระวังภัยพิบัติทางธรรมชาติ (Natural disaster prevention) การเฝ้าระวังในภาคอุตสาหกรรม (Industrial monitoring) การเก็บข้อมูลต่าง (Data logging) การติดตามการทำงานของเครื่องจักร (Machine health monitoring) การเฝ้าระวังน้ำเสีย (Waste water monitoring) การเฝ้าระวังในภาคเกษตรกรรม (Agriculture monitoring) การเฝ้าระวังในบ้าน (Smart home monitoring) จะเห็นได้ว่า WSN เริ่มเข้ามามีบทบาทในชีวิตประจำวันมากขึ้น จึงได้มีการศึกษาและพัฒนาขีดความสามารถของ WSN ให้สูงขึ้น เพื่อตอบโจทยความต้องการที่มากขึ้น

2.3 คุณลักษณะของ WSN

- 1) โหนดต่างๆ ในเครือข่ายจะใช้พลังงานจากแบตเตอรี่ หรือจากการผลิตขึ้นมาด้วยตัวเอง
- 2) มีการใช้ MAC ที่เน้นเรื่องการประหยัดพลังงาน
- 3) เครือข่ายยังสามารถทำงานได้เมื่อมีบางโหนดเสียหาย
- 4) โหนดจะต้องสามารถเคลื่อนย้ายได้
- 5) แต่ละโหนดมีลักษณะการทำงานเป็นลำดับขั้น
- 6) เครือข่ายต้องรองรับการขยาย หรือเพิ่มจำนวนของโหนด
- 7) สามารถทนอยู่ได้ภายในสภาพแวดล้อมที่เลวร้าย
- 8) สามารถต่อกับเซนเซอร์ต่างๆ ได้
- 9) ง่ายต่อการใช้งาน

ตัวโหนดภายใน WSN เปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กที่สามารถรับสัญญาณอินพุตและส่งสัญญาณเอาต์พุตผ่านทางอินเทอร์เน็ตเฟสต่างๆ โดยในการติดต่อสื่อสารนั้นจะใช้สัญญาณวิทยุ และติดต่อกับตัวเกตเวย์ เมื่อต้องการส่งข้อมูลออกนอกเครือข่าย

2.4 มาตรฐานและคุณสมบัติของ WSN

ใน WSN มีหลายสถาบันที่ช่วยกันพัฒนา ซึ่ง IEEE เป็นสถาบันที่มุ่งเน้นที่ชั้นที่ 1 กายภาพ (Physical layer) และชั้นที่ 2 ชั้นสื่อสารข้อมูล (Data link) ส่วน IETF จะมุ่งเน้นไปที่ชั้นที่ 3 ชั้นเครือข่าย (Network layer) และชั้นที่สูงกว่าขึ้นไป ตามการจัดเรียงใน OSI model

มาตรฐาน IEEE802.15.4 เป็นมาตรฐานที่เกี่ยวกับโครงสร้างของเครือข่ายสำหรับเครือข่ายเซนเซอร์ไร้สาย ตัวมาตรฐานเน้นไปที่ชั้นของกายภาพ (Physical layer) และชั้นเครือข่าย (MAC layer) สำหรับการประยุกต์ใช้งานเครือข่ายเซนเซอร์ไร้สายนั้น หลักในการออกแบบนั้นต้องการให้ตอบโจทย์เรื่องของการใช้พลังงานในการทำงานให้น้อยที่สุด ต้นทุนที่ต่ำ โหนดแต่ละโหนดครอบคลุมพื้นที่ขนาดเล็ก แต่มีการใช้เครือข่ายเมช (Mesh network) ในกรณีที่ต้องการติดต่อสื่อสารเป็นบริเวณกว้างและครอบคลุมโหนดจำนวนมาก โครงสร้างของเครือข่าย IEEE802.15.4 จะเริ่มจากลักษณะของเครือข่าย เช่น แบบดวงดาว โดยโหนดในเครือข่ายจะมีอยู่ 1 โหนด ที่จะคอยทำหน้าที่เป็นโหนดประสานงาน (Node Coordinator) ซึ่งจะทำหน้าที่ระบุช่องการสื่อสารที่ไม่ทับซ้อนกันเพื่อป้องกันการรบกวนกันของคลื่นความถี่ (Interference) โดยการสแกนหาระดับของพลังงานแต่ละช่องที่รับได้ ซึ่งตัวโหนดใดๆ ที่อยู่ในช่องสัญญาณเดียวกันกับตัวโหนดประสานงาน (Node Coordinator) และต้องการจะเชื่อมต่อ ก็จะทำการส่งสัญญาณแจ้งเตือนออกมา ร้องขอ (Beacon request) เพื่อขอเข้าร่วมเครือข่าย มีเพียงตัวโหนดประสานงาน (Node coordinator) เท่านั้นที่จะตอบสนองต่อการร้องขอนั้นๆ ซึ่งเรียกกระบวนการนี้ว่า แอสโซซิเอชัน (Association) และมี PAN Coordinator เป็นตัวประสานงาน ในการคุยกับตัวโหนดประสานงานอื่นๆ ที่อยู่ใน PAN เดียวกัน ในกรณีที่มี โหนดประสานงานมากกว่า 1 โหนด ซึ่งโหนด PAN Coordinator นอกจากการทำหน้าที่คอยประสานงานแล้ว ยังทำหน้าที่เป็นเกตเวย์ เพื่อเชื่อมต่อกับเครือข่ายอื่นๆ เพื่อให้ง่ายต่อการทำความเข้าใจในงานวิจัยนี้ จึงขอแทนตัวโหนด PAN Coordinator ว่า เกตเวย์โหนด ในมาตรฐาน IEEE802.15.4 นั้น เมื่อมีโหนดใดๆ เกาะกับตัวโหนด แล้ว โหนดเหล่านั้นสามารถที่จะกลายเป็นตัวโหนดเราเตอร์ (Node router) ได้ด้วยตัวเอง และสามารถยอมให้โหนดอื่นๆ เกาะกับตัวมันได้ต่อ ผลที่ได้คือทำให้เกิดการทำแอสโซซิเอชัน (Association) แบบหลายลำดับชั้น

โครงสร้างของเครือข่าย ZigBee นั้นจะประกอบไปด้วยโหนดต่างๆ จำนวนมากแต่สามารถแบ่งออกเป็นสองแบบหลักคือ โหนดที่มีฟังก์ชันการทำงานแบบเต็ม Full-function device (FFD) และโหนดที่ถูกตัดฟังก์ชันที่ไม่จำเป็นออก Reduced-function device (RFD) ซึ่งในเครือข่ายจะต้องประกอบไปด้วย FFD อย่างน้อย 1 ตัว เพื่อทำหน้าที่เป็น PAN coordinator ตัวโหนดที่เป็น FFD สามารถทำงานได้ 3 รูปแบบ ดังนี้

- 1) PAN coordinator node
- 2) Coordinator node
- 3) Device node

ตัวโหนดที่เป็น RFD จะถูกกำหนดเพื่อทำหน้าที่ในการติดต่อกับส่วนที่เป็นเซนเซอร์และมีการทำงานที่ไม่ซับซ้อน ซึ่งบางทีจะถูกเรียกว่าโหนดสุดท้าย และต้องการที่จะส่งข้อมูลที่มีขนาดไม่มากนัก เนื่องจากต้องการประหยัดพลังงานที่ตัวเองให้มากที่สุด โหนด FFD สามารถที่จะคุยกับโหนด RFD หรือ FFD ใดๆ ก็ได้ทั้งหมดในเครือข่ายเดียวกัน แต่โหนด RFD จะคุยกับโหนด FFD ได้เท่านั้น ไม่สามารถคุยกับ RFD ด้วยกันได้ ในบางครั้งตัวโหนด FFD ก็สามารถที่จะทำหน้าที่เป็นโหนดเซนเซอร์เพื่อใช้ในการเก็บข้อมูลก็ได้ขึ้นอยู่กับการออกแบบตัวเครือข่าย

2.5 โครงสร้างเครือข่ายเซนเซอร์ไร้สาย ZigBee

โครงสร้างเครือข่ายของ ZigBee จะแบ่งเป็น 3 แบบ คือแบบดาว (Star topology) แบบจุดต่อจุด (Peer-to-peer topology) และแบบกลุ่มต้นไม้ (Cluster tree topology)

2.5.1 โครงสร้างแบบดาว (Star topology)

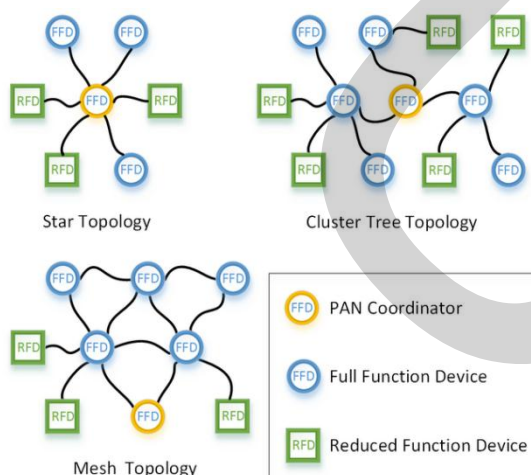
ในโครงสร้างแบบนี้การเริ่มการสื่อสารจะเริ่มขึ้นระหว่างโหนดโดยรอบกับโหนดควบคุมส่วนกลางหรือ (PAN coordinator) ตัวโหนดควบคุมส่วนกลางอาจจะจำเป็นต้องใช้พลังงานจากแหล่งพลังงานที่มากโหนดโดยรอบ เนื่องจากจะต้องติดต่อกับโหนดโดยรอบตลอดเวลา แต่โหนดโดยรอบอาจจะใช้เพียงแค่แบตเตอรี่อย่างเดียว การประยุกต์ใช้งานจากประโยชน์ที่ได้จากโครงสร้างแบบดาว คือการควบคุมระบบอัตโนมัติภายในครัวเรือน การทำงานของเครือข่ายเกิดขึ้นหลังจากโหนด FFD เริ่มต้นทำงานในครั้งแรก ตัวโหนด FFD ตัวนี้อาจจะทำหน้าที่เป็นโหนดควบคุมส่วนกลาง (PAN coordinator) ในแต่ละเครือข่ายดาว (Star Network) จะมีหมายเลขกำกับหรือ (PAN ID) ซึ่งจะไม่ซ้ำกับเครือข่ายอื่นๆ ภายในช่องสัญญาณสื่อสารเดียวกัน ซึ่งเป็นการทำให้เครือข่ายนี้ทำงานโดยอิสระกับเครือข่ายอื่นๆ

2.5.2 โครงสร้างแบบจุดต่อจุด (Peer-to-peer topology) หรือแบบตาข่าย (Mesh topology)

ในโครงสร้างแบบนี้ก็เช่นเดียวกับโครงสร้างแบบดาว คือ จะมีโหนดควบคุมส่วนกลาง (PAN coordinator) เพียงโหนดเดียวเท่านั้นในเครือข่าย แต่ตัวโหนดต่างๆ สามารถสื่อสารกับโหนดใดๆ ก็ได้เท่าที่อยู่ในย่านที่สามารถติดต่อกันกันได้ โครงสร้างแบบนี้เป็นเหมือนการทำงานในลักษณะของ Ad-Hoc คือสามารถที่จะสร้างโครงข่ายขึ้นมาด้วยตัวเองและดูแลรักษาโครงข่ายได้ด้วยตัวเอง ตัวอย่างการใช้งานเครือข่ายแบบนี้ เช่น ในการควบคุมในอุตสาหกรรม การเฝ้าติดตาม การเคลื่อนที่ต่างๆ เนื่องจากระบบเหล่านี้สามารถใช้ประโยชน์จากคุณสมบัติของโครงสร้างแบบนี้ได้เป็นอย่างดี ในโครงสร้างแบบนี้จะยอมให้มีการสร้างเส้นทางจากโหนดใดๆ ผ่าน ฮอปจำนวนมากได้ (Multiple hop) เพื่อไปยังโหนดใดๆ ในเครือข่าย อีกทั้งยังสามารถให้เส้นทางที่มีเสถียรภาพจากการทำการหาเส้นทางแบบหลายเส้น (Multipath routing)

2.5.3 โครงสร้างแบบกลุ่มต้นไม้ (Cluster tree topology)

เป็นโครงสร้างที่มีลักษณะจำเพาะหนึ่งของโครงสร้างแบบจุดต่อจุด (peer-to-peer topology) ซึ่งมีโหนดจำนวนมากเป็น FFD และมีโหนด RFD เชื่อมต่อเข้ากับโหนดเหล่านี้ ซึ่งมีลักษณะคล้ายกลุ่มของต้นไม้ โดยมีกิ่งก้านเป็น โหนด FFD และโหนด RFD เหมือนกับใบไม้ ซึ่งเป็นจุดปลายสุดของกิ่งก้านเหล่านั้น โหนด FFD ทุกๆ สามารถทำหน้าที่เป็นตัวประสานงานและคอยให้บริการกับโหนดอื่นๆ ที่เป็น FFD และ RFD ซึ่งในโครงสร้างแบบนี้จะมีเพียงโหนดเดียวทำหน้าที่เป็นตัวประสานงานเครือข่าย (PAN coordinator)



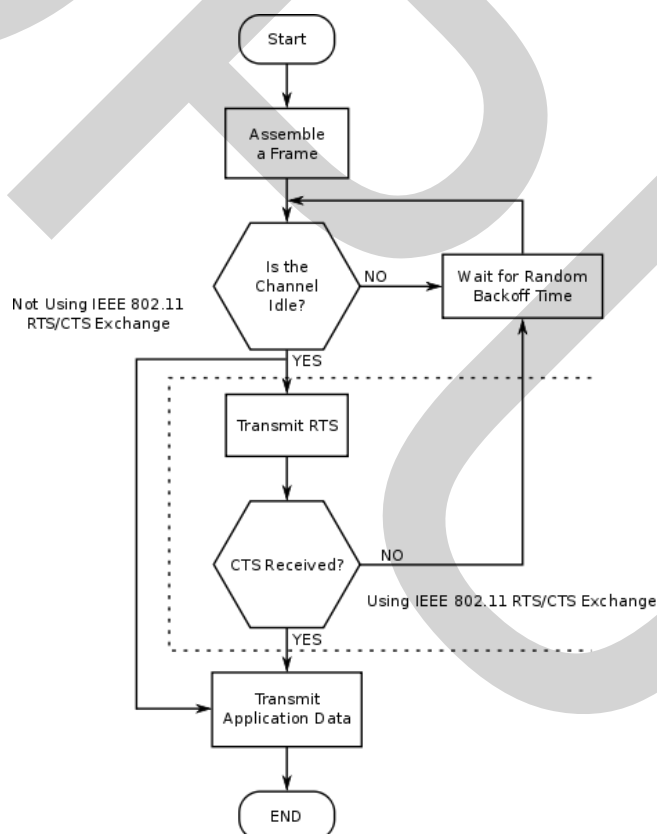
ภาพที่ 2.2 โครงสร้างเครือข่ายของ ZigBee

ที่มา: <http://www.pervcomconsulting.com/IEEE.html>

ตัวประสานงานเครือข่าย (PAN coordinator) ของกลุ่มแรกจะทำหน้าที่เป็นหัวหน้ากลุ่ม CLH และกำหนดตัวหมายเลขประจำกลุ่ม CL ID เป็น “0” หรือเลือกจาก PAN Identifier ที่ไม่ได้ใช้ จากนั้นก็จะกระจายเฟรมสัญญาณเตือน (frames broadcasting beacon) ไปยังตัวโหนดข้างเคียง

2.6 โครงสร้าง Low Rate Wireless Personal Area Network (LR-WPAN)

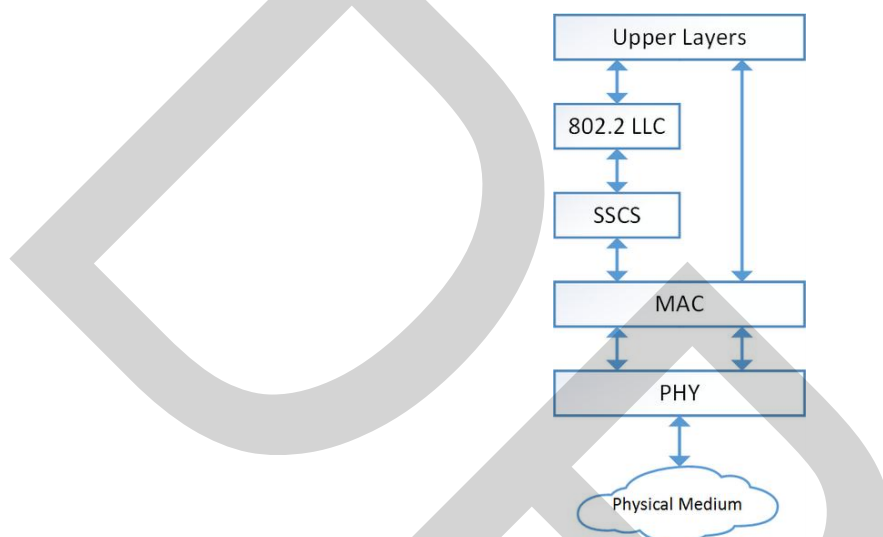
โครงสร้างของอุปกรณ์ที่เป็นแบบ LR-WPAN เนื่องจาก WSN มีจุดประสงค์หลักคือการส่งผ่านสัญญาณจากโหนดเซนเซอร์เพื่อมาประมวลผล โดยมีการส่งข้อมูลที่มีขนาดไม่ใหญ่มาก แต่จะเน้นที่การใช้พลังงานของตัวโหนดเซนเซอร์ให้น้อยที่สุด จึงต้องมีการออกแบบส่วนของ MAC เพื่อควบคุมการเข้าถึงสื่อกลางในการสื่อสารข้อมูล ระหว่างส่วนของ Upper Layers ไปยัง Physical Layers ในที่นี้คือคลื่นความถี่ โดยมีการใช้งานคลื่นความถี่ร่วมกันระหว่างโหนดเซนเซอร์ต่างๆ



ภาพที่ 2.3 กลไกการทำงาน CSMA/CA

ที่มา: http://en.wikipedia.org/wiki/File:Csma_ca.svg

ในกระบวนการติดต่อกับชั้นกายภาพของ WSN นั้นจะอาศัยกลไก ของ CSMA/CA ซึ่ง เป็นกระบวนการในการขอใช้ตัวกลางร่วมกัน และใช้กลไกของการตรวจจับว่าช่องสัญญาณว่างอยู่หรือไม่ ถ้าว่างก็จะมีการใช้ช่องสัญญาณนั้นๆ แต่ถ้าไม่ว่างก็จะจับเวลาถอยหลังเป็นเวลาเท่ากับ Backoff time จึงเริ่มทำการขอใช้ช่องสัญญาณใหม่



ภาพที่ 2.4 โครงสร้างของ LR-WPAN

ที่มา: <http://zigbeevsmbus.blogspot.com/2012/06/fossil-energy-crisis-havealways-lead-to.html>

โดยจะมีกลไกของตัว MAC เป็นตัวควบคุมการทำงานของตัว โหนด ว่าต้องการให้ตื่นหรือหลับ เมื่อไร โดยเน้นไปที่การใช้พลังงานให้ต่ำที่สุด เพื่อให้โหนดเซนเซอร์อยู่ได้นานที่สุด การทำงานของ MAC จะใช้การปรับเรื่องของค่า Duty Cycles โดยถ้าช่วงเวลาในการหลับมีมากกว่าช่วงเวลาตื่นมากๆ ก็จะทำให้ประหยัดพลังงานในการส่งข้อมูลได้ แต่ก็ต้องระวังเรื่องของช่วงเวลาในการตื่นเพื่อส่งข้อมูลนั้น จะต้องทำให้พร้อมกันกับตัว FFD ถ้าไม่ตรงอาจทำให้เกิดการส่งข้อมูลผิดพลาด หรือไม่ได้ส่งเลยก็ได้

2.7 พลังงานที่ใช้ใน WSN

ใน WSN สิ่งที่สำคัญคือเรื่องพลังงาน ในการทำงานของโหนดแต่ละโหนดนั้น จะต้องมี การใช้พลังงานไปในการทำงาน โดยรูปแบบของพลังงานที่ถูกใช้ไปใน WSN นั้น สามารถแบ่งได้ เป็นได้เป็น 2 แบบ คือ

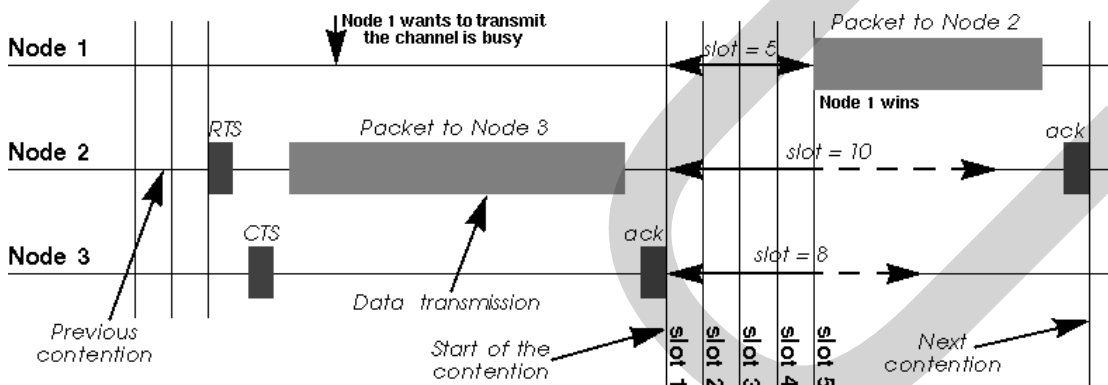
1) พลังงานที่ต้องใช้ (Useful power consumption) เป็นความต้องการใช้พลังงานสำหรับการส่ง หรือรับข้อมูล กระบวนการร้องขอการส่งข้อมูล การส่งต่อกระบวนการร้องขอต่างๆ หรือการส่งข้อมูลไปยังโหนดข้างเคียง (Node Neighbors)

2) พลังงานสูญเสียน (Wasteful power consumption) เป็นพลังงานที่สูญเสียนไปจากกระบวนการต่างๆ เช่น การรอช่องสัญญาณว่าง เพื่อจะส่งข้อมูล การส่งข้อมูลใหม่ ในกรณีที่ข้อมูลเดิมที่ถูกส่งมีการชนกันของข้อมูล กระบวนการที่เซนเซอร์ได้รับแพ็คเกจที่ไม่ใช่ของมัน

2.8 กระบวนการเข้าถึงช่องสัญญาณสื่อสารใน WSN

ใน WSN กระบวนการหรือกลไก ที่ตัวโหนดต่างๆ จะเข้าถึงช่องสัญญาณสื่อสาร หรือสื่อกลางที่ใช้ในการสื่อสารร่วมกันนั้น ล้วนแล้วแต่เป็นกระบวนการที่เกิดขึ้นในส่วนของ MAC ซึ่งสามารถแบ่งออกเป็นกลุ่มต่างๆ ได้ 3 วิธีการ คือ

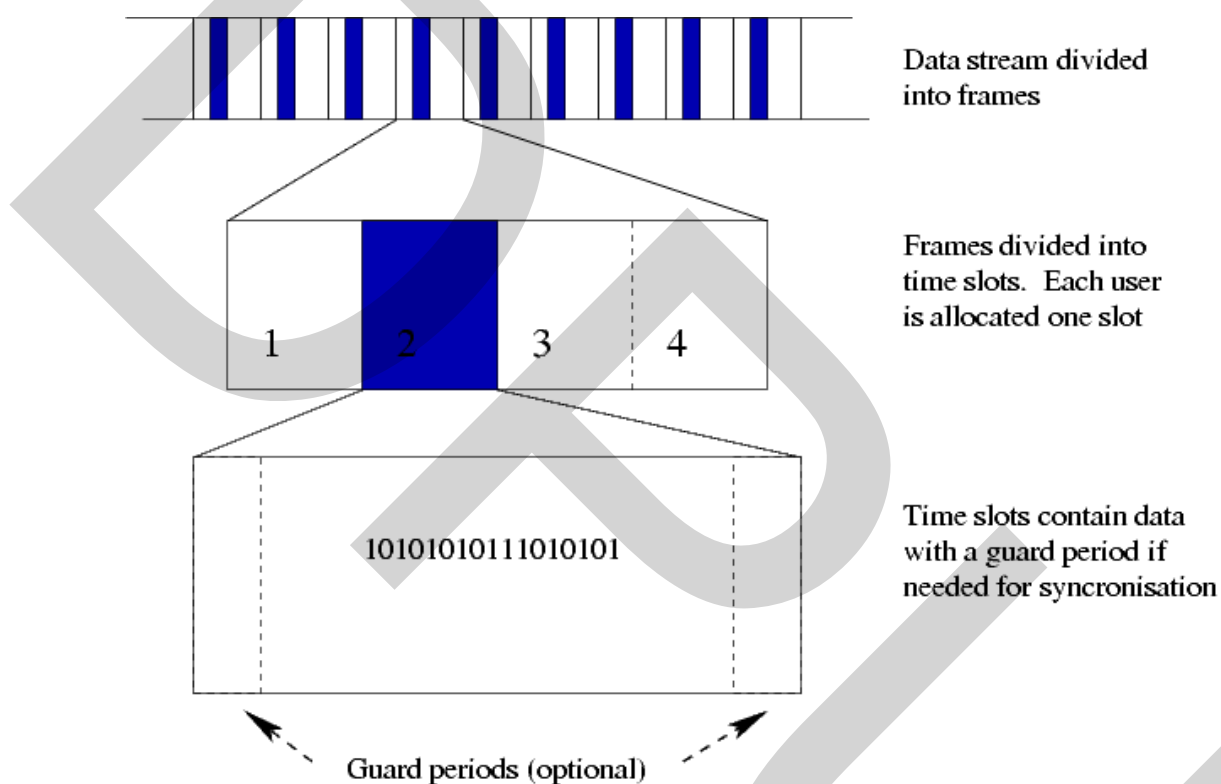
1) CSMA เป็นรูปแบบการเข้าถึงสื่อกลางแบบแย่งชิง (Contention based) คือจะไม่มีการจองสื่อกลางไว้ เหมือนแบบ TDMA ทำให้มีการใช้งานสื่อกลางได้อย่างมีประสิทธิภาพ แต่จะมีการเกิดการหน่วงเวลาบ้างในกรณีที่สื่อกลางมีการใช้หนาแน่น ตัวอย่างรูปแบบของ MAC ที่ทำงานในลักษณะนี้ ได้แก่ S-MAC, T-MAC, PCS-MAC, B-MAC, Wise-MAC, D-MAC, U-MAC, X-MAC, P-MAC, C-MAC, etc.



ภาพที่ 2.5 การเข้าถึงช่องสัญญาณของ CSMA/CA

ที่มา: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.mac.html

2) TDMA เป็นรูปแบบการเข้าถึงสื่อกลางแบบมีการจองช่วงเวลาไว้ก่อน (Time slot) หรือ (Reservation based) เมื่อต้องการส่งข้อมูลก็สามารถส่งได้เลย ทำให้มีประสิทธิภาพในการส่งข้อมูลสูง และเกิดการหน่วงเวลาในการรอการส่งน้อยเนื่องจากไม่ต้องแย่งสื่อกลางกัน ตัวอย่างรูปแบบของ MAC ที่ทำงานในลักษณะนี้ ได้แก่ ER-MAC, TRAMA, E-MACS, DE-MAC, BMA, SS-TDMA, L-MAC, etc.



ภาพที่ 2.6 โครงสร้างเฟรมของ TDMA

ที่มา: <http://en.wikipedia.org/wiki/File:Tdma-frame-structure.png>

3) Hybrid (CSMA/TDMA) เป็นรูปแบบการเข้าถึงสื่อกลาง โดยมีการประยุกต์เอาข้อดีของทั้ง CSMA และ TDMA เข้าด้วยกัน เช่นการปรับค่า Duty Cycle ในมีค่าเปลี่ยนแปลงตามความหนาแน่นของข้อมูล โดยการเพิ่มจำนวนช่องของ TDMA ตัวอย่างรูปแบบของ MAC ที่ทำงานในลักษณะนี้ ได้แก่ IEEE802.15.4, PTDMA, DEE-MAC, μ -MAC, SCP-MAC, R-MAC, A-MAC, SPARE-MAC, Y-MAC, Z-MAC, H-MAC, etc.

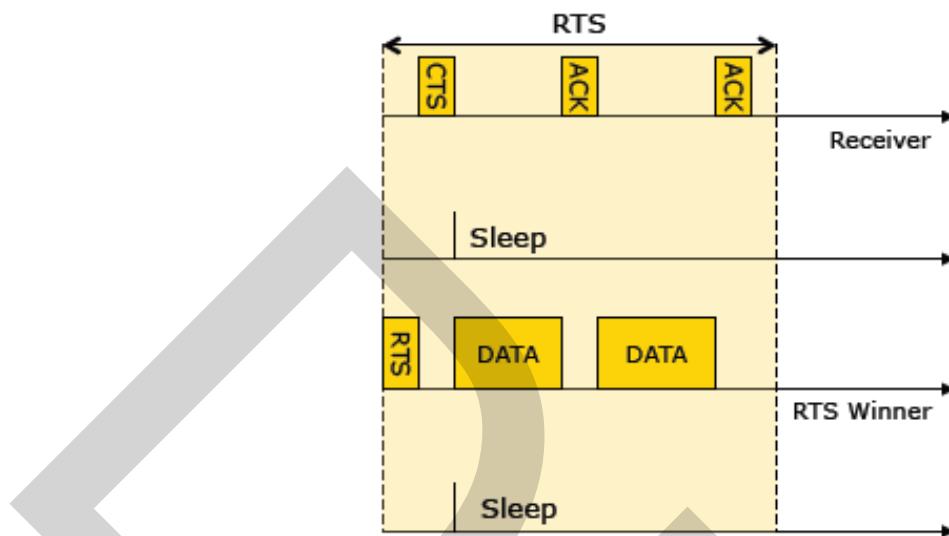
2.9 Media Access Control (MAC) ใน WSN

ใน WSN นั้น MAC มีความสำคัญ เนื่องจากถ้ามีการเลือกใช้ MAC ไม่เหมาะสมกับการใช้งาน ก็จะทำให้ไม่สามารถใช้งาน WSN ได้อย่างมีประสิทธิภาพ เพราะตัว MAC เป็นตัวแปรที่สำคัญที่จะทำให้โหนดต่างๆ ใน WSN สามารถประหยัดพลังงานในการส่งและรับข้อมูล (Energy efficiency) หรือมีประสิทธิภาพในแง่ของการส่งข้อมูลสำเร็จมากหรือน้อย (Throughput) การหลีกเลี่ยงการชนกันของข้อมูล (Collision avoidance) การเพิ่มจำนวนหรือลดจำนวนโหนดในเครือข่าย (Scalability) การหน่วงเวลาหรือการเข้าถึงข้อมูล (Delay and Latency) การใช้งานช่องสัญญาณอย่างมีประสิทธิภาพ (Bandwidth utilization) ดังนั้นส่วนใหญ่ MAC ต่างๆ ที่ใช้ใน WSN จะมุ่งเน้นไปในเรื่องของการ ประหยัดพลังงานในการส่งและรับข้อมูล โดยตัวอย่างการทำงานของ MAC แบบพื้นฐาน ที่ถูกใช้บ่อยใน WSN มีดังนี้

2.9.1 S-MAC

S-MAC เป็น MAC ที่มีการทำงานบนพื้นฐาน Contention based โดยจะมีช่วงเวลาหลับที่คงที่ (Regulates sleep periods) เพื่อประโยชน์ในเรื่องของการยืดอายุการใช้งานเครือข่ายเซนเซอร์ไร้สายให้ยาวนานขึ้น โดยอาศัยปัจจัย 4 อย่าง ในการบริหารจัดการพลังงาน

- 1) avoiding idle listening หลีกเลี่ยงการรับข้อมูลเข้ามา ในขณะที่ข้อมูลนั้นไม่ใช่ของตัวเอง
- 2) static sleep scheduling กำหนดช่วงเวลาในการหลับเป็นแบบคงที่
- 3) preamble sampling เมื่อตัวโหนดอยู่ในสถานะหลับ จะมีการตื่นขึ้นมาเป็นช่วงเวลาสั้นๆ เพื่อคอยดูว่ามีโหนดใดส่งข้อมูลมาให้หรือไม่ ถ้าไม่มีก็จะหลับต่อ ในขณะที่ตื่นขึ้นมาครั้งนั้น ถ้ามีโหนดใดส่งข้อมูลให้ ก็จะตื่นมารับข้อมูล
- 4) off-line scheduling จะทำงานเมื่อตัวโหนดอยู่ในสถานะหลับ โดยจะทำการปิดภาคส่ง และภาครับสัญญาณวิทยุ

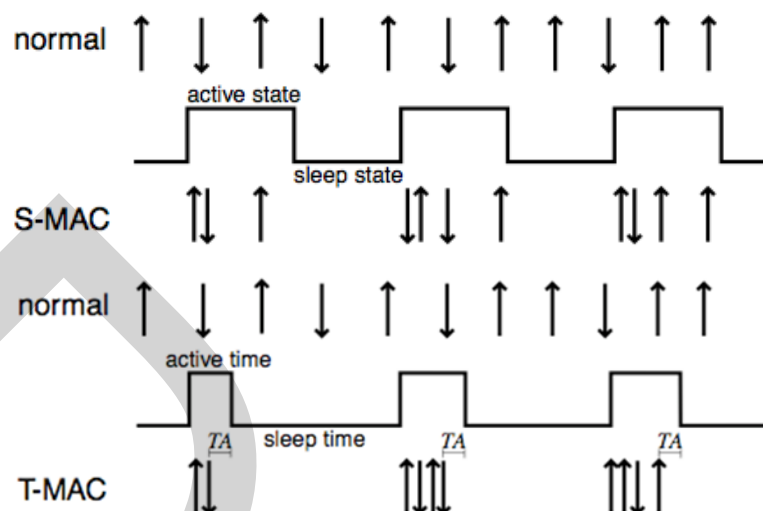


ภาพที่ 2.7 การทำงานของ S-MAC

ที่มา: http://www.cs.binghamton.edu/~kang/teaching/cs526/cs526_lect05_mac.ppt

2.9.2 T-MAC

T-MAC เป็น MAC ที่มีการทำงานบนพื้นฐาน Contention based โดยจะมีการทำงานคล้ายกับตัว S-MAC แต่จะมีการปรับเปลี่ยนวิธีการในการเข้าสู่โหมดการหลับ จากเดิมตัว S-MAC จะเป็นแบบ static sleep scheduling จะถูกปรับเปลี่ยนมาเป็น dynamic sleep scheduling โดยจะมีการปรับเปลี่ยนเวลาในการหลับ ตามปริมาณความหนาแน่นของการส่งและรับข้อมูล และให้ตัวจับเวลา TA เป็นตัวควบคุมการเข้าสู่โหมดหลับ เมื่อไม่การส่งหรือรับข้อมูลเป็นระยะเวลา $> TA$ ตัวโหนดนั้นก็จะเข้าสู่โหมดหลับ ทำให้เพิ่มเวลาในการหลับได้มากขึ้นและประหยัดพลังงานมากขึ้น จากภาพที่ 2.8 แสดงให้เห็นว่า ค่าเวลา TA ยิ่งน้อย ยิ่งทำให้เข้าสู่ Sleep mode เร็วขึ้น ทำให้ T-MAC ประหยัดพลังงานมากกว่า S-MAC



ภาพที่ 2.8 การเปรียบเทียบการทำงานของ S-MAC และ T-MAC

ที่มา: perso.ens-lyon.fr/eric.fleury/CPS/ART/Projet/MACPresentation.ppt

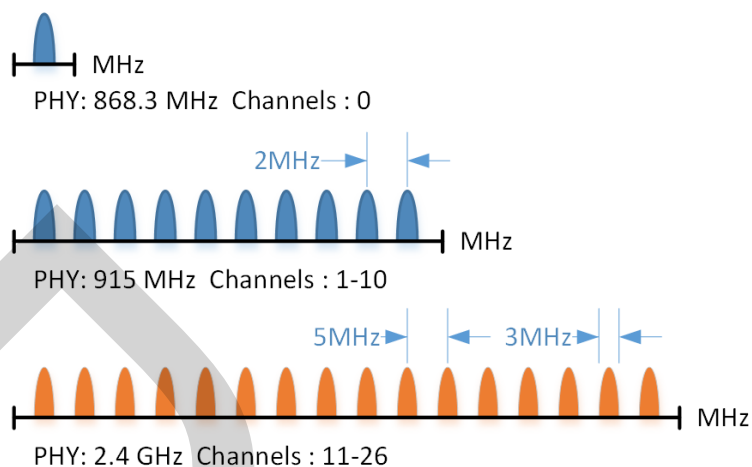
2.10 ย่านความถี่ที่ใช้กันใน IEEE802.15.4

ย่านความถี่และช่องสัญญาณข้อมูลที่ใช้ใน WSN จะมีอยู่ 3 ย่านความถี่ ซึ่งในแต่ละย่านความถี่ก็จะมีจำนวนช่องสัญญาณในการสื่อสารที่แตกต่างกัน

PHY (MHz)	Frequency Band (MHz)	Modulation	Channels	Channel spacing	Bit rate (kbps)	Symbol rate (ksymbol/s)	Symbols
868	868-868.6	BPSK	1	-	20	20	binary
868	868-868.6	ASK	1	-	250	12.5	20-bit PSSS
868	868-868.6	O-QPSK	1	-	100	25	16-ary orthogonal
915	902-928	BPSK	10	2MHz	40	40	binary
915	902-928	ASK	10	2MHz	250	50	5-bit PSSS
915	902-928	O-QPSK	10	2MHz	250	62.5	16-ary orthogonal
2450	2400-2483.5	O-QPSK	16	5MHz	250	62.5	16-ary orthogonal

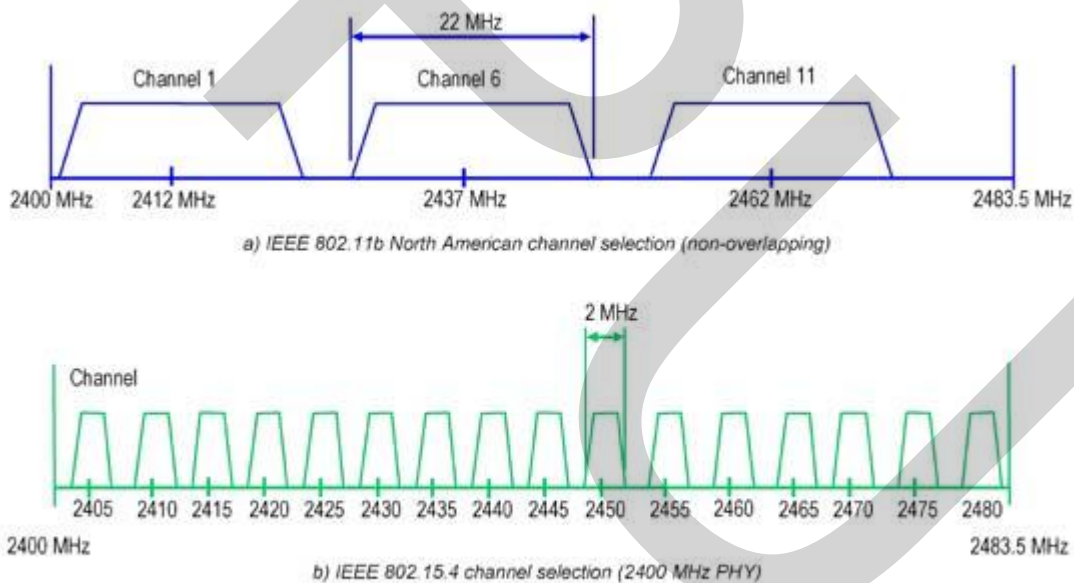
ภาพที่ 2.9 ค่าย่านความถี่ที่ใช้ใน IEEE802.15.4-2006

ที่มา: http://www.techonlineindia.com/techonline/design_centers/171033/introduction-wireless-sensor-network-concepts



ภาพที่ 2.10 ค่าความถี่มูลฐานของแต่ละช่องสัญญาณ

ที่มา: <http://thaieasyelec.com/article-wiki/basic-electronics/what-is-zigbee.html>



Source: IEEE 802.15.4 specification

ภาพที่ 2.11 ค่าความถี่ 2.4GHz ที่ทับซ้อนกันของ IEEE802.15.4 กับ IEEE802.11

ที่มา: http://www.eetasia.com/ART_8800550485_499488_NT_4a9289f5.HTM

จะเห็นได้ว่าในย่านความถี่ 2400MHz นั้นจะมีใช้ในส่วนของ IEEE802.15.4 กับ IEEE802.11b แต่มีส่วนของ Bandwidth ที่ต่างกัน ดังนั้นจึงอาจเกิดการกวนกันของสัญญาณได้

2.11 โพรโทคอลการค้นหาเส้นทางใน WSN

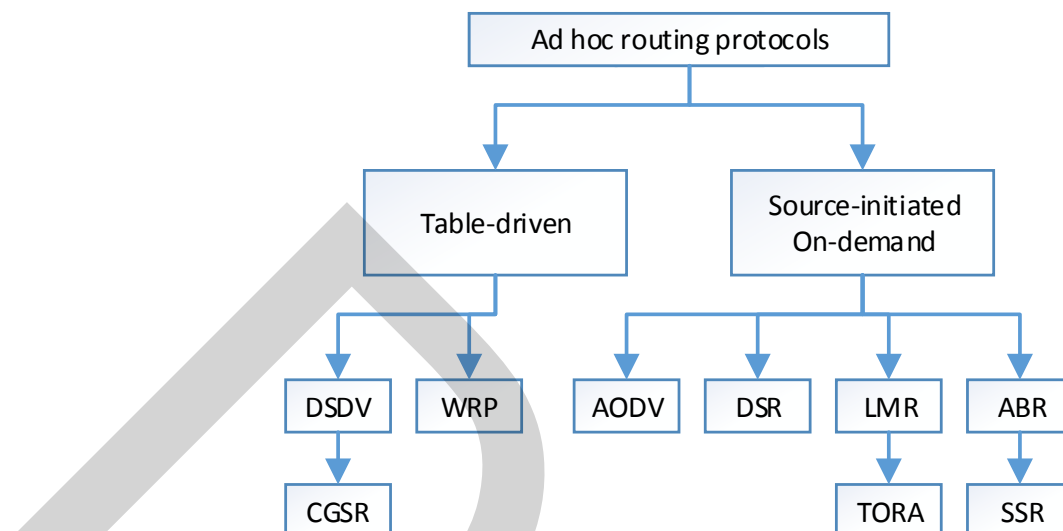
ใน WSN มีการเชื่อมกันแบบ Ad-Hoc mobile network ซึ่งเป็นวิธีการเชื่อมต่อเครือข่ายแบบหนึ่ง ในระบบการสื่อสารไร้สาย ที่โหนดนั้นมีการเคลื่อนที่อยู่ตลอดเวลา ดังนั้นเพื่อช่วยให้การติดต่อสื่อสารเป็นไปได้อย่างราบรื่นจึงได้มีการพัฒนาตัวโพรโทคอลในการค้นหาเส้นทาง ซึ่งถูกใช้ในการค้นหาเส้นทางระหว่างโหนดต้นทางไปยังโหนดปลายทาง ซึ่งเป้าหมายหลักของโพรโทคอลการค้นหาเส้นทางใน Ad-Hoc mobile network ก็เพื่อค้นหาเส้นทางที่ถูกต้องและมีประสิทธิภาพระหว่างคู่ของโหนดเพื่อให้ส่งข้อมูลไปยังเป้าหมายได้ตามเวลาที่เหมาะสม การสร้างเส้นทางนั้นควรจะทำให้เกิด Overhead และใช้ช่องสัญญาณ Bandwidth ให้น้อยที่สุด

ตั้งแต่โครงการวิจัย DARPA เกี่ยวกับเครือข่ายวิทยุแพ็คเกจ ในช่วงต้นปี 1970 มีโพรโทคอลจำนวนมากได้รับการพัฒนาเพื่อใช้งานบน Ad-Hoc mobile network หรือ MANET โพรโทคอลที่ถูกพัฒนาขึ้นมามากกว่าก็เพื่อจัดการกับข้อจำกัดต่างๆ เช่น การใช้พลังงานสูงในการทำงาน การได้ช่องสัญญาณที่ต่ำ และมีอัตราความผิดพลาดในการส่งข้อมูลที่สูง ซึ่งปัจจุบัน โพรโทคอลการค้นหาเส้นทางที่ถูกออกแบบมาเพื่อใช้ใน Ad-Hoc mobile network สามารถแบ่งออกเป็น 3 กลุ่ม คือ

1) Table-driven protocols (Proactive) ประกอบไปด้วยโพรโทคอลที่ใช้วิธีการแบบ Distance Vector เช่น DSDV, WRP และ โพรโทคอลที่ใช้วิธีการแบบ Link State เช่น OLSR, GSR, TBRPF

2) Source-initiated or Demand-driven protocols (Reactive) ประกอบไปด้วยโพรโทคอลที่ใช้วิธีการแบบ Distance Vector เช่น AODV, ABR และ โพรโทคอลที่ใช้วิธีการแบบ Link State เช่น DSR

3) Hybrid protocols ประกอบไปด้วยโพรโทคอลต่างๆ เช่น ARPAM, ZRP, OORP



ภาพที่ 2.12 โพรโทคอลการค้นหาเส้นทางที่ใช้ในเครือข่ายเฉพาะกิจ

ที่มา: <http://www.eexploria.com/routing-protocols-in-manets/>

2.11.1 Table-driven protocols (Proactive)

Table-driven protocols (Proactive) เป็นโพรโทคอลประเภท Table-Driven เป็นโพรโทคอลที่จะพยายามรักษาและอัปเดตข้อมูลเส้นทางจากโหนดหนึ่งไปยังทุกๆ โหนดในเครือข่ายให้เป็นปัจจุบัน โพรโทคอลเหล่านี้จำเป็นต้องใช้ในแต่ละโหนดเพื่อรักษาตารางเส้นทางหนึ่งหรือมากกว่าในการจัดเก็บข้อมูลเส้นทาง และโพรโทคอลเหล่านี้จะมีตอบสนองต่อการเปลี่ยนแปลงต่างๆ ในโครงสร้างของเครือข่ายโดยการปรับปรุงข้อมูลเส้นทางในตาราง โดยการกระจายไปให้ทั่วเครือข่าย เพื่อให้เครือข่ายยังคงมีตารางที่สอดคล้องกัน จำนวนโหนดที่มีจำนวนมากมีความเกี่ยวข้องกับขนาดของตารางเส้นทาง และมีความจำเป็นที่จะใช้วิธีการกระจายแบบ Broadcast เมื่อมีการเปลี่ยนแปลงของโครงสร้างของเครือข่าย

2.11.2 Source-initiated or Demand-driven protocols (Reactive)

Source-initiated or Demand-driven protocols (Reactive) เป็นโพรโทคอลประเภท Demand-Driven จะเริ่มทำการค้นหาเส้นทาง หรือจะสร้างเส้นทางก็ต่อเมื่อมีความต้องการที่จะส่งข้อมูลจากโหนดต้นทางเท่านั้น เมื่อโหนดต้นทางต้องการเส้นทางไปยังโหนดปลายทาง กระบวนการสำรวจเส้นทางจะเกิดขึ้นภายในเครือข่าย กระบวนการนี้จะเสร็จสิ้นก็ต่อเมื่อพบเส้นทางทั้งหมดที่เป็นไปได้และได้รับการตรวจสอบแล้ว เมื่อเส้นทางได้ถูกสร้างขึ้น เส้นทางจะถูกดูแลรักษาโดยกระบวนการซ่อมบำรุงเส้นทาง จนกระทั่งเส้นทางที่ไปยังปลายทางไม่สามารถเข้าถึง

ได้ทุกๆ เส้นทาง หรือจนกว่าเส้นทางนั้นจะไม่ใช่ที่ต้องการอีกต่อไป เส้นทางนั้นๆ ก็จะถูกกำจัดออกไป

2.12 โพรโทคอล AODV

โพรโทคอลการค้นหาเส้นทาง AODV เป็นโพรโทคอลแบบ Source Initiated On-Demand Driven/Reactive ซึ่งจะทำการหาเส้นทางก็ต่อเมื่อจุดเริ่มต้นต้องการหาเส้นทางโดยหาเส้นทางไปจนถึงปลายทาง ซึ่งจะเป็นการหาเส้นทางตามเส้นทางที่เป็นไปได้จนถึงปลายทางแล้ว จึงหยุดค้นหาหาเส้นทาง โพรโทคอล AODV ได้ปรับปรุงมาจาก DSDV เนื่องจากมีความคล้ายคลึงกันมาก แต่ AODV จะลดจำนวนครั้งในการที่จะค้นหาเส้นทางลง โดยจะทำการค้นหาเส้นทางเมื่อมีโหนดต้องการที่จะส่งข้อมูล หรือมีการเคลื่อนที่ของโหนดเท่านั้น ทำให้ไม่เปลือง Bandwidth ในช่องสัญญาณ

โพรโทคอล AODV เป็นโพรโทคอลค้นหาเส้นทางตามมาตรฐาน RFC3561 ที่ถูกใช้งานกันอย่างแพร่หลายในเครือข่ายการสื่อสารเคลื่อนที่แบบไร้สาย ที่มีเป็นแบบ ad hoc ซึ่งจะตอบสนองต่อการเปลี่ยนแปลงเส้นทางที่เกิดขึ้นอยู่บ่อยๆ ได้อย่างรวดเร็ว กินทรัพยากรในการประมวลผล และหน่วยความจำน้อย รองรับการสร้างเส้นทางแบบหลายฮอป (Multi Hop) มีการกำหนดเส้นทางแบบ Unicast เพื่อไปยังปลายทาง และปรับปรุงเส้นทาง ภายในเครือข่ายเฉพาะกิจ โดยจะใช้ลำดับเลขหมาย (Destination sequence numbers) คอยกำกับเส้นทางไปยังปลายทาง เพื่อให้มั่นใจได้ว่าจะไม่เกิดลูปตลอดเวลาและหลีกเลี่ยงการเกิดปัญหาการนับลำดับหมายเลขจนถึงค่าอนันต์ โดยใช้หลักการแบบ distance vector ของ Bellman-Ford โพรโทคอล AODV จะพยายามทำโหนดได้รับเส้นทางโดยเร็วที่สุดเมื่อมีการกำหนดโหนดปลายทางใหม่ และจะไม่ใช้โหนดที่ไม่ได้ทำงานอยู่เพื่อไปยังโหนดปลายทาง โพรโทคอล AODV จะแจ้งให้โหนดรับรู้ถึงการเชื่อมโยงที่ถูกตัดขาด และเมื่อมีการเปลี่ยนแปลงโครงสร้างของเครือข่าย เพื่อให้โหนดนั้นยกเลิกการใช้เส้นทางที่ถูกตัดขาดไป คุณลักษณะเด่นของโพรโทคอล AODV คือการใช้งานของลำดับหมายเลขเพื่อไปยังโหนดปลายทาง (Destination sequence numbers) สำหรับแต่ละเส้นทาง ซึ่งจะเป็นหมายเลขที่ไม่ซ้ำกัน และจะใส่ไปในแพ็คเกจข้อมูลที่ถูกส่งออกไป การใช้ลำดับหมายเลขปลายทางนี้จะเป็นวิธีการป้องกันการเกิดลูปในเครือข่าย และทำให้ง่ายต่อการเขียนโปรแกรม ในการเลือกเส้นทางไปยังโหนดปลายทางระหว่างสองเส้นทางที่มีนั้น โหนดปลายทางจะเลือกเพียงหนึ่งลำดับหมายเลขเท่านั้น

2.12.1 ความหมายของคำศัพท์ในโพรโทคอล AODV

ในตัวโพรโทคอล AODV มีการกำหนดความหมายของชื่อเรียกในตัวโพรโทคอลไว้ดังนี้

1) Active route เป็นเส้นทางที่สามารถไปยังโหนดปลายทางได้ ซึ่งมีรายการอยู่ในตารางเส้นทาง และมีการทำเครื่องหมายไว้ว่าสามารถใช้งานได้ ซึ่ง Active route จะเป็นเส้นทางที่ใช้ส่งข้อมูลได้

2) Broadcast เป็นการกระจายข้อมูลไปยังที่อยู่ไอพีที่ถูกกำหนดเป็น Broadcast address สำหรับไอพีเวอร์ชัน 4 ใช้ 255.255.255.255 โดยการกระจายแพ็คเก็ตนั้นจะช่วยให้ AODV สามารถกระจายข้อมูลไปได้ทั่วทั้งเครือข่ายเฉพาะกิจ

3) Destination เป็นโหนดปลายทางที่ต้องการส่งข้อมูลแพ็คเก็ต โดยข้อมูลนี้จะแสดงในส่วน IP Header การสร้างเส้นทางไปยังโหนดปลายทางนี้จะทำโดยโพรโทคอล AODV

4) Forwarding node เป็นโหนดที่ยอมให้มีการส่งผ่านแพ็คเก็ตไปยังโหนดปลายทาง โดยการส่งต่อแพ็คเก็ตเหล่านั้นไปยังโหนดข้างเคียงโดยใช้ Unicast ซึ่งเส้นทางนี้จะถูกใช้งานโดย Routing control message

5) Forward route เป็นเส้นทางที่จัดตั้งขึ้นเพื่อส่งแพ็คเก็ตข้อมูลจากโหนดต้นทางไปสู่โหนดปลายทางที่ต้องการ ซึ่งเกิดขึ้นในกระบวนการสำรวจเส้นทาง

6) Invalid route เป็นเส้นทางใดๆ ที่หมดอายุ และมีการทำเครื่องหมายยกเลิกไว้ในตารางเส้นทางแล้ว จะถูกนำมาใช้ในการจัดเก็บข้อมูลเส้นทางที่ถูกตัดก่อนหน้านี้ เส้นทางที่ถูกยกเลิกจะไม่สามารถใช้เพื่อส่งต่อแพ็คเก็ตข้อมูลได้ แต่ก็สามารถให้ข้อมูลที่เป็นประโยชน์ในการซ่อมแซมเส้นทางและเป็นข้อมูลให้ RREQ Message ในอนาคต

7) Originating node เป็นโหนดที่เป็นจุดเริ่มต้นของกระบวนการสำรวจเส้นทางใน AODV Message และอาจมีการส่งต่อข้อความนี้ไปยังโหนดอื่นๆ ภายในเครือข่ายเฉพาะกิจ ตัวอย่างเช่น โหนดที่เป็นโหนดเริ่มต้นในกระบวนการสำรวจเส้นทาง และทำการกระจาย RREQ Message นั้นจะถูกเรียกว่า “Originating node of the RREQ message”

8) Reverse route เป็นเส้นทางที่ถูกกำหนดขึ้นเพื่อการส่งต่อ RREP แพ็คเก็ต กลับไปยังโหนดเริ่มต้น จากโหนดปลายทาง หรือโหนดระหว่างทางที่มีเส้นทางไปยังโหนดปลายทาง

9) Sequence number เป็นจำนวนตัวเลขที่เพิ่มขึ้น โดยแต่ละโหนดต้นทาง ซึ่งเป็นกระบวนการสร้างเส้นทางใน AODV RREQ message ซึ่งจะถูกใช้โดยโหนดอื่นๆ เพื่อตรวจสอบความเป็นปัจจุบันของข้อมูลที่มาจากโหนดต้นทาง

10) Valid route เป็นเส้นทางที่มีการทำเครื่องหมายไว้ว่าสามารถใช้ในการส่งแพ็คเก็ตได้

โพรโทคอลสร้างเส้นทาง AODV ถูกออกแบบมาเพื่อใช้ในเครือข่ายเฉพาะกิจ (ad hoc network) โดยรองรับโหนดได้ถึงหลักพัน และรองรับการเคลื่อนที่ของโหนดจำนวนมากในเครือข่ายได้ดี อีกทั้งยังสามารถรองรับรูปแบบการส่งข้อมูลที่มีหลายระดับได้ AODV ถูกออกแบบมาเพื่อลดการแพร่กระจายแพ็กเก็ตเกิดในการปรับปรุงเส้นทาง และลด overhead ของข้อมูล เพื่อปรับปรุงประสิทธิภาพ และขยายขนาดของเครือข่ายให้ดีขึ้น

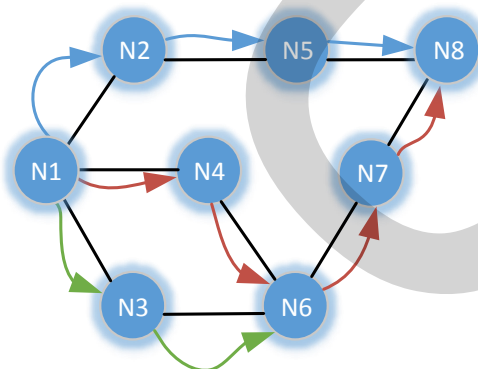
2.12.2 รูปแบบของข้อความ (Message Formats)

ในโพรโทคอลสร้างเส้นทาง AODV ได้มีการกำหนดรูปแบบของข้อความที่ส่งและรับระหว่างโหนดไว้ดังนี้

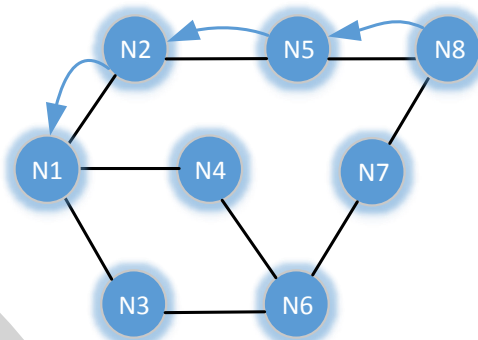
- 1) Route Request (RREQ) Message Format
- 2) Route Reply (RREP) Message Format
- 3) Route Error (RERR) Message Format
- 4) Route Reply Acknowledgment (RREP-ACK) Message Format

2.12.3 การทำงานของโพรโทคอล AODV

การทำงานของโพรโทคอล AODV จะเริ่มจากโหนดต้นทางมีการส่งร้องขอเส้นทาง (RREQ) มีการตอบ Route Reply (RREP) หรือ Route Error (RERR) สำหรับการสื่อสารแบบ Unicast ไปยังโหนดปลายทาง และเพื่อที่จะจัดการและประมวลผลข้อมูลเส้นทางเหล่านี้ได้ถูกต้อง จึงมีการจัดเก็บรักษาข้อมูลไว้ในตารางเส้นทางที่ไปยังโหนดเป้าหมาย โดยปกติแล้วโพรโทคอล AODV จะใช้ UDP ในการส่งข้อมูลไปยังพอร์ต 654



ภาพที่ 2.13 กระบวนการค้นหาเส้นทางไปยังโหนดปลายทาง (RREQ)



ภาพที่ 2.14 การตอบข้อความกลับ (RREP) ไปยังโหนดต้นทาง

ในโพรโทคอล AODV จะประกอบไปด้วย Message Route Request (RREQ) Route Replies (RREP) และ Route Errors (RERR) ซึ่งเป็นการรับแบบ UDP และมีการใช้ IP header ยกตัวอย่าง การร้องขอจากโหนดจะถูกระบุ IP Address ที่เป็นต้นกำเนิดของ Messages สำหรับการ Broadcast Message นั้น จะใช้ IP address (255.255.255.255) โดยขนาดของการ Broadcast Message RREQ นั้น จะถูกแสดงและกำหนดเป็นค่า TTL ที่อยู่ในส่วนหัวของไอพี เมื่อมีโหนดเชื่อมต่อเข้ามา และไม่มีเส้นทางไปยังโหนดนั้น ถ้าต้องการไปยังโหนดปลายทางนั้น จะมีการ Broadcasts RREQ เพื่อค้นหาเส้นทางไปยังโหนดปลายทางนั้น เส้นทางจะถูกสร้างขึ้นเมื่อ RREQ Message ไปถึงยังโหนดปลายทางที่ระบุไว้ด้วยตัวเอง หรือโหนดระหว่างทางมีเส้นทางที่สามารถไปยังโหนดปลายทาง ถ้าเส้นทางใหม่เป็นเส้นทางที่ถูกต้องสามารถไปยังปลายทางที่อยู่ใน RREQ ได้และมีลำดับหมายเลข เส้นทางนี้จะเป็นเส้นทางที่ใช้สำหรับส่ง RREP กลับไปยังต้นทาง โดยการ Unicast แต่ละโหนดจะคอยเฝ้าติดตามสถานะของการเชื่อมโยงถัดไปยังโหนดที่ทำงานอยู่ เมื่อการเชื่อมโยงดังกล่าวขาดลง ในระหว่างการค้นหาเส้นทาง จะมี RERR message ถูกส่งออกไปเพื่อแจ้งเตือนโหนดต่างๆ ถึงการสูญเสียการเชื่อมโยงดังกล่าว RERR Message จะแสดงให้โหนดปลายทางเหล่านั้นที่อยู่ใน Subnets เดียวกันถึงการไม่สามารถใช้งานได้จากการเชื่อมโยงที่มีปัญหา ซึ่งกระบวนการนี้จะต้องเปิดใช้งาน กลไกการรายงาน ซึ่งในแต่ละโหนดจะเก็บ Precursor List ภายในรายการจะประกอบไปด้วย IP address ของโหนดข้างเคียง ที่เป็นโหนดเพื่อใช้เส้นทางนำไปสู่ปลายทาง ข้อมูลที่อยู่ในรายการนี้ส่วนใหญ่เกิดจากการประมวลผลในการสร้าง RREP message ซึ่งเป็นโหนดที่จะต้องผ่านไปยังโหนดต้นทาง โพรโทคอล AODV จะใช้ข้อมูลในการสร้างเส้นทางดังนี้

- 1) หมายเลขไอพีที่อยู่ปลายทาง Destination IP Address
- 2) หมายเลขลำดับปลายทาง Destination Sequence Number

3) เครื่องหมายแสดงหมายเลขลำดับปลายทางที่ใช้งานได้ Valid Destination Sequence Number flag

4) สถานะอื่นๆ Other state and routing flags (เช่น valid, invalid, repairable, being repaired)

5) อินเทอร์เฟซเครือข่าย Network Interface

6) จำนวนโหนด Hop Count (number of hops needed to reach destination)

7) รายการโหนดถัดไป Next Hop List of Precursors

8) อายุของแพ็กเก็ต Lifetime (expiration or deletion time of the route)

2.12.4 การดูแลรักษา Sequence Number

การจัดการหมายเลขลำดับเป็นสิ่งสำคัญเพื่อที่จะหลีกเลี่ยงการเกิดลูปในเส้นทาง เมื่อการเชื่อมโยงถูกตัดขาด ทำให้ไม่สามารถเข้าถึงโหนดได้ โหนดปลายทางไม่สามารถเข้าถึงการเชื่อมโยงในเครือข่ายได้ หรือเส้นทางถูกปิดการใช้งาน เมื่อเงื่อนไขเหล่านี้เกิดขึ้น เส้นทางนั้นก็จะถูกยกเลิกโดยการทำเครื่องหมายที่ลำดับหมายเลขในตารางเส้นทางไว้ โดยทุกๆ โหนดที่มีข้อมูลอยู่ในตารางเส้นทาง จะต้องมีข้อมูลเกี่ยวกับ Sequence Number สำหรับไปยังหมายเลข IP address ของโหนดปลายทาง ซึ่ง Sequence Number นี้ถูกเรียกว่า destination sequence number โดยตัวเลขนี้จะถูกปรับปรุงอยู่เสมอเมื่อโหนดได้รับข้อมูล Sequence Number ใหม่จาก RREQ RREP หรือ RERR message ซึ่งเป็นรับข้อมูลที่เกี่ยวข้องกับโหนดปลายทาง โดยโหนดปลายทางจะมีการเพิ่มค่า Sequence Number ของตัวเองใน 2 กรณี

1) เพิ่มทันทีก่อนที่โหนดเริ่มต้นจะค้นพบเส้นทาง มันจะทำการเพิ่มค่า Sequence Number ของตัวมันเอง เพื่อป้องกันไม่ให้เกิดความขัดแย้งกับ Sequence Number ที่สร้างขึ้นก่อนหน้านี้ ในขณะที่ส่งข้อมูลย้อนกลับไปสู่เส้นทางเริ่มต้นของโหนดเริ่มต้นที่กระจาย RREQ ออกมา

2) เพิ่มทันทีก่อนที่โหนดปลายทางจะส่ง RREP เพื่อตอบสนองต่อ RREQ โดยมันจะต้องปรับปรุงค่า Sequence Number ของมันให้เป็นค่าสูงสุดของหมายเลขปัจจุบันและ Destination sequence number ในแพ็กเก็ต RREQ

โหนดปลายทางสามารถเพิ่มค่าหมายเลขลำดับ โดยหมายเลขลำดับจะเป็นตัวเลขจำนวนเต็มขนาด 32 บิต ซึ่งมีค่าสูงสุดที่ 4294967295 เมื่อเพิ่มจนถึงค่านี้แล้วก็จะเริ่มต้นที่ศูนย์ใหม่ ในทางกลับกันถ้าคิดเป็นจำนวนเต็มแบบมีเครื่องหมายค่าจะเริ่มตั้งแต่ -2147483648 ถึง 2147483647 และเพื่อเป็นการตรวจสอบให้แน่ใจว่าข้อมูลเกี่ยวกับโหนดปลายทางไม่เป็นข้อมูลเดิมที่ค้างอยู่ โหนดจะทำการเปรียบเทียบค่าเลขหมายลำดับปัจจุบันของมันกับค่าเลขหมายลำดับที่ได้รับมาใหม่จาก AODV Message ในการเปรียบเทียบนี้จะต้องทำการเปรียบเทียบทางคณิตศาสตร์แบบคิด

เครื่องหมายขนาด 32 บิต ถ้าผลลัพธ์ของการลบหมายเลขลำดับระหว่างหมายเลขเดิมกับหมายเลขใหม่มีค่าน้อยกว่าศูนย์ หมายความว่าข้อมูลของโหนดใหม่ที่ส่งเข้าเก่ากว่าข้อมูลเดิมที่เก็บอยู่ในโหนด

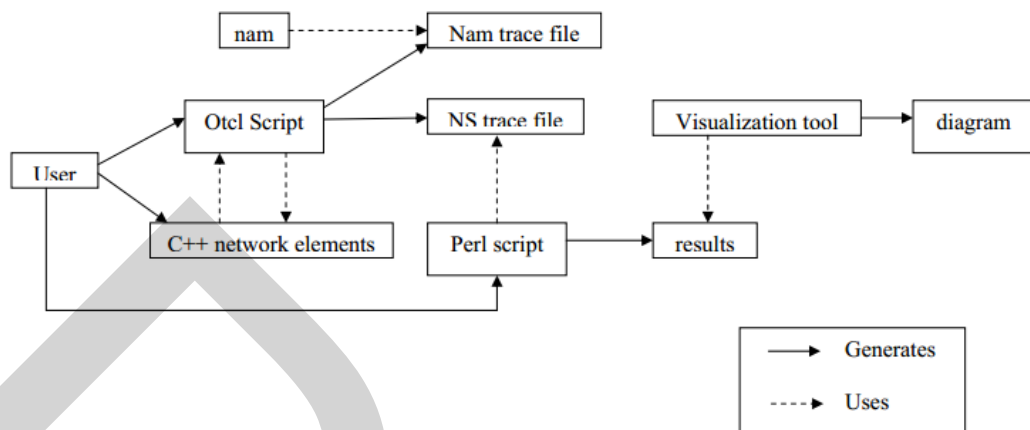
กรณีที่เส้นทางที่เชื่อมต่อระหว่างโหนดไปยังโหนดที่สามารถเชื่อมไปยังโหนดปลายทางได้มีการชำรุดหรือขาดลง โหนดนั้นอาจมีการเปลี่ยนแปลงค่าหมายเลขลำดับ โดยในกรณีนี้จะทำการเพิ่มค่าหมายเลขลำดับที่ไปยังโหนดปลายทางนี้ และทำการทำเครื่องหมาย “Invalid” เพื่อยกเลิกเส้นทางนี้ แต่เมื่อไรที่มีการรับข้อมูลใหม่เข้ามาและค่าหมายเลขลำดับน้อยกว่าหรือเท่ากับหมายเลขลำดับที่ถูกระงับไว้ นั่นหมายความว่าโหนดนั้นสมควรจะได้รับการปรับปรุง โหนดอาจจะมีการเปลี่ยนหมายเลขลำดับในตารางเส้นทางของโหนดปลายทางถ้าอยู่ในเงื่อนไขดังนี้

- 1) ถ้าโหนดนั้นเป็นโหนดปลายทางเอง และต้องการให้เส้นทางใหม่มาอย่างต่อเนื่อง
- 2) ถ้าได้รับ AODV message ที่เป็นข้อมูลใหม่เกี่ยวกับหมายเลขลำดับของโหนดปลายทาง
- 3) ถ้าได้รับ AODV message ที่เป็นข้อมูลว่าเส้นทางไปยังโหนดปลายทางนั้นชำรุดหรือเสียหาย

2.13 โปรแกรมจำลองการทำงานของเครือข่าย NS2

โปรแกรมการจำลองการทำงานของระบบเครือข่ายที่มีชื่อว่า NS2 เป็น discrete event simulator โดย NS2 ซึ่งสนับสนุนการทำงานแบบจำลองการทำงานของ UDP, TCP, Routing, Multicast protocols over wired และ wireless (local และ satellite) networks จากคุณสมบัติเหล่านี้เราสามารถจำลองการทำงานของเครือข่ายได้

NS2 มีภาษาที่ใช้อยู่ 2 ภาษาคือ ภาษา C++ และ Tcl โดย NS2 เป็น Open Source และสามารถที่จะทำงานได้ทั้งบน Linux , FreeBSD, SunOS, Solaris, Window ถูกพัฒนาขึ้นโดย ISI ซึ่ง NS2 เป็นโปรแกรมที่ใช้ในการจำลองการทำงานของ network ในแบบที่เป็น Discrete event simulator ซึ่งสนับสนุนการจำลองการเลือกเส้นทางในการขนส่ง packet สามารถจำลองการทำงานของ multicast protocol และ IP protocol เช่น UDP, TCP, RTP, SRM ที่อยู่บนเครือข่ายประเภทที่เป็น Wire และ Wireless (Local และ Satellite) ซึ่ง NS2 เป็นเครื่องมือที่มีประโยชน์มากทั้งยังสนับสนุน multiple protocol และยังสามารถในการแสดงรายละเอียดของ Network traffic ออกมาในรูปแบบของภาพเคลื่อนไหว รวมทั้งยังสนับสนุนขั้นตอนวิธีในการจัดเส้นทาง และ Queuing เช่น FIFO, Round-robin เป็นต้น



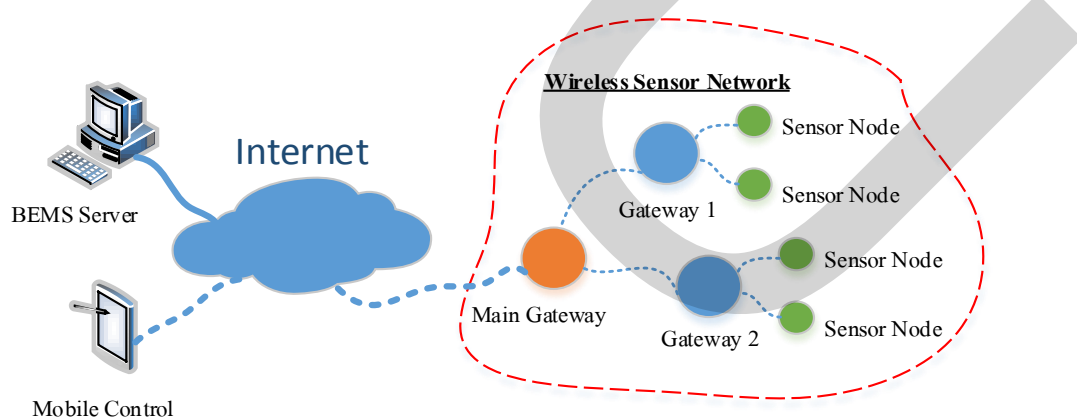
ภาพที่ 2.15 การทำงานของผู้ใช้งานผ่านโปรแกรม NS2

ที่มา: <http://wiki.nectec.or.th/ngiwiki/pub/Project/IntelligentNMS/bm-ns2.pdf>

2.14 งานวิจัยที่เกี่ยวข้อง

2.14.1 งานวิจัยเรื่อง Efficient Multiple Gateway System for WSN Management in BEMS จากงานวิจัยของ Xuan He, Yuanchen Ma และ Mika Mizutani

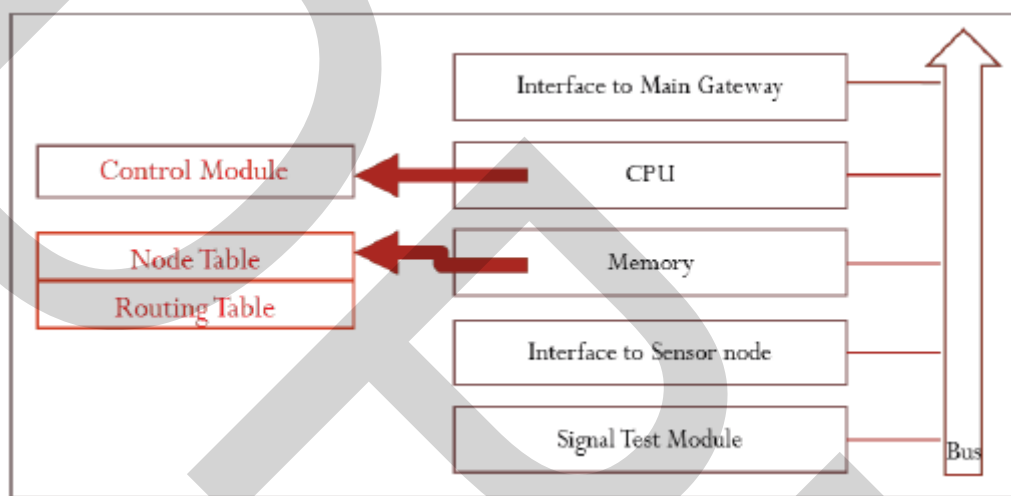
เป็นงานวิจัยเกี่ยวกับการใช้ระบบหลายเกตเวย์ ในเครือข่ายเซนเซอร์ไร้สาย เพื่อเพิ่มประสิทธิภาพในการบริหารจัดการพลังงานในอาคาร BEMS โดยมีการออกแบบไว้ให้มีสถาปัตยกรรมที่มีเกตเวย์หลักอยู่หนึ่งตัว โดยเชื่อมต่อกับกับเครื่องแม่ข่าย BEMS และรวบรวมข้อมูลจากเทอร์มินอลเกตเวย์ตัวอื่นๆ ที่ต่อกับโหนดเซนเซอร์



ภาพที่ 2.16 โครงสร้างของ Multiple Gateway System

ที่มา: Xuan He and Yuanchen Ma, "Efficient Multiple Gateway System for WSN Management in BEMS" Networked Sensing Systems (INSS), Belgium, June 2012.

ในกรณีที่ตัวเทอร์มินอลเกตเวย์ในระบบตัวใดตัวหนึ่งมีปัญหาตัวโหนดเซนเซอร์ที่ต่ออยู่กับเทอร์มินอลเกตเวย์ตัวนั้น ก็จะหาเส้นทางไปยังเทอร์มินอลเกตเวย์ตัวอื่นๆ ที่ยังทำงานอยู่ แล้วจึงส่งต่อข้อมูลออกไปยังเกตเวย์หลัก เพื่อส่งต่อไปยังเครื่องข่าย BEMS อีกที ตัวเทอร์มินอลเกตเวย์ในสถาปัตยกรรมนี้จะแตกต่างกับตัวเกตเวย์ปกติทั่วไป ตรงที่จะต้องให้ตัวโมดูลควบคุมและคอยส่งสัญญาณ ไปทดสอบแต่ละเกตเวย์ว่ายังทำงานอยู่หรือไม่ ซึ่งการส่งสัญญาณนี้ จะเป็นการกินทรัพยากรด้านช่องสัญญาณในเครือข่าย



ภาพที่ 2.17 โครงสร้างของ Terminal Gateway

ที่มา: Xuan He and Yuanchen Ma, “Efficient Multiple Gateway System for WSN Management in BEMS” Networked Sensing Systems (INSS), Belgium, June 2012.

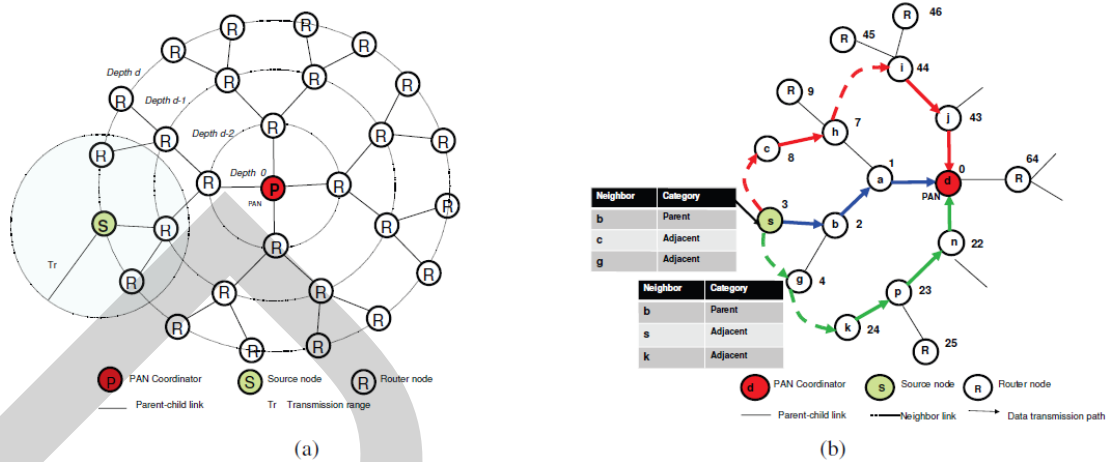
จากสถาปัตยกรรมแบบนี้ตัวกระบวนการหลัก (Main process) จะเริ่มต้นการทำงานโดยการสร้างตารางเส้นทางโดยมีเส้นทางทั้งหมดของเกตเวย์ข้างเคียง โดยตารางเส้นทางนี้จะมีพารามิเตอร์ของการหน่วงเวลา (delay time) การใช้พลังงาน (Power consumption) และสถานะของเส้นทาง ค่าการหน่วงเวลาจะหาจากขนาดของหน่วยความจำและสถานะของเส้นทาง ส่วนค่าการใช้พลังงาน จะคำนวณจากระยะทางของเกตเวย์ข้างเคียง หลังจากคำนวณทุกอย่างเสร็จแล้วเส้นทางที่ดีที่สุดจะถูกเลือก ซึ่งเป็นเส้นทางที่ประหยัดพลังงานและมีการหน่วงเวลาน้อยที่สุด ตัวสถานะของเส้นทางจะถูกระบุเป็น “ON” นอกนั้นจะระบุเป็น “OFF” ซึ่งตัวตารางเส้นทางนี้จะถูกปรับปรุงอยู่เป็นประจำ เมื่อเส้นทางที่ดีที่สุดขาด เส้นทางที่สองจะถูกเลือกขึ้นมาใช้แทน ตัวเกตเวย์หลัก จะทำ

หน้าที่เป็นเหมือนสะพานเพื่อเชื่อมระหว่างเครือข่ายเซนเซอร์ไร้สายกับเครือข่ายอินเทอร์เน็ต โดยจะรับข้อมูลจากเทอร์มินอลเกตเวย์ นอกจากนี้ยังทำหน้าที่ในการรับคำสั่งจากตัวผู้ใช้งานเพื่อส่งไปยังเทอร์มินอลเกตเวย์และ โหนดเซนเซอร์ปลายทางที่ต้องการ ในงานวิจัยนี้ได้ประโยชน์ในการช่วยให้ตัวเซนเซอร์โหนดสามารถหาทางออกไปยังเกตเวย์หลักได้ง่ายขึ้นเนื่องจากมีหลายเกตเวย์ และใช้ตัวแปรเวลากับพลังงาน มาเป็นส่วนหนึ่งในการเลือกเส้นทางที่ดีที่สุด ซึ่งการใช้วิธีนี้สามารถที่จะลดเรื่องของการหน่วงเวลาและการใช้พลังงาน ได้ทั้งคู่

ในส่วนองงานวิจัยนี้ จะมีปัญหาอยู่ตรงส่วนที่เป็นตัวเกตเวย์หลัก เนื่องจากมีแค่เพียงตัวเดียว ดังนั้นถ้าเกตเวย์ตัวนี้มีปัญหา SPOF ก็จะทำให้เครือข่ายเซนเซอร์ไร้สายไม่สามารถติดต่อกับเครือข่ายอินเทอร์เน็ตได้ นั่นหมายความว่าเครือข่ายเซนเซอร์ไร้สายจะไม่สามารถส่งข้อมูลไปเก็บบน BEMS เครื่องแม่ข่ายได้

2.14.2 งานวิจัยเรื่อง Node disjoint multi-path routing for ZigBee cluster-tree Wireless Sensor Network จากงานวิจัยของ Zahia Bidai, Hafid Haffaf และ Moufida Maimour

เป็นงานวิจัยที่เกี่ยวกับการส่งข้อมูลในเครือข่ายเซนเซอร์ไร้สายแบบหลายเส้นทาง เนื่องจาก ZigBee เป็นเครือข่ายแบบ Ad-hoc ที่มีรากฐานการพัฒนาจากมาตรฐาน IEEE802.15.4 ซึ่งมุ่งเน้นไปที่ชั้นกายภาพ (Physical layers) และชั้นการเชื่อมโยงข้อมูล (MAC layers) โดยมุ่งเน้นการใช้งานสำหรับงานที่ต้องการส่งผ่านข้อมูลที่มีขนาดต่ำ (Low data rate) และใช้พลังงานในการทำงานที่น้อย จากคุณลักษณะนี้ตัว ZigBee จึงถูกคาดหวังในการใช้งานเป็นเครือข่ายเซนเซอร์ไร้สายในอุตสาหกรรมเพื่อคอยวัดและควบคุมงานต่างๆ ดังนั้นการใช้งานที่ต้องการส่งผ่านข้อมูลที่มีขนาดสูง (High data rate) ผ่านเครือข่ายนี้ เช่น การส่งผ่านข้อมูลภาพเคลื่อนไหว จึงเป็นสิ่งที่น่าสนใจ จากงานวิจัยนี้ พวกเขาได้นำเสนอวิธีการที่จะส่งผ่านข้อมูลที่มีขนาดสูง (High data rate) ผ่านเครือข่ายนี้โดยใช้วิธีการส่งแบบหลายเส้นทาง โดยมีการจำลองการส่งข้อมูลจากต้นทางไปยังปลายทาง โดยใช้โปรโตคอล Z-MHTR ซึ่งเป็นโปรโตคอลที่สร้างขึ้นเพื่อใช้งานในการสร้างเส้นทางแบบหลายเส้นทางในการส่งข้อมูลบนเครือข่ายรูปแบบกลุ่มต้นไม้ (Cluster tree network) และทำการเปรียบเทียบกับโปรโตคอล AODV (Table driven or On-Demand) ซึ่งเป็นการสร้างเส้นทางแบบ Single-Path โดยใช้โปรแกรมจำลองการทำงานเครือข่าย NS2 ในการทดสอบ ทำการวัดประสิทธิภาพของเครือข่ายด้านการส่งผ่านข้อมูล (Throughput Data) การหน่วงเวลาในการส่งข้อมูลระหว่างต้นทางไปยังปลายทาง (Delay end to end) และระยะเวลาของการทำงาน (Lifetime) ระหว่างการส่งผ่านข้อมูลที่มีขนาดสูง (High data rate) และ ส่งผ่านข้อมูลที่มีขนาดต่ำ (Low data rate)



ภาพที่ 2.18 ภาพ (a) Circular Topology ภาพ (b) An example of the multipath routing in subnet ZigBee cluster-tree

ที่มา: Zahia Bidai and Hafid Haffaf “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”, Multimedia Computing and Systems (ICMCS), Ouarzazate, Morocco, April 2011.

จากภาพที่ 2.18 ภาพ a เป็นรูปแบบเครือข่ายที่จะนำมาจำลอง โดยการกำหนดให้โหนดปลายทางในที่นี้คือ โหนดประสานงานเครือข่าย (PAN Coordinator node) อยู่ที่ศูนย์กลางของเครือข่าย และมีการส่งข้อมูลจากโหนดต้นทาง (Source node) ผ่านไปยังเราเตอร์โหนดต่างๆ ภาพ b เป็นตัวอย่างของเส้นทางที่ถูกสร้างขึ้นซึ่งจะเห็นได้ว่ามีอยู่ 3 เส้นทางที่จะวิ่งไปยังโหนดปลายทางบนเครือข่ายแบบ กลุ่มต้นไม้ (Cluster tree)

ในการจำลองการทำงานของเครือข่ายเซนเซอร์ไร้สาย บนโปรแกรม NS2 มีการกำหนดค่าพารามิเตอร์ และตั้งค่าต่างๆ ดังนี้

ตารางที่ 2.1 ค่าการตั้งค่าพารามิเตอร์ในโปรแกรม NS2

พารามิเตอร์	ค่าที่กำหนด
Number of Nodes	29
Radio Transmission range (m)	17
Number of FFD	28
Number of Simulation Runs	(4*8) a total 32
Circle Radius R (m)	10
Number of Sources	1
Number of Sinks	1 (PAN Coordinator)
Sink Position	Center of the area
Node Mobility	None
ZigBee Parameters (Lm, Cm, Rm)	(3,4,4)
Initial Energy of Sensor nodes	1 j
Transmission Energy	0.3 μ j/bit
Reception Energy	0.3 μ j/bit
MAC Protocol	IEEE802.15.4 (Non-Beacon mode with unslotted CSMA/CA mechanism)
Propagation Model	Two Ray Ground Model
Queue Size	50
Data Transfer Model	Direct Data Transmission
Traffic Model	Poisson traffic
Packet Size (bytes)	80
Simulation Area (m*m)	70*70
Traffic load (pps)	200, 100, 50, 20, 10, 5, 2, 1
Simulation Time (sec)	100

ในการวัดผลของประสิทธิภาพของการทำงานของวิธี Z-MHTR นั้น พวกเขาพิจารณาตัวชี้วัดต่างๆ ดังนี้

1) Packet delivery fraction เป็นค่าความสำเร็จในการรับข้อมูลที่ปลายทางกับจำนวนข้อมูลที่ถูส่งจากต้นทาง

2) Average end-to-end delay เป็นค่าการหน่วงเวลาโดยเฉลี่ย จากการส่งข้อมูลจากต้นทางไปยังปลายทาง

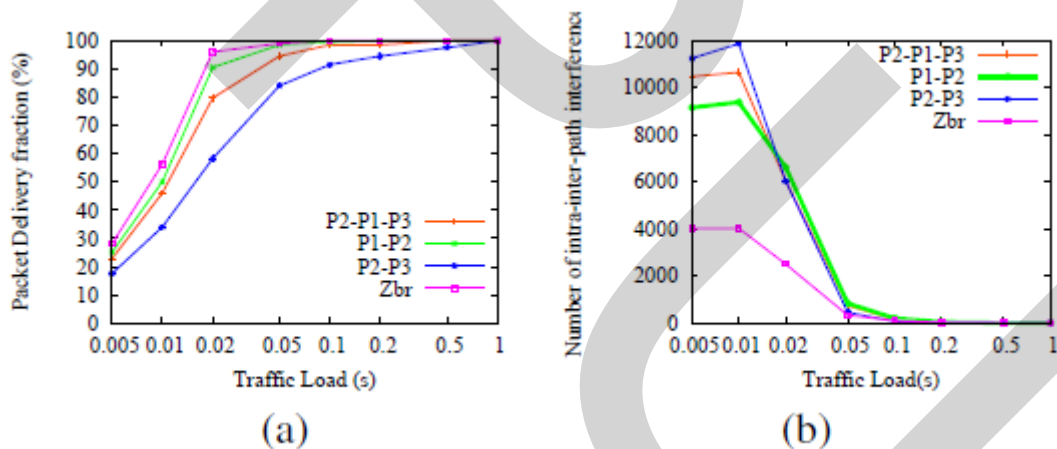
3) Network lifetime เป็นค่าเวลาที่หมดที่เครือข่ายทำงาน เมื่อปลายทางหยุดรับข้อมูลจากต้นทาง เครือข่ายก็จะหยุดทำงาน

จากตัวชี้วัดข้างต้นสามารถตีความหรือตั้งข้อสังเกตอื่นๆ ได้ดังนี้

1) เกิดการชนกันของข้อมูลที่โหนดปลายทาง ทำให้เกิดการหลุดของข้อมูลได้ ซึ่งเป็นปัจจัยที่เกี่ยวข้องกับจำนวนข้อมูลที่หลุดไป

2) เกิดการชนกันของข้อมูลภายในเส้นทางเดียวกัน ทำให้เกิดการหลุดของข้อมูลได้ ซึ่งเป็นปัจจัยที่เกี่ยวข้องกับจำนวนข้อมูลที่หลุดไป

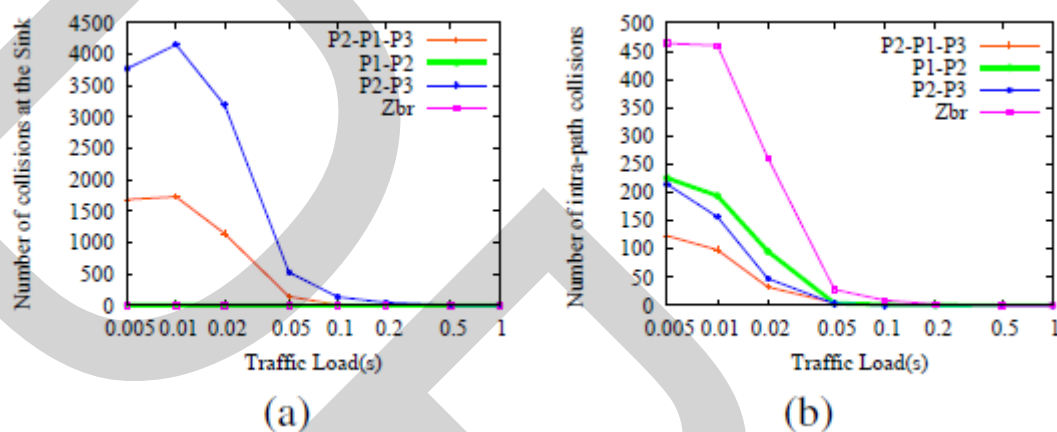
3) เกิดการรบกวนกันระหว่างเส้นทางภายใน กับเส้นทางภายนอก ทำให้เกิดการหลุดของข้อมูลได้ ซึ่งเป็นปัจจัยที่เกี่ยวข้องกับจำนวนข้อมูลที่หลุดไป



ภาพที่ 2.19 ภาพ (a) Packet delivery fraction ภาพ (b) Number of intra-path and inter-path interference in the different routing strategies

ที่มา: Zahia Bidai and Hafid Haffaf “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”, Multimedia Computing and Systems (ICMCS), Ouarzazate, Morocco, April 2011.

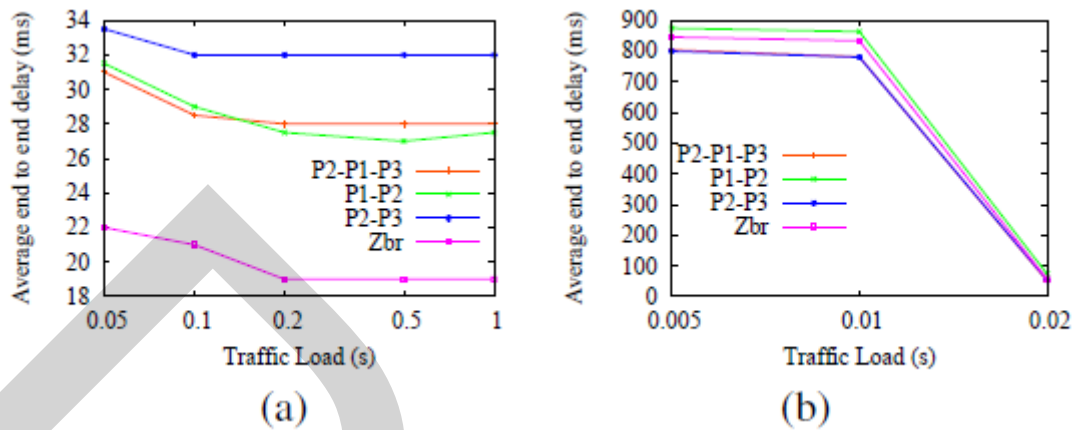
จากภาพที่ 2.19 ภาพ (a) จะเห็นได้ว่าเปอร์เซ็นต์ระหว่างการส่งและรับข้อมูลสำเร็จนั้นแบบที่เป็นเส้นทางเดียว (Zbr) จะมีค่าสูงกว่าการส่งข้อมูลแบบหลายๆ เส้นทางในทุกๆ ค่าภาระการจราจร (Traffic Load) ส่วนภาพ (b) แสดงถึงการรบกวนกันของเส้นทางระหว่างเส้นทางภายในกับเส้นทางภายนอก จะเห็นได้ว่าการรบกวนกันแบบที่เป็นเส้นทางเดียว (Zbr) มีค่าน้อยกว่าการส่งข้อมูลแบบหลายๆ เส้นทาง ตั้งแต่การส่งข้อมูลที่มีค่ามากกว่า 20 pps



ภาพที่ 2.20 ภาพ (a) Number of collisions at the sink ภาพ (b) Number of intra-path collisions in the different routing strategies

ที่มา: Zahia Bidai and Hafid Haffaf “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”, Multimedia Computing and Systems (ICMCS), Ouarzazate, Morocco, April 2011.

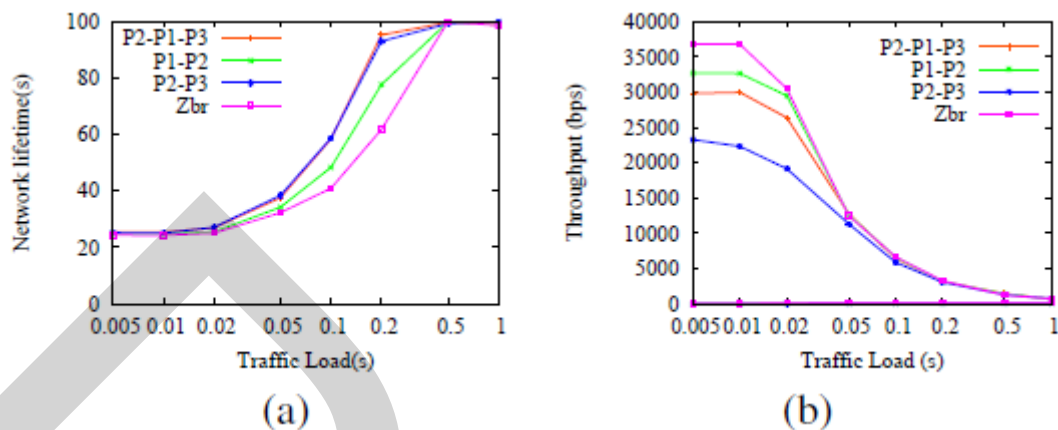
จากภาพที่ 2.20 ภาพ (a) จะแสดงให้เห็นถึงการชนกันของข้อมูลที่ตัวโหนดปลายทาง โดยการส่งข้อมูลแบบที่เป็นเส้นทางเดียว (Zbr) จะไม่เกิดการชนข้อมูลเลย แต่แบบการส่งข้อมูลแบบหลายๆ เส้นทางมีอยู่ 1 เส้นทางที่ไม่มีการชนกันของข้อมูล ส่วนอีก 2 เส้นทาง เริ่มมีการชนของข้อมูลมากขึ้นตั้งแต่การส่งข้อมูลที่มีค่ามากกว่า 20 pps ส่วนภาพ (b) แสดงถึงการชนกันของข้อมูลภายในเส้นทางเดียวกัน ซึ่งการส่งข้อมูลแบบหลายๆ เส้นทาง ทำได้ดีกว่าการส่งข้อมูลแบบเส้นทางเดียว แต่ก็เห็นได้ว่าเริ่มมีการชนกันเกิดขึ้นอย่างมากตั้งแต่การส่งข้อมูลที่มีค่ามากกว่า 20 pps



ภาพที่ 2.21 ภาพ (a) Average end to end delay in the different routing at time: 0.05s to 1s ภาพ (b) at time: 0.005s to 2s

ที่มา: Zahia Bidai and Hafid Haffaf “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”, Multimedia Computing and Systems (ICMCS), Ouarzazate, Morocco, April 2011.

จากภาพที่ 2.21 ภาพ (a) จะแสดงให้เห็นถึงค่าเฉลี่ยของการหน่วงเวลาของข้อมูลจากต้นทางไปยังปลายทาง โดยพิจารณาข้อมูลที่มีขนาดตั้งแต่ 1pps ถึง 20pps โดยการส่งข้อมูลแบบที่เป็นเส้นทางเดียว (Zbr) จะมีการหน่วงเวลาน้อยกว่าแบบการส่งข้อมูลแบบหลายๆ เส้นทาง ซึ่งมีการหน่วงเวลามาก ภาพ (b) จะแสดงให้เห็นถึงค่าเฉลี่ยของการหน่วงเวลาของข้อมูลจากต้นทางไปยังปลายทาง โดยพิจารณาข้อมูลที่มีขนาดตั้งแต่ 50pps ถึง 200pps ซึ่งในกรณีนี้ ทั้งการส่งข้อมูลแบบเส้นทางเดียว แบบหลายเส้นทาง เริ่มมีการหน่วงเวลามากขึ้นเรื่อยๆ



ภาพที่ 2.22 ภาพ (a) Network lifetime ภาพ (b) Throughput in the different routing strategies

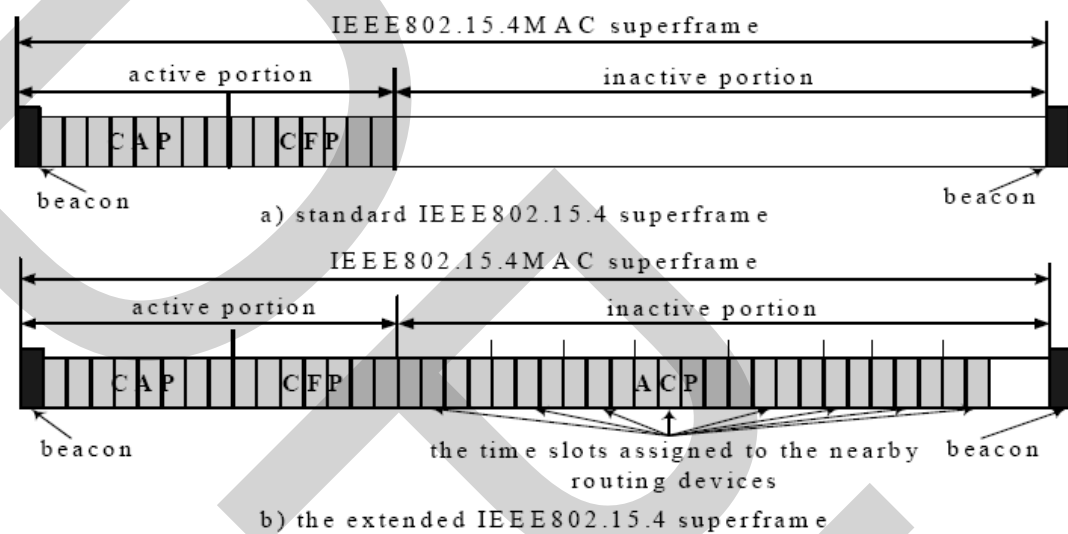
ที่มา: Zahia Bidai and Hafid Haffaf “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”, Multimedia Computing and Systems (ICMCS), Ouarzazate, Morocco, April 2011.

จากภาพที่ 2.22 ภาพ (a) จะแสดงให้เห็นถึงอายุการใช้งานเครือข่าย โดยแบบเส้นทางเดียว โหนดปลายทางจะหยุดรับข้อมูลก็เมื่อโหนดแรกที่เชื่อมต่อกับมันหยุดทำงาน ส่วนแบบหลายเส้นทาง ตัวโหนดปลายทางจะหยุดรับข้อมูลก็เมื่อโหนดแรกของทุกๆ เส้นทาง ที่เชื่อมต่อกับมันหยุดทำงาน ซึ่งที่ขนาดของข้อมูลที่มีการส่งเท่าๆ กันที่ 5pps ตัวการทำงานของเครือข่ายที่ใช้แบบเส้นทางเดียวจะทำงานน้อยกว่า ซึ่งทำให้อายุการใช้งานยาวนานกว่า ส่วนที่ $< 2\text{pps}$ และ $> 100\text{pps}$ การทำงานของเครือข่ายพอๆ กัน ในภาพ (b) จะแสดงให้เห็นถึงขนาดของข้อมูลที่ผ่านเครือข่ายโดยที่จราจรภายในเครือข่ายมากกว่า 20pps การใช้เส้นทางเดียว จะให้ข้อมูลผ่านได้มากที่สุด

ในส่วนของงานวิจัยนี้ จะมีปัญหาอยู่ในเรื่องของการชนกันของข้อมูลภายในเส้นทางเดียวกัน และข้ามเส้นทาง จึงเป็นสาเหตุให้เกิดการหน่วงเวลาของข้อมูลและเป็นผลทำให้การส่งผ่านข้อมูลทำได้ช้าลง ดังนั้นจึงคิดว่าการส่งข้อมูลแบบเส้นทางเดียวน่าจะทำได้ดีกว่าการส่งหลายเส้นทาง เนื่องจากเกิดการชนกันของข้อมูลที่โหนดปลายทาง และระหว่างเส้นทางน้อยกว่า อีกทั้งยังทำให้การส่งข้อมูลสามารถทำได้มากกว่าแบบหลายเส้นทาง

2.14.3 งานวิจัยเรื่อง Congestion Control for Industrial Wireless Communication Gateway
จากงานวิจัยของ Tang Zhong, Mei Zhan

เป็นงานวิจัยเกี่ยวกับการลดความคับคั่งของข้อมูลที่เกตเวย์ โดยอาศัยการเพิ่มประสิทธิภาพของ IEEE802.15.4 MAC super frame ซึ่งโดยปกติแล้วที่ตัวเกตเวย์ การทำงานจะทำงานแบ่งเป็น 2 ส่วนคือส่วนที่ Active กับส่วนที่เป็น Inactive ดังภาพที่ 2.23



ภาพที่ 2.23 การทำงานของ IEEE802.15.4 MAC super frame

ที่มา: Tang Zhong, Mei Zhan “Congestion Control for Industrial Wireless Communication Gateway”

จากภาพที่ 2.23 (b) จะเห็นได้ว่าการเพิ่มการส่งข้อมูลในส่วนที่เป็น Inactive portion ซึ่งจะช่วยลดปัญหาความคับคั่งของข้อมูลที่ตัวเกตเวย์ได้ แต่อย่างไรก็ตามงานวิจัยนี้ ไม่ได้กล่าวถึงปัญหาในกรณีที่เกตเวย์เกิดการเสียหาย ดังนั้นจึงคิดว่าอาจเกิดปัญหา SPOF ที่ตัวเกตเวย์ได้

บทที่ 3

ระเบียบวิธีวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาและเปรียบเทียบการทำงานของโพรโทคอล AODV สำหรับการสื่อสารข้อมูลภายในเครือข่ายเซนเซอร์ไร้สายตามมาตรฐาน IEEE 802.15.4 ระหว่าง Router Node กับ Coordinator Node ซึ่งตัว Coordinator Node จะทำหน้าที่เป็น Gateway ในการติดต่อสื่อสารกับเครือข่ายอื่นๆ โดยเปรียบเทียบกรณีที่มี Coordinator Node 1 ตัว กับ กรณีที่มี Coordinator Node 2 ตัว โดยงานวิจัยนี้จะแสดงให้เห็นถึงประโยชน์ของการมี Coordinator Node ที่มากกว่า 1 ตัว

3.1 แนวทางการวิจัยและพัฒนา

1) ศึกษาค้นคว้าทฤษฎีและงานวิจัยที่เกี่ยวข้อง โดยจะเริ่มจากเครือข่ายเซนเซอร์ไร้สายตามมาตรฐาน IEEE802.15.4 มีการอธิบายโครงสร้างและหลักการทำงานของอุปกรณ์ในเครือข่ายในชั้นกายภาพ (Physical layers) ชั้นการควบคุมเข้าถึงสื่อกลางการสื่อสารข้อมูล (Medium Access Control) จากนั้นจะเป็นส่วนของชั้นเครือข่าย (Network layers) ซึ่งเป็นชั้นที่พูดถึงการเดินทางของข้อมูล และจะเป็นส่วนที่สำคัญในงานวิจัยนี้ เพราะจะต้องมีการพัฒนาและออกแบบโพรโทคอลเส้นทางเพื่อใช้งานในชั้นนี้ ต่อมาจะศึกษาเกี่ยวกับโพรโทคอลเส้นทาง AODV ที่ใช้ในเครือข่ายเซนเซอร์ไร้สาย โดยดูเรื่องการทำงานของโพรโทคอล ข้อดีและข้อเสียของตัวโพรโทคอลนี้ เพื่อที่จะนำมาเป็นแนวทางในการพัฒนาโพรโทคอลใหม่ขึ้น สุดท้ายเป็นการศึกษาการใช้งานโปรแกรมจำลองการทำงานเครือข่าย (Network Simulator) ในงานวิจัยนี้จะใช้โปรแกรม NS2 ช่วยในการสร้างโหนดเซนเซอร์ไร้สายและโพรโทคอลเส้นทาง เพื่อที่จะเปรียบเทียบการทำงานของโพรโทคอล AODV ในกรณีที่มี Coordinator Node 1 ตัว กับ กรณีที่มี Coordinator Node 2 ตัว

2) ออกแบบโปรแกรมเพื่อใช้ในการจำลองการทำงานของเครือข่ายเซนเซอร์ไร้สายแบบ Multiple gateways โดยใช้ภาษา OTcL ซึ่งเป็นภาษาสคริป ในการสร้างลำดับขั้นตอนการทำงานของเครือข่าย แล้วจึงใช้โปรแกรม Network Simulator 2 หรือ NS2 ในการจำลองการทำงาน of เครือข่ายเซนเซอร์ไร้สาย โดยใช้โพรโทคอล AODV ในการค้นหาและสร้างเส้นทางจาก Router

Node ไปยัง Coordinator Node โดยมุ่งเน้นการออกแบบและพัฒนาโปรแกรมให้เป็นที่ระบุไว้ในวัตถุประสงค์ของวิจัย

3) สร้างโมเดลจำลองการทำงานของ IEEE802.15.4 โดยกำหนดให้มีโหนดต้นทางจำนวน 1 โหนด และโหนดปลายทางจำนวน 2 โหนด โดยโหนดทั้งหมดเป็นแบบ FFD (Full Function Device) และเป็นโหนดแบบไม่มีการเคลื่อนที่

4) ทดสอบโมเดลจำลอง จะทำโดยการใช้โปรแกรม NS2 ในการจำลองการทำงานของเครือข่ายเซนเซอร์ไร้สาย เพื่อเปรียบเทียบปริมาณ Throughput, Delay ของเครือข่าย ในกรณีที่มี Gateway 1, 2 ตัว โดยใช้โปรโตคอลเส้นทาง ระหว่างโปรโตคอล AODV

5) เปรียบเทียบ วิเคราะห์ผลที่ได้จากการทดสอบ และสรุปการทดสอบ เพื่อที่จะสรุปผลการจำลองระบบว่าประสิทธิภาพของระบบนั้นเป็นอย่างไร

6) รวบรวมข้อมูลที่ได้ทั้งหมด เพื่อจัดทำวิทยานิพนธ์ โดยแผนการดำเนินงานที่ได้วางแผนไว้ แสดงอยู่ในตารางที่ 3.1

ตารางที่ 3.1 แผนการดำเนินงาน

	พ.ย.	ม.ค.	เม.ย.	ก.ค.	ต.ค.	ม.ค.	พ.ค.
	ธ.ค.	มี.ค.	มิ.ย.	ก.ย.	ธ.ค.	เม.ย.	ก.ค.
	55	56	56	56	56	57	57
ศึกษาข้อมูลและงานวิจัยที่เกี่ยวข้อง							
ศึกษาการทำงานของเครือข่าย WSN และโปรโตคอล AODV							
ศึกษาการใช้งานโปรแกรม NS2							
วิเคราะห์และศึกษาแนวทางการพัฒนา ทำการเปรียบเทียบผลจากการจำลองระบบ							
รวบรวมข้อมูลและวิเคราะห์ผลจากการจำลองระบบ แล้วสรุปผล							
รวบรวมข้อมูลที่ได้ทั้งหมดจัดทำวิทยานิพนธ์							

3.2 เครื่องมือที่ใช้ในงานวิจัย

3.2.1 เครื่องคอมพิวเตอร์แบบพกพาจำนวน 1 เครื่อง ซึ่งมีคุณสมบัติดังนี้

- 1) CPU Intel® Core™ i3-3227U @ 1.90GHz
- 2) VGA Intel® HD Graphics 4000
- 3) RAM DDR3 4GB
- 4) Hard Disk, SATA 500GB
- 5) Network Adapter: Wireless LAN 802.11n
- 6) OS Microsoft Windows 8, 64-bit

3.2.2 ซอฟต์แวร์ที่ใช้ในการพัฒนาและวิจัย มีรายละเอียดดังนี้

- 1) VMWare Workstation Ver.9.01
 - a. CPU 2 Processor
 - b. Memory 2 GB
 - c. Hard Disk (SCSI) 20GB
 - d. Network Adapter: NAT
- 2) OS Ubuntu Desktop 12.04 LTS
- 3) Network Simulator 2.34 (NS2)
- 4) C++
- 5) OTcL Script

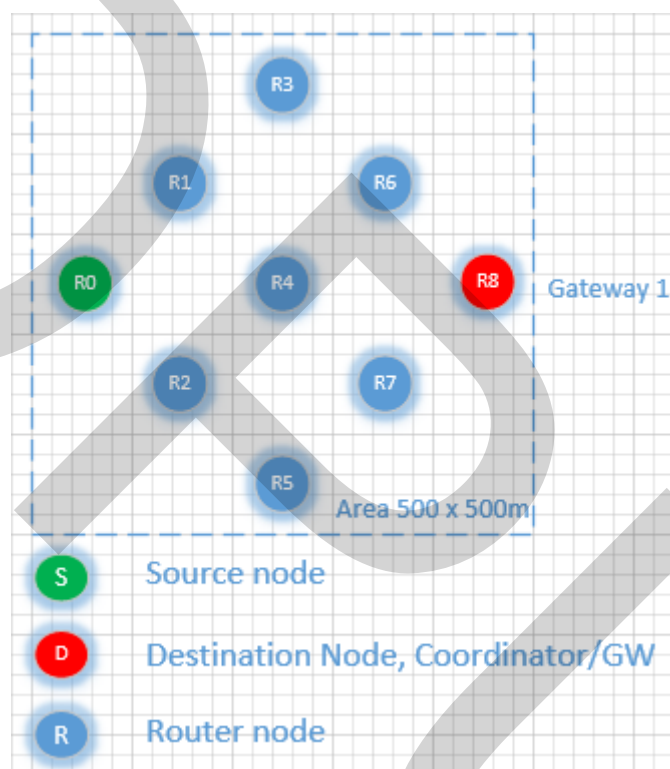
3.3 ขั้นตอนการดำเนินงานวิจัย

3.3.1 สร้างโมเดลโครงข่าย WSN

สร้างโมเดลโครงข่ายเพื่อใช้ในการจำลองการทำงานของโปรโตคอล AODV โดยมีการกำหนดโหนดที่เป็น Source และ Destination ในการสร้างโมเดลโครงข่ายจำลองของเครือข่ายเซนเซอร์ไร้สายเพื่อใช้งานงานวิจัยนี้ จะแบ่งเป็น 2 โมเดล โดยมีการอ้างอิงโครงข่ายต้นแบบของงานวิจัย Node disjoint multi-path routing for ZigBee cluster-tree Wireless Sensor Network ซึ่งเป็น Circular Topology โดยมีการปรับขนาดของจำนวนโหนดให้เล็กลง และเพิ่มโหนดที่เป็น Coordinator Node ดังนี้

1) แบบที่ 1 Coordinator Node (1 Gateway) จะประกอบไปด้วยโหนดที่เป็น FFD โหนด จำนวน 9 โหนด มีโหนดที่ทำหน้าที่เป็น Source จำนวน 1 โหนด และโหนดที่ทำหน้าที่เป็น ตัว Destination จำนวน 1 โหนด โดยจะมีการส่งข้อมูลที่เป็น CBR Packet จาก Source โหนด ไปยัง

Destination โหนด กำหนดขนาดของทอพอโลยี จะเป็น 500 x 500 เมตร จากรูปข้างล่าง จะประกอบไปด้วย โหนด R0 ถึง R8 รวมเป็นทั้งหมด 9 โหนด โดยโหนด R0 จะทำหน้าที่เป็น Source โหนด และโหนด R8 จะทำหน้าที่เป็น Destination โหนด ในที่นี่จะทำงานเป็น PAN Coordinator และ Gateway ในตัว นอกนั้นโหนดที่เหลือ R1 ถึง R7 จำนวน 7 โหนด จะทำหน้าที่เป็นโหนด Router โดยมีการใช้โพรโทคอล AODV ในการให้เส้นทางภายในเครือข่าย



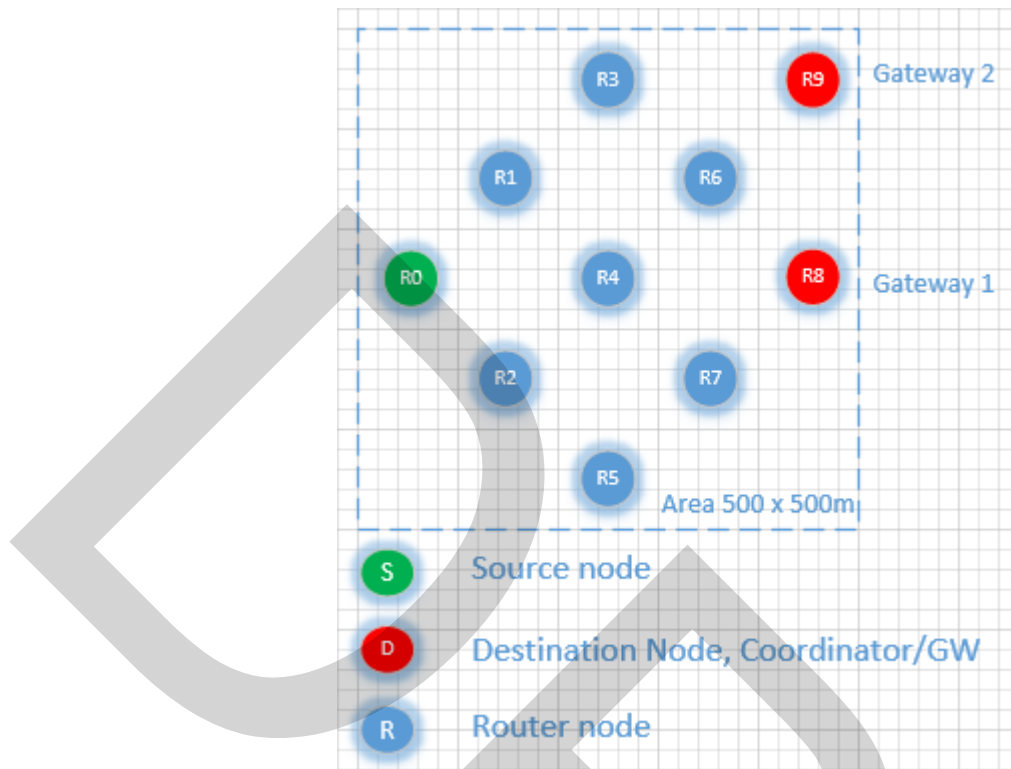
ภาพที่ 3.1 โครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบมี 1 Gateway

เมื่อมีการกำหนดตัวโครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบ 1 Gateway แล้ว ได้ทำการกำหนดค่าคุณสมบัติต่างๆ ในโหนด และเครือข่ายที่จะทำการจำลอง โดยมีรายละเอียดตามตารางข้างล่าง

ตารางที่ 3.2 การตั้งค่าต่างๆ บนทอพอโลยี แบบ 1 Coordinator/Gateway

พารามิเตอร์	ค่าที่กำหนด
Number of Nodes	9
Number of FFD	9
Circle Radius R (m)	10
Number of Sources	1
Number of Sink	1 (PAN Coordinator)
Sink Position	Right of the area
Node Mobility	None
MAC Protocol	IEEE 802.15.4 (Non-Beacon mode with unslotted CSMA/CA mechanism)
Propagation Model	Two Ray Ground Model
Queue Size	50
Data Transfer Model	Direct Data Transmission
Traffic Model	Poisson traffic
Packet Type	CBR
Packet Size (bytes)	100
Simulation Area (m*m)	500*500
Packet Quantity	5000, 10000, 15000, 20000, 25000
Simulation Time (sec)	45

2) แบบ 2 PAN Coordinator จะประกอบไปด้วยโหนดที่เป็น FFD โหนด จำนวน 10 โหนด มีโหนดที่ทำหน้าที่เป็น Source จำนวน 1 โหนด และโหนดที่ทำหน้าที่เป็นตัว Sink จำนวน 2 โหนด โดยจะมีการส่งข้อมูลที่เป็น CBR Packet จาก Source โหนด ไปยัง Sink โหนด กำหนดขนาดของทอพอโลยี จะเป็น 500 x 500 เมตร จากรูปข้างล่าง จะประกอบไปด้วย โหนด R0 ถึง R9 รวมเป็นทั้งหมด 10 โหนด โดยโหนด R0 จะทำหน้าที่เป็น Source โหนด และโหนด R8, R9 จะทำหน้าที่เป็น Destination โหนด ในที่นี้จะทำงานเป็น PAN Coordinator และ Gateway ในตัว นอกนั้นโหนดที่เหลือ R1 ถึง R7 จำนวน 7 โหนด จะทำหน้าที่เป็น โหนด Router โดยมีการใช้โปรโตคอล AODV ในการให้เส้นทางภายในเครือข่าย



ภาพที่ 3.2 โครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบมี 2 Gateways

เมื่อมีการกำหนดตัวโครงสร้างของเครือข่ายที่จะใช้ในการจำลอง แบบ 1 Gateway แล้ว ได้ทำการกำหนดค่าคุณสมบัติต่างๆ ในโหนด และเครือข่ายที่จะทำการจำลอง โดยมีรายละเอียดตามตารางข้างล่าง

ตารางที่ 3.3 การตั้งค่าต่างๆ บนทอพอโลยี แบบ 2 Coordinator/Gateway

พารามิเตอร์	ค่าที่กำหนด
Number of Nodes	10
Number of FFD	10
Circle Radius R (m)	10
Number of Sources	1

ตารางที่ 3.3 (ต่อ)

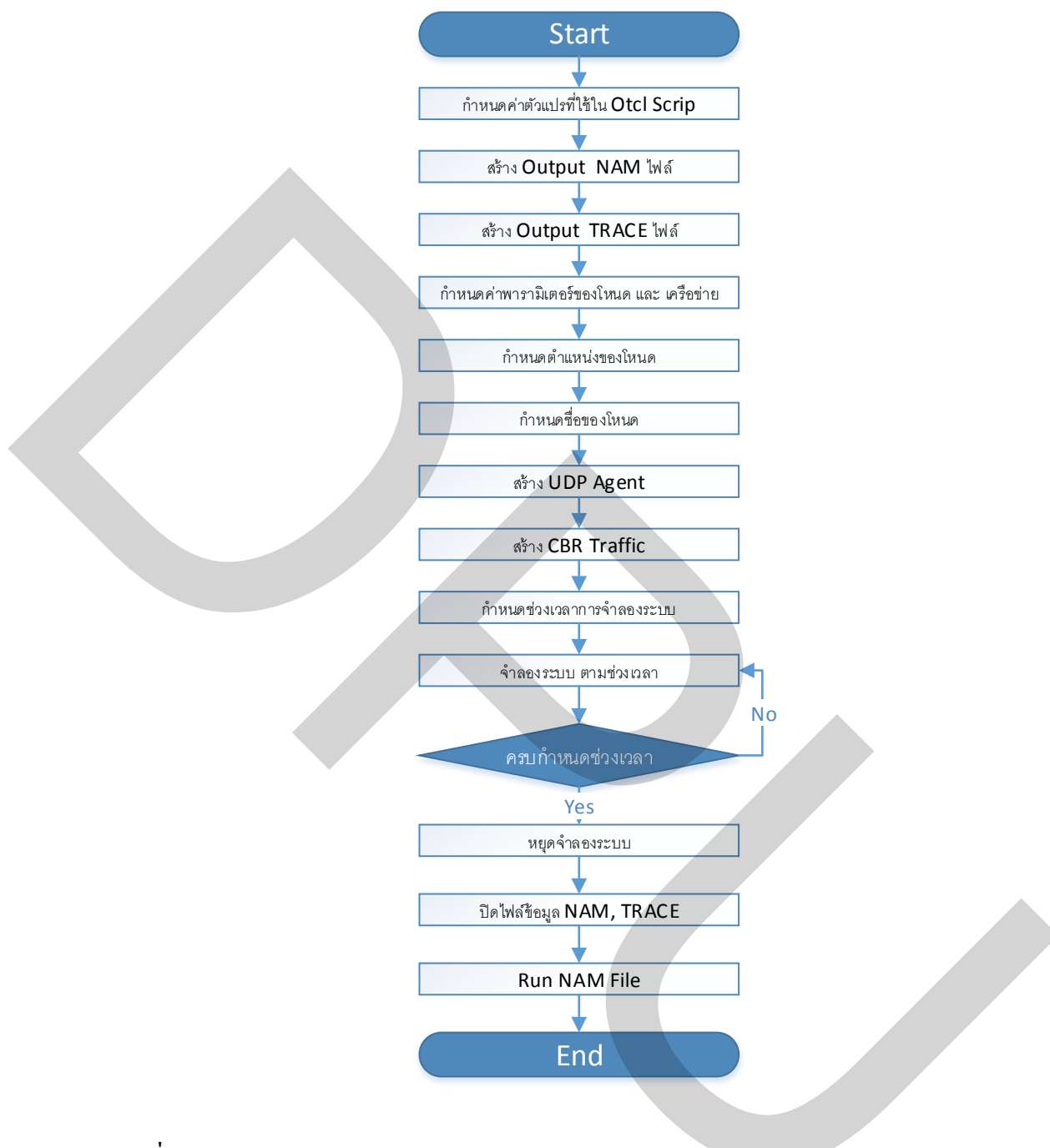
พารามิเตอร์	ค่าที่กำหนด
Number of Sink	2 (PAN Coordinator)
Sink Position	Right of the area
Node Mobility	None
MAC Protocol	IEEE 802.15.4 (Non-Beacon mode with unslotted CSMA/CA mechanism)
Propagation Model	Two Ray Ground Model
Queue Size	50
Data Transfer Model	Direct Data Transmission
Traffic Model	Poisson traffic
Packet Type	CBR
Packet Size (bytes)	100
Simulation Area (m*m)	500*500
Packet Quantity	5000, 10000, 15000, 20000, 25000
Simulation Time (sec)	45

3.3.2 เริ่มกำหนดค่าพารามิเตอร์ในโปรแกรม NS2

กำหนดค่าพารามิเตอร์ที่จะใช้ในตัวโปรแกรม Network Simulation 2 (NS-2) ตามตารางที่ 3.2 และ 3.3 โดยจะสามารถกำหนดได้โดยการเขียนโปรแกรมผ่าน Otcl Script โดยมีลำดับการกำหนดค่าพารามิเตอร์ต่างๆ ตามแผนผังการกำหนดค่าในโปรแกรมภาษา Otcl Script โดยตัวโปรแกรมที่เขียนขึ้นจะใช้โปรแกรม editor ชื่อ gedit ในการสร้างส่วนของ Otcl Scrip โดยใช้นามสกุล tcl ซึ่งเป็นโปรแกรมที่ทำงานอยู่บน Ubuntu 12.04 และใช้ตัวโปรแกรม Network Simulator 2.35 ในการจำลองการทำงาน โดยใช้คำสั่งตามรูปด้านล่างในการเริ่มโปรแกรม

```
chanadej@ubuntu:~/thesis$ ns wsn2gw.tcl
```

ภาพที่ 3.3 คำสั่งการใช้โปรแกรม Network Simulator 2.35 เพื่อสั่งงาน Otcl Script



ภาพที่ 3.4 แผนผังการกำหนดค่าในโปรแกรมภาษา Otcl Script สำหรับโมเดลจำลอง

โดยลำดับขั้นตอนการทำงานของแผนผังมีดังนี้

1) กำหนดค่าตัวแปรที่ใช้ใน Otcl Scrip ในงานวิจัยนี้มีการกำหนดตัวแปรต่างๆ ดังนี้ เพื่อใช้เก็บค่าพารามิเตอร์ต่างๆ ที่จะใช้ในการกำหนดโหนด และเครือข่าย

ตารางที่ 3.4 ชื่อและค่าตัวแปรที่กำหนด

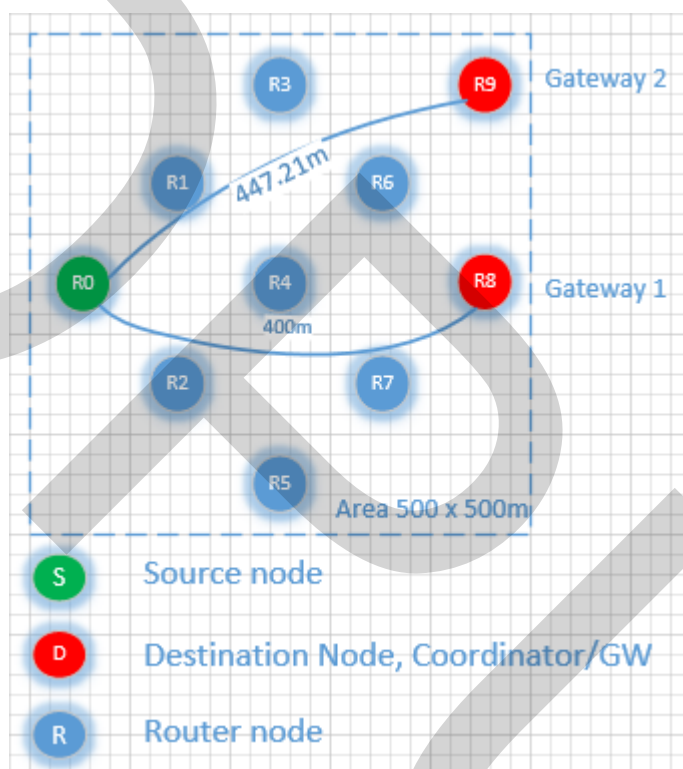
ชื่อตัวแปร	ค่าตัวแปร	การใช้งาน
Chan	WirelessChannel	การกำหนดประเภทของช่องการสื่อสารเป็นแบบไร้สาย
Prop	TwoRayGround	การกำหนดค่าชนิดของ radio-propagation
netif	802_15_4	การกำหนดค่าชนิดของ Network interface
mac	802_15_4	การกำหนดค่าชนิดของสื่อกลางที่ใช้ในการเข้าถึงช่องการสื่อสาร
ifq	PriQueue	การกำหนดค่าชนิดของการ Queue
ant	OmniAntenna	การกำหนดค่าชนิดของสายอากาศเป็นแบบ Omni
ifqlen	50	การกำหนดค่าขนาดของ Buffer สูงสุดในแต่ละโหนด
Nn	9	การกำหนดค่าจำนวนโหนดทั้งหมดของเครือข่าย เป็น 9 โหนด
rp	AODV	การกำหนดค่าชนิดของ Routing Protocol เป็น AODV
x	500	การกำหนดค่าขนาดของเครือข่ายในแนวนอนเป็น 500 เมตร
y	500	การกำหนดค่าขนาดของเครือข่ายในแนวตั้งเป็น 500 เมตร
stop	50	การกำหนดค่าช่วงเวลาทั้งหมดของการทำงานในโปรแกรม เป็น 50 วินาที
Energymodel	EnergyModel	การกำหนดค่าชนิดของโมเดลพลังงานที่ใช้
Initialenergy	100	การกำหนดค่าเริ่มต้นของพลังงานที่ใช้

2) สร้าง Output NAM ไฟล์ เพื่อใช้ในการแสดงผลบน โปรแกรม Network Animation โดยในงานวิจัยนี้มีการสร้างไฟล์ wsn.nam

3) สร้าง Output Trace ไฟล์ เพื่อใช้ในการเก็บผลการจำลองการทำงานที่เหตุการณ์ต่างๆ โดยในงานวิจัยนี้มีการสร้างไฟล์ wsn.tr

4) สร้างโหนด FFD และกำหนดค่าโหนด และค่าเริ่มต้นเครือข่าย โดยการเอาค่าที่ใส่ไว้ในตัวแปรมาใช้ในแต่ละโหนด

5) กำหนดตำแหน่งของโหนดต่างๆ บนเครือข่าย โดยโหนด Source มีระยะห่างจากโหนด Destination 1 (Gateway 1) อยู่ 400 เมตร และมีระยะห่างจากโหนด Destination 2 (Gateway 2) อยู่ 447.21 เมตร โดยมีลักษณะเป็น Circular Topology ตามรูปด้านล่าง



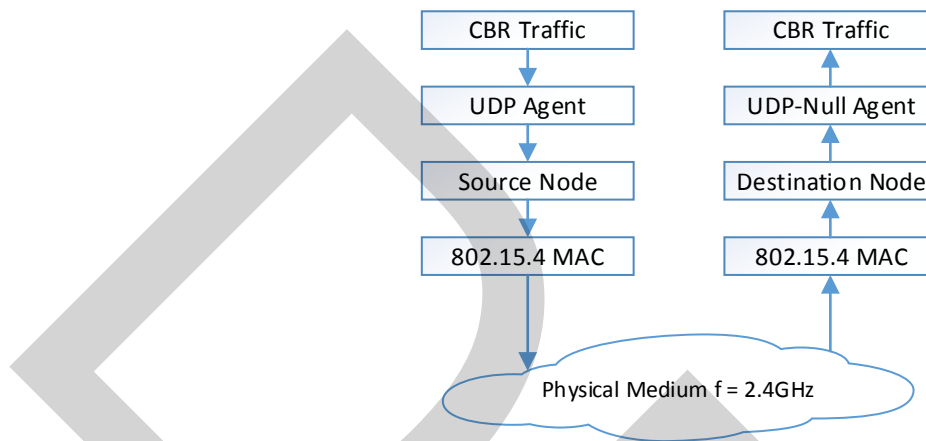
ภาพที่ 3.5 โครงสร้าง WSN และระยะห่างระหว่างโหนดต้นทาง กับโหนดปลายทาง

6) กำหนดชื่อโหนดเพื่อให้แสดงผลบนโปรแกรม Network Animation

7) สร้าง UDP Agent โดยจะมีการกำหนดส่วนของตัวที่เป็น Source Node ให้เป็น UDP และตัวที่เป็น Destination Node ให้เป็นแบบ Null เพื่อคอยรับข้อมูล

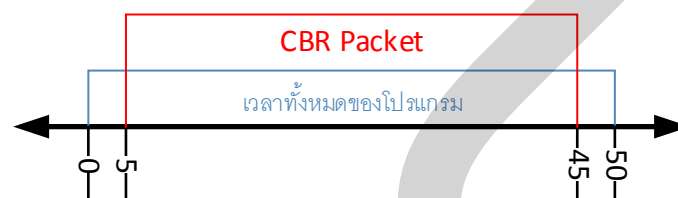
8) สร้าง Packet Traffic ที่เป็นแบบ CBR เพื่อส่งเข้าไปในตัว Source Node ผ่าน UDP Agent เมื่อ CBR Packet ถูกส่งเข้ามาในตัว UDP Agent ก็จะมีการเชื่อมต่อการส่งข้อมูลไปยังตัว Source node และส่งผ่านไปยัง MAC ที่เป็น 802.15.4 จากนั้นก็ส่งออกไปยัง Physical Layer ที่เป็น

แบบไร้สายโดยอาศัยคลื่นความถี่ 2.4GHz เป็นสื่อกลาง เพื่อส่งต่อไปยัง Destination Node ผ่านทาง 802.15.4 MAC ของโหนดปลายทางและส่งขึ้นไปยังตัว Agent เป็นการจบการส่ง CRB Packet



ภาพที่ 3.6 การส่งข้อมูลระหว่างโหนด Source กับโหนด Destination ใน NS2

9) กำหนดช่วงเวลาในการทดสอบแบบจำลอง โดยมีการกำหนดช่วงเวลาในการทดสอบ 50 วินาที ดังนี้



ภาพที่ 3.7 ช่วงเวลาในส่ง CBR Packet กรณีที่มี 1 Gateway

10) เริ่มทำงานโปรแกรมจำลองในระบบ โดยมีช่วงการทำงาน 50 วินาที และตัว CBR Packet เริ่มส่งไปยัง Destination Node ที่เวลา 5 วินาที และหยุดส่งที่เวลา 45 วินาที และโปรแกรมหยุดทำงานที่ 50 วินาที โดยมีช่วงเวลาการส่ง CBR Packet อยู่ที่ 40 วินาที

11) เมื่อโปรแกรมทำงานเสร็จก็จะทำการปิดไฟล์ wsn.nam กับ wsn.tr โดยจะบันทึกไว้ในที่อยู่อีกด้วยกับ wsn.tcl ไฟล์

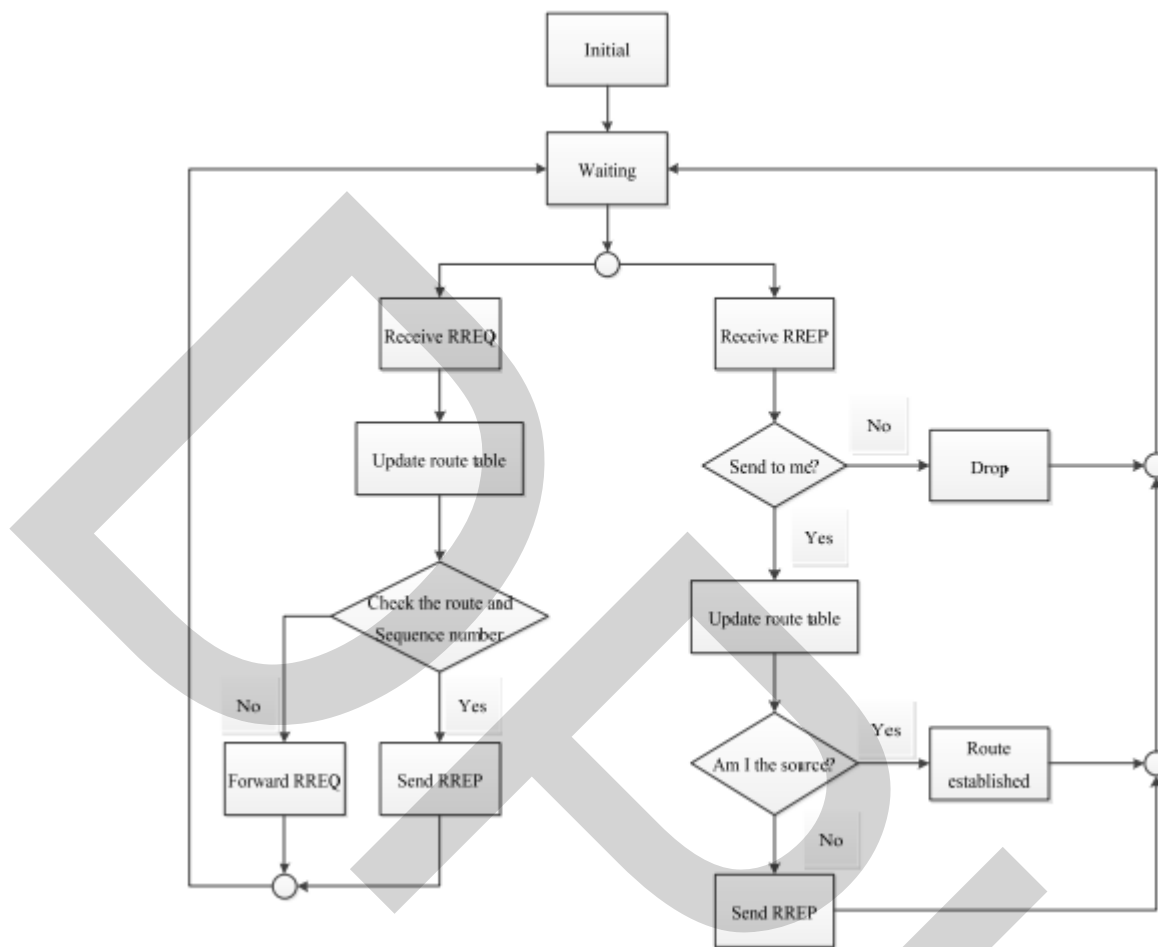
12) เรียกโปรแกรม Network Animation ขึ้นมาเพื่อแสดงการจำลองการทำงานของเครือข่ายในรูปของกราฟฟิคภาพเคลื่อนไหว

13) จบการทำงานของโปรแกรม นำผลของการจำลองเครือข่ายเช่นเซอร์ไรรายที่ได้ในรูปของ trace file ที่อยู่ในไฟล์ wsn.tr มาหาค่าที่ต้องการ

3.3.3 จำลองการทำงานของเครือข่ายในรูปแบบภาพเคลื่อนไหว

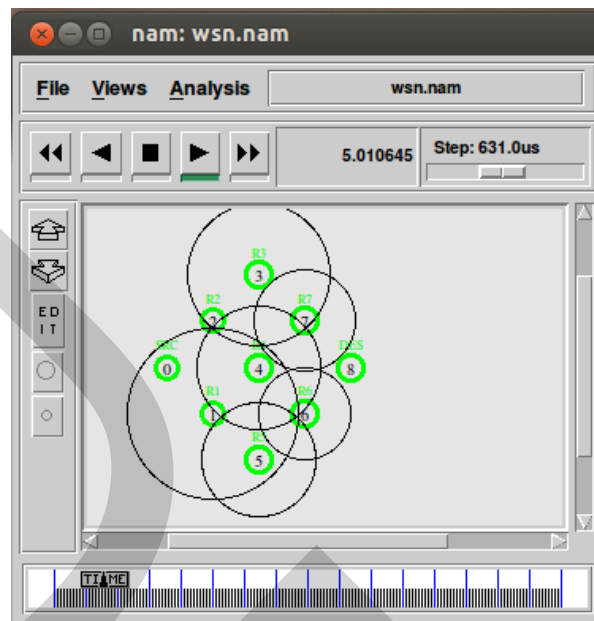
การจำลองการทำงานของเครือข่ายในรูปแบบภาพเคลื่อนไหว เพื่อให้เห็นเส้นทางการเคลื่อนที่ของข้อมูล CBR packets โดยเมื่อตัวโปรแกรม Network Simulator ทำงานเสร็จแล้วก็จะทำการสั่งให้มีการเรียกตัวโปรแกรม Network Animation เพื่อมาแสดงผลไฟล์ที่มีนามสกุล nam ในที่นี้ใช้ชื่อ wsn.nam โดยในการแสดงผลนั้นจะสามารถสั่งให้ตัวโปรแกรมแสดงเหตุการณ์ที่เกิดขึ้นต่างๆ ในเครือข่ายตามเวลาที่ได้กำหนดไว้ตั้งแต่เริ่มต้น ในงานวิจัยนี้ใช้เวลาทั้งหมด 50 วินาที โดยบนหน้าจอของตัวโปรแกรม Network Simulator จะแสดงรูปของโหนดที่สร้างขึ้น ในที่นี้จะมีโหนดทั้งหมด 9 และ 10 โหนด โดยกรณีที่มี 9 โหนด หมายถึงกรณีที่มีโหนดที่เป็นต้นทาง Source โหนดจำนวน 1 โหนด คือ R0 และมี Gateway จำนวน 1 โหนด คือ R8 และที่เหลืออีก 7 โหนด จะเป็นโหนด Router คือ R1 – R7 ส่วนกรณีที่มี 10 โหนด หมายถึงกรณีที่มีโหนดที่เป็นต้นทาง Source โหนดจำนวน 1 โหนด คือ R0 และมี Gateway จำนวน 2 โหนด คือ R8, R9 และที่เหลืออีก 7 โหนด คือ R1 – R7 จะเป็นโหนด Router

เมื่อมีการกดปุ่มเริ่มทำงาน ตัวโหนดที่ทำหน้าที่เป็น Source “R0” ก็จะเริ่มทำการ Broadcast packet “RREQ” ไปยังโหนดข้างเคียง “R1”, “R2”, “R4” เมื่อโหนดข้างเคียงได้รับ packet “RREQ” ก็จะตรวจสอบว่าตัวมันเองเป็นโหนดปลายทางหรือไม่ ถ้าไม่ใช่ก็จะส่งต่อ packet “RREQ” ไปยังโหนดข้างเคียงต่อไป “R3”, “R7”, “R6”, “R5”, “R8” เมื่อโหนด “R8” ได้รับ packet “RREQ” และตรวจสอบแล้วว่า ตัวเองเป็นโหนด Destination ก็จะสร้าง packet “RREP” ตอบกลับไปยังเส้นทางที่ส่ง packet “RREQ” ที่มีหมายเลขลำดับนี้มา คือโหนด “R4” จากนั้นเมื่อโหนด “R8” ได้รับ packet “RREQ” ที่มีหมายเลขลำดับเดียวกัน ที่มาจากโหนดอื่นๆ ก็จะ Drop packet “RREQ” นั้นทิ้งไป เพื่อป้องกันการเกิดลูปในเครือข่าย เมื่อ packet “RREP” ที่ตอบกลับ ไปถึงตัวโหนด Source “R0” ตัวโหนด Source “R0” ก็จะส่ง packet “CBR” ไปยังเส้นทางที่รับมา ในกรณีที่เส้นทางจากโหนด “R4” ที่ไปยังโหนด “R8” ไม่สามารถใช้งานได้ ตัวโหนด “R4” ก็จะกระจาย Broadcast packet “RRER” กลับไปยังโหนดต้นทาง ก็คือโหนด “R0” เพื่อให้เริ่มกระบวนการค้นหาเส้นทางใหม่

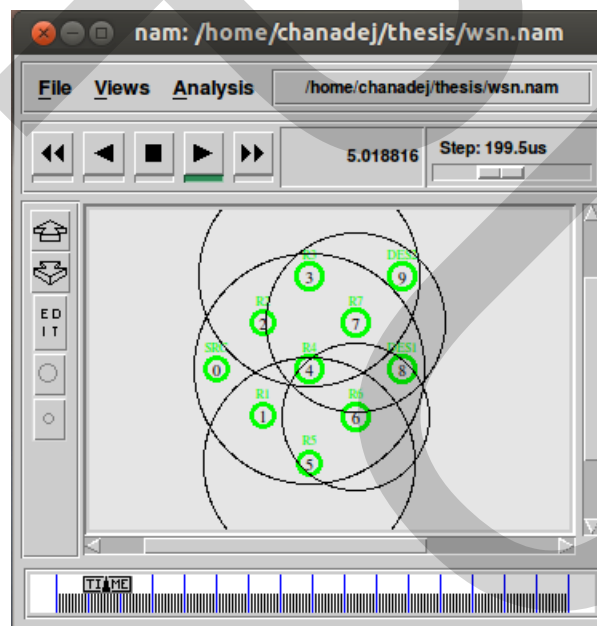


ภาพที่ 3.8 ลำดับขั้นตอนการส่งข้อมูลของตัวโปรโตคอล AODV

จากภาพที่ 3.8 ตัวโปรโตคอล AODV ในสถานะที่กำลังรอที่จะส่งข้อมูล เมื่อโหนดได้รับการร้องขอเส้นทาง “RREQ” จะมีการปรับปรุงเส้นทางในตารางเส้นทาง โดยการตรวจสอบเส้นทางและหมายเลขลำดับ ถ้ามีโหนดปลายทางในตารางเส้นทาง และหมายเลขลำดับเป็นปัจจุบัน ก็จะส่ง “RREP” กลับไปยังโหนดต้นทางตามเส้นทางเดิมที่รับ “RREP” มา แต่ถ้าไม่มีโหนดปลายทางในตารางเส้นทาง ก็จะทำการส่งต่อ “Forward RREQ” ไปยังโหนดข้างเคียง เมื่อโหนดใดๆ ได้รับ “RREP” ก็จะตรวจสอบว่าเป็นของตัวเองหรือไม่ ถ้าไม่ใช่ก็จะ Drop “RREP” นั้นทิ้งไป แต่ถ้าใช่ ก็จะปรับปรุงเส้นทางในตารางเส้นทางของตัวเอง แล้วคว้าตัวเองเป็นโหนดต้นทางหรือไม่ ถ้าไม่ใช่ก็จะส่ง “RREP” ต่อไป ถ้าใช่ ก็จะสร้างเส้นทาง เพื่อให้ UDP ส่งข้อมูลไปยังโหนดปลายทาง



ภาพที่ 3.9 โปรแกรม Network Animation ขณะจำลองการทำงานของ WSN 1 Gateway



ภาพที่ 3.10 โปรแกรม Network Animation ขณะจำลองการทำงาน WSN แบบ 2 Gateways

3.3.4 วิเคราะห์ผลการจำลองการทำงานของ WSN แบบ 1 Gateway

นำผลการจำลองการทำงานของเครือข่ายที่อยู่ใน trace file มาทำการวิเคราะห์ในโปรแกรม LibreOffice3 ซึ่งเป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการ Ubuntu โดยรายละเอียดของ Trace ไฟล์ที่ถูกสร้างขึ้น จะมีตัวอย่างรายละเอียดตามภาพที่ 3.8 ด้านล่าง

```

r 5.002962471 1 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002962471 2 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002962667 4 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002962943 3 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002962943 5 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002963054 6 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002963054 7 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
r 5.002963333 8 RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001]
s 5.002963333 8 RTR --- 0 AODV 44 [0 0 0] [energy 99.999229 ei 0.000 es 0.000 et 0.000 er 0.001] -----
N -t 5.003609 -n 7 -e 99.998711
N -t 5.003609 -n 6 -e 99.998711
N -t 5.003609 -n 4 -e 99.998711
N -t 5.003609 -n 5 -e 99.998711
N -t 5.003609 -n 3 -e 99.998711
N -t 5.003609 -n 2 -e 99.998711
N -t 5.003609 -n 1 -e 99.998711
N -t 5.003610 -n 0 -e 99.998067

```

ภาพที่ 3.11 โครงสร้างของ Trace ไฟล์ ที่สร้างขึ้น

โดยใน Otcl script ที่เขียนขึ้นจะกำหนดให้มีการสร้าง Trace ไฟล์ ชื่อ wsn.tr ที่จะบันทึกเหตุการณ์ที่เกิดขึ้น ที่ส่วนของ Router และ Agent เพื่อดูเหตุการณ์ที่เกิดขึ้นในแต่ละ packet โดยมีโครงสร้างของ Trace ไฟล์จะมีรายละเอียดของแต่ละคอลัมน์ดังนี้

รายละเอียดคอลัมน์ที่ 1-7

Event	Time	Node	Trace Level	Flags	SEQNO	Packet Type
-------	------	------	-------------	-------	-------	-------------

รายละเอียดคอลัมน์ที่ 8-13

Packet Size	MAC [a b c d]	Energy ei es et er	Flags	Network	Transport	Message
-------------	---------------	--------------------	-------	---------	-----------	---------

คอลัมน์ที่ 1 (Event) แสดงรายละเอียดของเหตุการณ์ที่เกิดขึ้นเวลานั้นๆ โดยมีความหมายของแต่ละ Event ดังนี้

- 1) s: มีการส่งข้อมูล
- 2) r: มีการรับข้อมูล
- 3) D: ข้อมูลถูก Drop

คอลลัมน์ที่ 2 (Time) แสดงเวลาที่เกิดขึ้นที่เหตุการณ์นั้นๆ มีหน่วยเป็นวินาที

คอลลัมน์ที่ 3 (Node) แสดงหมายเลขโหนดที่เกิดเหตุการณ์นั้นๆ

คอลลัมน์ที่ 4 (Trace Level) บอกถึงส่วนที่มีการเก็บข้อมูล แต่ละชื่อมีความหมายดังนี้

- 1) RTR: เกี่ยวกับเส้นทาง
- 2) AGT: เกี่ยวกับตัวโปรแกรมประยุกต์ เช่น UDP

คอลลัมน์ที่ 5 (Flags) เป็นการแสดงเครื่องหมายต่างๆ

คอลลัมน์ที่ 6 (SEQNO) บอกถึงหมายเลขลำดับของแพ็คเกจ

คอลลัมน์ที่ 7 (Packet Type) บอกถึงชนิดของแพ็คเกจที่เกิดขึ้น เช่น AODV, CBR

คอลลัมน์ที่ 8 (Packet Size) บอกถึงขนาดของแพ็คเกจมีหน่วยเป็น ไบต์ (Byte)

คอลลัมน์ที่ 9 (MAC [a b c d]) มีความหมายแต่ละตำแหน่งดังนี้

- 1) ตำแหน่ง a -- the packet duration in mac layer header
- 2) ตำแหน่ง b -- the mac address of destination
- 3) ตำแหน่ง c -- the mac address of source
- 4) ตำแหน่ง d -- the mac type of the packet body

คอลลัมน์ที่ 10 (Energy ei es et er) บอกถึงระดับพลังงานในโหนดนั้นๆ โดยมีรายละเอียดดังนี้

- 1) ตำแหน่ง Energy – ระดับพลังงานที่เหลืออยู่ในโหนดนั้นๆ
- 2) ตำแหน่ง ei – คือ Idle Power เป็นค่าพลังงานที่ถูกใช้ไปในขณะ idle มีหน่วยเป็นวัตต์
- 3) ตำแหน่ง es – คือ Sleep Power เป็นค่าพลังงานที่ถูกใช้ไปในขณะที่โหนดหลับอยู่ มีหน่วยเป็นวัตต์
- 4) ตำแหน่ง et – คือ Transmitting Power เป็นค่าพลังงานที่ถูกใช้ไปในขณะที่มีการส่งข้อมูล มีหน่วยเป็นวัตต์
- 5) ตำแหน่ง er -- คือ Receiving Power เป็นค่าพลังงานที่ถูกใช้ไปในขณะที่มีการรับข้อมูล มีหน่วยเป็นวัตต์

คอลลัมน์ที่ 11 (Flags) เป็นการแสดงเครื่องหมายต่างๆ

คอลลัมน์ที่ 12 (Network) เป็นการแสดงค่าในส่วนของเครือข่ายประกอบไปด้วย

- 1) src_addr, dst_addr, TTL, next_hop_addr

คอลลัมน์ที่ 12 (Transport) เป็นการแสดงค่าในส่วน of ชั้น Transport ประกอบไปด้วย

- 1) seq_no, ack_no, no_of_fwds, opt_no_of_fwds

คอลัมน์ที่ 13 (Message) บอกประเภทของข้อมูลที่โต้ตอบกัน เช่น Request, Reply

ตัวอย่างข้อมูลใน Trace ไฟล์ที่เขียนขึ้น เช่น `s 5.000000000 0 AGT --- 18455 cbr 100 [0 0 0 0] [energy 91.093916 ei 0.000 es 0.000 et 0.438 er 8.468] ----- [0:255 -1:255 30 0]`

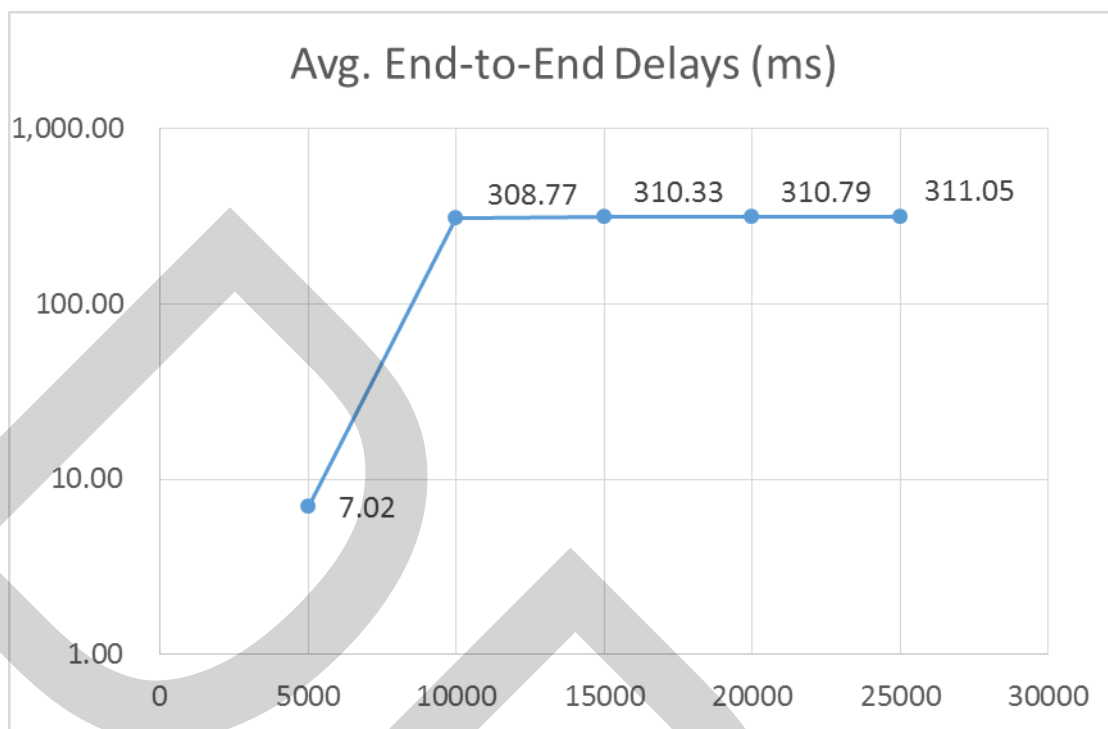
หมายความว่า มีการส่งข้อมูลที่เวลา 5.0 วินาที และเกิดขึ้นที่โหนด “0” โดยมีชนิดของข้อมูลเป็นแบบ “cbr” และหมายเลขลำดับของแพ็คเกจเป็น “18455” มีขนาดข้อมูลเป็น 100 Byte มีระดับพลังงานที่เหลืออยู่ 91.093916 วัตต์ พลังงานที่ใช้ในสภาวะ Idle = 0 วัตต์ พลังงานที่ใช้ในสภาวะ Sleep = 0 วัตต์ พลังงานที่ใช้ในการส่งข้อมูล = 0.438 วัตต์ พลังงานที่ใช้ในการรับข้อมูล = 8.468 วัตต์ มีโหนดต้นทาง source address “0” port “255” มีโหนดปลายทาง destination address “-1” คือการกระจาย Broadcast port “255” มี TTL “30” และมีโหนดถัดไปจำนวน “0”

การเปิดไฟล์ wsn.tr เพื่อทำการวิเคราะห์ข้อมูลที่ได้จากการจำลอง ในงานวิจัยนี้ได้ใช้โปรแกรม LibreOffice 3 เนื่องจากเป็นโปรแกรม Freeware ที่มีหาได้ในตัว Ubuntu ที่เป็นโปรแกรม Freeware GNU ด้วยคุณสมบัติของตัวโปรแกรม LibreOffice 3 ที่สามารถใช้ฟังก์ชันในการคำนวณทางสถิติ และการกำหนดเงื่อนไขได้ เช่น ฟังก์ชัน SUM, MAX, MIN, AVERAGE, IF ทำให้สามารถหาค่าผลลัพธ์ต่างๆ ได้

เมื่อนำข้อมูลที่ได้อาวิเคราะห์แล้วสามารถทำการสร้างกราฟแสดงผลการทดลองโดยใช้โปรแกรม LibreOffice 3 โดยในภาพที่ 3.12 แสดงตัวอย่างตารางข้อมูลที่ได้จากการวิเคราะห์ข้อมูลใน wsn.tr โดย เมื่อได้ข้อมูลที่ต้องการทั้งหมดแล้ว จะทำการสร้างกราฟแสดงผลข้อมูลที่ได้อิงภาพที่ 3.13

Parameter	AODV				
CBR Packet Size (byte)	100	100	100	100	100
CBR Packet Rate (Mb)	0.1	0.2	0.3	0.4	0.5
CBR Interval	0.008000	0.004000	0.002667	0.002000	0.001600
Gateway	1	1	1	1	1
Start Time (sec)	5.00	5.00	5.00	5.00	5.00
Stop Time (sec)	45.00	45.00	45.00	45.00	45.00
Data send (Packets)	5,000.00	10001	15000	20000	25000
Data Receive (Packets)	5,000.00	6443	6442	6443	6443
Avg. Throughput (Kbps) Source Node	12.50	24.81	37.22	49.61	62.02
Avg. Throughput (Kbps) Destination Node	15.00	19.18	19.18	19.18	19.18
Avg. Throughput Ratio	120%	77%	52%	39%	31%
Avg. Delay (ms)	7.02	308.77	310.33	310.79	311.05
Packet Delivery Ratio (%)	100%	64%	43%	32%	26%
Packet Drops (Packets)	-	3558	8558	13557	18557

ภาพที่ 3.12 ข้อมูลใน trace file หลังจากแบ่งข้อมูลแต่ละ column แล้ว



ภาพที่ 3.13 กราฟของค่าเฉลี่ยการหน่วงเวลาแบบจุดต่อจุด

จากภาพที่ 3.13 เป็นกราฟที่แสดงค่าเฉลี่ยการหน่วงเวลาแบบจุดต่อจุด โดยเริ่มจาก โหนดต้นทาง R0 ไปยังโหนดปลายทาง R8 โดยมีการเปลี่ยนแปลงปริมาณ packet CBR ที่ส่งเข้ามา ในเครือข่ายเซนเซอร์ไร้สายที่มีตัว Coordinator/Gateway Node เพียง 1 ตัว ในแกนนอนแสดงให้เห็นถึงปริมาณ ความหนาแน่นของ packet CBR ที่เวลาเท่ากันคือ 40 วินาที เมื่อความหนาแน่นของ packet CBR มีปริมาณสูงขึ้น จะทำให้เกิดค่าเฉลี่ยการหน่วงเวลาแบบจุดต่อจุด มากขึ้นตาม

3.3.5 สร้างทอพอโลยี ของ WSN แบบ 2 gateways

เพิ่มตัว Gateway ตัวที่ 2 เข้าไปใน WSN เพื่อให้เห็นถึงความแตกต่าง จึงทำการจำลอง ตัวเครือข่ายเซนเซอร์ไร้สายที่มี Gateway 1 ตัว โดยใช้ตัวโปรโตคอล AODV แล้วใส่รูปแบบของ ข้อมูลที่เป็น CBR (constant bit rate) โดยจะทำการปรับเปลี่ยนปริมาณความหนาแน่นของข้อมูลที่ ส่งเข้าไปยังตัวโหนดต้นทาง R0 โดยมีปริมาณความหนาแน่นของข้อมูลต่อเวลาดังนี้

ตารางที่ 3.5 ค่าปริมาณความหนาแน่นของข้อมูลต่อเวลา

ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5
5000 packets	10000 packets	15000 packets	20000 packets	25000 packets
125 packets/sec	250 packets/sec	375 packets/sec	500 packets/sec	625 packets/sec

ในการกำหนดค่าของปริมาณความหนาแน่นของข้อมูลในส่วนของโปรแกรมจะสามารถทำได้โดยกำหนดค่าของ CBR Packet Interval ในส่วนของ Otcl script ตอนกำหนดรายละเอียดของข้อมูล CBR โดยมีรายละเอียดดังนี้

ตารางที่ 3.6 ค่าที่ใช้ในการกำหนดค่า Packet Interval

CBR Interval 1	CBR Interval 2	CBR Interval 3	CBR Interval 4	CBR Interval 5
0.008	0.004	0.002667	0.002	0.0016

ตัวอย่างคำสั่งที่ใช้ในการกำหนดค่าชนิดข้อมูลที่เป็น CBR ในการกำหนดค่านั้นจะต้องมีการกำหนดค่าของ Packet Size : 100 และค่า Packet Interval เมื่อกำหนดเรียบร้อยแล้วก็ทำการเชื่อมต่อ CBR packet เข้ากับตัว Agent “udp_src”

```
#Setup CBR traffic
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp_src
$cbr set packetSize_ 100 #(CBR Packet is 100 byte)
$cbr set interval_ 0.008 #ใช้ในการกำหนดค่า Packet Interval
```

ภาพที่ 3.14 ตัวอย่างการกำหนดค่าของ CBR Packet

3.3.6 ประเมินผลการจำลองเบื้องต้นใน WSN

ในประเมินผลการวิจัยนี้ จะต้องหาตัวแปรต่างๆ โดยค่าตัวแปรที่ต้องการหาหลังจากจำลองการทำงานของเครือข่ายเซนเซอร์ไร้สาย จะประกอบไปด้วยตัวแปรต่างๆ ดังนี้

1) จำนวน CBR Packet ที่โหนดต้นทาง ในการจำลองเครือข่ายนี้คือ R0 โดยสามารถเขียนเป็นชุดโค้ด (Pseudo Code) ได้ดังนี้


```

ALGORITHM SOURCE_CBR (NODE, TYPE, TOTAL_DATA)
INPUT NODE          #NODE NUMBER
INPUT TYPE          #TYPE OF DATA
INPUT TOTAL_DATA    #NUMBER OF LINE
OUTPUT SOURCE_CBR   #TOTAL CBR_PACKET AT SOURCE NODE
SOURCE_CBR = 0
FOR I=0, I=TOTAL_DATA, I++
  IF NODE = R0 AND TYPE = CBR THEN
    SOURCE_CBR = SOURCE_CBR + 1
  NEXT
ELSE NEXT
END FOR
RETURN (SOURCE_CBR)

```

ภาพที่ 3.15 ชูโดโค้ด (Pseudo Code) การหาจำนวน CBR Packet ที่โหนดต้นทาง

2) จำนวน CBR Packet ที่โหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8 โดยสามารถเขียนเป็นชูโดโค้ด (Pseudo Code) ได้ดังนี้

```

ALGORITHM DESTINATION_CBR (NODE, TYPE, TOTAL_DATA)
INPUT NODE          #NODE NUMBER
INPUT TYPE          #TYPE OF DATA
INPUT TOTAL_DATA    #NUMBER OF LINE
OUTPUT DESTINATION_CBR #TOTAL CBR_PACKET AT DESTINATION NODE
DESTINATION_CBR = 0
FOR I=0, I=TOTAL_DATA, I++
  IF NODE = R8 AND TYPE = CBR THEN
    DESTINATION_CBR = DESTINATION_CBR + 1
  NEXT
ELSE NEXT
END FOR
RETURN (DESTINATION_CBR)

```

ภาพที่ 3.16 ชูโดโค้ด (Pseudo Code) การหาจำนวน CBR Packet ที่โหนดปลายทาง

3) ค่าเฉลี่ย CBR Packet ต่อเวลา ที่โหนดต้นทาง ในการจำลองเครือข่ายนี้คือ R0 โดยสามารถเขียนเป็นชูโดโค้ด (Pseudo Code) ได้ดังนี้

```

ALGORITHM SOURCE_AVG_CBR (NODE[],TYPE[],TOTAL_DATA[],TIME,TRACE_LEVEL[],PACKET_SIZE[])
INPUT NODE           #NODE NUMBER
INPUT TYPE           #TYPE OF DATA
INPUT TOTAL_DATA     #NUMBER OF LINE
INPUT TIME           #TOTAL TIME "SEC"
INPUT TRACE_LEVEL    #POINT OF MEASUREMENT "AGT"
INPUT PACKET_SIZE    #PACKET_SIZE "BYTE"
OUTPUT SOURCE_AVG_CBR # AVERAGE CBR_PACKET AT SOURCE NODE
SOURCE_AVG_CBR = 0
FOR I=0, I=TOTAL_DATA, I++
  IF NODE[I] = R0 AND TYPE[I] = CBR AND TRACE_LEVEL = AGT THEN
    SOURCE_AVG_CBR = SOURCE_AVG_CBR + PACKET_SIZE[I]
  NEXT
ELSE NEXT
END FOR
SOURCE_AVG_CBR = SOURCE_AVG_CBR/TIME
RETURN (SOURCE_AVG_CBR)

```

ภาพที่ 3.17 ชูโดโค้ด (Pseudo Code) การหาค่าเฉลี่ย CBR Packet ต่อเวลาที่โหนดต้นทาง

4) ค่าเฉลี่ย CBR Packet ต่อเวลาที่โหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8 โดยสามารถเขียนเป็นชูโดโค้ด (Pseudo Code) ได้ดังนี้

```

ALGORITHM DESTINATION_AVG_CBR (NODE[],TYPE[],TOTAL_DATA[],TIME,TRACE_LEVEL[],PACKET_SIZE[])
INPUT NODE           #NODE NUMBER
INPUT TYPE           #TYPE OF DATA
INPUT TOTAL_DATA     #NUMBER OF LINE
INPUT TIME           #TOTAL TIME "SEC"
INPUT TRACE_LEVEL    #POINT OF MEASUREMENT "AGT"
INPUT PACKET_SIZE    #PACKET_SIZE "BYTE"
OUTPUT DESTINATION_AVG_CBR # AVERAGE CBR_PACKET AT DESTINATION NODE
DESTINATION_AVG_CBR = 0
FOR I=0, I=TOTAL_DATA, I++
  IF NODE[I] = R8 AND TYPE[I] = CBR AND TRACE_LEVEL = AGT THEN
    DESTINATION_AVG_CBR = DESTINATION_AVG_CBR + PACKET_SIZE[I]
  NEXT
ELSE NEXT
END FOR
DESTINATION_AVG_CBR = DESTINATION_AVG_CBR/TIME
RETURN (DESTINATION_AVG_CBR)

```

ภาพที่ 3.18 ชูโดโค้ด (Pseudo Code) การหาค่าเฉลี่ย CBR Packet ต่อเวลาที่โหนดปลายทาง

5) ค่าสัดส่วนของค่าเฉลี่ย CBR Packet ต่อเวลาระหว่างที่โหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8 เทียบกับ โหนดต้นทาง ในการจำลองเครือข่ายนี้คือ R0

```

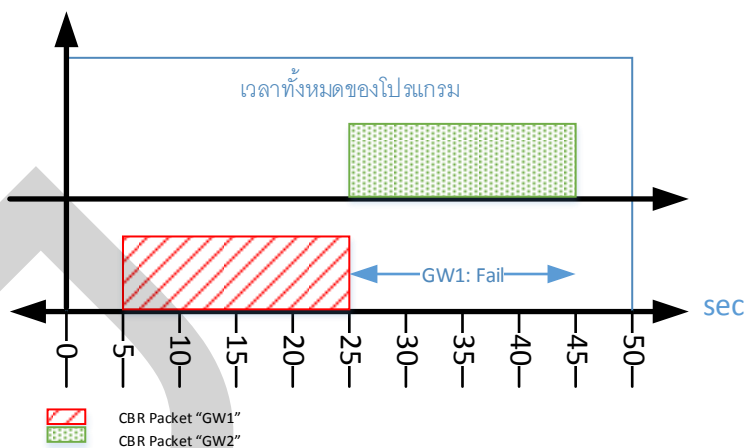
ALGORITHM THROUGHPUT_RATIO (SOURCE_AVG_CBR, DESTINATION_AVG_CBR)
INPUT SOURCE_AVG_CBR           # AVERAGE CBR_PACKET AT SOURCE NODE
INPUT DESTINATION_AVG_CBR      # AVERAGE CBR_PACKET AT DESTINATION NODE
OUTPUT THROUGHPUT_RATIO       # AVERAGE CBR_PACKET AT DESTINATION NODE
THROUGHPUT_RATIO = 0
    THROUGHPUT_RATIO = (DESTINATION_AVG_CBR/SOURCE_AVG_CBR) *100
RETURN (THROUGHPUT_RATIO)

```

ภาพที่ 3.19 ซูโดโค้ด (Pseudo Code) การหาค่าสัดส่วนของค่าเฉลี่ย CBR Packet ต่อเวลาระหว่าง โหนดต้นทาง กับ โหนดปลายทาง

- 6) จำนวน CBR Packet ที่ถูก Drop ที่โหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8
- 7) ค่าเฉลี่ยการหน่วงเวลาแบบ End-to-End delay ของ CBR Packet ระหว่างโหนดต้นทาง ในการจำลองเครือข่ายนี้คือ R0 กับโหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8
- 8) ค่าสัดส่วนของการส่งข้อมูลสำเร็จ PDR% ระหว่างโหนดต้นทาง ในการจำลองเครือข่ายนี้คือ R0 กับโหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8

เมื่อทำการจำลองการทำงานของ WSN แบบ 1 Gateway และได้ค่าตัวแปรที่ต้องการทราบค่าทั้งหมดแล้ว ก็จะทำการจำลองการทำงานของ WSN แบบ 2 Gateway โดยมีการปรับค่าในส่วนของเวลาที่ส่งข้อมูล CBR packet ไปยังโหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8 จากเดิมจะเริ่มส่งที่เวลา 5 วินาที และหยุดส่งที่เวลา 45 วินาที รวมเป็น 40 วินาที ให้มาเป็น ส่งเริ่มที่เวลา 5 วินาที และหยุดส่งที่เวลา 25 วินาที รวมเป็น 20 วินาที จากนั้นให้หยุดการทำงานของโหนดปลายทาง ในการจำลองเครือข่ายนี้คือ R8 และเปลี่ยนโหนดปลายทางใหม่ ในการจำลองเครือข่ายนี้คือ R9 และเริ่มส่งข้อมูล CBR packet ไปยังโหนดปลายทาง R9 ใหม่ที่เวลา 25 วินาที และหยุดส่งที่เวลา 45 วินาที รวมเป็นเวลา 20 วินาที เมื่อนำเวลาที่ส่งข้อมูลไปที่โหนด R8 และ R9 รวมกัน จะได้ 40 วินาที ซึ่งเท่ากับตอนที่ มี โหนดปลายทางเดียว โดยการโหนดปลายทางนี้ เพื่อเป็นการจำลองในการทำงานของเครือข่ายในกรณีที่มี Gateway อยู่ 2 ตัว เมื่อทำงานได้สักระยะแล้ว ตัว Gateway 1 มีปัญหาไม่สามารถรับข้อมูลได้ จึงจำเป็นต้องย้าย Gateway จาก Gateway 1 ไปเป็น Gateway 2 เพื่อส่งข้อมูลออกนอกเครือข่ายแทน รายละเอียดช่วงเวลาในการทำงานสามารถแสดงได้ดังภาพที่ 3.20



ภาพที่ 3.20 ช่วงเวลาในส่ง CBR Packet กรณีที่มี 2 Gateways

```
#Setup CBR1 for 1 Gateways
set cbr1 [new Application/Traffic/CBR]
$ns at $val(cbr_start) "$cbr1 attach-agent $udp_src1"
$cbr1 set packetSize_ 100 # (CBR Packet is 100 byte)
$cbr1 set rate_ 0.1Mb # (Unit bit) (1-100000: No drop) Step 1=0.1Mb, Step 2=0.2Mb
$cbr1 set random_ false

#Setup CBR2 for 2 Gateways
set cbr2 [new Application/Traffic/CBR]
$ns at $val(cbr_start) "$cbr2 attach-agent $udp_src2"
$cbr2 set packetSize_ 100 # (CBR Packet is 100 byte)
$cbr2 set rate_ 0.1Mb # (Unit bit) (1-100000: No drop) Step 1=0.1Mb, Step 2=0.2Mb
$cbr2 set random_ false

#Schedule Time to sent CBR packets for 2 Gateways
$ns at 5.0 "$cbr1 start"
$ns at 15.0 "$cbr1 stop"

$ns at 15.0 "$cbr2 start"
$ns at 25.0 "$cbr2 stop"

$ns at 25.0 "$cbr1 start"
$ns at 35.0 "$cbr1 stop"

$ns at 35.0 "$cbr2 start"
$ns at 45.0 "$cbr2 stop"
```

ภาพที่ 3.21 การปรับปรุงโปรแกรมกรณีที่มี 2 Gateways

ในภาพที่ 3.21 แสดงส่วนโปรแกรมที่ทำการปรับปรุง เพื่อให้สามารถส่งข้อมูล packets CBR ไปยัง Gateways ทั้ง 2 ตัว ตามช่วงเวลาที่กำหนดในภาพที่ 3.20 เพื่อจำลองให้เห็นความแตกต่างของ WSN ในกรณีที่มี 1 Gateway กับ 2 Gateway โดยอาศัยโพรโทคอล AODV

จากตารางที่ 3.7 ค่า Throughput และค่า PDR จะเห็นได้ว่าเมื่อจำนวน CBR packets มีปริมาณมากขึ้นจะทำให้ Throughput และค่า PDR มีค่าที่น้อยลง เนื่องจากตัวโหนดไม่สามารถรองรับปริมาณข้อมูลที่มีจำนวนมากขึ้น ดังจะเห็นได้จากข้อมูลที่รับได้ในโหนด R8 ขณะที่จำนวน CBR packets ตั้งแต่ 7000 ขึ้นไปมีค่าเท่าเดิม และเมื่อพิจารณาในส่วนของจำนวน Packets Drops จะเห็นว่ามีความเพิ่มขึ้นเรื่อยๆ อันเป็นผลมาจากการที่โหนด R8 หยุดทำงาน

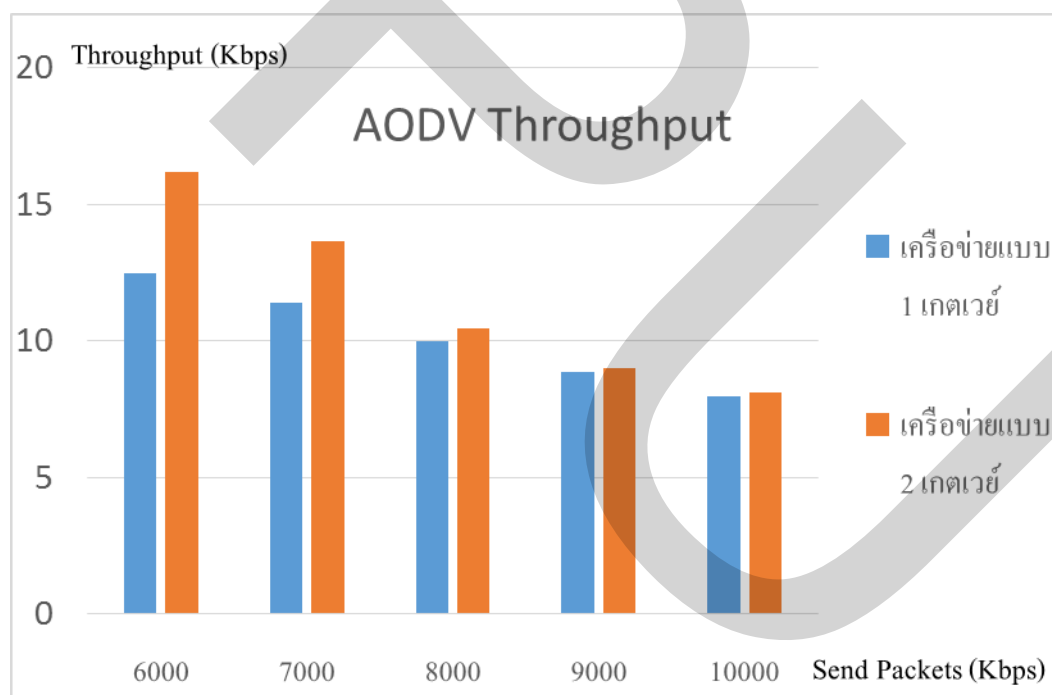
ตารางที่ 3.7 ผลการทดลองเบื้องต้นใน WSN แบบ 1 เกตเวย์

Parameter	Sending Data (Packets)				
	6000	7000	8000	9000	10000
Simulate time (s)	40	40	40	40	40
Data Receive R8 (Packets)	2999	3192	3192	3192	3192
Avg. Throughput R8 (Kbps)	12.50	11.40	9.98	8.87	7.98
Packet Delivery Ratio (R8) (%)	49.98	45.60	39.90	35.47	31.92
Packet Drops (R8) (Packets)	3001	3808	4808	5808	6808

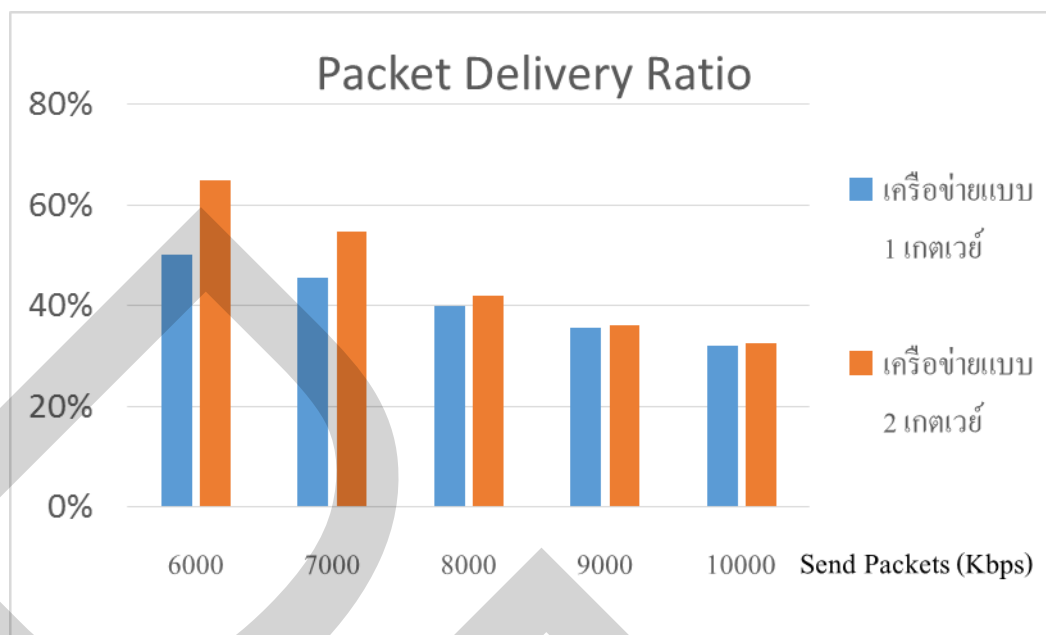
จากตารางที่ 3.8 ผลการทดลองเบื้องต้นใน WSN แบบ 2 เกตเวย์ โดยผลที่ได้เมื่อมีการหาค่าปริมาณ Throughput และค่า PDR จะเห็นได้ว่าที่จำนวนปริมาณ CBR packets ต่างๆ จะได้ปริมาณค่าของ Throughput และค่า PDR ของ WSN แบบ 2 เกตเวย์ มีค่ามากกว่าเมื่อเทียบกับแบบเครือข่ายที่มี 1 เกตเวย์ เนื่องจากตัวโหนด R9 ที่ทำหน้าที่เป็นเกตเวย์ 2 สามารถช่วยรองรับปริมาณข้อมูลที่มีจำนวนมากขึ้นได้

ตารางที่ 3.8 ผลการทดลองเบื้องต้นใน WSN แบบ 2 เกตเวย์

Parameter	Sending Data (Packets)				
	6000	7000	8000	9000	10000
Simulate time (s)	40	40	40	40	40
Data Received R8 (Packets)	2999	3192	3192	3192	3192
Data Received R9 (Packets)	887	634	159	54	54
Avg. Throughput R8+R9 (Kbps)	16.19	13.66	10.47	9.02	8.12
Packet Delivery Ratio R8+R9 (%)	64.77	54.66	41.89	36.07	32.46
Packet Drops R8+R9 (Packets)	2114	3174	4649	5754	6754



ภาพที่ 3.22 ค่าเปรียบเทียบ Throughput ระหว่าง WSN แบบ 1 เกตเวย์ กับ 2 เกตเวย์



ภาพที่ 3.23 ค่า PDR ระหว่าง WSN แบบ 1 เกตเวย์ กับ 2 เกตเวย์

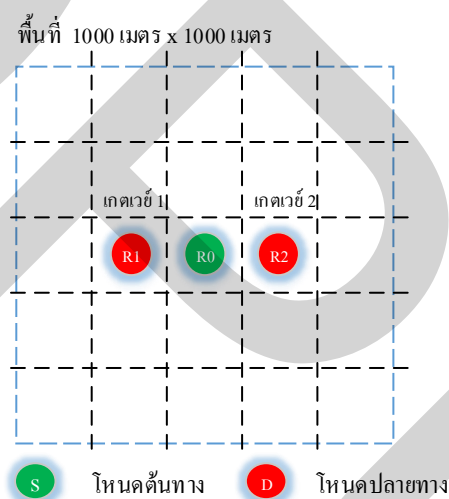
จากกราฟในภาพที่ 3.22 และ 3.23 แสดงการเปรียบเทียบค่า Throughput และค่า PDR ของเครือข่ายแบบ 1 และ 2 เกตเวย์ ซึ่งจะเห็นได้ว่าเครือข่ายแบบ 2 เกตเวย์ สามารถทำได้สูงกว่าแบบเครือข่าย 1 เกตเวย์ ซึ่งปริมาณ Throughput ที่การส่ง Packets 6000 มีค่ามากกว่าที่ส่ง Packets 10000 นั้น เนื่องจากถ้าดูตารางที่ 3.8 พบว่าจำนวน Packets Drop ของเกตเวย์ 2 มีค่ามากขึ้นเมื่อมีการส่งข้อมูลมากขึ้น เนื่องจากเกตเวย์ 2 อยู่ไกลสุด ทำให้ Buffer ที่โหนดระหว่างทางเต็ม มีผลทำให้ Packets ที่ถูกส่งมาใหม่ถูก Drop ไป แต่โดยรวมของการจำลองการส่ง CBR packets ที่ค่าต่างๆ พบว่าในเครือข่าย 2 เกตเวย์ สามารถทำได้สูงกว่าแบบเครือข่าย 1 เกตเวย์ โดยการใช้โพรโทคอลเส้นทาง AODV และโหนดภายในเครือข่ายไม่มีการเคลื่อนที่ แต่ในกรณีมีความหนาแน่นของข้อมูลในเครือข่ายจำนวนมาก ส่วนต่างของค่า Throughput ของเครือข่าย และค่า PDR นั้นจะมีค่าไม่สูงมาก โดยพบค่า Throughput มีค่าสูงขึ้นเฉลี่ย 1.35 Kbps และค่า Packet Delivery Ratio (PDR) ของ WSN แบบ 2 เกตเวย์ มีค่าสูงขึ้นเฉลี่ย 5.39% เมื่อเทียบกับ WSN แบบเดิมที่มี 1 เกตเวย์

จากการทดลองเบื้องต้นจะเห็นได้ว่า เครือข่าย WSN แบบ 2 เกตเวย์ สามารถเพิ่มปริมาณ Throughput ของเครือข่ายได้ เมื่อเทียบกับ WSN แบบ 1 เกตเวย์ หลังจากการทดลองเบื้องต้นแล้วได้มีการกำหนดเงื่อนไขในการทดลองเพิ่มเติมและมีการปรับปรุงตัวโพรโทคอล AODV เพิ่มเติมเพื่อให้สามารถทำงานบน WSN แบบเกตเวย์หลายตัวได้

หลังจากได้ผลการทดลองเบื้องต้นแล้ว ได้ทำการปรับปรุงการทดลองเครือข่ายเซนเซอร์ไร้สาย Wireless Sensor Network (WSN) แบบเกตเวย์หลายตัว เพื่อให้ครอบคลุมรูปแบบการใช้งานจริงของ WSN และลดปัญหาการเกิดการ drop ของข้อมูลระหว่างทาง โดยมุ่งเน้นการทดลองเพื่อหาค่าพลังงานที่ใช้ในเกตเวย์สำรอง และเวลาที่ใช้ในการเปลี่ยนเกตเวย์ รวมถึงแสดงให้เห็นถึงประสิทธิภาพจากการเปรียบเทียบการทำงานของ WSN แบบ 2 เกตเวย์ กับ แบบ 1 เกตเวย์

3.3.7 กำหนดรูปแบบเครือข่าย WSN ใหม่ที่ใช้ในการทดลองโพรโตคอล AODV

เครือข่าย WSN ที่ใช้ในการทดลองนี้ประกอบไปด้วยโหนดเกตเวย์ 2 โหนด ซึ่งมีโหนด R0 ทำหน้าที่เป็นโหนดต้นทาง โหนด R1 ทำหน้าที่เป็นเกตเวย์หลัก โหนด R2 ทำหน้าที่เป็นเกตเวย์สำรอง รายละเอียดแสดงให้ห็นดังรูปที่ 3.24



ภาพที่ 3.24 เครือข่ายเซนเซอร์ไร้สาย แบบ 2 เกตเวย์ ที่ใช้ในการทดลอง

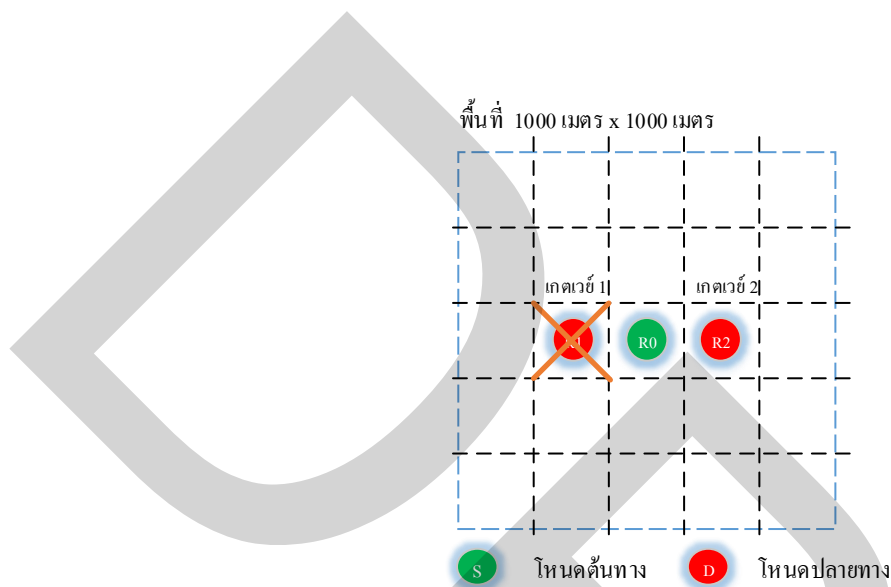
ในการทดลองมีการส่งข้อมูลจากโหนดต้นทางไปยังเกตเวย์ โดยใช้ชนิดของข้อมูลแบบ CBR ขนาด 100 Byte และมีการปรับเปลี่ยนปริมาณของข้อมูลที่ส่งเข้าไป ในช่วงเวลาที่กำหนด โดยมีรายละเอียดของลำดับข้อมูลที่ใช้ในการทดลองตามตารางที่ 3.9

ตารางที่ 3.9 จำนวน packets ที่ใช้ในการทดลอง

Time (sec)	40	40	40	40	40	40
Packets	3000	4000	5000	6000	7000	8000

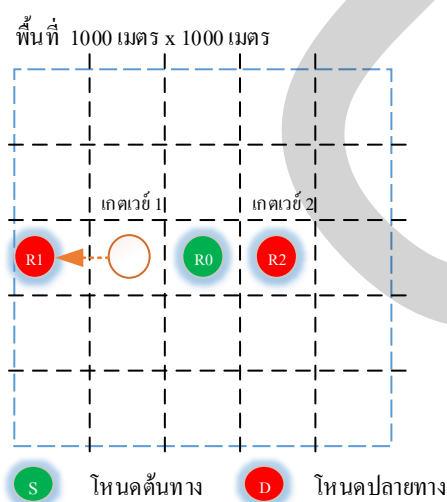
เพื่อให้การทดลองครอบคลุมรูปแบบการใช้งานจริงของเครือข่าย WSN ที่จะเกิดปัญหาที่ตัวเกตเวย์ จึงมีการกำหนดรูปแบบของการเกิดปัญหาที่ตัวเกตเวย์เดิมไว้ 2 แบบ ดังนี้

1. เมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากได้รับความเสียหาย แบบทันทีทันใด



ภาพที่ 3.25 เครือข่ายที่ใช้ทดลองแบบเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากได้รับความเสียหาย

2. เมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากมีการเคลื่อนที่ออกจากเครือข่าย



ภาพที่ 3.26 เครือข่ายที่ใช้ทดลองแบบเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากเคลื่อนที่ออกจากเครือข่าย

ในการทดลองจะมีการกำหนดเวลาที่เกิดการเสียหายที่ตัวเกตเวย์ _1_ ไว้ที่ วินาทีที่ 25 และมีการเคลื่อนที่ของเกตเวย์ _1_ ออกจากเครือข่าย จากตำแหน่ง Y: 250m, X: 250m มาเป็น Y: 0.1m, X: 250m ด้วยความเร็ว 100m/s หลังการกำหนดเวลาในการทดสอบแล้ว จะทำการทดสอบ โดยเริ่มส่ง packets ที่เวลา 5 วินาที และหยุดที่เวลา 45 วินาที

ในการทดลองนี้ได้มีการปรับปรุงตัวโปรโตคอล AODV ให้มีการแสดงเวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย เพื่อจะใช้เป็นเวลาที่เปลี่ยนตัวเกตเวย์ จากตัวที่ 1 เป็นตัวที่ 2 โดยมีส่วนที่ปรับเปลี่ยนใน AODV.CC เพื่อตรวจสอบ Message RERR เมื่อเกตเวย์ปลายทาง ที่โหนดต้นทางส่งข้อมูลไป ไม่สามารถรับข้อมูล และทำการเพิ่ม Time Stamp และ Error Status เพื่อให้ตัวโหนดต้นทางรับรู้ เพื่อจะได้ทำการเปลี่ยนเกตเวย์ได้ โดยมีปรับเปลี่ยนตามภาพที่ 3.27

```

1238 void
1239 AODV::sendError(Packet *p, bool jitter) {
1240     gw_ = gw_ + 1; //cnd
1241     gw_chgtime_ = CURRENT_TIME; //cnd
1242
1243     printf("Send error counting: %d \n",gw_);
1244     printf("Current time to send error: %f \n",CURRENT_TIME);

```

ภาพที่ 3.27 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC

ในส่วนของการทำงานของ NS2 จะมีการทำงานของภาษา C++ ภาษา OCTL และภาษา TCL ที่เป็นเช่นนี้เนื่องจาก ตัวโครงสร้างของภาษา C++ จะทำงานได้เร็ว จึงถูกใช้ในส่วนของการทำงานของโปรโตคอล ที่มีเรื่องของเวลาเข้ามาเกี่ยวข้อง แต่ในส่วนของการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการควบคุมการทำงานของตัวโหนดนั้น จะใช้ส่วนของตัว OTCL และ TCL ในการส่งข้อมูลให้เนื่องจากง่ายต่อการใช้งาน ดังนั้นในส่วนนี้จึงจำเป็นต้องสร้างตัวแปรที่ใช้ในการเชื่อมโยง แลกเปลี่ยนข้อมูล ระหว่าง ภาษา C++ ภาษา OCTL และTCL ซึ่งในที่นี้จะเป็นการสร้างตัวแปร “gw_chgtime”_ และ “gw_” ซึ่งเป็นตัวแปรประเภททศนิยม กับตัวเลข ตามลำดับ เพื่อส่งข้อมูลให้กับตัวโหนดต้นทาง ในการตัดสินใจในการเปลี่ยนเกตเวย์ ซึ่งส่วนที่เป็นการปรับปรุงใน AODV.CC เพื่อเชื่อมโยงข้อมูลมีรายละเอียดตาม ภาพที่ 3.28

```

142     AODV::AODV(nsaddr_t id) : Agent(PT_AODV),
143                               btimer(this), htimer(this), ntimer(this),
144                               rtimer(this), lrtimer(this), rqueue() {
145
146         bind("gw_", &gw_); //cnd
147         bind("gw_chgtime_", &gw_chgtime_); //cnd
148     }

```

ภาพที่ 3.28 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC เพื่อเชื่อมโยงข้อมูล

ในการทดลองนี้ได้มีการปรับปรุงตัวโพรโทคอล AODV ให้มีการแสดงเวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย เพื่อจะใช้เป็นเวลาที่เปลี่ยนตัวเกตเวย์ จากตัวที่ 1 เป็นตัวที่ 2 โดยมีส่วนที่ปรับเปลี่ยนใน AODV.CC เพื่อตรวจสอบ Message RERR เมื่อเกตเวย์ปลายทาง ที่โหนดต้นทางส่งข้อมูลไป ไม่สามารถรับข้อมูล และทำการเพิ่ม Time Stamp และ Error Status เพื่อให้ตัวโหนดต้นทางรับรู้ เพื่อจะได้ทำการเปลี่ยนเกตเวย์ได้ โดยมีปรับเปลี่ยนตามภาพที่ 3.29

```

1238     void
1239     AODV::sendError(Packet *p, bool jitter) {
1240         gw_ = gw_ + 1; //cnd
1241         gw_chgtime_ = CURRENT_TIME; //cnd
1242
1243         printf("Send error counting: %d \n", gw_);
1244         printf("Current time to send error: %f \n", CURRENT_TIME);

```

ภาพที่ 3.29 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC

ในส่วนของการทำงานของ NS2 จะมีการทำงานของภาษา C++ ภาษา OCTL และภาษา TCL ที่เป็นเช่นนี้เนื่องจาก ตัวโครงสร้างของภาษา C++ จะทำงานได้เร็ว จึงถูกใช้ในส่วนของการทำงานของโพรโทคอล ที่มีเรื่องของเวลาเข้ามาเกี่ยวข้อง แต่ในส่วนของการปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ในการควบคุมการทำงานของตัวโหนดนั้น จะใช้ส่วนของตัว OTCL และ TCL ในการส่งข้อมูลให้เนื่องจากง่ายต่อการใช้งาน ดังนั้นในส่วนนี้จึงจำเป็นต้องสร้างตัวแปรที่ใช้ในการเชื่อมโยง แลกเปลี่ยนข้อมูล ระหว่าง ภาษา C++ ภาษา OCTL และTCL ซึ่งในที่นี้จะเป็นการสร้างตัวแปร “gw_chgtime”_ และ “gw_” ซึ่งเป็นตัวแปรประเภทศนิยม กับตัวเลข ตามลำดับ เพื่อส่งข้อมูล

ให้กับตัวโหนดต้นทาง ในการตัดสินใจในการเปลี่ยนเกตเวย์ ซึ่งส่วนที่เป็นการปรับปรุงใน AODV.CC เพื่อเชื่อมโยงข้อมูลมีรายละเอียดตาม ภาพที่ 3.30

```

142 AODV::AODV(nsaddr_t id) : Agent(PT_AODV),
143     btimer(this), htimer(this), ntimer(this),
144     rtimer(this), lrtimer(this), rqueue() {
145
146     bind("gw_", &gw_); //cnd
147     bind("gw_chgtime_", &gw_chgtime_); //cnd
148

```

ภาพที่ 3.30 ส่วนที่ปรับปรุงเพิ่มเติมใน AODV.CC เพื่อเชื่อมโยงข้อมูล

จากภาพที่ 3.29 การเพิ่มและเชื่อมโยงตัวแปร “gw_chgtime_” และ “gw_” เมื่อโปรแกรมเริ่มทำงาน ในกรณีที่โหนดมีการส่ง Message RERR ก็จะเข้ามาทำงานในส่วนของโปรแกรมตามภาพที่ 3.25 ซึ่งจะทำให้การเพิ่มค่า “gw_” ขึ้น และทำการเก็บค่าเวลานั้น เพื่อใช้เป็นเวลาที่จะเริ่มทำการเปลี่ยนเกตเวย์ในส่วนของ TCL ซึ่งผลจากการทดลอง ได้มีการเก็บค่าเวลาที่จะทำการเปลี่ยนเกตเวย์ที่ปริมาณข้อมูล ตามตารางที่ 3.9

ในส่วนของการทำงานของตัวโปรแกรมที่ใช้ภาษา TCL เพื่อกำหนดการทำงานและเงื่อนไขต่างๆ บนเครือข่ายนั้น จะมีการกำหนดเงื่อนไขในการเปลี่ยนเกตเวย์ในเครือข่าย โดยการพัฒนาโปรแกรมตามรายละเอียดในภาพที่ 3.31 และ 3.32

```

98 #Setup CBR1
99 set cbr1 [new Application/Traffic/CBR]
100 $ns at 5.0 "$cbr1 attach-agent $udp_src1"
101 $cbr1 set packetSize_ 100 #(CBR Packet is 100 byte)
102 $cbr1 set rate_ 0.06Mb # (Unit bit)
103 $cbr1 set random_ false
104 $ns at 5.0 "$cbr1 start"
105 $ns at 45.0 "$cbr1 stop"
106
107 #Schedule Time to take action
108 $ns at $val(gw1_action_) "$n(1) setdest 0.1 250.0 100"
109 $ns at $val(gw1_action_) "$n(1) node-down"

```

ภาพที่ 3.31 ส่วนของโปรแกรมที่พัฒนาโดยภาษา TCL เพื่อสร้าง packets CBR

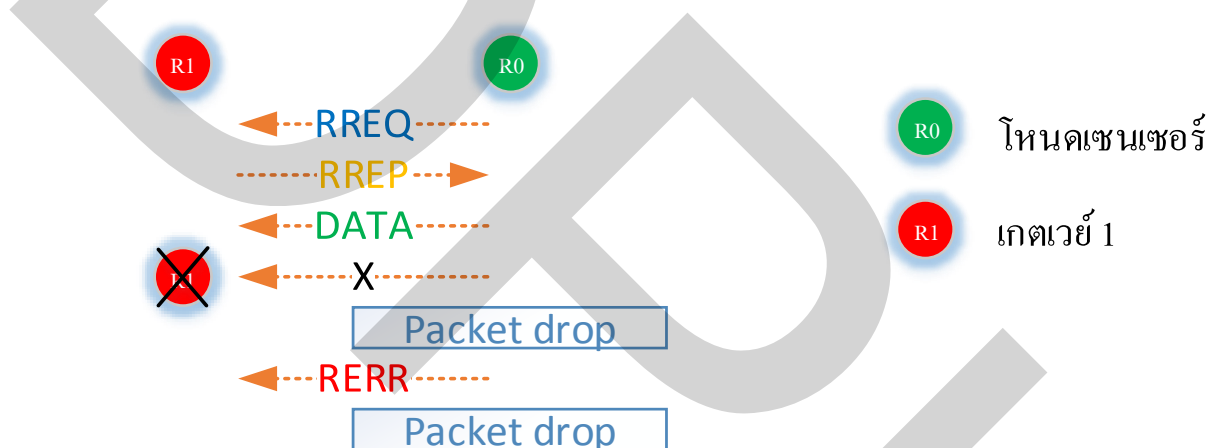
```

121 for {set i 1} {$i < $val(stop)} {incr i} {
122 if { $val(gw_) == 1 } {
123     $ns at $val(gw_chgtime_) "$ns attach-agent $n(2) $udp_des2"
124     $ns at $val(gw_chgtime_) "$ns connect $udp_srcl $udp_des2"
125 }
126 }

```

ภาพที่ 3.32 ส่วนของโปรแกรมที่พัฒนาโดยภาษา TCL เพื่อกำหนดเงื่อนไขในการเปลี่ยนเกตเวย์

ในส่วนของการเปลี่ยนเกตเวย์จะอาศัยเงื่อนไขเมื่อเกตเวย์หลักมีการส่ง Message RERR ออกไป เมื่อเกตเวย์หลักไม่สามารถทำงานได้ โดยแสดงในภาพที่ 3.33

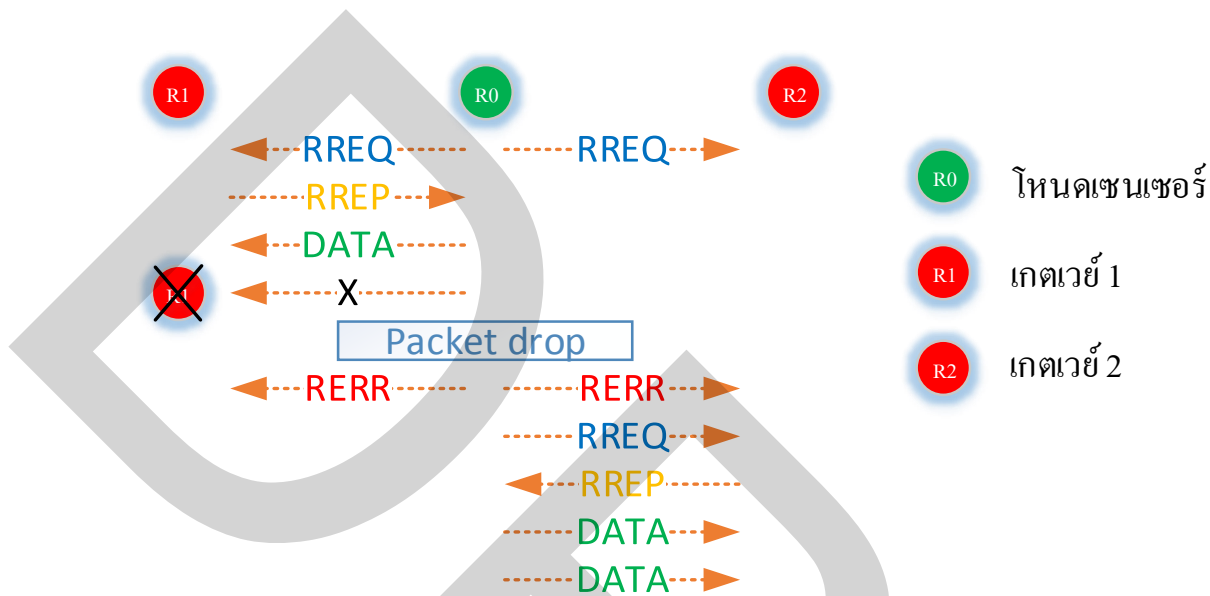


ภาพที่ 3.33 การส่ง Message RERR ออกไป เมื่อเกตเวย์หลักไม่สามารถทำงานได้

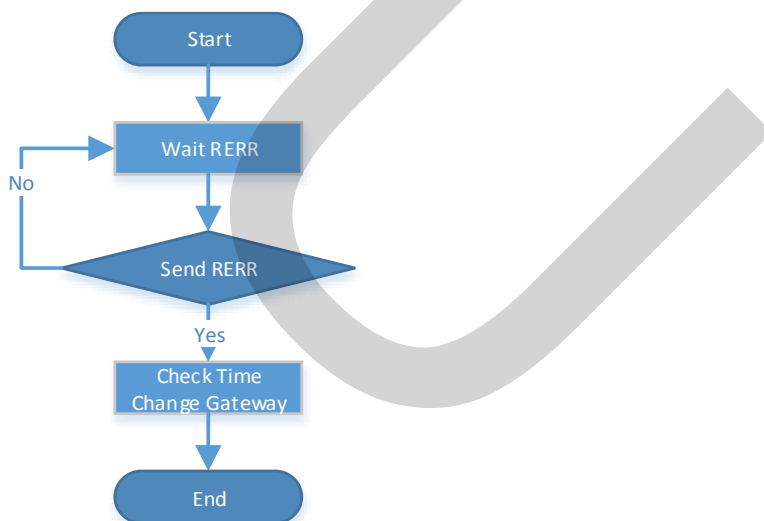
ในภาพที่ 3.33 เป็นการทำงานของโพรโทคอล AODV แบบเดิม ซึ่งเมื่อโหนดต้นทางต้องการส่งข้อมูลไปให้เกตเวย์หลัก ก็จะมีการส่ง Message RREQ ออกไปเพื่อร้องขอเส้นทาง เมื่อเกตเวย์หลักได้รับ Message RREQ แล้วก็จะมีการตอบกลับ Message RREP ไปยังเส้นทางที่ดีที่สุด เมื่อโหนดต้นทางได้รับ Message RREP ก็จะเริ่มกระบวนการส่งข้อมูล เมื่อเกตเวย์หลักไม่สามารถทำงานได้หรือเส้นทางไปยังเกตเวย์หลักไม่สามารถใช้งานได้ ตัวโหนดต้นทางก็จะส่ง Message RERR ออกไปยังโหนดข้างเคียงเพื่อร้องขอเส้นทางใหม่ที่ไปยังเกตเวย์หลักอีกครั้ง ซึ่งในกรณีนี้จะไม่สามารถร้องขอเส้นทางใหม่ได้ เนื่องจากเกตเวย์หลักเสียหาย ดังนั้นข้อมูลที่เหลือก็จะถูก drop ไป

ในภาพที่ 3.34 เป็นการทำงานของโพรโทคอล AODV แบบใหม่ที่ใช้กับ WSN แบบเกตเวย์หลายตัว ซึ่งจะอาศัยจังหวะที่โหนดต้นทางส่ง Message RERR ออกไป โดยโหนดต้นทางจะมี

เกตเวย์สำรองอยู่ในรายการแล้ว ก็จะเปลี่ยนไปใช้เกตเวย์สำรองทันที โดยเริ่มกระบวนการ RREQ ใหม่ แต่ใช้โหนดปลายทางเป็นเกตเวย์สำรองแทน โดยภาพที่ 3.35 เป็นลำดับขั้นตอนการทำงานของโปรโตคอล AODV แบบใหม่ที่ใช้กับ WSN แบบเกตเวย์หลายตัว



ภาพที่ 3.34 การส่ง Message RERR ออกไป เมื่อเกตเวย์หลักไม่สามารถทำงานได้



ภาพที่ 3.35 ขั้นตอนการรอ Message RERR เพื่อเปลี่ยนเกตเวย์

บทที่ 4

การทดลองและผลการทดลอง

ในงานวิจัยนี้ได้ทำการทดลองเครือข่ายเซนเซอร์ไร้สาย Wireless Sensor Network (WSN) แบบ 2 เกตเวย์ โดยมุ่งเน้นการทดลองเพื่อแสดงให้เห็นถึงประสิทธิภาพจากการเปรียบเทียบการทำงานของ WSN แบบ 2 เกตเวย์ คือเกตเวย์หลัก และเกตเวย์สำรอง กับ WSN แบบ 1 เกตเวย์ ที่มีเพียงเกตเวย์หลักเพียงอย่างเดียว

4.1 หาค่าพลังงานของเกตเวย์สำรองที่ใช้ไปและเวลาที่ใช้ในการเปลี่ยนเกตเวย์

ในการทดลองมีการส่งข้อมูลจากโหนดต้นทางไปยังเกตเวย์ โดยใช้ชนิดของข้อมูลแบบ CBR ขนาด 100 Byte และมีการปรับเปลี่ยนปริมาณของข้อมูลที่ส่งเข้าไป ในช่วงเวลาที่กำหนด โดยมีรายละเอียดของลำดับข้อมูลที่ใช้ในการทดลองตามตารางที่ 4.1

ตารางที่ 4.1 จำนวน packets ที่ใช้ในการทดลอง

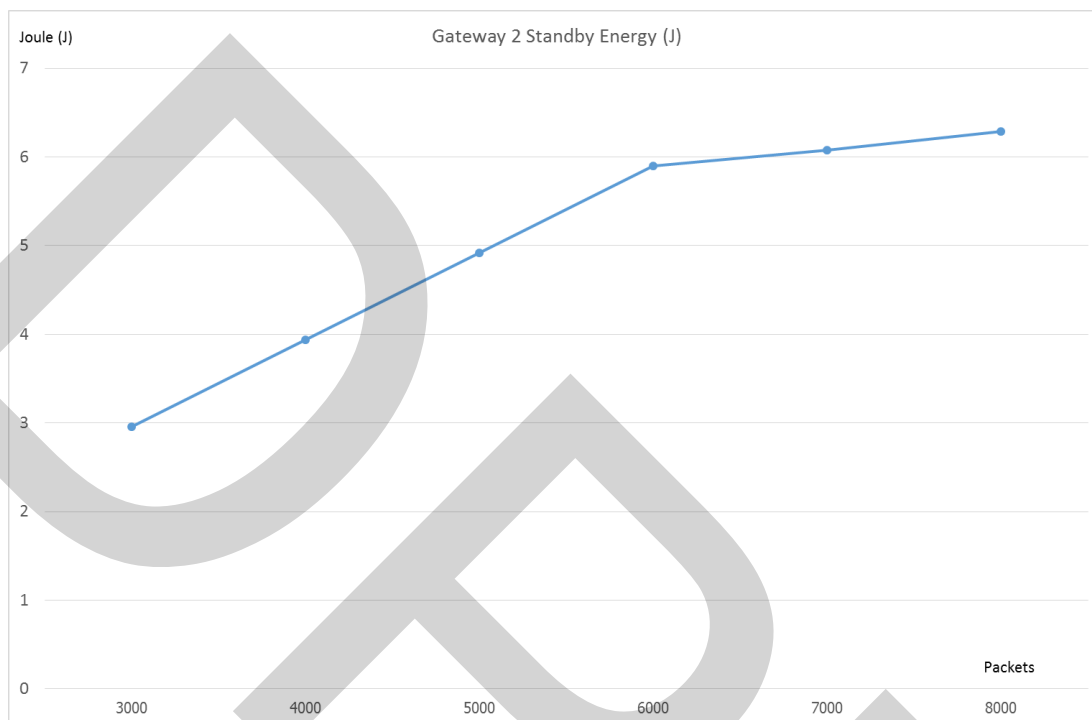
Time (sec)	40	40	40	40	40	40
Packets	3000	4000	5000	6000	7000	8000

ในการทดลองจะมีการกำหนดเวลาที่เกตเวย์หลัก จะเกิดปัญหาจนไม่สามารถทำงานได้ไว้ที่ วินาทีที่ 25 โดยมีลักษณะของปัญหาอยู่ 2 รูปแบบ คือ เกตเวย์หลักเสียหายแบบทันทีทันใด และเกตเวย์หลักมีการเคลื่อนที่ออกนอกเครือข่ายจากตำแหน่ง Y: 250m, X: 250m มาเป็น Y: 0.1m, X: 250m ด้วยความเร็ว 100m/s หลังการกำหนดเวลาในการทดสอบแล้ว จะทำการทดสอบ โดยเริ่มส่ง packets ที่เวลา 5 วินาที และหยุดที่เวลา 45 วินาที

4.1.1 หาค่าพลังงานของเกตเวย์สำรอง

ในการทดลองบนเครือข่าย WSN แบบเกตเวย์หลายตัว ได้มีการหาค่าพลังงานที่เกตเวย์สำรองใช้ในการรอเพื่อที่จะทำงานเมื่อเกตเวย์หลักไม่สามารถรับส่งข้อมูลได้ ซึ่งแสดงให้เห็นรูปที่ 4.1 โดยเมื่อดูจากกราฟแล้วจะเห็นได้ว่าตัวเกตเวย์สำรองมีการใช้พลังงานไปในขณะรอ แสดงมีการ

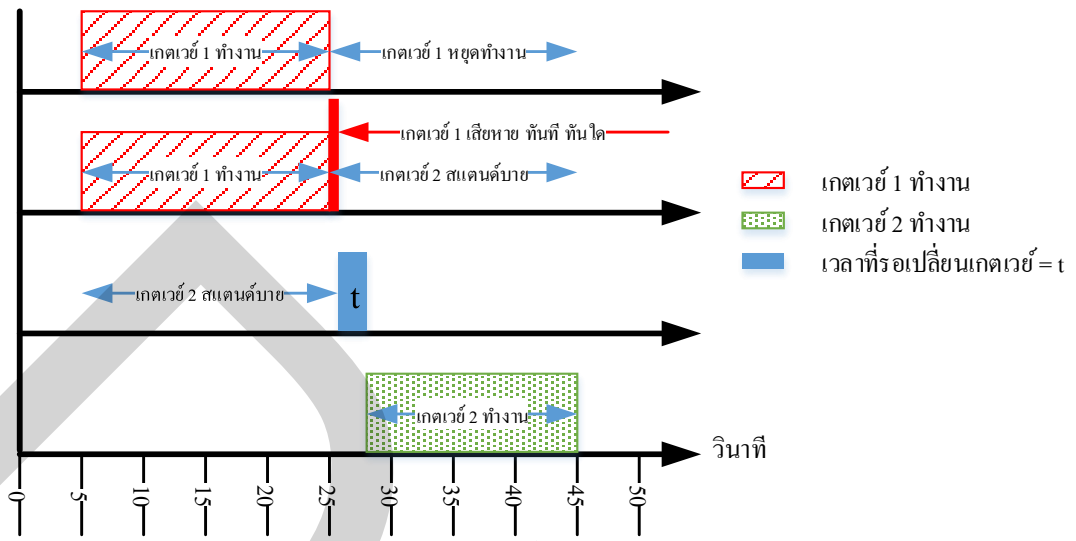
ใช้พลังงานมากขึ้นเรื่อยๆ เมื่อความหนาแน่นของข้อมูลมากขึ้น ทั้งนี้เนื่องมาจากว่าตัวเกตเวย์สำรองก็ จะทำหน้าที่ในการส่งต่อข้อมูลไปยังโหนดอื่นๆ ข้างเคียงด้วยเช่นกัน



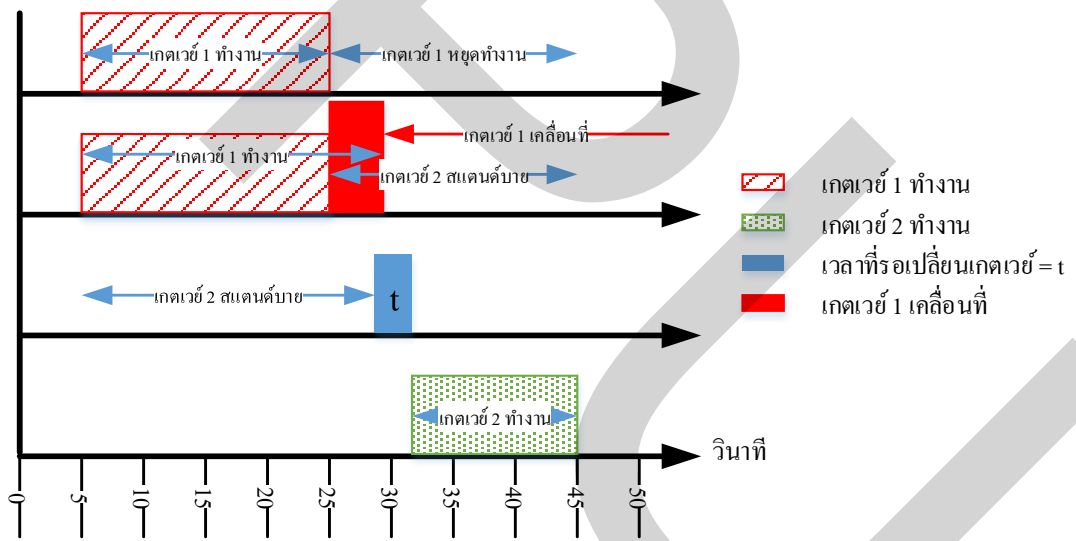
ภาพที่ 4.1 ค่าพลังงานของเกตเวย์สำรองที่ใช้ในขณะรอทำงาน

4.1.2 เวลาที่ใช้ในการเปลี่ยนเกตเวย์

ผลจากการทดลอง ได้มีการเก็บค่าเวลาที่จะทำเริ่มทำการเปลี่ยนเกตเวย์ ที่ปริมาณข้อมูล ตามตารางที่ 4.1 โดยจะมีค่าเวลาที่ใช้ในการเปลี่ยนเกตเวย์อยู่ 2 กรณีคือ ในกรณีเกตเวย์หลักเสียหาย แบบทันทีทันใด และในกรณีเกตเวย์หลักมีการเคลื่อนที่ออกนอกเครือข่าย ซึ่งแสดงช่วงเวลาในการ ทดลองให้เห็นในภาพที่ 4.2 และ 4.3 โดยภาพที่ 4.4 ถึงภาพที่ 4.9 เป็นผลการทดลองของเครือข่าย 2 เกตเวย์ ในกรณีที่ 1 คือเมื่อเกตเวย์หลัก ไม่สามารถติดต่อได้ เนื่องจากรับความเสียหาย แบบ ทันทีทันใด ส่วนภาพที่ 4.10 ถึงภาพที่ 4.15 เป็นผลการทดลองของเครือข่าย 2 เกตเวย์ ในกรณีที่ 2 คือเมื่อเกตเวย์หลัก ไม่สามารถติดต่อได้ เนื่องจากมีการเคลื่อนที่ออกจากเครือข่าย



ภาพที่ 4.2 เวลาที่ใช้ในทดสอบ WSN กรณีเกตเวย์หลักหยุดทำงานแบบทันที ทันใด



ภาพที่ 4.3 เวลาที่ใช้ในทดสอบ WSN กรณีเกตเวย์หลักเคลื่อนที่ออกนอกเครือข่าย

```

ns2@ubuntu:~/cnd$ ns 2gw_move.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 60000
CBR1 interval = 0.013333333333333334
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.026265
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.4 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 3000

```

ns2@ubuntu:~/cnd$ ns aodv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 80000
CBR1 interval = 0.01
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.026265
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.5 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 4000

```

ns2@ubuntu:~/cnd$ ns aodv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 100000
CBR1 interval = 0.008000000000000000002
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.027865
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.6 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 5000

```

ns2@ubuntu:~/cnd$ ns aodv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 120000
CBR1 interval = 0.0066666666666666671
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.026055
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.7 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 6000

```

ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 140000
CBR1 interval = 0.0057142857142857143
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.027335
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.8 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 7000

```

ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 160000
CBR1 interval = 0.0050000000000000001
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 25.027335
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.9 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเสียหาย จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 8000

จากภาพที่ 4.6 ถึงภาพที่ 4.11 เป็นผลการทดลองของเครือข่าย 2 เกตเวย์ ในกรณีที่ 1 คือเมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้ ซึ่งจากผลการทดลองจะเห็นได้ว่า เมื่อมีการปิดการทำงานของตัวเกตเวย์ 1 แล้วนั้น จะทำให้ตัวโพรโทคอล AODV ที่ปรับปรุงแล้ว มีการแสดงจำนวนครั้งที่มีการส่ง

ข้อมูล Error และแสดงค่าเวลาที่มีการส่ง message ERROR โดยผลการทดลองที่ได้แสดงในตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองเมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้

Packets	3000	4000	5000	6000	7000	8000
Even Time (sec)	25.000000	25.000000	25.000000	25.000000	25.000000	25.000000
Error Time (sec)	25.026265	25.026265	25.027865	25.026055	25.027335	25.027335
Response Time (sec)	0.02626	0.02626	0.02786	0.02605	0.02734	0.02734

จากตารางที่ 4.2 จะเห็นได้ว่าค่าเวลาที่มีการตอบสนอง (Response time) หลังจากมีการปิดการทำงานของตัวเกตเวย์ 1 แล้วนั้น ที่ค่า packets ต่างๆ มีค่าไม่ต่างกันเฉลี่ยอยู่ประมาณ 0.02685 วินาที และมีค่าต่ำสุดอยู่ที่ 0.02605 วินาที และมีค่าสูงสุดอยู่ที่ 0.02786 วินาที

```
ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 60000
CBR1 interval = 0.013333333333333334
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.695492
ns2@ubuntu:~/cnd$
```

ภาพที่ 4.10 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 3000

```
ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 80000
CBR1 interval = 0.01
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.693705
ns2@ubuntu:~/cnd$
```

ภาพที่ 4.11 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 4000

```
ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 100000
CBR1 interval = 0.0080000000000000000002
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.690585
ns2@ubuntu:~/cnd$
```

ภาพที่ 4.12 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 5000

```

ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 120000
CBR1 interval = 0.0066666666666666671
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.689049
ns2@ubuntu:~/cnd$

```

รูปที่ 4.13 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 6000

```

ns2@ubuntu:~/cnd$ ns aadv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 140000
CBR1 interval = 0.0057142857142857143
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.688724
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.14 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 7000

```

ns2@ubuntu:~/cnd$ ns aodv_wsn2gw.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
id: 0
INITIALIZE THE LIST xListHead
id: 1
id: 2
CBR1 packet size = 100
CBR1 packet rate = 160000
CBR1 interval = 0.0050000000000000001
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
Send error counting: 1
Current time to send error: 26.688724
ns2@ubuntu:~/cnd$

```

ภาพที่ 4.15 เวลาที่โหนดต้นทางได้รับ Message RERR เมื่อโหนดปลายทางเคลื่อนที่ออก จำนวน Packets ที่ใช้ทดสอบ เท่ากับ 8000

จากภาพที่ 4.10 ถึงภาพที่ 4.15 เป็นผลการทดลองของเครือข่าย 2 เกตเวย์ ในกรณีที่ 1 คือเมื่อเกตเวย์ 1 ไม่สามารถติดต่อได้ เนื่องจากมีการเคลื่อนที่ออกนอกเครือข่าย ซึ่งจากผลการทดลองจะเห็นได้ว่า เมื่อมีการปิดการทำงานของตัวเกตเวย์ 1 แล้วนั้น จะทำให้ตัวโพรโทคอล AODV ที่ปรับปรุงแล้ว มีการแสดงจำนวนครั้งที่มีการส่งข้อมูล Error และแสดงค่าเวลาที่มีการส่ง message ERROR โดยผลการทดลองที่ได้แสดงในตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองเมื่อเกตเวย์ 1 เคลื่อนที่ออกจากเครือข่าย

Packets	3000	4000	5000	6000	7000	8000
Even Time (sec)	25.00000	25.00000	25.00000	25.00000	25.00000	25.00000
Error Time (sec)	26.69549	26.69371	26.69059	26.68905	26.68872	26.68872
Response Time (sec)	1.69549	1.69371	1.69059	1.68905	1.68872	1.68872

จากตารางที่ 4.3 จะเห็นได้ว่าค่าเวลาที่มีการตอบสนอง (Response time) หลังจากในตัวเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่ายที่ค่า packets ต่างๆ ค่าเวลาตอบสนองที่ส่งออกมา มีค่าไม่

ต่างกันเฉลี่ยอยู่ประมาณ 1.69105 วินาที และมีค่าต่ำสุดอยู่ที่ 1.68872 วินาที และมีค่าสูงสุดอยู่ที่ 1.69549 วินาที

ตารางที่ 4.4 เปรียบเทียบค่าเวลาการตอบสนอง Error ของเครือข่าย WSN แบบ 2 เกตเวย์

Packets	3000	4000	5000	6000	7000	8000
Even Time (sec)	25.000	25.000	25.000	25.000	25.000	25.000
Response Time (sec)						
Turn-Down	0.02626	0.02626	0.02786	0.02605	0.02734	0.02734
Response Time (sec)						
Move	1.69549	1.69371	1.69059	1.68905	1.68872	1.68872

จากตารางที่ 4.4 ได้ว่าค่าเวลาการตอบสนอง Error หลังจากมีการทดลองการทำงานทั้ง 2 รูปแบบคือ แบบเกตเวย์หยุดทำงาน กับแบบเกตเวย์เคลื่อนที่นั้น พบว่าแบบเกตเวย์เคลื่อนที่ที่มีการตอบสนอง Error ช้ากว่าแบบเกตเวย์หยุดทำงาน ที่เป็นเช่นนี้เพราะว่า ตัวเกตเวย์ที่กำลังเคลื่อนที่ออกจากเครือข่ายนั้น ยังคงสามารถติดต่อกับตัวโหนดต้นทางได้อยู่ แม้กระทั่งในขณะที่ตัวเองเคลื่อนที่จะกระทั้งตัวเองหลุดออกจากเครือข่าย จึงมีการตอบสนองต่อ Error ดังนั้นจึงทำให้เกิดการตอบสนองช้ากว่า แต่อย่างไรก็ตามการทดลองทั้งสองแบบก็สามารถ ส่งค่าเวลาและสถานะ เพื่อทำการเปลี่ยนเกตเวย์ได้

4.2 หาประสิทธิภาพของเครือข่ายเซนเซอร์ไร้สายแบบ 2 เกตเวย์ เทียบกับ แบบ 1 เกตเวย์

หลังจากทำการทดลองเพื่อเก็บข้อมูลในรูปแบบของ trace file เรียบร้อยแล้ว ก็เอาข้อมูลที่ได้มาทำการวิเคราะห์ โดยการหาค่าที่ต้องการดังนี้

- 1 หาค่าการหน่วงเวลาจากโหนดต้นทาง ไปยังโหนดปลายทาง (packets delay)
- 2 หาค่าความสำเร็จในการส่งข้อมูลจากโหนดต้นทาง ไปยังโหนดปลายทาง (%PDR)
- 3 หาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (%packets drop)
- 4 หาค่าปริมาณข้อมูลที่เกตเวย์หลักและเกตเวย์สำรองรับได้เทียบกับเวลา (Throughput

kbps)

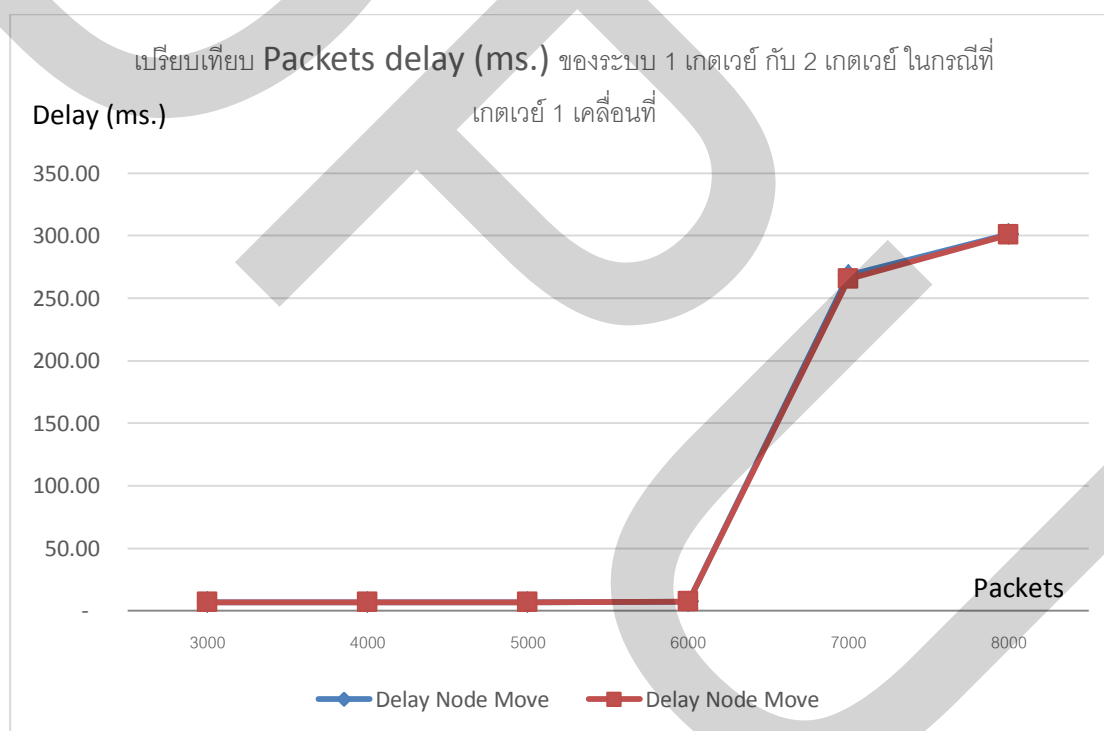
ในการหาประสิทธิภาพของเครือข่ายเซนเซอร์ไร้สายแบบ 2 เกตเวย์ เทียบกับ แบบ 1 เกตเวย์นั้น ในแต่ละแบบจะประกอบไปด้วยการทดลองแบบปิดการทำงานของเกตเวย์เดิม กับการย้ายตำแหน่งของเกตเวย์เดิมออกนอกเครือข่าย

4.2.1 หาค่าการหน่วงเวลาจากโหนดต้นทางไปยังโหนดปลายทาง (packets delay)

ในส่วนนี้จะเป็นการหาค่าการหน่วงเวลาจากโหนดต้นทาง ไปยังโหนดปลายทาง (packets delay) ซึ่งจะทำให้การหาค่าเวลาสูงสุด และต่ำสุดของแต่ละ packet จากนั้นจึงนำค่าที่ได้มาหาค่าส่วนต่าง แล้วนำมารวมกัน จากนั้นหารด้วยจำนวน packets ทั้งหมดเพื่อหาค่าเฉลี่ย ซึ่งได้ผลดังนี้

ตารางที่ 4.5 ค่าการหน่วงเวลา packets delay ของแบบเกตเวย์ 1 เคลื่อนที่ที่ออกนอกเครือข่าย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	6.980	6.977	6.977	7.335	268.475	301.322
2 Gateway	6.954	6.963	6.969	7.379	265.403	300.660



ภาพที่ 4.16 เปรียบเทียบ Packets delay (ms) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เคลื่อนที่

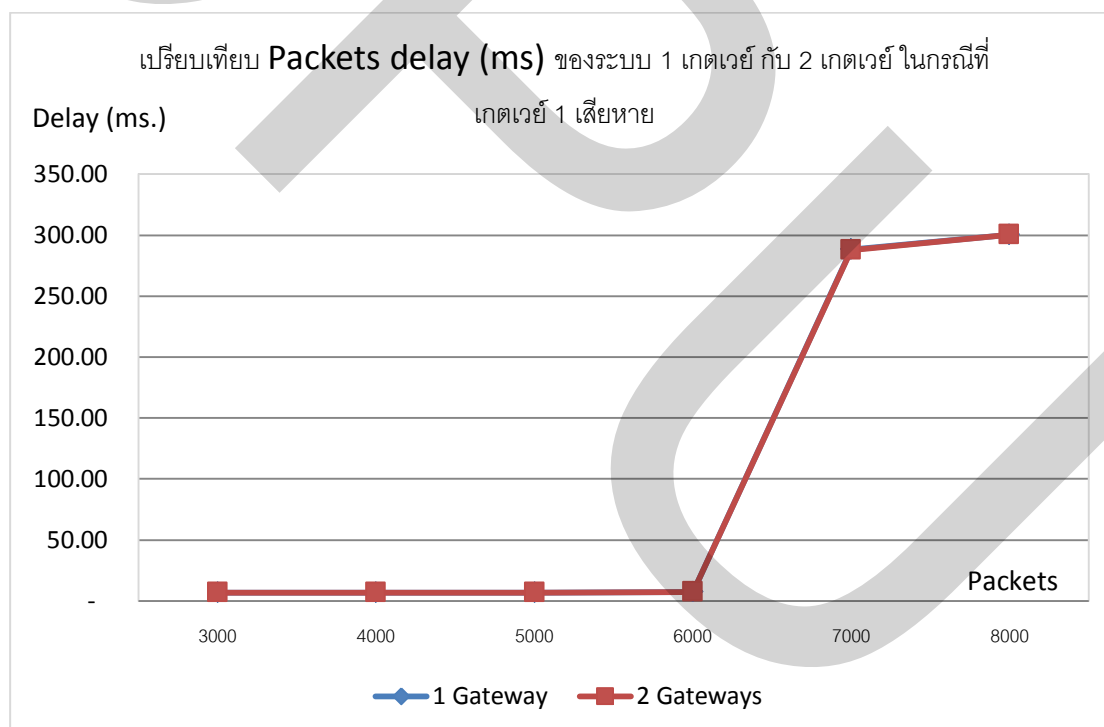
จากตารางที่ 4.5 นำมาแสดงในรูปแบบของกราฟเส้นได้ ตามภาพที่ 4.19 จะเห็นได้ว่าค่าการหน่วงเวลา packet delay ของเครือข่าย WSN แบบ 2 เกตเวย์ กับแบบ 1 เกตเวย์ จะมีค่าไม่ต่างกันมากเกือบทุกๆ ขนาด packets ที่ส่งเข้าไป เนื่องจากการทำงานของเครือข่าย WSN แบบ 2 เกตเวย์

นั้น ตัวเกตเวย์ 1 กับ 2 ไม่ได้ทำงานพร้อมกัน จึงไม่มีผลในการช่วยรับข้อมูลจาก โหนดต้นทาง โดยในตัวเกตเวย์ 2 จะทำงานก็ต่อเมื่อ เกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่ายเท่านั้น

ตารางที่ 4.6 ค่าการหน่วงเวลา packets delay ของแบบเกตเวย์ 1 เสียหาย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	6.988	6.982	6.980	7.340	288.135	300.478
2 Gateway	6.957	6.962	6.969	7.385	287.923	300.478

จากตารางที่ 4.6 เป็นผลการทดลองของเครือข่าย WSN แบบ 2 เกตเวย์ กับเครือข่ายแบบ 1 เกตเวย์ โดยเป็นการหาค่าการหน่วงเวลา packets delay ของเครือข่าย ในกรณีที่เกตเวย์ 1 ได้รับความเสียหาย ซึ่งนำมาแสดงเปรียบเทียบให้เห็นตามภาพที่ 4.20.



ภาพที่ 4.17 เปรียบเทียบ Packets delay (ms) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เสียหาย

ในภาพที่ 4.20 จะเห็นได้ว่าค่าการหน่วงเวลา ของเครือข่าย WSN แบบ 2 เกตเวย์ กับเครือข่ายแบบ 1 เกตเวย์ นั้นมีค่าไม่ต่างกันอย่างเห็นได้ชัด ซึ่งผลการทดลองมีลักษณะคล้ายกับแบบกรณีที่เกิดเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ทั้งนี้ก็เป็นเพราะการทำงานของเครือข่าย WSN แบบ 2 เกตเวย์ นั้น ตัวเกตเวย์ 1 กับ 2 ไม่ได้ทำงานพร้อมกัน จึงไม่มีผลในการช่วยรับข้อมูลจากโหนดคั่นทาง โดยในตัวเกตเวย์ 2 จะทำงานก็ต่อเมื่อ เกตเวย์ 1 เสียหายเท่านั้น

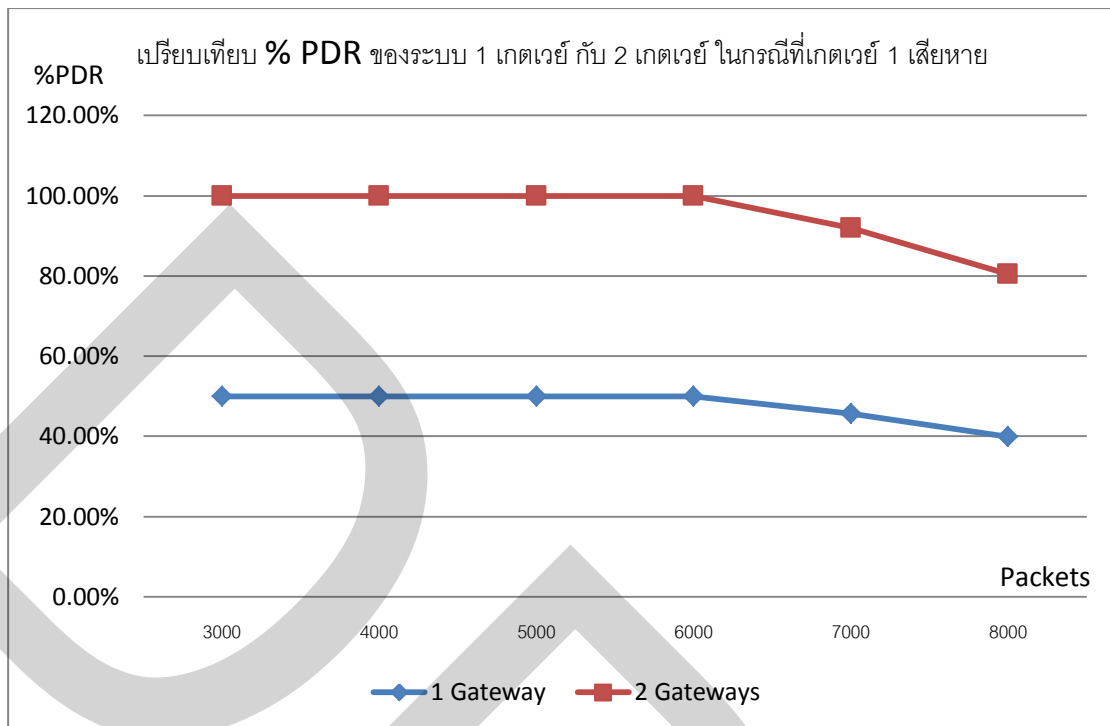
4.2.2 หาค่าความสำเร็จในการส่งข้อมูลจากโหนดคั่นทาง ไปยังโหนดปลายทาง (%PDR)

ในส่วนนี้จะเป็นการหาค่าความสำเร็จในการส่งข้อมูลจากโหนดคั่นทาง ไปยังโหนดปลายทาง (%PDR) โดยจะทำการเปรียบเทียบผลการทดลองทั้งแบบเครือข่าย WSN แบบ 2 เกตเวย์ และแบบเครือข่ายแบบ 1 เกตเวย์ ซึ่งมีทำการทดลองใน 2 กรณี คือ กรณีที่เกิดเวย์ 1 เคลื่อนที่ออกจากเครือข่าย กับกรณีที่เกิดเวย์ 1 เกิดการเสียหาย โดยผลการทดลองแสดงไว้ในตารางที่ 4.7 และต่างที่ 4.8

ตารางที่ 4.7 ค่าความสำเร็จในการส่งข้อมูล %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่เกิดเวย์ 1 เกิดการเสียหาย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	50.00%	49.99%	50.00%	49.98%	45.65%	39.95%
2 Gateway	99.93%	99.93%	99.92%	99.92%	91.97%	80.48%

จากตารางที่ 4.7 เป็นตารางแสดงผลการทดลองหาค่าความสำเร็จในการส่งข้อมูล %PDR ของ แบบเกตเวย์ 1 เสียหาย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.21



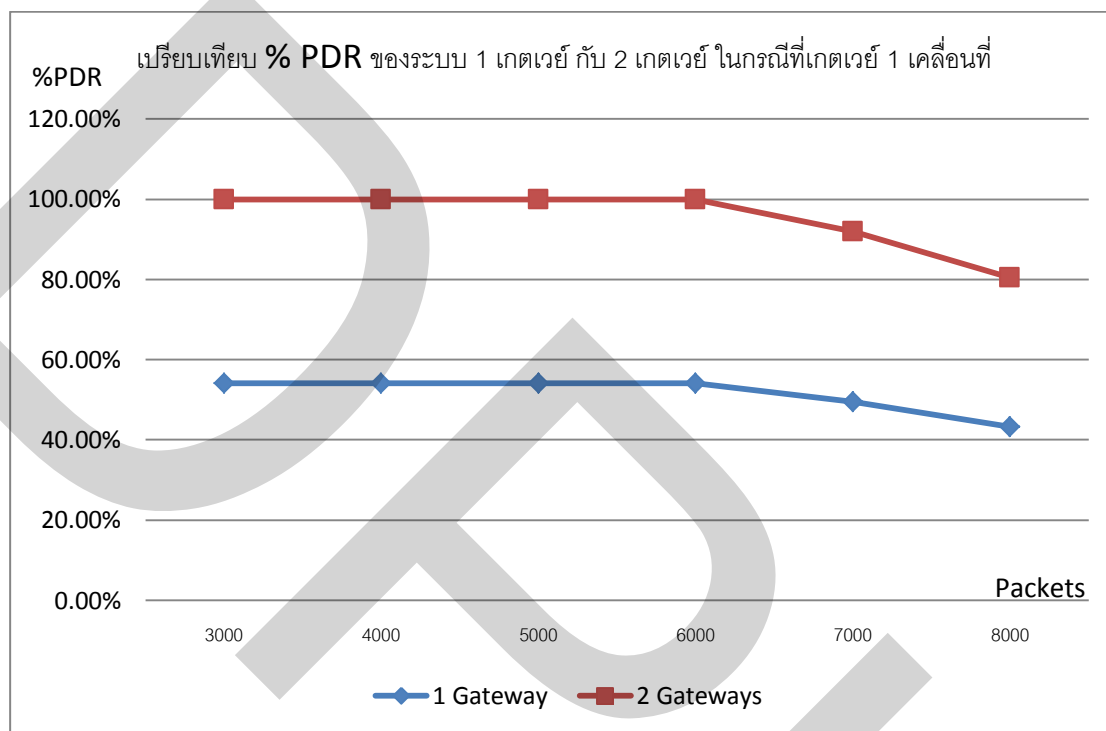
ภาพที่ 4.18 เปรียบเทียบ % PDR ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เสียหาย

ในภาพที่ 4.21 เป็นกราฟเส้นแสดงผลการทดลองหาค่าความสำเร็จในการส่งข้อมูล %PDR ของ แบบเกตเวย์ 1 เสียหาย ซึ่งจะเห็นได้ว่า ค่า %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ มีค่าสูงกว่า แบบ 1 เกตเวย์ ประมาณเกือบ 2 เท่า ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 6000 packets ค่า %PDR ของแบบ 2 เกตเวย์สูงเกือบ 100% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มลดลง ที่เป็นเช่นนี้น่าจะเกิดจาก ความเร็วของข้อมูลที่ถูส่งเข้ามามีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องของขนาด Buffer ที่จำกัดของตัวเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

ตารางที่ 4.8 ค่าความสำเร็จในการส่งข้อมูล %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	54.17%	54.16%	54.18%	54.17%	49.46%	43.29%
2 Gateway	99.90%	99.93%	99.94%	99.93%	92.00%	80.50%

จากตารางที่ 4.8 เป็นตารางแสดงผลการทดลองหาค่าความสำเร็จในการส่งข้อมูล %PDR ของ แบบเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.22



ภาพที่ 4.19 เปรียบเทียบ % PDR ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เคลื่อนที่

ในภาพที่ 4.19 เป็นกราฟเส้นแสดงผลการทดลองหาค่าความสำเร็จในการส่งข้อมูล %PDR ของ แบบเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งจะเห็นได้ว่า ค่า %PDR ของเครือข่าย WSN แบบ 2 เกตเวย์ มีค่าสูงกว่า แบบ 1 เกตเวย์ ประมาณเกือบ 2 เท่า ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 5000 packets ค่า %PDR ของแบบ 2 เกตเวย์สูงเกือบ 100% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มลดลง ที่เป็นเช่นนี้น่าจะเกิดจาก ความเร็วของข้อมูลที่ถูส่งเข้ามามีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องของขนาด Buffer ที่จำกัดของตัวเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

โดยเมื่อพิจารณาจากภาพที่ 4.18 และภาพที่ 4.19 แล้ว จะเห็นได้ว่าผลการทดลองมีแนวโน้มไปในทิศทางเดียวกัน คือตั้งแต่ 3000 ถึง 6000 packets ค่า %PDR ของแบบ 2 เกตเวย์สูง

เกือบ 100% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มลดลง ดังนั้นขนาดของ Buffer จึงน่าจะมีผลต่อค่าความสำเร็จการส่งข้อมูล %PDR

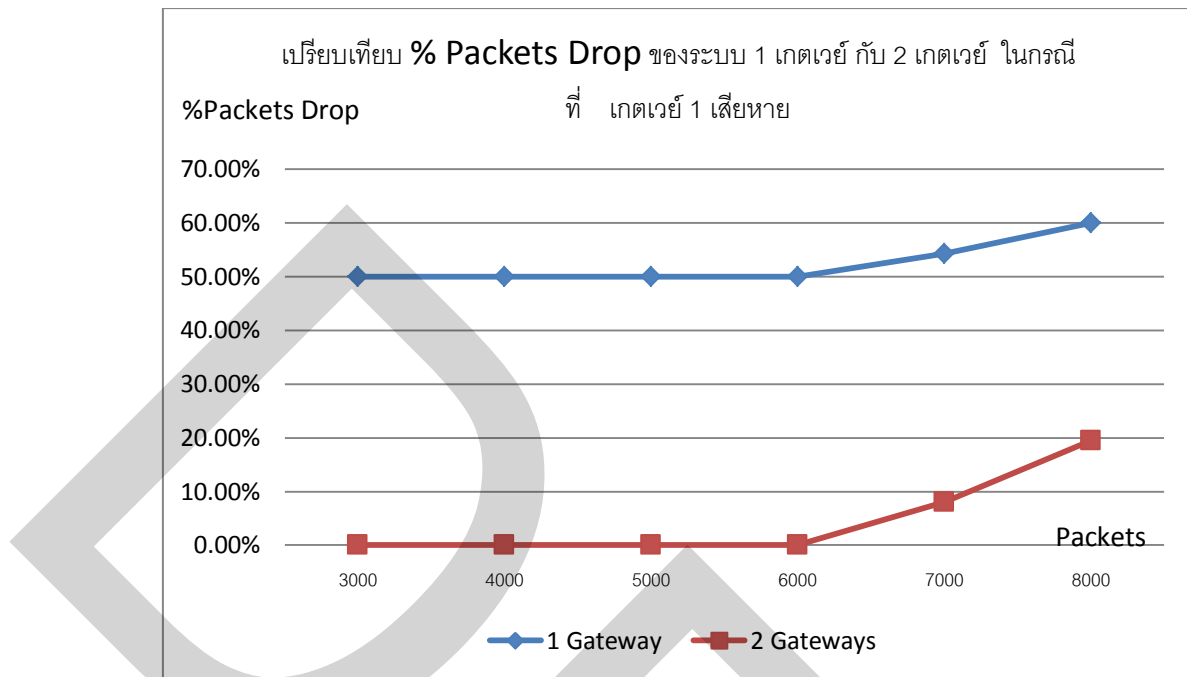
4.2.3 หาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop)

ในส่วนนี้จะเป็นการหาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ซึ่งจะทำการนับจำนวน packets drop ที่เกิดขึ้นในเครือข่าย ในการทดลองนี้จะนับในส่วนของ CBR packets drop ที่เกิดขึ้นที่ โหนดต่างๆ ในเครือข่าย WSN ทั้งแบบ 1 เกตเวย์ และแบบ 2 เกตเวย์ โดยพิจารณาทั้งในกรณี ที่เกตเวย์ 1 เกิดการเสียหาย และกรณีที่เกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย แล้วนำค่าที่นับได้มาเทียบกับจำนวน CBR packets ที่ส่งเข้าไปในเครือข่าย ที่ปริมาณความหนาแน่นของ packets ที่ต่างกัน ในช่วงเวลาเดียวกัน ซึ่งได้ผลการทดลองดังนี้

ตารางที่ 4.9 ค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยมีเกตเวย์ 1 เสียหาย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	50.00%	50.00%	50.00%	50.02%	54.34%	60.05%
2 Gateway	0.07%	0.05%	0.08%	0.08%	8.01%	19.53%

จากตารางที่ 4.9 เป็นตารางแสดงผลการทดลองหาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยมีเกตเวย์ 1 เสียหาย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.20



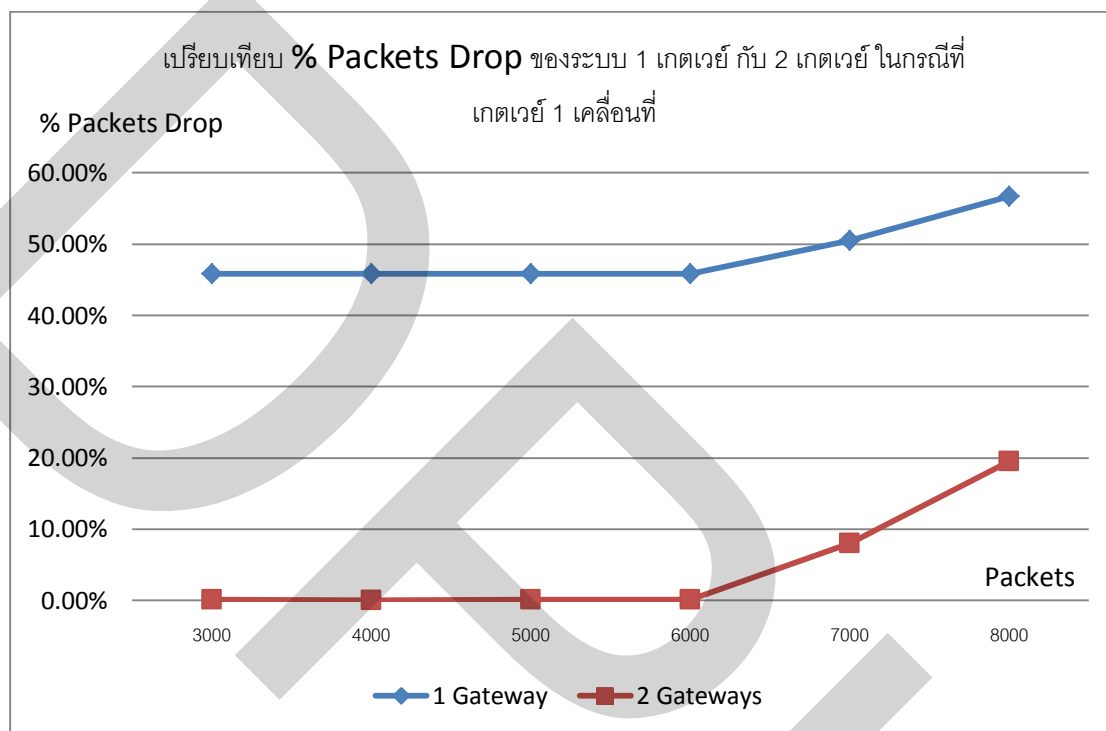
ภาพที่ 4.20 เปรียบเทียบ % Packets Drop ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เสียหาย

ในภาพที่ 4.20 เป็นกราฟเส้นแสดงผลการทดลองหาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เสียหาย ซึ่งจะเห็นได้ว่า ค่า %Packets drop ของเครือข่าย WSN แบบ 1 เกตเวย์ มีค่าสูงกว่า แบบ 2 เกตเวย์ ประมาณเกือบ 45% ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 6000 packets ค่า %Packets drop ของแบบ 2 เกตเวย์ต่ำเกือบ 0% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มมีค่ามากขึ้น ที่เป็นเช่นนี้น่าจะเกิดจากความเร็วของข้อมูลที่ส่งเข้ามามีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องขนาด Buffer ที่จำกัดของตัวโหนดเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

ตารางที่ 4.10 ค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	45.83%	45.83%	45.82%	45.83%	50.53%	56.71%
2 Gateway	0.10%	0.05%	0.06%	0.07%	7.99%	19.50%

จากตารางที่ 4.10 เป็นตารางแสดงผลการทดลองหาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.24



ภาพที่ 4.21 เปรียบเทียบ % Packets Drop ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 มีการเคลื่อนที่

ในภาพที่ 4.21 เป็นกราฟเส้นแสดงผลการทดลองหาค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งจะเห็นได้ว่า ค่า %Packets drop ของเครือข่าย WSN แบบ 1 เกตเวย์ มีค่าสูงกว่า แบบ 2 เกตเวย์ ประมาณเกือบ 45% ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 6000 packets ค่า %Packets drop ของแบบ 2 เกตเวย์ต่ำเกือบ 0% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มมีค่ามากขึ้น ที่เป็นเช่นนี้จะเกิดจากความเร็วของข้อมูลที่วิ่งส่งเข้ามาที่มีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องขนาด Buffer ที่จำกัดของตัวโหนดเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

โดยเมื่อพิจารณาจากภาพที่ 4.20 และภาพที่ 4.21 แล้ว จะเห็นได้ว่าผลการทดลองมีแนวโน้มไปในทิศทางเดียวกัน คือตั้งแต่ 3000 ถึง 6000 packets ค่า %Packets drop ของเครือข่าย WSN แบบ 2 เกตเวย์ มีค่าต่ำเกือบ 0% ส่วนตั้งแต่ 6000 ถึง 8000 เริ่มสูงขึ้น ดังนั้นขนาดของ Buffer จึงน่าจะมีผลต่อค่าปริมาณข้อมูลที่สูญเสียในเครือข่าย (% Packets drop)

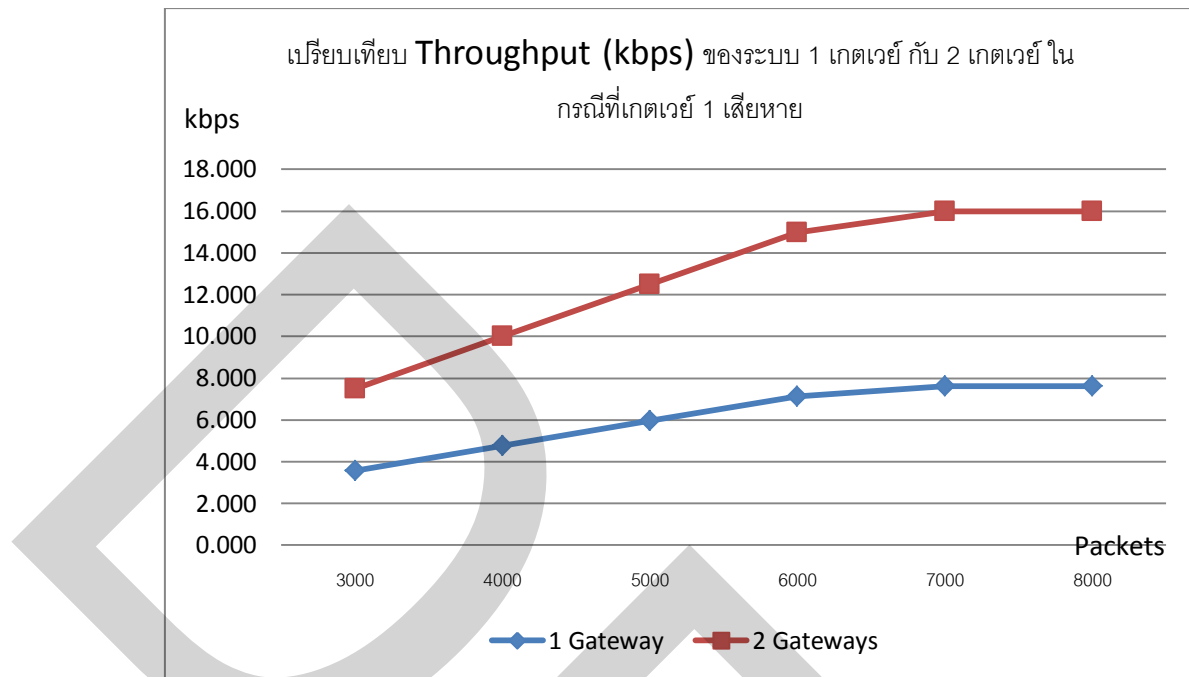
4.2.4 หาค่าปริมาณข้อมูลที่เกตเวย์ 1 และเกตเวย์ 2 รับผิดชอบกับเวลา (Throughput kbps)

ในการทดลองส่วนสุดท้ายนี้จะเป็นการหาค่าปริมาณข้อมูลที่เกตเวย์ 1 และเกตเวย์ 2 รับผิดชอบกับเวลา (Throughput kbps) ซึ่งจะทำให้การนับจำนวน CBR packets ที่ส่งเข้าไปในเครือข่าย เทียบกับ CBR packets ที่สามารถรับได้ที่เกตเวย์ 1 เกตเวย์ 2 แล้วนำค่าที่ได้มารวมกัน บนเครือข่าย WSN ทั้งแบบ 1 เกตเวย์ และแบบ 2 เกตเวย์ โดยพิจารณาทั้งในกรณี ที่เกตเวย์หลักเกิดการเสียหาย และกรณีที่เกตเวย์หลักเคลื่อนที่ออกนอกเครือข่าย ซึ่งได้ผลดังนี้

ตารางที่ 4.11 ปริมาณข้อมูลที่เกตเวย์ 1 และเกตเวย์ 2 รับผิดชอบกับเวลา (Throughput kbps) กรณีเกตเวย์ 1 เสียหาย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	3.57	4.76	5.95	7.14	7.61	7.61
2 Gateway	7.50	9.99	12.49	14.99	15.97	15.97

จากตารางที่ 4.11 เป็นตารางแสดงผลการทดลองหาค่าปริมาณข้อมูลที่เกตเวย์ 1 และที่เกตเวย์ 2 สามารถรับได้เทียบกับเวลา (Throughput kbps) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เสียหาย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.25



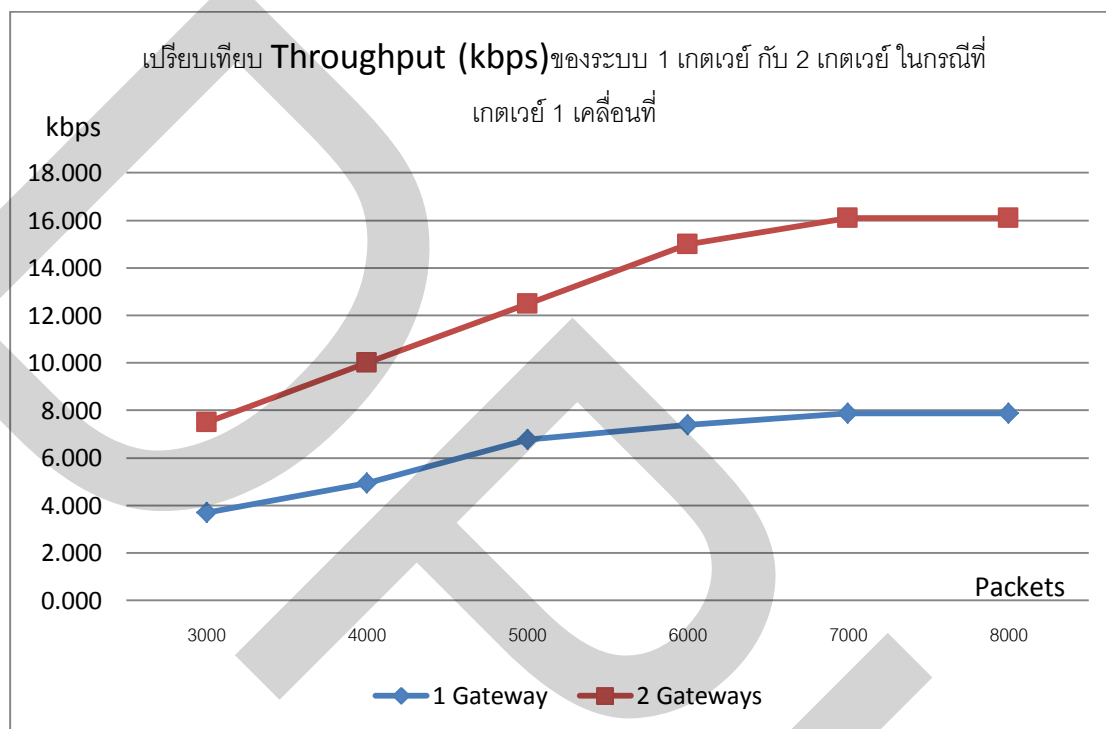
ภาพที่ 4.22 เปรียบเทียบ Throughput (kbps) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เสียหาย

ในภาพที่ 4.22 เป็นกราฟเส้นแสดงผลการทดลองหาค่าปริมาณข้อมูลที่เกตเวย์ 1 และที่เกตเวย์ 2 สามารถรับได้เทียบกับเวลา (Throughput kbps) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เสียหาย ซึ่งจะเห็นได้ว่า ค่า Throughput ของเครือข่าย WSN แบบ 2 เกตเวย์ มีค่าสูงกว่าแบบ 1 เกตเวย์ ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 7000 packets ค่า Throughput ของเครือข่ายมีความชันสูงกว่าค่า Throughput ของข้อมูลตั้งแต่ 7000 ถึง 8000 ที่เป็นเช่นนี้น่าจะเกิดจากความเร็วของข้อมูลที่ถูส่งเข้ามาที่มีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องของขนาด Buffer ที่จำกัดของตัว โหนดเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

ตารางที่ 4.12 ปริมาณข้อมูลที่เกตเวย์ 1 และเกตเวย์ 2 รับได้เทียบกับเวลา (Throughput kbps) กรณีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
1 Gateway	3.69	4.93	6.77	7.39	7.87	7.87
2 Gateway	7.49	10.00	12.49	14.99	16.10	16.10

จากตารางที่ 4.12 เป็นตารางแสดงผลการทดลองหาค่าปริมาณข้อมูลที่เกตเวย์ 1 และที่เกตเวย์ 2 สามารถรับได้เทียบกับเวลา (Throughput kbps) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.26



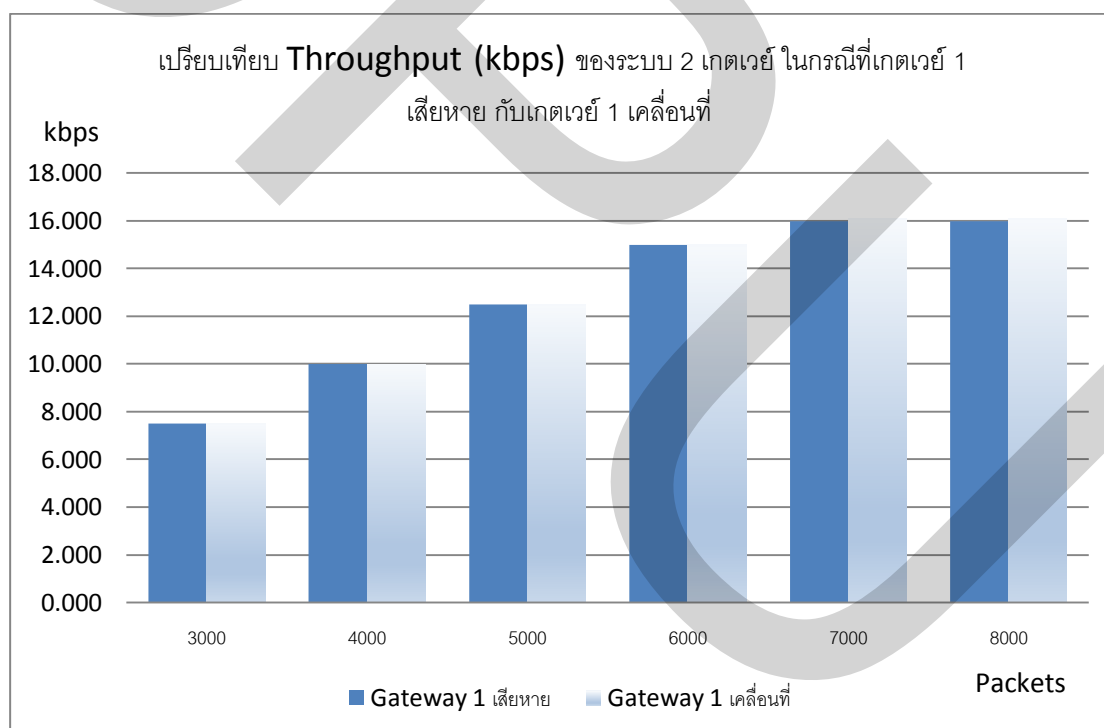
ภาพที่ 4.23 เปรียบเทียบ Throughput (kbps) ของระบบ 1 เกตเวย์ กับ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เคลื่อนที่

ในภาพที่ 4.23 เป็นกราฟเส้นแสดงผลการทดลองหาค่าปริมาณข้อมูลที่เกตเวย์ 1 และที่เกตเวย์ 2 สามารถรับได้เทียบกับเวลา (Throughput kbps) ของเครือข่าย WSN แบบ 2 เกตเวย์ โดยที่มีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งจะเห็นได้ว่า ค่า Throughput ของเครือข่าย WSN แบบ 2 เกตเวย์ มีค่าสูงกว่า แบบ 1 เกตเวย์ ที่ทุกๆ ปริมาณ packets ที่ส่งเข้าไป โดยตั้งแต่ 3000 ถึง 7000 packets ค่า Throughput ของเครือข่ายมีความชันสูงกว่าค่า Throughput ของข้อมูลตั้งแต่ 8000 ถึง 8000 ที่เป็นเช่นนี้น่าจะเกิดจากความเร็วของข้อมูลที่ถูส่งเข้ามาที่มีค่าสูงขึ้น จนเกิดปัญหาเรื่องจาก Drop ของข้อมูล จากปัญหาเรื่องของขนาด Buffer ที่จำกัดของตัวเอง ซึ่งลักษณะเช่นนี้เกิดขึ้นในทิศทางเดียวกันทั้งแบบ 1 เกตเวย์ และ 2 เกตเวย์

ตารางที่ 4.13 ปริมาณข้อมูลของเครือข่ายแบบ 2 เกตเวย์ ที่รับได้เทียบกับเวลา (Throughput kbps) โดยเปรียบเทียบกรณีเกตเวย์ 1 เสียหาย และกรณีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย

Packets	3,000	4,000	5,000	6,000	7,000	8,000
Gateway 1 damage	7.50	9.99	12.49	14.99	15.97	15.97
Gateway 1 move	7.49	10.00	12.49	14.99	16.10	16.10

จากตารางที่ 4.13 เป็นตารางแสดงผลการทดลองหาค่าปริมาณข้อมูลของเครือข่ายแบบ 2 เกตเวย์ ที่รับได้เทียบกับเวลา (Throughput kbps) โดยเปรียบเทียบกรณีเกตเวย์ 1 เสียหาย และกรณีเกตเวย์ 1 เคลื่อนที่ออกนอกเครือข่าย ซึ่งนำมาแสดงเป็นกราฟเส้นได้ตามภาพที่ 4.27



ภาพที่ 4.24 เปรียบเทียบ Throughput (kbps) ของระบบ 2 เกตเวย์ ในกรณีที่เกตเวย์ 1 เสียหาย กับเกตเวย์ 1 เคลื่อนที่

จากภาพที่ 4.24 แสดงให้เห็นถึงผลการเปรียบเทียบปริมาณ Throughput ของเครือข่าย WSN แบบใช้ 2 เกตเวย์ โดยเกตเวย์ตัวที่ 2 จะทำงานก็ต่อเมื่อ เกตเวย์ 1 เสียหาย หรือ เคลื่อนที่ออกนอกเครือข่าย ซึ่งเมื่อเปรียบเทียบผลการทดลองทั้ง 2 กรณีแล้ว พบว่าสามารถทำงานได้ใกล้เคียงกันมาก โดยแบบเกตเวย์ 1 เคลื่อนที่ จะเริ่มมีค่า Throughput มากกว่าเล็กน้อย ในกรณีที่ปริมาณข้อมูลในเครือข่ายมีค่าตั้งแต่ 6000 ถึง 8000 ซึ่งน่าจะเป็นผลมาจากความเร็วในการส่งข้อมูลต่อเวลาที่มากขึ้นจึงทำให้มีโอกาสที่ปริมาณข้อมูลที่รับได้ที่เกตเวย์ มีค่ามากกว่ากรณีที่มีความเร็วในการส่งข้อมูลต่อเวลาที่ต่ำกว่า และมีปริมาณ %Packets drop ในกรณีเกตเวย์เคลื่อนที่ มีค่าน้อยกว่ากรณีเกตเวย์เสียหาย

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอเครือข่ายเซนเซอร์ไร้สายตามมาตรฐาน IEEE802.15.4 แบบหลายเกตเวย์ เพื่อหาเส้นทางสำรองให้กับ โหนดในเครือข่ายเซนเซอร์ไร้สาย Wireless Sensor Network (WSN) ในกรณีที่ตัวเกตเวย์หลักที่ใช้งานอยู่ปัจจุบันเกิดปัญหาไม่สามารถส่งข้อมูลออกนอกเครือข่าย WSN ได้ โดยจากเครือข่าย WSN แบบหลายเกตเวย์ที่ได้สร้างขึ้นและทำการทดลอง พบว่าสามารถทำงานได้ดี และมีประสิทธิภาพ โดยดูได้จากการวัดผลประสิทธิภาพของระบบ เช่น ปริมาณงาน Throughput รวมของระบบที่ค่าสูงขึ้นเกือบเท่าตัวเมื่อเทียบกับเครือข่าย WSN เดิมที่มี 1 เกตเวย์

ถึงแม้ว่าเครือข่ายเซนเซอร์ไร้สายแบบเกตเวย์หลายตัวจะสามารถเพิ่มประสิทธิภาพในการส่งข้อมูลออกนอกเครือข่ายได้ แต่ก็ต้องสูญเสียพลังงานที่ใช้ไปกับการรอเพื่อที่จะทำงานของเกตเวย์สำรอง ซึ่งมีแนวโน้มที่จะเพิ่มขึ้นเมื่อความหนาแน่นของข้อมูลในเครือข่ายมีค่ามากขึ้น อันเนื่องมาจากการทำหน้าที่เปรียบเสมือน โหนดแบบ FFD ในเครือข่าย WSN

5.2 ปัญหาและข้อเสนอแนะ

5.2.1 ปัญหาที่พบในงานวิจัย

ในงานวิจัยนี้ใช้ตัวโปรแกรมจำลองการทำงานของเครือข่าย NS2 เวอร์ชัน 2.35 ซึ่งเป็นโปรแกรมที่ต้องอาศัยระยะเวลาในการศึกษาทำความเข้าใจที่นาน จึงทำให้ต้องเสียเวลาในการทดลองไปค่อนข้างมาก และในส่วนของ การอ่านข้อมูลจาก Trace ไฟล์ ที่มีขนาดใหญ่ จึงทำให้ต้องใช้เวลาในการวิเคราะห์ห้านาน

ในงานวิจัยนี้เป็นงานวิจัยเกี่ยวกับการศึกษาเรื่องของการเปลี่ยนเกตเวย์ในเครือข่าย WSN ซึ่งยังไม่ครอบคลุมถึงการวิเคราะห์เรื่องของข้อมูลที่จะต้องถูกส่งต่อไปยังเครือข่ายอื่นๆ เพื่อเก็บ ว่ามีข้อมูลสูญหายระหว่างทางหรือไม่

5.2.2 ข้อเสนอแนะ

ในเรื่องของการวิเคราะห์ข้อมูลน่าจะใช้การเขียนโปรแกรมเพื่อเปิดไฟล์มาวิเคราะห์ข้อมูลแทนการใช้โปรแกรมสำเร็จรูป

ในเรื่องของการวิเคราะห์ข้อมูลที่สูญหายในเครือข่ายน่าจะลองให้ส่วนของโปรโตคอลที่เป็น TCP แทนการใช้ UDP

ควรวางวิธีในการจัดการเรื่องพลังงานที่สูญเสียในเกตเวย์สำรองในระหว่างไม่ได้ทำงาน



บรรณานุกรม

บรรณานุกรม

ภาษาไทย

บทความ

อนุรักษ์ เขยชุ่ม และ ชัยพร เขมะภักตะพันธ์.(2553). "การประยุกต์เครือข่ายตรวจจับสัญญาณไร้สาย เพื่อประมวลผลข้อมูลความเร็วลมสำหรับช่วยตัดสินใจในการติดตั้งกังหันลมผลิตกระแสไฟฟ้า". การประชุมการวิจัยระดับบัณฑิตศึกษาแห่งชาติ มหาวิทยาลัยเชียงใหม่ ครั้งที่ 2. 26 พฤศจิกายน 2553. หน้า 620-629.

นภาพร พิมปรู และ ประโยชน์ คำสวัสดิ์.(2556). "การบีบอัดข้อมูลในเครือข่ายเซนเซอร์ไร้สาย" **ECTI-CARD 2013**. หน้า 173-178.

สารสนเทศจากสื่ออิเล็กทรอนิกส์

จิตติมา นิตยวรรณ โสฬส ชัคัตตยาพงษ์ (2550). Network Simulator (NS2)

สืบค้นข้อมูลเมื่อวันที่ 25 มกราคม 2557. จาก

<http://wiki.nectec.or.th/ngiwiki/pub/Project/IntelligentNMS/bm-ns2.pdf>

ภาษาต่างประเทศ

ARTICLES

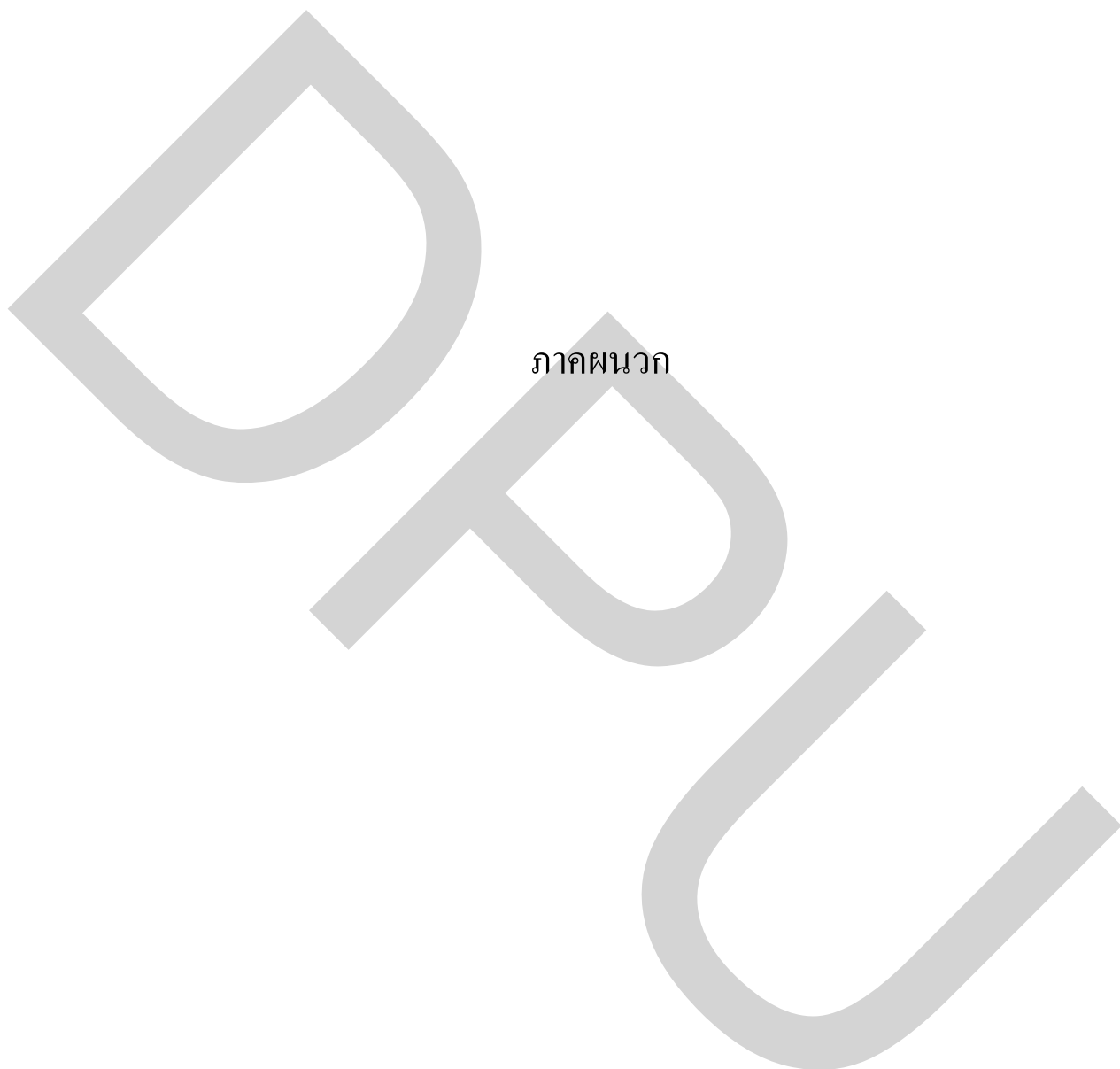
Luca De Nardis & Maria-Gabriella.(2007, March). "Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks" *WPNC 2007*. Germany.

P. Manoyut & W. Suntiamorntut.(2012, February). "Modified Local Repair in AODV for LoWPANs" *IMCOM 2012*. Kula Lumpur. Malaysia.

Xuan He & Yuanchen Ma.(2012, June). "Efficient Multiple Gateway System for WSN Management in BEMS" *Networked Sensing Systems (INSS)*. Belgium.

Zahia Bidai & Hafid Haffaf.(2011, April). “Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks”. *Multimedia Computing and Systems*. Morocco.





ภาคผนวก

ผลงานทางวิชาการ



THE 10TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

PROCEEDINGS OF NCCIT 2014

THE 10TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

8TH-9TH MAY 2014
ANGSANA LAGUNA PHUKET, THAILAND

WWW.NCCIT.NET

FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

บทความวิจัย

การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
ครั้งที่ 10

8-9 พฤษภาคม 2557
โรงแรมอังกูรา ลากูนา ภูเก็ต



คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

WWW.IT.KMUTNB.AC.TH

เครือข่ายเซนเซอร์ไร้สาย IEEE802.15.4 แบบหลายเกตเวย์ Multiple Gateways System for Wireless Sensor Network IEEE802.15.4

ชนะเดช แย้มคลี่ (Chanadej Yamclee)¹ และชัยพร เขมะภาคะพันธ์ (Chaiyaporn Khemapatapan)²

สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์

มหาวิทยาลัยธุรกิจบัณฑิตย์

¹y_chanadej@hotmail.com, ²chaiyaporn@dpu.ac.th

บทคัดย่อ

การทำงานของเครือข่ายเซนเซอร์ไร้สายโดยทั่วไปจะมีการส่งข้อมูลที่ได้จากโหนดเซนเซอร์ไปยังโหนดโคออดิเนเตอร์ที่ทำหน้าที่เป็นเกตเวย์เชื่อมต่อกับเครือข่ายภายนอกที่มีอยู่เพียง 1 โหนด เพื่อส่งต่อข้อมูลออกจากเครือข่ายไปยังปลายทางที่ต้องการ อย่างไรก็ตามเมื่อโหนดที่เป็นเกตเวย์เกิดการเสียหายจะทำให้ไม่สามารถส่งข้อมูลออกนอกเครือข่ายนั้นได้ งานวิจัยนี้ได้นำเสนอวิธีการเพิ่มโหนดที่เป็นเกตเวย์จำนวน 2 โหนด และประยุกต์โพรโทคอลการค้นหาเส้นทางแบบ AODV กับเครือข่ายไร้สายตามมาตรฐาน IEEE802.15.4 ร่วมกับการเปลี่ยนเกตเวย์ที่ใช้เป็นโหนดปลายทางในกรณีที่โหนดที่เป็นเกตเวย์โหนดดั้งเดิมไม่สามารถทำงานได้

ผลการทดสอบโดยการจำลองการทำงานผ่านโปรแกรม Network Simulator 2 (NS-2) พบว่าค่า Throughput ของเครือข่ายเซนเซอร์ไร้สายที่เสนอมีค่าสูงขึ้นเฉลี่ย 1.35 Kbps และค่า Packet Delivery Ratio (PDR) มีค่าสูงขึ้นเฉลี่ย 5.39% เมื่อเทียบกับเครือข่ายเซนเซอร์ไร้สายแบบดั้งเดิม

คำสำคัญ: เครือข่ายเซนเซอร์ไร้สาย AODV หลายเกตเวย์

Abstract

The operation of wireless sensor network (WSN) is normally forwarding data from sensor node to gateway node or coordination node which connects to another network. The gateway node will subsequently transfer the data to the specific destination. When the gateway node has been damaged, the data cannot be sent out. Thus, this research presents a method by adding 2 gateway nodes and applying the routing protocol AODV over

IEEE802.15.4 WSN including gateway changing when a default gateway node comes to shutdown.

The results from the simulation using NS2 program found that throughput and Packet Delivery Ratio (PDR) of the proposed WSN increase averagely 1.35 Kbps and 5.39%, respectively, in comparison with conventional WSN.

Keyword: AODV, Wireless Sensor Network, Multiple Gateway, IEEE802.15.4

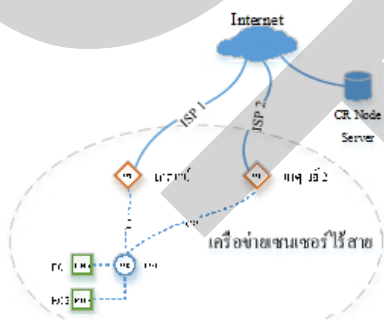
1. บทนำ

ในปัจจุบันเครือข่ายเซนเซอร์ไร้สายหรือ WSN ถูกนำมาใช้งานจริงในภาคสนามอย่างแพร่หลายทั่วโลก ในประเทศไทย มีการนำมาใช้ในงานด้านการสำรวจทางธรณีวิทยาเช่น การตรวจจัดการดินดินถล่ม งานด้านการทหาร งานด้านการเกษตร เช่นการตรวจวัด อุณหภูมิ ความชื้นในดิน [1], [2] ในการนำเอาตัวเครือข่ายเซนเซอร์ไร้สายมาประยุกต์ใช้ในงานด้านต่างๆ ก็เพื่อการเก็บข้อมูลที่ต้องการที่หน้างาน และส่งข้อมูลที่ได้เข้ามาที่ส่วนกลางเพื่อประมวลผล ซึ่งโดยโครงสร้างของ WSN จะมีการส่งข้อมูลจากโหนดเซนเซอร์มายังโหนดที่เป็นเกตเวย์ผ่านโหนดเรเตอร์ ซึ่งโหนดปลายทางที่เป็นโหนดโคออดิเนเตอร์ซึ่งทำหน้าที่เป็นเกตเวย์จะมีอยู่เพียงแค่โหนดเดียวภายในเครือข่าย [3] ดังนั้นเมื่อโหนดเกตเวย์ได้รับความเสียหายก็จะไม่สามารถรับและส่งข้อมูลที่ต้องการออกนอกเครือข่ายได้ จนทำให้ข้อมูลเกิดการสูญหายสาเหตุเนื่องจากปัญหาเรื่องพลังงานและขนาดของหน่วยความจำที่มีขนาดจำกัด

ถึงแม้ว่าจะมีการแก้ปัญหาโดยการเพิ่มขนาดหน่วยความจำของตัวโหนดที่เป็นเกตเวย์ให้มีขนาดมากขึ้น เพื่อเพิ่มการ

รองรับข้อมูลจากตัวโหนดเซนเซอร์ที่ต้องส่งเข้ามาที่ตัวโหนดที่เป็นเกตเวย์ ในขณะที่ไม่สามารถส่งข้อมูลออกนอกเครือข่ายได้ชั่วคราว แต่ก็ไม่สามารถแก้ปัญหาในกรณีที่โหนดเกิดความเสียหายจนไม่สามารถส่งและรับข้อมูลได้แบบถาวร

ดังนั้นงานวิจัยนี้จะแสดงให้เห็นถึงประโยชน์ และวิธีในการเพิ่มโหนดที่เป็นเกตเวย์จาก 1 โหนด เป็น 2 โหนด ซึ่งในเบื้องต้นจะทำการเพิ่มจำนวนโหนดที่เป็นเกตเวย์แค่ 1 โหนด เนื่องจากสามารถแสดงให้เห็นถึงกรณีที่เกิด SPOF ที่โหนดได้ในทอพอโลยีแบบเดียวกัน ซึ่งโหนดที่เพิ่มจะอยู่ไกลกว่าโหนดเดิมและทดสอบโดยการส่งข้อมูลแบบ Constant Bit Rate (CBR) ผ่านโปรโตคอล Ad Hoc On-Demand Distance Vector (AODV) โดยจะทำการเปรียบเทียบปริมาณ Throughput ระหว่างโหนดที่เป็นเกตเวย์จาก 1 และ 2 โหนด ในกรณีที่โหนดที่เป็นเกตเวย์ 1 หยุดทำงาน



ภาพที่ 1: เครือข่ายเซนเซอร์ไร้สายแบบ 2 เกตเวย์

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

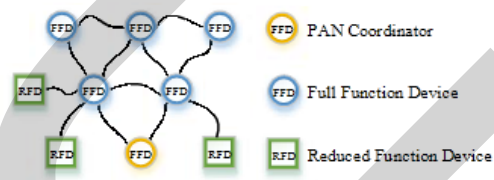
2.1 เครือข่ายเซนเซอร์ไร้สาย IEEE802.15.4

WSN มีหลายสถาบันที่ช่วยกันพัฒนา ซึ่ง Institute of Electrical and Electronics Engineers (IEEE) เป็นสถาบันที่มุ่งเน้นที่ชั้นที่ 1 กายภาพ (Physical layer) และชั้นที่ 2 ชั้นสื่อสารข้อมูล (Data link) ส่วน Internet Engineering Task Force (IETF) จะมุ่งเน้นไปที่ชั้นที่ 3 ชั้นเครือข่าย (Network layer) และชั้นที่สูงกว่าขึ้นไป ตามการจัดเรียงใน OSI model

โครงสร้างของ WSN จะมีโหนดที่เป็นเกตเวย์ ซึ่งทำหน้าที่ระบุช่องทางทางการสื่อสารภายในเครือข่ายระหว่างโหนดในเครือข่าย เพื่อไม่ทับซ้อนกัน และป้องกันการกวนกันของคลื่น

ความถี่ (Interference) โดยการสแกนหาระดับของพลังงานแต่ละช่องที่รับได้ ซึ่งโหนดใดๆ ที่อยู่ในช่วงสัญญาณเดียวกันกับโหนดเกตเวย์และต้องการจะเชื่อมต่อกับเครือข่าย เพื่อส่งข้อมูลไปยังตัวโหนดเกตเวย์ โหนดเหล่านั้นก็จะทำการส่งสัญญาณแจ้งเตือนออกมาร้องขอ (Beacon request) เพื่อขอเข้าร่วมเครือข่าย ซึ่งจะมีเพียงโหนดที่เป็นเกตเวย์เท่านั้นที่จะตอบสนองต่อการร้องขอเหล่านี้ได้

WSN จะประกอบไปด้วยโหนดต่างๆ จำนวนมาก แต่สามารถแบ่งออกเป็นสองแบบหลักคือ โหนดที่มีฟังก์ชันการทำงานแบบเต็ม Full-function device (FFD) และโหนดที่ถูกตัดฟังก์ชันที่ไม่จำเป็นออก Reduced-function device (RFD) ซึ่งในเครือข่ายจะต้องประกอบไปด้วย FFD อย่างน้อย 1 ตัวเพื่อทำหน้าที่เป็น PAN coordinator โดยตัวโหนดที่เป็น FFD จะสามารถทำหน้าที่เป็น โหนดเรดเดอร์ และโหนดที่เป็นเกตเวย์ โดยโหนดเรดเดอร์จะทำหน้าที่ในการกำหนดเส้นทางให้กับโหนดต่างๆ ที่ต้องการส่งข้อมูลไปยังโหนดที่เป็นเกตเวย์ โดยโหนดนี้จะทำหน้าที่ในการจัดการเครือข่าย และสามารถทำหน้าที่เป็นเกตเวย์ โดยการเพิ่มส่วนของการติดต่อกับเครือข่ายอินเตอร์เน็ต เพื่อนำข้อมูลที่ได้ใช้ และโหนดประเภท RFD จะทำหน้าที่เป็นโหนดเซนเซอร์ ซึ่งจะเก็บข้อมูลจากตัวเซนเซอร์ต่างๆ ที่ทำงาน เช่น อุณหภูมิ จากนั้นจะส่งข้อมูลไปยังโหนดที่เป็นเกตเวย์ผ่านทางโหนดเรดเดอร์



ภาพที่ 2: โครงสร้างของเครือข่ายเซนเซอร์ไร้สาย

2.2 Ad Hoc On-Demand Distance Vector (AODV)

AODV[4] เป็นโปรโตคอลเส้นทางตามมาตรฐาน RFC3561 ที่ใช้ทรัพยากรในการประมวลผล และหน่วยความจำน้อย จึงเหมาะกับ WSN ที่ต้องการเน้นเรื่องประหยัดพลังงาน AODV ใช้หลักการแบบ distance vector ของ Bellman-Ford ในการสร้างเส้นทาง และเป็น โปรโตคอลแบบ On-Demand ซึ่งจะทำ

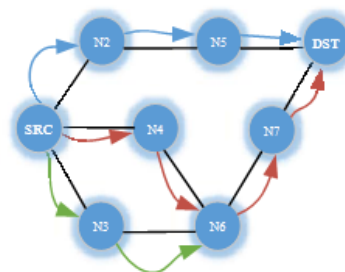
การสร้างเส้นทางเมื่อต้องการส่งข้อมูล ทำให้ลดการ Broadcast RREQ ได้ ตัวโปรโตคอล AODV จะพยายามทำให้โหนดนั้นๆ ได้รับเส้นทางโดยเร็วที่สุดเมื่อมีการกำหนดโหนดปลายทางใหม่ โดยอาศัยกระบวนการปรับปรุง และตรวจสอบเส้นทาง ที่ไปยังโหนดข้างเคียงว่าสามารถใช้งานได้หรือไม่ หรือเมื่อโครงสร้างเครือข่ายเปลี่ยนแปลง โดยให้โหนดนั้นๆ ยกเลิกการใช้เส้นทางที่ไม่สามารถใช้งานได้

โปรโตคอล AODV ประกอบไปด้วย Message ต่างๆ คือ Route Request (RREQ) Route Replies (RREP) Route Errors (RERR) เมื่อใดที่โหนดต้นทางต้องการส่งข้อมูลไปยังโหนดปลายทาง โหนดต้นทางจะ Broadcasts Message RREQ เพื่อค้นหาเส้นทางไปยังโหนดปลายทางนั้น และเส้นทางจะถูกสร้างขึ้นเมื่อ Message RREQ ไปถึงยังโหนดปลายทาง หรือในโหนดระหว่างเส้นทางนั้นมีเส้นทางที่สามารถไปยังโหนดปลายทาง อยู่ในตารางเส้นทาง โหนดนั้นก็จะส่งส่ง Message RREP กลับไปยังโหนดต้นทางโดยวิธีการ Unicast

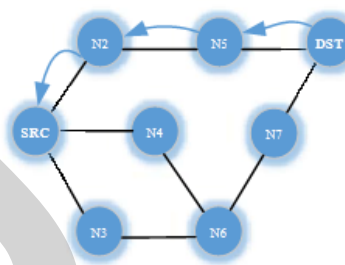
2.3 งานวิจัยที่เกี่ยวข้อง

งานวิจัยของ Xuan He, Yuanchen Ma และ Mika Mizutani ได้นำเสนอ Efficient Multiple Gateway System for WSN Management in BEMS [5] ซึ่งเป็นการเพิ่มโหนดโคออดิเนเตอร์ย่อยๆ ขึ้นมา แต่โหนดเหล่านี้ถ้าต้องการส่งข้อมูลออกนอกเครือข่าย จะต้องส่งผ่านโหนดที่เป็นเกตเวย์หลัก ดังนั้นเมื่อโหนดโคออดิเนเตอร์หลักได้รับความเสียหาย ทำให้เกิด Single of Failure (SPOF) ข้อมูลก็จะไม่สามารถส่งออกนอกเครือข่ายได้เช่นเดิม

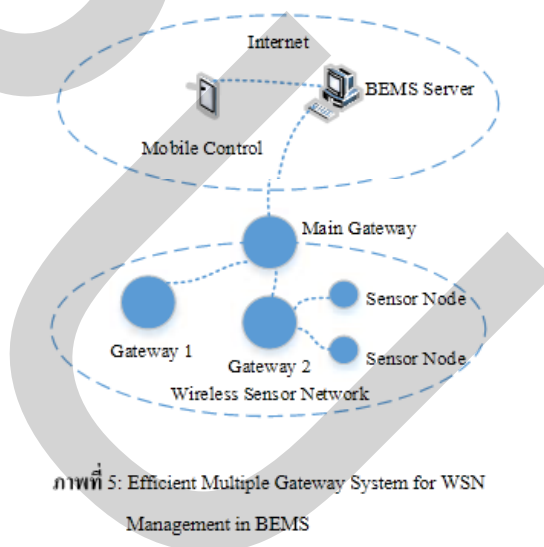
งานวิจัยของ Zahia Bidai, Hafid Haffaf, Moufida Maimour ได้นำเสนอ Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks [6] โดยต้องการส่งข้อมูลที่ขนาดใหญ่ เช่น ภาพเคลื่อนไหว ผ่านทางหลายเส้นทางไปยังโหนดปลายทางโคออดิเนเตอร์เดียว ปัญหาที่เกิดขึ้นคือมีการชนกันของข้อมูลที่โหนดปลายทาง และมีการกวนกันของข้อมูลระหว่างเส้นทาง จึงไม่สามารถเพิ่มปริมาณ Throughput ได้มาก และไม่ส่งข้อมูลออกนอกเครือข่ายไปใช้งานได้ ในกรณีที่โหนดที่เป็นเกตเวย์เสีย จึงทำให้เกิด SPOF ในเครือข่ายได้



ภาพที่ 3: โหนดต้นทางกระจาย RREQ Message



ภาพที่ 4: โหนดปลายทางตอบ RREP Message



ภาพที่ 5: Efficient Multiple Gateway System for WSN Management in BEMS

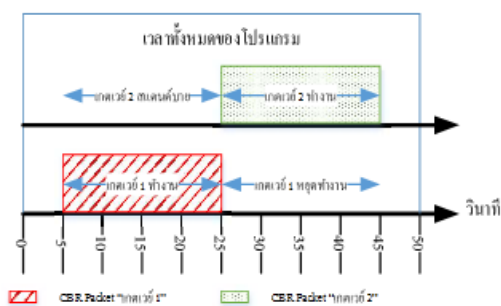
3. วิธีการดำเนินงาน

การจำลองการทำงานของระบบผ่าน โปรแกรม Network Simulator (NS-2) โดยมีการสร้างทอพอโลยีที่ประกอบด้วยโหนดต้นทางจำนวน 1 โหนด และโหนดปลายทางจำนวน 1 และ 2 โหนด โดยมีกำหนดช่วงเวลาการทำงานของเกตเวย์

ดังภาพที่ 5 และทำการส่งข้อมูล CBR ที่มีจำนวนต่างๆ คือ 6000, 7000, 8000, 9000, 10000 packets เพื่อตรวจวัดค่า Throughput และค่า Packets Delivery Ratio (PDR) ของเครือข่าย ในกระบวนการจำลองการทำงานจะแบ่งเป็น 2 ส่วน

ส่วนที่ 1 จำลองการทำงานในกรณีที่มี 1 เกตเวย์ ทำงานเป็นระยะเวลา 50 วินาที โดยวินาทีที่ 5 จะเริ่มส่งข้อมูล CBR เข้าไปในโหนดต้นทาง R0 และกำหนดโหนดปลายทางเป็น R8 เมื่อเวลาผ่านไป 20 วินาที คือวินาทีที่ 25 กำหนดให้โหนดปลายทาง R8 หยุดทำงานจนกระทั่งถึงวินาทีที่ 45 รวมเป็นระยะเวลา 40 วินาที

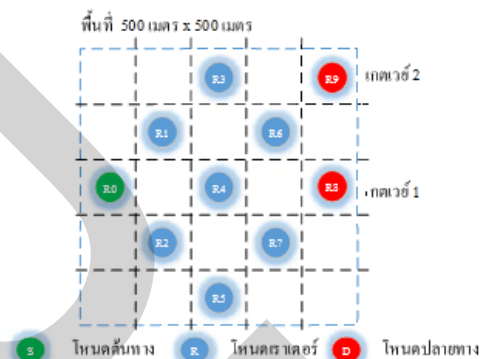
ส่วนที่ 2 จำลองการทำงานในกรณีที่มี 2 เกตเวย์ ทำงานเป็นระยะเวลา 50 วินาที โดยวินาทีที่ 5 จะเริ่มส่งข้อมูล CBR เข้าไปในโหนดต้นทาง R0 และกำหนดโหนดปลายทางเป็น R8 เมื่อเวลาผ่านไป 20 วินาที คือวินาทีที่ 25 กำหนดให้โหนดปลายทาง R8 หยุดทำงาน ในขณะที่โหนดที่เป็นเกตเวย์ 2 จะเริ่มทำงานแทนเมื่อรู้ว่าโหนดที่เป็นเกตเวย์ 1 หยุดทำงาน ซึ่งตัวโหนดที่เป็นเกตเวย์ 2 จะทำงานจนกระทั่งถึงวินาทีที่ 45 เป็นระยะเวลา 20 วินาที รวมเวลาในการจำลองทั้งหมด 40 วินาที ซึ่งแบ่งเป็น โหนดที่เป็นเกตเวย์ 1 ทำงาน 20 วินาที และโหนดที่เป็นเกตเวย์ 2 ทำงาน 20 วินาที



ภาพที่ 6: ช่วงเวลาการทำงานของเกตเวย์



ภาพที่ 7: ทอพอโลยีแบบ 1 เกตเวย์



ภาพที่ 8: ทอพอโลยีแบบ 2 เกตเวย์

การจำลองการทำงานจะกำหนดค่าพารามิเตอร์ในการทดสอบดังนี้

จำนวน โหนด (1 เกตเวย์)	9	โหนด
จำนวน โหนด (2 เกตเวย์)	10	โหนด
ชนิดข้อมูลที่ส่ง	Constant Bit Rate (CBR)	
รูปแบบการแพร่กระจายคลื่น	Two Ray Ground Model	
ขนาดข้อมูลที่ส่ง	100	ไบต์
ขนาดทอพอโลยี	500 เมตร x 500 เมตร	
โพรโทคอลระดับ MAC	IEEE 802.15.4	
โหนดต้นทาง	1	โหนด
โหนดปลายทาง (1 เกตเวย์)	1	โหนด
โหนดปลายทาง (2 เกตเวย์)	1	โหนด
ระยะเวลาในการจำลอง	40	วินาที
โพรโทคอลเส้นทาง	AODV	

หลังจากจำลองการทำงานของ WSN ทั้ง 2 แบบ แล้วจะได้ข้อมูลเป็น trace ไฟล์ ซึ่งบันทึกค่าเหตุการณ์ต่างๆ ไว้ที่เวลาต่างๆ โดยจะนำข้อมูลที่ได้เหล่านี้มาหาค่าเฉลี่ย Throughput และค่า Packets Delivery Ratio (PDR) ของเครือข่ายจากชุดโค้ด (Pseudo Code) ดังนี้

```

1 ALGORITHM THROUGHPUT (SOURCE_CBR, DESTINATION_CBR, TIME)
2 INPUT SOURCE_CBR # CBR_PACKET AT SOURCE NODE
3 INPUT DESTINATION_CBR # CBR_PACKET AT DESTINATION NODE
4 INPUT TIME # SIMULATION TIME
5 OUTPUT THROUGHPUT
6 THROUGHPUT_RATIO = 0
7 THROUGHPUT = ((DESTINATION_CBR/SOURCE_CBR)/TIME)*1000
8 RETURN (THROUGHPUT) #Throughput (Kbps)
9

```

ภาพที่ 9: การหาค่า Throughput

```

10 ALGORITHM PDR (SOURCE_CBR, DESTINATION_CBR)
11 INPUT SOURCE_CBR # CBR_PACKET AT SOURCE NODE
12 INPUT DESTINATION_CBR # CBR_PACKET AT DESTINATION NODE
13 OUTPUT PDR
14 PDR = 0
15 PDR = ((DESTINATION_CBR/SOURCE_CBR)*100
16 RETURN (PDR) # Packet Delivery Ratio (PDR%)
17
18

```

ภาพที่ 10: การหาค่า PDR

ภาพที่ 9 แสดงการหาค่า Throughput ของ WSN แบบที่มี 1 เกตเวย์ โดยการหาค่า CBR packets ขาออกที่โหนด R8 เทียบกับขาเข้าที่โหนด R0 แล้วเทียบกับเวลาที่ใช้ในการจำลองคือ 40 วินาที ส่วน WSN แบบ 2 เกตเวย์ โดยการหาค่า CBR packets ขาออกที่โหนด R8 บวกกับโหนด R9 เทียบกับขาเข้าที่โหนด R0 แล้วเทียบกับเวลาที่ใช้ในการจำลองคือ 40 วินาที

ภาพที่ 10 แสดงการหาค่า PDR ของ WSN แบบ 1 เกตเวย์ โดยการหาค่า CBR packets ขาออกที่โหนด R8 เทียบกับขาเข้าที่โหนด R0 ส่วนของเครือข่ายแบบ 2 เกตเวย์ โดยการหาค่า CBR packets ขาออกที่โหนด R8 บวกกับโหนด R9 เทียบกับขาเข้าที่โหนด R0

4. ผลการทดลอง

4.1 การทดสอบการจำลองส่งข้อมูล CBR ใน WSN แบบ 1 เกตเวย์

ทำการทดสอบโดยการส่งข้อมูล CBR ที่มีอัตราข้อมูลที่แตกต่างกันเข้าไปที่โหนดต้นทาง R0 โดยมีโหนดปลายทาง R8

ตารางที่ 1: ผลการทดลองใน WSN แบบ 1 เกตเวย์

Parameter	Sending Data (Packets)				
	6000	7000	8000	9000	10000
Simulate time (s)	40	40	40	40	40
Data Receive R8 (Packets)	2999	3192	3192	3192	3192
Avg. Throughput R8 (Kbps)	12.50	11.40	9.98	8.87	7.98
Packet Delivery Ratio (R8) (%)	49.98	45.60	39.90	35.47	31.92
Packet Drops (R8) (Packets)	3001	3808	4808	5808	6808

จากตารางที่ 1 ค่า Throughput และค่า PDR จะเห็นได้ว่าเมื่อจำนวน CBR packets มีปริมาณมากขึ้นจะทำให้ Throughput และค่า PDR มีค่าที่น้อยลง เนื่องจากตัวโหนดไม่สามารถรองรับปริมาณข้อมูลที่มีจำนวนมากขึ้น ดังจะเห็นได้จากข้อมูลที่รับได้ในโหนด R8 ขณะที่จำนวน CBR packets ตั้งแต่ 7000 ขึ้นไปมีค่าเท่าเดิม และเมื่อพิจารณาในส่วนของการจำนวน Packets Drops จะเห็นว่ามีความมากขึ้นเรื่อยๆ อันเป็นผลมาจากการที่โหนด R8 หุดทำงาน

4.2 การทดสอบการจำลองส่งข้อมูล CBR ใน WSN แบบ 2 เกตเวย์

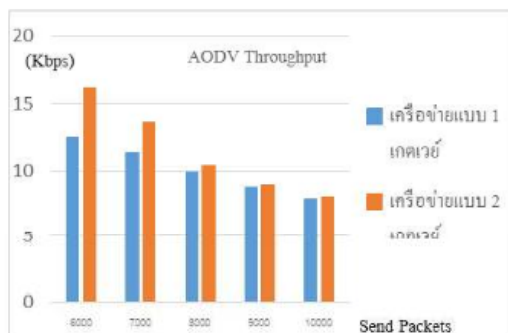
ทำการทดสอบโดยการส่งข้อมูล CBR เข้าไปที่โหนดต้นทาง R0 โดยมีโหนดปลายทาง R8 และ R9 จากตารางที่ 2 ผลที่ได้เมื่อมีการหาค่าปริมาณ Throughput และค่า PDR จะเห็นว่าที่จำนวนปริมาณ CBR packets ต่างๆ ปริมาณค่าของ Throughput และค่า PDR ของเครือข่ายที่มี 2 เกตเวย์ มีค่ามากกว่าเมื่อเทียบกับแบบเครือข่ายที่มี 1 เกตเวย์ เนื่องจากตัวโหนด R9 ที่ทำหน้าที่เป็นเกตเวย์ 2 สามารถช่วยรองรับปริมาณข้อมูลที่มีจำนวนมากขึ้นได้

จากกราฟในภาพที่ 11 และ 12 แสดงการเปรียบเทียบค่า Throughput และค่า PDR ของเครือข่ายแบบ 1 และ 2 เกตเวย์ ซึ่งจะเห็นได้ว่าเครือข่ายแบบ 2 เกตเวย์ สามารถทำได้สูงกว่าแบบเครือข่าย 1 เกตเวย์ ซึ่งปริมาณ Throughput ที่การส่ง Packets 6000 มีค่ามากกว่าที่ส่ง Packets 10000 นั้น เนื่องจากถ้าดูที่ตารางที่ 2 พบว่าจำนวน Packets Drop ของเกตเวย์ 2 มีค่า

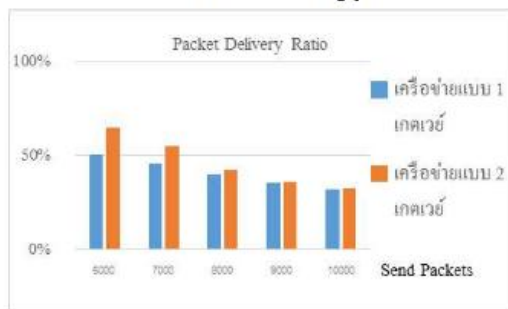
มากขึ้นเมื่อมีการส่งข้อมูลมากขึ้น เนื่องจากเกตเวย์ 2 อยู่ไกลสุด ทำให้ Buffer ที่โหนดระหว่างทางเต็ม ทำให้ Packets ที่ถูกส่งมาใหม่ถูก Drop ไป แต่โดยรวมของการจำลองที่ระดับ CBR packets ต่างๆ พบว่าในเครือข่าย 2 เกตเวย์ สามารถทำได้สูงกว่าแบบเครือข่าย 1 เกตเวย์

ตารางที่ 2: ผลการทดลองใน WSN แบบ 2 เกตเวย์

Parameter	Sending Data (Packets)				
	6000	7000	8000	9000	10000
Simulate time (s)	40	40	40	40	40
Data Received R8 (Packets)	2999	3192	3192	3192	3192
Data Received R9 (Packets)	887	634	159	54	54
Avg. Throughput R8+R9 (Kbps)	16.19	13.66	10.47	9.02	8.12
Packet Delivery Ratio R8+R9 (%)	64.77	54.66	41.89	36.07	32.46
Packet Drops R8+R9 (Packets)	2114	3174	4649	5754	6754



ภาพที่ 11: ปริมาณ Throughput



ภาพที่ 12: สัดส่วนการส่งข้อมูลสำเร็จ PDR

5. ข้อสรุปและเสนอแนะ

จากการทดลองแสดงให้เห็นว่า WSN แบบ 2 เกตเวย์ สามารถให้ปริมาณ Throughput ของเครือข่าย และค่า PDR% ที่ดีกว่าการใช้ WSN แบบ 1 เกตเวย์ โดยการใช้โปรโตคอลเส้นทาง AODV และโหนดภายในเครือข่ายไม่มีการเคลื่อนที่ แต่ในกรณีที่มีความหนาแน่นของข้อมูลในเครือข่ายจำนวนมาก ส่วนต่างของค่า Throughput ของเครือข่าย และค่า PDR นั้นจะมีค่าไม่สูงมาก โดยพบว่าค่า Throughput มีค่าสูงขึ้นเฉลี่ย 1.35 Kbps และค่า Packet Delivery Ratio (PDR) มีค่าสูงขึ้นเฉลี่ย 5.39% เมื่อเทียบกับ WSN แบบเดิมที่มี 1 เกตเวย์

ข้อเสนอแนะควรทำการปรับปรุงตัวโปรโตคอลเส้นทาง Ad-hoc On-Demand Distance Vector (AODV) ให้สามารถหาตัวเกตเวย์ใหม่ในเครือข่ายได้เองโดยอัตโนมัติ และแบ่งภาระการทำงานของตัวโหนดที่เป็นเกตเวย์ เมื่อเกตเวย์ปัจจุบันไม่สามารถส่งหรือรับข้อมูลได้

เอกสารอ้างอิง

- [1] อนุรักษ์ เขษม และ ชัยพร เขมะภาคพันธ์, "การประยุกต์เครือข่ายตรวจจับสัญญาณไร้สายเพื่อประมวลผลข้อมูลความเร็วสูงสำหรับช่วยตัดสินใจในการติดตั้งกังหันลมผลิตกระแสไฟฟ้า, การประชุมการวิจัยระดับบัณฑิตศึกษาแห่งชาติ มหาวิทยาลัยเชียงใหม่ ครั้งที่ 2, 26 พฤศจิกายน 2553, หน้า 620-629.
- [2] นภาพร พิมพ์ และ ประโยชน์ คำสวัสดิ์, "การบีบอัดข้อมูลในเครือข่ายเซนเซอร์ไร้สาย" *ECTI-CARD 2013* หน้า 173-178.
- [3] Luca De Nardis and Maria-Gabriella, "Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks" *WPNC 2007, Handover, Germany, March 2007*.
- [4] P. Manoyut and W. Suntiamorntut, "Modified Local Repair in AODV for LoWPANs" (*IMCOM 2012*), Kuala Lumpur, Malaysia, February 2012.
- [5] Xuan He and Yuanchen Ma, "Efficient Multiple Gateway System for WSN Management in BEMS" *Networked Sensing Systems (INSS)*, Belgium, June 2012.
- [6] Zahia Bidai and Hafid Haffaf "Node disjoint multi-path routing for zigbee cluster-tree Wireless Sensor Networks", *Multimedia Computing and Systems (ICMCS)*, Ouarzazate, Morocco, April 2011.

ประวัติผู้เขียน

ชื่อ-นามสกุล

นายชนะเดช แหม่มคลี่

ประวัติการศึกษา

ปีการศึกษา 2545 สำเร็จการศึกษาระดับปริญญาตรี
สาขาวิศวกรรมระบบวัดคุม คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีมหานคร

ตำแหน่งและสถานที่ทำงานปัจจุบัน

ผู้จัดการฝ่ายขาย
บริษัท แอมต้า จำกัด