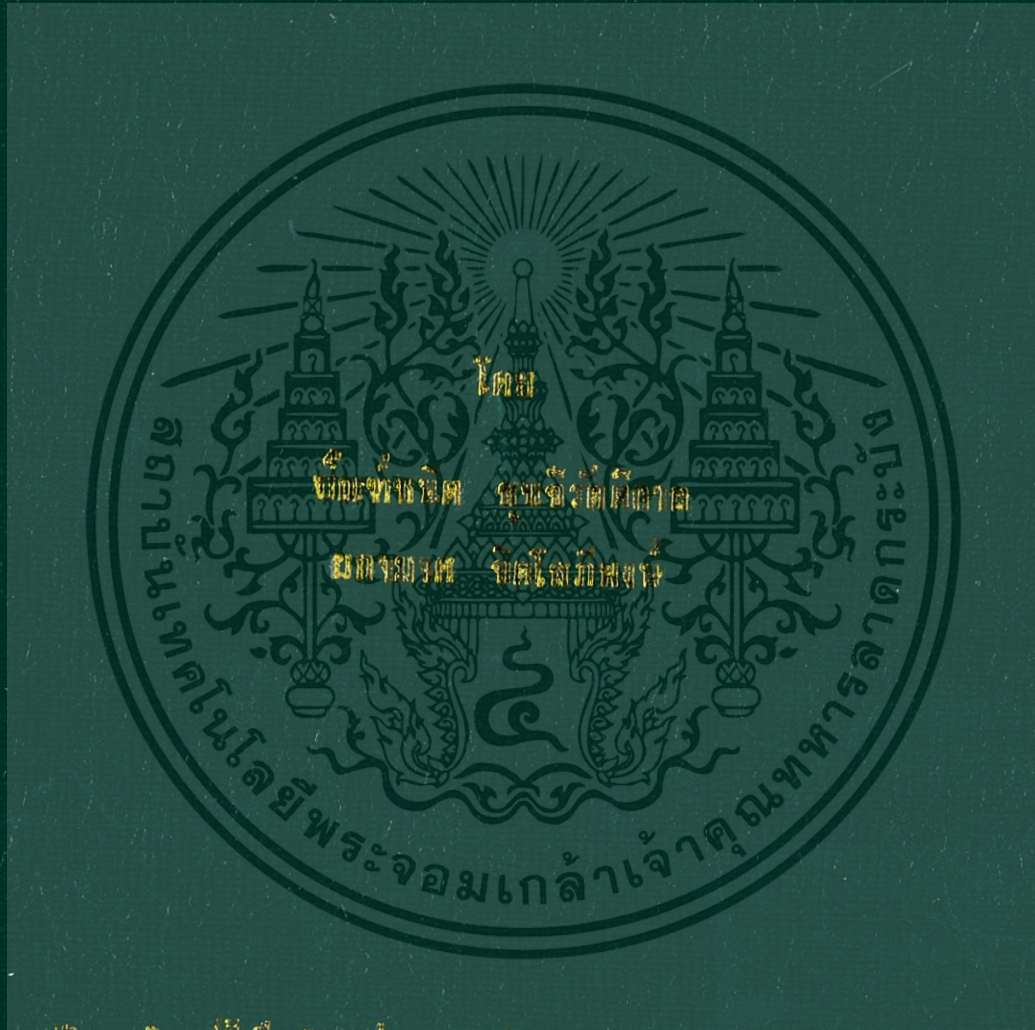


การศึกษาด้านประสิทธิภาพการดำเนินงานในระบบตลาดการเงิน
A STUDY ON MARKET PERFORMANCE IN MULTIPATH ENVIRONMENT



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คณะวิศวกรรมศาสตร์และเทคโนโลยีสารสนเทศ

สาขาวิชาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

ภาคเรียนที่ ๒ ปีการศึกษา ๒๕๕๗

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใดโดยไม่ได้รับอนุญาต
สงวนลิขสิทธิ์ไว้ก่อน. ถ้ามีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สถาบันจะดำเนินการฟ้องดำเนินคดี

การศึกษาประสิทธิภาพบนเครือข่ายแมนเนทบนสภาพแวดล้อมแบบหลายทาง

A STUDY ON MANET PERFORMANCE IN MULTIPATH ENVIRONMENT



โดย

ปัทม์ชนิต สุขจิรัตติกาล

PANCHANIT SUKJIRATTIKAL

ผกามาศ จิตโสภิพงษ์

PHAKAMAS JITSOPEEPONG

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต

6001681 95
b. 42812201
i.

เลขหมู่.....
เลขทะเบียน...144529
วัน,เดือน,ปี...2.5...พ.ค. 2559

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาคเรียนที่ 2 ปีการศึกษา 2557 ขออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาประสิทธิภาพบนเครือข่ายแมนเนทบนสภาพแวดล้อมแบบหลายทาง

A STUDY ON MANET PERFORMANCE IN MULTIPATH
ENVIRONMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาคเรียนที่ 2 ปีการศึกษา 2557 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**A STUDY ON MANET PERFORMANCE IN MULTIPATH
ENVIRONMENT**



**PANCHANIT SUKJIRATTIKAL
PHAKAMAS JITSOPEEPONG**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ 2/2014 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2015

FACULTY OF INFORMATION TECHNOLOGY

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โยชนด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2557

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาประสิทธิภาพบนเครือข่ายแมนเนทบนสภาพแวดล้อมแบบหลาย
ทาง

A STUDY ON MANET PERFORMANCE IN MULTIPATH
ENVIRONMENT

ผู้จัดทำ

1. นางสาวปิ่นทชณิต สุขจิรัตติกาล รหัสนักศึกษา 54070053
2. นางสาวผกามาต จิตโสภีพงษ์ รหัสนักศึกษา 54070054

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	การศึกษาประสิทธิภาพบนเครือข่ายแมนเนทบนสภาพแวดล้อมแบบหลายทาง
นักศึกษา	นางสาวปณัฏชานิต สุขจิรัตติกาล รหัสนักศึกษา 54070053 นางสาวพกามาศ จิตโสภีพงษ์ รหัสนักศึกษา 54070054
ปริญญา	วิทยาศาสตร์บัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต

บทคัดย่อ

ในการหาเส้นทางบนเครือข่ายเฉพาะกิจ การกระจายภาระงานมีผลกระทบมากต่อประสิทธิภาพเครือข่าย การหาเส้นทางที่ดีควรมีการกระจายภาระงานให้โหนดอย่างทั่วถึง มิฉะนั้นจะทำให้โหนดที่ถูกใช้เป็นทางผ่านการรับส่งข้อมูลอยู่ตลอด เกิดความคับคั่งจากปริมาณข้อมูลผ่านมากเกินไป ในขณะที่โหนดที่ไม่ค่อยถูกใช้ มีปริมาณข้อมูลผ่านน้อย ความไม่สมดุลในการกระจายภาระงานนี้ทำให้เกิดปัญหาความล่าช้าในการส่งข้อมูล การส่งข้อมูลล้มเหลว และเกิดโอเวอร์เฮดเพิ่มสูงขึ้นจากการต้องส่งข้อมูลที่ล้มเหลวซ้ำ บทความนี้ศึกษาแนวทางการแก้ปัญหาความไม่สมดุลดังกล่าว งานวิจัยก่อนหน้าเสนอแนวทางแก้ปัญหาโดยให้แต่ละโหนดจะพิจารณาความสามารถตัวเองในการรับภาระงานเพิ่ม ก่อนตัดสินใจว่าจะยอมเป็นทางผ่านในการรับส่งข้อมูลให้โหนดต้นทางที่ร้องขอมาหรือไม่ เพื่อให้เกิดความสมดุลในการกระจายภาระงาน แต่ในทางตรงกันข้ามโหนดต้นทางอาจต้องใช้เวลาและจำนวนครั้งในการร้องขอเส้นทางมากขึ้น ทำให้เกิดความล่าช้า ผู้วิจัยจึงเสนอการกระจายภาระงานอย่างสมดุลที่มีความยืดหยุ่นขึ้น เพื่อแก้ไขปัญหาที่กล่าวมานี้ และผลการประเมินจากการจำลองเหตุการณ์แสดงให้เห็นว่าเวลาที่ใช้ในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลลดลง

Project Title	A Study on MANET Performance in Multipath Environment	
Student	Miss Panchanit Sukjirattikal	Student ID 54070053
	Miss Phakamas Jitsopeepong	Student ID 54070054
Degree	Bachelor of Science	
Program	Information Technology	
Academic Year	2014	
Advisor	Asst.Prof.Dr. Sumet Prabhavat	

ABSTRACT

In Mobile Ad Hoc Network, Load-balancing is a major factor of network performance. An effective route discovery is needed to balance the traffic; otherwise, some nodes will incur congestion problem. Imbalance problem causes large delay, low packet delivery ratio, and large routing overhead. This paper studies about load-balancing techniques and develops a flexible multipath load-balancing algorithm for multipath routing protocol. The result shows that the proposed algorithm reduces delay and overhead and increases packet delivery ratio.

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ที่ปรึกษาโครงการ ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต ด้วยความกรุณาจากอาจารย์ ทำให้โครงการการศึกษาประสิทธิภาพบนเครือข่ายเฉพาะกิจบนสภาพแวดล้อมแบบหลายทางบรรลุผลสำเร็จไปได้ด้วยดี อาจารย์ได้คอยให้คำปรึกษา และเอาใจใส่ในการพัฒนาโครงการนี้มาโดยตลอด

ขอขอบคุณ บิดา มารดา และ คณาจารย์ทุกท่านที่ได้อบรมสั่งสอนความรู้ที่มีประโยชน์ต่อการประยุกต์ใช้ในการทำงานและแก้ไขปัญหาต่างๆ รวมไปถึงเพื่อนๆ พี่ๆ และน้องๆ ที่คอยเป็นกำลังใจและช่วยสนับสนุน หากปราศจากบุคคลเหล่านี้ โครงการการศึกษาประสิทธิภาพบนเครือข่ายเฉพาะกิจบนสภาพแวดล้อมแบบหลายทาง ก็มีอาจบรรลุผลสำเร็จไปได้ด้วยดี

ปิ่นกัญชนิศ สุขจิรัตติกาล

ศกามาศ จิตโสภีพงษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VI
สารบัญตาราง.....	VII
คำนิยามศัพท์.....	VIII

บทที่

1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 วิธีการดำเนินงาน.....	2
1.4 ขอบเขตของงาน.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2 ทฤษฎีและงานที่เกี่ยวข้อง.....	4
2.1 เครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของ โมบายล์โหนด หรือ เฉพาะกิจ.....	4
2.1.1 Proactive Routing Protocol หรือ Table Driven Protocol.....	5
2.1.2 Reactive Routing Protocol หรือ On-demand Protocol.....	5
2.2 โมเดลการค้นหาเส้นทางของเฉพาะกิจแบบทางเดียว.....	5
2.2.1 Ad-hoc On demand Distance Vector (AODV).....	5
2.3 โมเดลการค้นหาเส้นทางของเฉพาะกิจในสภาพแวดล้อมแบบหลายทาง.....	7
2.3.1 Ad Hoc On-demand Multipath Distance Vector Routing (AOMDV).....	7
2.3.2 A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network.....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ IV ศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3 งานที่นำเสนอ	12
3.1 งานที่นำเสนอ.....	12
3.1.2 หลักการทำงานของอัลกอริทึมที่นำเสนอ	13
3.1.3 สมการที่ใช้คำนวณความน่าจะเป็นในการละทิ้ง RREQ.....	14
4 การทดลองและประเมินผล	15
4.1 ผลการทดลอง	17
5 สรุปผล	21
บรรณานุกรม	23
ภาคผนวก	24
ประวัติผู้เขียน	30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 แสดงการเปรียบเทียบระหว่างแบบโครงสร้างพื้นฐาน กับ แบบเฉพาะกิจ.....	4
2.2 แสดงการทำงานของโปรโตคอล WAL	10
3.1 แสดงการทำงานของโปรโตคอลที่นำเสนอ	13
4.1 แสดงการเปรียบเทียบเวลาที่ใช้ในการสร้างเส้นทาง	17
4.2 แสดงการเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูล	18
4.3 แสดงการเปรียบเทียบโอเวอร์เฮดต่อหนึ่งแพคเกจข้อมูล.....	20
4.4 แสดงการเปรียบเทียบอัตราการส่งแพคเกจสำเร็จ	20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อVLศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 แสดงข้อมูล โครงสร้างตารางเส้นทางของโพรโทคอล AODV.....	9
2.2 แสดงข้อมูล โครงสร้างตารางเส้นทางของโพรโทคอล AOMDV.....	9
4.1 แสดงข้อมูล สิ่งแวดล้อมของการจำลองเครือข่าย.....	15



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ VII ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนิยามศัพท์

1. Mobile Ad Hoc Network (MANET) คือ เครือข่ายที่ประกอบด้วยกลุ่มของโหนดเคลื่อนที่ไร้สายที่สามารถเคลื่อนไหวได้อย่างอิสระ สร้างเครือข่ายชั่วคราวโดยที่ไม่จำเป็นต้องอาศัยโครงสร้างพื้นฐานและไม่จำเป็นต้องมีศูนย์กลางการควบคุม
2. Route Request (RREQ) คือ แพคเกจที่ใช้ในกระบวนการค้นหาเส้นทาง (route discovery) เพื่อทำการร้องขอค้นหาเส้นทาง
3. Route Reply (RREP) คือ แพคเกจที่โหนดปลายทางหรือโหนดระหว่างทางตอบเส้นทางกลับไปยังโหนดต้นทาง
4. Route Error (RRER) คือ แพคเกจที่ใช้ในกระบวนการรักษาเส้นทาง (route recovery) เพื่อทำการแจ้งโหนดต้นทางว่าเส้นทางเสียหาย
5. Route Discovery Error Message (RDER) คือ แพคเกจที่โหนดระหว่างทางส่งไปหาโหนดที่ส่ง RREP มาให้ เมื่อหาเส้นทางที่ส่ง RREP ต่อไม่เจอ
6. Route Request Confirmation Message (RRCM) คือ แพคเกจที่โหนดต้นทางส่งกลับไปยังโหนดปลายทางเมื่อได้รับ RREP
7. Quality of Service (QoS) คือ คุณภาพของการบริการ เป็นตัวกำหนดคุณสมบัติของประสิทธิภาพในการติดต่อสื่อสาร
8. Time-to-Live (TTL) คือ เวลาหมดอายุของแพคเกจ เป็นการกำหนดจำนวนฮอปสูงสุดที่แพคเกจสามารถถูกส่งต่อได้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีการติดต่อสื่อสารได้มีความก้าวหน้าและมีบทบาทสำคัญเป็นอย่างมาก การใช้งานอินเทอร์เน็ตสามารถใช้ได้ไม่จำกัดเฉพาะคอมพิวเตอร์ รูปแบบการใช้งานในปัจจุบันจึงเปลี่ยนไป เกิดการสื่อสารแบบไร้สายขึ้น ทำให้ผู้ใช้สามารถติดต่อสื่อสารกันได้แบบไม่จำกัดสถานที่และไม่จำกัดเวลา สะดวกและรวดเร็วยิ่งขึ้น ทำให้อุปกรณ์และเทคโนโลยีการติดต่อสื่อสารเหล่านี้เป็นส่วนหนึ่งในชีวิตประจำวัน และมีบทบาทในธุรกิจเป็นอย่างมาก

เทคโนโลยีการสื่อสารของระบบเครือข่าย ปัจจุบันมีทั้งระบบเครือข่ายแบบโครงสร้างพื้นฐาน ซึ่งจำเป็นต้องมีอุปกรณ์ตัวกลางทำหน้าที่รับ-ส่งข้อมูล เมื่อเกิดเหตุไม่คาดฝันทำให้ระบบโครงสร้างพื้นฐานเกิดขัดข้อง ใช้งานไม่ได้ในสถานะจำเป็น จึงได้เกิดเป็นเครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของโมบายล์โหนด หรือ Mobile Ad-hoc Network (MANETs) เป็นเครือข่ายที่ประกอบด้วยกลุ่มของโหนดเคลื่อนที่ไร้สายที่สามารถเคลื่อนไหวได้อย่างอิสระ สร้างเครือข่ายชั่วคราวโดยที่ไม่จำเป็นต้องอาศัยโครงสร้างพื้นฐานและไม่จำเป็นต้องมีศูนย์กลางการควบคุม ด้วยความยืดหยุ่นนี้ จึงทำให้สามารถสร้างเครือข่ายที่ใดและเมื่อใดก็ได้ トラาบใดที่มีผู้ใช้อย่างน้อย 2 คน ต้องการติดต่อสื่อสารกัน แต่ละโหนดจะทำหน้าที่เป็นเราท์เตอร์และโอสต์ ส่งผ่านข้อมูลแพคเกจไปยังโหนดอื่นต่อกันไป ดังนั้นการมีอยู่ของแต่ละโหนดจึงมีความสำคัญเท่าๆกัน จึงต้องคำนึงถึงเรื่องการใช้ทรัพยากรที่มีอยู่อย่างจำกัด เช่น พลังงาน หน่วยประมวลผล แรม

เนื่องจากโหนดมีการเคลื่อนย้ายที่ได้ จึงทำให้เกิดการเปลี่ยนแปลงโทโพโลยี ขึ้นได้ตลอดเวลา ส่งผลให้เส้นทางในการรับส่งข้อมูลขาดได้ ความน่าเชื่อถือของการสื่อสารลดลง เพื่อแก้ปัญหาเช่นนี้ จึงมีความสนใจที่จะพัฒนาโพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทาง

โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทาง จะช่วยพัฒนาประสิทธิภาพและความน่าเชื่อถือ เพราะว่าเมื่อเส้นทางเกิดขาด ก็จะมีเส้นทางสำรองให้ใช้แทนได้ทันที โดยไม่จำเป็นต้องทำกระบวนการค้นหาเส้นทางใหม่ ทำให้ช่วยลดเวลาที่ใช้ในการส่งข้อมูลตรงส่วนนั้นได้นอกจากนั้นโพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทางยังมีการใช้ทรัพยากรคุ่มค่ามากกว่าโพรโทคอลการค้นหาเส้นทางแบบเส้นทางเดียว เนื่องจากในการค้นหาเส้นทางแบบเส้นทางเดียวนั้น ยังมีอีกหลายเส้นทางที่สามารถเลือกใช้ส่งข้อมูลไปยังปลายทางได้แต่ไม่ได้ถูกนำมาใช้

นอกจากนั้น โพรโทคอลปัจจุบันยังขาดประสิทธิภาพในการกระจายภาระงาน ทำให้คุณภาพของการบริการลดลง โดยเฉพาะในเครือข่ายที่มีปริมาณทราฟฟิกมาก เนื่องจากทราฟฟิกจะ

เอกสำรฉบับนี้สงวนลิขสิทธิ์ไว้เพื่อประโยชน์ของนักศึกษาและผู้สนใจทั่วไปในการศึกษาหาความรู้โดยไม่มีการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอใจหลายอย่าง เช่น เวลาที่ใช้ในการส่งข้อมูลสูง อัตราการส่งข้อมูลสำเร็จต่ำ โอเวอร์เฮดสูง และ ยังใช้ทรัพยากรของเครือข่าย เช่น แบนวิธ พลังงานอย่างไม่มีประสิทธิภาพ

ทางผู้จัดทำจึงได้ทำการศึกษา ค้นคว้า และนำเสนอ โพรโทคอลที่มีการใช้เทคนิคกระจายภาระงานเพื่อปรับปรุงประสิทธิภาพไม่ให้โหนดนั้นๆ มีการใช้เส้นทางที่คับคั่ง โดยจะมีการกำหนดว่าจะยอมให้มีการสร้างเส้นทางผ่านโหนดนั้นๆ ไปยังโหนดปลายทางหรือไม่ มีการนำเสนอแนวคิดที่จะใช้หลักของความน่าจะเป็นมาเป็นตัวชี้วัดในการจำกัดการสร้างเส้นทางของแต่ละโหนด ซึ่ง โพรโทคอลที่ทางผู้จัดทำใช้ในการศึกษาและทดลองคือ โพรโทคอล AOMDV เนื่องจากเป็นโพรโทคอลที่ถูกออกแบบมาเพื่อลดโอเวอร์เฮดโดยเฉพาะ [1] ดังนั้นจึงมีโอเวอร์เฮดน้อยกว่าโพรโทคอลในการค้นหาเส้นทางอื่นๆ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ปริญญานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาประสิทธิภาพและพัฒนาอัลกอริทึม ในการติดต่อสื่อสารของอุปกรณ์ในเครือข่ายเฉพาะกิจ เพื่อเพิ่มประสิทธิภาพให้กับเครือข่ายเฉพาะกิจในสภาพแวดล้อมแบบหลายทาง ในด้านการกระจายภาระงาน การลดเวลาที่ใช้ในการส่งข้อมูล การเพิ่มอัตราการส่งข้อมูลสำเร็จ และการลดโอเวอร์เฮด

1.3 วิธีการดำเนินงาน

โครงการการศึกษาประสิทธิภาพของเครือข่ายเฉพาะกิจบนสภาพแวดล้อมแบบหลายเส้นทางนั้นจะทำการศึกษาและพัฒนาอัลกอริทึมแบบสภาพแวดล้อมแบบหลายทางในการค้นหาและคัดเลือกเส้นทางในการสื่อสาร โดยมีวิธีการดำเนินงานวิจัยดังนี้

1. ศึกษางานวิจัย โพรโทคอลเดิมที่มีอยู่ก่อนหน้า ทั้งเนื้อหาในเรื่องของสภาพแวดล้อมแบบเส้นทางเดียวและหลายเส้นทาง รวมไปถึงทฤษฎีอื่นๆที่เกี่ยวข้อง
2. ศึกษาโปรแกรมที่ใช้ในการจำลองเครือข่ายเพื่อหาโปรแกรมที่เหมาะสมในการใช้งาน
3. ทดลองจำลองโพรโทคอลเดิมที่มีอยู่เพื่อศึกษาการใช้โปรแกรมและเพื่อดูประสิทธิภาพการทำงานของโพรโทคอลนั้นๆ
4. ศึกษาหาวิธีปรับปรุงประสิทธิภาพการทำงานของโพรโทคอล
5. ทดลองจำลองโพรโทคอลที่นำเสนอ ดูประสิทธิภาพเพื่อนำไปใช้ในการศึกษา วิเคราะห์ผลการทดลอง และพัฒนาต่อยอด

1.4 ขอบเขตของงาน

ศึกษาการทำงานของโพรโทคอล Ad Hoc On-demand Distance Vector routing (AODV) กับ Ad Hoc On-demand Multipath Distance Vector routing (AOMDV) และนำเทคนิคการกระจายเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาระงานมาปรับใช้งานร่วมกับโปรโตคอล AOMDV เพื่อหาวิธีการคัดเลือกเส้นทางแบบหลายเส้นทางให้มีประสิทธิภาพมากที่สุด โดยทำการศึกษาค้นคว้าจากงานวิจัยก่อนหน้า และนำมาจำลองเพื่อศึกษาประสิทธิภาพในปัจจัยด้านต่างๆ รวมไปถึงคิดวิธีการที่สามารถนำมาปรับใช้กับโปรโตคอลพื้นฐานเพื่อเพิ่มประสิทธิภาพ แล้วเปรียบเทียบวิเคราะห์ผลการทดลอง

การสื่อสารแบบเฉพาะกิจที่ทำการค้นคว้า จะเป็นเรื่องของสภาพแวดล้อมหลายเส้นทาง โดยจะศึกษา (1) ค้นหาเส้นทางแบบหลายเส้นทาง (2) วิธีในการเลือกเส้นทาง (3) กระจายการภาระระหว่างเส้นทาง เพื่อให้มีประสิทธิภาพกว่าเดิม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้และความเข้าใจในกระบวนการทำงานของโปรโตคอลในสภาพแวดล้อมแบบหลายเส้นทางมากยิ่งขึ้น
2. ได้เรียนรู้กระบวนการทำงานในการพัฒนาโปรโตคอลให้มีประสิทธิภาพมากขึ้น
3. ได้เรียนรู้วิธีการใช้โปรแกรมจำลองเครือข่ายต่างๆ เพื่อใช้ในการจำลองโปรโตคอลที่พัฒนาขึ้น
4. ได้ฝึกทักษะทางกระบวนการคิด วิเคราะห์ปัญหา และหาสาเหตุ เพื่อใช้วิเคราะห์ผลการทดลอง และนำมาปรับปรุงแก้ไข

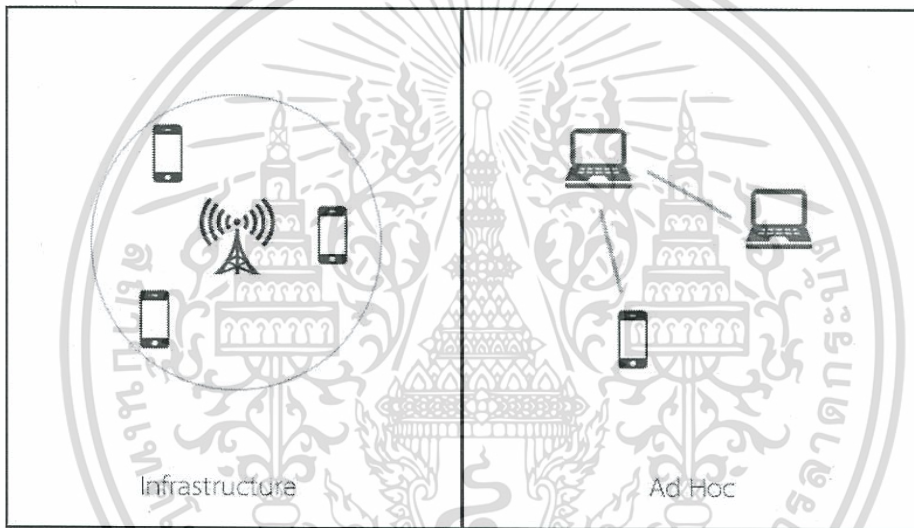
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยต่างๆที่เกี่ยวข้องที่เป็นพื้นฐานในการพัฒนาโพรโทคอลในการค้นหาเส้นทางแบบหลายเส้นทาง ซึ่งจะกล่าวถึง เครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของโมบายล์โหนด หรือ Mobile Ad-hoc Network (MANETs) และโพรโทคอลในการค้นหาเส้นทางประเภทต่างๆ รวมไปถึงงานวิจัยที่เกี่ยวข้อง

2.1 เครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของโมบายล์โหนด หรือ เฉพาะกิจ



รูปที่ 2.1 แสดงการเปรียบเทียบระหว่างแบบโครงสร้างพื้นฐาน กับ แบบเฉพาะกิจ

เครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของโมบายล์โหนด หรือ Mobile Ad-hoc Network (MANETs) สามารถสร้างเครือข่ายชั่วคราวโดยที่ไม่ต้องพึ่งโครงสร้างพื้นฐานและไม่จำเป็นต้องมีศูนย์กลางควบคุม ซึ่งทุกโหนดสามารถเคลื่อนที่ไปได้อย่างอิสระและสามารถทำการตั้งค่าตัวเองได้ แต่ละโหนดจะทำหน้าที่เป็นเราท์เตอร์และโฮสต์แม้ว่าโทโพโลยีอาจมีการเปลี่ยนแปลงอย่างรวดเร็ว ซึ่งรูปแบบการติดต่อสื่อสารจะถูกกำหนดโดยโพรโทคอล ซึ่งแต่ละโพรโทคอลจะมีคุณสมบัติที่แตกต่างกันออกไป โดยในระบบการสื่อสารแบบเฉพาะกิจนั้น สามารถแบ่งประเภทของโพรโทคอลได้เป็น 2 ประเภทหลักๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 Proactive Routing Protocol หรือ Table Driven Protocol

โพรโทคอลในการค้นหาเส้นทางแบบ proactive protocol นั้นจะทำการค้นหาเส้นทางระหว่างโหนดทุกโหนดที่สามารถเชื่อมต่อกันได้ในเครือข่ายแบบเป็นระยะ ๆ อีกทั้งยังมีการปรับปรุงบำรุงรักษาเส้นทางอย่างต่อเนื่อง

2.1.2 Reactive Routing Protocol หรือ On-demand Protocol

โพรโทคอลในการค้นหาเส้นทางแบบ reactive protocols จะทำการค้นหาเส้นทางในการส่งข้อมูลจากต้นทางไปยังปลายทางเมื่อต้นทางต้องการส่งข้อมูลเท่านั้น หรือกล่าวได้ว่าโพรโทคอลในการค้นหาเส้นทางในรูปแบบนี้จะทำการเปลี่ยนแปลงหรือค้นหาเส้นทางเมื่อต้องการ

2.2 โมเดลการค้นหาเส้นทางของเฉพาะกิจแบบทางเดียว

2.2.1 Ad-hoc On demand Distance Vector (AODV)

โพรโทคอล Ad-hoc On demand Distance Vector หรือ AODV [1] จะไม่เก็บเส้นทางจากแต่ละโหนดมันจะทำการค้นหาเส้นทางเมื่อต้องการจะใช้เท่านั้น โดยทั่วไปแล้ว โพรโทคอล AODV ไม่จำเป็นต้องมีโครงสร้างพื้นฐานและตัวควบคุม จึงทำให้สามารถจัดการและตั้งค่าตัวเองได้

โพรโทคอล AODV ประกอบด้วยกลไกพื้นฐาน 2 อย่าง คือ (1) กลไกการค้นหาเส้นทาง (route discovery) และ (2) กลไกการรักษาเส้นทาง (route maintenance) โดยที่โหนดต้นทางจะค้นหาเส้นทางไปยังปลายทาง เมื่อโหนดต้นทางต้องการส่งข้อมูล

กลไกการค้นหาเส้นทาง

เมื่อต้องการจะส่งข้อมูลไปยังโหนดปลายทาง โหนดต้นทางจะเข้าไปเช็คในตารางเส้นทาง (routing table) ก่อนเพื่อดูว่ามีเส้นทางไปยังปลายทางแล้วหรือไม่ ถ้ามีก็จะทำการส่งข้อมูลไปยังโหนดถัดไปตามตารางเส้นทางจนถึงปลายทาง แต่หากไม่มีเส้นทาง จะใช้กระบวนการค้นหาเส้นทาง ซึ่งโพรโทคอล AODV จะใช้ route request (RREQ) และ route reply (RREP) โดยโหนดต้นทางจะทำการสร้างแพคเกจ RREQ ซึ่งเก็บข้อมูลดังนี้ source address source sequence number destination address destination sequence number และ broadcast ID ซึ่ง broadcast ID จะเพิ่มขึ้นทุกครั้งที่โหนดต้นทางสร้าง RREQ ดังนั้น source address และ broadcast ID จะแตกต่างกันไปในแต่ละแพคเกจ ใช้ระบุความแตกต่างของ RREQ แต่ละครั้ง

เริ่มต้นกระบวนการค้นหาเส้นทางโดยโหนดต้นทางจะ broadcast แคมแพคเกจ RREQ ไปยังโหนดเพื่อนบ้าน เมื่อโหนดระหว่างทางได้รับ RREQ ก่อนที่จะส่งต่อจะมีการตั้ง reverse route entry (เส้นทางย้อนกลับมายังต้นทาง) ไว้ในตารางเส้นทาง และเพิ่มค่า hop count ถ้าหากได้ RREQ เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือมี source address และ broadcast ID เหมือนกับของ RREQ ก่อนหน้า ก็จะทำการละทิ้ง RREQ นั้นไป

การสร้าง reverse path

จะใช้ source sequence number ในการบอกความใหม่ของข้อมูลเส้นทาง reverse route ไปยังโหนดต้นทาง และใช้ destination sequence number ในการบอกความสดใหม่ของข้อมูลไปยังปลายทาง ในขณะที่ RREQ ถูกส่งต่อผ่านโหนดระหว่างทาง โหนดจะทำการตั้ง reverse path กลับไปยังโหนดต้นทางเสมอ โดยการบันทึกแอดเดรสของเพื่อนบ้านที่ส่ง RREQ มาให้ แล้วทำการจับเวลา ถ้าหากครบระยะเวลาที่กำหนดแล้ว ไม่มีการใช้ reverse route entry นั้นเลย ก็จะทำให้การลบเส้นทางนั้นออกจากตารางเส้นทาง

การสร้าง forward path

เมื่อโหนดระหว่างทางได้รับ RREQ จะตรวจสอบว่ามีเส้นทางไปยังปลายทางหรือไม่ จะตรวจสอบว่าเป็นเส้นทางที่สดใหม่หรือไม่โดยการเปรียบเทียบ destination sequence number ภายใน RREQ กับที่อยู่ภายใน routing table ของตน โดยจะใช้เส้นทางที่มี destination sequence number มากกว่า ถ้าภายใน RREQ มีมากกว่า จะทำการบรอดแคสต์ RREQ ต่อ แต่ถ้าหากภายใน routing table มากกว่าหรือเท่ากับที่อยู่ใน RREQ โหนดจะทำการตอบกลับ RREQ แบบยูนิแคสต์ไปยังเพื่อนบ้านที่ส่ง RREQ มาให้ ภายใน RREP ประกอบด้วย source address destination address destination sequence number และ lifetime

ในขณะที่ RREP ถูกส่งกลับไปยังโหนดต้นทาง แต่ละโหนดระหว่างทางจะตั้ง forward path เป็นโหนดก่อนหน้าที่ส่ง RREP มาให้ แล้วอัปเดตเวลาหมดเวลาของเส้นทางและบันทึก destination sequence number สำหรับปลายทางนั้น ถ้าโหนดระหว่างทางได้รับหลาย RREP จะส่งต่อเฉพาะ RREP ที่มีค่า destination sequence number มากกว่าใน RREP ก่อนหน้า หรือถ้า destination sequence number มีขนาดเท่ากันจะเลือกส่งเฉพาะที่มีจำนวนฮอปน้อยกว่า เป็นการป้องกันการเกิดลูปได้

กลไกการรักษาเส้นทาง

ถ้าโหนดต้นทางมีการเคลื่อนที่ออกจนทำให้ไม่มีเส้นทางในขณะที่ต้องการใช้เส้นทางอยู่ จะเรียกใช้กระบวนการค้นหาเส้นทางใหม่อีกครั้ง แต่ถ้าหากโหนดปลายทางหรือโหนดระหว่างทางมีการเคลื่อนที่จากเส้นทาง โหนดจะสร้าง RREP แบบพิเศษ เป็นข้อความส่งกลับไปยังโหนดต้นทางเพื่อแจ้งว่าเส้นทางเสียหาย อาจใช้ข้อความ HELLO ส่งเป็นระยะๆ เพื่อตรวจสอบว่าเส้นทางเสียหายหรือไม่

เมื่อส่งไปยังฮอปถัดไปไม่ได้ โหนดจะส่ง RREP ที่มี sequence number มากกว่าเดิม โดยมี hop count คือ ∞ ไปยังโหนดต้นน้ำทั้งหมด จนกระทั่งถึงโหนดต้นทาง โหนดต้นทางก็จะเริ่มกระบวนการค้นหาเส้นทางใหม่

โพรโทคอล AODV มีการใช้เทคนิค ring search คือ เมื่อโหนดต้นทางส่งแพ็คเกจ RREQ ไปยังโหนดเพื่อนบ้าน สำหรับเครือข่ายขนาดใหญ่จำเป็นที่จะต้องมีการควบคุมการบรอดแคสต์ RREQ เพื่อที่จะลดโอเวอร์เฮดจากการบรอดแคสต์ค้นหาเส้นทางภายในเครือข่าย โหนดต้นทางจะใช้เทคนิค ring search ซึ่งเทคนิคนี้จะใช้การตั้ง Time-to-Live (TTL) หรือก็คือเวลาหมดอายุของแพ็คเกจ ถ้าไม่มีการตอบรับภายในระยะเวลาที่กำหนด ก็จะทำการ บรอดแคสต์ RREQ ออกไปอีกครั้งแต่จะเพิ่มค่า TTL ขึ้นด้วย เพื่อให้ RREQ สามารถกระจายไปในเครือข่ายได้ไกลขึ้น กระบวนการเพิ่ม TTL จะเพิ่มจนกระทั่งมีค่าถึงค่าที่กำหนดไว้ (threshold) หลังจากนั้นจะแพร่กระจายจนทั่วเครือข่าย

2.3 โมเดลการค้นหาเส้นทางของเฉพาะกิจในสภาพแวดล้อมแบบหลายทาง

เนื่องจากโมเดลการค้นหาเส้นทางของเฉพาะกิจแบบทางเดียวนั้น ทำการค้นหาเส้นทางมาใช้ในการสื่อสารแค่เส้นทางเดียว ทำให้เมื่อเส้นทางเกิดความเสียหาย จะต้องทำการค้นหาเส้นทางใหม่ทันที ซึ่งทำให้เกิดโอเวอร์เฮดในการทำกระบวนการค้นหาเส้นทางทุกครั้งที่เส้นทางเกิดความเสียหายและส่งผลให้เกิดเวลาที่ใช้ในการส่งข้อมูล

ดังนั้นจึงเกิดเป็น โมเดลการค้นหาเส้นทางของเฉพาะกิจในสภาพแวดล้อมแบบหลายทาง ซึ่งจะทำการค้นหาเส้นทางมาใช้ในการสื่อสารหลายเส้นทาง โดยจะใช้เป็นเส้นทางหลักและเส้นทางสำรอง เมื่อเส้นทางหลักเกิดความเสียหาย ก็จะใช้เส้นทางสำรองแทน เมื่อเส้นทางเกิดความเสียหายจนหมด ค่อยทำการค้นหาเส้นทางใหม่ ทำให้ไม่ต้องทำการค้นหาเส้นทางทุกครั้งเมื่อเส้นทางเสียหาย ช่วยลดโอเวอร์เฮดและเวลาที่ใช้ในการส่งข้อมูลได้เป็นอย่างมาก

2.3.1 Ad Hoc On-demand Multipath Distance Vector Routing (AOMDV)

โพรโทคอล Ad Hoc On-demand Multipath Distance Vector Routing หรือ AOMDV [2] เป็นโพรโทคอลที่จัดทำขึ้นเพื่อมีจุดประสงค์ที่จะลดโอเวอร์เฮดในการค้นหาเส้นทาง เป็นแบบการค้นหาเส้นทางที่เริ่มจากต้นทาง (source initiate) คือเมื่อต้นทางมีข้อมูลที่ต้องการจะส่ง จะเริ่มกระบวนการค้นหาเส้นทางโดยส่ง RREQ เพื่อหาเส้นทางไปยังปลายทาง ซึ่งการค้นหาเส้นทางแต่ละครั้งจะใช้เวลาและโอเวอร์เฮดสูง ดังนั้นเพื่อที่จะเพิ่มประสิทธิภาพของ on-demand routing จึงไม่ควรจะทำการค้นหาเส้นทางบ่อยๆ ซึ่งการใช้โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทางจะลดจำนวนการค้นหาเส้นทางได้มากกว่าโพรโทคอลการค้นหาเส้นทางแบบทางเดียว

โหนดต้นทางจะค้นหาเส้นทางไปยังปลายทางหลายเส้นทางภายในการบรอดแคสต์ RREQ หนึ่งครั้ง เพราะฉะนั้นจะทำการค้นหาเส้นทางใหม่อีกครั้งก็ต่อเมื่อเส้นทางเหล่านั้นใช้การไม่ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมดแล้ว ต่างจากโพรโทคอลการค้นหาเส้นทางแบบทางเดียว ที่ถ้าหากเส้นทางเสียหาย จะต้องทำการค้นหาเส้นทางใหม่ทุกครั้ง ดังนั้นทำให้โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทางสามารถลดจำนวนการค้นหาเส้นทางได้ ทำให้มีโอเวอร์เฮดน้อยกว่า

กระบวนการค้นหาเส้นทางของโพรโทคอล AOMDV จะทำการค้นหาหลายเส้นทางจากโหนดต้นทางไปถึงโหนดปลายทาง และเป็นเส้นทางที่ไม่มีลูป

โพรโทคอล AOMDV ได้นำคุณลักษณะของโพรโทคอล AODV มาใช้ดังนี้

1. กลไกการค้นหาเส้นทาง และกลไกการรักษาเส้นทาง

กระบวนการค้นหาเส้นทาง คือกระบวนการที่โหนดต้นทางใช้หาเส้นทางไปยังปลายทาง เมื่อโหนดต้นทางต้องการเส้นทางไปยังปลายทาง โดยการฟลัด RREQ และรอรับ RREP แต่ละโหนดจะตั้ง next hop ของ reverse path เป็นฮอปก่อนหน้าที่ส่ง RREQ มาให้มัน ถ้าหากโหนดมีเส้นทางไปยังปลายทางก็จะตอบกลับ RREP ไปยังต้นทางแบบยูนิแคสกลับ ไปทาง reverse path ถ้าหากได้รับ RREP ซ้ำก็จะทำการละทิ้งทิ้ง เมื่อปลายทางได้รับ RREQ ก็จะทำการตอบกลับไปหาต้นทางทาง reverse path เมื่อต้นทางได้รับ RREP ก็จะทำการสร้างเส้นทาง forward path ได้

กระบวนการรักษาเส้นทาง เมื่อโหนดระหว่างทางตรวจพบว่าเส้นทางขาด จะสร้าง RERR (route error) ส่งไปยังต้นทางที่ใช้เส้นทางผ่านทางนี้ และลบเส้นทางออก

2. เลขลำดับ (Sequence number) และเป็นเส้นทางที่ไม่มีลูป

ทุกๆ โหนดจะเพิ่มค่า sequence number ของตนก่อนบรอดแคสแพคเกจ และจะเก็บค่า destination sequence number คือค่า sequence number สูงสุดของเส้นทางไปยังปลายทางต่างๆ ใน routing table จะถูกแนบไปในทุก routing message เพื่อให้ดูความสดใหม่ของเส้นทาง โดยที่เลขมากกว่าจะมีค่า destination sequence number มากกว่า และโพรโทคอลนี้ป้องกันการเกิดลูปของเส้นทางโดยใช้กฎ sequence number rule คือโหนดจะรักษาเส้นทางที่มีค่า destination sequence number สูงสุดเท่านั้น ในการเลือกเก็บเส้นทางจึงเก็บเฉพาะเส้นทางที่มีค่า destination sequence number มากกว่าหรือเท่ากัน ในกรณีที่มี destination sequence number เท่ากัน

- route advertisement rule โหนดที่มีเส้นทางจะประกาศเส้นทางที่มี hop count มากที่สุดออกไปก่อน (เส้นทางยาวที่สุด)
- route acceptance rule โหนดที่รับเส้นทางจะยอมรับเส้นทางจากโหนดเพื่อนบ้านก็ต่อเมื่อมีจำนวนฮอปน้อยกว่าหรือเท่ากันกับที่ได้รับประกาศก่อนหน้านี้เท่านั้น

ทุกโหนดจะมีการประกาศ hop count ไปยังแต่ละปลายทาง โดยภายในนั้นจะมีความยาวของเส้นทางไปยังปลายทาง โดยครั้งแรกจะประกาศเส้นทางที่ยาวที่สุดไปยังปลายทางด้วยค่า destination sequence number หนึ่ง ค่า advertised hop count จะเปลี่ยนแปลงเมื่อ sequence number เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนงานวิชาการ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยน การประกาศเส้นทางที่ยาวที่สุดทำให้เพิ่มโอกาสในการรับได้หลายเส้นทางเนื่องจากโหนด จะยอมรับเส้นทางที่ประกาศจากโหนดเพื่อนบ้านเฉพาะในกรณีที่มี hop count ต่ำกว่าที่เคยได้รับ ประกาศมาเท่านั้น

Disjoint path

AOMDV จะมีกลไกในการเลือกเส้นทางที่เป็น disjoint path คือถ้าแต่ละเส้นทางจากโหนด ต้นทางไปยังโหนดปลายทางเป็นเส้นทางที่ไม่มีการใช้ลิงค์เดียวกันเลย จะเกิดขึ้นได้เมื่อ โหนดมี next hop ต่างกันและ last hop (โหนดสุดท้ายก่อนถึงปลายทาง) ต่างกันเท่านั้น ซึ่ง จะต้องมีการเก็บ ค่า last hop ของทุกๆเส้นทางใน routing table และภายใน RREQ และ RREP ก็ต้องมีข้อมูล last hop ด้วย

ตารางเส้นทาง (Routing Table)

ตารางที่ 2.1 แสดงข้อมูล โครงสร้างตารางเส้นทางของโพรโทคอล AODV

Destination	Sequence number	Hop count	Next hop	Timeout
-------------	-----------------	-----------	----------	---------

ตารางที่ 2.2 แสดงข้อมูล โครงสร้างตารางเส้นทางของโพรโทคอล AOMDV

Destination	Sequence number	Advertised hop count	Route list			
			Next hop	Last hop	Hop count	Timeout
			Next hop	Last hop	Hop count	Timeout
			Next hop	Last hop	Hop count	Timeout
		

จะมีฟิลด์ประกาศ hop count (Advertised hop count) และรายการเส้นทาง (route list) เพื่อ เก็บข้อมูลสำหรับเส้นทางสำรอง (alternative path) เมื่อใดก็ตามที่ destination sequence number ที่ โหนดใด ๆ มีการอัปเดต โหนดจะเริ่มทำการประกาศ hop count สำหรับ destination sequence number นั้น โดยก่อนที่จะประกาศจะทำการอัปเดตค่าประกาศ hop count ตามสมการนี้

$$\text{Advertised_hop_count}_i^d := \max_k \left\{ \text{hop}_{\text{count}_{ik}}^d \right\}, i \neq d$$

$$:= 0, \text{ otherwise.} \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลไกการค้นหาเส้นทาง

เมื่อต้นทางต้องการเส้นทางไปยังปลายทางจะสร้าง RREQ แล้วฟลัดไปในเครือข่าย โหนดจะไม่ละทิ้ง RREQ ที่ได้ซ้ำเนื่องจากอาจใช้สร้างเส้นทางสำรองได้ แต่จะใช้ก็ต่อเมื่อเป็นเส้นทางที่ไม่มีลูป และ disjoint จากเส้นทางอื่นๆเท่านั้น

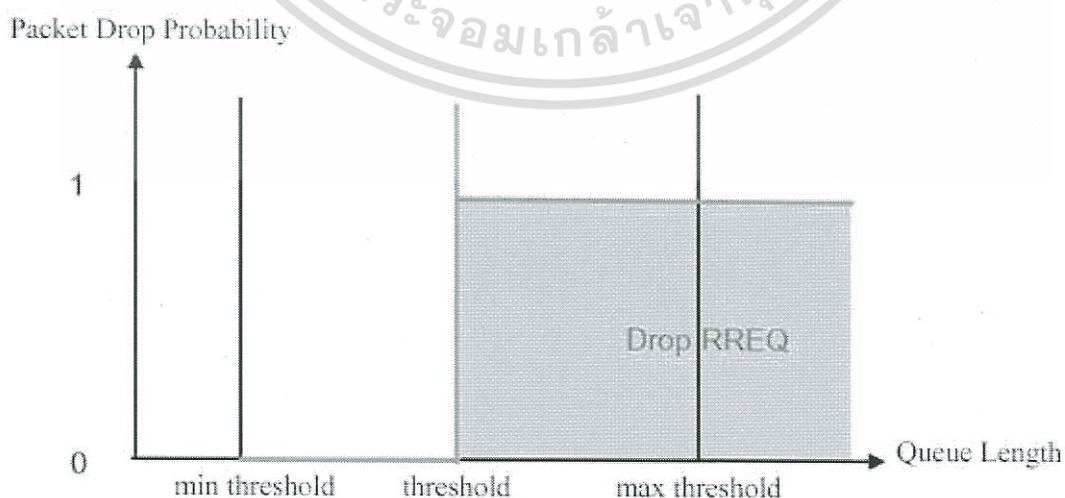
เมื่อโหนดระหว่างทางได้รับ RREQ จะทำการตรวจสอบว่ามีเส้นทางที่จะส่งต่อไปยังปลายทางได้หรือไม่ ถ้ามีโหนดก็จะสร้าง RREP แล้วส่งกลับไปให้โหนดต้นทางผ่านทาง reverse path แต่ถ้าหากไม่มีเส้นทางก็จะรีบรอดแคสต่อไปอีกและตั้ง reverse path เมื่อปลายทางได้รับ RREQ ก็จะทำการตั้ง reverse path เช่นกัน

กลไกการรักษาเส้นทาง

จะใช้ RERR เหมือนโพรโทคอล AODV โดยที่โหนดจะสร้าง RERR ก็ต่อเมื่อเส้นทางสุดท้ายที่มีไปยังปลายทางขาด แต่โพรโทคอล AOMDV จะให้ส่ง salvage packet ไปยังเส้นทางที่เสียหายด้วย โดยการส่งอีกรอบไปยังเส้นทางสำรอง

2.3.2 A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network

Workload-based Adaptive Load-Balancing Technique หรือ WAL [3] เป็นโพรโทคอลที่มีการกระจายภาระงานในการสร้างเส้นทาง เพื่อลดความคับคั่งของโหนด โดยแต่ละโหนดจะสามารถเลือกส่ง route request (RREQ) ได้ จะมีแค่โหนดที่ไม่คับคั่งเป็นโหนดระหว่างทางในการส่งต่อ RREQ โหนดที่อยู่ในสถานะที่มีภาระงานมากเกินไป (overload) จะไม่ยอมให้มีการสร้างเส้นทางผ่านตนเองเพิ่ม โดยจะไม่ส่งต่อ RREQ เป็นระยะเวลาหนึ่ง โดยโหนดจะกลับมาส่งต่อ RREQ เพื่อสร้างเส้นทางสร้างผ่านตนเองได้อีกก็ต่อเมื่อโหนดนั้นไม่อยู่ในสถานะที่มีภาระงานมากเกินไปแล้ว



รูปที่ 2.2 แสดงการทำงานของโพรโทคอล WAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอล WAL จะมีการกำหนดค่า threshold ที่น้อยที่สุด (\min_{th}) และ ค่า threshold ที่มากที่สุด (\max_{th}) เอาไว้ เมื่อมี RREQ เข้ามายังโหนด จะมีการตรวจสอบขนาดคิวว่า เกินค่า \min_{th} หรือไม่ ถ้าหากขนาดคิวไม่เกิน จะทำการส่ง RREQ นั้นทันที เป็นการยอมให้สร้างเส้นทางผ่านโหนดนั้น ๆ และตรวจสอบขนาดคิวว่า เกินค่า \max_{th} หรือไม่ ถ้าเกินก็จะทำการละทิ้ง RREQ จะไม่อนุญาตให้สร้างเส้นทางผ่านเพราะ ในขณะนั้นมีความคับคั่งอยู่ แต่ถ้าหากอยู่ระหว่าง \min_{th} และ \max_{th} จะต้องใช้ค่า threshold เป็นตัวกำหนดว่าจะยอมให้สร้างเส้นทางผ่านโหนดนั้นได้หรือไม่ ถ้าขนาดคิวไม่เกินค่า threshold ก็จะอนุญาตให้ส่ง RREQ ต่อ แต่ถ้าขนาดคิวมีค่าเกิน threshold จะทำการละทิ้ง RREQ นั้น ซึ่งค่า threshold นี้จะมีการเปลี่ยนแปลงและมีการอัปเดตค่า threshold เสมอ เมื่อมี RREQ เข้ามายังโหนดนั้น ๆ โดยค่าเริ่มแรกของ threshold จะถูกกำหนดไว้ที่ค่า \max_{th} จนกระทั่งเกิดความคับคั่งของข้อมูลในโหนดนั้นหรือเกิดการะงันมากเกิน ตาม 2 เงื่อนไขดังนี้ (1) ขนาดของคิวในโหนดนั้นเริ่มเพิ่มมากขึ้น จนเกินค่าเฉลี่ยของค่า \min_{th} และ \max_{th} (2) ระยะเวลาของงานที่เหลืออยู่ มีค่ามากกว่าค่าที่กำหนดไว้เพื่อเป็นตัววัดสถานะภาระงาน ซึ่งระยะเวลาของงานที่เหลืออยู่มีหน่วยเป็นวินาที จะเริ่มนับเมื่อขนาดของคิวของโหนดนั้นเพิ่มมากขึ้นจนเกินค่าเฉลี่ยของ \min_{th} และ \max_{th}

เมื่อเกิดสถานะที่มีภาระงานมากเกิน threshold จะปรับลดลงมาเป็นระยะ thres_{dec} (ค่าที่กำหนดว่าจะลดลง threshold ครั้งละเท่าไร) ถ้าคิวที่มีอยู่ยังคงเกินค่า threshold จะไม่อนุญาตให้สร้างเส้นทางผ่านได้อีกเพราะสถานะของปริมาณภาระงานที่ผ่านโหนดในตอนนี้มีความคับคั่งมากอยู่แล้ว และจะค่า threshold ลดทุกครั้งที่ได้รับ RREQ จนกระทั่ง ไปหยุดอยู่ที่ \min_{th} เป็นระยะเวลาหนึ่ง จะหมายความว่า คิวที่มีอยู่ของโหนดได้ทำการส่งออกไปจนกระทั่งโหนดนั้นไม่อยู่ในสถานะที่คับคั่งแล้ว เมื่อนั้นค่า threshold จะกลับไปอยู่ที่ค่าเริ่มต้นหรือที่ \max_{th} อีกครั้ง

แม้ว่าการละทิ้ง RREQ จะช่วยในเรื่องการกระจายภาระงาน แต่ถ้าหากว่าละทิ้งมากไปอาจทำให้การหาเส้นทางไปยังโหนดปลายทางไม่สามารถทำได้ โพรโทคอลนี้จึงมีการกำหนดให้ใช้ priority flag คือ ถ้าหากโหนดต้นทางส่ง RREQ ไปแล้วยังไม่ได้รับ route reply (RREP) ไปจนครบเวลา timeout โหนดนั้นจะทำการส่ง RREQ ใหม่โดยตั้ง flag เป็น 1 เมื่อโหนดระหว่างทางได้รับ RREQ แม้ว่าโหนดนั้นมีขนาดคิวเกินกว่าค่า threshold โหนดก็จะยอมส่ง RREQ นั้นต่อให้ เพื่อให้โหนดต้นทางได้เส้นทางในการส่งข้อมูล

บทที่ 3

งานที่นำเสนอ

3.1 งานที่นำเสนอ

จากการศึกษางานวิจัย A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network มีการใช้เทคนิคการกระจายการไหลลด โดยการใช้ค่า threshold ที่ปรับเปลี่ยนได้ตามความคับคั่งของข้อมูลที่ผ่านมาในโหนดนั้นๆ และใช้ขนาดคิวเป็นตัวจำกัดการสร้างเส้นทางผ่านโหนดที่มีความคับคั่ง เมื่อมี RREQ เข้ามายังโหนด โหนดจะตรวจสอบสถานะความคับคั่ง ถ้าขนาดคิวที่มีอยู่มากเกินกว่าค่า threshold โหนดนั้นจะไม่ยอมให้มีการสร้างเส้นทางผ่านเพิ่มเลย จนกระทั่งสถานะความคับคั่งของโหนดนั้นจะบรรเทาลง การกระทำเช่นนี้จะทำให้ตัดโอกาสการสร้างเส้นทาง และสำหรับกรณีที่เครือข่ายมีสถานะที่โหนดส่วนมากในเครือข่ายมีความคับคั่ง จะทำให้โหนดต้นทางไม่สามารถสร้างเส้นทางไปยังโหนดปลายทางได้ เนื่องจากขนาดคิวของโหนดส่วนมากในเครือข่ายจะเกินค่า threshold ส่งผลให้ไม่มี RREQ ส่งไปถึงปลายทาง จึงทำให้โหนดต้นทางต้องส่ง RREQ เพื่อขอสร้างเส้นทางอีกครั้ง โดยที่ส่ง RREQ ที่มีค่า flag เป็น 1 ซึ่งหมายความว่าแม้โหนดระหว่างทางที่ได้รับ RREQ นั้นจะมีขนาดคิวเกินค่า threshold ก็จะไม่ส่งต่อ RREQ นั้นเสมอ ในกรณีเช่นนี้ จะทำให้ใช้เวลาในการสร้างเส้นทางมากขึ้น และส่งผลถึงเวลาที่ใช้ในการส่งข้อมูลอีกด้วย ทางผู้จัดทำจึงมีแนวคิดให้โหนดมีความยืดหยุ่นยอมให้มีการสร้างเส้นทางผ่านโหนดนั้นได้อีกแม้ว่าจะมีขนาดคิวเกินค่า threshold แล้ว แต่ยังไม่เกินค่า max_h (โหนดไม่คับคั่งมาก) โดยนำการประยุกต์ใช้ค่าความน่าจะเป็นมาเป็นตัวชี้วัดในการละทิ้ง RREQ ตามสมการต่าง ๆ เพื่อจำกัดการสร้างเส้นทางไม่ให้โหนดนั้น ๆ มีความคับคั่ง ซึ่งสมการที่ใช้คำนวณค่าความน่าจะเป็นในการละทิ้ง 3 สมการ ดังนี้

- สมการเส้นตรง
- สมการเอ็กซ์โพเนนเชียล
- สมการลอการิทึม

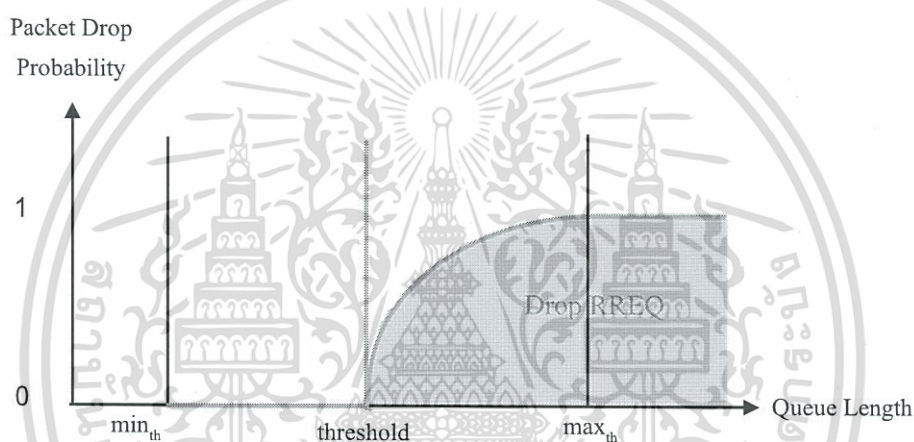
ทางผู้จัดทำได้นำอัลกอริทึมมาใช้กับโปรโตคอลการค้นหาเส้นทางแบบหลายเส้นทาง คือ โปรโตคอล AOMDV จากผลการทดลองชี้ให้เห็นว่า ค่าความน่าจะเป็นตามสมการลอการิทึมสามารถเพิ่มประสิทธิภาพในด้านการลดเวลาที่ใช้ในการส่งข้อมูลและการสร้างเส้นทาง และไม่ส่งผลกระทบต่อโอเวอร์เฮดกับอัตราการส่งข้อมูลสำเร็จ ระหว่างโปรโตคอลที่นำเสนอ และโปรโตคอลที่ใช้เทคนิค WAL

3.1.1 แนวคิดใช้ขนาดคิวเป็นตัวบ่งบอกความคับคั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางผู้จัดทำได้ทำการทดลองและพบว่าการใช้ขนาดคิวเป็นปัจจัยในการเลือกสร้างเส้นทาง และเป็นตัวชี้วัดที่บ่งบอกความคับคั่งของโพรโทคอล WAL มีประสิทธิภาพดีกว่าการใช้ Round Trip Time (RTT) ของโพรโทคอล AOMDV เนื่องจาก RTT เป็นตัวชี้วัดที่มีการเก็บสะสมค่าจากค่าก่อนหน้า จึงไม่สามารถบ่งบอกได้ชัดเจนว่ามีความคับคั่งหรือไม่ ดังนั้นเส้นทางที่มีเวลาที่ใช้ในการส่งข้อมูลน้อย ไม่ได้หมายความว่า จะมีความคับคั่งน้อยที่สุดเสมอไป แต่ขนาดคิวสามารถบ่งบอกความคับคั่ง ณ ขณะนั้นได้ชัดเจนกว่า ดังนั้นผู้จัดทำจึงสนใจแนวคิดของโพรโทคอล WAL ที่ใช้ RTT เป็นตัวชี้วัดในการเลือกเส้นทางเพื่อหลีกเลี่ยงเส้นทางที่คับคั่งจะได้ผลไม่แม่นยำเท่าการใช้ขนาดคิวที่วัดก่อนการเลือกเส้นทาง

3.1.2 หลักการทำงานของอัลกอริทึมที่นำเสนอ



รูปที่ 3.1 แสดงการทำงานของโพรโทคอลที่นำเสนอ

หลักการทำงานจะคล้ายกับโพรโทคอล WAL แต่ต่างกันตรงที่มีการคำนวณค่าความน่าจะมีแนวโน้มในการละทิ้งแพคเกจเพิ่มเข้าไปในการทำงานของโพรโทคอล WAL ที่เมื่อขนาดคิวของโพรโทคอล WAL มีขนาดเกินค่า $threshold$ จะทำการละทิ้ง RREQ ทันที

แต่สำหรับการประยุกต์โพรโทคอลที่นำเสนอนั้น เมื่อโหนดระหว่างทางที่ได้รับ RREQ มีขนาดคิวเกินค่า $threshold$ แล้ว แต่ยังไม่เกินค่า max_q จะไม่ทำการละทิ้ง RREQ นั้นทันที แต่จะทำการคำนวณค่าความน่าจะเป็นในการละทิ้งจากสมการลอการิทึม โดยจากรูปที่ 3.1 โดยความน่าจะเป็นจะแปรผันตามปริมาณคิวของโหนดนั้น ๆ คือ แม้ขนาดคิวเกินค่า $threshold$ แล้ว แต่ก็ยังยอมให้สร้างเส้นทางผ่านอยู่ แต่หากขนาดคิวเริ่มเข้าใกล้ค่า max_q โอกาสถูกละทิ้ง RREQ ก็จะมีมากขึ้น จนกระทั่งเมื่อขนาดคิวเกินค่า max_q ความน่าจะเป็นที่ RREQ จะถูกละทิ้งจะเท่ากับ 1 หมายความว่าเมื่อขนาดคิวเกินค่า max_q แล้ว RREQ จะถูกละทิ้งเสมอ (ไม่ยอมให้มีการสร้างเส้นทางผ่าน) เพื่อควบคุมไม่ให้มีการรับสร้างเส้นทางผ่านมากเกินไป เพื่อป้องกันการละทิ้งแพคเกจที่เกิดจากขนาด

คิวเต็ม ผู้จัดทำจึงเชื่อว่าการดำเนินการเช่นนี้จะช่วยลดเวลาในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลลงได้

3.1.3 สมการที่ใช้คำนวณความน่าจะเป็นในการละทิ้ง RREQ

สมการเส้นตรง

เมื่อขนาดของคิวอยู่ระหว่างค่า threshold และค่า threshold สูงที่สุด จะใช้ค่าความน่าจะเป็นเป็นตัวกำหนดว่าจะทำการละทิ้งแพ็คเกจหรือไม่ โดยค่าความน่าจะเป็นนั้นจะอิงตามสมการเส้นตรง ดังนี้

$$P = \frac{1}{(\text{max_th} - \text{threshold})} * (\text{queue_length} - \text{threshold}) \quad (3.1)$$

ซึ่งมีข้อเสียก็คือ หากมีแพ็คเกจส่งมาแบบ burst ขนาดคิวจะเพิ่มขึ้นสูงอย่างรวดเร็ว จะทำให้ไม่สามารถควบคุมขนาดคิวได้ทัน ส่งผลให้เกิดการละทิ้ง RREQ หรือปฏิเสธการสร้างเส้นทาง ซึ่งเป็นการใช้แบนวิธอย่างไม่มีประสิทธิภาพ เพราะโหนดจะเกิดความคับคั่งในช่วงระยะเวลาสั้นๆ เท่านั้น

สมการเอ็กซ์โพเนนเชียล

เมื่อขนาดของคิวอยู่ระหว่างค่า threshold และค่า threshold สูงที่สุด จะใช้ค่าความน่าจะเป็นเป็นตัวกำหนดว่าจะทำการละทิ้งแพ็คเกจหรือไม่ โดยค่าความน่าจะเป็นนั้นจะอิงตามสมการเอ็กซ์โพเนนเชียล ดังนี้

$$P = e^{-\frac{(\text{queue_length} * \ln 2)}{\text{max_th}}} - 1 \quad (3.2)$$

ซึ่งมีข้อเสียคือการที่มีโอกาสยอมให้สร้างเส้นทางผ่านมาก แม้ขนาดคิวจะเกินค่า threshold มาแล้ว ค่อยไปควบคุมไม่ให้สร้างผ่านเมื่อขนาดคิวถึงค่า max threshold จะทำให้มีโอกาสเกิด buffer overflow ได้ง่าย

สมการลอการิทึม

เมื่อขนาดของคิวอยู่ระหว่างค่า threshold และค่า threshold สูงที่สุด จะทำการคำนวณค่าความน่าจะเป็นเพื่อเป็นแนวโน้มนในการละทิ้งแพ็คเกจ โดยค่าความน่าจะเป็นนั้นจะอิงตามสมการลอการิทึม ดังนี้

$$P = \log_{\text{max_th}+1} (\text{queue_length} + 1) \quad (3.3)$$

ข้อดีคือ สามารถรองรับการที่จะมีแพ็คเกจเข้ามาแบบ burst เนื่องจากมีการกำหนดความน่าจะเป็นในการละทิ้งให้สูงในช่วงแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและประเมินผล

จากการศึกษางานวิจัยที่เกี่ยวข้อง ทางผู้จัดทำจึงได้ทำการทดสอบประสิทธิภาพการทำงานของโพรโทคอลต่างๆ และนำมาเปรียบเทียบประสิทธิภาพ 3 โพรโทคอล ดังนี้

- AOMDV เป็นโพรโทคอลพื้นฐาน
- AOMDV-WAL คือ โพรโทคอลที่นำเทคนิค WAL มาประยุกต์ใช้กับโพรโทคอล AOMDV ซึ่งใช้ขนาดคิวเป็นตัวตัดสินใจและทิ้ง RREQ เพื่อควบคุมการสร้างเส้นทาง
- Proposed Protocol คือ โพรโทคอลที่พัฒนามาจากโพรโทคอล WAL โดยเพิ่มความยืดหยุ่นในการรับสร้างเส้นทางในกรณีที่โหนดมีความคับคั่งของภาระงาน

ตารางที่ 4.1 แสดงข้อมูลสิ่งแวดล้อมของการจำลองเครือข่าย

สิ่งแวดล้อม	ค่าที่กำหนด
แบนวิธ	2 เมกะบิต/วินาที
จำนวนโหนด	100 โหนด
ความเร็วเฉลี่ย	10 เมตร/วินาที
ขนาดแฉกคอย	50 แฉกเกต
ประเภทของ interface queue	DropTail
ขนาดแฉกเกต	512 ไบต์
Pause time	100 วินาที
ขนาดเครือข่าย	2200 X 600 เมตร
บัฟเฟอร์ขาออก	64 แฉกเกต
เวลาที่ใช้จำลอง	500 วินาที
ระยะทางในการส่งข้อมูล	250 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางผู้จัดทำได้ใช้ NS-2 (Network simulator) ในการจำลองเครือข่าย โดยกำหนดให้ใช้แบนวิธเท่ากับ 2 Mbps และใช้โปรโตคอล AOMDV เป็นโปรโตคอลพื้นฐาน จำลองเปรียบเทียบประสิทธิภาพระหว่าง โปรโตคอล Workload-based Adaptive Load-Balancing Technique และโปรโตคอลที่นำเสนอ โดยมีบัพเฟอร์ที่ฝั่งส่งขนาด 64 แพคเกจเพื่อใช้สำหรับแพคเกจที่รอรับ RREP โดยกำหนดเวลาหมดอายุไว้ที่ 30 วินาที ขนาดของแฉกคอยจะกำหนดไว้ที่ 50 แพคเกจ ซึ่งแฉกคอยจะให้ความสำคัญกับ routing protocol packet มากกว่าแพคเกจข้อมูล

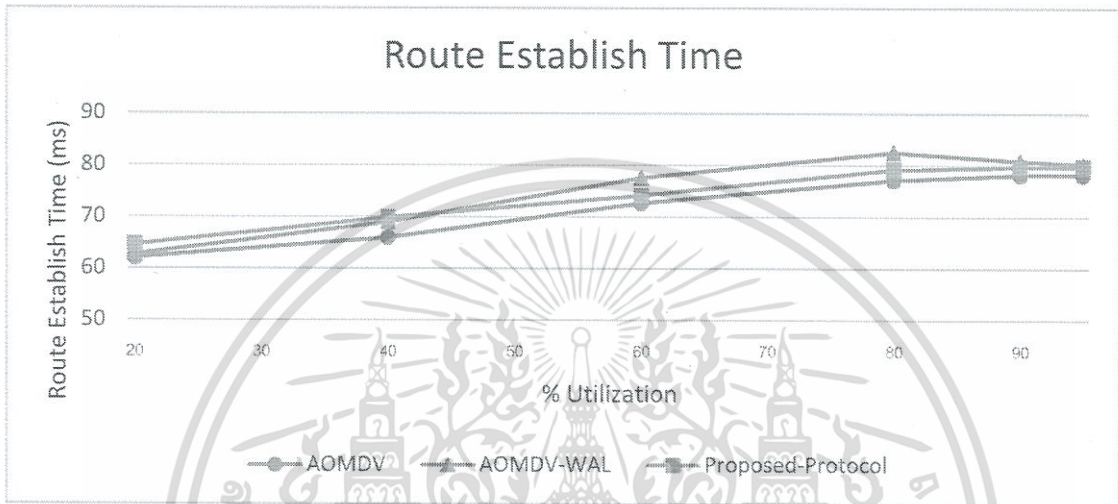
เราจะใช้การจำลองทั้งหมด 20 ครั้ง ซึ่งแต่ละครั้งจะมีรูปแบบของการเคลื่อนที่ต่างกัน กำหนด pause time (ระยะเวลาที่โหนดหยุดก่อนเคลื่อนที่) ไว้ที่ 100 วินาที ความเร็วของโหนดเฉลี่ยอยู่ที่ 10 เมตรต่อวินาที

เครือข่ายจะประกอบด้วย 100 โหนด มีระยะทางในการส่งข้อมูล 250 เมตร ใช้ทั้งหมด 40 traffic flow ที่มีขนาดของแพคเกจคือ 512 ไบต์ โดยค่อย ๆ เพิ่มอัตราการใช้แบนวิธจากร้อยละ 20 ถึง 95 ของแบนวิธ ในพื้นที่การจำลองขนาด 2200*600 เมตร เป็นระยะเวลา 500 วินาที (เวลาจำลอง)

ในการเปรียบเทียบประสิทธิภาพ ทางผู้วิจัยจะทำการวัดประสิทธิภาพในด้านต่าง ๆ ดังนี้ Packet Delivery Rate (PDR) คืออัตราของแพคเกจข้อมูลที่ถูกส่งได้สำเร็จ ต่อจำนวนแพคเกจที่โหนดต้นทางส่งออกไปทั้งหมด Routing Overhead (OH) คือจำนวนโอเวอร์เฮดต่อหนึ่งแพคเกจข้อมูล Route Establish Time (RET) คือเวลาที่ใช้ในการสร้างเส้นทาง (ms) End-to-end delay (EE) คือเวลาเฉลี่ยที่ใช้ในการรับส่งข้อมูลจากต้นทางไปถึงปลายทาง (ms)

4.1 ผลการทดลอง

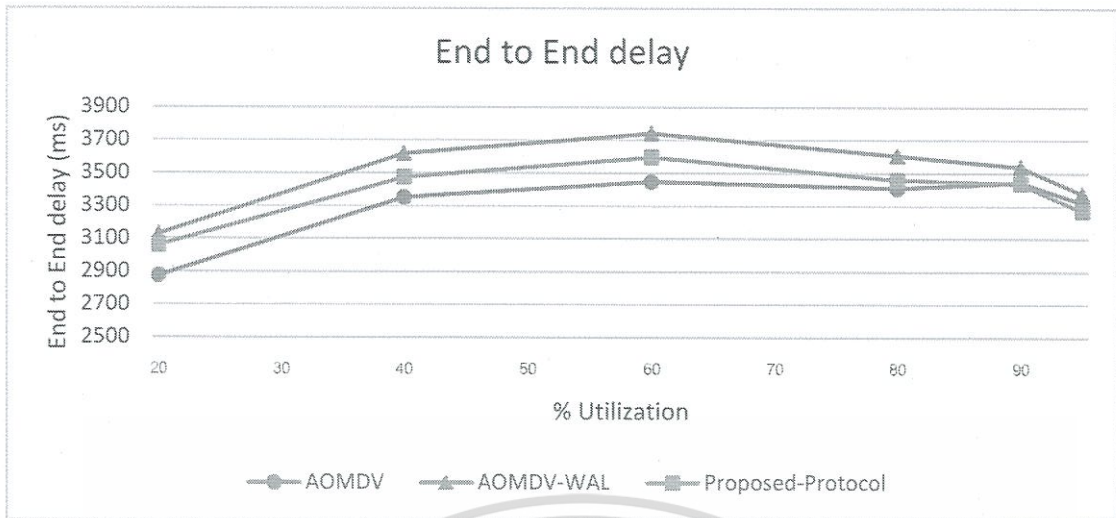
ในส่วนนี้จะแสดงถึงผลการจำลองเปรียบเทียบ โพรโทคอลที่นำเสนอกับ AOMDV ซึ่งเป็น โพรโทคอลพื้นฐาน และ A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network (WAL)



รูปที่ 4.1 แสดงการเปรียบเทียบเวลาที่ใช้ในการสร้างเส้นทาง

จากรูปที่ 4.1 แสดงผลลัพธ์จากการทดลองในค่านเวลาในการสร้างเส้นทาง โดยแสดงเป็นกราฟเปรียบเทียบระหว่างเวลาในการสร้างเส้นทางซึ่งมีหน่วยเป็นมิลลิวินาที (ms) กับร้อยละ (%) ของการใช้แบนวิธ ซึ่งพบว่าโพรโทคอล AOMDV จะใช้เวลาในการสร้างเส้นทางต่ำที่สุด เนื่องจากโพรโทคอล WAL และโพรโทคอลที่นำเสนอ นั้น มีการควบคุมการสร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้นๆ ส่งผลให้ต้องใช้เวลาเพิ่มขึ้นในการหาเส้นทางไปยังโหนดที่ไม่มีความคับคั่งแทน

เมื่อเปรียบเทียบโพรโทคอลที่นำเสนอกับโพรโทคอล WAL พบว่า โพรโทคอลที่นำเสนอ นั้นใช้เวลาในการสร้างเส้นทางต่ำกว่าโพรโทคอล WAL เนื่องจาก การที่โพรโทคอลที่นำเสนอ นั้น มีความยืดหยุ่นในการรับสร้างเส้นทางผ่านมากกว่า คือถ้าหากโหนดนั้นมีความคับคั่งเล็กน้อยก็จะยอมเป็นโหนดทางผ่านให้อยู่ ซึ่งถ้าหากป็นโพรโทคอล WAL จะทำการปฏิเสธการเป็นโหนดทางผ่านทันที ซึ่งในกรณีที่มีส่วนส่วนมากในเครือข่ายมีความคับคั่งแล้ว อาจทำให้โหนดต้นทางในโพรโทคอล WAL นั้นหาเส้นทางไปยังปลายทางไม่ได้ จนทำให้ RREQ นั้น timeout และต้องทำการส่ง RREQ อีกครั้ง ส่งผลให้ใช้เวลาในการสร้างเส้นทางสูงขึ้น

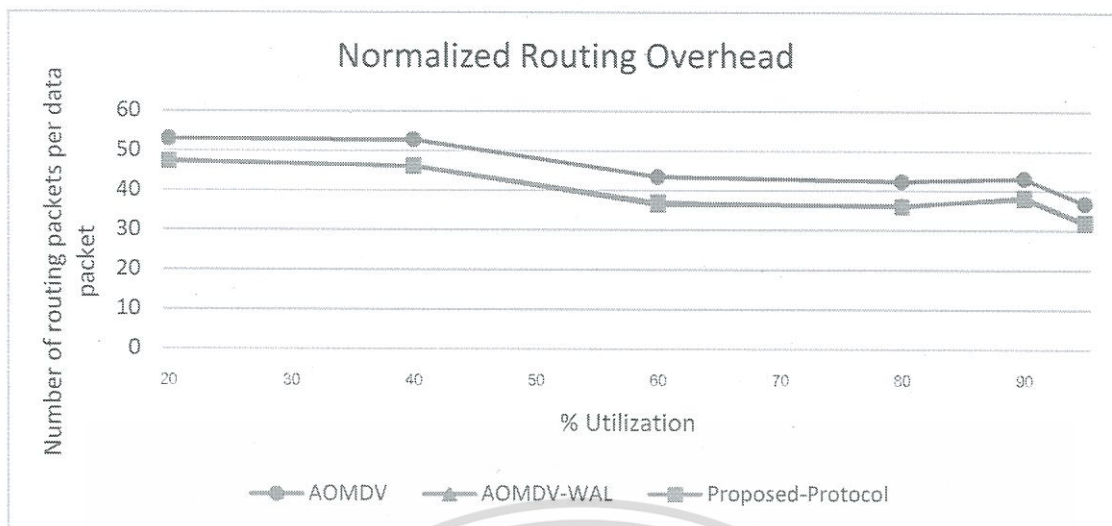


รูปที่ 4.2 แสดงการเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูล

จากรูป 4.2 แสดงเวลาเฉลี่ยที่ใช้ในการส่งหนึ่งแพคเกจข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง ซึ่งจะเห็นว่าโพรโทคอล AOMDV จะใช้เวลาในการส่งแพคเกจข้อมูลต่ำกว่าโพรโทคอล WAL และโพรโทคอลที่นำเสนอในช่วงแรก เนื่องจาก โพรโทคอล WAL และโพรโทคอลที่นำเสนอ นั้น มีการควบคุมการสร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้นๆ ส่งผลให้ต้องใช้เวลาในการสร้างเส้นทางเพิ่มขึ้น แต่จะเห็นว่าในช่วงที่มีการใช้แบนวิดมาก ๆ โพรโทคอลที่นำเสนอจะใช้เวลาในการส่งข้อมูลต่ำกว่า AOMDV เนื่องจากการที่ความเร็วในการส่งข้อมูลสูง จะทำให้โหนดระหว่างทางมีความคับคั่งมากการที่ควบคุมการสร้างเส้นทางจึงเป็นผลดี ทำให้สามารถใช้เส้นทางที่ไม่มี ความคับคั่งได้ และยังมี ความยืดหยุ่นในการยอมให้สร้างเส้นทางดีกว่าโพรโทคอล WAL ซึ่งช่วยลดการละทิ้ง RREQ ได้เหมาะสมกว่า

แต่เมื่อเทียบโพรโทคอลที่นำเสนอกับโพรโทคอล WAL จะเห็นว่าโพรโทคอลที่นำเสนอ นั้นใช้เวลาเฉลี่ยในการส่งแพคเกจข้อมูลต่ำกว่า เนื่องจากเวลาเฉลี่ยที่ใช้ในการส่งข้อมูลนั้นจะหมายถึงเวลาดังแต่โหนดต้นทางค้นหาเส้นทาง ไปจนถึงเวลาที่แพคเกจข้อมูลถูกส่งไปถึงโหนดปลายทาง ซึ่งจะรวมถึงเวลาที่ค้นหาเส้นทางใหม่ในกรณีที่เส้นทางขาดหรือเสียหายด้วย สำหรับการจำลองเครือข่ายในสถานะแวดล้อมที่โหนดมีการเคลื่อนไหวยู่ตลอดเวลา จะทำให้เส้นทางในการส่งข้อมูลเสียหายบ่อยครั้ง ถ้าในขณะที่โหนดมีการส่งข้อมูลอยู่แล้วเส้นทางเกิดความเสียหายขึ้น โหนดต้นทางจะต้องทำการค้นหาเส้นทางใหม่ ซึ่งโพรโทคอล WAL ที่ไม่มีความยืดหยุ่นในการรับสร้างเส้นทาง ทำให้โหนดต้นทางต้องใช้เวลาในการค้นหาเส้นทางใหม่ มากกว่าโพรโทคอลที่นำเสนอ

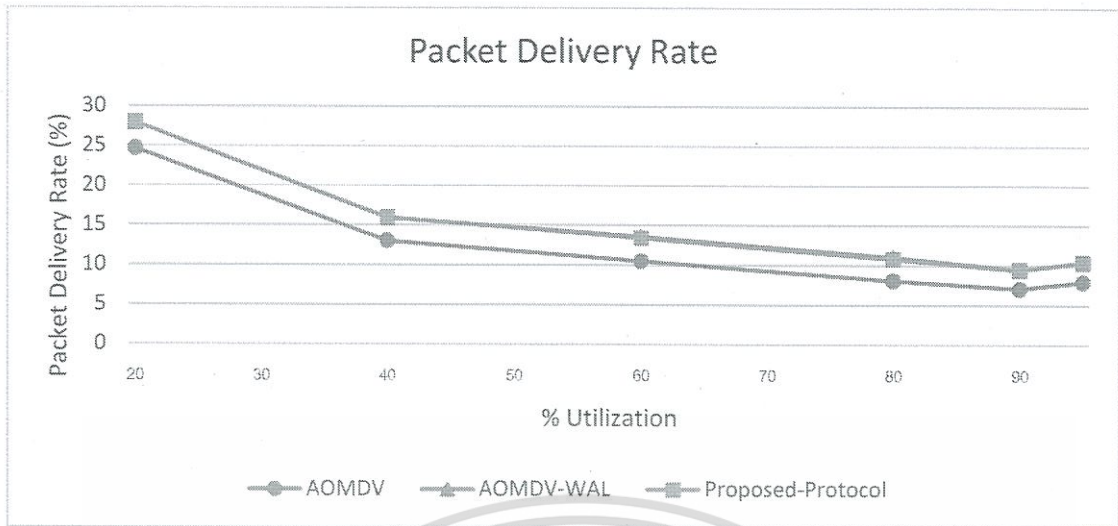
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการเปรียบเทียบโอเวอร์เฮดต่อหนึ่งแพคเกจข้อมูล

จากรูป 4.3 แสดงจำนวนแพคเกจที่ใช้ในกลไกค้นหาเส้นทางต่อหนึ่งแพคเกจข้อมูลที่ส่งถึงปลายทาง แสดงให้เห็นว่า โพรโทคอล WAL และโพรโทคอลที่นำเสนอมีจำนวนโอเวอร์เฮดต่อหนึ่งแพคเกจข้อมูลต่ำกว่าโพรโทคอล AOMDV เนื่องจากการที่มีการจำกัดเส้นทางของกลไกการค้นหาเส้นทางที่ไม่ยอมให้โหนดที่มีภาระงานมากอยู่แล้วยอมรับเป็นทางผ่านส่งต่อ RREQ ทำให้โหนดที่มีภาระงานมากนั้นละทิ้ง RREQ จึงทำให้สามารถลดจำนวน RREQ ได้อย่างมีประสิทธิภาพ และโอเวอร์เฮดส่วนใหญ่ของโพรโทคอล AOMDV นั้นมาจาก RREQ จึงเป็นการลดจำนวนโอเวอร์เฮดที่ไม่จำเป็นภายในเครือข่ายลงได้ โดยไม่ส่งผลกระทบต่อประสิทธิภาพด้านอื่นๆ

เมื่อเปรียบเทียบระหว่างโพรโทคอลที่นำเสนอกับโพรโทคอล WAL แล้วพบว่าโพรโทคอลที่นำเสนอมีจำนวนโอเวอร์เฮดสูงกว่าเล็กน้อย เนื่องมาจากการที่โพรโทคอลที่นำเสนอมีความยืดหยุ่นยอมให้ส่งต่อ RREQ แม้ขณะนั้นโหนดเริ่มมีความคับคั่งขึ้นแล้วโดยการสุ่มความน่าจะเป็นตามสมการ ซึ่งก็ยังคงมีการจำกัดการสร้างเส้นทางอยู่ จึงทำให้มีบางช่วงที่จำนวนโอเวอร์เฮดสูงกว่าโพรโทคอล WAL เพียงเล็กน้อย



รูปที่ 4.4 แสดงการเปรียบเทียบอัตราการส่งแพ็คเกจสำเร็จ

จากรูปที่ 4.4 อัตราการส่งแพ็คเกจสำเร็จ คืออัตราการของแพ็คเกจที่ถูกส่งได้สำเร็จต่อแพ็คเกจที่โหนดต้นทางส่งทั้งหมด จะแสดงให้เห็นว่าทั้งโพรโทคอล WAL และโพรโทคอลที่นำเสนอ มีอัตราการส่งข้อมูลได้สำเร็จสูงกว่า AOMDV มาก เนื่องจากการที่โหนดได้ใช้เส้นทางที่มีความคับคั่งของข้อมูลน้อยกว่าจะทำให้ลดโอกาสในการที่แพ็คเกจจะถูกละทิ้ง เนื่องจากขนาดคิวของโหนดระหว่างทางเต็ม และเมื่อเทียบประสิทธิภาพในการส่งข้อมูลได้สำเร็จของโพรโทคอล WAL และโพรโทคอลที่นำเสนอพบว่าอัตราการส่งข้อมูลได้สำเร็จไม่แตกต่างกันมากนัก

บทที่ 5

สรุปผล

จากการศึกษาและวิเคราะห์ประสิทธิภาพของโพรโทคอลสำหรับเครือข่ายเฉพาะกิจในสภาพแวดล้อมหลายเส้นทาง ผู้จัดทำได้เลือกพัฒนาโพรโทคอล WAL เนื่องจากมีการใช้ขนาดคิวเป็นปัจจัยในการเลือกสร้างเส้นทางและใช้เป็นตัวเลือกสร้างเส้นทาง ซึ่งมีประสิทธิภาพดีกว่าการใช้ Round Trip Time (RTT) ของโพรโทคอล AOMDV (การใช้เส้นทางตาม RREP แรกที่มาถึง) ผู้จัดทำจึงสนใจที่จะพัฒนาประสิทธิภาพของโพรโทคอล WAL และเห็นว่าในการทำงานของโพรโทคอล WAL มีการเลือกเส้นทางที่ไม่ยืดหยุ่น คือจะไม่ยอมให้มีการสร้างเส้นทางผ่านโหนดที่มีความคับคั่ง ทำให้โหนดต้นทางต้องใช้เวลาในการค้นหาเส้นทางเพิ่มขึ้น ส่งผลให้เวลาเฉลี่ยในการส่งข้อมูลไม่ดีเท่าที่ควร จึงมีแนวคิดที่จะลดเวลาในการค้นหาเส้นทางและปรับปรุงเวลาที่ใช้ในการส่งแพคเกจข้อมูล โดยมีแนวคิดคือ จะให้โหนดระหว่างทางจะมีความยืดหยุ่นในการรับเป็นทางผ่านมากขึ้น คือเมื่อขนาดคิวเกินค่า threshold แล้ว จะไม่ละทิ้ง RREQ ทันที แต่จะทำการสุ่มว่าจะรับเป็นทางผ่านหรือไม่ โดยความน่าจะเป็นในการสุ่มจะขึ้นอยู่กับปริมาณขนาดคิว ถ้าหากขนาดคิวมาก เข้าใกล้ค่า max_{th} (ค่าสูงสุดที่จะยอมให้สร้างเป็นทางผ่าน) โอกาสที่ RREQ จะถูกละทิ้งจะมีมาก และถ้าหากขนาดคิวมากกว่าค่า threshold แต่ยังมีน้อยอยู่ก็จะมีโอกาสที่จะละทิ้ง RREQ ต่ำ ซึ่งจะทำให้โหนดต้นทางมีโอกาสได้เส้นทางผ่านโหนดนั้น ๆ เป็นการลดเวลาในการสร้างเส้นทางลงได้

จากผลการทดลองพบว่าในช่วงที่มีอัตราการส่งข้อมูลต่ำ เวลาที่ใช้ในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลของโพรโทคอลที่นำเสนอกับโพรโทคอล WAL ใช้เวลามากกว่าโพรโทคอล AOMDV เนื่องจากมีการควบคุมการสร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้นๆ ส่งผลให้ต้องใช้เวลาในการสร้างเส้นทางเพิ่มขึ้น และเมื่อถูกปฏิเสธการสร้างเส้นทางผ่านโหนดนั้นๆ จะต้องทำการค้นหาเส้นทางและส่งข้อมูลใหม่อยู่ตลอด ซึ่งก็ส่งผลให้เวลาที่ใช้ในการส่งเส้นทางเพิ่มขึ้นด้วย แต่เมื่อมีการใช้อัตราเร็วในการส่งข้อมูลสูง โพรโทคอล AOMDV จะใช้เวลาในการสร้างเส้นทางและส่งข้อมูลสูงกว่าโพรโทคอลที่นำเสนอ ซึ่งการที่มีการควบคุมการสร้างเส้นทางก็ยังส่งผลกระทบต่อค่าโอเวอร์เฮด ทำให้ค่าโอเวอร์เฮดของโพรโทคอลที่นำเสนอกับโพรโทคอล WAL ต่ำลง เนื่องจากทำให้โหนดที่มีภาระงานมากนั้นละทิ้ง RREQ จึงทำให้สามารถลดจำนวน RREQ ได้อย่างมีประสิทธิภาพ และโอเวอร์เฮดส่วนใหญ่ของโพรโทคอล AOMDV นั้นมาจาก RREQ นอกจากนี้เนื่องจากการที่โหนดได้ใช้เส้นทางที่มีความคับคั่งของข้อมูลน้อยกว่าจะทำให้ลดโอกาสในการที่แพคเกจจะถูกละทิ้งเนื่องจากรูปแบบของโหนดระหว่างทางเต็ม จึงส่งผลให้อัตราการส่งข้อมูลไม่ต่ำกว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำเร็จ ของโพรโทคอลที่นำเสนอและโพรโทคอล WAL มีอัตราการส่งข้อมูลสำเร็จมากกว่าโพรโทคอล AOMDV



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] C. E. Perkins, E. M. Belding-Royer, S. R. Das. “**Ad hoc On-Demand Distance Vector (AODV) Routing.**” Internet Engineering Task Force (IETF) draft, 2003
- [2] Mahesh K. Marina, Samir R. Das. “**Ad hoc on-demand multipath distance vector routing.**” Wirel. Commun. Mob. Comput. 2006
- [3] Young J. Lee, George F. Riley. “**A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network.**” Wireless Communications and Networking Conference, 2005



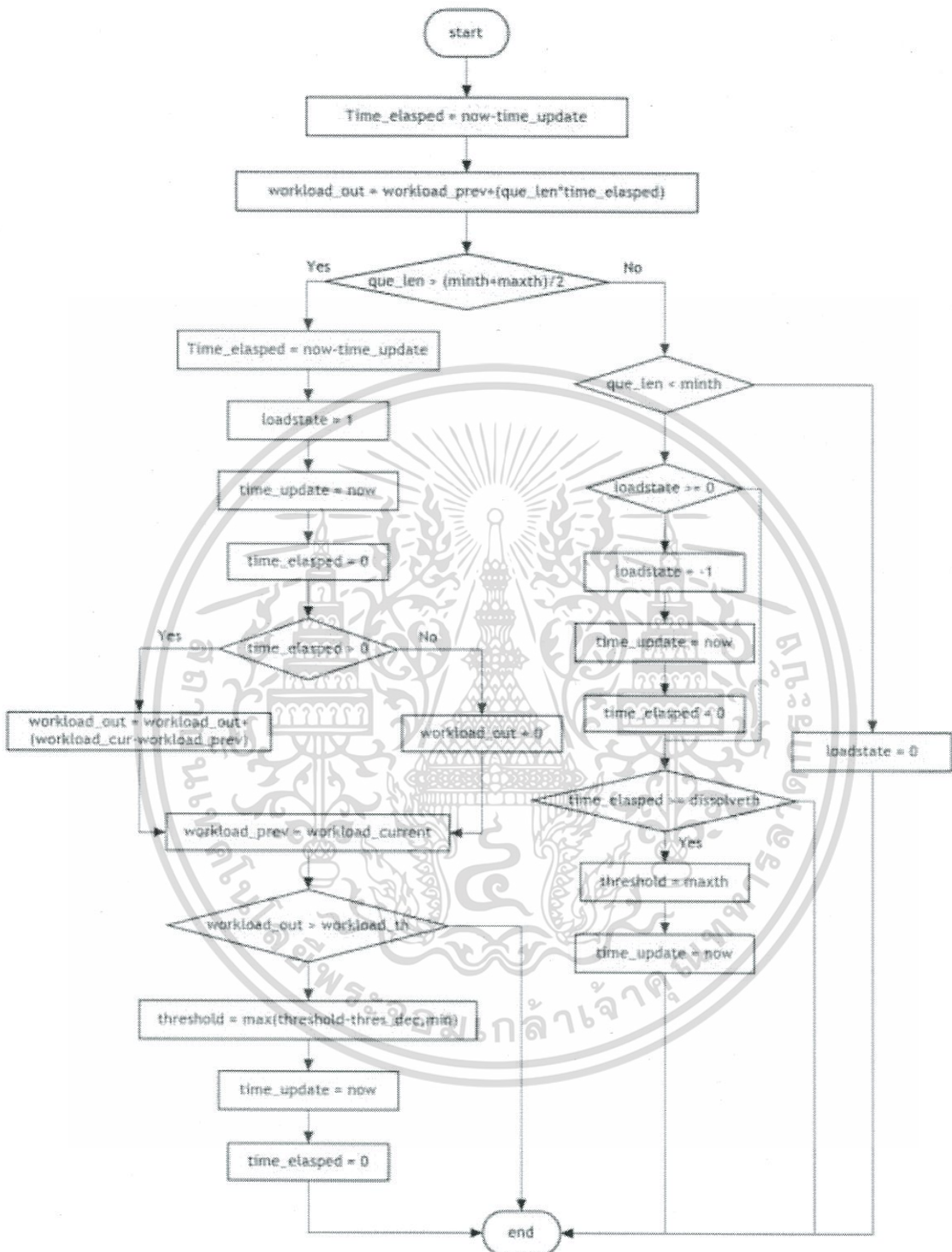
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart การอัปเดตค่า threshold



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโค้ดแสดงการอัปเดตค่า threshold

```

void
AOMDV::updateThreshold(){
    int que_len = AOMDVifqueue->length();
    time_elapsed = now-time_updated;
    workload_current = workload_prev+(que_len*time_elapsed);

    if(que_len>(minth+maxth)/2){
        if(load_state<=0){
            load_state=1;
            time_updated=Scheduler::instance().clock();
            time_elapsed=0;
        }
        if(time_elapsed>0){
            workload_out += workload_current-workload_prev;
        }
        else{
            workload_out=0;
        }
        workload_prev=workload_current;
        if(workload_out>workload_th){
            threshold=max(threshold-thres_dec,minth);
            time_updated=Scheduler::instance().clock();
            workload_out=0;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(que_len<minth){
    if(load_state>=0){
        load_state=-1;
        time_updated=Scheduler::instance().clock();
        time_elpased=0;
    }
    if(time_elpased>=dissolve_th){
        threshold=maxth;
        time_updated=Scheduler::instance().clock();
    }
}
else{
    load_state=0;
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโค้ดแสดงการคำนวณความน่าจะเป็นในการละทิ้ง RREQ

```
double
```

```
AOMDV::prob(double threshold,double queue){
```

```
    p = (log((AOMDVifqueue->length()-threshold)+1)/log(maxth-threshold+1));
```

```
    return p;
```

```
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ นางสาวปัทม์ชนิต สุขจิรัตติกาล
 วัน เดือน ปีเกิด 30 มีนาคม 2536
 ที่อยู่ บ้านเลขที่ 21/38 ซอยรามคำแหง 142 ถนนรามคำแหง แขวงสะพานสูง
 เขตสะพานสูง กรุงเทพฯ 10240
 ประวัติการศึกษา วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ นางสาวพกามาศ จิตโสภิพงษ์
 วัน เดือน ปีเกิด 30 มกราคม 2536
 ที่อยู่ บ้านเลขที่ 380 ซอยรามคำแหง 148 ถนนรามคำแหง แขวงสะพานสูง
 เขตสะพานสูง กรุงเทพฯ 10240
 ประวัติการศึกษา วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาประสิทธิภาพบนเครือข่ายแมนเนทบนสภาพแวดล้อม แบบหลายทาง

ปณตช์ชนิต สุขจิรัตติกาล และ ผกามาศ จิตโสภิงษ์

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: khaopun_panchanit@hotmail.com, j_janz@hotmail.com

บทคัดย่อ

ในการหาเส้นทางบนเครือข่ายเฉพาะกิจ การกระจายภาระงานมีผลกระทบต่อประสิทธิภาพเครือข่าย การหาเส้นทางที่ดีควรมีการกระจายภาระงานให้โหลดอย่างทั่วถึง มิฉะนั้นจะทำให้โหนดที่ถูกใช้เป็นทางผ่านการรับส่งข้อมูลอยู่ตลอด เกิดความคับคั่งจากปริมาณข้อมูลผ่านมากเกินไป ในขณะที่โหนดที่ไม่ค่อยถูกใช้ มีปริมาณข้อมูลผ่านน้อย ความไม่สมดุลในการกระจายภาระงานนี้ทำให้เกิดปัญหาความล่าช้าในการส่งข้อมูล การส่งข้อมูลล้มเหลว และเกิดโอเวอร์เฮดเพิ่มสูงขึ้นจากการต้องส่งข้อมูลที่ล้มเหลวซ้ำ บทความนี้ศึกษาแนวทางการแก้ปัญหาความไม่สมดุลดังกล่าว งานวิจัยก่อนหน้าเสนอแนวทางแก้ปัญหาโดยให้แต่ละโหนดจะพิจารณาความสามารถตัวเองในการรับภาระงานเพิ่ม ก่อนตัดสินใจว่าจะยอมเป็นทางผ่านในการรับส่งข้อมูลให้โหนดต้นทางที่ร้องขอมาหรือไม่ เพื่อให้เกิดความสมดุลในการกระจายภาระงาน แต่ในทางตรงกันข้าม โหนดต้นทางอาจต้องใช้เวลาและจำนวนครั้งในการร้องขอเส้นทางมากขึ้น ทำให้เกิดความล่าช้า ผู้วิจัยจึงเสนอการกระจายภาระงานอย่างสมดุลที่มีความยืดหยุ่นขึ้น เพื่อแก้ไขปัญหากล่าวมานี้ และผลการประเมินจากการจำลองเหตุการณ์แสดงให้เห็นว่าเวลาที่ใช้ในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลลดลง

คำสำคัญ – เครือข่ายเฉพาะกิจ; สภาพแวดล้อมแบบหลายเส้นทาง; การกระจายภาระงาน

1. บทนำ

เทคโนโลยีการสื่อสารของระบบเครือข่าย ปัจจุบันมีระบบเครือข่ายแบบโครงสร้างพื้นฐาน ซึ่งจำเป็นต้องมีอุปกรณ์ตัวกลางทำหน้าที่รับ-ส่งข้อมูล เมื่อเกิดเหตุไม่คาดฝันทำให้ระบบโครงสร้างพื้นฐานเกิดขัดข้อง ใช้งานไม่ได้ในสภาวะจำเป็น จึงได้เกิดเป็นเครือข่ายสื่อสารเฉพาะกิจเคลื่อนที่ของโมบายล์โหนด หรือ Mobile Ad-hoc Network (MANETs) เป็นเครือข่ายที่ประกอบด้วยกลุ่มของโหนดเคลื่อนที่ไร้สายที่สามารถเคลื่อนไหวได้อย่างอิสระ สร้างเครือข่ายชั่วคราวโดยที่ไม่จำเป็นต้องอาศัยโครงสร้างพื้นฐานและไม่จำเป็นต้องมีศูนย์กลางการควบคุม ด้วยความยืดหยุ่นนี้ จึงทำให้สามารถสร้างเครือข่ายที่ใดและเมื่อใดก็ได้ トラบโดที่มีผู้ใช้อย่างน้อย 2 คนต้องการทำติดต่อสื่อสารกัน และเนื่องจากโหนดมีการเคลื่อนย้ายที่ได้ จึงทำให้เกิดการเปลี่ยนแปลงโทโพโลยี ขึ้นได้ตลอดเวลา ส่งผลให้เส้นทางในการรับส่งข้อมูลขาดได้ ความน่าเชื่อถือของการสื่อสารลดลง

เพื่อแก้ปัญหาเช่นนี้ จึงมีความสนใจที่จะพัฒนาโพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทาง

โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทาง จะช่วยพัฒนาประสิทธิภาพและความน่าเชื่อถือ เพราะว่าเมื่อเส้นทางเกิดขาด ก็จะมีเส้นทางสำรองให้ใช้แทนได้ทันที โดยไม่จำเป็นต้องทำกระบวนการค้นหาเส้นทางใหม่ ทำให้ช่วยลดเวลาที่ใช้ในการส่งข้อมูลตรงส่วนนั้นได้ นอกจากนี้โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทางยังมีการใช้ทรัพยากรคุ่มค่ามากกว่าโพรโทคอลการค้นหาเส้นทางแบบเส้นทางเดียว เนื่องจากในการค้นหาเส้นทางแบบเส้นทางเดียวนั้น ยังมีอีกหลายเส้นทางที่สามารถเลือกใช้ส่งข้อมูลไปยังปลายทางได้แต่ไม่ได้ถูกนำมาใช้

นอกจากนั้นโพรโทคอลปัจจุบันยังขาดประสิทธิภาพในการกระจายภาระงาน ทำให้คุณภาพของการบริการลดลง โดยเฉพาะในเครือข่ายที่มีปริมาณข้อมูลมาก เนื่องจากข้อมูลจะไปหนาแน่นที่บางโหนด ส่งผลให้เกิดความคับคั่งของเครือข่าย ก่อให้เกิดผลกระทบที่ไม่น่าพึงพอใจหลายอย่าง เช่น เวลาที่ใช้ในการส่งข้อมูลสูง อัตราการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำเร็จต่ำ โอเวอร์เฮดสูง และยังใช้ทรัพยากรของเครือข่ายอย่างไม่มีประสิทธิภาพ

ทางผู้จัดทำจึงได้ทำการศึกษา ค้นคว้า และนำเสนอโพรโทคอลที่มีการใช้เทคนิคกระจายภาระงานเพื่อปรับปรุงประสิทธิภาพไม่ให้โหนดนั้น ๆ มีการใช้เส้นทางที่คับคั่ง โดยจะมีการกำหนดว่าจะยอมให้มีการสร้างเส้นทางผ่านโหนดนั้น ๆ ไปยังโหนดปลายทางหรือไม่ มีการนำเสนอแนวคิดที่จะใช้หลักของความน่าจะเป็นมาเป็นตัวชี้วัดในการจำกัดการสร้างเส้นทางของแต่ละโหนด ซึ่งโพรโทคอลที่ทางผู้จัดทำใช้ในการศึกษาและทดลองคือโพรโทคอล AOMDV เนื่องจากเป็นโพรโทคอลที่ถูกออกแบบมาเพื่อลดโอเวอร์เฮด โดยเฉพาะ [1] ดังนั้นจึงมีโอเวอร์เฮดน้อยกว่าโพรโทคอลในการค้นหาเส้นทางอื่น ๆ

2. งานวิจัยที่เกี่ยวข้อง

ในส่วนนี้จะกล่าวถึงทฤษฎีและงานวิจัยต่าง ๆ ที่เกี่ยวข้องในการพัฒนาโพรโทคอลค้นหาเส้นทางในสภาพแวดล้อมหลายเส้นทาง รวมไปถึงงานวิจัยที่เกี่ยวข้องกับโพรโทคอลที่มีการใช้เทคนิคการบาลานซ์ภาระงาน

2.1. Ad-hoc On demand Multipath Distance Vector (AOMDV)

โพรโท AOMDV [1] เป็นโพรโทคอลที่จัดทำขึ้นเพื่อมีจุดประสงค์ที่จะลดโอเวอร์เฮดในการค้นหาเส้นทาง เป็นการค้นหาเส้นทางที่เริ่มจากต้นทาง (source initiate) คือเมื่อต้นทางมีข้อมูลที่ต้องการจะส่ง จะเริ่มกระบวนการค้นหาเส้นทางโดยส่ง RREQ เพื่อหาเส้นทางไปยังปลายทาง ซึ่งการค้นหาเส้นทางแต่ละครั้งจะใช้เวลาและโอเวอร์เฮดสูง ดังนั้นเพื่อที่จะเพิ่มประสิทธิภาพของ on-demand routing จึงไม่ควรจะทำการค้นหาเส้นทางบ่อย ๆ ซึ่งการใช้โพรโทคอลการค้นหาเส้นทางแบบหลายเส้นทางจะลดจำนวนการค้นหาเส้นทางได้มากกว่าโพรโทคอลการค้นหาเส้นทางแบบทางเดียว

โหนดต้นทางจะค้นหาเส้นทางไปยังปลายทางหลายเส้นทางภายในการบรอดแคสต์ RREQ หนึ่งครั้ง เพราะฉะนั้นจะทำการค้นหาเส้นทางใหม่อีกครั้งก็ต่อเมื่อเส้นทางเหล่านั้นใช้การไม่ได้หมดแล้ว ต่างจากโพรโทคอลการค้นหาเส้นทางแบบทางเดียว ที่ถ้าหากเส้นทางเสียหาย จะต้องทำการค้นหาเส้นทางใหม่ทุกครั้ง ดังนั้นทำให้โพรโทคอลการค้นหา

เส้นทางแบบหลายเส้นทาง สามารถลดจำนวนการค้นหาเส้นทางได้ ทำให้มีโอเวอร์เฮดน้อยกว่า

กระบวนการค้นหาเส้นทางของโพรโทคอล AOMDV จะทำการค้นหาหลายเส้นทางจากโหนดต้นทางไปยังโหนดปลายทาง ซึ่งเส้นทางจะเป็นเส้นทางที่ไม่มีการเกิดลูปและเป็นเส้นทางที่ไม่มีการใช้ลิงค์ ร่วมกัน

ในกระบวนการค้นหาเส้นทางจะเริ่มขึ้นเมื่อโหนดต้นทางต้องการสร้างเส้นทางไปยังปลายทาง โดยจะบรอดแคสต์ RREQ ออกไปและรอรับ RREP แต่ละโหนดจะตั้ง next hop ของ reverse path เป็นฮอปก่อนหน้าที่ส่ง RREQ มาให้มัน ถ้าหากโหนดมีเส้นทางไปยังปลายทางก็จะตอบกลับ RREP ไปยังต้นทางแบบยูนิแคสกลับไปทาง reverse path ถ้าหากได้รับ RREP ซ้ำก็จะทำการละทิ้งทิ้ง เมื่อปลายทางได้รับ RREQ ก็จะทำการตอบกลับไปที่ต้นทางทาง reverse path เมื่อต้นทางได้รับ RREP ก็จะทำการสร้างเส้นทาง forward path ได้

ทุกโหนดจะมีการประกาศ hop count ไปยังแต่ละปลายทาง โดยภายในนั้นจะมีความยาวของเส้นทางไปยังปลายทาง โดยครั้งแรกจะประกาศเส้นทางที่ยาวที่สุดไปยังปลายทางด้วยค่า destination sequence number หนึ่ง ค่า advertised hop count จะเปลี่ยนแปลงเมื่อ sequence number เปลี่ยน การประกาศเส้นทางที่ยาวที่สุดทำให้เพิ่มโอกาสในการรับได้หลายเส้นทางเนื่องจากโหนดจะยอมรับเส้นทางที่ประกาศจากโหนดเพื่อนบ้าน เฉพาะในกรณีที่มี hop count ต่ำกว่าที่เคยได้รับประกาศมาเท่านั้น

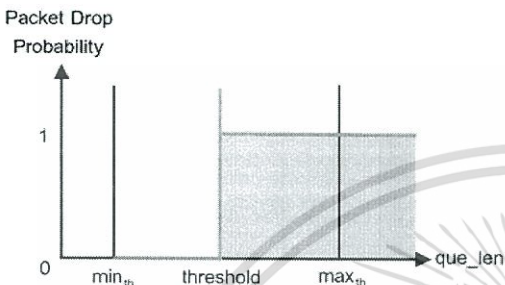
AOMDV จะมีกลไกในการเลือกเส้นทางที่เป็น disjoint path คือ แต่ละเส้นทางจากโหนดต้นทางไปยังโหนดปลายทางเป็นเส้นทางที่ไม่มีการใช้ลิงค์เดียวกันเลย จะเกิดขึ้นได้เมื่อโหนดมี next hop ต่างกันและ last hop (โหนดสุดท้ายก่อนถึงปลายทาง) ต่างกันเท่านั้น ซึ่งจะต้องมีการเก็บค่า last hop ของทุก ๆ เส้นทางใน routing table และภายใน RREQ และ RREP ก็ต้องมีข้อมูล last hop ด้วย

2.2 A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network (WAL)

โพรโทคอล WAL [2] มีการกระจายภาระงานในการสร้างเส้นทาง เพื่อลดความคับคั่งของโหนด โดยแต่ละโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นต้นการคำ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสามารถเลือกส่ง route request (RREQ) ได้ จะมีแค่ โหนดที่ไม่คับคั่งเป็นโหนดระหว่างทางในการส่งต่อ RREQ โหนดที่อยู่ในสถานะที่มีภาระงานมากเกินไป (overload) จะไม่ยอมให้มีการสร้างเส้นทางผ่านตนเองเพิ่ม โดยจะไม่ส่งต่อ RREQ เป็นระยะเวลาหนึ่ง โดยโหนดจะกลับมาส่งต่อ RREQ เพื่อสร้างเส้นทางสร้างผ่านตนเองได้อีกก็ต่อเมื่อโหนดนั้นไม่อยู่ในสถานะที่มีภาระงานมากเกินไปแล้ว



รูปที่ 1. กลไกการทำงานของโพรโทคอล WAL

จากรูปที่ 1 โพรโทคอล WAL จะมีการกำหนดค่า threshold ที่น้อยที่สุด (min_{th}) และ ค่า threshold ที่มากที่สุด (max_{th}) เอาไว้ เมื่อมี RREQ เข้ามายังโหนด จะมีการตรวจสอบขนาดคิวว่า เกินค่า min_{th} หรือไม่ ถ้าไม่เกิน จะทำการส่ง RREQ นั้นทันที เป็นการยอมให้สร้างเส้นทางผ่านโหนดนั้น ๆ และตรวจสอบขนาดคิวว่า เกินค่า max_{th} หรือไม่ ถ้าเกินก็จะทำการละทิ้ง RREQ จะไม่อนุญาตให้สร้างเส้นทางผ่านเพราะในขณะนั้นมีความคับคั่งอยู่ แต่ถ้าหากอยู่ระหว่าง min_{th} และ max_{th} จะต้องใช้ค่า threshold เป็นตัวกำหนดว่าจะยอมให้สร้างเส้นทางผ่านโหนดนั้นได้หรือไม่ ถ้าขนาดคิวไม่เกินค่า threshold ก็จะอนุญาตให้ส่ง RREQ ต่อ แต่ถ้าขนาดคิวมีค่าเกิน threshold จะทำการละทิ้ง RREQ นั้น ซึ่งค่า threshold นี้จะมีการเปลี่ยนแปลงและมีการอัปเดตค่า threshold เสมอเมื่อมี RREQ เข้ามายังโหนดนั้น ๆ โดยค่าเริ่มแรกของ threshold จะถูกกำหนดไว้ที่ค่า max_{th} จนกระทั่งเกิดความคับคั่งของข้อมูลในโหนดนั้นหรือเกิดภาระงานมากเกินไป ตาม 2 เงื่อนไข ดังนี้ (1) ขนาดของคิวในโหนดนั้นเริ่มเพิ่มมากขึ้น จนเกินค่าเฉลี่ยของค่า min_{th} และ max_{th} (2) ระยะเวลาของงานที่เหลืออยู่ มีค่ามากกว่าค่าที่กำหนดไว้เพื่อเป็นตัววัดสถานะภาระงาน ซึ่งระยะเวลาของงานที่เหลืออยู่มีหน่วยเป็นวินาที จะเริ่มนับเมื่อขนาดของคิวของโหนดนั้นเพิ่มมากขึ้นจนเกินค่าเฉลี่ยของ min_{th} และ max_{th}

เมื่อเกิดสถานะที่มีภาระงานมากเกินไป threshold จะปรับลดลงมาเป็นระยะ $thres_{dec}$ (ค่าที่กำหนดว่าจะลดลง threshold ครั้งละเท่าไร) ถ้าคิวที่มีอยู่ยังคงเกิน threshold จะไม่อนุญาตให้สร้างเส้นทางผ่านได้อีกเพราะสถานะในตอนนี้มีความคับคั่งอยู่แล้ว และจะลดทุกครั้งที่ได้รับ RREQ จนกระทั่ง threshold ไปหยุดอยู่ที่ min_{th} เป็นระยะเวลาหนึ่ง จะหมายความว่า คิวที่มีอยู่ของโหนดได้ทำการส่งออกจนไม่คับคั่งแล้ว เมื่อนั้นค่า threshold จะกลับไปอยู่ที่ค่าเริ่มต้นหรือที่ max_{th} อีกครั้ง

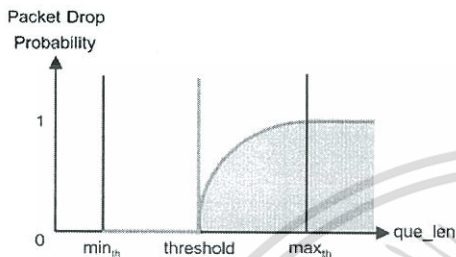
แม้ว่าการละทิ้ง RREQ จะช่วยในเรื่องการกระจายภาระงาน แต่ถ้าหากว่าละทิ้งมากเกินไปอาจทำให้การหาเส้นทางไปยังโหนดปลายทางไม่สามารถทำได้ จึงกำหนดให้ใช้ priority flag คือ ถ้าหากโหนดต้นทางส่ง RREQ ไปแล้วยังไม่ได้รับ route reply (RREP) ไปจนครบเวลา timeout โหนดนั้น จะทำการส่ง RREQ ใหม่โดยตั้ง flag เป็น 1 เมื่อโหนดระหว่างทางได้รับ RREQ แม้ว่าโหนดนั้นมีขนาดคิวเกินกว่าค่า threshold โหนดก็จะยอมส่ง RREQ นั้นต่อให้ เพื่อให้โหนดต้นทางได้เส้นทางในการส่งข้อมูล

3. สิ่งที่น่าสนใจ

จากการศึกษางานวิจัย [1], [2], [3] และ [4] โดยนำผลมาเปรียบเทียบในหลาย ๆ ด้าน เพื่อแสดงให้เห็นถึงความแตกต่างของโพรโทคอลที่ไม่มีการกระจายภาระงาน และโพรโทคอลที่มีการกระจายภาระงาน โดยทำการวัดประสิทธิภาพในด้านต่าง ๆ ดังนี้ เวลาที่ใช้ในการส่งข้อมูล อัตราการส่งข้อมูลสำเร็จ โอเวอร์เฮด เวลาที่ใช้ในการสร้างเส้นทาง และจำนวนแพคเกจข้อมูลที่ส่งผ่านในแต่ละโหนด เพื่อแสดงให้เห็นถึงการเลือกสร้างเส้นทาง อัตราการละทิ้งแพคเกจ และ ค่าความเบี่ยงเบนของการสร้างเส้นทาง พบว่าการใช้ขนาดคิวเป็นปัจจัยในการเลือกสร้างเส้นทาง และเป็นตัวชี้วัดที่บ่งบอกความคับคั่งของโพรโทคอล WAL มีประสิทธิภาพดีกว่าการใช้ Round Trip Time (RTT) ของโพรโทคอล AODV เนื่องจาก RTT เป็นตัวชี้วัดที่มีการเก็บสะสมค่าจากค่าก่อนหน้า จึงไม่สามารถบ่งบอกได้ชัดเจนว่ามีความคับคั่งหรือไม่ ดังนั้นเส้นทางที่มีดีเลย์น้อย ไม่ได้หมายความว่า จะมีความคับคั่งน้อยที่สุดเสมอไป แต่ขนาดคิวสามารถบ่งบอกความคับคั่ง ณ ขณะนั้นได้ชัดเจนกว่า ดังนั้นผู้วิจัยจึงสนใจแนวคิดของโพรโทคอล WAL ที่ใช้ RTT เป็นตัวชี้วัดในการเลือกเส้นทางเพื่อหลีกเลี่ยงเส้นทางที่คับคั่งจะได้ผลไม่แน่นอนถ้าหากการใช้ขนาดคิวที่วัดก่อนการเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าวิจัยเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

เส้นทาง และได้สังเกตเห็นว่ายังมีโอกาสที่จะพัฒนาประสิทธิภาพทางด้านเวลาในการสร้างเส้นทาง (RET) และเวลาที่ใช้ในการส่งข้อมูล (EE) โดยไม่กระทบต่อประสิทธิภาพในด้านอื่น ๆ ซึ่งก็ยังมีแนวคิดให้โหนดมีความยืดหยุ่นยอมให้มีการสร้างเส้นทางผ่านโหนดนั้นได้อีกแม้ว่าจะมีขนาดคิวเกินค่า threshold แล้ว แต่ยังไม่เกินค่า max_{th}



รูปที่ 2. กลไกการทำงานของโพรโทคอลที่นำเสนอ

ผู้วิจัยจึงได้นำเสนอการประยุกต์ใช้ค่าความน่าจะเป็นมาเป็นตัวชี้วัดในการละทิ้ง RREQ เพื่อจำกัดการสร้างเส้นทางไม่ให้โหนดนั้น ๆ มีความคับคั่ง ซึ่งหลักการทำงานจะคล้ายกับโพรโทคอล WAL แต่ต่างกันตรงที่มีการใช้ความน่าจะเป็นในการละทิ้งแพ็คเกจเพิ่มเข้าไปในการทำงานของโพรโทคอล WAL ที่เมื่อขนาดคิวของโพรโทคอล WAL มีขนาดเกินค่า threshold จะทำการละทิ้ง RREQ ทันที การกระทำเช่นนี้จะทำให้ตัดโอกาสการสร้างเส้นทาง และสำหรับกรณีที่เครือข่ายมีสถานะที่โหนดส่วนมากในเครือข่ายมีความคับคั่งจะทำให้โหนดต้นทางไม่สามารถสร้างเส้นทางไปยังปลายทางได้ เนื่องจากขนาดคิวของโหนดส่วนมากเกินค่า threshold ส่งผลให้มี RREQ ส่งไปถึงปลายทาง จึงทำให้โหนดต้นทางต้องส่ง RREQ เพื่อขอสร้างเส้นทางอีกครั้ง โดยที่ส่ง RREQ จะมีค่า flag เป็น 1 ซึ่งจะหมายถึงว่าแม้โหนดระหว่างทางที่ได้รับ RREQ นั้นจะมีขนาดคิวเกินค่า threshold ก็จะไม่ส่งต่อ RREQ นั้นเสมอ ซึ่งในกรณีเช่นนี้ จะทำให้ใช้เวลาในการสร้างเส้นทางมากขึ้น และส่งผลถึงเวลาที่ใช้ในการส่งข้อมูลอีกด้วย

สำหรับการประยุกต์โพรโทคอลที่นำเสนอ นั้น เมื่อโหนดระหว่างทางที่ได้รับ RREQ มีขนาดคิวเกินค่า threshold แล้ว แต่ยังไม่เกินค่า max_{th} จะไม่ทำการละทิ้ง RREQ นั้นทันที แต่จะทำการคำนวณค่าความน่าจะเป็นในการละทิ้งจากสมการลอการิทึม ดังเช่นรูปที่ 2 ความน่าจะเป็นในการละทิ้งจะแปรผันตามขนาดคิว คือ ถ้าหากขนาดคิวเกินค่า

threshold ไม่มาก โอกาสที่ RREQ จะถูกละทิ้งจะมีน้อย และถ้าหากขนาดคิวเข้าใกล้ค่า max_{th} โอกาสถูกละทิ้ง RREQ ก็จะมีมากขึ้น จนกระทั่งเมื่อขนาดคิวเกินค่า max_{th} ความน่าจะเป็นที่ RREQ จะถูกละทิ้งจะเท่ากับ 1 หมายความว่าเมื่อขนาดคิวเกินค่า max_{th} แล้ว RREQ จะถูกละทิ้งเสมอ (ไม่ยอมให้มีการสร้างเส้นทางผ่าน) เพื่อควบคุมไม่ให้มีการรับสร้างเส้นทางผ่านมากเกินไป เพื่อป้องกันการละทิ้งแพ็คเกจที่เกิดจากขนาดคิวเต็ม ผู้วิจัยจึงเชื่อว่าการดำเนินการเช่นนี้จะช่วยลดเวลาในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลลงได้

4. การทดลองและประเมินผล

จากการศึกษางานวิจัยที่เกี่ยวข้อง ทางผู้จัดทำจึงได้ทำการทดสอบประสิทธิภาพการทำงานของโพรโทคอลต่าง ๆ และนำมาเปรียบเทียบประสิทธิภาพ 3 โพรโทคอล ดังนี้ AOMDV เป็นโพรโทคอลพื้นฐาน AOMDV-WAL คือ โพรโทคอลที่นำเทคนิค WAL มาประยุกต์ใช้กับโพรโทคอล AOMDV ซึ่งใช้ขนาดคิวเป็นตัวตัดสินใจละทิ้ง RREQ เพื่อควบคุมการสร้างเส้นทาง Proposed Protocol คือ โพรโทคอลที่พัฒนามาจากโพรโทคอล WAL โดยเพิ่มความยืดหยุ่นในการรับสร้างเส้นทางในกรณีที่โหนดมีความคับคั่งของภาระงาน

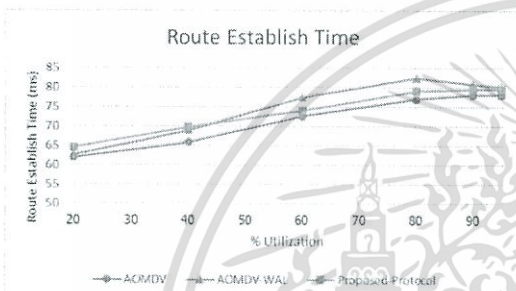
ทางผู้จัดทำได้ใช้ NS-2 (Network simulator) ในการจำลองเครือข่าย โดยกำหนดให้ใช้แบนวิธเท่ากับ 2 Mbps และใช้โพรโทคอล AOMDV เป็นโพรโทคอลพื้นฐาน จำลองเปรียบเทียบประสิทธิภาพระหว่าง โพรโทคอล AOMDV ที่ใช้เทคนิค WAL และโพรโทคอลที่นำเสนอ โดยมีบัฟเฟอร์ที่ฝั่งส่งขนาด 64 แพ็คเกจเพื่อใช้สำหรับแพ็คเกจที่รอรับ RREQ โดยกำหนดเวลารอคอยไว้ที่ 30 วินาที ขนาดของแถวคอยจะกำหนดไว้ที่ 50 แพ็คเกจ ซึ่งแถวคอยจะให้ความสำคัญกับ routing protocol packet มากกว่าแพ็คเกจข้อมูล

เราจะใช้การจำลองทั้งหมด 20 ครั้ง ซึ่งแต่ละครั้งจะมีรูปแบบของการเคลื่อนที่ต่างกัน กำหนด pause time (ระยะเวลาที่โหนดหยุดก่อนเคลื่อนที่) ไว้ที่ 100 วินาที ความเร็วของโหนดเฉลี่ยอยู่ที่ 10 เมตรต่อวินาทีเครือข่ายจะประกอบด้วย 100 โหนด มีระยะทางในการส่งข้อมูล 250 เมตร ใช้ทั้งหมด 40 traffic flow ที่มีขนาดของแพ็คเกจคือ 512 ไบต์ โดยค่อย ๆ เพิ่มอัตราการใช้แบนวิธจากร้อยละ

เมตร เป็นระยะเวลา 500 วินาที (เวลาจำลอง)

ในการเปรียบเทียบประสิทธิภาพ ทางผู้วิจัยจะทำการวัดประสิทธิภาพในด้านต่าง ๆ ดังนี้ Packet Delivery Rate (PDR) คืออัตราของแพคเกจข้อมูลที่ถูกส่งได้สำเร็จ ต่อจำนวนแพคเกจที่โหนดต้นทางส่งออกทั้งหมด Routing Overhead (OH) คือจำนวนโอเวอร์เฮดต่อหนึ่งแพคเกจข้อมูล Route Establish Time (RET) คือเวลาที่ใช้ในการสร้างเส้นทาง (ms) End-to-end delay (EE) คือเวลาเฉลี่ยที่ใช้ในการรับส่งข้อมูลจากต้นทางไปถึงปลายทาง (ms)

4.1 ผลการทดลอง

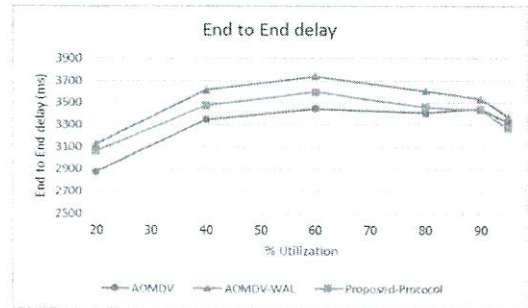


รูปที่ 3. แสดงการเปรียบเทียบเวลาที่ใช้ในการสร้างเส้นทาง

จากรูปที่ 3 แสดงผลลัพธ์จากการทดลองในด้านเวลาในการสร้างเส้นทาง โดยแสดงเป็นกราฟเทียบระหว่างเวลาในการสร้างเส้นทางซึ่งมีหน่วยเป็นมิลลิวินาที (ms) กับร้อยละ (%) ของการใช้แบนวิธ ซึ่งพบว่าโปรโตคอล AOMDV จะใช้เวลาในการสร้างเส้นทางต่ำที่สุด เนื่องจากโปรโตคอล WAL และโปรโตคอลที่นำเสนอ นั้น มีการควบคุมการสร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้น ๆ ส่งผลให้ต้องใช้เวลาเพิ่มขึ้นในการหาเส้นทางไปยังโหนดที่ไม่มีความคับคั่งแทน

เมื่อเปรียบเทียบโปรโตคอลที่นำเสนอกับโปรโตคอล WAL พบว่า โปรโตคอลที่นำเสนอ นั้นใช้เวลาในการสร้างเส้นทางต่ำกว่าโปรโตคอล WAL เนื่องจาก การที่โปรโตคอลที่นำเสนอ นั้นมีความยืดหยุ่นในการรับสร้างเส้นทางผ่านมากกว่า คือถ้าหากโหนดนั้นมีความคับคั่งเล็กน้อยก็จะยอมเป็นโหนดทางผ่านให้อยู่ ซึ่งถ้าหากเป็นโปรโตคอล WAL จะทำการปฏิเสธการเป็นโหนดทางผ่านทันที ซึ่งในกรณีที่มีส่วนมากในเครือข่ายมีความคับคั่งแล้ว อาจทำให้โหนดต้นทางในโปรโตคอล WAL นั้นหาเส้นทางไปยังปลายทางไม่ได้

จนทำให้ RREQ นั้น timeout และต้องทำการส่ง RREQ อีกครั้ง ส่งผลให้ใช้เวลาในการสร้างเส้นทางสูงขึ้น



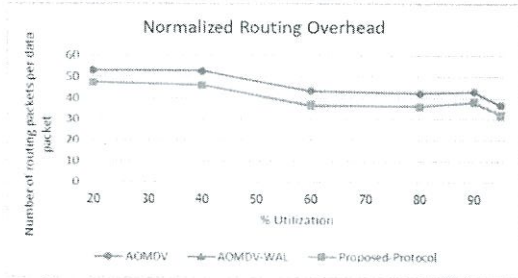
รูปที่ 4. แสดงการเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูล

จากรูป 4 แสดงเวลาเฉลี่ยที่ใช้ในการส่งหนึ่งแพคเกจข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง ซึ่งจะเห็นว่าโปรโตคอล AOMDV จะใช้เวลาในการส่งแพคเกจข้อมูลต่ำกว่าโปรโตคอล WAL และโปรโตคอลที่นำเสนอในช่วงแรก เนื่องจาก โปรโตคอล WAL และโปรโตคอลที่นำเสนอ นั้น มีการควบคุมการสร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้น ๆ ส่งผลให้ต้องใช้เวลาในการสร้างเส้นทางเพิ่มขึ้น แต่จะเห็นว่าในช่วงที่มีการใช้แบนวิธมาก ๆ โปรโตคอลที่นำเสนอ จะใช้เวลาในการส่งข้อมูลต่ำกว่า AOMDV เนื่องจากการที่ความเร็วในการส่งข้อมูลสูง จะทำให้โหนดระหว่างทางมีความคับคั่งมาก การที่ควบคุมการสร้างเส้นทางจึงเป็นผลดี ทำให้สามารถใช้เส้นทางที่ไม่มี ความคับคั่งได้ และยังมี ความยืดหยุ่นในการยอมให้สร้างเส้นทางดีกว่าโปรโตคอล WAL ซึ่งช่วยลดการละทิ้ง RREQ ได้เหมาะสมกว่า

แต่เมื่อเทียบโปรโตคอลที่นำเสนอ กับโปรโตคอล WAL จะเห็นว่าโปรโตคอลที่นำเสนอ นั้นใช้เวลาเฉลี่ยในการส่งแพคเกจข้อมูลต่ำกว่า เนื่องจากเวลาเฉลี่ยที่ใช้ในการส่งข้อมูลนั้นจะหมายถึงเวลาตั้งแต่โหนดต้นทางค้นหาเส้นทาง ไปจนถึงเวลาที่แพคเกจข้อมูลถูกส่งไปถึงโหนดปลายทาง ซึ่ง จะรวมถึงเวลาที่ค้นหาเส้นทางใหม่ในกรณีที่เส้นทางขาดหรือเสียหายด้วย สำหรับการจำลองเครือข่ายในสภาวะแวดล้อมที่โหนดมีการเคลื่อนไหวอยู่ตลอดเวลา จะทำให้เส้นทางในการส่งข้อมูลเสียหายบ่อยครั้ง ถ้าในขณะที่โหนดมีการส่งข้อมูลอยู่แล้วเส้นทางเกิดความเสียหายขึ้น โหนดต้นทางจะต้องทำการค้นหาเส้นทางใหม่ ซึ่งโปรโตคอล WAL ใช้เวลาในการค้นหาเส้นทางมากกว่าโปรโตคอลที่นำเสนออยู่แล้ว ยิ่งเส้นทางเสียหายบ่อยครั้ง ยิ่งทำให้โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

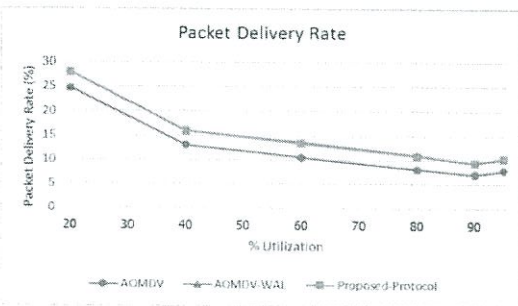
ต้องค้นหาเส้นทางใหม่ ส่งผลให้เวลาในการส่งข้อมูลสูงกว่า โพรโทคอลที่นำเสนอ



รูปที่ 5. แสดงการเปรียบเทียบโอเวอร์เฮดต่อหนึ่งแพ็คเกจข้อมูล

จากรูป 5 แสดงจำนวนแพ็คเกจที่ใช้ในกลไกค้นหาเส้นทางต่อหนึ่งแพ็คเกจข้อมูลที่ส่งถึงปลายทาง แสดงให้เห็นว่า โพรโทคอล WAL และโพรโทคอลที่นำเสนอนั้นมีจำนวนโอเวอร์เฮดต่อหนึ่งแพ็คเกจข้อมูลต่ำกว่าโพรโทคอล AOMDV เนื่องจากการที่มีการจำกัดเส้นทางของกลไกการค้นหาเส้นทางที่ไม่ยอมให้โหนดที่มีภาระงานมากอยู่แล้ว ยอมรับเป็นทางผ่านส่งต่อ RREQ ทำให้โหนดที่มีภาระงานมากนั้นละทิ้ง RREQ จึงทำให้สามารถลดจำนวน RREQ ได้อย่างมีประสิทธิภาพ และโอเวอร์เฮดส่วนใหญ่ของโพรโทคอล AOMDV นั้นมาจาก RREQ จึงเป็นการลดจำนวนโอเวอร์เฮดที่ไม่จำเป็นภายในเครือข่ายลงได้ โดยไม่ส่งผลกระทบต่อประสิทธิภาพด้านอื่น ๆ

เมื่อเปรียบเทียบระหว่างโพรโทคอลที่นำเสนอกับโพรโทคอล WAL แล้ว พบว่าโพรโทคอลที่นำเสนอมีจำนวนโอเวอร์เฮดสูงกว่าเล็กน้อย เนื่องมาจากการที่โพรโทคอลที่นำเสนอมีความยืดหยุ่นยอมให้ส่งต่อ RREQ แม้ขณะนั้นโหนดเริ่มมีความคับคั่งขึ้นแล้วโดยการสุ่มความน่าจะเป็นตามสมการ ซึ่งก็ยังมีกรจำกัดการสร้างเส้นทางอยู่ จึงทำให้มีบางช่วงที่จำนวนโอเวอร์เฮดสูงกว่าโพรโทคอล WAL เพียงเล็กน้อย



รูปที่ 6. แสดงการเปรียบเทียบอัตราการส่งแพ็คเกจสำเร็จ

จากรูปที่ 6 อัตราการส่งแพ็คเกจสำเร็จ คืออัตราการของแพ็คเกจที่ถูกส่งได้สำเร็จต่อแพ็คเกจที่โหนดต้นทางส่งทั้งหมด จะแสดงให้เห็นว่าทั้งโพรโทคอล WAL และโพรโทคอลที่นำเสนอมีอัตราการส่งข้อมูลได้สำเร็จสูงกว่า AOMDV มาก เนื่องจากการที่โหนดได้ใช้เส้นทางที่มีความคับคั่งของข้อมูลน้อยกว่าจะทำให้ลดโอกาสในการที่แพ็คเกจจะถูกละทิ้ง เนื่องจากขนาดคิวของโหนดระหว่างทางเต็ม และเมื่อเทียบประสิทธิภาพในการส่งข้อมูลได้สำเร็จของโพรโทคอล WAL และโพรโทคอลที่นำเสนอพบว่าอัตราการส่งข้อมูลได้สำเร็จไม่แตกต่างกันมากนัก

5. สรุป

จากการศึกษาและวิเคราะห์ประสิทธิภาพของโพรโทคอลสำหรับเครือข่ายเฉพาะกิจในสภาพแวดล้อมหลายเส้นทาง ผู้จัดทำได้เลือกพัฒนาโพรโทคอล WAL เนื่องจากมีการใช้ขนาดคิวเป็นปัจจัยในการเลือกสร้างเส้นทางและใช้เป็นตัวเลือกสร้างเส้นทาง ซึ่งมีประสิทธิภาพดีกว่าการใช้ Round Trip Time (RTT) ของโพรโทคอล AOMDV (การใช้เส้นทางตาม RREQ แรกที่มาถึง) ผู้จัดทำจึงสนใจที่จะพัฒนาประสิทธิภาพของโพรโทคอล WAL และเห็นว่าในการทำงานของโพรโทคอล WAL มีการเลือกเส้นทางที่ไม่ยืดหยุ่น คือจะไม่ยอมให้มีการสร้างเส้นทางผ่านโหนดที่มีความคับคั่ง ทำให้โหนดต้นทางต้องใช้เวลาในการค้นหาเส้นทางเพิ่มขึ้น ส่งผลให้เวลาเฉลี่ยในการส่งข้อมูลไม่ดีเท่าที่ควร จึงมีแนวคิดที่จะลดเวลาในการค้นหาเส้นทางและปรับปรุงเวลาที่ใช้ในการส่งแพ็คเกจข้อมูล โดยมีแนวคิดคือจะให้โหนดระหว่างทางจะมีความยืดหยุ่นในการรับเป็นทางผ่านมากขึ้น คือเมื่อขนาดคิวเกินค่า threshold แล้วจะไม่ละทิ้ง RREQ ทันที แต่จะทำการสุ่มว่าจะรับเป็นทางผ่านหรือไม่ โดยความน่าจะเป็นในการสุ่มจะขึ้นอยู่กับปริมาณขนาดคิว ถ้าหากขนาดคิวมาก เข้าใกล้ค่า max_{th} (ค่าสูงสุดที่จะยอมให้สร้างเป็นทางผ่าน) โอกาสที่ RREQ จะถูกละทิ้งจะมีมาก และถ้าหากขนาดคิวมากกว่าค่า threshold แต่อย่างน้อยอยู่ก็จะมีโอกาสที่จะละทิ้ง RREQ ต่ำ ซึ่งจะทำให้โหนดต้นทางมีโอกาสได้เส้นทางผ่านโหนดนั้น ๆ เป็นการลดเวลาในการสร้างเส้นทางลงได้

จากผลการทดลองพบว่าในช่วงที่มีอัตราการส่งข้อมูลต่ำ เวลาที่ใช้ในการสร้างเส้นทางและเวลาที่ใช้ในการส่งข้อมูลของโพรโทคอลที่นำเสนอกับโพรโทคอล WAL ใช้เวลามากกว่าโพรโทคอล AOMDV เนื่องจากการควบคุมการ

สร้างเส้นทางไม่ให้สร้างเส้นทางผ่านไปยังโหนดที่มีปริมาณข้อมูลส่งผ่านมากอยู่แล้ว จึงทำให้อาจถูกปฏิเสธไม่ให้ผ่านโหนดนั้น ๆ ส่งผลให้ต้องใช้เวลาในการสร้างเส้นทางเพิ่มขึ้น และเมื่อถูกปฏิเสธการสร้างเส้นทางผ่านโหนดนั้น ๆ จะต้องทำการค้นหาเส้นทางและส่งข้อมูลใหม่อยู่ตลอด ซึ่งก็ส่งผลให้เวลาที่ใช้ในการส่งเส้นทางเพิ่มขึ้นด้วย แต่เมื่อมีการใช้อัตราเร็วในการส่งข้อมูลสูง โพรโทคอล AOMDV จะใช้เวลาในการสร้างเส้นทางและส่งข้อมูลสูงกว่าโพรโทคอลที่นำเสนอ ซึ่งการที่มีการควบคุมการสร้างเส้นทางก็ยังส่งผลกระทบถึงค่าโอเวอร์เฮด ทำให้ค่าโอเวอร์เฮดของโพรโทคอลที่นำเสนอกับโพรโทคอล WAL ต่ำลง เนื่องจากทำให้โหนดที่มีภาระงานมากนั้นละทิ้ง RREQ จึงทำให้สามารถลดจำนวน RREQ ได้อย่างมีประสิทธิภาพ และโอเวอร์เฮดส่วนใหญ่ของโพรโทคอล AOMDV นั้นมาจาก RREQ นอกจากนี้เนื่องจากการที่โหนดได้ใช้เส้นทางที่มีความคับคั่งของข้อมูลน้อยกว่าจะทำให้ลดโอกาสในการที่แพคเกจจะถูกละทิ้งเนื่องจากขนาดคิวของโหนดระหว่างทางเต็ม จึงส่งผลให้อัตราการส่งข้อมูลสำเร็จ ของโพรโทคอลที่นำเสนอและโพรโทคอล WAL มีอัตราการส่งข้อมูลสำเร็จมากกว่าโพรโทคอล AOMDV

เอกสารอ้างอิง

- [1] C. E. Perkins, E. M. Belding-Royer, S. R. Das.
“Ad hoc On-Demand Distance Vector (AODV) Routing.” Internet Engineering Task Force (IETF) draft, 2003
- [2] Mahesh K. Marina, Samir R. Das.
“Ad hoc on-demand multipath distance vector routing.” Wirel. Commun. Mob. Comput. 2006
- [3] Young J. Lee, George F. Riley.
“A Workload-based Adaptive Load-Balancing Technique for Mobile Ad Hoc Network.” Wireless Communications and Networking Conference, 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้