

การวิเคราะห์สมรรถนะของโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายแอตฮอก
สำหรับยานพาหนะ โดยใช้สถาปัตยกรรมชื่อข้อมูล

นางสาวศศิธรมย์ เทียนน้อย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the Graduate School.

PERFORMANCE ANALYSIS OF DATA DISSEMINATION PROTOCOL IN
VEHICULAR AD HOC NETWORK BY USING NAMED DATA
ARCHITECTURE

Ms. Sasirom Tiennoy

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2016
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การวิเคราะห์สมรรถนะของโพรโทคอลการแจกจ่ายข้อมูล
ในโครงข่ายแอตสอกสำหรับยานพาหนะโดยใช้สถาปัตยกรรม
ชื่อข้อมูล

โดย

นางสาวศศิธรมย์ เทียนน้อย

สาขาวิชา

วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์
ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ศาสตราจารย์ ดร.วาทิต เบญจพลกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร)

.....กรรมการ
(รองศาสตราจารย์ ดร.เชาวน์ดิศ อิศวกุล)

.....กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร.ศุภกร สิทธิไชย)

ศศิธรมย์ เทียนน้อย : การวิเคราะห์สมรรถนะของโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายแอดฮอคสำหรับยานพาหนะ โดยใช้สถาปัตยกรรมชื่อข้อมูล. PERFORMANCE ANALYSIS OF DATA DISSEMINATION PROTOCOL IN VEHICULAR AD HOC NETWORK BY USING NAMED DATA ARCHITECTURE อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร. ชัยเชษฐ์ สบายวิจิตร, 110 หน้า.

ปัจจุบันงานวิจัยในโครงข่ายแบบไร้สายที่เกี่ยวข้องกับโครงข่ายแอดฮอคสำหรับยานพาหนะได้รับความสนใจเป็นจำนวนมาก เนื่องจากยานพาหนะมีจำนวนบนท้องถนนเพิ่มมากขึ้น ส่งผลให้ต้องบริหารจัดการระบบการจราจรให้มีประสิทธิภาพเพื่อลดปัญหาต่าง ๆ ที่เกิดขึ้น เช่น การเกิดอุบัติเหตุและการจราจรติดขัด โดยสาเหตุของปัญหาที่สำคัญอย่างหนึ่งคือ การขาดการติดต่อสื่อสารระหว่างกันเพื่อแลกเปลี่ยนข้อมูลสภาพการจราจรแบบทันทีทันใด ด้วยสาเหตุนี้ การแลกเปลี่ยนข้อมูลระหว่างยานพาหนะกับยานพาหนะ หรือยานพาหนะกับโหนดข้างถนน จำเป็นต้องได้รับการประเมินและการแพร่กระจายอย่างถูกต้องภายในระยะเวลาที่เหมาะสมทันที เช่น ข้อมูลเตือนภัย ดังนั้นเวลาในการแจกจ่ายข้อมูลจึงมีความสำคัญมาก แต่เนื่องจากโครงข่ายแอดฮอคสำหรับยานพาหนะเป็นโครงข่ายที่มีสภาพแวดล้อมเป็นแบบเปลี่ยนแปลงตลอดเวลา ทำให้พบปัญหาการขาดความต่อเนื่องในการรับส่งข้อมูล อย่างไรก็ตาม โครงข่ายการสื่อสารในปัจจุบันที่ใช้โพรโทคอลแบบ TCP/IP ซึ่งใช้ที่อยู่ไอพีในการส่งข้อมูล อาจจะไม่เหมาะสมกับสภาพแวดล้อมที่โหนดมีลักษณะการเปลี่ยนแปลงตลอดเวลา เพราะโหนดต้องมีการคำนวณหาเส้นทางในการส่งข้อมูลใหม่ หลังจากที่เส้นทางขาดความต่อเนื่องในการรับส่งข้อมูล ดังนั้นสถาปัตยกรรมชื่อข้อมูล (named data architecture) หรือ โครงข่ายชื่อข้อมูล (named data network : NDN) เป็นสถาปัตยกรรมอินเทอร์เน็ตในอนาคตที่นำมาแก้ปัญหาของสถาปัตยกรรมอินเทอร์เน็ต TCP/IP ซึ่งมุ่งเน้นการให้บริการข้อมูลตามที่อยู่ของผู้ร้องขอ โดยไม่จำเป็นต้องสร้างเส้นทางก่อนการร้องขอข้อมูล อีกทั้งยังสามารถส่งการร้องขอข้อมูลด้วยชื่อออกไปและสามารถเก็บข้อมูลไว้ในขณะที่โครงข่ายมีจำนวนโหนดเบาบาง ในงานวิจัยนี้จึงได้เสนอโพรโทคอลการแจกจ่ายข้อมูลสำหรับโครงข่ายแอดฮอคสำหรับยานพาหนะ โดยใช้สถาปัตยกรรมชื่อข้อมูลซึ่งมีการติดตั้งโหนดข้างถนนเพื่อช่วยในการแลกเปลี่ยนข้อมูลการจราจรให้มีประสิทธิภาพ โพรโทคอลที่เสนอนี้เรียกว่า โครงข่ายชื่อข้อมูลโดยมีโหนดข้างถนนเป็นผู้ช่วย (roadside unit assisted of named data network : RA-NDN) ซึ่งกำหนดให้โหนดข้างถนนไม่ต้องเชื่อมต่อกับศูนย์กลางข้อมูลหรือศูนย์ประมวลผล และให้คุณลักษณะของโหนดข้างถนนเหมือนกันกับยานพาหนะ เช่น รัศมีการสื่อสาร เป็นต้น

ในการประเมินสมรรถนะของโพรโทคอลที่นำเสนอ โดยใช้โปรแกรมจำลองการสื่อสาร (NS3) และโปรแกรมจำลองการจราจร (SUMO) บนพื้นฐานข้อมูลจริงจากโครงการสาทรโมเดล ถนนสาทรกรุงเทพมหานคร ผลการจำลองที่ได้เมื่อเปรียบเทียบโพรโทคอลที่นำเสนอ (RA-NDN) กับโครงข่ายชื่อข้อมูลแบบดั้งเดิม (V-NDN) ภายใต้เงื่อนไขความหนาแน่นของยานพาหนะ รัศมีการสื่อสาร และจำนวนผู้ร้องขอ พบว่า โพรโทคอลที่นำเสนอสามารถลดปริมาณกราฟฟิกในระบบ และเวลาการแจกจ่ายข้อมูลโดยรวม อีกทั้งยังสามารถเพิ่มอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จและค่าวิสัยสามารถ

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อ.ที่ปรึกษาหลัก

5471425521 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS: VANET/ NDN/ ITS/ Caching.

SASIROM TIENNOY : PERFORMANCE ANALYSIS OF DATA DISSEMINATION PROTOCOL IN VEHICULAR AD HOC NETWORK BY USING NAMED DATA ARCHITECTURE. ADVISOR: ASST. PROF. CHAIYACHET SAIVICHIT, Ph.D., 110 pp.

Nowadays, vehicular ad hoc network (VANET) becomes one of highly active research area of wireless networking. In a road network where road information is exchanged among vehicles, all traffic information must be processed and disseminated in a timely manner. Since VANET is multi-hop wireless networks with extremely high mobility and short connection lifetimes, it is important to address the data dissemination problem in a partitioned and rapid changed topology. However, the current TCP/IP implementation regularly does not fit into the highly dynamic vehicular environments because the routed must be recalculated due to the intermittent connectivity. Hence, Name data networking (NDN) is considered here. NDN is a newly proposed architecture for the future Internet which focuses on delivering what data information drivers want. In this research, the data dissemination protocol for VANET using Named Data Architecture, which is deployed in a roadside unit to assist vehicles exchange their traffic information efficiently is proposed. The proposed protocol is called Roadside Unit Assisted of Named Data Network or RA-NDN. The RSU can operate on a standalone node without connecting to the data center, data collection or processing center, nor connect the other RSUs via wireless adhoc network. The characteristics of RSU are defined to be the same as the characteristics of vehicles such as its transmission range.

In order to verify the performance of the proposed protocol, this research uses the network simulator (NS3) and the traffic simulator (SUMO) software for this model. To reduce network latency under vehicular density, vehicular transmission range and the number of requesters, the proposed approach is compared with V-NDN via a real-world data set in the urban area of Sathorn road in Bangkok, Thailand. Simulation results show that using the proposed protocol with the roadside unit decreases traffic load, total dissemination time and increases the data received ratio and throughput.

Department : Electrical Engineering
Field of Study : Electrical Engineering

Student's Signature
Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลืออย่างมากของผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งกรุณาให้ความรู้ ให้คำแนะนำต่าง ๆ ที่เป็นประโยชน์ในการวิจัยทั้งทางตรงและทางอ้อม อีกทั้งตรวจสอบและแก้ไข รวมไปถึงคำวิจารณ์ในเชิงสร้างสรรค์ ตลอดจนความเมตตา และเอาใจใส่ต่อผู้ทำวิจัยมาโดยตลอด ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้

ขอขอบพระคุณศาสตราจารย์ ดร.วาทีต เบญจพลกุล ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.เชาวน์ดิศ อัครกุล กรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ และ ดร.ศุภกร สิทธิไชยกรรมการสอบวิทยานิพนธ์จากภายนอกมหาวิทยาลัย ที่ได้สละเวลาตรวจสอบและให้คำแนะนำเพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณกลุ่มวิจัยโครงข่าย (Network research group: NRG) ซึ่งดูแลโดย รองศาสตราจารย์ ดร.เชาวน์ดิศ อัครกุล และ ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร ได้จัดกิจกรรมเพื่อส่งเสริมการเรียนรู้ซึ่งมีส่วนสำคัญในการเสริมสร้างความมั่นใจและความกล้าแสดงความคิดเห็นในเชิงวิชาการมากขึ้น ทำให้งานวิทยานิพนธ์นี้สำเร็จได้อย่างสะดวกราบรื่น

ขอขอบคุณกลุ่มโครงการสาทรโมเดล ที่อนุเคราะห์ข้อมูลจริงสำหรับการจำลองต่าง ๆ

ขอกราบขอบพระคุณบิดา มารดา ที่เป็นกำลังใจและกำลังทรัพย์ตลอดเวลา รวมทั้งให้โอกาสผู้วิจัยได้ศึกษาต่อในระดับปริญญาตรีบัณฑิต

งานวิจัยนี้ได้รับการสนับสนุนจากหน่วยปฏิบัติการวิจัยโครงข่ายไร้สาย และอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Unit) กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย และทุนอุดหนุนการวิจัยจากทุน 90 ปี กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย

สุดท้ายนี้ ขอขอบคุณห้องปฏิบัติการวิจัยระบบโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สำหรับทรัพยากรต่างๆ ในการศึกษา ค้นคว้าและวิจัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฅ
สารบัญรูป	ญ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิทยานิพนธ์	6
1.3 ขอบเขตวิทยานิพนธ์	6
1.4 ระเบียบวิธีการดำเนินงาน	6
1.5 ประโยชน์ที่ได้รับ	7
1.6 ประมวลวิทยานิพนธ์	7
2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง	9
2.1 โครงข่ายแอดฮอค (ad hoc network)	9
2.1.1 คุณลักษณะของโครงข่ายแบบแอดฮอค(characteristics of ad hoc network)	9
2.1.2 การควบคุมการเข้าถึงตัวกลางในโครงข่ายแอดฮอค	10
2.2 โครงข่ายแอดฮอคสำหรับยานพาหนะ (vehicular ad hoc network หรือ VANET)	14
2.2.1 เทคนิคการแจกจ่ายข้อมูล(data dissemination)	15
2.2.2 มาตรฐานการสื่อสารของโครงข่ายแอดฮอคสำหรับยานพาหนะ	16
2.2.3 แบบจำลองการเคลื่อนที่ในโครงข่ายแอดฮอค	19
2.3 โครงข่ายชื่อข้อมูล (named data network หรือ NDN)	24
2.4 โปรแกรมการจำลอง	25
2.4.1 โปรแกรมจำลองระบบการจราจร	26
2.4.2 โปรแกรมจำลองโครงข่ายเอ็นเอส-3	29
2.4.3 โปรแกรมจำลองโครงข่ายเอ็นดีเอ็นซิม (ndnSim)	31
3 การออกแบบโพรโทคอล	34
3.1 โครงสร้างของระบบ	35
3.2 กระบวนการทำงานของโนดข้างถนนและยานพาหนะ	37
3.3 กระบวนการทำงานของโพรโทคอล	37
3.3.1 ขั้นตอนการทำงานของโพรโทคอลในโหมดบรอดแคสต์ (broadcast mode)	37
3.3.2 ขั้นตอนการทำงานของโพรโทคอลในโหมดแอกทีเวท (activate mode)	41
3.3.3 ขั้นตอนการเลือกโนดส่งต่อข้อมูล	45
3.4 การพัฒนาโปรแกรมสำหรับใช้ในการจำลอง	47
3.5 สรุป	48

บทที่	หน้า
4 สภาพแวดล้อมของการจำลอง ผลการจำลองและวิเคราะห์ผลการจำลองระบบ	49
4.1 สภาพแวดล้อมแบบจำลองจากโครงการสาทรโมเดล	49
4.2 ผลการจำลองเมื่อเปรียบเทียบโครงข่ายชื่อข้อมูลกับโครงข่ายแอดฮอกแบบ TCP/IP	51
4.2.1 โครงข่ายแบบง่าย : โครงข่ายแบบกริดขนาด 3x3	52
4.2.2 โครงข่ายจากข้อมูลจริง : สาทรโมเดล	55
4.3 ผลการจำลองเพื่อศึกษาการทำงานของโพรโทคอลชื่อข้อมูล	58
4.4 ผลการจำลองสำหรับประเมินสมรรถนะของโพรโทคอลที่ออกแบบ	62
4.4.1 พารามิเตอร์ที่ใช้สำหรับประเมินสมรรถนะของโพรโทคอลที่ออกแบบ	62
4.4.2 โครงสร้างระบบที่จำลอง	63
4.4.3 ผลการจำลองของโพรโทคอลที่ออกแบบในโหมดบรอดแคสต์	66
4.4.4 ผลการจำลองของโพรโทคอลที่ออกแบบในโหมดแอดทีเวท	68
4.5 สรุป	81
5 การวิเคราะห์เชิงคณิตศาสตร์	82
6 บทสรุปและข้อเสนอแนะ	87
6.1 บทสรุป	87
6.2 ข้อเสนอแนะ	87
รายการอ้างอิง	89
ภาคผนวก	93
ก โปรแกรมสำหรับการตั้งค่าพารามิเตอร์ต่าง ๆ และ โปรแกรมหลักที่ใช้ในการจำลอง (configuration and code simulation)	94
ข โปรแกรมแพ็คเกจการร้องขอและโปรแกรมเมื่อได้รับแพ็คเกจการร้องขอ	107
ประวัติผู้เขียนวิทยานิพนธ์	110

สารบัญตาราง

	หน้า
ตารางที่ 2.1 การประยุกต์ใช้งานในโครงข่ายแอดฮอคสำหรับยานพาหนะ [31]	15
ตารางที่ 2.2 ชั้นโพรโทคอลของ WAVE [35]	18
ตารางที่ 2.3 พารามิเตอร์ระหว่างมาตรฐาน IEEE 802.11a และ IEEE802.11p	18
ตารางที่ 3.1 โครงสร้างส่วนหัวของแพ็กเก็ต	36
ตารางที่ 4.1 การตั้งค่าเทียบระหว่างโปรแกรม SUMO และ NETSTREAM	51
ตารางที่ 4.2 การตั้งค่าของผลการจำลองโครงข่ายแบบกริดขนาด 3x3	53
ตารางที่ 4.3 การตั้งค่าของผลการจำลองโครงข่ายจากข้อมูลจริง	56
ตารางที่ 4.4 การตั้งค่าของผลการจำลองเพื่อศึกษาการทำงานของโพรโทคอลชื่อข้อมูล	59
ตารางที่ 4.5 การตั้งค่าของผลการจำลองของโพรโทคอลที่ออกแบบในโมดบรอดแคสต์	66
ตารางที่ 4.6 การตั้งค่าของผลการจำลองเมื่อปรับเปลี่ยนจำนวนแพ็กเก็ตการร้องขอ	69
ตารางที่ 4.7 การตั้งค่าของผลการจำลองเมื่อปรับเปลี่ยนรัศมีการสื่อสาร	73
ตารางที่ 4.8 การตั้งค่าของผลการจำลองเมื่อปรับขนาดของกลุ่มข้อมูล	78
ตารางที่ 4.9 ตารางเปรียบเทียบสมรรถนะของโพรโทคอล	81

สารบัญรูป

	หน้า
รูปที่ 1.1 เปรียบเทียบโครงข่าย	3
รูปที่ 1.2 รูปแบบการกระจายข้อมูล [10]	4
รูปที่ 1.3 เปรียบการติดต่อสื่อสารระหว่างโครงข่าย TCP/IP กับ โครงข่ายชื่อข้อมูล	5
รูปที่ 2.1 โครงข่ายพื้นที่ท้องถิ่นแบบไร้สายแบบไม่มีสถานีฐานและมีสถานีฐาน	9
รูปที่ 2.2 การสื่อสารแบบหลายช่วงเชื่อมต่อ (multi-hop communication)	10
รูปที่ 2.3 กลไกการเข้าใช้ตัวกลางของ IEEE 802.11 [30]	11
รูปที่ 2.4 ปัญหาโนดซ่อนเร้นและปัญหาโนดที่มองเห็น	12
รูปที่ 2.5 ตัวอย่างการเข้าถึงช่องสัญญาณในมาตรฐาน IEEE802.11 แบบ DCF	13
รูปที่ 2.6 การเพิ่มแบบ Exponential ของ contention window (CW)	13
รูปที่ 2.7 เปรียบเทียบการแจกจ่ายข้อมูล	16
รูปที่ 2.8 ช่องสัญญาณของมาตรฐาน WAVE. [35]	17
รูปที่ 2.9 รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่โดยใช้ random walk ใน 2 มิติ	20
รูปที่ 2.10 รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่โดยใช้ random walk ใน 2 มิติ (ระยะทางคงที่)	21
รูปที่ 2.11 รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่โดยใช้แบบแรนดอมเวย์พอยต์	22
รูปที่ 2.12 รูปแบบแพ็กเก็ตในสถาปัตยกรรมข้อมูลชื่อ	24
รูปที่ 2.13 การเปรียบเทียบโครงสร้างและการทำงาน	26
รูปที่ 2.14 แสดงการจำลองของระบบ	26
รูปที่ 2.15 การสร้างแบบจำลองการเคลื่อนที่ของยานพาหนะบนถนน	27
รูปที่ 2.16 ผลลัพธ์ของการจำลองโปรแกรม sumo	28
รูปที่ 2.17 อธิบายความหมายของ ns3mobility.tcl	28
รูปที่ 2.18 ตัวเลือกของการประมวลผลสำหรับ trace file	29
รูปที่ 2.19 องค์ประกอบของโปรแกรมเอ็นเอส-3 [45]	30
รูปที่ 2.20 รูปแบบของโครงสร้างของโปรแกรมเอ็นดีเอ็นซีเอ็ม	31
รูปที่ 2.21 รูปแบบการจำลองของโปรแกรมเอ็นเอส-3	32
รูปที่ 2.22 รูปแบบการจำลองของโปรแกรมเอ็นดีเอ็นซีเอ็มโดยเพิ่มเติมจากโปรแกรมเอ็นเอส-3	32
รูปที่ 3.1 รูปแบบของโครงสร้างของโพรโทคอลที่นำเสนอ	35
รูปที่ 3.2 แผนภาพแสดงการทำงานของโพรโทคอล	38
รูปที่ 3.3 ผังงานแสดงกระบวนการของแพ็กเก็ต <i>Polling</i>	40
รูปที่ 3.4 ผังงานแสดงกระบวนการของแพ็กเก็ต <i>Data</i>	41
รูปที่ 3.5 โครงสร้างของ <i>data d_{n,i} message</i>	42
รูปที่ 3.6 ผังงานแสดงกระบวนการของแพ็กเก็ต <i>Request Info</i>	43
รูปที่ 3.7 ผังงานแสดงกระบวนการของแพ็กเก็ต <i>Reply Info</i>	44
รูปที่ 3.8 ระยะเวลาการจัด	45
รูปที่ 3.9 การพิจารณาโนดที่ทำการส่งต่อ	47
รูปที่ 3.10 รูปแบบการพัฒนาโปรแกรมโดยเพิ่มเติมจากโปรแกรม ndnSIM และโปรแกรม NS-3	48
รูปที่ 4.1 การจำลองของระบบ	49

รูปที่ 4.2	แผนที่ของสาทรโมเดลบน Google map	50
รูปที่ 4.3	การจำลองของระบบ	50
รูปที่ 4.4	รูปแบบจำลองถนนของการทดลองที่ 1	52
รูปที่ 4.5	อัตราร้อยละของการส่งข้อมูลสำเร็จบนแบบจำลองยานพาหนะ เชื่อมต่อ	54
รูปที่ 4.6	เวลาประวิงเฉลี่ยในการส่งข้อมูลบนแบบจำลองยานพาหนะ เชื่อมต่อ	54
รูปที่ 4.7	แบบจำลองบนถนนสาทรและเส้นทางการเคลื่อนที่ของคู่ยานพาหนะ ณ วันที่ 7 กรกฎาคม พ.ศ. 2557 เวลา 15.30.00 – 15.46.40 น.	55
รูปที่ 4.8	อัตราร้อยละของการส่งข้อมูลสำเร็จของยานพาหนะบนถนนสาทร	57
รูปที่ 4.9	เวลาประวิงเฉลี่ยในการส่งข้อมูลของยานพาหนะบนถนนสาทร	57
รูปที่ 4.10	เวลาประวิงในการส่งข้อมูลของยานพาหนะบนถนนสาทรตามดัชนีลำดับของแพ็กเกต	58
รูปที่ 4.11	แสดงรูปแบบการเคลื่อนที่ของยานพาหนะ	59
รูปที่ 4.12	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ	60
รูปที่ 4.13	จำนวนการส่งต่อข้อมูลซ้ำ	60
รูปที่ 4.14	เวลาประวิงข้อมูลเฉลี่ย	61
รูปที่ 4.15	ตำแหน่งของโนดข้างถนน บนแผนที่จริง	64
รูปที่ 4.16	ตำแหน่งของโนดข้างถนน บนโปรแกรม SUMO	64
รูปที่ 4.17	จำนวนความหนาแน่นบนถนนสาทร	65
รูปที่ 4.18	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ	67
รูปที่ 4.19	เวลาประวิงเฉลี่ย	67
รูปที่ 4.20	เวลาการแจกจ่ายข้อมูล	68
รูปที่ 4.21	ปริมาณกราฟฟิกของโครงข่าย	69
รูปที่ 4.22	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ	70
รูปที่ 4.23	เวลาการแจกจ่ายข้อมูลโดยรวม	71
รูปที่ 4.24	เวลาการแจกจ่ายข้อมูล	72
รูปที่ 4.25	ค่าวิสัยสมารถ	72
รูปที่ 4.26	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 10	73
รูปที่ 4.27	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 70	74
รูปที่ 4.28	เวลาการแจกจ่ายข้อมูล กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 10	75
รูปที่ 4.29	เวลาการแจกจ่ายข้อมูล กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 70	76
รูปที่ 4.30	เวลาการแจกจ่ายข้อมูลโดยรวม กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 10	76
รูปที่ 4.31	เวลาการแจกจ่ายข้อมูลโดยรวม กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 70	77
รูปที่ 4.32	อัตราส่วนการรับแพ็กเกตข้อมูลสำเร็จ กรณีความหนาแน่นจำนวน 128 และ 1282 โหนด	79
รูปที่ 4.33	จำนวนของการส่งต่อข้อมูลซ้ำ กรณีความหนาแน่นจำนวน 128 และ 1282 โหนด	79
รูปที่ 4.34	เวลาประวิงข้อมูลเฉลี่ย ความหนาแน่นจำนวน 128 โหนด	80
รูปที่ 4.35	เวลาประวิงข้อมูลเฉลี่ย ความหนาแน่นจำนวน 1282 โหนด	81
รูปที่ 5.1	โครงสร้างระบบที่พิจารณา	82
รูปที่ 5.2	ความน่าจะเป็นของความสำเร็จเมื่อมีแพ็กเกตข้อมูลในตัวเอง ($x=0$)	84

รูปที่ 5.3 ความน่าจะเป็นของความสำเร็จเมื่อได้รับการตอบกลับแพ็กเก็ตข้อมูลก่อนหน้า	1
ฮอป ($x=1$)	85
รูปที่ 5.4 ความผิดพลาดสัมบูรณ์ของ $P(x N, n, k)$ เมื่อจำลองด้วย Monti-Carlo 1 ล้านครั้ง	86

บทที่ 1

บทนำ

เทคโนโลยีการสื่อสารไร้สายจากอดีตจนถึงปัจจุบันได้มีการปรับเปลี่ยนและถูกพัฒนาไปอย่างมาก ซึ่งการพัฒนาศักยภาพของโครงข่ายเป็นหัวข้อที่นักวิจัยและผู้เชี่ยวชาญในสาขาเฉพาะให้ความสนใจ วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งที่นำเสนอแนวทางการพัฒนารูปแบบและวิธีการเพื่อช่วยพัฒนาโครงข่ายไร้สายให้มีคุณภาพที่ดีขึ้น โดยเนื้อหาในบทนี้ได้กล่าวถึงความเป็นมาและความสำคัญของปัญหา จากนั้นได้เสนอแนวทางของวิทยานิพนธ์ วัตถุประสงค์ของวิทยานิพนธ์ ขอบเขตของวิทยานิพนธ์ รวมไปถึงขั้นตอนดำเนินงาน และประโยชน์ที่คาดว่าจะได้รับ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีการสื่อสารแบบไร้สายได้เข้ามามีบทบาทที่สำคัญในชีวิตประจำวันของมนุษย์ ตลอดจนความต้องการของผู้ใช้งานที่มีมากขึ้น ทำให้มีการพัฒนาเทคโนโลยีของการสื่อสารไร้สาย ออกมาเป็นจำนวนมากและหลากหลายชนิด โดยเฉพาะการสื่อสารแบบโครงข่ายแอดฮอค (ad hoc network) กำลังเป็นที่สนใจในตำนานงานวิจัย ซึ่งโครงข่ายแอดฮอคนั้นจะมีการติดต่อสื่อสารกันโดยตรง และไม่จำเป็นต้องมีตัวกลางหรือจุดเข้าถึง (access point) อย่างไรก็ตาม เมื่อจุดโหนดที่ต้องการสื่อสารกัน มีระยะการสื่อสารที่เกินระยะของการส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง การสื่อสารจึงจำเป็นต้องอาศัยโหนดระหว่างทาง (intermediate node) สำหรับการส่งต่อข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง ในอดีตที่ผ่านมา โครงข่ายแอดฮอคส่วนใหญ่ถูกนำไปใช้ด้านการทหาร เนื่องจากในสนามรบนั้น การวางโครงข่ายที่เป็นแบบศูนย์กลาง หรือ โครงข่ายแบบเซลลูลาร์มีความยากลำบาก อีกทั้งอุปกรณ์การสื่อสารมีความเสี่ยงต่อการถูกทำลาย แต่ในปัจจุบัน โครงข่ายแอดฮอคมีความสะดวกสบายในการพกพา และการติดตั้งอุปกรณ์ไม่มีความซับซ้อนเหมือนกับโครงข่ายแบบศูนย์กลาง โครงข่ายแอดฮอกริงถูกนำมาใช้งานในเชิงพาณิชย์และสามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น การขนส่งมวลชน เป็นต้น

ระบบการขนส่งมวลชนได้พัฒนาระบบการจัดการจราจรอัจฉริยะ (ITS or Intelligent Transportation System) เพื่อให้บริการด้านข้อมูลการจราจรกับประชาชน เช่น ในประเทศญี่ปุ่น ได้มีการให้บริการ VICS (Vehicle Information and Communication System Center) หรือ ศูนย์ระบบสื่อสารและให้ข้อมูลการจราจร [1] ซึ่งเป็นการรายงานข้อมูลสภาพการจราจรที่เกิดขึ้น จากการสื่อสารระหว่างอุปกรณ์บนยานพาหนะกับถนน โดยมีเสาอากาศติดตั้งอยู่ริมถนนหรือตามเสาไฟฟ้า ใช้ในการถ่ายทอดสภาพการจราจร ซึ่งรับสัญญาณมาจากเซ็นเซอร์ที่ติดตั้งตามพื้นผิวบนถนน เมื่อมียานพาหนะวิ่งผ่านเซ็นเซอร์ เซ็นเซอร์จะส่งข้อมูลให้คอมพิวเตอร์ประมวลผลและวิเคราะห์สภาพการจราจร เพื่อแจ้งข้อมูลการจราจรให้กับผู้ใช้งานพาหนะที่ติดตั้งเครื่องระบบนำร่องที่รับสัญญาณจากดาวเทียมจีพีเอส และมีแผนที่ดิจิทัลของถนนในเขตเมือง แม้ว่าระบบ VICS มีประโยชน์ในการให้ข้อมูลอย่างมาก แต่เมื่อพิจารณาการทำงานแล้วจะเห็นได้ว่าระบบใช้อุปกรณ์และงบประมาณการลงทุนสูง อีกทั้งยังเกิดช่องว่างในการรับข้อมูล กล่าวคือ VICS จะมีการรายงานตามเวลาที่กำหนด ไม่รายงานแบบทันทีทันใด โดยต้องรอข้อมูลจากศูนย์กลาง ที่ต้องรวบรวมข้อมูลและประมวลผล ซึ่งจะมีผลกระทบต่อผู้ใช้งานพาหนะในกรณีที่เกิดอุบัติเหตุ รวมถึงในกรณีที่มีการจราจรคับคั่ง เนื่องจากจากผู้ใช้งานพาหนะทุกคนได้รับข้อมูลในช่วงเวลาที่กำหนด และตัดสินใจเลือกเส้นทางที่แนะนำพร้อมกัน ดังนั้นปัญหา

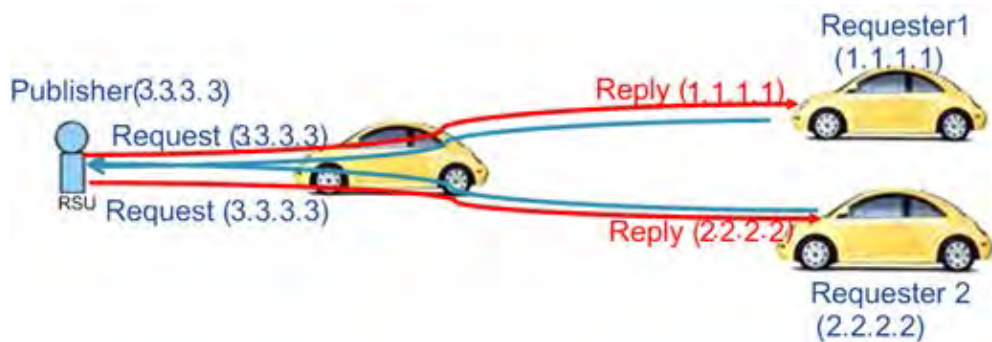
เหล่านี้สามารถแก้ไขโดยใช้ระบบโครงข่ายแอดฮอคสำหรับยานพาหนะ (vehicular ad hoc network or VANET) ติดต่อกันสื่อสารข้อมูลแบบทันทีทันใดระหว่างยานพาหนะด้วยกัน หรือระหว่างยานพาหนะกับอุปกรณ์ที่ติดตั้งอยู่กับที่ เพื่อช่วยในการให้ข้อมูลเกี่ยวกับระบบเตือนภัยล่วงหน้าในช่วงเวลาสั้นๆ ก่อนถึงจุดเกิดเหตุ หรือภาวะแวดล้อมไม่ดี เช่น เมื่อมีหมอกลงจัด, ฝนตกหนัก จนผู้ขับขี่ไม่สามารถมองเห็นเหตุการณ์ชัดเจนจากระยะไกล ทำให้การตัดสินใจควบคุมยานพาหนะ ทำได้ยาก หรือการจราจรในตัวเมืองขนาดใหญ่ มีสภาพการจราจรที่คับคั่งและไม่สามารถคาดการณ์สภาพการจราจรได้ ข้อมูลจึงมีความจำเป็นสำหรับการตัดสินใจในการเดินทาง ดังนั้นการรับข้อมูลจากศูนย์กลางที่ได้มาจากกล้องที่ติดตั้งตามท้องถนน หรืออุปกรณ์ตรวจจราจรบนถนนนั้น นอกจากจะมีปัญหาในการรายงานข้อมูลตามช่วงเวลาที่กำหนด ไม่ทันที่ทันใดแล้ว ระบบศูนย์กลางยังมีการลงทุนที่สูง เพราะต้องมีการวางระบบและติดตั้งโครงข่ายเชื่อมต่อกับศูนย์กลางการควบคุม

โครงข่ายการสื่อสารในปัจจุบันใช้โพรโทคอลแบบ TCP/IP โดยแต่ละโหนดจะกำหนดที่อยู่ไอพี (IP address) เมื่อผู้ร้องขอต้องการข้อมูล ผู้ร้องขอจะต้องระบุไอพีปลายทางเพื่อส่งข้อมูล และเมื่อโหนดระหว่างทางได้รับข้อมูล โหนดระหว่างทางจะตรวจสอบตารางไอพี และส่งต่อไปยังไอพีปลายทาง ซึ่งเป็นการติดต่อกันสื่อสารแบบ end-to-end communication ในกรณีที่ผู้ร้องขอรายอื่นต้องการข้อมูลชนิดเดียวกัน ผู้ร้องขอนั้นต้องส่งการร้องขอไปที่ไอพีของโหนดปลายทางเดียวกัน แสดงในรูปที่ 1.1ก) สำหรับโครงข่ายแอดฮอคสำหรับยานพาหนะ โหนดจะมีลักษณะการเคลื่อนที่เร็ว ซึ่งจะทำให้พบปัญหาการขาดความต่อเนื่องในการรับส่งข้อมูล ทำให้งานวิจัยที่มีอยู่ส่วนใหญ่มุ่งเน้นไปที่การแพร่กระจายข้อมูลให้มีประสิทธิภาพ สามารถสรุปได้เป็นสามรูปแบบ ได้แก่ การกำหนดเส้นทางทางภูมิศาสตร์ (geo-based routing) การส่งข้อมูลแบบฮอปต่อฮอป (hop-by-hop forwarding) และเทคนิคการถือและนำส่งต่อข้อมูล (carry-and-forward technique)

โพรโทคอลกำหนดเส้นทางทางภูมิศาสตร์ เป็นโพรโทคอลการกำหนดเส้นทางตามตำแหน่งแบบทั่วไปสำหรับโครงข่ายแอดฮอค โดยใช้ข้อมูลตำแหน่งทางภูมิศาสตร์เพื่อกำหนดเส้นทางตามแพ็กเก็ต GPSR [2] ซึ่งแต่ละโหนดจะมีการระบุตำแหน่งในส่วนหัวของแพ็กเก็ต เพื่อใช้คำนวณเลือกโหนดระหว่างทางและส่งต่อไปยังโหนดปลายทางแบบกตัญญู (Greedy) มีงานวิจัยจำนวนมากใช้ GPSR เพื่อปรับปรุงประสิทธิภาพการกำหนดเส้นทางสำหรับโครงข่ายแอดฮอคสำหรับยานพาหนะ ยกตัวอย่างเช่น A-STAR [3] คำนวณเส้นทางที่สั้นที่สุดโดยใช้ถนนที่มีน้ำหนักตามจำนวนรถประจำทางที่ผ่าน ในงานวิจัยที่ [4] ถูกออกแบบมาสำหรับระบบสื่อสารระหว่างยานพาหนะที่มีจำนวนยานพาหนะเบาบาง โดยใช้หลักการของ Carry-and-forward ซึ่งยานพาหนะจะเก็บข้อมูลไว้ จนกว่าจะมียานพาหนะคันอื่นผ่านเข้ามาในรัศมีการส่งสัญญาณ จากนั้นจึงเลือกยานพาหนะที่จะส่งข้อมูลต่อไปให้ถึงผู้รับโดยเร็วที่สุด โดยอาศัยข้อมูลด้านการเคลื่อนที่ ความเร็วและความน่าจะเป็นของความหนาแน่นของยานพาหนะที่ใช้พิจารณา แต่วิธีนี้มีข้อเสียคือใช้เวลามากในการคำนวณหาเส้นทางที่เหมาะสม นอกจากนี้ การได้มาซึ่งความน่าจะเป็นของการเคลื่อนที่ของยานพาหนะนั้นเป็นไปได้ยาก

การส่งต่อแบบฮอปต่อฮอปใช้กันอย่างแพร่หลาย เนื่องจากการโครงข่ายแอดฮอคสำหรับยานพาหนะมีการเปลี่ยนแปลงอย่างรวดเร็ว และการสร้างเส้นทางส่งต่อทั้งหมดเป็นไปได้ยาก การเลือกส่งต่อข้อมูลอาจขึ้นอยู่กับเมตริกต่าง ๆ เช่น GPCR [5] จะเลือกยานพาหนะที่อยู่ตรงกลางระหว่างจุดเชื่อมต่อเพื่อหลีกเลี่ยงอุปสรรคต่อการสื่อสารแบบไร้สาย ขณะที่ CLWPR [6] จะพิจารณาจากหลายปัจจัย ได้แก่ ระยะทาง ความยาวของคิวและอัตราความผิดพลาดของเฟรม เป็นต้น

เทคนิคการถือและนำส่งต่อข้อมูล จะถูกพิจารณาในสถานการณ์ที่โครงข่ายขาดความเชื่อมโยง กล่าวคือ ยานพาหนะบางตัวเลือกที่จะจัดเก็บแพ็กเก็ตหากไม่สามารถเลือกผู้ส่งต่อที่ดีกว่า แพ็กเก็ตจะถูกเก็บในยานพาหนะเคลื่อนที่ จนกว่าจะเจอยานพาหนะที่ถูกเลือกเพื่อส่งต่อ ยานพาหนะที่ได้รับ



ก)



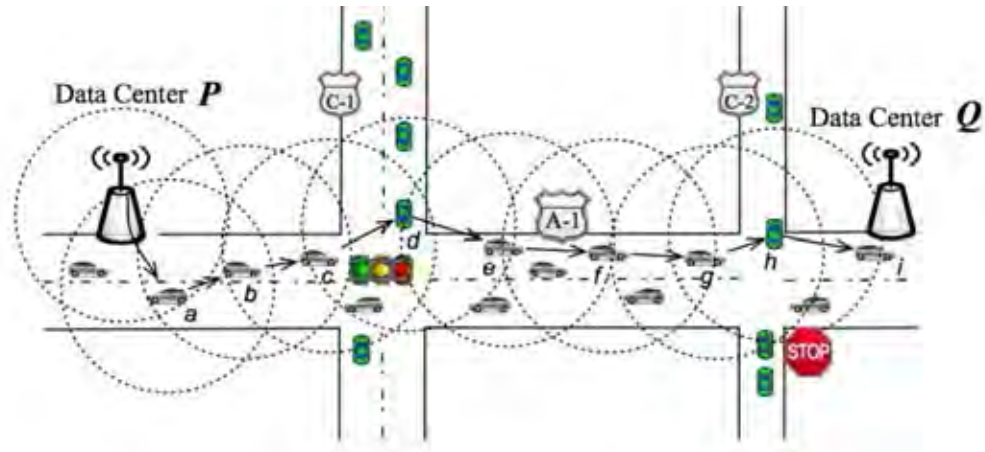
ข)

รูปที่ 1.1: ก) โครงข่ายที่ใช้ในปัจจุบัน และ ข) โครงข่ายชื่อข้อมูล

เลือกเหล่านี้เรียกว่ายานพาหนะ SCF (Store-Carry-and-Forward) งานวิจัย DFF-FSFD [7] จะเลือกยานพาหนะที่เคลื่อนที่ไปยังจุดหมายปลายทางในฐานะยานพาหนะ SCF และในงานวิจัย UV-CAST [8] จะเลือกยานพาหนะที่อยู่ในเส้นทางของการเชื่อมต่อกันจนถึงปลายทาง ซึ่งในทางปฏิบัติความหนาแน่นของการจราจร จะเปลี่ยนแปลงไปตามเวลาและเส้นทางถนน ดังนั้นเส้นทางที่ถูกเลือกนี้ อาจจะไม่ใช่ว่าเส้นทางที่ดีที่สุด อีกทั้งยังทำให้เกิดปัญหาการสูญหายของข้อมูลอีกด้วย

นอกจากนี้ ในงานวิจัยที่ [9] ได้ทำการเพิ่มสถานีที่ไม่ได้มีการติดต่อกับศูนย์กลาง (static node) เพื่อช่วยในการสื่อสาร ในสถานะที่มียานพาหนะเบาบาง และงานวิจัยที่ [10] ได้เสนอวิธีการกระจายข้อมูล โดยข้อมูลที่กระจายมาจากศูนย์กลาง เรียกว่า Data Center P สู่วานพาหนะบนท้องถนน ดังรูปที่ 1.2 จะเห็นได้ว่าการรับข้อมูลจากศูนย์กลางนั้น ข้อมูลที่ได้มาจากกล้องที่ติดตามท้องถนน หรืออุปกรณ์ตรวจจับสนุยานยนต์ ซึ่งจะมีปัญหาในเรื่องการลงทุนที่สูง เพราะจะต้องคำนึงถึงการวางระบบและติดตั้งโครงข่ายเพื่อเชื่อมต่อเข้ากับศูนย์กลางการควบคุม ในการแก้ปัญหาเรื่องนี้จึงนำโน้ตข้างถนน (roadside unit) มาติดตั้ง (เช่น สี่แยกไฟแดง ตรงป้อมตำรวจ ป้ายรถเมล์ หรือตู้โทรศัพท์) ซึ่งข้อมูลที่ได้มาจากการสื่อสารกับยานพาหนะที่รายงานสถานะของตนเองให้กับโน้ตข้างถนน เป็นการติดต่อสื่อสารแบบโครงข่ายแอดฮอก ดังนั้นจึงมีความจำเป็นที่ต้องมีการพัฒนาโพรโทคอลการสื่อสารระหว่างยานพาหนะกับโน้ตข้างถนน อย่างไรก็ตามวิธีการที่เสนอนี้ไม่ได้แก้ปัญหาการแจกจ่ายข้อมูลการจราจรที่มีเวลาล่าช้าสูงและปัญหาคอขวดของศูนย์กลางการควบคุม

งานวิจัยที่ผ่านมา การเสนอโครงข่ายแอดฮอกสำหรับยานพาหนะยังคงไม่มีประสิทธิภาพ เนื่องจากมีเวลาประวิงในการแจกจ่ายข้อมูลการจราจรที่สูงและปัญหาคอขวดของศูนย์กลางการควบคุม อีกทั้งเมื่อพิจารณาระยะเวลาห่างระหว่างผู้ส่งและผู้รับที่มีขนาดใหญ่ [11, 12] การใช้งานโพรโทคอลแบบ

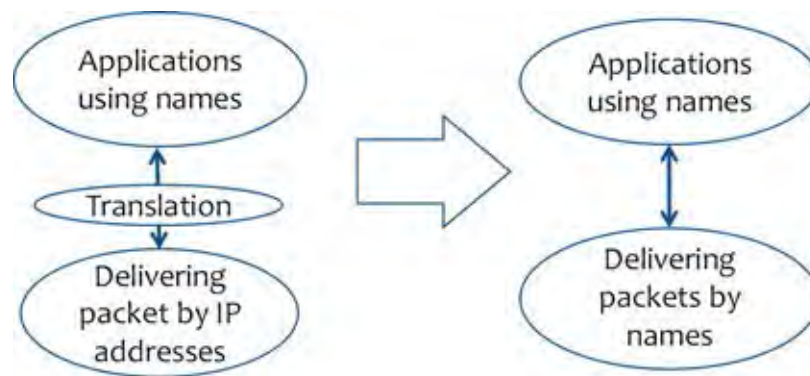


รูปที่ 1.2: รูปแบบการกระจายข้อมูล [10]

TCP/IP จะไม่เหมาะสมกับโครงข่ายแอตฮ็อกสำหรับยานพาหนะที่มีสภาพแวดล้อมเป็นแบบเปลี่ยนแปลงตลอดเวลา และเมื่อพิจารณาในมุมมองของผู้ใช้บนท้องถนนที่มุ่งเน้นไปที่เนื้อหาของข้อมูลที่ต้องการ ซึ่งไม่จำเป็นต้องสนใจว่าข้อมูลที่ต้องการมาจากยานพาหนะคันใด หรือที่อยู่ไอพีของแหล่งข้อมูล รวมถึงการร้องขอข้อมูล โดยไม่ต้องระบุที่อยู่ไอพีปลายทาง กล่าวคือ ถ้ายานพาหนะคันใดมีชื่อข้อมูลดังกล่าวแล้ว ยานพาหนะคันนั้นสามารถส่งต่อข้อมูลกลับไปยังผู้ร้องขอได้ทันที จากปัญหาดังกล่าว ทำให้สถาปัตยกรรมชื่อข้อมูล หรือ โครงข่ายชื่อข้อมูล (Named Data Network (NDN)) [13, 14] ได้รับการเสนอขึ้นใหม่สำหรับอินเทอร์เน็ตในอนาคต (future internet) ซึ่งมุ่งเน้นการให้บริการข้อมูลตามที่ผู้ร้องขอต้องการ แสดงในรูปที่ 1.3

สำหรับกระบวนการทำงานของโครงข่ายชื่อข้อมูลนั้น มีแพ็กเก็ตสองชนิด คือ แพ็กเก็ตการร้องขอ (Interest packet) และ แพ็กเก็ตข้อมูล (Data packet) แสดงในรูปที่ 1.1) เมื่อผู้ร้องขอมีความต้องการข้อมูล ผู้ร้องขอจะ flooding แพ็กเก็ตการร้องขอออกไป หลังจากนั้น โหนดใด ๆ ที่ได้รับแพ็กเก็ตการร้องขอ จะเข้าสู่กระบวนการทำงานแบบโครงข่ายชื่อข้อมูล หากพบว่ามีข้อมูลที่ตรงกับความสนใจก็จะตอบกลับแพ็กเก็ตข้อมูลไปยังผู้ร้องขอที่ส่งแพ็กเก็ตการร้องขอมา ซึ่งเส้นทางที่มีการส่งแพ็กเก็ตข้อมูล โหนดที่อยู่ระหว่างทาง สามารถจัดเก็บข้อมูลไว้ก่อนที่จะส่งแพ็กเก็ตข้อมูลกลับไปยังผู้ร้องขอ ซึ่งข้อดีสำหรับการจัดเก็บข้อมูลไว้ในโหนดระหว่างทาง คือ เมื่อมีผู้ร้องขอรายอื่นต้องการข้อมูลแบบเดียวกัน จะสามารถตอบกลับข้อมูลได้ทันที เพื่อลดปัญหาคอขวดที่โหนดใดโหนดหนึ่ง อีกทั้งยังช่วยแก้ปัญหาล่าช้าในการส่งที่ล่าช้า และเพิ่มประสิทธิภาพของการใช้ช่องสัญญาณได้อีกด้วย

เริ่มแรกงานวิจัยเกี่ยวกับโครงข่ายชื่อข้อมูลแบบดั้งเดิมจะให้ความสนใจกับการเพิ่มประสิทธิภาพของการออกแบบโครงสร้างข้อมูล และกลไกเส้นทางในโครงข่ายของอินเทอร์เน็ต (Internet network) โดยใช้กับโครงสร้างระบบที่ไม่มีเคลื่อนที่ [15–18] ต่อมางานวิจัยส่วนหนึ่งได้นำโครงข่ายชื่อข้อมูลมาประยุกต์ใช้ในโครงข่ายแอตฮ็อกสำหรับยานพาหนะ ซึ่งโครงสร้างระบบจะมีการเคลื่อนที่ ดังนั้นระบบโครงข่ายชื่อข้อมูลแบบดั้งเดิมจึงไม่เหมาะสมเมื่อนำมาใช้กับโครงข่ายแอตฮ็อกสำหรับยานพาหนะ งานวิจัยในปัจจุบันจึงมุ่งเน้นการออกแบบและสร้างโพรโทคอลการสื่อสารให้เหมาะสมกับระบบที่พิจารณา มุ่งงานวิจัยที่ได้ออกแบบเบื้องต้นสำหรับการประยุกต์ใช้สถาปัตยกรรมชื่อข้อมูลชื่อกับโครงข่ายแอตฮ็อกสำหรับยานพาหนะใน [19] และ [20] และได้มีการพัฒนาต่อยอดใน [21–26] ยกตัวอย่างเช่น Bai และ Krishnamachari [19] ใช้พิจารณาการเคลื่อนที่ของรถ



รูปที่ 1.3: เปรียบการติดต่อสื่อสารระหว่างโครงข่าย TCP/IP กับ โครงข่ายชื่อข้อมูล

ในพื้นที่แบบเมือง เพื่อสำรวจว่าแอปพลิเคชันการแจ้งเตือนความแออัดของรถสามารถใช้งานได้อย่างไร สำหรับ Wang et al. [20] มุ่งเน้นไปที่การรวบรวมข้อมูลที่โนดข้างถนน การรวบรวมข้อมูลจากยานพาหนะจากการออกแบบแพ็กเก็ตการร้องขอ และแพ็กเก็ตข้อมูลบนพื้นฐานของสถาปัตยกรรมชื่อแบบดั้งเดิม ซึ่งสถาปัตยกรรมที่นำเสนอนี้มีการประเมินประสิทธิภาพเปรียบเทียบระหว่างแบบจำลองที่นำเสนอกับการทำงานแบบโพรโทคอลอินเทอร์เน็ตเคลื่อนที่ (Mobile IP)

ในงานวิจัยที่ Wang et al. [21,22] มีการวิเคราะห์ข้อดีของการใช้โครงสร้างชื่อแพ็กเก็ตการร้องขอ และแพ็กเก็ตข้อมูลสำหรับข้อมูลการจราจร งานวิจัยของ Arnould et al. [23] ใช้หลักการ ICN เพื่อเผยแพร่ข้อมูลด้านความปลอดภัยระหว่างยานพาหนะที่มีส่วนต่อประสานของวิทยุหลายเฟส ต่อมา TalebiFard และ Leung [24] ใช้เทคนิคการเขียนโค้ดเพื่อปรับปรุงการเผยแพร่เนื้อหาผ่านเส้นทาง ICN หลายเส้นทาง Amadeo et al. [25, 26] เสนอการทำงานของสถาปัตยกรรมชื่อสำหรับแอปพลิเคชันการค้นหาข้อมูลและตอบกลับข้อมูล ซึ่งโนดสามารถเป็นได้ทั้งโนดต้นทาง โนดระหว่างทางและ โนดปลายทาง อีกทั้ง Lucas et al. [27] ได้เสนอการลดการชนกันของข้อมูลการแพร่กระจายของข้อมูลการจราจร ตามอัลกอริทึมแบบโครงข่ายชื่อข้อมูล โดยอัลกอริทึมจะแยกแยะข้อมูลการจราจรออกเป็นประเภทต่างๆ เพื่อตั้งเวลาที่แตกต่างกันสำหรับการส่งต่อข้อมูลในระบบ เป็นการเพิ่มประสิทธิภาพให้สูงขึ้นและใช้ต้นทุนต่ำ นอกจากนี้ Giulio et al. [28] ได้แนะนำสถานที่ตั้งของโหนดปลายทางตามขั้นตอนวิธีโครงข่ายชื่อข้อมูล เมื่อมีการร้องขอข้อมูล อัลกอริทึมจะเข้ารหัสตำแหน่งทางภูมิศาสตร์ (Geo-Faces) และแทรกในชื่อข้อมูลเพื่อให้มีการแพร่กระจายแพ็กเก็ตการร้องขอไปยังตำแหน่งทางภูมิศาสตร์ที่ผู้ร้องขอสนใจ หากแพ็กเก็ตข้อมูลตรงกับรถที่มีข้อมูลที่ผู้ร้องขอต้องการ รถคันดังกล่าวจะตอบกลับทันทีเพื่อเพิ่มประสิทธิภาพในระบบ สำหรับ TalebiFard et al. [29] ใช้ประโยชน์จากคุณสมบัติการเพิ่มจำนวนของโนดที่อยู่ในคลัสเตอร์เดียวกันในการส่งต่อข้อมูลการร้องขอในระบบ

สังเกตได้ว่าโครงข่ายชื่อข้อมูลที่ถูกนำไปประยุกต์ใช้งานกับโครงข่ายแอตฮอกของยานพาหนะจะมีการสร้างและออกแบบโดยใช้กับยานพาหนะโดยตรง ทำให้เกิดการเปลี่ยนแปลงเส้นทางบ่อย ประสิทธิภาพและสมรรถนะการทำงานของโพรโทคอลจะลดลง ซึ่งงานวิจัยดังกล่าวยังไม่ได้พิจารณาการแจกจ่ายข้อมูลระหว่างยานพาหนะกับโนดข้างถนนที่มีข้อมูลที่ต้องการ เนื่องจากแหล่งของข้อมูลมีความสำคัญในการให้บริการข้อมูลด้านต่าง ๆ กับยานพาหนะ ผู้วิจัยจึงมุ่งเน้นความสนใจไปที่แหล่งข้อมูล เพื่อกำหนดทิศทางและพื้นที่จำกัดในการร้องขอข้อมูล

วิทยานิพนธ์ฉบับนี้จะนำเสนอโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายแอตฮอกสำหรับยาน-

พาหนะโดยใช้สถาปัตยกรรมชื่อข้อมูล และคำนึงถึงแหล่งที่มาของข้อมูลจากตำแหน่งของโนดข้างถนน ซึ่งนำไปใช้ในสภาพแวดล้อมได้ดีทั้งในโครงข่ายที่ไม่เชื่อมต่อในสภาพแวดล้อมที่รถเบาบางและเชื่อมต่อกันอย่างดีในสภาพแวดล้อมที่รถหนาแน่น บนพื้นฐานข้อมูลจริงจากโครงการสาทรโมเดล [47] โพรโทคอลที่เสนอได้รับการประเมินโดยการจำลองโปรแกรมจำลองโครงข่าย NS-3 และโปรแกรมจำลองระบบการจราจร SUMO ในสภาพแวดล้อมที่เป็นจริง อีกทั้งมีการวิเคราะห์ทางคณิตศาสตร์เพื่อหาความน่าจะเป็นที่จะได้รับแพ็กเก็ตข้อมูลสำเร็จ

1.2 วัตถุประสงค์ของงานวิทยานิพนธ์

เสนอโพรโทคอลการแจกจ่ายข้อมูลแบบสถาปัตยกรรมข้อมูลที่มีชื่อสำหรับการสื่อสารระหว่างยานพาหนะด้วยตัวเอง และ ยานพาหนะกับโนดข้างถนน และกำหนดกรอบทางคณิตศาสตร์ที่คำนึงถึงรูปแบบการเคลื่อนที่ของโนด เพื่อวิเคราะห์การทำงานของโพรโทคอล

1.3 ขอบเขตวิทยานิพนธ์

ขอบเขตของวิทยานิพนธ์นี้ในการตรวจสอบโพรโทคอลการแจกจ่ายข้อมูลของโครงข่ายแอดฮอกสำหรับยานพาหนะโดยใช้สถาปัตยกรรมชื่อข้อมูลมีดังต่อไปนี้

1. พัฒนาโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายไร้สายแบบแอดฮอกและสถาปัตยกรรมชื่อข้อมูล สำหรับใช้ในการสื่อสารระหว่างยานพาหนะด้วยตัวเอง และ ยานพาหนะกับโนดข้างถนน เพื่อให้ยานพาหนะสามารถมีข้อมูลการจราจรครบถ้วนเพื่อการประมวลผลข้อมูลการจราจรและเส้นทางที่ต้องการได้
2. พิจารณารูปแบบการเคลื่อนที่ของโนดตามข้อมูลจริงที่ได้รวบรวมจากโครงการสาทร บนถนนสาทร กรุงเทพมหานคร โดยปรับรูปแบบการเคลื่อนที่ของโนดบนถนนโดยใช้โปรแกรม SUMO
3. ใช้โปรแกรมจำลองระบบโครงข่าย NS-3 และ ndnSIM เพื่อประเมินสมรรถนะของโพรโทคอลที่ออกแบบด้วยพารามิเตอร์ต่าง ๆ เช่น อัตราส่วนการรับแพ็กเก็ตสำเร็จ (Data receive ratio) เวลาการแจกจ่ายข้อมูลโดยรวม (Total dissemination time) เวลาการแจกจ่ายข้อมูล (Dissemination time) ปริมาณทราฟฟิกในระบบ (Network traffic load) และค่าวิสัยสามารถ (Throughput) โดยใช้เทคโนโลยี IEEE 802.11p ในระดับเลเยอร์ดาต้าลิงค์
4. ศึกษาผลของวิธีการที่นำเสนอและนำเสนอการศึกษาเปรียบเทียบกับวิธีการโครงข่ายชื่อข้อมูลแบบดั้งเดิม
5. เสนอสมการคณิตศาสตร์สำหรับการหาความน่าจะเป็นที่จะได้รับแพ็กเก็ตข้อมูลสำเร็จ

1.4 ระเบียบวิธีการดำเนินงาน

1. ศึกษางานวิจัยที่เกี่ยวข้องกับปัญหาที่กำหนด:
 - งานวิจัยในอดีตที่เกี่ยวข้องกับโพรโทคอลการแจกจ่ายข้อมูลโครงข่ายแอดฮอกสำหรับยานพาหนะ

- งานวิจัยในอดีตที่เกี่ยวข้องกับแบบจำลองโครงข่ายชื่อข้อมูล
 - งานวิจัยที่ผ่านมาเกี่ยวกับโครงข่ายชื่อข้อมูล ในโครงข่ายแอตฮอกสำหรับยานพาหนะ
2. เสนอโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายแอตฮอกสำหรับยานพาหนะ โดยใช้สถาปัตยกรรมชื่อข้อมูล
 3. วิเคราะห์สมรรถนะของโพรโทคอลการแจกจ่ายข้อมูลที่น่าเสนอ
 4. จำลองโปรแกรมคอมพิวเตอร์และปรับเปลี่ยนพารามิเตอร์ตามที่ออกแบบไว้ เพื่อประเมินโพรโทคอลการแจกจ่ายข้อมูลที่น่าเสนอ
 5. สรุป วิเคราะห์ผลที่ได้ และรวบรวมข้อมูล
 6. เขียนบทความวิชาการระดับนานาชาติและวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่ได้รับ

1. เครื่องมือคำนวณสำหรับการวิเคราะห์ความสำเร็จของการสื่อสารแบบโครงข่ายชื่อข้อมูลกับโครงข่ายแอตฮอกสำหรับยานพาหนะ
2. สามารถนำวิธีการนี้ไปใช้ในการสื่อสารข้อมูลการจราจรที่ใช้โครงข่ายแบบแอตฮอกระหว่างยานพาหนะและยานพาหนะกับโหนดข้างถนน
3. เพื่อให้การเรียกใช้และการประมวลผลข้อมูลการจราจร มีประสิทธิภาพและมีความน่าเชื่อถือ
4. โปรแกรมจำลองการสื่อสารข้อมูลการจราจรในโครงข่ายแอตฮอกที่น่าเสนอ
5. แนวทางการวิจัยสำหรับการปรับปรุงสมรรถนะการติดต่อสื่อสารในโครงข่ายแอตฮอกสำหรับยานพาหนะต่อไป

1.6 ประมวลวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งรายละเอียดออกเป็น 5 บท ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์และแนวทางของวิทยานิพนธ์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการดำเนินงาน ประโยชน์ที่ได้รับ และประมวลวิทยานิพนธ์

บทที่ 2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง โดยกล่าวถึงความรู้พื้นฐานของงานวิจัย ประกอบด้วยความหมายของโครงข่ายแอตฮอก กลไกพื้นฐานของการเข้าใช้ช่องสัญญาณร่วมกัน แบบจำลองการเคลื่อนที่ของโหนด ภาพรวมของโครงข่ายข้อมูลชื่อ และโปรแกรมที่ใช้ทดสอบการทำงานที่เกี่ยวข้องกับการจำลองผลในวิทยานิพนธ์นี้

บทที่ 3 อธิบายเกี่ยวกับโครงสร้างของระบบและกระบวนการทำงานของโพรโทคอลการแจกจ่ายข้อมูล ซึ่งประกอบด้วย พารามิเตอร์ต่าง ๆ ของโพรโทคอล ชนิดของแพ็กเก็ตที่ใช้ในของโพรโทคอล

บทที่ 4 บทนี้จะกล่าวถึงสภาพแวดล้อมของการจำลอง ผลการจำลอง และวิเคราะห์ผลการจำลองระบบ ซึ่งประกอบด้วย พารามิเตอร์ ๆ ในการจำลอง ทอพอโลยีที่ใช้ในการทดสอบ และผลการทดสอบของโพรโทคอลที่น่าเสนอในสภาพแวดล้อมแบบต่าง ๆ

บทที่ 5 บทนี้จะกล่าวถึงการวิเคราะห์เชิงคณิตศาสตร์ เพื่อหาความน่าจะเป็นที่จะได้รับแพ็คเกจข้อมูลสำเร็จ

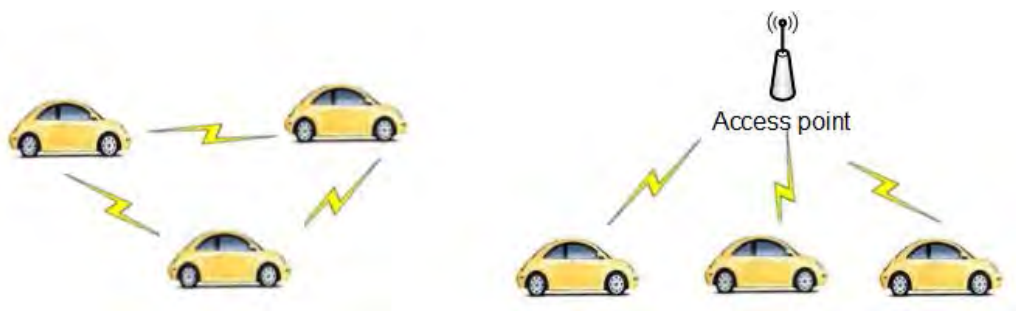
บทที่ 6 บทสรุป บทนี้จะกล่าวถึงบทสรุป และข้อเสนอแนะสำหรับงานวิจัยในอนาคต

บทที่ 2

ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1 โครงข่ายแอดฮอค (ad hoc network)

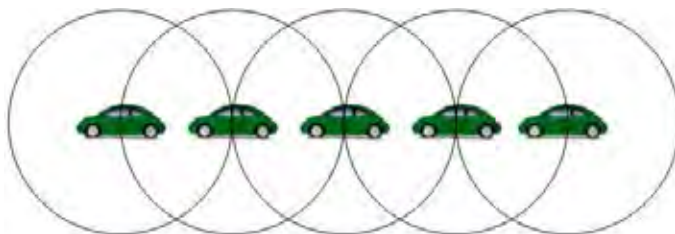
ในปัจจุบันโครงข่ายพื้นที่ท้องถิ่นแบบไร้สาย (wireless LAN) เป็นที่นิยมอย่างมาก เนื่องจากมีความสะดวกและมีความยืดหยุ่นในการใช้งาน โครงข่ายพื้นที่ท้องถิ่นแบบไร้สายมีการทำงาน 2 รูปแบบ คือ แบบที่มีสถานีฐาน (base station) และแบบที่ไม่มีสถานีฐาน ซึ่งแบบที่มีสถานีฐาน เป็นการสื่อสารระหว่างโหนดต้นทางไปยังโหนดปลายทาง โดยต้องส่งผ่านจุดเข้าถึง (access point) ก่อน จากนั้นข้อมูลจะถูกส่งไปยังโหนดปลายทาง และแบบที่ไม่มีสถานีฐาน เป็นการสื่อสารระหว่างโหนดต้นทางไปยังโหนดปลายทาง ซึ่งสามารถส่งผ่านกันได้โดยตรง โดยไม่อาศัยจุดเข้าถึง เรียกการทำงานแบบนี้ว่า แอดฮอค(ad hoc) ดังแสดงการเปรียบเทียบในรูปที่ 2.1



รูปที่ 2.1: โครงข่ายพื้นที่ท้องถิ่นแบบไร้สายแบบไม่มีสถานีฐานและมีสถานีฐาน

2.1.1 คุณลักษณะของโครงข่ายแบบแอดฮอค(characteristics of ad hoc network)

- ทอพอโลยีแบบพลวัต (dynamic topology) โหนดในโครงข่ายแอดฮอกระบุจะมีการเคลื่อนที่ที่เป็นแบบสุ่ม ดังนั้นระบบจะไม่สามารถคาดการณ์การเคลื่อนที่ของโหนดได้ ซึ่งส่งผลให้ทอพอโลยีของโครงข่ายมีการเปลี่ยนแปลงอยู่ตลอดเวลาขึ้นอยู่กับตำแหน่งของโหนดต่างๆ ณ ช่วงเวลาใดเวลาหนึ่ง อาจทำให้โหนดขาดการเชื่อมต่อระหว่างกัน รวมไปถึงความยากในการควบคุมการเข้าถึงตัวกลางในโครงข่ายแอดฮอค
- การสื่อสารเป็นแบบหลายช่วงเชื่อมต่อ (multi-hop communication) การสื่อสารในโครงข่ายแอดฮอกระบุเป็นการสื่อสารกันโดยตรง ซึ่งไม่อาศัยจุดเข้าถึง การควบคุมสั่งการและการจัดการสามารถดำเนินการได้ด้วยตนเอง ดังนั้นโหนดแต่ละโหนดจะสามารถเป็นโหนดต้นทาง โหนดปลายทาง และโหนดระหว่างทางได้ หากการสื่อสารเกินระยะของการส่งข้อมูล (transmission range) จำเป็นต้องอาศัยโหนดระหว่างทาง (intermediate node) เพื่อแจกจ่ายข้อมูลนั้นไปยังโหนดปลายทาง ซึ่งจะเห็นได้ว่ายังมีการแจกจ่ายของโหนดระหว่างทางมากขึ้นเท่าใดความซับซ้อนของโครงข่ายก็จะมากขึ้นเท่านั้น



รูปที่ 2.2: การสื่อสารแบบหลายช่วงเชื่อมต่อ (multi-hop communication)

- ข้อจำกัดเกี่ยวกับความปลอดภัย (security) สถาปัตยกรรมของโครงข่ายแอตฮอกมีโครงสร้างที่ไม่แน่นอน ทำให้มีความเสี่ยงที่อุปกรณ์การสื่อสารจะโดนดักฟังหรือถูกขโมยข้อมูล เพราะสัญญาณของข้อมูลแบบไร้สายอาจถูกส่งต่อไปยังแฮกเกอร์ได้ง่ายกว่าระบบสื่อสารแบบมีสาย
- ข้อจำกัดทางด้านแบนด์วิดท์ (bandwidth constrained) การสื่อสารแบบไร้สายจะมีข้อจำกัดของแบนด์วิดท์มากกว่าการสื่อสารแบบมีสาย เช่น ผลกระทบของการเข้าถึงแบบหลายทาง (multiple access) เฟดดิ้ง (fading) สัญญาณรบกวน (noise) ปัญหาของสถานีที่ซ่อนเร้น (hidden station problem) และปัญหาสถานีที่รับฟังได้ (exposed station problem) เป็นต้น ซึ่งผลกระทบของปัญหาดังกล่าว จะส่งผลให้การใช้แบนด์วิดท์ที่มีค่าต่ำกว่าค่าแบนด์วิดท์สูงสุดที่สามารถใช้ได้
- ข้อจำกัดทางด้านพลังงาน (energy constrained) โหนดในระบบโครงข่ายแอตฮอกมีการเคลื่อนที่โดยใช้พลังงานจากแบตเตอรี่ การออกแบบระบบแอตฮอกจึงต้องสนใจเกี่ยวกับพลังงานของอุปกรณ์ที่ใช้ในการติดต่อสื่อสารระหว่างกัน ถ้าพลังงานของอุปกรณ์ตัวหนึ่งตัวใดหมดไปหรือมีไม่เพียงพอในการส่งข้อมูล อาจส่งผลกระทบต่อข้อมูลที่มีการสูญหายภายในโครงข่ายได้

2.1.2 การควบคุมการเข้าถึงตัวกลางในโครงข่ายแอตฮอก

การควบคุมการเข้าถึงตัวกลางตามมาตรฐาน IEEE 802.11 ได้แบ่งโมดการทำงานเป็น 2 โมด

- โมด PCF (point coordinate function) คือ วิธีการเข้าถึงแบบใช้ศูนย์กลางในการควบคุมการเข้าถึง
- โมด DCF (distributed coordinate function) คือ วิธีการเข้าถึงตัวกลางแบบไม่ใช้ศูนย์กลางในการควบคุมการเข้าถึงตัวกลาง

โครงข่ายแอตฮอก มีการทำงานในโมด DCF โดยที่โมด DCF ใช้โพรโทคอล CSMA/CA (carrier sense multiple access with collision avoidance) ร่วมกับ binary exponential back-off เพื่อลดโอกาสในการเกิดการชนกันของข้อมูล กระบวนการทำงานของ CSMA คือ เมื่อโหนดหนึ่งต้องการเข้าถึงตัวกลาง โหนดดังกล่าวจะต้องตรวจสอบช่องสัญญาณก่อนว่ามีโหนดอื่นกำลังรับส่งสัญญาณข้อมูลอยู่หรือไม่และรอจนกว่าช่องสัญญาณจะว่าง ซึ่งช่วงเวลาว่างนี้มีค่าเท่ากับ DIFS (DCF inter-frame spacing) เมื่อช่องสัญญาณว่างเท่ากับค่า DIFS แล้วโหนดที่ต้องการเข้าถึงตัวกลางจะต้องรอต่อไปอีกระยะเวลาหนึ่งซึ่งเรียกช่วงเวลานี้ว่า ช่วงเวลา back-off ซึ่งแต่ละโหนดได้กำหนดระยะเวลาในการ

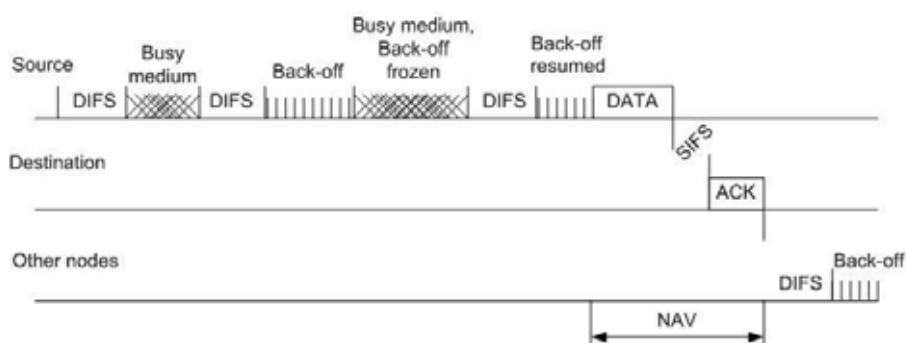
รอดังกล่าวไว้แล้วด้วยการสุ่มค่าจากความกว้างของหน้าต่างการช่วงชิง โหนดที่สุ่มได้ค่าระยะเวลาในการรอน้อยกว่าก็จะมีโอกาสในการเข้าถึงตัวกลางก่อน ซึ่งค่า back-off นี้ สามารถคำนวณได้จากสมการที่ (2.1)

$$\text{back-off} = \text{rand}(0, CW) \times \text{slottime} \quad (2.1)$$

โดยที่ CW คือ ขนาดความกว้างของหน้าต่างการช่วงชิง (contention window size)

slottime คือ ผลรวมของเวลาที่ต้องใช้ในการตรวจวัดเฟรม เวลาการประวิงในการเดินทาง เวลาที่ใช้ในการเปลี่ยนสถานะจากเครื่องรับไปเป็นเครื่องส่ง และเวลาที่ใช้ในการส่งสัญญาณไปยังชั้นควบคุมการเข้าถึงตัวกลาง (MAC layer) เพื่อบอกสถานะของตัวกลาง ตามมาตรฐาน IEEE 802.11 ค่า slottime มีค่าเท่ากับ 20 ไมโครวินาที

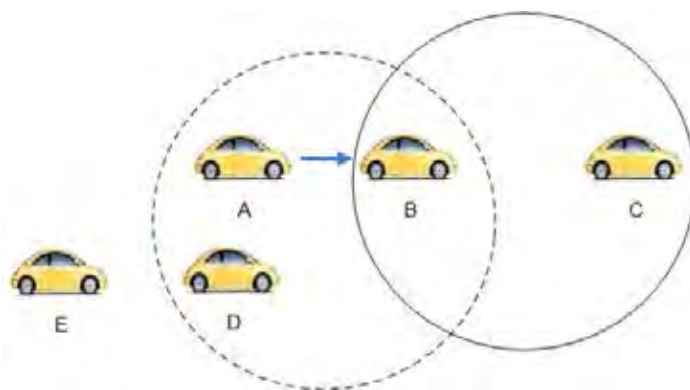
ถ้ามีโหนดอื่นส่งข้อมูลในช่วงเวลา back-off นี้ การนับค่า back-off นี้จะหยุดและจะเริ่มนับอีกครั้งหนึ่งหลังจากช่องสัญญาณว่างเป็นช่วงเวลาเท่ากับ DIFS ดังรูปที่ 2.3 จากการใช้ Binary exponential back-off เพื่อลดโอกาสการเกิดการชนกันของข้อมูลแล้ว ก็ยังมีโอกาสที่จะเกิดการชนกันของข้อมูลอยู่



รูปที่ 2.3: กลไกการเข้าใช้ตัวกลางของ IEEE 802.11 [30]

ปัญหาโหนดซ่อนเร้น (hidden terminal problem) เป็นปัญหาหนึ่งที่เกิดในโครงข่ายไร้สาย เนื่องจากมีโหนดที่ต้องการส่งข้อมูลบางโหนดไม่ได้อยู่ในขอบเขตการส่งคลื่นสัญญาณของโหนดที่กำลังส่งสัญญาณออกมา แต่อยู่ในขอบเขตการส่งคลื่นสัญญาณของโหนดรับเดียวกัน และเมื่อส่งข้อมูลออกมาส่งผลกระทบหรือทำให้เกิดการชนกันของแพ็กเก็ตขึ้นดังแสดงในรูปที่ 2.4 จากรูปโหนด A กำลังส่งแพ็กเก็ตไปยังโหนด B ในขณะเดียวกันโหนด C ตรวจสอบช่องสัญญาณพบว่าช่องสัญญาณว่างจึงส่งแพ็กเก็ตไปยังโหนด B เช่นกัน ซึ่งส่งผลให้เกิดการชนกันของแพ็กเก็ตที่โหนด B ขึ้น

ปัญหาโหนดที่รับสัญญาณได้ (exposed terminal problem) เป็นปัญหาที่เกิดจากโหนดที่ต้องการส่งข้อมูลโหนดหนึ่งอยู่ในขอบเขตการส่งคลื่นสัญญาณของโหนดที่ส่งข้อมูลอยู่อีกโหนดหนึ่ง ทำให้โหนดไม่สามารถส่งข้อมูลได้ทั้ง ๆ ที่การส่งข้อมูลทั้งสองไม่ได้อยู่ในเส้นทางเดียวกันดังแสดงในรูปที่ 2.4 โหนด A กำลังส่งข้อมูลไปยังโหนด B ในขณะเดียวกัน โหนด D ต้องการส่งข้อมูลไปยังโหนด E แต่เนื่องจากโหนด D อยู่ในขอบเขตการส่งคลื่นสัญญาณของโหนด A ทำให้ไม่สามารถส่งข้อมูลได้ ซึ่งปัญหานี้ส่งผลให้ประสิทธิภาพของการส่งแพ็กเก็ตในโครงข่ายลดลง

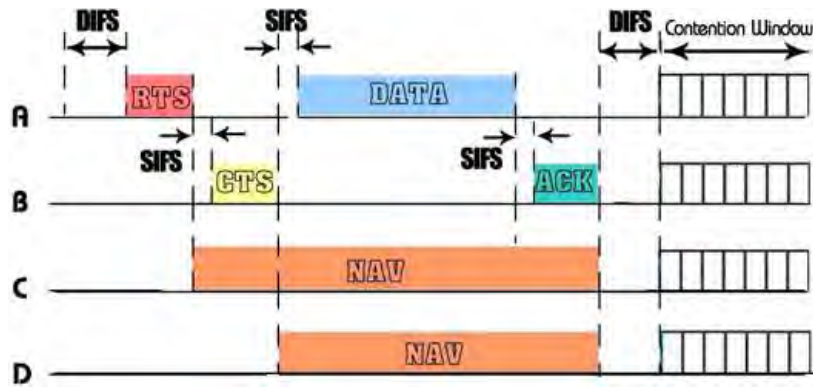


รูปที่ 2.4: ปัญหาโนดซ่อนเร้นและปัญหาโนดที่มองเห็น

จากปัญหาทั้งสองแบบมีวิธีการแก้ไขโดยการตรวจสอบการชนกันของข้อมูล ใช้โปรโตคอล CSMA/CD ซึ่งเป็นโปรโตคอลที่มีประสิทธิภาพในการตรวจสอบช่องสัญญาณอย่างมาก แต่เนื่องจากเทคนิค CSMA/CD ไม่สามารถนำมาใช้กับ WLAN ซึ่งใช้การสื่อสารแบบไร้สายได้ สาเหตุหลัก ๆ ก็คือการตรวจสอบการชนกันของสัญญาณในระหว่างที่ทำการส่งสัญญาณจะต้องใช้อุปกรณ์รับส่งคลื่นวิทยุที่สามารถรับและส่งสัญญาณในเวลาเดียวกันได้ (full duplex) ซึ่งจะมีราคาแพงกว่าอุปกรณ์รับส่งคลื่นวิทยุที่ไม่สามารถรับและส่งสัญญาณในเวลาเดียวกัน

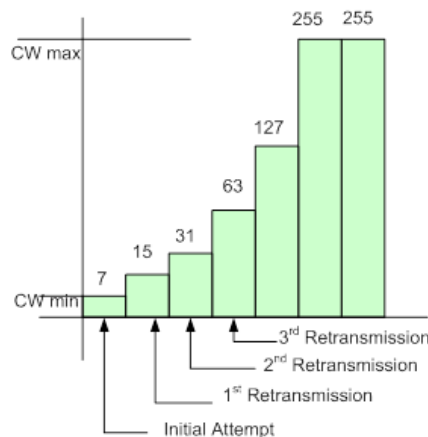
การหลีกเลี่ยงไม่ให้เกิดการชนกันของสัญญาณนั้น ตามมาตรฐาน IEEE 802.11 ได้ใช้กลไกที่เรียกว่า virtual carrier sense เพื่อแก้ไขปัญหานั้นแต่ละโหนดอาจไม่ได้ยินสัญญาณจากโหนดอื่นบางโหนด กลไกดังกล่าวมีการทำงานแสดงดังรูปที่ 2.5 เมื่อโหนดที่ต้องการจะส่งข้อมูลได้รับสิทธิในการเข้าถึงตัวกลางแล้วจะส่งแพ็กเก็ตสัญญาณที่เรียกว่า RTS (request to send) เพื่อเป็นการจองช่องสัญญาณ ก่อนที่จะส่งแพ็กเก็ตข้อมูลจริง ซึ่งแพ็กเก็ต RTS ประกอบไปด้วยระยะเวลาที่คาดว่าจะใช้ช่องสัญญาณจนแล้วเสร็จ (duration ID) รวมถึงที่อยู่ของโหนดต้นทางและโหนดปลายทาง เมื่อโหนดปลายทางได้รับแพ็กเก็ต RTS ก็จะตอบรับกลับมาด้วยการส่งแพ็กเก็ต CTS (clear to send) ซึ่งมีการบอกข้อมูลระยะเวลาที่คาดว่าโหนดต้นทางจะส่งข้อมูลนั้นจะใช้ช่องสัญญาณจนแล้วเสร็จ หลักการก็คือทุกๆ โหนดจะได้รับแพ็กเก็ต RTS หรือ CTS อย่างใดอย่างหนึ่งหรือทั้งสองอย่าง เมื่อได้รับ RTS หรือ CTS โหนดทุก ๆ โหนดจะทราบถึงว่าช่วงเวลาที่จะระบุไว้ใน duration ID ซึ่งเป็นช่วงเวลาที่ช่องสัญญาณจะถูกใช้และทุกโหนดที่ยังไม่ได้รับสิทธิในการเข้าถึงตัวกลางจะตั้งค่า NAV (network allocation vector) ให้เท่ากับ Duration ID ซึ่งแสดงถึงช่วงเวลาที่ยังไม่สามารถเข้าใช้ช่องสัญญาณได้ หรือเพื่อเป็นการเตือนให้ทราบว่าช่องสื่อสารไม่ว่าง ซึ่งสัญญาณ NAV เป็นเพียงสัญญาณภายในแต่ละโหนดที่สมมติขึ้นมาเพื่อบอกให้โหนดนั้น ๆ หยุดการส่งสัญญาณในช่วงเวลานั้น ๆ ซึ่งไม่มีการส่งสัญญาณนั้นจริงในโครงข่าย โดยโหนดทุก ๆ โหนดจะใช้กลไก virtual carrier sense ดังกล่าวผนวกกับการฟังสัญญาณในช่องสัญญาณจริงๆ ในการตรวจสอบว่าช่องสัญญาณว่างอยู่หรือไม่ โหนดที่ส่งข้อมูลจะต้องรอรับ ACK (Acknowledgement) จากโหนดที่ส่งข้อมูลไปให้ หากไม่ได้รับ ACK กลับมาภายในเวลาที่กำหนดจะถือว่าการชนกันของแพ็กเก็ตขึ้นและต้องทำการส่งข้อมูลเดิมซ้ำต่อไป

ในแต่ละขั้นตอนของการส่งและรับนั้นจะมีช่องว่างระหว่างเฟรม (inter-frame spacing) โดยช่องว่างระหว่างเฟรมนี้มี 4 ช่วง เพื่อวัตถุประสงค์ที่แตกต่างกัน โดยช่วงระยะหยุดนิ่งแบบแรกคือช่วง SIFS (short inter-frame spacing) ซึ่งจะมีช่วงที่สั้นที่สุด ที่จะคอยเปิดโอกาสให้โหนดที่กำลังสื่อสารกันอยู่นั้นสามารถสื่อสารกันได้ ซึ่งช่วงนี้จะใช้กับการส่งสัญญาณควบคุมต่างๆ และจะรวมถึงการยอมให้



รูปที่ 2.5: ตัวอย่างการเข้าถึงช่องสัญญาณในมาตรฐาน IEEE802.11 แบบ DCF

ผู้ส่งสามารถส่งเฟรม fragment burst ต่อกันเป็นลำดับได้โดยไม่ต้องส่งเฟรม RTS อีกรอบ แต่จะมีเพียงโหนดเดียวเท่านั้นที่สามารถตอบสนองในช่วงเวลานี้ และถ้าโหนดนั้นสละสิทธิ์ในการใช้ประโยชน์จากช่วงเวลาดังกล่าว ก็จะเข้าสู่ช่วงเวลาถัดไป คือช่วง PIFS (PCF inter-frame spacing) คือ ค่าช่องว่างระหว่างเฟรมที่ให้โหนดฐานสามารถครอบครองช่องสื่อสารได้โดยไม่ต้องแข่งขันกับผู้ใด โหนดฐานจะส่ง beacon frame หรือ poll frame ได้ กลไกนี้ช่วยให้โหนดที่กำลังส่งเฟรมข้อมูลหรือ fragment sequence สามารถส่งเฟรมต่อไปได้จนเสร็จสิ้นโดยที่ไม่มีผู้ใดมาขัดจังหวะได้ แต่ถ้ากรณีที่โหนดฐานไม่มีข้อมูลที่จะส่งก็จะถึงช่วงเวลา DIFS (DCF inter-frame spacing) ซึ่งเป็นช่องว่างระหว่างเฟรมที่ใช้สำหรับการส่งข้อมูลที่มีการเข้าถึงแบบ DCF ที่เปิดโอกาสให้ทุกโหนดสามารถส่งข้อมูลออกมาเพื่อขอใช้ช่องสัญญาณในกรณีที่เกิดการส่งสัญญาณซ้อนมาจากหลายโหนด กลไก binary exponential back-off จะถูกนำมาใช้แก้ปัญหานี้ ซึ่งจะกล่าวถึงวิธีการทำงานต่อไป และช่วงสุดท้ายคือ EIFS (extended inter-frame spacing) เป็นช่องว่างระหว่างเฟรมที่ใช้สำหรับตรวจสอบและรายงานความถูกต้องในการได้รับเฟรมข้อมูล เหตุที่กำหนดช่วงเวลานี้ไว้เป็นลำดับสุดท้าย ก็เพื่อให้โหนดที่ได้รับเฟรมนั้นมีโอกาสตรวจสอบให้ชัดเจนก่อน เพื่อไม่ให้ส่งข้อมูลออกมารบกวนการสื่อสารที่อาจกำลังดำเนินอยู่ตามปกติ



รูปที่ 2.6: การเพิ่มแบบ Exponential ของ contention window (CW)

วิธีการทำงานของ binary exponential back-off นั้น จะทำงานเมื่อโหนดที่มีเฟรมข้อมูลที่จะส่ง จะตรวจดูช่องสัญญาณจนกระทั่งช่องสัญญาณว่างซึ่งช่วงเวลานี้มีค่าเท่ากับ DIFS (distributed inter-frame space) หลังจากนั้นโหนดต้นทางจะรอเป็นระยะเวลาเท่ากับ random back-off interval หรือ binary exponential back-off จึงจะส่งข้อมูลได้ โดยค่า random back-off interval นี้จะมีค่าอยู่ในช่วง $[0, CW-1]$ โดยที่ CW คือ back-off window size ณ เวลาปัจจุบัน ในการพยายามส่งข้อมูลครั้งแรกนี้ CW จะมีค่าเท่ากับ CW_{\min} (initial back-off window size) ถ้าการส่งข้อมูลล้มเหลวหรือเกิดการชนกันของแพ็กของข้อมูลขึ้น ค่า CW จะมีค่าเพิ่มขึ้นเป็น 2 เท่าของค่าเดิมจนมีค่าเท่ากับ CW_{\max} (maximum window size) ดังแสดงในรูปที่ 2.6 หลังจากนั้นเมื่อโหนดปลายทางได้รับข้อมูลแล้วจะส่ง ACK (acknowledgement frame) ในช่วงเวลา SIFS (short inter-frame space) ในขณะที่โหนดต้นทางจะคอย ACK จากโหนดปลายทาง ถ้าไม่ได้รับ ACK ภายในช่วงเวลา ACK timeout หรือตรวจพบว่ามีเฟรมข้อมูลอื่นส่งข้อมูลอยู่ด้วยก็จะกลับไปทำขั้นตอนการกำหนดค่าช่วง random back-off ใหม่

2.2 โครงข่ายแอตฮอกสำหรับยานพาหนะ (vehicular ad hoc network หรือ VANET)

โครงข่ายแอตฮอกสำหรับยานพาหนะ (vehicular ad-Hoc network หรือ VANET) เป็นรูปแบบหนึ่งของโครงข่ายแอตฮอกแบบเคลื่อนที่ (mobile ad-hoc network หรือ MANET) โดยเป็นการติดต่อสื่อสารข้อมูลระหว่างยานพาหนะ หรือการติดต่อสื่อสารระหว่างยานพาหนะกับอุปกรณ์ที่ติดตั้งอยู่กับที่บริเวณของถนน (fixed equipment) การพิจารณาระหว่างระบบโครงข่ายแอตฮอกแบบเคลื่อนที่กับโครงข่ายแอตฮอกสำหรับยานพาหนะ มีส่วนคล้ายคลึงกันมาก แต่ก็มีความลักษณะพิเศษที่แตกต่างกัน ดังต่อไปนี้

- ระบบสมองกลอัจฉริยะ (embedded system): เป็นโปรแกรมที่ถูกติดตั้งอยู่ในยานพาหนะสำหรับการติดต่อสื่อสารระหว่างยานพาหนะได้ จึงมีความสำคัญที่จะต้องออกแบบโปรโตคอลการสื่อสารของโครงข่ายแอตฮอกสำหรับยานพาหนะ
- แหล่งพลังงานไม่จำกัด (infinite energy supply): อุปกรณ์ที่ติดตั้งในยานพาหนะ สามารถใช้แหล่งพลังงานจากแบตเตอรี่ของยานพาหนะได้โดยตรง ขณะเดินทางเครื่องยนต์สามารถชาร์จแบตเตอรี่ได้ ซึ่งแตกต่างจากเซนเซอร์ ที่ต้องคำนึงถึงพลังงานที่มีอยู่อย่างจำกัด โปรโตคอลของโครงข่ายแอตฮอกสำหรับยานพาหนะจึงไม่จำเป็นต้องพิจารณาในเรื่องของพลังงานที่ถูกใช้อย่างจำกัด
- ช่วงเวลาเชื่อมต่อที่สั้น (short connection time): เนื่องจากการเคลื่อนที่ของยานพาหนะมีการเคลื่อนที่ด้วยความเร็วสูงและเปลี่ยนแปลงอย่างรวดเร็ว ทำให้การติดต่อสื่อสารมีระยะเวลาที่สั้น อยู่ในหน่วยไมโครวินาที ทำให้ยากต่อการสร้างการเชื่อมต่อระหว่างโหนดต้นทางถึงโหนดปลายทางได้
- การเคลื่อนที่ที่สามารถคาดเดาได้ (predictable mobility): ยานพาหนะต้องเคลื่อนที่ไปตามรูปแบบของถนน จึงทำให้สามารถคาดเดาทิศทางและการเคลื่อนที่ของโหนดได้ อีกทั้งโหนดสามารถติด GPS เพื่อระบุตำแหน่งและเวลาในขณะนั้น และมีระบบนำทางที่บอกรายละเอียดบนแผน-

ตารางที่ 2.1: การประยุกต์ใช้งานในโครงข่ายแอตสอกสำหรับยานพาหนะ [31]

Application type	Content type	Local interest	Local validity	Lifetime
Active safety	Dangerous road warning	All vehicles	100 m	10 s
	Car accident warning	All vehicles	500 m	30 s
	Work zone warning	All vehicles	1 km	Construction duration
Public service	Emergency vehicle warning	All vehicles	500 m	10 min
	Highway information	All vehicles	5 km	All days
Improved driving	Road congestion region information	All vehicles	5 km	30 min
	Traffic navigation map	Subscribers only	10 km	30 min
Business& entertainment	Commercial advertisement	Subscribers only	1-5 km	1-10 days
	Multimedia file sharing	Subscribers only	1 km	10 min

ที่ได้ ข้อมูลที่ได้จาก GPS และ แผนที่ถนนนั้น สามารถนำมาใช้เพื่อเพิ่มประสิทธิภาพของโครงข่ายแอตสอกการจราจรและโครงข่ายแอตสอกการแจกจ่ายข้อมูล

- การประยุกต์ใช้งาน (applications): สามารถแบ่งได้ตามวัตถุประสงค์การใช้งาน ได้แก่ การประยุกต์ใช้งานด้านความปลอดภัย เมื่อยานพาหนะคันหน้าพบสิ่งกีดขวางหรือประสบอุบัติเหตุ ทำการแจ้งเตือนให้ยานพาหนะคันหลังที่ตามมาเพื่อให้ระวังและสามารถหลีกเลี่ยงอุบัติเหตุได้ทันท่วงทีหรือการสื่อสารระหว่างรถพยาบาลกับสัญญาณไฟจราจรเพื่อควบคุมให้ได้สัญญาณไฟเขียวตลอดเส้นทางในยามเมื่อเกิดเหตุฉุกเฉิน เป็นต้น สำหรับการประยุกต์ใช้งานที่ไม่เกี่ยวข้องกับด้านความปลอดภัย เช่น ข้อมูลการจราจร สภาพภูมิอากาศ สถานที่เติมน้ำมัน และพื้นที่ในการจอดรถ เป็นต้น

ตารางที่ 2.1 [31] แสดงภาพรวมของการประยุกต์ใช้งานในโครงข่ายแอตสอกของยานพาหนะ โดยคำนึงถึงประเภทของข้อมูล ความสนใจเฉพาะพื้นที่ บริเวณเฉพาะ และอายุการใช้งานของข้อมูล

1. *บริการด้านความปลอดภัย:* เพื่อเพิ่มความปลอดภัยให้กับผู้โดยสารโดยการแลกเปลี่ยนข้อมูลด้านความปลอดภัยเช่นระบบการเตือนภัยการชนกัน และระบบเตือนภัยของสภาพอากาศ เป็นต้น
2. *บริการข้อมูลการจราจร:* เพื่อแจ้งข้อมูลการจราจรแก่ผู้ขับขี่ เช่น ค่าเตือนของรถฉุกเฉิน และข้อมูลการจราจรบนทางด่วน เป็นต้น
3. *การปรับปรุงการขับขี่:* เพื่อเพิ่มความสะดวกของผู้โดยสารในการตัดสินใจเลือกเส้นทาง เช่น เวลาเดินทาง ความคับคั่ง และเส้นทางที่รวดเร็ว เป็นต้น
4. *ธุรกิจและบันเทิง:* เพื่อใช้งานเชิงธุรกิจและความบันเทิง เช่น การโฆษณา และการแชร์ไฟล์มัลติมีเดีย เป็นต้น

2.2.1 เทคนิคการแจกจ่ายข้อมูล(data dissemination)

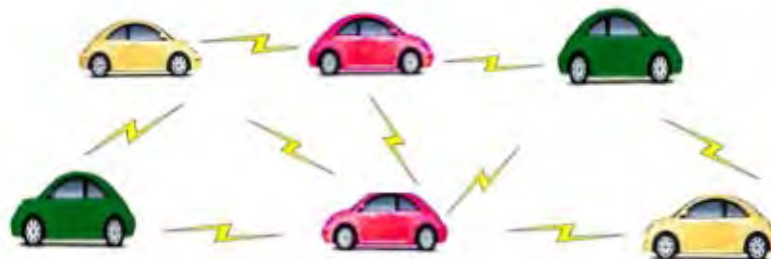
ในส่วนนี้จะอธิบายเทคนิคการแจกจ่ายข้อมูล เนื่องจากแบนด์วิธที่มีจำกัดโครงข่ายแอตสอกสำหรับยานพาหนะ ทำให้การแจกจ่ายข้อมูลมีความสำคัญมาก เทคนิคของการแจกจ่ายข้อมูลสามารถแบ่ง

ได้ดังนี้:

- *V2R / R2V dissemination*: การติดต่อสื่อสารระหว่างยานพาหนะกับโนดข้างถนน (RSU) ดังแสดงในรูป 2.7ก) ประกอบด้วยการแจกจ่ายข้อมูลสองประเภท คือ การผลักข้อมูล(push) และการดึงข้อมูล (pull) จากโนดข้างถนน โดยข้อมูลที่ยานพาหนะต้องการ เช่น สภาพการจราจรหรือการโฆษณาทางอิเล็กทรอนิกส์ ที่จอดรถหรือตำแหน่งของสถานีบริการน้ำมัน เป็นต้น
- *V2V dissemination*: การติดต่อสื่อสารระหว่างยานพาหนะด้วยกันเอง ดังแสดงในรูป 2.7ข) การติดต่อสื่อสารใช้วิธีการแพร่กระจายข้อมูลหรือการแจกจ่ายข้อมูล จะใช้การสื่อสารแบบแพร่กระจายข้อมูลในสภาพการจราจรเบาบาง เมื่อสภาพการจราจรหนาแน่น จะใช้การสื่อสารแบบแจกจ่ายข้อมูล แต่วิธีนี้ไม่น่าเชื่อถือ เมื่อมีความต้องการเลือกโนดส่งที่ดีที่สุด



ก)



ข)

รูปที่ 2.7: ก) การแจกจ่ายข้อมูลระหว่างยานพาหนะกับโนดข้างถนน และ ข) การแจกจ่ายข้อมูลระหว่างยานพาหนะด้วยกันเอง

2.2.2 มาตรฐานการสื่อสารของโครงข่ายแอตฮอกสำหรับยานพาหนะ

มาตรฐานการสื่อสารของโครงข่ายแอตฮอกสำหรับยานพาหนะ ดังแสดงในตารางที่ 2.2 กลุ่มวิจัย IEEE 1609 working group [34] ได้ออกมาตรฐานดังนี้

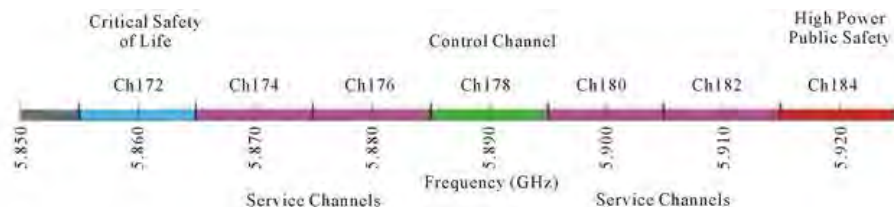
- IEEE1609.1 เป็นมาตรฐานสำหรับ Wireless Access For Vehicular Environment หรือ WAVE ซึ่งมีการจัดสรรทรัพยากรเพื่อนำไปใช้งานพื้นฐาน รวมถึงโพรโทคอลการสื่อสารที่ใช้รับส่งข้อมูลระหว่างยานพาหนะกับโนดข้างถนน
- IEEE1609.2 เป็นมาตรฐานความปลอดภัยใน WAVE คลื่นความถี่ 5.9 GHz. เพื่อใช้ระบุตัวตนและรักษาความลับ

- IEEE P1609.3 เป็นมาตรฐานด้านการจัดการโครงข่าย
- IEEE P1609.4 เป็นมาตรฐานสำหรับกำหนดการทำงานของคลื่นสัญญาณแบบหลายช่องสัญญาณ

การจัดสรรความถี่เป็นสิ่งที่จะต้องคำนึงถึงเป็นอันดับแรก โดยแต่ละกลุ่มประเทศมีความแตกต่างในการจัดสรรคลื่นความถี่ ได้แก่ อเมริกาเหนือ ยุโรปและญี่ปุ่น ดังนี้

- กลุ่มประเทศอเมริกาเหนือ (North America): สถาบันวิจัยอิเล็กทรอนิกส์และโทรคมนาคม (ETRI) ได้พัฒนา Dedicated Short Range Communications (DSRC) ซึ่งเป็นระบบการสื่อสารแบบแพ็กเก็ตและทำงานในย่านความถี่ 5.850-5.925 GHz แลบบความถี่แบ่งออกเป็น 7 ช่องละ 10 MHz แต่ละช่อง ระบบนี้ถูกกำหนดตั้งแต่ 300 ถึง 1000 เมตร ความเร็วในการเข้าถึงข้อมูลสูง 6 ถึง 27 Mbps และทำงานใน Half-Duplex (ไม่สามารถรับและส่งข้อมูลในเวลาเดียวกันได้)
- กลุ่มประเทศญี่ปุ่น (Japan): มาตรฐานการใช้งานในกลุ่มธุรกิจและอุตสาหกรรมด้านวิทยุ (Association of Radio Industries and Business standard หรือ มาตรฐาน ARIB STD-T55) มาตรฐานนี้มีระบบโครงสร้างเหมือน DSRC ในกลุ่มอเมริกาเหนือ มีการกำหนดย่านความถี่ 5.835-5.940 GHz และ 5.845-5.850 GHz สำหรับใช้งานในการเชื่อมต่อขาขึ้น (uplink) และ 5.790-5.795 GHz และ 5.800-5.805 GHz สำหรับใช้งานในการเชื่อมต่อขาลง (downlink) มาตรฐานนี้จะใช้ ASK modulation สำหรับอัตรารับส่งข้อมูลที่ 1 Mbps ด้วยการ ใช้ 8 ช่องสัญญาณ (Slot) ของ TDMA/FDD (Time Division Multiplex Access/ Frequency Division Duplex) เพื่อให้บริการกับรถได้สูงสุด 8 คันในระยะ 30 m. ปัจจุบันกำลังพัฒนา มาตรฐานใหม่ ARIB STD-T75
- กลุ่มประเทศยุโรป: กลุ่ม EC (European Commission) ได้กำหนดการจัดสรรย่านความถี่ 5.875-5.905 GHz สำหรับด้านความปลอดภัยบนท้องถนน (road safety applications) และ ย่าน 0.020 GHz ขึ้นไปจากย่านนี้ สำหรับการใช้งานในอนาคต และย่านความถี่ 5.855- 5.875 GHz สำหรับการใช้งานที่ไม่เกี่ยวข้องกับความปลอดภัย (non safety related applications)

2.2.2.1 802.11p PHY layer



รูปที่ 2.8: ช่องสัญญาณของมาตรฐาน WAVE. [35]

คุณสมบัติทางกายภาพ ของมาตรฐาน IEEE 802.11p มีพื้นฐานมาจากมาตรฐาน IEEE 802.11a ที่ใช้กันอยู่ทั่วไป ซึ่ง IEEE 802.11p ใช้ OFDM บนช่องสัญญาณ 10 MHz ในย่านความถี่ 5.9 GHz

ตารางที่ 2.2: ชั้นโพรโทคอลของ WAVE [35]

OSI Model	Data Plane	
	Resource Manager (IEEE1609.1)	WME (IEEE1609.3)
Layer 4	UDP/TCP (IEEE1609.3)	Security Services (IEEE1609.2)
	WSMP (IEEE1609.3)	
Layer 3	IPv6 (IEEE1609.3)	
Layer 2	LLC (IEEE1609.3)	
	Multichannel Operation (IEEE1609.4)	MLME Extension (IEEE1609.4)
Layer 1	WAVE MAC (802.11p)	MLME (802.11p)
	WAVE PHY (802.11p)	PLME (802.11p)

(ตามปกติของ IEEE 802.11 จะใช้ช่องสัญญาณ 20 MHz) โดยย่านดังกล่าวจะถูกแบ่งออกเป็น 7 ช่อง ดังแสดงในรูป 2.8 มีการใช้เทคนิค OFDM (orthogonal frequency division multiplexing) สำหรับการมอดูเลชัน (modulation) และมีการพัฒนาระดับชั้นการเชื่อมต่อ (link layer) เพื่อปรับลดข้อมูลในส่วนหัวของแพ็กเก็ต (Overhead) ให้ลดน้อยลง และมีการปรับรูปแบบข้อมูลในชั้น MAC (medium access control) เทียบกับ IEEE 802.11a ดังแสดงในตารางที่ 2.3 สำหรับมาตรฐาน IEEE 802.11p เนื่องจากโน้ตมีการเคลื่อนที่ตลอดเวลา จึงนำเทคนิคของ Diversity เพื่อลดปัญหาของ ICI (inter carrier interference) และผลกระทบจาก Doppler effect

ตารางที่ 2.3: พารามิเตอร์ระหว่างมาตรฐาน IEEE 802.11a และ IEEE802.11p

Parameters	IEEE 802.11a	IEEE 802.11p
Bit rate (Mbit/s)	6, 9, 12, 18, 24, 36, 48, 54	3, 4.5, 6, 9, 12, 18, 24, 27
Modulation mode	BPSK, QPSK, 16-QAM, 64-QAM	BPSK, QPSK, 16-QAM, 64-QAM
Code rate	1/2, 2/3, 3/4	1/2, 2/3, 3/4
Number of subcarriers	52	52
Symbol duration	4 μ s	8 μ s
Guard time	0.8 μ s	1.6 μ s
FFT period	3.2 μ s	6.4 μ s
Preamble duration	16 μ s	32 μ s
Subcarrier spacing	0.3125 MHz	0.15625 MHz

2.2.2.2 802.11p MAC layer

สำหรับมาตรฐาน IEEE 802.11p ยังอยู่ในขั้นการพัฒนาฉบับร่าง (Draft version) ของ WAVE โดยทำงานที่อัตราส่งข้อมูล 3 ถึง 27 Mbps ในช่องสัญญาณ 10 MHz และอัตราส่งข้อมูลเพิ่มขึ้นเป็น 6 ถึง 54 Mbps สำหรับช่องสัญญาณ 20 MHz ในโครงข่ายแอตสอกสำหรับยานพาหนะมีการเคลื่อนที่ของโน้ตความเร็วสูง การส่งแพ็กเก็ตรูปแบบ DCF (distributed coordination function)

มีความเสียหายหรือเกิดการชนของแพ็กเก็ตทำให้การใช้ RTS/CTS (request to send/clear to send) และวินโดว์ (Windows) ใน IEEE 802.11p ไม่ได้ช่วยแก้ปัญหาปัญหาโนดซ่อนเร้น (hidden terminal problem) และปัญหาโนดที่รับสัญญาณได้ (exposed terminal problem)

2.2.3 แบบจำลองการเคลื่อนที่ในโครงข่ายแอตฮอก

การจำลองแบบการเคลื่อนที่ของโนดมีการพิจารณาถึงการเคลื่อนที่ของโนดให้ใกล้เคียงกับสถานะจริง โดยมีการเปลี่ยนความเร็วและทิศทางการเคลื่อนที่ที่เกิดขึ้นตามช่วงเวลาที่เหมาะสม ในหัวข้อนี้อธิบายการทำงานของแบบจำลองการเคลื่อนที่ทั้งหมด 3 แบบ สำหรับโครงข่ายแอตฮอกที่เคลื่อนที่แบบสุ่มบนพื้นที่เปิดกว้าง (2 แบบ) และการเคลื่อนที่แบบจำลองการเคลื่อนที่ Car following model [38]

- Random walk mobility model: เป็นรูปแบบการจำลองการเคลื่อนที่ที่ง่ายซึ่งมีการสุ่มทิศทางและความเร็ว
- Random waypoint mobility model: แบบจำลองการเคลื่อนที่นี้จะมีช่วงเวลาทีโนดเคลื่อนที่จะหยุด ระหว่างการเปลี่ยนทิศทางและความเร็ว
- Car following model เป็นการจำลองการเคลื่อนที่เสมือนยานพาหนะวิ่งตามกันในสภาพแวดล้อมเมือง

แบบจำลองการเคลื่อนที่ของโครงข่ายแอตฮอกที่เคลื่อนที่แบบสุ่ม ในส่วนนี้จะอธิบายถึง 2 รูปแบบของแบบจำลองการเคลื่อนที่ ที่ถูกเสนอสำหรับประเมินประสิทธิภาพของโครงข่ายแอตฮอกคือ random walk mobility model และ random waypoint mobility model ซึ่งเป็นแบบจำลองพื้นฐานที่ถูกใช้โดยนักวิจัยมากที่สุด

2.2.3.1 แบบจำลองการเคลื่อนที่การเดินทางแบบสุ่ม (random walk mobility model)

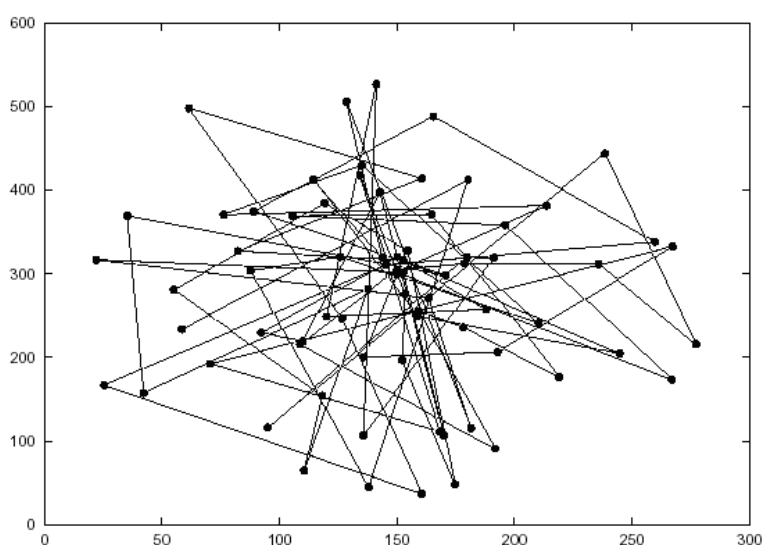
แบบจำลองการเคลื่อนที่นี้ โหนดเคลื่อนที่จะเคลื่อนที่จากตำแหน่งปัจจุบันไปยังตำแหน่งใหม่โดยการสุ่มเลือกทิศทางและความเร็วเพื่อที่จะเดินทางไป ความเร็วและทิศทางการเคลื่อนที่ใหม่นั้นเลือกแบบสุ่มจากช่วงที่สามารถระบุได้คือ $[-speedmin, speedmax]$ และ $[-0, 2\pi]$ ตามลำดับ แต่ละการเคลื่อนที่ใน random walk mobility model เกิดขึ้นในช่วงค่าคงที่เวลา t หรือค่าคงที่ระยะทางที่เคลื่อนที่ d ไปได้ ที่จุดสิ้นสุดการเคลื่อนที่ค่าทิศทางและความเร็วใหม่จะถูกสุ่มอีกครั้ง ถ้าโนดเคลื่อนที่เคลื่อนที่ไปชนขอบของพื้นที่จำลองมันจะกระเด็นออกมากลับมุมที่ถูกระบุ (ทิศทางตั้งฉาก) กับทิศทางที่เข้ามาจากนั้นโนดเคลื่อนที่ก็เคลื่อนที่ต่อตามเส้นทางนั้น

การจำลองแบบ random walk mobility model ได้ถูกพัฒนาขึ้นในหลายรูปแบบทั้งใน 1 มิติ 2 มิติ 3 มิติ และ d มิติ เพื่อพิสูจน์ว่า Random Walk ในหนึ่งหรือสองมิติจะวิ่งกลับเข้าหาจุดเริ่มต้นด้วยความน่าจะเป็นเท่ากับ 1 คุณสมบัตินี้ทำให้ทราบว่า random walk เป็นการแสดงแบบจำลองการเคลื่อนที่ ซึ่งการเคลื่อนที่จะอยู่รอบๆจุดเริ่มต้น โดยปราศจากการเคลื่อนที่ที่โนดเคลื่อนที่เคลื่อนที่ไปไกลจากจุดเริ่มต้นแล้วไม่กลับมาจุดเดิม

สำหรับ random walk mobility model 2 มิติ ได้รับความสนใจเป็นอย่างมาก เพราะว่าพื้นผิวโลกถูกจำลองเป็น 2 มิติ รูปที่ 2.9 แสดงตัวอย่างของการเคลื่อนที่ที่ใช้แบบจำลอง 2 มิติโดยโนดเคลื่อนที่จะเดินทางแต่ละครั้งด้วยเวลา 60 วินาทีก่อนเปลี่ยนทิศทางการเคลื่อนที่ ในรูป

ที่ 2.10 เป็นตัวอย่างการเคลื่อนที่ของโนดเคลื่อนที่โดยการเคลื่อนที่แต่ละครั้งมีระยะทางที่คงที่ก่อนการสุ่มเลือกเส้นทางใหม่

สำหรับ random walk mobility model เป็นรูปแบบการเคลื่อนที่ที่ไม่มีควมจำ (memory less) เพราะทิวทิศทางและความเร็วการเคลื่อนที่ในปัจจุบันของโนดเคลื่อนที่ที่ไม่ขึ้นกับทิวทิศทางและความเร็วของการเคลื่อนที่ในอดีต คุณสมบัตินี้ทำให้เกิดการเคลื่อนที่ที่ไม่เป็นจริงขึ้น เช่น เกิดการหยุดกะทันหัน และมีการเปลี่ยนทิวทิศทางกะทันหัน ถ้าระยะเวลาในการเคลื่อนที่แต่ละครั้ง (ระยะทาง) ถูกกำหนดให้มีค่าน้อยแล้วรูปแบบการเคลื่อนที่จะถูกสุ่มอยู่ในพื้นที่ที่จำกัดเป็นส่วนเล็กๆของพื้นที่จำลอง รูปที่ 2.10 แสดงประเภทการเคลื่อนที่แบบธรรมชาติสถิต (static nature) ตามที่แสดง โหนดเคลื่อนที่ไม่เคลื่อนที่ไปไกลจากตำแหน่งเริ่มต้น ดังนั้นถ้าเป้าหมายของการสังเกตประสิทธิภาพเพื่อสำหรับประเมินเครือข่ายกึ่งสถิต จะต้องตั้งค่าคงที่ที่จะเปลี่ยนทิวทิศทางการเคลื่อนที่ให้มีค่าน้อย

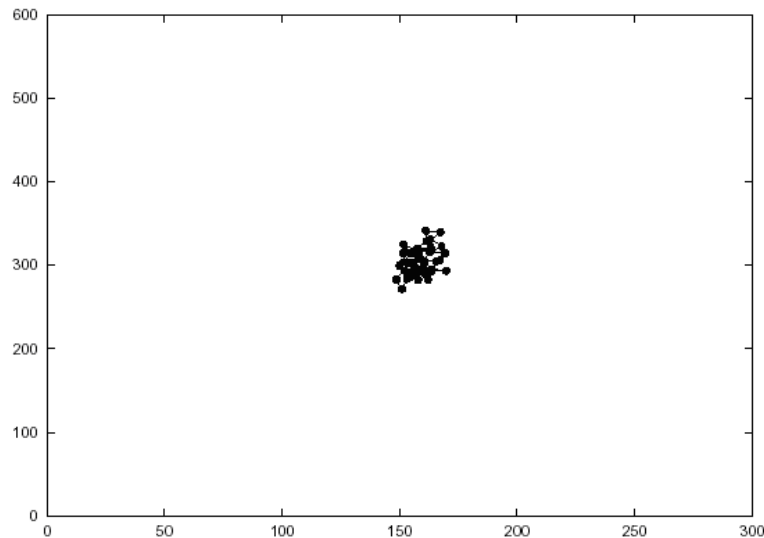


รูปที่ 2.9: รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่ที่ใช้ random walk ใน 2 มิติ

2.2.3.2 แบบจำลองการเคลื่อนที่แบบแรนดอมเวย์พอยต์ (random waypoint mobility model)

แบบจำลองการเคลื่อนที่แบบแรนดอมเวย์พอยต์ หรือ Random waypoint mobility model นั้นจะมีช่วงเวลาที่โนดหยุด (pause time) ในระหว่างการเปลี่ยนทิวทิศทางการเคลื่อนที่หรือความเร็ว โหนดเคลื่อนที่จะอยู่ที่ตำแหน่งหนึ่งในช่วงเวลาที่กำหนด เมื่อหมดช่วงเวลานั้นโนดเคลื่อนที่จะสุ่มเลือกตำแหน่งที่จะไปในพื้นที่จำลอง และความเร็วจะสุ่มระหว่าง $[-\text{minspeed}, \text{maxspeed}]$ หลังจากนั้นโนดเคลื่อนที่จะเคลื่อนที่ไปยังตำแหน่งที่ได้เลือกกับค่าความเร็วที่สุ่มได้ เมื่อเคลื่อนที่มาถึงตำแหน่งเป้าหมายโนดเคลื่อนที่จะหยุดที่ระยะเวลาที่กำหนด แล้วทำการสุ่มเลือกตำแหน่งกับความเร็วอีกครั้ง

จากรูปที่ 2.11 แสดงตัวอย่างการเคลื่อนที่ของโนดเคลื่อนที่ที่ใช้ random waypoint mobility model รูปแบบการเคลื่อนที่นี้มีความคล้ายกับ random walk mobility model คือถ้าช่วงเวลาที่หยุดเป็นศูนย์ และ $[-\text{minspeed}, \text{maxspeed}] = [-\text{speedmin}, \text{speedmax}]$



รูปที่ 2.10: รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่โดยใช้ random walk ใน 2 มิติ (ระยะทางคงที่)

2.2.3.3 แบบจำลองของการเดินทาง (Car following model)

รูปแบบของกระบวนการเดินทางที่มีการขับตามๆกันไปบนถนนที่มี ช่องทางเดินเพียงช่องทางเดียว (car following) การเดินทางลักษณะนี้เป็นรูปแบบที่พบเห็นในเมือง มีรูปแบบจำลองทางคณิตศาสตร์ที่ครบสมบูรณ์ และยังเป็นรูปแบบที่ง่ายเมื่อเทียบกับการเดินทางรูปแบบอื่นอีกด้วย ในการใช้งานจริงบนถนนที่มีหลายช่องทางเดินแต่มีสภาพการไหลที่ไม่คล่องตัว แบบจำลองการเดินทางแบบนี้จึงเป็นแบบจำลองที่ให้ความแม่นยำสูง เมื่อยานพาหนะเคลื่อนที่ตามกันไปด้วยระยะห่างเฉลี่ย S (space) และความเร็วเฉลี่ย V (velocity) จะได้ความสัมพันธ์ระหว่างระยะห่างและความเร็วสัมพันธ์กับความจุของช่องทางเดินทางเดียว C (capacity) ดังสมการที่ (2.2)

$$c = (1000) \frac{V}{S} \quad (2.2)$$

เมื่อ

C คือ ความจุของช่องทางเดินทางเดียว (คันต่อชั่วโมง)

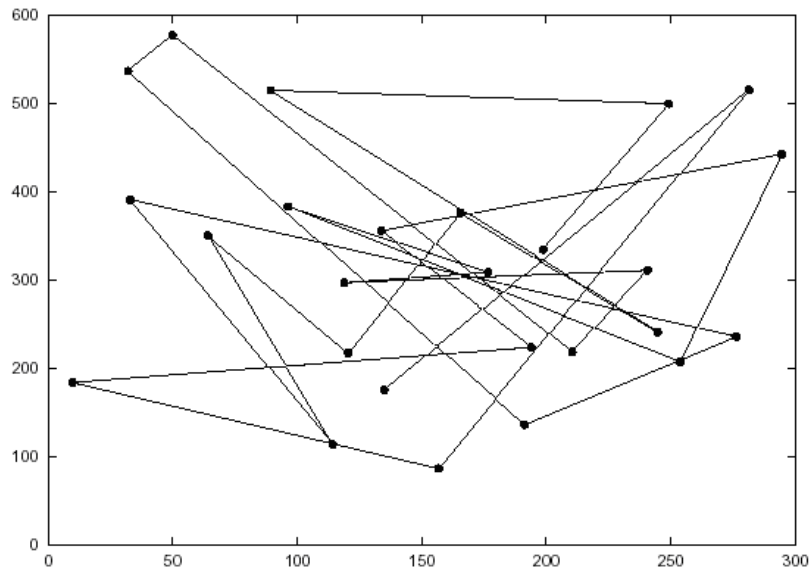
V คือ ความเร็วเฉลี่ย (กิโลเมตรต่อชั่วโมง)

S คือ ระยะห่างเฉลี่ยจากกันชนหลังของยานพาหนะคันหน้าถึงกันชนหน้าของยานพาหนะคันหลัง (เมตร)

นอกจากนี้ผลจากการศึกษาของ highway capacity manual (1950) กล่าวถึงความสัมพันธ์ ระหว่างความเร็วและระยะห่าง ดังสมการที่ (2.3)

$$S = \alpha + \beta V + \gamma V^2 \quad (2.3)$$

เมื่อค่าของสัมประสิทธิ์ α , β และ γ นั้นเป็นได้หลายค่า โดยความหมายทางกายภาพ สำหรับวิทยานิพนธ์ฉบับนี้ ให้ค่าจำกัดความ ของสัมประสิทธิ์ดังกล่าว ดังนี้



รูปที่ 2.11: รูปแบบการเคลื่อนที่ของโนดเคลื่อนที่โดยใช้แบบแรนดอมเวย์พอยด์

α คือ ความยาวยังผลของตัวถังรถ (effective vehicle length), L (เมตร)

β คือ ช่วงเวลาปฏิกิริยา (reaction time), T (วินาที)

γ คือ ส่วนกลับของสองเท่าอัตราชะลอสูงสุดของรถคันหลัง (reciprocal of twice the maximum average deceleration of a following vehicle)

ในส่วนของเทอม γV^2 เป็นการเพิ่มเข้ามาเพื่อสำรองระยะห่างระหว่างรถ เมื่อรถคันหน้าต้องการเบรคอย่างฉับพลันและไม่ให้เกิดการชน โดยมีค่าปกติ $\gamma \approx 0.023 \text{ sec}^2 / ft$

นอกจากนั้นยังสามารถตีความหมายทางกายภาพของสัมประสิทธิ์ γ ในรูปแบบไม่เป็นเชิงเส้นที่แม่นยำขึ้นได้สมการที่ (2.4)

$$\gamma = 0.5(a_f^{-1} - a_l^{-1}) \quad (2.4)$$

เมื่อ a_f และ a_l คืออัตราชะลอเฉลี่ยสูงสุดของรถคันตามและรถคันหน้าตามลำดับ สมการนี้ต้องการบอกถึงการอนุญาตให้ประสิทธิภาพการเบรคของรถทั้งสองนั้นต่างกันได้โดยไม่เกิดการชนกัน

สมการที่ (2.4) นี้สามารถใช้วิเคราะห์ได้จริงกับระบบที่รถแต่ละคันพยายามรักษาความเร็ว และระยะห่างให้คงที่ (การจราจรกระแสคงตัว (steady-state traffic stream)) การวิเคราะห์รถที่เคลื่อนที่ตามกันไปนี้จะเป็นการเชื่อมต่อการวิเคราะห์แบบจุลภาค (microscopic approaches) ซึ่งคำนึงถึงพฤติกรรมเคลื่อนที่ตามกันของรถแต่ละคันเป็นหลักกับการวิเคราะห์แบบมหภาค (macroscopic approaches) ซึ่งคำนึงถึงการไหลและความคงตัวของจราจรเป็นหลัก

ในส่วนของการพัฒนาแบบจำลอง(model development) แบบจำลองรถเคลื่อนที่ตามกันนี้มีสมมุติฐานว่า รถที่เคลื่อนที่ตามกันด้วยระยะห่างประมาณ 0 ถึง 125 เมตรนั้นจะมีความสัมพันธ์ เชื่อมโยงต่อกัน และมีการสมมุติว่าคนขับทุกคนอยู่ในสถานะตื่นตัวสามารถควบคุมระบบขับที่ ' และสั่งการควบคุมรถได้ โดยขับที่บนถนนทางเดินรถเดียวที่ไม่มีการแซงเกิดขึ้น ขึ้นอยู่กับปัจจัยดังนี้

- การรับรู้ (perception) คนขับจะรวบรวมข้อมูลรอบข้างที่มาจากการมองเห็นเป็นหลัก ข้อ-

มูลโดยมากจะมาจากการเคลื่อนไหวของรถคันหน้าและสิ่งที่คุณซบนั้นรับรู้ได้อย่างรวดเร็ว เช่น ความเร็วยานพาหนะ ความเร็วสัมพัทธ์ ความเร่ง การเบรคกะทันหัน ระยะห่าง อัตราการเปลี่ยนแปลงระยะห่างการชน เป็นต้น

- การตัดสินใจ (decision making) คนซบจะตีความข้อมูลนั้นจากตัวอย่างและเหตุการณ์ต่างๆ เป็นระยะเวลาหนึ่ง การตีความเพื่อตัดสินใจนั้นอยู่ บนพื้นฐานของความทราปรู้อย่างยานพาหนะ คันนั้นๆ รวมทั้งประสบการณ์ของผู้ซบที่เอง จากประสบการณ์ในปัจจุบันและที่ผ่านมาซึ่งนำไปสู่การตอบสนองอัตโนมัติหรือที่เรียกว่า ทักษะการซบซี้
- การควบคุม (control) คนซบที่มีทักษะจะสามารถกระทำการควบคุมรถได้อย่างนุ่มนวลและฉับพลัน ผ่านการป้อนกลับแบบพลวัตจากสถานะของรถและถนน

ที่ผ่านมา Tustin (1947) Ellson (1949) และ Taylor (1949) มีการจำลองคนซบออกเป็นแบบจำลองทางคณิตศาสตร์ผ่านฟังก์ชันถ่ายโอน แต่ฟังก์ชันถ่ายโอนที่เป็นได้นั้นขึ้นอยู่กับ การตอบสนองของผู้ซบซี้ ต่อเมื่อมีสิ่งเร้าที่เกิดขึ้น ผู้ซบซี้ก็จะตอบสนองดังสมการความสัมพันธ์สิ่งเร้าต่อการตอบสนอง ดังสมการที่ (2.5)

$$Respond(t) = \lambda \times Stimulus(t) \quad (2.5)$$

เมื่อ λ เป็นตัวประกอบสัดส่วนสัมพันธ์สิ่งเร้าต่อการตอบสนองหรือการควบคุม โดยที่สิ่งเร้าประกอบด้วยหลายส่วน เช่น ความเร็ว ความเร็วสัมพัทธ์ ระยะห่างระหว่างรถ อัตราเร่ง สมรรถนะของยานพาหนะ ระดับกระตุ้นขีดเริ่มของผู้ซบซี้ อย่างไรก็ตามปัจจัยบางตัวอาจส่งผลเพียงเล็กน้อยต่อการตัดสินใจ ดังนั้นแบบจำลองจึงสมมุติว่าผู้ซบซี้พยายามตามรถคันหน้า หลีกเลี่ยงการชนด้วยความเร็วสัมพัทธ์มีการป้องกันการชนและการเร่งหน้อยอย่างรวดเร็วของรถคันหน้า กำหนดตัวแปรความเร็วสัมพัทธ์เป็นตัวแปรหลักที่ส่งผลกระทบต่อการตอบสนอง ทันทีที่ผู้ซบซี้สังเกตเห็นสิ่งเร้า ผู้ซบซี้จะตอบสนองเมื่อเวลาผ่านไปเพียงเล็กน้อย ทำให้เกิดการประวิงเวลา (T) ผู้ซบซี้จะตอบสนองกลับมาในรูปของความเร่งและความหน่วงด้วยการเหยียบคันเร่งหรือเหยียบเบรค ผลของการตอบสนอง สามารถเขียนได้ในรูปของความเร่งได้ดังสมการที่ (2.6)

$$Respond(t) = a_f(t) = \ddot{x}_f(t) \quad (2.6)$$

เมื่อ $x_i(t)$ ตำแหน่งของรถคันที่ i บนถนน ณ เวลา t จากสมการ (2.5) และ (2.6) จะได้ความสัมพันธ์ของสิ่งเร้าต่อการตอบสนอง (Chandler et al. 1958) เป็น

$$\ddot{x}_f(t) = \lambda [\dot{x}_l(t - T) - \dot{x}_f(t - T)] \quad (2.7)$$

ซึ่งเทียบเท่ากับ

$$\ddot{x}_f(t + T) = \lambda [\dot{x}_l(t) - \dot{x}_f(t)] \quad (2.8)$$

สำหรับแบบจำลองที่แม่นยำกว่านั้นต้องการการจำลององค์ประกอบย่อย ได้แก่ แบบจำลองของคนซบ ปฏิกริยาโต้ตอบระหว่างยานพาหนะและถนนความเชื่อมโยงระหว่างคนซบ และยานพาหนะ นอกจากนี้ยังรวมถึงความสัมพันธ์ระหว่างรถที่พิจารณาเกี่ยวกับยานพาหนะที่อยู่ไกลออกไปอีกด้วย

2.3 โครงข่ายชื่อข้อมูล (named data network หรือ NDN)

สถาปัตยกรรมชื่อข้อมูล หรือ โครงข่ายชื่อข้อมูล (named data network (NDN)) [14] เป็นสถาปัตยกรรมอินเทอร์เน็ตในอนาคตที่นำมาแก้ปัญหาของสถาปัตยกรรมอินเทอร์เน็ต TCP/IP ซึ่ง NDN มีรากฐานมาจากโครงการก่อนหน้านี้ที่มีชื่อว่า โครงข่ายเนื้อหา (content-centric networking (CCN)) โดย Van Jacobson [13] ได้นำเสนอในเดือนพฤศจิกายน 2553 และได้รับทุนจากมูลนิธิวิทยาศาสตร์แห่งชาติ (National Science Foundation - NSF) ภายใต้ชื่อโครงการ Future Internet Architecture (FIA) ซึ่งเกิดแนวความคิดมาจากสถาปัตยกรรม TCP/IP ที่เน้นการเข้าถึงข้อมูลจากที่อยู่ไอพี มาเป็นการเข้าถึงข้อมูลด้วยชื่อของข้อมูล อีกทั้งโครงข่ายการสื่อสารควรเน้นให้ผู้ใช้งานเข้าถึงข้อมูลด้วยชื่อของข้อมูลแทนที่การอ้างอิงตำแหน่งทางกายภาพ หรือที่อยู่ไอพีของข้อมูล

รูปแบบแพ็กเก็ตการสื่อสารของ NDN แบ่งเป็น 2 รูปแบบคือ แพ็กเก็ตการร้องขอ (interesting packet) หรือเทียบได้ว่าเป็นคำถาม และ แพ็กเก็ตข้อมูล (data packet) หรือเทียบได้ว่าเป็นคำตอบ โดยมีรูปแบบของแพ็กเก็ต ดังรูปที่ 2.12

จากรูปที่ 2.12 แสดงส่วนประกอบของแพ็กเก็ตการร้องขอ ประกอบไปด้วย

- Name สำหรับระบุชื่อข้อมูลที่ต้องการ
- Selector สำหรับระบุว่าข้อมูลต่าง ๆ เช่น ลำดับการอ้างอิงถึง ผู้ให้บริการ เป็นต้น
- NONCE สำหรับระบุแพ็กเก็ตการร้องขอแต่ละแพ็กเก็ต โดยสร้างตัวเลขสุ่ม เพื่อไม่ให้ข้อมูลแต่ละข้อมูลมีความซ้ำซ้อน
- Guider สำหรับระบุขอบเขต หรือ เวลาหมดอายุของแพ็กเก็ตการร้องขอ

และส่วนประกอบของแพ็กเก็ตข้อมูล ประกอบไปด้วย

- Name สำหรับระบุชื่อข้อมูลที่ต้องการ
- Meta Info. สำหรับระบุชนิดของข้อมูล ข้อมูลใหม่ หรือ ช่วงเวลา เป็นต้น
- Content สำหรับบรรจุข้อมูล
- Signature สำหรับเข้ารหัสของแพ็กเก็ต

Interest Packet	Data Packet
Name	Name
Selector: order preference, publisher filter, exclude filter, ...	Meta Info: content type, freshness period, ...
NONCE	Content
Guiders: scope, Interest lifetime	Signature: signature type, key locator, signature bits, ...

รูปที่ 2.12: รูปแบบแพ็กเก็ตในสถาปัตยกรรมข้อมูลชื่อ

โพรโทคอลแบบ TCP/IP ที่ใช้อยู่ในปัจจุบัน จะค้นหาโนดปลายทางจากที่อยู่ของไอพี ดังรูปที่ 2.13ก กล่าวคือ เมื่อโนดได้รับแพ็กเก็ตการร้องขอ โนดจะตรวจสอบที่อยู่ไอพีที่ส่วนหัวของแพ็กเก็ตในตาราง FIB เพื่อที่จะส่งต่อไปยังโนดปลายทางที่ถูกระบุที่อยู่ไอพีนั้นไว้

สถาปัตยกรรมชื่อข้อมูลมีโครงสร้างการแจกจ่ายข้อมูล 3 ส่วน ดังรูปที่ 2.13ข โดยแบ่งออกเป็น

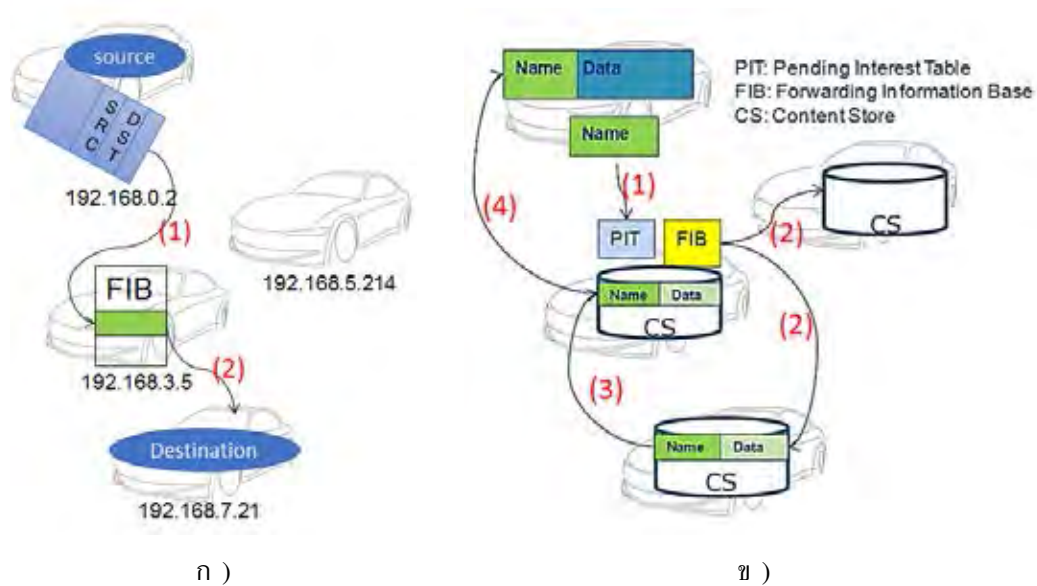
1. **Pending interest table : PIT** ใช้สำหรับจัดเก็บแพ็กเก็ตการร้องขอทั้งหมดที่โนดส่งออกไป แต่ยังไม่ได้รับการตอบกลับมา โดยที่รายการแต่ละรายการจะถูกบันทึกชื่อข้อมูลที่ส่งออกไปจากโนดที่รับและโนดที่ได้ส่งต่อ
2. **Forwarding information base: FIB** ใช้สำหรับเก็บรายละเอียดของเส้นทางที่ส่งแพ็กเก็ต ซึ่งการร้องขออาจจะมีการส่งแพ็กเก็ตการร้องขอหลาย ๆ แพ็กเก็ต โดยที่แพ็กเก็ตแต่ละตัวจะถูกบันทึกลงตารางเส้นทาง เพื่อบอกส่วนต่อประสานขาออกจากโนด
3. **Content store : CS** ใช้สำหรับเก็บข้อมูล

เมื่อโนดได้รับแพ็กเก็ตการร้องขอ สถาปัตยกรรมชื่อข้อมูลจะทำงานดังรูปที่ 2.13ข) โดยมีขั้นตอนดังนี้

- เมื่อโนดต้นทางต้องการข้อมูล โนดต้นทางจะส่งแพ็กเก็ตการร้องขอที่มีชื่อข้อมูลที่ต้องการแพร่กระจายออกไป
- เมื่อโนดเพื่อนบ้านได้รับแพ็กเก็ตการร้องขอที่มีชื่อข้อมูล โนดเพื่อนบ้านจะตรวจสอบใน CS ก่อน หากพบว่ามีแพ็กเก็ตข้อมูลใน CS ที่ตรงกับแพ็กเก็ตการร้องขอ โนดเพื่อนบ้านจะส่งแพ็กเก็ตข้อมูลบนส่วนต่อประสานเดียวกันกลับไปยังโนดต้นทาง แต่หากไม่พบแพ็กเก็ตข้อมูลใน CS โนดเพื่อนบ้านจะตรวจสอบที่ตาราง PIT ต่อไป
- ในขั้นตอนการตรวจสอบที่ตาราง PIT หากโนดเพื่อนบ้านตรวจสอบชื่อข้อมูลในแพ็กเก็ตการร้องขอที่ตาราง PIT แล้วพบว่าชื่อข้อมูลในแพ็กเก็ตการร้องขอตรงกับข้อมูลในตารางแพ็กเก็ตการร้องขอนี้จะถูกยกเลิก แต่หากชื่อข้อมูลในแพ็กเก็ตการร้องขอไม่ตรงกับข้อมูลในตาราง โนดเพื่อนบ้านจะตรวจสอบที่ตาราง FIB ต่อไป
- ในขั้นตอนการตรวจสอบที่ตาราง FIB หากโนดเพื่อนบ้านตรวจสอบชื่อข้อมูลในแพ็กเก็ตการร้องขอที่ตาราง FIB แล้วพบว่าชื่อข้อมูลในแพ็กเก็ตการร้องขอตรงกับข้อมูลในตารางแพ็กเก็ตการร้องขอจะถูกส่งออกไปบนส่วนต่อประสานขาออกที่ระบุไว้ในรายการ แต่หากชื่อข้อมูลในแพ็กเก็ตการร้องขอไม่ตรงกับข้อมูลในตาราง แพ็กเก็ตการร้องขอจะถูกส่งแพร่กระจายออกไปบนส่วนต่อประสานขาออก ซึ่งไม่ว่าชื่อข้อมูลในแพ็กเก็ตการร้องขอจะตรงกับข้อมูลในตาราง FIB หรือไม่ ข้อมูลจะถูกเพิ่มในตาราง PIT เสมอ เพื่อรอการตอบกลับแพ็กเก็ตข้อมูลไปยังโนดที่ต้องการข้อมูลต่อไป

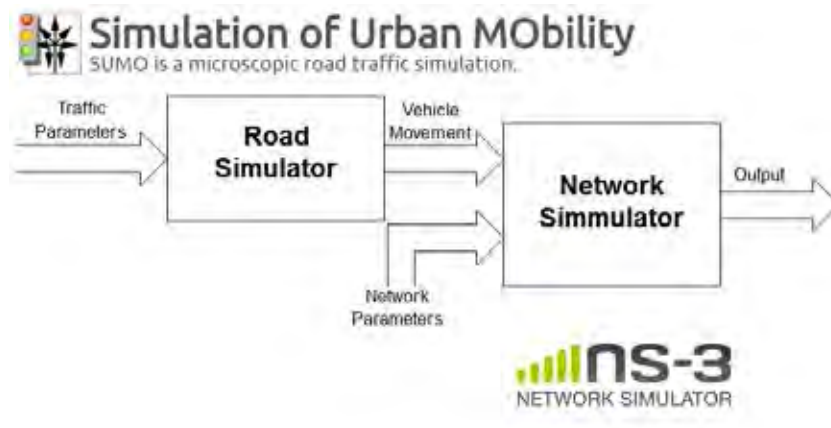
2.4 โปรแกรมการจำลอง

ในรูปที่ 2.14 แสดงขั้นตอนการทำงานระหว่างโปรแกรมการจำลองระบบการจราจร และโปรแกรมการจำลองระบบการสื่อสาร โดยใช้โปรแกรม SUMO และโปรแกรม NS-3 ตามลำดับ กระบวนการทำงานเริ่มจากใส่พารามิเตอร์ที่ใช้ในการสร้างถนนและยานพาหนะเข้าสู่โปรแกรมการจำลองระบบการจราจร หลังจากนั้น เมื่อได้ข้อมูลที่แสดงการเคลื่อนที่ของยานพาหนะจากการจำลอง



รูปที่ 2.13: ก) โครงข่ายของระบบที่มีอยู่ (TCP/IP) และ ข) โครงข่ายชื่อข้อมูล

ระบบการจราจรแล้ว ข้อมูลนี้จะถูกใส่ไปพร้อมกับพารามิเตอร์การจำลองระบบการสื่อสาร เพื่อวิเคราะห์การจำลองระบบการสื่อสารต่อไป

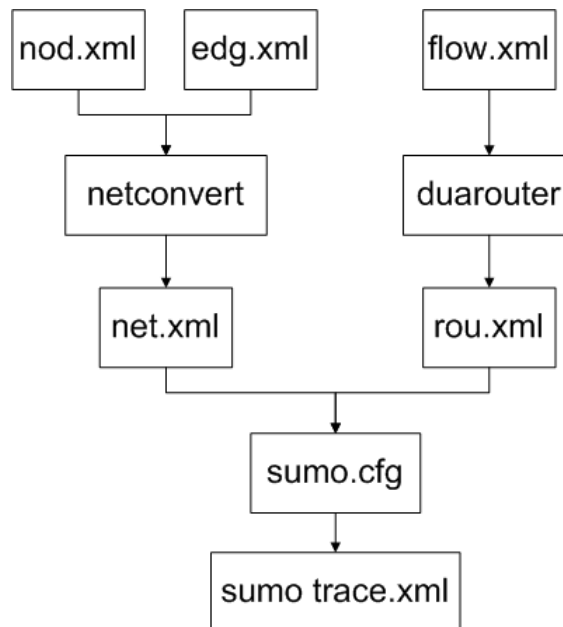


รูปที่ 2.14: แสดงการจำลองของระบบ

2.4.1 โปรแกรมจำลองระบบการจราจร

โปรแกรมการจำลองระบบการจราจรในเมือง หรือ Simulation of Urban MObility (SUMO) [46] ซึ่งเป็นโปรแกรมการจำลองระบบการจราจรประเภทจุลภาคที่ใช้การวิเคราะห์ที่ค่อนข้างละเอียดแบบคันท่อคั้น โดยจะตรงข้ามกับประเภทมหภาคที่พิจารณาในระบบการจราจรที่เป็นกลุ่มก้อน เนื่องจากรูปแบบไฟล์จากการจำลองโปรแกรม SUMO ไม่สามารถนำไปใช้งานในโปรแกรม NS-3 ได้

จึงจำเป็นต้องเปลี่ยนรูปแบบไฟล์ที่ได้จากโปรแกรม SUMO ให้อยู่ในรูปแบบไฟล์ของโปรแกรม NS-2 ซึ่งได้แก่ข้อมูลที่แสดงการเคลื่อนที่ของโหนด ก่อนที่จะเข้าสู่การจำลองในโปรแกรม NS-3 ต่อไป



รูปที่ 2.15: การสร้างแบบจำลองการเคลื่อนที่ของยานพาหนะบนถนน

ในรูปที่ 2.15 ได้แสดงขั้นตอนในการสร้างแบบจำลองการเคลื่อนที่ของยานพาหนะบนถนน โดยมีรายละเอียดของขั้นตอนดังนี้

- nod.xml และ edg.xml ป้อนค่าเพื่อสร้างเส้นทางถนน เช่น ขนาด 900 x 900 ตารางเมตร
- netconvert เปลี่ยนค่าที่ป้อนใน nod.xml และ edg.xml ให้เป็นถนนใน net.xml
- flow.xml ป้อนค่าเพื่อจำลองยานพาหนะ และเส้นทางที่ยานพาหนะเคลื่อนที่
- duarouter จะทำการสร้างข้อมูลที่ใส่ภายใน flow.xml ออกมาเป็นการเคลื่อนที่ของยานพาหนะ ในไฟล์ rou.xml
- sumo.cfg รวมไฟล์ระหว่าง net.xml และ rou.xml ได้ไฟล์ sumotrace.xml ซึ่งแสดงผลในรูปที่ 2.16g พร้อมกับตารางบอกตำแหน่งของยานพาหนะ ณ แต่ละเวลาทุกๆวินาที

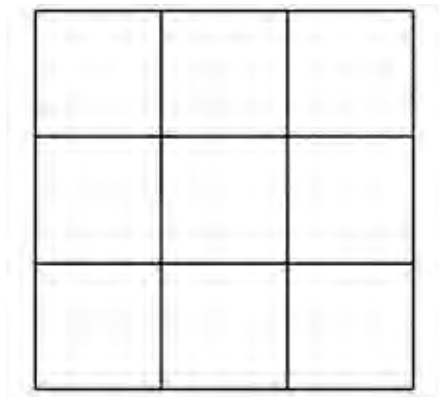
หลังจากที่สร้างไฟล์ sumotrace.xml แล้วจะใช้ชุดคำสั่งแปลงไฟล์จากโปรแกรม sumo ให้สามารถนำไปใช้ในโปรแกรม NS-3 ดังนี้

```

1 traceExporter.py --fcd-input sumotrace.xml --ns2mobility-output
2 ns3mobility.tcl

```

ซึ่งจะแปลงไฟล์ sumotrace.xml ให้กลายเป็นไฟล์สำหรับนำไปใช้ใน NS-3 โดยได้ไฟล์ชื่อ ns3mobility.tcl ซึ่งแสดงผลในรูปที่ 2.16ข



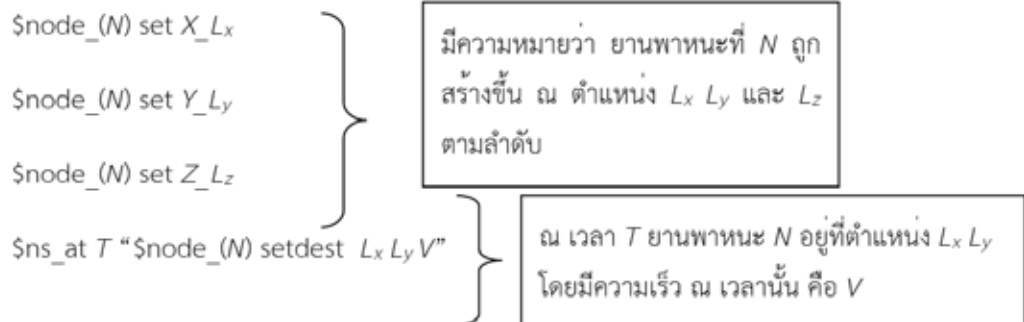
ก)

```

14 $node_(2) set X_ 1.65
15 $node_(2) set Y_ 8.35
16 $node_(2) set Z_ 0
17 $ns_ at 3.0 "$node_(2) setdest 1.65 8.35 0.00"
18 $ns_ at 3.0 "$node_(1) setdest 18.6 898.35 4.94"
19 $ns_ at 4.0 "$node_(0) setdest 1.65 22.04 4.03"
20 $ns_ at 4.0 "$node_(2) setdest 1.65 18.24 1.89"
21 $ns_ at 4.0 "$node_(1) setdest 22.89 898.35 4.29"
22 $node_(3) set X_ 8.35
23 $node_(3) set Y_ 898.35
24 $node_(3) set Z_ 0
25 $ns_ at 4.0 "$node_(3) setdest 8.35 898.35 0.00"
26 $ns_ at 5.0 "$node_(0) setdest 1.65 26.07 4.03"
27 $ns_ at 5.0 "$node_(2) setdest 1.65 14.53 4.29"
28 $ns_ at 5.0 "$node_(1) setdest 27.02 898.35 4.13"
29 $ns_ at 5.0 "$node_(3) setdest 10.34 898.35 1.99"
30 $ns_ at 6.0 "$node_(0) setdest 1.65 30.48 4.41"
31 $ns_ at 6.0 "$node_(2) setdest 1.65 17.84 3.30"
    
```

ข)

รูปที่ 2.16: ก) การเคลื่อนที่ของยานพาหนะบนแบบจำลอง และ ข) ส่วนหนึ่งของไฟล์ ns3mobility.tcl บรรจุข้อมูลสถานะภาพของทุกยานพาหนะ ณ เวลาต่างๆ



รูปที่ 2.17: อธิบายความหมายของ ns3mobility.tcl

จากรูปที่ 2.17 แสดงความหมายของไฟล์ ns3mobility.tcl โดยที่ N แทน ยานพาหนะลำดับที่ N , T คือ ณ เวลาที่ T , L_x คือพิกัดในแกน X , L_y คือพิกัดในแกน Y , L_z คือพิกัดในแกน Z (ในที่นี้เป็น 0) และ V คือ ความเร็ว เมตร/วินาที

นอกจากนี้ ในโปรแกรม sumo มีคำสั่งเพิ่มเติมในการสร้างไฟล์ผลลัพธ์ หรือ (trace files) ดังแสดงในรูปที่ 2.18

Option	Description
--persons <BOOL>	Whether persons should be included in the trace. default: False
--penetration <FLOAT>	A float between 0 and 1, defining the probability of a vehicle to be included in the output. default: 1
--gps-blur <FLOAT>	A float defining a random offset applied on the vehicle positions before exporting them. default: 0
--boundary <FLOAT>,<FLOAT>,<FLOAT>,<FLOAT>	Defines the bounding box as 'xmin ymin xmax ymax'
--delta-t <TIME>	Defines the time sampling. If given, only every <TIME> time step will be exported.
-d <TIME>	
--begin <TIME>	Time steps before <TIME> are not exported
-b <TIME>	
--end <TIME>	Time steps after <TIME> are not exported. Please note that some issues are known, here
-e <TIME>	
--seed <INT>	The randomizer seed to use.
-s <INT>	
--base-date <INT>	The base date in seconds since the epoch (1970-01-01 00:00:00), the default is the current local time.
-S <INT>	

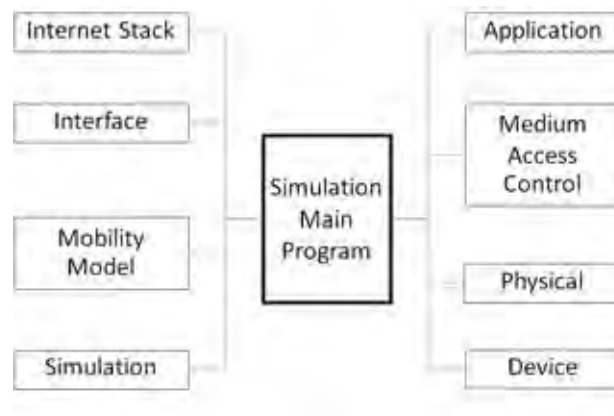
รูปที่ 2.18: ตัวเลือกของการประมวลผลสำหรับ trace file

เนื่องจากต้องการให้มีจำนวนรถที่มีค่าแตกต่างกัน จากการเพิ่มคำสั่ง --penetration ซึ่งสามารถเรียกใช้ค่าเป็นแบบ float มีค่าตั้งแต่ 0-1 ดังนั้นจะทำการลดจำนวนยานพาหนะทั้งหมดให้เหลือ 10% จะใช้ชุดคำสั่งดังนี้

```
1 traceExporter.py --penetration 0.1 --fcd-input sumotrace.xml
2 --ns2mobility-output ns3mobility.tcl
```

2.4.2 โปรแกรมจำลองโครงข่ายเอ็นเอส-3

ในกลุ่มของนักวิจัยมีการใช้โปรแกรมจำลองโครงข่ายเพราะเป็นกระบวนการวิจัยที่ได้รับความนิยมอย่างแพร่หลาย เนื่องจากการจำลองโครงข่ายมีความยืดหยุ่นในเรื่องของการออกแบบสภาพแวดล้อม เทคโนโลยีที่ใช้ ลักษณะอุปกรณ์ และต้นทุนของงานวิจัยที่ต่ำกว่าการใช้อุปกรณ์จริง ปัจจุบันมีโปรแกรมจำลองโครงข่ายหลายโปรแกรมที่ได้รับความนิยมอย่างกว้างขวาง และงานวิจัย [39] ทำการทดสอบประสิทธิภาพของโปรแกรมจำลองโครงข่าย ได้แก่ เอ็นเอส-2 (ns-2) [40] เอ็นเอส-3 [41] OMNeT++ [42] Jist [43] และ SimPy [44] ได้ทำการเปรียบเทียบจากการใช้หน่วยความจำ (memory usage) และการใช้ซีพียู (CPU utilization) เมื่อโปรแกรมจำลองเน็ตในระบบตั้งแต่ 400-2000 โหนด ในเวลา 500 วินาที บนพื้นที่จำลองขนาด 1000 x 1000 แล้วพบว่า NS-3 แสดงให้เห็นถึงประสิทธิภาพทั้งหมดที่ดีที่สุด และ NS-3 มีตัวช่วยในการพัฒนาทำให้ผู้ใช้งานไม่จำเป็นต้องสร้างโครงสร้างของโครงข่ายเองทั้งหมด เพียงแต่ทำการเขียนโปรแกรมหลักเพื่อเรียกใช้ตัวช่วยต่าง ๆ และยังสามารถเพิ่มโมดูลการทำงานของโครงข่ายอื่นเข้ามาบนโปรแกรม NS-3 ได้ ดังแสดงในรูปที่ 2.19 ดังนั้นการเลือกใช้โปรแกรม NS-3 ในการทดลองเพื่อศึกษาการทำงานของโพรโทคอล จึงเป็นตัวเลือกที่ดีที่สุด



รูปที่ 2.19: องค์ประกอบของโปรแกรมเอ็นเอส-3 [45]

โปรแกรม NS-3 เป็นโปรแกรมจำลองโครงข่ายแบบเหตุการณ์วิฤต (discrete-event network simulator) โดยมีจุดมุ่งหมายเพื่อการวิจัยสำหรับนักวิจัยแบบสาธารณะ ได้รับการอนุญาตภายใต้โครงการ GNU GPLv2 โปรแกรม NS-3 เป็นซอฟต์แวร์ที่ไม่เสียค่าใช้จ่าย สามารถดาวน์โหลดและศึกษาเพิ่มเติมได้จาก <http://www.nsnam.org> สำหรับภาษาที่ถูกใช้ในโปรแกรมคือ ภาษา C++ หรือ Python โดยที่โมเดลส่วนใหญ่จะใช้ภาษา C++ ดังนั้นก่อนที่จะสามารถใช้งานโปรแกรม NS-3 ได้จะต้องมีความรู้พื้นฐานในการเขียนภาษา C++ มาก่อน ในส่วนของการรันโปรแกรมนั้น แนะนำให้ใช้ใน Linux เพื่อใช้สำหรับการจำลอง แต่ก็สามารถรันโปรแกรมใน Window ได้ เมื่อลง Cygwin แต่ในบางโมเดลไม่สามารถทำงานได้เนื่องจากขาดไลบรารีบางอย่างที่มีเฉพาะใน Linux เท่านั้น

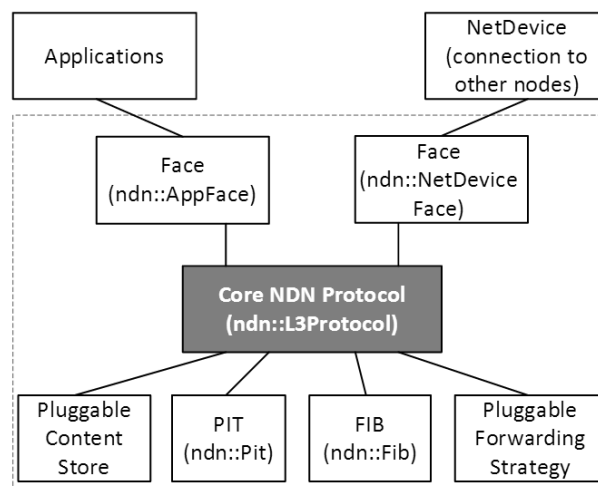
องค์ประกอบของโปรแกรมจำลองโครงข่ายเอ็นเอส-3 ประกอบด้วย

1. อินเทอร์เน็ตสแตค (**internet stack**) ช่วยจำลองโพรโทคอลของระบบอินเทอร์เน็ต
2. ส่วนต่อประสาน (**interface**) ช่วยจำลองส่วนต่อประสานระหว่างอุปกรณ์เชื่อมต่อจำลองและระบบโครงข่ายจำลอง
3. แบบจำลองการเคลื่อนที่ (**mobility model**) ช่วยจำลองการเคลื่อนที่ของโหนด เช่น การเคลื่อนที่แบบความเร็วคงที่ เคลื่อนที่แบบมีความเร่งคงที่ เคลื่อนที่แบบสุ่มความเร็ว เป็นต้น ถ้าไม่มีการตั้งค่าโหนดจะไม่มีเคลื่อนที่
4. โปรแกรมจำลอง (**simulation**) ช่วยควบคุมการจำลองระบบ เช่น ให้เริ่มจำลองที่วินาทีที่เท่าไร หยุดการจำลองวินาทีที่เท่าไร เริ่มจำลองครั้งต่อไปที่วินาทีที่เท่าไร เป็นต้น
5. โปรแกรมประยุกต์ (**application**) ช่วยจำลองโปรแกรมการทำงานของโหนด ปกติโปรแกรมคอมพิวเตอร์จะถูกแบ่งออกเป็น 2 ประเภท คือ โปรแกรมที่ใช้ในการจัดการทรัพยากรของเครื่องคอมพิวเตอร์และโปรแกรมประยุกต์ ในเอ็นเอส-3 จะไม่มีการสร้างระบบปฏิบัติการจำลองจะมีเฉพาะการจำลองโปรแกรมประยุกต์ขึ้นมาใน โปรแกรมโดยใช้ ภาษาC++
6. ส่วนควบคุมการเข้าใช้งานสื่อกลาง (**medium access control**) ช่วยจำลองการทำงานของส่วนควบคุมการเข้าใช้งานสื่อกลาง ซึ่งอยู่ในแผงวงจรของอุปกรณ์เชื่อมต่อ

7. **ชั้นกายภาพ (physical)** ช่วยจำลองการทำงานในชั้นกายภาพและสิ่งแวดล้อมที่จำลองขึ้นมาในระบบ เช่น แบบจำลองการสูญเสียชนิดต่าง ๆ เป็นต้น
8. **อุปกรณ์ (device)** ช่วยจำลองชนิดของอุปกรณ์เชื่อมต่อและโปรแกรมขับอุปกรณ์ (software driver) ที่จะใช้ติดตั้งในโนด เช่น แผ่นวงจรต่อประสานข่ายงาน (network interface card) เป็นต้น

2.4.3 โปรแกรมจำลองโครงข่ายเอ็นดีเอ็นซิม (ndnSim)

โปรแกรมจำลองเอ็นดีเอ็นซิม (ndnSim) เป็นโมดูลส่วนเสริมของโปรแกรม NS-3 ที่มีการใช้โมเดลการสื่อสารระบบโครงข่ายชื่อข้อมูล (Named Data Networking: NDN) ทำให้ ndnSIM สามารถใช้งานร่วมกันกับรูปแบบโพรโทคอลชั้นเชื่อมโยง (point-to-point CSMA wireless) ชั้นโครงข่าย (IPv4 IPv6) และชั้นขนส่ง (TCP UDP) ในโปรแกรม NS-3 อย่างไรก็ตามการจำลองจะดำเนินการในรูปแบบโมดูล (module) โดยใช้คลาส C++ แยกต่างหาก เพื่อจำลองพฤติกรรมของแต่ละเอนทิตีของโครงข่ายชื่อข้อมูล (PIT FIB CS) และส่วนต่อประสานสำหรับโครงข่าย เป็นต้น โครงสร้างแบบแยกส่วนนี้ช่วยให้เอนทิตีใด ๆ สามารถปรับเปลี่ยนหรือเปลี่ยนได้โดยง่ายโดยไม่มีผลกระทบใด ๆ นอกจากนี้ชุดคำสั่งโปรแกรมต่าง ๆ ยังมีชุดของส่วนต่อประสาน ดังแสดงในรูปที่ 2.20

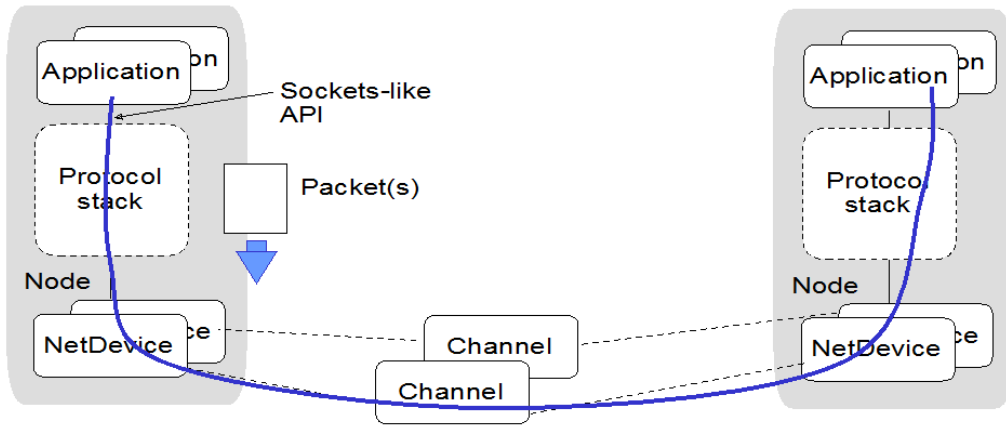


รูปที่ 2.20: รูปแบบของโครงสร้างของโปรแกรมเอ็นดีเอ็นซิม

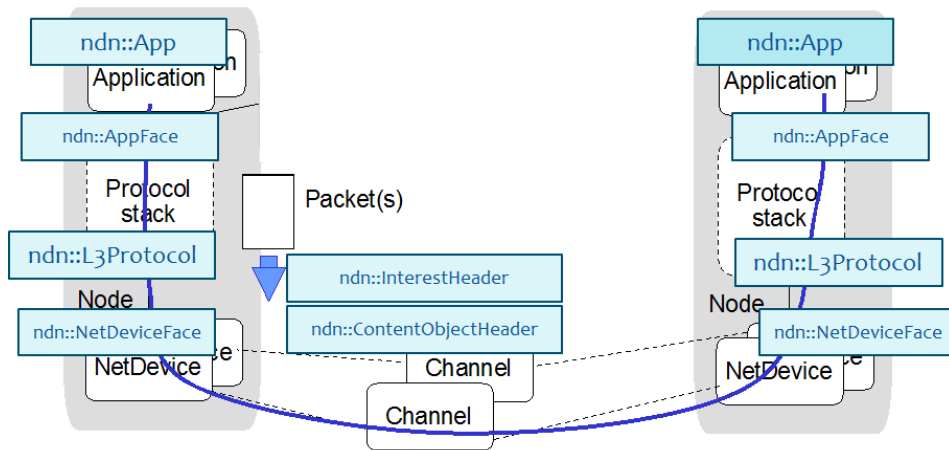
จากส่วนประกอบที่เพิ่มเข้าสามารถแสดงเพิ่มเติมจากรูปที่ 2.21 แสดงรูปแบบจำลองโครงสร้างของโปรแกรมเอ็นเอส-3 โดยโมดูลที่เพิ่มเข้ามาแสดงในรูปที่ 2.22

องค์ประกอบที่แสดงเพิ่มเติมของโปรแกรมเอ็นดีเอ็นซิม ได้แก่

- **ndn::L3Protocol:** เป็นองค์ประกอบหลักของโปรแกรม สำหรับโพรโทคอลของแพ็กเก็ตการร้องขอและแพ็กเก็ตข้อมูล จากเลเยอร์บนสู่เลเยอร์ล่างผ่าน ndn::Face
- **ndn::Face:** เป็นส่วนต่อประสาน เพื่อใช้สื่อสารระหว่างโปรแกรมประยุกต์และส่วนประกอบอื่น ๆ



รูปที่ 2.21: รูปแบบการจำลองของโปรแกรมเอ็นเอส-3



รูปที่ 2.22: รูปแบบการจำลองของโปรแกรมเอ็นดีเอ็นซีเอ็มโดยเพิ่มเติมจากโปรแกรมเอ็นเอส-3

- ndn::ContentStore: เป็นที่จัดเก็บข้อมูล
- ndn::Pit: เป็นตารางสำหรับจัดเก็บแพ็กเก็ตการร้องขอที่รอการตอบกลับ
- ndn::Fib: เป็นตารางสำหรับจัดเก็บส่วนต่อประสานขาออกของการส่งต่อแพ็กเก็ตการร้องขอ
- ndn::ForwardingStrategy: เป็นขั้นตอนของกระบวนการส่งต่อในโครงข่ายชื่อข้อมูล

บทที่ 3

การออกแบบโพรโทคอล

บทที่ผ่านมาได้กล่าวถึงแนวทางและความสำคัญของระบบการจราจรอัจฉริยะบนท้องถนน สำหรับบทนี้จะกล่าวถึงการพัฒนาโพรโทคอลการแจกจ่ายข้อมูลสำหรับโครงข่ายแอตฮอกสำหรับยานพาหนะ เมื่อพิจารณาในเรื่องของอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จที่เกิดขึ้นโดยปกติมีค่าน้อยมาก ดังนั้นจุดประสงค์ในการออกแบบโพรโทคอลเพื่อให้อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จมีความสำเร็จสูงขึ้น และใช้เวลาการแจกจ่ายข้อมูลน้อยลง ในงานวิจัยที่ผ่านมา ได้นำโครงข่ายชื่อข้อมูลมาปรับใช้กับโครงข่ายแอตฮอกของยานพาหนะเพื่อใช้ในการแจกจ่ายข้อมูลในระบบ เนื่องมาจากโครงสร้างพื้นฐานของโครงข่ายชื่อข้อมูลไม่เหมาะสมที่จะนำมาใช้กับโครงข่ายแอตฮอกสำหรับยานพาหนะได้โดยตรง จึงได้มีการปรับเปลี่ยนโครงสร้างพื้นฐานของโครงข่ายชื่อข้อมูลแบบดั้งเดิม ดังนี้

- มีการใช้ช่องสัญญาณของโครงข่ายไร้สายในการส่งแพ็กเก็ตแบบแพร่กระจาย (Broadcast) และรับแพ็กเก็ตข้อมูลด้วยส่วนต่อประสานเดียวกัน ดังนั้น FIB (Forwarding information base) หรือ ตารางเส้นทางของส่วนต่อประสานขาออกจากโหนด จะไม่ถูกพิจารณาสำหรับโครงข่ายแอตฮอกสำหรับยานพาหนะ เพราะมีส่วนต่อประสาน 1 ชนิด
- มีการใช้งาน CS (Content store) หรือ ที่เก็บข้อมูล และฟังก์ชัน PIT (Pending interest table) หรือ ตารางบันทึกการร้องขอข้อมูลที่รอดำเนินการ ตามโครงข่ายชื่อข้อมูลแบบดั้งเดิม
- มีการป้องกันการเกิดการชนกันของข้อมูลในลักษณะเป็นกลุ่มก้อนขนาดใหญ่ระหว่างการแพร่กระจายข้อมูล (Broadcast storm) ในระบบโดยใช้กลไกการจับเวลา เมื่อยานพาหนะได้รับข้อความจะต้องรอเวลาเพื่อที่จะส่งต่อข้อมูล แทนที่จะส่งต่อข้อมูลในทันที เพื่อให้การส่งต่อข้อมูลเป็นไปอย่างมีประสิทธิภาพ

สำหรับโพรโทคอล การแจกจ่ายข้อมูลในโครงข่ายแอตฮอกของยานพาหนะที่นำเสนอนี้ มีการนำโหนดมาติดตั้งข้างถนน เรียกว่า โหนดข้างถนน (roadside unit: RSU) ที่ไม่เชื่อมต่อกับศูนย์กลางข้อมูลหรือศูนย์ประมวลผล โหนดข้างถนนสามารถติดตั้งตรงป้อมตำรวจ ป้ายรถเมล์ เสาไฟฟ้า หรือตู้โทรศัพท์ โดยติดต่อสื่อสารข้อมูลการจราจร ได้แก่ การรายงานสถานะการจราจรบนท้องถนน การประมาณเวลาที่ใช้ในการเดินทางจากจุดเริ่มต้นไปยังปลายทาง (traveling time) สภาพอากาศในบริเวณนั้น เหตุการณ์อุบัติเหตุต่าง ๆ ที่เกิดขึ้น การให้ข้อมูลศูนย์บริการน้ำมัน ร้านอาหาร หรือห้างสรรพสินค้าบริเวณนั้น เป็นต้น

ข้อดีของการนำโหนดข้างถนนมาติดตั้งนั้น สรุปได้ 2 กรณี

กรณีที่ 1 ในสภาวะบนท้องถนนที่มียานพาหนะเบาบาง ทำให้อานพาหนะไม่สามารถติดต่อสื่อสารกันได้ การติดต่อสื่อสารข้อมูลการจราจรในพื้นที่นี้ อาจเกิดการสูญหายของข้อมูล แต่เมื่อนำโหนดข้างถนนมาติดตั้งเพื่อช่วยเก็บรักษาข้อมูลจากยานพาหนะไว้ระยะเวลาหนึ่ง ต่อมาเมื่อยานพาหนะเข้ามาในพื้นที่ดังกล่าว ข้อมูลการจราจรนี้จะถูกแจกจ่ายจากโหนดข้างถนนไปยังยานพาหนะได้ เพื่อเป็นการป้องกันการสูญหายของข้อมูลจากกรณีที่ไม่มียานพาหนะในพื้นที่นั้น

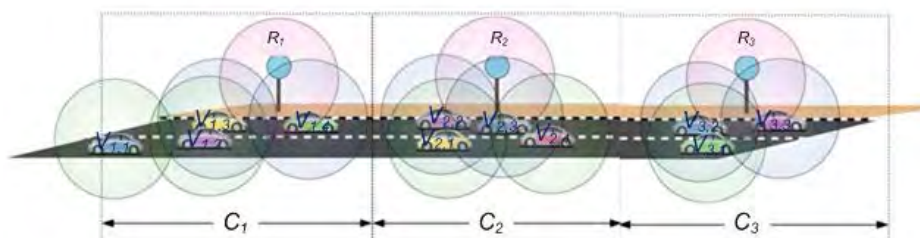
กรณีที่ 2 ในสภาวะบนท้องถนนที่มียานพาหนะหนาแน่น ทำให้เกิดการแย่งใช้ช่องสัญญาณ เมื่อมีการติดตั้งโหนดข้างถนนเข้ามาในระบบแล้ว โหนดข้างถนนนี้ทำหน้าที่เป็นตัวกลางในการรวบรวมข้อมูลการจราจร เพื่อลดการชนกันของข้อมูลให้น้อยลงจากการตอบกลับข้อมูลจากยานพาหนะมา

ที่โนดข้างถนน อีกทั้งเพื่อพิจารณาเมื่อมีการร้องขอข้อมูลการจราจรที่เป็นสาเหตุของปัญหาการชนกันของข้อมูลในลักษณะเป็นกลุ่มก้อนขนาดใหญ่ระหว่างการแพร่กระจายข้อมูล เมื่อยานพาหนะมีการร้องขอข้อมูลที่โนดข้างถนน จะเป็นการช่วยกำหนดทิศทางการร้องขอ แทนที่แพ็กเก็ตการร้องขอจะกระจายไปทุกทิศทาง ดังนั้นโนดข้างถนนช่วยในการลดการส่งแพ็กเก็ตการร้องขอได้อีกด้วย

ในการออกแบบโพรโทคอลที่นำเสนอนี้ มีการใช้งานตามแบบโครงสร้างโครงข่ายชื่อข้อมูลที่ได้กล่าวข้างต้น แต่มีคุณลักษณะเพิ่มเติม ดังนี้

- รูปแบบที่แก้ไขแล้วของแพ็กเก็ตการร้องขอและแพ็กเก็ตข้อมูลซึ่งมีข้อมูลของตำแหน่งทางภูมิศาสตร์
- วิธีคำนวณตัวจับเวลาของการส่งต่อแพ็กเก็ตการร้องขอและแพ็กเก็ตข้อมูล เมื่อคำนึงถึงโนดที่วิ่งเข้าสู่โนดข้างถนน
- การป้องกันการเกิดการชนกันของข้อมูลในลักษณะเป็นกลุ่มก้อนขนาดใหญ่ระหว่างการแพร่กระจายข้อมูล ด้วยการกำหนดทิศทางของแพ็กเก็ตการร้องขอข้อมูล
- การส่งแพ็กเก็ตการร้องขอซ้ำพิจารณาแบบฮอปต่อฮอป

3.1 โครงสร้างของระบบ



รูปที่ 3.1: รูปแบบของโครงสร้างของโพรโทคอลที่นำเสนอ

ในส่วนนี้ อธิบายถึงความหมายและสัญลักษณ์ที่ใช้ ตามรูปที่ 3.1 ดังนี้

- ถนนจะถูกแบ่งออกเป็นเซลล์เสมือนเล็กๆ เขียนแทนด้วยสัญลักษณ์ C_1, C_2, \dots, C_n โดยแบ่งตามโนดข้างถนนที่มีประจำอยู่ในแต่ละช่วงถนน
- ยานพาหนะแต่ละคัน เขียนแทนด้วยสัญลักษณ์ $V_{n,i}$ โดยเป็นยานพาหนะคันที่ i ในบริเวณเซลล์เสมือน C_n
- ยานพาหนะแต่ละคันมีการติดตั้งอุปกรณ์การสื่อสารไร้สาย และใช้เสาอากาศแบบรอบทิศทาง (Omni-directional) ทำให้เวลาส่งข้อมูลแบบการแพร่กระจายข้อมูล (Broadcast) จะมีรัศมีเป็นวงกลมและมีระยะการส่งสัญญาณที่จำกัด

- ยานพาหนะแต่ละคันมีการติดตั้งระบบอุปกรณ์ที่มีระบบนำทาง (GPS) และมีแผนที่ดิจิทัล ทำให้สามารถระบุตำแหน่ง ทิศทาง และความเร็วของยานพาหนะได้ อีกทั้งยานพาหนะจะรู้ตำแหน่งของโน้ตข้างถนนทุกตัว
- กำหนดโน้ตข้างถนน หรือ Roadside unit (RSU) เขียนแทนด้วยสัญลักษณ์ R_1, R_2, \dots, R_n โดยที่ n -th คือ ลำดับที่กำกับโน้ตข้างถนนแต่ละตัว โน้ตข้างถนนแต่ละตัวจะทำหน้าที่เก็บข้อมูลจากยานพาหนะในเซลล์เสมือนนั้น ๆ โดยที่โน้ตข้างถนนนี้จะติดตั้งอุปกรณ์ไร้สายชนิดเดียวกับยานพาหนะ ทำให้โน้ตข้างถนนมีคุณลักษณะเหมือนกับยานพาหนะทุกประการ

ตารางที่ 3.1: โครงสร้างส่วนหัวของแพ็กเก็ต

Types packet	<i>Polling</i>	<i>Data</i>	<i>Request info.</i>	<i>Reply info.</i>
Name	Polling	Polling	Traffic	Traffic
Selector	Node Selector	RSU ID	RSU ID	Node Selector
Guilters	Conditions	Conditions	Conditions	Conditions
Position	Position	Position	Position	Position
Time stamp	Time stamp	Time stamp	Time stamp	Time stamp
Signature	NONCE	Signature	NONCE	Signature

จากตารางที่ (3.1) แสดงชนิดของแพ็กเก็ตทั้ง 4 ชนิด ดังนี้

1. *Polling message* คือ แพ็กเก็ตการร้องขอข้อมูลจากโน้ตข้างถนน R_n โน้ตข้างถนนแต่ละตัวจะส่งข้อมูลการร้องขอเป็นช่วงเวลา เพื่อรวบรวมข้อมูลจากยานพาหนะในเซลล์เสมือนนั้น ๆ ตัวอย่าง prefix name ที่อยู่ในแพ็กเก็ตการร้องขอ **/polling/node selector/conditions/position/timestamp/NONCE** โดยที่
 - “Polling” แสดงชื่อแพ็กเก็ตการร้องขอจากโน้ตข้างถนน
 - “Node Selector” ทำหน้าที่เป็นตัวเลือกประเภทของโน้ตที่ต้องการรวบรวม
 - “Conditions” แสดงข้อมูลเฉพาะเจาะจงที่ผู้ร้องขอต้องการในบริเวณที่เกี่ยวข้อง เช่น อดีของโน้ต ความเร็ว เลน และตำแหน่ง
 - “Position” แสดงตำแหน่งของโน้ตผู้ร้องขอ
 - “Timestamp” แสดงเวลาเหตุการณ์เกิดขึ้น
 - “NONCE” แสดงกลุ่มตัวเลขแบบสุ่มเพื่อระบุแพ็กเก็ตการร้องขอของแต่ละแพ็กเก็ต
2. *Data $d_{n,i}$ message* คือ แพ็กเก็ตข้อมูลของ $V_{n,i}$ หลังจาก $V_{n,i}$ ได้รับข้อความ *polling message* จาก R_n
3. *Request info. message* คือ แพ็กเก็ตการร้องขอข้อมูลการจราจร เมื่อยานพาหนะมีความต้องการข้อมูล จะส่งแพ็กเก็ตการร้องขอข้อมูล ตัวอย่าง prefix name ที่อยู่ในแพ็กเก็ตการร้องขอข้อมูลการจราจร **/traffic/RSU ID/conditions/position/timestamp/NONCE** โดยที่

- “Traffic” แสดงชื่อแพ็กเก็ตเกิดการร้องขอข้อมูลจากโนดยานพาหนะ
- “RSU ID” แสดงการระบุตำแหน่งของแหล่งข้อมูลที่ต้องการ
- “Conditions” แสดงข้อมูลเฉพาะเจาะจงที่ผู้ร้องขอต้องการในบริเวณที่เกี่ยวข้อง เช่น ไอทีของโนด ความเร็ว เลน และตำแหน่ง
- “Position” แสดงตำแหน่งของโนดผู้ร้องขอ
- “Timestamp” แสดงเวลาเหตุการณ์เกิดขึ้น
- “NONCE” แสดงกลุ่มตัวเลขแบบสุ่มเพื่อระบุแพ็กเก็ตเกิดการร้องขอของแต่ละแพ็กเก็ต

4. *Reply info. message* คือ แพ็กเก็ตข้อมูลการจราจร หลังจากที่มีการส่งแพ็กเก็ตเกิดการร้องขอข้อมูลการจราจร

3.2 กระบวนการทำงานของโนดข้างถนนและยานพาหนะ

ในโครงข่ายแอตฮอกของยานพาหนะนั้น มีการใช้ช่องสัญญาณเดียวกันในการรับและส่งข้อมูล เมื่อโนดข้างถนนต้องการรวบรวมข้อมูลจากยานพาหนะ อาจเกิดปัญหาการชนกันของข้อมูล จึงมีการกำหนดให้โนดข้างถนนทำงานได้ 2 โหมด ซึ่งมีการทำงานดังแสดงในรูปที่ 3.2 คือ

- โหมดบรอดแคสต์ (Broadcast mode) จะทำงานเมื่อโนดข้างถนนส่งแพ็กเก็ตเกิดการร้องขอ (polling packet) ในทุก ๆ ช่วงเวลาไปยังยานพาหนะที่อยู่ในเซลล์เสมือนภายในรัศมีการส่งของโนดข้างถนน เพื่อรับข้อมูลตอบกลับจากยานพาหนะ
- โหมดแอคทีเวท (Activate mode) จะทำงานเมื่อยานพาหนะมีความต้องการข้อมูลการจราจร ยานพาหนะจะส่งแพ็กเก็ตเกิดการร้องขอข้อมูลการจราจร (request info.) ออกไปยังยานพาหนะคันอื่น ๆ ในระบบ หากพบว่ายานพาหนะในระบบไม่มีข้อมูลการจราจรดังกล่าว โหนดข้างถนนจะตอบกลับด้วยแพ็กเก็ตข้อมูลการจราจร(reply info.) ไปยังยานพาหนะนั้นแทน

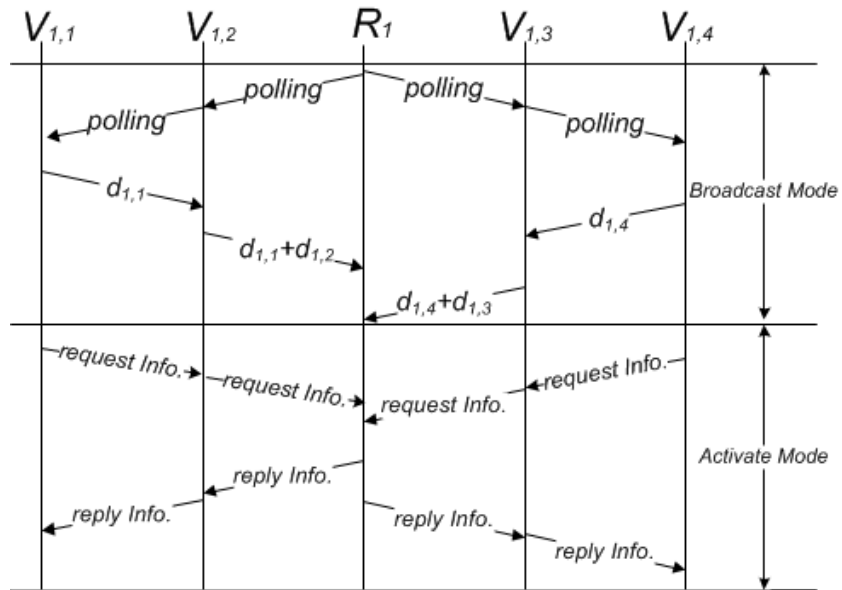
ยานพาหนะจะทำงานสลับกันระหว่าง 2 โหมด คือ โหมดทำงาน (Busy mode) กับ โหมดว่าง (Idle mode) กล่าวคือ เริ่มต้นยานพาหนะจะอยู่ในโหมดว่าง และจะปรับเข้าสู่โหมดทำงานทันที เมื่อยานพาหนะได้รับแพ็กเก็ตเกิดการร้องขอ (polling packet) จากโนดข้างถนน เพื่อรายงานข้อมูลกลับไปยังโนดข้างถนน เมื่อรายงานเสร็จ ยานพาหนะจะกลับเข้าสู่โหมดว่าง และหากมียานพาหนะคันอื่น ๆ ในระบบต้องการข้อมูลการจราจร ยานพาหนะนั้นจะปรับเข้าสู่โหมดทำงานอีกครั้ง โดยส่วนใหญ่ยานพาหนะจะอยู่ในโหมดว่าง และจะเปลี่ยนเข้าสู่โหมดทำงานเมื่อมีการร้องขอข้อมูลเกิดขึ้นในระบบ

3.3 กระบวนการทำงานของโพรโทคอล

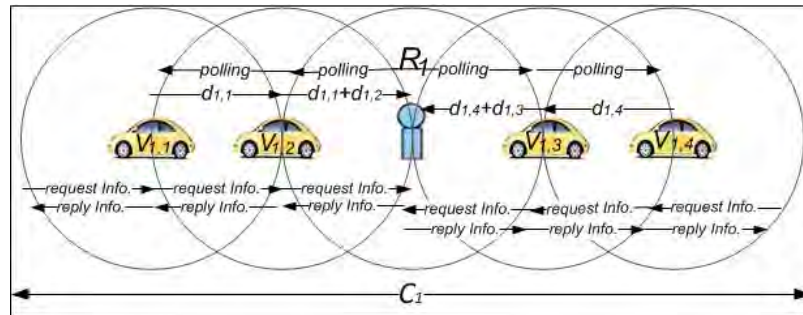
3.3.1 ขั้นตอนการทำงานของโพรโทคอลในโหมดบรอดแคสต์ (broadcast mode)

โครงสร้างของโครงข่ายชื่อข้อมูลแบ่งหน้าที่ของโนดออกเป็น 3 ชนิด โดยโพรโทคอลที่นำเสนอได้ออกแบบตามโครงสร้างดังกล่าว มีการกำหนดหน้าที่ของโนด ดังต่อไปนี้

- โหนดผู้ร้องขอ (requester node) หมายถึง โหนดข้างถนนส่งแพ็กเก็ต *Polling* เพื่อร้องขอข้อมูลจากยานพาหนะเพื่อรวบรวมข้อมูล



ก)



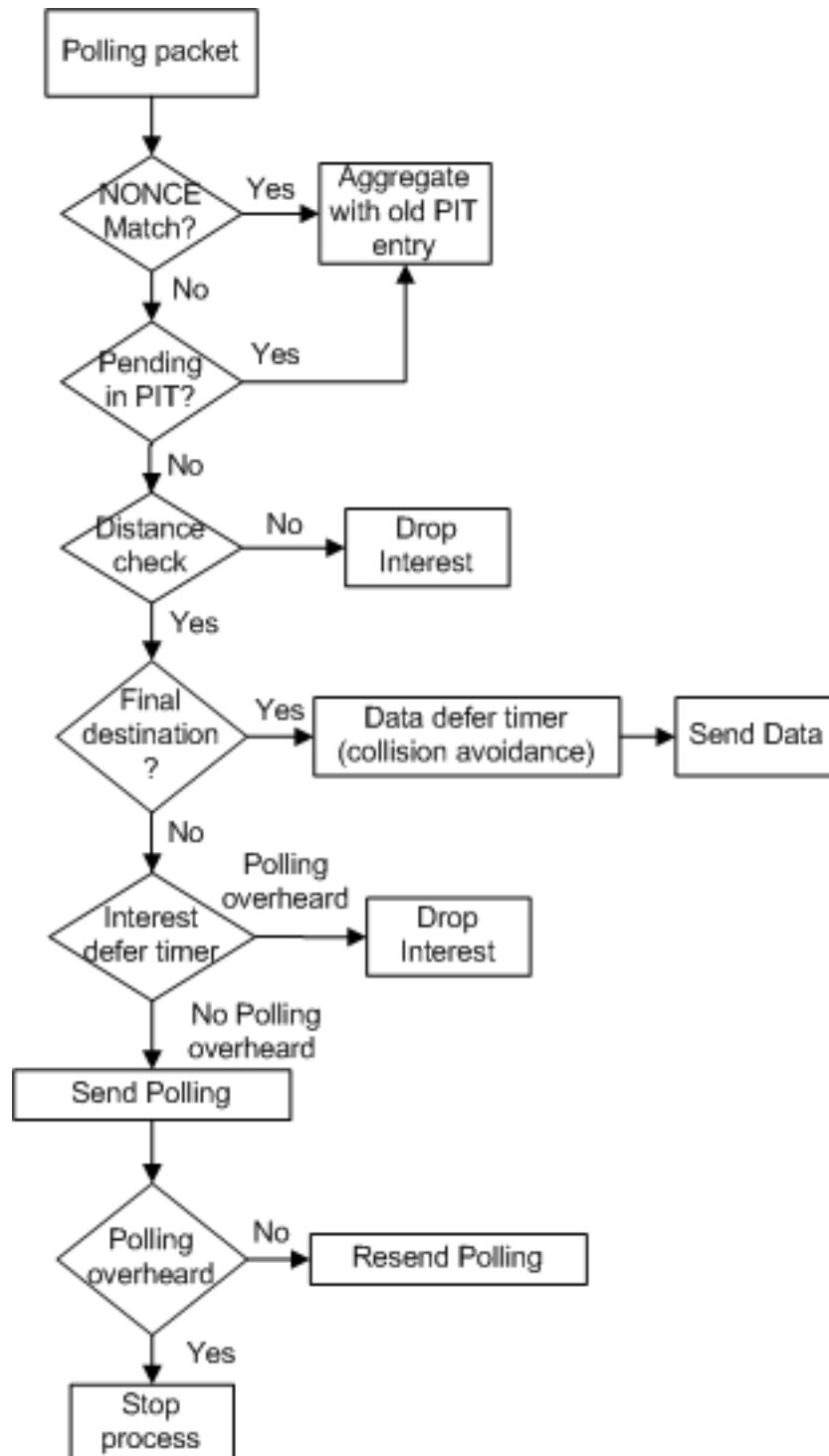
ข)

รูปที่ 3.2: ก) โพรโทคอลที่ใช้ในการสื่อสารระหว่างโหนดข้างถนนและยานพาหนะ และ ข) แสดงรัศมีมีการสื่อสารข้อมูลของโพรโทคอลระหว่างโหนดข้างถนนและยานพาหนะ

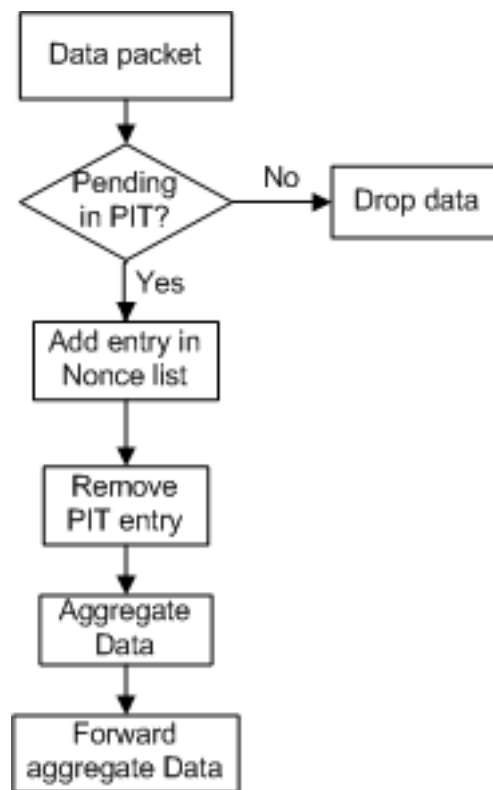
- โหนดระหว่างทาง (intermediate node) หมายถึง ยานพาหนะที่ถูกเลือกให้แพร่กระจายแพ็กเก็ต *Polling*
- โหนดแหล่งข้อมูล (provider node) หมายถึง ยานพาหนะที่ได้รับแพ็กเก็ต *Polling* และจะตอบกลับด้วย *data* $d_{n,i}$

ในขั้นตอนการทำงานของโพรโทคอลในโมดบรอดแคสต์ โหนดข้างถนนจะเริ่มรวบรวมข้อมูลจากยานพาหนะ โดยการแพร่กระจายแพ็กเก็ตการร้องขอ *polling message* ออกไป โดยใช้ค่านำหน้าของแพ็กเก็ตชื่อ **/polling** ซึ่งประกอบไปด้วยค่า NONCE จำนวน 4 ไบต์ และอายุของแพ็กเก็ต (lifetime) มีค่าเท่ากับ 4 วินาที ในบริเวณเซลล์เสมือนที่ตนเองดูแลอยู่ โพรโทคอลที่ออกแบบมีการปรับลำดับของการทำงานตามโครงสร้างชื่อข้อมูลแบบดั้งเดิม กล่าวคือ โพรโทคอลแบบดั้งเดิมมีการตรวจสอบภายใน Content store (CS) ก่อน เพื่อเป็นการลดความล่าช้าในการค้นหาของ CS โดยโพรโทคอลจะเปลี่ยนไปตรวจสอบภายใน PIT ก่อน เนื่องจากขนาดของ PIT มีขนาดเล็กกว่า CS ทำให้โพรโทคอลสามารถเพิ่มประสิทธิภาพของโครงข่ายทั้งหมดได้ ทั้งนี้โพรโทคอลจะกระทำการใด ๆ กับแพ็กเก็ตเพียงครั้งเดียว และโหนดจะละเลยแพ็กเก็ตที่ได้เคยพิจารณาไปแล้ว ดังนั้นกระบวนการทำงานตามผังงานลำดับเมื่อโหนดได้รับแพ็กเก็ต *polling message* ในรูปที่ 3.3 มีการทำงานต่อไปนี้

- ยานพาหนะจะตรวจสอบค่าของ NONCE เป็นลำดับแรกว่าเคยได้รับแพ็กเก็ตการร้องขอก่อนหน้านี้แล้วหรือไม่ หากได้รับแล้ว แพ็กเก็ตจะถูกยกเลิก แต่ถ้าไม่ได้รับ แพ็กเก็ตจะถูกเก็บไว้ใน PIT
- หลังจากนั้นเมื่อตรวจสอบในรายการรอดำเนินการใน PIT แล้ว หากพบว่าการร้องขออยู่ในรายการรอดำเนินการการร้องขอดังกล่าว แพ็กเก็ตจะถูกยกเลิก แต่หากไม่ตรงกับรายการรอดำเนินการใน PIT จะตรวจสอบระยะทางต่อไป
- เมื่อแพ็กเก็ตการร้องขอตรวจสอบระยะทางว่าอยู่ในพื้นที่ของโหนดข้างถนนที่ร้องขอหรือไม่ แต่หากไม่ แพ็กเก็ตจะถูกยกเลิก หรือถ้าอยู่ในพื้นที่ดังกล่าว จะตรวจสอบในขั้นต่อไป
- แพ็กเก็ตการร้องขอตรวจสอบว่าอยู่บริเวณขอบเขตพื้นที่การร้องขอหรือไม่ แต่หากไม่ แพ็กเก็ตจะถูกยกเลิก หรือถ้าอยู่ในพื้นที่ดังกล่าว จะตรวจสอบในขั้นต่อไป
- เมื่อแพ็กเก็ตการร้องขอตรวจสอบระยะทางว่าโหนดอยู่ในพื้นที่ของโหนดข้างถนนที่ร้องขอหรือไม่ แต่หากไม่ แพ็กเก็ตการร้องขอจะถูกส่งต่อ หรือถ้าโหนดอยู่ในพื้นที่ดังกล่าว โหนดดังกล่าวจะส่งแพ็กเก็ตข้อมูลกลับไป เนื่องจากการป้องกันการชนกันแพ็กเก็ต โหนดจะมีช่วงเวลารอดอยเป็นระยะเวลาหนึ่ง
- หลังจากตรวจสอบแพ็กเก็ตการร้องขอ โหนดจะมีการรอดอยเป็นช่วงเวลาหนึ่ง เพื่อให้โพรโทคอลมีการคำนวณว่า ใครเป็นผู้เหมาะสมในการแพร่กระจายแพ็กเก็ตการร้องขอ ถ้าได้ยื่นการกระจายข้อมูลการร้องขอ แพ็กเก็ตจะถูกยกเลิก แต่หากไม่ ยานพาหนะจะแพร่กระจายแพ็กเก็ตการร้องขอออกไป
- หลังจากที่โหนดได้รับการแพร่กระจายแพ็กเก็ตการร้องขอ โหนดจะมีการรอดอยเป็นช่วงเวลาหนึ่ง เพื่อตรวจสอบว่าในบริเวณนั้น มียานพาหนะหรือไม่ ถ้าได้ยื่นการแพร่กระจายข้อมูลการร้องขอ แพ็กเก็ตจะถูกยกเลิก แต่หากไม่ แพ็กเก็ตจะทำการส่งแพ็กเก็ตการร้องขอนี้ใหม่อีกครั้ง



รูปที่ 3.3: ผังงานแสดงกระบวนการของแพ็กเก็ต *Polling*.



รูปที่ 3.4: ผังงานแสดงกระบวนการของแพ็กเก็ต *Data*.

กระบวนการทำงานตามผังงานลำดับเมื่อโหนดได้รับแพ็กเก็ต *data message* ในรูปที่ 3.4 มีการทำงานต่อไปนี้

- ยานพาหนะจะตรวจสอบในรายการรอดำเนินการใน PIT หากการร้องขอไม่อยู่ในรายการรอดำเนินการการร้องขอ แพ็กเก็ตจะถูกยกเลิก ถ้าตรงกับรายการรอดำเนินการ จะเพิ่มชื่อแพ็กเก็ตลงในตารางของ Nonce เพื่อดำเนินการตรวจสอบแพ็กเก็ตในอนาคตว่าเคยได้รับแล้วหรือไม่
- หลังจากนั้นแพ็กเก็ตนี้จะถูกลบไปจากตารางรายการรอดำเนินการใน PIT
- แพ็กเก็ตข้อมูลจะรวบรวมแพ็กเก็ตข้อมูลที่ได้รับกับข้อมูลของตัวเองแบบ piggy back ($Data V_1 + V_2 + \dots + V_n$) จากงานวิจัยโดย Ai H. Ho และคณะ [49] ดังแสดงในรูปที่ 3.5
- แพ็กเก็ตข้อมูลจะถูกส่งกลับไปยังโหนดข้างเคียง

3.3.2 ขั้นตอนการทำงานของโพรโทคอลในโหมดแอคทีฟ (activate mode)

โพรโทคอลจะเริ่มทำงานเมื่อยานพาหนะต้องการทราบข้อมูลการจราจร โดยการส่งแพ็กเก็ตการร้องขอข้อมูลการจราจร *Request Info. packet* ใช้คำนำหน้าของแพ็กเก็ตคือ **/traffic** ประกอบไปด้วยค่า NONCE จำนวน 4 ไบต์ และค่าอายุของแพ็กเก็ต (lifetime) ซึ่งมีค่าเท่ากับ 4 วินาที ซึ่งแบ่งชนิดของโหนดตามโครงข่ายชื่อข้อมูลออกเป็น 3 รูปแบบ ได้แก่

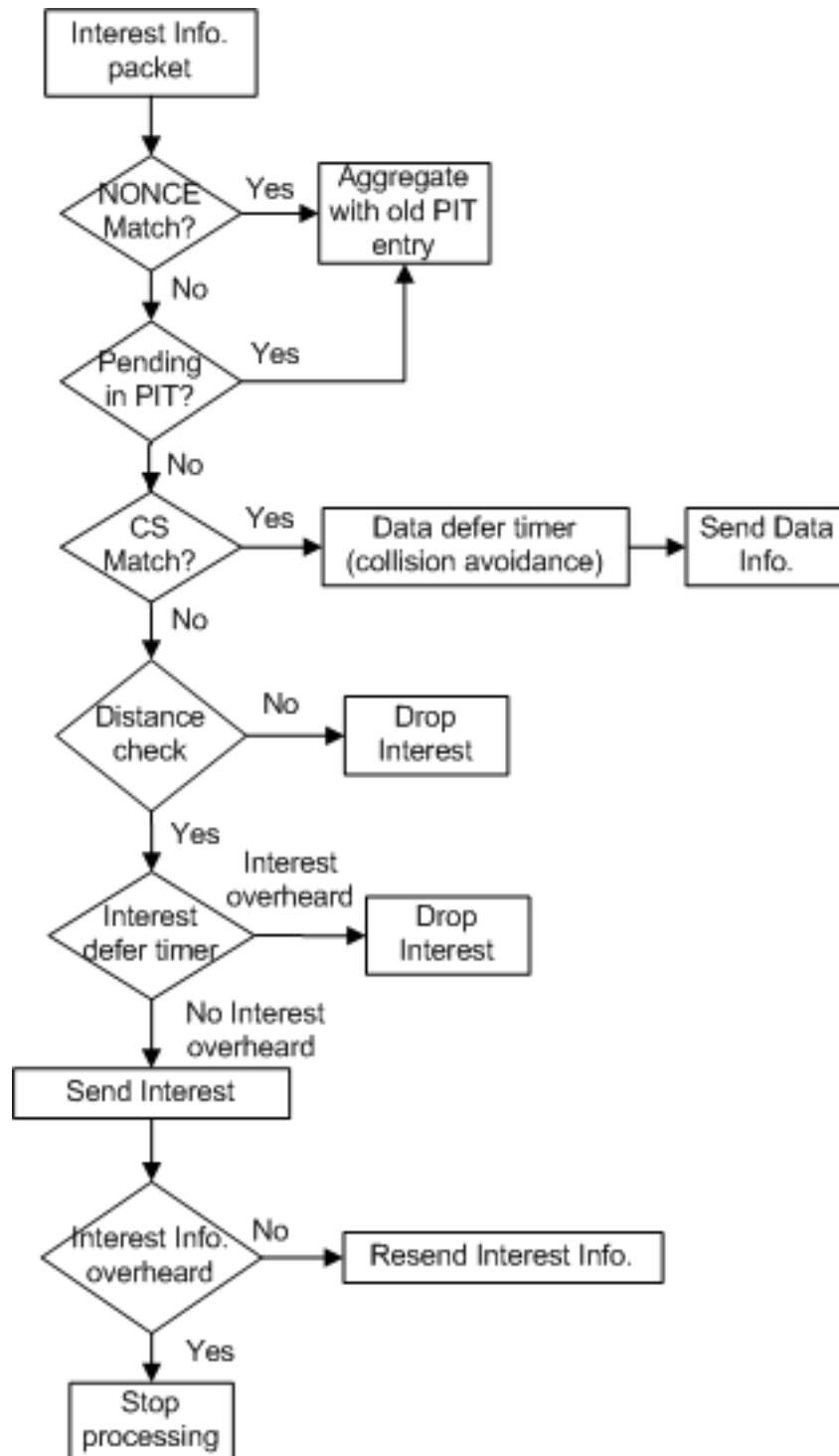
RSU ID
Hash or message ID
Number of messages
Message 1 length
Message 1
Message 2 length
Message 2

รูปที่ 3.5: โครงสร้างของ $data\ d_{n,i}\ message$.

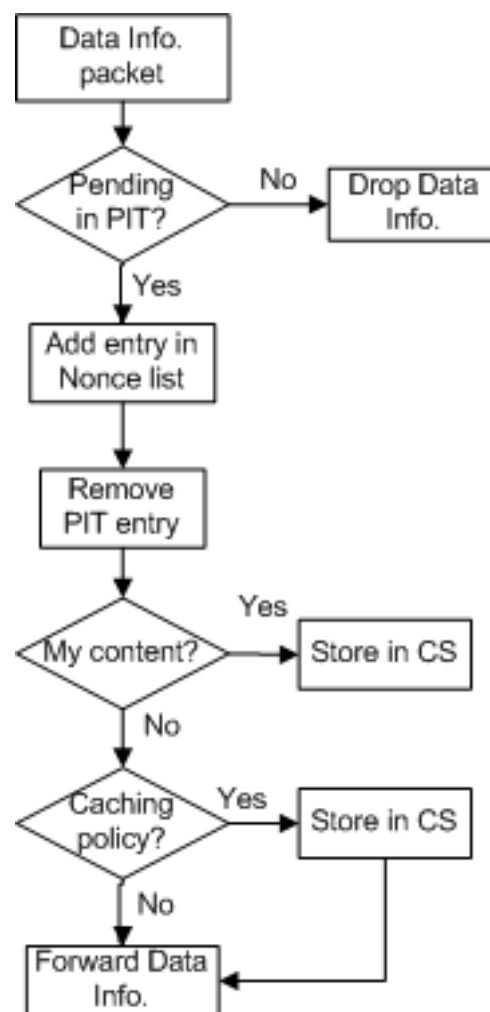
- โหนดผู้ร้องขอ (requester node) หมายถึง ยานพาหนะที่ต้องการทราบข้อมูลการจราจร จะส่งแพ็กเก็ต *Interest Info*.
- โหนดระหว่างทาง (intermediate node) หมายถึง ยานพาหนะที่ถูกเลือกในการส่งต่อข้อมูล เป็นโหนดระหว่างทางที่มีการจัดเก็บเนื้อหาข้อมูล เมื่อได้รับ แพ็กเก็ต *Data Info*.
- โหนดแหล่งข้อมูล (provider node) หมายถึง โหนดข้างถนนที่ให้บริการข้อมูล หรือ ยานพาหนะที่มีข้อมูลจากการจัดเก็บข้อมูล จะตอบกลับข้อมูลด้วยแพ็กเก็ต *Data Info*.

กระบวนการทำงานตามผังงานลำดับเมื่อโหนดได้รับแพ็กเก็ต *Request Info. packet* ในรูปที่ 3.3 มีขั้นตอนการทำงานดังนี้

- ยานพาหนะจะตรวจสอบค่าของ NONCE เป็นลำดับแรกว่าเคยได้รับแพ็กเก็ตการร้องขอก่อนหน้านี้แล้วหรือไม่ หากได้รับแล้ว แพ็กเก็ตจะถูกยกเลิก แต่ถ้าไม่ได้รับ แพ็กเก็ตจะถูกเก็บไว้ใน PIT
- หลังจากนั้นแพ็กเก็ตจะตรวจสอบรายการรอดำเนินการใน PIT หากการร้องขออยู่ในรายการรอดำเนินการการร้องขอ แพ็กเก็ตจะถูกยกเลิก แต่ถ้าไม่ตรงกับรายการรอดำเนินการ แพ็กเก็ตจะตรวจสอบข้อมูลใน CS ต่อไป
- เมื่อแพ็กเก็ตการร้องขอตรวจสอบว่าข้อมูลตรงกับชื่อของแพ็กเก็ตการร้องขอข้อมูลการจราจร อยู่ใน CS โหนดจะส่งแพ็กเก็ตข้อมูลการจราจรกลับไป เพื่อป้องกันการชนกันแพ็กเก็ต โหนดจะมีช่วงเวลารอคอยเป็นระยะเวลาหนึ่ง แต่หากไม่ตรงกับ CS จะตรวจสอบระยะทางต่อไป
- เมื่อแพ็กเก็ตการร้องขอตรวจสอบระยะทางว่าระยะห่างระหว่างตนเองกับโหนดข้างถนนเทียบกับระยะห่างโหนดผู้ร้องขอกับโหนดข้างถนน มีค่าน้อยกว่าหรือไม่ แต่หากไม่ แพ็กเก็ตการร้องขอจะถูกยกเลิก หรือแต่หากอยู่ในระยะดังกล่าว โหนดจะแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลการจราจรออกไป
- ก่อนที่โหนดจะแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลการจราจร โหนดจะรอคอยเป็นช่วงเวลาหนึ่ง ถ้าได้รับแพ็กเก็ตการกระจายข้อมูลการร้องขอเดิม แพ็กเก็ตจะถูกยกเลิก แต่ถ้าไม่ ยานพาหนะจะแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลการจราจรออกไป



รูปที่ 3.6: ฟังงานแสดงกระบวนการของแพ็กเก็ต *Request Info.* .



รูปที่ 3.7: ฟังงานแสดงกระบวนการของแพ็กเก็ต *Reply Info.*

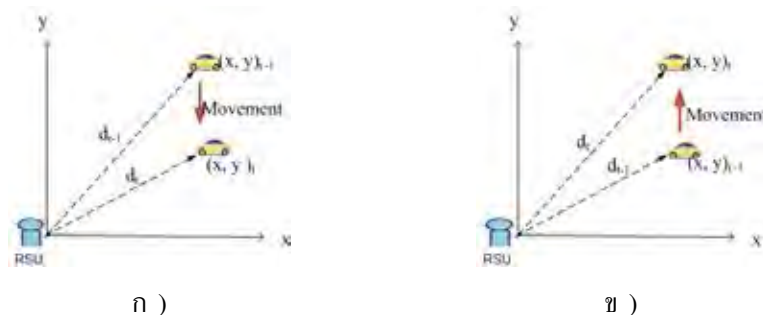
- หลังจากที่ได้แพร่กระจายแพ็กเก็ตการร้องขอข้อมูลการจราจร โหนดจะมีการรอเป็นช่วงเวลาหนึ่งเพื่อตรวจสอบว่ามียานพาหนะในบริเวณนั้นหรือไม่ แต่หากได้ยืนยันการกระจายข้อมูลการร้องขอแพ็กเก็ตจะถูกยกเลิก แต่ถ้าหากไม่ แพ็กเก็ตการร้องขอนี้จะถูกส่งใหม่อีกครั้ง

กระบวนการทำงานตามผังงานลำดับเมื่อโหนดได้รับแพ็กเก็ต *Reply Info.* ในรูปที่ 3.7 มีกระบวนการทำงาน ต่อไปนี้

- ยานพาหนะจะตรวจสอบแพ็กเก็ตการร้องขอในรายการรอดำเนินการ PIT หากแพ็กเก็ตการร้องขอไม่อยู่ในรายการรอดำเนินการการร้องขอ แพ็กเก็ตจะถูกยกเลิก แต่ถ้าหากตรงกับรายการรอดำเนินการจะเพิ่มแพ็กเก็ตการร้องขอลงในตารางของ Nonce เพื่อดำเนินการตรวจสอบแพ็กเก็ตในอนาคตว่าเคยได้รับแล้วหรือไม่
- หลังจากนั้นแพ็กเก็ตนี้จะถูกลบไปจากตารางรายการรอดำเนินการใน PIT
- โหนดมีการตรวจสอบว่าแพ็กเก็ตข้อมูลนี้ จากโหนดตนเองหรือไม่ หากใช่ โหนดจะเก็บแพ็กเก็ตลงไปใน CS แต่ถ้าหากไม่ใช่ โหนดจะตรวจสอบว่าจะมีการจัดเก็บข้อมูลหรือไม่
- แพ็กเก็ตข้อมูลจะถูกตรวจสอบตาม Caching policy ว่าต้องการเก็บข้อมูลหรือไม่ หากใช่ จะเก็บแพ็กเก็ตข้อมูลการจราจรลงไปใน CS แต่ถ้าหากไม่ จะส่งต่อแพ็กเก็ตข้อมูลการจราจรออกไป

3.3.3 ขั้นตอนการเลือกโหนดส่งต่อข้อมูล

สำหรับกระบวนการการเลือกโหนดส่งต่อข้อมูลจะใช้ตำแหน่งของโหนดผู้ร้องขอ ตำแหน่งของโหนดผู้รับซึ่งอยู่ในรัศมีการสื่อสารของผู้ร้องขอ และตำแหน่งของโหนดข้างถนน โดยที่โหนดจะมีการติดตั้ง GPS เพื่อระบุตำแหน่งของโหนด และมีการบันทึกตำแหน่งของโหนด ณ เวลานั้นๆ เพื่อใช้ตัดสินใจเลือกโหนดส่งต่อที่เหมาะสมที่สุด กระบวนการตัดสินใจนี้จะเกิดขึ้นที่ **โหนดผู้รับ** เพื่อป้องกันการเกิดการชนกันของข้อมูลในลักษณะเป็นกลุ่มก้อนขนาดใหญ่ระหว่างการแพร่กระจายข้อมูล โดยกระบวนการตัดสินใจที่โหนดผู้รับแบ่งได้ 3 ขั้นตอนดังนี้



รูปที่ 3.8: ก) ระยะการกระจายลดลงแสดงถึงยานพาหนะที่กำลังมุ่งหน้าสู่เหตุการณ์ และ ข) ระยะการกระจายเพิ่มขึ้นแสดงถึงยานพาหนะที่เคลื่อนที่ออกจากเหตุการณ์

1. เมื่อโหนดผู้รับทุกโหนดได้รับแพ็กเก็ตการร้องขอข้อมูล โหนดผู้รับจะคำนวณหาระยะห่างระหว่างผู้ร้องขอกับโหนดข้างถนน และระยะห่างระหว่างตนเองกับโหนดข้างถนน ใช้การคำนวณหา

ระยะทางตั้งสมการที่ (3.1) ถ้าค่าระยะห่างระหว่างผู้ร้องขอกับโนดข้างถนนมีค่ามากกว่า ระยะห่างระหว่างตนเองกับโนดข้างถนน โหนดผู้รับจะละทิ้งแพ็กเก็ตการร้องขอ เพื่อช่วยลด จำนวนแพ็กเก็ตการร้องขอในระบบ แต่ถ้าค่าระยะห่างระหว่างผู้ร้องขอกับโนดข้างถนนมีค่าน้อยกว่าระยะห่างระหว่างตนเองกับโนดข้างถนน จะหมายถึงโนดผู้รับนี้อยู่ใกล้กับโนดข้างถนน และโนดผู้รับนี้จะเป็นหนึ่งในตัวเลือกที่จะถูกส่งต่อแพ็กเก็ต

$$d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (3.1)$$

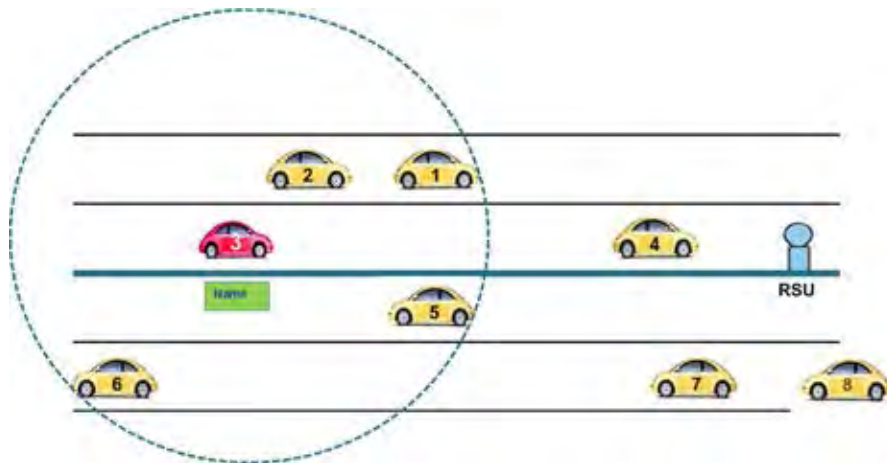
เมื่อ d_{AB} คือระยะห่างระหว่างโนด A กับโนด B $(x, y)_A$ คือพิกัด x, y ของโนด A และ $(x, y)_B$ คือพิกัด x, y ของโนด B

2. โหนดผู้รับจะตรวจสอบว่าโนดตนเองเคลื่อนที่เข้าหาโนดข้างถนนหรือไม่ ดังรูปที่ 3.3.3 โดยคำนวณจากตำแหน่งของโนดตนเองในช่วงเวลาปัจจุบัน และช่วงเวลาก่อนหน้า ใช้การคำนวณหาระยะทางตั้งสมการที่ (3.1) กำหนดให้ t_d คือเวลาขีดแบ่ง(threshold)เมื่อโนดมีระยะการจัดมากขึ้นหรือน้อยลง ซึ่งถ้าโนดผู้รับนี้เคลื่อนที่ไปในทิศทางเดียวกับโนดผู้ร้องขอ และมีการเคลื่อนที่เข้าหาโนดข้างทาง การกระจัดจะมีค่าลดลง ซึ่งโนดผู้รับนี้จะเลือกช่วงเวลา t ในช่วง $[0, t_d]$ และถ้าโนดผู้รับนี้เคลื่อนที่ในทิศทางตรงข้ามกับโนดผู้ร้องขอ และมีการเคลื่อนที่ออกจากโนดข้างถนน การกระจัดจะมีค่าเพิ่มขึ้น ซึ่งโนดผู้รับนี้จะเลือกช่วงเวลา t ในช่วง $[t_d, t_{max}]$
3. หลังจากนั้น โหนดผู้รับจะกำหนดช่วงเวลารอคอยก่อนที่จะส่งต่อแพ็กเก็ต ซึ่งสามารถคำนวณได้จากสมการที่ (3.2) โดยปรับมาจากสมการ [22]

$$T_{slots} = \left\lfloor \frac{R - d}{R} \right\rfloor \times (t_{max} - t) + t \quad (3.2)$$

เมื่อ T_{slots} คือระยะเวลาที่ใช้ในการรอคอยเพื่อส่งต่อข้อมูล R คือรัศมีของการรับส่งข้อมูล d คือระยะห่างระหว่างโนด นั้น ๆ กับโนดก่อนหน้าที่ส่งต่อแพ็กเก็ต t คือเวลาในการเลือกสล็อต และ t_{max} คือเวลาในการรอสูงสุดของชั้น MAC ซึ่งเวลาดังกล่าวจะเป็นสัดส่วนตรงกับระยะทางระหว่างโนดตนเองกับโนดข้างถนน โดยโนดผู้รับใดที่รอคอยจนครบเวลาก่อน โหนดผู้รับนั้นจะเป็นผู้มีสิทธิในการส่งต่อแพ็กเก็ต และโนดผู้รับที่ได้ส่งต่อแพ็กเก็ตจะไปยับยั้งการส่งต่อแพ็กเก็ตของโนดผู้รับอื่น ๆ

ในรูปที่ 3.9 เมื่อพิจารณายานพาหนะ V_3 เป็นโนดผู้ร้องขอ ยานพาหนะที่อยู่ในรัศมีการสื่อสารของ V_3 จะมีการทำงานตามกระบวนการรับแพ็กเก็ตการร้องขอข้อมูล เพื่อให้การส่งต่อแพ็กเก็ตมีประสิทธิภาพ โปรโทคอลกำหนดช่วงเวลารอคอยก่อนที่จะจัดส่งแพ็กเก็ตไปยังยานพาหนะอื่น ๆ ยานพาหนะใดที่รอคอยจนครบเวลาก่อน จะเป็นผู้มีสิทธิในการส่งต่อแพ็กเก็ตนี้ ซึ่งยานพาหนะ V_1 และ V_5 อยู่ในตำแหน่งเดียวกันในระยะการสื่อสาร จากการคำนวณตามกระบวนการเลือกโนดส่งต่อข้อมูลข้างต้น เมื่อยานพาหนะมีการเคลื่อนที่เข้าสู่โนดข้างถนนและอยู่ในทิศทางเดียวกับยานพาหนะ V_3 ที่ต้องการส่งต่อแพ็กเก็ต เพราะฉะนั้นยานพาหนะ V_1 จะถูกเลือกเป็นยานพาหนะที่จะแพร่กระจายแพ็กเก็ตต่อไป

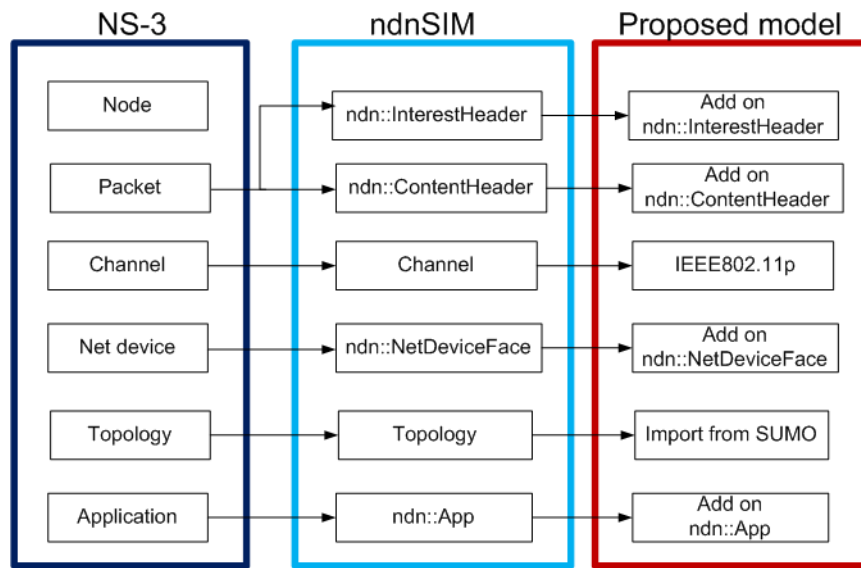


รูปที่ 3.9: การพิจารณาโหนดที่ทำการส่งต่อ

3.4 การพัฒนาโปรแกรมสำหรับใช้ในการจำลอง

ในวิทยานิพนธ์ฉบับนี้ได้มีการพัฒนาโปรแกรม ndnSIM ซึ่งเป็นโมดูลส่วนเสริมของโปรแกรม NS-3 เพื่อใช้จำลองการทำงานของโปรโตคอลที่ออกแบบ โดยโปรแกรมหลักจะแสดงในภาคผนวก ก โปรแกรมแพ็กเก็ตการร้องขอและโปรแกรมเมื่อได้รับแพ็กเก็ตการร้องขอที่ได้พัฒนาแสดงในภาคผนวก ข การพัฒนาโปรแกรมแสดงดังรูปที่ 3.10 โดยได้เพิ่มฟังก์ชันที่มีการทำงานตามโปรโตคอลที่ออกแบบไว้ในโปรแกรม NS-3 ซึ่งมีส่วนประกอบต่าง ๆ ดังนี้

- Node :ใช้โครงสร้างพื้นฐานจากโปรแกรม NS-3
- Packet :ใช้โครงสร้างของ `ndn::InterestHeader` และ `ndn::ContentHeader` โดยมีการเพิ่มเติมค่าของส่วนหัวแพ็กเก็ตคือ ชื่อของแพ็กเก็ต ตำแหน่งของโหนดข้างถนน ตำแหน่งของตนเอง และเวลาที่แพ็กเก็ตถูกสร้างขึ้นมา ตามรูปแบบที่แสดงใน ตารางที่ 3.1
- Channel :ใช้โครงสร้างพื้นฐานจากโปรแกรม NS-3 โดยเลือกใช้มาตรฐาน IEEE802.11p
- Net device :ใช้โครงสร้างของ `ndn::NetDeviceFace` โดยมีการเพิ่มเติมการทำงานแบบโครงข่ายไร้สาย เนื่องจากโครงสร้างเดิมของโปรแกรม ndnSIM จะมีการทำงานแบบโครงข่ายมีสาย ทำให้ต้องสร้างรูปแบบการทำงานเป็นแบบโครงข่ายไร้สายเพื่อให้ทำงานร่วมกับโครงสร้างพื้นฐานหลักของโปรแกรม ndnSIM ได้
- Topology :ใช้โครงสร้างพื้นฐานจากโปรแกรม NS-3 โดยเพิ่มเติมการเรียกใช้งานจาก ไฟล์ที่ถูกสร้างมาจากโปรแกรม SUMO
- Application :ใช้โครงสร้างของ `ndn::App` โดย `ndn::App` มีโครงสร้างหลักของโปรโตคอล NDN คือ `ndn::Pit` `ndn::Fib` และ `ndn::ForwardingStrategy` (แสดงในบทที่ 2 รูปที่ 2.20) โดยโปรโตคอลที่ออกแบบจะถูกดัดแปลงเฉพาะ `ndn::ForwardingStrategy` ซึ่งเป็นขั้นตอนของกระบวนการแจกจ่ายข้อมูลในโครงข่ายชื่อข้อมูล เพื่อให้ทำงานตามกระบวนการแจกจ่ายข้อมูลที่ได้ออกแบบใน 3.3.3 ได้



รูปที่ 3.10: รูปแบบการพัฒนาโปรแกรมโดยเพิ่มเติมจากโปรแกรม ndnSIM และโปรแกรม NS-3

3.5 สรุป

ในบทนี้ได้นำเสนอโปรโตคอลการแจกจ่ายข้อมูล โดยนำโนดข้างถนนมาติดตั้ง เพื่อปรับปรุงสมรรถนะในระบบแบบโครงข่ายแอดฮอคเคลื่อนที่ในยานพาหนะที่ดีขึ้น มีการอธิบายการทำงานของโปรโตคอล ซึ่งมีจุดประสงค์เพื่อให้โปรโตคอลการแจกจ่ายข้อมูลนี้ มีอัตราส่วนการรับแพ็กเก็ตมีความสำเร็จสูงขึ้น อัตราส่วนการสูญเสียแพ็กเก็ตมีค่าลดลง เวลาการแจกจ่ายข้อมูลโดยรวมน้อยลง และปริมาณทราฟฟิกในระบบมีค่าต่ำลง โปรโตคอลที่ออกแบบได้นำโนดข้างถนนมาติดตั้งในระบบเพื่อให้เป็นตัวกลางในการติดต่อสื่อสารข้อมูลการจราจร อีกทั้งมีการพิจารณาถึงในสถานะบนท้องถนนที่มียานพาหนะเบาบาง และสถานะบนท้องถนนที่มียานพาหนะหนาแน่น

บทที่ 4

สภาพแวดล้อมของการจำลอง ผลการจำลองและวิเคราะห์ผลการจำลองระบบ

เนื้อหาในบทนี้จะกล่าวถึงผลการประเมินสมรรถนะของโพรโทคอลการแจกจ่ายข้อมูลที่ได้นำเสนอในบทที่ 3 ในระบบโครงข่ายแอตสอกสำหรับยานพาหนะตามสถานการณ์ต่าง ๆ ที่เกิดขึ้นจริงบนท้องถนน สำหรับส่วนแรกของบทนี้ กล่าวถึงสภาพแวดล้อมของการจำลอง ซึ่งประกอบด้วยพารามิเตอร์ในการจำลอง และพารามิเตอร์สำหรับการประเมินสมรรถนะของโพรโทคอลที่นำเสนอ ตลอดจนทอพอโลยีที่ใช้ในการจำลอง ในส่วนที่สองจะกล่าวถึงผลการจำลองและการวิเคราะห์ผลการจำลองของวิธีที่ได้นำเสนอในงานวิทยานิพนธ์ฉบับนี้ ซึ่งมีการเปรียบเทียบกับโครงข่ายชื่อข้อมูลแบบดั้งเดิม ว่ามีสมรรถนะที่ดีกว่าหรือด้อยกว่าอย่างไร เพื่อเป็นแนวทางในการพัฒนาโครงข่ายแอตสอกสำหรับยานพาหนะต่อไปในอนาคต

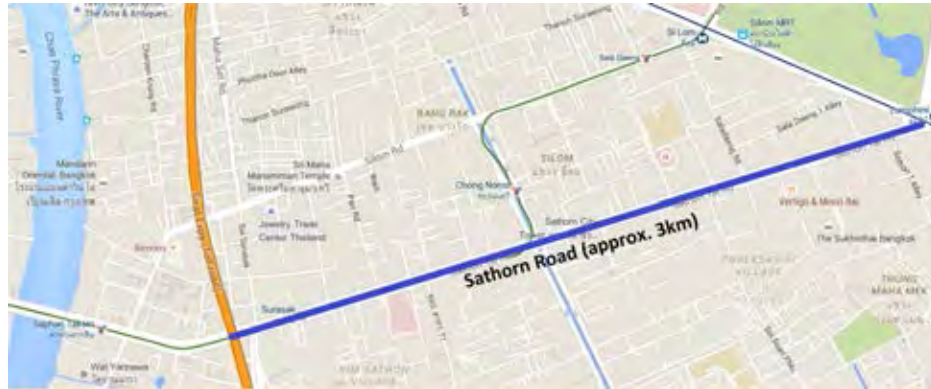
ในการจำลองเพื่อประเมินสมรรถนะการทำงานของโพรโทคอลการแจกจ่ายข้อมูลนั้น สมรรถนะของโพรโทคอลจะเปลี่ยนไปเมื่อมีการจำลองภายใต้สถานการณ์ที่แตกต่างกัน ดังนั้น เพื่อสร้างสถานการณ์ให้ใกล้เคียงกับความเป็นจริง จึงใช้โปรแกรมจำลอง 2 ส่วนตามที่ได้อธิบายรายละเอียดไว้ในบทที่ 2 หัวข้อที่ 2.4 ดังแสดงในรูปที่ 4.1 ซึ่งแสดงขั้นตอนการทำงานระหว่างโปรแกรมการจำลองระบบการจราจร และโปรแกรมการจำลองระบบการสื่อสาร โดยใช้เลือกใช้โปรแกรม SUMO และ NS-3 ตามลำดับ



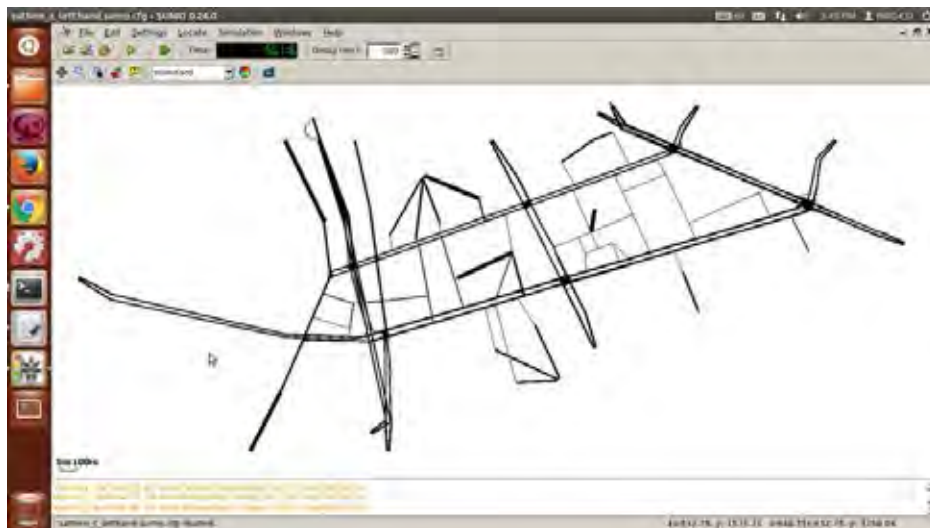
รูปที่ 4.1: การจำลองของระบบ

4.1 สภาพแวดล้อมแบบจำลองจากโครงการสาทรโมเดล

สภาธุรกิจโลกเพื่อการพัฒนาที่ยั่งยืน (WBCSD) ได้เปิดตัวโครงการ **Sustainable Mobility Project 2.0** กับ 15 ธุรกิจและคู่ค้าโดยมีเมืองที่ได้รับการคัดเลือกไว้ 6 เมืองเช่นอินดอร์ (อินเดีย) เฉิงตู (จีน) ฮัมบูร์ก (เยอรมนี) กัมเปเซ (บราซิล) ลิสบอน (โปรตุเกส) และกรุงเทพฯ (ประเทศไทย) ในประเทศไทยนั้น สนใจศึกษาถนนสาทรเป็นพื้นที่การทดลอง ตั้งชื่อโครงการว่า Sathorn Model [47] ซึ่งแสดงในรูป 4.2 เพราะเป็นพื้นที่สำคัญของย่านธุรกิจกลางเมือง ทำให้ลักษณะของพื้นที่นี้มีสภาพการจราจรที่ไม่อึดอัด เหมาะสำหรับการทดลอง ในส่วนของการจำลองระบบนั้น ผู้วิจัยจะใช้โปรแกรม SUMO ที่ถูกปรับพารามิเตอร์ให้ใกล้เคียงกับโปรแกรมจำลองการจราจร NETSTREAM ซึ่งได้รับการพัฒนาจากโตโยต้าเซ็นทรัล R&D Labs [48] ที่ได้นำข้อมูลจริงจากการสังเกตการณ์ในบริเวณถนนสาทรของวันที่ 7 กรกฎาคม 2014 มาตั้งค่าในโปรแกรมจำลองการจราจร



รูปที่ 4.2: แผนที่ของสายรถโมเดลบน Google map



รูปที่ 4.3: การจำลองของระบบ

การปรับพารามิเตอร์ระหว่าง SUMO และ NETSTREAM [47] จะใช้พารามิเตอร์ดังแสดงตารางที่ 4.1

ตารางที่ 4.1: การตั้งค่าเทียบระหว่างโปรแกรม SUMO และ NETSTREAM

พารามิเตอร์ ในโปรแกรม NETSTREAM	
ค่าความผิดพลาดหลักของการไหล (the root mean square error of flow)	9856.2 <i>vehicles/15 min</i>
เวลาในการเดินทาง (root mean square of travel time)	9450.9 <i>s/15 min</i>
การไหลเฉลี่ย (mean flow)	0.82 <i>vehicles</i>
ระยะเวลาการเดินทางเฉลี่ย (mean travel time)	0.33 <i>s</i>
พารามิเตอร์ ในโปรแกรม SUMO	
ความเร็วสูงสุด (maximum velocity)	14 <i>m/s</i>
ระยะห่างต่ำสุดระหว่างสองคัน (minimum spacing between two cars)	2.37 <i>m</i>
ความเร่ง (acceleration)	1.6 <i>m/s²</i>
ความหน่วง (latency)	3.1 <i>s</i>

4.2 ผลการจำลองเมื่อเปรียบเทียบโครงข่ายชื่อข้อมูลกับโครงข่ายแอตฮอกแบบ TCP/IP

ในปัจจุบันโครงข่ายแอตฮอกสำหรับยานพาหนะที่ใช้งานอยู่เป็นแบบ TCP/IP ซึ่งในโครงข่ายดังกล่าวใช้โพรโทคอลการค้นหาเส้นทางที่เป็นที่นิยมในการส่งข้อมูล ซึ่งสามารถแบ่งได้ 2 ชนิดหลัก ๆ คือ โพรโทคอลค้นหาเส้นทางเชิงรุก (Proactive routing protocol) และโพรโทคอลค้นหาเส้นทางเชิงรับ (Reactive routing protocol) สำหรับโพรโทคอลค้นหาเส้นทางเชิงรุก เป็นโพรโทคอลที่ใช้ตารางในการค้นหาค้นหาเส้นทาง โดยบรรจุข้อมูลที่อยู่ที่ของโหนดข้างเคียง และมีการอัปเดตข้อมูลระหว่างโหนดใด ๆ เป็นระยะ ๆ ข้อดีของโพรโทคอลชนิดนี้คือ สามารถส่งข้อมูลได้ทันทีที่ต้องการเพราะมีการบรรจุข้อมูลใหม่อยู่เสมอ ข้อเสียคือต้องบรรจุข้อมูลเส้นทางปริมาณมาก ทำให้มี overhead ที่มากขึ้น ตัวอย่างของโพรโทคอลมาตรฐานชนิดนี้ที่จะถูกนำมาศึกษาได้แก่ DSDV (Destination-Sequenced Distance Vector) และ OLSR (Optimized Link State Routing) สำหรับโพรโทคอลค้นหาเส้นทางเชิงรับ ถูกออกแบบมาเพื่อให้สูญเสียพลังงานในส่วน overhead น้อยลง โดยใช้เวกเตอร์บ่งชี้ระยะทางและจะระบุตำแหน่งของโหนดเป้าหมายก็ต่อเมื่อมีการร้องขอข้อมูลจากโหนดที่จะส่งข้อมูลเท่านั้น ตัวอย่างของโพรโทคอลชนิดนี้ที่จะนำมาเปรียบเทียบ คือ DSR (the Dynamic Source Routing) และ AODV (Ad Hoc On-Demand Distance-Vector)

การทดลองที่หนึ่ง เป็นการศึกษาการทำงานของโครงข่ายชื่อข้อมูล (NDN) เปรียบเทียบกับโครงข่ายแบบ TCP/IP ที่ใช้โพรโทคอลค้นหาเส้นทาง 4 ชนิด ได้แก่ DSR OLSR AODV และ DSDV การศึกษาโพรโทคอลดังกล่าวนี้ จะมีการวิเคราะห์และเปรียบเทียบสมรรถนะในด้านอัตราความสำเร็จของการส่งข้อมูลและค่าเฉลี่ยของการประวิงเวลา เพื่อรับประกันว่าผู้รับจะได้รับข้อมูลอย่างแน่นอนและใช้เวลาโดยเฉลี่ยที่เหมาะสม โดยแบ่งการจำลองออกเป็น 2 กรณี ได้แก่

- แบบจำลองเบื้องต้นการเคลื่อนที่แบบกิริต ขนาด 3x3
- แบบจำลองการเคลื่อนที่ที่เก็บจากข้อมูลจริงในสภาพโมเดล

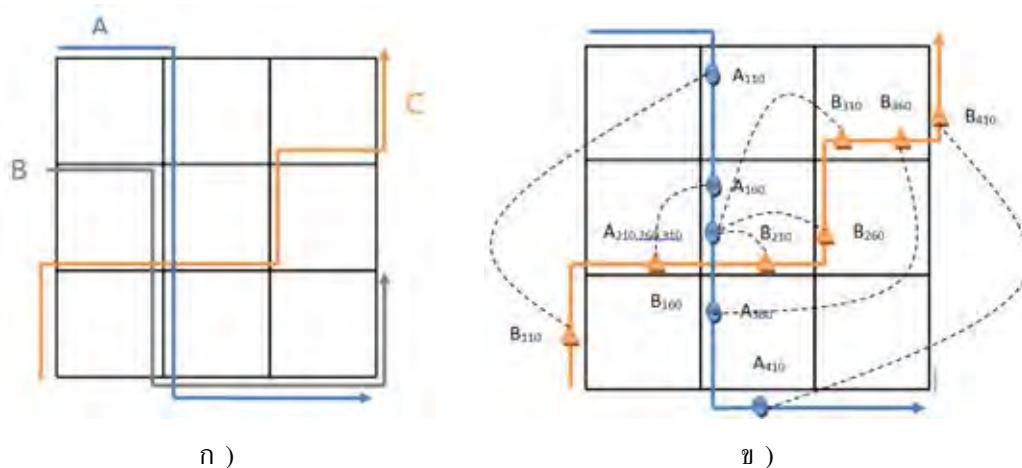
4.2.1 โครงข่ายแบบง่าย : โครงข่ายแบบกริดขนาด 3x3

การทดลองนี้ มีการจำลองยานพาหนะที่เคลื่อนที่ไปบนถนนที่ถูกสร้างขึ้นมาจากโปรแกรม SUMO โดยมีการคำนวณการเคลื่อนที่ของยานพาหนะตามรูปแบบการจำลองการเคลื่อนที่ของโหนดแบบ car-following model (ตามที่ได้อธิบายไปแล้วในบทที่ 2) รูปแบบของถนนมีลักษณะแบบกริดขนาด 3x3 พื้นที่ทั้งหมดมีขนาด 900 x 900 ตารางเมตร มียานพาหนะที่ถูกสร้างขึ้นจากเส้นทาง A จำนวน 20 คัน เส้นทาง B จำนวน 30 คัน เส้นทาง C จำนวน 50 คัน รวมจำนวนยานพาหนะทั้งหมด 100 คัน ดังแสดงในรูปที่ 4.4ก) โดยบังคับให้ความเร็วของยานพาหนะเป็นแบบสุ่มและมีความเร็วไม่เกิน 5 เมตรต่อวินาที

การติดต่อสื่อสารระหว่างโหนดใด ๆ มีทั้งหมด 10 คู่ โดยโปรแกรมจะสุ่มเลือกสถานะการณั้ดังนี้

- คู่ยานพาหนะที่เดินบนเส้นทาง A สื่อสารกับยานพาหนะที่เดินบนเส้นทาง C ทั้งหมด 4 คู่ยานพาหนะ
- คู่ยานพาหนะที่เดินบนเส้นทาง B สื่อสารกับยานพาหนะที่เดินบนเส้นทาง C ทั้งหมด 3 คู่ยานพาหนะ
- คู่ยานพาหนะที่เดินบนเส้นทาง B เดียวกัน ทั้งหมด 1 คู่ยานพาหนะ
- คู่ยานพาหนะที่เดินบนเส้นทาง C เดียวกัน ทั้งหมด 2 คู่ยานพาหนะ

จากรูปที่ 4.4ข) แสดงตัวอย่างโหนด (A_t, B_t) มีการเคลื่อนที่ตามเส้นทาง A และ B โดย t แทนเวลาที่ตำแหน่งต่าง ๆ มีหน่วยเป็นวินาที



รูปที่ 4.4: ก) รูปแบบการเคลื่อนที่ของยานพาหนะบนถนนตามเส้นทาง A B และ C และ ข) ตัวอย่างการสื่อสารระหว่างโหนด (A_t, B_t) ณ เวลาต่าง ๆ

การจำลองระบบการสื่อสารของโครงข่ายแบบ TCP/IP การส่งแพ็กเก็ตของโพรโทคอลการค้นหาเส้นทางในระดับชั้นประยุกต์ จะมีการจำลองการส่งข้อมูลแบบ UDP server และ UDP client ใน

ตารางที่ 4.2: การตั้งค่าของผลการจำลองโครงข่ายแบบกริดขนาด 3x3

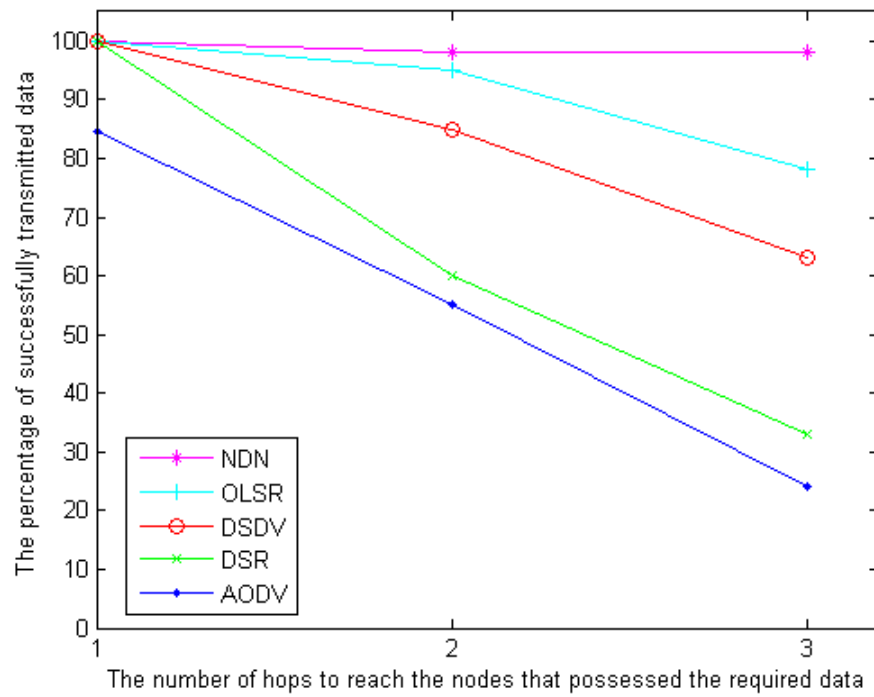
Parameter	Value
Number of simulations (times)	100
Simulation time (s)	700
MAC protocol	IEEE802.11p
Routing protocol	DSR, DSDV, OLSR, AODV, NDN
Packet size (bytes)	1024
Transmission range (m)	300
Mobility model	Car following
Maximum mobile speed (m/s)	15
Number of nodes	100
Area (m^2)	900x900
Transmit power	30 dBm
Speed	5 m/s
Number of pairs between server and client	10

อัตรา 2.045 kbps ซึ่งการทำงานชนิดนี้ มีความใกล้เคียงกับการทำงานแบบโครงข่ายข้อมูลชื่อ กล่าวคือ เมื่อยานพาหนะต้องการข้อมูล ยานพาหนะจะทำงานแบบ UDP client เพื่อส่งแพ็กเก็ตเกิดการร้องขอไปยังโนดเซิร์ฟเวอร์ที่ทำงานแบบ UDP server ซึ่งพารามิเตอร์ที่ใช้ในการจำลองแสดงในตารางที่ 4.2

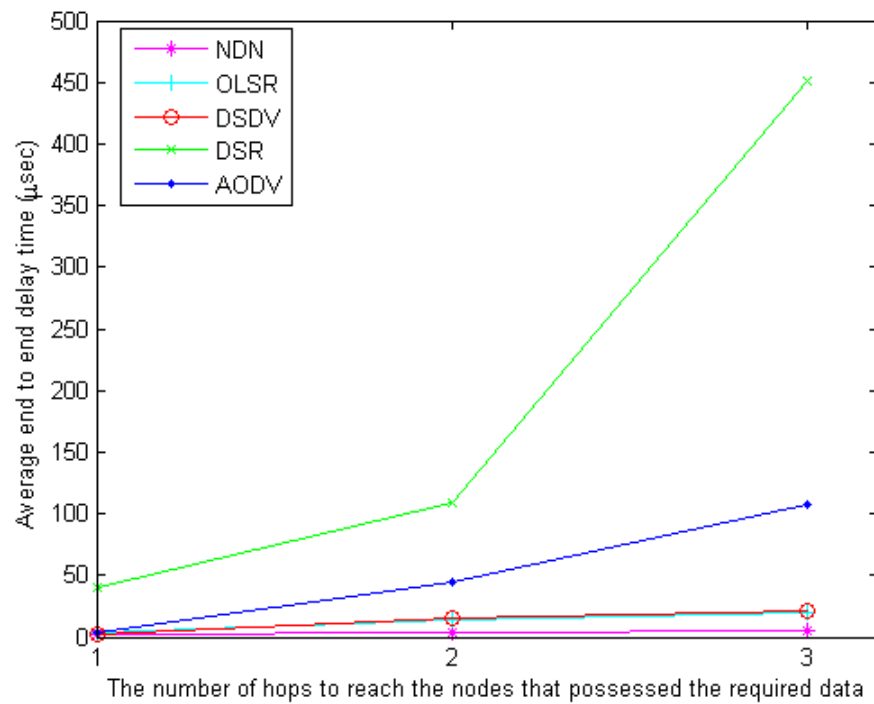
ผลการจำลองและการวิเคราะห์ พารามิเตอร์ที่ใช้เป็นตัวชี้วัดสมรรถนะในการทดลองนี้ ประกอบไปด้วย

1. อัตราร้อยละของการส่งข้อมูลสำเร็จ (The percentage of successfully transmitting data) คือ จำนวนแพ็กเก็ตข้อมูลที่โนดปลายทางได้รับสำเร็จต่อปริมาณโหนดขอบริการที่โนดต้นทาง
2. เวลาประวิงเฉลี่ยจากปลายถึงปลาย (Average end-to-end delay) คือ เวลาเฉลี่ยที่ใช้ในการส่งแพ็กเก็ตจำนวนหนึ่งแพ็กเก็ตจากโนดต้นทางไปยังโนดปลายทาง ซึ่งมีหน่วยเป็น วินาที

ในรูปที่ 4.5 แสดงถึงอัตราร้อยละของการส่งข้อมูลสำเร็จบนแบบจำลองยานพาหนะ ณ แต่ละจำนวนช่วงเชื่อมต่อ ที่มีแกน Y เป็นค่าอัตราร้อยละของการส่งข้อมูลสำเร็จ และแกน X เป็นจำนวนช่วงเชื่อมต่อมีค่า 1 2 และ 3 ฮอป สังเกตได้ว่า 1 ช่วงเชื่อมต่อ โพรโทคอล AODV มีอัตราร้อยละของการส่งข้อมูลสำเร็จประมาณ 85% ซึ่งมิต้าน้อยที่สุด และโพรโทคอลค้นหาเส้นทางอีก 3 ชนิด คือ OLSR DSDV และ DSR มีอัตราการประสบความสำเร็จในการส่งข้อมูลต่ำลงแบบเชิงเส้นตามจำนวนช่วงเชื่อมต่อ โดยมีสาเหตุมาจากยานพาหนะนั้นมีการเคลื่อนที่ ดังนั้นการปรับปรุงตารางข้อมูลเส้นทางตามคาบที่กำหนดไม่สามารถปรับปรุงได้ทันที โพรโทคอลค้นหาเส้นทางที่สร้างเส้นทางเพื่อใช้ในการส่งข้อมูลจึงไม่สามารถส่งข้อมูลได้ เพราะ ณ เวลาที่ข้อมูลเดินทางถึงตำแหน่งปลายทาง ยานพาหนะได้มีการเคลื่อนที่ออกจากตำแหน่งนั้นแล้ว ซึ่งเมื่อเปรียบเทียบกับโครงข่ายชื่อข้อมูล โครงข่ายชื่อข้อมูลจะมีอัตราร้อยละของการส่งข้อมูลสำเร็จถึง 100% เนื่องจากโพรโทคอลไม่ต้องสร้างตารางข้อมูล



รูปที่ 4.5: อัตราร้อยละของการส่งข้อมูลสำเร็จบนแบบจำลองยานพาหนะ แต่ละจำนวนช่วงเชื่อมต่อ



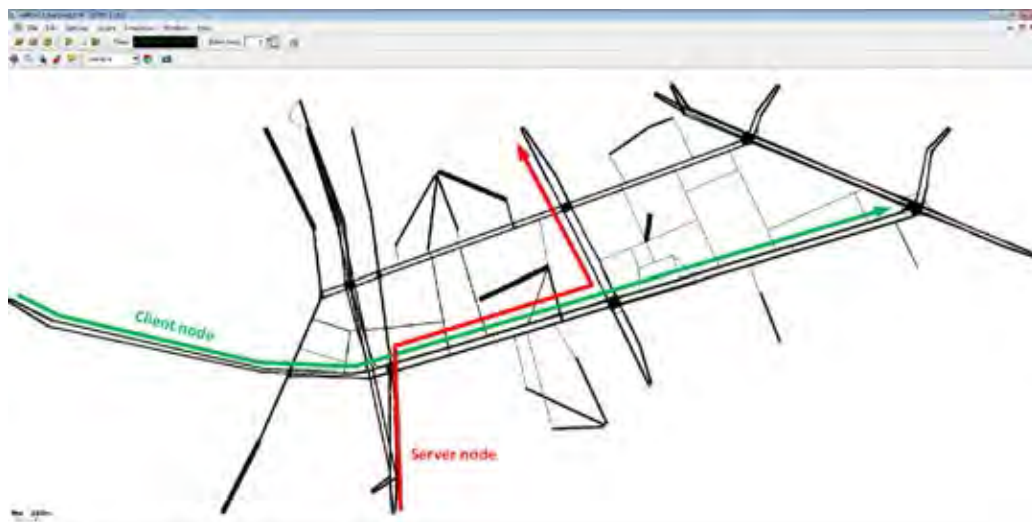
รูปที่ 4.6: เวลาประวิงเฉลี่ยในการส่งข้อมูลบนแบบจำลองยานพาหนะ แต่ละจำนวนช่วงเชื่อมต่อ

เส้นทาง และยานพาหนะสามารถแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลออกไปได้เรื่อยๆ จนกว่าจะได้รับการตอบกลับของแพ็กเก็ตข้อมูลนั้นในขณะที่ยานพาหนะมีการเคลื่อนที่

ในรูปที่ 4.6 แสดงถึงเวลาประวิงเฉลี่ยในการส่งข้อมูลบนแบบจำลองยานพาหนะ ณ แต่ละจำนวนช่วงเชื่อมต่อ ที่มีแกน Y เป็นเวลาประวิงเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนช่วงเชื่อมต่อมีค่า 1 2 และ 3 ฮอป สังเกตได้ว่า โพรโทคอล DSR มีเวลาประวิงเฉลี่ยตามจำนวนช่วงเชื่อมต่อสูงที่สุด เนื่องจากยานพาหนะที่เป็นจุดเชื่อมต่อระหว่างทางในเส้นทางการส่งข้อมูลได้เคลื่อนที่ออกไปนอกขอบเขตรัศมีที่ยานพาหนะข้างเคียงเดิมจะติดต่อกับได้ จึงทำให้ไม่สามารถส่งข้อมูลต่อไปได้นอกจากนี้ โพรโทคอล DSR จะค้นหาเส้นทางทั้งหมดไปพร้อม ๆ กับส่งข้อมูล ดังนั้นหากยานพาหนะมีการเคลื่อนไหวจะส่งผลให้โพรโทคอล DSR ต้องเก็บข้อมูลเส้นทางใหม่ทั้งหมดตลอดเวลา ส่งผลให้มีเวลาประวิงที่สูงขึ้นตามจำนวนช่วงเชื่อมต่อ ซึ่งสูงที่สุดถึงประมาณ 0.451 มิลลิวินาที ในขณะที่โพรโทคอลแบบเชิงรับทั้ง DSDV และ OLSR มีเวลาประวิงเฉลี่ยที่เกือบเท่ากัน และโพรโทคอลโครงข่ายชื่อข้อมูล จะมีเวลาประวิงเฉลี่ยต่ำที่สุด เพราะโครงข่ายชื่อข้อมูลไม่มีการสร้างเส้นทางการส่งข้อมูลก่อน ทำให้ไม่เสียเวลาในการส่งข้อมูลเมื่อเทียบกับโพรโทคอลอีก 4 ชนิด

4.2.2 โครงข่ายจากข้อมูลจริง : สาทรโมเดล

การทดลองที่ 2 นี้ มีการนำข้อมูลจริงจากข้อมูลการจัดการจราจรของถนนสาทร ของวันที่ 7 กรกฎาคม พ.ศ. 2557 ดังในรูปที่ 4.7 ซึ่งแสดงแบบจำลองบนถนนสาทรและเส้นทางการเคลื่อนที่ของคู่ยานพาหนะระหว่าง Server node และ Client node ช่วงเวลา 15.30.00 – 15.46.40 น. หรือ 1000s จากข้อมูลจริง โดยมียานพาหนะบนถนนทั้งหมดประมาณ 6410 คัน แต่ในการทดลองนี้ ได้ลดจำนวนของยานพาหนะเป็น 0.2%, 2% และ 10% หรือมีจำนวนยานพาหนะ 12 64 และ 641 คัน ตามลำดับโดยพารามิเตอร์ที่ใช้ในการจำลองแสดงในตารางที่ 4.3



รูปที่ 4.7: แบบจำลองบนถนนสาทรและเส้นทางการเคลื่อนที่ของคู่ยานพาหนะ ณ วันที่ 7 กรกฎาคม พ.ศ. 2557 เวลา 15.30.00 – 15.46.40 น.

ผลการจำลองและการวิเคราะห์ พารามิเตอร์ที่ใช้เป็นตัวชี้วัดสมรรถนะในการทดลองนี้ ประกอบไปด้วย

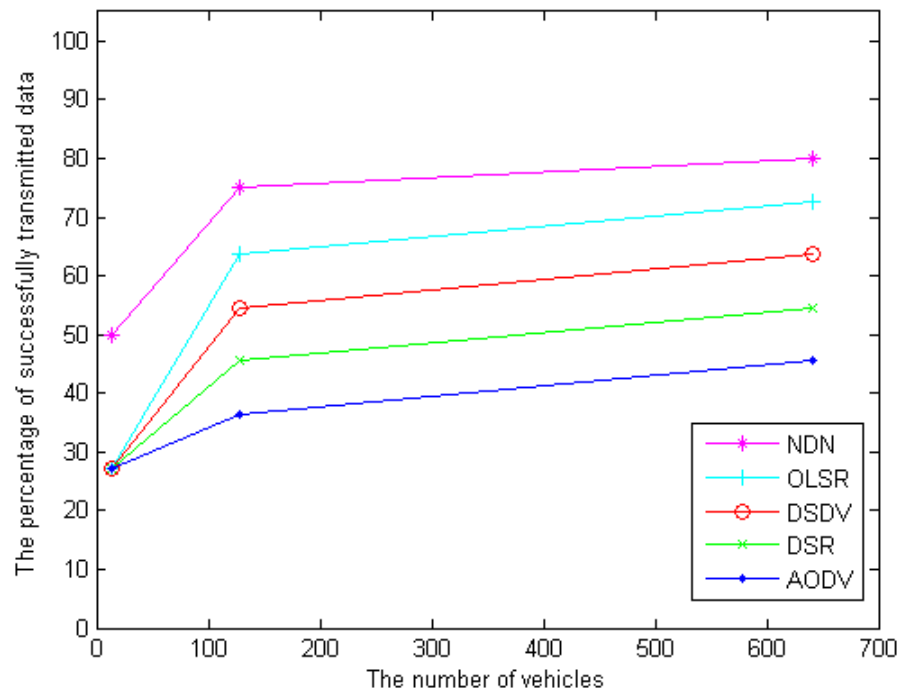
ตารางที่ 4.3: การตั้งค่าของผลการจำลองโครงข่ายจากข้อมูลจริง

Parameter	Value
Number of simulations (times)	100
Simulation time (s)	1000
MAC protocol	IEEE802.11p
Packet size (byte)	1024
Transmission range (m)	500
Maximum mobile speed (m/s)	15
Transmit power	30 dBm
Number of nodes	12, 128, 641
Bandwidth	11 Mb/s
Number of pairs between server and client	10

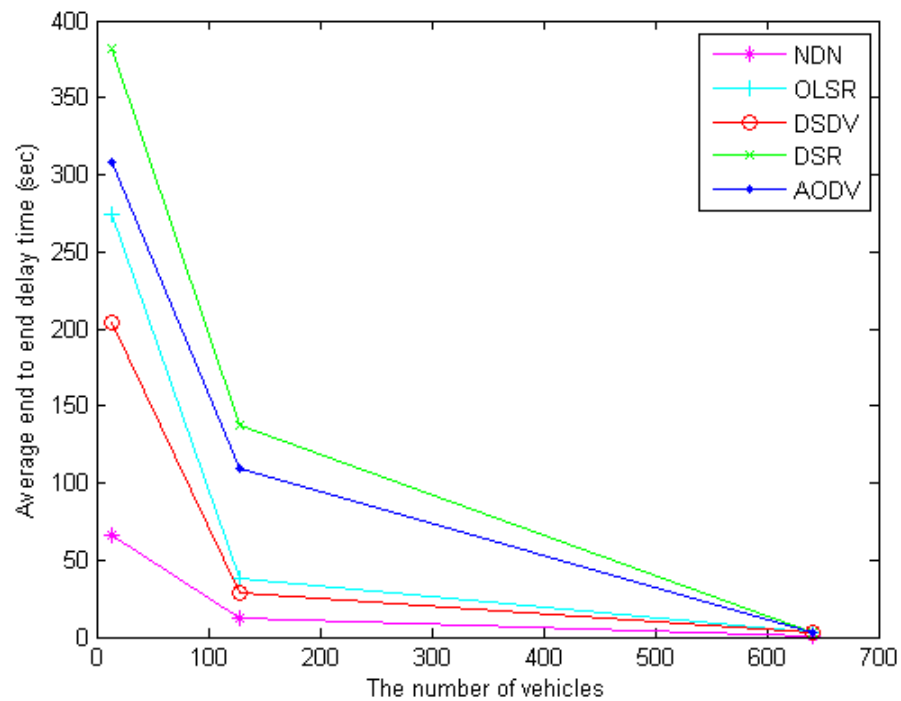
1. อัตราร้อยละของการส่งข้อมูลสำเร็จ (The percentage of successfully transmitting data) คือ จำนวนแพ็กเก็ตข้อมูลที่โหนดปลายทางได้รับสำเร็จต่อปริมาณโหนดขอบริการที่โหนดต้นทาง
2. เวลาประวิงเฉลี่ยจากปลายถึงปลาย (Average end-to-end delay) คือ เวลาเฉลี่ยที่ใช้ในการส่งแพ็กเก็ตจำนวนหนึ่งแพ็กเก็ตจากโหนดต้นทางไปยังโหนดปลายทาง มีหน่วยเป็น วินาที

ในรูปที่ 4.8 แสดงถึงอัตราร้อยละของการส่งข้อมูลสำเร็จของยานพาหนะบนถนนสาทร ที่มีแกน Y เป็นค่าอัตราร้อยละของการส่งข้อมูลสำเร็จ และแกน X เป็นจำนวนยานพาหนะมีค่า 12 128 และ 641 คัน สังเกตได้ว่าโพรโทคอล AODV มีอัตราร้อยละของการส่งข้อมูลสำเร็จประมาณ 30-33% ซึ่งมีค่าน้อยที่สุด และโพรโทคอล OLSR มีอัตราร้อยละของการส่งข้อมูลสำเร็จประมาณ 30-72% ซึ่งมีค่ามากที่สุดในโพรโทคอลแบบ TCP/IP โพรโทคอลค้นหาเส้นทางทั้ง 4 ชนิดเรียงลำดับ อัตราร้อยละของการส่งข้อมูลสำเร็จจากมากไปน้อย คือ OLSR DSDV DSR และ AODV ตามลำดับ ซึ่งอัตราการประสบความสำเร็จในการส่งข้อมูลจะมีค่าสูงขึ้นตามจำนวนยานพาหนะที่อยู่บนท้องถนน เนื่องจากโอกาสที่โหนดต้นทางจะสามารถส่งข้อมูลถึงโหนดปลายทางได้สำเร็จจะมากขึ้นตามจำนวนยานพาหนะที่เพิ่มขึ้น อย่างไรก็ตาม เมื่อเปรียบเทียบกับโครงข่ายชื่อข้อมูลที่มีอัตราร้อยละของการส่งข้อมูลสำเร็จถึง 50-80% เนื่องจากยานพาหนะสามารถแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลออกไปได้เรื่อยๆ จนกว่าจะได้รับการตอบกลับของแพ็กเก็ตข้อมูลนั้นในขณะที่ยานพาหนะมีการเคลื่อนที่ ทำให้จำนวนยานพาหนะที่เพิ่มขึ้นในท้องถนนส่งผลให้โพรโทคอลทำงานได้ดีขึ้น

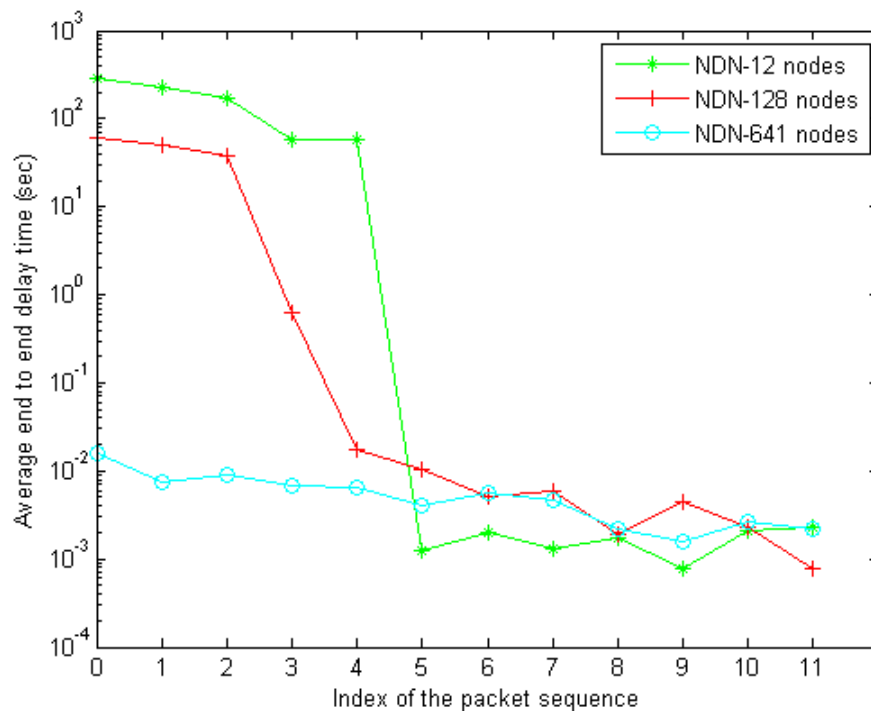
ในรูปที่ 4.9 แสดงถึงเวลาประวิงเฉลี่ยในการส่งข้อมูลของยานพาหนะบนถนนสาทร ที่มีแกน Y เป็นเวลาประวิงเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนยานพาหนะมีค่า 12 128 และ 641 คัน สังเกตได้ว่า โพรโทคอล DSR มีเวลาประวิงเฉลี่ยสูงที่สุด เท่ากับ 370 วินาที รองลงมาคือโพรโทคอล AODV ในขณะที่โพรโทคอลแบบเชิงรับทั้ง DSDV และ OLSR มีเวลาประวิงเฉลี่ยที่เกือบเท่ากัน ซึ่งการทดลองนี้มีแนวโน้มเหมือนกับการทดลองที่ 1 เมื่อเปรียบเทียบกับโพรโทคอลโครงข่ายชื่อข้อมูล จะมีเวลาประวิงเฉลี่ยต่ำที่สุด และโพรโทคอลทั้ง 5 ชนิด จะมีแนวโน้มของเวลาประวิงเฉลี่ยในการส่งข้อมูลที่ต่ำลง เนื่องจากจำนวนยานพาหนะที่มากขึ้น ทำให้มีโอกาสที่โหนดต้นทางจะสามารถส่งข้อมูลถึงโหนดปลายทางได้สำเร็จมีมากขึ้นตาม



รูปที่ 4.8: อัตราร้อยละของการส่งข้อมูลสำเร็จของยานพาหนะบนถนนสาทร



รูปที่ 4.9: เวลาประวิงเฉลี่ยในการส่งข้อมูลของยานพาหนะบนถนนสาทร



รูปที่ 4.10: เวลาประวิงในการส่งข้อมูลของยานพาหนะบนถนนสาทรตามดัชนีลำดับของแพ็กเกต

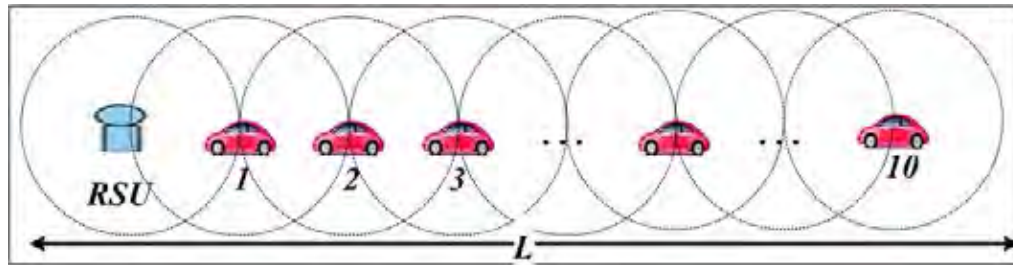
อย่างไรก็ตามเมื่อพิจารณาค่าประวิงเวลาในการส่งข้อมูลของยานพาหนะบนถนนสาทรเฉพาะในโพรโทคอลโครงข่ายข้อมูลชื่อ ที่มีแกน Y เป็นเวลาประวิงเฉลี่ยในการส่งข้อมูล และแกน X เป็นดัชนีลำดับของแพ็กเกต มีค่าตั้งแต่ 0-11 ดังแสดงในรูปที่ 4.10 ซึ่งเปรียบเทียบที่ยานพาหนะจำนวน 12 128 และ 641 คัน สังเกตได้ว่าเมื่อเริ่มการส่งของดัชนีที่ 0 จำนวนยานพาหนะ 12 คัน จะมีเวลาประวิงที่สูงที่สุด มีค่าประมาณ 290 วินาที จนเวลาผ่านไป โหนดต้นทางกับโหนดปลายทางสามารถส่งข้อมูลกันได้ ทำให้เวลาประวิงเฉลี่ยมีค่าลดลง

จากการทดลองที่ผ่านมา ถึงแม้ว่าโพรโทคอลโครงข่ายชื่อข้อมูลจะมีอัตราร้อยละของการส่งข้อมูลสำเร็จที่สูงกว่าโพรโทคอลแบบ TCP/IP แต่ก็ยังมีปัญหาเรื่องค่าเฉลี่ยประวิงเวลาที่สูง เนื่องจากโหนดไม่ทราบว่าจะแหล่งข้อมูลอยู่ตรงไหน จำเป็นต้องปรับปรุงประสิทธิภาพในการทำงาน ทำให้เกิดแนวความคิดในการออกแบบโพรโทคอล โดยการเพิ่มโหนดข้างถนนเข้ามาในระบบเพื่อเป็นแหล่งข้อมูลหลัก เพื่อช่วยลดปัญหาดังกล่าวได้

4.3 ผลการจำลองเพื่อศึกษาการทำงานของโพรโทคอลชื่อข้อมูล

การทดลองที่ 3 นี้ เพื่อศึกษาการทำงานของโพรโทคอลชื่อข้อมูล เมื่อโหนดมีการเคลื่อนที่บนถนนเส้นตรงจำนวน 1 เลน เริ่มต้นยานพาหนะมีระยะห่างเท่ากับ 200 เมตร และใช้การเคลื่อนที่แบบแรนดอมเวย์พอย์ ที่ความเร็ว 5 และ 10 เมตรต่อวินาที โดยยานพาหนะแต่ละคันจะมีระยะห่างระหว่างโหนดข้างถนนเป็นระยะ 2-10 ฮอป ดังแสดงในรูปที่ 4.11 กล่าวคือ เมื่อยานพาหนะที่อยู่ในช่วงเชื่อมต่อต่าง ๆ ต้องการข้อมูล ยานพาหนะคันนั้นจะส่งแพ็กเกตการร้องขอไปที่โหนดข้างถนน ซึ่ง

แต่ละเวลามียานพาหนะต้องการข้อมูลเพียง 1 คันเท่านั้น โดยส่งการร้องขอข้อมูลทุก ๆ 100 วินาที และพารามิเตอร์ที่ใช้ในการทดลองนี้ แสดงในตารางที่ 4.4



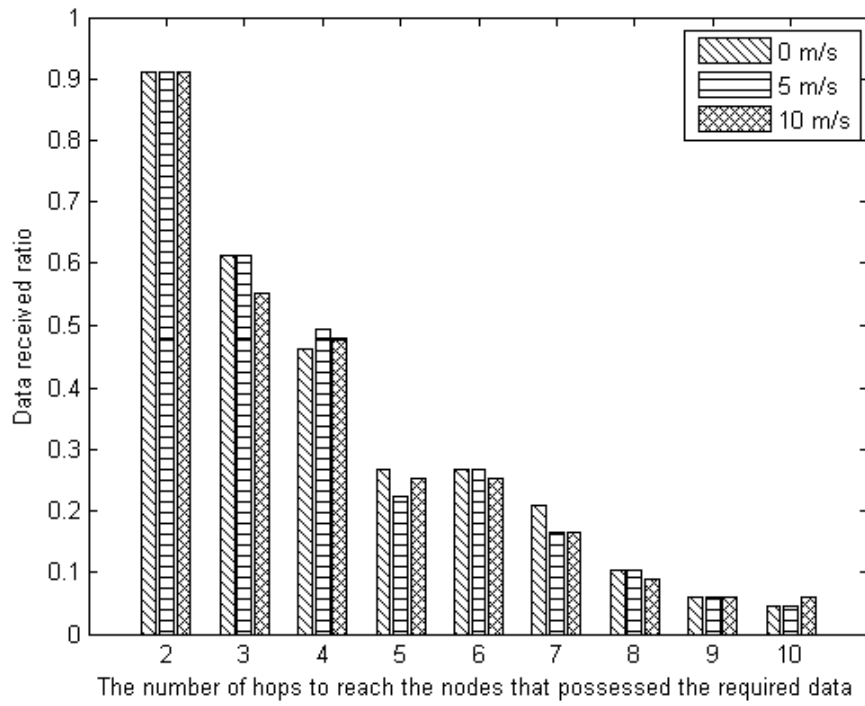
รูปที่ 4.11: แสดงรูปแบบการเคลื่อนที่ของยานพาหนะ

ตารางที่ 4.4: การตั้งค่าของผลการจำลองเพื่อศึกษาการทำงานของโพรโทคอลชื่อข้อมูล

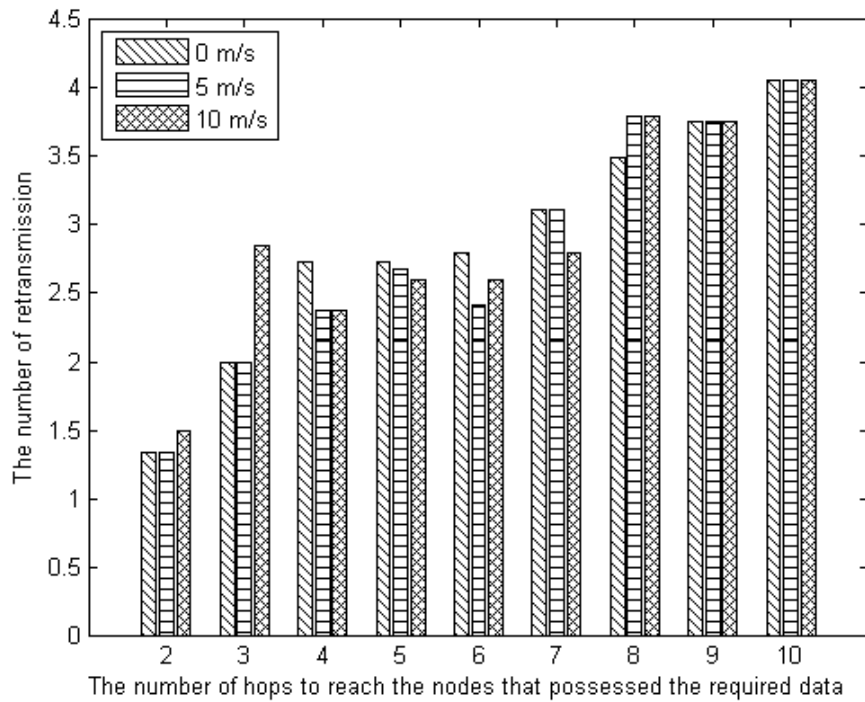
Parameter	Value
Number of simulations (times)	100
Simulation time (s)	1000
MAC protocol	IEEE802.11p
Packet size (byte)	1024
Total of packet (packets)	10
Transmission range (m)	200
Road length (km)	2
Maximum mobile speed (m/s)	0, 5, 10
Transmit power	16.0206 dBm
Number of nodes	2-10
Bandwidth	11 Mb/s

ผลการจำลองและการวิเคราะห์ พารามิเตอร์ที่ใช้เป็นตัวชี้วัดสมรรถนะในการทดลองนี้ ประกอบไปด้วย

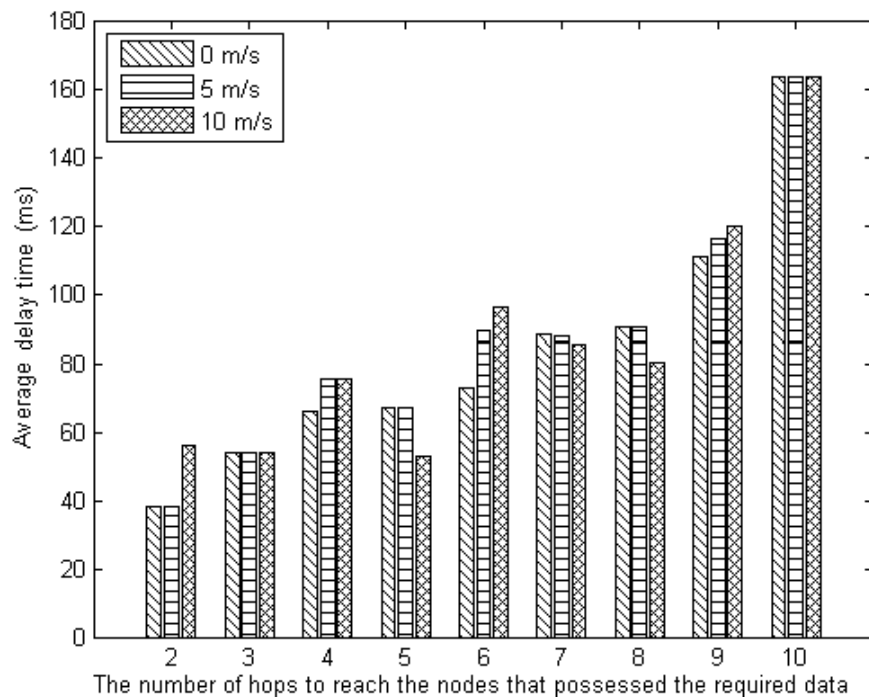
1. อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ (Data Received Ratio หรือ *DRR*) คือ จำนวนแพ็กเก็ตข้อมูลที่โหนดทั้งหมดได้รับสำเร็จในระบบต่อจำนวนแพ็กเก็ตที่โหนดทั้งหมดส่งสำเร็จในระบบ
2. เวลาประวิงเฉลี่ย (Average delay time) คือ เวลาเฉลี่ยที่ใช้ในการส่งแพ็กเก็ตจำนวนหนึ่งแพ็กเก็ตจากโหนดผู้ร้องขอไปยังโหนดข้างถนน ซึ่งมีหน่วยเป็น วินาที
3. จำนวนการส่งต่อข้อมูลซ้ำ (The number of retransmission) คือ จำนวนการส่งแพ็กเก็ตการร้องขอซ้ำ



รูปที่ 4.12: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ



รูปที่ 4.13: จำนวนการส่งต่อข้อมูลซ้ำ



รูปที่ 4.14: เวลาประวิงข้อมูลเฉลี่ย

จากรูปที่ 4.12 แสดงถึงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ ที่มีแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X เป็นจำนวนช่วงเชื่อมต่อมีค่า 2-10 ฮอป สังเกตได้ว่า 2 ช่วงเชื่อมต่อ โพรโทคอลจะมีอัตราส่วนการรับข้อมูลสำเร็จประมาณ 90% และจะมีอัตราส่วนการรับข้อมูลสำเร็จต่ำสุดประมาณ 10% ที่ช่วงเชื่อมต่อ 10 ฮอป โดยที่แนวโน้มอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จมีค่าต่ำลงแบบ exponential ตามจำนวนช่วงเชื่อมต่อ เนื่องจากว่าระยะช่วงเชื่อมต่อมากขึ้น การแพร่กระจายแพ็กเก็ตการร้องขอข้อมูลในแต่ละช่วงเชื่อมต่อจะมีโอกาสที่แพ็กเก็ตการร้องขอชนกันสูงขึ้น และเมื่อเปรียบเทียบกันระหว่างยานพาหนะที่มีการเคลื่อนที่ 0 5 และ 10 เมตรต่อวินาที ตามลำดับ พบว่าแนวโน้มมีค่าไม่แตกต่างกันมาก เพราะเมื่อพิจารณาในสเกลเวลาของการสื่อสารที่ใช้ความเร็วแสงในการส่งแพ็กเก็ต เทียบกับสเกลเวลาของการเคลื่อนที่ของยานพาหนะมีหน่วยที่ต่างกันมาก จึงทำให้เสมือนว่ายานพาหนะมีการหยุดนิ่งในขณะที่มีการส่งแพ็กเก็ต

ในรูปที่ 4.13 แสดงจำนวนการส่งต่อข้อมูลซ้ำ ณ แต่ละจำนวนช่วงเชื่อมต่อ ที่มีแกน Y เป็นจำนวนการส่งต่อข้อมูลซ้ำ และแกน X เป็นจำนวนช่วงเชื่อมต่อมีค่า 2-10 ฮอป สังเกตได้ว่า โพรโทคอลจะมีอัตราส่วนการรับข้อมูลที่ต่ำลง เนื่องจากโหนดที่ร้องขอข้อมูลไม่ได้รับแพ็กเก็ตข้อมูล ทำให้โหนดจำเป็นต้องส่งแพ็กเก็ตการร้องขอซ้ำ ๆ จึงส่งผลให้มีจำนวนการส่งต่อข้อมูลซ้ำมีค่ามากขึ้นตามจำนวนช่วงเชื่อมต่อ

ในรูปที่ 4.14 แสดงเวลาประวิงเฉลี่ยในการส่งข้อมูลบนแบบจำลองยานพาหนะ ณ แต่ละจำนวนช่วงเชื่อมต่อ ที่มีแกน Y เป็นเวลาประวิงเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนช่วงเชื่อมต่อ มีค่า 2-10 ฮอป สังเกตได้ว่า เมื่อจำนวนช่วงเชื่อมต่อมากขึ้น เวลาประวิงเฉลี่ยมีค่าสูงขึ้นแบบ exponential เนื่องจากว่าโหนดที่ร้องขอข้อมูลตามจำนวนช่วงเชื่อมต่อที่มากขึ้นเป็นการเพิ่มจำ-

นวนการติดต่อที่มากขึ้นตามไปด้วย จึงทำให้เกิดการชนกันในเลเยอร์ดาต้าลิงค์ที่สูงขึ้น และแพ็กเก็ตที่ชนกันจะต้องรอเวลาการส่งต่อข้อมูลออกไปแบบ exponential ค่าเวลาการประวิงเฉลี่ยจึงมีแนวโน้มแบบ exponential

สรุป การทดลองนี้พบปัญหาของการชนกันของแพ็กเก็ตเมื่อช่วงเชื่อมต่อมีมากขึ้น ส่งผลให้สมรรถนะของระบบมีค่าลดลง การออกแบบโพรโทคอลนี้จึงต้องคำนึงถึงปัญหาดังกล่าว โดยออกแบบสมการเวลาการรอคอยเพื่อส่งต่อข้อมูลให้มีความเหมาะสมกับโครงข่ายแอตชอกสำหรับยานพาหนะ

4.4 ผลการจำลองสำหรับประเมินสมรรถนะของโพรโทคอลที่ออกแบบ

4.4.1 พารามิเตอร์ที่ใช้สำหรับประเมินสมรรถนะของโพรโทคอลที่ออกแบบ

การประเมินสมรรถนะของโพรโทคอลที่นำเสนอ นั้น มีพารามิเตอร์ที่ใช้เป็นตัวชี้วัดสมรรถนะของการออกแบบโพรโทคอลการแจกจ่ายข้อมูล ซึ่งพารามิเตอร์ที่ใช้ในการประเมินสมรรถนะในงานวิทยานิพนธ์ฉบับนี้ประกอบด้วยพารามิเตอร์ดังนี้

- อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ (Data Received Ratio หรือ *DRR*) คือ จำนวนแพ็กเก็ตข้อมูลทีโหนดทั้งหมดรับสำเร็จในระบบต่อจำนวนแพ็กเก็ตทีโหนดทั้งหมดส่งสำเร็จในระบบ ดังแสดงในสมการที่ (4.1)

$$DRR = \frac{\sum_{i=1}^N R_i}{\sum_{i=1}^N S_i} \quad (4.1)$$

เมื่อ

R_i คือ จำนวนแพ็กเก็ตที่ได้รับสำเร็จในแต่ละโหนด i มีค่าตั้งแต่ 1, 2, ..., N

S_i คือ จำนวนแพ็กเก็ตที่ได้ส่งสำเร็จในแต่ละโหนด i มีค่าตั้งแต่ 1, 2, ..., N

N คือ จำนวนโหนดทั้งหมดในระบบ

- เวลาการแจกจ่ายข้อมูลโดยรวม (Total dissemination time หรือ *TDT*) คือ เวลาการแจกจ่ายข้อมูลโดยรวม สำหรับการรับข้อมูลทั้งหมดในระบบ โดยคำนวณจากช่วงเวลาทีโหนดส่งแพ็กเก็ตการร้องขอตั้งแต่ครั้งแรก จนได้รับการตอบกลับแพ็กเก็ตข้อมูล เทียบกับอัตราส่วนการรับแพ็กเก็ตสำเร็จในพารามิเตอร์ข้างต้น มีหน่วยเป็น วินาที (s) ดังแสดงในสมการที่ (4.2)

$$TDT = \frac{\bar{T}_f}{DRR} \quad (4.2)$$

เมื่อ

\bar{T}_f คือ เวลาประวิงเฉลี่ยระหว่างการส่งแพ็กเก็ตร้องขอข้อมูลครั้งแรกและได้รับการตอบกลับแพ็กเก็ตข้อมูล (รวมช่วงเวลาที่มีการส่งแพ็กเก็ตการร้องขอข้อมูลซ้ำ)

DRR คือ อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ

- เวลาแจกจ่ายข้อมูล (Dissemination time หรือ DT) คือ เวลาแจกจ่ายข้อมูล โดย คำนวณจาก ช่วงเวลาที่โหนดส่งแพ็กเก็ตการร้องขอครั้งล่าสุดก่อนจะได้รับการตอบกลับแพ็กเก็ตข้อมูล เทียบ กับอัตราส่วนการรับแพ็กเก็ตสำเร็จในพารามิเตอร์ข้างต้น มีหน่วยเป็น วินาที (s) ดังแสดงใน สมการที่ (4.3)

$$DT = \frac{\bar{T}_l}{DRR}, \quad (4.3)$$

เมื่อ

\bar{T}_l คือ เวลาประวิงเฉลี่ยระหว่างส่งแพ็กเก็ตร้องขอข้อมูลครั้งล่าสุดก่อนจะได้รับการตอบกลับแพ็กเก็ตข้อมูล

DRR คือ อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ

- ปริมาณทราฟฟิกในระบบ (Traffic load หรือ TL) คือ แพ็กเก็ตข้อมูลที่โหนดทั้งหมดส่งสำเร็จในระบบ มีหน่วยเป็น แพ็กเก็ต ดังแสดงในสมการที่ (4.4)

$$TL = \sum_{i=1}^N S_i \quad (4.4)$$

- ค่าวิสัยสามารถ (Throughput หรือ TP) คือ ค่าวิสัยสามารถที่ระบบทำได้ในช่วงเวลาหนึ่ง สามารถคำนวณจากขนาดของแพ็กเก็ตข้อมูลเทียบกับเวลาประวิงเต็มทั้งหมด มีหน่วยเป็น ไบต์ต่อวินาที ดังแสดงในสมการที่ (4.5)

$$TP = \frac{PS}{TDT} \quad (4.5)$$

เมื่อ

PS คือ ขนาดของแพ็กเก็ตข้อมูลที่ถูกส่งไปในระบบ (packet size)

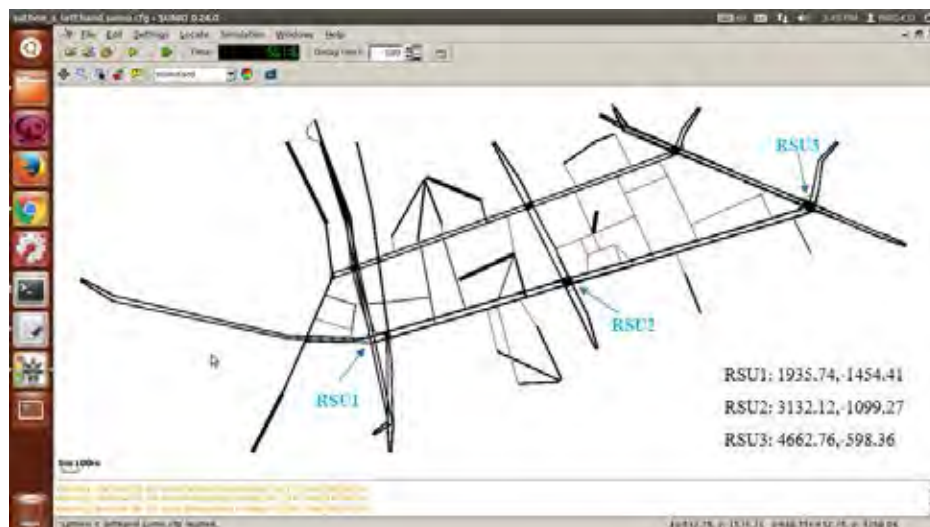
TDT คือ เวลาการแจกจ่ายข้อมูลโดยรวม

4.4.2 โครงสร้างระบบที่จำลอง

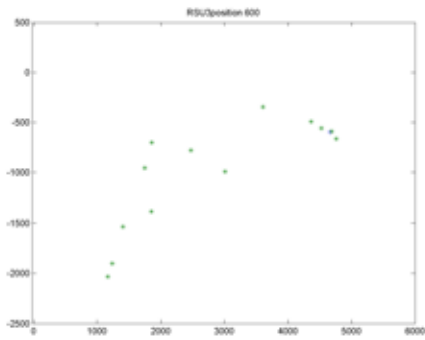
ในโครงสร้างระบบที่จำลอง จะให้มีตำแหน่ง RSU1 ตั้งที่แยกของถนนสาทร RSU2 ตั้งที่แยกถนนราชีวาส และ RSU3 ตั้งที่แยกถนนวิฑู ตั้งแสดงในรูปที่ 4.15 และ รูปที่ 4.16 โดยแสดงในแผนที่จริงและในโปรแกรม SUMO ตามลำดับ ในส่วนของการจำลองนี้ เราจะอธิบายเกี่ยวกับผลกระทบจากการปรับลดจำนวนยานพาหนะบนถนน ซึ่งในระบบมียานพาหนะทั้งหมด 6,410 คัน โดยปรับลดจำนวนยานพาหนะในอัตราส่วนร้อยละ 0.003 - 0.2 ของจำนวนยานพาหนะทั้งหมด ดังแสดงในรูปที่ 4.17 บนถนนสาทร ณ วันที่ 7 กรกฎาคม พ.ศ. 2557 เวลา 15.30-17.30 น. จำนวน 10,000 วินาที



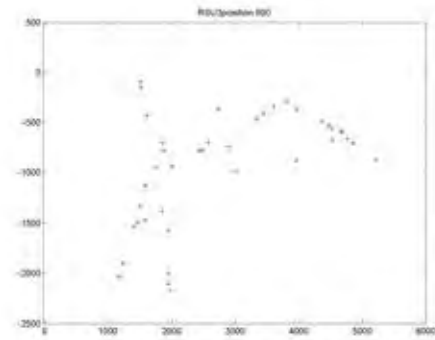
รูปที่ 4.15: ตำแหน่งของโนดข้างถนน บนแผนที่จริง



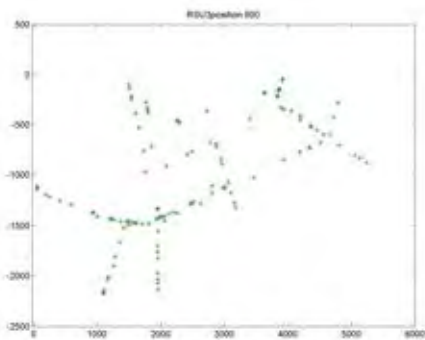
รูปที่ 4.16: ตำแหน่งของโนดข้างถนน บนโปรแกรม SUMO



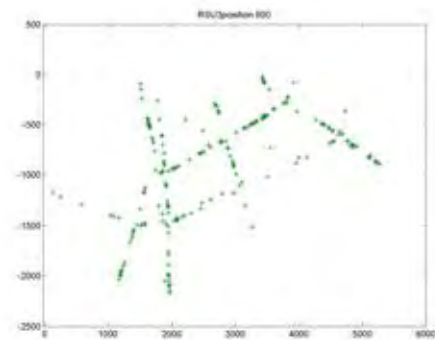
ก)



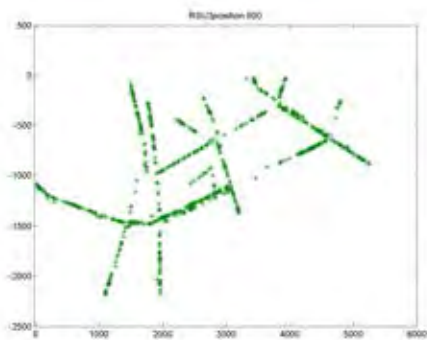
ข)



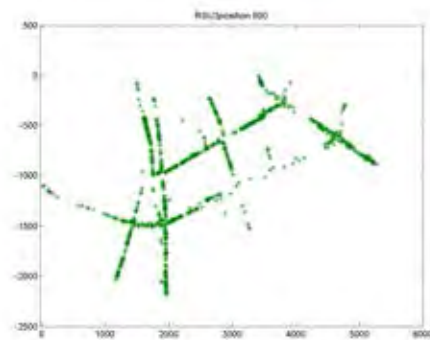
ค)



ง)



จ)



ฉ)

รูปที่ 4.17: ก) the penetration = 0.03% ข) the penetration = 0.09% ค) the penetration = 2%
 ง) the penetration = 6% จ) the penetration = 10% และ ฉ) the penetration = 20%.

4.4.3 ผลการจำลองของโพรโทคอลที่ออกแบบในโมดบรอดแคสต์

ในส่วนนี้เป็นการประเมินสมรรถนะของโพรโทคอลเมื่อมีการเพิ่มจำนวนยานพาหนะในโมดบรอดแคสต์ ซึ่งโพรโทคอลกำหนดให้มีโนดข้างถนน ซึ่งแสดงดังรูปที่ 4.15 โดยโนดข้างถนนมีการรวบรวมข้อมูลจากยานพาหนะในบริเวณที่อยู่ห่างออกไป 1 กิโลเมตร โดยพารามิเตอร์ที่ใช้ในการจำลองดังแสดงในตารางที่ 4.5

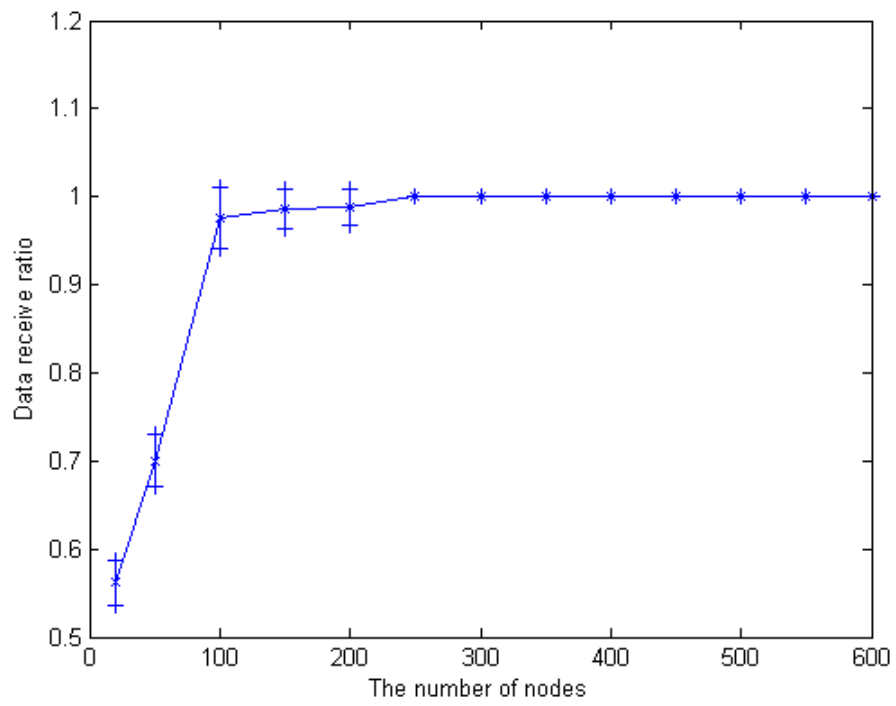
ตารางที่ 4.5: การตั้งค่าของผลการจำลองของโพรโทคอลที่ออกแบบในโมดบรอดแคสต์

Parameter	Value
Number of simulations (times)	100
Simulation time (s)	10,000
Physical/MAC protocols	IEEE802.11p
CW (min/max)	31/1023
Data rate(Mb/s)	11
Payload (bytes)	1024
Transmission range (m)	200
Carrier frequency (GHz)	5.9
Bit rate (Mbps)	2
Maximum mobile speed (m/s)	1.6
Number of vehicles on road	20, 50, 100, 150, ... , 600

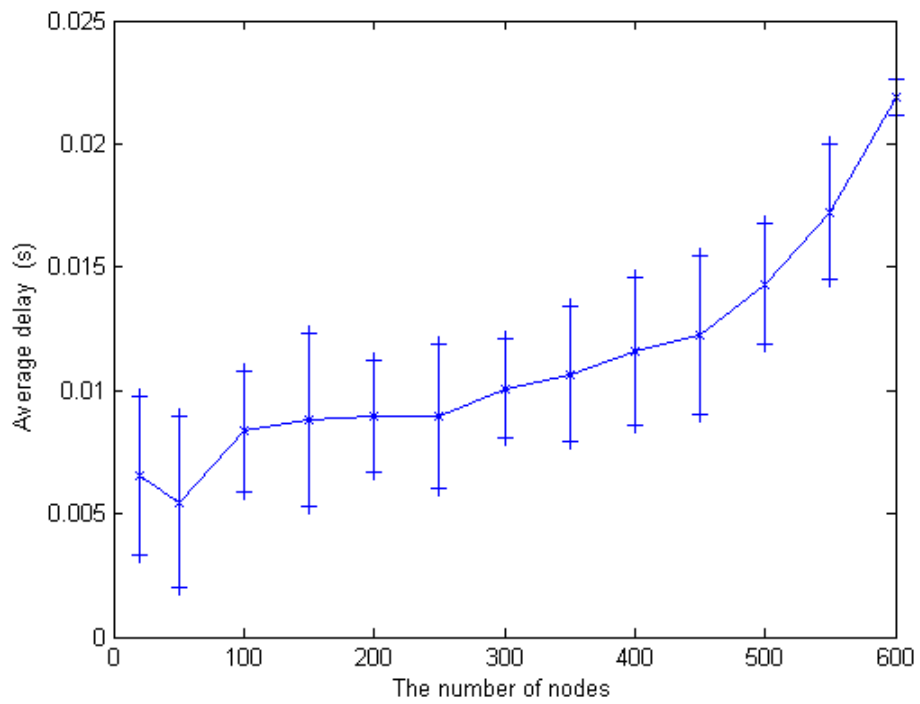
จากรูปที่ 4.18 แสดงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จที่มีช่วงของค่าความเชื่อมั่นอัตราร้อยละ 95 โดยมีแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X เป็นจำนวนโนดบนถนน โพรโทคอลจะมีอัตราส่วนการรับข้อมูลสำเร็จประมาณ 55-100% ตามจำนวนโนดที่เพิ่มขึ้นสังเกตได้ว่า อัตราส่วนการรับแพ็กเก็ตเข้าสู่ค่า 100% เมื่อจำนวนโนดมีค่า 250 แสดงให้เห็นว่าโพรโทคอลการเก็บข้อมูลมีสมรรถนะในการทำงานได้อย่างมีประสิทธิภาพตามจำนวนโนดที่เพิ่มขึ้น

จากรูปที่ 4.19 แสดงเวลาประวิงเฉลี่ยในการรวบรวมข้อมูลจากยานพาหนะที่มีช่วงของค่าความเชื่อมั่นอัตราร้อยละ 95 โดยมีแกน Y เป็นเวลาประวิงเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนโนดบนถนน สังเกตได้ว่า เมื่อจำนวนยานพาหนะในระบบมีค่ามากขึ้น เวลาประวิงเฉลี่ยจะมีค่าสูงขึ้น เนื่องจากเมื่อยานพาหนะได้รับแพ็กเก็ต *polling message* จากโนดข้างถนน จะมีช่วงเวลารอเพื่อรับแพ็กเก็ตข้อมูลจากโนดข้างเคียงก่อน จากนั้นจึงส่งกลับไปยังโนดข้างถนนโดยใช้เทคนิคแบบ piggybacking ทำให้ค่าสูงสุดของการรวบรวมข้อมูลของโนดข้างถนนอยู่ที่ 0.22 วินาที

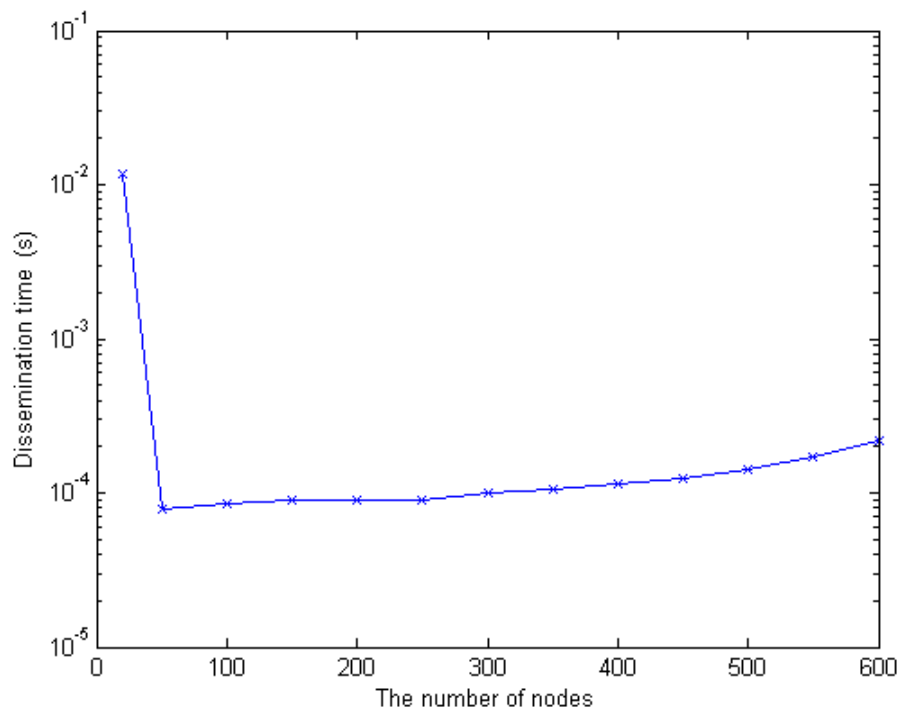
จากรูปที่ 4.20 แสดงเวลาการแจกจ่ายข้อมูล เพื่อต้องการหาจำนวนยานพาหนะที่เหมาะสมที่สุดสำหรับโนดข้างถนนที่ทำหน้าที่รวบรวมข้อมูลจากยานพาหนะ โดยมีแกน Y เป็นเวลาการแจกจ่ายข้อมูล และแกน X เป็นจำนวนโนดบนถนน สังเกตได้ว่า ที่จำนวนโนด 50 คัน จะมีเวลาการแจกจ่ายข้อมูลน้อยที่สุด ซึ่งสามารถอธิบายได้ว่าโพรโทคอลที่ออกแบบจะมีสมรรถนะในการทำงานสูงสุด เมื่อมีจำนวนยานพาหนะบนถนนสาทร 50 คัน



รูปที่ 4.18: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ



รูปที่ 4.19: เวลาประวิงเฉลี่ย



รูปที่ 4.20: เวลาการแจกจ่ายข้อมูล

4.4.4 ผลการจำลองของโพรโทคอลที่ออกแบบในโมดแอคทีเวท

ในส่วนนี้ เป็นการประเมินสมรรถนะของโพรโทคอลที่ออกแบบเปรียบเทียบกับโพรโทคอลแบบดั้งเดิมสำหรับโครงข่ายแอตฮ็อกสำหรับยานพาหนะ ซึ่งโพรโทคอลดังกล่าวยังไม่ได้พิจารณาการแจกจ่ายข้อมูลระหว่างยานพาหนะกับโหนดข้างถนนที่มีข้อมูลที่ต้องการ ทำให้แพ็กเก็ตการร้องขอส่งออกไปรอบทิศทาง ดังนั้นโพรโทคอลที่นำเสนอนี้ มีการกำหนดทิศทางและพื้นที่จำกัดในการร้องขอข้อมูล เราจึงแบ่งผลการจำลองออกเป็น 3 แบบ ดังนี้

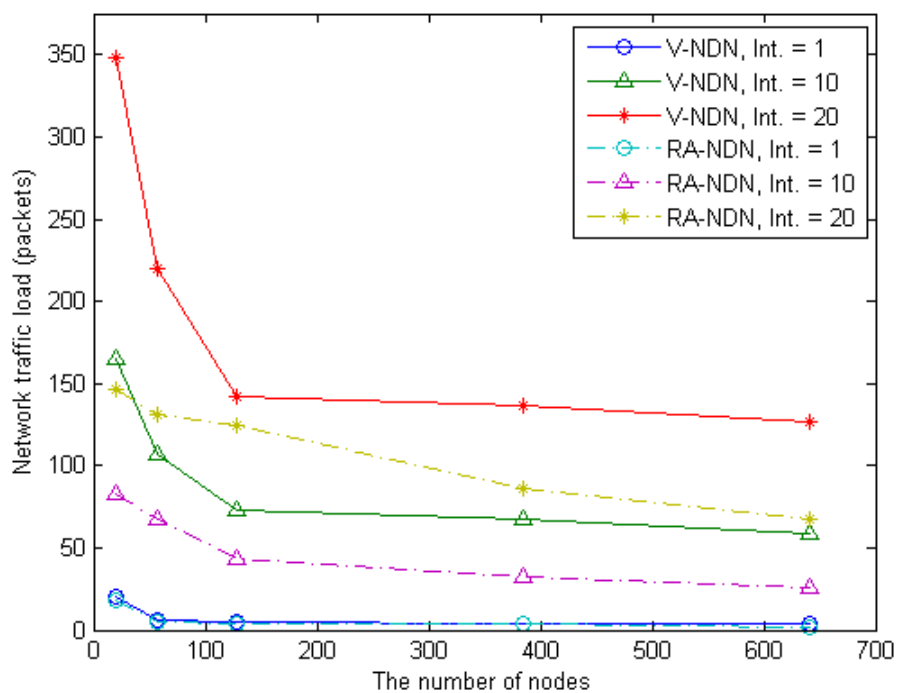
1. ผลการจำลองเมื่อปรับเปลี่ยนจำนวนแพ็กเก็ตการร้องขอ
2. ผลการจำลองเมื่อปรับเปลี่ยนรัศมีการสื่อสาร
3. ผลการจำลองเมื่อปรับขนาดของกลุ่มข้อมูล

4.4.4.1 ผลการจำลองเมื่อปรับเปลี่ยนจำนวนแพ็กเก็ตการร้องขอ

ในส่วนนี้ศึกษาผลการจำลองเมื่อปรับเปลี่ยนจำนวนแพ็กเก็ตการร้องขอ มีค่า 1 10 และ 20 ครั้ง การทดลองนี้ มีการนำข้อมูลจริงจากข้อมูลการจัดการจราจรของถนนสาทร ของวันที่ 7 กรกฎาคม พ.ศ. 2557 ช่วงเวลา 15.30.00 – 15.46.40 น. หรือ 1,000 วินาที โดยมียานพาหนะบนถนนทั้งหมดประมาณ 6,410 คัน แต่ในการทดลองนี้ ได้ลดจำนวนของยานพาหนะเป็นอัตราร้อยละ 0.03-10 ของจำนวนยานพาหนะทั้งหมด หรือมีจำนวนยานพาหนะ 12-641 คัน โดยพารามิเตอร์ที่ใช้ในการจำลองแสดงในตารางที่ 4.6

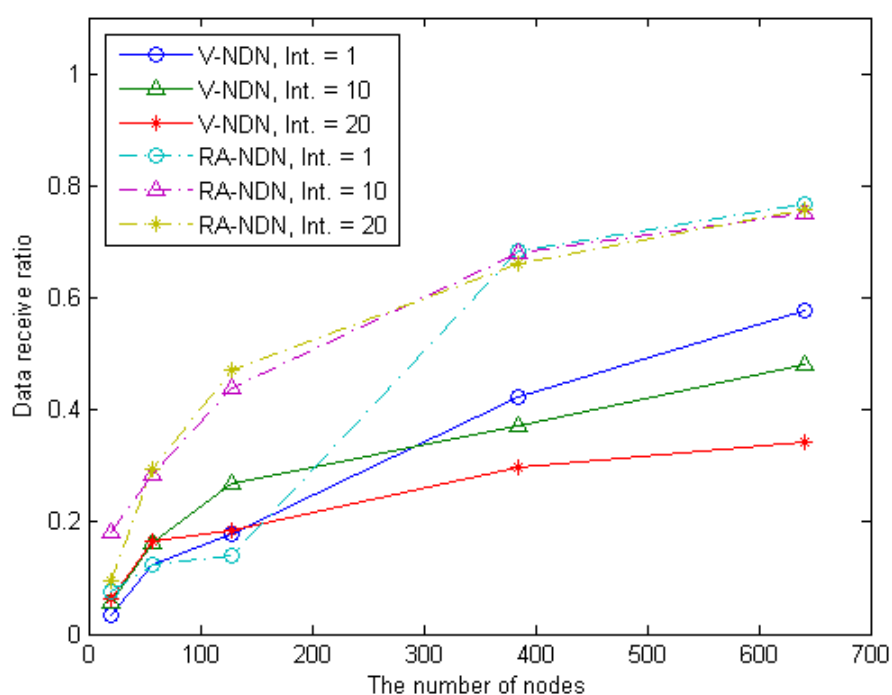
ตารางที่ 4.6: การตั้งค่าของผลการจำลองเมื่อปรับเปลี่ยนจำนวนแพ็กเก็ตการร้องขอ

Parameter	Value
Number of simulations (times)	100
Simulation time (s)	10,000
Physical/MAC protocols	IEEE802.11p
CW Min/Max	31/1023
Data rate(Mb/s)	11
Payload (byte)	1024
Transmission range (m)	200
Carrier frequency (GHz)	5.9
Bit rate (Mbps)	2
Maximum mobile speed (m/s)	1.6
Number of vehicles on road	0.03%-10%
Number of interest	1 10 20



รูปที่ 4.21: ปริมาณกราฟฟิกของโครงข่าย

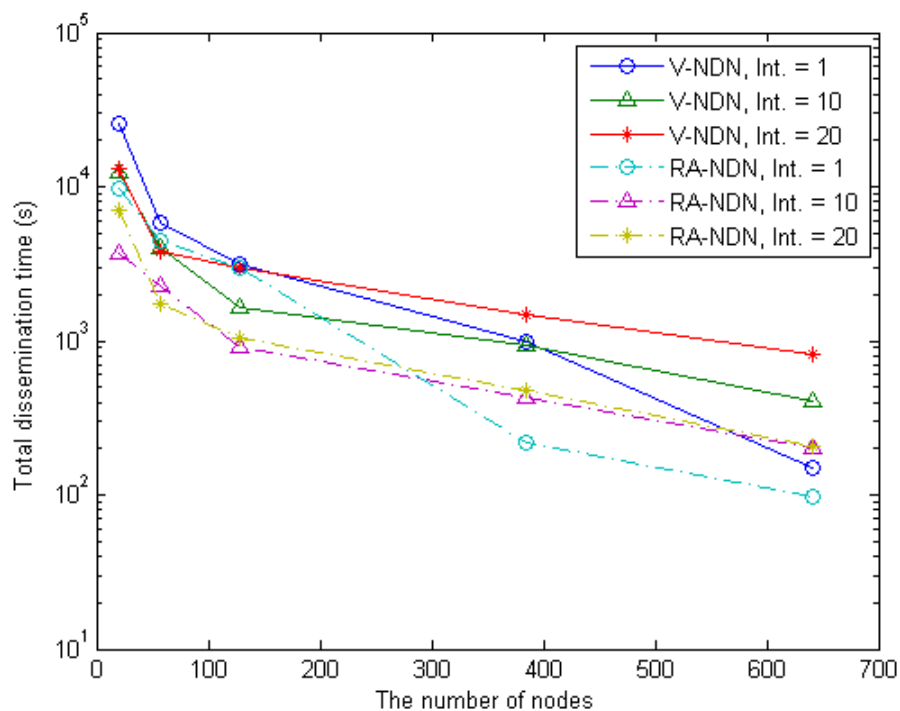
จากรูปที่ 4.21 แสดงปริมาณกราฟฟิกรที่อยู่ในระบบ ที่มีแกน Y เป็นปริมาณกราฟฟิก และแกน X เป็นจำนวนยานพาหนะบนถนน สังเกตได้ว่าปริมาณกราฟฟิกจะมีแนวโน้มลดลง เพราะว่าการเชื่อมต่อมีความต่อเนื่องกัน นอกจากนั้น โพรโทคอลที่นำเสนอ ที่มีการนำโนดข้างถนน (RA-NDN) มาใช้งานร่วม สามารถลดปริมาณกราฟฟิกในการรับส่งข้อมูลได้ประมาณสองเท่าเมื่อเปรียบเทียบกับโครงข่ายชื่อข้อมูล (V-NDN) แบบดั้งเดิม เนื่องจากยานพาหนะที่จะส่งแพ็กเก็ตเกิดการร้องขอสามารถรู้ตำแหน่งของโนดข้างถนน จึงสามารถเลือกส่งต่อแพ็กเก็ตเกิดการร้องขอผ่านโนดระหว่างทางไปยังโนดข้างถนน แสดงให้เห็นว่าปริมาณกราฟฟิกที่ลดลงช่วยเพิ่มควมมีเสถียรภาพในระบบโครงข่ายได้



รูปที่ 4.22: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ

จากรูปที่ 4.22 แสดงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ ที่มีแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X เป็นจำนวนยานพาหนะบนถนน สังเกตได้ว่า โพรโทคอลที่นำเสนอจะมีอัตราส่วนการรับข้อมูลสำเร็จมากกว่า โพรโทคอลแบบ V-NDN เนื่องจากเรากำหนดให้เพิ่มส่วนหัวของแพ็กเก็ตการร้องขอด้วย RSU ID ทำให้รู้ตำแหน่งและทิศทางของแหล่งข้อมูล ทำให้ความถูกต้องในการรับข้อมูลมีความเที่ยงตรงมากขึ้น

จากรูปที่ 4.23 แสดงเวลาการแจกจ่ายข้อมูลโดยรวม ที่มีแกน Y เป็นเวลาการแจกจ่ายข้อมูลโดยรวม และแกน X เป็นจำนวนยานพาหนะบนถนน สังเกตได้ว่า เวลาการแจกจ่ายข้อมูลโดยรวมของโพรโทคอลที่นำเสนอมีค่าต่ำกว่าโพรโทคอลแบบดั้งเดิมประมาณ 2 เท่า เพราะโพรโทคอลที่นำเสนอได้เพิ่มตำแหน่งข้อมูลของโนดข้างถนนในส่วนหัวของแพ็กเก็ตการร้องขอ เพื่อช่วยในการส่งข้อมูล ในขณะที่โพรโทคอลแบบดั้งเดิมจะส่งแพ็กเก็ตการร้องขอไปทุกทิศทางในระบบ โดยโพรโทคอลทั้งสองจะส่งข้อมูลซ้ำเรื่อย ๆ จนกว่าจะประสบความสำเร็จในการ



รูปที่ 4.23: เวลาการแจกจ่ายข้อมูลโดยรวม

รับแพ็กเก็ตข้อมูล

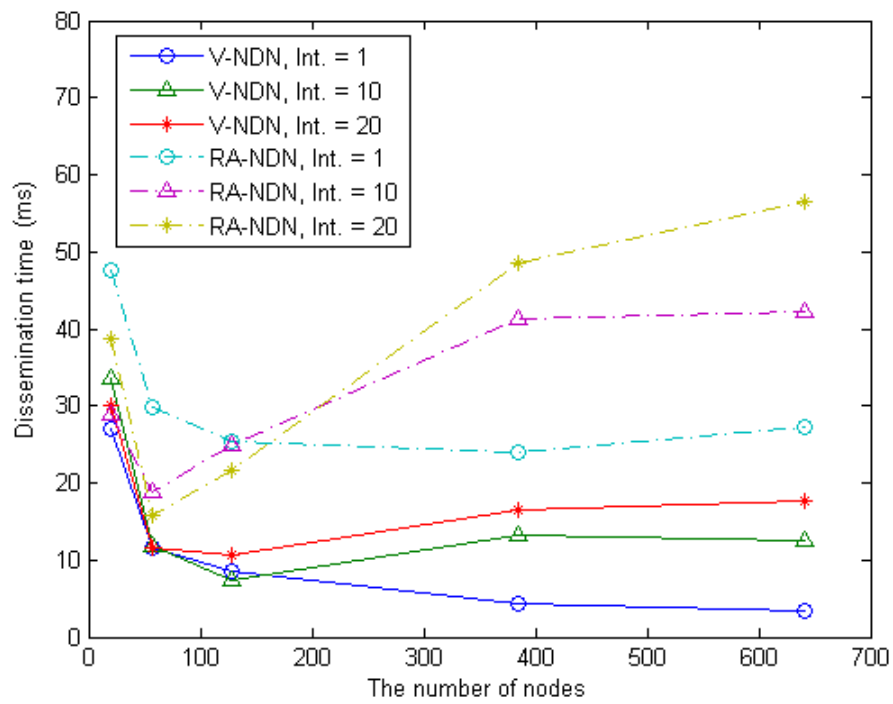
จากรูปที่ 4.24 แสดงเวลาการแจกจ่ายข้อมูล ที่มีแกน Y เป็นเวลาการแจกจ่ายข้อมูล และแกน X เป็นจำนวนยานพาหนะบนถนน จากผลการทดลองที่ได้พบว่าเวลาการแจกจ่ายข้อมูลของ RA-NDN มีค่าประมาณ 50 วินาที เมื่อเทียบกับ V-NDN ที่มีค่า 28 วินาที ภายใต้เงื่อนไขการส่งแพ็กเก็ตการร้องขอ 1 แพ็กเก็ต และมีโหนดเบาบางบนถนน ซึ่งโพรโทคอลแบบ RA-NDN มีการคำนวณหาตำแหน่งก่อนที่จะคำนวณหาเวลาการรอคอยในการส่งต่อข้อมูล จึงทำให้เวลาการแจกจ่ายข้อมูลมีค่าที่สูงกว่าเมื่อเทียบกับ V-NDN ซึ่งจะส่งต่อข้อมูลทันที ทำให้ V-NDN จึงมีเวลาการแจกจ่ายข้อมูลที่ต่ำกว่า

จากรูปที่ 4.25 แสดงค่าวิสัยสามารถ ที่มีแกน Y เป็นค่าวิสัยสามารถ และแกน X เป็นจำนวนยานพาหนะบนถนน พบว่าโพรโทคอลที่ออกแบบมีค่าวิสัยสามารถสูงกว่าโพรโทคอลแบบดั้งเดิม เนื่องจากเวลาการแจกจ่ายข้อมูลโดยรวมต่ำ จึงทำให้ค่าวิสัยสามารถสูง โดยสามารถคำนวณได้ในสมการที่ (4.5) ซึ่งโพรโทคอลที่ออกแบบมีค่าวิสัยสามารถมากกว่าโพรโทคอลแบบดั้งเดิมประมาณ 10 เท่า

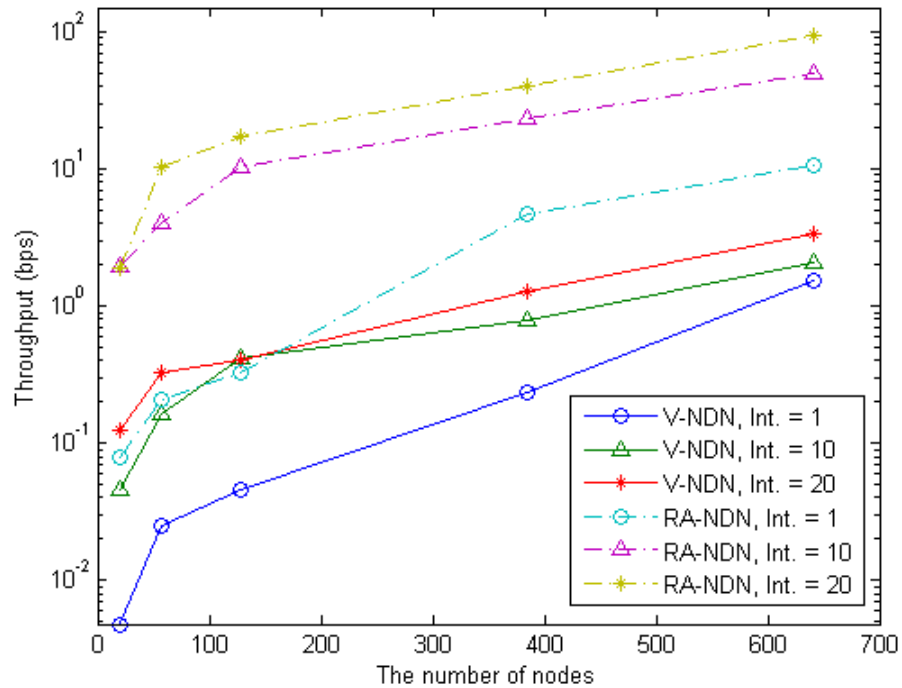
4.4.4.2 ผลการจำลองเมื่อปรับเปลี่ยนรัศมีการสื่อสาร

ในส่วนนี้ มีการจำลองระบบเพื่อศึกษาผลกระทบเมื่อปรับเปลี่ยนรัศมีการสื่อสารที่ระยะ 200 600 และ 1,000 เมตร กำหนดให้มีจำนวนแพ็กเก็ตการร้องขอเป็นอัตราร้อยละ 10 และ 70 ของจำนวนยานพาหนะทั้งหมดบนถนน พารามิเตอร์ที่ใช้ในการทดลองนี้ แสดงในตารางที่ 4.7

จากรูปที่ 4.26 แสดงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ ในกรณีที่ให้มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 โดยแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X



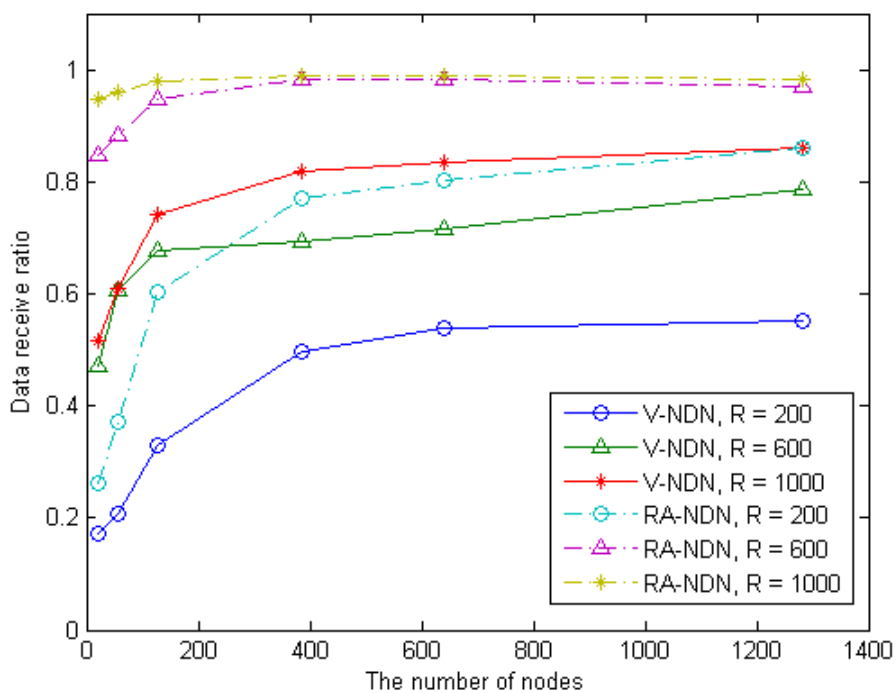
รูปที่ 4.24: เวลาการแจกจ่ายข้อมูล



รูปที่ 4.25: ค่าวิสัยสามารถ

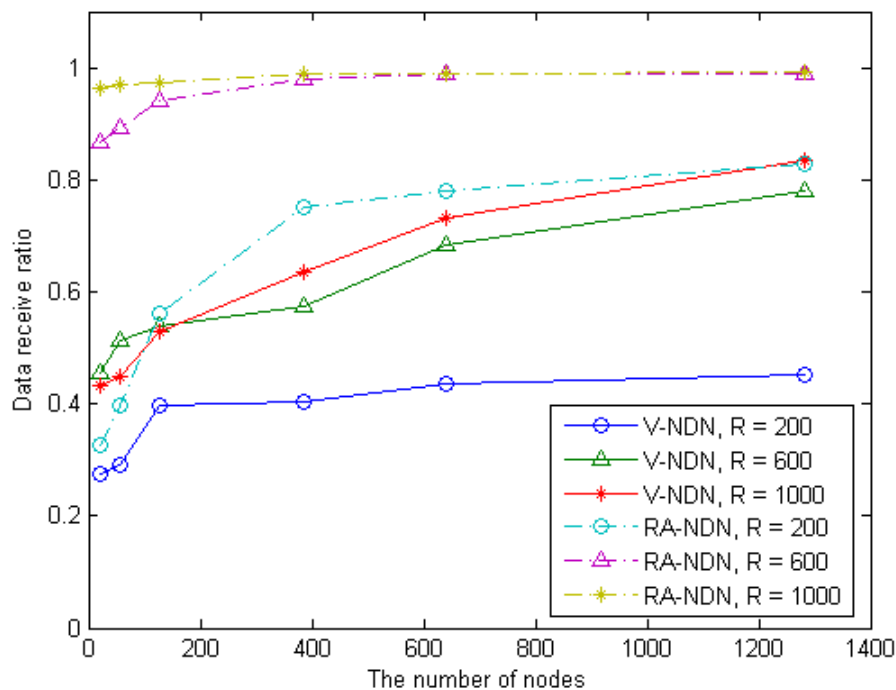
ตารางที่ 4.7: การตั้งค่าของผลการจำลองเมื่อปรับเปลี่ยนรัศมีการสื่อสาร

Parameter	Value
Number of simulations (times)	100
Simulation time (s)	10,000
Physical/MAC protocols	IEEE802.11p
Transmission range (m)	200 600 1000
CW Min/Max	31/1023
Data rate(Mb/s)	11
Payload (byte)	1024
Carrier frequency (GHz)	5.9
Bit rate (MBps)	2
Maximum mobile speed (m/s)	1.6
Number of vehicles on road	0.2%-20%
Number of interest	10% 70%



รูปที่ 4.26: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ กรณีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10

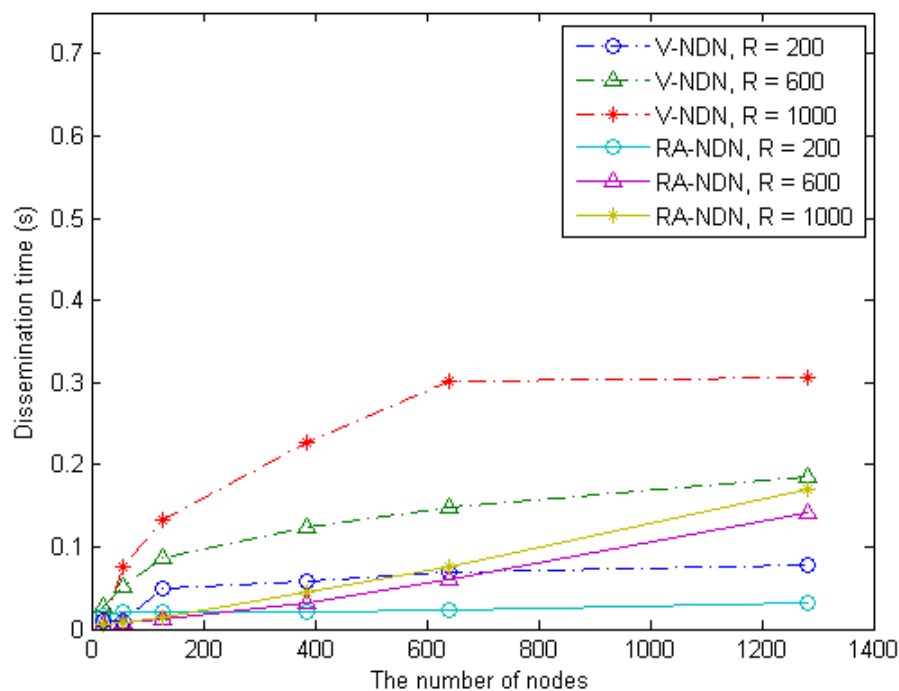
เป็นจำนวนยานพาหนะบนถนน สังเกตได้ว่า เมื่อปรับเปลี่ยนรัศมีการสื่อสาร พบว่า มีอัตราการรับแพ็กเก็ตข้อมูลสำเร็จมากที่สุดไปน้อยที่สุดที่รัศมีการสื่อสารระยะ 200 600 และ 1,000 เมตร ตามลำดับ เนื่องจากเมื่อเพิ่มรัศมีการสื่อสาร ระยะการสื่อสารระหว่างยานพาหนะจะเพิ่มขึ้น ส่งผลให้อัตราการรับแพ็กเก็ตข้อมูลสำเร็จเพิ่มขึ้นด้วย ซึ่งโพรโทคอลที่นำเสนอจะมีอัตราส่วนการรับข้อมูลสำเร็จสูงถึง 100% มากกว่า โพรโทคอลแบบ V-NDN ที่มีอัตราส่วนการรับข้อมูลสำเร็จสูงสุดเท่ากับ 80% เนื่องจากเรากำหนดให้เพิ่มส่วนหัวของแพ็กเก็ตการร้องขอด้วย RSU ID ทำให้รู้ตำแหน่งและทิศทางของแหล่งข้อมูล ความถูกต้องในการรับข้อมูลจึงมีความเที่ยงตรงมากขึ้น



รูปที่ 4.27: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ กรณีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 70

จากรูปที่ 4.27 แสดงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ ในกรณีที่มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 70 โดยแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X เป็นจำนวนยานพาหนะบนถนน สังเกตได้ว่า เมื่อปรับเปลี่ยนรัศมีการสื่อสาร พบว่า มีอัตราการรับแพ็กเก็ตข้อมูลสำเร็จมากที่สุดไปน้อยที่สุดที่รัศมีการสื่อสารระยะ 200 600 และ 1,000 เมตร ตามลำดับ ซึ่งมีแนวโน้มเหมือนกันกับกรณีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 จึงสรุปได้ว่าจำนวนแพ็กเก็ตการร้องขอที่มีมากขึ้นในระบบไม่ส่งผลต่อการทำงานของโพรโทคอล เพราะมีอัตราส่วนการรับข้อมูลสำเร็จสูงถึง 100%

จากรูปที่ 4.28 แสดงเวลาการแจกจ่ายข้อมูล ในกรณีที่มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 โดยแกน Y เป็นเวลาการแจกจ่ายข้อมูล และแกน X เป็นค่าจำนวนยานพาหนะบนถนน จากผลการทดลองที่ได้พบว่า เวลาการแจกจ่ายข้อมูลของ RA-NDN สูงกว่า V-NDN เมื่อเทียบที่รัศมีการสื่อสารเดียวกันที่ระยะ 200 600 และ 1,000 เมตร ตามลำดับ เนื่องจากโพรโทคอลแบบ RA-NDN มีการคำนวณหาระยะตำแหน่งก่อนที่จะคำนวณหาเวลาการ



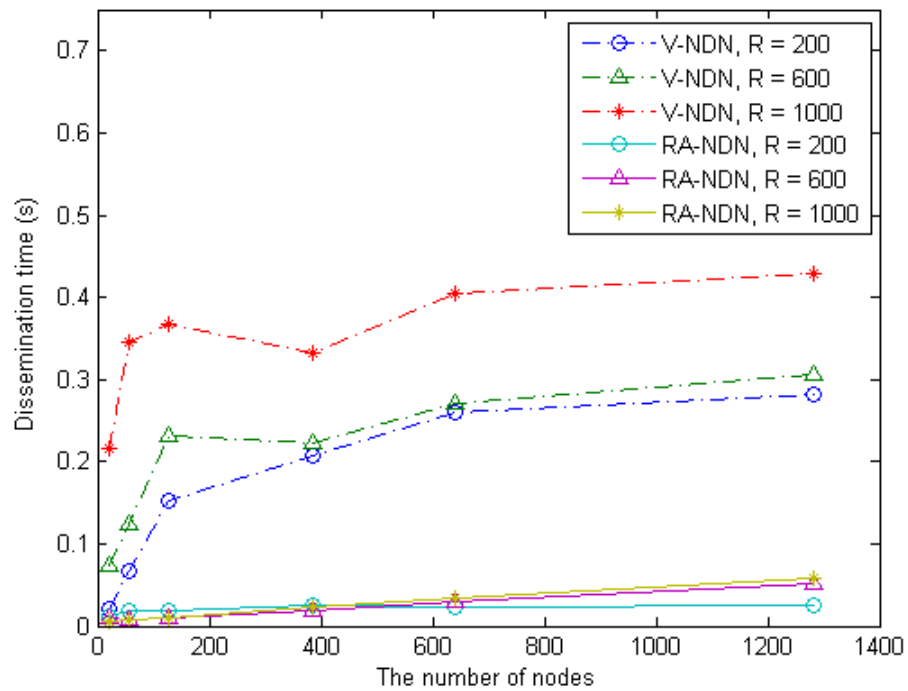
รูปที่ 4.28: เวลาการแจกจ่ายข้อมูล กรณีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10

รอดอกในการส่งต่อข้อมูล แต่โพรโทคอลแบบ V-NDN จะส่งต่อข้อมูลทันที ดังนั้น V-NDN จึงมีค่าเวลาการแจกจ่ายข้อมูลต่ำกว่า

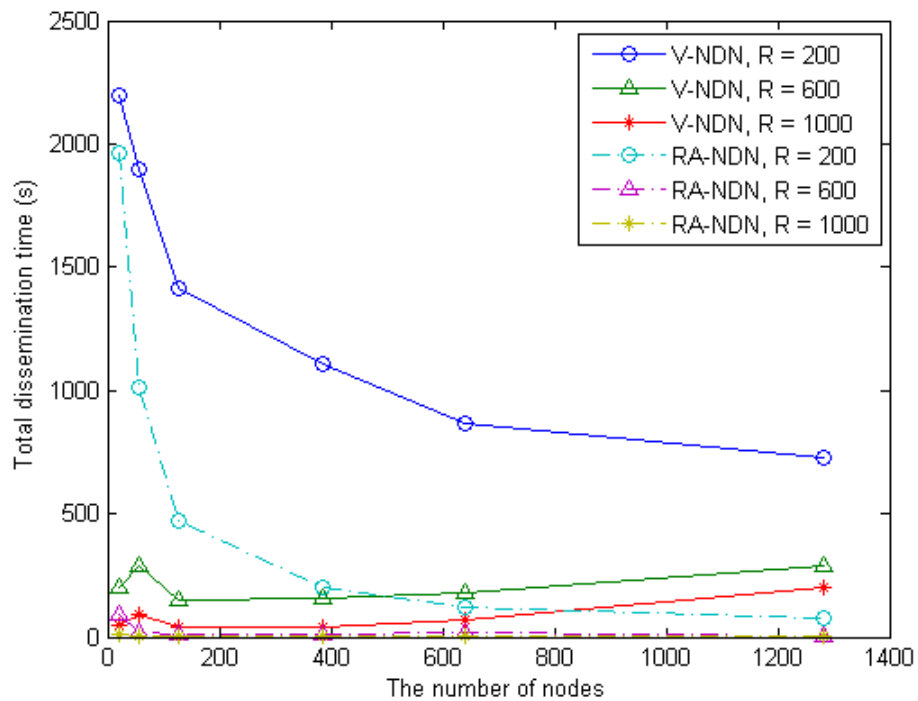
เราสามารถสังเกตได้ว่าเวลาการแจกจ่ายข้อมูล จากรูปที่ 4.28 มีค่ามากที่สุดที่รัศมีการสื่อสารระยะ 1,000 เมตร เนื่องจากที่รัศมีการสื่อสารระยะ 1,000 เมตร จะมีจำนวนโหนดในระยะเวลาการส่งต่อมาก ทำให้แพ็กเก็ตมีโอกาสชนกันสูงขึ้น ดังนั้นเวลาการแจกจ่ายข้อมูลจึงมีค่ามากที่สุดที่รัศมีการสื่อสารระยะ 1,000 เมตร

จากรูปที่ 4.29 แสดงเวลาการแจกจ่ายข้อมูล ในกรณีที่มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 70 โดยแกน Y เป็นเวลาการแจกจ่ายข้อมูล และแกน X เป็นค่าจำนวนยานพาหนะบนถนน จากผลการทดลองที่ได้พบว่า เวลาการแจกจ่ายข้อมูลของ RA-NDN สูงกว่า V-NDN เมื่อเทียบที่รัศมีการสื่อสารเดียวกันที่ระยะ 200 600 และ 1,000 เมตร ตามลำดับ ซึ่งมีแนวโน้มเหมือนกันกับกรณีที่มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 แต่จะมีเวลาประวิงข้อมูลล่าสุดสูงกว่ากรณีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 เพราะเมื่อมีจำนวนแพ็กเก็ตการร้องขอมากขึ้น ระบบจะมีโอกาสชนกันของข้อมูลมากขึ้น

จากรูปที่ 4.30 แสดงเวลาการแจกจ่ายข้อมูลโดยรวม ในกรณีที่มีจำนวนแพ็กเก็ตการร้องขออัตราร้อยละ 10 โดยแกน Y เป็นเวลาการแจกจ่ายข้อมูลโดยรวม และแกน X เป็นจำนวนยานพาหนะบนถนน จากผลการทดลองที่ได้พบว่า เวลาการแจกจ่ายข้อมูลโดยรวมของโพรโทคอลที่นำเสนอต่ำกว่าโพรโทคอลแบบดั้งเดิม เมื่อเทียบที่รัศมีการสื่อสารเดียวกันที่ระยะ 200 600 และ 1000 เมตร ตามลำดับ เนื่องจากเมื่อเพิ่มรัศมีการสื่อสาร ระยะการสื่อสารระหว่างยานพาหนะจะเพิ่มขึ้น จึงทำให้โหนดผู้ร้องขอมีโอกาสเจอโหนดที่มีแหล่งข้อมูลได้เร็วขึ้น

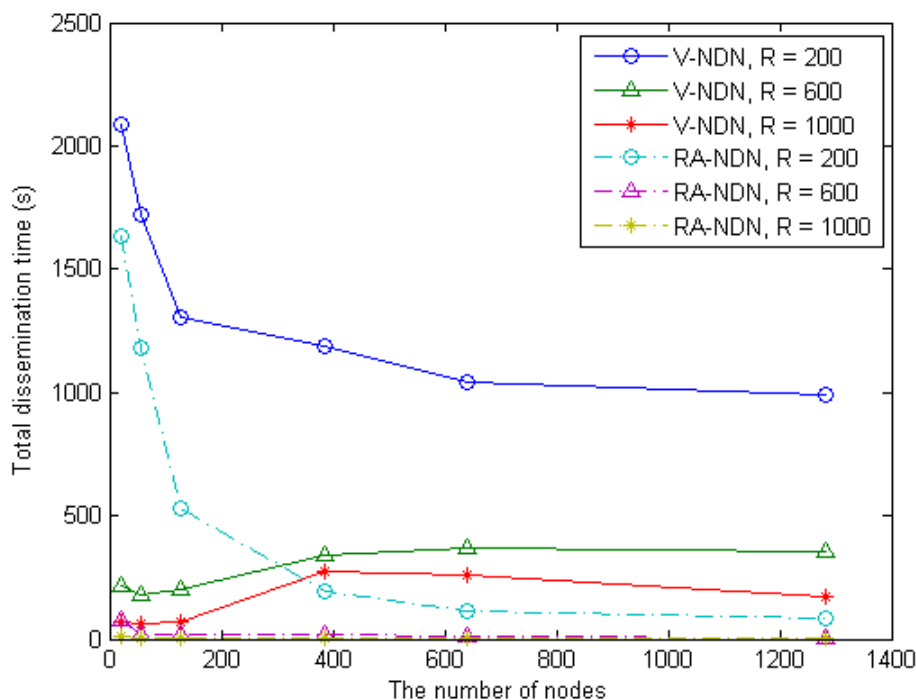


รูปที่ 4.29: เวลาการแจกจ่ายข้อมูล กรณีจำนวนแพ็กเก็ตเกิดการร้องขออัตราร้อยละ 70



รูปที่ 4.30: เวลาการแจกจ่ายข้อมูลโดยรวม กรณีจำนวนแพ็กเก็ตเกิดการร้องขออัตราร้อยละ 10

เราสามารถสังเกตได้ว่าเวลาการแจกจ่ายข้อมูลโดยรวม จากรูปที่ 4.30 มีค่ามากที่สุดที่รัศมีการสื่อสารระยะ 200 เมตร เนื่องจากที่รัศมีการสื่อสารระยะ 200 เมตร มีจำนวนช่วงเชื่อมต่อมาก จึงทำให้โนดระหว่างทางส่งแพ็กเกตมากขึ้น ส่งผลให้เกิดการชนกันของแพ็กเกตเพิ่มขึ้น จึงเกิดการส่งแพ็กเกตซ้ำเพิ่มมากขึ้นในระบบ ดังนั้นเวลาการแจกจ่ายข้อมูลโดยรวมจึงมีค่ามากที่สุดที่รัศมีการสื่อสารระยะ 200 เมตร



รูปที่ 4.31: เวลาการแจกจ่ายข้อมูลโดยรวม กรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 70

จากรูปที่ 4.31 แสดงเวลาการแจกจ่ายข้อมูลโดยรวม ในกรณีที่ที่มีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 70 โดยแกน Y เป็นเวลาการแจกจ่ายข้อมูลโดยรวม และแกน X เป็นจำนวนยานพาหนะบนถนน พบว่า เวลาการแจกจ่ายข้อมูลโดยรวมของโพรโทคอลที่นำเสนอมีค่าต่ำกว่าโพรโทคอลแบบดั้งเดิม ซึ่งมีแนวโน้มเหมือนกันกับกรณีจำนวนแพ็กเกตการร้องขออัตราร้อยละ 10

4.4.4.3 ผลการจำลองเมื่อปรับขนาดของกลุ่มข้อมูล

ในส่วนนี้ เป็นการจำลองระบบเพื่อศึกษาผลกระทบเมื่อมีการปรับเปลี่ยนขนาดของกลุ่มข้อมูล โดยแบ่งเป็น 4 กลุ่มข้อมูล ได้แก่ 256 512 1024 และ 2048 ไบต์ (byte) เปรียบเทียบโพรโทคอล RA-NDN กับ V-NDN และกำหนดให้มีจำนวนแพ็กเกตการร้องขอเป็นอัตราร้อยละ 50 ของจำนวนยานพาหนะทั้งหมดบนถนน ที่มีจำนวน 128 และ 1282 ไบต์ ซึ่งพารามิเตอร์ที่ใช้ในการทดลองนี้แสดงในตารางที่ 4.8

ผลการจำลองและการวิเคราะห์ พารามิเตอร์ที่ใช้เป็นตัวชี้วัดสมรรถนะในการทดลองนี้ ประกอบไปด้วย

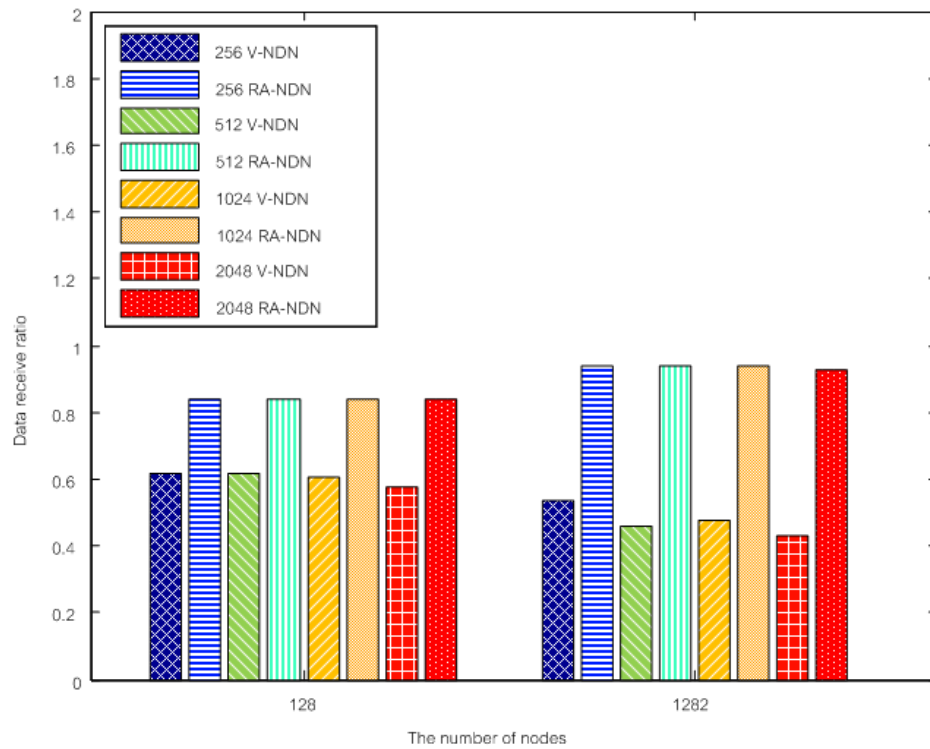
1. อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ (Data Received Ratio หรือ *DRR*) คือ จำนวนแพ็กเก็ตข้อมูลที่โนดทั้งหมดได้รับสำเร็จในระบบต่อจำนวนแพ็กเก็ตที่โนดทั้งหมดส่งสำเร็จในระบบ
2. เวลาประวิงข้อมูลเฉลี่ย (Average delay time) คือ เวลาเฉลี่ยที่โนดผู้ร้องขอรับแพ็กเก็ตข้อมูลจากโนดแหล่งข้อมูล มีหน่วยเป็น วินาที
3. จำนวนการส่งต่อข้อมูลซ้ำ (The number of retransmission) คือ จำนวนการส่งแพ็กเก็ตการร้องขอซ้ำ

ตารางที่ 4.8: การตั้งค่าของผลการจำลองเมื่อปรับขนาดของกลุ่มข้อมูล

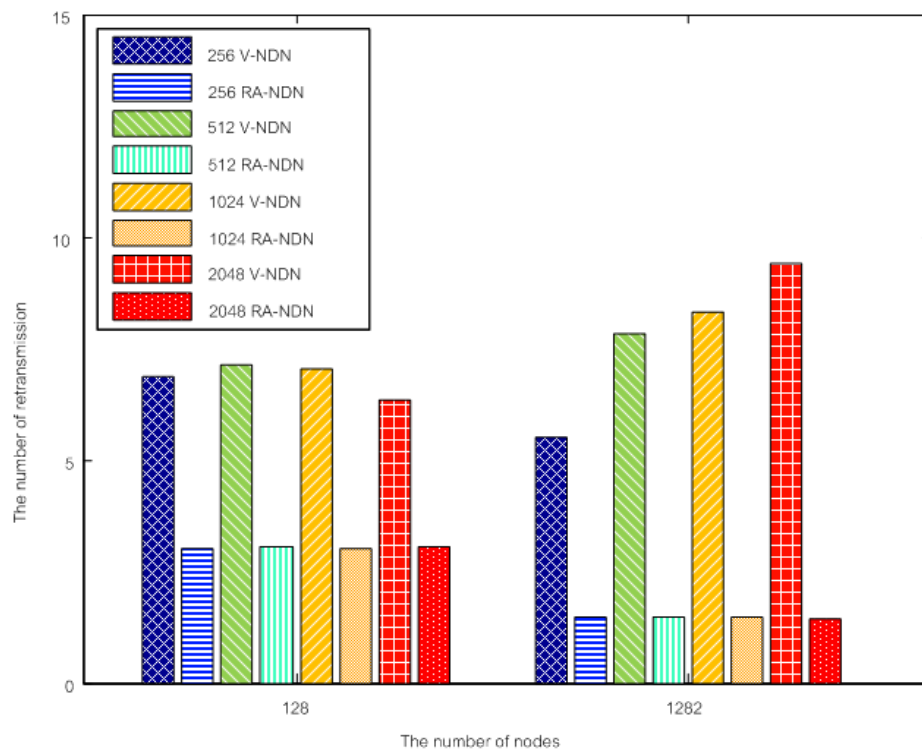
Parameter	Value
Number of simulations (times)	100
Simulation time (s)	1000
MAC protocol	IEEE802.11p
Payload (bytes)	256 512 1024 2048
Transmission range (m)	600
Carrier frequency (GHz)	5.9
Bit rate (Mbps)	2
Maximum mobile speed (m/s)	1.6
Number of vehicles on road	128 1282
Number of interest	50%

จากรูปที่ 4.32 แสดงอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ ของโพรโทคอล V-NDN และ RA-NDN โดยแกน Y เป็นอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ และแกน X เป็นจำนวนโหนดบนถนน สังเกตได้ว่า โพรโทคอล V-NDN มีอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จอยู่ที่ 60% ขณะที่โพรโทคอล RA-NDN มีอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จอยู่ที่ 100% ทำให้ระบบมีความน่าเชื่อถือและเพิ่มสมรรถนะของโพรโทคอล อีกทั้งเมื่อพิจารณาการเพิ่มขนาดของกลุ่มข้อมูล พบว่าอัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จมีค่าลดลงน้อยมาก แนวโน้มการเพิ่มจำนวนกลุ่มข้อมูลจึงมีค่าไม่แตกต่างกัน

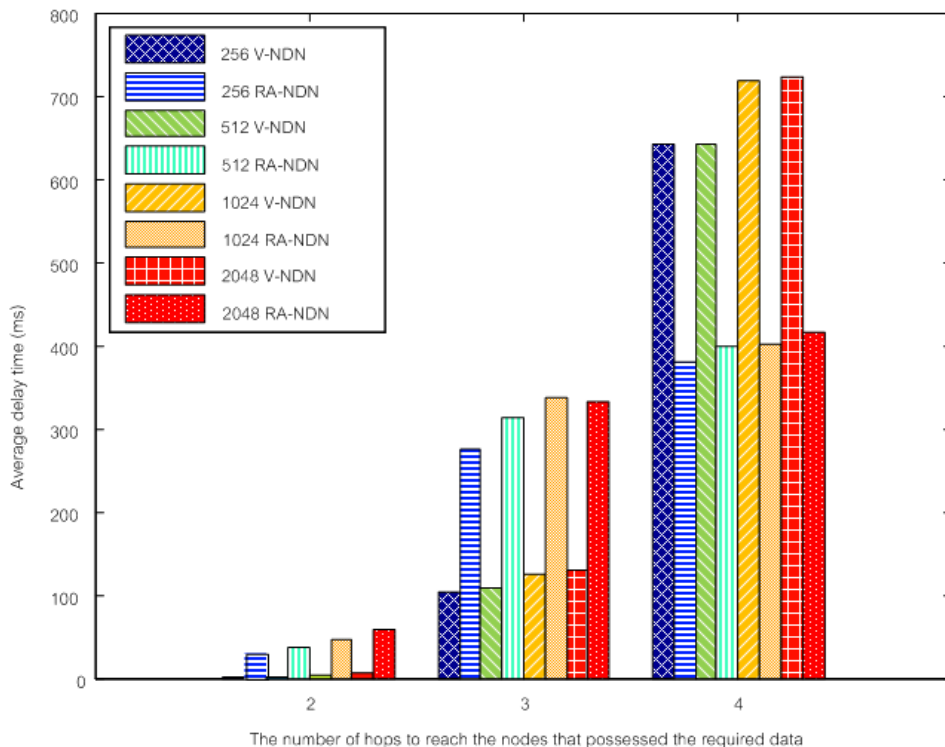
ในรูปที่ 4.33 แสดงจำนวนการส่งต่อข้อมูลซ้ำ ของโพรโทคอล V-NDN และ RA-NDN โดย แกน Y เป็นจำนวนการส่งต่อข้อมูลซ้ำ และแกน X เป็นจำนวนโหนดบนถนน สังเกตได้ว่า โพรโทคอลมีการส่งแพ็กเก็ตการร้องขอซ้ำเพิ่มขึ้นเมื่อขนาดของกลุ่มข้อมูลมีค่ามากขึ้น เพราะโอกาสที่แพ็กเก็ตจะเกิดการชนกันของข้อมูลขนาดใหญ่มีค่าสูงกว่าข้อมูลขนาดเล็ก ทำให้โนดผู้ร้องขอไม่ได้รับแพ็กเก็ตข้อมูล โหนดจึงจำเป็นต้องส่งแพ็กเก็ตการร้องขอซ้ำ ๆ ส่งผลให้จำนวนการส่งต่อข้อมูลซ้ำมีค่ามากขึ้น ซึ่งจะเห็นได้ว่าโพรโทคอล RA-NDN มีจำนวนการส่งต่อข้อมูลซ้ำน้อยกว่าประมาณ 2 เท่า เมื่อเทียบกับ V-NDN ทำให้สามารถใช้ช่องสัญญาณได้เต็มประสิทธิภาพและสามารถลดจำนวนแพ็กเก็ตการร้องขอที่เกิดขึ้นในระบบ



รูปที่ 4.32: อัตราส่วนการรับแพ็กเก็ตข้อมูลสำเร็จ กรณีความหนาแน่นจำนวน 128 และ 1282 โหนด



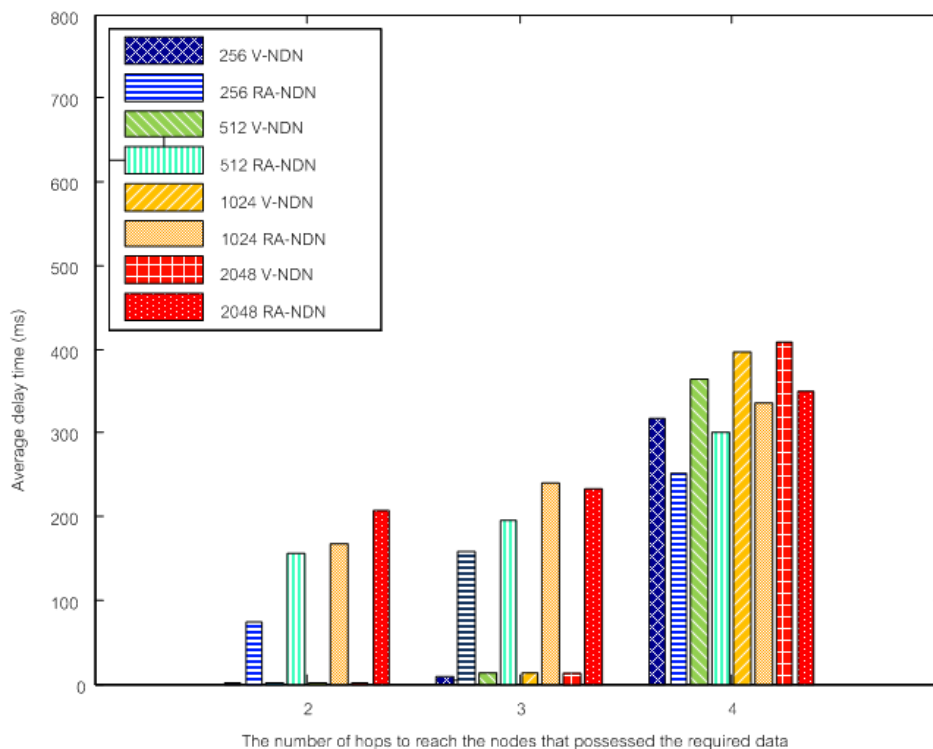
รูปที่ 4.33: จำนวนของการส่งต่อข้อมูลซ้ำ กรณีความหนาแน่นจำนวน 128 และ 1282 โหนด



รูปที่ 4.34: เวลาประวิงข้อมูลเฉลี่ย ความหนาแน่นจำนวน 128 โหนด

ในรูปที่ 4.34 แสดงเวลาประวิงข้อมูลเฉลี่ย จำนวน 128 โหนด ของโพรโทคอล V-NDN และ RA-NDN โดยแกน Y เป็นเวลาประวิงข้อมูลเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนช่วงเชื่อมต่อ มีค่า 2 3 และ 4 ฮอป สังเกตได้ว่า เมื่อจำนวนช่วงเชื่อมต่อมากขึ้น โพรโทคอล V-NDN มีเวลาประวิงข้อมูลเฉลี่ยสูงขึ้นแบบ exponential เนื่องจากจำนวนโหนดที่ร้องขอข้อมูลตามจำนวนช่วงเชื่อมต่อมากขึ้น การเพิ่มจำนวนการติดต่อจึงมากขึ้นตามไปด้วย ทำให้เกิดการชนกันในระดับ MAC สูงขึ้น อีกทั้งแพ็กเก็ตที่ชนกันจะต้องรอเวลาการส่งต่อข้อมูลออกไปแบบ exponential เวลาประวิงข้อมูลเฉลี่ยจึงมีแนวโน้มแบบ exponential ในขณะที่โพรโทคอล RA-NDN มีเวลาประวิงข้อมูลเฉลี่ยมากกว่าโพรโทคอล V-NDN เมื่อจำนวนช่วงเชื่อมต่อมีค่า 2 และ 3 ฮอป แต่ที่จำนวนช่วงเชื่อมต่อ 4 ฮอป เวลาประวิงข้อมูลเฉลี่ยจะมีค่าน้อยกว่าโพรโทคอล V-NDN ซึ่งการเพิ่มขนาดของกลุ่มข้อมูลจะทำให้เวลาประวิงข้อมูลเฉลี่ยมีแนวโน้มเพิ่มขึ้น เนื่องจากขนาดแพ็กเก็ตที่มีขนาดใหญ่จะส่งผลให้ประวิงข้อมูลเฉลี่ยมีค่าสูงขึ้น

ในรูปที่ 4.35 แสดงเวลาประวิงข้อมูลเฉลี่ย จำนวน 1282 โหนด ของโพรโทคอล V-NDN และ RA-NDN โดยแกน Y เป็นเวลาประวิงข้อมูลเฉลี่ยในการส่งข้อมูล และแกน X เป็นจำนวนช่วงเชื่อมต่อ มีค่า 2 3 และ 4 ฮอป สังเกตได้ว่า เมื่อจำนวนโหนดในถนนเพิ่มขึ้น เวลาประวิงเฉลี่ยมีค่าเพิ่มขึ้น ในขณะที่โพรโทคอล RA-NDN มีเวลาประวิงข้อมูลเฉลี่ยมากกว่าโพรโทคอล V-NDN เมื่อจำนวนช่วงเชื่อมต่อมีค่า 2 และ 3 ฮอป แต่ที่จำนวนช่วงเชื่อมต่อ 4 ฮอป เวลาประวิงข้อมูลเฉลี่ยจะมีใกล้เคียงกับโพรโทคอล V-NDN และเมื่อปรับเปลี่ยนขนาดของกลุ่มข้อมูล แนวโน้มของเวลาประวิงข้อมูลเฉลี่ยเหมือนกันกับกรณีของเวลาประวิงข้อมูลเฉลี่ยที่จำนวน 128 โหนด ตามที่ได้อธิบายไปข้างต้น



รูปที่ 4.35: เวลาประวิงข้อมูลเฉลี่ย ความหนาแน่นจำนวน 1282 โหนด

4.5 สรุป

จากผลการจำลองข้างต้น สามารถสรุปตามกรณี ความหนาแน่น ช่วงเชื่อมต่อ จำนวนการร้องขอ และรัศมีการสื่อสาร ได้ดังตารางที่ 4.9 ซึ่งโพรโทคอลการแจกจ่ายข้อมูลที่น่าเสนอโดยใช้โหนดข้างถนน ให้อัตราการรับข้อมูลสำเร็จมากกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม ปริมาณกราฟิกในระบบมีค่าลดลง และมีระยะเวลาการแจกจ่ายข้อมูลน้อยกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม อีกทั้งค่าวิสัยสามารถยังมีสมรรถนะที่ดีกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม แม้ว่าเวลาการแจกจ่ายข้อมูลของโพรโทคอลที่น่าเสนอมากกว่าโพรโทคอลชื่อแบบดั้งเดิม

ตารางที่ 4.9: ตารางเปรียบเทียบสมรรถนะของโพรโทคอล

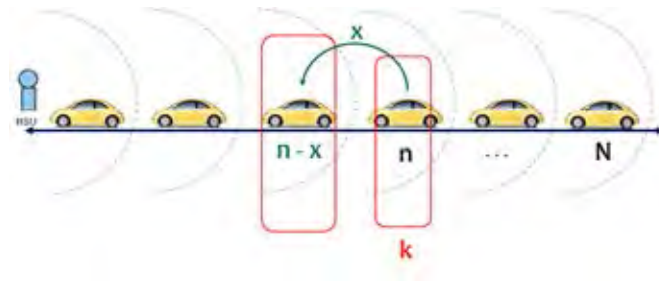
พารามิเตอร์	ค่า	สมรรถนะ		เวลาการแจกจ่ายข้อมูลโดยรวม(วินาที)		เวลาการแจกจ่ายข้อมูล(วินาที)	
		V-NDN	RA-NDN	V-NDN	RA-NDN	V-NDN	RA-NDN
ความหนาแน่น	ต่ำ	5-25%	20-40%	12566-1654	3682-921	0.0018-0.0019	0.0052-0.0109
	สูง	40-55%	60-80%	944-403	432-203	0.0048-0.0060	0.0280-0.0317
ช่วงเชื่อมต่อ	สั้น	80%	90%	1.0125	73.8403	0.0125	29.6866
	ยาว	60%	85%	641.6029	378.6426	7.0810	13.4422
จำนวนการร้องขอ	ต่ำ	18-60%	25-98%	25427-149	9718-97	0.0009-0.0019	0.0040-0.0444
	สูง	45-80%	78-100%	13127-827	6979-205	0.0018-0.0060	0.0036-0.0427
รัศมีการสื่อสาร	สั้น	18-55%	22-80%	2197-730	1964-77	0.0109-0.0786	0.0167-0.0312
	ยาว	45-80%	88-100%	49-200	10-0.0143	0.0054-0.1689	0.0056-0.3064

บทที่ 5

การวิเคราะห์เชิงคณิตศาสตร์

ในบทนี้เป็นการวิเคราะห์เชิงคณิตศาสตร์เพื่อหาความน่าจะเป็นที่ได้รับแพ็กเก็ตข้อมูลสำเร็จ หลังจากส่งแพ็กเก็ตการร้องขอข้อมูลระหว่างโนดผู้ร้องขอและโนดแหล่งข้อมูล ดังแสดงในรูปที่ 5.1 โดยในโครงสร้างระบบที่พิจารณาระเบียงชนิดของโนดได้ 3 ชนิด ดังนี้

1. โหนดผู้ร้องขอ คือ โหนดที่ส่งแพ็กเก็ตการร้องขอ
2. โหนดแหล่งข้อมูล (โนดผู้ให้บริการ) คือ โหนดที่มีข้อมูลที่โนดผู้ร้องขอต้องการ
3. โหนดระหว่างทาง (โนดส่งต่อข้อมูล) คือ โหนดที่อยู่ระหว่างทาง ซึ่งทำหน้าที่ส่งต่อข้อมูล และสามารถทำหน้าที่เป็นโนดแหล่งข้อมูลได้ โดยการเก็บข้อมูลจากโนดข้างถนน และโนดระหว่างทางอื่น ๆ



รูปที่ 5.1: โครงสร้างระบบที่พิจารณา

ในโครงสร้างระบบที่พิจารณากำหนดให้

- r คือ โหนดผู้ร้องขอ
- p คือ โหนดผู้ให้บริการ
- D_{rp} คือ ระยะทางระหว่างโนดผู้ร้องขอ และโนดผู้ให้บริการ
- R คือ รัศมีการสื่อสารของยานพาหนะ
- N คือ ระยะห่างสูงสุดระหว่างโนดข้างถนนกับโนดผู้ร้องขอเป็นจำนวนเท่าของรัศมีการสื่อสาร
- n คือ ระยะห่างระหว่างโนดข้างถนนกับโนดผู้ร้องขอเป็นจำนวนเท่าของรัศมีการสื่อสาร
- k คือ ครั้งที่ร้องขอ
- x คือ ระยะห่างระหว่างโนดผู้ร้องขอกับโนดที่มีแพ็กเก็ตข้อมูลเป็นจำนวนเท่าของรัศมีการสื่อสาร

กำหนดสมมุติฐาน ให้

- โหนดข้างถนน 1 โหนดอยู่ขอบของถนน
- โหนดผู้ร้องขอ จะไม่ร้องขอข้อมูลเดิมซ้ำ
- โหนดแต่ละโหนดไม่ส่งข้อมูลพร้อมกัน
- โหนดที่เคยร้องขอข้อมูลและโหนดที่อยู่ในเส้นทางการร้องขอ จะมีการเก็บข้อมูลไว้
- โหนดทุกโหนดมีโอกาสในการร้องขอเท่ากัน

โดยหากโหนดผู้ร้องขอและโหนดผู้ให้บริการไม่สามารถติดต่อสื่อสารกันได้โดยตรง แพ็กเก็ตการร้องขอ และแพ็กเก็ตข้อมูลจะถูกส่งผ่านไปยังโหนดระหว่างทาง ดังนั้นระยะทางสูงสุดระหว่างโหนดผู้ร้องขอ ไปยังโหนดผู้ให้บริการจะได้ D_{rp} เท่ากับ $i \times R$ โดยที่ i เป็นจำนวนฮอปน้อยที่สุดที่แพ็กเก็ตข้อมูลจะถูกส่งผ่านโหนดระหว่างทาง เรากำหนดให้ $i \leq N$ โดยที่ N เป็นจำนวนฮอปสูงสุดที่โหนดข้างถนนสามารถส่งแพ็กเก็ตข้อมูลไปยังโหนดผู้ร้องขอ

เราจะพบแพ็กเก็ตข้อมูลได้ที่โหนดระหว่างทาง หากโหนดผู้ร้องขออื่น ๆ เคยร้องขอแพ็กเก็ตข้อมูลเดียวกันจากโหนดข้างถนนมาก่อน โดยเราสามารถเรียกโหนดระหว่างทางได้อีกอย่างว่า โหนดส่งต่อ ซึ่งจะทำหน้าที่แพร่กระจายแพ็กเก็ตการร้องขอไปยังโหนดผู้ให้บริการ และส่งแพ็กเก็ตข้อมูลกลับมายังโหนดผู้ร้องขอ

กำหนดให้ระยะทางระหว่างตำแหน่งของโหนดผู้ร้องขอไปยังโหนดข้างถนนเท่ากับ $n \times R$ และให้ระยะทางจากตำแหน่งของโหนดผู้ร้องขอไปยังตำแหน่งของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูล เท่ากับ $x \times R$

ในการหาเหตุการณ์ที่โหนดผู้ร้องขอจะได้รับแพ็กเก็ตข้อมูลครั้งที่ร้องขอ k นั้น เราต้องพิจารณาครั้งที่ร้องขอ $k - 1$ และ $k - 2$ ด้วย เนื่องจากการที่โหนดผู้ร้องขอจะได้รับแพ็กเก็ตข้อมูลนั้น โหนดร้องขออื่น ๆ จะต้องเคยร้องขอข้อมูลก่อนหน้านั้นแล้ว จึงต้องพิจารณการร้องขอในอดีต เพื่อหาตำแหน่งของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูลนั้น เมื่อเราพิจารณาที่โหนดผู้ร้องขอส่งแพ็กเก็ตการร้องขอ และได้รับแพ็กเก็ตข้อมูลตอบกลับมา ในครั้งที่ร้องขอ $k - 1$ จะได้ระยะทางจากตำแหน่งของโหนดผู้ร้องขอไปยังตำแหน่งของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูลเท่ากับ $(n - x) \times R$ และในครั้งที่ร้องขอ $k - 2$ จะได้ระยะทางจากตำแหน่งของโหนดผู้ร้องขอไปยังตำแหน่งของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูลเท่ากับ $(n - x - 1) \times R$

ดังนั้นเราสามารถหาเหตุการณ์ที่โหนดผู้ร้องขอจะได้รับแพ็กเก็ตข้อมูลครั้งที่ร้องขอ k ซึ่งมีระยะทางที่อยู่ไม่ไกลเกิน $(n - x - 1) \times R$ ได้เท่ากับ $\binom{n - x - 1}{k - 2} (k - 2)!$ และเมื่อพิจารณาโหนดที่เหลือในระบบทั้งหมดจำนวน $N - k$ โหนด จะได้เหตุการณ์ที่โหนดผู้ร้องขอจะได้รับแพ็กเก็ตข้อมูลครั้งที่ร้องขอ k คือ

$$E_1(x, N, n, k) \text{ เท่ากับ } (k - 1) \binom{n - x - 1}{k - 2} (k - 2)! (N - k)!$$

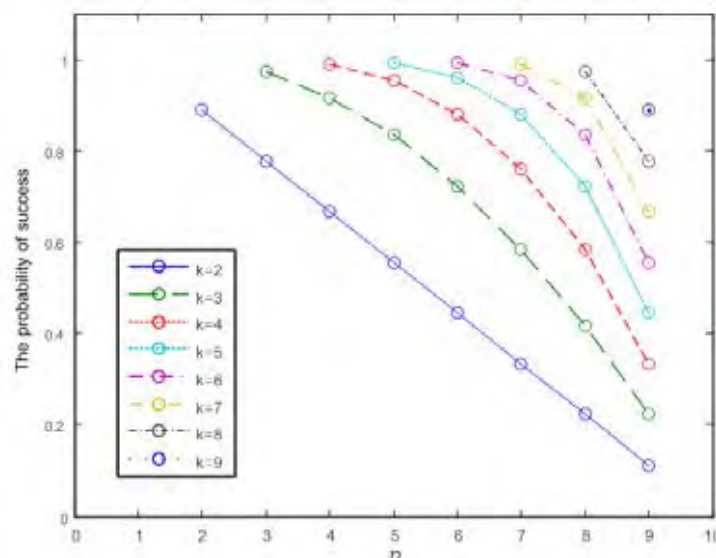
โดยที่ $E_1(x, N, n, k)$ คือ เหตุการณ์ที่ตำแหน่ง n ของโหนดผู้ร้องขอ จะได้รับแพ็กเก็ตข้อมูลจากตำแหน่ง x ของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูล โดยมีจำนวนฮอปสูงสุด N ที่โหนดข้างถนนสามารถส่งแพ็กเก็ตข้อมูลไปยังโหนดผู้ร้องขอ และครั้งที่ร้องขอ k

กำหนดให้ $E_2(N, n, k)$ เป็นเหตุการณ์ทั้งหมดที่ตำแหน่ง n ของโนดผู้ร้องขอ จะได้รับแพ็กเก็ตข้อมูล โดยมีจำนวนฮอปสูงสุด N ที่โนดข้างถนนสามารถส่งแพ็กเก็ตข้อมูลไปยังโนดผู้ร้องขอ ครั้งที่ร้องขอ k เท่ากับ $(N - 1)!$

ดังนั้นเราจะได้ความน่าจะเป็นที่จะได้รับแพ็กเก็ตข้อมูลที่ระยะทางระหว่างโนดผู้ร้องขอ และตำแหน่งของโนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูลที่ระยะ $x \times R$ คือ

$$\begin{aligned}
 P(x|N, n, k) &= \frac{E_1(x, N, n, k)}{E_2(N, n, k)} \\
 &= \frac{\binom{n-x-1}{k-2} (k-1)! (N-k)!}{(N-1)!} \\
 &= \frac{\binom{n-x-1}{k-2}}{\binom{N-1}{k-1}} ; \quad n-k+1 \geq x \\
 &\quad ; \quad N \geq k
 \end{aligned} \tag{5.1}$$

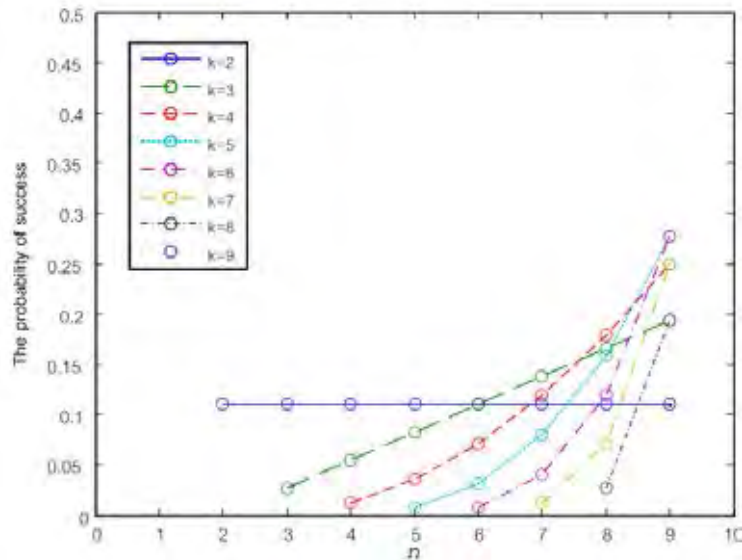
จากรูปที่ 5.2 คือความน่าจะเป็นของความสำเร็จเมื่อมีแพ็กเก็ตข้อมูลในตัวเอง ($x=0$) มีแกน x คือระยะห่างระหว่างโนดร้องขอและโนดข้างถนน หรือ n และแกน y คือ ความน่าจะเป็นของความสำเร็จ ผลการทดลองแสดงให้เห็นว่าความน่าจะเป็นของความสำเร็จจะลดลงในขณะที่ n เพิ่มขึ้น ซึ่งโนดที่ใกล้เคียงกับโนดข้างถนนอาจไม่จำเป็นต้องร้องขอแพ็กเก็ตข้อมูล เนื่องจากโนดที่อยู่ไกลกว่าออกไปมีการร้องขอ จึงทำให้โนดที่ใกล้เคียงกับโนดข้างถนนได้รับแพ็กเก็ตข้อมูลจากการเป็นส่วนหนึ่งในเส้นทางการร้องขอข้อมูลของโนดที่ไกลกว่า



รูปที่ 5.2: ความน่าจะเป็นของความสำเร็จเมื่อมีแพ็กเก็ตข้อมูลในตัวเอง ($x=0$)

จากรูปที่ 5.3 คือ ความน่าจะเป็นเมื่อมีการส่งแพ็กเก็ตการร้องขอและได้รับแพ็กเก็ตข้อมูลกลับมา ก่อนหน้า 1 ฮอป ($x=1$) มีแกน x คือระยะห่างระหว่างโนดร้องขอและโนดข้างถนน หรือ n และแกน

y คือ ความน่าจะเป็นของความสำเร็จ ผลการทดลองแสดงให้เห็นว่าความน่าจะเป็นของความสำเร็จจะเพิ่มขึ้น ในขณะที่ n เพิ่มขึ้น เนื่องจากโอกาสที่จะได้รับแพ็กเก็ตข้อมูลจากโหนดก่อนหน้ามีระยะ 1 ฮอป ซึ่งจะช่วยให้ประสิทธิภาพในระบบ เพราะโหนดร้องขอไม่ต้องส่งแพ็กเก็ตการร้องขอไปหลายช่วงฮอป



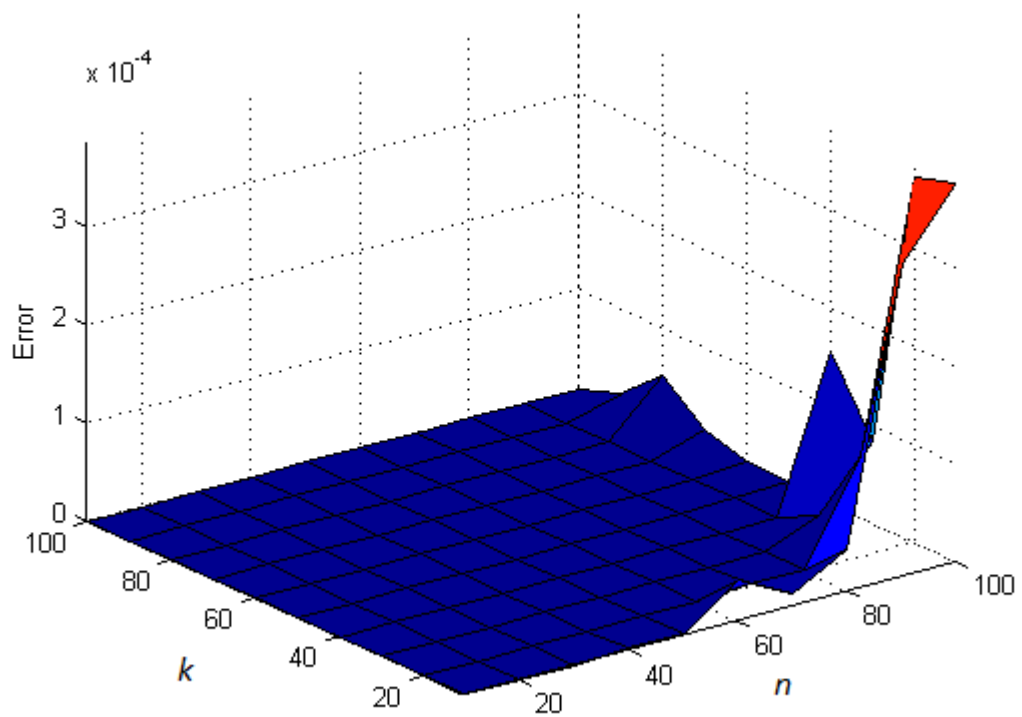
รูปที่ 5.3: ความน่าจะเป็นของความสำเร็จเมื่อได้รับการตอบกลับแพ็กเก็ตข้อมูลก่อนหน้า 1 ฮอป ($x=1$)

จากสมการที่ (5.2) เราจะจำลองระบบโดยใช้การจำลองแบบ Monte-Carlo จำนวน 1 ล้านครั้ง ดังแสดงในรูปที่ 5.4 แกน x ของกราฟมีค่า n แตกต่างกันตั้งแต่ 10 ถึง 100 ตำแหน่ง และแกน y ของกราฟมีค่า k ตั้งแต่ 10 ถึง 100 ตำแหน่ง กราฟแสดงให้เห็นว่าข้อผิดพลาดสูงสุดของสมการ (5.2) เท่ากับ 3.86×10^{-4} ซึ่งสามารถใช้งานได้จริง และเราสามารถคำนวณหาค่า (Requester-Provider round trip time Delay) หรือ RPD โดยใช้ $P(x|N, n, k)$ ได้ดังนี้

$$RPD = \sum_{x=0}^n P(x|N, n, k) T(x) \quad (5.2)$$

โดยที่ $T(x)$ เป็นเวลาประวิงที่ระยะทางระหว่างโหนดผู้ร้องขอไปยังตำแหน่งของโหนดที่ผู้ร้องขอสามารถเจอแพ็กเก็ตข้อมูลเท่ากับ $x \times R$

สำหรับการหาค่า $T(x)$ ในวิทยานิพนธ์ฉบับนี้ไม่ได้กำหนดสูตรไว้ และเป็นงานที่จะต้องทำต่อเนื่องในอนาคต



รูปที่ 5.4: ความผิดพลาดสัมบูรณ์ของ $P(x|N, n, k)$ เมื่อจำลองด้วย Monte-Carlo 1 ล้านครั้ง

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

ในวิทยานิพนธ์ฉบับนี้ได้พัฒนาโพรโทคอลการแจกจ่ายข้อมูลในโครงข่ายแอตสอกสำหรับยานพาหนะโดยใช้สถาปัตยกรรมชื่อข้อมูล เนื่องจากการใช้งานสถาปัตยกรรมแบบ TCP/IP มีความไม่เหมาะสมกับโครงข่ายแอตสอกสำหรับยานพาหนะที่มีสภาพแวดล้อมเป็นแบบเปลี่ยนแปลงตลอดเวลา และเมื่อพิจารณาในมุมมองของผู้ใช้บนท้องถนนที่สนใจเฉพาะเนื้อหาของข้อมูลที่ต้องการ ซึ่งผู้ใช้ไม่จำเป็นต้องสนใจว่าข้อมูลที่ต้องการมาจากยานพาหนะคันใด รวมถึงกระบวนการของการร้องขอข้อมูล ผู้ใช้ไม่จำเป็นต้องระบุที่อยู่ไอพีปลายทาง ซึ่งโพรโทคอลการแจกจ่ายข้อมูลนี้ออกแบบนี้ได้นำโนตข้างถนนเข้ามาช่วยในการรับข้อมูลจากยานพาหนะและแจกจ่ายข้อมูลที่ได้ต่อไปยังยานพาหนะคันอื่น ๆ

งานวิทยานิพนธ์นี้ได้อธิบายการออกแบบการทำงานของโพรโทคอลการแจกจ่ายข้อมูลที่มีความแตกต่างกับโครงข่ายชื่อข้อมูลอื่น ๆ ที่ประยุกต์ใช้งานในโครงข่ายแอตสอกสำหรับยานพาหนะ ซึ่งงานวิจัยที่ผ่านมา ยังไม่ได้พิจารณาการแจกจ่ายข้อมูลระหว่างยานพาหนะกับโนตข้างถนนที่มีข้อมูลที่ต้องการ งานวิจัยฉบับนี้จึงมุ่งเน้นความสนใจไปที่การแจกจ่ายข้อมูลระหว่างยานพาหนะกับโนตข้างถนนที่มีข้อมูลที่ต้องการ เพื่อกำหนดทิศทางและพื้นที่จำกัดในการร้องขอข้อมูล

ผลจากการจำลองด้วยโปรแกรม NS-3 โดยใช้ข้อมูลตำแหน่งจริงของยานพาหนะจากโครงการสาธารณสุข สรุปลงได้ว่า โพรโทคอลการแจกจ่ายข้อมูลที่น่าเสนอโดยใช้โนตข้างถนน ทำให้อัตราการรับข้อมูลสำเร็จมากกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม ปริมาณทราฟฟิกในระบบมีค่าลดลง และมีระยะเวลาการแจกจ่ายข้อมูลโดยรวมน้อยกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม อีกทั้งค่าวิสัยสามารถยังมีสมรรถนะที่ดีกว่าโพรโทคอลชื่อข้อมูลแบบดั้งเดิม แม้ว่าเวลาการแจกจ่ายข้อมูลของโพรโทคอลที่น่าเสนอมากกว่าโพรโทคอลชื่อแบบดั้งเดิม

6.2 ข้อเสนอแนะ

1. พิจารณาเพิ่มเติมในเรื่องของการสร้างเส้นทางก่อนการส่งแพ็กเก็ตการร้องขอข้อมูล เพื่อเพิ่มความน่าเชื่อถือในระบบมากขึ้น เนื่องจากการสร้างเส้นทางรอไว้ก่อนที่จะส่งแพ็กเก็ตการร้องขอออกไป ระบบจะไม่ต้องส่งแพ็กเก็ตการร้องขอซ้ำ ๆ
2. พิจารณาเพิ่มเติมในเรื่องการใช้งานร่วมกันระหว่างโปรแกรมจำลองการจราจรและโปรแกรมการสื่อสาร ให้มีการส่งข้อมูลสื่อสารกันไปมาระหว่างโปรแกรมทั้งสอง โดยการปรับพารามิเตอร์ในการควบคุมสัญญาณไฟจราจร เพื่อลดความคับคั่งของการจราจร
3. พิจารณาการวิเคราะห์เพิ่มเติมเกี่ยวกับการนำโครงข่ายชื่อข้อมูลไปใช้กับโนตเคลื่อนที่อื่น ๆ เช่น เรือหรือเครื่องบิน โดยกำหนดโนตข้างถนน เป็น ทูบที่อยู่ในแม่น้ำหรือหอบังคับการบิน เป็นต้น
4. พิจารณาเพิ่มเติมเกี่ยวกับการเชื่อมต่อของโนตข้างถนนกับโครงข่ายแบบมีสายหรือโครงข่ายแบบเซลลูลาร์

5. พิจารณาเรื่องความปลอดภัยในโครงข่าย เนื่องจากโครงข่ายแอตชอกสำหรับยานพาหนะเป็นโครงข่ายไร้สาย ทำให้มีโอกาสที่จะถูกโจมตี การออกแบบโพรโทคอลควรคำนึงถึงความปลอดภัยจากผู้ไม่หวังดีบุกรุกเข้ามาในโครงข่าย โดยการส่งข้อมูลปลอม หรือ การส่งข้อความซ้ำ ๆ เพื่อรบกวนการสื่อสารในระบบ ดังนั้น ข้อความที่ถูกส่งออกไป ควรจะต้องมีการเข้ารหัสและระบุตัวตนของผู้ใช้งาน อีกทั้งควรมีการรับมือจากการถูกโจมตีจากผู้บุกรุก
6. พิจารณาเพิ่มโนดข้างถนนเข้ามาช่วยในการรับข้อมูล เนื่องจากหากโนดข้างถนนเกิดความเสียหาย หรือไม่สามารติดต่อสื่อสารกับยานพาหนะ โพรโทคอลการแจกจ่ายข้อมูลจะไม่สามารถทำงานได้
7. พิจารณาใช้งานร่วมกับโครงข่าย SDN (Software-Defined Networking) โดยให้โนดข้างถนนทำงานบน control plane และยานพาหนะทำงานบน data plane

รายการอ้างอิง

- [1] Ministry of Land Infrastructure and Transport. The system outline of VICS [online]. Available from: <http://www.vics.or.jp/english/vics/index.html> [27 March 2560.]
- [2] B. Karp, H.-T. Kung. GPSR: greedy perimeter stateless routing for wireless net-works. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, 243–254. 2000.
- [3] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong and K.-K. Lee. A-STAR: a mobile ad hoc routing strategy for metropolis vehicular communications, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications, 989–999. 2004.
- [4] J. Zhao, G. Cao. VADD: vehicle-assisted data delivery in vehicular ad hoc net-works. IEEE Transactions on Vehicular Technology(2008):1910–1922.
- [5] C. Lochert, M. Mauve, H. Fubler and H. Hartenstein Geographic routing in city scenarios. ACM SIGMOBILE Mobile Computing and Communications (2005):69–72.
- [6] K. Katsaros, M. Dianati, R. Tafazolli and R. Kernchen. CLWPR: a novel cross-layer optimized position based routing protocol for VANETs. Proceedings of Vehicular Networking Conference, 139–146. 2011.
- [7] P. Costa, D. Frey, M. Migliavacca and L. Mottola. Towards lightweight information dissemination in inter-vehicular networks. Proceedings of the 3rd International Workshop on Vehicular ad Hoc Networks, 20–29. 2006.
- [8] W. Viriyasitavat, F. Bai and O.K. Tonguz. UV-CAST: an urban vehicular broad-cast protocol Proceedings of Vehicular Networking Conference, 25–32. 2010.
- [9] Y. Ding, C. Wang, and L. Xiao. A Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks. IEEE Transactions on Vehicular Technology (2007): 59-68.
- [10] J. Zhao, Y. Zhang, and G. Cao. Data Pouring and Buffering on The Road: A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks. IEEE Transactions on Vehicular Technology (2007).
- [11] C. Wu, S. Ohzahata, T. Kato. Fuzzy logic based multi-hop broadcast for high-density vehicular ad hoc networks. Proceedings of Vehicular Networking Conference, 17–24. 2010.
- [12] C. Wu, S. Ohzahata and T. Kato. A broadcast path diversity mechanism for delay sensitive VANET safety applications. Proceedings of IEEE Vehicular Networking Conference, 171–176. 2011.

- [13] V. Jacobson, D.K. Smetters, J.D Thornton, M. Plass, N. Briggs and R. L. Braynard. Networking Named Content. the 5th International Conference on Emerging Networking Experiments and Technologies, Dec. 2009.
- [14] L. Zhang. Named Data Networking (NDN) Project. Technical report, PARC, (Oct. 2010).
- [15] C. Yi, A. Afanasyev, L. Wang,. Adaptive forwarding in named data networking. ACM SIGCOMM Computer Communication vol. 42, no. 3(2012): 62–67.
- [16] Yuan, H., Crowley, P.: Scalable pending interest table design: from principles to practice. Proceedings of IEEE International Conference on Computer Communications(April 2014):2049–2057.
- [17] So, W., Narayanan, A., Oran, D., et al. Toward fast NDN software forwarding lookup engine based on hash tables. the Eighth ACM/IEEE Symposium on Architectures for Networking and Communications Systems (October 2012):85–86.
- [18] Wang, J., Peng, C., Li, C., et al.: Implementing instant messaging using named data. Proceedings of the Sixth Asian Internet Engineering Conference, pp. 40–47, (November 2010).
- [19] F. Bai, and B. Krishnamachari. Exploiting the wisdom of the crowd: localized, distributed information-centric VANETs IEEE Magazine on communications vol. 48 no. 5 (May 2010).
- [20] J. Wang, R.Wakikawa, and L. Zhang. DMND: collecting data from mobiles using named data. Proceedings of the IEEE International Conference on Vehicular Networking (2010).
- [21] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, L. Zhang. Data Naming in Vehicle-to-Vehicle Communications. Proceedings of the IEEE Conference on Computer Communications Workshops, 2012.
- [22] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R.Wakikawa, and L. Zhang. Rapid Traffic Information Dissemination Using Named Data. Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design (2012).
- [23] G. Arnould, D. Khadraoui, and Z. Habbas. A Self-Organizing Content Centric Network Model for Hybrid Vehicular Ad-Hoc Networks. Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications (2011).
- [24] P. TalebiFard and V.C.M. Leung. A Content Centric Approach to Dissemination of Information in Vehicular Networks. Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications (2012).

- [25] C. Campolo, M. Amadeo and A. Molinaro. Content-Centric Networking: is that a Solution for Upcoming Vehicular Networks. Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications (2012).
- [26] M. Amadeo, C. Campolo, and A. Molinaro. CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks. IEEE Letters on Communications ,vol. 16, no. 9. (2012).
- [27] W. Lucas, A. Alexander, K. Romain. Rapid traffic information dissemination using named data. Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design(2012):7–12.
- [28] G. Giulio, P. Davide, W. Lucas. ACM HotMobile 2013 poster: vehicular inter-networking via named data. ACM SIGMOBILE Mob. Comput. Commun. Rev. (2013):23–24.
- [29] P. TalebiFard, V. Leung. A content centric approach to dissemination of information in vehicular networks. Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (October 2012):17–24.
- [30] IEEE. Std. 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, 1999.
- [31] Euisin Lee, Eun-Kyu Lee, Mario Gerla and Soon Y. Oh. Vehicular cloud networking: architecture and design principles. Proceeding of the IEEE Magazine, on communications vol. 52, no. 2. (2014).
- [32] M. Meisel, V. Pappas, and L. Zhang. Ad Hoc Networking via Named Data. Proceedings of the Fifth ACM Workshop on Mobility in the Evolving Internet Architecture (September 2010).
- [33] S. Panichpapiboon and W. Pattara-Atikom. Connectivity requirements for a self-organizing traffic information systems. IEEE Transactions on Vehicular Technology (2008): 3333-3340.
- [34] E. Schoch et al. Communication Patterns in VANETs. IEEE Communications Magazine vol. 46(11)(2008): 119-125.
- [35] Lusheng Miao, Karim Djouani, Barend J. van Wyk and Yskandar Hamam. A Survey of IEEE 802.11p MAC Protocol. Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), 2011.

- [36] IEEE802-WG IEEE 802.11 part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. ANSI/IEEE Std 802.11 1999 Edition (R2007), 2007.
- [37] IEEE P802.11p TM/D3.0 Draft Standard for Information Technology-Amendment 7: Wireless Access in Vehicular Environments. ANSI/IEEE Std 802.11 1999 Edition (R2007) IEEE 802.11 working group, 2007.
- [38] Richard W. Rothery. Traffic Flow Theory - Car following models (Chapter 4) Transportation Research Board report United States Department of Transportation - Federal. Highway Administration, 1999).
- [39] Weingartner, E., Lehn, H., and Wehrle, K. A Performance Comparison of Recent Network Simulators. in Proc. of IEEE ICC, 2009.
- [40] The network simulator ns-2 [Online]. Available from : <http://www.isi.edu/nsnam/ns/> [2014, March 5]
- [41] Henderson, T. R., Roy, S., Floyd, S., and Riley, G. F. ns-3 project goals. in proc. of WNS2, 2006 : 13.
- [42] Varga, A., and Hornig, R. An overview of the OMNeT++ simulation environment. in proc. of SIMUTools, 2008.
- [43] Barr, R., Haas Z. J., and Renesse, R. V. JiST: an efficient approach to simulation using virtual machines. Software, Practice. Experiment, 2004.
- [44] Mueller, K. SimPy documentation [Online]. Available from : <http://simpy.sourceforge.net/discuss.htm>.
- [45] เพียงพูน จักรแก้ว. การประเมินเปรียบเทียบแบบจำลองบน ns-3 และระบบทดสอบจริง : กรณีศึกษาของระบบแจ้งเตือนอุบัติเหตุการที่ใช้ VANET. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ, ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2553.
- [46] SUMO : Simulation of Urban Mobility, Available from: <http://sumo.sourceforge.net/>.
- [47] U. Kijawattanee, A. Teeratat, W. Sorawee, K. Patrachart, R. Jittichai and A. Chaodit. Traffic Data Analysis on Sathorn Road with Synchro Optimization and Traffic Simulation. Proceedings of the Electrical Engineering Conferance, 2016.
- [48] E. Teramoto, M. Baba, H. Mori, Y. Asano, and H. Morita. NETSTREAM: Traffic Simulator for Evaluating Traffic Information Systems. Proceedings of the IEEE International Conference on Intelligent Transportation Systems, 1997.
- [49] Ai H. Ho, Yao H. Ho, and Kien A. Hua. A Connectionless Approach to Mobile Ad Hoc Networks. Street Environment, 2003.

ภาคผนวก

ก โปรแกรมสำหรับการตั้งค่าพารามิเตอร์ต่าง ๆ และ โปรแกรมหลักที่ใช้ในการจำลอง (configuration and code simulation)

รายละเอียดการทำงานของโปรแกรมหาดังนี้

- บรรทัดที่ 1-5: ตัวอย่างคำสั่งการจำลองใน terminal
- บรรทัดที่ 8-27: ไลบรารีที่ถูกเรียกใช้
- บรรทัดที่ 29: ตั้งชื่อสำหรับ log file
- บรรทัดที่ 37: การประกาศตัวแปร ns3 boost และ std ตามลำดับ
- บรรทัดที่ 43-98: ฟังก์ชันสำหรับแฮนด์พุท เพื่อเก็บค่าของการรับส่งแพ็กเก็ตที่เกิดขึ้นในการจำลอง
- บรรทัดที่ 101-110: ฟังก์ชันของการเก็บตำแหน่งของโหนดในช่วงเวลาต่าง ๆ และ ฟังก์ชันการรับแพ็กเก็ต
- บรรทัดที่ 112-133: คลาสสำหรับการเก็บค่า pcap
- บรรทัดที่ 135-148: ฟังก์ชันสำหรับการบันทึกค่าของตารางรอดำเนินการของแพ็กเก็ตการร้องขอ (PIT)
- บรรทัดที่ 151-161: ฟังก์ชันการทำงานของส่วนต่อประสานในโครงข่ายแอตสอกสำหรับยานพาหนะ
- บรรทัดที่ 165-218: ตั้งค่าเริ่มต้นสำหรับการทำงานในโปรแกรมหลัก
- บรรทัดที่ 221-233: ตรวจสอบคำสั่งใน terminal
- บรรทัดที่ 236: ปรับเปลี่ยนค่า seed ในการจำลองแต่ละครั้ง
- บรรทัดที่ 238-256: ตั้งค่าพารามิเตอร์ในระดับชั้น MAC
- บรรทัดที่ 259-268: เรียกใช้ไฟล์การเคลื่อนที่ของโหนด
- บรรทัดที่ 271-281: กำหนดตำแหน่งของโหนดข้างถนน
- บรรทัดที่ 290-292: กำหนดชนิดของโหนดข้างถนนและยานพาหนะ
- บรรทัดที่ 295-296: กำหนดคุณลักษณะส่วนต่อประสานของโหนดทั้งหมด
- บรรทัดที่ 298-306: กำหนดการทำงานแบบโครงข่ายชื่อข้อมูล
- บรรทัดที่ 309-340: กำหนดคุณลักษณะการทำงานในชั้น application
- บรรทัดที่ 343-405: กำหนดชื่อสำหรับการเก็บค่าต่าง ๆ หลังจากการจำลอง
- บรรทัดที่ 407-418: จบการทำงานของโปรแกรม

```

1 //how to run
2 //NS_LOG=ndn.Consumer:ndn.Producer:ndn.fw.V2v ./waf --run "VanetRSU
  --traceFile=/home/nrg-cu/sathon_road/nsmobility/ns2mobility0.001.
  tcl --nodeNum=16 --duration=100.0" > outputRSU.txt 2>&1
3 //NS_LOG=ndn.Consumer:ndn.Producer:ndn.fw.V2v:vanetRSU ./waf --run "
  VanetRSU --traceFile=/home/nrg-cu/sathon_road/nsmobility/
  ns2mobility0.001.tcl --nodeNum=16 --duration=100.0" > outputRSU.
  txt 2>&1
4 //./waf --run "VanetRSU --traceFile=/home/nrg-cu/sathon_road/
  nsmobility/ns2mobility0.001.tcl --nodeNum=16 --duration=100.0"
5 //NS_LOG=vanetRSU ./waf --run "VanetRSU --traceFile=/home/nrg-cu/
  sathon_road/nsmobility/ns2mobility0.001.tcl --nodeNum=16 --
  duration=100.0" > /home/nrg-cu/text/outputRSU.txt 2>&1
6 //////////////////////////////////////
7
8 #include "ns3/core-module.h"
9 #include "ns3/network-module.h"
10 #include "ns3/applications-module.h"
11 #include "ns3/wifi-module.h"
12 #include "ns3/mobility-module.h"
13 #include "ns3/internet-module.h"
14 #include "ns3/ndnSIM-module.h"
15 #include "ns3/ns2-mobility-helper.h"
16 #include <boost/shared_ptr.hpp>
17 #include <boost/make_shared.hpp>
18 #include <boost/lexical_cast.hpp>
19 #include <boost/tokenizer.hpp>
20 #include "ndn-v2v-net-device-face.h"
21 #include "v2v-tracer.h"
22 #include "car-relay-tracer.h"
23 #include <ns3/ndnSIM/utils/tracers/ndn-cs-tracer.h>
24 #include <ns3/ndnSIM/utils/tracers/ndn-app-delay-tracer.h>
25 #include <ns3/ndnSIM/utils/tracers/ndn-l3-aggregate-tracer.h>
26 #include <ns3/ndnSIM/utils/tracers/ndn-l3-rate-tracer.h>
27 #include <ns3/ndnSIM/utils/tracers/l2-rate-tracer.h>
28 //////////////////////////////////////
29 NS_LOG_COMPONENT_DEFINE ("vanetRSUall");
30 //////////////////////////////////////
31 using namespace ns3;
32 using namespace boost;
33 using namespace std;
34
35 #define MAX_NODES 2000
36 static int packet_count = 0;
37 static int mac_tx_count = 0;
38 static int mac_tx_drop = 0;

```

```

39 static int mac_rx_count = 0;
40 static int mac_rx_drop = 0;
41 ///////////////////////////////////////////////////////////////////
42 // Callback for simulation traces//
43 void MacTxCount ( Ptr<const Packet> p ) {
44 mac_tx_count++;
45   NS_LOG_DEBUG ( "[mac_tx_count] Packet is going to be transmitted
      ...mac_tx_count= "<< mac_tx_count );
46 }
47
48 void MacTxDrop ( Ptr<const Packet> p ) {
49 mac_tx_drop++;
50   NS_LOG_DEBUG ( "[mac_tx_drop] Packet has been dropped before
      transmission...mac_tx_drop= "<< mac_tx_drop );
51 }
52
53 void MacRxCount ( Ptr<const Packet> p ) {
54 mac_rx_count++;
55   NS_LOG_DEBUG ( "[mac_rx_count] Packet has been received , going to
      forward...mac_rx_drop= "<< mac_rx_count );
56 }
57
58
59 void MacRxDrop ( Ptr<const Packet> p ) {
60 mac_rx_drop++;
61   NS_LOG_DEBUG ( "[mac_rx_drop] Packet has been dropped by Mac
      before forward...mac_rx_drop= "<< mac_rx_drop );
62 }
63
64 void PrintTrasmission() {
65 //std::cout<<Simulator::Now().GetSeconds()<<"\t"<< mac_tx_count <<"\
      t"<<mac_tx_drop <<"\t"<< mac_rx_count <<"\t"<<mac_rx_drop <<"\n";
66   NS_LOG_DEBUG ( "[PRINT_TRANSMISSION] Time= "<< Simulator::Now().
      GetSeconds()<<"\t"<<"mac_tx_count= "<<mac_tx_count<<"\t"<<"
      mac_tx_drop= "<<mac_tx_drop <<"\t"<<"Mac rx coun t= "<<
      mac_rx_count<<"\t"<<"mac_rx_drop = "<<mac_rx_drop );
67 Simulator::Schedule(Seconds(50.0), &PrintTrasmission);
68 }
69 ///////////////////////////////////////////////////////////////////
70 void OutInterest (Ptr<const ns3::ndn::InterestHeader> interestHeader
      , Ptr<const ns3::ndn::Face> face) {
71   NS_LOG_DEBUG ( "[OUT INTEREST] Time= "<< Simulator::Now().
      GetSeconds()<< "\t" << "NID= " << face->GetNode()->GetId()<< "\
      t" <<"Nonce= " << interestHeader->GetNonce()<< "\t" <<"Name= "
      << interestHeader->GetName() <<"\n ");
72 }

```

```

73
74
75 void InInterest ( Ptr<const ns3::ndn::InterestHeader> interestHeader
    , Ptr<const ns3::ndn::Face> face) {
76   NS_LOG_DEBUG ( "[ IN INTEREST ] Time= "<< Simulator::Now().
        GetSeconds()<< "\t" <<"NID= "<< face->GetNode()->GetId()<<"\
        tNonce= "<< interestHeader ->GetNonce()<<"\tName= "<<
        interestHeader ->GetName()<<"\tPosition= "<< face->GetNode()->
        GetObject<MobilityModel> ( )->GetPosition()<<"\n " );
77 }
78
79
80 void DropInterest ( Ptr<const ns3::ndn::InterestHeader>
    interestHeader , Ptr<const ns3::ndn::Face> face) {
81   NS_LOG_DEBUG ( "[ DROP INTEREST ] Time= "<< Simulator::Now().
        GetSeconds()<<"\tNID= "<< face->GetNode()->GetId()<<"\tNonce= "
        << interestHeader ->GetNonce()<<"\tName= "<< interestHeader ->
        GetName()<<"\n " );
82 }
83
84
85 void InData ( Ptr<const ns3::ndn::ContentObjectHeader> contentHeader
    , Ptr<const ns3::Packet> packet , Ptr<const ns3::ndn::Face> face
    ) {
86   NS_LOG_DEBUG ( "[ IN DATA ] Time= "<< Simulator::Now().GetSeconds()
        <<"\tNID= "<<face->GetNode()->GetId()<<"\tName= "<<
        contentHeader->GetName()<<"\tPosition= "<<face->GetNode()->
        GetObject<MobilityModel> ()->GetPosition()<<"\n " );
87 }
88
89
90 void OutData ( Ptr<const ns3::ndn::ContentObjectHeader>
    contentHeader , Ptr<const ns3::Packet> packet , bool value , Ptr<
    const ns3::ndn::Face> face) {
91
92   NS_LOG_DEBUG ( "[ OUT DATA ] Time= "<< Simulator::Now().GetSeconds()
        <<"\tNID= "<<face->GetNode()->GetId()<<"\tName= "<<
        contentHeader-> GetName()<<"\n " );
93 }
94
95
96 void DropData ( Ptr<const ns3::ndn::ContentObjectHeader>
    contentHeader , Ptr<const ns3::Packet> packet , Ptr <const ns3::
    ndn::Face> face) {
97   NS_LOG_DEBUG ( "[ DROP DATA ] Time= "<< Simulator::Now().GetSeconds()
        <<"\tNID="<<face->GetNode()->GetId()<<"\tName= "<<contentHeader

```

```

        ->GetName () << "\n ";
98 }
99
100 ///////////////////////////////////////////////////////////////////
101 void printPosition ( Ptr<const MobilityModel> mobility){
102     NS_LOG_INFO("Time= "<< Simulator::Now().GetSeconds() << "\tPosition
        of:"<< mobility->GetObject<Node>()->GetId()<< ":"<<mobility->
        GetPosition());
103 }
104
105 void ReceivePacket ( Ptr<Socket> socket)
106 {
107     NS_LOG_UNCOND ( "ReceivePacket!");
108     packet_count++;
109 }
110 ///////////////////////////////////////////////////////////////////
111
112 class PcapWriter
113 {
114 public:
115     PcapWriter (const std::string &file)
116     {
117         PcapHelper helper;
118         m_pcap = helper.CreateFile (file, std::ios::out, PcapHelper::
            DLT_IEEE802_11);
119         //m_pcap = helper.CreateFile (file, std::ios::out, PcapHelper::
            DLT_IEEE802_11_RADIO);
120     }
121
122     void
123     TracePacket (Ptr<const Packet> packet)
124     {
125         // static PppHeader pppHeader;
126         //pppHeader.SetProtocol (0x0077);
127         //m_pcap->Write (Simulator::Now (), pppHeader, packet);
128         m_pcap->Write (Simulator::Now (), packet);
129     }
130
131 private:
132     Ptr<PcapFileWrapper> m_pcap;
133 };
134 ///////////////////////////////////////////////////////////////////
135 void
136 PeriodicStatsPrinter (Ptr<Node> node, Time next)
137 {
138     Ptr<ndn::Pit> pit = node->GetObject<ndn::Pit> ();

```

```

139
140 NS_LOG_DEBUG ( "[PIT] Time= "<< Simulator::Now().GetSeconds() << "\tNID
      = "<< node->GetId() << "\tNumberOfPitEntries=" << pit->GetSize () << "\n
      ");
141
142     std::cout << Simulator::Now().ToDouble (Time::S) << "\t"
143         << node->GetId () << "\t"
144         << pit->GetSize () << "\n";
145
146     Simulator::Schedule (next, PeriodicStatsPrinter, node, next);
147     // Simulator::Schedule (next, PeriodicStatsPrinter, node, next);
148 }
149
150 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
151 Ptr<ndn::NetDeviceFace>
152 V2vNetDeviceFaceCallback (Ptr<Node> node, Ptr<ndn::L3Protocol> ndn,
      Ptr<NetDevice> device)
153 {
154     NS_LOG_DEBUG ("Creating ndn::V2vNetDeviceFace on node " << node->
      GetId ());
155
156     Ptr<ndn::NetDeviceFace> face = CreateObject<ndn::V2vNetDeviceFace>
      (node, device);
157     ndn->AddFace (face);
158     // NS_LOG_LOGIC ("Node " << node->GetId () << ": added
      NetDeviceFace as face #" << *face);
159
160     return face;
161 }
162 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
163 int main (int argc, char *argv[])
164 {
165     // disable fragmentation
166     Config::SetDefault ("ns3::WifiRemoteStationManager::
      FragmentationThreshold", StringValue ("2200"));
167     Config::SetDefault ("ns3::WifiRemoteStationManager::
      RtsCtsThreshold", StringValue ("2200"));
168     Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode
      ", StringValue ("OfdmRate6MbpsBW10MHz"));
169
170     // vanet hacks to CcnxL3Protocol
171     Config::SetDefault ("ns3::ndn::ForwardingStrategy::
      CacheUnsolicitedData", StringValue ("true")); // not necessary,
      but for clarity
172
173     // v2v device

```

```

174 Config::SetDefault ("ns3::ndn::V2vNetDeviceFace::MaxDelay",
    StringValue ("2ms"));
175 Config::SetDefault ("ns3::ndn::V2vNetDeviceFace::
    MaxDelayLowPriority", StringValue ("5ms"));
176 Config::SetDefault ("ns3::ndn::V2vNetDeviceFace::MaxDistance",
    StringValue ("150"));
177 Config::SetDefault ("ns3::ndn::V2vNetDeviceFace::
    MaxRetransmissionAttempts", StringValue ("0"));
178 // !!! very important parameter !!!
179 // Should keep PIT entry to prevent duplicate interests from re-
    propagating
180 Config::SetDefault ("ns3::ndn::Pit::PitEntryPruningTimeout",
    StringValue ("1s"));
181 ///////////////////////////////////////////////////////////////////
182 CommandLine cmd;
183     std::string traceFile;
184     cmd.AddValue ("traceFile", "Ns2 movement trace file", traceFile);
185
186     std::string floder;
187     cmd.AddValue ("floder", "name of floder output", floder);
188
189     int nodeNum;
190     cmd.AddValue ("nodeNum", "Number of nodes", nodeNum);
191
192     double duration;
193     cmd.AddValue ("duration", "Duration of Simulation", duration);
194
195     uint32_t run = 1;
196     cmd.AddValue ("run", "Run", run);
197
198     string batches = "50s 1 550s 1 1050s 1 1550s 1 2050s 1 2550s 1
        3050s 1 3550s 1 4050s 1 4550s 1 5050s 1 5550s 1 6050s 1 6550s 1
        7050s 1 7550s 1 8050s 1 8550s 1 9050s 1 9550s 1";
199
200     cmd.AddValue ("batches", "Consumer interest batches", batches);
201
202     double frequency= 1.0;
203     cmd.AddValue ("frequency", "Frequency of interest", frequency);
204
205     double range =1000.0;
206     cmd.AddValue ("range", "Transmission Range", range);
207
208     uint32_t user = 10;
209     cmd.AddValue ("user", "Number of user that sending interest",
        user);
210

```



```

211     bool distanceDelay = false;
212     bool jumpDistance = false;
213     bool retx = false;
214     bool cachedTime = false;
215
216     // Parse command line attribute
217
218     cmd.Parse (argc,argv);
219
220     // Check command line arguments
221     if (traceFile.empty () || nodeNum <= 0 || duration <= 0 )
222     {
223         std::cout << "Usage of " << argv[0] << " :\n\n"
224         ". /waf --run \"VanetRSU\"
225         \" --traceFile=/examples/vanet/mobility.tcl\"
226         \" --nodeNum=300 --duration=100.0 \" \n\n"
227         "NOTE: ns2-traces-file could be an absolute or relative path.
228         \n\n"
229         "NOTE 2: Number of nodes present in the trace file must match
230         with the command line argument and must\n"
231         " be a positive number. Note that you must know it
232         before to be able to load it.\n\n"
233         "NOTE 3: Duration must be a positive number. Note that you
234         must know it before to be able to load it.\n\n";
235
236         return 0;
237     }
238
239     //////////////////////////////////////
240     Config::SetGlobal ("RngRun", IntegerValue (run));
241     //////////////////////////////////////
242     WifiHelper wifi = WifiHelper::Default ();
243     // wifi.SetRemoteStationManager ("ns3::AarfWifiManager");
244     wifi.SetStandard (WIFI_PHY_STANDARD_80211p_SCH);
245     wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
246     "DataMode", StringValue ("
247     OfdmRate6MbpsBW10MHz"));
248
249     YansWifiChannelHelper wifiChannel;// = YansWifiChannelHelper::
250     Default ();
251     wifiChannel.SetPropagationDelay ("ns3::
252     ConstantSpeedPropagationDelayModel");
253     wifiChannel.AddPropagationLoss ("ns3::
254     ThreeLogDistancePropagationLossModel");
255     wifiChannel.AddPropagationLoss ("ns3::
256     NakagamiPropagationLossModel");

```

```

248     wifiChannel.AddPropagationLoss ("ns3::RangePropagationLossModel"
        , "MaxRange", DoubleValue (range));
249     //YansWifiPhy wifiPhy = YansWifiPhy::Default ();
250     YansWifiPhyHelper wifiPhyHelper = YansWifiPhyHelper::Default ();
251     wifiPhyHelper.SetChannel (wifiChannel.Create ());
252     wifiPhyHelper.Set ("TxPowerStart", DoubleValue (30));
253     wifiPhyHelper.Set ("TxPowerEnd", DoubleValue (30));
254
255     NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
256     wifiMac.SetType ("ns3::AdhocWifiMac");
257
258
259     // 1. Install Mobility Model to Vehicles
260
261     // Create Ns2MobilityHelper with the specified trace log file as
        parameter
262     Ns2MobilityHelper ns2MobH = Ns2MobilityHelper (traceFile);
263
264     // Create all nodes.
265     NodeContainer vehicles;
266     vehicles.Create (nodeNum);
267
268     ns2MobH.Install (); // configure movements for each node, while
        reading trace file
269
270
271     // 2. Create RSUs as Static Nodes at intersection
272     NodeContainer rsu;
273     int numRsu = 3;
274     rsu.Create (numRsu);
275
276     MobilityHelper mobility;
277     Ptr<ListPositionAllocator> positionAlloc = CreateObject<
        ListPositionAllocator> ();
278
279     positionAlloc->Add (Vector (1952.75, -1436.66, 0.0)); // RSU at
        sathon road
280     positionAlloc->Add (Vector (3132.12, -1099.27, 0.0)); // RSU at
        narativas road
281     positionAlloc->Add (Vector (4662.76, -598.36, 0.0)); // RSU at
        radio road
282
283     mobility.SetPositionAllocator (positionAlloc);
284     mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
285
286     mobility.Install (rsu);

```

```

287
288 // Merge the two Node Containers
289
290 NodeContainer allNodes;
291 allNodes.Add(vehicles);
292 allNodes.Add(rsu);
293
294
295 // 2. Install Wifi
296 NetDeviceContainer wifiNetDevices = wifi.Install (wifiPhyHelper,
    wifiMac, allNodes);
297
298 // 3. Install NDN stack
299 NS_LOG_INFO ("Installing NDN stack");
300 ndn::StackHelper ndnHelper;
301 ndnHelper.AddNetDeviceFaceCreateCallback (WifiNetDevice::
    GetTypeId (), MakeCallback (V2vNetDeviceFaceCallback));
302 ndnHelper.SetForwardingStrategy ("ns3::ndn::fw::V2v");
303 ndnHelper.SetContentStore ("ns3::ndn::cs::Lru", "MaxSize", "
    10000");
304 ndnHelper.SetDefaultRoutes (true);
305 ndnHelper.SetPit ("ns3::ndn::pit::Persistent::AggregateStats");
    //add pit stat
306 ndnHelper.Install(allNodes);
307
308
309 // 4. Set up applications
310 NS_LOG_INFO ("Installing Applications");
311 ////////////////////////////////////////////////////
312 Ptr<UniformRandomVariable> x = CreateObject<
    UniformRandomVariable> ();
313 ApplicationContainer consumer[nodeNum-1];
314 ////////////////////////////////////////////////////
315 ///ConsumerBatches
316 ndn::AppHelper consumerHelper ("ns3::ndn::ConsumerBatches");
317 consumerHelper.SetPrefix ("/traffic/RamaIV/westbound
    /{400,410}/{120002202014,130002202014}/speed/19375887");
318 consumerHelper.SetAttribute ("Batches", StringValue (batches));
319 ////////////////////////////////////////////////////
320 //ndn::AppHelper consumerHelper ("ns3::ndn::ConsumerCbr");
321 //consumerHelper.SetPrefix ("/prefix");
322 // consumerHelper.SetAttribute ("Frequency", DoubleValue (frequency)
    );
323 //consumerHelper.SetAttribute ("Frequency", StringValue ("10")); //
    10 interests a second
324

```

```

325
326 ///////////////////////////////////////////////////////////////////
327     ndn::AppHelper producerHelper ("ns3::ndn::Producer");
328     producerHelper.SetPrefix ("/");
329     producerHelper.SetAttribute ("PayloadSize", StringValue("1024"))
        ;
330
331     // Install producer and consumer both on vehicles and RSUs.
332     producerHelper.Install(rsu);
333     //consumerHelper.Install(vehicles);
334 ///////////////////////////////////////////////////////////////////
335 for ( int i=0; i < user;i++)
336 {
337     int userNode=x->GetInteger (0,nodeNum-1);
338     consumer[i]=consumerHelper.Install(vehicles.Get (userNode));
339     //std::cout <<"user is "<<userNode<< std::endl;
340 }
341 ///////////////////////////////////////////////////////////////////
342
343     //string floder = "outputRSU2-mobility57900_58900s"; //edit 2
344 string prefix = "car-" + lexical_cast<string> (range) + "-" +
        lexical_cast<string> (nodeNum) + "-" + lexical_cast<string> (user)
        + "-" + lexical_cast<string> (run) + "-";
345 ///////////////////////////////////////////////////////////////////
346     //PcapWriter trace (floder+"/pcap/"+prefix+"wifi-pcap-IEEE802_11.
        pcap");
347     //PcapWriter trace ("wifi-pcap-IEEE802_11_RADIO.pcap");
348     //wifiPhyHelper.EnablePcap (floder+"/pcap-all/"+prefix+"ndn",
        wifiNetDevices);
349
350
351     //Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
        L3Protocol/FaceList*/Ndn",MakeCallback (&PcapWriter::
        TracePacket, &trace));
352 ///////////////////////////////////////////////////////////////////
353 // Trace of level 2.5
354 Config::ConnectWithoutContext ("/NodeList*/DeviceList*/ $ns3::
        WifiNetDevice/Mac/MacTx" ,MakeCallback (&MacTxCount));
355 Config::ConnectWithoutContext ("/NodeList*/DeviceList*/ $ns3::
        WifiNetDevice/Mac/MacRx" ,MakeCallback (&MacRxCount));
356 Config::ConnectWithoutContext ("/NodeList*/DeviceList*/ $ns3::
        WifiNetDevice/Mac/MacTxDrop" , MakeCallback (&MacTxDrop));
357 Config::ConnectWithoutContext ("/NodeList*/DeviceList*/ $ns3::
        WifiNetDevice/Mac/MacRxDrop" , MakeCallback (&MacRxDrop));
358
359 // Trace

```

```

360 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/OutInterests" ,MakeCallback(&OutInterest));
361 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/InInterests" ,MakeCallback(&InInterest));
362 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/DropInterests" ,MakeCallback(&DropInterest));
363 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/InData" , MakeCallback(&InData));
364 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/DropData" ,MakeCallback(&DropData));
365 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::
    ForwardingStrategy/OutData" ,MakeCallback
366 (&OutData));
367 //PIT
368 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::ndn::Pit/
    MaxPitEntryLifetime/PeriodicStatsPrinter" ,MakeCallback(&
    PeriodicStatsPrinter));
369 /** Tracing position */
370 Config::ConnectWithoutContext ("/NodeList/*/ $ns3::MobilityModel/
    CourseChange" ,MakeCallback(&printPosition));
371
372 //////////////////////////////////////
373 boost::tuple< boost::shared_ptr<std::ostream>, std::list<boost::
    shared_ptr<ndn::CarRelayTracer> > >
374 tracing1 = ndn::CarRelayTracer::InstallAll (floder+"/distance/"+
    prefix+"distance.txt", ndn::CarRelayTracer::DISTANCE_WAITING)
    ;
375
376 boost::tuple< boost::shared_ptr<std::ostream>, std::list<boost::
    shared_ptr<ndn::CarRelayTracer> > >
377 tracing2 = ndn::CarRelayTracer::InstallAll (floder+"/jump-
    distance/"+prefix+"jump-distance.txt", ndn::CarRelayTracer::
    JUMP_DISTANCE);
378
379 boost::tuple< boost::shared_ptr<std::ostream>, std::list<boost::
    shared_ptr<ndn::CarRelayTracer> > >
380 tracing3 = ndn::CarRelayTracer::InstallAll (floder+"/tx/"+prefix
    +"tx.txt", ndn::CarRelayTracer::TX);
381
382 boost::tuple< boost::shared_ptr<std::ostream>, std::list<boost::
    shared_ptr<ndn::V2vTracer> > >
383 tracing = ndn::V2vTracer::InstallAll (floder+"/V2v/"+prefix+"V2v
    .txt");
384
385 boost::tuple< boost::shared_ptr<std::ostream>, std::list<boost::
    shared_ptr<ndn::CarRelayTracer> > >

```

```

386     tracing4 = ndn::CarRelayTracer::InstallAll (floder+"/in-cache/"+
           prefix+"in-cache.txt", ndn::CarRelayTracer::IN_CACHE);
387
388
389 ////////////////////////////////////////////////////////////////// the following should be put just
           before calling Simulator::Run in the scenario
390
391 boost::tuple< boost::shared_ptr<std::ostream>, std::list<Ptr<ndn::
           CsTracer> > >
392     aggTracers = ndn::CsTracer::InstallAll (floder+"/cs-trace/"+
           prefix+ "cs-trace.txt", Seconds (50));
393
394 boost::tuple< boost::shared_ptr<std::ostream>, std::list<Ptr<ndn::
           L3RateTracer> > >
395     rateTracers = ndn::L3RateTracer::InstallAll (floder+"/rate/"+
           prefix+"rate-trace.txt", Seconds (50));
396
397     boost::tuple< boost::shared_ptr<std::ostream>, std::list<Ptr<
           L2RateTracer> > >
398     l2tracers = L2RateTracer::InstallAll (floder+"/drop/"+prefix+"drop
           -trace.txt", Seconds (50));
399
400     boost::tuple< boost::shared_ptr<std::ostream>, std::list<Ptr<ndn::
           AppDelayTracer> > >
401     tracers = ndn::AppDelayTracer::InstallAll (floder+"/app-delays/"+
           prefix+"app-delays-trace.txt");
402
403     boost::tuple< boost::shared_ptr<std::ostream>, std::list<Ptr<ndn::
           L3AggregateTracer> > >
404     aggreTracers = ndn::L3AggregateTracer::InstallAll (floder+"/
           aggregate/"+prefix+"aggregate-trace.txt", Seconds (50));
405
406 //////////////////////////////////////////////////////////////////
407     Simulator::Stop (Seconds (duration));
408 //////////////////////////////////////////////////////////////////
409     NS_LOG_INFO ("Starting");
410
411     Simulator::Run ();
412
413     NS_LOG_INFO ("Done");
414
415     Simulator::Destroy ();
416
417     return 0;
418 }

```

ข โปรแกรมแพ็กเก็ตการร้องขอและโปรแกรมเมื่อได้รับแพ็กเก็ตการร้องขอ

ในส่วนนี้เป็นโปรแกรมที่ได้พัฒนาเพิ่มเติมประกอบไปด้วย

- แพ็กเก็ตการร้องขอและแพ็กเก็ตข้อมูล ปรับแก้ใน `ndn::InterHeader` และ `ndn::ContentObjectHeader` ตามลำดับ

```

1 void Requester::SendInterest ()
2 {
3     if (!m_isRunning) return;
4     Ptr<ndn::Name> prefix = Create<ndn::Name> (m_name); //
        another way to create name
5     // Create and configure ndn::Interest
6     ndn::Interest interestHeader;
7     UniformVariable rand (0, std::numeric_limits<uint32_t>::max ()
        );
8     interestHeader.SetNonce (rand.GetValue ());
9     interestHeader.SetName (prefix);
10    interestHeader.SetInterestLifetime (Seconds (3.0));
11    interestHeader.SetPosition (socket->GetNode()->GetId());
12    interestHeader.SetRSUID (socket->GetNode()->GetId());
13    // Create packet and add ndn::Interest
14    Ptr<Packet> packet = Create<Packet> ();
15    packet->AddHeader (interestHeader);
16    NS_LOG_DEBUG ("Sending Interest packet for " << *prefix);
17    // Forward packet to lower (network) layer
18    m_protocolHandler (packet);
19    // Call trace (for logging purposes)
20    m_transmittedInterests (&interestHeader, this, m_face);
21    Simulator::Schedule (Seconds (100.0), &Requester::
        SendInterest, this);
22 }
23 void Requester::OnContentObject (const Ptr<const ndn::
        ContentObject> &contentObject, Ptr<Packet> payload)
24 {
25     NS_LOG_DEBUG ("Receiving ContentObject packet for " <<
        contentObject->GetName ());
26 }

```

- กระบวนการทำงานเมื่อได้รับแพ็กเก็ตการร้องขอ

```

1 //-----Receive packet -----
2 static void ReceivePacket (Ptr<Socket> socket) //Receive
        function
3 {

```

```

4  uint32_t a = chk[socket->GetNode () ->GetId()]++; // received
    packet
5
6  message_received++;
7
8  Ptr<Packet> packet = socket->Recv(); // return Packet object
9  MyHeader destinationHeader;
10  packet->RemoveHeader (destinationHeader); // remove header
11
12  packetId[a] = destinationHeader.GetData(); // get data
13
14  double x_current = position[socket->GetNode ()->GetId ()
    ] [0] [0];
15  double y_current = position[socket->GetNode ()->GetId ()
    ] [0] [1];
16  double x_relay = relay[packetId[a]] [0] [0];
17  double y_relay = relay[packetId[a]] [0] [1];
18  double x_previous = position[socket->GetNode ()->GetId ()
    ] [1] [0];
19
20  dist[socket->GetNode ()->GetId ()] = sqrt((x_current-x_relay)
    *(x_current-x_relay)+(y_current-y_relay)*(y_current-
    y_relay));
21
22  double max_differ = 0.000011636; // Data rate= 11Mbps
23
24  differ[socket->GetNode()->GetId()] = max_differ*((transrange-
    dist[socket->GetNode ()->GetId ()])/transrange)+0.00006;
25
26  double direction;
27  direction = ((x_current-x_previous))*((incidence[0] [0]-
    incidence[1] [0]));
28  if (chk[socket->GetNode () ->GetId()] < 2)
29  {
30  if ( direction > 0 ) //same direction
31  {
32  if ( sqrt((x_current-incidence[0] [0])*(x_current-incidence
    [0] [0])) < sqrt((x_previous-incidence[0] [0])*(x_previous
    -incidence[0] [0]) ) ) // same direction moving toward
33  {
34
35  noticed_message++;
36  Simulator::Schedule (Seconds (differ[socket->GetNode()->
    GetId ()]), &Rebroadcast, socket);
37
38  }else {

```



```

39         irrelevance_message++;
40     }
41 }
42 else if ( direction < 0) //opposite direction
43 {
44     if (sqrt((x_current-incidence[0][0])*(x_current-incidence
45         [0][0])) > sqrt((x_previous-incidence[0][0])*(x_previous
46         -incidence[0][0])))
47     {
48         relay_message++;
49         Simulator::Schedule (Seconds (differ[socket->GetNode()->
50             GetId()]), &Rebroadcast, socket);
51     }else{
52         irrelevance_message++;
53     }
54 }
55 //-----Get Position -----
56 static Vector
57 GetPosition (Ptr<Node> node)
58 {
59     Ptr<MobilityModel> mobility = node->GetObject<MobilityModel>
60         ();
61     return mobility->GetPosition ();
62 }
63 //-----Hold Position -----
64 static void
65 HoldPosition (Ptr<Node> node)
66 {
67
68     position[node->GetId()][1][0] = position[node->GetId()
69         ][0][0];//shift index x
70     position[node->GetId()][1][1] = position[node->GetId()
71         ][0][1]; // shift index y
72
73     Vector pos = GetPosition (node);
74     position[node->GetId()][0][0] = pos.x;
75     position[node->GetId()][0][1] = pos.y;
76 }

```

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวศศิธรมย์ เทียนน้อย เกิดเมื่อวันที่ 15 มีนาคม พ.ศ. 2527 ที่จังหวัดนครนายก สำเร็จ การศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัย บูรพา เมื่อปีการศึกษา 2548 และหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2552 จากนั้นได้เข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรดุษฎีบัณฑิต ในปีการศึกษา 2554 ณ จุฬาลงกรณ์มหาวิทยาลัย โดยสังกัดห้องปฏิบัติ- การวิจัยโทรคมนาคม