



รายงานการวิจัยฉบับสมบูรณ์

การออกแบบระบบการทำงานด้วยวิธีการ Cross layer
โดยใช้โปรโตคอล SCTP เพื่อเพิ่มสมรรถนะให้กับ
เครือข่ายแบบ Ad hoc

ผู้วิจัย ดร.สกุณา เจริญปัญญาศักดิ์
ที่ปรึกษา พศ.ดร.วรรณรัช สันติอมรทัต

งานวิจัยนี้ได้รับทุนอุดหนุนการวิจัย จากเงินรายได้
คณะวิศวกรรมศาสตร์ ประจำปีงบประมาณ 2552

บทคัดย่อ

ลักษณะเด่นของเครือข่ายแบบ Ad hoc คือ โหนดทุกโหนดในเครือข่ายสามารถเคลื่อนที่ได้อย่างอิสระ ซึ่งเป็นสาเหตุให้ลักษณะของทอพอโลยีในเครือข่ายมีโครงสร้างที่ไม่แน่นอน รวมทั้งสามารถเปลี่ยนได้ตลอดเวลา จากคุณสมบัติของเครือข่ายดังกล่าว ทำให้เส้นทางการสื่อสารมีโอกาสที่จะขาดการติดต่อได้บ่อยครั้ง และเป็นผลให้เกิดการสูญหายของข้อมูลขึ้น บทความนี้ได้เสนอการวิเคราะห์เชิงเปรียบเทียบในโพรโทคอล SCTP multihoming กับการทำงานของโพรโทคอลการค้นหาเส้นทาง ในกรณีที่เส้นทางหลักเกิดปัญหาขาดการติดต่อไปยังปลายทาง และเสนอทอพอโลยีที่แสดงถึงผลกระทบต่อโพรโทคอลการค้นหาเส้นทางสำหรับเครือข่ายแบบ Ad hoc โดยการจำลองการทำงานผ่านโปรแกรม Network Simulation 2 (NS2) โพรโทคอลการค้นหาเส้นทางที่นำมาเปรียบเทียบในการทดลองนี้ได้แก่ DSDV DSR และ AODV จากผลการทดสอบพบว่า เวลาที่ใช้ในการค้นหาเส้นทางใหม่ในแต่ละโพรโทคอล จะขึ้นกับทอพอโลยีของเครือข่าย นอกจากนั้นกระบวนการที่เรียกว่า Real Multiroute System (RMS) ซึ่งเป็นวิธีการทำแบบ Cross layer ยังคงสามารถลดการสูญเสียของแพ็กเก็ต และเพิ่มสมรรถนะให้กับระบบ

คำสำคัญ: โพรโทคอลการค้นหาเส้นทาง ทอพอโลยี เครือข่ายไร้สายแบบ Ad hoc

ABSTRACT

The distinctive feature of wireless ad hoc network protocol is that every node in network can move randomly and freely so that the network topology may change rapidly and unpredictably. According to this characteristic of network, path loss and packet drop can easily occur. This paper proposes the network analysis of SCTP multihoming protocol and routing protocol in case of the main path loss and the network topologies for routing wireless ad hoc network. We prove these network topologies by using Network Simulation 2 (NS2) simulator. DSDV, AODV and DSR protocol are compared in the network. The results show that the time of the re-route discovery in each protocol is depended on network topologies. Moreover the Real Multiroute System (RMS) process that is the Cross layer method can reduce packet losses and improve the performance systems.

Keyword: Routing protocols, Topologies, Wireless Ad hoc Networks.

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จได้ ด้วยความกรุณาของ ผศ. ดร. วรณรัช สันติอมรทัต ที่ปรึกษางานวิจัย ซึ่งได้ให้คำแนะนำถึงประเด็นต่างๆ ในการศึกษา และชี้แนวทางในการแก้ไขปัญหา รวมทั้งแนวทางการค้นคว้าหาข้อมูลเพิ่มเติมอันเป็นประโยชน์อย่างยิ่งในการทำงานวิจัยนี้ นอกจากนี้ขอขอบพระคุณภาคีวิชาการวิศวกรรมคอมพิวเตอร์ ที่ให้สถานที่ทำงานวิจัยรวมถึงให้ยืมอุปกรณ์ที่ใช้ในการทำงานวิจัย

ขอขอบคุณ นายวาริช จันธิ นักศึกษาปริญญาโท ซึ่งมีส่วนช่วยในงานวิจัยทางด้านของการทดสอบทอพอโลยีแบบต่างๆ กับโปรโตคอลในระดับเครือข่าย

งานวิจัยนี้ดำเนินการภายใต้การสนับสนุนจากเงินรายได้คณะวิศวกรรมศาสตร์ ประเภททุนพัฒนานักวิจัย จากมหาวิทยาลัยสงขลานครินทร์ ในโครงการหมายเลข ENG-52-2-7-01-0039-S คณะวิจัยขอขอบพระคุณมา ณ ที่นี้

สารบัญ

| | |
|--|------------|
| บทคัดย่อ | II |
| ABSTRACT | III |
| กิตติกรรมประกาศ | IV |
| สารบัญ | V |
| สารบัญรูปภาพ | VII |
| สารบัญตาราง | IX |
| บทที่ 1 | 1 |
| บทนำ | 1 |
| 1.1 วัตถุประสงค์ | 2 |
| 1.2 ขอบเขตการวิจัย | 2 |
| 1.3 วิธีการแบบ Cross layer และแนวทางการดำเนินงานเบื้องต้น | 3 |
| 1.4 ระเบียบวิธีการวิจัย | 4 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับจากผลการวิจัย | 5 |
| บทที่ 2 | 6 |
| ความรู้พื้นฐาน | 6 |
| 2.1 เครือข่ายเคลื่อนที่แบบ Ad hoc..... | 6 |
| 2.2 โพรโทคอลการค้นหาเส้นทางในเครือข่ายเคลื่อนที่แบบ Ad hoc | 7 |
| 2.2.1 Destination Sequenced Distance Vector (DSDV) | 7 |
| 2.2.2 Ad-hoc On-Demand Distance Vector (AODV) | 8 |
| 2.2.3 Dynamic Source Routing (DSR)..... | 9 |
| 2.3 มาตรฐาน IEEE 802.11 | 10 |
| 2.4 สถาปัตยกรรม IEEE 802.11 | 11 |
| บทที่ 3 | 15 |
| ระบบการทำงานแบบ Cross layer | 15 |

| | |
|---|-----------|
| 3.1 รูปแบบของการสื่อสารแบบ Cross layer | 15 |
| 3.2 การออกแบบ Real Multiroute System (RMS)..... | 16 |
| 3.3 การเริ่มต้นการทำงานของระบบ..... | 17 |
| 3.4 ระบบเกิดปัญหาการสูญหายของข้อมูล..... | 18 |
| บทที่ 4 | 20 |
| ผลการวิจัย | 20 |
| 4.1 ทดสอบการทำงานของทอพอโลยี | 20 |
| 4.2 ทดสอบการทำ Cross layer ด้วยระบบ RMS..... | 29 |
| บทที่ 5 | 31 |
| สรุปผลการวิจัย และข้อเสนอแนะ..... | 31 |
| 5.1 สรุปผลการวิจัย | 31 |
| 5.2 ข้อเสนอแนะ | 31 |
| บรรณานุกรม | 32 |
| ภาคผนวก ก..... | 35 |
| ภาคผนวก ข..... | 38 |
| ภาคผนวก ค..... | 42 |

สารบัญรูปภาพ

| | |
|--|----|
| รูปที่ 1 สถาปัตยกรรมของการสื่อสารแบบ Cross layer..... | 3 |
| รูปที่ 2 Cross layer ระหว่างระดับชั้น Transport และ Network..... | 4 |
| รูปที่ 3 การหาเส้นทางการส่งข้อมูลบนเครือข่ายไร้สายแบบ Ad hoc..... | 6 |
| รูปที่ 4 การหาเส้นทางของโพรโทคอล DSDV | 8 |
| รูปที่ 5 การหาเส้นทางของโพรโทคอล AODV | 9 |
| รูปที่ 6 การหาเส้นทางของโพรโทคอล DSR | 9 |
| รูปที่ 7 เปรียบเทียบลักษณะตารางเก็บค่าเส้นทางที่แตกต่างกัน..... | 10 |
| รูปที่ 8 แสดงสถาปัตยกรรมของ IEEE 802.11 | 11 |
| รูปที่ 9 แสดงขนาดของช่องว่างระหว่างเฟรมต่างๆของมาตรฐาน 802.11 | 12 |
| รูปที่ 10 แสดงขั้นตอนการส่งข้อมูลในการควบคุมใช้สื่อบนแบบ DCF แบบ 2-way handshake | 12 |
| รูปที่ 11 การเข้าใช้สื่อบนของโพรโทคอล CSMA/CA ที่ใช้ RTS/CTS..... | 13 |
| รูปที่ 12 ปัญหาการชนกันของข้อมูลเนื่องจากสถานีที่ซ่อนอยู่ (Hidden terminal)..... | 14 |
| รูปที่ 13 หลักการทำงานของ CSMA/CA | 14 |
| รูปที่ 14 การสื่อสารโดยตรงระหว่างระดับชั้นสื่อสาร | 15 |
| รูปที่ 15 การสื่อสารแบบส่งข้อมูลถึงกัน..... | 16 |
| รูปที่ 16 การส่งข้อมูลด้วยโหนดเพื่อนบ้านที่ต่างกัน | 16 |
| รูปที่ 17 การส่งข้อมูลด้วยโหนดเพื่อนบ้านที่เหมือนกัน..... | 16 |
| รูปที่ 18 การทำงานของ RMS | 17 |
| รูปที่ 19 กระบวนการเริ่มต้นการทำงาน | 17 |
| รูปที่ 20 การสื่อสารที่เรียกว่า Multihop..... | 18 |
| รูปที่ 21 เส้นทางหลักขาดการติดต่อสื่อสาร | 18 |
| รูปที่ 22 การแสดงปัญหาที่เกิดขึ้นในระดับ MAC..... | 19 |
| รูปที่ 23 แสดงทอพอโลยีรูปแบบที่ 1..... | 21 |
| รูปที่ 24 แสดงทอพอโลยีรูปแบบที่ 2..... | 21 |
| รูปที่ 25 แสดงทอพอโลยีรูปแบบที่ 3..... | 21 |
| รูปที่ 26 แผนภูมิเปรียบเทียบค่า Delay..... | 23 |
| รูปที่ 27 แผนภูมิเปรียบเทียบค่า Routing Packet | 24 |
| รูปที่ 28 แผนภูมิเปรียบเทียบค่า Packet Delivery Ratio | 24 |
| รูปที่ 29 เปรียบเทียบ Instants Throughput ของโพรโทคอล AODV ที่มีทอพอโลยีแตกต่างกัน..... | 24 |
| รูปที่ 30 เปรียบเทียบ Instants Throughput ของ DSR โพรโทคอลที่มีทอพอโลยีแตกต่างกัน | 25 |
| รูปที่ 31 เปรียบเทียบ Instants Throughput ของ DSDV โพรโทคอลที่มีทอพอโลยีแตกต่างกัน | 25 |
| รูปที่ 32 แสดงการเคลื่อนที่ออกของโหนดเชื่อมต่อในทอพอโลยีรูปแบบที่ 2 | 27 |
| รูปที่ 33 แสดงการเคลื่อนที่ออกของโหนดเชื่อมต่อในทอพอโลยีรูปแบบที่ 3 | 28 |
| รูปที่ 34 การทำ Local Repair | 28 |

| | |
|--|----|
| รูปที่ 35 เปรียบเทียบค่าสมรรถนะของระบบ | 29 |
| รูปที่ 36 จำนวนการส่งแพ็กเก็ตเข้า | 30 |
| รูปที่ 37 อัตราการสูญหายของข้อมูล | 30 |

สารบัญตาราง

| | | |
|------------|---|----|
| ตารางที่ 1 | ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 1 | 22 |
| ตารางที่ 2 | ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 2 | 22 |
| ตารางที่ 3 | ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 3 | 23 |

บทที่ 1

บทนำ

ระบบเครือข่ายการทำงานแบบ Ad hoc หรือเรียกว่าระบบ MANETs (Mobile Ad hoc NETworks) เป็นการจัดตั้งขึ้นโดยกลุ่ม Internet Engineering Task Force (IETF) [1] โดยที่อุปกรณ์การทำงานแต่ละตัวเป็นอุปกรณ์เคลื่อนที่หรือไร้สาย เช่น โทรศัพท์มือถือ และเครื่องโน้ตบุ๊ก เป็นต้น ซึ่งรูปแบบสถาปัตยกรรมจะมีการเปลี่ยนแปลงอยู่ตลอดเวลา ดังนั้นเครือข่ายจึงมีรูปแบบที่ไม่แน่นอน หรือ Dynamic topology ซึ่งระบบเครือข่ายแบบ Ad hoc ใช้สำหรับการสื่อสารข้อมูลจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่งโดยไม่ใช้สถานีฐาน (Base station) หมายความว่า ตัวอุปกรณ์แต่ละตัวสามารถสื่อสารกันโดยตรงโดยไม่ต้องผ่านสถานีฐาน ซึ่งอุปกรณ์แต่ละตัวทำหน้าที่เป็นทั้งตัวส่งหรือรับข้อมูล รวมไปถึงการทำหน้าที่ในตำแหน่งที่เป็นเราเตอร์ (router) ในกรณีที่ผู้ส่งข้อมูลไม่สามารถติดต่อกับผู้รับโดยตรง และจะต้องส่งข้อมูลผ่านตัวกลาง (Intermediate node) ซึ่งจะทำหน้าที่เสมือนเป็นเราเตอร์เพื่อให้ข้อมูลดังกล่าวไปถึงปลายทางได้ หรือเรียกเป็นการสื่อสารแบบ Multihop

จากวิวัฒนาการแรกของงานระบบเครือข่ายไร้สายหรือ Single hop ในขั้นแรกตัวอุปกรณ์แต่ละตัวจะต้องส่งข้อมูลไปยังสถานีฐาน แล้วสถานีฐานจะทำการส่งข้อมูลต่อไปยังปลายทาง โดยระบบนี้จะมีโครงสร้างที่แน่นอน (Infrastructure) ต่อมาในปัจจุบันระบบเครือข่ายไร้สายแบบ Multihop ได้เริ่มเข้ามา มีบทบาทมากขึ้น และมีนักวิจัยจำนวนมากให้ความสนใจที่จะจัดการกับปัญหาสำหรับงานระบบเครือข่ายไร้สาย เพื่อพัฒนาสมรรถนะให้ดีขึ้น [2, 3]

ส่วนใหญ่ปัญหาของเครือข่ายไร้สาย เกิดจากคุณสมบัติส่วนตัวของลักษณะอุปกรณ์ โดยอุปกรณ์แต่ละตัว จะเคลื่อนที่อยู่ตลอดเวลา เป็นเหตุให้เกิดปัญหาขาดการติดต่อของเส้นทางการส่งข้อมูลจากต้นทางไปยังปลายทางเกิดขึ้นบ่อยครั้ง รวมทั้งข้อมูลที่ส่งไปอาจเกิดการสูญหายได้ เป็นผลให้สมรรถนะของระบบลดลง ระบบเครือข่ายจากกรณีดังกล่าวจะมีผลต่อโพรโทคอลในระดับที่สูงกว่าระดับเครือข่าย (Network layer) ซึ่งได้แก่ ระดับชั้น Transport โดยลักษณะของโพรโทคอลในระดับนี้ จะออกแบบเพื่อปรับปรุงปัญหาความคับคั่งของการส่งข้อมูลและเหมาะสมสำหรับระบบเครือข่ายแบบมีสาย ยกตัวอย่างเช่น โพรโทคอล Transmission Control Protocol (TCP) หรือโพรโทคอล Stream Control Transmission Protocol (SCTP) ดังนั้นจึงมีการคิดค้นวิธีการเพิ่มสมรรถนะให้กับระบบ เช่น การทำ Routing optimization หรือพัฒนาจากโพรโทคอลมาตรฐาน TCP ยกตัวอย่างเช่น TCP Westwood [4] หรือ TCP Sack [5] เพื่อให้สามารถใช้งานได้ดีกับระบบเครือข่ายไร้สาย

สำหรับโครงการนี้ ได้สังเกตเห็นถึงปัญหาหลักสำหรับระบบเครือข่ายแบบ Ad hoc กล่าวคือ เมื่ออุปกรณ์หนึ่งไม่สามารถติดต่อกับอีกอุปกรณ์หนึ่ง โพรโทคอลในระดับเครือข่ายแบบ Ad hoc เช่นโพรโทคอล Ad hoc On Demand Distance Vector (AODV) [6] จะทำการค้นหาเส้นทางใหม่ แทนเส้นทางเก่า

ที่ไม่สามารถติดต่อกันระหว่างต้นทางและปลายทางได้ แต่ในระบบที่ระดับโพรโทคอล Transport จะคิดว่าขณะนี้บัฟเฟอร์หรือขนาดเก็บข้อมูลที่ปลายทางเต็ม และทำการลดอัตราการส่งข้อมูลที่ต้นทางแทน ด้วยปัญหานี้ระบบจะวิเคราะห์การทำงานไม่ถูกต้อง เพราะในระหว่างทำการค้นหาเส้นทางใหม่ให้กับระบบ ต้นทางจะต้องไม่ลดขนาดการส่งหรือ Window size ที่ระดับโพรโทคอล Transport เพื่อแก้ไขปัญหาดังกล่าว โครงการนี้ขอเสนอวิธีการทำที่เรียกว่า Cross layer [7,8,9]

จากรูปแบบการทำงานของเครือข่าย OSI (Open Systems Interconnection) ซึ่งมีทั้งหมด 7 ระดับ โดยลักษณะการทำงานในโพรโทคอลระดับ Transport อยู่ระดับ 4 จะไม่สามารถทำการติดต่อกับโพรโทคอลที่สูงกว่าหรือต่ำกว่าระดับ Transport ได้ ยกตัวอย่างเช่น มีการชนกันของข้อมูล (Collisions) ที่ระดับ MAC (Medium Access Control) ซึ่งอยู่ระดับ 2 หรือ มีการค้นหาเส้นทางใหม่เมื่อเส้นทางเก่าขาดการติดต่อที่ระดับโพรโทคอล Network ระดับ 3 ในโครงการนี้จึงเสนอการใช้ข้อมูลร่วมกันระหว่างโพรโทคอล อย่างเช่นการทำงานระหว่างโพรโทคอลระดับ 2 กับ 4 หรือ โพรโทคอลระดับ 3 กับ 4 โดยใช้วิธีการที่เรียกว่า Cross layer

โครงการนี้ จะใช้โพรโทคอล SCTP ซึ่งเป็นโพรโทคอลระดับ Transport โดยคุณสมบัติของโพรโทคอลตัวนี้มีลักษณะคล้ายกับ TCP แต่ SCTP มีคุณสมบัติการทำงานซึ่ง TCP ไม่มีก็คือ จะมีการส่งข้อมูลทั้งแบบ Multistreaming และ Multihoming ดังนั้นจึงเป็นเหตุผลหลักที่นำ SCTP มาใช้ในโครงการนี้ และนอกจากนี้จะนำเอาวิธีการแบบ Cross layer มาใช้งานเพิ่มสมรรถนะการทำงานให้กับระบบเครือข่ายแบบ Ad hoc

1.1 วัตถุประสงค์

1. ศึกษาถึงความสำคัญของโพรโทคอลระดับ Routing ที่มีผลต่อสมรรถนะในโพรโทคอลระดับ Transport ในเครือข่ายการทำงานแบบ Ad hoc
2. ออกแบบ และทดสอบอัลกอริทึม ที่มีผลต่อการทำงานของเครือข่าย Ad hoc
3. เพื่อพัฒนาสมรรถนะการทำงานของเครือข่ายแบบ Ad hoc โดยใช้โพรโทคอล SCTP
4. เพื่อเป็นการนำร่องการนำวิธีการที่เรียกว่า Cross layer ไปประยุกต์ใช้ในระบบการทำงานจริง
5. ผลิตงานวิจัยเชิงวิชาการที่สามารถออกสู่เวทีในระดับชาติ และนานาชาติได้

1.2 ขอบเขตการวิจัย

1. ศึกษาความน่าจะเป็นในการเพิ่มสมรรถนะในเครือข่ายแบบ Ad hoc ที่ทำงานร่วมกับโพรโทคอล SCTP โดยจะแสดงถึงความสำคัญของโพรโทคอลในระดับ 3 ซึ่งมีผลต่อสมรรถนะของโพรโทคอลในระดับ 4 รวมไปถึงความสำคัญในการทำ Cross layer
2. ออกแบบอัลกอริทึมเพื่อรองรับเครือข่ายแบบ Ad hoc ยกตัวอย่างเช่น ในกรณีที่อุปกรณ์ในเครือข่ายมีการเคลื่อนที่ออกจากรัศมีที่สามารถส่งข้อมูลได้ และไม่มีเส้นทางเชื่อมต่อจากต้นทางไปยังปลายทาง โดยโครงการวิจัยนี้จะใช้ข้อมูลเช่นจำนวนแพ็กเก็ตที่สูญหายของโพรโทคอลในระดับ

3. ควบคุมการทำงานของโพรโทคอลในระดับ 4 และทดสอบการทำงานโดยใช้โปรแกรม Network Simulation

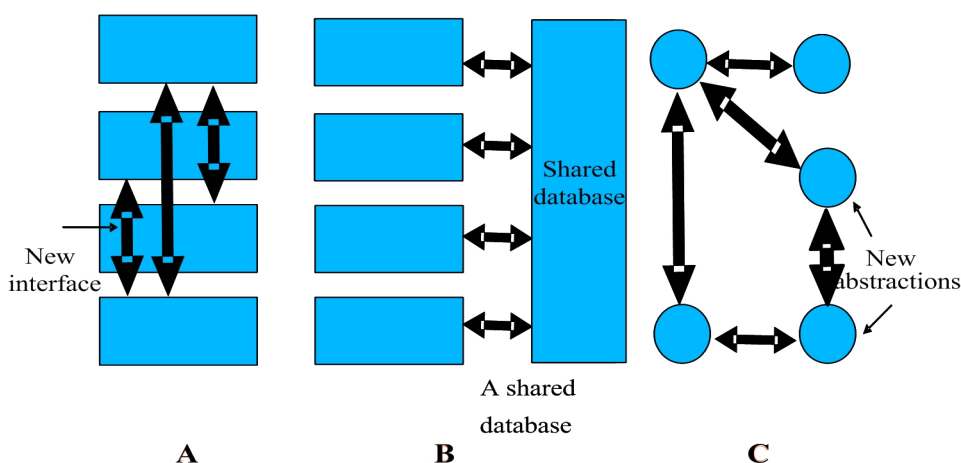
1.3 วิธีการแบบ Cross layer และแนวทางการดำเนินงานเบื้องต้น

1. สถาปัตยกรรมของการสื่อสารแบบ Cross layer

วิธีการแบบ Cross layer เป็นการแลกเปลี่ยนข้อมูลกันระหว่างระดับชั้นของการสื่อสาร ซึ่งอาจจะเป็นการสื่อสารกับระดับชั้นที่ไม่ติดกัน นั่นก็คือ สิ่งที่ต้องการเป็นการใช้ข้อมูลที่มีอยู่ในแต่ละระดับชั้น มาประยุกต์ใช้เพื่อให้ระบบมีประสิทธิภาพ สถาปัตยกรรมแบบนี้เป็นการสร้างกลไกการทำงาน เพื่อให้เกิดความยืดหยุ่น ซึ่งแบ่งการทำงานเป็น 3 รูปแบบ [10]

แบบที่ 1 การสื่อสารกันโดยตรงระหว่างระดับชั้น

วิธีการนี้เพื่อให้โพรโทคอลสามารถสื่อสารกันโดยตรงระหว่างระดับชั้น ซึ่งอาจจะเป็นระดับชั้นที่ติดกัน หรือห่างกัน แสดงดังรูปที่ 1A อินเตอร์เฟซที่เกิดขึ้นมาใหม่นี้จะถูกใช้ในการแลกเปลี่ยนข้อมูลกันระหว่างระดับชั้น



รูปที่ 1 สถาปัตยกรรมของการสื่อสารแบบ Cross layer

แบบที่ 2 การสื่อสารกันโดยผ่านทางสื่อกลาง

จากรูปที่ 1B เป็นการแลกเปลี่ยนข้อมูลผ่านทางสื่อกลาง หรือเรียกว่า Database ซึ่งทางสื่อกลางจะเป็นตัวรับ และส่งข้อมูลระหว่างระดับชั้นที่เกิดขึ้นในระบบที่ได้ออกแบบ

แบบที่ 3 การสื่อสารโดยใช้โมดูลใหม่

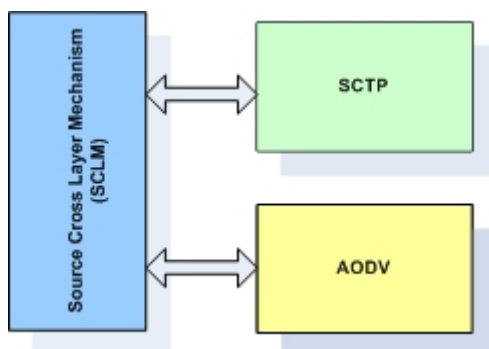
จากรูปที่ 1C แสดงการจัดการโพรโทคอลอีกวิธีการหนึ่ง ซึ่งไม่เพียงจะพิจารณาจากข้อมูลระหว่างระดับชั้นของการสื่อสาร และยังคงสื่อสารกันเองระหว่างโมดูล (New abstractions)

2. แนวทางการดำเนินงานเบื้องต้น

-จากสถาปัตยกรรมที่ได้กล่าวมาข้างต้น นักวิจัยในปัจจุบันจะนำวิธีการใดมาใช้ นั่น ก็ต้องพิจารณาจากข้อมูลของการสื่อสารระหว่างระดับชั้น เช่นในเบื้องต้นของโครงการนี้จะพิจารณาจากการนำ

ข้อมูลของจำนวนการสูญหายของแพ็กเก็ต ซึ่งอยู่ที่ระดับชั้น Network มาทำ Cross layer กับระดับชั้น โพรโทคอล Transport จึงคิดว่าจะออกแบบการสื่อสารโดยผ่านทางสื่อกลาง หรือแบบที่ 2 เพื่อเก็บ จำนวนการสูญหายของแพ็กเก็ตไว้ใน Database

-จำลองการทำงานโดยใช้ จำนวนโหนด (Mobile nodes) ที่ประมาณ 10 โมบายล์โหนด เพื่อดู จำนวนการสูญหายของแพ็กเก็ต ดังรูปที่ 2 แสดงการรูปแบบการทำงานกันระหว่าง 2 ระดับชั้น โดยโพรโทคอลที่ใช้ในระดับ Network คือ AODV (Ad hoc On Demand Distance Vector) และระดับชั้นโพรโทคอล Transport ใช้โพรโทคอล SCTP



รูปที่ 2 Cross layer ระหว่างระดับชั้น Transport และ Network

1.4 ระเบียบวิธีการวิจัย

ลำดับขั้นตอนระเบียบวิธีวิจัยสามารถแจกแจงได้ดังนี้

1. สืบค้นข้อมูลเพื่อค้นหา/ทดสอบแนวคิด
สืบค้นเทคนิคต่างๆ ที่เคยนำเสนอจากนักวิจัยใน และต่างประเทศ
2. การออกแบบระบบโดยใช้การสูญหายของแพ็กเก็ต มาใช้ในรูปแบบที่เราเรียกว่า Cross layer ในขั้นตอนนี้จะเป็นการทดสอบแนวคิดที่ได้นำเสนอๆ ไปแล้ว โดยทดสอบหลักการคร่าวๆ ด้วยการจำลองการทำงานโดยใช้ โปรแกรม Network Simulation
3. การพัฒนาทฤษฎีการออกแบบและการตรวจสอบ (ถ้าหากจำนวนการสูญหายของแพ็กเก็ต ทำให้สมรรถนะของระบบดีขึ้นเล็กน้อย อาจจะมีการพิจารณาเพิ่มเติมจากจำนวนการชนกันของแพ็กเก็ตที่ระดับชั้น MAC)
4. การจำลองการทำงาน และทดสอบ
ทดสอบการทำงานโดยจำลองเครือข่ายแบบ Ad hoc ที่ออกแบบไว้โดยใช้โปรแกรม Network Simulation
5. การตรวจสอบทฤษฎี (นำเทคนิคที่เรียกว่า Multihoming มาใช้ในการออกแบบ หมายความว่า จะมี Network cards มากกว่า 1 ตัว ต่อหนึ่งคอมพิวเตอร์ ซึ่งวิธีการนี้อยู่ที่ระดับ Transport และโครงการนี้จะนำมาทำ Cross layer กับระดับ network) โดยคิดว่าเมื่อได้รับข้อมูลเกี่ยวกับการสูญหาย (RERR: Route ERRor) จากระดับ network จะนำข้อมูลนี้ไปใช้เพื่อเปลี่ยนเส้นทางการติดต่อทันที แทนการรอคอยการ drop ของแพ็กเก็ต

6. การทดสอบในขั้นการจำลองการทำงาน และปรับแต่งสมรรถนะ
ปรับอัลกอริทึมเพื่อให้ได้สมรรถนะตามที่ต้องการ และทดสอบการทำงานอีกครั้ง
7. การจำลองการทำงาน ทดสอบ และวิเคราะห์ผล
วิเคราะห์ผลการทดสอบ และเสนอวิธีการใหม่ รวมถึงผลการจำลองการทำงานที่ได้สามารถ
ประยุกต์ใช้ในงานจริงต่อไป
8. สรุปงานและเขียนบทความ
เขียนบทความและรายงานฉบับสมบูรณ์เพื่อสรุปผลลัพธ์และข้อเสนอแนะในการทำวิจัยต่อไป

1.5 ประโยชน์ที่คาดว่าจะได้รับจากผลการวิจัย

1. สามารถผลิตงานวิจัยเชิงวิชาการเพื่อออกสู่เวทีในระดับชาติ และนานาชาติได้
2. อัลกอริทึมที่ออกแบบสามารถนำมาประยุกต์ใช้กับระบบการใช้งานจริงได้
3. ระบบที่ได้ สามารถนำไปใช้กับเครือข่ายอื่นๆ ได้เช่น เครือข่าย Vehicular Ad-Hoc Networks (VANETs) หรือเป็นการติดตั้งไว้ในรถที่มีการเคลื่อนที่
4. ผลการทดสอบเป็นประโยชน์ต่อการเรียนการสอนในระดับปริญญาโท และปริญญาเอกต่อไป

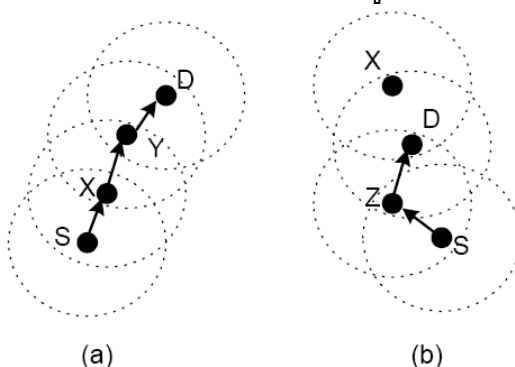
บทที่ 2

ความรู้พื้นฐาน

2.1 เครือข่ายเคลื่อนที่แบบ Ad hoc

เครือข่ายเคลื่อนที่แบบ Ad hoc [15] เริ่มมีการศึกษาคิดค้นตั้งแต่ช่วงกลางทศวรรษที่ 1960 ซึ่งแรกเริ่มใช้ชื่อว่า Packet Radio Network แต่ได้หยุดการพัฒนาไปในช่วงทศวรรษที่ 1980 เนื่องจากเทคโนโลยีด้านหน่วยประมวลผลและหน่วยความจำยังมีราคาแพงมาก ปัจจุบันปัญหาที่ว่านั้นได้หมดไป แนวความคิดเรื่องเครือข่ายเคลื่อนที่แบบ Ad hoc กลับมาได้รับความสนใจและมีการพัฒนาอีกครั้งอย่างต่อเนื่องตั้งแต่ปี 1995

เครือข่ายเคลื่อนที่แบบ Ad hoc หรือกลุ่มของโหนดเคลื่อนที่ไร้สาย (Wireless Mobile Nodes) รวมกันก่อให้เกิดเป็นเครือข่ายอิสระที่มีโครงสร้างการเชื่อมโยงแบบไร้สายทั้งหมด และมีลักษณะการทำงานแบบ Dynamic Autonomous Network โหนดต่าง ๆ จะทำการติดต่อสื่อสารระหว่างกันได้โดยไม่ต้องผ่านการเชื่อมโยงที่สถานีฐาน (Base Stations) หรือแอคเซสพอยต์ (Access Points) ซึ่งก็หมายความว่า โหนดแต่ละโหนดจะเป็นได้ทั้งเราท์เตอร์ (Router) และโฮสต์ (Host) และเนื่องจากโหนดเคลื่อนที่ไร้สายมีข้อจำกัดในการส่งสัญญาณติดต่อกันภายในระยะทางช่วงหนึ่ง ดังนั้น การติดต่อสื่อสารระหว่างโหนดต่าง ๆ ที่อยู่เครือข่ายจะมีลักษณะที่เรียกว่า Multihop และจากรูปที่ 3 จะเห็นว่าภายใน MANETs มีการติดต่อสื่อสารระหว่างโหนด S และโหนด D โดยโหนด S ส่งข้อมูลไปยังโหนด D การเชื่อมโยงระหว่างโหนด เป็นดังนี้ $S > X > Y > D$ (ดังรูป 3 (a)) และในเวลาต่อมา โหนดต่าง ๆ มีการเคลื่อนที่ การเชื่อมโยงระหว่างโหนดเปลี่ยนเป็น $S > Z > D$ (ดังรูป 3 (b)) เป็นต้น



รูปที่ 3 การหาเส้นทางการส่งข้อมูลบนเครือข่ายไร้สายแบบ Ad hoc

เครือข่ายเคลื่อนที่แบบ Ad hoc มีข้อได้เปรียบกว่าเครือข่ายไร้สายแบบอื่น ๆ เช่น เครือข่ายสามารถสร้างขึ้นได้อย่างรวดเร็ว ไม่ยุ่งยากซับซ้อน ลดการก่อสร้างในส่วนของเครือข่ายที่ต้องติดตั้งสถานีฐาน ซึ่งทำให้ใช้งบประมาณในการสร้างเครือข่ายลดลง ดังนั้นเครือข่ายเคลื่อนที่แบบ Ad hoc

สามารถสร้างขึ้นได้ทันทีโดยไม่ต้องมีโครงสร้างของสถานีฐาน และไม่ต้องมีผู้ดูแลระบบเครือข่าย สำหรับเครือข่ายเคลื่อนที่แบบ Ad hoc สามารถนำไปประยุกต์ใช้กับระบบต่าง ๆ ได้อย่างเหมาะสมได้แก่ ใช้ติดต่อส่วนบุคคล ซึ่งอุปกรณ์ที่ใช้ติดต่อมีดังนี้ เครื่องโทรศัพท์มือถือ คอมพิวเตอร์พกพา และเครื่อง PDA หรือในกรณีใช้ติดต่อภายในกลุ่มเช่น สร้างเครือข่ายติดต่อสื่อสารภายในงานแสดงนิทรรศการ งานสัมมนาต่างๆ ใช้ในงานด้านการทหาร หรือใช้ในกรณีฉุกเฉินต่างๆ เช่น กรณีที่เกิดภัยพิบัติ หรือใช้เป็นเครือข่ายสื่อสารภายในเมืองเป็นต้น จากการนำไปประยุกต์ดังกล่าวข้างต้น เครือข่ายเคลื่อนที่แบบ Ad hoc ซึ่งปัจจุบันยังคงมีการพัฒนาระบบอย่างต่อเนื่อง และหาทางแก้ไขปัญหาต่างๆ สำหรับการทำงานของเครือข่ายเคลื่อนที่แบบ Ad hoc มีข้อจำกัดที่สำคัญได้แก่

- ทอพอโลยีแบบพลวัต (Dynamic Topology) โหนดต่างๆ เคลื่อนที่ได้ และสามารถติดต่อกับโหนดใดๆ ก็ได้ในเครือข่าย ดังนั้นการเชื่อมโยงโหนดต่างๆ ในเครือข่ายเป็นแบบไม่ถาวร มีการเปลี่ยนแปลงไปตามเวลา ขึ้นอยู่กับตำแหน่งของโหนดต่างๆ ณ เวลาขณะใดขณะหนึ่ง ซึ่งทำให้เกิดหัวข้อในการแก้ปัญหาเรื่องขาดการติดต่อสื่อสารเมื่อโหนดมีการเคลื่อนที่

- ข้อจำกัดการใช้แถบความถี่ ความจุในการส่งสัญญาณข้อมูลทางการเชื่อมโยงไร้สายมีน้อยกว่าการเชื่อมโยงโดยใช้สายเป็นอย่างมาก และการเชื่อมโยงไร้สายยังมีปัญหาในเรื่องของสัญญาณรบกวนทำให้สัญญาณข้อมูลด้านรับมีคุณภาพลดลง

- ข้อจำกัดทางด้านการรักษาความปลอดภัย เครือข่ายที่โหนดมีการเคลื่อนที่และมีทอพอโลยีแบบพลวัต มีความเสี่ยงด้านความปลอดภัยของข้อมูลมากกว่าเครือข่ายแบบถาวร เนื่องจากอุปกรณ์สื่อสารที่โหนดหรือของผู้ใช้อาจถูกขโมยได้ หรือสัญญาณข้อมูลอาจถูกส่งไปในการเชื่อมโยงไร้สายที่ไม่ปลอดภัยเป็นต้น ดังนั้นจะต้องมีการป้องกันปัญหาในเรื่องการลักลอบหรือดักสัญญาณข้อมูล และปัญหาด้านความปลอดภัยของเครือข่าย

- ข้อจำกัดด้านพลังงาน พลังงานที่ใช้ในโหนดเคลื่อนที่นั้นก็คือแบตเตอรี่ ดังนั้น การออกแบบระบบเครือข่ายเคลื่อนที่แบบ Ad hoc จะต้องเกี่ยวข้องกับเทคโนโลยีด้านการอนุรักษ์พลังงานด้วย

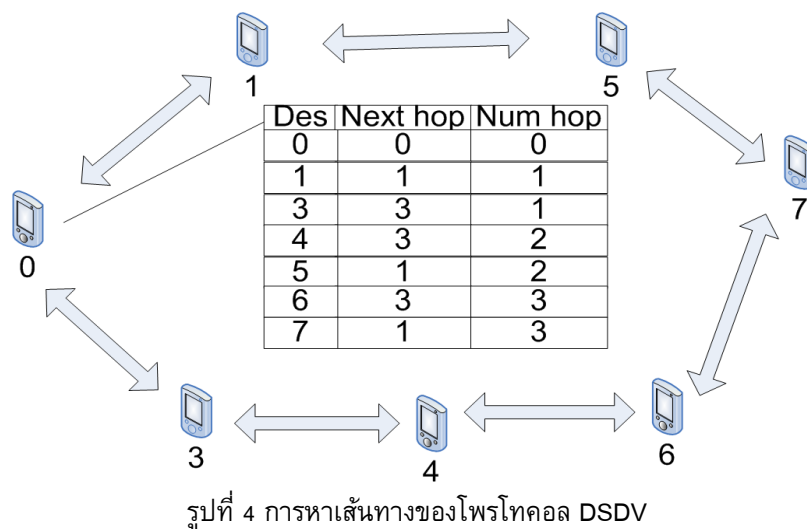
- เครือข่ายอิสระคือ ไม่มีการควบคุมสั่งการ หรือการจัดการจากเครือข่ายที่เป็นส่วนกลาง เครือข่ายเคลื่อนที่แบบ Ad hoc สามารถดำเนินการต่างๆ ด้วยตัวเองได้ ดังนั้นการออกแบบระบบจะต้องมีการเพิ่มความสามารถในการจัดการของโพรโทคอลการค้นหาเส้นทาง

2.2 โพรโทคอลการค้นหาเส้นทางในเครือข่ายเคลื่อนที่แบบ Ad hoc

2.2.1 Destination Sequenced Distance Vector (DSDV)

เป็นโพรโทคอลการจัดเส้นทางแบบเวกเตอร์บอกระยะ ที่มีลักษณะเป็นช่วงเชื่อมโยงต่อช่วงเชื่อมโยง (hop-by-hop) โดยแต่ละโหนดการเชื่อมต่อ มีตารางเก็บค่าเส้นทางไปยังโหนดปลายทางของแต่ละโหนดในระบบเครือข่าย จะเก็บค่าในเส้นทางไปยังโหนดปลายทางด้วยช่วงเชื่อมโยงต่อไป (Next hop) ดังรูปที่ 4 และจำนวนช่วงเชื่อมโยงที่ใช้ในเส้นทางนี้เหมือนกับเวกเตอร์บอกระยะ โพรโทคอลชนิดนี้มีการปรับปรุงข้อมูลในตารางเส้นทางด้วยการแพร่สัญญาณ (Broadcast) เป็นช่วงๆ ซึ่ง DSDV [11]

รับประกันว่าจะไม่เกิดเส้นทางวนรอบเพื่อไม่เกิดเส้นทางซ้ำกัน DSDV ใช้ตัวเลขลำดับเพิ่มในข้อมูลเส้นทางแต่ละเส้น โพรโทคอล DSDV ทำการปรับข้อมูลเส้นทางโดยการรับข้อมูลที่แพร่สัญญาณมาเป็นช่วงๆ จากสถานีโหนดข้างเคียง เวลาที่ข้อมูลทั้งหมดจากสถานีเชื่อมโยงจากข้างเคียง ทุกสถานีเชื่อมโยงได้มาครบแล้ว จึงมาประเมินหาค่าเส้นทางได้

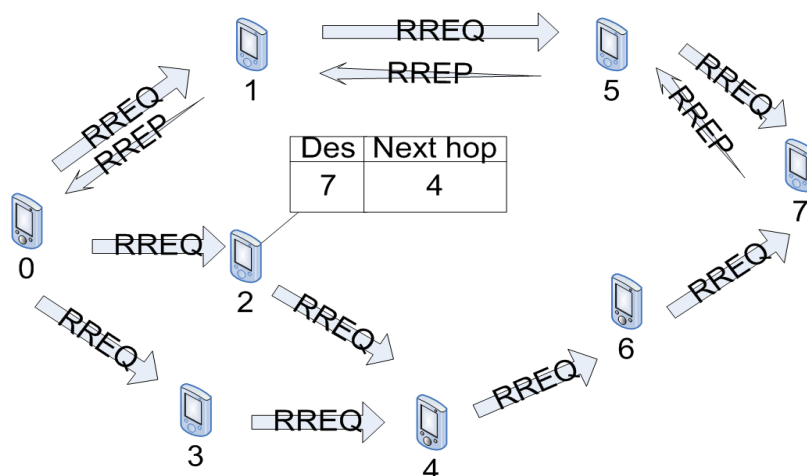


2.2.2 Ad-hoc On-Demand Distance Vector (AODV)

โพรโทคอล AODV [12] เป็นโพรโทคอลการจัดเส้นทางในเน็ตเวิร์คไร้สายแบบ Ad hoc ทำให้สถานีเชื่อมโยงสามารถติดต่อกันได้ โดยที่เส้นทางอาจมีหลายช่วงการเชื่อมต่อ โพรโทคอลมีพื้นฐานมาจากโพรโทคอลเวกเตอร์ระยะ (Distance Vector) แต่ AODV จะมีการทำงานเป็นแบบรีแอกทีฟ คือ ขบวนการค้นหาเส้นทางเกิดขึ้นเมื่อมีการร้องขอใช้เส้นทางนั้นเท่านั้น และสถานีเชื่อมโยงไม่จำเป็นต้องทำการปรับปรุงข้อมูลเส้นทางไปยังสถานีเชื่อมโยงปลายทางที่ยังไม่ใช้งานในขณะนั้น และในขณะการสื่อสารดำเนินอยู่ โดยเส้นทางยังทำงานได้ AODV ก็จะไม่ทำงานใดๆเลย ข้อเด่นอย่างหนึ่งของโพรโทคอล AODV คือการค้นหาเส้นทางและเลือกใช้เส้นทางของคู่สถานีเชื่อมโยงต้นทาง และปลายทางที่มีอยู่ เพื่อให้การส่งข้อมูลนั้นเป็นไปอย่างถูกต้อง โพรโทคอลสถานะลิงค์และเวกเตอร์ระยะทำงานได้ในเน็ตเวิร์คไร้สายแบบเฉพาะกิจที่มีการเคลื่อนที่ของสถานีเชื่อมโยงน้อย ทำให้การเปลี่ยนแปลงของ ภูมิภาคของเน็ตเวิร์คไม่มากนัก แต่นอกจากปัญหาเกี่ยวกับการเปลี่ยนแปลงของภูมิภาคของเน็ตเวิร์คบ่อยแล้ว ในการทำงานของโพรโทคอลเหล่านี้คือมีการส่งข้อความควบคุม (Control Messages) เป็นช่วงๆเพื่อใช้ในการกำหนดเส้นทางหรือปรับปรุงข้อมูลเส้นทาง

การทำงานของ การจัดสรรเส้นทางแบบ AODV เป็นดังรูปที่ 5 โดยกำหนดให้โหนด 0 เป็นโหนดต้นทางและโหนด 7 เป็นโหนดปลายทางโหนดต้นทางจะทำการส่ง route request (RREQ) ไปยังโหนดข้างเคียง และโหนดที่ได้รับก็จะทำการส่งต่อไปยังโหนดที่ใกล้เคียงต่อไปเรื่อย ๆ โหนดข้างเคียงแต่ละโหนดจะทำการปรับปรุงตารางเก็บค่าเส้นทาง เมื่อโหนดปลายทางได้รับ RREQ โหนดปลายทางก็จะทำการส่ง route reply (RREP) กลับไปยังโหนดต้นทางที่ทำการส่ง RREQ มาให้โดยจะส่งกลับไป

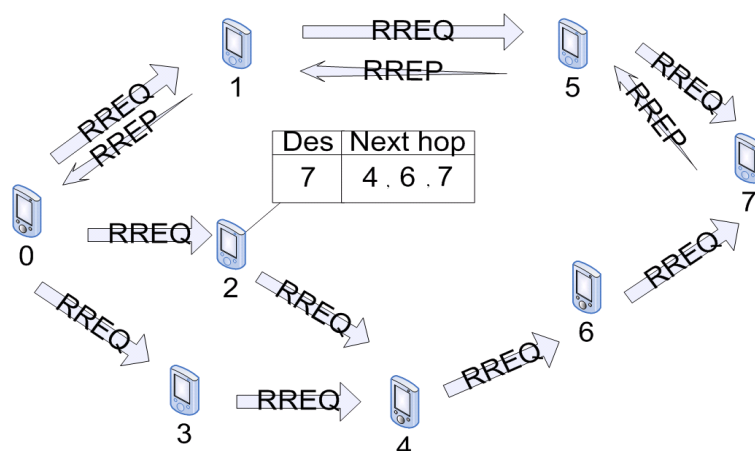
เส้นทางที่ RREQ ตัวแรกมาถึง เพราะถือว่าใช้เวลาน้อยที่สุดในการส่ง RREQ มาจาก ต้นทาง



รูปที่ 5 การหาเส้นทางของโปรโตคอล AODV

2.2.3 Dynamic Source Routing (DSR)

โปรโตคอล DSR [13] เป็นโปรโตคอลการจัด เส้นทางแบบหนึ่งในโปรโตคอลที่มีลักษณะเป็น รีแอดทีฟ ลักษณะการทำงานคล้ายกับ AODV ดังรูปที่ 6 คือ จะทำการส่ง RREQ แฝกเก็บไปจุด ข้างเคียงจนกว่าจะถึงปลายทาง แต่ลักษณะตารางเก็บค่าเส้นทาง ที่โหนดต้นทาง จะต่างจาก AODV คือ ในส่วนของเส้นทางไปยังโหนดปลายทางแทนที่จะมีแค่ค่าโหนดถัดไป (Next Hop) แต่จะบรรจุ ทั้งหมดของเส้นทางไว้รวมกับข้อมูลในแพ็คเก็ต ดังรูปที่ 7 หากโหนดต้นทางไม่มีเส้นทางไปยังโหนด ปลายทาง จะทำกระบวนการหาเส้นทางเช่นเดียวกับ AODV เพียงแต่ต่างกันว่า RREQ จะสะสม รายละเอียดเส้นทางครบทุกโหนดไปยังจุดหมาย และตอบกลับมาให้โหนดต้นทางทราบผ่าน RREP เพื่อให้สถานีเชื่อมโยงสามารถค้นหาเส้นทาง ไปยัง จุดหมาย และตอบกลับมาให้โหนดต้นทางทราบผ่าน RREP



รูปที่ 6 การหาเส้นทางของโปรโตคอล DSR



รูปที่ 7 เปรียบเทียบลักษณะตารางเก็บค่าเส้นทางที่แตกต่างกัน

2.3 มาตรฐาน IEEE 802.11

เทคโนโลยีพื้นฐานที่รองรับเครือข่ายเคลื่อนที่แบบ Ad hoc คือ เทคโนโลยีไร้สายโดยใช้มาตรฐานของ IEEE 802.11 สร้างขึ้นในปี ค.ศ. 1997 โดยองค์กร IEEE (The Institute of Electronics and Electrical Engineers) ซึ่งระบบมีอัตราการความเร็วในการรับส่งข้อมูลที่ 2 ล้านบิตต่อวินาที มาตรฐานนี้อ้างอิงถึงระดับชั้น Physical และระดับชั้น Medium Access Control (MAC)

- ระดับชั้น Physical (PHY)

ในระดับชั้น PHY มาตรฐาน IEEE 802.11 ได้กำหนดให้อุปกรณ์มีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1 2 5.5 11 และ 54 Mbps โดยมีสเปกให้เลือกใช้ได้แก่ คลื่นวิทยุที่มีความถี่สาธารณะ 2.4 GHz และ 5 GHz และ อินฟราเรด ที่ 1 Mbps และ 2 Mbps เท่านั้น

- ระดับชั้น Media Access Control (MAC)

ในระดับชั้น MAC มาตรฐาน IEEE802.11 ได้กำหนดให้มีกลไกการทำงานที่เรียกว่า CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ซึ่งมีความคล้ายคลึงกับหลักการ CSMA/CD (Collision Detection) ของมาตรฐาน IEEE 802.3 Ethernet ซึ่งเป็นที่นิยมใช้กันทั่วไปในเครือข่าย LAN แบบไร้สายนำสัญญาณ นอกจากนี้ในมาตรฐาน IEEE 802.11 ยังกำหนดให้มีทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย

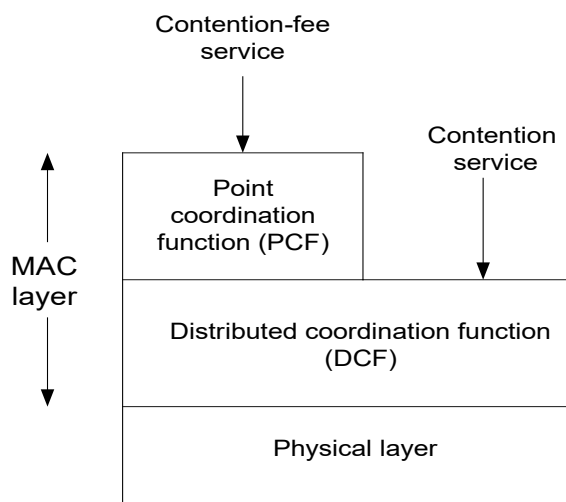
มาตรฐาน IEEE 802.11 เวอร์ชันแรกเริ่มมีประสิทธิภาพค่อนข้างต่ำและไม่มีการรองรับหลักการ Quality of Service (QoS) ซึ่งเป็นที่ต้องการของตลาด อีกทั้งกลไกรักษาความปลอดภัยที่ใช้ยังมีช่องโหว่อยู่มาก IEEE จึงได้จัดตั้งคณะทำงาน (Task Group) ขึ้นมาหลายชุดด้วยกัน เพื่อทำการปรับปรุงเพิ่มมาตรฐานให้มีศักยภาพสูงขึ้น โดยคณะทำงานกลุ่มที่มีผลงานที่น่าสนใจและเป็นที่รู้จักกันดีได้แก่ IEEE 802.11a, IEEE 802.11b, IEEE 802.11e และ IEEE 802.11g

มาตรฐาน IEEE 802.11e เป็นมาตรฐานที่ออกแบบมา เพื่อควบคุมและรับประกันคุณภาพของการใช้งานตามหลักการ QoS โดยการปรับปรุงโพรโทคอล MAC ซึ่งทำให้คุณสมบัติในการรับประกันคุณภาพดีขึ้น และเป็นการเพิ่มประสิทธิภาพในการใช้งานของแอปพลิเคชันทางด้านมัลติมีเดียมากขึ้น อย่างเช่น กลุ่มการทำงานที่ใช้ Voice over IP (VoIP)

มาตรฐาน IEEE 802.11g มาตรฐานนี้เป็นมาตรฐานใหม่ที่ความถี่ 2.4 GHz โดยสามารถรับส่งข้อมูลที่ความเร็ว 36 ถึง 54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน IEEE 802.11b ดังนั้นลักษณะของ IEEE 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ (ตามสภาพแวดล้อม

ของเครือข่ายที่ใช้งาน) มาตรฐานนี้เป็นที่ยอมรับจากผู้ใช้เป็นจำนวนมาก และกำลังจะเข้ามาแทนที่ 802.11b ในอนาคตอันใกล้

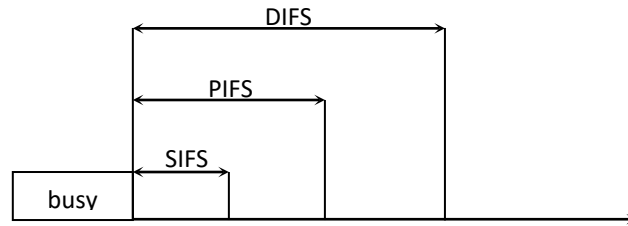
2.4 สถาปัตยกรรม IEEE 802.11



รูปที่ 8 แสดงสถาปัตยกรรมของ IEEE 802.11

สถาปัตยกรรมตัวนี้จะอยู่ที่ระดับชั้น MAC ซึ่งสามารถแบ่งเป็น 2 ฟังก์ชันได้แก่ 1) Point Coordination Function (PCF) จะเป็นองค์ประกอบหลักของเครือข่ายไร้สายที่ใช้สถานีฐาน และ 2) Distributed Coordination Function (DCF) [14] หรือใช้ในกรณีที่เป็นระบบเครือข่ายแบบ Ad hoc โดยมีโพรโทคอล Carrier Sense Multiple Access / Carrier Access (CSMA/CA) ทำหน้าที่ในการจัดการระบบนั้นก็คือบริการที่ต้องมีการแข่งขันกันเพื่อเข้าไปใช้ช่องสัญญาณ โดยจะต้องตรวจสอบก่อนที่จะส่งข้อมูล แต่ถ้าช่องสัญญาณไม่ว่าง จะทำการสุ่มข้อมูลแทน

จากรูปที่ 8 ลักษณะการทำหน้าที่ในส่วนที่ 2 หรือเป็นการควบคุมการใช้สื่อแบบกระจายเป็นการควบคุมพื้นฐานของมาตรฐาน IEEE 802.11 ซึ่งแต่ละโหนดสามารถติดต่อสื่อสารกันได้โดยตรง โดยไม่ต้องอาศัยสถานีฐานหรือโครงสร้างทางเครือข่าย ส่วนลักษณะการส่งข้อมูลเป็นการแย่งกันของช่องสัญญาณ และมีโพรโทคอล CSMA/CA ทำหน้าที่ในการควบคุมการใช้สื่อแบบกระจายได้แก่ 2 way-handshake และแบบ 4 way-handshake หรือที่เรียกว่า Request to send / Clear to send (RTS/CTS) การทำงานของโพรโทคอล CSMA/CA แต่ละสถานีจะได้ส่งข้อมูลก็ต่อเมื่อตรวจพบว่าช่องสัญญาณว่างเป็นระยะเวลาหนึ่ง เรียกระยะเวลาดังกล่าวว่าช่องว่างระหว่างเฟรม ตามมาตรฐานของ IEEE 802.11 แบ่งเฟรมออกเป็น 3 ประเภทคือ เฟรมการจัดการ (Management frame) เฟรมควบคุม (Control frame) และเฟรมข้อมูล (Data frame) ซึ่งเฟรมควบคุมทำหน้าที่ควบคุมการทำงานของ การรับส่งข้อมูลในเครือข่ายเช่นเฟรมตอบรับ หรือเฟรม RTS/CTS เป็นต้น ดังนั้นเฟรมควบคุมจึงควรได้ส่งทันทีเมื่อถึงเวลาส่ง



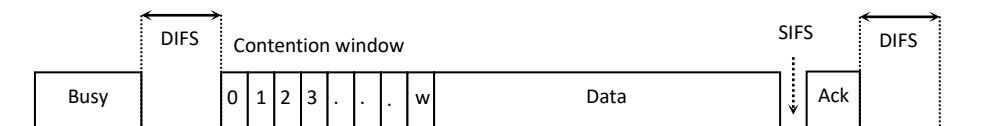
รูปที่ 9 แสดงขนาดของช่องว่างระหว่างเฟรมต่างๆของมาตรฐาน 802.11

Distributed Coordination Function Inter-Frame Spacing (DIFS) เป็นช่องว่างระหว่างเฟรมที่ยาวที่สุด จึงมีโพรบอริตีต่ำสุด ใช้สำหรับการส่งข้อมูลที่มีการควบคุมใช้สื่อแบบกระจาย

Point Coordination Function Inter-Frame Spacing (PIFS) เป็นช่องว่างระหว่างเฟรมที่สั้นกว่า DIFS ใช้สำหรับเฟรมที่ใช้ในการควบคุมใช้สื่อแบบรวมศูนย์ และเฟรมการจัดการ

Short Inter-Frame Spacing (SIFS) เป็นช่องว่างระหว่างเฟรมที่สั้นที่สุดจึงมีโพรบอริตีสูงสุดคือถูกออกแบบมาสำหรับเฟรมที่ใช้ควบคุมการทำงานของเครือข่าย ซึ่งต้องการการตอบรับทันทีเช่น เฟรมตอบรับ เฟรมร้องขอการส่งข้อมูล เป็นต้น

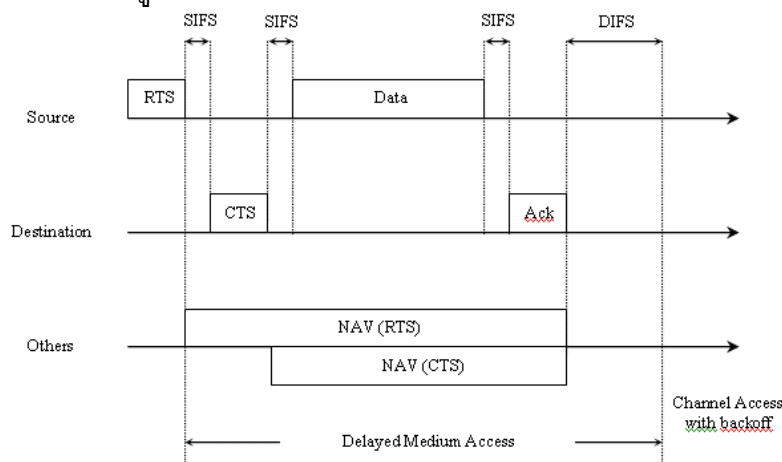
การควบคุมอย่างง่ายแสดงดังรูปที่ 9 โดยแต่ละสถานีที่ต้องการส่งข้อมูลจะตรวจดูการใช้ช่องสัญญาณด้วยฮาร์ดแวร์ซึ่งเป็นการตรวจช่องสัญญาณในระดับชั้น Physical เมื่อตรวจพบว่าช่องสัญญาณว่างเป็นระยะเวลา DIFS สถานีนั้นจึงเริ่มใช้วิธีการสุ่มช่วงเวลาการเข้าใช้สื่อด้วยวิธี random back-off และจะเริ่มทำงานของตัวนับเวลา



รูปที่ 10 แสดงขั้นตอนการส่งข้อมูลในการควบคุมใช้สื่อแบบ DCF แบบ 2-way handshake

การควบคุมการใช้สื่อแบบกระจายในเครือข่ายไร้สาย ใช้การสุ่มเวลาเข้าใช้สื่อ โดยแต่ละสถานีจะสุ่มเวลาเข้าใช้สื่ออยู่ในช่วง $(0, w-1)$ ค่า w คือช่วงเวลาสำหรับแข่งขันของแต่ละสถานี ซึ่งค่า w จะขึ้นอยู่กับจำนวนครั้งการชนกันของข้อมูล ในภาวะเริ่มต้นค่า w จะถูกตั้งไว้เท่ากับ CW_{min} ซึ่งเป็นช่วงการแข่งขันที่สั้นที่สุด ค่า w จะเพิ่มขึ้นเป็น 2 เท่าทุกครั้งที่เกิดการชนกันของข้อมูล ดังนั้นค่า w จึงเท่ากับ $2^i CW_{min}$ เมื่อ i คือจำนวนครั้งของการชนกัน แต่ละสถานีจะเริ่มส่งข้อมูลเมื่อตัวนับลดค่า w จนเท่ากับศูนย์ซึ่งตัวนับ จะทำงานก็ต่อเมื่อตรวจพบว่าช่องสัญญาณว่างและจะหยุดทำงานเมื่อตรวจพบว่าช่องสัญญาณถูกใช้อยู่ เมื่อตัวนับเวลามีค่าเท่ากับศูนย์สถานีจึงเริ่มส่งข้อมูล หลังจากสถานีปลายทางได้รับข้อมูลเป็นระยะเวลา SIFS สถานีปลายทางจะส่งเฟรมตอบรับมาที่สถานีต้นทาง หากเกิดการชนกันของข้อมูล สถานีปลายทางจะไม่ส่งเฟรมตอบรับมาที่สถานีต้นทาง เมื่อสถานีต้นทางไม่ได้รับเฟรมตอบรับในช่วงเวลาที่กำหนดจะถือว่าเกิดการชนกันของข้อมูล สถานีต้นทางจะเริ่มสุ่มเวลาเข้าใช้สื่อและตรวจช่องสัญญาณเพื่อส่งข้อมูลใหม่

การควบคุมการใช้สื่อแบบกระจายอีกวิธีหนึ่งคือ 4 way-handshake หรือที่เรียกกันว่า RTS/CTS [16] ถูกใช้กันมากจนกลายเป็นส่วนหนึ่งของมาตรฐาน แสดงการทำงานดังรูปที่ 11 ทำงานโดยจะส่งเฟรมควบคุมขนาดเล็กไปก่อนเฟรมข้อมูลเรียกว่าเฟรม Request to send (RTS) หลังจากที่สถานีปลายทางได้รับเฟรม RTS แล้วรอจนช่องสัญญาณว่างเป็นเวลาเท่ากับ SIFS จะส่งเฟรม Clear to send (CTS) กลับมาที่สถานีต้นทาง สถานีที่ไม่ใช่สถานีต้นทางและปลายทางหากได้รับเฟรม RTS หรือเฟรม CTS จะเซตค่า Network Allocation Vector (NAV) ให้มีค่าเท่ากับช่วงเวลาเท่ากับที่สถานีต้นทางส่งข้อมูลไปที่สถานีปลายทางจนเสร็จ ซึ่งหากค่า NAV ไม่เท่ากับศูนย์สถานีนั้นจะไม่สามารถส่งข้อมูลได้ ซึ่งเป็นการตรวจช่องสัญญาณแบบเสมือนที่ชั้นควบคุมการใช้สื่อ เมื่อสถานีต้นทางได้รับเฟรม CTS เป็นเวลา SIFS จึงเริ่มส่งข้อมูลได้



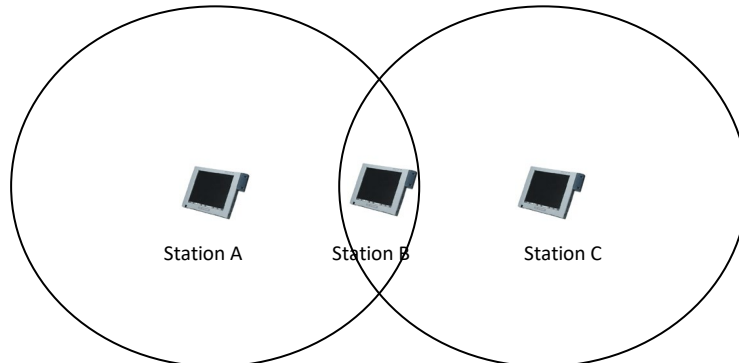
รูปที่ 11 การเข้าใช้สื่อของโพรโทคอล CSMA/CA ที่ใช้ RTS/CTS

เนื่องจากว่าในเครือข่ายแบบไร้สาย ไม่สามารถตรวจการชนกันของข้อมูลได้เหมือนเครือข่ายแบบมีสาย ดังนั้นโพรโทคอล CSMA/CA จึงตรวจการชนโดยการรอฟังสัญญาณตอบรับของสถานีปลายทาง เมื่อได้รับการตอบรับที่ถูกต้องจากสถานีปลายทางแสดงว่าการส่งข้อมูลนั้นสำเร็จ ถ้าสถานีต้นทางไม่ได้รับเฟรมตอบรับจากปลายทางเป็นระยะเวลามากกว่าที่ระบุไว้ หรือได้รับข้อมูลอื่นที่ไม่ใช่เฟรมตอบรับ สถานีต้นทางจะถือว่าเกิดการชนกันของข้อมูลและจะเริ่มการสุ่มเวลาเข้าใช้สื่อเพื่อส่งข้อมูลซ้ำ

ปัญหาชนกันของข้อมูลเนื่องจากสถานีที่หลบซ่อน (Hidden terminal) [17] เกิดขึ้นเนื่องจากแต่ละสถานีไม่สามารถตรวจการชนกันได้โดยตรง เมื่อเกิดการชนกันขึ้นที่สถานีปลายทาง สถานีต้นทางไม่สามารถตรวจพบได้ดังรูปที่ 12 ขณะที่สถานี A ส่งข้อมูลให้สถานี B สถานี C ไม่สามารถตรวจการใช้สื่อที่สถานี B ได้เมื่อสถานี C ส่งข้อมูลไปที่สถานี B จึงเกิดการชนกันของข้อมูลที่สถานี B ขึ้น

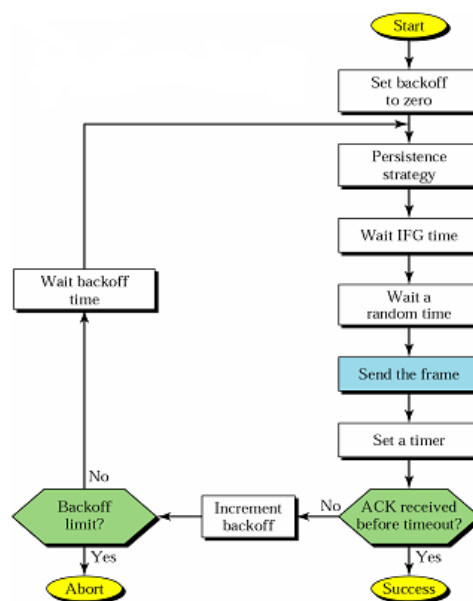
การแก้ปัญหาดังกล่าวทำได้โดยใช้วิธี 4-way handshaking หรือที่เรียกว่า RTS/CTS ดังรูปที่ 13 สถานีที่ต้องการส่งข้อมูลจะตรวจช่องสัญญาณจนกระทั่งช่องสัญญาณว่างเป็นระยะเวลาระหว่างเฟรมเท่ากับ DIFS จึงเริ่มเข้าสู่ช่วงการสุ่มเข้าใช้สื่อ เมื่อได้ส่งสถานีต้นทางจะส่งเฟรม RTS ไปที่สถานีปลายทางเพื่อจองช่องสัญญาณ สถานีอื่นที่ไม่ใช่สถานีปลายทางเมื่อได้รับเฟรมนี้จะเซตค่าการจราจรเครือข่าย (NAV) ไว้ค่าหนึ่งซึ่ง เมื่อค่า NAV ของสถานีใดไม่เท่ากับศูนย์สถานีนั้นจะไม่สามารถส่งข้อมูล

ได้ ค่า NAV นี้จะถูกตั้งไว้เท่ากับช่วงเวลาเท่ากับสถานีต้นทางจะส่งข้อมูลไปยังสถานีปลายทางเสร็จ เมื่อสถานีปลายทางเมื่อได้รับเฟรม RTS จะตอบกลับมาด้วยเฟรม CTS เพื่อเป็นการจองช่องสัญญาณเช่นกัน เมื่อสถานีต้นทางได้รับเฟรม CTS จึงเริ่มส่งข้อมูลไปให้สถานีปลายทาง



รูปที่ 12 ปัญหาการชนกันของข้อมูลเนื่องจากสถานีที่ซ่อนอยู่ (Hidden terminal)

Network Allocation Vector (NAV) คือ ตัวที่ระบุว่าเวลาที่มีสถานีที่จะทำการส่งข้อมูลเข้าไป มันจะต้องการ Reserve Channel เป็นระยะเวลาสั้นเท่าไร โดยจะถูกกำหนดใน Header ของเฟรมที่ส่ง ซึ่งจะมีฟิลด์ Duration อยู่ซึ่ง NAV เป็นแบบ Virtual Carrier Sense Function



รูปที่ 13 หลักการทำงานของ CSMA/CA

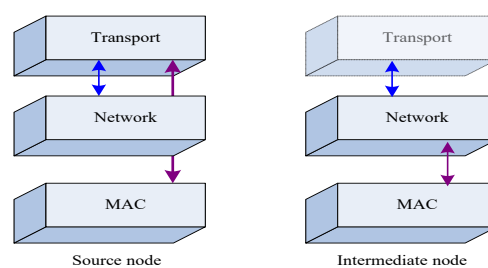
บทที่ 3

ระบบการทำงานแบบ Cross layer

การทำงานของระบบสื่อสารไร้สาย โดยส่วนมากจะพบปัญหาเมื่อโหนดสื่อสารเคลื่อนที่ออกจากรัศมีการส่งสัญญาณ จึงทำให้ผู้คิดค้นวิธีการการทำงานที่จะช่วยลดปัญหาการสูญหายของข้อมูล ดังนั้นในงานวิจัยนี้จึงได้เสนอแนวทางการทำงานที่เรียกว่า Cross layer

3.1 รูปแบบของการสื่อสารแบบ Cross layer

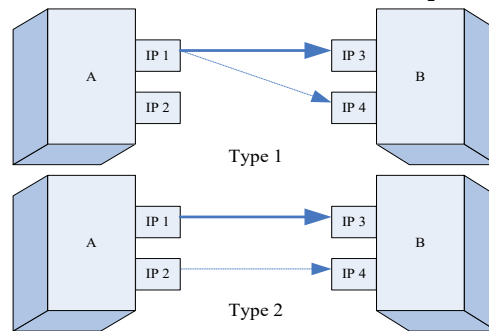
ปัญหาเบื้องต้นของการสื่อสารแบบนี้ก็คือ ระบบจะลดอัตราการส่งข้อมูล แทนการแก้ปัญหาที่เกิดจากระดับชั้น Network หรือระดับชั้นที่ต่ำกว่านี้ ดังนั้นงานวิจัยนี้จึงออกแบบให้ระบบทำงานร่วมกับระดับชั้น Transport เพื่อช่วยแก้ปัญหาลดอัตราการส่งข้อมูล โดยการหน่วงเวลาการทำงานของระดับชั้น Transport เมื่อเกิดการปัญหาของเส้นทางขาดในระดับชั้น Network และปัญหาสัญญาณการส่งของระดับชั้น MAC วิธีการทำงานของการออกแบบได้แก่ การสื่อสารเริ่มต้นจากตัวส่งข้อมูลที่เรารู้จักว่า Source node ซึ่งการออกแบบงานวิจัยนี้จะให้ระบบทำงานออกเป็น 2 ส่วนได้แก่ การติดต่อสื่อสารในระดับ Network เมื่อเกิดการสูญหายของข้อมูล และ การสื่อสารในระดับ MAC เมื่อระบบเกิดปัญหาจากสัญญาณการส่ง ทั้งสองส่วนนี้จะทำงานเมื่อเกิดปัญหา นอกจากนั้นได้ออกแบบเพิ่มเติมสำหรับโหนดเพื่อนบ้าน ให้ทำงานเชื่อมต่อกับระดับชั้นดังรูปที่ 14 การทำงานดังกล่าวจะเกิดขึ้นเมื่อเกิดปัญหาการสื่อสารที่ระดับชั้น Network และ MAC



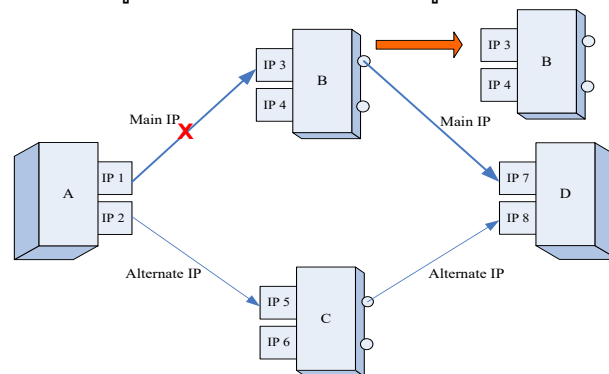
รูปที่ 14 การสื่อสารโดยตรงระหว่างระดับชั้นสื่อสาร

ส่วนรูปที่ 15 เป็นการแสดงให้เห็นถึงโหนดสื่อสารแต่ละตัวว่ามี 2 IP ดังนั้นการสื่อสารแบบที่ 1 และแบบที่ 2 จะมีเส้นทางการสื่อสาร 2 เส้นทาง สำหรับการสื่อสารแบบที่ 1 ถ้าหากเส้นทางที่ใช้อยู่เป็นประจำไม่สามารถส่งข้อมูลได้ ก็สามารถใช้อีกเส้นทางการสื่อสารได้ ดังแสดงในรูปที่ 16 แต่อีกกรณีหนึ่งถ้าการสื่อสารทั้งสองเส้นทางใช้โหนดเพื่อนบ้านที่เหมือนกัน ดังรูปที่ 17 จะทำให้เกิดปัญหากับระบบการ

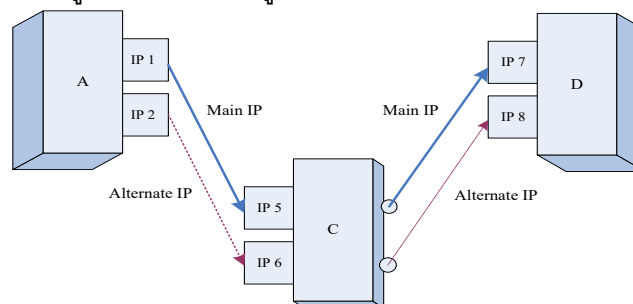
สื่อสารขึ้น ดังนั้นการออกแบบที่เรียกว่า Cross layer จะช่วยให้ระบบรู้จัก กลุ่ม IP สื่อสาร โดยการ ทำงานตั้งแต่การเริ่มต้นการทำงานของระบบ เพื่อให้ระดับชั้นอื่นๆ รู้จักกลุ่ม IP ของตนเอง



รูปที่ 15 การสื่อสารแบบส่งข้อมูลถึงกัน



รูปที่ 16 การส่งข้อมูลด้วยโหนดเพื่อนบ้านที่ต่างกัน



รูปที่ 17 การส่งข้อมูลด้วยโหนดเพื่อนบ้านที่เหมือนกัน

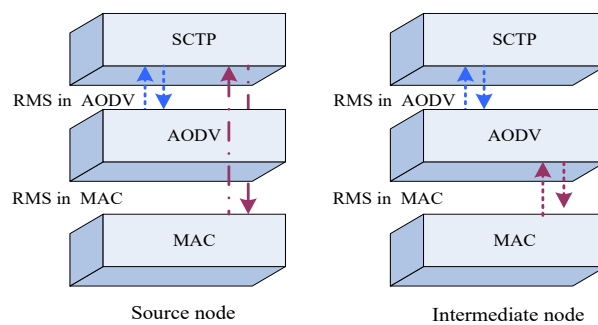
3.2 การออกแบบ Real Multiroute System (RMS)

จึงได้ออกแบบการทำงานที่เรียกว่า Real Multiroute System (RMS) การทำงานในระดับชั้น Transport จะใช้โพรโทคอลที่มีชื่อว่า Stream Control Transmission Protocol (SCTP) ซึ่งเป็นโพรโทคอลที่ต้องทำการติดต่อสื่อสาร จากต้นทางไปยังปลายทางก่อนที่จะส่งข้อมูล สำหรับโพรโทคอลตัวนี้ งานวิจัยนี้จะใช้ การสื่อสารที่เรียกว่า Multihoming หมายความว่าโหนดการสื่อสาร 1 ตัวจะมี IP มากกว่า 1 IP สื่อสาร ดังนั้นงานวิจัยจะให้โหนดแต่ละตัวมี 2 IP สื่อสาร

สำหรับโพรโทคอลในระดับชั้น Network ได้แก่ AODV ซึ่งจะทำงานเมื่อเส้นทางการสื่อสารหลัก เกิดปัญหา โดยที่ระบบจะส่งข้อมูลไปยังระดับชั้น SCTP ให้หน่วงเวลาการของการลดอัตราการส่งข้อมูล ก่อน จนกว่าระดับชั้น AODV จะสามารถหาเส้นทางใหม่ได้ และเมื่อระดับชั้น AODV สามารถหาเส้นทางใหม่ได้ เราก็จะเอาระบบการหน่วงเวลาที่ระดับชั้น SCTP ออก เพื่อให้ระบบสามารถทำงานได้ตามปกติ

ส่วนระดับชั้น MAC จะให้ทำงานเมื่อสัญญาณมีปัญหา และใช้หลักการเดียวกันกับระดับชั้น AODV หมายความว่า โพรโทคอล SCTP จะทำงานร่วมกับ โพรโทคอล MAC ดังรูปที่ 18

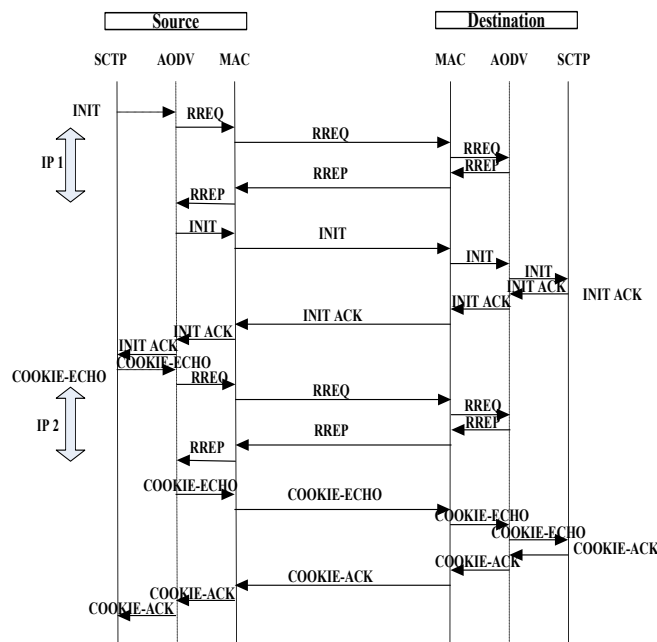
ส่วนโหนดเพื่อนบ้านจะทำงานแตกต่างจากโหนดต้นทาง หรือ Source node คือ เมื่อเกิดปัญหาเส้นทางหลักสูญหาย โพรโทคอล SCTP จะทำงานร่วมกับ โพรโทคอล AODV และ ปัญหาที่เกิดจากสัญญาณสูญหาย โพรโทคอล AODV จะทำงานร่วมกับ โพรโทคอล MAC ในส่วนนี้ที่แตกต่างจากโหนดต้นทางก็เพราะว่า โหนดเพื่อนบ้านไม่จำเป็นที่จะต้องไปหน่วงเวลาการลดอัตราการส่งข้อมูล สำหรับโพรโทคอล AODV จะส่งสัญญาณไปยัง โพรโทคอล SCTP ก็เพื่อตรวจสอบกลุ่ม IP สื่อสาร และ โพรโทคอล MAC จะใช้ในการจัดการเมื่อไม่ได้รับข้อมูลที่เรียกว่า Hello message ซึ่งข้อมูลอันนี้จะเป็นตัวบอกให้รู้ว่าระบบขาดการติดต่อสื่อสารข้อมูล ซึ่งแสดงดังรูปที่ 18



รูปที่ 18 การทำงานของ RMS

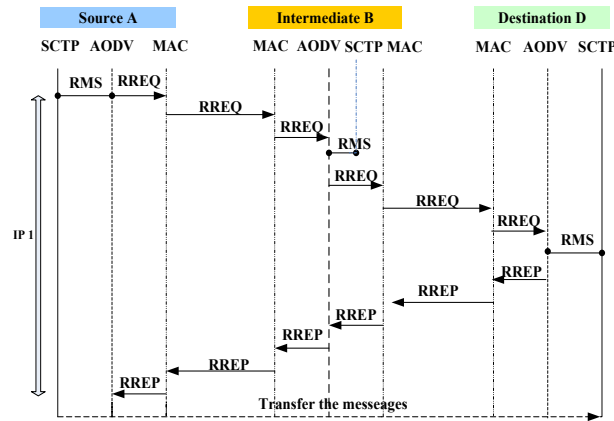
3.3 การเริ่มต้นการทำงานของระบบ

กระบวนการทำงานเกิดจากระดับชั้น Transport จะส่งสัญญาณทั้งหมด 4 ตัวเพื่อเริ่มต้นการส่งข้อมูลในเส้นทางที่ 1 หรือเรียกว่า IP 1 ดังรูปที่ 19 และหาเส้นทางที่สอง หรือ IP 2



รูปที่ 19 กระบวนการเริ่มต้นการทำงาน

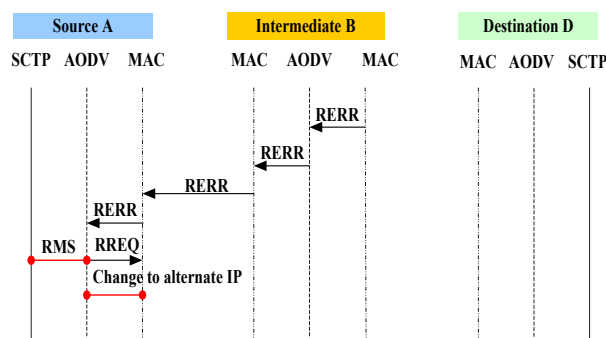
ทั้ง 2 IP สื่อสาร จะเกิดจากกระบวนการทำงานจาก INIT, INIT-ACK, COOKIE-ECHO และ COOOKIE-ACK ซึ่งเป็นหลักการทำงานพื้นฐานของโปรโตคอล SCTP ดังนั้นหลังจากที่ IP 1 ทำงานเสร็จเรียบร้อยแล้ว ก็สามารถส่งข้อมูลทันที



รูปที่ 20 การสื่อสารที่เรียกว่า Multihop

3.4 ระบบเกิดปัญหาการสูญหายของข้อมูล

ปัญหาจะเกิดจาก 2 ส่วน ได้แก่ ระดับชั้น Network และ MAC ซึ่งระบบโดยปกติจะมีการตรวจสอบจากโปรโตคอลในระดับชั้นต่างๆ และเมื่อระดับชั้น Network ได้รับข้อมูลที่ชื่อว่า Route Error (RERR) ก็จะส่งไปยังต้นทางเพื่อที่จะทำ Cross layer ที่เรียกว่า RMS หลังจากทีระบบทำการค้นหาเส้นทางใหม่ได้ ก็ส่งข้อมูลไปยังปลายทางต่อไป ดังแสดงในรูปที่ 21

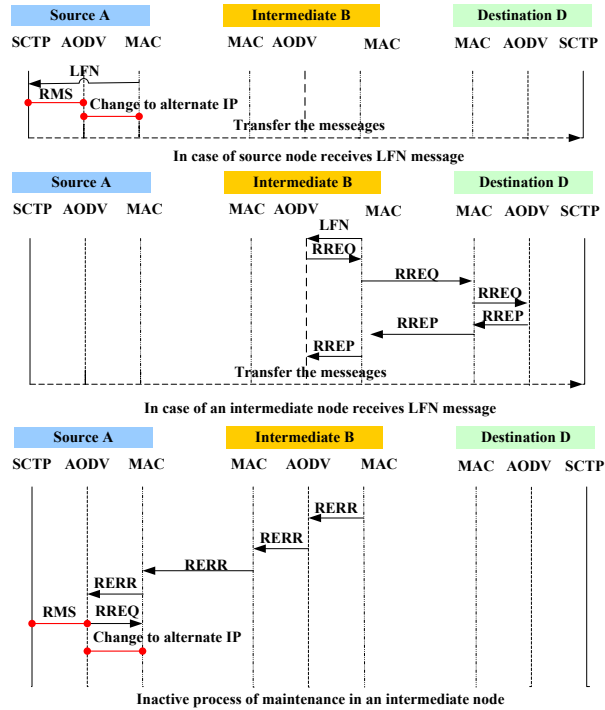


รูปที่ 21 เส้นทางหลักขาดการติดต่อสื่อสาร

ส่วนปัญหาที่เกิดจากสัญญาณขาดหาย หรือกำลังส่งมีปัญหา ในระดับชั้น MAC จะส่งข้อมูลที่ชื่อว่า Link Failure Notification (LFN) ไปยังโปรโตคอล AODV ซึ่งแสดงในรูปที่ 22 สำหรับกรณีนี้ เราจะแบ่งการทำงานออกเป็น 2 ช่วง ได้แก่

1. ถ้าหากรับข้อมูล LFN อยู่ใกล้กับต้นทางให้ส่งข้อมูลนี้ไปยังต้นทางเพื่อหาเส้นทางใหม่

2. ถ้าหากรับข้อมูล LFN อยู่ใกล้ปลายทางมากกว่าต้นทาง ให้โหนดเพื่อนบ้านทำการหาเส้นทางใหม่ทันที แต่ถ้าหากไม่สามารถหาเส้นทางใหม่ได้ ก็ให้ส่งข้อมูล RERR กลับไปยังต้นทาง เพื่อให้ต้นทางหาเส้นทางส่งข้อมูลอีกครั้ง



รูปที่ 22 การแสดงปัญหาที่เกิดขึ้นในระดับ MAC

บทที่ 4

ผลการวิจัย

งานวิจัยนี้เราจะเริ่มต้นจากการวิเคราะห์ปัญหาที่เกิดจากเส้นทางการส่งข้อมูล ที่ไม่สามารถสื่อสารกันได้ หลังจากนั้นจะทำการทดสอบการทำงานที่ได้ออกแบบไว้ที่เรียกว่า Real Multiroute System (RMS) โดยจะแบ่งการทดสอบออกเป็น 2 ช่วงได้แก่

4.1 ทดสอบการทำงานของทอพอโลยี

การจำลองการทำงานของโพรโทคอลการค้นหาเส้นทางผ่านโปรแกรม NS-2 ได้แก่โพรโทคอล DSDV AODV และ DSR โดยเปรียบเทียบการทำงานแต่ละโพรโทคอล เมื่อมีทอพอโลยีที่แตกต่างกัน โดยแต่ละทอพอโลยีจะมีการเซตให้แต่ละโหนดอยู่กับที่ และจะเปรียบเทียบกับการให้โหนดบางโหนดมีการเคลื่อนที่ เพื่อดูลักษณะการทำงานของแต่ละโพรโทคอล เมื่อมีการทำการค้นหาเส้นทางใหม่ให้กับเครือข่าย แทนเส้นทางหลักที่ไม่สามารถติดต่อสื่อสารได้ ซึ่งในการทดสอบมีทอพอโลยีที่แตกต่างกัน 3 รูปแบบ

พารามิเตอร์ที่ใช้ในการจำลอง

ในการกำหนดค่าพารามิเตอร์จะทำการแก้ไขไฟล์ .tcl ซึ่งอยู่ในโปรแกรม NS-2 [18] ดูการเซตค่าดังกล่าวในภาคผนวก ก โดยในการทดสอบจำกำหนดค่าพารามิเตอร์ต่างๆ ดังนี้

| | |
|----------------------|---------------------|
| จำนวนโหนด | 6 โหนด |
| จำนวน Connection | 1 |
| ระยะห่างของแต่ละโหนด | 350 เมตร |
| ระยะเวลาในการจำลอง | 500 วินาที |
| ชนิดข้อมูลที่ส่ง | Constant bit rate |
| ขนาดของแพ็คเก็ต | 512 ไบต์ |
| ขนาดของทอพอโลยี | 2000x2000 ตารางเมตร |
| Traffic Interval | 0.1 |

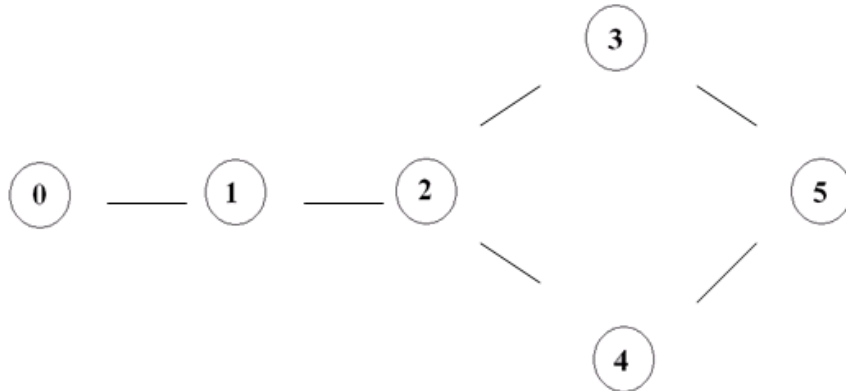
ทำงานบน MAC 802.11

ทอพอโลยีที่ใช้ในการจำลอง

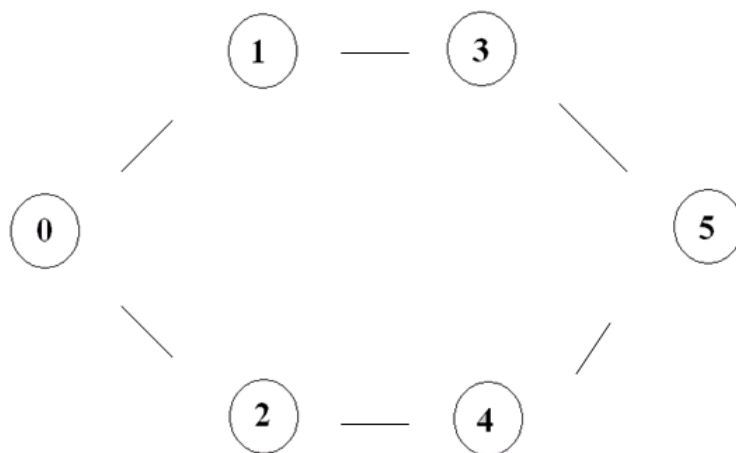
การทดสอบจะจำลองการทำงานของโพรโทคอล AODV DSR และ DSDV โดยเปรียบเทียบการทำงานแต่ละโพรโทคอลเมื่อมีทอพอโลยีที่แตกต่างกัน โดยแต่ละทอพอโลยีจะมีการเซตให้แต่ละโหนดอยู่กับที่และจะเปรียบเทียบกับเซตให้โหนดบางโหนดมีการเคลื่อนที่ เพื่อดูลักษณะการทำงานของแต่ละโพรโทคอล ซึ่งในการทดสอบมีทอพอโลยีที่แตกต่างกัน 3 รูปแบบดังนี้



รูปที่ 23 แสดงทอพอโลยีรูปแบบที่ 1



รูปที่ 24 แสดงทอพอโลยีรูปแบบที่ 2



รูปที่ 25 แสดงทอพอโลยีรูปแบบที่ 3

ในการเซตรูปแบบของทอพอโลยีที่ใช้ในการจำลองจะต้องเข้าไปแก้ไขไฟล์ .tcl ดูได้จากภาคผนวก ก โดยจะทำการเซตพิกัดของแต่ละโหนด กำหนดเวลาและความเร็วที่โหนดเคลื่อนที่ออก

จากรูปที่ 23 ในวินาทีที่ 180 ของการจำลอง กำหนดให้โหนดที่ 3 เคลื่อนที่ออกไปด้วยความเร็ว 20 m/s

จากรูปที่ 24 ในวินาทีที่ 180 ของการจำลอง กำหนดให้โหนดที่ 3 เคลื่อนที่ออกไปด้วยความเร็ว 20 m/s และเมื่อถึงวินาทีที่ 300 กำหนดให้โหนดที่ 2 เคลื่อนที่ออกไปด้วยความเร็ว 20 m/s

จากรูปที่ 25 ในวินาทีที่ 180 ของการจำลอง กำหนดให้โหนดที่ 4 เคลื่อนที่ออกไปด้วยความเร็ว 20 m/s และเมื่อถึงวินาทีที่ 300 กำหนดให้โหนดที่ 2 เคลื่อนที่ออกไปด้วยความเร็ว 20 m/s

ผลลัพธ์ที่ได้จากการจำลอง

ผลลัพธ์จากการจำลองจะได้ไฟล์ .tr ซึ่งต้องเขียนสคริปไฟล์ .awk ในการดึงค่าผลลัพธ์ออกมา โดยรูปแบบของไฟล์ .tr ดูได้จากภาคผนวก ข เมื่อใช้ไฟล์ .awk ดึงค่าออกมาแล้วแสดงค่าที่ได้จากการจำลองดังนี้

| | DSDV | | AODV | | DSR | |
|-----------------------|-------|--------|-------|--------|-------|--------|
| | Fix | Mobile | Fix | Mobile | Fix | Mobile |
| Start time[s] | 60 | 60 | 60 | 60 | 60 | 60 |
| Stop time[s] | 500 | 205 | 500 | 204 | 500 | 204 |
| Data send[packets] | 4400 | 4400 | 4400 | 4400 | 4400 | 4400 |
| Data receive[packets] | 4400 | 1460 | 4400 | 1444 | 4400 | 1444 |
| avgThroughput [kbps] | 40.97 | 40.96 | 40.97 | 40.96 | 40.97 | 40.97 |
| avgDelay[ms] | 17.94 | 17.94 | 30.37 | 31 | 30.13 | 42.97 |
| Routing Packet | 257 | 258 | 10 | 154 | 11 | 254 |
| Packet Delivery Ratio | 100 | 33.18 | 100 | 32.82 | 100 | 32.82 |
| Packets dropped | 0 | 3327 | 0 | 2865 | 0 | 2957 |

ตารางที่ 1 ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 1

| | DSDV | | AODV | | DSR | |
|-----------------------|-------|--------|-------|--------|-------|--------|
| | Fix | Mobile | Fix | Mobile | Fix | Mobile |
| Start time[s] | 60 | 60 | 60 | 60 | 60 | 60 |
| Stop time[s] | 500 | 314 | 500 | 314 | 500 | 314 |
| Data send[packets] | 4400 | 4400 | 4400 | 4400 | 4400 | 4400 |
| Data receive[packets] | 4400 | 2411 | 4400 | 2543 | 4400 | 2544 |
| avgThroughput [kbps] | 40.97 | 40.96 | 40.97 | 40.96 | 40.97 | 40.97 |
| avgDelay[ms] | 24.01 | 24.01 | 24.23 | 24.15 | 24.04 | 36.36 |
| Routing Packet | 253 | 265 | 9 | 141 | 12 | 395 |
| Packet Delivery Ratio | 100 | 54.8 | 100 | 57.8 | 100 | 57.82 |
| Packets dropped | 0 | 2557 | 0 | 817 | 0 | 2544 |

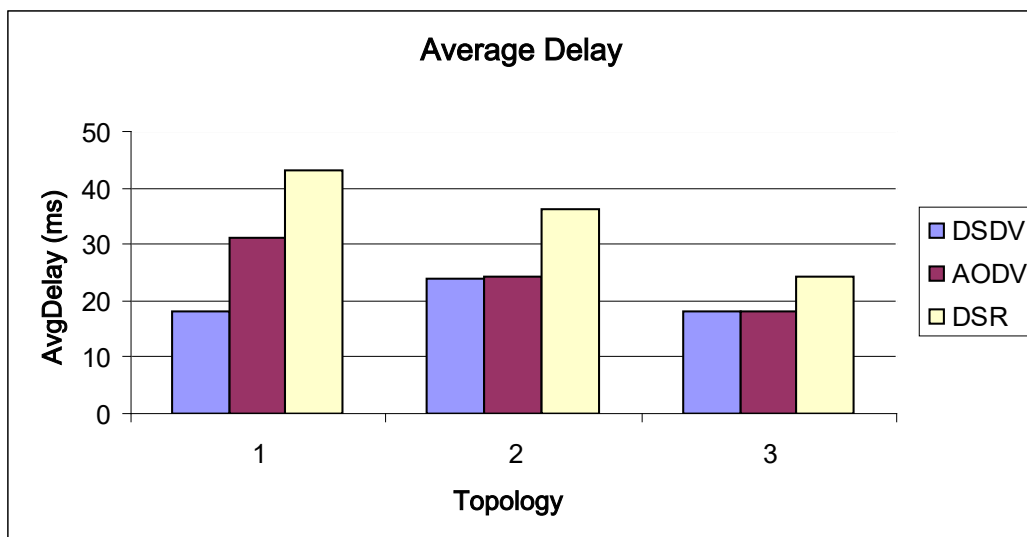
ตารางที่ 2 ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 2

| | DSDV | | AODV | | DSR | |
|-----------------------|-------|--------|-------|--------|-------|--------|
| | Fix | Mobile | Fix | Mobile | Fix | Mobile |
| Start time[s] | 60 | 60 | 60 | 60 | 60 | 60 |
| Stop time[s] | 500 | 500 | 500 | 500 | 500 | 500 |
| Data send[packets] | 4400 | 4400 | 4400 | 4400 | 4400 | 4400 |
| Data receive[packets] | 4400 | 4400 | 4400 | 3199 | 4400 | 4400 |
| avgThroughput [kbps] | 40.97 | 40.96 | 40.97 | 29.79 | 40.97 | 40.97 |
| avgDelay[ms] | 17.94 | 17.94 | 18.18 | 18.12 | 24.46 | 24.11 |
| Routing Packet | 257 | 260 | 8 | 42 | 20 | 64 |
| Packet Delivery Ratio | 100 | 99.98 | 100 | 72.7 | 100 | 100 |
| Packets dropped | 0 | 2 | 0 | 1139 | 0 | 4 |

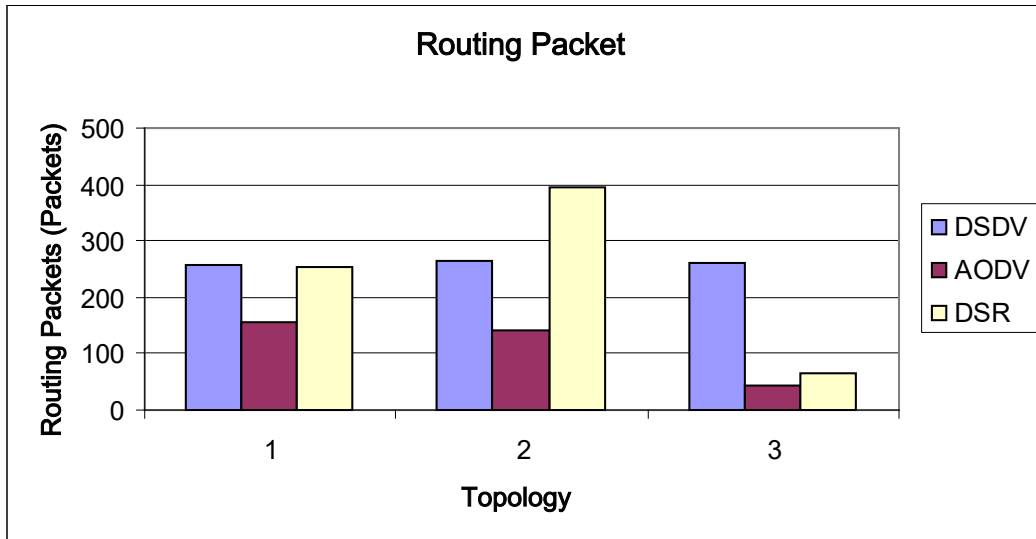
ตารางที่ 3 ผลลัพธ์ที่ได้จากการทดสอบในทอพอโลยีรูปแบบที่ 3

- * Fix แต่ละโหนดไม่มีการเคลื่อนที่
- * Mobile มีบางโหนดเคลื่อนที่ออกจากแนวเดิม
- * avgThroughput ค่า throughput เฉลี่ยทั้งหมด
- * avgDelay ค่าเฉลี่ย end – to – end Delay

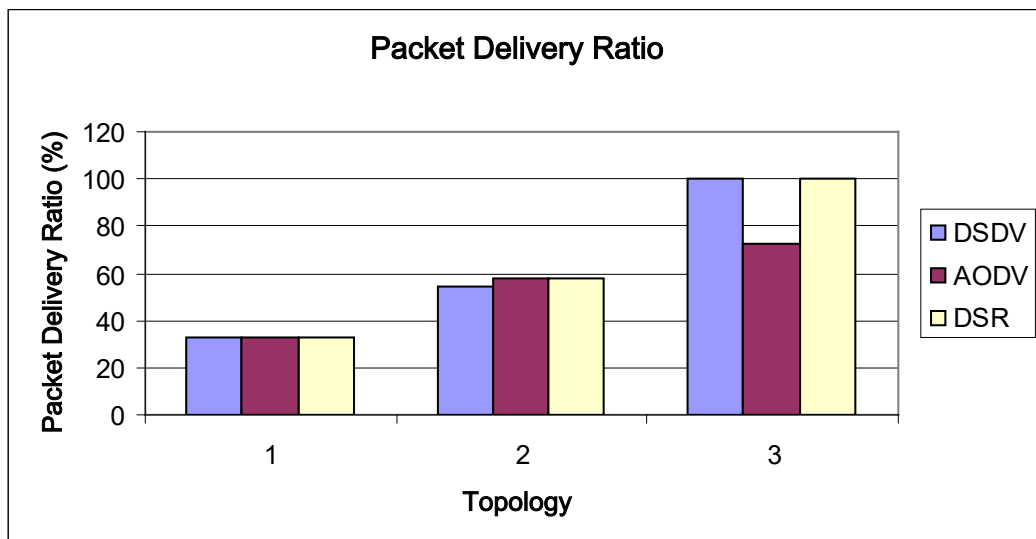
กราฟผลลัพธ์ที่ได้จากการจำลอง



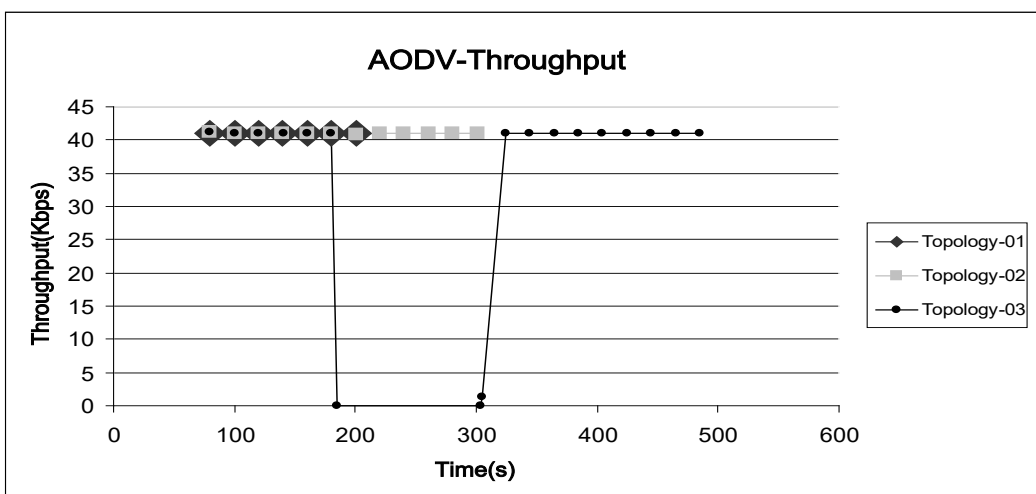
รูปที่ 26 แผนภูมิเปรียบเทียบค่า Delay



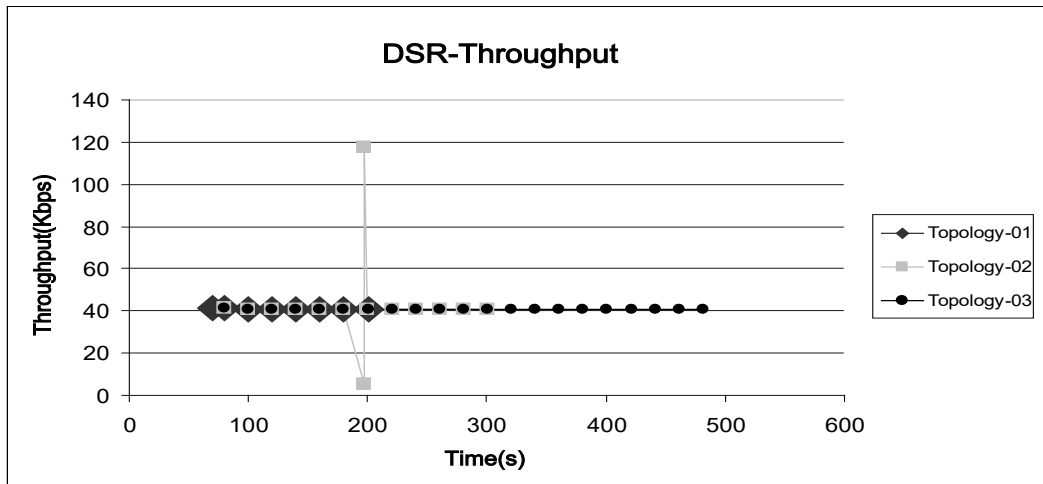
รูปที่ 27 แผนภูมิเปรียบเทียบค่า Routing Packet



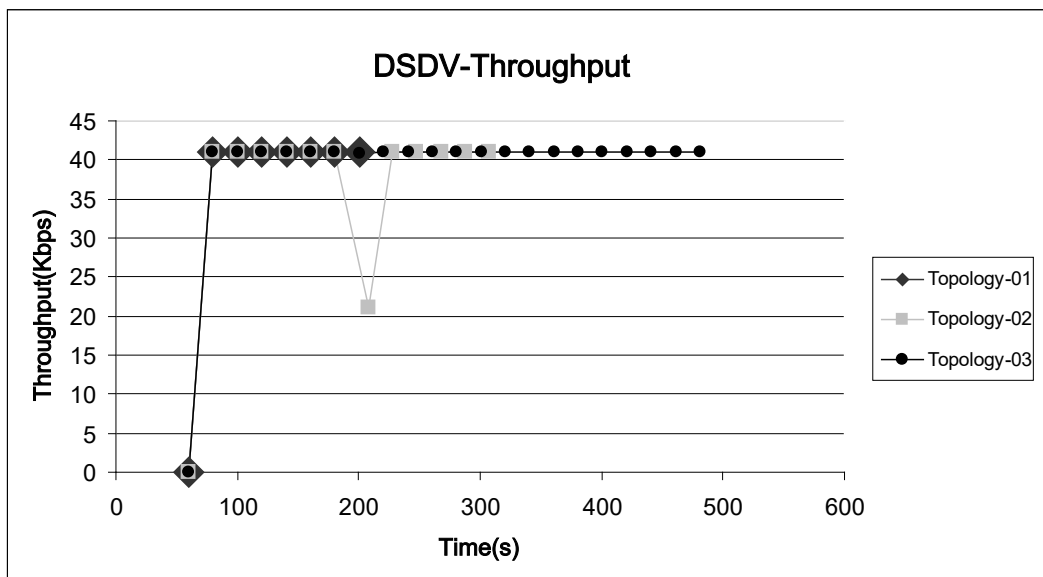
รูปที่ 28 แผนภูมิเปรียบเทียบค่า Packet Delivery Ratio



รูปที่ 299 เปรียบเทียบ Instants Throughput ของโปรโตคอล AODV ที่มีทอพอโลยีแตกต่างกัน



รูปที่ 30 เปรียบเทียบ Instants Throughput ของ DSR โพรโทคอลที่มีทอพอโลยีแตกต่างกัน



รูปที่ 31 เปรียบเทียบ Instants Throughput ของ DSDV โพรโทคอลที่มีทอพอโลยีแตกต่างกัน

วิเคราะห์ผลการทดสอบ

ตารางที่ 1 เปรียบเทียบการทำงานของโพรโทคอลการค้นหาเส้นทางบนทอพอโลยีรูปแบบที่ 1 เมื่อกำหนดให้โหนดหมายเลข 3 (หรือตัวใดก็ได้ที่อยู่ในเส้นทางเชื่อมต่อ) เคลื่อนที่ออกในวินาทีที่ 180 จะทำให้เส้นทางขาดการเชื่อมต่อ และหยุดการจำลองการทำงานในวินาทีที่ 205 สำหรับในช่วงเวลาของโหนดที่มีการเคลื่อนที่ออกแสดงให้เห็นว่า โพรโทคอล DSDV มีจำนวนการสูญหายของข้อมูล (Packet Drops) สูงที่สุด เมื่อเปรียบเทียบกับโพรโทคอล AODV และ DSR สำหรับในส่วนของค่า Delay พบว่าโพรโทคอล DSDV มีค่าน้อยที่สุด เนื่องจากทราบเส้นทางที่แน่นอน ดังนั้นการจัดการเส้นทาง และการรักษาเส้นทางสำหรับโพรโทคอล DSDV จะใช้ทรัพยากรมากที่สุด เมื่อเปรียบเทียบกับโพรโทคอลตัวอื่นๆ

จากตารางที่ 2 เปรียบเทียบการทำงานของโพรโทคอลการค้นหาเส้นทางบนทอพอโลยีรูปแบบที่ 2 เมื่อกำหนดให้โหนดหมายเลข 3 เคลื่อนที่ออกในวินาทีที่ 180 หลังจากช่วงเวลานี้ระบบจะทำการหา

เส้นทางใหม่ เพื่อทำการส่งข้อมูลต่อไป จนกระทั่งถึงวินาทีที่ 300 ของการจำลอง เมื่อกำหนดให้โหนดหมายเลข 2 เคลื่อนที่ออก เป็นเหตุให้เส้นทางขาดการเชื่อมต่อและหยุดการจำลองการทำงานในวินาทีที่ 314 สำหรับโปรโตคอล DSDV มีจำนวนการสูญหายของข้อมูลสูงที่สุด ในขณะที่พารามิเตอร์ในค่าอื่นๆ มีแนวโน้มใกล้เคียงกัน

จากตารางที่ 3 เปรียบเทียบการทำงานของโปรโตคอลการค้นหาเส้นทางบนทอพอโลยีรูปแบบที่ 3 เมื่อให้โหนดหมายเลข 2 เคลื่อนที่ออก จะพบว่าระบบเครือข่ายก็ยังสามารถทำงานได้ สำหรับโปรโตคอล AODV มีจำนวนการสูญหายของข้อมูลสูงที่สุด ในขณะที่ค่าของ Delay นั้น โปรโตคอล DSR ใช้เวลามากที่สุด ส่วนการค้นหาเส้นทาง และการรักษาเส้นทางก็ยังพบว่า โปรโตคอล DSDV จะใช้ทรัพยากรมากที่สุด เนื่องจากมีการส่งข้อความตรวจสอบเส้นทางในทุกช่วงเวลา ตามลักษณะการทำงานของตัวโปรโตคอล

สรุปจากตารางที่ 1-3 พบว่าเมื่อเปรียบเทียบการทำงานของโปรโตคอลการค้นหาเส้นทาง หากโหนดอยู่กับที่การทำงานจะไม่มีปัญหา โดยสามารถจำลองการทำงานจนครบ 500 วินาที แต่เมื่อกำหนดให้มีบางโหนดเคลื่อนที่ออกจากเส้นทางจะพบว่าในทอพอโลยีที่ 1 สามารถส่งข้อมูลได้จนถึงวินาทีที่ 205 สำหรับทอพอโลยีที่ 2 สามารถส่งข้อมูลได้จนถึงวินาทีที่ 314 ส่วนในทอพอโลยีที่ 3 สามารถส่งข้อมูลได้จนถึงวินาทีที่ 500 ซึ่งมีค่าเท่ากับแบบจำลองที่ไม่มีการเคลื่อนที่ และในส่วนของค่าพารามิเตอร์ตัวอื่นๆ อย่างเช่น โปรโตคอล DSDV จะให้ค่า Delay ที่น้อยกว่าโปรโตคอลอื่น แต่ค่าของ Routing Packets และ Packet Drops จะมีปัญหามากกว่าโปรโตคอลตัวอื่น

จากรูปที่ 26 จะพบว่าค่าเฉลี่ย Delay ของโปรโตคอลการค้นหาเส้นทางของโปรโตคอลแต่ละตัว มีความแตกต่างกันชัดเจน เนื่องจากกระบวนการทำงานที่แตกต่างกัน โดยพบว่า ค่า Delay ของโปรโตคอล DSR มีค่าสูงที่สุดในทุกๆ ทอพอโลยีเมื่อเทียบกับโปรโตคอลตัวอื่น และ โปรโตคอล DSDV มีค่า Delay ต่ำที่สุด เนื่องจากในลักษณะในกระบวนการทำงานของโปรโตคอล DSDV ที่จะทราบเส้นทางที่จะใช้ในการรับ - ส่งข้อมูลอยู่ตลอด เนื่องจากการทำงานเป็นแบบ Proactive ดังนั้นเมื่อต้องการส่งข้อมูลจะสามารถส่งได้ในทันทีทำให้ค่า Delay น้อยกว่าโปรโตคอล AODV และ DSR เพราะการทำงานของโปรโตคอลทั้งสองเป็นแบบ Reactive คือเมื่อต้องการที่จะส่งข้อมูลจึงจะทำการหาเส้นทาง ในส่วนของโปรโตคอล DSR จะต้องรู้ข้อมูลของทั้งเส้นทางดังนั้นค่า Delay จึงสูงกว่าโปรโตคอลตัวอื่นๆ

จากรูปที่ 27 พบว่า Routing Packet ของโปรโตคอล AODV มีค่าน้อยที่สุดเนื่องจากจะทำการส่ง Packet ในการค้นหาเส้นทางเมื่อต้องการส่งข้อมูลเท่านั้น ซึ่งโดยปกติ DSDV จะต้องมีการส่ง Routing Packet มากที่สุดสังเกตได้จาก ทอพอโลยีที่ 1 และ 3 เนื่องจากตัวโปรโตคอลทำการแพร่กระจาย Routing Packet ตลอดคาบเวลา แต่ในทอพอโลยีที่ 2 จะพบว่า โปรโตคอล DSR มี Routing Packet ที่สูงกว่า DSDV เนื่องจากเมื่อโหนดเชื่อมต่อเคลื่อนที่ออก DSR จะทำการหาเส้นทางใหม่ จึงทำการแพร่กระจาย Routing Packet ออกมามากกว่าปกติเนื่องจากไม่สามารถหาเส้นทางใหม่ได้ทันที

จากรูปที่ 28 พบว่ารูปแบบของทอพอโลยีมีผลต่อความสำเร็จในการส่งข้อมูล โดยในทอพอโลยีรูปแบบที่ 1 จะพบว่ามีเปอร์เซ็นต์ในการส่งสำเร็จต่ำกว่า ทอพอโลยีรูปแบบอื่น โดยทอพอโลยีในรูปแบบที่ 3 จะมีเปอร์เซ็นต์ในการส่งสำเร็จสูงที่สุดเนื่องจากในลักษณะของทอพอโลยีเมื่อโหนดเคลื่อนที่ออกไปเพียง

บางโหนดระบบเครือข่ายยังสามารถทำงานได้เนื่องจากสามารถเปลี่ยนไปใช้เส้นทางอื่นได้ ซึ่งเมื่อเปรียบเทียบกับทุก ๆ ทอพอโลยีค่าของความสำเร็จของแต่ละโพรโทคอลควรมีความใกล้เคียงกัน แต่ในทอพอโลยีรูปแบบที่ 3 พบว่า โพรโทคอล AODV มีเปอร์เซ็นต์ในการส่งสำเร็จน้อยกว่าโพรโทคอลทั้งสอง เนื่องจากเมื่อให้โหนดเชื่อมต่อเคลื่อนที่ออก โพรโทคอล AODV ไม่สามารถหาเส้นทางใหม่ได้ในทันที จึงส่งผลกระทบต่อประสิทธิภาพของระบบเครือข่าย

จากรูปที่ 29 ถึง 31 จะพบว่าโพรโทคอลแต่ละตัวจะมีความเหมาะสมกับทอพอโลยีแต่ละรูปแบบที่แตกต่างกัน โดยพบว่าในทอพอโลยีรูปแบบที่ 1 ซึ่งการเรียงตัวกันของโหนดไม่มีความซับซ้อนมากนัก เมื่อมีโหนดเคลื่อนที่ออกจากจุดเดิม เป็นเหตุให้ขาดการเชื่อมต่อ เนื่องจากไม่สามารถเชื่อมต่อกับโหนดข้างเคียงโหนดอื่นได้

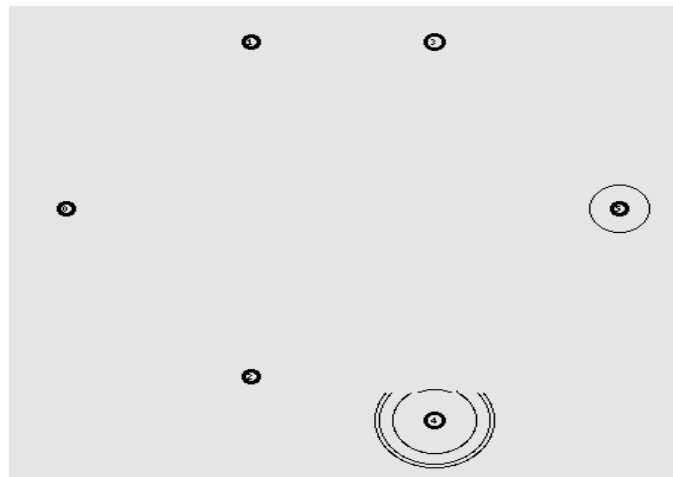
ในทอพอโลยีรูปแบบที่ 2 หากโหนดหมายเลข 3 เคลื่อนที่ออก ระบบเครือข่ายจะยังสามารถใช้งานได้ โดย AODV และ DSDV สามารถทำงานได้ปกติโดยเมื่อโหนดเชื่อมต่อเคลื่อนที่ออก AODV และ DSDV สามารถหาเส้นทางใหม่ได้ทันที แต่ในโพรโทคอล DSR สามารถหาเส้นทางใหม่ได้แต่จะมีการขึ้นลงของ Throughput ไม่คงที่ก่อนที่จะเปลี่ยนไปใช้เส้นทางใหม่เมื่อมีโหนดเคลื่อนที่ออกดังรูปที่ 32 โดยโพรโทคอลหาเส้นทางทั้ง 3 สามโพรโทคอลจะทำการหาเส้นทางใหม่โดยใช้เส้นทางผ่านโหนดหมายเลข 4 แทน แต่หากให้โหนดที่ 4 เคลื่อนที่ออกจากเส้นทางระบบเครือข่ายจะไม่สามารถใช้ทำงานได้เนื่องจากโหนดต้นทางจะไม่สามารถเชื่อมต่อกับโหนดปลายทางได้



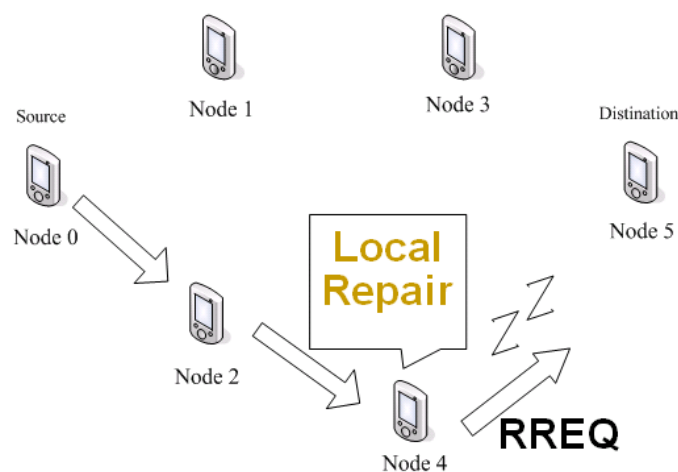
รูปที่ 32 แสดงการเคลื่อนที่ออกของโหนดเชื่อมต่อในทอพอโลยีรูปแบบที่ 2

ในทอพอโลยีรูปแบบที่ 3 เมื่อมีบางโหนดเคลื่อนที่ออกดังรูปที่ 33 พบว่าโพรโทคอล DSR สามารถทำงานได้ปกติเมื่อเส้นทางขาดการเชื่อมต่อสามารถหาเส้นทางใหม่ได้ทันที แต่โพรโทคอล DSDV มีกราฟของ Throughput เป็นศูนย์ชั่วขณะในขณะที่โพรโทคอลการค้นหาเส้นทางกำลังทำการค้นหาเส้นทางในการส่งข้อมูลเส้นทางใหม่ ในส่วนของโพรโทคอล AODV ไม่สามารถหาเส้นทางใหม่ได้ทันทีเช่นกันแต่ใช้เวลาในการหาเส้นทางใหม่นานที่สุดเนื่องจากเมื่อโหนดที่สี่ เคลื่อนที่ออกจากเส้นทางเดิมทำให้ไม่สามารถส่งข้อมูลผ่านไปยังโหนดที่ 5 ซึ่งเป็นโหนดปลายทางได้ ในโพรโทคอล AODV เมื่อโหนดขาดการเชื่อมต่อ จะทำการตรวจสอบว่าจุดที่เส้นทางขาด อยู่ใกล้กับโหนดต้นทางมากกว่าโหนด

ปลายทางหรือไม่ หากจุดที่เส้นทางขาดอยู่ใกล้กับโหนดต้นทางมากกว่าโหนดปลายทาง โปรโตคอล AODV จะทำการ Route Discovery โดยให้โหนดต้นทางทำการหาเส้นทางใหม่โดยจะทำการ Broadcast ข้อความ Route Request ที่โหนดต้นทาง แต่ถ้าหากจุดที่เส้นทางขาดอยู่ใกล้โหนดปลายทางมากกว่า จะให้โหนดที่ตรวจพบเส้นทางขาด ทำการ Broadcast ข้อความ Route Request แทนเรียกว่าการทำ Local Repair ซึ่งจะทำให้ Overhead ของระบบลดลงเนื่องจากไม่ต้องทำการ Broadcast ที่ต้นทาง ดังรูป 25 จะพบว่าโหนดที่สี่ตรวจพบเส้นทางขาด และพบว่าจุดที่เส้นทางขาดอยู่ใกล้กับโหนดปลายทางมากกว่า ดังนั้นโหนดที่สี่จะทำการ Broadcast ข้อความ Route Request แต่ที่โหนดใกล้เคียงโหนดที่สี่มีแค่โหนดเดียวและไม่สามารถส่งข้อมูลไปยังโหนดปลายทางได้ จึงทำให้ Throughput ลดต่ำลงในขณะที่ โหนดที่สี่กำลังทำ Local Repair ทำให้กราฟของค่า Throughput เป็นศูนย์เป็นระยะเวลานานกว่าจะหาเส้นทางใหม่ได้



รูปที่ 33 แสดงการเคลื่อนที่ออกของโหนดเชื่อมต่อในทอพอโลยีรูปแบบที่ 3



รูปที่ 34 การทำ Local Repair

4.2 ทดสอบการทำ Cross layer ด้วยระบบ RMS

การจำลองการทำงานของโพรโทคอลการค้นหาเส้นทางผ่านโปรแกรม NS-2 ด้วยวิธีการทำ Cross layer ส่วนรูปแบบการเคลื่อนที่ของโหนดจะเป็นแบบ Random waypoint ซึ่งจะแตกต่างจากการทดสอบในหัวข้อ 4.1 ที่ใช้โหนดเพียง 6 โหนด เพื่อเป็นการทดสอบอย่างง่าย สำหรับการทดสอบในส่วนนี้ งานวิจัยต้องการให้โหนดเคลื่อนที่ตามรูปแบบ Random waypoint และการทดสอบแต่ละความเร็ว เรา จะใช้การเคลื่อนที่ที่แตกต่างกัน 10 ครั้ง แล้วนำมาหาค่าเฉลี่ย

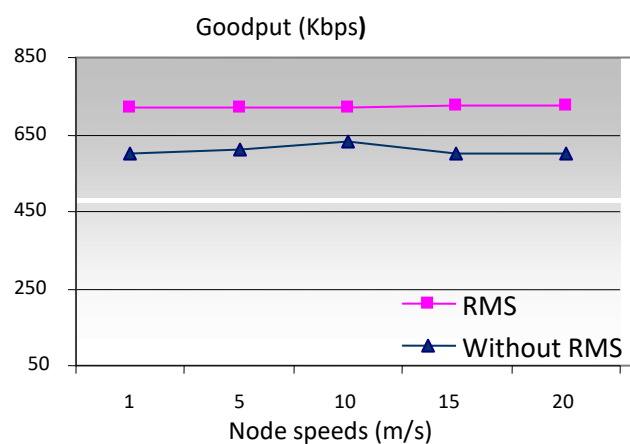
พารามิเตอร์ที่ใช้ในการจำลอง

ในการกำหนดค่าพารามิเตอร์จะทำการแก้ไขไฟล์ .tcl ซึ่งอยู่ในโปรแกรม NS-2 โดยในการทดสอบจำกำหนดค่าพารามิเตอร์ต่างๆ ดังนี้

| | |
|----------------------|---------------------|
| จำนวนโหนด | 50 โหนด |
| จำนวน Connection | 10 คู่ |
| ระยะห่างของแต่ละโหนด | 250 เมตร |
| ระยะเวลาในการจำลอง | 1000 วินาที |
| ชนิดข้อมูลที่ส่ง | Constant bit rate |
| ขนาดของแพ็คเก็ต | 512 ไบต์ |
| ขนาดของทอพอโลยี | 1000x1000 ตารางเมตร |
| Traffic Interval | 0.1 |
| ทำงานบน MAC 802.11 | |

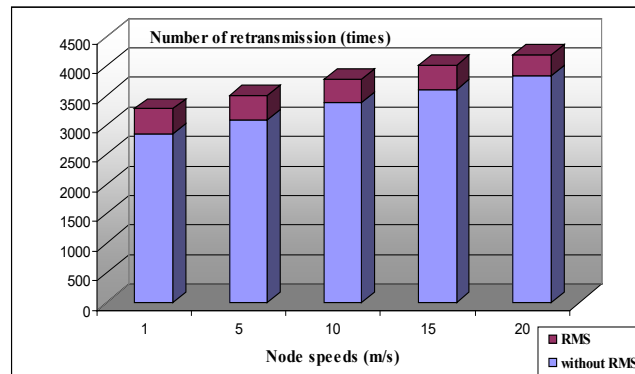
ผลการทดสอบ

ค่าสมรรถนะการทำงานของระบบ เมื่อทดสอบเปรียบเทียบกันระหว่างการทำงานแบบ RMS และโพรโทคอลมาตรฐาน พบว่าการทำงานแบบ RMS สามารถให้ค่า goodput ที่สูงกว่าโพรโทคอลมาตรฐาน ซึ่งค่านี้เป็นอัตราข้อมูลที่รับได้ที่ปลายทาง เทียบกับเวลาดังแสดงในรูปที่ 35

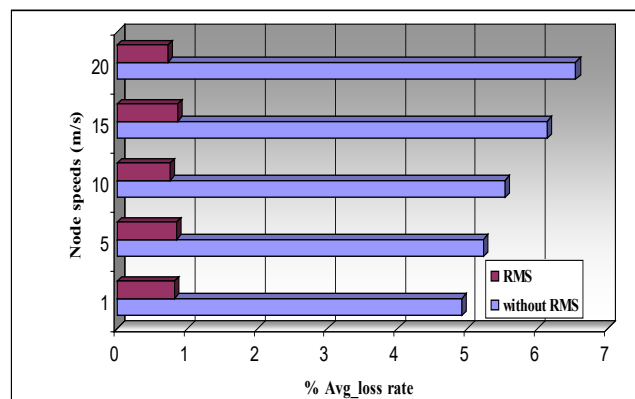


รูปที่ 35 เปรียบเทียบค่าสมรรถนะของระบบ

เมื่อทำการวัดอัตราการส่งแพ็กเก็ตที่ระดับชั้น Transport พบว่าการทำงานแบบ RMS ทำการส่งแพ็กเก็ตช้ากว่าโปรโตคอลมาตรฐาน ซึ่งแสดงให้เห็นถึงประสิทธิภาพที่มีต่อการสูญหายของข้อมูลในระดับชั้น Network และ MAC แสดงผลการทดสอบดังรูปที่ 36 ส่วนการวัดการสูญหายของข้อมูลได้วัดที่ระดับชั้น Network ซึ่งได้แสดงดังรูปที่ 37 RMS จะช่วยให้ระบบทำงานได้ดีขึ้นเมื่อมีการเปลี่ยนเส้นทางการติดต่อสื่อสาร และปัญหาการขาดหายของสัญญาณ



รูปที่ 36 จำนวนการส่งแพ็กเก็ตซ้ำ



รูปที่ 37 อัตราการสูญหายของข้อมูล

จากผลการทดสอบดังกล่าวแสดงให้เห็นว่า RMS สามารถที่จะช่วยลดปัญหาการสูญหายของข้อมูล ดังนั้นวิธีการแบบ Cross layer ก็เป็นวิธีหนึ่งที่เหมาะสมกับการเพิ่มประสิทธิภาพให้กับการทำงานแบบเครือข่ายไร้สายแบบ Ad hoc

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการทดสอบแสดงให้เห็นว่า ทอพอโลยีแต่ละรูปแบบ มีผลต่อการเชื่อมต่อของระบบเครือข่ายแบบ Ad hoc โดยพบว่าทอพอโลยีรูปแบบที่ 1 หากให้โหนดเคลื่อนที่ออกเพียงโหนดใดโหนดหนึ่งจะขาดการเชื่อมต่อทันที แต่ทอพอโลยีรูปแบบที่ 2 และ 3 โหนดเชื่อมต่อสามารถเคลื่อนที่ออกได้มากกว่าหนึ่งโหนด ก็ยังไม่ส่งผลกระทบต่อการทำงานของระบบมากนัก และจากการทดสอบพบว่า โพรโทคอลแต่ละตัวมีความเหมาะสม-สมของการใช้งานในทอพอโลยีที่แตกต่างกัน เช่นในทอพอโลยีรูปแบบที่ 2 จะเกิดปัญหาค่าเฉลี่ยปริมาณงานไม่คงที่กับโพรโทคอล DSR และ DSDV แต่ทอพอโลยีรูปแบบที่ 3 จะเกิดปัญหากับโพรโทคอล AODV

สำหรับงานวิจัยส่วนใหญ่ [19] และ [20] จะทำการเปรียบเทียบแต่ละโพรโทคอลของการค้นหาเส้นทางเท่านั้น แต่ไม่ได้ชี้ให้เห็นถึงปัญหาของเวลาที่ใช้ในการหาเส้นทางใหม่ ดังนั้นจึงเป็นจุดเด่นของงานวิจัยนี้ที่แตกต่างจากงานวิจัยอื่น ส่วนโพรโท-คอลในการค้นหาเส้นทาง กับทอพอโลยีของเครือข่าย จะเห็นว่ามีความสัมพันธ์กัน และเวลาที่ใช้ในการค้นหาเส้นทางใหม่ จะมีผลกระทบต่อเครือข่าย Ad hoc ยิ่งใช้เวลาในการหาเส้นทางใหม่มาก ก็จะมีผลให้เกิดการสูญหายของข้อมูลมากขึ้นเช่นกัน นอกจากนี้รูปแบบของทอพอโลยีที่นำมาทดสอบ ต้องการชี้ให้เห็นชัดเจนระหว่างเกิดปัญหาทันที เมื่อโหนดเคลื่อนที่ออกจากเส้นทางหลัก กับในกรณีที่มีเส้นทางอีกเส้นทางหนึ่งมาช่วยส่งข้อมูลแทน เมื่อเส้นทางหลักเกิดปัญหาขึ้น

งานวิจัยนี้ได้แก้ปัญหาโพรโทคอลการค้นหาเส้นทางเมื่อเส้นทางหลักไม่สามารถใช้งานได้ โดยจะทำให้การใช้เวลาของการค้นหาเส้นทางใหม่ลดลง กับวิธีการที่เรียกว่า Cross layer system [21] และ [22] ในรูปแบบที่ได้ออกแบบไว้ ซึ่งได้แก่ Real Multiroute System (RMS) [23]

5.2 ข้อเสนอแนะ

ทอพอโลยีที่ใช้ในการทดสอบมีจำนวน 3 ทอพอโลยีที่แตกต่างกัน แต่สามารถทำการทดสอบเพิ่มเติมโดยใช้ทอพอโลยีอื่นๆได้อีก ซึ่งในส่วนของงานวิจัยนี้ได้แสดงให้เห็นในส่วนของการทำงานของโพรโทคอลค้นหาเส้นทาง และความสำคัญของโหนดเชื่อมต่อในระบบเครือข่ายที่มีผลกระทบต่อการทำงานของโพรโทคอลหาเส้นทาง นอกจากนี้ควรพัฒนาระบบให้สามารถนำไปใช้ได้จริง

บรรณานุกรม

1. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, IETF, January 1999.
2. S. Bansal, R. Gupta, R. Shorey, A. Misra, I. Ali and A. Razdan, "Energy efficiency and throughput for TCP traffic in multi-hop wireless networks", The twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM), New York, 2002.
3. S. Charoenpanyasak, B. Paillassa and F. Jaddi, "Experimental Study on TCP Enhancement Interest in Ad Hoc Networks", IEEE International Conference on Wireless and Mobile Communications (ICWMC), Guadeloupe, March 2007.
4. S. Mascolo, C. Casetti, M. Gerla, S.S. Lee and M. Sanadidi, "TCP Westwood: congestion control with faster recovery", The 21st Century Military Communications Conference (MILCOM), Los Angeles California USA, October 2000.
5. S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", RFC 2883, IETF, July 2000.
6. C. Perkins, E. Belding-Royer and S. Das, "Ad hoc on-demand distance vector (AODV) routing", RFC 3561, IETF, July 2003.
7. E. Borgia, M. Conti and F. Delmastro, "MobileMAN: Design, Integration and Experimentation of Cross-layer Mobile Multi-hop Ad Hoc Networks", IEEE Communication Magazine, Ad Hoc sensor Network Series, vol. 44, no. 7, July 2006.
8. M. Conti, G. Maselli, G. Turi and S. Giordano, "Cross layering in mobile Ad hoc network design", IEEE Computer Society, pp. 48-51, February 2004.
9. M.A. Razzaque, S. Dobson and P. Nixon, "Cross-layer architectures for autonomic communications", Journal of Network and Systems Management, pp. 13-27. March 2007.

10. V. Srivastava and M. Motani, "Cross-layer design: a survey and the road ahead", IEEE communication magazine, vol. 43, no. 12, pp. 112-119, December 2005.
11. Charles E. Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," SIGCOMM, August 1994.
12. Charles E. Perkins, Elisabeth M. Belding-Royer and Samir R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003.
13. David B. Johnson, David A. Maltz and Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," RFC 4728, February 2007.
14. T. Larsson and Y. Liu, "Study of EDCA and DCF in Multihop Ad Hoc," Networks Department of Department of Electrical and Information Technology Signal and System Lund Institute of Technology, March 13, 2008.
15. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANETs) Routing Protocol Performance Issue and Evaluation Consideration," RFC 2501, January 1999.
16. Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1999, The Institute of Electrical and Electronics Engineers, NY, 1999.
17. W. Christopher, W. Tadeusz and C. Joe, "Hidden terminal jamming problems in IEEE 802.11 mobile ad hoc networks," IEEE International Conference on Communications, 2001.
18. NS2 website; "<http://www.isi.edu/nsnam/ns>".
19. Saleh Ali K. and S. Putra, "An Overview of mobile Ad hoc Networks for The Existing Protocol and Applications," Journal on Application of Graph Theory in Wireless Ad hoc Networks and Sensor Networks, USA, March 2010.

20. N. Surayati Mohamad Usop, A. Abdullah and A. Faisal Amri Abidin, "Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment", International Journal of Computer Science and Network Security (IJCSNS), Korea, July 2009.
21. D. Felice and M. Relatore, "Cross-Layer Optimizations in Multi-HopAd Hoc Networks," Dottorato di Ricerca, March, 2008.
22. M. Conti, G. Maselli, and G. Turi, "Cross-layering in mobile ad hoc network design," Network Computer, 2004.
23. S. Charoenpanyasak, "Real Multiroute System (RMS) for Mobile Ad hoc Networks: A Cross layer Approach," ACM The 7th International Conference on Advances in Mobile Computing & Multimedia (MoMM), Kuala Lumpur, Malaysia, December 2009.

ภาคผนวก ก

ในการจำลองการทำงานใช้โปรแกรม NS2 ในการจำลองโดยใช้ File Tcl ในการเซตค่าพารามิเตอร์ต่างๆ ตัวอย่าง code Tcl ไฟล์ที่ใช้ในการจำลอง

กำหนดค่าพารามิเตอร์ที่ใช้ในการทดสอบเริ่มต้น

```
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy           ;# network interface type
set val(mac)       Mac/802_11                ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)        LL                        ;# link layer type
set val(ant)       Antenna/OmniAntenna       ;# antenna model
set val(ifqlen)    50                        ;# max packet in ifq
set val(nn)        6                         ;# number of mobilenodes
set val(rp)        AODV                      ;# routing protocol
set val(x) 2000    ;# X dimension of the topography in meters
set val(y) 2000    ;# Y dimension of the topography in meters
set val(time)      500.0                     ;# Simulation time in seconds
set data_interval  0.1                       ;# CBR Traffic Interval
set start_time     60.0                      ; # traffic start time
```

เซตค่า Out put ของไฟล์

```
# Tracde Files
set tracefile [open 6aodv03.tr w]
$ns_ use-newtrace
$ns_ trace-all $tracefile
set namfile [open 6aodv03.nam w]
$ns_ namtrace-all-wireless $namfile $val(x) $val(y)
# set up topography object
set topo      [new Topography]
$topo load_flatgrid $val(x) $val(y)
กำหนดรูปแบบของ Topology
```

```
# Provide initial (X,Y, for now Z=0) co-ordinates for mobilenodes
# and produce some simple node movements
```

```
$node_(0) set X_ 0.0
$node_(0) set Y_ 600.0
$node_(0) set Z_ 200.0
```

```
$node_(1) set X_ 350.0
$node_(1) set Y_ 1000.0
$node_(1) set Z_ 200.0
```

```
$node_(2) set X_ 350.0
$node_(2) set Y_ 200.0
$node_(2) set Z_ 200.0
```

```
$node_(3) set X_ 700.0
$node_(3) set Y_ 1000.0
$node_(3) set Z_ 200.0
```

```
$node_(4) set X_ 700.0
$node_(4) set Y_ 200.0
$node_(4) set Z_ 200.0
```

```
$node_(5) set X_ 1050.0
$node_(5) set Y_ 600.0
$node_(5) set Z_ 200.0
```

เซตการเคลื่อนที่ของโหนดที่วินาทีที่ 180 ของการจำลอง กำหนดให้โหนดที่สี่เคลื่อนที่ไปยังจุดที่กำหนด และในวินาทีที่ 300 ของการจำลอง กำหนดให้โหนดที่ 2 เคลื่อนที่ไปยังจุดที่กำหนด

```
$ns_ at 180.0 "$node_(4) setdest 700.0 1.0 20.0"
$ns_ at 300.0 "$node_(2) setdest 350.0 1.0 20.0"
```

เซตค่า การส่งข้อมูลของ โหนดเริ่มต้นและโหนดปลายทาง

กำหนดให้โหนดศูนย์เป็นโหนดเริ่มต้น และโหนดที่ ห้าเป็นโหนดปลายทาง ส่งข้อมูลแบบ Constant Bit rate โดยที่มีขนาดของแพ็คเกต 512 Bytes

```
# Data Source , Nodes
```

```
set udp_(0) [new Agent/UDP]
```

```
$udp_(0) set fid_ 0
```

```
$ns_ attach-agent $node_(0) $udp_(0)
```

```
# Null Agent to receive Packets for Node 0
```

```
set null_(0) [new Agent/Null]
```

```
$null_(0) set fid_ 1
```

```
$ns_ attach-agent $node_(5) $null_(0)
```

```
//ทำการกำหนดขนาดของแพ็คเกต
```

```
# Constant Bit rate traffic genrator
```

```
set cbr_(0) [new Application/Traffic/CBR]
```

```
$cbr_(0) set packetSize_ 512
```

```
$cbr_(0) set interval_ $data_interval
```

```
$cbr_(0) set random_ 0
```

```
$cbr_(0) set maxpkts_ 10000
```

```
$cbr_(0) attach-agent $udp_(0)
```

```
$ns_ connect $udp_(0) $null_(0)
```

```
$ns_ at $start_time "$cbr_(0) start"
```

ภาคผนวก ข

คำอธิบายสำหรับไฟล์ .tr ที่เป็นรูปแบบ new trace

รูปแบบของ new trace format จะถูกแบ่งออกเป็นฟิลด์ข้อมูลต่างๆ :

ชนิดของเหตุการณ์และคำอธิบาย

ข้อมูลตัวแรก

| | |
|---|-----------------------|
| s | โหนดทำการส่งข้อมูล |
| r | โหนดทำการรับข้อมูล |
| d | โหนดทำการ drop packet |
| f | ทำการส่งต่อ |

Tag ทั่วไป

ในส่วนของ ฟิลด์ที่สอง จะขึ้นต้นด้วย "-t "

| | |
|----|--------------|
| -t | เวลาที่จำลอง |
|----|--------------|

รายละเอียดของโหนด "-N" and are listed as below:

| | |
|------|---|
| -Ni: | node id |
| -Nx: | จุดพิกัดในแนวแกน x ที่โหนดอยู่ |
| -Ny: | จุดพิกัดในแนวแกน y ที่โหนดอยู่ |
| -Nz: | จุดพิกัดในแนวแกน z ที่โหนดอยู่ |
| -Ne: | ระดับพลังงานของโหนด |
| -NI: | การทำงานอยู่ในระดับชั้นไหน |
| | AGT ระดับชั้น Transport |
| | RTR ระดับชั้น Routing |
| | MAC ระดับชั้น MAC |
| -Nw: | คำอธิบายเกี่ยวกับสาเหตุของการ Drop ได้แก่ |
| : | "END" DROP_END_OF_SIMULATION |
| | "COL" DROP_MAC_COLLISION |
| | "DUP" DROP_MAC_DUPLICATE |
| | "ERR" DROP_MAC_PACKET_ERROR |
| | "RET" DROP_MAC_RETRY_COUNT_EXCEEDED |

| | |
|--------|---|
| "STA" | DROP_MAC_INVALID_STATE |
| "BSY" | DROP_MAC_BUSY |
| "NRTE" | DROP_RTR_NO_ROUTE i.e no route is available. |
| "LOOP" | DROP_RTR_ROUTE_LOOP i.e there is a routing loop |
| "TTL" | DROP_RTR_TTL i.e TTL has reached zero. |
| "TOUT" | DROP_RTR_QTIMEOUT i.e packet has expired. |
| "CBK" | DROP_RTR_MAC_CALLBACK |
| "IFQ" | DROP_IFQ_QFULL i.e no buffer space in IFQ. |
| "ARP" | DROP_IFQ_ARP_FULL i.e dropped by ARP |
| "OUT" | DROP_OUTSIDE_SUBNET |

ข้อมูลของแพ็คเก็ตที่ระดับชั้น IP

จะเริ่มต้นด้วย "-I" :

- Is: ที่อยู่ของโหนดต้นทาง
- Id: ที่อยู่ของโหนดปลายทาง
- It: ชนิดของแพ็คเก็ต
- Il: ขนาดของแพ็คเก็ต
- If: flow id
- Ii: unique id
- Iv: ค่า TTL

ข้อมูล Next hop

เริ่มต้นด้วย "-H".

- Hs: id ของโหนด
- Hd: id ของโหนดถัดไป

ข้อมูลของแพ็คเก็ตที่ระดับชั้น MAC

เริ่มต้นด้วย "-M":

- Ma: duration
- Md: ethernet address ของโหนดปลายทาง
- Ms: ethernet address ของโหนดต้นทาง
- Mt: ชนิดของ ethernet

ข้อมูลของแพ็คเก็ตที่ระดับชั้น Application "

-P arp มีรายละเอียดดังนี้ :

- Po: ARP Request/Reply
- Pm: mac address ของโหนดต้นทาง
- Ps: address ของโหนดต้นทาง
- Pa: mac address ของโหนดปลายทาง
- Pd: address ของโหนดปลายทาง

-P dsr

ข้อมูลของ Dynamic source routing. Information มีรายละเอียดดังนี้ :

- Pn: จำนวนโหนดที่ผ่านมา
- Pq: routing request flag
- Pi: route request sequence number
- Pp: routing reply flag
- Pl: reply length
- Pe: src of srcrouting->dst of the source routing
- Pw: error report flag ?
- Pm: จำนวน errors
- Pc: รายงาน
- Pb: link error จาก linka->linkb

-P cbr

แสดงรายละเอียด เมื่อส่งข้อมูล Constant bit rate :

- Pi: sequence number
- Pf: จำนวนที่ส่งแพ็คเกต
- Po: จำนวนที่ forwards

-P tcp แสดงข้อมูลที่มีการส่งแบบ TCP :

- Ps: seq number
- Pa: ack number
- Pf: เวลาที่แพ็คเกตถูกส่ง

ตัวอย่างของไฟล์ .tr

M 0.00000 0 (0.00, 600.00, 200.00), (10.00, 10.00), 0.00

M 0.00000 1 (350.00, 1000.00, 200.00), (10.00, 10.00), 0.00

M 0.00000 2 (350.00, 200.00, 200.00), (10.00, 10.00), 0.00

M 0.00000 3 (700.00, 1000.00, 200.00), (10.00, 10.00), 0.00

M 0.00000 4 (700.00, 200.00, 200.00), (10.00, 10.00), 0.00

M 0.00000 5 (1050.00, 600.00, 200.00), (10.00, 10.00), 0.00

s -t 5.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI AGT -
Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -ls 0.0 -ld 5.0 -lt cbr -ll 1000 -lf 0 -li 0 -lv 32 -Pn cbr -Pi 0 -Pf 0
-Po 0

r -t 5.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI RTR -
Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -ls 0.0 -ld 5.0 -lt cbr -ll 1000 -lf 0 -li 0 -lv 32 -Pn cbr -Pi 0 -Pf 0
-Po 0

s -t 5.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI AGT -
Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -ls 0.0 -ld 5.0 -lt cbr -ll 24 -lf 0 -li 1 -lv 32 -Pn cbr -Pi 1 -Pf 0 -
Po 0

r -t 5.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI RTR -
Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -ls 0.0 -ld 5.0 -lt cbr -ll 24 -lf 0 -li 1 -lv 32 -Pn cbr -Pi 1 -Pf 0 -
Po 0

s -t 5.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI RTR -
Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -ls 0.255 -ld -1.255 -lt AODV -ll 48 -lf 0 -li 0 -lv 30 -P aodv -Pt
0x2 -Ph 1 -Pb 1 -Pd 5 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

s -t 5.000535000 -Hs 0 -Hd -2 -Ni 0 -Nx 0.00 -Ny 600.00 -Nz 200.00 -Ne -1.000000 -NI MAC -
Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -ls 0.255 -ld -1.255 -lt AODV -ll 106 -lf 0 -li 0 -lv 30 -P
aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 5 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 5.001384772 -Hs 1 -Hd -2 -Ni 1 -Nx 350.00 -Ny 1000.00 -Nz 200.00 -Ne -1.000000 -NI
MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -ls 0.255 -ld -1.255 -lt AODV -ll 48 -lf 0 -li 0 -lv 30
-P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 5 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

r -t 5.001384772 -Hs 2 -Hd -2 -Ni 2 -Nx 350.00 -Ny 200.00 -Nz 200.00 -Ne -1.000000 -NI MAC
-Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 800 -ls 0.255 -ld -1.255 -lt AODV -ll 48 -lf 0 -li 0 -lv 30 -P
aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 5 -Pds 0 -Ps 0 -Pss 4 -Pc REQUEST

ภาคผนวก ค

1. Waris Chanei and S. Charoenpanyasak, "Comparative Analysis of topologies and routing protocols for Wireless Ad hoc Networks", The 6th National Conference on Computing and Information Technology, Bangkok, Thailand, June 3-5, 2010.
2. S. Charoenpanyasak, "Real Multiroute System (RMS) for Mobile Ad hoc Networks: A Cross layer Approach", ACM The 7th International Conference on Advances in Mobile Computing & Multimedia (MoMM), Kuala Lumpur, Malaysia, December 14-16, 2009.
3. S. Charoenpanyasak, "ORDERED VS NON-ORDERED messages in SCTP multihoming over Ad hoc networks", IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009), Barcelona, Spain, November 4 - 6, 2009.