การวิเคราะห์เพื่อปรับปรุงประสิทธิภาพเครือข่ายไร้สาย

Analysis for Optimize Performance Wireless Network

โดย

นาย สุรสิทธิ์ มานะกสิกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

การวิเคราะห์เพื่อปรับปรุงประสิทธิภาพเครือข่ายไร้สาย

Analysis for Optimize Performance Wireless Network

โดย

นาย สุรสิทธิ์ มานะกสิกิจ 47015032

อาจารย์ที่ปรึกษา

ผศ. นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์เพื่อปรับปรุงประสิทธิภาพเครือข่ายไร้สาย

Analysis for Optimize Performance Wireless Network

ผู้จัดทำ

1. นาย สุรสิทธิ์ มานะกสิกิจ 47015032

.....อาจารย์ที่ปรึกษา

(ผศ. นภัทร สระเอี่ยม)

การวิเคราะห์เพื่อปรับปรุงประสิทธิภาพเครือข่ายไร้สาย

Analysis for Optimize Performance Wireless Network

โดย นาย สุรสิทธิ์ มานะกสิกิจ 47015032

อาจารย์ที่ปรึกษา ผศ. นภัทร สระเอี่ยม

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการวิเคราะห์โปรโตคอลทีซีพี/ไอพี เพื่อปรับปรุงประสิทธิภาพการใช้งานเครือข่ายไร้สาย เนื่องจากการสื่อสารข้อมูลทั่วไปในปัจจุบันเป็นการสื่อสาร โดยระบบเครือข่าย ในอนาคตเครือข่ายไร้สายจะมีแนวโน้มใช้งานที่มาก ดังนั้นการใช้งานเครือข่ายไร้สายที่มีประสิทธิภาพจึงต้องมีการศึกษาการวิเคราะห์ระบบเครือข่ายไร้สายบนโปรโตคอลทีซีพี/ไอพี ที่เหมาะสม

ABSTRACT

This Thesis is Analysis for Optimize Performance Wireless Network. Nowadays, Trend of Data Communication usage is moving toward to the Wireless Network System, So, We will study about the performance optimum of TCP/IP Network in the Wireless environment.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 แนวความคิดของปริญญานิพนธ์

ปัจจุบันความต้องการในการสื่อสารข้อมูลเพิ่มมากขึ้น เพิ่มทั้งปริมาณข้อมูลและรูปแบบข้อมูลในการรับ – ส่งที่มากขึ้น จากเดิมการสื่อสารข้อมูลที่จำกัดอยู่ตามองค์กร เนื่องจากค่าใช้จ่ายสูงเพราะการแข่งขันในทางธุรกิจยังมีไม่มากและเทคโนโลยีที่พัฒนาให้ใช้งานสำหรับการสื่อสารผู้ใช้จำนวนน้อย แต่ในปัจจุบันการแข่งขันทางด้านธุรกิจมีมากทำให้ค่าใช้จ่ายในการสื่อสารลดลง ทางด้านเทคโนโลยีก็มีการพัฒนาให้รองรับผู้ใช้จำนวนมากขึ้น ในปัจจุบันการสื่อสารข้อมูลบนระบบเครือข่ายเป็นที่นิยมใช้กันมาก และภายใต้การพัฒนาเทคโนโลยีการสื่อสารข้อมูล ระบบเครือข่ายใช้สายแบบดั้งเดิมกำลังจะถูกแทนที่ด้วยระบบเครือข่ายไร้สาย เนื่องจากสามารถอำนวยความสะดวกในการให้กับผู้ใช้เพราะสามารถให้บริการได้ทุกสถานที่ทุกเวลาความก้าวหน้าของเทคโนโลยีการสื่อสารแบบไร้สาย และความสามารถที่เพิ่มขึ้นของอุปกรณ์คำนวณที่สามารถพกพาได้ เช่น โน้ตบุ๊กคอมพิวเตอร์ พีดีเอ มีราคาถูกลงและอัตราการส่งข้อมูลเพิ่มขึ้น ทำให้การสร้างเน็ตเวิร์คแบบไร้สาย โครงสร้างพื้นฐานหรือเน็ตเวิร์คไร้สายแบบเฉพาะกิจ (Ad-hoc) มีความน่าสนใจเพิ่มขึ้นมากและระบบโครงข่ายไร้สายมีการลงทุนค่าใช้จ่ายน้อยเมื่อเทียบกับระบบเครือข่ายใช้สาย

จากการพัฒนาเทคโนโลยีการสื่อสารข้อมูลระบบเครือข่ายไร้สายในปัจจุบันจึงเกิดการเปลี่ยนแปลงทางด้านเทคโนโลยีที่มีลักษณะการทำงานที่รองรับจำนวนผู้ใช้มาก ดังนั้นการใช้งานระบบที่มีความน่าเชื่อถือ มีประสิทธิภาพต่อการใช้งาน เป็นสิ่งสำคัญในการสื่อสารข้อมูล จึงเหมาะสมอย่างมากที่มีการศึกษา การวิเคราะห์โพรโตคอลเลือกเส้นทางบนระบบเครือข่ายไร้สาย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ในการทำปริญญานิพนธ์เรื่องการวิเคราะห์เพื่อปรับปรุงประสิทธิภาพเครือข่ายไร้สาย ได้ศึกษาถึงรูปแบบการสื่อสารข้อมูลของชุดโพรโตคอลทีซีพี/ไอพี และได้กำหนดจุดประสงค์ไว้ดังนี้

- เพื่อศึกษาโพรโตคอลเลือกเส้นทาง
- เพื่อศึกษาการจำลองระบบเครือข่ายไร้สาย
- เพื่อนำข้อมูลที่ได้จากการจำลองระบบมาใช้ในการวิเคราะห์ปัญหาที่เกิดขึ้นในระบบเครือข่ายได้

1.3 ขอบเขตของปริญญานิพนธ์

จะเป็นการศึกษารายละเอียดของเครือข่ายไร้สายและลักษณะโปรโตคอลการเลือกเส้นทาง รวมทั้งศึกษาการใช้งานโปรแกรม NS และทำการจำลองการทำงานของระบบเครือข่ายโดยใช้โปรแกรม NS แล้วนำข้อมูลที่ได้จากการจำลองระบบมาไปใช้ในการวิเคราะห์ ปัญหาที่เกิดขึ้นในระบบเครือข่าย

1.4 รายละเอียดในปริญญานิพนธ์

ในปริญญานิพนธ์นี้ ได้แบ่งเนื้อหาออกเป็นบทได้ทั้งหมด 5 บทโดยในบทที่ 1 จะเป็นการกล่าวนำถึง แนวความคิด วัตถุประสงค์และขอบเขตในการทำปริญญานิพนธ์ และได้กล่าวถึงเนื้อหาโดยย่อของแต่ละบท ซึ่งในบทอื่นๆจะมีเนื้อหาดังนี้

บทที่ 2 กล่าวถึงทฤษฎีและหลักการ ระบบเครือข่ายไร้สาย การเลือกเส้นทางและโปรโตคอลการเลือกเส้นทาง ชนิดของโปรโตคอล TCP

บทที่ 3 วิธีการดำเนินการทดลอง

บทที่ 4 ผลการทดลอง

บทที่ 5 สรุปและวิจารณ์

บทที่ 2
ทฤษฎีและหลักการ

2.1 ระบบเครือข่ายไร้สาย (Wireless LAN)

ระบบเครือข่ายไร้สาย (Wireless LANs) เกิดขึ้นครั้งแรก ในปี ค.ศ. 1971 บนเกาะฮาวาย โดยโปรเจ็คของนักศึกษาของมหาวิทยาลัยฮาวาย ที่ชื่อว่า "ALOHNET" ขณะนั้นลักษณะการส่งข้อมูลเป็นแบบ Bi-directionalส่งไป-กลับง่ายๆ ผ่านคลื่นวิทยุ สื่อสารกันระหว่างคอมพิวเตอร์ 7 เครื่อง ซึ่งตั้งอยู่บนเกาะ 4 เกาะโดยรอบ และมีศูนย์กลางการเชื่อมต่ออยู่ที่เกาะๆหนึ่ง ที่ชื่อว่า Oahu

ระบบเครือข่ายไร้สาย (WLAN = Wireless Local Area Network) คือ ระบบการสื่อสารข้อมูลที่มีความคล่องตัวมาก ซึ่งอาจจะนำมาใช้ทดแทนหรือเพิ่มต่อกับระบบเครือข่ายแลนใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และ คลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่อง ผ่านอากาศ โดยปราศจากความต้องการของการเดินสาย นอกจากนั้นระบบเครือข่ายไร้สายก็ยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบ LAN แบบใช้สาย

ที่สำคัญก็คือ การที่มันไม่ต้องใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก ไม่เหมือนระบบ LAN แบบใช้สาย ที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์

ปัจจุบันนี้ โลกของเราเป็นยุคแห่งการติดต่อสื่อสาร เทคโนโลยีต่างๆ เช่นโทรศัพท์มือถือ เป็นสิ่งจำเป็นต่อการดำเนินธุรกิจและการใช้ชีวิตประจำวัน ความต้องการข้อมูลและการบริการต่างๆ มีความจำเป็นสำหรับนักธุรกิจ เทคโนโลยีที่สนองต่อความต้องการเหล่านั้น มีมากมาย เช่น โทรศัพท์มือถือ เครื่องคอมพิวเตอร์โน้ตบุ๊กเครื่อง พีดีเอ ได้ถูกนำมาใช้เป็นอย่างมากและ ผู้ที่น่าจะได้ประโยชน์จากการใช้ระบบเครือข่ายไร้สาย มีมากมายไม่ว่าจะเป็น

-หมอหรือพยาบาลในโรงพยาบาล เพราะสามารถดึงข้อมูลสมารักษาผู้ป่วยได้จาก เครื่องคอมพิวเตอร์โน้ตบุ๊ก ที่เชื่อมต่อกับ ระบบเครือข่ายไร้สายได้ทันที

-นักศึกษาในมหาวิทยาลัยก็สามารถใช้งานโน้ตบุ๊กเพื่อค้นคว้าข้อมูลในห้องสมุดของมหาวิทยาลัยหรือใช้อินเตอร์เน็ท จากสนามหญ้าในมหาลัยได้

-นักธุรกิจที่มีความจำเป็นต้องใช้งานเครื่องคอมพิวเตอร์นอกสถานที่ที่ทำงานปกติ ไม่ว่าจะเป็นการนำเสนองานยังบริษัทลูกค้า หรือการนำเครื่องคอมพิวเตอร์ติดตัวไปงานประชุมสัมมนาต่างๆ บุคคลเหล่านี้มีความจำเป็นที่จะต้องเชื่อมต่อเข้า กับเครือข่ายคอมพิวเตอร์ ไม่ว่าจะเป็นเครือข่ายคอมพิวเตอร์ขององค์กรซึ่งอยู่ห่างออกไปหรือเครือข่ายคอมพิวเตอร์สาธารณะ เช่นเครือข่ายอินเทอร์เน็ต เทคโนโลยีเครือข่ายไร้สายจึงน่าจะอำนวยความสะดวกให้กับบุคคลเหล่านี้ได้ ซึ่งในปัจจุบันได้มีการเปิดให้บริการเชื่อมต่อเครือข่ายอินเตอร์เน็ตแบบไร้สาย ตามสนามบินใหญ่ทั่วโลก และนำมาใช้งานแพร่หลายในห้างสรรพสินค้า และโรงแรมต่างๆแล้ว

ประโยชน์ของระบบเครือข่ายไร้สาย

1. mobility improves productivity & service มีความคล่องตัวสูง ดังนั้นไม่ว่าเราจะเคลื่อนที่ไปที่ไหน หรือเคลื่อนย้ายคอมพิวเตอร์ไปตำแหน่งใด ก็ยังมีการเชื่อมต่อ กับเครือข่ายตลอดเวลา ตราบใดที่ยังอยู่ในระยะการส่งข้อมูล

2. installation speed and simplicity สามารถติดตั้งได้ง่ายและรวดเร็ว เพราะไม่ต้องเสียเวลาติดตั้งสายเคเบิล และไม่รกรุงรัง

3. installation flexibility สามารถขยายระบบเครือข่ายได้ง่าย เพราะเพียงแค่มี พีซีการ์ดมาต่อเข้ากับโน้ตบุ๊ก หรือพีซี ก็เข้าสู่เครือข่ายได้ทันที

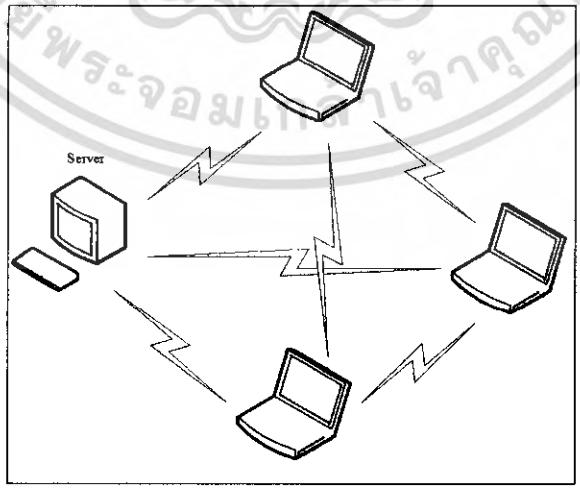
4. reduced cost- of-ownership ลดค่าใช้จ่ายโดยรวม ที่ผู้ลงทุนต้องลงทุน ซึ่งมีราคาสูง เพราะในระยะยาวแล้ว ระบบเครือข่ายไร้สายไม่จำเป็นต้องเสียค่าบำรุงรักษา และการขยายเครือข่ายก็ลงทุนน้อยกว่าเดิมหลายเท่า เนื่องด้วยความง่ายในการติดตั้ง

5.scalability เครือข่ายไร้สายทำให้องค์กรสามารถปรับขนาดและความเหมาะสมได้ง่ายไม่ยุ่งยาก เพราะสามารถโยกย้ายตำแหน่งการใช้งาน โดยเฉพาะระบบที่มีการเชื่อมระหว่างจุดต่อจุด เช่น ระหว่างตึก

2.1.1 รูปแบบการเชื่อมต่อของระบบเครือข่ายไร้สาย

1. Peer-to-peer (ad hoc mode)

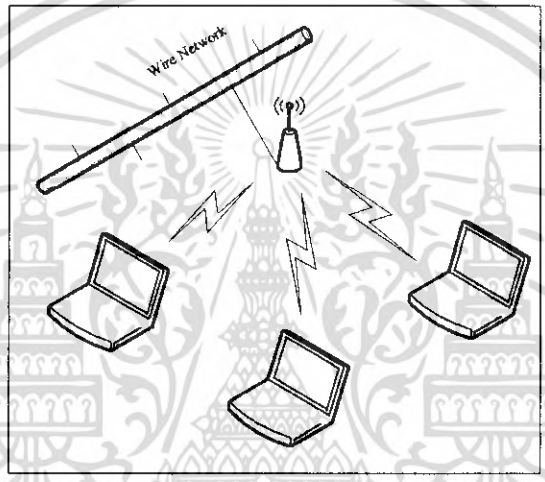
รูปแบบการเชื่อมต่อระบบแลนไร้สายแบบ Peer to Peer เป็นลักษณะ การเชื่อมต่อแบบโครงข่ายโดยตรงระหว่างเครื่องคอมพิวเตอร์ จำนวน 2 เครื่องหรือมากกว่านั้น เป็นการใช้งานร่วมกันของ wireless adapter cards โดยไม่ได้มีการเชื่อมต่อกับเครือข่ายแบบใช้สายเลย โดยที่เครื่องคอมพิวเตอร์แต่ละเครื่องจะมีความเท่าเทียมกัน สามารถทำงานของตนเองได้และขอใช้บริการเครื่องอื่นได้ เหมาะสำหรับการนำมาใช้งานเพื่อจุดประสงค์ในด้านความรวดเร็วหรือติดตั้งได้โดยง่าย เมื่อไม่มีโครงสร้างพื้นฐานที่จะรองรับ ยกตัวอย่างเช่น ในศูนย์ประชุมหรือการประชุมที่จัดขึ้นนอกสถานที่



รูปที่ 2.1 แสดงการทำงานแบบ Ac hoc mode

2. Client/server (Infrastructure modc)

ระบบเครือข่ายไร้สายแบบ Client / server หรือ Infrastructure mode เป็นลักษณะการรับส่งข้อมูลโดยอาศัย Access Point (AP) หรือเรียกว่า "Hot spot" ทำหน้าที่เป็นสะพานเชื่อมต่อระหว่างระบบเครือข่ายแบบใช้สายกับเครื่องคอมพิวเตอร์ลูกข่าย (client) โดยจะกระจายสัญญาณคลื่นวิทยุเพื่อ รับ-ส่งข้อมูลเป็นรัศมีโดยรอบเครื่องคอมพิวเตอร์ที่อยู่ในรัศมีของ AP จะกลายเป็น เครือข่ายกลุ่มเดียวกันทันที โดยเครื่องคอมพิวเตอร์ จะสามารถติดต่อกัน หรือติดต่อกับ Server เพื่อแลกเปลี่ยนและค้นหาข้อมูลได้ โดยต้องติดต่อผ่าน AP เท่านั้น ซึ่ง AP 1 จุด สามารถให้บริการเครื่องลูกข่ายได้ถึง 15-50 อุปกรณ์ ของเครื่องลูกข่าย เหมาะสำหรับการนำไปขยายเครือข่ายหรือใช้ร่วมกับระบบเครือข่ายแบบใช้สายเดิมในออฟฟิต ห้องสมุด หรือในห้องประชุม เพื่อเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น



รูปที่ 2.2 แสดงการทำงานแบบ access point

2.1.2 มาตรฐาน Wireless LANs

มาตรฐานหลักของระบบเครือข่ายไร้สายและอุปกรณ์เครือข่ายไร้สาย คือ มาตรฐาน IEEE 802.11 เป็นมาตรฐานระบบเครือข่ายไร้สายที่ถูกกำหนดขึ้นโดย Institute of Electrical and Electronic Engineers ซึ่งเป็นองค์กรกำหนดมาตรฐานเกี่ยวกับการสื่อสารของอุตสาหกรรมคอมพิวเตอร์ โดยในส่วนมาตรฐาน IEEE 802.XX นั้นจะเป็นเรื่องเกี่ยวกับการสื่อสารผ่านครือข่าย เช่น IEEE 802.3 ก็คือมาตรฐานของเครือข่ายแบบ Ethernet โดยในส่วนย่อย IEEE 802.11 ก็จะเป็นการสื่อสารกับเครือข่าย แต่เป็นแบบไร้สายนั่นเอง

มาตรฐาน IEEE 802.11 นั้นเริ่มประกาศใช้ตั้งแต่ปี ค.ศ. 1997 มาตรฐานที่เกิดขึ้นนี้ยังมีข้อจำกัดในด้านเทคโนโลยี ซึ่งกำหนดระบบการส่งสัญญาณด้วยความเร็ว 2 Mbps และได้มีการพัฒนาเรื่อยมา โดยมีส่วนย่อยอยู่ด้วยกันถึง 9 ส่วน คือ a, b, c, d, e, f, g, h และ I โดยแต่ละชนิดนั้นก็จะมีลักษณะหรือมาตรฐานของรายละเอียดต่างกันไป ซึ่งหลังจาก 9 กลุ่มย่อยนี้ พัฒนามาตรฐาน IEEE 802.11 ในด้านต่างๆ จนเสร็จสิ้นแล้ว จึงได้มีการนำเอามาตรฐานที่พัฒนาเสร็จแล้วมานำเสนอและผลิตออกเป็นผลิตภัณฑ์ออกวางจำหน่าย โดยผลิตภัณฑ์ แรกที่ออกวางจำหน่ายเป็นผลิตภัณฑ์ที่พัฒนาโดยกลุ่มย่อย b จึงทำให้เกิด

มาตรฐาน IEEE 802.11b ในปี ค.ศ.1999 ย่านความถี่ที่เริ่มใช้ เบื้องต้น คือ 2.4 GHz โดยมีความเร็วในการรับ-ส่งข้อมูลสูงสุดอยู่ที่ 11 Mbps ได้วางตลาดก่อนผลิตภัณฑ์กลุ่มอื่น จึงเป็นกลุ่มที่มาตรฐานได้รับการยอมรับและเป็นที่รู้จัก มากที่สุดในช่วงนี้ จากนั้นจึงตามด้วยกลุ่ม a ที่ออกความถี่สูงสุดถึง 5 GHz และมีความเร็วสูงสุดถึง 54 Mbps ใน ทั้งนี้ไม่เกี่ยวว่า a จะเก่ากว่า b และ c จะออกมาใหม่ในอนาคตตามตัวอักษร แต่จะขึ้นอยู่กับว่ามาตรฐานของกลุ่มใดทำเสร็จก่อนก็จะออกเปิดตัวก่อนโดยไม่เรียงลำดับตามตัวอักษร

1. มาตรฐาน IEEE 802.11

มาตรฐาน IEEE 802.11 เหมือนกับมาตรฐาน IEEE 802.3 Ethernet ซึ่งใช้กับเครือข่าย LAN แบบใช้สาย และ IEEE 802.5 สำหรับเครือข่าย Token Ring ตรงที่ มาตรฐาน IEEE 802.11 จะมุ่งความสนใจไปที่ระดับล่างสุดสองระดับของ ISO model (คือ physical layer และ data link layer) ซึ่งจะทำให้ application, network OS, protocol, รวมทั้ง TCP/IP ใดๆก็ตามสามารถใช้งานบน 802.11 compliant WLANs ได้ง่ายๆ เช่นเดียวกับใช้งานบน Ethernet โดยทั่วไป

มาตรฐาน 802.11 นี้ใช้การส่งสัญญาณแบบคลื่นวิทยุที่ความถี่ 2.4 GHz ซึ่งเป็นความถี่ ISM (Industrial Scientific and Medical) band สามารถส่งข้อมูลได้ด้วยอัตราความเร็ว ค่อนข้างต่ำ คือ 1 และ 2 Mbps เท่านั้นโดยใช้เทคนิคการส่งสัญญาณหลักอยู่ 2 รูปแบบ คือ DSSS (Direct Sequent Spread Spectrum) และ FHSS(Frequency Hopping Spread Spectrum) ซึ่งถูกคิดค้นมาจากหน่วยงานทหาร การส่งสัญญาณทั้ง 2 รูปแบบจะใช้ความกว้างของช่องสัญญาณ (bandwidth) ที่มากกว่าการส่งสัญญาณแบบ narrow band แต่ทำให้สัญญาณมีความแรงมากกว่าซึ่งง่ายต่อการตรวจจับมากกว่า แบบ narrow band หน่วยงานทหารใช้วิธีการเหล่านี้ในการป้องกันการใช้งานจากอุปกรณ์อื่นๆ ที่จะมาทำให้ระบบเกิดปัญหา โดยการส่งสัญญาณแบบ FHSS สัญญาณจะกระโดดจากความถี่หนึ่งไปยังอีกความถี่หนึ่งในอัตราที่ได้กำหนดไว้แล้ว ซึ่งจะรู้กันเฉพาะตัวรับกับตัวส่งเท่านั้น ส่วนการส่งสัญญาณแบบ DSSS จะมีการส่ง chipping code ไปกับสัญญาณแต่ละครั้งด้วย ซึ่งจะมีเฉพาะตัวรับกับตัวส่งเท่านั้นที่จะรู้ลำดับของ chip สำหรับการใช้งานระบบเครือข่ายแบบไร้สายทุกวันนี้ DSSS มีคุณสมบัติที่โดดเด่นและให้ throughput ที่มากกว่า เมื่อเร็วๆนี้เองที่ได้มีการพัฒนาจนได้อัตราการส่งข้อมูล 11 Mbps ผ่านการส่งแบบ DSSS และเป็นมาตรฐานที่โดดเด่นของ WLAN ผลิตภัณฑ์ซึ่งรองรับมาตรฐาน 802.11b (อัตราส่งถ่ายข้อมูลสูง 11 Mbps) นี้สามารถทำงานร่วมกับผลิตภัณฑ์ซึ่งทำงานกับมาตรฐาน DSSS แบบเก่า 802.11 (อัตราส่งถ่ายข้อมูล 1 และ 2 Mbps) ได้ แต่ ระบบ FHHS จะถูกใช้กับอุปกรณ์ที่มีกำลังส่งต่ำเป็น application ที่ใช้งานในย่านต่ำๆ เช่น โทรศัพท์ไร้สายความถี่ 2.4 GHz แต่จะใช้งานร่วมกับผลิตภัณฑ์ DSSS ไม่ได้

2. มาตรฐาน IEEE 802.11b

มาตรฐาน IEEE 802.11b ซึ่งเป็นมาตรฐานระบบเครือข่ายไร้สายที่ได้รับการยอมรับมากที่สุดในโลกเพราะมีการเปิดตัวก่อนมาตรฐานอื่นและมีผลิตภัณฑ์ออกวางจำหน่ายแล้วมากและแพร่หลายที่สุด

มาตรฐาน IEEE 802.11b นั้นล่าสุดได้รับการตั้งชื่อใหม่ว่า Wi-Fi โดยได้รับการรับรองมาตรฐานและกำหนดรายละเอียดโดยกลุ่ม WECA หรือ Wireless Ethernet Compatibility Alliance ที่ประกอบด้วยสมาชิกจากผู้ผลิตในอุตสาหกรรมคอมพิวเตอร์ชื่อดังอย่าง 3com, Cisco Systems, Intersil, Agere Systems,

Nokia และ Symbol Technologies ซึ่งปัจจุบันก็ยังมีสมาชิกจากบริษัทต่างๆ อีกกว่า 110 บริษัทเข้าร่วมอยู่ในมาตรฐานนี้

สำหรับรายละเอียดด้านคุณสมบัติ ของ IEEE 802.11b จะสามารถรับ-ส่งข้อมูลได้ด้วยความเร็วสูงสุดที่11 Mbps โดยใช้ความถี่คลื่นวิทยุที่ 2.4 GHz ใช้เทคนิคการส่งสัญญาณแบบ DSSS โดยย่านความถี่ที่ใช้เป็น ISM (Industrial Scientific and Medical) band จากระดับความเร็วที่ค่อนข้างต่ำ คือทำได้เพียง 11 Mbps เท่านั้นเมื่อเทียบกับ ระบบ LAN แบบมีสาย ที่มาตรฐานปัจจุบัน อยู่ที่ระดับ 100 Mbps และล่าสุดมาตรฐานความเร็ว 1Gbps กำลังเป็นที่ยอมรับและนิยมใช้งานมากขึ้นเรื่อย ๆ ก็จะเห็นว่า IEEE 802.11b นั้นค่อนข้างช้ากว่ามาก ไม่เพียงเท่านั้น คลื่นความถี่วิทยุที่ 2.4 GHz ที่ IEEE 802.11b ใช้อยู่นั้นยังมีอุปกรณ์อื่นๆ ร่วมใช้งานอยู่ด้วยหลายชนิด เช่น เตาไมโครเวฟ หรือ โทรศัพท์มือถือ ซึ่งหากมีอุปกรณ์เหล่านี้ทำงานอยู่ใกล้ๆ กับเครือข่าย IEEE802.11b ก็จะทำให้ความเร็วในการรับส่งข้อมูลช้าลง แต่จุดเด่นก็คือการใช้ความถี่คลื่นวิทยุที่ค่อนข้างต่ำ เพียง2.4 GHz นั้นทำให้ IEEE 802.11b มีระยะทางในการติดต่อระหว่างอุปกรณ์ค่อนข้างไกล ทำให้ชุดเครือข่ายไร้สายแบบ IEEE 802.11b ไม่จำเป็นต้องมีจุด รับส่งสัญญาณ หรือที่เรียกกันว่า Access Point หรือ Hot Spot มากนัก ซึ่งช่วยประหยัดค่าใช้จ่ายได้ดี

3. มาตรฐาน IEEE 802.11a

มาตรฐาน IEEE 802.11a นั้นเกิดขึ้นหลังการวางตลาดของมาตรฐาน IEEE 802.11b โดยผลิตภัณฑ์ IEEE 802.11a มีจุดเด่นที่เหนือกว่า IEEE 802.11b ตรงที่ความเร็วในการรับส่งข้อมูลนั้นจะเร็วกว่า คือ ทำได้สูงสุดถึง 54 Mbps และเร็วกว่า IEEE 802.11b ในทุกระยะทาง (ความเร็วของเครือข่ายไร้สายทุกมาตรฐานจะลดลงเมื่อระยะทางมากขึ้น) โดยมีความถี่คลื่นวิทยุอยู่ที่ 5 GHz ซึ่งเป็นย่านความถี่วิทยุ ของ Unlicensed National Information Infrastructure (U-NII) band มีความกว้างของความถี่ทั้งหมด 300 MHz โดยแบ่งเป็น 3 ระดับระดับละ 100 MHz คือ ต่ำ ปานกลาง และสูง ซึ่งแต่ล่ะระดับมีระดับมีการสามารถใช้งานและกำลังส่งแตกต่างกัน

- ย่านความถี่ระดับต่ำ (low band) ย่านความถี่ที่ทำงานจาก 5.15 ถึง 5.25 GHz กำลังส่งสูงสุดเท่ากับ50 mW

- ย่านความถี่ระดับปานกลาง (middle band) ย่านความถี่ที่ทำงานจาก 5.25 ถึง 5.35 GHz ด้วยกำลังส่งสูงสุด เท่ากับ 250 mW

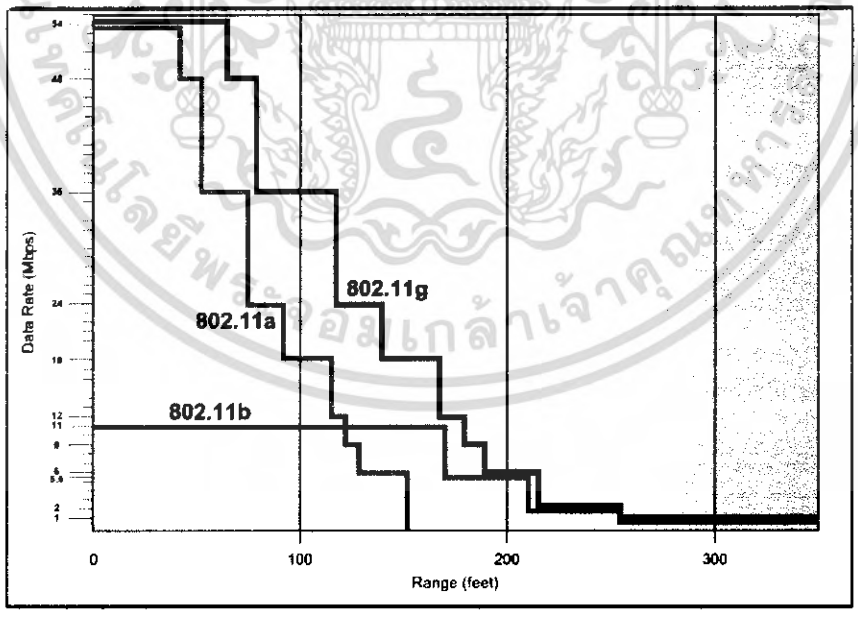
- ย่านความถี่ระดับสูง (high band) ย่านความถี่ที่ทำงานจาก 5.725 ถึง 5.825 GHz ด้วยกำลังส่งสูงสุดเท่ากับ 1000 mW

โดยกำลังส่งที่สูงของเครื่องรับ-ส่งสัญญาณของระบบเครือข่ายไร้สายและช่วงความถี่ 5.8 GHz จะทำให้สามารถส่งสัญญาณติดต่อกัน ระหว่างอาคารหนึ่ง กับอีกอาคารหนึ่งได้ ส่วนการใช้งานภายในอาคารจะใช้งานในย่านความถี่ระดับปานกลางและต่ำ ซึ่งในอเมริกาสามารถใช้งานได้ทั้ง 3 ย่านความถี่ แต่ปัญหาเรื่องของกฎหมายเกี่ยวกับคลื่นความถี่ระดับ 5 GHz ที่ในแถบยุโรปและประเทศญี่ปุ่นมีข้อกำหนดค่อนข้างเคร่งครัด คือ ในยุโรปกำลังทำข้อตกลงร่วมกันระหว่าง IEEE และ European Telecommunications Standards Institute (ETSI) ส่วนในประเทศญี่ปุ่นอนุญาตให้ใช้ได้เฉพาะ ย่านความถี่ต่ำเท่านั้น

ดังนั้นการใช้งานในย่านความถี่ปานกลางและต่ำ จึงมีความกว้างของสัญญาณรวมกันเท่ากับ 200 MHz สามารถส่งข้อมูลได้ด้วยอัตราเร็วสูงสุดถึง 54 Mbps ได้สำเร็จ โดยใช้หลักการ ส่งสัญญาณความถี่ย่อยโดยอัตราเร็วต่ำๆ พร้อมๆกัน เมื่อนำทั้งหมดมารวมกัน ก็จะสามารถสร้างช่องสัญญาณที่มีอัตราเร็วสูงขึ้นได้ ตามที่ได้รับอนุญาตให้ใช้ช่วงความถี่ดังกล่าว สามารถแบ่งการใช้งานได้ ถึง 8 ช่องสัญญาณโดยไม่ทับซ้อนกัน แต่ละช่องสัญญาณมีความกว้าง เท่ากับ 20 MHz ใช้การมอดูเลชั่นแบบ OFDM (Orthogonal Frequency division Multiplex) ในการส่งสัญญาณ ซึ่งเป็นเทคนิคการส่งสัญญาณแบบแยกส่งเป็นความถี่ย่อยๆ (Narrow-bandsubcarriers) และมีความเป็นอิสระต่อกัน แต่ละความถี่ย่อยจะมีความกว้างเท่ากับ 300 KHz จำนวน 52 ช่องสัญญาณความถี่ย่อย สัญญาณความถี่ย่อยจะทำการรับและส่งข้อมูลโดยส่งไปแบบขนาน ด้านรับสัญญาณจะได้รับข้อมูลทั้งหมดพร้อมกัน ซึ่งนั้นก็หมายความว่าข้อมูลที่ส่งจะมีขนาดใหญ่ และต้องการความต่อเนื่องในการส่งสัญญาณ เพราะฉะนั้นเพื่อป้องกัน การสูญหายของข้อมูล (data loss feature) จึงเพิ่ม Forward Error Correction (FEC) เข้าไปใน 802.11a ด้วย ซึ่งจะมีเฉพาะใน 802.11a เท่านั้น (ไม่พบใน 802.11b)

มาตรฐาน 802.11a รองรับอัตราความเร็วของการส่งข้อมูล เท่ากับ 6, 9,12, 18, 24, 36, 48 และ 54 Mbps อัตราความเร็วจะลดลงเองอย่างอัตโนมัติขึ้นอยู่กับระยะทางระหว่าง Access point กับ เครื่องคอมพิวเตอร์ลูกข่าย โดยที่ความเร็วสูงสุดที่ 54 Mbps นั้นใช้การมอดูเลชั่นสัญญาณความถี่ย่อย แบบ 64-level Quadrature Amplitude Modulation (64 QAM)

คล้ายกันกับ 802.11b ที่ เครื่องลูกข่ายมาตรฐาน 802.11a จะมีอัตราเร็วลดลงเหมือนระยะทางจาก Access Point มากขึ้น แต่เมื่อนำมาเปรียบเทียบกันแล้ว 802.11a ยังมีความเร็วที่เหนือกว่าในทุกระยะทาง



รูปที่ 2.3 แสดงการเปรียบเทียบอัตราเร็วกับระยะทางระหว่าง มาตรฐาน 802.11a 802.11bและ802.11g

4. มาตรฐาน IEEE 802.11g

เป็นมาตรฐานที่มีจุดเด่นก็คือการใช้คลื่นความถี่วิทยุ 2.4 GHz ซึ่งเป็นคลื่นสาธารณะที่ได้รับอนุญาตให้ใช้งานได้โดยไม่ผิดกฎหมาย เหมือนมาตรฐาน IEEE802.11b แต่ใช้เทคโนโลยีแบบ OFDM ในการส่งสัญญาณ ทำให้มีความเร็วสูงสุดมากกว่า 20 Mbps เหมือนมาตรฐาน IEEE 802.11a จุดเด่นที่สำคัญของ 802.11g ก็คือสามารถใช้งานร่วมกับ 802.11b ที่มีอยู่แล้วได้

2.1.3 โครงสร้างการทำงานของระบบเครือข่ายไร้สาย

ในระบบเครือข่ายไร้สาย IEEE 802.11 นั้นจะแบ่งระดับชั้นของเทคโนโลยีออกเป็น 4 ระดับ นั่นคือPHY (Physical Layer หรือ ชั้นกายภาพ) MAC (Media Access Controller หรือตัวควบคุมการเข้าถึงสื่อ) OS (ระบบปฏิบัติการ) และ Application (แอพพลิเคชั่น) โดย PHY หรือขั้นกายภาพนั้นก็คือส่วนของฮาร์ดแวร์ที่แบ่งมาตรฐานออกเป็น a, b และ g โดยหากเลือกต่างชนิดกันก็ไม่สามารถสื่อสารกันได้รู้เรื่อง เพราะเป็นความถี่ที่ต่างกันจะติดต่อรับส่งข้อมูลกันไม่ได้ โดยปัจจุบันในส่วนของ PHY นี้มีอยู่ทั้งสิ้น 4 มาตรฐาน คือ a, b, g และ IR(อินฟราเรด)

ส่วนต่อมาคือ MAC นั้น เป็นส่วนของการทำงานเกี่ยวกับระบบรักษาความปลอดภัยของเครือข่าย การจัดการโครงสร้างหรือรูปแบบของข้อมูล การแปลงข้อมูล ซึ่งมาตรฐาน IEEE 802.11 นั้นใช้มาตรฐาน MAC เดียวกันทั้งหมด คือ ได้กำหนดทางเลือกของการเข้ารหัสไว้ก่อนทำการส่งข้อมูล โดยใช้อัลกอริธึมการเข้ารหัสแบบ 40 บิตซึ่งรู้จักกันในชื่อ RC4 นอกจากนั้นผู้ผลิตบางรายก็ยังเสนอให้มีการตรวจสอบก่อนใช้งานโครงข่ายด้วยวิธีการที่เรียกว่า Wired Equivalent Privacy (WEP) shared-key อันเดียวกันจะใช้ในการตรวจสอบก่อนที่จะทำการเข้ารหัสหรือถอดรหัสข้อมูล ซึ่งจะมีเพียงผู้ใช้งานที่ถูกต้องเท่านั้นจึงจะมี shared-key ที่ถูกต้องในการถอดรหัสข้อมูลออกมาได้ เนื่องด้วยเทคโนโลยีไร้สายถูกคิดค้นขึ้นมาจากหน่วยงานทางทหาร ฉะนั้นเรื่องความปลอดภัยจึงเป็นหัวใจสำคัญอย่างยิ่ง นอกจากเรื่องความน่าเชื่อถือกับเรื่องความปลอดภัยแล้ว มาตรฐาน 802.11 ในส่วน MAC นี้ ยังมีโหมดสนับสนุนการจัดการพลังงานอีก 2 รูปแบบ คือ Continuous Aware Mode และ Power Saving Polling Mode โดยโหมดแรกสัญญาณวิทยุจะส่งอยู่ตลอดและทำให้สูญเสียพลังงาน ในขณะที่โหมดต่อมาสัญญาณวิทยุจะอยู่ในภาวะนอนหลับหรือ sleep เพื่อที่จะถนอมพลังงาน

ส่วนของ OS และ Application นั้นก็คือระบบปฏิบัติการภายในเครื่องและแอพพลิเคชั่นควบคุมการสื่อสาร ซึ่งตรงนี้ก็ใช้งานเหมือนอย่างที่ใช้งานกันอยู่กับเครือข่ายแบบมีสายในปัจจุบัน

2.1.4 เทคโนโลยีเครือข่ายไร้สาย

โดยทั่วไปแล้วระบบเครือข่ายไร้สายจะใช้เทคโนโลยีในการส่งสัญญาณอยู่ 2 ประเภท คือ ประเภทที่ใช้สัญญาณคลื่นความถี่วิทยุซึ่งแบ่งเป็น 2 แบบ คือ Narrow band และ Spread spectrum โดยมีรายละเอียด ดังต่อไปนี้

1. Narrow band Technology

ระบบวิทยุแบบความถี่แคบ เป็นการรับ – ส่ง สัญญาณคลื่นวิทยุบนความถี่เฉพาะ โดยคลื่นความถี่ดังกล่าว เป็นที่รู้จักในชื่อของแถบความถี่ ISM (Industrial Scientific and Medical) ที่มีความถี่แบ่งเป็น 3 ช่วง ได้แก่ 902 MHz ถึง 928 MHz 2.14 MHz ถึง 2.484 MHz และ 5.725 MHz ถึง 5.850 MHz

สัญญาณจะมีกำลังต่ำ (โดยทั่วไปประมาณ 1 มิลลิวัตต์) และใช้ในการรับ-ส่ง ข้อมูลระหว่างต้นทางกับปลายทางเพียง 1 คู่เท่านั้น และไม่สามารถส่งสัญญาณข้ามโหนดไปมาได้ การส่งข้อมูลแบบนี้เปรียบได้กับคู่สายโทรศัพท์ที่สามารถคุยได้เฉพาะต้นทางกับปลายทางแต่ ไม่สามารถคุยพร้อมกันได้หลายๆ คน

ข้อจำกัดของการใช้สัญญาณแบบนี้ คือจะต้องของอนุญาตจาก FCC (Federal Communication Committee) ซึ่งเป็นหน่วยงานที่กำหนดความถี่ในการใช้สัญญาณคลื่นวิทยุแบบ Narrow band นี้

2. Spread spectrum technology

ระบบเครือข่ายไร้สายส่วนใหญ่นิยมใช้เทคนิค Spread spectrum technology ซึ่งใช้ความถี่ที่กว้างกว่า Narrow band Technology ซึ่ง Spread Spectrum ก็คือ วิธีการเปลี่ยนแปลงสัญญาณข้อมูลเพื่อให้ครอบคลุมพื้นที่ความถี่วิทยุมากเกินความจำเป็น แรกทีเดียวเทคนิคนี้ได้รับการพัฒนาขึ้นมาเพื่อใช้ในกิจการทางทหารซึ่งต้องการความเชื่อถือได้ในระดับสูงมากในระหว่างการรบ ข้าศึกอาจใช้อุปกรณ์อิเล็กทรอนิกส์ดักฟังสัญญาณเพื่อขโมยความลับหรือรบกวนการทำงาน แต่ในระบบนี้การส่งสัญญาณถูกส่งออกไปหลายความถี่พร้อมกันจึงทำให้การดักฟังเป็นไปได้ยากขึ้น รวมทั้งการรบกวนการสื่อสารก็ยากมากขึ้นด้วยเพราะจะต้องค้นหาคลื่นความถี่ทั้งหมดให้ได้ โดยการส่งสัญญาณจะใช้แถบความถี่ ISM ที่ช่วงความถี่ ระหว่าง 902-928 MHz และ 2.4-2.484 GHz เทคนิค Spread Spectrum สามารถแบ่งได้ เป็น 2 แบบ คือ Direct Sequence และ Frequency – Hopping

3. Direct Sequence Spread Speetrum (DSSS)

Direct Sequence Spread Spectrum เป็นเทคนิคที่ยังใช้คลื่นพาหะที่ต้องระบุความถี่ที่ใช้ สามารถส่งข้อมูลได้มากกว่า แบบ narrow band ข้อมูลจะถูกกระจายให้ช่วงความถี่กว้างขึ้น (RF bandwidth) ในรูปแบบของรหัสเฉพาะ รูปแบบของรหัสเฉพาะที่เป็นที่รู้จักกันดีคือ Pseudo-noise Sequence หรือ PN sequence

รูปแบบนี้จะใช้การเข้ารหัสในวิธีพิเศษ โดยการแปลงเลขฐานสองแต่ละบิตในข้อมูลดั้งเดิมที่จะส่งไปให้อยู่ในรูปแบบเลขฐานสองที่มีความยาวเพิ่มมากขึ้น ตัวอย่างเช่น ข้อมูลเลขฐานสอง 1 อาจจะถูกแปลงเป็น0010010101 และข้อมูล 0 จะถูกแปลงเป็น Inverse ของ 1 คือ 1101101010 แล้วข้อมูลที่แปลงแล้วเหล่านี้จะถูกส่งไปพร้อมๆกัน ในลักษณะขนาน ซึ่งหากผู้รับสามารถจดจำรูปแบบการแปลงข้อมูลได้ก็จะถูกส่งไป โดยที่สัญญาณรบกวนไม่สามารถทำให้ข้อมูลเสียหายไปได้ หรือหากรูปแบบที่ส่งไปเกิดผิดพลาดไปไม่ว่าจะด้วยสาเหตุใดก็ตาม ทางฝ่ายรับก็สามารถที่จะใช้เทคนิคในทางสถิติเพื่อกู้ข้อมูลที่ผิดพลาดไปให้กลับคืนมาได้ วิธีนี้จะใช้ในมาตรฐาน IEEE802.11 และIEEE 802.11b ผู้ผลิตระบบเครือข่ายไร้สายส่วนใหญ่จะเลือกใช้วิธีการนี้เพราะว่าเป็นวิธีที่เหมาะมากกว่าวิธีอื่นในสภาพแวดล้อมที่มีการแทรกสอดรบกวนจากคลื่นวิทยุอื่นๆ อย่างรุนแรง นอกจากนี้ยังเปิดโอกาสให้ผู้ใช้ตัดสินใจได้ว่าจะทำการจัดสรรแถบความถี่ในการส่งข้อมูลอย่างไรบ้าง เช่น อาจจัดแบ่งแถบความถี่เป็นช่วงย่อยหลายช่วงเพื่อใช้ส่งข่าวสารหลายชิ้นไปพร้อมกัน

4. Frequency – Hopping Spread Spectrum (FHSS)

การส่งสัญญาณรูปแบบนี้จะใช้ความถี่แคบพาหะเพียงความถี่เดียว (narrow band) และจะเปลี่ยนแปลงความถี่(กระโดด)ไปมาอย่างต่อเนื่อง ในลักษณะหรือรูปแบบที่เป็นที่เข้าใจตรงกันระหว่างเครื่องส่งกับเครื่องรับสามารถทำงานประสานกันได้แล้ว

วิธีการส่งแบบนี้ป้องกันสัญญาณรบกวนที่เกิดจากความถี่ข้างเคียงได้เป็นอย่างดี เพราะว่าความถี่จะมีการเปลี่ยนแปลงตลอดเวลา โดยการส่งและรับแต่ละครั้งที่ส่วนหัวของ packet ข้อมูลจะบอก รับก็สามารถที่จะปรับเปลี่ยนไปได้ตลอดเวลาอันจะทำให้เกิดความปลอดภัยของข้อมูลสูงมากขึ้น ผู้ผลิตระบบเครือข่ายเฉพาะที่ไร้สายแบบ Frequency Hopping ให้ความเห็นว่าการส่งข้อมูลวิธีนี้สามารถส่งข้อมูลไปพร้อมๆกันหลายช่องสัญญาณได้ด้วยการกำหนดให้มีรูปแบบของการเปลี่ยนแปลงหลายๆ รูปแบบทำงานไปพร้อมกัน ซึ่งจะทำสามารถใช้ประโยชน์แถบความถี่ได้ดีกว่าและทำให้เครือข่ายมีประสิทธิภาพสูงกว่า

ในการตัดสินใจเลือกใช้วิธีใดวิธีหนึ่งนั้น การนำไปใช้งานจะเป็นตัวกำหนดว่า ถ้าคำนึงถึงปัญหาทางด้านประสิทธิภาพและคลื่นรบกวนก็ควรใช้วิธี DSSS ถ้าต้องการใช้อะแดปเตอร์ไร้สายขนาดเล็กและราคาไม่แพงสำหรับเครื่องโน้ตบุ๊ก หรือ เครื่อง PDA ก็ควรเลือกแบบ FHSS

5. Orthogonal frequency division multiplex (OFDM)

เทคนิคนี้ถูกนำมาใช้ในเพื่อเพิ่มความเร็วในการส่งข้อมูลในมาตรฐาน ใหม่ๆ ของระบบเครือข่ายไร้สายคือ IEEE 802.11a และ 802.11g การส่งสัญญาณคลื่นวิทยุแบบนี้ เป็นการมัลติเพล็กสัญญาณ โดยช่องสัญญาณความถี่จะถูกแบ่งออกเป็นความถี่พาหะย่อย (subcarrier) หลายๆความถี่ โดยแต่ละความถี่พาหะย่อยจะตั้งฉากซึ่งกันและกัน ทำให้มันเป็นอิสระต่อกัน ความถี่ที่คลื่นพาหะที่ตั้งฉากกันนั้นทำให้ไม่มีปัญหาการซ้อนทับกันของสัญญาณที่อยู่ติดกัน

OFDM เป็นเทคนิคการมัลติเพล็กโดยการแบ่งความถี่ เมื่อช่องความถี่ถูกแบ่งออกเป็นขนาดเล็กๆ N ช่องแต่ละช่องมีขนาดเท่ากับขนาดของสัญลักษณ์ (bit rate) ดิจิตอล ทางด้านส่งจะมีสัญญาณดิจิตอลความเร็วสูงที่ถูกแบ่งออกเป็นกลุ่มข้อมูลย่อยๆ ที่มีความถี่ต่ำกว่า จะถูกมอดูเลตกับสัญญาณพาหะย่อย 1 สัญญาณ และนำสัญญาณทั้งหมดส่งขนานกันออกไป รูปแบบในการมอดูเลตสัญญาณพาหะย่อยที่นิยมทั่วไปได้แก่ 16 QAM หรือ64 QAM เป็นต้น ใน OFDM กลุ่มของข้อมูลจะถูกแปลงให้อยู่ในรูปขนานกันโดยการมอดูเลตกับสัญญาณพาหะย่อย ดังนั้น จะกลายมาเป็นสัญญาณบนแกนความถี่ ซึ่งการแปลงสัญญาณกลับให้อยู่บนแกนเวลาอีกครั้งโดยการแปลงกลับฟาสท์ฟูเรียร์ (IFFT) จากนั้นจะใช้สัญลักษณ์บนแกนเวลาจะถูกมัลติเพล็กเข้าด้วยกันให้เป็นอนุกรมของสัญญาณ แล้วจึงส่งสัญญาณออกไปทางเสาอากาศ

หลังจากการมอดูเลตแบบ OFDM จะมีการสอดแทรกช่วงแถบป้องกันแคบๆ เพื่อลดสัญญาณรบกวนระหว่างสัญลักษณ์ (Inter symbol Interference: ISI) ที่เกิดจากสัญญาณหลายเส้นทาง (multi-path) เราเรียกแถบป้องกันแคบๆนี้ว่า การเสริมไซคริก (cyclic prefix) ส่วนในเครื่องรับจะดำเนินกระบวนการตรงข้ามกับเครื่องส่งในเครื่องรับจะใช้การแปลงฟาสท์ฟูเรียร์แปลงสัญญาณที่อยู่บนแกนเวลาไปเป็นแถบความถี่สมมูลย์

ข้อดีของ OFDM คือสามารถใช้งานแถบความถี่ในระบบที่เคยใช้สัญญาณพาหะเดี่ยวได้อย่างเต็มประสิทธิภาพ (spectral efficiency) สามารถป้องกันผลกระทบจากการเคลื่อนที่ของสัญญาณหลายเส้นทาง

(immunity to multi-path) และมีความไวต่ำต่อการเลื่อนหายไปของความถี่ที่เลือก (less sensitivity to frequency selective fading)

2.2 โปรโตคอล

การเชื่อมโยงเครือข่ายที่มีฮาร์ดแวร์ต่างกันจำเป็นต้องกำหนดข้อตกลงร่วมเรียกว่าโปรโตคอล (Protocol) ซึ่งการกำหนดโปรโตคอลมีไว้เพื่อให้คอมพิวเตอร์สื่อสารกันตามข้อกำหนด ทีซีพี/ไอพี (TCP/IP) จัดเป็นโปรโตคอลหนึ่งที่ออกแบบมาเพื่อแก้ปัญหาการเชื่อมโยงดังกล่าว

โปรโตคอลในความหมายของระบบเครือข่ายคือข้อกำหนดการสื่อสารคอมพิวเตอร์หรืออุปกรณ์เครือข่ายจะมีซอฟต์แวร์ที่ปฏิบัติงานตามโปรโตคอลที่กำหนดพร้อมทั้งมีกรรมวิธีแก้ไขปัญหาที่เกิดขึ้น เช่น หากข้อมูลที่ขนถ่ายมีข้อผิดพลาด คอมพิวเตอร์จะดำเนินการตามแบบแผนในโปรโตคอลเช่นส่งข้อมูลซ้ำใหม่

ในระบบเครือข่ายขนาดใหญ่ อาจมีเส้นทางเชื่อมโยงระหว่างกันได้เป็นจำนวนมากข้อมูลที่ส่งออกไป อาจไม่ได้ใช้เส้นทางเดียวกันตลอด ข้อมูลที่ส่งออกไปก่อนอาจไปถึงปลายทางช้ากว่ากรณีนี้เครื่องปลายทางจำเป็นต้องจัดลำดับข้อมูลใหม่ กรณีที่คอมพิวเตอร์ต้นทางสามารถส่งข้อมูลได้เร็วเกินกว่าปลายทางจะรับได้ทัน โปรโตคอลจะกำหนดกรรมวิธีควบคุมการลำเลียงข้อมูลระหว่างต้นทางและปลายทางให้สัมพันธ์กันข้อกำหนดตามโปรโตคอลที่กล่าวถึงนี้จะอธิบายโดยละเอียดในแต่ละหัวข้อต่อไป

2.2.1 โพรโตคอลทีซีพี/ไอพี

1. ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

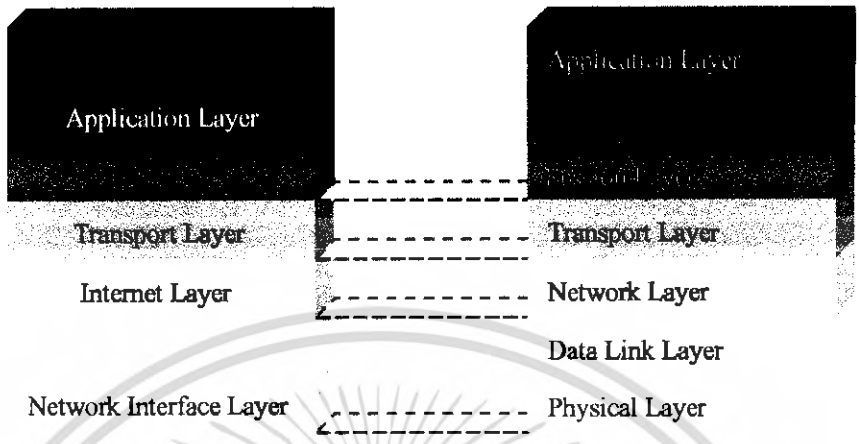
ทีซีพี/ไอพี (TCP/IP: Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลมาตรฐานที่ใช้กันอยู่ในระบบปฏิบัติการแบบยูนิกซ์ เริ่มพัฒนาโดยกระทรวงกลาโหมของสหรัฐฯ ในปี ค.ศ. 1969 เพื่อเชื่อมโยงเครื่องคอมพิวเตอร์ทางทหารของแต่ละหน่วยที่อยู่ห่างไกลกัน โดยมีจุดประสงค์คือสร้างระบบเครือข่ายให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้แม้ว่าสายส่งข้อมูลบางส่วนจะถูกทำลายเสียหายไปก็ตามเพื่อใช้งานในยามเกิดสงคราม โดยเครือข่ายที่จัดตั้งในระยะแรกชื่อว่า Advanced Research Projects Agency Network หรือ อาร์พาเน็ต (ARPANET)

ต่อมาได้พัฒนาเป็นเครือข่ายอินเทอร์เน็ต (INTERNET) โพรโตคอลนี้เหมาะสำหรับเชื่อมต่อคอมพิวเตอร์ทั้งใกล้และไกลเข้าด้วยกัน เนื่องจากมีมาตรฐานรองรับทำให้ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์สามารถสร้างอุปกรณ์และโปรแกรมที่จะรองรับการทำงานของโพรโตคอลนี้ ทำให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ไม่ว่าจะเป็นเครื่องขนาดเล็กหรือขนาดใหญ่หรือใช้ระบบปฏิบัติการอะไรก็ได้

2. การเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลที่ใช้ในการสื่อสารในระบบอินเทอร์เน็ต และอินทราเน็ต มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ของฝ่ายรับและฝ่ายส่งให้ได้รับข้อมูลที่ถูกต้องครบถ้วน หากข้อมูลที่ส่งมาเกิดการสูญหายระหว่างทางจะมีการแจ้งให้ต้นทางส่งข้อมูลมาใหม่ การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับ

โมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2.4



รูปที่ 2.4 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังนี้

- ชั้นแอพพลิเคชัน (Application Layer)

รองรับการทำงานของแอพพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซสหรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่างๆ มีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอพพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง

- ชั้นทรานสปอร์ต (Transport Layer)

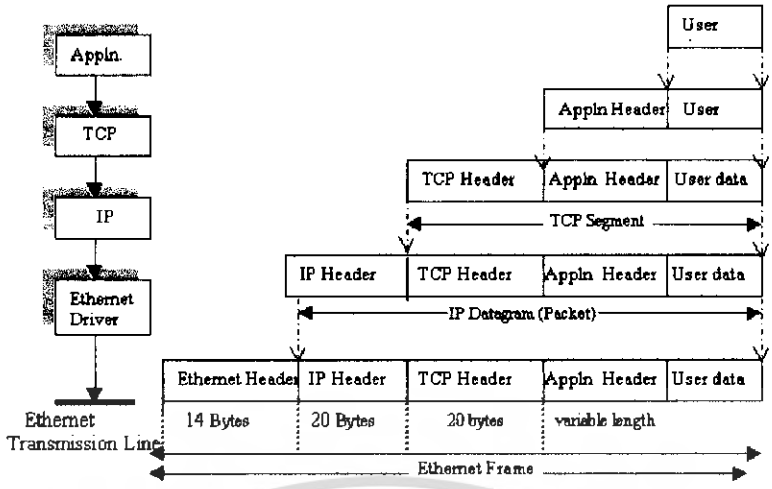
สร้างการเชื่อมต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือ ซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol)

- ชั้นอินเทอร์เน็ต (Internet Layer)

ส่งผ่านข้อมูลระหว่างเครือข่ายโดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ตคือ ไอพี (Internet Protocol: IP) นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP)

- ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer)

แปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย

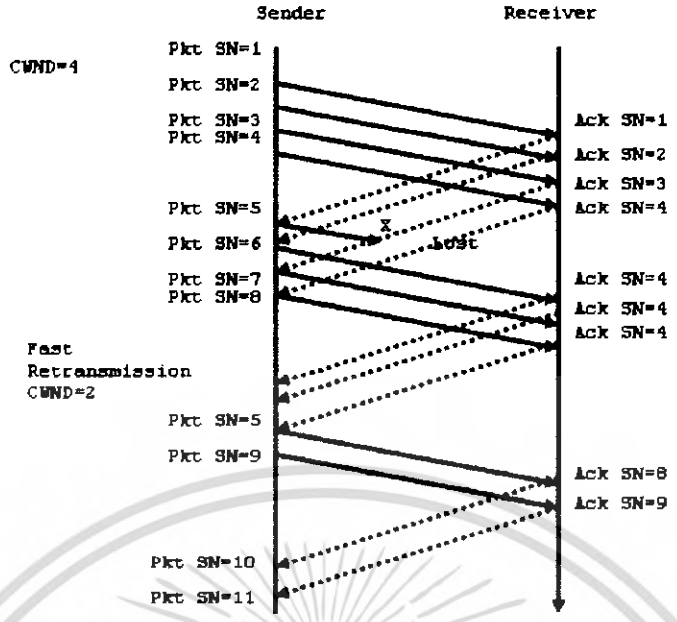


รูปที่ 2.5 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

3. การทำงานและโครงสร้างของระบบโพรโตคอลทีซีพี/ไอพี

ในชุดโพรโตคอลทีซีพี/ไอพีนี้มีโพรโตคอลหลักที่ขอกล่าวถึง 3 โพรโตคอล ได้แก่โพรโตคอลทีซีพี โพรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโตคอลไอพีซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

โพรโตคอลทีซีพี (TCP: Transmission Control Protocol) การทำงานของโพรโตคอลทีซีพีที่สำคัญอย่างหนึ่งของโพรโตคอลทีซีพี คือ การทำ "3-Way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างผู้ใช้และผู้ให้บริการในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมาจึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2.6 แสดงเหตุการณ์การรับส่งข้อมูลขณะที่ขนาดของหน้าต่างควบคุมความคับดั่งเท่ากับ 4 และจะเห็นได้ว่าเมื่อเกิดการสูญหายของข้อมูลในที่นี้คือแพคเกจที่ 5 หายไปผู้รับก็จะได้แค่รับแพคเกจที่ 6, 7 และ 8 และผู้รับจะทำการส่ง ACK ที่ 4 ตอบกลับไปเพราะว่ามีแพคเกจในชุดข้อมูลนี้หายไปให้ส่งแพคเกจที่ 5 กลับมาใหม่ จากการส่ง ACK ที่ 4 กลับไปถึง 3 ครั้งกรณีเราเรียกเหตุการณ์นี้ว่าการเกิดการตอบกลับซ้ำ (DUPACK : Duplicate Acknowledgements) ซึ่งส่งผลให้เป็นภาระของระบบขึ้นและทำให้ขนาดของหน้าต่างควบคุมความคับดั่งจะต้องลดขนาดลงในที่นี้เหลือเท่ากับ 2 ในที่นี้เรียกว่า Slow start จากนั้นทีซีพีก็ทำการส่งแพคเกจขนาดหน้าต่างเท่ากับ 2 ซึ่งส่งผลให้ประสิทธิภาพการทำงานของทีซีพีลดลง



รูปที่ 2.6 แสดงการทำงานของทีซีพีและเมื่อเกิดข้อมูลสูญหาย

การเชื่อมต่อแบบ 3-Way Handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่ง และฝ่ายรับ และเพื่อกำหนดค่าเริ่มต้นของพารามิเตอร์ต่าง ๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-Way Handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับและส่งข้อมูลซึ่งกันและกันได้

ดังนั้นโพรโตคอลทีซีพีจึงเป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบ “Connection Oriented” ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น

หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อคือ

ควบคุมการรับส่งข้อมูล (Basic Data Transfer)

ความน่าเชื่อถือ ในการรับส่งข้อมูล (Reliability)

ควบคุมการไหลของข้อมูล (Flow Control)

การทำมัลติเพล็ก (Multiplexing)

ควบคุมการเชื่อมต่อ (Connection)

ความปลอดภัยในการรับส่งข้อมูล (Security)

ส่วนประกอบของทีซีพีเฮดเดอร์

Source Port : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง

Destination Port : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง

Sequence Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล

Acknowledgement Number : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ด้านรับข้อมูลปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกด้านหนึ่ง) + 1 เสมอ

Data Offset: เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก

Flag : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่

URG : Urgent Pointer Field Significant - แสดง Urgent Pointer

ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement

PSH : Push Function

RST : Reset The Connection - แสดงเมื่อรีเซ็ทการเชื่อมต่อ

SYN : Synchronize Sequence Number - หมายเลขแพคเกจที่ส่งแบบซิงโครนัส

FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว

Window : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control

Checksum : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล

Urgent Pointer : เป็นตัวชี้ตำแหน่งของ Urgent Data

Option and Padding : เป็นตัวบอกออปชันของโพรเซสที่ใช้ทีซีพี

Data : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้ และกำหนดให้เป็นศูนย์)

<table>
<tr><td colspan="8">Source Port</td><td colspan="2">Destination Port</td></tr>
<tr><td colspan="10">Sequence Number</td></tr>
<tr><td colspan="10">Acknowledgement Number</td></tr>
<tr><td>Offset</td><td colspan="2">Reserved</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td><td>Window</td></tr>
<tr><td colspan="8">Checksum</td><td colspan="2">Urgent Pointer</td></tr>
<tr><td colspan="10">Options + Padding</td></tr>
<tr><td colspan="10">Data</td></tr>
</table>

รูปที่ 2.7 แสดงส่วนประกอบของเซ็กเมนต์

โพรโตคอลยูดีพี เป็นโพรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) เช่นเดียวกับทีซีพีแต่เป็นแบบไม่ต่อเนื่อง (Connectionless) คือทั้งฝ่ายส่งและฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโตคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

โพรโตคอลไอพี (IP: Internet Protocol) เป็นโพรโตคอลที่จัดการเกี่ยวกับการหาเส้นทางของแต่ละแพคเกจเพื่อให้ส่งแพคเกจต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไป

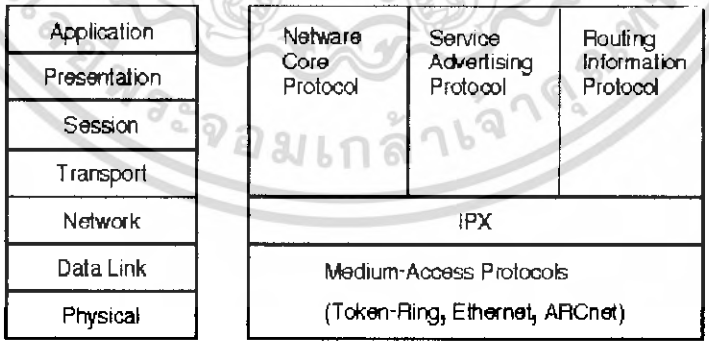
ยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอบริการหรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบไม่ต่อเนื่อง ซึ่งเป็นไปตามสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่ เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพคเกจที่ละมาตรฐานรองรับได้จึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพคเกจย่อย ๆ ในระหว่างการส่ง เรียกว่า การทำ แฟร็กเมนเตชัน (Fragmentation) เช่น แพคเกจของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพคเกจสูงสุดเพียง 1,500 ไบต์ ดังนั้นการส่งแพคเกจไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพคเกจย่อย และเมื่อแพคเกจย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพคเกจเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่า การรีแอสเซมเบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง

2.2.2 Netware Protocol

Novell ได้ทำการพัฒนา Protocol สำหรับ Netware เช่นเดียวกับ TCP/IP ซึ่ง Protocol หลักที่ถูกใช้โดย Netware มี Protocol 5 ประเภทประกอบด้วย

- Media Access Protocol
- Internetwork Packet Exchanger and Sequence Packet Exchanger (IPX/SPX)
- Routing Information Protocol (RIP)
- Service Advertising Protocol (SAP)
- NetWare Cone Protocol (NCP)

เนื่องจาก Protocol เหล่านี้มีใช้ก่อนที่จะมีการกำหนดมาตรฐาน OSI จึงทำให้มีบางส่วนไม่ตรงกับ OSI รูปที่ 2.8 แสดงถึงการจับคู่ระหว่าง Netware Protocol กับ OSI model สังเกตได้ว่ามาตรฐานทั้งสองมีขอบเขตที่ไม่เหมือนกันโดย Protocol จะทำงานในลักษณะที่มีรูปแบบเฉพาะห่อหุ้มกัน โดย Protocol ส่วนบน (NCP, SAP, RIP) จะถูกห่อหุ้มด้วย IPS/SPX และ Media Access Protocols จะห่อหุ้ม IPX/SPX ไว้อีกครั้งหนึ่ง



รูปที่ 2.8 การเปรียบเทียบ Netware กับ OSI Reference Model

1. Media Access Protocol

Media Access Protocol จะทำการกำหนดที่อยู่โดยทำให้แต่ละ Node ใน NetWare มีAddress ต่างกันโดยถูกสร้างอยู่ใน NIC โดยรูปแบบที่นิยมใช้ได้แก่

- 802.5 Token Ring

- 802.3 Ethernet
- Ethernet 2.0

Protocol นี้จะมีหน้าที่ในการจัดการกับ Header ของ Protocol โดยใน Header จะประกอบด้วยที่อยู่ต้นทางและปลายทาง เมื่อ Packet ถูกส่งออกไปยังคอมพิวเตอร์เครื่องต่างๆ ในระบบคอมพิวเตอร์แต่ละเครื่องจะทำการตรวจสอบว่าใช่ของตนหรือไม่ หรือเมื่อต้องการ Broadcast ข้อมูลออกมา NIC จะทำการคัดลอกข้อมูลเหล่านั้นให้ Protocol Stack นอกจากนั้น การกำหนด Address แล้ว Protocol ยังให้บริการตรวจสอบข้อผิดพลาดในระดับ bit แบบ Cyclical Redundancy Check (CRC) โดย CRC จะถูกส่งลงไปใน Packets มาใช้การตรวจสอบว่า Packet นั้นสมบูรณ์หรือไม่

2. Internetwork Packet Exchange and Sequence Packet Exchange (IPX/SPX)

IPX ถูกนำมาใช้ในการกำหนด ที่อยู่ในเครือข่าย ของ Network และ SPX ช่วยในการรักษาความปลอดภัยและเพิ่มความน่าเชื่อถือ แก่ IPX โดยที่ IPX เป็น Protocol ที่ทำงานอยู่ใน Network Layer มีลักษณะเชื่อมต่อแบบ Connectionless จึงไม่ค่อยน่าเชื่อถือเท่าใด และไม่ต้องการ Acknwleagement สำหรับข้อมูลที่ส่งออกไป สำหรับกลยุทธในการควบคุม และเชื่อมต่อจะทำงานโดย Protocol ที่อยู่เหนือ IPX ขึ้นไป SPX ทำงานในลักษณะ Connection-Oriented จึงมีความน่าเชื่อถือสูง

Novell ได้ใช้ Xerox Network System (XNS) Internet Datagram Protocol ในการปรับปรุง IPX Protocol โดยที่ IPX มีการจัดการกับที่อยู่ 2 แบบด้วยกัน

- Internetwork Addressing ที่อยู่ของกลุ่มเครื่องในระบบเครือข่าย ถูกกำหนดโดยหมายเลขเครือข่าย ที่กำหนดในขณะทำการติดตั้ง
- Intranode Addressing ที่อยู่ของบริการภายใน Node ถูกกำหนดโดยหมายเลข Socket Protocol แบบ IPX จะใช้กับเครือข่ายที่มี Netware Server และบ่อยครั้งที่จะถูกติดตั้งพร้อม Protocol อื่นๆ เช่น TCP/IP

3. Routing Information Protocol (RIP)

RIP ช่วยในการแลกเปลี่ยนข้อมูลในเครือข่าย Netware ถูกพัฒนาบน XNS เหมือนกับ IPX แต่ในการใช้ RIP จะมีการเพิ่มข้อมูลบาง Field เข้าไปใน Packet เพื่อช่วยในการเลือกเส้นทางในการติดต่อได้ดีขึ้น การกระจายข้อมูลของ RIP อนุญาตให้หลายเหตุการณ์เกิดขึ้น ดังนี้

- เครื่อง Workstation สามารถค้นหาเส้นทางที่เร็วที่สุดในการส่งข้อมูลได้
- Route สามารถร้องขอข้อมูลจาก Route อื่นๆ เพื่อปรับปรุงข้อมูลใน Routing Table ให้ทันสมัยตลอดเวลา
- Router สามารถสนองการร้องขอในการส่งข้อมูลจากเครื่อง Workstation และ Router ตัวอื่นๆได้
- Router สามารถตรวจพบความเปลี่ยนแปลงในระบบเครือข่าย

4. Service Address Protocol (SAP)

SAP อนุญาตให้ Node ที่ให้บริการ (รวมถึง File Service, Print Service, Gateway Service และ Application Service) ประกาศบริการที่ให้และที่อยู่ของเครื่องที่ให้บริการ ทำให้เครื่องลูกข่ายสามารถ

Access ข้อมูลของทรัพยากรในระบบเครือข่ายได้ และนอกจากการใช้ SAP Server จะทำการ Broadcast ข้อมูลทุก ๆ 60 วินาที โดย Packet ของ SAP ประกอบด้วย

- Operating Information ทำให้ทราบถึงกิจกรรมที่ Packet กำลังทำ
- Service Type ทำให้ทราบชนิดของบริการที่ให้โดยเครื่อง Server
- Service Name ทำให้ทราบชื่อเครื่องที่ใช้บริการอยู่
- Network Address ทำให้ทราบจำนวนเครื่อง Server ที่ใช้บริการอยู่
- Node Address ทำให้เราทราบหมายเลข 100 เครื่อง Server ที่ให้บริการอยู่
- Socket Address ทำให้ทราบหมายเลข Socket ของเครื่อง Server ที่ให้บริการอยู่
- Total Hops to Server ทำให้ทราบจำนวน Hop ที่จะไปถึงเครื่อง Server
- Operation Field ทำให้ทราบชนิดของการร้องขอ
- Additional Information ข้อมูล I-2 Field ที่ต่อท้าย Operation Field ประกอบด้วยข้อมูลเครื่อง Server จำนวน 1 ตัว หรือมากกว่า

5. NetWare Core Protocol (NCP)

NCP กำหนดการควบคุมการเชื่อมต่อและร้องขอการใช้บริการ ทำให้เครื่อง Server และ เครื่องลูกข่ายสามารถติดต่อระหว่างกันได้ โดย Protocol นี้จะให้บริการเชื่อมต่อระบบรักษาความปลอดภัยของ NetWare ที่รวมอยู่ในการให้บริการของ Protocol นี้

2.2.3 Network Basic Input/Output System (NetBIOS)

โปรแกรมส่วนใหญ่ที่ถูกใช้งานภายใต้ระบบปฏิบัติการ Windows ใช้ NetBIOS ในการเชื่อมต่อเรียกว่า Inter Poses Communication (IPC) NetBIOS ถูกออกแบบให้ใช้ใน LAN และถูกผลักดันให้กลายเป็นมาตรฐานสำหรับโปรแกรมในการเข้าถึงระบบเครือข่ายใน Transport Layer สำหรับ Connection-Oriented และ Non Connection-Oriented โดย NetBIOS ถูกนำมาใช้กับ NetBEUI, NWLink และ TCP/IP โดย NetBIOS ต้องการหมายเลข IP และชื่อใน NetBIOS เพื่อใช้อ้างอิงถึงเครื่องคอมพิวเตอร์

NetBIOS มีหน้าที่ 4 ประการ คือ

- NetBIOS Name Resolution เครื่องในระบบเครือข่ายต้องมีชื่ออย่างน้อย 1 ชื่อ โดย NetBIOS จะทำการเก็บข้อมูลชื่อเหล่านั้น ชื่อแรกนั้นจะเป็นชื่อ เฉพาะของ NIC และชื่อผู้ใช้สามารถถูกนำมาเสริม เพื่อช่วยให้ง่ายในการแสดงตนในระบบ โดย NetBIOS จะอ้างอิงชื่อทั้งสองสลับกันไปตามต้องการ
- NetBIOS Datagram service หน้าที่นี่จะอนุญาตให้สามารถทำการส่งข้อความ (Message) ไปยังกลุ่มของชื่อ หรือผู้ใช้ทั้งหมดของระบบก็ได้ อย่างไรก็ตาม เนื่องจาก NetBIOS ไม่ได้ใช้การเชื่อมต่อแบบ Point-to-Point ทำให้ไม่มีการรับรองว่าข้อความจะถูกส่งไปถึงเครื่องปลายทาง

- NetBIOS Session Service บริการนี้ทำให้เกิดการเชื่อมต่อแบบ Point-to-Point ระหว่างเครื่อง Workstation ในระบบเรือข่าย โดยเครื่องต้นทางจะร้องขอไปยังเครื่องอื่นๆ เพื่อเปิดการติดต่อ ทำให้สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน
- NetBIOS NIC/Session Status หน้าที่ทำให้สามารถเรียกใช้โปรแกรมอื่นๆ ที่ใช้ NetBIOS ได้ โดยใช้ข้อมูลเกี่ยวกับ NIC ของเครื่องรวมถึงเครื่องอื่นๆ ในระบบที่ทำงานอยู่แรกเริ่มเดิมที บริษัท IBM ให้แยก NetBIOS เป็นผลิตภัณฑ์ต่างหาก โดยพัฒนาเป็นโปรแกรม Terminate-and-Stay-Resident (TSR) แต่ในปัจจุบันไม่ได้ใช้โปรแกรม TSR แล้วหากคุณพบว่ามีการใช้ระบบใด ๆ อยู่ ก็สามารถทำการแทนที่ได้โดย Windows NetBIOS

2.2.4 NetBEUI

NetBEUI ย่อมาจาก NetBIOS Extended User Interface โดยแรกเริ่ม NetBIOS และ NetBEUI ถูกรวมกันอยู่ใน Protocol เดียวกัน แต่อย่างไรก็ตามบริษัทผู้ผลิตเครือข่ายได้แยก NetBIOS ที่อยู่ใน Session Layer ออกมาทำให้สามารถทำการ Route ร่วมกับ Protocol อื่นได้ NetBIOS เป็นส่วนที่ใช้ในการเชื่อมต่อเครือข่ายแบบ LAN ของ IBM ทำงานอยู่ใน Session Layer โดย NETBIOS ให้การช่วยเหลือกับโปรแกรมในการติดต่อกับเครื่องอื่นๆ ในระบบเครือข่าย และจากการที่มีโปรแกรมหลายโปรแกรมรองรับการทำงานกับ NetBEUI ทำให้ NetBEUI ถูกนำมาใช้งานอย่างแพร่หลาย

NetBEUI มีขนาดเล็ก ทำงานได้เร็วและเป็น Transport Protocol ที่มีพร้อมกับผลิตภัณฑ์ทุกชนิดของ Microsoft โดย NetBEUI มีให้มาตั้งแต่กลางปี 1980 และถูกใช้มาตั้งแต่ผลิตภัณฑ์ของระบบเครือข่ายอันแรกของ Microsoft คือ MS/NET

ข้อดีของ NetBEUI คือมีขนาดเล็ก (เหมาะสำหรับมากสำหรับเครื่องที่ใช้ MS-DOS) ความเร็วในการส่งข้อมูลผ่านสายสัญญาณ และสามารถทำงานได้กับผลิตภัณฑ์ทุกชนิดของ Microsoft

ข้อเสียของ NetBEUI คือไม่สนับสนุนการ Route ทำให้เกิดจากการจำกัดของระบบเครือข่ายที่ใช้ผลิตภัณฑ์ Microsoft และ NetBEUI จึงดีและเหมาะสมกับเครือข่ายขนาดเล็กที่ใช้ผลิตภัณฑ์ของMicrosoft

2.2.5 X.25 Product Switching

กลุ่มของ Protocol สำหรับระบบ WAN จะประกอบด้วย x.25 ซึ่งให้บริการ Switching Service มีการใช้บริการ Switching ครั้งแรกในการเชื่อมต่อเครื่องจากระบบทางไกล สู่ระบบ Mainframe โดยจะทำการแยกข้อมูลออกเป็นส่วนๆ และส่งไปในระบบเครือข่าย โดยเส้นทางระหว่าง Node จะทำผ่าน Virtual Circuit ข้อมูลจะสามารถถูกส่งผ่านทางเส้นใดก็ได้ระหว่างต้นทางและปลายทางและถึงปลายทาง Packet จะถูกนำมารวมกันอีกครั้ง

โดยปกติ Protocol จะประกอบด้วยข้อมูลจำนวน 128 Byte อย่างไรก็ตามเมื่อมีการทำการเชื่อมต่อแล้ว เครื่องต้นทางและปลายทางก็สามารถตกลงกันในเรื่องขนาดของ Packet ได้ ตามทฤษฎีแล้ว Protocol แบบ x.25 สามารถมีเส้นทางระหว่างต้นทางและปลายทางได้ เว้นตามปกติ x.25 จะทำการส่งข้อมูลความเร็ว 64 Kbps

Protocol แบบ x.25 จะทำงานใน Physical Data-Link และ Network Layer ของ OSI Model มีการใช้งานมาตั้งแต่ปี 1970 และได้รับการพัฒนามาเรื่อย ๆ ดังนั้น จึงจัดว่าเป็นระบบที่มีความน่าเชื่อถือ อย่างไรก็ตาม x.25 มีข้อเสียหลักอยู่ 2 ประการคือ

- ขบวนการ Store-and-Forward ทำให้เกิดความล่าช้า โดยปกติประมาณ 0.6 วินาที ไม่มีผลกระทบกับข้อมูลขนาดใหญ่ แต่จะสามารถสังเกตในการส่งข้อมูลแบบ Flip-Flop

- มีความต้องการ Buffer ขนาดใหญ่ในการสนับสนุนขบวนการ Store-and-Forward x.25 และ TCP/IP มีความเหมือนกันตรงที่ต่างก็เป็น Protocol แบบ Packet Switched แต่ก็มีข้อแตกต่างระหว่าง Protocol ทั้งสองตัว คือ

- TPC/IP มีการตรวจสอบข้อผิดพลาด (Error Checking) และ Flop Control ในลักษณะ End-to-End เท่านั้น แต่ x.25 มีการตรวจสอบข้อผิดพลาดในลักษณะจาก Node-to-Node

- TPC/IP มีความซับซ้อนของ Flop Control และ Window Mechanism มากกว่า x.25

- X.25 ผูกติดกับลักษณะของการเชื่อมต่อ แต่ TCP/IP ถูกออกแบบให้สามารถใช้รูปแบบการเชื่อมต่อได้หลายชนิด

2.3 การควบคุมความคับคั่งของของมูลในโปรโตคอล TCP

เมื่อปริมาณข้อมูลในระบบเครือข่ายมีมากกว่าความสามารถในการรับส่งข้อมูลเมื่อใด ก็จะเกิดปัญหาความคับคั่งของของมูล ซึ่งมีลักษณะคล้ายกับปัญหาการจราจรติดขัดในช่วงเวลาเร่งด่วน แม้ว่าโปรโตคอลในชั้นควบคุมเครือข่ายจะพยายามแก้ไขปัญหานี้แล้วก็ตาม แต่ส่วนที่จะแก้ปัญหาได้อย่างแท้จริงเป็นส่วนของโปรโตคอล TCP ซึ่งเป็นตัวที่ควบคุมการส่งหรือไม่ส่งข้อมูลไปยังชั้นสื่อสารควบคุมเครือข่าย การควบคุมในจุดนี้จึงเท่ากับเป็นการส่งข้อมูลในอัตราที่ช้าลงซึ่งจะช่วยแก้ปัญหาได้ดีกว่า

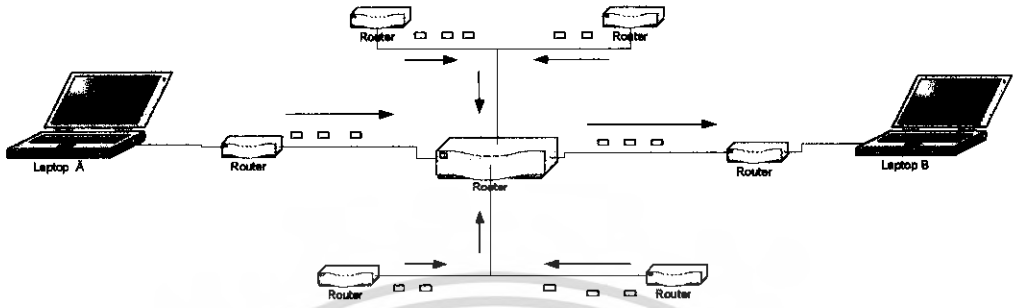
การแก้ปัญหาในทางทฤษฎีนั้นใช้หลักการทางฟิสิกส์เข้ามาช่วย คือ กฎการอนุรักษ์แพคเกจ (The law of conversation of packets) แนวความคิดพื้นฐานคือการไม่ส่งแพคเกจเข้าสู่ระบบเครือข่ายจนกว่าแพคเกจเดิมจะถูกส่งออกไปเรียบร้อยแล้ว โปรโตคอล TCP พยายามนำหลักการนี้ไปใช้โดยจัดการปรับขนาดของเซ็กเมนต์ตลอดเวลาที่ยังมีการสื่อสารเกิดขึ้น

ขั้นตอนแรกของการแก้ปัญหา คือ จะต้องสามารถตรวจพบว่ามีความคับคั่งเกิดขึ้นให้ได้ เมื่อก่อนนี้การตรวจจับทำได้ลำบากมาก เนื่องจากแพคเกจที่เดินทางมาไม่ทันระยะเวลารอคอยนั้นอาจเกิดได้จากสัญญาณรบกวนที่เกิดขึ้นในสายสื่อสาร หรือ แพคเกจถูกลบทิ้งโดย Router ที่เกิดความคับคั่งสูง ผู้รับและผู้ส่งข้อมูลไม่สามารถแยกความแตกต่างระหว่าง 2 เหตุการณ์นี้ได้

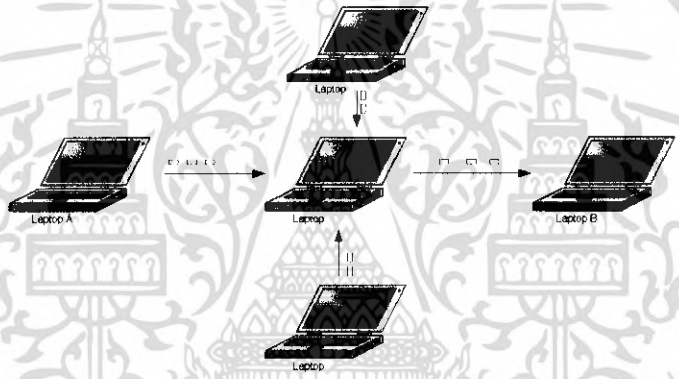
ในปัจจุบันนี้แพคเกจที่สูญหายในระหว่างการนำส่งเนื่องจากสัญญาณรบกวนนั้นเกิดขึ้นน้อยมาก (แม้ว่าในบางระบบ เช่น การสื่อสารไร้สายจะยังคงมีปัญหานี้อยู่ก็ตาม) ดังนั้นการที่แพคเกจที่เดินทางมาถึงจุดหมายไม่ทันระยะเวลารอคอยจึงเกิดขึ้นจากความคับคั่งของข้อมูลเพียงสาเหตุเดียว

เมื่อการเชื่อมต่อเริ่มเกิดขึ้นก็ต้องมีการกำหนดขนาดของหน้าต่างสื่อสาร ซึ่งก็คือขนาดสูงสุดของเซ็กเมนต์ที่อนุญาตให้ใช้ในการสื่อสาร โดยปกติผู้รับจะใช้ขนาดบัฟเฟอร์เป็นตัวกำหนด ถ้าผู้ส่งใช้ตัวเลข

นี้ในการสื่อสารอย่างเคร่งครัด ปัญหาในการสื่อสารจะไม่เกิดขึ้นที่ทางฝ่ายผู้รับอย่างแน่นอน แต่อาจทำให้เกิดความคับคั่งขึ้นในระบบเครือข่าย



(a) การเชื่อมโยงผ่านเราเตอร์ระบบใช้สาย



(b) การเชื่อมโยงผ่านเราเตอร์ระบบไร้สาย

รูปที่ 2.9 การเชื่อมโยงระหว่างสถานีที่เกิดความคับคั่ง

เมื่อ TCP ควบคุมกระแสข้อมูลด้วยการประกาศค่าหน้าต่าง เพื่อให้อีกฝ่ายทราบขนาดข้อมูลที่จะส่ง แต่วิธีนี้เป็นการควบคุมกระแสข้อมูลที่มองโดยรวมระหว่างต้นทางกับปลายทางโดยไม่คำนึงถึงกระแสข้อมูลระหว่างทางแต่อย่างใด เมื่อเราเตอร์ศูนย์กลางมีปัญหาข้อมูลล้นเกินไปแต่สถานี A ที่เป็นฝ่ายส่งไม่มีทางทราบถึงปัญหานี้ทำได้แต่รอการตอบรับจากสถานี B

เมื่อสถานี A รอการตอบรับจนกระทั่งหมดเวลาก็จะถือว่าแพคเกจที่ส่งออกไปสูญหายและต้องส่งซ้ำใหม่ ในทำนองเดียวกับสถานีอื่นในเครือข่ายที่ส่งข้อมูลผ่านเราเตอร์ตัวเดียวกันก็จะส่งแพคเกจซ้ำเช่นเดียวกับสถานี A โดยแพคเกจที่ส่งซ้ำนั้นจะยิ่งสร้างความคับคั่งให้สูงขึ้น และเป็นตัวถ่วงเวลาแพคเกจอื่นให้เดินทางช้าลงไปอีกทำให้ต้องมีการส่งซ้ำเพิ่มตามมา จึงทำให้ปัญหายิ่งลุกลามมากขึ้นจนอาจทำให้การสื่อสารทั้งหมดในเครือข่ายหยุดชะงักได้

หนทางแก้ปัญหาในระบบดังกล่าวจึงขึ้นอยู่กับความสามารถในการรับทราบปัญหาทั้ง 2 แบบ คือ ความจุข้อมูลของระบบเครือข่ายและความจุข้อมูลของผู้รับซึ่งมีวิธีแก้ไขแตกต่างกัน ผู้ส่งจะต้องเก็บตัวเลขที่นอกเหนือจากขนาดของหน้าต่างสื่อสารที่ผู้รับแจ้งให้ทราบ นั่นคือผู้ส่งต้องเก็บขนาดของหน้าต่าง

ความคับคั่ง (congestion window) ด้วย ซึ่งหน้าต่างความคับคั่งนี้มีไว้กำหนดปริมาณข้อมูลที่ส่งได้ ซึ่งเป็นเสมือนหน้าต่างกำหนดความจุตลอดทั้งเส้นทางเชื่อมโยงว่าสามารถรับแพคเกจได้ปริมาณเท่าใด

โดยตัวเลขทั้งสองตัวนี้ใช้กำหนดขนาดของเซ็กเมนต์ที่ผู้ส่งจะส่งไปยังผู้รับซึ่งจะต้องเลือกค่าที่น้อยกว่าในระหว่างตัวเลขทั้งสองตัวนี้ เช่น ถ้าผู้ใช้อนุญาตให้ใช้ขนาด 8 กิโลไบต์ แต่ผู้ส่งทราบว่าขนาดที่จะไม่ทำให้เกิดความคับคั่งจะต้องไม่เกิน 4 กิโลไบต์ ผู้ส่งก็จะส่งเซ็กเมนต์ 4 กิโลไบต์ ในทางกลับกันผู้รับอนุญาตให้ใช้ขนาด 8 กิโลไบต์ แต่ระบบเครือข่ายสามารถรองรับข้อมูลได้มากถึง 32 กิโลไบต์ ผู้ส่งก็จะส่งข้อมูลขนาด 8 กิโลไบต์ได้เต็มตามขนาดที่ผู้รับอนุญาต

เมื่อการเชื่อมต่อเริ่มขึ้น ผู้ส่งจะใช้ขนาดเซ็กเมนต์สูงสุดที่ผู้ใช้อนุญาตเป็นขนาดหน้าต่างสื่อสารความคับคั่ง หลังจากนั้นจึงเริ่มส่งเซ็กเมนต์ขนาดสูงสุดไปยังผู้รับ ถ้าได้รับการตอบรับก่อนหมดระยะเวลารอคอย ผู้ส่งจะเริ่มขนาดหน้าต่างสื่อสารความคับคั่งเป็นสองเท่าแล้วจัดการส่งเซ็กเมนต์ขนาดเดิมจำนวนสองเซ็กเมนต์ออกมาติดกัน ถ้าประสบผลสำเร็จผู้ส่งก็จะพยายามเพิ่มขนาดขึ้นไปเรื่อยๆ ครั้งละ 1 เท่าตัวของขนาดหน้าต่างสื่อสารความคับคั่งล่าสุด(ที่ประสบผลสำเร็จ)

ผู้ส่งจะปรับขนาดหน้าต่างสื่อสารความคับคั่งไปเรื่อยๆจนกว่าจะเกิดปัญหาขึ้น คือ แพคเกจตอบรับเดินทางมาไม่ถึงภายในเวลาที่กำหนดหรือส่งข้อมูลไปจนเต็มขนาดหน้าต่างสื่อสารที่ผู้ใช้กำหนด เช่น ถ้าผู้ส่งประสบความสำเร็จในการส่งข้อมูลขนาด 1024 , 2048 และ 4096 ไบต์ แต่ล้มเหลวเมื่อส่งข้อมูลขนาด 8192 ไบต์ ผู้ส่งจะต้องกำหนดขนาดหน้าต่างสื่อสารความคับคั่งไว้ที่ 4096 ไบต์ หลังจากนั้นผู้ส่งจะส่งข้อมูลออกมาครั้งละไม่เกิน 4096 ไบต์ แม้ว่าหน้าต่างสื่อสารที่ผู้ใช้กำหนดจะมีขนาดใหญ่กว่านี้ก็ตาม

การทำงานข้างต้นนี้ Jacobson เป็นคนเสนอซึ่งเป็นวิธีสองวิธีให้การทำงานร่วมกันแก้ปัญหาความคับคั่งเรียกว่า การเริ่มต้นอย่างช้า (Slow Start) และการหลีกเลี่ยงความคับคั่ง (Congestion Avoidance)

การเริ่มต้นอย่างช้าหมายถึงให้ TCP นำส่งแพคเกจทีละน้อยหลังจากเริ่มต้นสถาปนาการเชื่อมโยงและเพิ่มขึ้นจนกระทั่งถึงจุดที่เครือข่ายจะรองรับได้ ข้อดีของการเริ่มต้นอย่างช้าคือ ไม่เพิ่มภาระเครือข่ายในทันทีทันใดและเป็นการทดสอบไปพร้อมกันว่าเครือข่ายในขณะนั้นสามารถรองรับปริมาณข้อมูลได้มากน้อยพียงใด แต่ถ้าเพิ่มปริมาณส่งไปถึงจุดๆหนึ่งจนแพคเกจเกิดสูญหายจากความคับคั่ง ขั้นตอนหลีกเลี่ยงความแออัดจะรับหน้าที่ต่อเพื่อรักษาระดับการนำส่งแพคเกจให้อยู่ในปริมาณที่เหมาะสม

2.4 แนวคิดพื้นฐานของการจัดการโปรโตคอล TCP

เมื่อทั้งสองฝ่ายสถาปนาการเชื่อมต่อได้แล้ว ฝ่ายส่งจะทำการกำหนดค่า cwnd (ค่าสูงสุดของเซ็กเมนต์) เท่ากับหนึ่งเซ็กเมนต์ ซึ่งหมายถึง TCP ส่งข้อมูลโดยไม่ต้องรอการตอบรับที่ได้เพียงหนึ่งเซ็กเมนต์ และค่าอีกค่าหนึ่งกำหนดเพดานของเซ็กเมนต์เรียกว่า ssthresh (Slow Start threshold) ซึ่งปกติกำหนดให้เท่ากับ 64 กิโลไบต์

ค่า cwnd เริ่มต้นจาก 1 และจะเพิ่มขึ้นครั้งละ 1 ต่อเซ็กเมนต์ตอบรับที่ได้ ดังนั้นในขั้นแรกเมื่อได้รับเซ็กเมนต์ตอบรับแล้ว ค่า cwnd จะเพิ่มเป็น 4 และจะเพิ่มเป็น 8 หรือเพิ่มขึ้นแบบเอ็กโพเนนเชียลจนกระทั่งถึงขนาดหน้าต่างที่กำหนดหรือเกิดแพคเกจสูญหายทำให้ต้องส่งซ้ำ

หากเกิดกรณีส่งเซ็กเมนต์ซ้ำ ค่า cwnd จะนำมาหาร 2 แล้วเก็บเข้าไปในตัวแปรอีกตัวแปรหนึ่งคือค่าที่เรียกว่า ssthresh (Slow Start threshold) ถัดจากนั้นให้เริ่มต้นกระบวนการเริ่มต้นอย่างช้าใหม่อีกครั้งหรือกลับไปเซ็ตค่า cwnd เท่ากับ 1 แต่ในขั้นต่อมานี้ค่า cwnd จะมีขีดจำกัดการเพิ่มแบบเอ็กซ์โพเนนเชียลที่ไม่เกินค่า ssthresh

เมื่อ cwnd เพิ่มค่าขึ้น ssthresh จะหยุดการเพิ่มแบบเอ็กโพเนชียล (หยุดขั้นตอนเริ่มต้นอย่างช้า) แล้วเริ่มเข้าสู่กระบวนการใหม่คือหลีกเลี่ยงความคับคั่ง ในขั้นนี้ค่า cwnd จะเพิ่มขึ้นครั้งละ 1/cwnd ต่อทุกเซ็กเมนต์ตอบรับหรือเพิ่มแบบเชิงเส้น ถ้าหากไม่ปัญหาการส่งเซ็กเมนต์ซ้ำอีก ค่า cwnd ก็จะเข้าสู่สมดุลตามค่าหน้าต่างฝ่ายรับอีกครั้ง

โดยปกติแล้วหากไม่มีปัญหาใดๆ ค่า cwnd ก็จะเพิ่มขึ้นไปได้ถึง 64 เซ็กเมนต์ (ให้เซ็กเมนต์ใหญ่สุดเท่ากับ 1 กิโลไบต์) ในที่นี้สมมติว่ามีปัญหาเซ็กเมนต์สูญหายทำให้ต่อส่งซ้ำดังนั้นค่า ssthresh จะมีค่าเท่ากับครึ่งหนึ่งของค่า cwnd ล่าสุดหรือเท่ากับ 32 จากนั้นกระบวนการเริ่มต้นอย่างช้าจะเริ่มต้นขึ้น

ถัดจากนั้นกระบวนการหลีกเลี่ยงความคับคั่งจะเริ่มทำงานโดยเพิ่มค่า cwnd เป็นฟังก์ชันเชิงเส้น สมมติให้การส่งครั้งที่ 13 เกิดปัญหาอีก ค่า ssthresh จะถูกปรับเปลี่ยนเป็น 20 และเริ่มส่งเซ็กเมนต์ตามการเริ่มต้นอย่างช้าใหม่ในคราวนี้ cwnd จะเพิ่มค่าแบบเอ็กซ์โพเนนเชียลได้ไม่เกิน 20

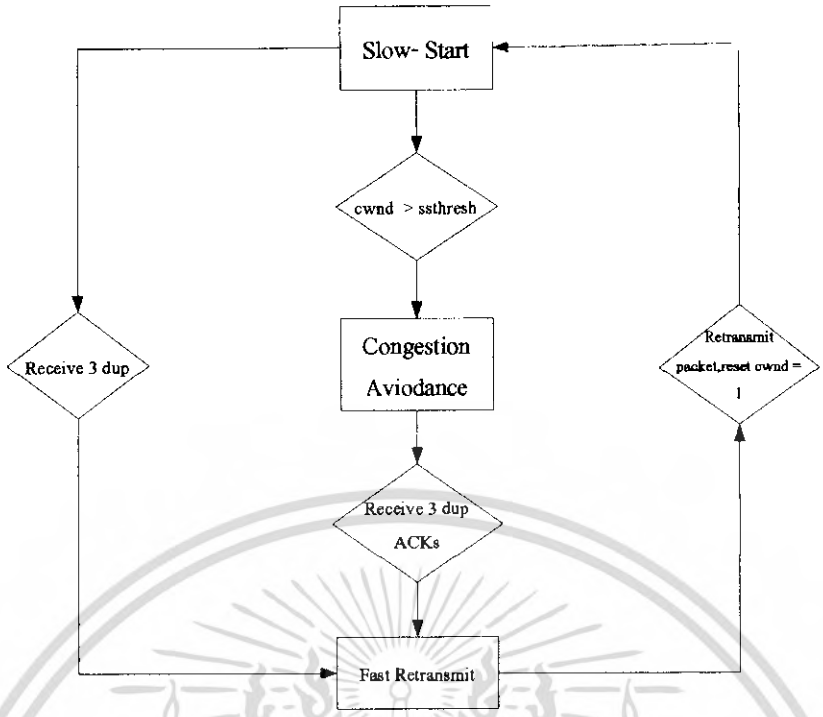
2.5 ชนิดของโปรโตคอล TCP

2.5.1 Tahoe TCP

TCP สมัยใหม่จะประกอบไปด้วยจำนวนของอัลกอริธึมที่ใช้ควบคุมความคับคั่งของเครือข่ายเพื่อช่วยทำให้ค่า Throughput ในการใช้งานมีค่าสูงขึ้น ในระยะแรก TCP จะใช้กระบวนการ Go-back-n ในการให้การให้สัญญาณการ Acknowledgement และใช้การหน่วงเวลาเพื่อทำกระบวนการ Retransmit เพื่อทำการส่งข้อมูลซ้ำในระหว่างการส่งข้อมูล สิ่งเหล่านี้เป็นส่วนช่วยให้เครือข่ายมีความคับคั่งไม่มากนัก

Tahoe TCP จะมีทั้งอัลกอริธึมใหม่และส่วนของการปรับปรุง เพื่อให้การใช้งานได้ง่ายขึ้น อัลกอริธึมนี้ประกอบด้วย Slow Start Congestion Avoidance และ Fast Retransmit ส่วนการปรับปรุงนั้นประกอบด้วยการใช้ค่า round-trip time ที่ใช้ในการตั้งค่าเวลา (Timeout) นับถอยหลังในการ Retransmit

ส่วนของอัลกอริธึมในการ Fast Retransmit เป็นส่วนที่ได้ถูกเพิ่มเติมขึ้นภายหลังโดยมีหลักการดังนี้คือ หลังจากได้รับสัญญาณ Acknowledgement ซ้ำๆ กันแล้ว ผู้ส่งจะอนุมานว่า แพคเกจได้เกิดการผิดพลาดขึ้นและจะทำให้การส่งข้อมูลซ้ำ (Retransmit) โดยไม่มีการรอให้เวลาหมด (Timeout) ในการตอบรับข้อมูลและจะเป็นการนำไปสู่การใช้ช่องสัญญาณที่คุ้มค่ารวมทั้งให้ค่า Throughput ที่ดีขึ้น



รูปที่ 2.10 Flowchart ของ Tahoe TCP

2.5.2 Reno TCP

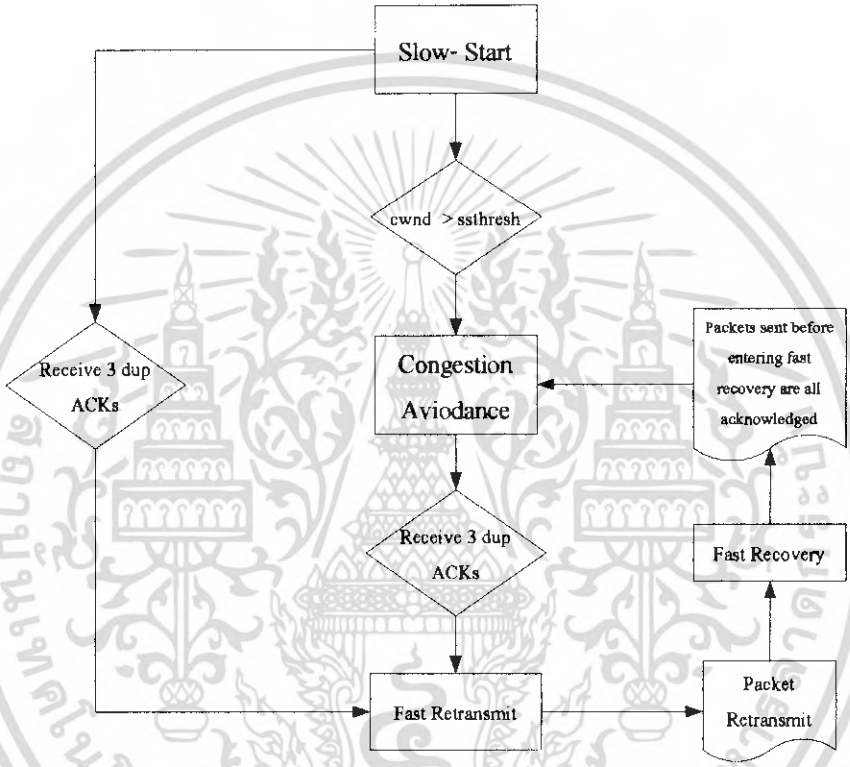
Reno TCP นั้นมีรูปแบบเหมือนกับ Tahoe แต่มีการปรับปรุงกระบวนการ Fast Retransmit โดยเพิ่มการ Fast Recovery อัลกอริธึมใหม่นี้จะป้องกันเส้นทางการสื่อสาร (ท่อ) ที่จะว่างหลังจาก Fast Retransmit และเพื่อหลีกเลี่ยงความจำเป็นในการเริ่มต้นทำกระบวนการ Slow-Start หลังจากที่แพคเกจสูญหายไป กระบวนการ Fast Recovery จะทำงานหลังจากการรับสัญญาณ ACK ที่ถูกส่งซ้ำมาเมื่อแพคเกจได้ออกจากท่อ (pipe) ไปแล้ว 1 แพคเกจ ดังนั้นในระหว่างการ Fast Recovery ตัวส่ง TCP จะสามารถคาดคะเนจำนวนข้อมูลที่ยังคงไม่ถูกส่งออกไปได้

กระบวนการ Fast Recovery จะเริ่มขึ้นโดยตัวส่งทีซีพีได้รับ Threshold ของสัญญาณ ACK ที่ส่งซ้ำมา Threshold นี้รู้จักกันในชื่อ tcprexmtthresh ที่ถูกตั้งค่าไว้ที่ 3 แต่ละครั้งของการรับ Threshold ของสัญญาณ ACK ที่ส่งซ้ำมา ผู้ส่งจะทำการ Retransmit 1 แพคเกจและลดความคับคั่งของวินโดว์ลง ½ เท่าใน Reno ตัวส่งจะใช้การเพิ่มของ ACK ที่ส่งซ้ำมาเพื่อกำหนดสัญญาณ clock ของ แพคเกจขาออก แทนการ Slow-Start ที่เป็นรูปแบบของ Tahoe TCP

ใน Reno ความสามารถในการใช้งานวินโดว์ของตัวส่งเป็น min(awin,cwnd+ndup) ซึ่ง awin เป็นวินโดว์ของตัวรับ, cwnd เป็น Congestion Window ความคับคั่งของวินโดว์ของตัวส่ง และ ndup ถูกตั้งค่าไว้ที่ 0 จนถึงจำนวนของ ACK ที่ถูกส่งซ้ำมาถึงค่า Threshold tcprexmtthresh ดังนั้นระหว่างการ Fast Recovery ตัวส่งจะทำการขยายขนาดวินโดว์จากการได้รับ ACK ที่ถูกส่งซ้ำ จากการสังเกตในแต่ละ ACK ที่ถูกส่งซ้ำนั้นบ่งชี้ว่าบางแพคเกจได้ถูกส่งออกจากเน็ตเวิร์คและไปถึงยังด้านรับแล้ว หลังจากกระบวนการ Fast Recovery และ Retransmitting 1 แพคเกจแล้ว ด้านส่งจะรอจนกระทั่งได้รับสัญญาณ ACK ที่ส่งซ้ำกลับมาเพียงครึ่งวินโดว์ และจะส่งแพคเกจใหม่สำหรับแต่ละการส่ง ACK ซ้ำกลับมา การรับสัญญาณ

ACK ของ Data ใหม่ที่เรียกว่าการ Recovery ACK นั้นด้านส่งจะทำการ Fast Recovery โดยการตั้งค่า ndup เป็น 0

อัลกอริธึมในการ Fast Recovery ของ Reno ถูกใช้ในกรณีที่แพคเกจเดี่ยวถูก drop จากวินโดว์ของข้อมูล ที่ด้านส่งใน Reno จะ Retransmit แพคเกจที่ drop 1 แพคเกจต่อ round-trip time Reno ได้ปรับปรุงจาก Tahoe TCP เมื่อแพคเกจ 1 แพคเกจถูก drop จากวินโดว์ของข้อมูลจะมีผลจากปัญหาเมื่อแพคเกจหลายๆอันถูก drop ลงจากวินโดว์ของข้อมูล ปัญหานี้ง่ายในการ simulates เมื่อเชื่อมต่อ Reno TCP ด้วยวินโดว์ขนาดใหญ่ที่จะส่งผลต่อแพคเกจที่สูญเสียหลังจากกระบวนการ Slow-Start ในเครือข่ายด้วย drop-tail gateways



รูปที่ 2.11 Flowchart ของ Reno TCP

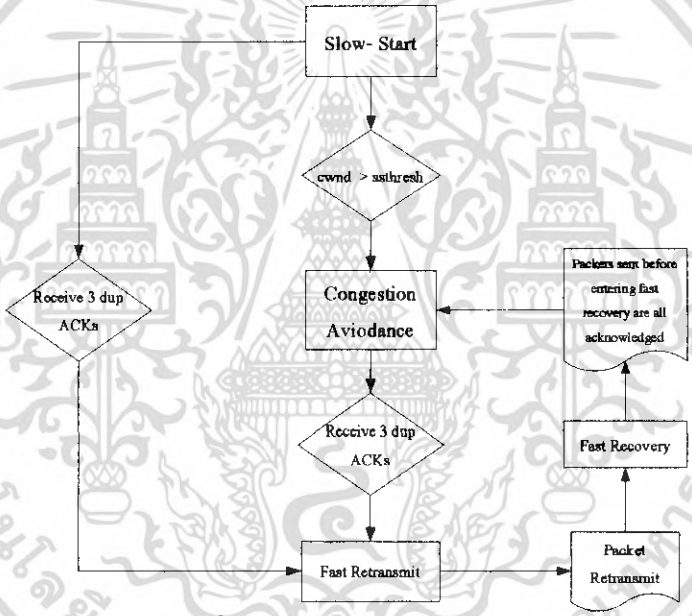
2.5.3 NewReno TCP

NewReno TCP แสดงการเปลี่ยนแปลงอย่างง่ายๆของ TCP ที่จะหลีกเลี่ยงบางปัญหาของ Reno TCP โดยปราศจากการเพิ่มของ SACK ในเวลาเดียวกัน เราใช้ NewReno TCP เพื่อสำรวจข้อจำกัดพื้นฐานของคุณสมบัติของ TCP ที่ขาดไปของ SACK

NewReno TCP จะมีการเปลี่ยนแปลงของ Reno algorithm ที่ด้านส่งจะไม่รอเวลาการ Retransmit ของ Reno เมื่อแพคเกจหลายๆ อันได้สูญเสียจากวินโดว์ การเปลี่ยนแปลงเกี่ยวข้องกับพฤติกรรมของด้านส่งระหว่างการ Fast Recovery เมื่อส่วนเล็กๆของ ACK ถูกรับได้แต่ไม่ทั้งหมด ซึ่งจะค้างอยู่ที่จุดเริ่มของช่วงระยะเวลาของการ Fast Recovery ใน Reno ส่วนเล็กๆของ ACK จะนำ TCP ออกจากการ Fast Recovery โดยการทำให้น้อยลง (deflating) การใช้วินโดว์จะกลับไปยังขนาดของ Congestion Window ส่วนใน NewReno ส่วนเล็กๆ ของ ACK จะไม่นำ TCP ออกจากการ Fast Recovery การแทนด้วยส่วน

เล็กๆ ของ ACK ที่ได้รับระหว่างการ Fast Recovery จะถูกแก้ไขด้วยการบ่งชี้ว่า แพคเกจ จะติดตามสัญญาณACK ในลำดับของส่วนที่สูญหายและจะทำการ Retransmit ดังนั้นเมื่อแพคเกจหลายๆอันสูญหายในวินโดว์ของข้อมูล NewReno จะสามารถสร้างใหม่ได้ โดยปราศจากกระบวนการรอการ Retransmission การ Retransmit ของแพคเกจ 1 แพคเกจที่สูญหายต่อ round-trip time จนกระทั่งการสูญหายของแพคเกจทั้งหมดจากวินโดว์จะได้ทำการ Retransmit แล้ว NewReno ยังคงอยู่ในกระบวนการ Fast Recovery จนกระทั่งส่วนทั้งหมดของข้อมูลที่ยังค้างอยู่ เมื่อ Fast Recovery ได้เริ่มการส่ง ACK

การแนะนำการเปลี่ยนแปลงเพิ่มเติมของอัลกอริธึมการ Fast Recovery ของ TCP ได้แนะนำว่าด้านส่งข้อมูลจะส่งแพคเกจใหม่สำหรับทุกๆ สองค่าของการส่งสัญญาณ ACK ที่ได้รับระหว่างการ Fast Recovery เพื่อเป็นการลดภาระของการ ACK สิ่งนี้มิใช่เครื่องมือใน NewReno เพราะว่าเราต้องการที่จะพิจารณาชุดของการเปลี่ยนแปลงของการเปลี่ยนแปลงของ Reno ที่จะจำเป็นในการหลีกเลี่ยงการรอเวลา (Timeout) โดยไม่จำเป็นของการ Retransmission



รูปที่ 2.12 Flowchart ของ NewReno TCP

2.5.4 SACK TCP

ดังเช่น Reno นั้น SACK TCP จะใช้การ Fast Recovery เมื่อด้านส่งข้อมูลได้รับ tcprexmtthresh ซ้ำด้านส่งจะทำการ Retransmit แพคเกจและตัดความคับคั่งของวินโดว์ (Congestion Window) ลงครึ่งเท่าระหว่างการ Fast Recovery SACK จะปรับเท่าที่เรียกว่า pipe ซึ่งแสดงจำนวนของแพคเกจที่ค้างอยู่ในเส้นทาง ด้านส่งจะส่งข้อมูลใหม่ หรือจะส่งข้อมูลใหม่ หรือส่งข้อมูลเดิมเมื่อจำนวนของแพคเกจในเส้นทางไม่น้อยกว่าค่าความคับคั่งของวินโดว์ (Congestion Windows) ค่าของ pipe จะเพิ่มขึ้นเมื่อด้านส่งส่งแพคเกจใหม่หรือส่งแพคเกจเดิมซ้ำ (Retransmit) และจะลดลงเมื่อด้านส่งได้รับแพคเกจของ ACK ซ้ำๆ ที่มี SACK ซึ่งข้อมูลใหม่จะถูกรับที่ด้านรับ

การใช้ pipe จะใช้เมื่อส่งแพคเกจจากการตัดสินของการส่งแต่ละแพคเกจ ผู้ส่งจะรักษาโครงสร้างของข้อมูล ซึ่งจะทำการ acknowledgement ได้จาก SACK ก่อนหน้านี้ เมื่อผู้ส่งได้ยอมให้ส่งแพคเกจ มัน

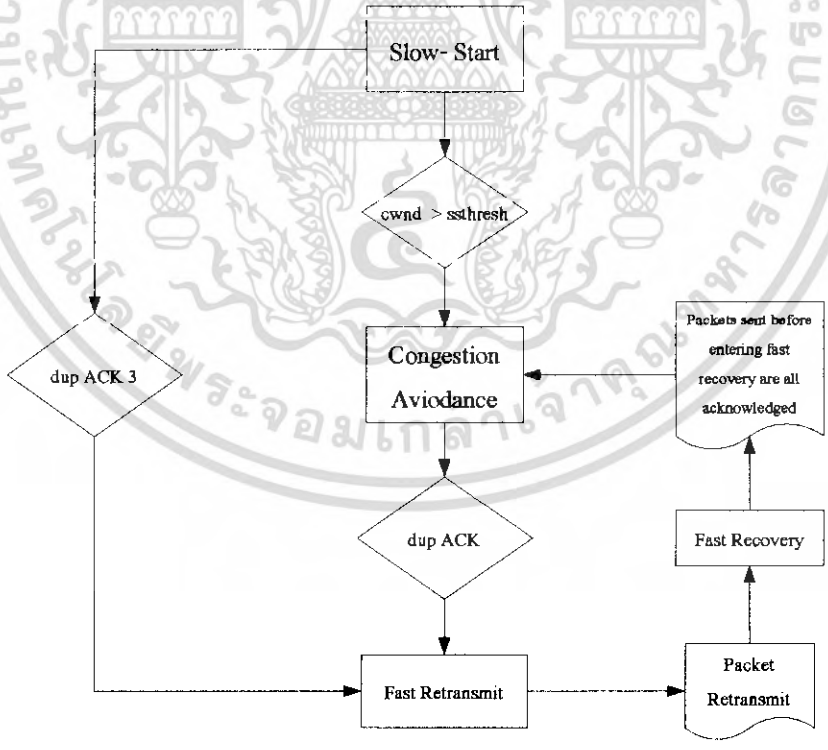
จะทำการ Retransmit แพคเกจต่อไปจากรายการของแพคเกจที่อนุมานจากการผิดพลาดที่ด้านรับ ถ้าแพคเกจ และวินโดว์ของด้านรับมีขนาดใหญ่ไม่ใหญ่มากผู้ส่งจะส่งแพคเกจใหม่

เมื่อทำการส่ง แพคเกจ ใหม่จะเกิดการ drop ลง การใช้ SACK จะป้องกันการ drop ของ timeout ในการ Retransmit โดยจะส่งแพคเกจที่ถูก drop และทำการ Slow-Start หลังจากนั้น

ด้านส่งจะทำการ Fast Recovery เมื่อได้รับ acknowledgement ของข้อมูลทั้งหมด ซึ่งกำลังค้างอยู่เมื่อได้ทำการ Fast Recovery

ด้านส่ง SACK จะมีการตรวจจับ ACK ในส่วนของ ACK ด้านส่งจะลด pipe ลงโดยใช้ 2 แพคเกจ เมื่อเริ่มการ Fast Retransmit pipe จะถูกลดสำหรับแพคเกจที่จะถูก drop ลงและจะเพิ่มขึ้นสำหรับแพคเกจที่ถูก Retransmit ดังนั้นการลดของ pipe ที่เกิดจาก 2 แพคเกจ เมื่อ ACK แรกได้รับและได้แสดงว่าแพคเกจ ได้ส่งออกไป อย่างไรก็ตามเพื่อการรับ ACK ได้ pipe จะถูกเพิ่มเมื่อแพคเกจที่ถูกส่งซ้ำถูกส่งเข้าไปใน pipe แต่จะไม่ลดสำหรับแพคเกจที่ถูก drop ดังนั้นเมื่อ ACK ส่งมาถึงมันจะแสดงว่า 2 แพคเกจได้ถูกส่งออกจาก pipe เป็นแพคเกจเริ่มต้นและแพคเกจที่ส่งซ้ำ เพราะว่าด้านส่งจะลด pipe ด้วย 2 แพคเกจที่มากกว่า สำหรับ ACK SACK ที่ด้านส่งจะไม่ทำการเร็วกว่า Slow-Start

มีจำนวนของสัดส่วนเพื่อการควบคุมอัลกอริธึมของความคับดั่งของ TCP ที่ใช้การเลือกการ Acknowledgement การใช้ SACK ในการ simulate จะถูกออกแบบให้มีการขยายของอัลกอริธึมที่ควบคุมความคับดั่งของ Reno ซึ่งจะทำให้เกิดการเปลี่ยนของอัลกอริธึมที่ควบคุมความคับดั่งของ Reno เพียงเล็กน้อย



รูปที่ 2.13 Flowchart ของ SACK TCP

2.5.5 Vegas TCP

ทุกวิธีที่นำสู่การพัฒนาของ Tahoe TCP และ Reno TCP ในปี 1996 Brakmo ได้แนะนำ TCP Vegas ที่ดีกว่าโดย Vegas จะเพิ่มกลไกในการ Retransmit ของ Reno ขั้นแรก Vegas จะอ่านและบันทึก clock ของระบบในแต่ละส่วนที่ถูกส่ง เมื่อสัญญาณ ACK มาถึง Vegas จะอ่านค่า clock อีกครั้งและทำการคำนวณแบบ RTT Vegas จะใช้ค่า RTT ที่จะตัดสินว่าจะทำการ Retransmit ใน 2 กรณีดังนี้

1. เมื่อได้รับสัญญาณ ACK ซ้ำ

Vegas จะตรวจความแตกต่างระหว่างค่าปัจจุบัน และบันทึกค่า Timestamp เพื่อให้ส่วนที่สัมพันธ์กันมีความสำคัญกว่าค่า Timeout ถ้าเป็นดังนี้ Vegas จะทำการ Retransmit ในส่วนที่ไม่ต้องรอให้มีการส่งสัญญาณ ACK ซ้ำกันถึง 3 ครั้ง ในกรณีอื่นที่มีการสูญเสีย ด้านส่งจะไม่รับ ACK ทั้ง 3 ครั้ง ดังนั้น Reno จะต้องคอบในเวลา Timeout

2. เมื่อไม่มีการรับสัญญาณ ACK ซ้ำ

ถ้า ACK ที่ถูกส่งมาอันแรกหรืออันที่สองหลังจากการ Retransmit Vegas จะตรวจถ้าเวลาช่วงในตั้งแต่ส่วนที่ถูกส่งมีขนาดใหญ่กว่าเวลา Timeout ถ้าเป็นดังนี้ Vegas จะทำการ Retransmit จะทำให้ส่วนอื่นๆที่สูญเสียก่อนหน้าที่จะมีการ Retransmit ที่ไม่มีการรอสัญญาณ ACK ที่ซ้ำถึง 3 ครั้ง

สำหรับกระบวนการ Congestion Avoidance ของ Vegas จะใช้พื้นฐานการเปลี่ยนแปลงจำนวนของข้อมูลในเครือข่าย Vegas จะกำหนดให้ BaseRTT ของการเชื่อมต่อให้เป็น RTT ของเซ็กเมนต์เมื่อมีการเชื่อมต่อไม่มีความคับคั่ง ในทางปฏิบัติ Vegas จะตั้งค่า BaseRTT ให้น้อยที่สุดของ round trip time ที่วัดได้ RTT ของเซ็กเมนต์แรกส่งโดยการเชื่อมต่อ ก่อนที่เราเตอร์จะเข้าคิวเพิ่มขึ้นเนื่องมาจากทราฟฟิกที่สร้างจากการเชื่อมต่อนี้ Vegas ใช้ค่านี้ทำการคำนวณค่า Throughput ที่คาดไว้ ในขั้นที่สอง Vegas จะคำนวณค่าอัตราเร็วในการส่งข้อมูล สิ่งนี้ทำได้โดยบันทึกเวลาในการส่งสำหรับแบ่งเซ็กเมนต์ บันทึกจำนวนของ bytes ที่ถูกส่งระหว่างเวลาที่เซ็กเมนต์ถูกส่งและได้รับสัญญาณ ACK ทำการประมวลผล RTT สำหรับแบ่งเซ็กเมนต์เมื่อสัญญาณ ACK มาถึง และแบ่งจำนวนของ bytes ที่ถูกส่งโดย sample RTT การคำนวณนี้ถูกทำหนึ่งครั้งต่อ round trip time สำหรับในขั้นที่สาม Vegas จะเปรียบเทียบค่า Throughput ที่คาดไว้แล้วและปรับขนาดของวินโดว์ ค่า Diff เป็นค่าความแตกต่างระหว่างค่าของ Throughput ที่ได้จากการปฏิบัติและค่าที่ได้จากการคาดการณ์ไว้ ดังนั้น Vegas จะกำหนดค่า Threshold 2 ค่าคือค่า α และ β ที่จะมีน้อยกว่าและมากกว่าข้อมูลในเครือข่าย ตามลำดับ

cwnd +1 if Diff $< \alpha$

cwnd – 1 if Diff $> \beta$

cwnd otherwise ($\alpha \leq Diff \leq \beta$)

Vegas มี 3 เทคนิคในการเพิ่ม throughput และลดการสูญเสียของแพคเกจและการ Retransmit เทคนิคแรกคือ Congestion Avoidance เป็นการปรับค่าความคับคั่งวินโดว์ของ TCP แบบเส้นตรงทั้งขาขึ้นและขาลงโดยจะมีค่าที่คงที่ของจำนวนบัฟเฟอร์ในเครือข่าย อัลกอริธึมของ Congestion Avoidance ของ Vegas จะรักษาจำนวนของส่วนในการส่งให้พอที่จะสามารถตั้งสัญญาณ clock ได้ เพื่อการป้องกันเทคนิค

ที่ 2 จะใช้ 2 สิ่งที่จะป้องกันการสูญหายของแพคเกจง่ายกว่า Reno และใช้การลดที่ต่ำกว่า Reno ในเทคนิคที่ 3 Vegas จะลดลงครึ่งหนึ่งในการเพิ่มของอัตราการ Slow-Start ที่จะป้องกันการสูญหายของแพคเกจ

2.6 โปรโตคอลการจัดเส้นทาง (Routing Protocol)

โปรโตคอลการจัดเส้นทางมีความจำเป็น เนื่องจากการส่งเพ็กเกตข้อมูลจากโหนดต้นทางไปยังโหนดปลายทาง อาจมีการใช้เส้นทางที่มีจำนวนฮอปมากกว่าหนึ่งฮอป หน้าที่หลักอย่างหนึ่งของโปรโตคอลชนิดนี้คือการค้นหาและเลือกใช้เส้นทางของคู่โหนดต้นทางและปลายทางที่มีอยู่ เพื่อให้การส่งข้อมูลนั้นเป็นไปอย่างถูกต้อง

เมื่อโปรโตคอลเพื่อการใช้เส้นทางมีความจำเป็น การไม่นำโปรโตคอลเดิมที่มีอยู่แล้ว เช่น เวกเตอร์บอกระยะ หรือสถานะลิงค์มาใช้กับเน็ตเวิร์คไร้สายแบบเฉพาะกิจ เนื่องจากหลายการทดสอบในกลุ่มที่ทำงานด้านเน็ตเวิร์คไร้สายแบบเฉพาะกิจ ระบุปัญหาหลักของการใช้โปรโตคอลเหล่านี้ คือ มักถูกออกแบบมาเพื่อใช้ในเน็ตเวิร์คที่มีภูมิลักษณะของเน็ตเวิร์คแบบสถิต (Static Topology) ทำให้โปรโตคอลเกิดปัญหาเมื่อภูมิลักษณะของเน็ตเวิร์คมีการเปลี่ยนแปลงอยู่ตลอดเวลา

โปรโตคอลสถานะลิงค์ และเวกเตอร์บอกระยะทำงานได้ในเน็ตเวิร์คไร้สายแบบเฉพาะกิจที่มีการเคลื่อน ที่ของโหนดน้อย ทำให้การเปลี่ยนแปลงของภูมิลักษณะของเน็ตเวิร์คไม่มากนัก แต่นอกจากปัญหาเกี่ยวกับการเปลี่ยนแปลงของภูมิลักษณะของเน็ตเวิร์คบ่อยแล้ว ในการทำงานของโปรโตคอลเหล่านี้คือมีการส่งข้อความควบคุม (Control Messages) เป็นช่วงๆ เพื่อใช้ในการกำหนดเส้นทางหรือปรับปรุงข้อมูลเส้นทาง และถ้าจำนวนโหนดในเน็ตเวิร์คมากขึ้นข้อมูลเหล่านั้นก็เพิ่มมากขึ้นและบ่อยขึ้นตามไปด้วย ทำให้เกิดการใช้ทรัพยากรของเน็ตเวิร์คในส่วนแบนด์วิดธ์ พลังงานและความสามารถของหน่วยประมวลผลกลางมากเกินไป

นอกจากนั้นโปรโตคอลการจัดเส้นทางแบบเดิมเหล่านี้ ยังใช้สมมติฐานว่าลิงค์ที่เกิดขึ้นเป็นแบบสอง ทิศทาง (Bi-directional Link) นั้นคือการส่งผ่านข้อมูลระหว่างโหนดสามารถทำงานได้เช่นเดียวกันทั้งไปและกลับ ซึ่งในเน็ตเวิร์คไร้สายแบบเฉพาะกิจอาจไม่เป็นเช่นนั้นเสมอไปเนื่องจากโปรโตคอลการจัดเส้นทางในเน็ตเวิร์คไร้สายแบบเฉพาะกิจ ที่ถูกนำเสนอขึ้นมานั้นส่วนใหญ่มีการพัฒนาจากอัลกอลิธึมของโปรโตคอลการจัดเส้นทางแบบเดิม ดังนั้นจึงจำเป็นต้องทำความเข้าใจการทำงานของโปรโตคอลการจัดเส้นทางแบบเดิม

2.6.1 โปรโตคอลแบบสถานะลิงค์ (Link State)

ในการทำงานของโปรโตคอลสถานะลิงค์ แต่ละโหนดจะทำการปรับปรุงข้อมูลของเส้น ทางของตัวเอง โดยการมองภูมิลักษณะของเน็ตเวิร์คทั้งหมดเพื่อหาค่าของแต่ละลิงค์ ในการหาค่าของลิงค์เหล่านี้แต่ละโหนดจะทำ การบรอดแคสต์ (Broadcast) ค่าของลิงค์ของตัวเองไปยังโหนดอื่นๆด้วยวิธีฟลัดดิง (Flooding) โหนดที่ได้รับข้อมูลนี้จะทำการปรับข้อมูลค่าลิงค์ของตัวเองและทำการคำนวณฮอปต่อไปสำหรับแต่ละโหนดปลายทาง

ค่าลิงค์ของแต่ละโหนดมีความผิดพลาดได้ เนื่องจาก ระยะเวลาหน่วงในการกระจายข้อมูลบางส่วนของเน็ตเวิร์คแยกออกหรืออย่างอื่น ทำให้การมองเครือข่ายของโหนดผิดไปนำไปสู่การเกิด

เส้นทางวนรอบ (Routing Loop) แต่ลูปเหล่านี้จะมีช่วงชีวิตสั้นเพราะจะหายไปเมื่อได้รับข้อมูลค่าลิงค์ในครั้งต่อมา

2.6.2 โปรโตคอลเวกเตอร์บอกระยะ (DV: Distance Vector)

การทำงานของโปรโตคอลเวกเตอร์บอกระยะ (Distance Vector) แต่ละโหนดจะทำการเฝ้าตรวจสอบเฉพาะค่าลิงค์ที่โหนดส่งข้อมูลออกไปเท่านั้น ไม่บรอดแคสต์ไปทุกๆโหนดในเน็ตเวิร์คโหนดทำการส่งข้อมูลการประเมินเส้นทางไปทุกโหนดปลายทางในเน็ตเวิร์คที่สั้นที่สุด ไปให้โหนดข้างเคียงเท่านั้น และโหนดที่ได้รับข้อมูลจะทำประเมินเส้นทางที่สั้นที่สุดไปยังโหนดปลายทางอีกครั้ง

เปรียบเทียบกับโปรโตคอลแบบสถานะลิงค์แล้ว โปรโตคอลเวกเตอร์บอกระยะใช้ประสิทธิภาพในการคำนวณมากกว่า แต่ง่ายกว่าในการใช้งานจริงเพราะมีความต้องการการจัดเก็บข้อมูลน้อยกว่า แต่อย่างไรก็ตามเป็นที่รู้กันดีว่าโปรโตคอลชนิดนี้ สามารถทำให้เกิดสาเหตุของการเกิดลูปทั้งชนิดช่วงชีวิตสั้นและช่วงชีวิตยาวได้สาเหตุหลักคือ โหนดเลือกฮอปต่อไปในการส่งข้อมูลโดยการใช้พื้นฐานจากข้อมูลเก่าที่ค้างอยู่

2.6.3 โปรโตคอลแบบซอสเราติง (Source Routing)

ซอสเราติง (Source Routing) เป็นวิธีการที่แต่ละแพคเกจต้องมีค่าของเส้นทางที่สมบูรณ์ที่แพคเกจนั้นๆ ใช้เพื่อเดินทางสู่โหนดปลายทาง การตัดสินใจและพิจารณาการใช้เส้นทางจะเกิดขึ้นที่โหนดต้นทาง ข้อได้เปรียบของวิธีการนี้คือเป็นวิธีการที่ง่ายที่จะหลีกเลี่ยงการเกิดเส้นทางวนรอบข้อเสียก็คือแต่ละแพคเกจมีเฮดเดอร์มากขึ้น

2.6.4 โปรโตคอลแบบฟลัดดิง (Flooding)

โปรโตคอลใช้การบรอดแคสต์เพื่อกระจายข้อมูลการควบคุม (Control Information) ไปในเน็ตเวิร์คจากจุดโหนดเริ่มต้นไปยังทุกๆ โหนดในเน็ตเวิร์ค การกระจายข้อมูลในลักษณะการฟลัดดิงถูกนำไปใช้อย่างกว้างขวาง วิธีการคือโหนดเริ่มต้นจะทำการกระจายข้อมูลไปยังโหนดใกล้เคียงโหนดที่ได้รับข้อมูลจะพิจารณาข้อมูลเหล่านั้นก่อน แล้วจึงทำการส่งต่อข้อมูลนั้นไปยังโหนดข้างเคียงอีกครั้งจนกว่าข้อมูลจะไปถึงทุกโหนดในเน็ตเวิร์ค แต่ละโหนดจะรับข้อมูลนี้เพียงครั้งเดียวเพื่อให้มั่นใจจึงมีการใช้ส่วนของตัวเลขลำดับ (Sequence Number) ในการสร้างข้อมูลแต่ละครั้งและตัวเลขลำดับนี้จะเพิ่มค่าทุกครั้งที่มีการสร้างข้อมูลใหม่

2.7 การแบ่งประเภทโปรโตคอลการจัดเส้นทาง

โปรโตคอลการจัดเส้นทางสามารถแบ่งประเภทได้หลายลักษณะตามคุณสมบัติต่างๆ ขึ้นกับคุณสมบัติที่ใช้ในการแบ่งประเภท เช่น

1) รวมศูนย์ (Centralized) และ กระจาย (Distributed)

2) สถิต (Static) และ อะแดปทีฟ (Adaptive)

3) รีแอคทีฟ (Reactive) และ โปรแอคทีฟ (Proactive)

วิธีการหนึ่งในการจัดประเภทโปรโตคอลการจัดเส้นทาง คือการแบ่งด้วยลักษณะรวมศูนย์หรือแบบกระจาย ลักษณะรวมศูนย์ คือการเลือกเส้นทางทุกครั้งจะทำในโหนดศูนย์กลางโหนดหนึ่ง ในขณะที่แบบกระจายเป็นการร่วมกันของโหนดในเน็ตเวิร์คในการคำนวณเส้นทาง

ในการแบ่งประเภทอีกอย่างคือพิจารณาจากการตอบสนองต่อสภาวะปริมาณการใช้ข้อมูล(Traffic Pattern) เพื่อทำการเปลี่ยนเส้นทางที่ใช้ในขณะนั้น โปรโตคอลที่เป็นแบบสถิต ใช้เส้น ทางระหว่างต้นทางและปลายทางคงที่ โดยไม่สนใจการเปลี่ยนแปลงของสภาวะความหนาแน่นของข้อมูล แต่เส้นทางสามารถเปลี่ยนได้เมื่อเส้นทางที่ใช้อยู่ไม่สามารถใช้ได้อีกต่อไป โปรโตคอลเหล่านี้ไม่สามารถเพิ่มทรูพุตได้เมื่อมีรูปแบบปริมาณการใช้ข้อมูลที่แตกต่างกัน ส่วนโปรโตคอลแบบอะแดปทีฟ ใช้ในเน็ตเวิร์คที่ต้องการการตอบสนองต่อความดับดั่งของการสัญจรข้อมูล

การแบ่งประเภทแบบสุดท้ายเกี่ยวข้องกับโปรโตคอลการจัดเส้นทางที่ใช้เน็ตเวิร์คไร้สายแบบเฉพาะกิจมากกว่าแบบอื่น โปรโตคอลแบ่งออกเป็นชนิดโปรแอคทีฟ (Proactive) และ รีแอคทีฟ (Reactive) โปรโตคอลแบบโปรแอคทีฟคือ โปรโตคอลที่มีการประเมินเส้นทางทั้งเน็ตเวิร์คเป็นช่วงๆ อยู่ตลอดเวลา ดังนั้นเมื่อมีแพคเกจที่ต้องการส่งออกไปยังปลายทาง จะมีเส้นทางพร้อมใช้อยู่แล้ว สามารถส่งออกไปได้ทันที โปรโตคอลในตระกูลของเวกเตอร์บอกระยะจะมีแบบแผนการทำงานเป็นแบบโปรแอคทีฟ ในส่วนของโปรโตคอลแบบรีแอคทีฟมีแบบแผนการทำงานอีกอย่างคือมีการเริ่มต้นกลไกการค้นหาเส้นทางก็ต่อเมื่อมีการร้องขอใช้เส้นทางเท่านั้น เมื่อมีความต้องการใช้เส้นทางขบวนการค้นหาเส้นทางจะเริ่มขึ้น มีการใช้แนวทางการทำงานของโปรโตคอลในตระกูลฟลัดดิง การหน่วงในช่วงการเริ่มต้นการส่งข้อมูลอาจมากกว่าโปรโตคอลแบบโปรแอคทีฟแต่เหมาะสมกว่าในเน็ตเวิร์คที่มีการเปลี่ยนแปลงภูมิลักษณะของเน็ตเวิร์คบ่อยๆ

2.8 อัลกอริธึมการจัดเส้นทาง

2.8.1 โปรโตคอล Destination Sequenced Distance Vector (DSDV)

Destination Sequenced Distance Vector (DSDV) เป็นโปรโตคอลการจัดเส้นทางแบบเวกเตอร์บอกระยะ ที่มีลักษณะเป็นฮอปต่อฮอป (Hop-by-Hop) คือแต่ละโหนดมีตารางเก็บค่าเส้นทางไปยังโหนดปลายทางทุกโหนดในเน็ตเวิร์ค ซึ่งจะเก็บค่าในแต่ละเส้นทางที่ไปยังโหนดปลายทางด้วยฮอปต่อไป และจำนวนฮอปที่ใช้ในเส้นทางนี้เหมือนกับเวกเตอร์บอกระยะโปรโตคอลชนิดนี้มีปรับปรุงข้อมูลในตารางเส้นทางด้วยการบรอดแคสต์เป็นช่วงๆ DSDV มีข้อดีมากกว่าเวกเตอร์บอกระยะแบบเดิมคือการรับประกันว่าจะไม่เกิดเส้นทางวนรอบ

เพื่อไม่เกิดเส้นทางวนรอบ DSDV ใช้ตัวเลขลำดับเพิ่มเข้าไปในข้อมูลเส้นทางแต่ละเส้น ตัวเลขลำดับเหล่านี้จะเป็นการแสดงความใหม่ของข้อมูลเส้นทาง ข้อมูลเส้นทางมีเลขลำดับมากกว่าเป็นเส้นทางที่เหมาะสมมากกว่า หากเส้นทาง R จะมีความเหมาะสมมากกว่าเส้นทาง R′ เมื่อเลขลำดับของ R มากกว่า R′ หรือถ้าเลขลำดับทั้งสองเท่ากันแต่จำนวนฮอปของ R น้อยกว่า R′ เลขลำดับของเส้นทางจะเพิ่มขึ้นเมื่อโหนด A ตรวจพบว่าเส้นทางไปยังโหนด D ไม่สามารถใช้งานได้อีกต่อไป ดังนั้นในการประกาศข้อมูล

เส้นทางครั้งต่อไปของโหนด A จะประกาศข้อมูลเส้นทางไปยังโหนด D ด้วยจำนวนฮอบเท่ากับค่าไม่รู้จบ (Infinite) และเลขลำดับเพิ่มขึ้นจากเดิม

โปรโตคอล DSDV มีพื้นฐานมาจากเวกเตอร์บอกระยะ โดยมีการปรับบางส่วนเพื่อให้ใช้งานได้ดีขึ้นในเน็ตเวิร์คไร้สายแบบเฉพาะกิจ มีการปรับในส่วนการปรับปรุงข้อมูลเส้นทางโดยลดขนาดของข้อมูลเหล่านั้นซึ่งจะมีการกำหนดเป็นสองแบบคือ การถ่ายข้อมูลแบบเต็ม (Full Dump) คือการส่งข้อมูลเส้นทางทั้งหมดที่มีอยู่และการถ่ายข้อมูลแบบส่วนเพิ่มเติม (Incremental Dump) คือส่งข้อมูลเฉพาะในส่วนที่มีการเปลี่ยนแปลง

เนื่องจาก DSDV ปรับข้อมูลเส้นทางโดยการรับข้อมูลที่บรอดแคสต์มาเป็นช่วงๆ จากโหนดข้างเคียงเวลาที่ข้อมูลทั้งหมดจากโหนดจากข้างเคียงทุกโหนดได้มาครบแล้ว จึงมาประเมินหาค่าเส้นทางได้ ช่วงเวลาเหล่านี้ไม่ได้นำมาพิจารณาในเน็ตเวิร์คแบบสถิตซึ่งมีภูมิลักษณะมีการเปลี่ยนแปลงน้อย แต่ในเน็ตเวิร์คที่มี การเปลี่ยนแปลงภูมิลักษณะบ่อยๆ อย่างในเน็ตเวิร์คไร้สายแบบเฉพาะกิจในช่วงเวลาเหล่านั้นอาจทำให้เกิดการละทิ้งแพคเกจข้อมูลที่ส่งไปได้ เนื่องจากเส้นทางนั้นไม่สามารถใช้งานได้ และในขณะเดียวกันการเพิ่มความถี่ในการบรอดแคสต์ก็จะทำให้เป็นการเพิ่มโอเวอร์เฮดในเน็ตเวิร์ค

2.8.2 โปรโตคอล Ad-hoc On Demand Distance Vector (AODV)

Ad-hoc On Demand Distance Vector (AODV) เป็นโปรโตคอลการจัดเส้นทางในเน็ตเวิร์คไร้สายแบบเฉพาะกิจ ทำให้โหนดสามารถติดต่อกันได้โดยที่เส้นทางอาจเป็นมีหลายฮอปโปรโตคอลมีพื้นฐานจากโปรโตคอลเวกเตอร์บอกระยะ แต่ AODV จะมีการทำงานเป็นแบบรีแอคทีฟ คือขบวนการค้นหาเส้นทางเกิดขึ้นเมื่อมีการร้องขอใช้เส้นทางนั้นเท่านั้น และโหนดไม่จำเป็นต้องทำการปรับปรุงข้อมูลเส้นทางไปยังโหนดปลายทางที่ยังไม่ใช้งานในขณะนั้น และในขณะการสื่อสารดำเนินอยู่โดยเส้นทางยังทำงานได้ AODV ก็จะไม่ทำงานใดๆเลย

ลักษณะเฉพาะของโปรโตคอล มีทั้งการปราศจากการเกิดเส้นทางวนรอบ มีการแจ้งการเกิดลิงค์ล้มเหลวไปยังโหนดที่มีผลกระทบจากเส้นทางที่ไม่สามารถใช้งานได้ทันที นอกจากนั้นโปรโตคอลยังสนับสนุนการใช้เส้นทางแบบมัลติแคสต์ (Multicast) ใช้เลขลำดับของแต่ละปลายทางเพื่อระบุความใหม่ของเส้นทาง

โปรโตคอลใช้ข้อความควบคุมแตกต่างกันในขบวนการค้นหาเส้นทาง และขบวนการปรับปรุงข้อมูลเส้นทาง เมื่อใดก็ตามที่มีโหนดต้องการเส้นทางไปยังโหนดปลายทาง โหนดจะทำการบรอดแคสต์ ROUTE REQUEST (RREQ) ไปยังทุกโหนดข้างเคียง ข้อความนี้จะกระจายไปทั่วเน็ตเวิร์คจนกระทั่งไปถึงโหนดปลายทางหรือโหนดที่มีเส้นทางไปยังโหนดปลายทางและเป็นเส้นทางที่ใหม่พอ เมื่อได้เส้นทางแล้วจะมีการส่ง ROUTE REPLY (RREP) แบบยูนิแคส (Unicast) กลับไปยังโหนดต้นทาง

มีอัลกอลิทึมใช้ข้อความทักทาย (Hello Messages) เป็น RREP พิเศษแบบหนึ่ง (บรอดแคสต์เป็นช่วงๆ ไปยังโหนดข้างเคียง) เพื่อเป็นการประกาศความคงอยู่ของโหนดเองต่อโหนดข้างเคียงโหนดที่ได้รับข้อความนี้นำไปปรับปรุงข้อมูลเส้นทางในตารางของตนเอง เส้นทางใดที่ผ่านโหนดนั้นยังทำงานได้ แต่ถ้าไม่ได้รับข้อความนี้ในระยะหนึ่ง ทำให้เข้าได้ว่าโหนดอาจเคลื่อนออกจากระยะที่จะติดต่อได้ ทำให้ข้อมูล

เส้นทางที่ผ่านทางโหนดนั้นถูกกำจัดออกไปจากตารางและแจ้งการเกิดลิงค์ล้มเหลวไปยังโหนดที่ใช้งานเส้นทางนี้

การจัดการตารางข้อมูลเส้นทาง AODV แต่ละโหนดมีการเก็บข้อมูลเส้นทางในตารางแต่ละเส้นทางด้วยข้อมูลต่อไปนี้

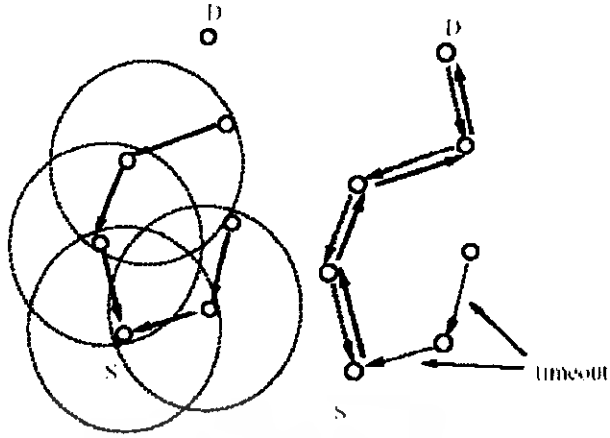
- IP address โหนดปลายทาง
- เลขลำดับของเส้นทางไปโหนดปลายทาง
- จำนวนฮอป
- ช่วงชีวิต เป็นระยะเวลาที่เส้นทางยังเป็นเส้นทางที่ใช้งานได้
- รายชื่อโหนดข้างเคียงที่ต้องยังอยู่ เพื่อให้เส้นทางใช้งานได้
- ส่วนพักของรีเควสต์ (Request Buffer)

การค้นหาเส้นทางเริ่มขึ้นเมื่อโหนดต้นทางมีความต้องการสื่อสารกับโหนดอื่น แต่ไม่มีเส้นทางไปยังโหนดนั้นในตารางข้อมูลเส้นทาง แต่ละโหนดมีการปรับปรุงข้อมูลของเลขลำดับของโหนดอยู่สองค่าคือ เลขลำดับของโหนดและเลขลำดับของข้อความบรอดแคสต์ โหนดต้นทางเริ่มการค้นหาเส้นทางโดยการส่งข้อความ RREQ ไปสู่โหนดข้างเคียง ข้อความ RREQ ประกอบด้วย <source_addr, source_sequence_#, broadcast_id, dest_addr, dest_sequence_#, hop_cnt >

คู่ของค่าข้อมูล < source_addr, broadcast_id > ทำให้สามารถอ้างข้อความได้เพียงหนึ่งเดียว broadcast_id มีค่าเพิ่มขึ้นเมื่อทุกครั้งที่โหนดต้นทางทำการส่งข้อความ RREQ ใหม่ทุกครั้ง โหนดข้างเคียงที่ได้รับข้อความนี้ถ้ามีเส้นทางหรือเป็นโหนดปลายทางก็จะทำการส่งข้อความ RREP กลับสู่โหนดต้นทาง หรือไม่ก็จะทำส่งต่อข้อความนี้ออกไปสู่โหนดข้างเคียงอีกครั้งโดยทำการเพิ่มค่า hop_cnt ก่อนส่งออกไป โหนดอาจได้รับข้อความนี้หลายครั้งแต่จะละทิ้งข้อความนี้ถ้าโหนดเคยรับมาก่อนแล้ว เมื่อได้รับข้อความโหนดจะเก็บค่าต่อไปนี้เพื่อทำการสร้างเส้นทางกลับยังโหนดต้นทาง(Reverse Path) เช่นเดียวกันเมื่อได้รับข้อความ RREP ก็จะเก็บค่าข้อมูลเพื่อสร้างเส้นทางส่งต่อไปยังโหนดปลายทาง (Forward Path)

- Destination IP address
- Source IP address
- Broadcast_ID
- เวลาเพื่อยกเลิกเส้นทางกลับสู่โหนดต้นทาง
- เลขลำดับของโหนดต้นทาง

ในที่สุดข้อความ RREQ ไปถึงโหนด (อาจเป็นโหนดปลายทาง) ที่มีเส้นทางไปยังโหนดปลายทางก็จะทำการสร้างข้อความ RREP ส่งกลับเส้นทางเดิมไปยังโหนดต้นทาง โหนดที่รับข้อความนี้ก็จะปรับปรุงข้อมูลเส้นทางไปยังโหนดปลายทางในตารางข้อมูลเส้นทางของตัวเอง



รูปที่ 2.14 แสดงการค้นหาเส้นทางของโปรโตคอล AODV

การปรับปรุงข้อมูลเส้นทาง เมื่อโหนดตรวจพบว่าเส้นทางไปโหนดข้างเคียงไม่สามารถใช้งานได้ มันจะทำการกำจัดข้อมูลเส้นทางนั้นออกจากตารางและทำการส่งข้อความแจ้งการเกิดลิงค์ล้มเหลว ข้อความนั้นจะถูกส่งต่อไปยังโหนดข้างเคียงเพื่อบอกโหนดที่ยังใช้งานเส้นทางนี้ โดยใช้ข้อมูลในส่วนของรายชื่อโหนดข้างเคียงของเส้นทางนั้น ขั้นตอนเหล่านี้จะทำไปเรื่อยๆ จนกว่าจะไม่มีโหนดต้นทางที่ใช้เส้นทางนี้

คุณสมบัติของโปรโตคอลเปรียบเทียบโปรโตคอล AODV กับโปรโตคอลเวกเตอร์บอกระยะแบบเดิม AODV ลดการเกิดโอเวอร์เฮดของเน็ตเวิร์คได้มากกว่ามากเนื่องจากโปรโตคอลทำงานแบบรีแอคทีฟ และ AODVยังมีการทำงานใกล้เคียงกับโปรโตคอลแบบเดิมมากกว่า DSR ทำให้เมื่อมีการเชื่อมกันระหว่างเน็ตเวิร์คไร้สายแบบเฉพาะกิจกับเน็ตเวิร์คที่มีการเดินสายแบบดั้งเดิมเป็นไปได้ง่ายกว่า

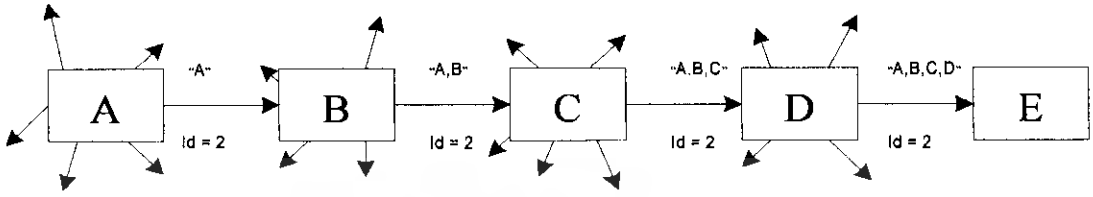
โปรโตคอล AODV สนับสนุนการใช้เพียงเส้นทางเดียวต่อแต่ละโหนดปลายทาง จึงมีแนวคิดที่จะปรับปรุงโปรโตคอล โดยใช้สามารถเกิดเส้นทางได้มากกว่าหนึ่งเส้นทางต่อหนึ่งโหนดปลายทาง

โปรโตคอล AODV ใช้ข้อความทักทายเป็นข้อความในระดับ IP ทำให้การทำงานของAODV ไม่จำเป็นต้องใช้การสนับสนุนจากระดับลิงค์เลเยอร์ (Link Layer) เพื่อให้โปรโตคอลสามารถทำงานได้ แต่ AODV ไม่สนับสนุนการทำงานเพื่อรองรับลิงค์แบบทิศทางเดียว เพราะในค้นหาเส้นทางของโปรโตคอลอยู่บนสมมติฐานของลิงค์แบบสองทิศทาง

2.8.3 โปรโตคอล Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) เป็นโปรโตคอลการจัดเส้นทางแบบหนึ่งในโปรโตคอลที่มีลักษณะเป็นรีแอคทีฟ เพื่อให้โหนดสามารถค้นหาเส้นทางไปยังโหนดปลายทางในเน็ตเวิร์คแบบพลวัต ซึ่งแต่ละแพคเกจมีเฮดเดอร์ (Header) บอกถึงข้อมูลรายชื่อโหนดต่างๆตลอดเส้นทางไปยังปลายทาง DSR ไม่ใช้การส่งข้อความเพื่อการทำนายเส้นทาง (ไม่มีการประกาศเส้นทางออกสู่เน็ตเวิร์ค) เป็นการลดการใช้แบนด์วิดธ์ของเน็ตเวิร์ค ประหยัดการใช้พลังงาน และหลีกเลี่ยงการปรับปรุงของข้อมูลเส้นทางในทั้งเน็ตเวิร์คซึ่งจะมีขนาดใหญ่ DSR ใช้การสนับสนุนจาก MAC Layer แทน (MAC Layer ควรมีการแจ้งให้ทราบเมื่อเกิดการล่มของลิงค์) รูปแบบการทำงานพื้นฐานของโปรโตคอลมีสองอย่างด้วยกันคือ การค้นหาเส้นทาง และปรับปรุงข้อมูลเส้นทาง

การค้นหาเส้นทางตัวอย่าง รูปที่ 2.15 แสดงตัวอย่างการค้นหาเส้นทาง โหนด A ต้องการค้นหาเส้นทางไปยังโหนด E ในการเริ่มต้นขบวนการค้นหา โหนด A ส่งข้อความ ROUTE REQUEST สู่โหนดข้างเคียงแต่ข้อความของ RREQ มีการระบุโหนดเริ่มต้น โหนดเป้าหมายและ Request ID ซึ่งขึ้นกับโหนดเริ่มต้น



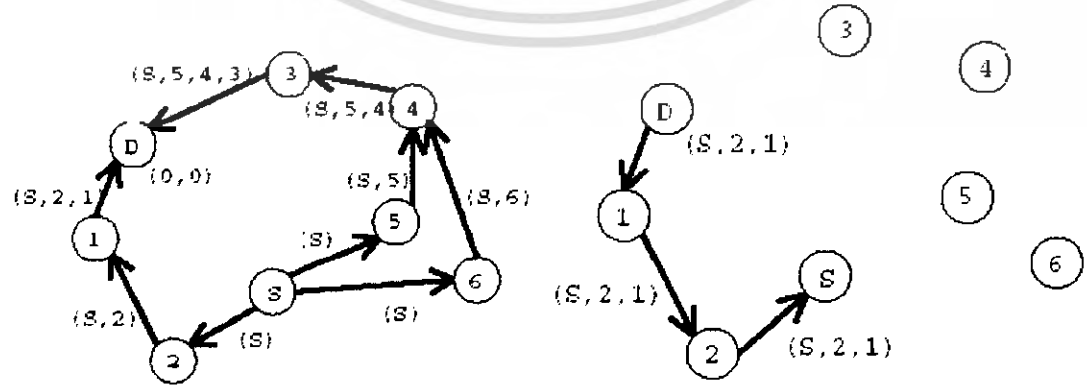
รูปที่ 2.15 แสดงการค้นหาเส้นทางอย่างง่ายของโปรโตคอล DSR

เมื่อโหนดอื่นได้รับ RREQ นี้แต่ไม่ใช่โหนดเป้าหมาย โหนดจะค้นหาข้อมูลเส้นทางไปยังโหนดเป้าหมายในหน่วยความจำข้อมูลเส้นทาง (Route Cache) ของโหนดก่อน ถ้าไม่มีข้อมูลเส้นทางนี้ก็จะทำการเพิ่มข้อมูลลำดับฮอปตัวเองลงในข้อมูลลำดับฮอปของข้อความ แล้วทำการบรอดแคสต์ข้อความนั้นอีกครั้ง แต่ถ้ามีข้อมูลเส้นทางนั้นโหนดส่งข้อความ ROUTE REPLY (RREP) กลับสู่โหนดที่เริ่มต้นทำการร้องขอเส้นทาง RREP มีการระบุลำดับ ฮอปของเส้นทางจากปลายทางไปยังจุดเริ่มต้น

ในขบวนการค้นหาเส้นทาง โหนดทำการส่ง RREQ ครั้งแรกด้วยกำหนดจำกัดการกระจายข้อมูลสูงสุดเป็นศูนย์ เพื่อป้องกันไม่ให้โหนดที่ได้รับข้อความทำการบรอดแคสต์ข้อความนั้นอีก ใช้เป็นกลไกเพื่อสอบถามข้อมูลเส้นทางในหน่วยความจำข้อมูลเส้นทางของโหนดข้างเคียง

จากรูปที่ 2.16 เมื่อโหนด S ต้องการส่งข้อมูลไปสู่โหนด D โหนด S ทำการสร้างเฮดเดอร์ของแพคเกจ ที่จะส่งไปยังโหนด D ด้วยข้อมูลเส้นทางทั้งหมดที่แพคเกจต้องผ่านไป โดยปกติ S หาข้อมูลเส้นทางในหน่วยความจำข้อมูลเส้นทาง (Route Cache) ของตัวเองก่อนซึ่งได้จากการเรียนรู้ก่อนหน้านี้ แต่ถ้าไม่มีข้อมูลเส้นทางนั้นมาก่อน S จะเริ่มขบวนการค้นหาเส้นทาง

โหนดสามารถทำงานในโหมดไม่เลือกหน้า (Promiscuous Mode) หมายถึงการปิดการทำงานของการกรองเลือกเฉพาะข้อมูลที่ส่งมายังโหนดเท่านั้น ทำให้โหนดสามารถรับทุกแพคเกจที่ผ่านเข้ามาโดยไม่สนใจ Address ในเฮดเดอร์ของแพคเกจ เพื่อนำข้อมูลเส้นทางในแพคเกจเหล่านั้นมาปรับปรุงข้อมูลเส้นทางในหน่วยความจำข้อมูลเส้นทาง



รูปที่ 2.16 แสดงการค้นหาเส้นทางของโปรโตคอล DSR

การส่ง RREP กลับสู่โหนดเริ่มต้น สามารถมีได้หลายวิธี อย่างง่ายที่สุดคือการกลับลำดับฮอปของข้อความ RREQ อย่างไรก็ตามวิธีนี้ใช้สมมติฐานว่าลิงค์ที่เกิดขึ้นเป็นลิงค์แบบสองทิศทาง(Bi-Directional Link) หรืออีกวิธีคือโหนดที่ส่ง RREP มีข้อมูลเส้นทางไปยังโหนดเริ่มต้นก็สามารถใช้ข้อมูลเส้นทางนั้นได้ทันที หรืออาจใช้วิธีอัดข้อความ RREP ลงในข้อความ RREQ ที่ส่งไปหาโหนดเริ่มต้น ด้วยวิธีนี้โหนดเริ่มต้นสามารถคำนวณได้ว่าเส้นทางยังโหนดเป้าหมายมีส่วนที่เป็นลิงค์แบบทิศทางเดียว เส้นทางที่ได้มานั้นจะถูกบันทึกในหน่วยความจำข้อมูลเส้นทาง (Route Cache)และข้อมูลเวลาที่ได้มาด้วยจะถูกนำมาใช้อีกครั้งในเฟสของการปรับปรุงข้อมูลเส้นทาง

การปรับปรุงข้อมูลเส้นทางเป็นกลไกที่เกิดเมื่อโหนด S ได้รับการแจ้งการหรือตรวจสอบ พบการล้มเหลวของลิงค์ใดลิงค์หนึ่ง ดังนั้นเส้นทางไปยังโหนด D จึงไม่สามารถใช้ได้อีกต่อไป ซึ่งอาจเกิดจากโฮสต์ที่อยู่ในรายชื่อของเส้นทาง เคลื่อนที่ออกนอกพื้นที่เดิมหรือออกไปจากเน็ตเวิร์คเมื่อมีการตรวจพบการล้มเหลวของลิงค์ในเส้นทางโหนดจะส่งข้อความแจ้งความผิดพลาดไปยังโหนดต้นทาง เมื่อได้รับข้อความแจ้งความผิดพลาดของลิงค์ใด โหนดค้นหาข้อมูลเส้นทางในหน่วยความจำข้อมูลเส้นทาง (Route Cache) ที่มีการใช้ลิงค์เหล่านั้นเพื่อลบข้อมูลเส้นทางนั้นออกโปรโตคอล DSR ใช้ข้อได้เปรียบของรูปแบบซอสเราติง โหนดข้างเคียงที่ใช้ในการส่งต่อแพคเกจไม่จำเป็นต้องทำการปรับปรุงข้อมูลเส้นทางเพื่อให้ได้เส้นทางที่จะทำการต่อแพคเกจนี้ออกไป และไม่จำเป็นต้องมีการประกาศข้อมูลเส้นทางเป็นช่วงๆ เป็นการประหยัดการใช้แบนด์วิดธ์ของเน็ตเวิร์ค รวมทั้งเป็นการประหยัดการใช้พลังงานของโหนดด้วยเช่นกัน

ข้อได้เปรียบอีกอย่างของโปรโตคอล คือการเรียนรู้เส้นทางสามารถทำได้ด้วยการอ่านข้อมูลในแต่ละแพคเกจที่ส่งผ่านมา การเรียนรู้เส้นทางแบบนี้ทำให้ลดโอเวอร์เฮดของเน็ตเวิร์คลงได้อย่างไรก็ตามเมื่อแพคเกจต้องใช้เส้นทางที่มีฮอปมากขึ้น โอเวอร์เฮดของแพคเกจก็จะเพิ่มขึ้นเพราะแพคเกจจะมีส่วนของข้อมูลเส้นทางรวมอยู่ด้วย โอเวอร์เฮดเพิ่มขึ้นทำให้แพคเกจมีขนาดใหญ่ขึ้นตามไปด้วย

การอ่านข้อมูลของทุกแพคเกจที่ผ่านเข้ามาก็เป็นปัญหาสำคัญในด้านความปลอดภัย การปิดระบบการกรองข้อมูลของส่วนเชื่อมต่อเพื่อให้สามารถอ่านข้อมูลทุกแพคเกจได้ อาจนำไปสู่การเข้าสู่ข้อมูลสำคัญใน แพคเกจนั้น อาจเป็นพาสเวิร์ด หรือ ตัวเลขบัตรเครดิต ดังนั้นส่วนของแอปปลิเคชันอาจต้องทำการเข้ารหัสข้อมูลก่อนทำการส่งข้อมูลเหล่านั้น

นอกจากนี้โปรโตคอลชนิดนี้ยังสนับสนุนการทำงานเมื่อเกิดลิงค์แบบทิศทางเดียวในเน็ตเวิร์คได้ด้วย

2.8.4 โปรโตคอล Temporally Ordered Routing Algorithm (TORA)

Temporally Ordered Routing Algorithm (TORA) เป็นโปโตคอลที่สามารถจัดเส้นทางได้หลายเส้นทางระหว่างต้นทางกับปลายทาง TORA ประกอบด้วย 3 ส่วน คือ การสร้างเส้นทาง การจัดการเส้นทางและการลบเส้นทาง ที่แต่ละโหนดจะสำเนาเส้นทาง ของแต่ละปลายทางแยกกัน โหนดแต่ละโหนดในเน็ตเวิร์คมีความสัมพันธ์กันสูง ข้อความความคุมจากโหนดที่มีความสำคัญสูงไปสู่โหนดที่มีความสำคัญต่ำกว่า การหาเส้นทางจะใช้ แพคเกจสอบถาม (QRY) และแพคเกจอัพเดท (UPD)

เมื่อโหนดไม่มีเส้นทางไปยังปลายทางมันจะกระจายแพคเกจ QRY ออกไปจนสามารถหาโหนดกับเส้นทางที่ปลายทาง โหนดนั้นจะตอบกับด้วยการกระจาย UPD แพคเกจ ประกอบด้วยโหนดที่สำคัญเมื่อโหนดรับ UPD แพคเกจก็จะได้จำนวนเส้นทางจากต้นทางไปยังปลายทาง

ถ้าโหนดพบปัญหาไม่สามารถไปถึงปลายทางได้ ก็ตั้งค่าการติดต่อไว้สูง ในกรณีนี้โหนดจะพยายามสร้างเส้นทางใหม่ โดยจะกระจาย CLR (Clear message) ตั้งค่าสถานะเส้นทางทั้งหมดและลบเส้นทางที่ใช้งานไม่ได้ออก

TORA ทำงานชั้นบนของ IMEP (Internet MANNET Encapsulation Protocol) จัดความน่าเชื่อถือของการส่งข้อความเส้นทางและแจ้งการเปลี่ยนแปลงโปรโตคอลของจัดเส้นทางของโหนดใกล้เคียง IMEP พยายามสรุป ข้อความ IMEPและTORA ที่จะลดส่วนหัวของแพคเกจ

บทที่ 3
วิธีการดำเนินการทดลอง

การจำลอง (Simulation) คือการจำลองหรือเลียนแบบเหตุการณ์ต่างๆ ซึ่งบางครั้งเหตุการณ์นั้นไม่สามารถที่จะทดลองด้วยของจริงได้ การจำลองเหตุการณ์สามารถทำได้หลายๆแบบ เช่น การใช้สมการคณิตศาสตร์ การใช้คอมพิวเตอร์ เป็นต้น การใช้คอมพิวเตอร์ในการจำลองเหตุการณ์นั้นสามารถที่จะทำได้หลายเหตุการณ์

3.1 รูปแบบการจำลอง

งานวิเคราะห์นี้ได้จำลองการทำงานของเครือข่ายไร้สายโดยมี สภาวะแวดล้อมซึ่งมีพารามิเตอร์ต่างๆ โดยในการทำงานสภาพแวดล้อมนี้เพื่อศึกษาปัญหาต่างๆมีผลต่อประสิทธิภาพของโปรโตคอลเลือกเส้นทาง

การจำลองการทำงานในสภาพแวดล้อม

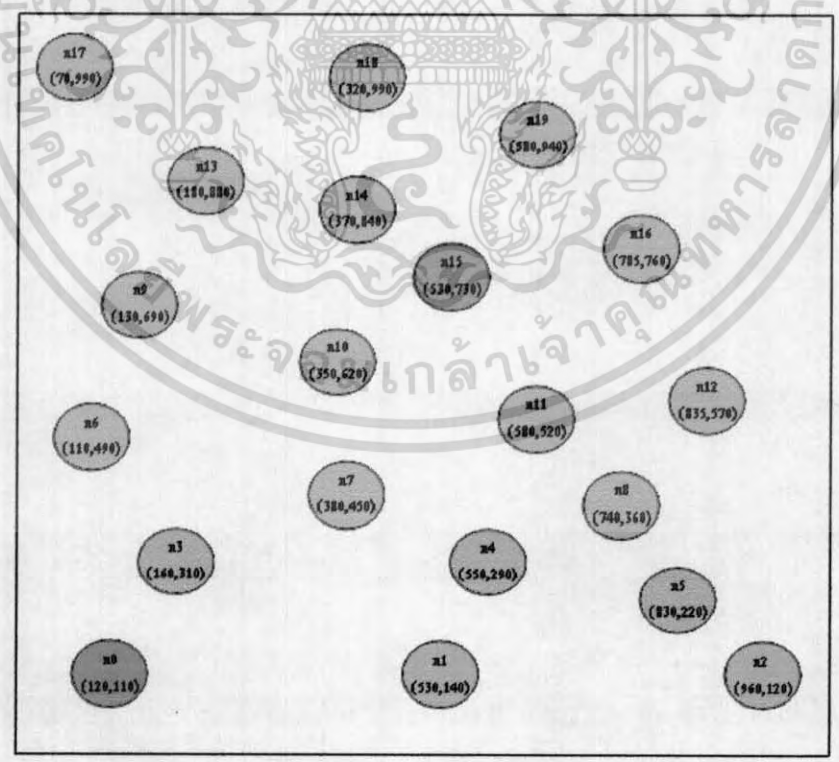
พื้นที่จำลองการทำงาน 1000เมตร x 1000เมตร

จำนวนโมไบล์โฮสต์ 20

จำนวนคู่การสื่อสาร 1, 4, 9

เวลาจำลองการทำงาน 240 วินาที

รูปแบบการกระจาย



รูปที่ 3.1 ลักษณะรูปแบบการจำลอง

3.2 การจำลองโดยไม่มีการเคลื่อนที่ของโหนด

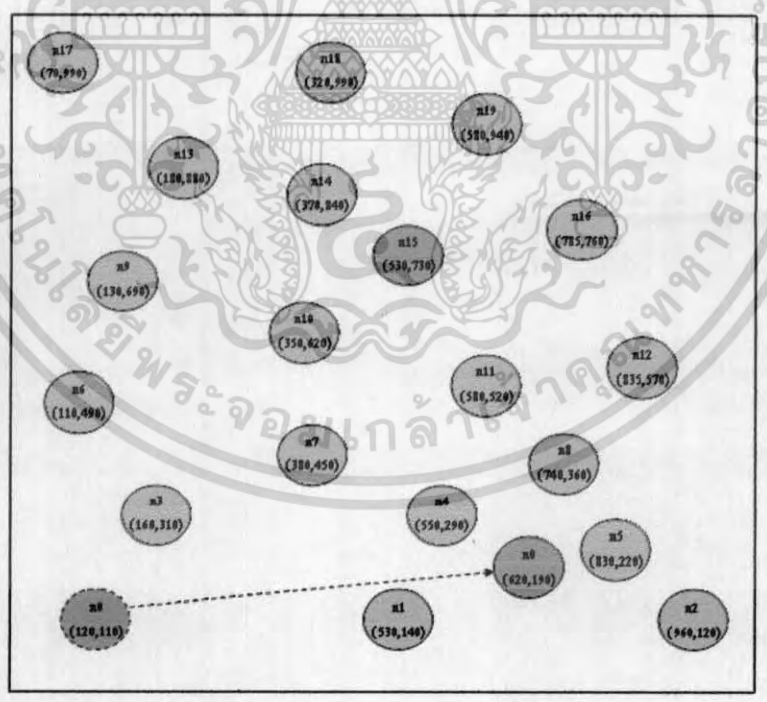
จากลักษณะรูปแบบการจำลองดังรูปที่ 3.1 จะไม่กำหนดให้มีการเคลื่อนที่ของโหนดใดเลยและคู่การสื่อสาร 1 คู่กำหนดให้โหนด n0 โอนถ่ายข้อมูลแบบเอฟทีพีกับโหนด n15 การสื่อสาร 4 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (10,16) (11,19) และการสื่อสาร 9 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (2,9) (5,17) (12,13) (10,16) (7,18) (11,19) (8,14)

เพื่อศึกษาผลของประสิทธิภาพของโปรโตคอลการจัดเส้นทางระหว่าง โหนด n0 กับ โหนด n15

3.3 การจำลองโดยเคลื่อนที่โหนดต้นทาง

จากลักษณะรูปแบบการจำลองดังรูปที่ 3.2 จะกำหนดให้มีการเคลื่อนที่ของโหนดต้นทางหรือโหนด n0 จากตำแหน่ง (120,110)ไปยังตำแหน่ง(620,190) โดยกำหนดความเร็วการเคลื่อนที่ของโหนด n0 เท่ากับ 5, 15, 30 เมตรต่อวินาที และคู่การสื่อสาร 1 คู่กำหนดให้โหนด n0 โอนถ่ายข้อมูลแบบเอฟทีพีกับโหนด n15 การสื่อสาร 4 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (10,16) (11,19) และการสื่อสาร 9 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (2,9) (5,17) (12,13) (10,16) (7,18) (11,19) (8,14)

เพื่อศึกษาผลของความเร็วและลักษณะรูปแบบเครือข่ายที่เปลี่ยนแปลงมีผลต่อของประสิทธิภาพของโปรโตคอลเลือกเส้นทางระหว่าง โหนด n0 กับ โหนด n15

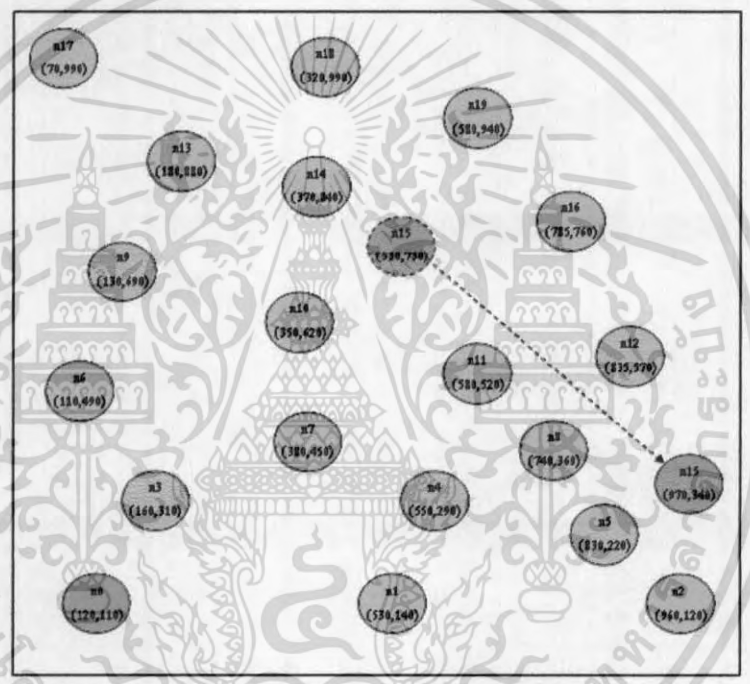


รูปที่ 3.2 แบบจำลองการเคลื่อนที่ของโหนดต้นทางหรือโหนด n0

3.4 การจำลองโดยเคลื่อนที่โหนดปลายทาง

จากลักษณะรูปแบบการจำลองดังรูปที่3.3 จะกำหนดให้มีการเคลื่อนที่ของโหนดปลายทางหรือโหนด n15 จากตำแหน่ง (530,730)ไปยังตำแหน่ง(970,340) โดยกำหนดความเร็วการเคลื่อนที่ของโหนด n15 เท่ากับ 5, 15, 30 เมตรต่อวินาที และคู่การสื่อสาร 1 คู่กำหนดให้โหนด n0 โอนถ่ายข้อมูลแบบเอฟทีพีกับโหนด n15 การสื่อสาร 4 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (10,16) (11,19) และการสื่อสาร 9 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (2,9) (5,17) (12,13) (10,16) (7,18) (11,19) (8,14)

เพื่อศึกษาผลของความเร็วและลักษณะรูปแบบเครือข่ายที่เปลี่ยนแปลงมีผลต่อของประสิทธิภาพของโปรโตคอลเลือกเส้นทางระหว่าง โหนด n0 กับ โหนด n15

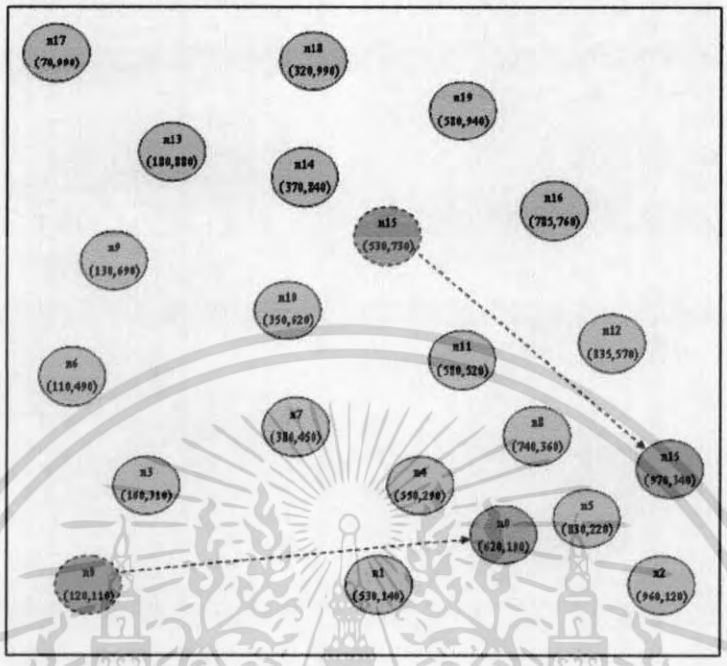


รูปที่ 3.3 แบบจำลองการเคลื่อนที่ของโหนดปลายทางหรือโหนด n15

3.5 การจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน

จากลักษณะรูปแบบการจำลองดังรูปที่ 3.4 จะกำหนดให้มีการเคลื่อนที่ของโหนดต้นทางหรือโหนด n0 จากตำแหน่ง (120,110)ไปยังตำแหน่ง(620,190) และโหนดปลายทางหรือโหนด n15 จากตำแหน่ง (530,730)ไปยังตำแหน่ง(970,340) โดยกำหนดความเร็วการเคลื่อนที่ของโหนด n0กับn15 เท่ากับ 5, 15, 30 เมตรต่อวินาที และคู่การสื่อสาร 1 คู่กำหนดให้โหนด n0 โอนถ่ายข้อมูลแบบเอฟทีพีกับโหนด n15 การสื่อสาร 4 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (10,16) (11,19) และการสื่อสาร 9 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (2,9) (5,17) (12,13) (10,16) (7,18) (11,19) (8,14)

เพื่อศึกษาผลของความเร็วและลักษณะรูปแบบเครือข่ายที่เปลี่ยนแปลงมีผลต่อของประสิทธิภาพของโปรโตคอลเลือกเส้นทางระหว่าง โหนด n0 กับ โหนด n15

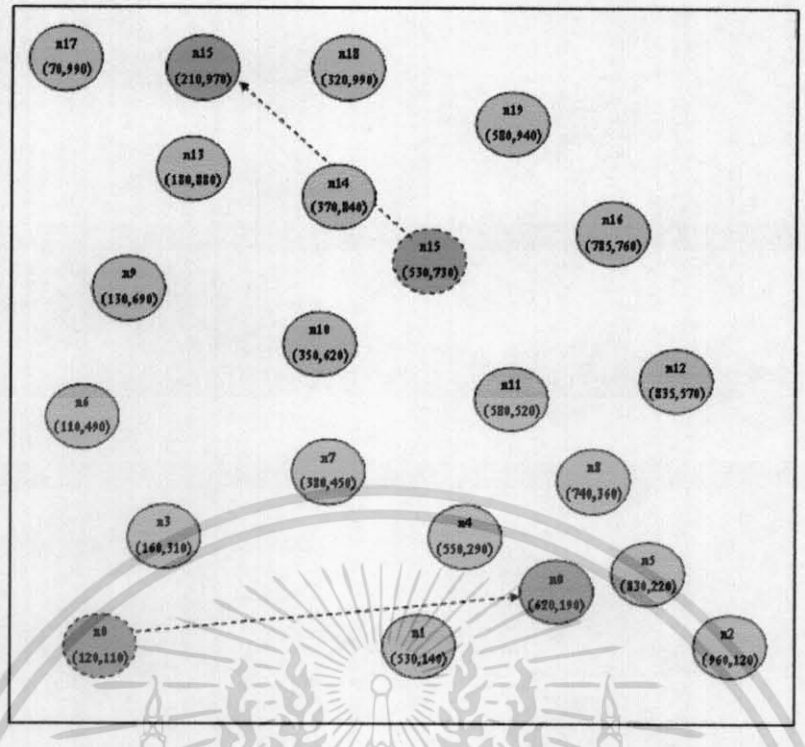


รูปที่ 3.4 แบบจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน

3.6 การจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางออกจากกัน

จากลักษณะรูปแบบการจำลองดังรูปที่ 3.5 จะกำหนดให้มีการเคลื่อนที่ของโหนดต้นทางหรือโหนด n0 จากตำแหน่ง (120,110)ไปยังตำแหน่ง(620,190)และโหนดปลายทางหรือโหนด n15 จากตำแหน่ง (530,730)ไปยังตำแหน่ง(210,970) โดยกำหนดความเร็วการเคลื่อนที่ของโหนด n0กับn15 เท่ากับ 5, 15, 30 เมตรต่อวินาที และคู่การสื่อสาร 1 คู่กำหนดให้โหนด n0 โอนถ่ายข้อมูลแบบเอฟทีพีกับโหนด n15 การสื่อสาร 4 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (10,16) (11,19) และการสื่อสาร 9 คู่ กำหนดให้คู่โหนดการสื่อสารมีดังนี้ (0,15) (1,6) (2,9) (5,17) (12,13) (10,16) (7,18) (11,19) (8,14)

เพื่อศึกษาผลของความเร็วและลักษณะรูปแบบเครือข่ายที่เปลี่ยนแปลงมีผลต่อของประสิทธิภาพของโปรโตคอลเลือกเส้นทางระหว่าง โหนด n0 กับ โหนด n15



รูปที่ 3.5 แบบจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางออกจากกัน

บทที่ 4
ผลการทดลอง

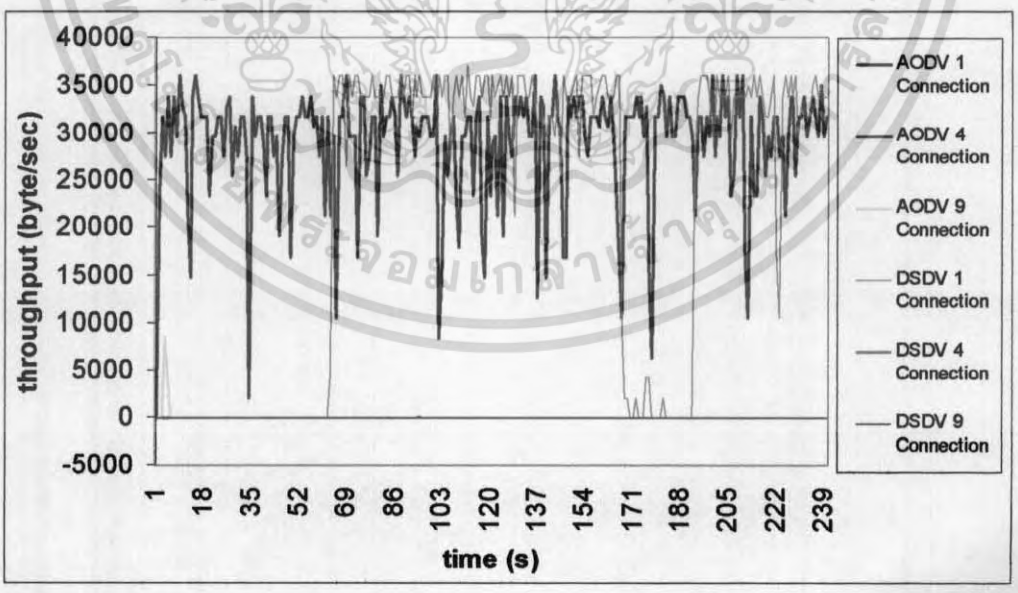
ในปริญญานิพนธ์นี้จะวิเคราะห์ผลการทดลองที่ได้จากการดำเนินการทดลอง เป็นการวิเคราะห์จากการวัดค่าประสิทธิภาพของเน็ตเวิร์คไร้สาย ด้วยค่าพารามิเตอร์ที่สนใจซึ่งมี ค่าคอนโทรลโอเวอร์เฮด และค่าประสิทธิภาพในการสื่อสารข้อมูล ในการทดลองจะค่า Periods Time เท่ากับ 1 วินาที

$$\text{Average control overhead} = \frac{\sum \text{routed packets size in the network (RTR)}}{\text{Periods Time}}$$

$$\text{Average Data Throughput} = \frac{\sum \text{size of all data packets}}{\text{Periods Time}}$$

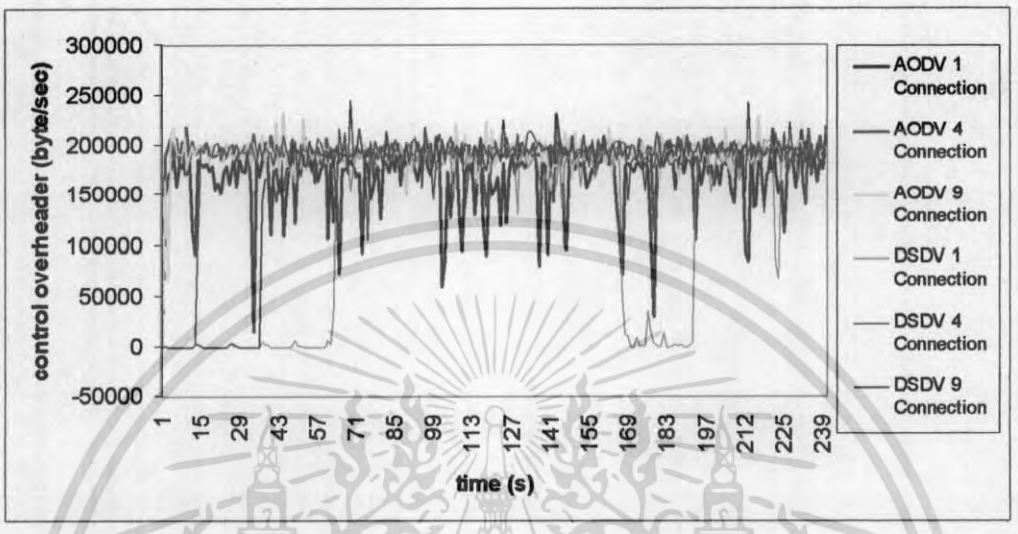
4.1 ผลการจำลองโดยไม่มีการเคลื่อนที่ของโหนด

รูปที่ 4.1 เปรียบเทียบค่าทรูพุตของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV โปรโตคอล AODV จะให้ค่าทรูพุตมากกว่าโปรโตคอล DSDV เนื่องจากโปรโตคอล AODV จะค้นหาเส้นทางทันทีที่มีการร้องขอ แต่จะไม่ต้องรอข้อมูลเส้นทางข้างเคียงทุกโหนดจนครบอย่างโปรโตคอล DSDV ที่เกิดการหน่วงในช่วงการบรอดแคสต์ปรับข้อมูลเส้นทาง ถ้าคู่การสื่อสารเพิ่มขึ้นจะทำให้สื่อสารไม่ได้



รูปที่ 4.1 ค่าทรูพุตการจำลองโดยไม่มีการเคลื่อนที่ของโหนด

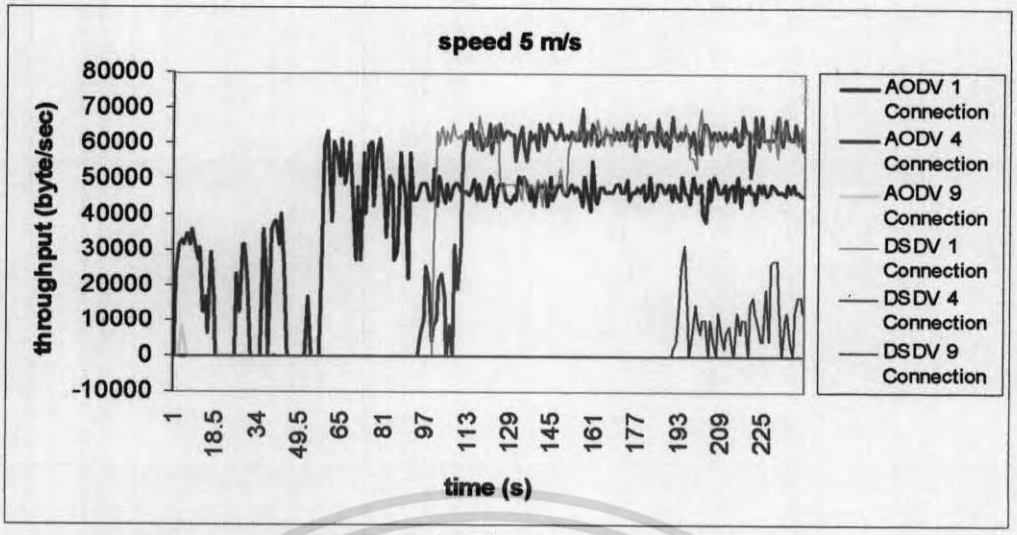
รูปที่ 4.2 เปรียบเทียบค่าโอเวอร์เฮดที่ใช้ในการค้นหาเส้นทางของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV โปรโตคอล AODV จะมีค่าโอเวอร์เฮดมากกว่าโปรโตคอล DSDV เนื่องจากโปรโตคอล AODV มีการใช้ข้อความทักทายเพื่อประกาศความคงอยู่ของโหนดข้างเคียงและจากเป็นแบบซอสเราติง



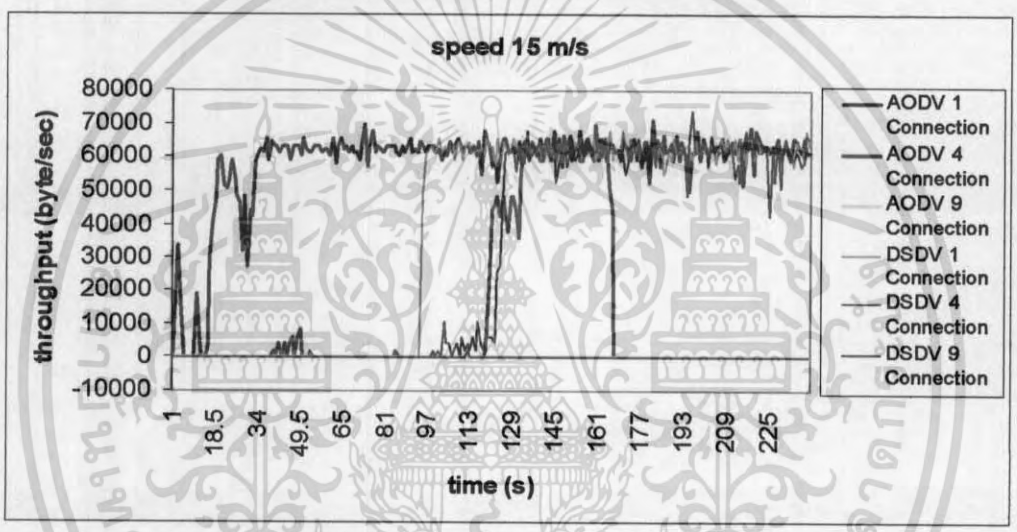
รูปที่ 4.2 ค่าโอเวอร์เฮดการจำลองโดยไม่มีการเคลื่อนที่ของโหนด

4.2 ผลการจำลองโดยเคลื่อนที่โหนดต้นทาง

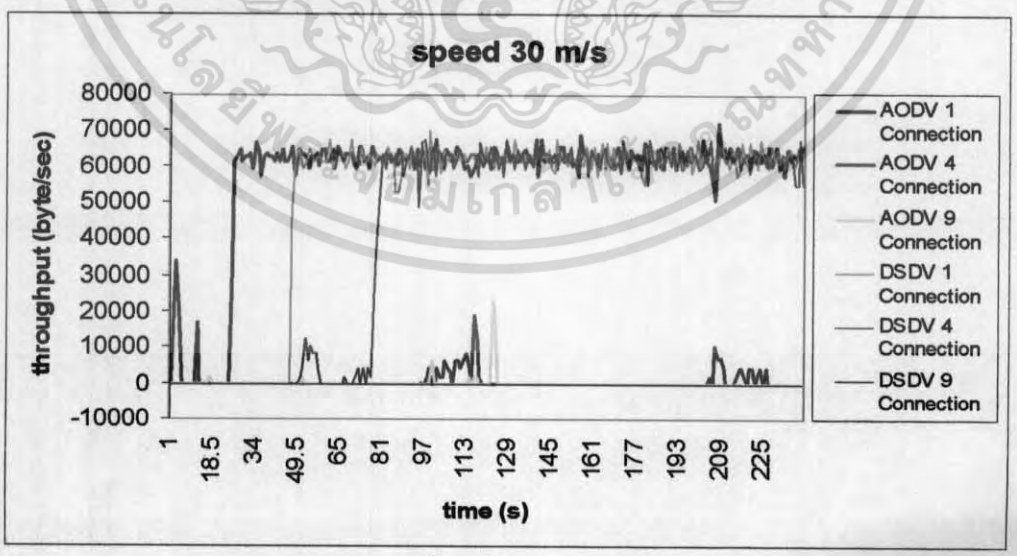
รูปที่ 4.3 เปรียบเทียบค่าทรูพุตของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV โปรโตคอล AODV จะให้ค่าทรูพุตมากกว่าโปรโตคอล DSDV เนื่องจากไม่มีการหน่วงที่ต้องรอการค้นหาเส้นทางที่จะต้องรอการตอบกลับของโหนดข้างเคียงจนครบอย่างเช่นโปรโตคอล DSDV แต่โปรโตคอล AODV ก็อาจไม่สามารถเลือกเส้นทางที่ดีที่สุดได้ ถ้าเส้นทางเก่ายังคงใช้ได้อยู่จะไม่มีการค้นหาเส้นทางใหม่ แต่ที่ความเร็วต่างๆก็อาจจะทำให้โปรโตคอล AODV มีโอกาสเลือกเส้นทางที่ดีที่สุดได้ดังในรูปที่ 4.3(ก) จะเห็นว่าที่ AODV 1 Connention จะมีค่า ทรูพุตน้อยกว่า รูปที่ 4.3(ข) และรูปที่4.3(ค) แต่ถ้าเป็นโปรโตคอล DSDV จะเลือกเส้นทางที่ดีที่สุดเพราะมีการปรับปรุงข้อมูลเส้นทางอยู่เป็นช่วงๆดังรูปที่ 4.3 (ก) (ข)และ(ค) จะเห็นว่าที่ DSDV 1 Connection จะมีค่าทรูพุตใกล้เคียงกัน ที่คู่การสื่อสารเพิ่มขึ้นถ้ามีการร้องขอเส้นทางสำเร็จก็สามารถสื่อสาร แต่ถ้าไม่สามารถร้องขอเส้นทางสำเร็จก็สื่อสารไม่ได้จะเห็นได้ชัดทั้งโปรโตคอล AODV และ โปรโตคอล DSDV ในรูปที่ 4.3 (ก) (ข)และ(ค)



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



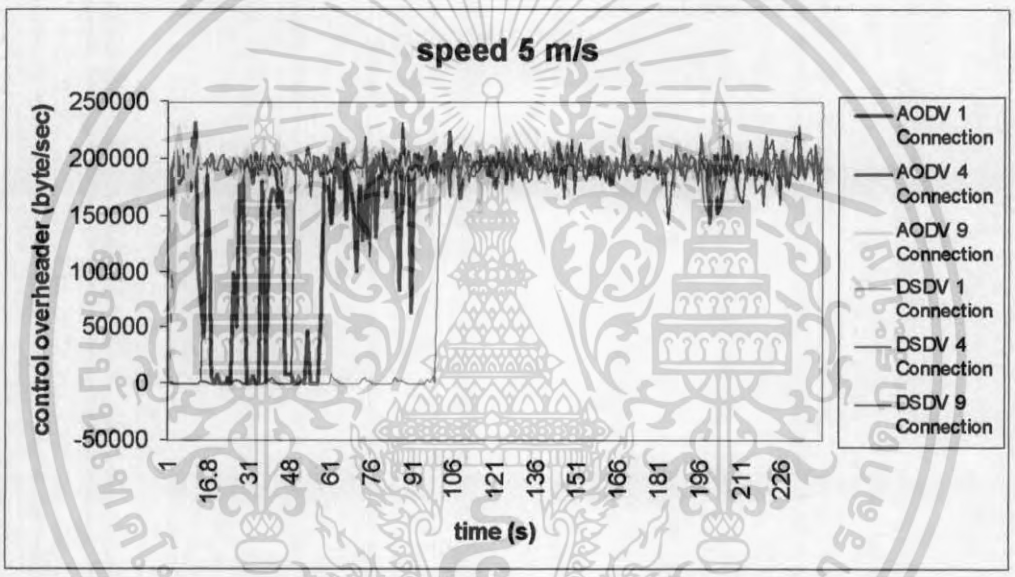
(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



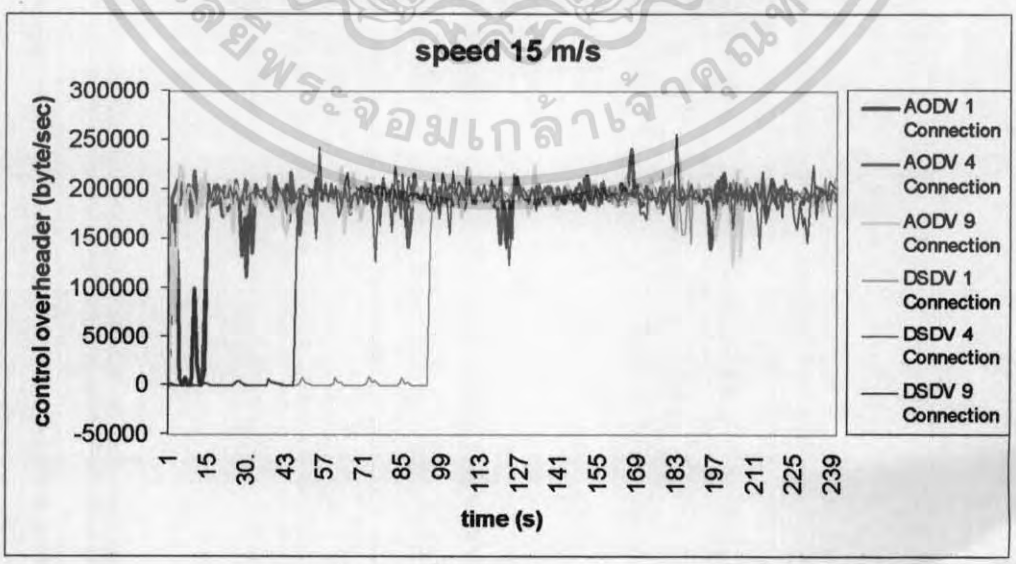
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.3 ค่าทรูพุตการจำลองโดยเคลื่อนที่โหนดต้นทาง

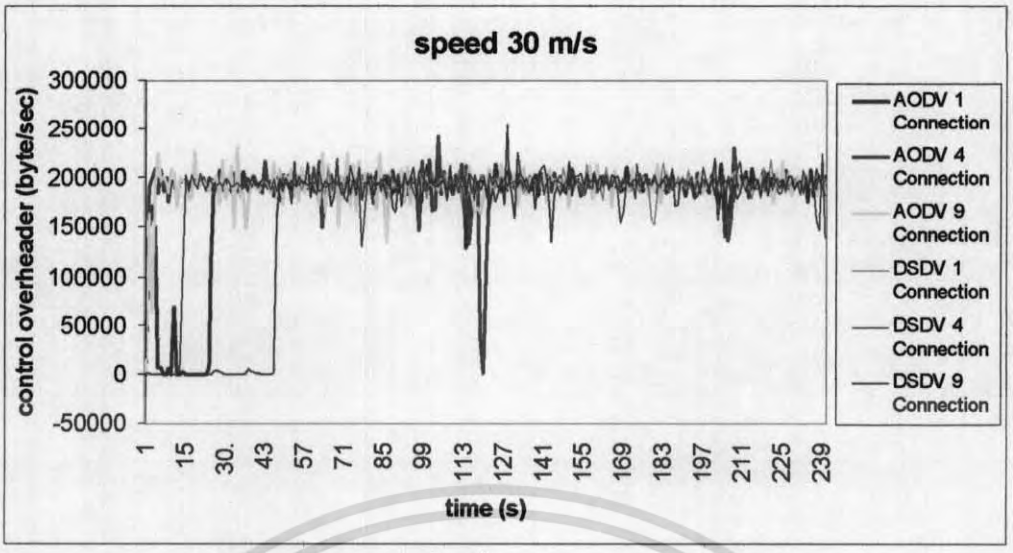
รูปที่ 4.4 เปรียบเทียบค่าโอเวอร์เฮดที่ใช้ในการค้นหาเส้นทางของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV ในการจำลองของโหนดต้นทางมีการเคลื่อนที่จะเห็นว่าค่าโอเวอร์เฮดของโปรโตคอล AODV มากกว่าโปรโตคอล DSDV ในรูปที่ 4.4 (ก) (ข)และ(ค)ช่วงที่เกิดการหน่วงโอเวอร์เฮดของโปรโตคอล DSDV มีน้อยก็จะทำให้โอเวอร์เฮดของโปรโตคอล AODV มากกว่า แต่ในช่วงเวลาที่โปรโตคอล DSDV มีการถ่ายโอนข้อมูลจากการบรอดแคสต์เป็นช่วงๆเพื่อปรับปรุงการเลือกเส้นทางและจะสังเกตเห็นได้ว่าที่รูปที่ 4.3 (ก) ความเร็วช้าๆจะมีการเปลี่ยนแปลงลักษณะเครือข่ายบ่อยจะทำให้เกิดการหน่วงเพื่อปรับปรุงข้อมูลเส้นมากกว่าช่วงเวลาที่รูปที่ 4.3 (ข)และ(ค) ความเร็วเพิ่มขึ้นเพราะที่เวลาความเร็วเพิ่มขึ้นจะทำให้โหนดที่เคลื่อนที่ไปถึงเส้นทางได้เร็วกว่าการเคลื่อนที่ของโหนดที่เคลื่อนที่ช้ากว่าแต่การจำลองในสภาวะแบบนี้โปรโตคอล AODV ก็มีอัลกอทึมในการประกาศความคงอยู่ของโหนดข้างเคียงแต่ก็จะทำให้มีโอเวอร์เฮดไม่มากเท่ากับการบรอดแคสต์เป็นช่วงๆของโปรโตคอล DSDV



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที

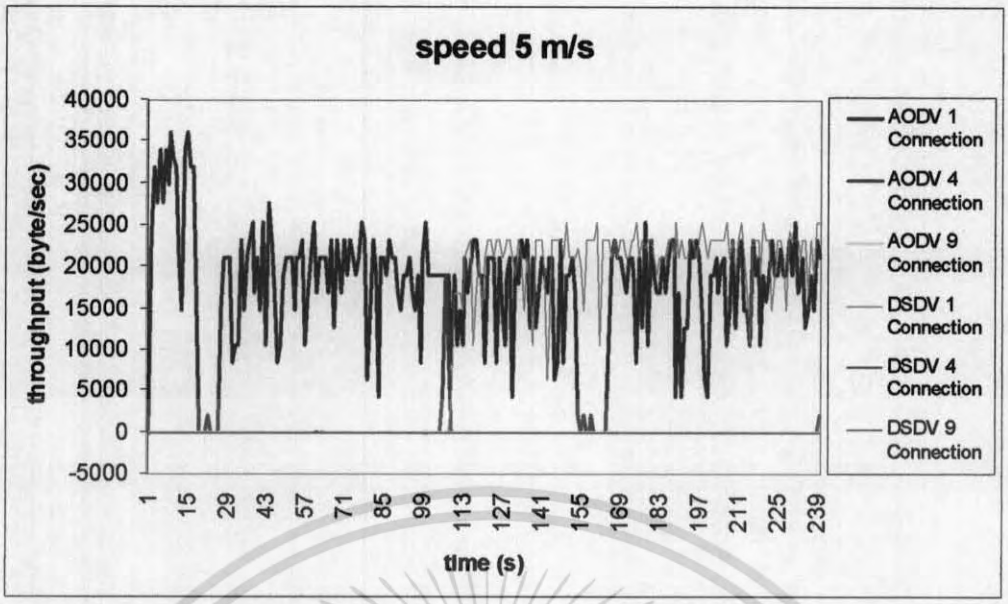


(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

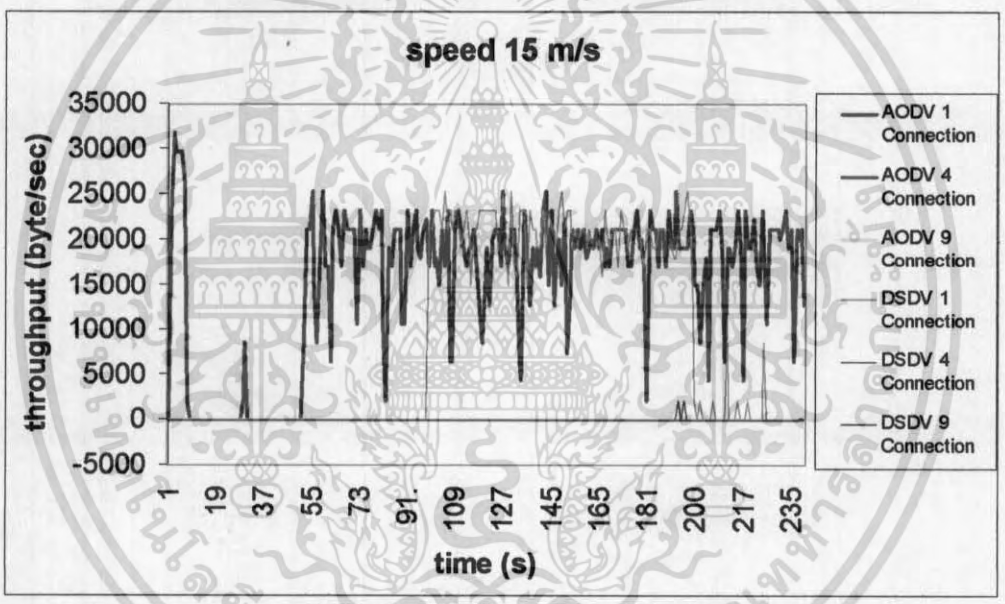
รูปที่ 4.4 ค่าโอเวอร์เฮดการจำลองโดยเคลื่อนที่โหนดต้นทาง

4.3 ผลการจำลองโดยเคลื่อนที่โหนดปลายทาง

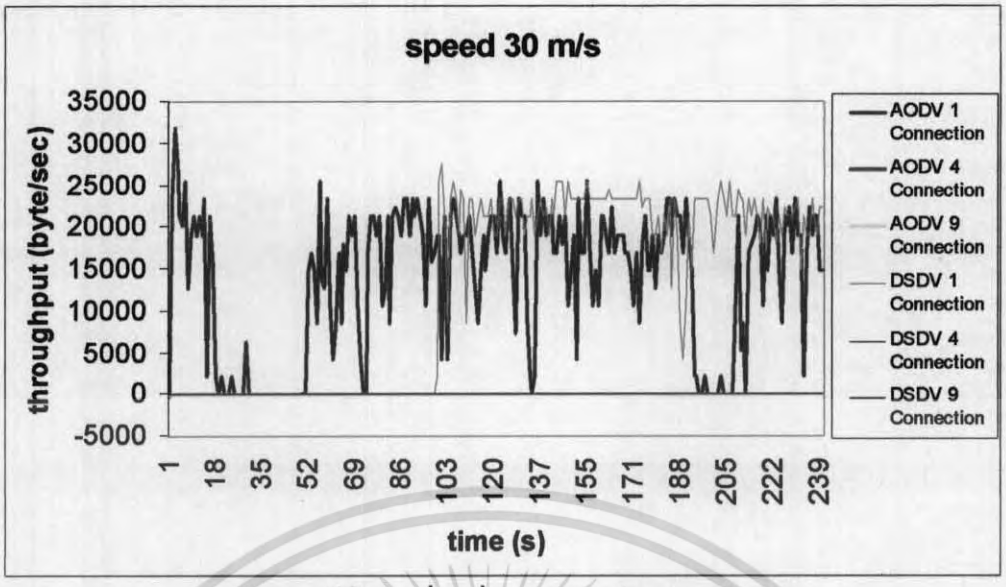
รูปที่ 4.5 เปรียบเทียบค่าทรูพุตของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และ DSDV โปรโตคอล AODV จะให้ค่าทรูพุตมากกว่าโปรโตคอล DSDV เนื่องจากไม่มีการหน่วงช่วงเริ่มต้นทำมีการส่งข้อมูลทันทีที่ต้องการได้ทัน ที่ช่วงเริ่มต้นจะสังเกตเห็นได้ว่าเมื่อโหนดเคลื่อนที่แต่ก็จะมีการถ่ายโอนข้อมูลที่ให้ค่าทรูพุตมากเนื่องการเคลื่อนที่ช้าดังรูปที่ 4.5 (ก) จนทำให้เส้นทางที่อยู่ใกล้สามารถใช้ติดต่อได้นานทำให้ มีค่าทรูพุตสูง แต่เมื่อความเร็วการเคลื่อนที่ของโหนดเพิ่มขึ้นดังรูปที่ 4.5 (ข)และ(ค) ทำให้เส้นที่อยู่ใกล้ใช้ติดต่อสื่อสารไม่ได้จึงทำการฟลัดดิงหาเส้นทางใหม่จึงเริ่มถ่ายโอนข้อมูลใหม่ และรูปที่ 4.5 (ก) โปรโตคอล DSDV ก็จะมีการรอการปรับปรุงเส้นทางข้อมูลในช่วงเริ่มต้นและจะมีการบรอดแคสต์ปรับปรุงข้อมูลเป็นช่วงๆแต่เมื่อความเร็วเพิ่มดังรูปที่ 4.5 (ข)และ(ค) ขึ้นทำให้โหนดเคลื่อนที่ไปถึงได้เร็วขึ้นจึงมีลักษณะการเปลี่ยนแปลงของเครือข่ายเร็วขึ้นค่าทรูพุตของโปรโตคอล DSDV มีค่าเพิ่มขึ้นในขณะที่ความเร็วเพิ่มขึ้น ในขณะที่คู่การสื่อสารเพิ่มขึ้นถ้ามีการร้องขอเส้นทางสำเร็จก็สามารถสื่อสาร แต่ถ้าไม่สามารถร้องขอเส้นทางสำเร็จก็สื่อสารไม่ได้



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



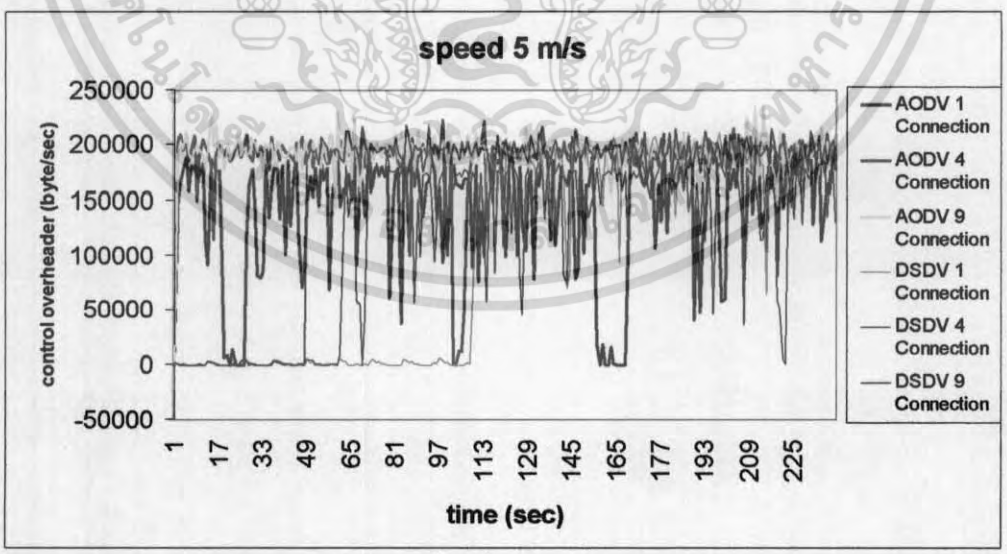
(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



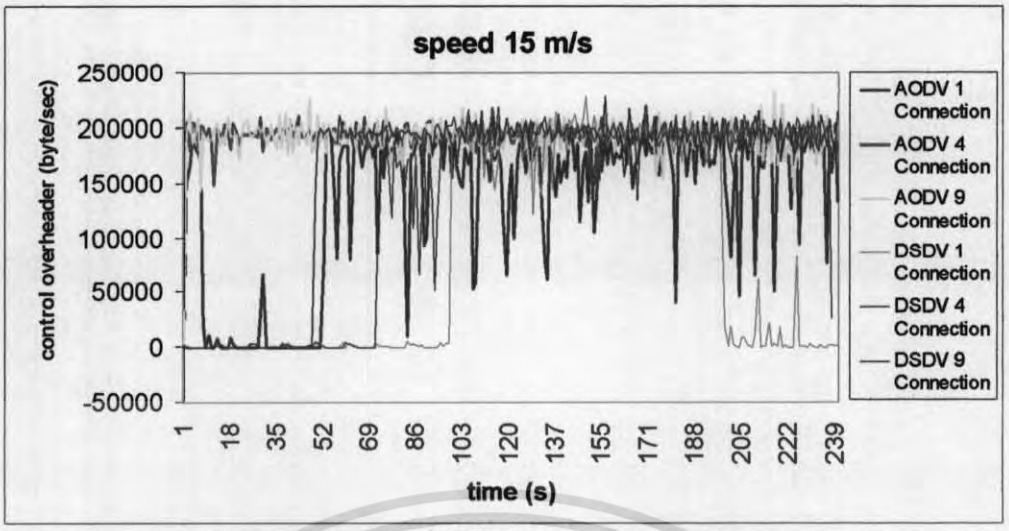
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.5 ค่าทรูพุตการจำลองโดยเคลื่อนที่โหนดปลายทาง

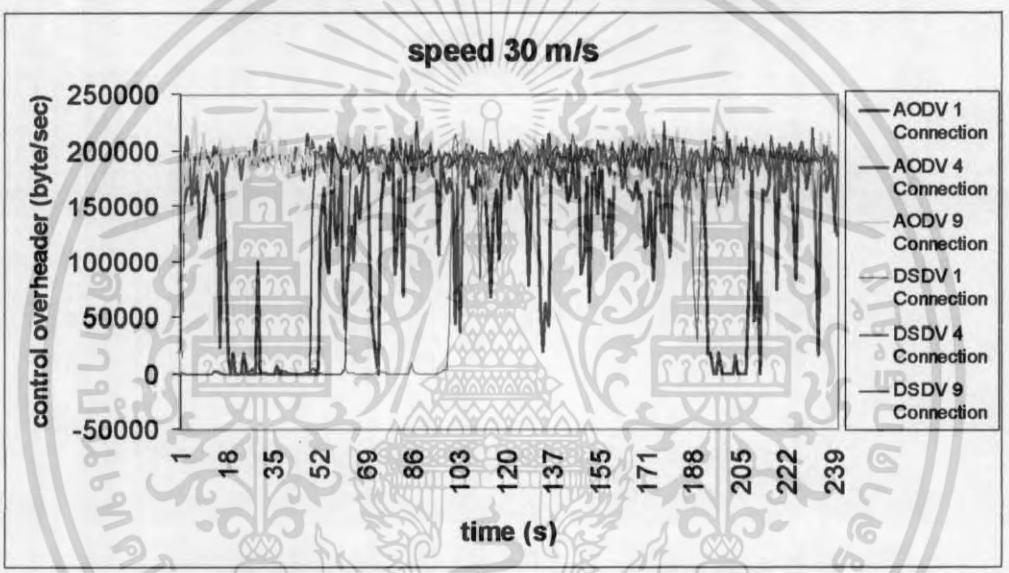
รูปที่ 4.6 (ก) (ข)และ(ค) เปรียบเทียบค่าโอเวอร์เฮดที่ใช้ในการค้นหาเส้นทางของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และ DSDV ค่าโอเวอร์เฮดของโปรโตคอล AODV มากกว่าโปรโตคอล DSDV และเมื่อคู่การสื่อสารเพิ่มขึ้นก็จะทำให้ค่าโอเวอร์เฮดสูงตามไปด้วยเนื่องจากการเก็บค่าโอเวอร์เฮดของโปรโตคอล AODV จะมีการเก็บข้อมูลเส้นทางตลอดเส้นทางและเมื่อคู่การสื่อสารเพิ่มขึ้นก็ทำให้การหาเส้นทางต้อง พยายามส่งการร้องขอเส้นทางมากขึ้น ส่วนโปรโตคอล DSDV ก็จะมีการปรับปรุงแบบส่วนเพิ่มเติมของเส้นทางได้



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



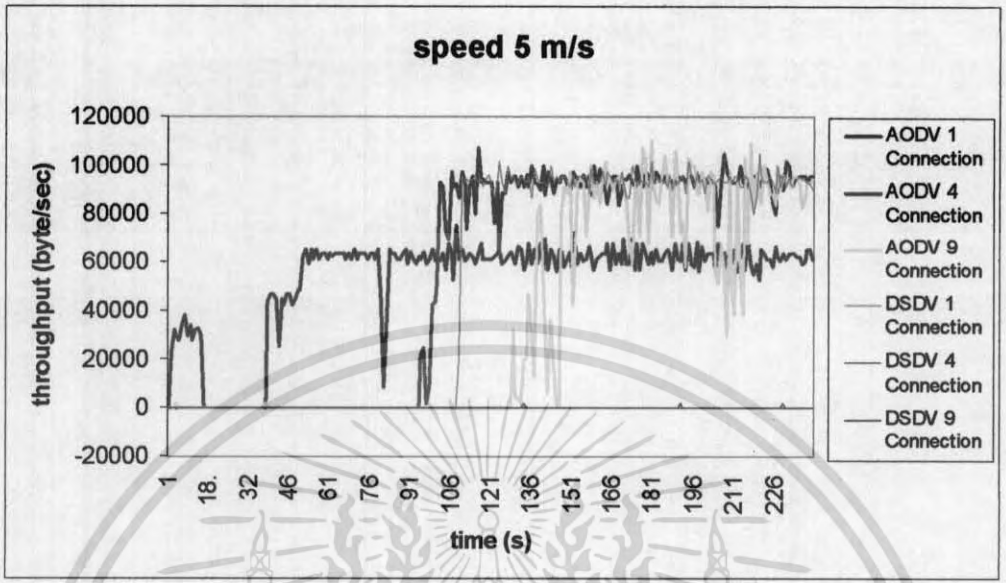
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.6 ค่าโอเวอร์เฮดการจำลองโดยเคลื่อนที่โหนดปลายทาง

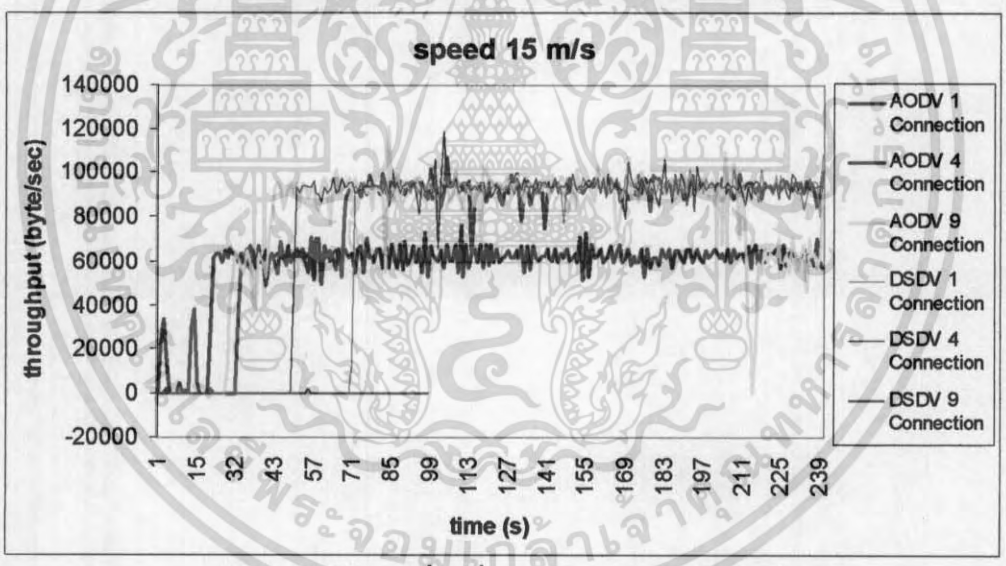
4.4 ผลการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน

รูปที่ 4.7 เปรียบเทียบค่าทรูพุตของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV จากการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน โปรโตคอล AODV จะให้ค่าทรูพุตมากกว่าโปรโตคอล DSDV เนื่องจากไม่มีการหน่วงช่วงเริ่มต้นดังรูปที่ 4.7 (ก) (ข)และ(ค) แต่การเลือกเส้นทางของโปรโตคอล AODV จะไม่ได้คำนึงถึงการเลือกเส้นทางที่ดีที่สุด แต่โปรโตคอล DSDV จะทำการบรอดแคสต์เป็นช่วงๆจึงทำให้มีการปรับปรุงและเลือกเส้นทางที่ดีที่สุดหลังทำการบรอดแคสต์ แต่ที่ความเร็วเปลี่ยนและคู่การสื่อสารเพิ่มขึ้นอาจทำให้ โปรโตคอล AODV มีค่าทรูพุตดีเท่ากับโปรโตคอล DSDV เนื่องจากการที่โหนดเคลื่อนที่เปลี่ยนทำให้เส้นทางเดิมที่ใช้ในการติดต่อไม่สามารถใช้งานได้จึงทำให้โปรโตคอล AODV ทำการฟลัดดิงหาเส้นทางใหม่ในช่วงเวลาที่ทำให้สามารถเลือกเส้นที่ดีที่สุดได้ และการร้องขอเส้นทางสำเร็จในบางช่วงเวลาที่ทำให้สามารถเลือกเส้นที่ดีที่สุดได้ ในขณะที่คู่การสื่อสาร

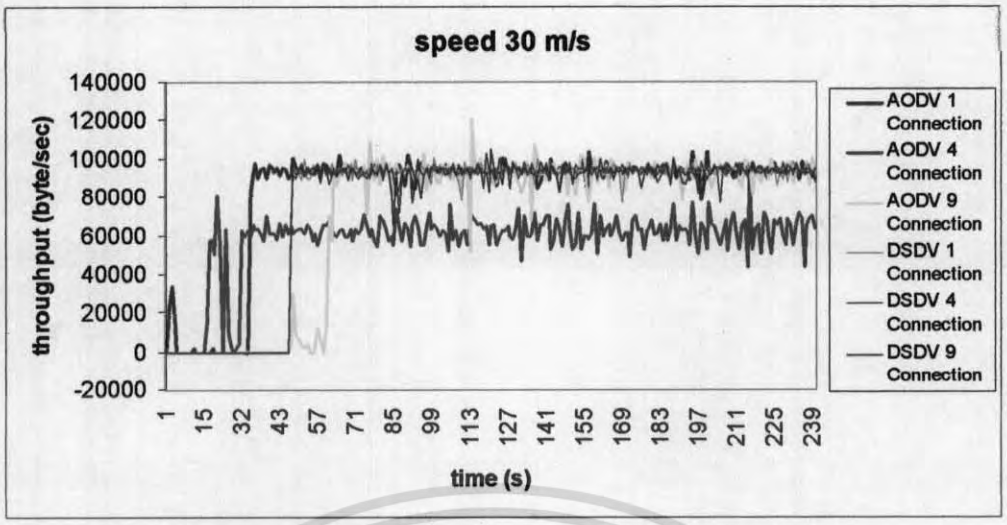
เพิ่มขึ้นถ้ามีการร้องขอเส้นทางสำเร็จก็สามารถสื่อสาร แต่ถ้าไม่สามารถร้องขอเส้นทางสำเร็จก็สื่อสารไม่ได้



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



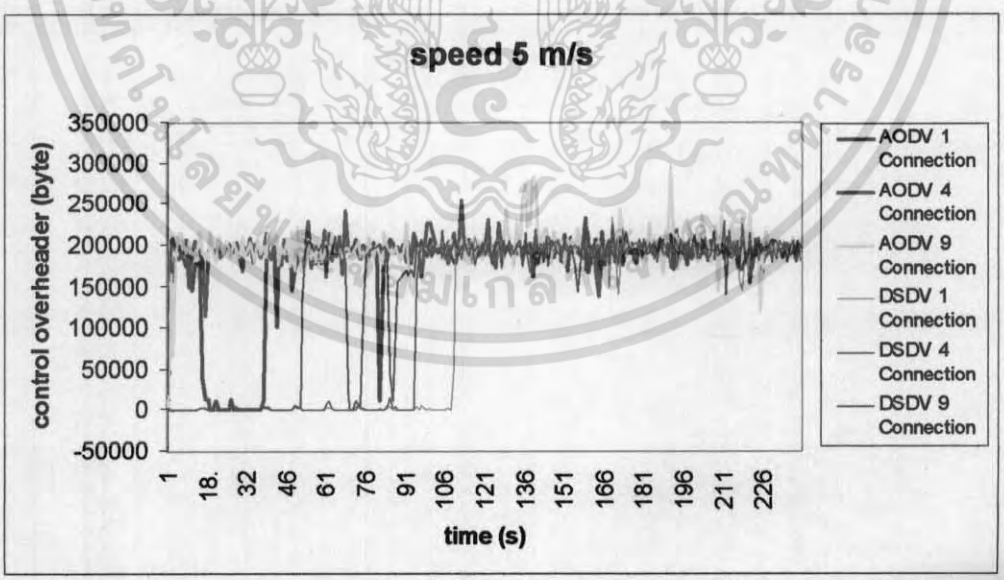
(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



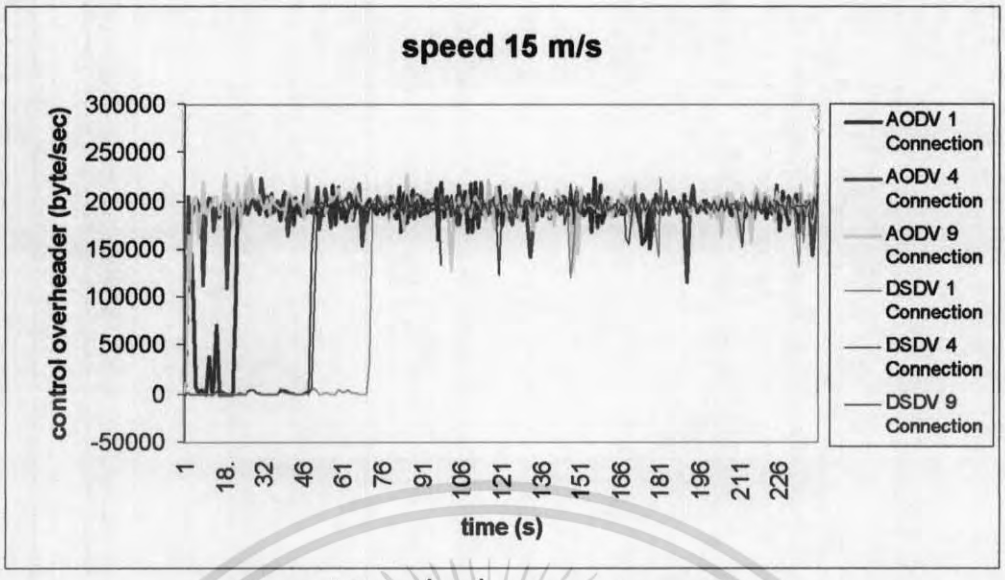
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.7 ค่าทรูพุตการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน

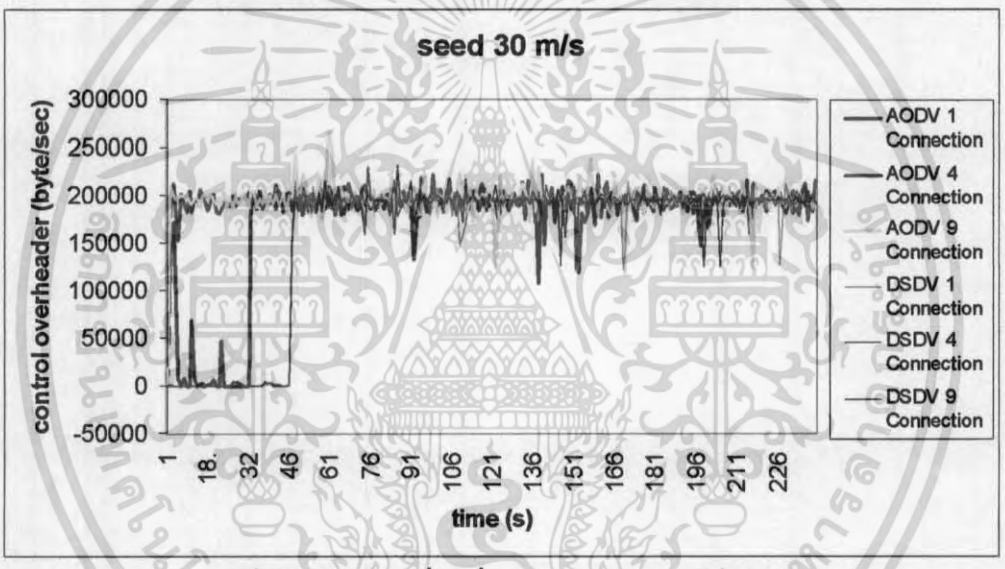
รูปที่ 4.8 (ก) (ข)และ(ค) เปรียบเทียบค่าโอเวอร์เฮดที่ใช้ในการค้นหาเส้นทางของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และ DSDV โปรโตคอล AODV จะมีค่าโอเวอร์เฮดมากกว่าโปรโตคอล DSDVเนื่องจากลักษณะเครือข่ายจะมีโหนดข้างเคียงอยู่มากจึงทำให้มีโอเวอร์เฮดที่ใช้ในอัลกอลิทึมทักทายเพื่อประกาศความคงอยู่ของโหนดข้างเคียงเพิ่มตามขึ้นไปด้วย และค่าโอเวอร์เฮดของการบรอดแคสต์ของโปรโตคอล DSDV มีค่าน้อยลงเมื่อเทียบค่าโอเวอร์เฮดของอัลกอลิทึมทักทายของโปรโตคอล AODV



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



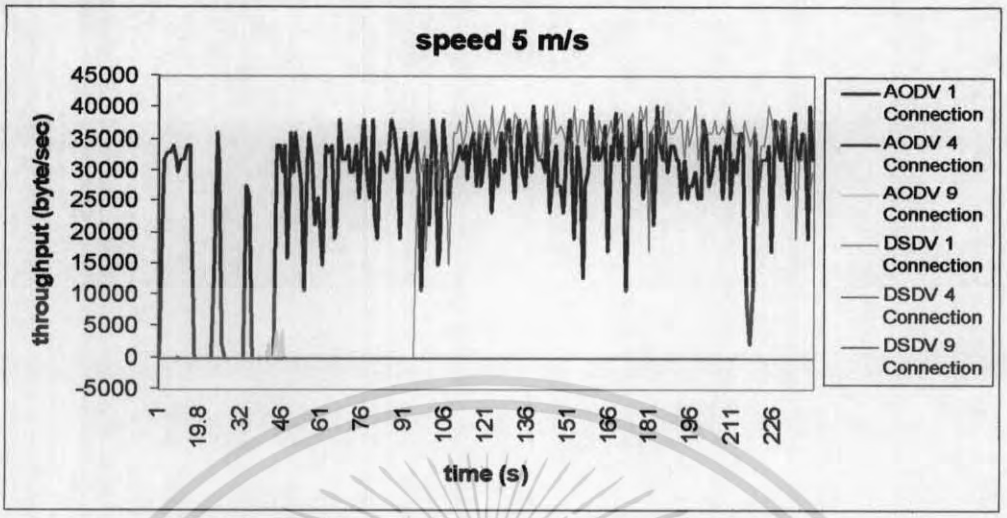
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.8 โอเวอร์เฮดการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางเข้าหากัน

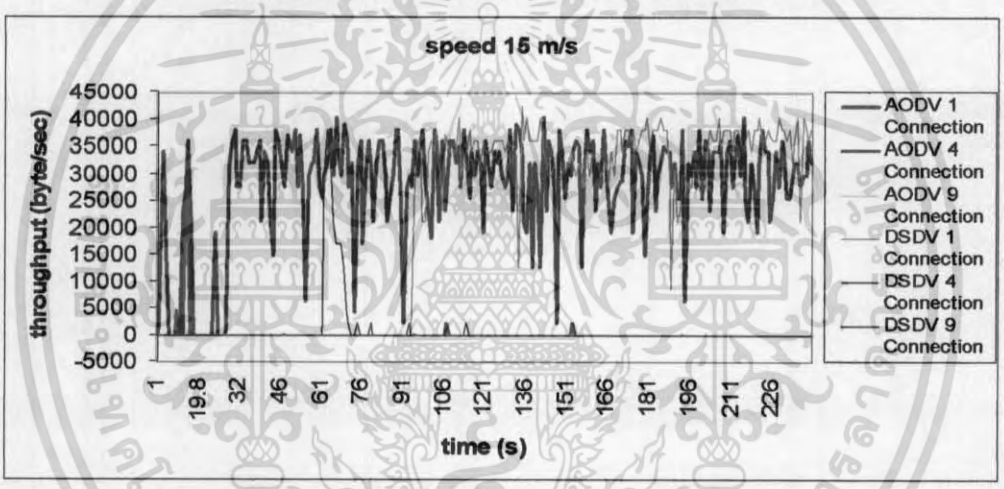
4.5 ผลการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางออกจากกัน

รูปที่ 4.9 เปรียบเทียบค่าทรูพุตของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV โปรโตคอล AODV จะให้ค่าทรูพุตมากกว่าโปรโตคอล DSDV เนื่องจากไม่มีการหน่วงรอค่าการปรับปรุงเส้นทางจากโหนดข้างเคียงจนครบอย่างเช่นโปรโตคอล DSDV แต่ในที่มีการถ่ายโอนข้อมูลโปรโตคอลจะมีการฟลัดดิงบ่อยเนื่องจากรูปแบบเครือข่ายในการจำลองมีระยะทางห่างมากขึ้นทำให้มีการส่งต่อบรอดแคสต์จะใช้ระยะเวลามากกว่าโปรโตคอล DSDV เพราะโปรโตคอล DSDV จะมีการปรับปรุงการถ่ายข้อมูลแบบเต็มและการถ่ายข้อมูลแบบส่วนเพิ่มเติมแต่โปรโตคอล AODV จะมีแต่การถ่ายข้อมูลแบบเต็มจึงทำให้ค่าทรูพุตโดยเฉลี่ยน้อยกว่าโปรโตคอล DSDV ในขณะที่คู่การสื่อสารเพิ่มขึ้นถ้ามีการร้องขอ

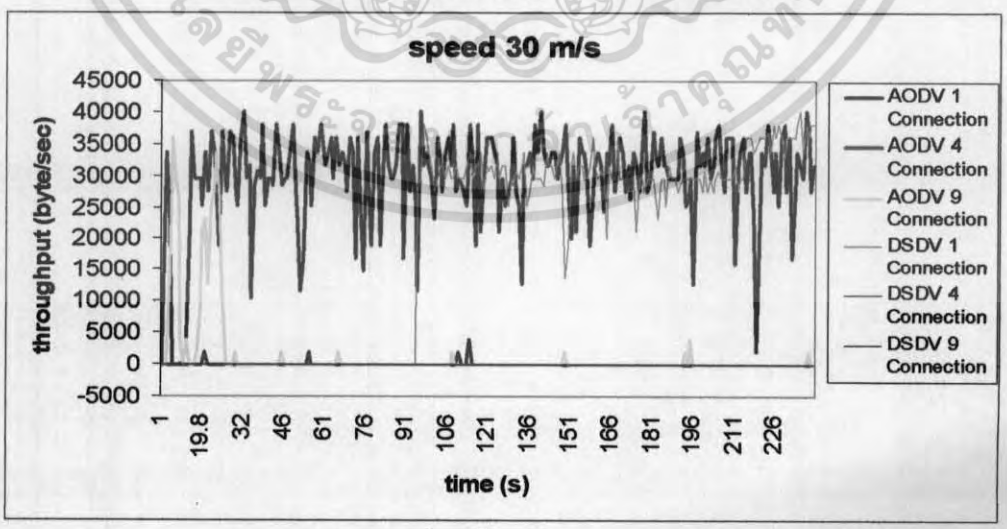
เส้นทางสำเร็จก็สามารถสื่อสารดังรูปที่ 4.9 (ข) จะเห็น DSDV 4 Connection จะร้องขอเส้นทางได้ก็มีการสื่อสารกัน แต่ถ้าไม่สามารถร้องขอเส้นทางสำเร็จก็สื่อสารไม่ได้



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



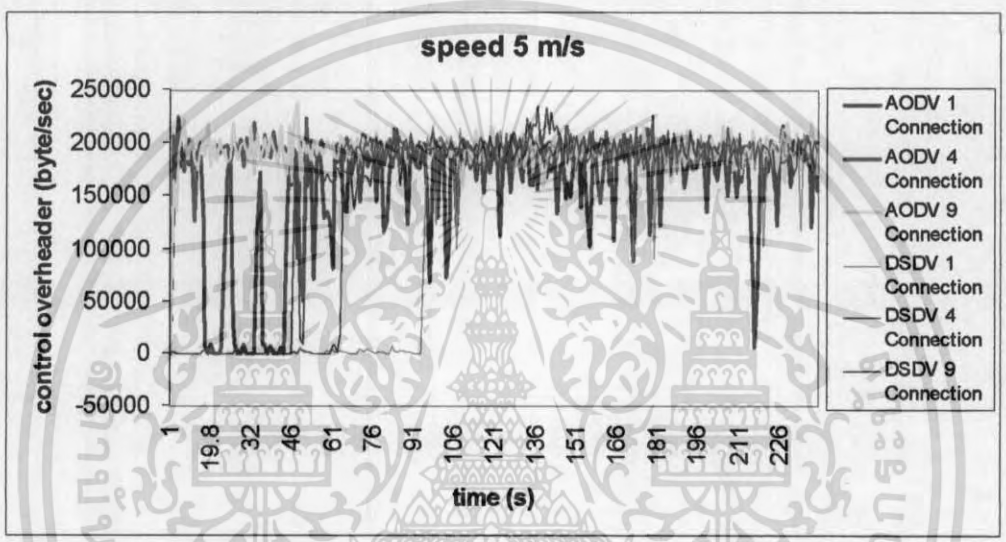
(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



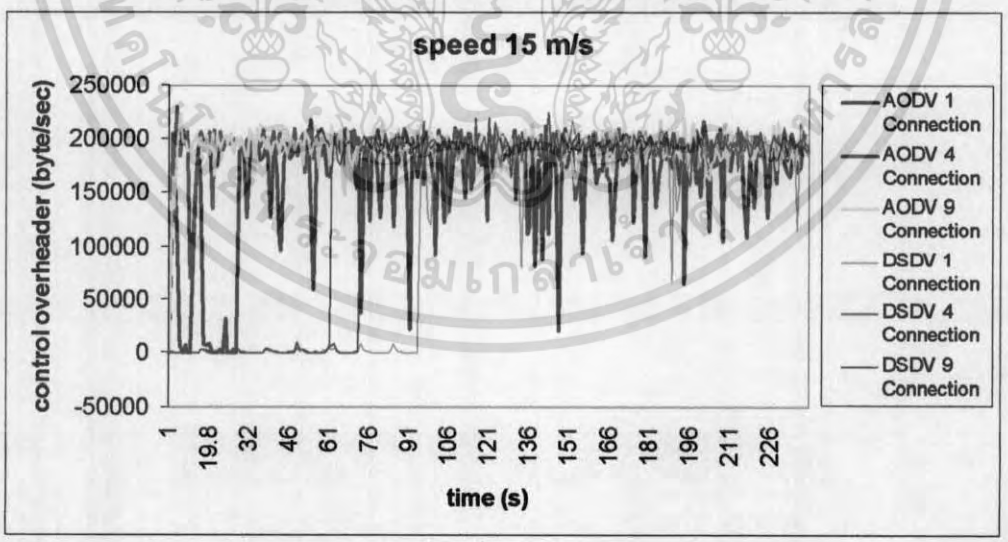
(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.9 ค่าทรูพุตการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางออกจากกัน

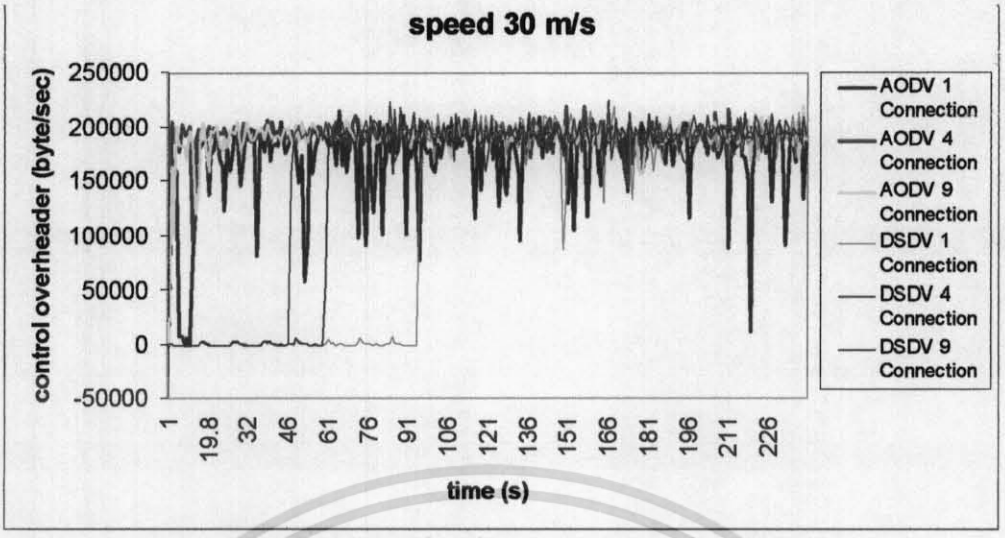
รูปที่ 4.10 (ก) (ข)และ(ค) เปรียบเทียบค่าโอเวอร์เฮดที่ใช้ในการค้นหาเส้นทางของระบบที่มีโปรโตคอลการจัดเส้นทางแบบ AODV และDSDV ค่าโอเวอร์เฮดของโปรโตคอล AODV มากกว่าโปรโตคอล DSDV แต่ถ้าไม่คิดในช่วงที่เกิดการหน่วงค่าโอเวอร์เฮดของโปรโตคอล DSDV มีค่ามากกว่าโปรโตคอล AODV เนื่องจากมีการบรอดแคสต์เป็นช่วงๆเพื่อปรับปรุงการเลือกเส้นทาง ส่วนโปรโตคอล AODV จะมีการปรับปรุงเส้นทางก็ต่อเมื่อเส้นทางเดิมนั้นไม่สามารถใช้งานได้ และจากรูปแบบเครือข่ายในการจำลองโหนดข้างเคียงจะมีน้อยจึงทำให้โปรโตคอล AODV มีค่าโอเวอร์เฮดที่ใช้ในอัลกอลิทึมประกาศความคงอยู่ของโหนดข้างเคียงน้อยเมื่อเทียบกับการบรอดแคสต์เพื่อปรับปรุงเส้นทางเป็นช่วงๆ ของโปรดตคอล DSDV



(ก) ความเร็วการเคลื่อนที่ของโหนด 5 เมตรต่อวินาที



(ข) ความเร็วการเคลื่อนที่ของโหนด 15 เมตรต่อวินาที



(ค) ความเร็วการเคลื่อนที่ของโหนด 30 เมตรต่อวินาที

รูปที่ 4.10 ค่าโอเวอร์เฮดการจำลองโดยเคลื่อนที่โหนดต้นทางและโหนดปลายทางออกจากกัน

บทที่ 5
สรุปและวิจารณ์

5.1 สรุป

ปัจจุบันเครื่องมือสื่อสารข้อมูลในแบบไร้สายเป็นสิ่งที่อำนวยความสะดวกเป็นอย่างมากและด้วยราคาที่ลดลงของเทคโนโลยีทางด้านการผลิตทำให้ความนิยมใช้เครือข่ายไร้สายเพิ่มมากขึ้น สำหรับเครือข่ายไร้สายแอดฮอคเป็นอีกหนึ่งทางเลือกที่ให้ความสะดวกเนื่องจากความอิสระในการเคลื่อนที่ของโฮสต์ที่อยู่ในเครือข่าย และสามารถนำไปใช้ในพื้นที่โดยไม่ต้องทำการติดตั้งสถานีฐาน เช่นงานด้านทางทหาร การสำรวจ ซึ่งพื้นที่การใช้งานไม่เหมาะสมหรือไม่สะดวกต่อการติดตั้งสถานีฐานได้

เนื่องจากโปรโตคอลในการเลือกเส้นทางที่ทำการพิจารณาเป็นโปรโตคอลการเลือกเส้นทางที่มีใช้อยู่ทั่วไป แต่ในการทดลองเพื่อเป็นการพิจารณาค่าประสิทธิภาพของโปรโตคอลที่พิจารณาต่อรูปแบบที่มีเครือข่ายเฉพาะ โดยจะพิจารณาองค์ประกอบโดยรวมข้อเด่นข้อด้อยของโปรโตคอลการเลือกเส้นทางที่มีต่อรูปแบบเครือข่ายเฉพาะ

จากผลการทดลองโปรโตคอล DSDV ข้อเด่นจะเป็นโปรแอคทีฟคือมีการประเมินเส้นทางอยู่เป็นช่วงๆจะทำให้ได้เส้นทางที่ดีที่สุด ก็จะทำให้ค่าทรูพุตมีค่าสูง และการปรับปรุงเส้นทางข้อมูลจะมีทั้งแบบเต็มและแบบส่วนเพิ่มเติม ในแบบส่วนเพิ่มเติม ทำให้มีโอเวอร์เฮดน้อยจากรูปแบบที่มีการเปลี่ยนแปลงน้อย ข้อด้อยโปรโตคอล DSDV จะมีการหน่วงในช่วงการเริ่มต้นการหาเส้นทางเพราะต้องรอการปรับปรุงเส้นทางจากโหนดข้างเคียงจนครบ และถ้าเครือข่ายมีการเปลี่ยนแปลงรูปแบบมากหรือจำนวนโหนดมากก็จะทำให้การหน่วงมากไปด้วย การปรับปรุงเส้นทางเป็นช่วงทำให้มีโอเวอร์เฮดมาก

จากผลการทดลองโปรโตคอล AODV ข้อเด่นคือจะสามารถหาเส้นทางได้รวดเร็วโดยวิธีฟลัดดิงสามารถส่งข้อมูลได้ใกล้เคียงเวลาที่ร้องการใช้เส้นทางมีโอเวอร์เฮดน้อยเพราะเป็นโปรโตคอลที่ทำงานแบบรีแอคทีฟคือจะมีการหาเส้นทางต่อเมื่อมีการร้องขอ เมื่อเส้นทางเดิมไม่สามารถติดต่อได้จะมีการค้นหาเส้นได้รวดเร็วดังนั้นค่าความเร็วในการเคลื่อนที่โหนดที่ค่าต่างๆไม่มีผลต่อโปรโตคอล AODV ข้อด้อยโปรโตคอล AODV จะไม่มีการเลือกเส้นทางที่ดีที่สุด เพราะทางเส้นทางเดิมใช้งานได้ก็จะไม่ทำการหาเส้นทางใหม่เป็นผลทำให้ค่าทรูพุตมีโอกาสน้อยได้

ผลการทดลองแสดงให้เห็นว่าเครือข่ายที่มีรูปแบบในการจำลองดังวิธีการทดลองควรใช้โปรโตคอลการเลือกเส้นทางแบบ AODV เพราะทำให้ได้ค่าประสิทธิภาพต่างๆที่ยอมรับได้แต่ในสภาวะแวดล้อมจริงยังมีปัจจัยทางกายภาพอื่นอีกมากที่มีผลกระทบต่อการทำงานของเครือข่าย แต่การจำลองยังเป็นทางเลือกที่ดีในการทำนายแนวโน้มประสิทธิภาพของเครือข่ายเพื่อนำไปสู่การพัฒนาการทำงานของเครือข่ายต่อไป

5.2 แนวทางการพัฒนาต่อ

การนำเน็ตเวิร์คไร้สายแบบเฉพาะกิจมาใช้งาน ถือเป็นการขยายความสามารถอุปกรณ์ที่มีใช้อยู่แล้วให้สามารถใช้งานได้เพิ่มขึ้นโดยการพัฒนาซอฟต์แวร์ การพัฒนาโปรโตคอลการจัดเส้นทางโดยคำนึงให้มีความเหมาะสมกับงานที่มีลักษณ์เฉพาะและคำนึงถึงปัจจัยทางกายภาพในการจำลอง การสื่อสารโดยทั่วไปยังต้องการความปลอดภัยในการสื่อสาร การประหยัดพลังงานในการสื่อสารจะสามารถเพิ่มประสิทธิภาพให้กับอุปกรณ์ได้ ดังนั้นการพัฒนาโปรโตคอลการจัดเส้นทางให้โปรโตคอลมีความปลอดภัยมากขึ้น หรืออาจพัฒนาให้ประหยัดการใช้พลังงานมากขึ้น