

โครงการวิจัยย่อยลำดับที่ 11

เรื่อง การประเมินประสิทธิภาพสำหรับโครงข่าย TCP/IP ผ่านดาวเทียม ปีที่ 4

1. ผู้รับผิดชอบโครงการ รศ.ดร. ประสิทธิ์ ทิมพุดิ

2. วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาประสิทธิภาพในระบบสื่อสารดาวเทียม เพื่อประเมินและวิเคราะห์หาวิธีปรับปรุงประสิทธิภาพให้ดีขึ้น
2. เพื่อพัฒนาและปรับปรุงวิธีการจัดหาเส้นทางในการส่งข้อมูลของการสื่อสารดาวเทียม ซึ่งจะทำให้ได้เส้นทางในการส่งข้อมูลที่มีการประวิงเวลาที่ต่ำ และหลีกเลี่ยงการใช้การเชื่อมโยงที่เกิดความคับคั่งบนโครงข่าย ส่งผลให้โครงข่ายมีประสิทธิภาพที่ดี
3. เพื่อศึกษาประสิทธิภาพในระบบสื่อสารดาวเทียมที่ทำการส่งผ่านข้อมูลโดยโปรโตคอล TCP/IP
4. เพื่อพัฒนาและปรับปรุงวิธีการส่งผ่านข้อมูลสำหรับโครงข่าย TCP/IP ผ่านดาวเทียมให้มีประสิทธิภาพมากขึ้น

3. ขอบเขตหรือเป้าหมายของโครงการ

1. ทำการศึกษาและวิเคราะห์รูปแบบปัจจุบันของการสื่อสารผ่านดาวเทียมที่มีการนำเสนอ
2. ศึกษาปัญหาที่เกิดขึ้นและหาแนวทางการแก้ไขเพื่อเพิ่มประสิทธิภาพของการสื่อสารดาวเทียม
3. ปรับปรุงวิธีการจัดหาเส้นทางในการส่งข้อมูลของการสื่อสารดาวเทียม โดยนำการค้นหาเส้นทางแบบเรียลไทม์ประยุกต์ใช้กับวิธีการหลีกเลี่ยงการใช้เส้นทางที่คับคั่ง ให้ได้เส้นทางในการส่งข้อมูลที่มีการประวิงเวลาที่ต่ำ
4. ทำการศึกษาระบบการทำงานของการทำงานของการส่งข้อมูลโดยการใช้โปรโตคอล TCP/IP บนดาวเทียม ซึ่งปัจจุบันจะพัฒนาให้เป็น back bone สำหรับเชื่อมโยงระบบอินเทอร์เน็ตที่มีประสิทธิภาพ
5. ศึกษาปัจจัยที่มีผลกระทบต่อประสิทธิภาพของการส่งผ่านข้อมูล TCP/IP ผ่านดาวเทียม โดยพิจารณาปัจจัยต่างๆ เช่น เทคนิคการมอดูเลต, การควบคุม error correction etc... และหาแนวทางแก้ไข
6. เขียนโปรแกรมเพื่อจำลองการทำงานของระบบ วิเคราะห์และเปรียบเทียบผลการทดลองที่ได้จากการจำลองผล

4. ส่วนงานที่ได้ดำเนินการไปแล้ว

1. ศึกษาการจัดเรียงตัวของดาวเทียมวงโคจรต่ำ โดยแบ่งออกเป็น 2 แบบคือ
 - 1.1 การจัดเรียงตัวแบบ ๆ
 - 1.2 การจัดเรียงตัวแบบ 2 ๆ
2. ศึกษาและวิเคราะห์การประวิงเวลาของดาวเทียม

ในการวิเคราะห์การประวิงเวลานี้ได้วิเคราะห์การประวิงเวลาแบบจุดถึงจุด (End-to-End delay) โดยได้แบ่งวิเคราะห์เป็น 2 กรณีคือ การประวิงเวลาของดาวเทียม GEO และ การประวิงเวลาของดาวเทียม LEO

3. ศึกษาและใช้งาน โปรแกรม Network Simulator (NS) บนระบบปฏิบัติการ Windows ในการจำลองผลของโปรโตคอลการจัดเส้นทางที่ได้มีการปรับปรุงใหม่นั้นจะต้องทดสอบบน simulator ก่อนที่จะนำไปทดสอบบนอุปกรณ์การใช้งานจริงได้ ซึ่ง simulator ที่ใช้ในการจำลองโปรโตคอลการจัดเส้นทางบนโครงข่ายแอตดอกคือ NS-2 ซึ่ง NS เป็น discrete event simulator ซึ่งจะรองรับการจำลองของ TCP การจัดเส้นทาง และ multicast protocol บนโครงข่ายทั้งแบบมีสายและไร้สาย NS เป็น OTCL script driven simulator มีส่วนประกอบของโครงข่ายที่เป็น C++ เช่น โปรโตคอลของโครงข่าย ตัววัดเวลา และ โครงสร้างโครงข่าย เป็นต้น เพื่อที่จะให้ผู้ใช้งานสร้างสถานการณ์จำลองโครงข่ายได้ง่าย

4. เขียนโปรแกรมการจำลอง และวิเคราะห์ผล โดยให้ดาวเทียมมีการส่งข้อมูลข้ามวงโคจรระหว่างดาวเทียม GEO และ LEO โดยตรง และได้มีการจำลองผลโดยกำหนดให้มีปริมาณทราฟฟิกแบ่งเป็น 3 ระดับต่างๆ กัน คือ ให้มีปริมาณทราฟฟิกน้อย ปานกลาง และมาก จากนั้นนำผลที่ได้จากการจำลองปริมาณทราฟฟิกที่ต่างกันนี้มาเปรียบเทียบกัน

5. ศึกษาฟังก์ชันและพารามิเตอร์ของ โหนด ตำแหน่ง การเชื่อมโยง การทำแฮนด์โอเวอร์ และการจัดเส้นทางของดาวเทียมในการเขียนโปรแกรม Network Simulator เพื่อใช้ในการปรับปรุงการจำลองระบบ

6. เขียนฟังก์ชันเพิ่มเติมเพื่อตรวจสอบว่า ระยะห่างของดาวเทียมมากเกินที่จะมีการเชื่อมโยงได้หรือไม่

7. แก้ไขฟังก์ชันการทำ hand off เพื่อตรวจสอบว่า การเชื่อมโยงนั้นๆ ต้องการการ hand off หรือไม่ ถ้ามีการเปลี่ยนแปลงของการเชื่อมโยงใดๆ จะต้องทำ rerouting

8. แก้ไขฟังก์ชันในการเชื่อมโยงโดยมีการหาประเภทของดาวเทียมและเก็บไว้ในแพ็คเก็ตและปรับตารางเวลาใหม่สำหรับการเริ่มสล็อตต่อไป

9. เขียนโปรแกรมฟังก์ชันการจัดเส้นทางการส่งต่อข้อมูลในระบบการสื่อสารดาวเทียม เพื่อเพิ่มประสิทธิภาพของระบบให้ดียิ่งขึ้น

10. เก็บผลการทดลองทั้งหมด วิเคราะห์ผลการทดลอง ข้อดีข้อเสียของงาน เขียนรูปเล่มรายงาน

11. ศึกษารูปแบบการส่งข้อมูลบนโปรโตคอล TCP/IP ผ่านความถี่ระหว่างสถานีภาคพื้นดิน
ต้น ทางไปยังสถานีปลายทาง ปัจจัยหลักที่เป็นผลกระทบต่อประสิทธิภาพการส่งผ่านของ TCP/IP บน
ช่องสัญญาณความถี่ และการหลีกเลี่ยงความแออัดของช่องสัญญาณ จากบทความ “ TCP OVER
SATELLITE ... The Final Frontier ” และ “ TCP/IP Enhancement for Satellite Networks ”

โดยปัจจัยที่มีผลกระทบต่อประสิทธิภาพ โครงข่ายหลัก ได้แก่ 3 ปัจจัยคือ

- Long Feedback Delay
- Large Bandwidth Delay Product
- Transmission Errors

12. ศึกษาประสิทธิภาพของโปรโตคอล TCP/IP และกระบวนการควบคุมบิตผิดพลาดและ
การตอบรับ ของสัญญาณข้อมูลบนช่องสัญญาณความถี่จากบทความ “ TCP/IP Enhancement for Satellite
Networks ” และ “ TCP/IP Performance over Satellite Links ”

มีหัวข้อหลัก 4 เรื่อง

- Issues and Challenges (Channel Error Rates, Propagations Delays, Channel
Asymmetry)
- Link Layer Enhancements (Forward Error Correction Schemes, Automatic Repeat
Request Protocols)
- Improved TCP Versions (Basic TCP Enhancements)
- Intelligent Interworking (Multiple TCP Sessions, Link Layer Interworking, ACK Control
Schemes, Improved Satellite Features)

13. ศึกษาการประยุกต์ TCP/IP เพื่อเพิ่มประสิทธิภาพในการส่งสัญญาณข้อมูลผ่าน
ช่องสัญญาณความถี่ในรูปแบบ Intelligent Interworking

14. ศึกษาระบบ Broadband IP Network via Satellites

15. ศึกษาบทความที่ทำการวิเคราะห์ประสิทธิภาพอัลกอริทึมของการส่ง TCP รูปแบบใหม่ที่
คาดว่าจะ เป็น อัลกอริทึมที่จะเข้ามาแทนที่ในกลไกการควบคุม current congestion บน standard TCP ใน
อินเทอร์เน็ตยุคถัดไป

5. ส่วนงานที่จะดำเนินการต่อไป

1. ศึกษาบทความวิจัยที่คาดว่าน่าจะเป็นแนวทางและกรณีศึกษาที่ดีในการพัฒนาระบบการ
ส่งข้อมูลโดยการใช้โปรโตคอล TCP/IP บน ความถี่
2. เรียนรู้อัลกอริทึมรูปแบบใหม่ๆ ที่สามารถนำมาประยุกต์ใช้งานกับการส่งผ่าน โปรโตคอล
TCP/IP
3. ศึกษาการปรับปรุงประสิทธิภาพเครือข่ายความถี่ให้ดีขึ้น
4. ศึกษาและพัฒนาวิธีการค้นหาเส้นทางในการส่งต่อข้อมูลในงานวิจัยต่อไป

5. พัฒนาซอฟต์แวร์ที่ใช้ในการจำลองระบบ
6. กำหนดแบบจำลองและข้อกำหนดต่างๆ ขึ้นมา และเขียนโปรแกรมประมวลผลที่ได้ศึกษาและวิเคราะห์ผลที่ได้เพิ่มเติมเพื่อเป็นแนวทางในการพัฒนางานวิจัยต่อไป



สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

การประวิงเวลาของดาวเทียม GEO

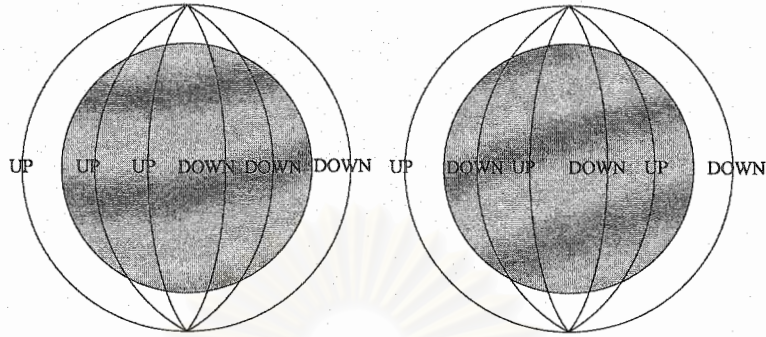
$$\begin{aligned}
 D_{End-to-End} = & \left(A_1 + \frac{L}{B_1} + P \right) \\
 & + 2 * (N - 1) * \left(\frac{A_1}{2} + D + \frac{L}{B_2} + I \right) \\
 & + \left(A_3 + D + \frac{L}{B_1} + P \right)
 \end{aligned} \tag{1}$$

โดยที่ $D_{End-to-End}$ คือการประวิงเวลารวม (end-to-end delay) N คือจำนวนฮอป A_1 คือการประวิงเวลาจากสถานีพื้นฐานไปยังการเข้าถึงดาวเทียม ($A_1 = 50$ มิลลิวินาที) A_2 คือการประวิงเวลาจากดาวเทียมไปยังการเข้าถึงดาวเทียม A_3 คือการประวิงเวลาจากดาวเทียมไปยังการเข้าถึงสถานีพื้นฐาน ($A_3 = 1$ มิลลิวินาที) B_1 คืออัตราบิตสำหรับพื้นดินไปยังการเชื่อมโยงดาวเทียม ($B_1 = 1.5$ Mbps) B_2 คืออัตราบิตสำหรับ ISL H คือความสูง ($H = 35,000$ กิโลเมตร) P คือการประวิงเวลาการแพร่ของ GSL ($I = 128.6$ มิลลิวินาที) I คือการประวิงเวลาการแพร่ของการเชื่อมโยงระหว่างดาวเทียมและพื้นดิน D คือการประวิงเวลาแบบประมวลผล (processing delay) ของโนดหรือดาวเทียม และ L คือความยาวของข้อมูล ($L = 512$ ไบต์)

จากสมการสามารถแบ่งออกได้เป็น 3 ส่วน ส่วนแรก แสดงเวลาที่ใช้ในการเข้าถึงดาวเทียม การส่งข้อมูลและการแพร่ไปยังดาวเทียม ส่วนที่สอง แสดงเวลาของฮอปจากดาวเทียมดวงหนึ่งไปยังอีกดวงหนึ่งซึ่งประกอบด้วย การประวิงเวลาแบบประมวลผล, การประวิงเวลาในการเข้าถึงฮอปถัดไป (next hop access delay), การประวิงเวลาในการส่งข้อมูล (transmission delay) และการประวิงเวลาการแพร่ ส่วนสุดท้าย แสดงการประวิงเวลาแบบประมวลผล, การประวิงเวลาการเข้าถึงสถานีภาคพื้นดิน (ground station access delay), การประวิงเวลาในการส่งข้อมูลและการประวิงเวลาการแพร่

การจัดเรียงตัวของดาวเทียม LEO

การจัดเรียงตัวของดาวเทียม LEO สามารถแบ่งได้เป็น 2 แบบคือ การจัดเรียงตัวแบบ π ครั้งหนึ่งของโลกดาวเทียมจะเคลื่อนที่ไปทางทิศใต้ และอีกครั้งหนึ่งดาวเทียมจะเคลื่อนที่ไปทางทิศเหนือ อีกแบบหนึ่งคือ การจัดเรียงตัวแบบ 2π ดาวเทียมที่อยู่ระนาบติดกันจะเคลื่อนที่ไปในทิศตรงข้ามกัน ดังแสดงในรูปที่ 1 cross-seam [1] เป็นจุดที่ดาวเทียมที่ระนาบติดกันเคลื่อนที่ไปทิศทางตรงข้ามกัน ซึ่งการจัดเรียงแบบแรกจะมี cross-seam อยู่ 2 ที่



รูปที่ 1 การจัดเรียงตัวแบบ π และการจัดเรียงตัวแบบ 2π ตามลำดับ

ข้อเสียของดาวเทียม LEO คือ มีความเร็วในการเคลื่อนที่สูงและมีรูปแบบโครงข่ายแบบไดนามิก ทำให้มีการจัดเส้นทางได้ลำบากเนื่องจากการเปลี่ยนแปลงรูปแบบโครงข่ายบ่อย นอกจากนี้การสื่อสารข้าม cross-seam ทำได้ยาก [1] เนื่องจากการเคลื่อนที่ผ่านกันเร็วมาก

การประวิงเวลาของดาวเทียม LEO

$$\begin{aligned}
 D_{End-to-End} = & \left(A_1 + \frac{L}{B_1} + P \right) \\
 & + (N-1) * \left(A_2 + D + \frac{L}{B_2} + I \right) \\
 & + \left(A_3 + D + \frac{L}{B_1} + P \right)
 \end{aligned} \tag{2}$$

โดยที่ $A_1 = 50$ มิลลิวินาที, $A_2 = 5$ มิลลิวินาที, $A_3 = 1$ มิลลิวินาที, $B_1 = 1.5$ Mbps, $B_2 = 155$ Mbps, $H = 1,350$ กิโลเมตร, $I = 7.5$ มิลลิวินาที, $P = 4.5$ มิลลิวินาที และ $L = 512$ ไบต์

สมการที่สองนี้มีส่วนแตกต่างจากสมการแรกคือ เทอมที่สอง นั่นคือ ดาวเทียมแบบ LEO มีการเชื่อมโยงระหว่างดาวเทียม โดยไม่จำเป็นต้องใช้สถานีภาคพื้นดิน และมีการเข้าถึงระหว่างดาวเทียม แทนการเชื่อมโยงระหว่างภาคพื้นดินกับดาวเทียม

การจัดสรรเส้นทางของดาวเทียม

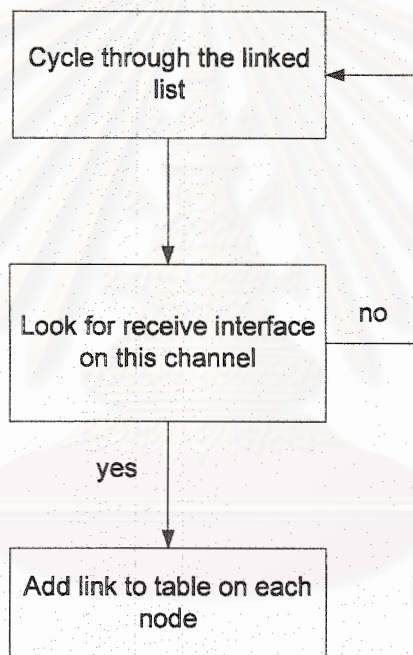
การจัดสรรเส้นทางใน NS ใช้การจัดสรรเส้นทางแบบศูนย์กลาง (centralized routing) ซึ่งเมื่อมีการเปลี่ยนแปลงรูปแบบโครงสร้างของโครงข่าย (topology) จะหารูปแบบโครงสร้างของโครงข่ายใหม่ แล้ว

คำนวณเส้นทางใหม่สำหรับทุกโหนด และสร้างตารางการส่งต่อข้อมูลบนแต่ละโหนด ตารางการส่งต่อข้อมูลนี้ประกอบด้วยตัวชี้ซึ่งจะชี้ไปยังการเชื่อมโยงขาออกใดๆ ที่จะนำไปสู่โหนดปลายทาง

1. การคำนวณรูปแบบโครงสร้างของโครงข่าย

ในการคำนวณรูปแบบโครงสร้างของโครงข่าย จะต้องหาข้อมูลความอยู่ใกล้ชิดกันของการเชื่อมโยง (link adjacency) จากโหนดต่างๆ และให้ข้อมูลปัจจุบันของรูปแบบโครงสร้างของโครงข่ายไปยังโหนดต่างๆ นั้นด้วย

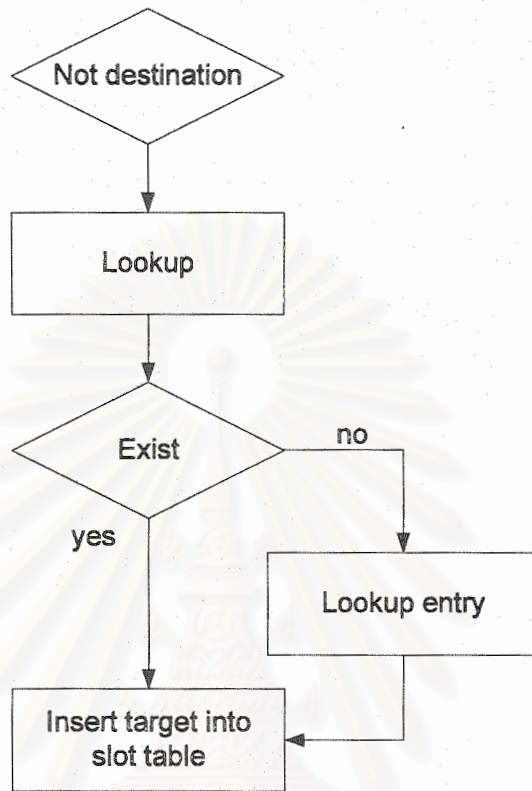
การหาความอยู่ใกล้ชิดกัน (adjacency) นั้นทำได้โดยพิจารณา linked list ของโหนดต่างๆ โดยตลอด จากนั้น จึงหาอินเตอร์เฟซภาครับ (receive interface) บนช่องสัญญาณ เมื่อพบความสัมพันธ์ของความอยู่ใกล้ชิดกัน (adjacency) แล้ว ให้เพิ่มการเชื่อมโยงนี้ไปยังตารางข้อมูลของแต่ละโหนด การทำงานของการคำนวณรูปแบบโครงสร้างของโครงข่ายแสดงในรูปที่ 2



รูปที่ 2 การคำนวณรูปแบบโครงสร้างของโครงข่ายของการจัดเส้นทางแบบศูนย์กลาง

2. การกระจายตารางการจัดเส้นทาง

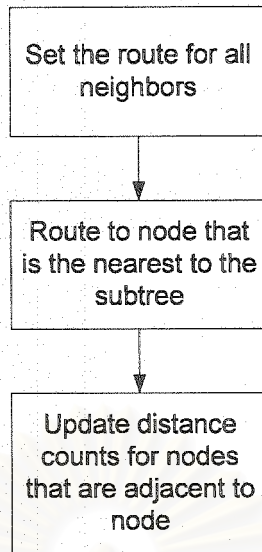
ในการกระจายตารางการจัดเส้นทาง เริ่มจากการทำสล็อตของตารางการจัดเส้นทางปัจจุบันให้ว่าง แล้วนำจุดหมายปลายทางมาพิจารณาเส้นทาง และเพิ่มจุดหมายลงในตาราง ไดอะแกรมการกระจายตารางการจัดเส้นทางแสดงในรูปที่ 3



รูปที่ 3 การกระจายตารางการจัดเส้นทาง

3. การคำนวณการจัดเส้นทาง

ในการคำนวณเส้นทางสำหรับโหนดต่างๆ สามารถทำได้โดย จัดเส้นทางไปยังโหนดข้างเคียงทั้งหมดก่อน จากนั้นจึงจัดเส้นทางไปยังโหนดที่มีระยะทางใกล้กับ subtree ที่สุด และอัปเดตระยะทาง (distance counts) สำหรับโหนดที่อยู่ใกล้ชิดกับโหนด วิธีการคำนวณเส้นทางของโหนดแสดงในรูปที่ 4



รูปที่ 4 การคำนวณเส้นทางของโหนด

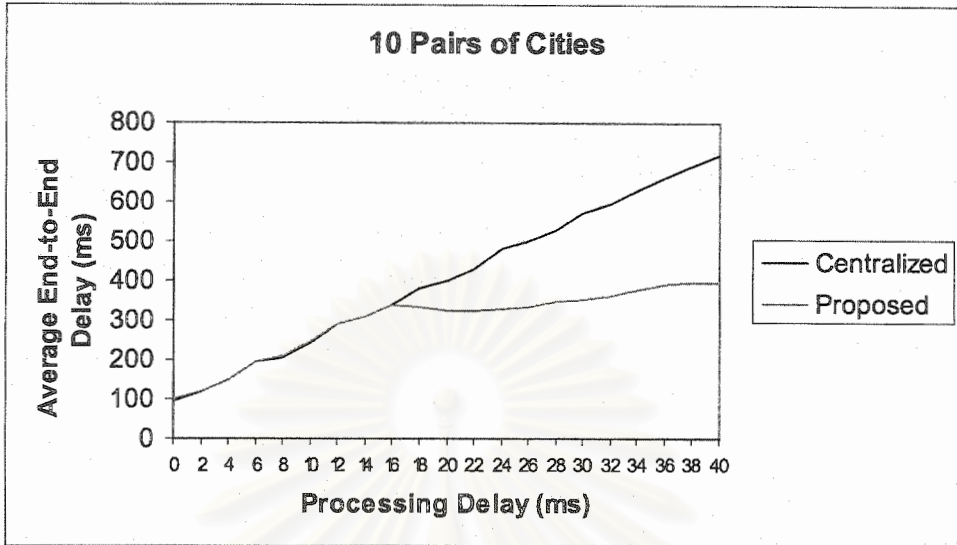
4. ข้อดีและข้อเสียของการจัดเส้นทางแบบศูนย์กลาง

ข้อดีของการจัดเส้นทางแบบศูนย์กลาง คือ ถ้าข้อมูล เช่น คุณสมบัติของพารามิเตอร์ ที่เปลี่ยนแปลงไม่บ่อย ข้อมูลต่างๆ นี้จะอยู่ในฐานข้อมูลส่วนกลาง และไม่จำเป็นต้องมีการประกาศ (advertise) ส่วนข้อเสียของการจัดเส้นทางแบบนี้คือ ถ้าต้องการซ่อมแซมเส้นทางอย่างรวดเร็ว จะไม่สามารถเชื่อถือระบบนี้ให้คำนวณเส้นทางที่ล้มเหลวได้ตามต้องการ และสำหรับรูปแบบโครงสร้างของโครงข่ายที่ใหญ่มากจะเสียเวลาในการคำนวณเส้นทางมาก

ผลการจำลอง

ในการจำลองผลได้จำลองให้มีการส่งข้อมูล โดยให้จุดรับส่งอยู่ตามที่ต่างๆ ทั่วโลก นอกจากนี้ ในการจำลองการรับส่งนั้น ได้แบ่งออกเป็น 3 กรณี คือ ให้มีทราฟฟิกต่างกันเป็น น้อย ปานกลาง และมาก โดยในงานวิจัยนี้ได้มีการวิเคราะห์การประวิงเวลารวมโดยเฉลี่ยและค่าวิสัยสามารถ

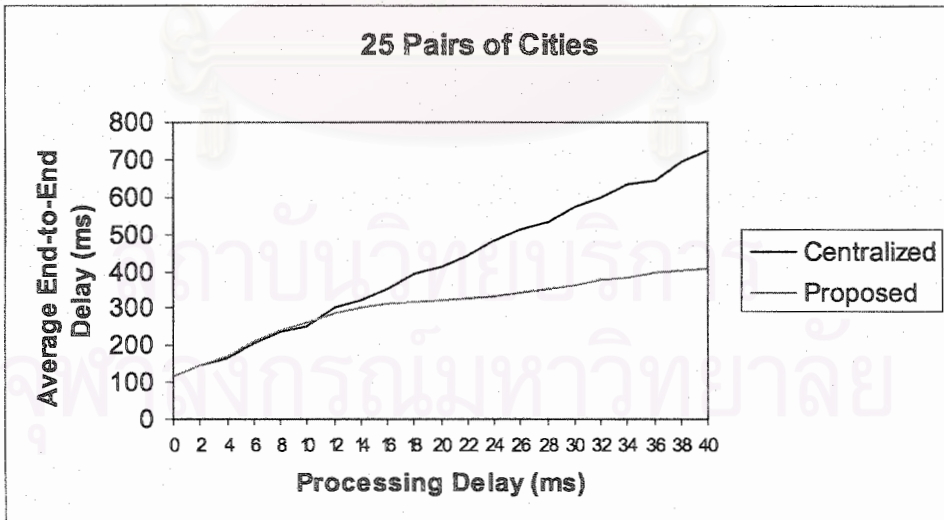
การจำลองผลได้แบ่งเป็น 3 กรณี คือ ให้มีคู่โหนดเป็น 10, 25 และ 50 คู่ โหนดต่างๆ นี้จะอยู่ตามเมืองต่างๆ กันทั่วโลก เพื่อเป็นการจำลองเสมือนกับการใช้โครงข่ายดาวเทียมในชีวิตจริง ผลการจำลองเมื่อมีคู่โหนดเป็น 10 คู่แสดงในรูปที่ 5



รูปที่ 5 ผลการจำลองเมื่อมีคู่โหนดเป็น 10 คู่

จากผลการจำลองจะเห็นได้ว่า ระบบที่ใช้การจัดเส้นทางที่นำเสนอมีค่าดีกว่าการจัดเส้นทางแบบศูนย์กลาง ค่าความต่างของการประวิงเวลาแสดงในตารางที่ 1

ผลการจำลองในกรณีที่ 2 เมื่อมีคู่โหนดเป็น 25 คู่แสดงในรูปที่ 6

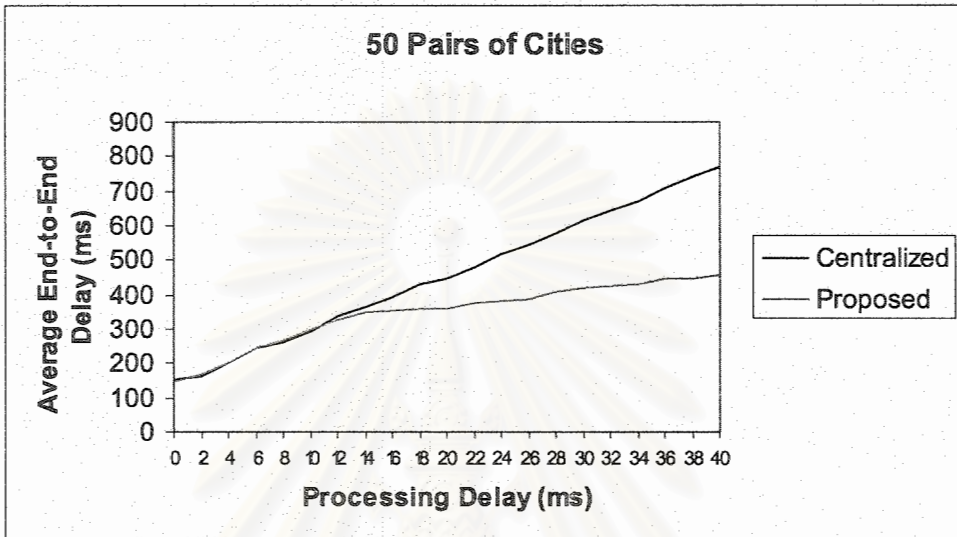


รูปที่ 6 ผลการจำลองเมื่อมีคู่โหนดเป็น 25 คู่

การจำลองในส่วนนี้ได้เพิ่มคู่โหนดจาก 10 เป็น 25 มีการประวิงเวลารวมเพิ่มขึ้นเพียงเล็กน้อย (20 มิลลิวินาที) ทั้งนี้เนื่องจากโหนดจะกระจายไปตามเมืองต่างๆ ทั่วโลก แต่ถึงอย่างไรก็ตาม การจัดเส้นทางที่ได้

นำเสนอที่ยังคงมีแนวโน้มการประวิงเวลาที่ดีกว่า และจะเห็นได้ว่ามีประสิทธิภาพดีขึ้นเร็วกว่าเดิม นั่นคือเริ่มมีความแตกต่างที่การประวิงเวลาแบบประมวลผลที่ 12 มิลลิวินาที ค่าความต่างของการประวิงเวลาและเปอร์เซ็นต์การลดลงแสดงในตารางที่ 1

ผลการจำลองในกรณีที่ 3 เมื่อมีคู่โหนดเป็น 50 คู่แสดงในรูปที่ 7



รูปที่ 7 ผลการจำลองเมื่อมีคู่โหนดเป็น 50 คู่

จากผลการจำลองจะเห็นได้ว่า จำนวนโหนดที่เพิ่มขึ้นไม่ได้มีผลกระทบทำให้การประวิงเวลาเพิ่มขึ้นมากนัก ทั้งนี้เนื่องจากโหนดอยู่กระจายกันไป ทำให้ผลของการเข้าถึงแบบหลากหลาย ที่มีต่อการประวิงเวลาลดลง และเช่นเดียวกันกับในกรณีที่ 2 คือ การจัดเส้นทางที่ได้นำเสนอ ยังคงมีแนวโน้มการประวิงเวลาที่ดีกว่า และมีประสิทธิภาพดีขึ้นเร็วกว่าเดิม นั่นคือเริ่มมีความแตกต่างที่การประวิงเวลาแบบประมวลผลที่ 12 มิลลิวินาที ค่าความต่างของการประวิงเวลาและเปอร์เซ็นต์การลดลงของเวลาแสดงในตารางที่ 1

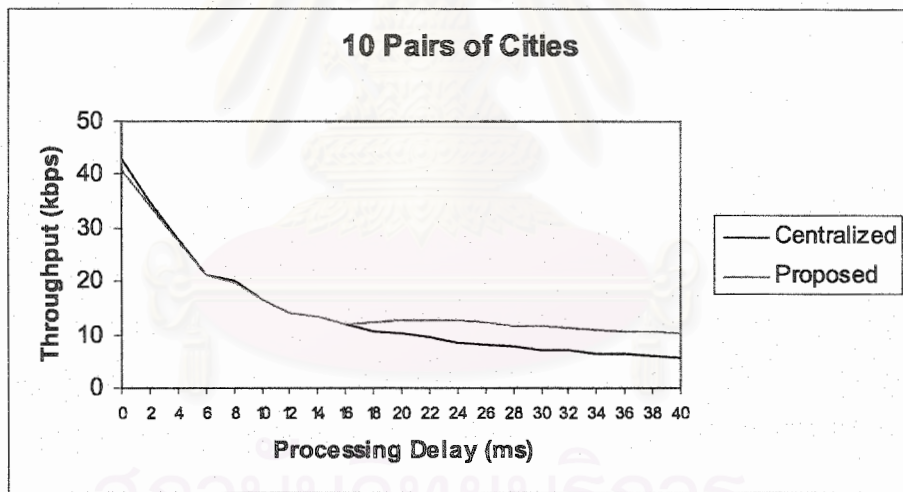
ตารางที่ 1 ความแตกต่างของเวลาและเปอร์เซ็นต์การลดลงของเวลาระหว่างการจัดสรรเส้นทางแบบศูนย์กลาง และการจัดสรรเส้นทางแบบแอดฮอก ในการส่งข้อมูลโดยให้โหนดอยู่ตามเมืองต่างๆ ทั่วโลก

	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3
12 มิลลิวินาที	-	16 มิลลิวินาที (5%)	11 มิลลิวินาที (3%)
14 มิลลิวินาที	-	24 มิลลิวินาที (7%)	16 มิลลิวินาที (4%)

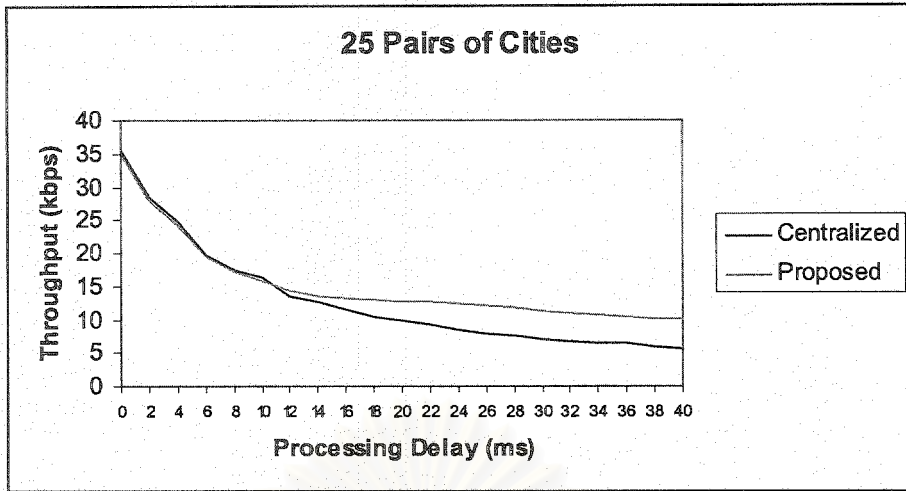
16 มิลลิวินาที	-	43 มิลลิวินาที (12%)	41 มิลลิวินาที (10%)
18 มิลลิวินาที	50 มิลลิวินาที (13%)	76 มิลลิวินาที (19%)	72 มิลลิวินาที (17%)
26 มิลลิวินาที	168 มิลลิวินาที (34%)	172 มิลลิวินาที (34%)	158 มิลลิวินาที (29%)
34 มิลลิวินาที	252 มิลลิวินาที (40%)	255 มิลลิวินาที (40%)	242 มิลลิวินาที (36%)
40 มิลลิวินาที	326 มิลลิวินาที (45%)	317 มิลลิวินาที (44%)	310 มิลลิวินาที (40%)

ผลของค่าวิสัยสามารถจากการส่งโดยให้โนดรับส่งอยู่ตามที่ต่างๆ ทั่วโลก

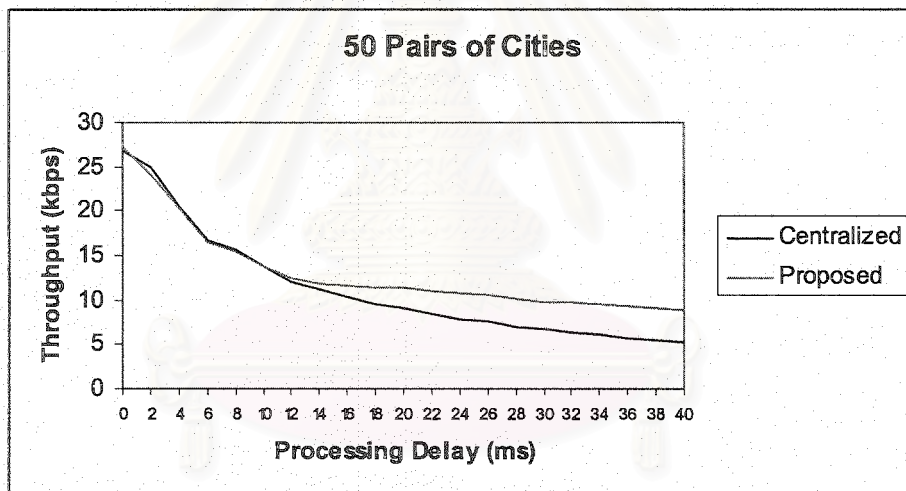
ต่อเนื่องจากส่วนก่อนหน้านี้ได้มีการวิเคราะห์ค่าการประวิงเวลา เมื่อหาค่าวิสัยสามารถของการส่งโดยให้จุดรับส่งอยู่ตามเมืองต่างๆ ทั่วโลก โดยมีคู่โนดเป็น 10, 25 และ 50 โนด จะเห็นได้ว่าระบบที่ใช้การจัดสรรเส้นทางแบบที่ได้นำเสนอจะมีค่าวิสัยสามารถที่ดีกว่า ดังผลการจำลองที่แสดงในรูปที่ 8 - 10



รูปที่ 8 ค่าวิสัยสามารถเทียบกับการประวิงเวลาแบบประมวลผล เมื่อมีโนดรับส่งอยู่ที่แตกต่างกัน ทั่วโลก และมีคู่โนดเป็น 10 คู่

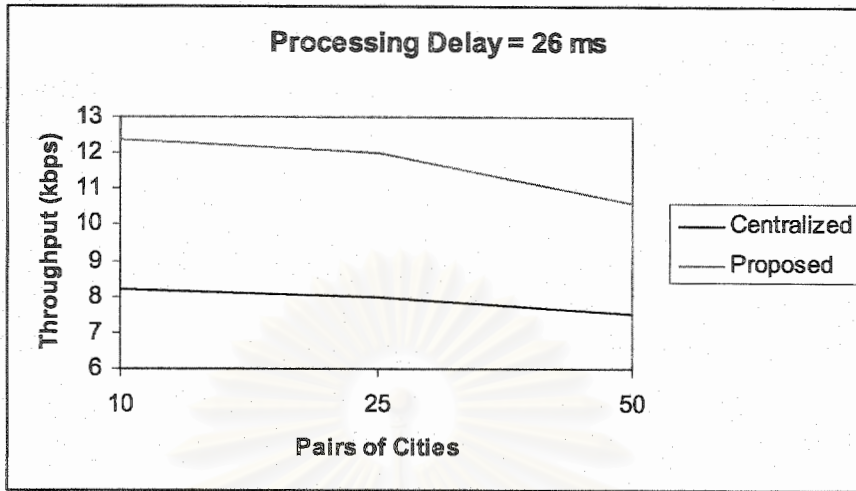


รูปที่ 9 ค่าวิสัยสามารถเทียบกับการประวิงเวลาแบบประมวลผล เมื่อมีโนดรับส่งอยู่ที่ต่างๆ ทั่วโลก และมีคู่โนดเป็น 25 คู่

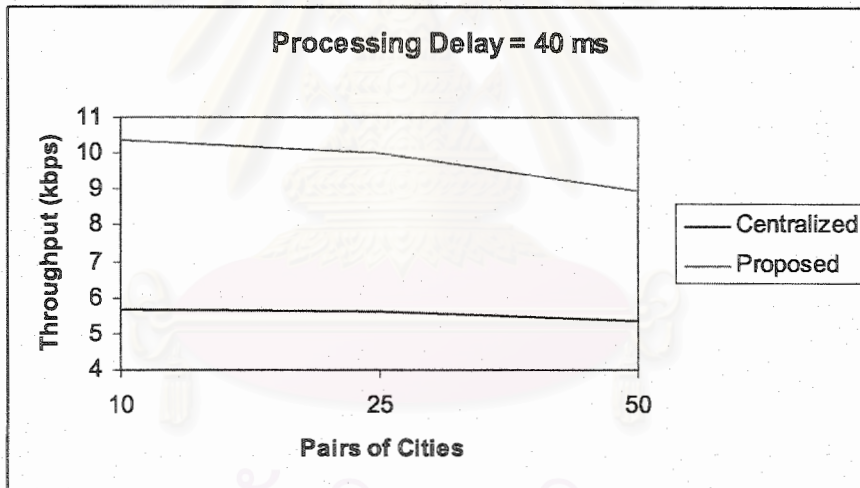


รูปที่ 10 ค่าวิสัยสามารถเทียบกับการประวิงเวลาแบบประมวลผล เมื่อมีโนดรับส่งอยู่ที่ต่างๆ ทั่วโลก และมีคู่โนดเป็น 50 คู่

เมื่อพิจารณาค่าวิสัยสามารถเทียบกับจำนวนคู่โนด พบว่า เมื่อให้จำนวนคู่โนดเพิ่มขึ้นค่าวิสัยสามารถจะลดลง ทั้งนี้เนื่องมาจากผลกระทบจากการเข้าถึงแบบหลากหลาย ดังที่ได้กล่าวไปแล้วในส่วนก่อนหน้านี้ แต่อย่างไรก็ตามจะเห็นได้ว่าระบบที่ใช้การจัดการที่นำเสนอจะมีค่าวิสัยสามารถมากกว่า ดังแสดงในรูปที่ 11 - 12



รูปที่ 11 ค่าวิสัยสามารถเทียบกับจำนวนคู่โหนด ที่การประวิงเวลาแบบประมวลผลเป็น 26 มิลลิวินาที



รูปที่ 12 ค่าวิสัยสามารถเทียบกับจำนวนคู่โหนด ที่การประวิงเวลาแบบประมวลผลเป็น 40 มิลลิวินาที

ในการศึกษาการใช้งานของโปรแกรม NS-2 (Network Simulator) เพื่อนำไปใช้ในการจำลองระบบเครือข่ายในงานวิจัย โดยโปรแกรมจำลองระบบ NS-2 มีลักษณะดังนี้

Network Simulator (NS)

ในการวิจัยทางด้านเครือข่าย จะมีโปรแกรมไว้จำลองระบบเครือข่ายเพื่อทำการศึกษาโปรโตคอลต่างๆ ซึ่งมีอยู่หลายตัวด้วยกัน เช่น

- NS-2 (Network Simulation)
- OpNet
- SSF (Scalable Simulation Framework)
- PARSEC (Parallel Simulation Environment for Complex Systems)

โดยในการวิจัยโครงการนี้ ได้เลือกใช้ NS-2 (Network Simulation) เนื่องจาก NS-2 เป็น Open Source นั่นก็คือ สามารถนำมาใช้ได้อย่างถูกต้องสิทธิ์

NS Simulator เป็นเครื่องมือที่ใช้จำลองระบบเครือข่ายแบบ discrete event simulator โดยมีการเขียนโปรแกรมบน 2 ภาษา คือ การเขียนโปรแกรมเป็นเชิงวัตถุ แบบ C++ และมีการใช้ภาษา OTcl (Object oriented of Tcl) โดยมีรายละเอียดดังนี้

- TCL (Tool Command Language) Script เป็นการเขียนในลักษณะ front end โดยสามารถสร้าง node และกำหนดคุณสมบัติต่างๆของมัน ซึ่งเหมือนกับการใช้ library ที่มีการสร้างไว้อยู่แล้ว
- C++ Script เป็นการเขียนในลักษณะ low level ที่สามารถสร้างโปรโตคอล และอัลกอริทึมต่างๆได้เอง เพื่อทำการศึกษาวิจัยในเรื่องต่างๆ

โดยในตัวโปรแกรม NS-2 ได้มีการแบ่งไฟล์ออกเป็น 2 ลักษณะเพื่อการใช้งาน คือ

- trace file (*.tr) จะมีลักษณะเป็น ascii file
- NAM file (*.nam) จะมีลักษณะเป็น Network Animator เพื่อทำการตรวจสอบ network ที่ออกแบบไว้

การทำงานของ NS-2 จะมีแพ็คเกจมากมายเพื่อนำไปใช้งาน โดยจะมีการปล่อย (release) ออกมาเป็นเวอร์ชันต่างๆ ซึ่งเป็น component ตามที่จำเป็นต้องใช้งาน สิ่งหนึ่งที่ทำให้ NS-2 เป็นที่นิยมก็คือ สามารถจำลองการทำงานของ TCP, UDP, Routing รวมถึงการทำ Multicast ทั้งบนเครือข่ายไร้สาย และเครือข่ายแบบปกติ

OTcl

Tcl เป็นภาษาคอมพิวเตอร์ที่มีคนนิยมใช้มากภาษาหนึ่ง เนื่องจากมี syntax ที่ไม่ยากมาก และสามารถนำไปรวมกับภาษาอื่นๆ ได้ นอกจากนี้โปรแกรมภาษานี้ยังเป็นโปรแกรมที่ไม่มีค่าใช้จ่ายอีกด้วย (ไม่เก็บค่าลิขสิทธิ์) ต่อมาได้พัฒนาส่วนของ Tcl ให้มีลักษณะการเขียนเชิงวัตถุ เพื่อให้เข้ากับภาษา C++ จึงกลายเป็น OTcl (Object oriented of Tcl)

CygWin

เนื่องจาก NS-2 ทำงานบนระบบปฏิบัติการ Linux แต่เมื่อต้องการให้ทำงานบนระบบปฏิบัติการ Windows จึงต้องทำการจำลองการทำงานของระบบ Linux บน Windows ซึ่งมีโปรแกรมชุดหนึ่งที่จะต้องนำมาช่วยคือ CygWin

Cygwin เป็นชุดโปรแกรมที่สร้างขึ้นเพื่อให้ใช้งานบนระบบปฏิบัติการ Windows เพื่อทำการจำลองระบบ Linux แต่ไม่ได้ทำทุกอย่างเหมือนระบบปฏิบัติการเดิม เพียงแต่ทำการจำลองเพื่อให้สะดวกในการพัฒนาโปรแกรมเท่านั้นเอง

ความแตกต่างระหว่างการใช้งาน NS-2 บนระบบปฏิบัติการ Windows และ Linux จะให้ผลที่แตกต่างกันบ้างในบางประเด็น คือ

- การสร้าง (build) จะทำในรูปแบบ allinone ไม่ได้ทำจาก components ย่อยต่างๆ
- การสร้าง (build) จะไม่ต้องการ Visual Studio สำหรับการ compile
- การใช้งาน NAM ไม่สามารถใช้งานใน stand alone Windows application แต่สามารถทำได้ใน Cygwin X server

ภาคผนวก ก

ในการศึกษาบทความที่ทำการวิเคราะห์ประสิทธิภาพอัลกอริทึมของการส่ง TCP รูปแบบใหม่ที่คาดว่าจะ เป็น อัลกอริทึมที่จะเข้ามาแทนที่ในกลไกการควบคุม current congestion บน standard TCP ในอินเทอร์เน็ตยุคถัดไป ซึ่งในบทความได้ทำการประเมินประสิทธิภาพด้วยการ simulate บน ns2 ใช้สภาวะเช่นเดียวกันกับ mobile satellite ผลของทั้ง 2 โปรโตคอล โดยเฉพาะ S-TCP แสดงให้เห็นความเป็นไปได้ของทางเลือกเพื่อปรับปรุง standard TCP ซึ่งหนึ่งในปัจจัยในการกำหนดประสิทธิภาพหลักของอินเทอร์เน็ตปัจจุบัน คือ ความสามารถของ standard Transport Control Protocol(TCP) ความสามารถของแบนด์วิธบน high speed backbone โดยที่จริงกลไก การควบคุม congestion ตั้งอยู่บน Additive Increase Multiplicative Decrease (AIMD) algorithm และตามมาด้วยการวิเคราะห์โครงข่ายที่ประพุดิตัวเป็น large Bandwidth Delay Product (BDP) ในการปรับปรุงรูปแบบ TCP ต่างๆ ในไม่กี่ปีมานี้มีจุดมุ่งหมายเพื่อแก้ปัญหาที่กล่าวมา ในบทความได้มุ่งเน้นความสนใจบน 2 ตัวหลักสำหรับทรานสปอร์ตโปรโตคอลของอินเทอร์เน็ตยุคถัดไป Scalable TCP(S-TCP)และ HighSpeed TCP (HSTCP) ทั้งคู่ถูกประยุกต์สำหรับ algorithm การ

ควบคุม congestion ของ TCP และจัดอยู่ในอันดับที่ดีกว่าของความสามารถในด้านแบนด์วิธ ดังเช่น high speed backbone ที่มีค่า BDP มาก มันเป็นเหตุผลในการคาดหวังในสภาวะการสื่อสารดาวเทียม สำหรับ ทั้ง S-TCP และ HSTCP ที่ถูกออกแบบขึ้น อันดับแรกค่าของ BDP ในกรณีของดาวเทียม GEO ปกติจะน้อยกว่า (มากกว่า 2เท่า) ใน backbone link บนการผิดพลาดในการส่งผ่านที่เกิดขึ้นจาก noise โดยทั่วไปแล้วจะละทิ้งในโครงข่ายดาวเทียมทั่วไปจะแสดงแพ็ค เกจ โดยพิจารณาที่มีอัตราการผิดพลาดสูง

ในบทความแสดงการประเมินประสิทธิภาพเชิงลึกของ S-TCP และ HSTCP ในสภาวะที่ใช้กับดาวเทียม GEO การประเมินชี้ให้เห็นโดยการ simulate บน ns2 โดยการกำหนดการเชื่อมต่อ TCP และ ประเมินอัตราการส่งผ่านโดย multiple connection sharing ในกรณี static user จะทำการเปลี่ยนแปลงจำนวนของผู้ใช้ เช่นเดียวกับค่าเฉลี่ย Frame Error Rate (FER) ใน transport level ในส่วนของผู้ใช้ที่มีการเคลื่อนที่ ให้ใช้ classical two-state GOOD/BAD Markovian channel model[4] ในการลำดับการนับสำหรับการ shadowing ในกรณีนี้จะเปลี่ยน ค่า FER และ ความเร็วการเคลื่อนที่ ในการใช้ พารามิเตอร์จากการวัด การปรับปรุงกลไกการควบคุม congestion โดย standard TCP ขึ้นอยู่กับ การ sliding window ซึ่งเป็นตัวกำหนดจำนวนแพ็คเกจที่ป้อนเข้าไปยังโครงข่าย ค่า congestion window จะถูก update หลังจากการรับแพ็คเกจตอบกลับ acknowledgment packet (ACK) และ ขึ้นกับ การตรวจจับแพ็คเกจที่สูญหาย การรับของ ACK ที่แสดงใน TCP คือสัญญาณของแบนด์วิธที่เหมาะสม ดังนั้น ค่า congestion window ,W อาจจะถูกเพิ่ม การเพิ่มอย่างรวดเร็วที่การเริ่มต้นของการเชื่อมต่อครั้งใหม่ (ดังที่เรียกว่า slow start phase) ขณะที่ผู้ส่ง TCP ขยาย W ด้วย 1 segment สำหรับแต่ละผู้รับ ACK ในวิธีนี้ congestion window ขยายตัวขึ้นแบบ exponentially บน Round Trip Time (RTT) เมื่อจุด threshold ถูกตัด การเชื่อมต่อจะเข้าสู่ช่วง congestion avoidance ขณะที่ผู้ส่งเพิ่มอัตราการส่งผ่านในรูปแบบเชิงเส้นบน RTT (congestion windows จะถูกเพิ่มขึ้น โดย $\frac{1}{W}$ ขึ้นกับการรับ ACK) การลดลงของ phase จะเริ่มขึ้น โดยการตรวจจับแพ็คเกจที่สูญหายไป แพ็คเกจที่สูญหายสามารถตรวจจับได้แต่ละส่วนเมื่อสิ้นสุดเวลาการส่งออก (ในกรณีที่ W กลับไปที่ 1) หรือ โดยการรับ ซ้ำกัน 3 ครั้งของ ACKs (ตอบสนอง congestion window ครั้งต่อครั้ง) เมื่อระยะเวลา timeout แต่ละส่วนรวมค่ามากพอ กรณีแรกคือการลดอัตราการส่งผ่านให้น้อยลง ในกรณีอื่น เมื่อบางแพ็คเกจรับได้โดยถูกต้องหลังจากเกิด loss แล้ว congestion จะถูกสมมติให้เป็น สภาวะ transient

ถ้าจุดที่ให้ความสนใจในสภาวะ steady-state เช่นกลไกการควบคุม congestion และ สมมติว่า loss ทั้งหมดถูกตรวจจับโดยการรับ ACKs ซ้ำ 3 ครั้ง (dupACKs) กลไกการควบคุมของ TCP อาจเขียนเป็นรูป classical AIMD

$$W \leftarrow W + \frac{1}{w} \quad \text{ขึ้นกับ ACK arrival}$$

$$W \leftarrow \frac{w}{2} \quad \text{ขึ้นกับ การรับ 3 dupACKs}$$

สำหรับ S-TCP ที่สร้างขึ้นด้วยหลักการ กลไก sliding window สำหรับการควบคุมอัตราการส่งข้อมูลและอยู่บนพื้นฐานของ Multiplicative Increase Multiplicative Decrease (MIMD) ซึ่งปรับเปลี่ยน congestion window เป็น

$$W \leftarrow W + \alpha \quad \text{ขึ้นกับ ACK arrival}$$

$$W \leftarrow [1 - \beta] \cdot W \quad \text{ขึ้นกับ การรับ 3 dupACKs}$$

ใน[1] $\alpha = 0.1$ และ $\beta = 0.125$

สำหรับ HSTCP ใช้หลักการ adaptive window algorithm ที่เสนอโดย S.Floyd สำหรับโครงข่ายที่มี BDP ใหญ่มากๆ การเพิ่มจำนวนและลดจำนวนของ congestion window ซึ่งเป็นผลตอบสนองการรับ ACK แสดงได้เป็น

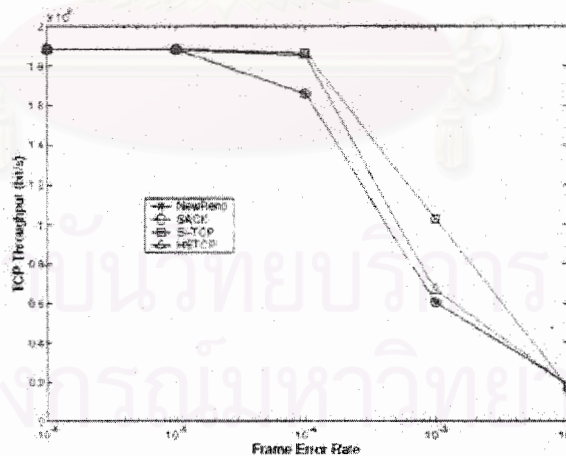
$$W \leftarrow W + \frac{a(W)}{w} \alpha \beta \quad \text{ขึ้นกับ ACK arrival}$$

10^{-2} ขึ้นกับ การรับ 3 dupACKs

โดยที่ $a(W)$ และ $b(W)$ อธิบายอยู่ใน [2]

ทั้ง S-TCP และ HSTCP ใช้กลไกการควบคุม congestion ใน slow start phase ซึ่งตั้งให้ทั้ง 2 เป็น W^* มีค่าไปถึง 16 และ 32 แพ็กเกจ

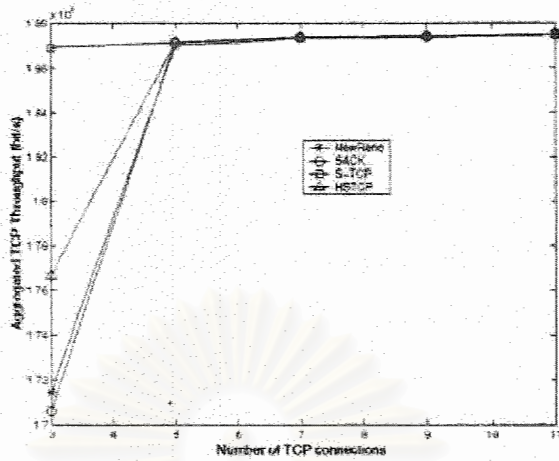
ได้ทำการเปรียบเทียบ S-TCP และ HSTCP ด้วย TCP โพรโตคอลหลัก 2 ตัว คือ NewReno และ New Reno ด้วย SACK โดยเริ่มจากการพิจารณา การเชื่อมต่อ TCP แบบเดี่ยว ในสถานะคงที่และศึกษาผลกระทบของ FER บนประสิทธิภาพระบบ ผลลัพธ์แสดงไว้ในรูป



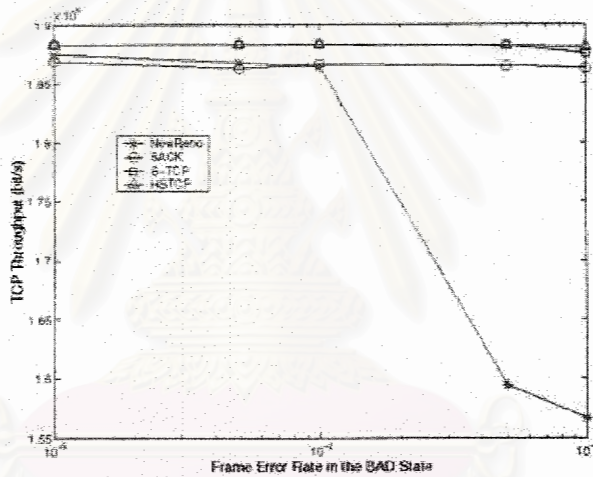
รูปแสดง TCP throughput vs. frame error rate สำหรับการเชื่อมต่อแบบเดี่ยว ($n = 1$) ในสถานะ fixed user (memoryless channel)

ในกรณีนี้จะสังเกตเห็นว่า TCP NewReno และ SACK พฤติกรรมคล้ายกัน โดยสามารถอธิบายด้วยความน่าจะเป็นของ multiple losses ด้วย window เดียวกับข้อมูลใน memoryless channel ส่วน HSTCP มีความสามารถเหนือกว่าเล็กน้อย ส่วน S-TCP เห็นได้อย่างชัดเจน ในช่วง FER ($10^{-5}, 10^{-2}$) แสดงให้เห็นว่า

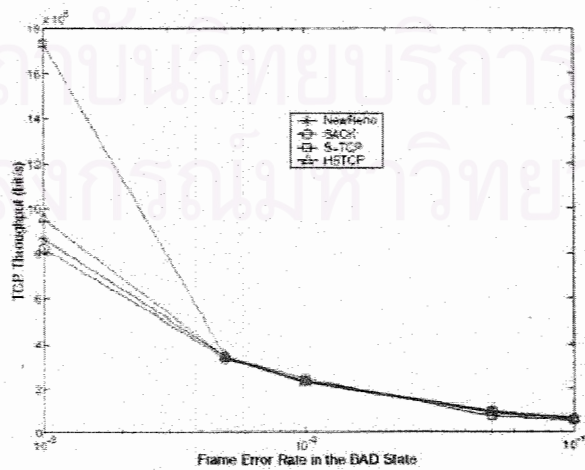
MIMD algorithm สามารถทำงานได้อย่างมีประสิทธิภาพที่ค่า α และ β ที่นำมาใช้ สำหรับค่า small congestion window ที่ขึ้นกับค่าอัตรา high error



รูปแสดง การรวม TCP throughput vs. number of concurrent persistent ที่ $FER = 10^{-3}$



รูปแสดง TCP throughput ในสภาวะ mobile vs. สภาวะ BAD state ความเร็วอุปกรณ์ mobile ที่ 60 km/h



รูปแสดง TCP throughput ในสถานะ mobile vs. frame error สถานะ BAD state ความเร็วอุปกรณ์ mobile ที่ 40 Km/h

จากผลทั้งหมดรวบรวมได้ว่า S-TCP และ HSTCP สามารถปรับปรุงประสิทธิภาพของ TCP ในโครงข่ายดาวเทียมขึ้นมาเล็กน้อย สังเกตได้ว่าหนึ่งในเหตุผลที่ทำให้ประสิทธิภาพถูกจำกัด ของทั้ง S-TCP และ HSTCP บน standard TCP คือ ค่าของ BDP ที่ถูกจำกัด ขณะที่ค่า 2 Mb/s สำหรับความจุแทนด้วยค่าที่เทคโนโลยีสามารถทำได้ ถ้าในอนาคตเทคโนโลยีก้าวหน้าไป ทั้ง S-TCP และ HSTCP จะให้ประโยชน์มาก เมื่อค่า BDP มีขนาดใหญ่ และอัตราส่วนระหว่าง การส่งผ่านและlink capacity พบว่า S-TCP มีประสิทธิภาพสูงกว่าตัวอื่น S-TCP จึงเป็นทางเลือกที่น่าสนใจสำหรับนำมาใช้งานบน link ของ geostationary satellite



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารอ้างอิง

- [1] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers." *Proceedings of ACM SIGCOMM*, (1994) : 234-244.
- [2] Hong-Seong Chang, Byoung-Wan Kim, Chang Gun Lee, Yanghee Choi, Sang Lyul Min, Hyun Suk Yang, and Chong Sang Kim, "Topological Design and Routing for Low Earth Orbit Satellite Networks," *Proceedings of IEEE GLOBECOM*, (November 1995) : 529-535.
- [3] C-C. Shen, G. Borkar, S. Rajagopalan, and C. Jaikao, "Interrogation-Based Relay Routing for Ad Hoc Satellite Networks," *IEEE, Global Telecommunications Conference*, vol.3, November 2002, pp. 2920-2924.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย