รหัสโครงการ SUT1-104-46-12-51

รายงานการวิจัย

การหาเส้นทางโดยมีเงื่อนไขบนเครือข่าย MPLS
(Constraint based routing for MPLS Networks)

คณะผู้วิจัย

หัวหน้าโครงการ

อาจารย์ ปิยาภรณ์ กระฉอดนอก

สาขาวิชาวิศวกรรมโทรคมนาคม

สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี ปีงบประมาณ พ.ศ. 2546

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

มิถุนายน 2548

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สามารถดำเนินการได้ และได้รับผลสำเร็จบรรลุตามวัตถุประสงค์ที่ตั้งไว้ทุกประการ โดยได้รับทุนอุดหนุนการวิจัย ปีงบประมาณ 2546 จากมหาวิทยาลัยเทคโนโลยีสุรนารี และขอขอบคุณ คุณอภิศักดิ์ สัณธนะและคุณวิศิษศักดิ์ เสงี่ยมศักดิ์ ที่ให้คำแนะนำที่เป็นประโยชน์

บทคัดย่อ

งานวิจัยนี้นำเสนอวิธีการหาเส้นทางแบบออนไลน์ที่มีการรับประกันแบนด์วิดท์ในโครงข่ายการสวิตช์ลาเบลหลายโปรโตคอล (Multi-Protocol Label Switching network) หรือโครงข่ายเอ็มพีแอลเอส (MPLS) ซึ่งมีชื่อเรียกว่า วิธีการหาเส้นทางที่มีแบนด์วิดท์คงเหลือและความจุของลิงค์สูง – มีจำนวนโฟลทั้งหมดต่ำ (Maximize Residual bandwidth and link Capacity – Minimize total Flows) หรือ MaxRC-MinF วิธีการหาเส้นทางที่นำเสนอนี้สามารถจัดอยู่ในกลุ่มของปัญหาการหาเส้นทางแบบมีเงื่อนไขบนลิงค์ (link-constrained routing) และมีเงื่อนไขบนเส้นทาง (path-constrained routing) ในการออกแบบมีเป้าหมายหลัก 3 ประการ คือ การลดระดับการรบกวนกันระหว่างคู่โหนด การรักษาความสมดุลของทราฟฟิกโหลดให้กระจายไปบนเส้นทางที่มีการใช้งานต่ำ และการสงวนแบนด์วิดท์คงเหลือของโครงข่ายสำหรับรองรับทราฟฟิกที่จะร้องขอในอนาคต ในตอนท้ายได้มีการเปรียบเทียบประสิทธิภาพของวิธีการ MaxRC-MinF กับวิธีการอื่นๆ ที่ได้เคยมีผู้นำเสนอ พบว่าวิธีการ MaxRC-MinF สามารถลดความน่าจะเป็นในการถูกปฏิเสธการร้องขอเส้นทาง (rejection probability) เพิ่มค่าวิสัยสามารถรวม (total throughput) เพิ่มปริมาณการใช้งานสูงสุด (maximum link utilization) และการใช้งานเฉลี่ยของลิงค์ (average link utilization) อย่างไรก็ตามเนื่องด้วยความซับซ้อนในการคำนวณทำให้วิธีการที่นำเสนอใช้เวลาในการคำนวณของหน่วยประมวลผลกลาง (CPU calculation time) สูงกว่าวิธีการอื่น

Abstract

This research proposes an on-line routing algorithm for bandwidth-based guaranteed tunnels in the Multi-Protocol Label Switching (MPLS) networks, called the Maximize Residual bandwidth and link Capacity – Minimize total Flows (MaxRC-MinF) routing algorithm. The proposed algorithm can be categorized into link-constrained and path-constrained routing problems. It is based on three objectives: minimizing the interference level among ingress-egress pairs, balancing the traffic load over under-utilized paths, and trying to reserve bandwidth for future request. Finally, we have compared the performance of the MaxRC-MinF algorithm with other previously proposed algorithms. We found that the MaxRC-MinF algorithm achieves lower rejection probability and higher total throughput, maximum and average link utilization. However, because of its computational complexity, the proposed algorithm has a few higher CPU calculation time.

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญภาพ	ฉ
บทที่ 1 บทนำ	
ความสำคัญและที่มาของปัญหาการวิจัย	1
วัตถุประสงค์ของการวิจัย	2
ระเบียบวิธีวิจัย	2
ขอบเขตของการวิจัย	2
ประโยชน์ที่ได้รับจากการวิจัย	2
บทที่ 2 วรรณกรรมปริทัศน์	
ทฤษฎีเกี่ยวกับโครงข่ายเอ็มพีแอลเอส	3
งานวิจัยที่เกี่ยวข้อง	8
บทที่ 3 วิธีการที่นำเสนอ	
เป้าหมายในการออกแบบ	12
วิธีการออกแบบ	12
ขั้นตอนของวิธีการที่นำเสนอ	14
บทที่ 4 ผลการทดสอบ	
โปรแกรมจำลองแบบ	15
ข้อกำหนดในการจำลองแบบ	15
ผลการจำลองโครงข่าย NET1	16
วิเคราะห์ผลการจำลองโครงข่าย NET1	30
ผลการจำลองโครงข่าย NET2	33
วิเคราะห์ผลการจำลองโครงข่าย NET2	48

สารบัญ (ต่อ)

หน้า

บทที่ 5 บทสรุป

สรุปผลการวิจัย 52

ข้อเสนอแนะ 52

บรรณานุกรม 53

ประวัติผู้วิจัย 55

สารบัญภาพ

หน้า

ภาพที่ 1 MPLS Architecture 3

ภาพที่ 2 ตัวอย่างการสร้างเส้นทาง LSP และการส่งข้อมูลบนโครงข่าย MPLS 4

ภาพที่ 3 โครงสร้างข้อมูลของลาเบล 6

ภาพที่ 4 การผนวกลาเบลเข้ากับแพ็กเก็ตของโครงข่ายแบบต่างๆ 6

ภาพที่ 5 การจัดสรรลาเบล 6

ภาพที่ 6 ขั้นตอนการทำงานของวิธีการที่นำเสนอ 14

ภาพที่ 7 โฟลว์ชาร์ตของโปรแกรมจำลองแบบ 16

ภาพที่ 8 โครงข่าย NET1 17

ภาพที่ 9 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4 17

ภาพที่ 10 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4 18

ภาพที่ 11 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4 18

ภาพที่ 12 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4 19

ภาพที่ 13 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4 19

ภาพที่ 14 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15 20

ภาพที่ 15 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15 20

ภาพที่ 16 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15 21

ภาพที่ 17 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15 21

ภาพที่ 18 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15 22

ภาพที่ 19 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15 22

ภาพที่ 20 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15 23

ภาพที่ 21 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15 23

ภาพที่ 22 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15 24

ภาพที่ 23 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15 24

ภาพที่ 24 Rejection Probability ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4 25

ภาพที่ 25 Total Throughput ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4 25

ภาพที่ 26 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4 26

ภาพที่ 27 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4 26

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 28 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4 27
ภาพที่ 29 Rejection Probability ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5 27
ภาพที่ 30 Total Throughput ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5 28
ภาพที่ 31 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5 28
ภาพที่ 32 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5 29
ภาพที่ 33 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5 29
ภาพที่ 34 Rejection Probability โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1 31
ภาพที่ 35 Total Throughput โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1 31
ภาพที่ 36 Maximum Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1 32
ภาพที่ 37 Average Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1 32
ภาพที่ 38 CPU Calculation Time โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1 33
ภาพที่ 39 โครงข่าย NET2 34
ภาพที่ 40 Rejection Probability ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 35
ภาพที่ 41 Total Throughput ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 36
ภาพที่ 42 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 36
ภาพที่ 43 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 37
ภาพที่ 44 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 37
ภาพที่ 45 Rejection Probability ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 38
ภาพที่ 46 Total Throughput ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 38
ภาพที่ 47 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 39
ภาพที่ 48 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 39
ภาพที่ 49 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 40
ภาพที่ 50 Rejection Probability ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 40
ภาพที่ 51 Total Throughput ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 41
ภาพที่ 52 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 41

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 53 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 42
ภาพที่ 54 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 42
ภาพที่ 55 Rejection Probability ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 43
ภาพที่ 56 Total Throughput ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 43
ภาพที่ 57 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 44
ภาพที่ 58 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 44
ภาพที่ 59 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 45
ภาพที่ 60 Rejection Probability ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 45
ภาพที่ 61 Total Throughput ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 46
ภาพที่ 62 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 46
ภาพที่ 63 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 47
ภาพที่ 64 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 47
ภาพที่ 65 Rejection Probability โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2 49
ภาพที่ 66 Total Throughput โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2 49
ภาพที่ 67 Maximum Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2 50
ภาพที่ 68 Average Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2 50
ภาพที่ 69 CPU Calculation Time โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2 51

บทที่ 1
บทนำ

1. ความสำคัญ ที่มาของปัญหาที่ทำการวิจัย

ปัจจุบันการติดต่อสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต (WWW) เข้ามามีบทบาทสำคัญต่อการดำเนินชีวิตมาก เช่น การโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต (VoIP) ไปยังต่างประเทศ การโทรศัพท์ภายในประเทศโดยใช้หมายเลข 1234 การรับชมภาพวิดีโอผ่านเครือข่ายอินเทอร์เน็ต การโอนเงิน การตรวจสอบยอดบัญชีของลูกค้าของธนาคารผ่านอินเทอร์เน็ต การจองตั๋วเครื่องบิน การค้นคว้าข้อมูล และการส่ง E-Mail เป็นต้น ซึ่งการติดต่อสื่อสารภายในประเทศไทยในลักษณะนี้มีการขยายตัวเพิ่มขึ้นอย่างรวดเร็ว ทำให้เกิดความแออัดหรือความคับคั่งของข้อมูลในเครือข่าย ซึ่งส่งผลให้เครือข่ายเกิดความขัดข้อง เกิดความล่าช้าหรือการประวิงเวลาและการสูญหายของข้อมูลขึ้น สร้างความเสียหายให้กับธุรกิจภายในประเทศ

เครือข่ายอินเทอร์เน็ตนั้นใช้การส่งข้อมูลโดยพิจารณาจากหมายเลข IP ของต้นทางและปลายทางซึ่งมีความยาวมาก และยังใช้วิธีการตรวจสอบข้อผิดพลาด (Error Correction) ของข้อมูลในทุกสวิตช์ที่ผ่าน [1] จึงทำให้เกิดความล่าช้าหรือการประวิงเวลา ถ้ามีข้อมูลส่งผ่านสวิตช์แต่ละตัวจำนวนมาก จะส่งผลให้การประวิงเวลามีมากขึ้น ทำให้ข้อมูลปลายทางขาดความต่อเนื่อง ไม่ชัดเจน ปัญหาเหล่านี้แก้ไขได้โดยใช้เครือข่าย MPLS [2] ซึ่งเป็นเครือข่ายที่จะช่วยให้การจัดการทราฟฟิก (Traffic Management) คุณภาพการให้บริการ (Quality of Service) และเครือข่ายส่วนตัวเสมือน (Virtual Private Network) สามารถใช้งานได้ง่ายยิ่งขึ้นในเครือข่ายอินเทอร์เน็ต (IP) และเครือข่ายอื่นๆ ในอนาคต [3] การหาเส้นทางที่เหมาะสมในการส่งผ่านข้อมูลบนเครือข่าย MPLS มีความสำคัญเป็นอย่างยิ่ง เช่น หากใช้วิธีการหาระยะทางที่สั้นที่สุดในการส่งผ่านข้อมูล จะทำให้เกิดความแออัดของข้อมูลในเส้นทางที่เลือกเท่านั้น และส่งผลให้บางเส้นทางซึ่งไม่ถูกเลือกมีแบนด์วิดท์เหลือ ประสิทธิภาพของเครือข่ายจะลดลง [4] ถ้าพิจารณาประเภทข้อมูลจะเห็นว่าสัญญาณเสียงและวิดีโอต้องการความเร่งด่วนในการส่งมากกว่า E-Mail [5] เป็นต้น สำหรับงานวิจัยนี้จะพิจารณาเงื่อนไขต่างๆในการหาเส้นทาง (Constraint based routing) ได้แก่ ระยะทางที่สั้นที่สุดหรือจำนวน Hop [4] ราคา (Cost) แบนด์วิดท์ (Bandwidth) ความล่าช้า (Delay) และการรบกวน (Interference) จากข้อมูลแหล่งอื่น ฯลฯ บนเครือข่าย MPLS เพื่อให้ได้เส้นทางการส่งข้อมูลผ่านเครือข่ายมีความเหมาะสมกับประเภทข้อมูล มีระยะทางสั้น ประหยัดเวลาในการส่งข้อมูล และลดค่าใช้จ่าย

2. วัตถุประสงค์ของโครงการวิจัย

2.1 เพื่อเป็นพื้นฐานสำหรับงานวิจัยที่เกี่ยวข้อง รวมทั้งเป็นข้อมูลความรู้ให้แก่บริษัทที่มีความเกี่ยวข้องกับการออกแบบและผลิตสวิตซ์ (Switch), เราเตอร์ (Router) และการติดตั้งระบบโครงข่ายคอมพิวเตอร์และการสื่อสาร

2.2 เพื่อปรับปรุงและพัฒนาระบบการสื่อสารผ่านเครือข่ายอินเตอร์เน็ต ให้มีประสิทธิภาพ (Efficiency) เพิ่มขึ้น

3. ระเบียบวิธีวิจัย

12.1. ศึกษากลไกการทำงานของเครือข่ายอินเตอร์เน็ต

12.2. ศึกษามาตรฐานและทฤษฎีต่างๆ ของ MPLS และส่วนที่เกี่ยวข้อง

12.3. ศึกษาวิธีการเลือกเส้นทางในเครือข่ายแบบต่างๆ

12.4. หาวิธีการปรับปรุงการเลือกเส้นทางในเราเตอร์

12.5. เขียนโปรแกรมสร้างแบบจำลอง เพื่อนำมาทดสอบกับกลไกที่ปรับปรุงขึ้นมา

12.6. วิเคราะห์และเปรียบเทียบผลการทดสอบจากแบบจำลอง

12.7. ตรวจสอบและแก้ไขข้อผิดพลาด

12.8. สรุปผล จัดทำรายงานวิจัยและบทความวิชาการ

4. ขอบเขตของการวิจัย

งานวิจัยนี้มุ่งเน้นไปถึงการออกแบบวิธีการการเลือกเส้นทางส่งข้อมูลในเครือข่ายอินเตอร์เน็ต โดยจะทำการศึกษาและออกแบบโดยพิจารณาจากพารามิเตอร์หลายๆอย่าง ซึ่งมีผลกระทบกับข้อมูลเพื่อให้ได้ข้อมูลปลายทางที่มีความถูกต้อง และใช้ทรัพยากรต่ำ

5. ประโยชน์ที่จะได้รับจากการวิจัย

เป็นองค์ความรู้ในการวิจัยต่อไป และเป็นบริการความรู้แก่ภาคธุรกิจ ทำให้การสื่อสารมีความสะดวก สามารถส่งข่าวสารทุกชนิดบนเครือข่ายอินเตอร์เน็ตอย่างเดียวได้โดยไม่เกิดการสูญหายของข้อมูล และปลายทางได้รับข้อมูลตามที่ต้องการโดยมีคุณภาพสัญญาณชัดเจน ถูกต้อง

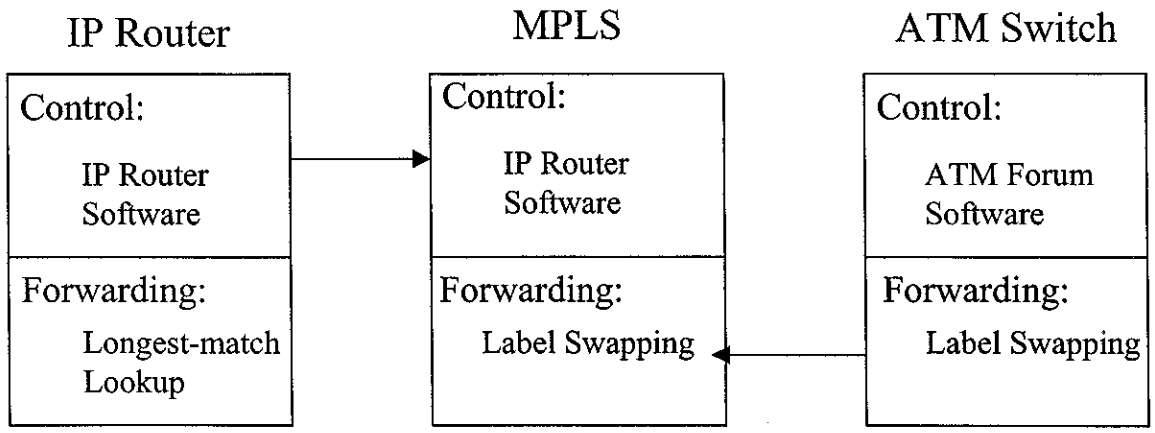
บทที่ 2
วรรณกรรมปริทัศน์

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการหาเส้นทางหรือ LSP บนโครงข่าย MPLS ซึ่งได้เคยมีผู้เสนอมาแล้ว [6]-[14] เพื่อใช้แก้ปัญหา ได้แก่วิธีการ Minimum Hop (MH) routing, Widest Shortest Path (WSP) routing, Dynamic Link Weight (DLW) algorithm, Wisitsak's algorithm (WSS) and Fabio's Link Weight algorithm (WM)

1. ทฤษฎีเกี่ยวกับโครงข่ายเอ็มพีแอลเอส

1.1 โครงข่ายเอ็มพีแอลเอส

โครงข่ายการสวิตช์ลาเบลได้หลายโปรโตคอล (MultiProtocol Label Switching หรือ MPLS) หรือเอ็มพีแอลเอส [1]-[5] เป็นโครงข่ายที่กำลังได้รับการสนใจในการพัฒนาการวิจัยอยู่ในปัจจุบัน ข้อดีของโครงข่ายเอ็มพีแอลเอสนี้ คือสามารถเชื่อมต่อระหว่างโปรโตคอลชั้นที่สองและชั้นที่สามใดๆ บนโมเดลโอเอสไอ (OSI model) ได้โดยไม่ต้องมีการแปลหรือจับคู่ (Mapping) ระหว่างตำแหน่งของโปรโตคอลชั้นที่สองและสาม ทำให้ลดความซับซ้อนในการทำงานระหว่างชั้นของโปรโตคอลลง และสามารถทำงานได้แม้ในโครงข่ายที่มีขนาดใหญ่ขึ้น (Scalability) เป็นการรวมเอา Control Plane ของ IP Router มาทำงานร่วมกับ Forwarding Plane ของ ATM Switch ดังภาพที่ 1



ภาพที่ 1 MPLS Architecture

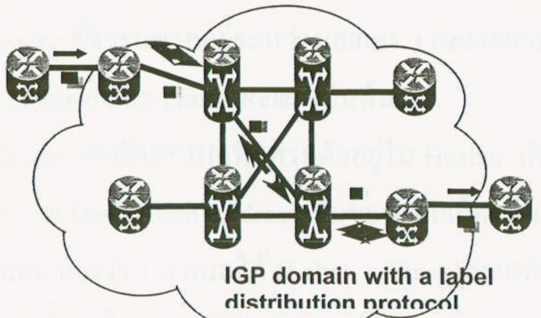
ปัจจุบันโปรโตคอลที่กำลังได้รับความสนใจในการพัฒนาบนโครงข่ายเอ็มพีแอลเอส คือ โปรโตคอลอินเตอร์เน็ต หรือ Internet Protocol (IP) ซึ่งเป็นโปรโตคอลในชั้นที่สาม ซึ่งถูกใช้งานบนโปรโตคอล Asynchronous Transfer Mode (ATM) ซึ่งเป็นโปรโตคอลในชั้นที่สอง เดิมนั้นจะต้องมีการ

แปลและจับคู่รหัสตำแหน่ง (Address Mapping) ระหว่างทั้งสองโปรโตคอลทำให้มีความซับซ้อนสูง แต่เมื่อนำสถาปัตยกรรมโครงข่ายเอ็มพีแอลเอสเข้ามาใช้ จะทำให้ช่วยลดความซับซ้อนนี้ลง โดยการนำซอฟแวร์ของ ATM เดิมออก แล้วนำซอฟแวร์การสวิตช์ลาเบลของเราท์เตอร์ (Label Switching Router, LSR) มาใช้งานแทน ทำให้สามารถใช้ฮาร์ดแวร์ของ ATM แต่ทำงานเสมือนเป็นเราท์เตอร์ตัวหนึ่ง ซึ่งสามารถเชื่อมต่อกับโปรโตคอล IP ได้โดยตรง โดย ATM Switch ที่มีการนำซอฟแวร์การสวิตช์ลาเบลมาใช้งาน จะมีชื่อใหม่เรียกว่า ATM-LSR นอกจากนี้เราท์เตอร์ที่อยู่บริเวณขอบของโครงข่ายจะมีชื่อเรียกว่า EDGE-LSR

การสร้างเส้นทางในโครงข่ายเอ็มพีแอลเอสจะต้องมีการสร้างเส้นทางการสวิตช์ลาเบล (Label Switched Path, LSP) ขึ้นมาก่อน แล้วจึงมีการส่งข้อมูลอย่างเรียงลำดับ (Connection oriented transmission) ไปบนเส้นทาง LSP โปรโตคอลที่ใช้หาเส้นทางเพื่อสร้าง LSP นี้ เรียกว่า MPLS Routing Protocol หรือ Label Distribution Protocol (LDP) โดยหลังจากการสร้างเส้นทาง LSP แล้ว LDP จะทำหน้าที่จัดสรรลาเบล (Label) หรือ Header ขนาดเล็กๆ ที่ใช้เป็นตัวบอกเส้นทางว่าแพ็กเก็ตข้อมูลนี้จะถูกส่งออกไปทางพอร์ตใด ซึ่งลาเบลจะถูกสลับไปเรื่อยๆ ในแต่ละ LSR จนถึงปลายทาง เทคนิคการส่งแพ็กเก็ตข้อมูลไปบนโครงข่าย MPLS นี้จะเรียกว่าการสลับลาเบล (Label Switching) ซึ่งคล้ายกับการส่งข้อมูลบนโครงข่าย ATM แต่การกำหนดลาเบลนี้จะไม่ได้ขึ้นอยู่กับหมายเลข IP ของแพ็กเก๊ตเท่านั้น แต่สามารถกำหนดได้จากหลายๆ ข้อกำหนดตามความต้องการของผู้บริหารโครงข่าย ได้แก่

1. หมายเลขโหนดปลายทาง (Destination Unicast address)
2. การวิศวกรรมทราฟฟิก (Traffic Engineering)
3. โครงข่ายเสมือนส่วนตัว (Virtual Private Network หรือ VPN) หรือ
4. กำหนดตามคุณภาพของการให้บริการ (Quality of Service หรือ QoS)

โดยการกำหนดหมายเลขลาเบลจะสัมพันธ์กับกลุ่มของแพ็กเก็ตเดียวกันหรือ Forward Equivalent Class (FEC) ในโครงข่าย IP



ภาพที่ 2 ตัวอย่างการสร้างเส้นทาง LSP และการส่งข้อมูลบนโครงข่าย MPLS

ขั้นตอนการส่งข้อมูลบนโครงข่าย MPLS ดังภาพที่ 2 มีดังนี้

1. ใช้โปรโตคอลหาเส้นทาง เช่น OSPF i-ISIS
2. ใช้ Label Distribution Protocol จัดสรรหมายเลขลาเบลลงบนเส้นทางที่สร้างไว้
3. เมื่อได้รับแพ็กเก็ตข้อมูลเข้ามาในโครงข่าย LSR ขาเข้า (Ingress LSR) จะตรวจสอบชนิดของแพ็กเก็ต และจัดสรรลาเบลให้กับแพ็กเก็ต จากนั้นแพ็กเก็ตที่ได้รับการกำหนดลาเบลแล้วจะถูกส่งไปในโครงข่าย
4. LSR ที่อยู่ภายในโครงข่าย หรือ Core LSR จะตรวจสอบลาเบลของแพ็กเก็ต ตรวจสอบว่าลาเบลขาเข้าตรงกับพอร์ตขาออกพอร์ตใด และลาเบลขาออกใด หลังจากนั้นลาเบลเดิมจะถูกเปลี่ยนเป็นลาเบลใหม่ (Label Switching) และแพ็กเก็ตถูกส่งต่อไปยังโหนดถัดไป
5. เมื่อแพ็กเก็ตถูกส่งมาถึง LSR ขาออก (Egress LSR) ลาเบลจะถูกถอดออกก่อนที่แพ็กเก็ตถูกส่งออกไปยังโครงข่ายอื่นนอกโครงข่าย MPLS

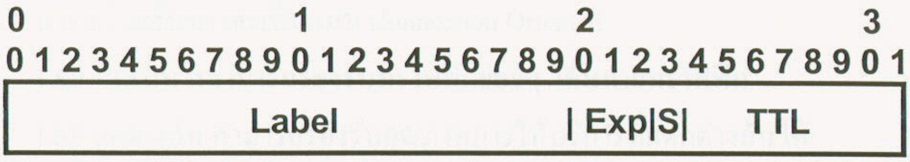
1.2 คุณลักษณะของลาเบล

ในหัวข้อนี้จะกล่าวถึงลักษณะของลาเบล (Label) ที่นำมาผนวกเข้ากับแพ็กเก็ตข้อมูล โดยลาเบลจะมีลักษณะดังนี้

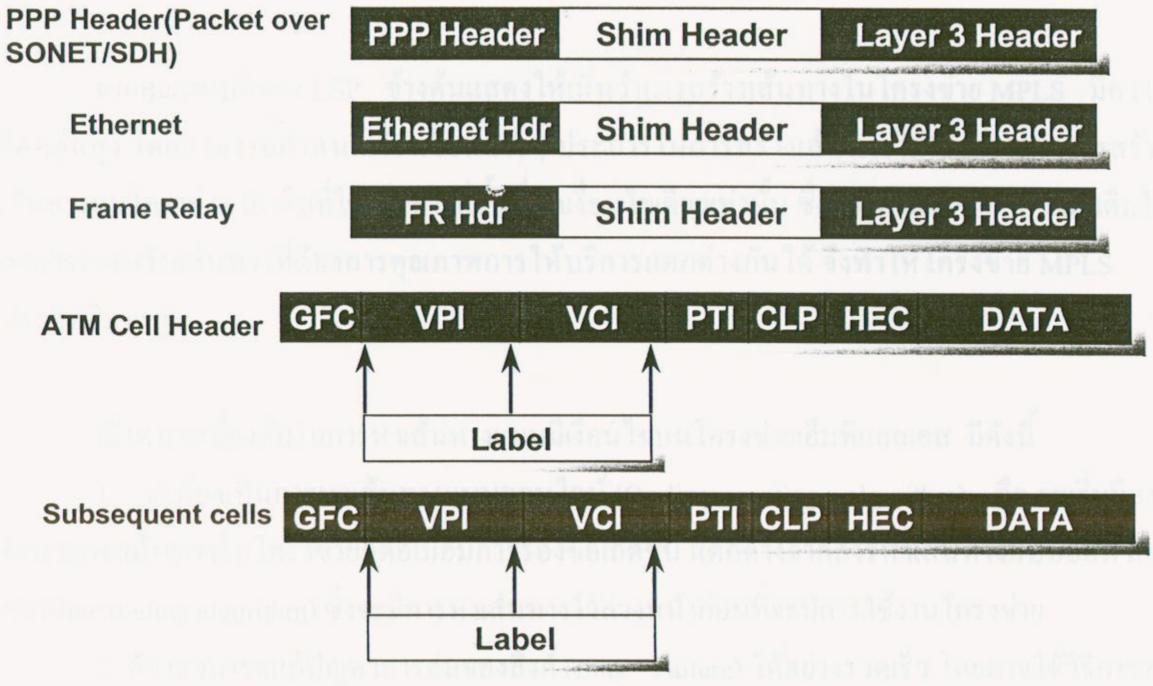
1. มีความยาวคงที่ เพื่อให้คำนวณและประมวลผลง่าย
2. มีขนาดเล็ก เพื่อให้ไม่เป็นการเพิ่ม Overhead ของการส่งข้อมูล
3. ขนาดของลาเบลจะเท่ากับ 32 บิต ดังภาพที่ 3 แบ่งเป็น
    a. ลาเบล 20 บิต
    b. Exp (Experimental) 3 บิต
    c. S (Bottom of Stack) 1 บิต
    d. TTL (Time To Live) 8 บิต
4. ตำแหน่งของการผนวกลาเบลกับแพ็กเก็ต ดังภาพที่ 4 แบ่งได้ 2 ประเภทคือ
    a. Shim Header ซึ่งจะแทรกอยู่ระหว่างเลเยอร์ 3 และเลเยอร์ 2 ของแพ็กเก็ต เช่น PPP Header, Ethernet และ Frame Relay เป็นต้น
    b. Label Header จะเป็นลาเบลที่แทรกตัวอยู่ใน Header เดิมของแพ็กเก็ตในเลเยอร์ 2 แทนที่บริเวณ Destination Address เดิมของแพ็กเก็ต เช่น ATM Cells
5. อนุญาตให้มีลาเบลมากกว่า 1 ลาเบลได้ (Label Stack) หากมีการส่งผ่านแพ็กเก็ตไปในโครงข่ายแบบลำดับขั้น หรือ Hierarchical Network
6. การจัดสรรลาเบลจะเป็นหน้าที่ของโหนดปลายทาง หรือ Egress LSR ไปบนเส้นทางที่โหนดขาเข้า (Ingress LSR) สร้างไว้ ดังตัวอย่างในภาพที่ 5 คือ

a. โหนด A ไปโหนด B ใช้ลาเบล 30 (แทนปลายทางของ Subnet 171.68.10)

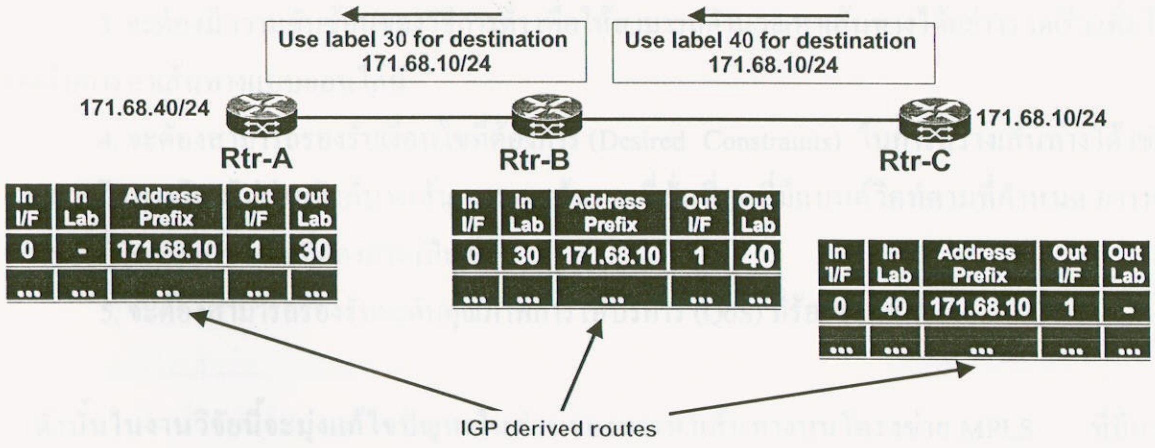
b. โหนด B ไปโหนด C ใช้ลาเบล 40 (แทนปลายทางของ Subnet 171.68.10)



ภาพที่ 3 โครงสร้างข้อมูลของลาเบล



ภาพที่ 4 การผนวกลาเบลเข้ากับแพ็กเก็ตของโครงข่ายแบบต่างๆ



ภาพที่ 5 การจัดสรรลาเบล

1.3 เส้นทางการสวิตช์ลาเบล

เส้นทางการสวิตช์ลาเบล (Label Switched Path หรือ LSP) เป็นเส้นทางที่ใช้ส่งแพ็กเก็ตข้อมูลจากโหนดต้นทางไปถึงโหนดปลายทาง โดยมีคุณลักษณะดังนี้

1. การสร้างเส้นทางจะเป็นแบบ Connection Oriented
2. LSP 1 เส้นทางสามารถรองรับทราฟฟิกย่อยๆ ได้หลายทราฟฟิก
3. LSP แต่ละเส้น สามารถรองรับคุณภาพการให้บริการที่แตกต่างกันได้
4. ในการสร้างเส้นทางสามารถสร้างโดยกำหนดเงื่อนไขบางอย่างได้ หรือสามารถนำเอาเทคโนโลยีการสร้างเส้นทางแบบมีเงื่อนไข (Constraint-Based Routing หรือ CBR) มาใช้ได้

จากคุณสมบัติของ LSP ข้างต้นแสดงให้เห็นว่าการสร้างเส้นทางในโครงข่าย MPLS มีความยืดหยุ่นสูง โดยสามารถกำหนดเงื่อนไขหลายๆ ประการในการสร้างเส้นทางได้ ซึ่งต่างจากการสร้างเส้นทางบนโครงข่าย IP เดิมที่ใช้เส้นทางที่สั้นที่สุดเงื่อนไขเดียวเท่านั้น ซึ่งทำให้โครงข่าย IP แบบเดิมไม่สามารถรองรับเส้นทางที่ต้องการคุณภาพการให้บริการแตกต่างกันได้ จึงทำให้โครงข่าย MPLS มีประสิทธิภาพสูง

เป้าหมายเบื้องต้นในการหาเส้นทางแบบมีเงื่อนไขบนโครงข่ายเอ็มพีแอลเอส มีดังนี้

1. จะต้องเป็นการหาเส้นทางแบบออนไลน์ (On-line routing algorithm) คือ จะเริ่มมีการคำนวณหาเส้นทางในโครงข่ายก็ต่อเมื่อมีการร้องขอเกิดขึ้น แตกต่างจากการหาเส้นทางแบบออฟไลน์ (Off-line routing algorithm) ซึ่งจะมีการหาเส้นทางไว้ล่วงหน้าก่อนที่จะมีการใช้งานโครงข่าย

2. ต้องสามารถแก้ปัญหาการล่มของลิงค์ (Link Failure) ได้อย่างรวดเร็ว โดยอาจใช้วิธีการหาเส้นทางซ้ำ (Rerouting) หรือใช้วิธีการสร้างเส้นทางสำรองไว้รองรับการล่มของลิงค์

3. จะต้องมีความซับซ้อนของวิธีการต่ำ เพื่อให้สามารถคำนวณหาเส้นทางได้อย่างรวดเร็ว เพื่อใช้รองรับการหาเส้นทางแบบออนไลน์

4. จะต้องสามารถรองรับเงื่อนไขที่ต้องการ (Desired Constraints) ในการสร้างเส้นทางได้ เช่น การหาเส้นทางโดยไม่ผ่านลิงค์บางเส้น การหาเส้นทางที่สั้นที่สุดที่มีแบนด์วิดท์ตามที่กำหนด การหาเส้นทางที่มีเวลาหน่วงตามที่ต้องการ เป็นต้น

5. จะต้องสามารถรองรับระดับคุณภาพการให้บริการ (QoS) ที่ร้องขอได้

ดังนั้นในงานวิจัยนี้จะมุ่งแก้ไขปัญหาในส่วนของการหาเส้นทางบนโครงข่าย MPLS ที่มีการรับประกันขนาดของแบนด์วิดท์ (Bandwidth Guaranteed Tunnels) โดยต้องการพัฒนาอัลกอริทึมในการหาเส้นทางแบบออนไลน์ (On-line Routing Algorithm) เพื่อสามารถหาเส้นทางได้อย่างมีประสิทธิภาพ

กล่าวคือ มีความน่าจะเป็นในการปฏิเสธการร้องขอเส้นทาง (Rejection Probability) ต่ำ มีค่าวิสัยสามารถ (Throughput) สูง มีค่าการใช้งานสูงสุดของลิงค์ (Maximum Link Utilization) ต่ำ มีค่าการใช้งานเฉลี่ยของลิงค์ (Average Link Utilization) สูง และมีค่าเวลาของหน่วยประมวลผลกลางในการคำนวณหาเส้นทาง (CPU Calculation Time) มีค่าต่ำ ซึ่งพารามิเตอร์ทั้ง 5 ชนิดสามารถคำนวณได้ดังนี้

$$Rejection\ Probaility = \frac{Succesful\ Requested\ Traffic}{Total\ Requested\ Traffic} \tag{1}$$

$$Total\ Throughput = \sum Successful\ Requested\ Bandwidth \tag{2}$$

$$Maximum\ Link\ Utilization = Max(Link\ Banwidth\ Usage) \tag{3}$$

$$Average\ Link\ Utilization = \frac{\sum(Banwidth\ Usage\ of\ Link)}{Total\ Number\ of\ Links}, \tag{4}$$

$$CPU\ Calculation\ Time = \frac{Total\ Simulation\ Time}{Total\ Number\ of\ Requested\ Traffic} \tag{5}$$

2. งานวิจัยที่เกี่ยวข้อง

2.1 วิธีการ Minimum Hop (MH) routing [11]

วิธีการหาเส้นทางที่สั้นที่สุด (Minimum Hop routing algorithm) หรือ MH มีขั้นตอนการทำงานดังนี้

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 หาเส้นทางที่สั้นที่สุด

ขั้นที่ 3 สร้างเส้นทางบนเส้นทางที่พบ หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ข้อดีของวิธีการ MH คือ เป็นวิธีการที่ง่าย ไม่มีความซับซ้อน สามารถทำงานได้อย่างรวดเร็ว

แต่มีข้อเสียคือ จะใช้เส้นทางที่สั้นที่สุดเท่านั้นเป็นเงื่อนไข และไม่มีการนำแบนด์วิดท์ที่ต้องการมาเป็นเงื่อนไขในการพิจารณา ทำให้ไม่สามารถสร้างเส้นทางได้ ถึงแม้ว่าจะมีเส้นทางอื่นๆ ที่ยังว่างอยู่ แต่เส้นทางเหล่านั้นไม่ใช้เส้นทางที่สั้นที่สุด

2.2 วิธีการ Widest Shortest Path (WSP) routing [11]

วิธีการเลือกเส้นทางที่มีกว้างที่สุดจากเส้นทางที่สั้นที่สุด (Widest Shortest Path routing algorithm) หรือ WSP มีขั้นตอนการทำงานดังนี้

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 คำนวณเส้นทางที่สั้นที่สุด (ฮอปต่ำที่สุด)

ขั้นที่ 3 คำนวณหาเส้นทางกว้างที่สุด หรือมีขนาดแบนด์วิดท์สูงที่สุด จากเส้นทางทั้งหมดในข้อ 2

ขั้นที่ 4 สร้างเส้นทางบนเส้นทางที่พบตามข้อ 3

ขั้นที่ 5 หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ข้อดีของวิธีการ WSP คือ เป็นวิธีการที่ง่าย มีความซับซ้อนในระดับต่ำแต่สูงกว่าวิธีการ MH สามารถทำงานได้อย่างรวดเร็ว สามารถหาเส้นทางที่กว้างที่สุดหรือมีแบนด์วิดท์สูงที่สุด จากระยะทางที่สั้นที่สุดออกมาได้ ทำให้มีการใช้งานในเส้นทางที่ว่างก่อน เพื่อให้สามารถรองรับการร้องขอในอนาคตได้

แต่มีข้อเสียคือ ในบางกรณีที่ปริมาณแบนด์วิดท์ที่ร้องขอเข้ามานั้นมีขนาดเล็ก แต่ในรอบต่อไปที่มีการร้องเข้ามาเป็นแบนด์วิดท์ขนาดใหญ่กว่า แต่ไม่สามารถสร้างเส้นทางได้เนื่องจากแบนด์วิดท์ไม่เพียงพอ เช่น

เส้นทางที่ 1 มีแบนด์วิดท์คงเหลือ 60 Mbps

เส้นทางที่ 2 มีแบนด์วิดท์คงเหลือ 20 Mbps ทั้งสองเส้นทางมีระยะฮอปเท่ากัน

หากมีการร้องขอการสร้างเส้นทางจำนวน 20 Mbps วิธีการ WSP จะเลือกเส้นทางที่ 1 ก่อน ทำให้เส้นทางที่ 1 มีแบนด์วิดท์คงเหลือ 40 Mbps หากหลังจากนั้นมีการร้องขอการสร้างเส้นทางขนาด 50 Mbps เข้ามา จะไม่สามารถทำได้เนื่องจากมีแบนด์วิดท์ไม่เพียงพอ

2.3 วิธีการ Dynamic Link Weight (DLW) [12]

วิธีการ Dynamic Link Weight (DLW) จะมีการนำเอาขนาดแบนด์วิดท์ที่คงเหลือ (Residual Bandwidth) ของแต่ละลิงค์ในโครงข่ายเข้ามาพิจารณา โดยต้องการให้มีกระจาย ทราฟฟิกไปบนแส้นทางที่มีแบนด์วิดท์คงเหลือสูงก่อน

DLW มีขั้นตอนการทำงานดังนี้

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 คำนวณหาค่าน้ำหนักโดยให้ $w = 1 + \frac{k}{R}$ เมื่อ $k = 0.1$ หรือ 10 เปอร์เซ็นต์

ขั้นที่ 3 ใช้น้ำหนักจากข้อ 2 เป็นน้ำหนักในการหาเส้นทางที่สั้นที่สุด

ขั้นที่ 4 สร้างเส้นทางบนเส้นทางที่พบตามข้อ 3

ขั้นที่ 5 หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ข้อดีของวิธีการ DLW คือ เป็นวิธีการที่สามารถกระจายทราฟฟิกไปบนโครงข่ายได้มากกว่าวิธีการ WSP เนื่องจากไม่มีข้อจำกัดที่ต้องใช้เส้นทางที่สั้นที่สุดเท่านั้น แต่สามารถใช้เส้นทางที่ยาวกว่าตามค่าน้ำหนักของลิงค์

DLW มีข้อเสียคือ ไม่มีการนำขนาดความจุของลิงค์มาพิจารณา และการรบกวนกันของคู่โหนดมาพิจารณา

2.4 วิธีการ Wisitsak's Algorithm (WSS) [13]

วิธีการ Wisitsak's Algorithm หรือ WSS เป็นวิธีการที่ใช้การคำนวณความน่าจะเป็นของการใช้งานลิงค์ล่วงหน้าของแต่ละคู่โหนดต้นทาง-ปลายทาง ดังนั้นการหาเส้นทางจะพยายามเลี่ยงการใช้งานเส้นทางที่มีความน่าจะเป็นของการใช้งานสูง เพื่อลดความคับคั่งของโครงข่าย

WSS มีขั้นตอนการทำงานแบ่งได้ 2 ช่วง คือ ช่วงการคำนวณน้ำหนักของลิงค์ (Link Weight) ก่อนการใช้งาน และช่วงการหาเส้นทาง

ช่วงที่ 1 การคำนวณน้ำหนักของลิงค์

ขั้นที่ 1 หาเส้นทาะที่เป็นไปได้ที่ไม่มีการซ้อนทับกันทั้งหมดระหว่างคู่โหนดจาก

ขั้นที่ 2 สร้างเส้นทางทั้งหมดที่พบโดยกำหนดแบนด์วิดท์ที่ร้องขอมีค่าเท่ากับ 1 หน่วย

ขั้นที่ 3 คำนวณคำนวณความน่าจะเป็นของการใช้งานลิงค์เท่ากับแบนด์วิดท์ที่ใช้บนลิงค์หารด้วยจำนวนแบนด์วิดท์รวมทั้งหมดของทุกลิงค์

<u>หมายเหตุ</u> การคำนวณในช่วงที่ 1 จะถูกใช้งานก็ต่อเมื่อรูปร่างของโครงข่าย (Network Topology) มีการเปลี่ยนแปลงไป หรือก่อนการใช้งานโครงข่าย เช่น เกิดการล่มของโหนดหรือลิงค์ในโครงข่าย เป็นต้น

ช่วงที่ 2 การหาเส้นทาง

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 หาเส้นทางที่มีผลรวมของความน่าจะเป็นของการใช้งานลิงค์ หรือน้ำหนักของลิงค์) ต่ำที่สุด เป็นน้ำหนักของลิงค์

ขั้นที่ 3 ในขั้นตอนที่ 2 หากพบมากกว่า 1 เส้นทางให้เลือกเส้นทางที่สั้นที่สุด (Shortest Path)

ขั้นที่ 4 ในขั้นตอนที่ 3 หากพบมากกว่า 1 เส้นทางให้เลือกเส้นทางที่มีแบนด์วิดท์คงเหลือ (Residual Bandwidth) มากที่สุด

ขั้นที่ 5 สร้างเส้นทางบนเส้นทางที่พบตามข้อ 4

ขั้นที่ 6 หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ข้อดีของวิธีการ WSS คือ เป็นวิธีการที่สามารถหลีกเลี่ยงเส้นทางที่มีความน่าจะเป็นในการถูกใช้งานสูง จึงเป็นการลดความคับคั่งและกระจายโหลดของโครงข่ายไปในเส้นทางอื่นๆ ที่มีความน่าจะเป็นในการใช้งานต่ำ นอกจากนี้วิธีการ WSS ยังมีความซับซ้อนต่ำกว่าวิธีการ MIRA เนื่องจากมีการคำนวณความน่าจะเป็นในการถูกใช้งานเพียง 1 ครั้งเดียวก่อนการใช้งาน หรือคำนวณเมื่อมีการเปลี่ยนแปลงของรูปร่างของโครงข่ายไป

แต่มีข้อเสียคือ ไม่ได้มีการนำแบนด์วิดท์ที่เหลือ และขนาดความจุของลิงค์มาพิจารณา

2.5 วิธีการ Fabio's Weight Function algorithm (WF) [14]

วิธีการ Fabio's Link Weight algorithm (WF) เป็นวิธีการที่นำแบนด์วิดท์ที่เหลือ และขนาดความจุของลิงค์มาคำนวณหาน้ำหนักของลิงค์ เพื่อใช้คำนวณหาเส้นทางที่มีแบนด์วิดท์เหลือมากที่สุด และมีขนาดความจุของลิงค์มากที่สุดมาใช้งานก่อน เพื่อลดความคับคั่ง

WM มีขั้นตอนการทำงานดังนี้

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 คำนวณค่าน้ำหนักของลิงค์โดยนำขนาดความจุของลิงค์หารด้วยแบนด์วิดท์ที่เหลือ

ขั้นที่ 3 ใช้น้ำหนักของลิงค์จากข้อ 2 ในการหาเส้นทางที่สั้นที่สุด

ขั้นที่ 4 สร้างเส้นทางบนเส้นทางที่พบตามข้อ 3

ขั้นที่ 5 หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ข้อดีของวิธีการ WM คือ เป็นวิธีการที่สามารถหลีกเลี่ยงเส้นทางที่พยายามกระจาย ทราฟฟิกไปเส้นทางที่มีการใช้งานต่ำและมีขนาดความจุของลิงค์สูงๆ ก่อน ทำให้ลดความคับคั่งของโครงข่าย

แต่มีข้อเสียคือ ไม่ได้นำข้อมูลของคู่โหนดมาคำนวณในการหาเส้นทาง

บทที่ 3
วิธีการที่นำเสนอ

ในบทนี้จะกล่าวถึงเป้าหมายในการออกแบบ แนวความคิดในการแก้ปัญหา และรายละเอียดของวิธีการที่ได้นำเสนอ

1. เป้าหมายในการออกแบบ

จากบทที่ 2 จะพบว่าแนวทางของการออกแบบวิธีการหาเส้นทางแบบมีเงื่อนไขสำหรับโครงข่าย MPLS ที่ได้มีผู้นำเสนอมาแล้วนั้น สามารถสรุปได้ดังนี้

1. ต้องการลดการรบกวนกันระหว่างคู่โหนด โดยการกระจายทราฟฟิกไปในเส้นทาง ที่มีการรบกวนต่ำ เพื่อให้มีแบนด์วิดท์สำหรับคู่โหนดมากพอที่จะรองรับทราฟฟิกที่จะเกิดขึ้นในอนาคต
2. นำแบนด์วิดท์ที่เหลือในปัจจุบัน และขนาดความจุของลิงค์มาพิจารณา เพื่อต้องการใช้เส้นทางที่มีขนาดแบนด์วิดท์ที่เหลือและขนาดความจุของลิงค์สูงมาใช้งานก่อน เพื่อเป็นการกระจายทราฟฟิกไปในเส้นทางที่มีการใช้งานต่ำ
3. พยายามใช้เส้นทางที่สั้นที่สุดก่อน เนื่องจากเป็นการใช้ทรัพยากรโครงข่ายต่ำที่สุด ทำให้มีความน่าจะเป็นที่จะคงเหลือแบนด์วิดท์ได้มาก จึงสามารถรองรับปริมาณทราฟฟิกได้สูง

2. วิธีการออกแบบ

เพื่อให้บรรลุเป้าหมายในการออกแบบทั้ง 3 ประการข้างต้น งานวิจัยนี้จึงขอนำเสนอแนวทางในการแก้ปัญหาดังนี้

1. ใช้การบันทึกจำนวนโฟลหรือทราฟฟิกทั้งหมดบนลิงค์ในปัจจุบัน เพื่อใช้แสดงระดับของการใช้งานในลิงค์นั้นๆ วิธีการที่นำเสนอจะเลี่ยงการใช้งานบนลิงค์ที่มีจำนวนโฟลสูง เพื่อให้บรรลุเป้าหมายในข้อที่ 1
2. นำแบนด์วิดท์ที่เหลือในปัจจุบัน และขนาดความจุของลิงค์มาพิจารณา เพื่อให้บรรลุเป้าหมายในข้อที่ 2
3. หากพบว่ามีเส้นทางมากกว่า 1 เส้นทางที่เป็นตามเงื่อนไข 2 ข้อแรกคือเป็นเส้นทางที่มีจำนวนโฟลต่ำที่สุด มีแบนด์วิดท์คงเหลือมากที่สุด และมีขนาดความจุของลิงค์สูงสุด ให้เลือกเส้นทางที่สั้นที่สุดหรือมีจำนวน Hop ต่ำที่สุดเพื่อให้บรรลุเป้าหมายในข้อที่ 3

กำหนดให้

G(N,L) : กราฟของโครงข่าย

N : เซ็ตของโหนดบนโครงข่าย

L : เซ็ตของลิงค์บนโครงข่าย

S : โหนดต้นทาง เมื่อ $S \in N$

D : โหนดปลายทาง เมื่อ $D \in N$

{S,D} : เซ็ตของคู่โหนดต้นทาง-ปลายทาง

F_j : จำนวนโฟลทั้งหมดบนลิงค์ j เมื่อ $j \in L$

R_j : ปริมาณแบนด์วิดท์คงเหลือบนลิงค์ j เมื่อ $j \in L$

C_j : ความจุทั้งหมดของลิงค์ j เมื่อ $j \in L$

w_j : น้ำหนักของลิงค์ j เมื่อ $j \in L$

$W_{\{S,D\}}$: น้ำหนักของเส้นทางระหว่างคู่โหนด {S,D}

จะสามารถหาน้ำหนักของลิงค์ j ได้จากสมการที่ 6 ดังนี้

$$w_j = \frac{F_j}{R_j C_j} \tag{6}$$

ลิงค์ที่มีค่า w_j สูง หมายถึงลิงค์ที่มีจำนวนโฟลสูง (มีโอกาสในการรบกวนสูง) มีแบนด์วิดท์คงเหลือต่ำ และมีความจุของลิงค์ต่ำ ในทางตรงข้ามลิงค์ที่มีค่า w_j ต่ำ หมายถึงลิงค์ที่มีจำนวนโฟลต่ำ (มีโอกาสในการรบกวนต่ำ) มีแบนด์วิดท์คงเหลือสูง และมีความจุของลิงค์สูง ดังนั้นลิงค์ที่ควรเลือกคือลิงค์ที่มีค่า w_j ต่ำ

จากนั้นสามารถหาน้ำหนักของเส้นทางได้จากสมการที่ 7

$$W_{\{s,d\}} = \sum_{j \in L_{\{S,D\}}} w_j \tag{7}$$

เงื่อนไขสำหรับการหาเส้นทาง คือ การเลือกเส้นทางที่มีน้ำหนักของเส้นทางระหว่างคู่โหนด {S,D} มีค่าต่ำ ดังสมการที่ 8

$$\min\left(W_{\{s,d\}}\right) \tag{8}$$

ในกรณีที่พบเส้นทางที่มีค่า $W_{\{s,d\}}$ มากกว่า 1 เส้นทางให้เลือกใช้เส้นทางที่สั้นที่สุดหรือมีจำนวน Hop ต่ำที่สุดก่อน

3. ขั้นตอนของวิธีการที่นำเสนอ

สามารถสรุปวิธีการที่นำเสนอได้ดังภาพที่ 6

วิธีการที่นำเสนอ

ขั้นที่ 1 ตัดเส้นทางที่มีแบนด์วิดท์ไม่เพียงพอออก

ขั้นที่ 2 คำนวณค่าน้ำหนักของลิงค์ w_j

ขั้นที่ 3 ใช้น้ำหนักของลิงค์จากข้อ 2 ในการหาเส้นทางที่สั้นที่สุด หรือ $\min(W_{\{s,d\}})$

ขั้นที่ 4 สร้างเส้นทางบนเส้นทางที่พบตามข้อ 3

ขั้นที่ 5 หากไม่สามารถสร้างได้ถือว่าเกิดการปฏิเสธการสร้างเส้นทาง

ภาพที่ 6 ขั้นตอนการทำงานของวิธีการที่นำเสนอ

เนื่องจากวิธีการที่นำเสนอใช้หลักการหาเส้นทางที่สั้นที่สุด ที่มีการรบกวนต่ำ มีแบนด์วิดท์คงเหลือสูง และมีความจุของลิงค์สูง จึงขอตั้งชื่อวิธีการที่นำเสนอนี้ว่า Maximize Residual bandwidth and link Capacity – Minimize total Flow (MaxRC-MinF) routing algorithm

บทที่ 4

ผลการทดสอบ

ในบทนี้จะเป็นการนำวิธีการที่ได้นำเสนอในบทที่ 3 หรือวิธีการ MaxRC-MinF ทำการทดสอบวัดประสิทธิภาพ ได้แก่ ความน่าจะเป็นในการปฏิเสธการร้องขอเส้นทาง (Rejection Probability) ค่าวิสัยสามารถ (Throughput) ค่าการใช้งานสูงสุดของลิงค์ (Maximum Link Utilization) ค่าการใช้งานเฉลี่ยของลิงค์ (Average Link Utilization) และค่าเวลาของหน่วยประมวลผลกลางในการคำนวณหาเส้นทาง (CPU Calculation Time) ดังที่กล่าวไว้ในบทที่ 2 (สมการที่ 1 – 5)

1. โปรแกรมจำลองแบบ

งานวิจัยนี้ได้พัฒนาโปรแกรมจำลองแบบการหาเส้นทางบนโปรแกรม MATLAB 6.5 โดยได้รันบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่มีคุณลักษณะดังนี้

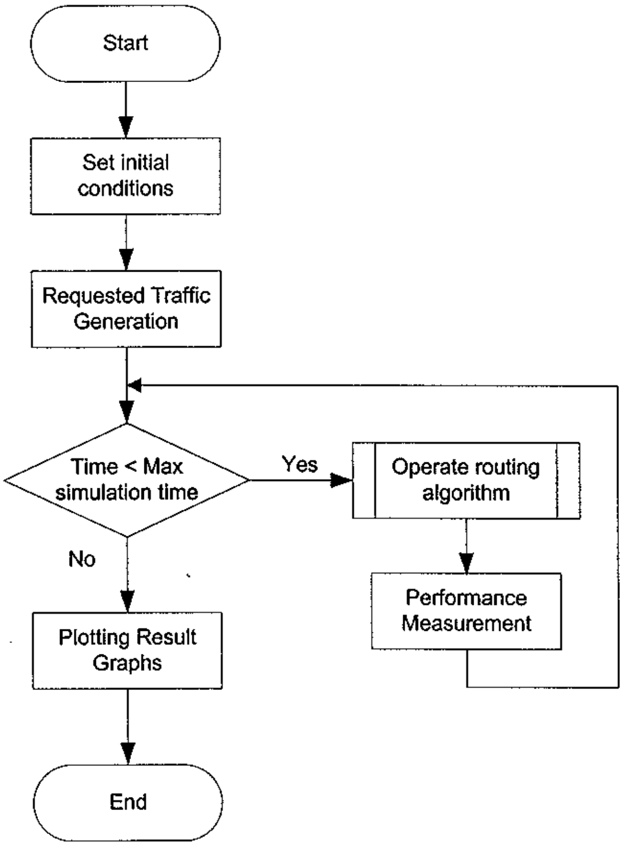
1. ระบบปฏิบัติการ Microsoft Windows XP SP1
2. หน่วยประมวลผลกลาง Pentium 4 ความเร็ว 1.5 GHz
3. RAM 256 MB
4. Hard drive 60 GB

ในภาพที่ 7 จะแสดงโฟลว์ชาร์ตของโปรแกรมจำลองแบบที่ได้พัฒนาขึ้น

2. ข้อกำหนดในการจำลองแบบ

ในการจำลองแบบมีข้อกำหนดดังนี้

1. โครงข่ายที่นำมาทดสอบมี 2 โครงข่าย คือ NET1 และ NET2
2. ลิงค์ทั้งหมดเป็นชนิด bi-directional link
3. ลิงค์สีเข้มมีความจุ 480 หน่วย (แทน OC-48)
4. ลิงค์สีจางมีความจุ 120 หน่วย (แทน OC-12)
5. ทราฟฟิกมีการร้องขอแบนด์วิดท์ขนาด 1-3 หน่วย เกิดจากการสุ่มโดยการกระจายอย่างมีรูปแบบ (Uniformly Distribution)
6. ปริมาณทราฟฟิกจะเพิ่มขึ้นจาก 0 – 1600 connections/seconds
7. การจำลองจะมีการสุ่มจำนวนคู่โหนด 4 และ 5 คู่ สำหรับโครงข่าย NET1 และ NET2 ตามลำดับ ซึ่งจะทำการจำลอง 5 ครั้ง แล้วหาค่าเฉลี่ย



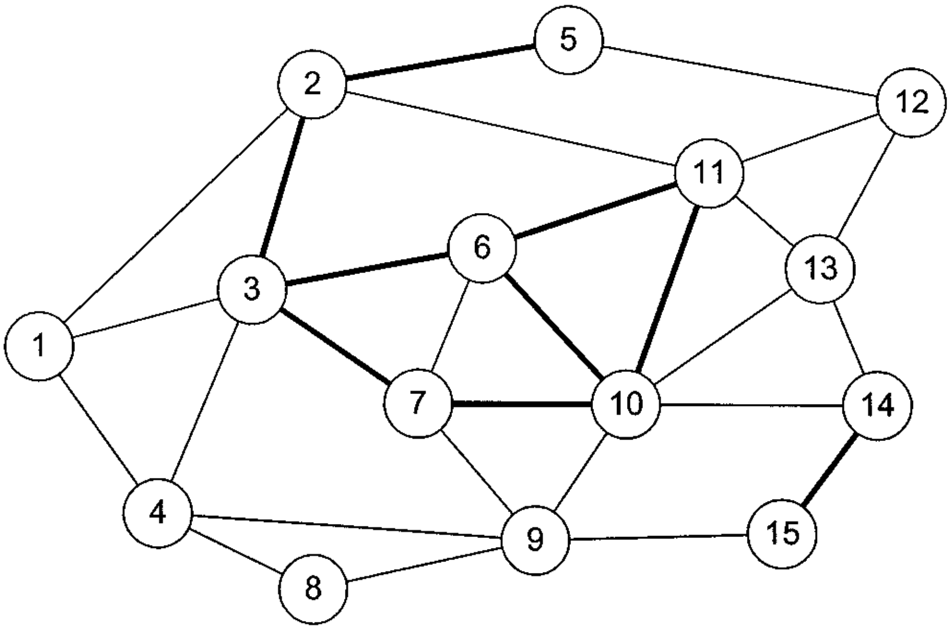
ภาพที่ 7 โฟลว์ชาร์ตของโปรแกรมจำลองแบบ

3. ผลการจำลองโครงข่าย NET1

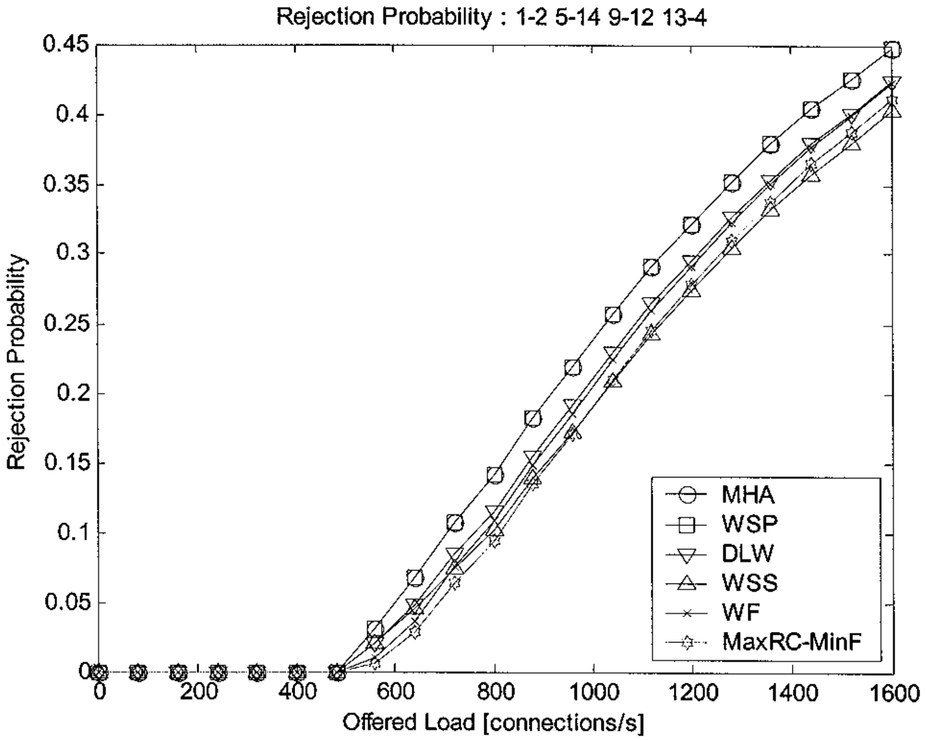
ภาพที่ 8 แสดงโครงข่าย NET1 ซึ่งเป็นโครงข่ายเดียวกันกับ [3] ซึ่งประกอบด้วย 15 โหนด และ 28 ลิงค์ การทดสอบจะมีการสุ่มเลือกคู่โหนด 4 คู่ จำนวน 5 ครั้ง ได้แก่

1. 1-2, 5-14, 9-12 และ 13-4
2. 1-5, 2-14, 4-12 และ 5-15
3. 1-13, 4-2, 5-9 และ 5-15
4. 2-12, 5-9, 8-13 และ 9-4
5. 5-8, 9-2, 12-4 และ 13-5

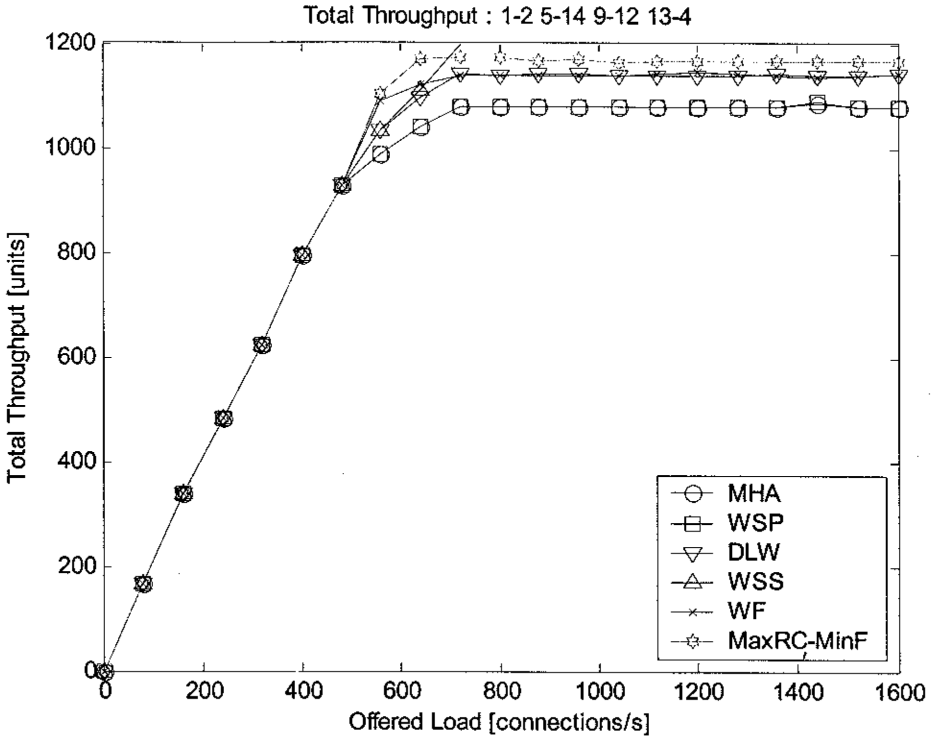
ซึ่งมีผลการจำลองดังนี้



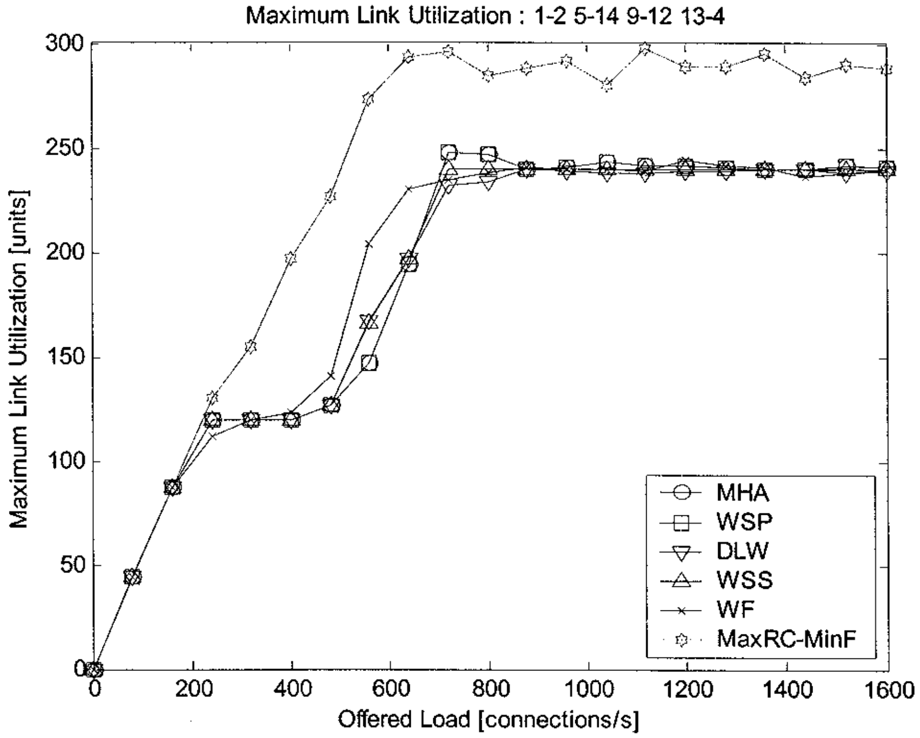
ภาพที่ 8 โครงข่าย NET1



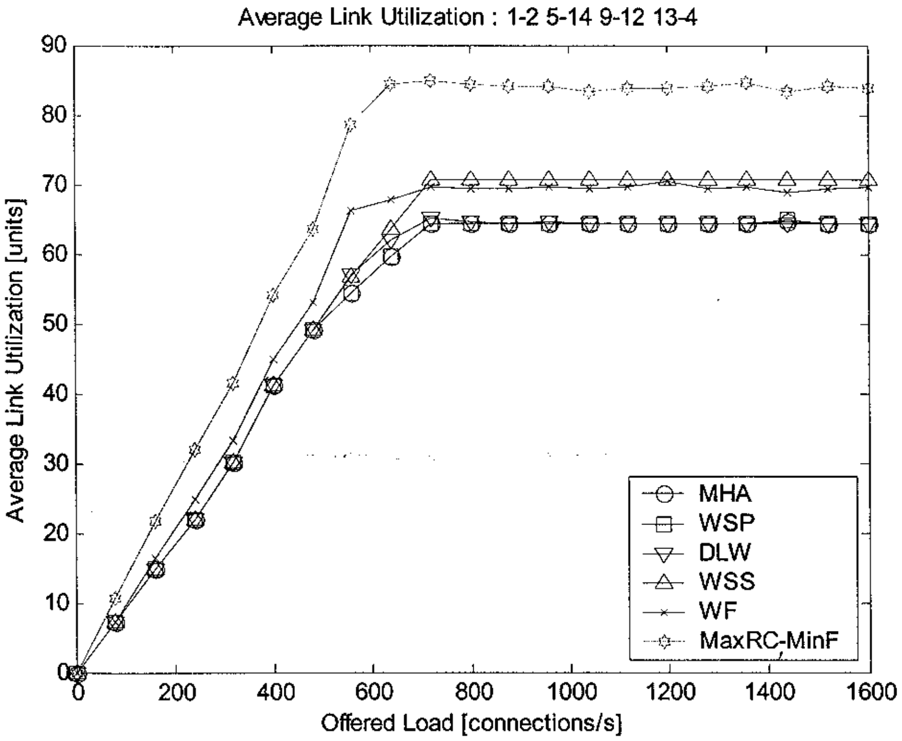
ภาพที่ 9 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4



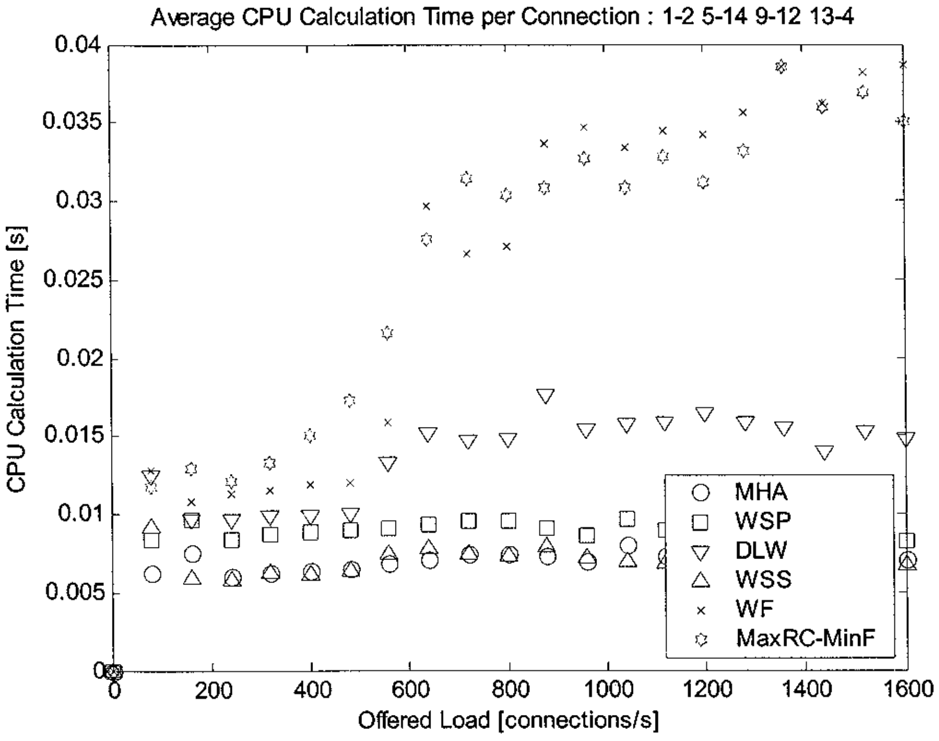
ภาพที่ 10 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4



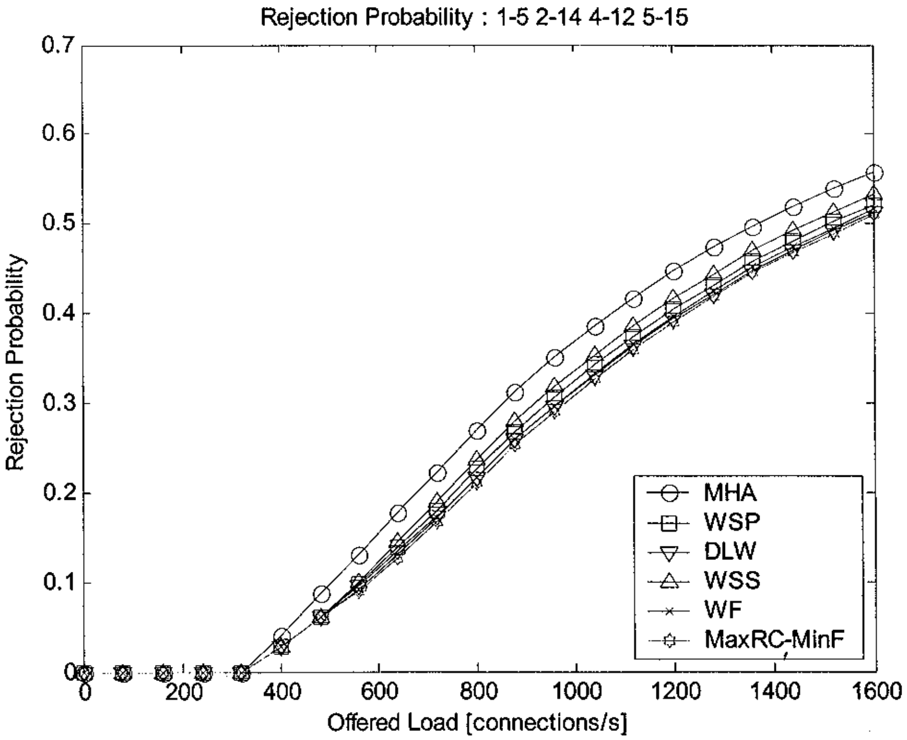
ภาพที่ 11 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4



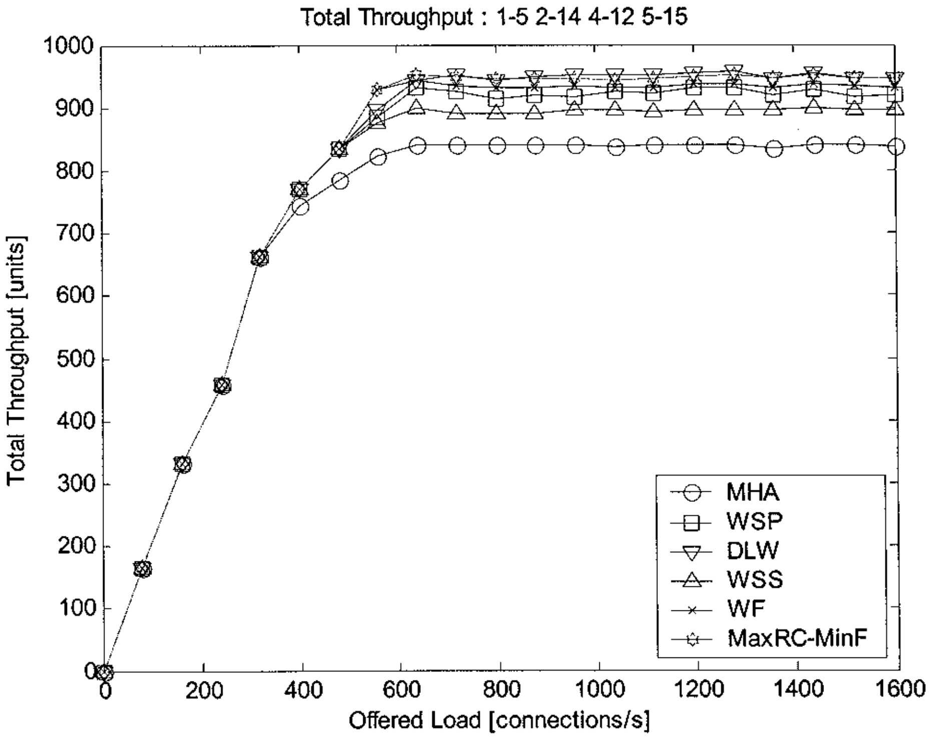
ภาพที่ 12 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4



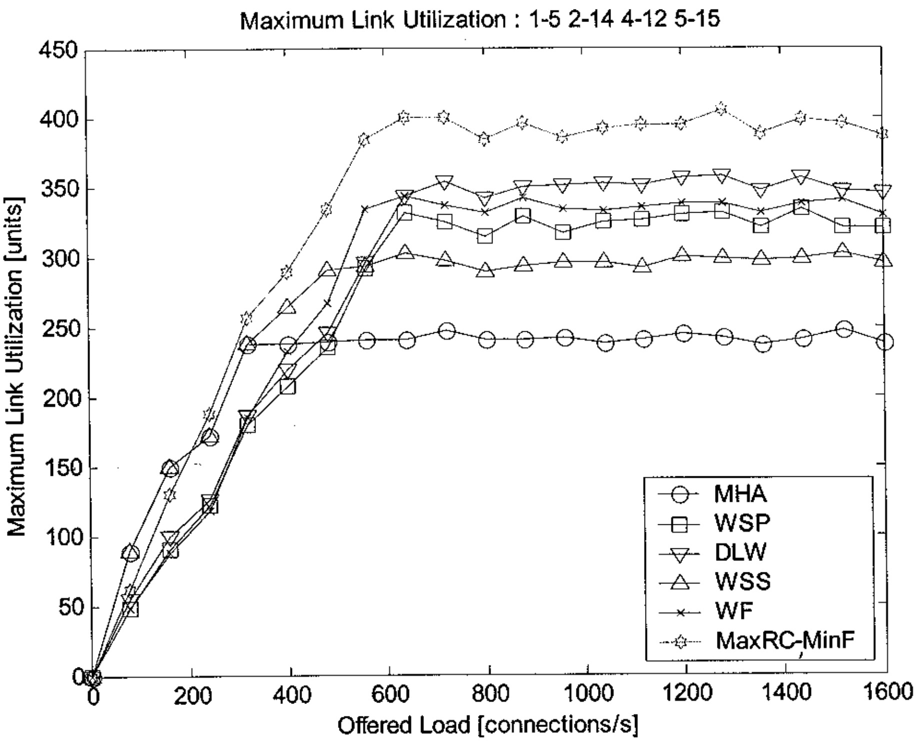
ภาพที่ 13 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-2, 5-14, 9-12 และ 13-4



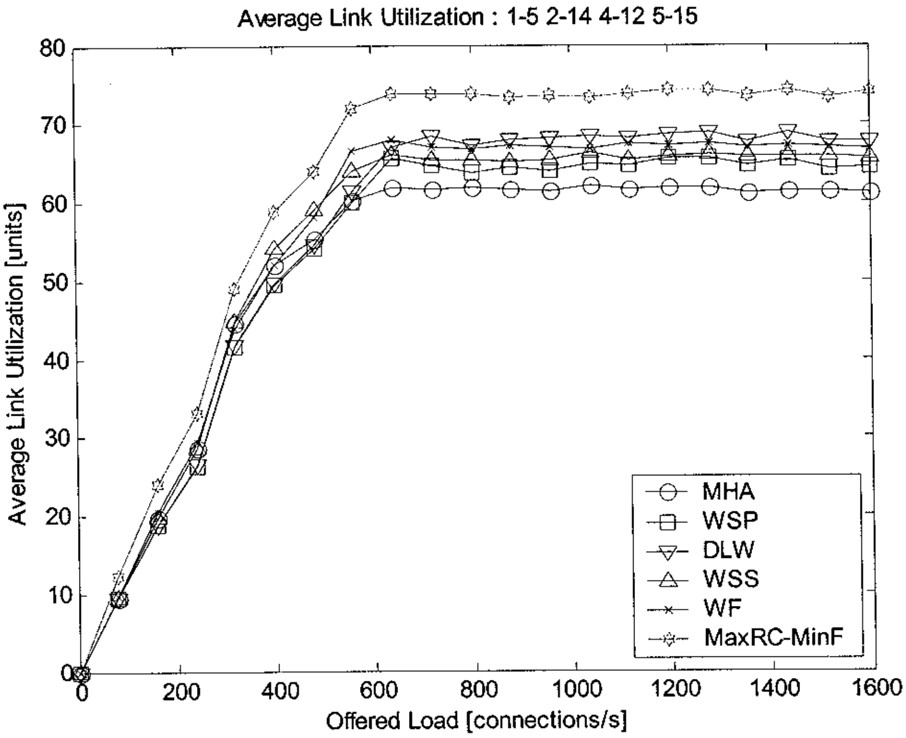
ภาพที่ 14 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15



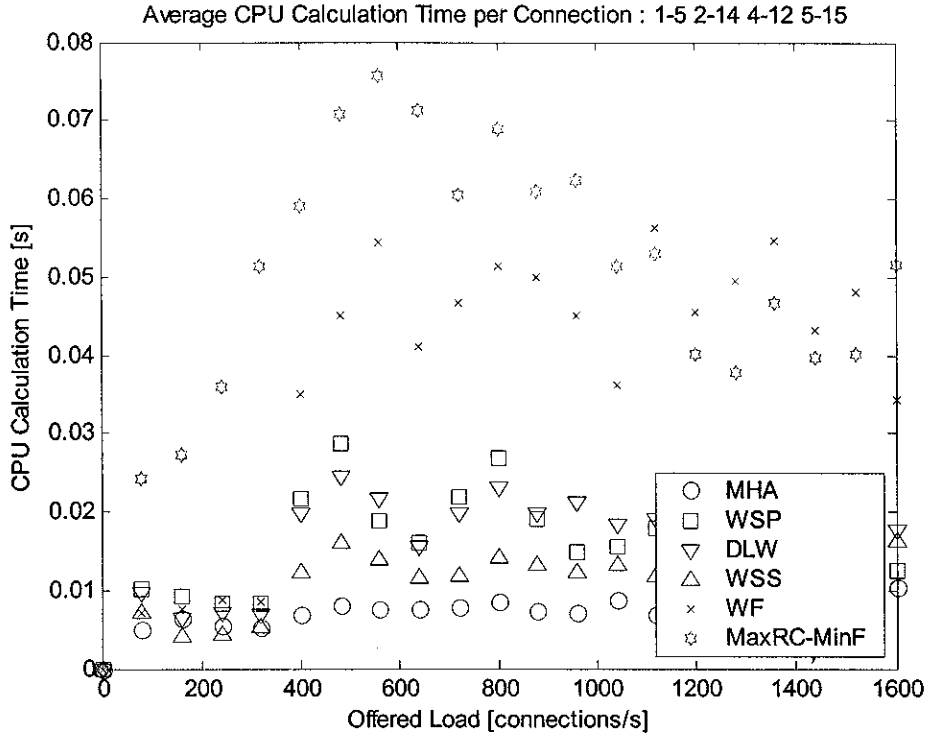
ภาพที่ 15 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15



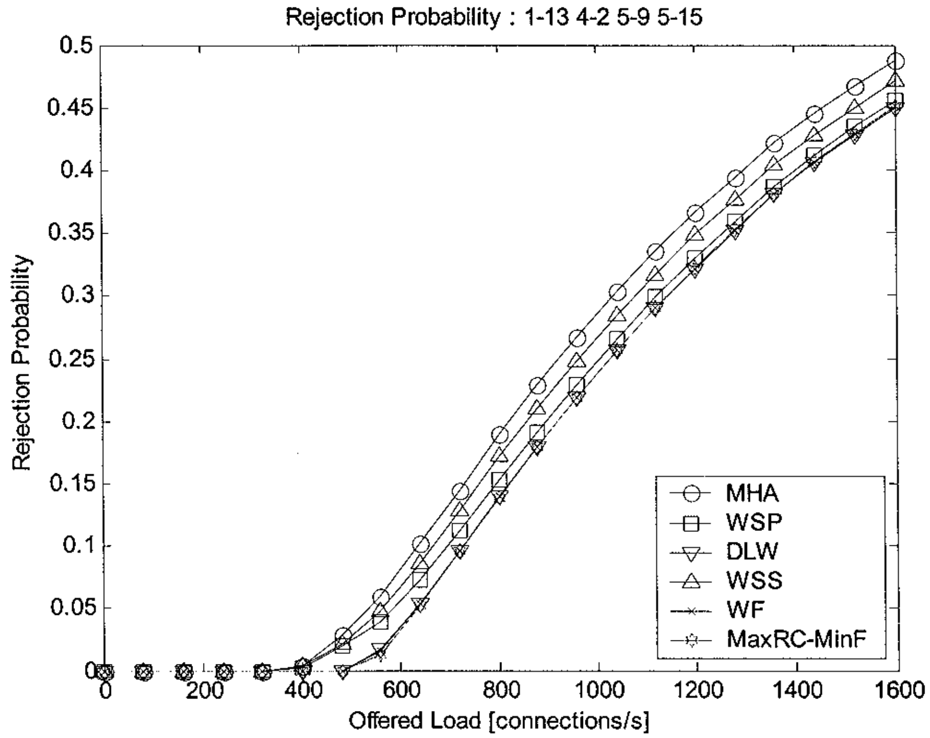
ภาพที่ 16 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15



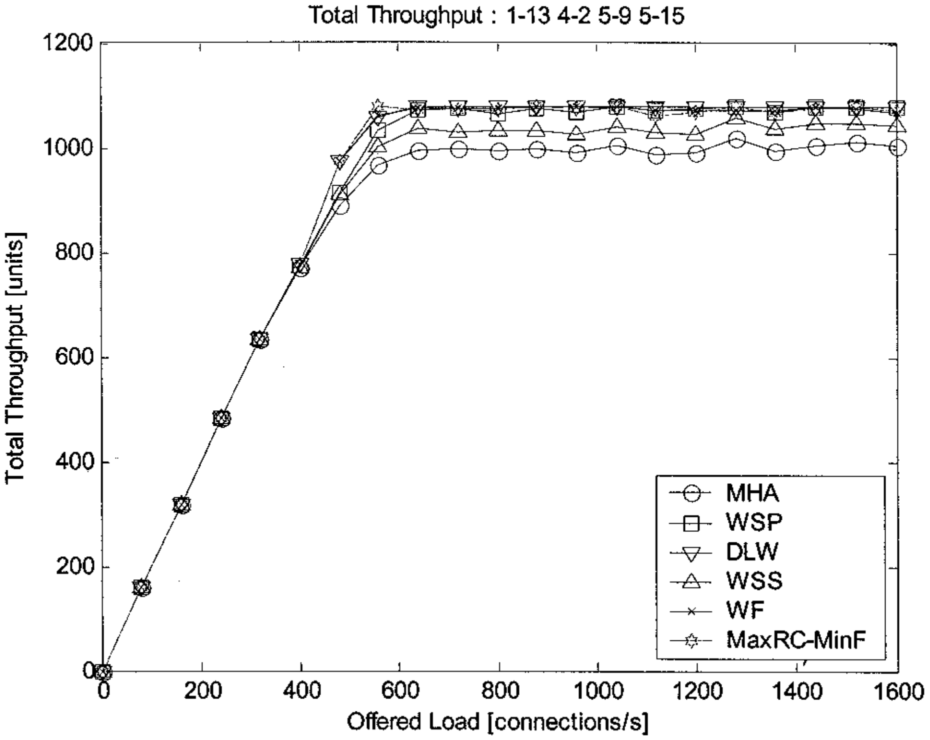
ภาพที่ 17 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15



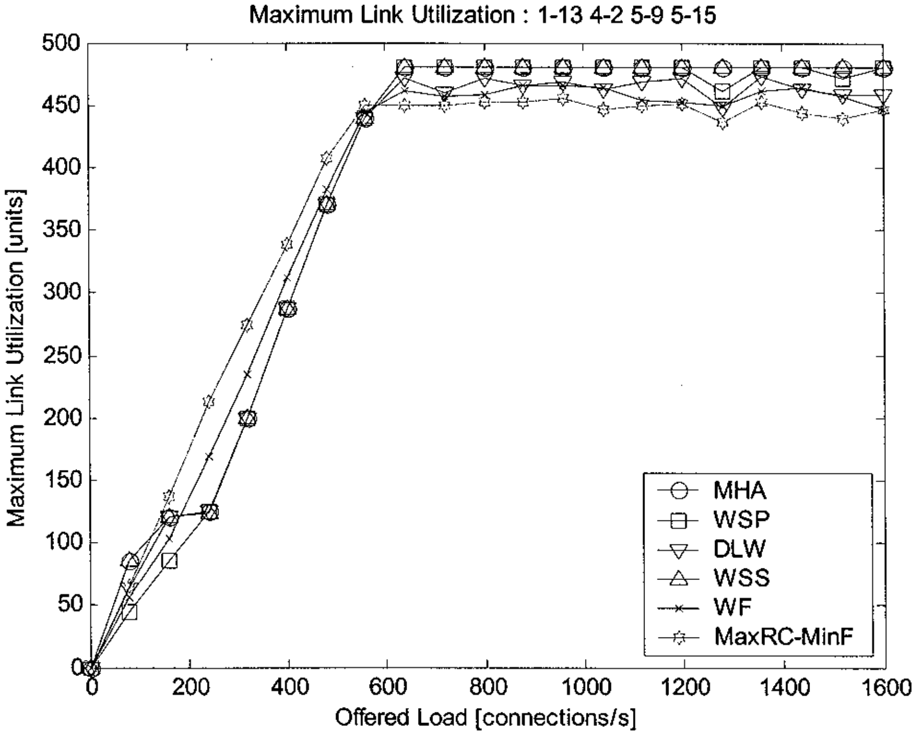
ภาพที่ 18 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-5, 2-14, 4-12 และ 5-15



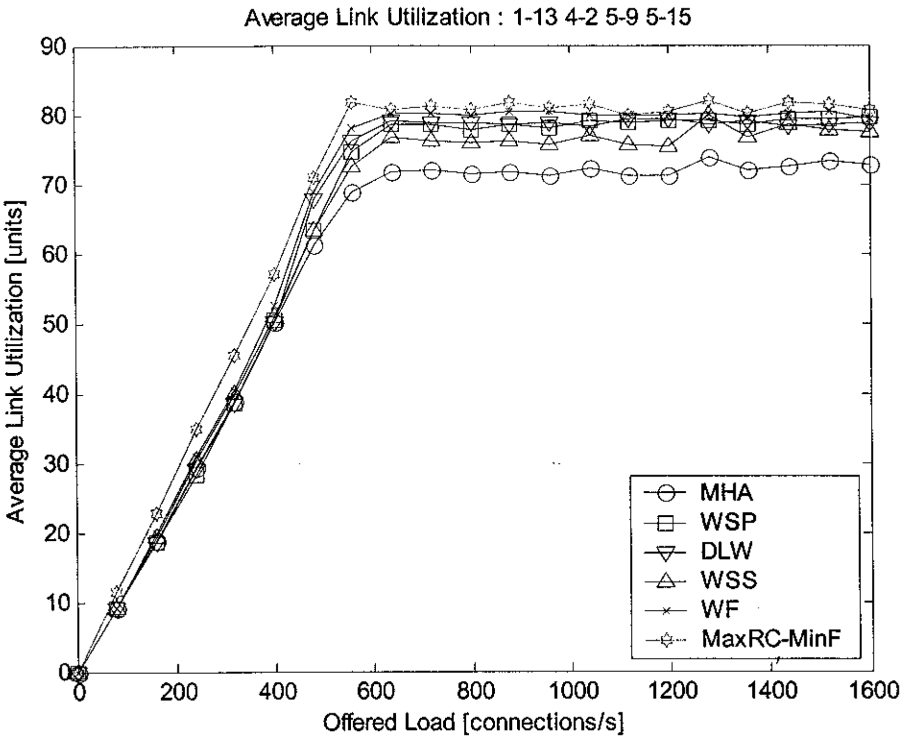
ภาพที่ 19 Rejection Probability ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15



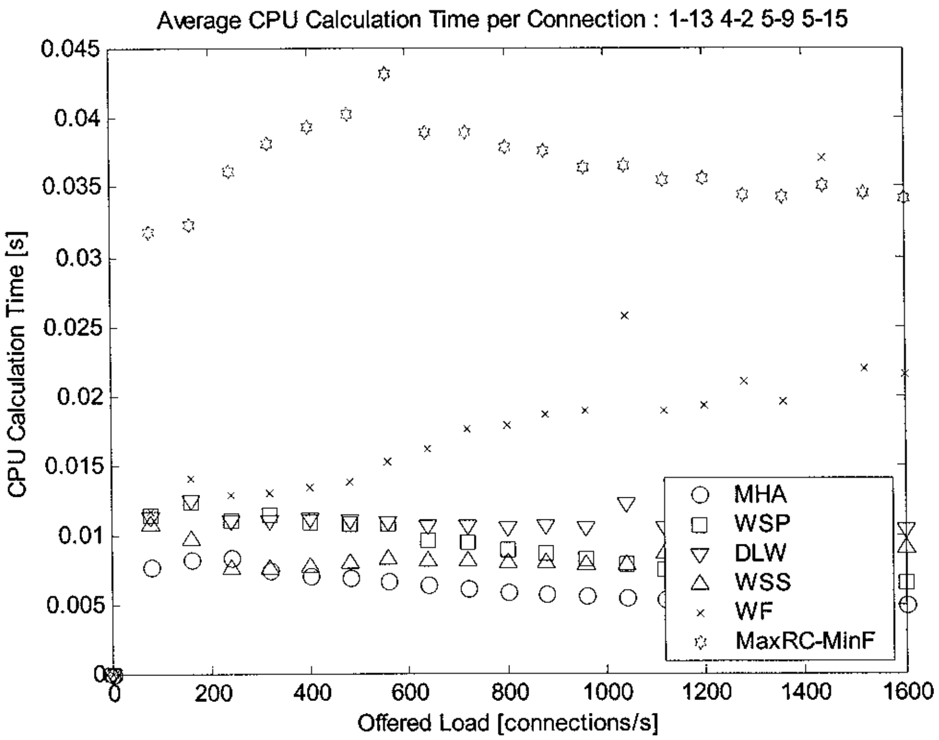
ภาพที่ 20 Total Throughput ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15



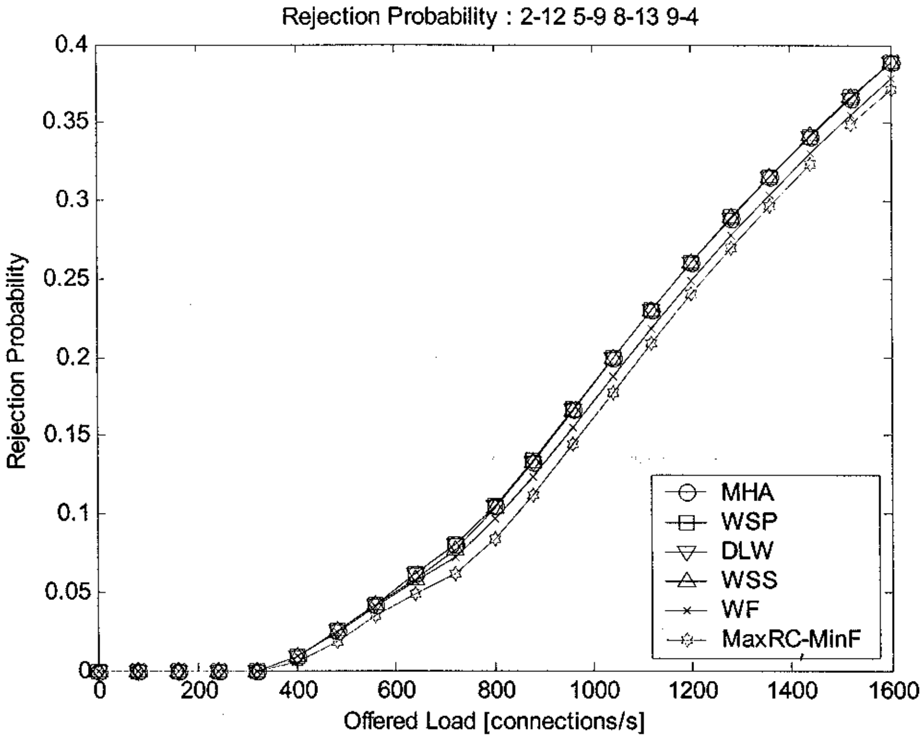
ภาพที่ 21 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15



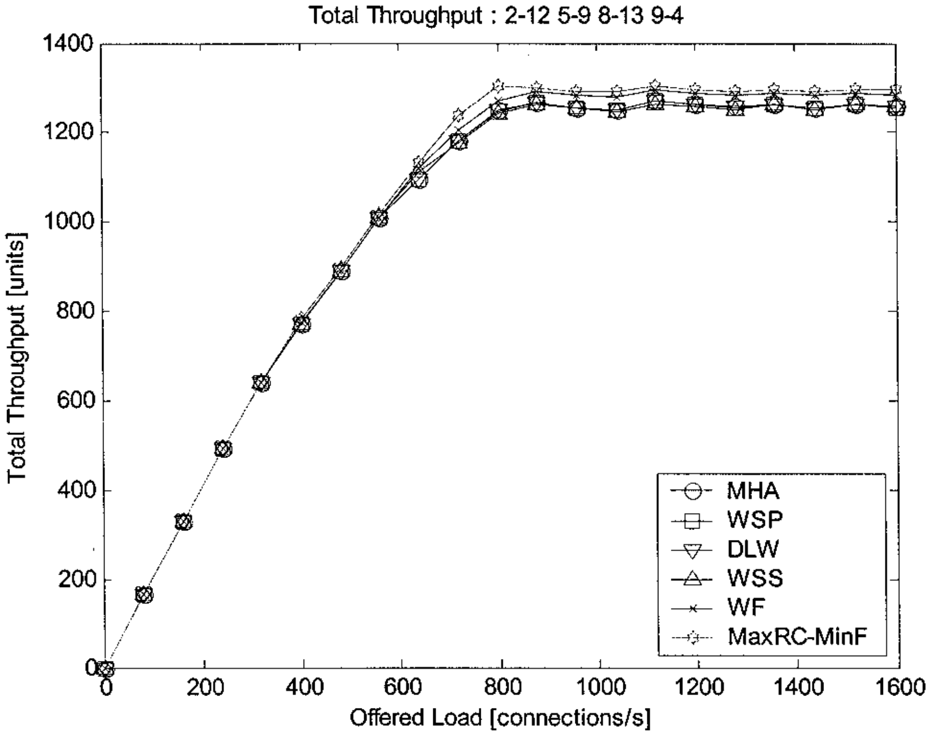
ภาพที่ 22 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15



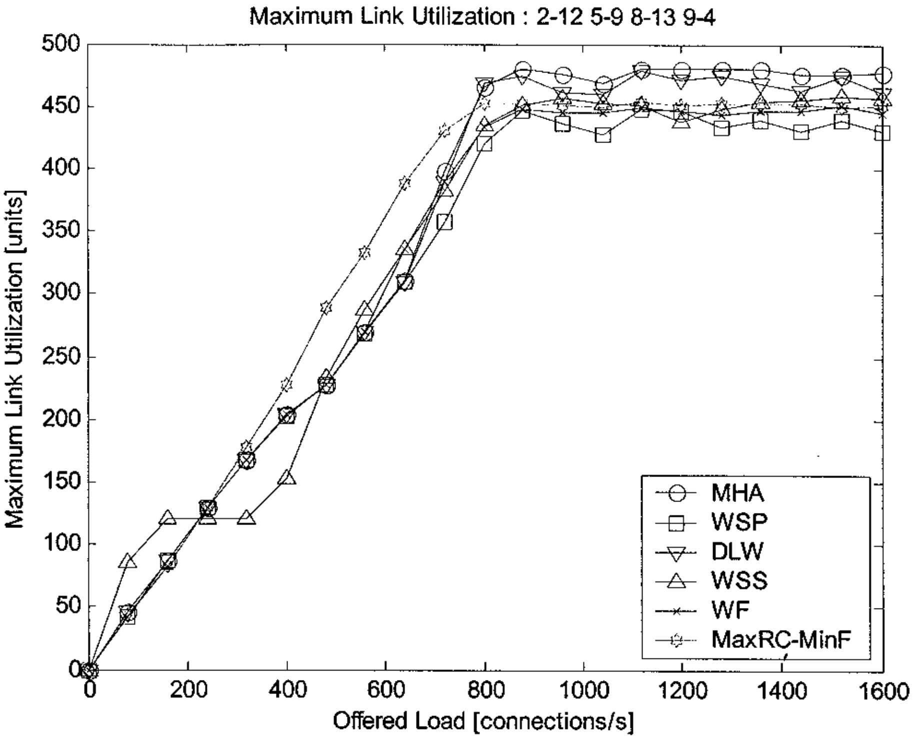
ภาพที่ 23 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 1-13, 4-2, 5-9 และ 5-15



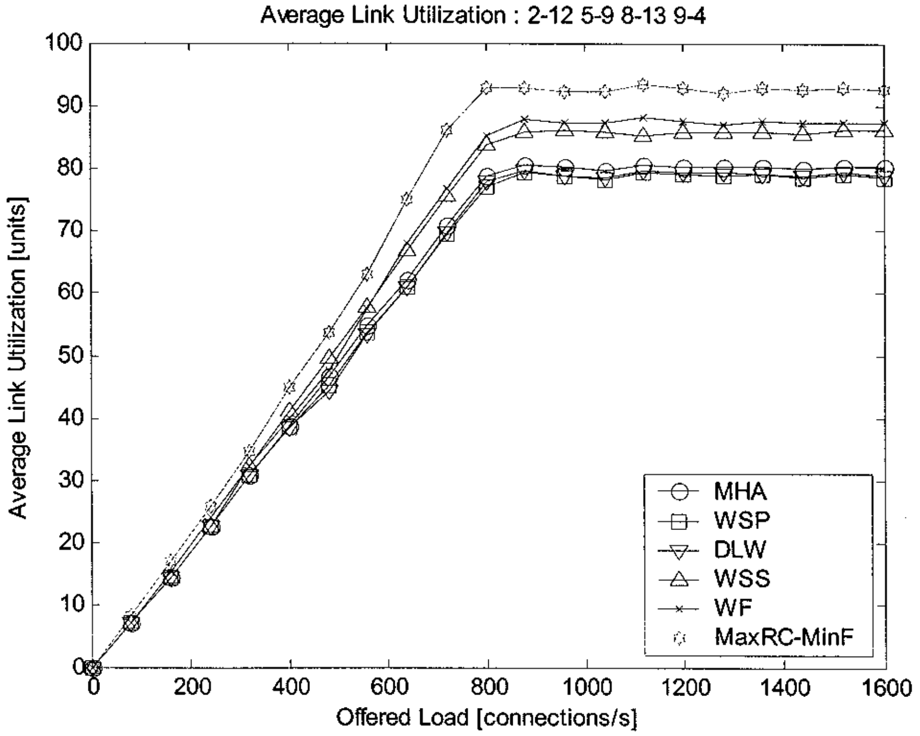
ภาพที่ 24 Rejection Probability ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4



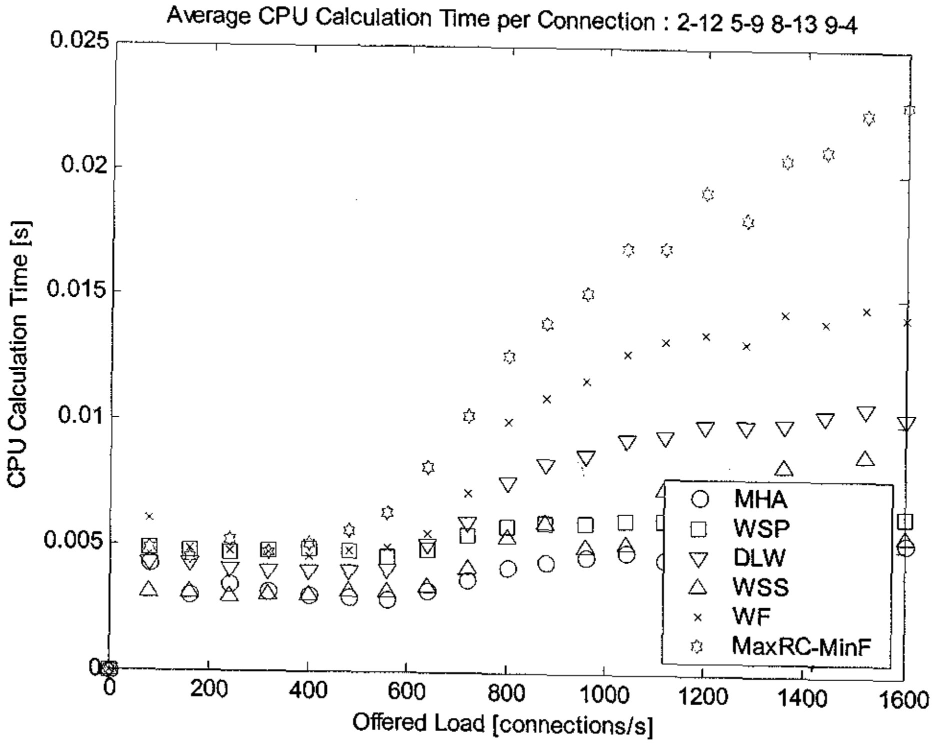
ภาพที่ 25 Total Throughput ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4



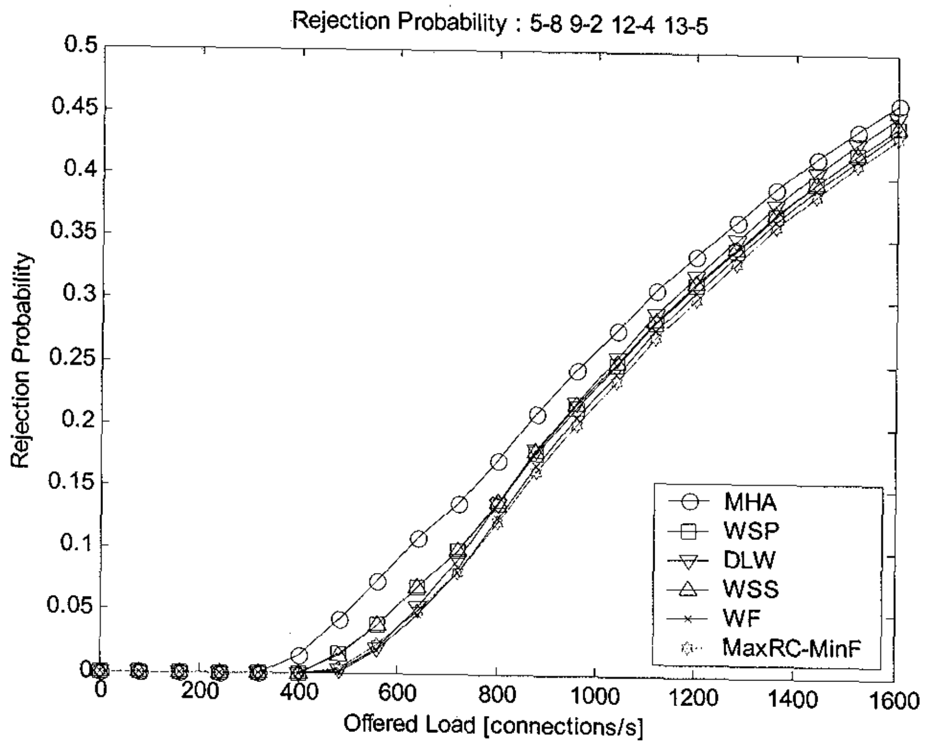
ภาพที่ 26 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4



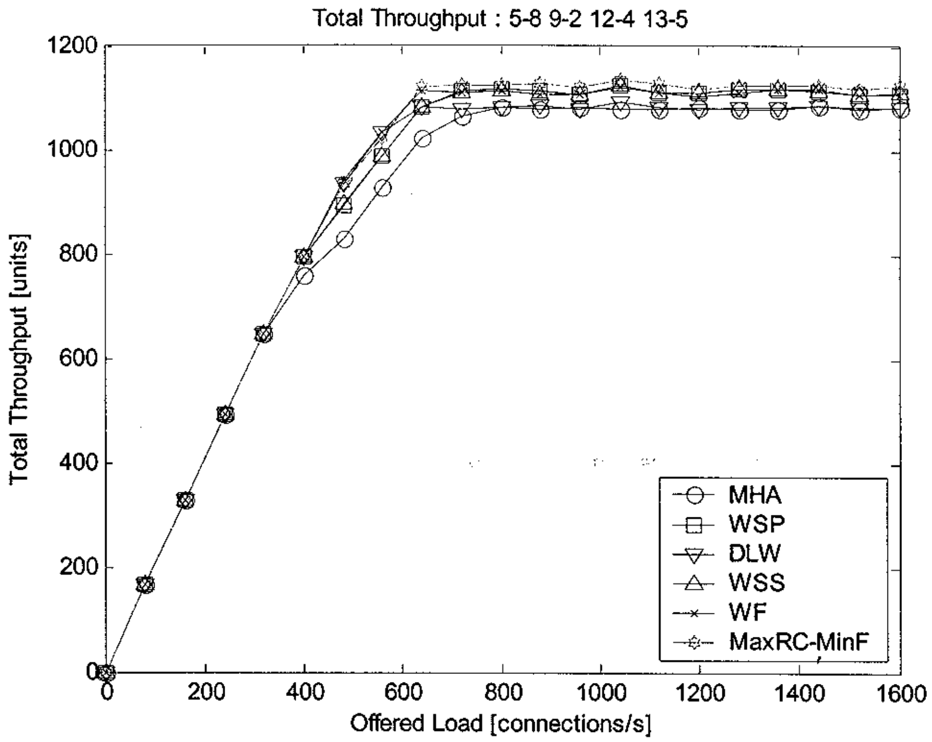
ภาพที่ 27 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4



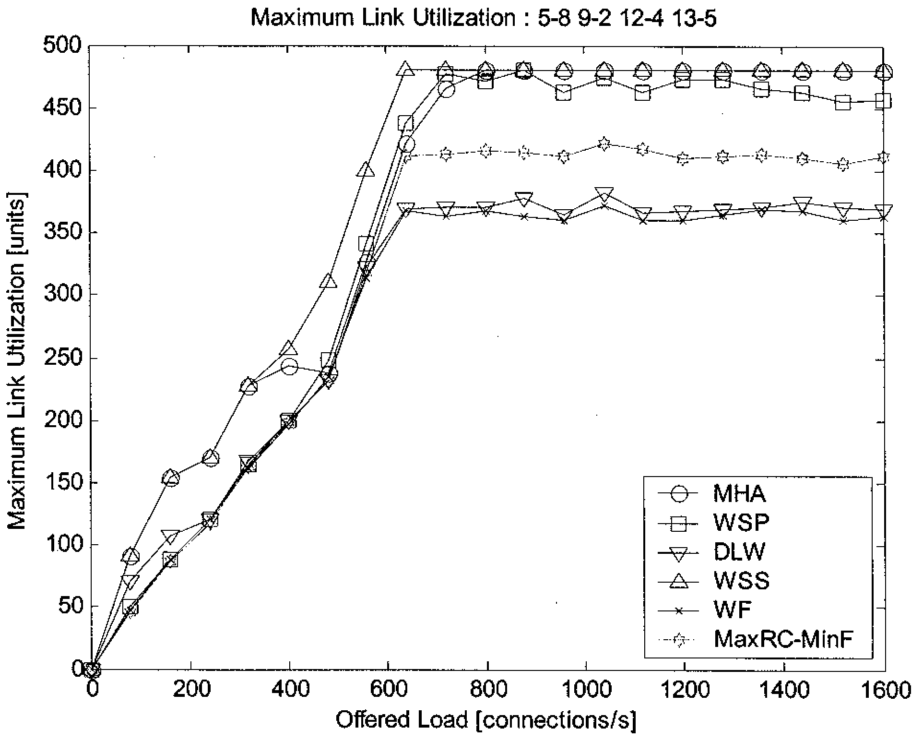
ภาพที่ 28 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 2-12, 5-9, 8-13 และ 9-4



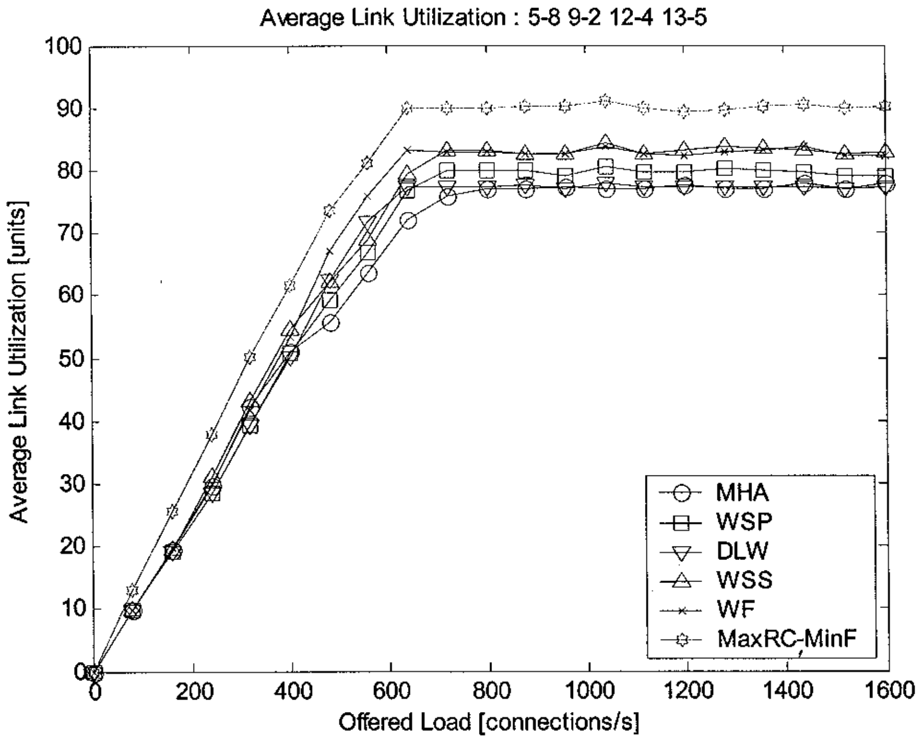
ภาพที่ 29 Rejection Probability ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5



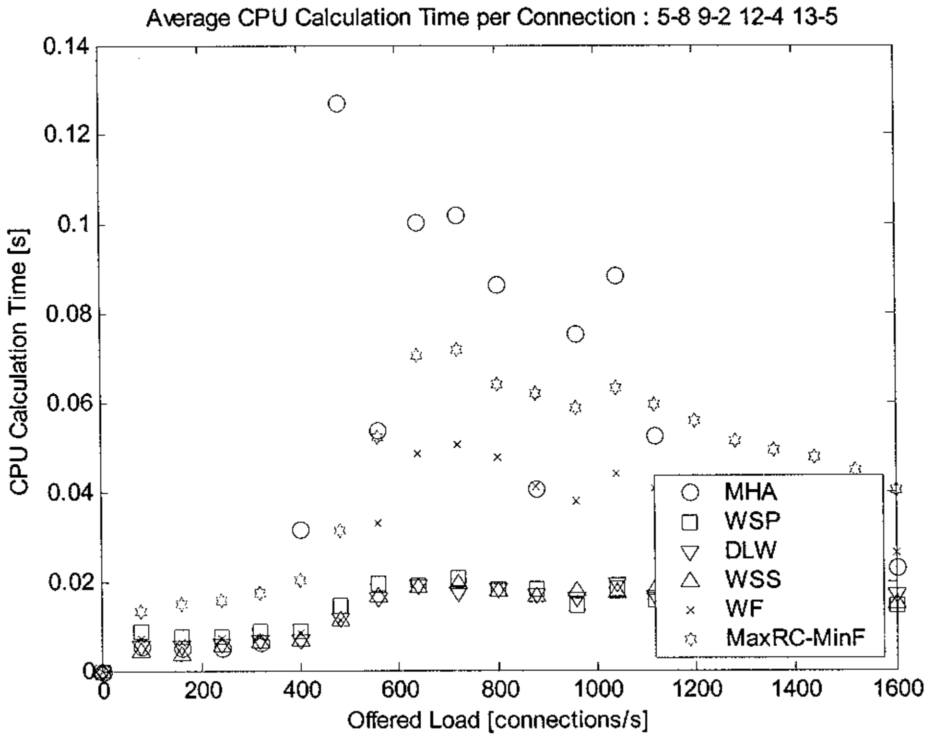
ภาพที่ 30 Total Throughput ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5



ภาพที่ 31 Maximum Link Utilization ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5



ภาพที่ 32 Average Link Utilization ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5



ภาพที่ 33 CPU Calculation Time ของโครงข่าย NET1 คู่โหนด 5-8, 9-2, 12-4 และ 13-5

4. วิเคราะห์ผลการจำลองโครงข่าย NET1

ภาพที่ 9 ถึง ภาพที่ 13 เป็นผลการทดลองของคู่โหนด 1-2, 5-15, 9-12 และ 13-4 ในภาพที่ 9 จะเห็นได้ว่าวิธีการ MaxRC-MinF มีค่า Rejection Probability ต่ำกว่าวิธีการอื่นๆ ในช่วงที่มีทราฟฟิกโหลดระหว่าง 0 – 1000 connections/s ซึ่งถือว่าดี แต่ว่าช่วง 1200 – 1600 connections/s มีค่าสูงกว่าวิธีการ WSS เล็กน้อย

สำหรับภาพที่ 10 จะเห็นว่าวิธีการ MaxRC-MinF มีค่า Total Throughput สูงสุด ซึ่งแสดงว่าสามารถส่งข้อมูลได้ปริมาณมากที่สุด

จากภาพที่ 11 และภาพที่ 12 วิธีการที่นำเสนอ หรือ MaxRC-MinF มีค่า Maximum Link Utilization และ Average Link Utilization สูงสุด ซึ่งแสดงว่าสามารถใช้งานแบนด์วิดท์ของลิงค์ได้สูง

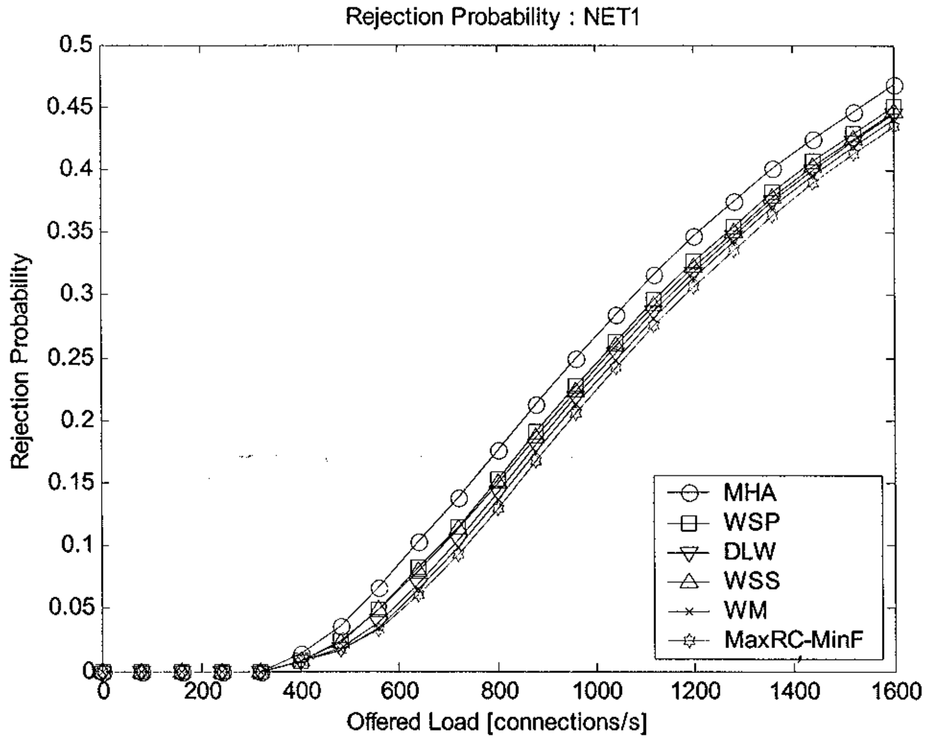
ในภาพที่ 13 จะเปรียบเทียบเวลาในการคำนวณในการหาเส้นทาง พบว่าวิธีการที่นำเสนอใช้เวลาในการคำนวณ (หรือความซับซ้อน) สูงกว่าวิธีการอื่นๆ เพียงเล็กน้อยในระดับที่ยอมรับได้

สำหรับภาพที่ 14 ถึง ภาพที่ 18 จะแสดงผลการทดลองระหว่างคู่โหนด 1-5, 2-14, 4-12 และ 5-15 พบว่าวิธีการที่นำเสนอมีค่าความน่าจะเป็นในการปฏิเสธเส้นทางต่ำที่สุด (ภาพที่ 14) ค่าวิสัยสามารถสูง (ภาพที่ 15) ค่าการใช้งานลิงค์สูงสุด (ภาพที่ 16 และ ภาพที่ 17) แต่ใช้เวลาในการคำนวณสูง (ภาพที่ 18)

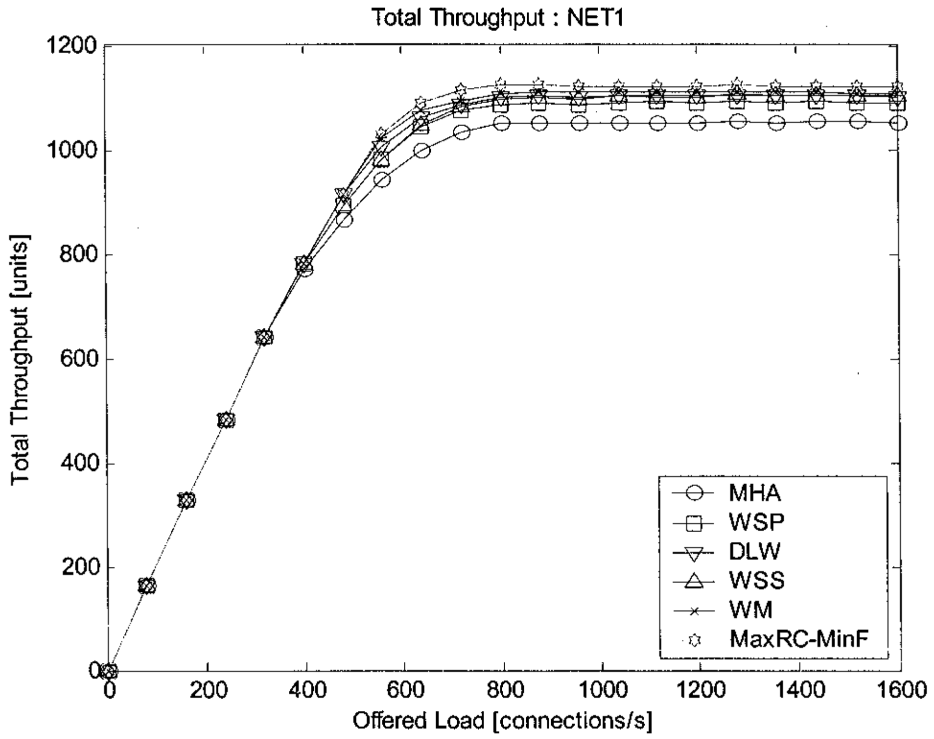
สำหรับภาพที่ 19 ถึง ภาพที่ 23 จะแสดงผลการทดลองระหว่างคู่โหนด 1-13, 4-2, 5-9 และ 5-15 พบว่าวิธีการ MaxRC-MinF มีค่าความน่าจะเป็นในการปฏิเสธเส้นทางต่ำ (ภาพที่ 19) ค่าวิสัยสามารถสูง (ภาพที่ 20) ค่าการใช้งานลิงค์สูงในช่วง 0 - 600 connections/s (ภาพที่ 21) มีค่าการใช้งานลิงค์เฉลี่ยสูงสุด (ภาพที่ 22) แต่ใช้เวลาในการคำนวณสูง (ภาพที่ 23) คล้ายกับในกรณีคู่โหนด 1-5, 2-14, 4-12 และ 5-15

จากภาพที่ 24 ถึง ภาพที่ 28 เป็นผลการทดลองระหว่างคู่โหนด 2-12, 5-9, 8-13 และ 9-4 พบว่าวิธีการ MaxRC-MinF มีค่า Rejection Probability ต่ำที่สุด (ภาพที่ 24) มีค่า Total Throughput สูงที่สุด (ภาพที่ 25) มีค่า Maximum Link Utilization สูงในช่วง 400 – 800 connections/s (ภาพที่ 26) มีค่า Average Link Utilization สูงสุด (ภาพที่ 27) แต่ใช้ CPU Calculation Time สูงสุด (ภาพที่ 28)

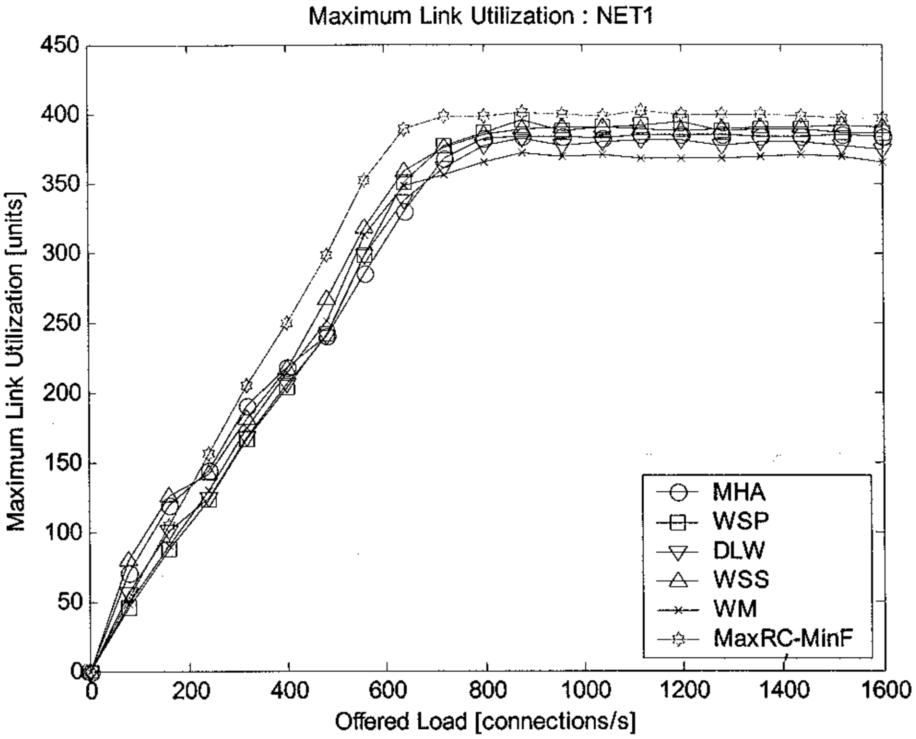
ภาพที่ 29 ถึง ภาพที่ 33 แสดงผลการทดลองของคู่โหนด 5-8, 9-2, 12-4 และ 13-5 พบว่าวิธีการที่นำเสนอมีค่า Rejection Probability สูงสุด (ภาพที่ 29) มีค่า Total Throughput สูงสุด (ภาพที่ 30) ค่า Maximum Link Utilization ปานกลาง (ภาพที่ 31) มีค่า Average Link Utilization สูง (ภาพที่ 32) และมีค่า CPU Calculation Time สูง (ภาพที่ 33)



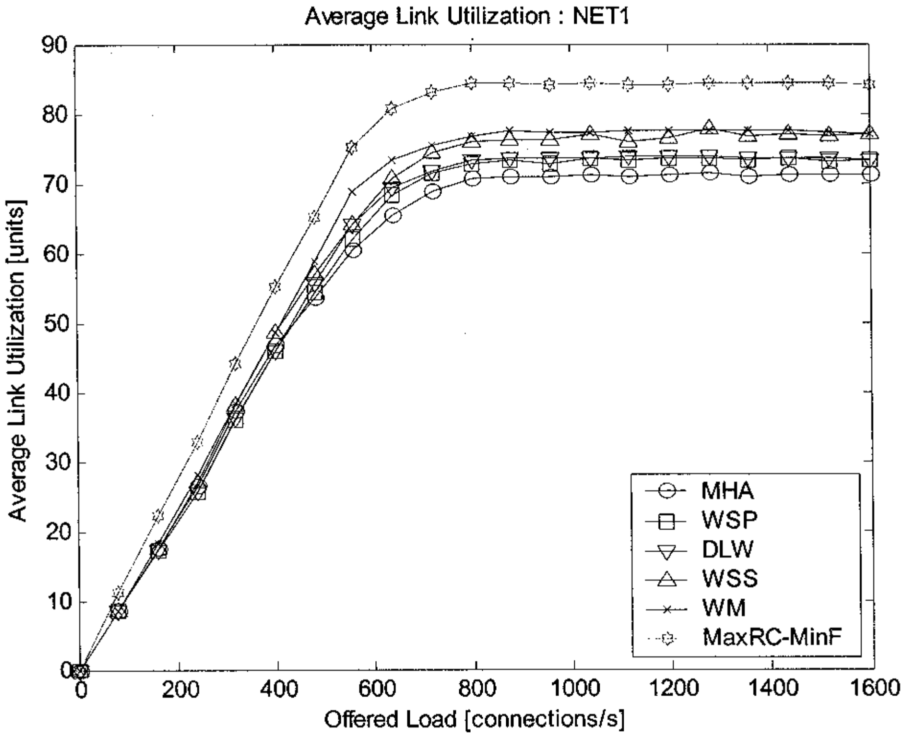
ภาพที่ 34 Rejection Probability โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1



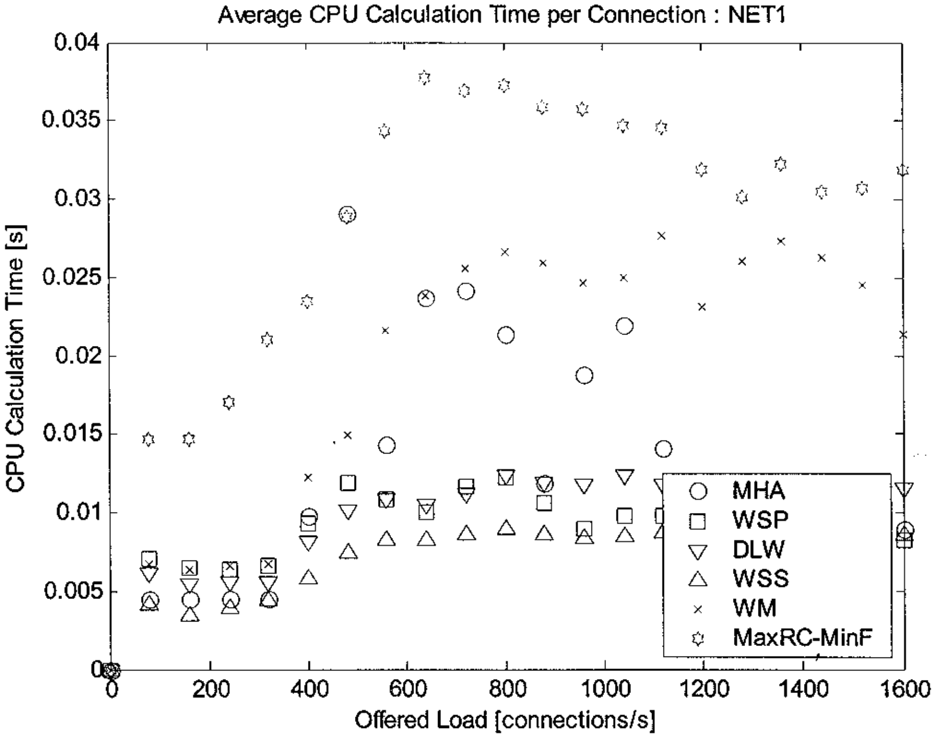
ภาพที่ 35 Total Throughput โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1



ภาพที่ 36 Maximum Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1



ภาพที่ 37 Average Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1



ภาพที่ 38 CPU Calculation Time โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET1

จากผลการทดลองทั้ง 5 คู่โหนดพบว่าผลการทดลองมีค่าไม่แน่นอน งานวิจัยนี้จึงได้นำผลการทดลองทั้ง 5 ครั้งมาเฉลี่ย พบว่าวิธีการที่นำเสนอมีค่า Rejection Probability ต่ำที่สุด มีค่า Total Throughput, Maximum Link Utilization, และ Average Link สูงที่สุด แต่มีค่า CPU Calculation Time สูงที่สุด ดังภาพที่ 34, ภาพที่ 35, ภาพที่ 36, ภาพที่ 37 และภาพที่ 38 ตามลำดับ

ซึ่งจากผลการทดลองเฉลี่ยเป็นการแสดงให้เห็นว่าวิธีการที่นำเสนอสามารถบรรลุเป้าหมายของการหาเส้นทางแบบมีเงื่อนไขที่รับประกันขนาดของแบนด์วิดท์ในโครงข่าย MPLS ได้เป็นที่น่าพอใจ แต่ยังมีข้อด้อยด้านความซับซ้อนของวิธีการซึ่งต้องใช้เวลาในการประมวลผลสูงกว่าวิธีการอื่นๆ

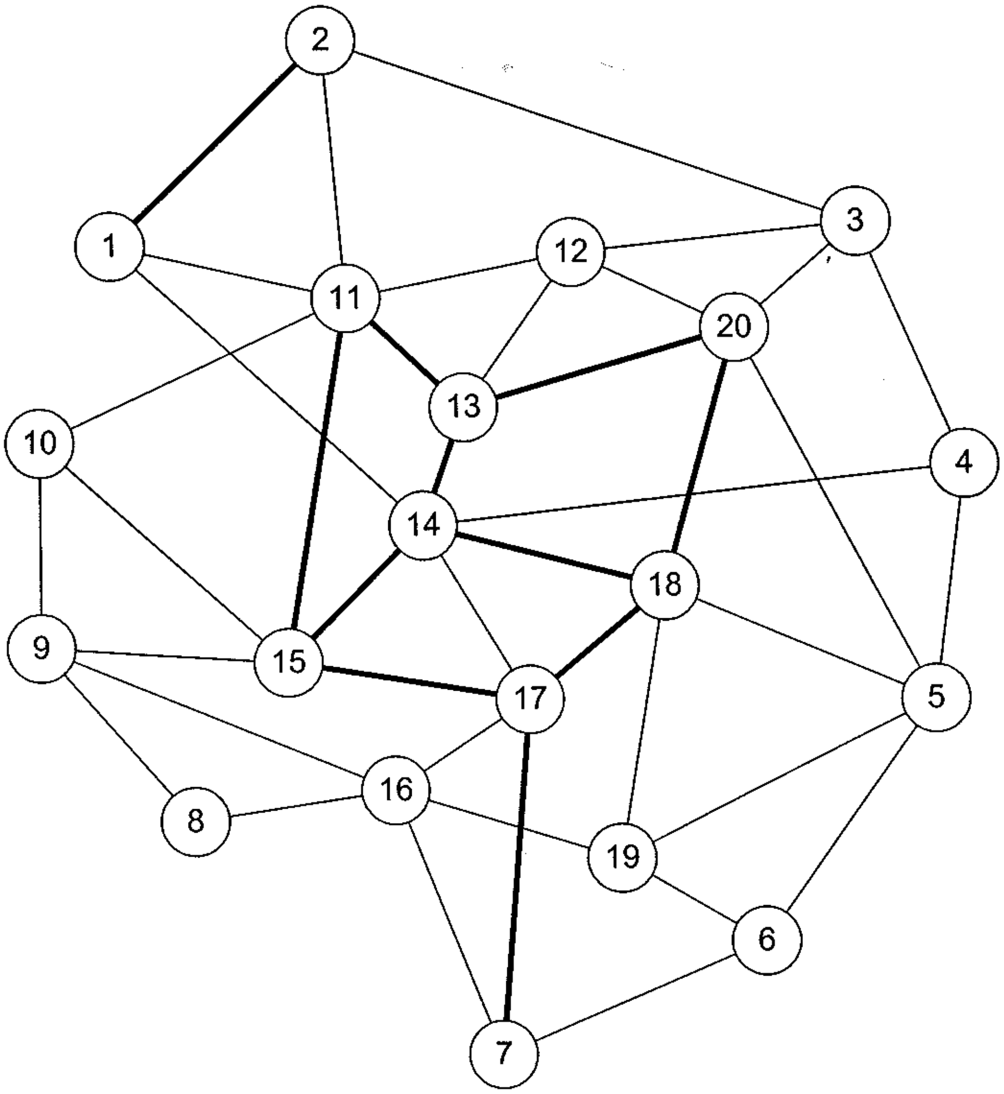
5. ผลการจำลองโครงข่าย NET2

ภาพที่ 39 แสดงโครงข่าย NET2 ซึ่งเป็นโครงข่ายที่เกิดจากการสุ่ม ซึ่งประกอบด้วย 20 โหนดและ 41 ลิงค์ การทดสอบจะมีการสุ่มเลือกคู่โหนด 5 คู่ จำนวน 5 ครั้ง ได้แก่

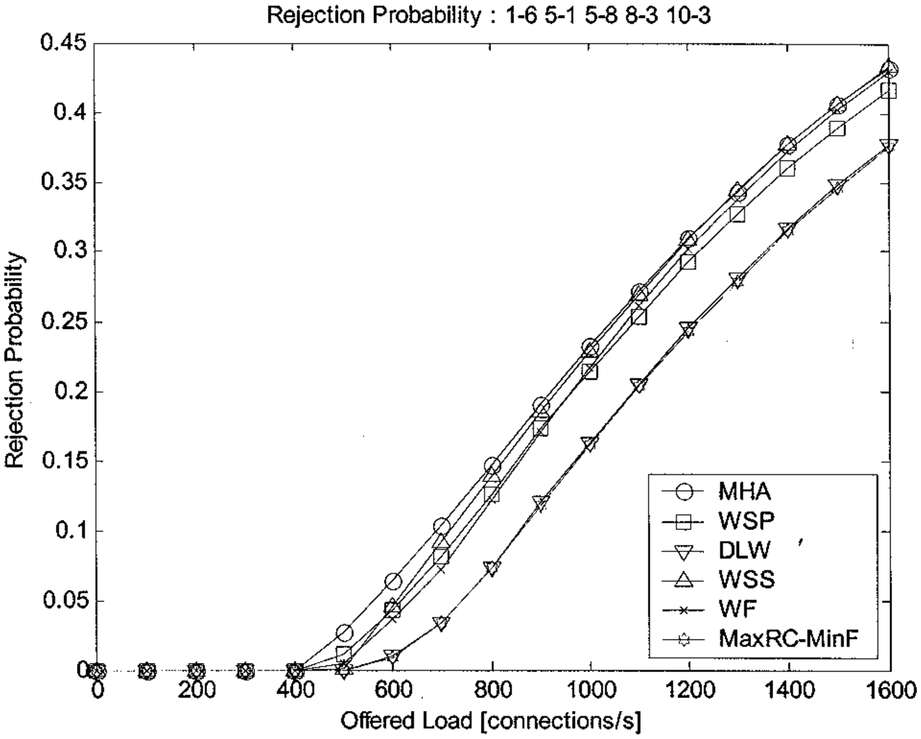
1. 1-6, 5-1, 5-8, 8-3 และ 10-3
2. 1-10, 2-9, 3-8, 4-7 และ 5-6

3. 2-1, 2-10, 8-9, 9-3 และ 10-5
4. 2-6, 3-4, 5-3, 5-6 และ 9-1
5. 3-6, 5-1, 6-4, 9-7 และ 9-8

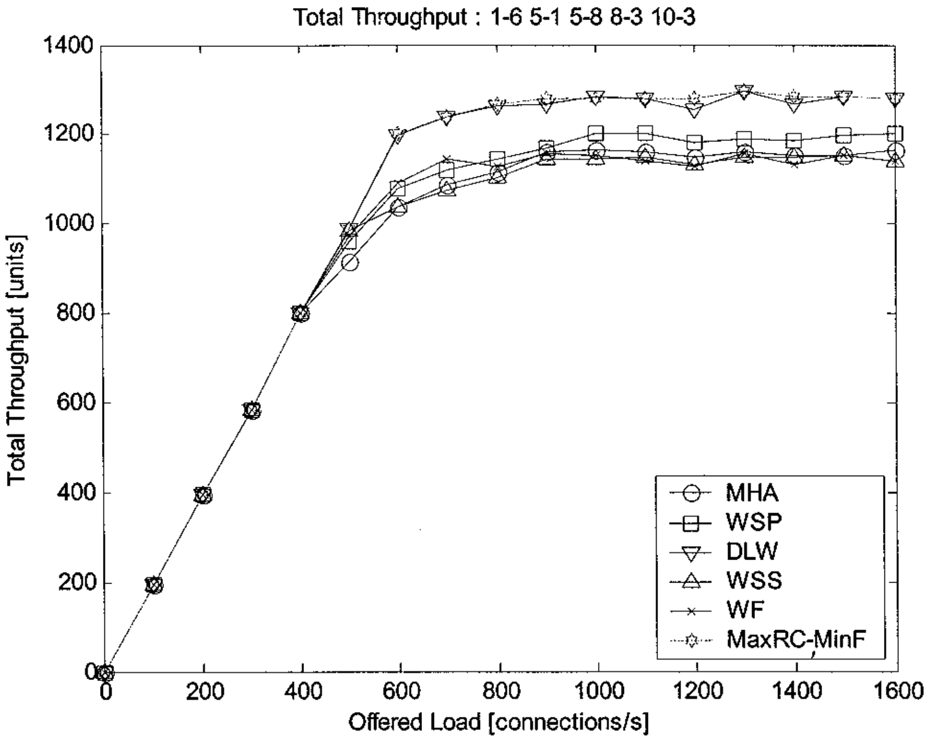
ซึ่งมีผลการจำลองดังนี้



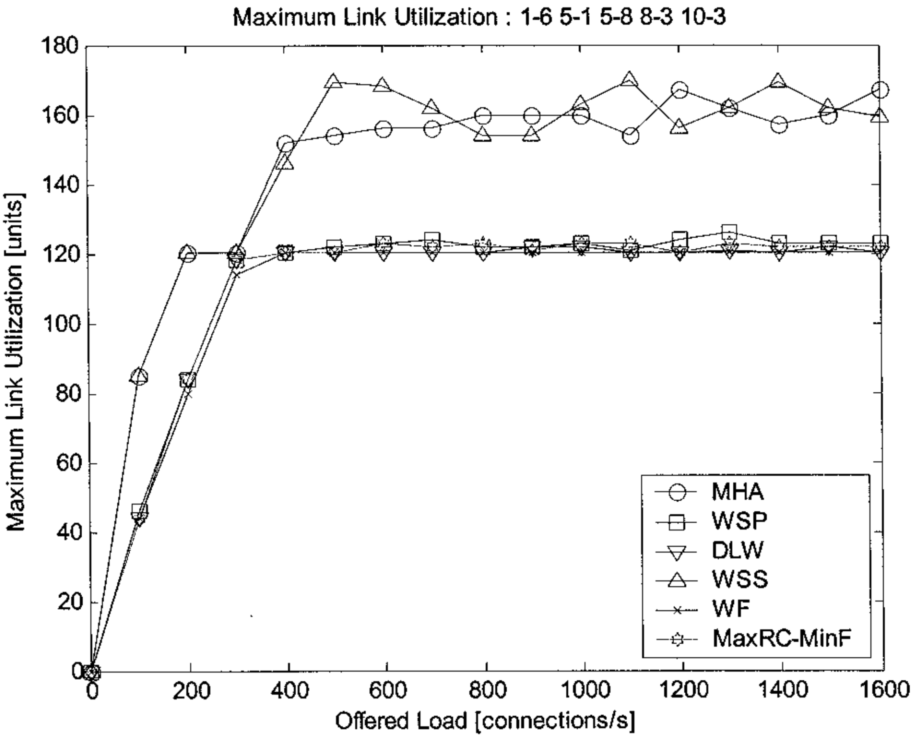
ภาพที่ 39 โครงข่าย NET2



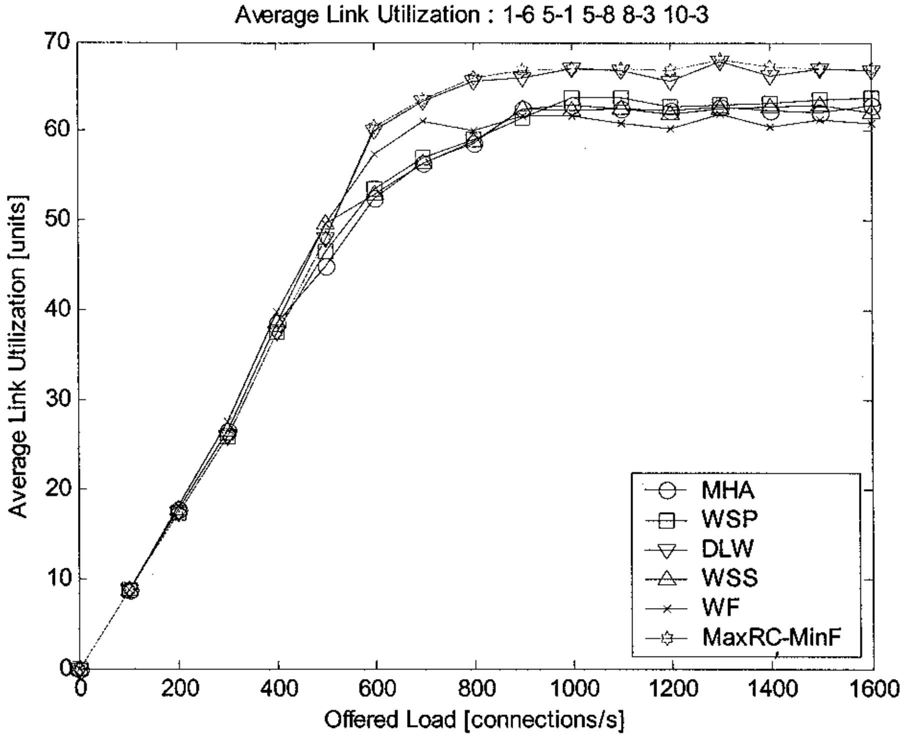
ภาพที่ 40 Rejection Probability ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3



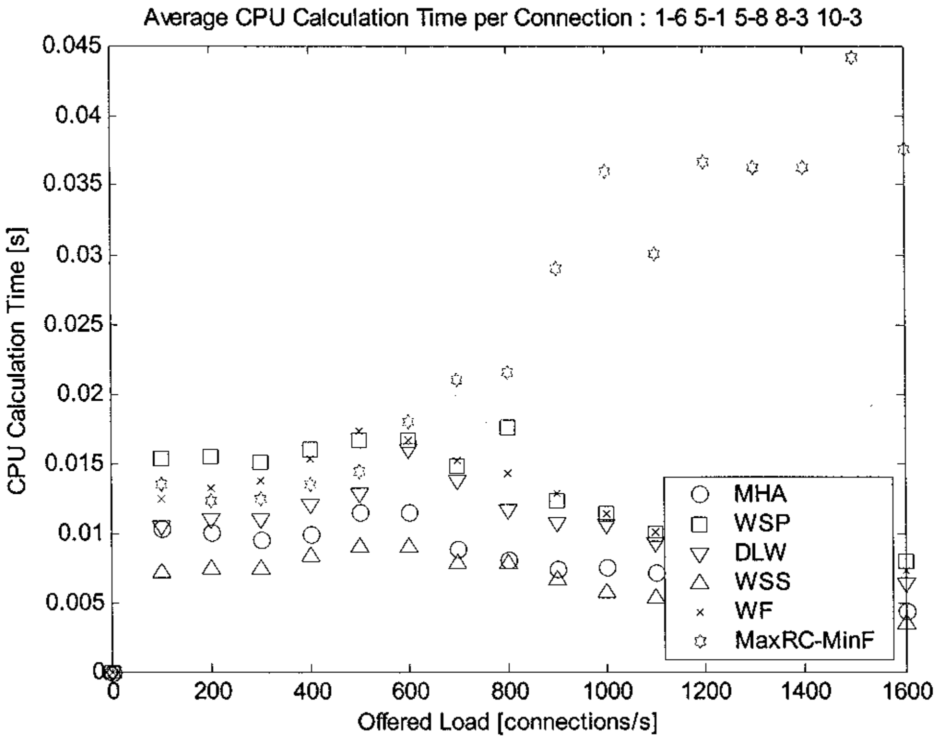
ภาพที่ 41 Total Throughput ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3



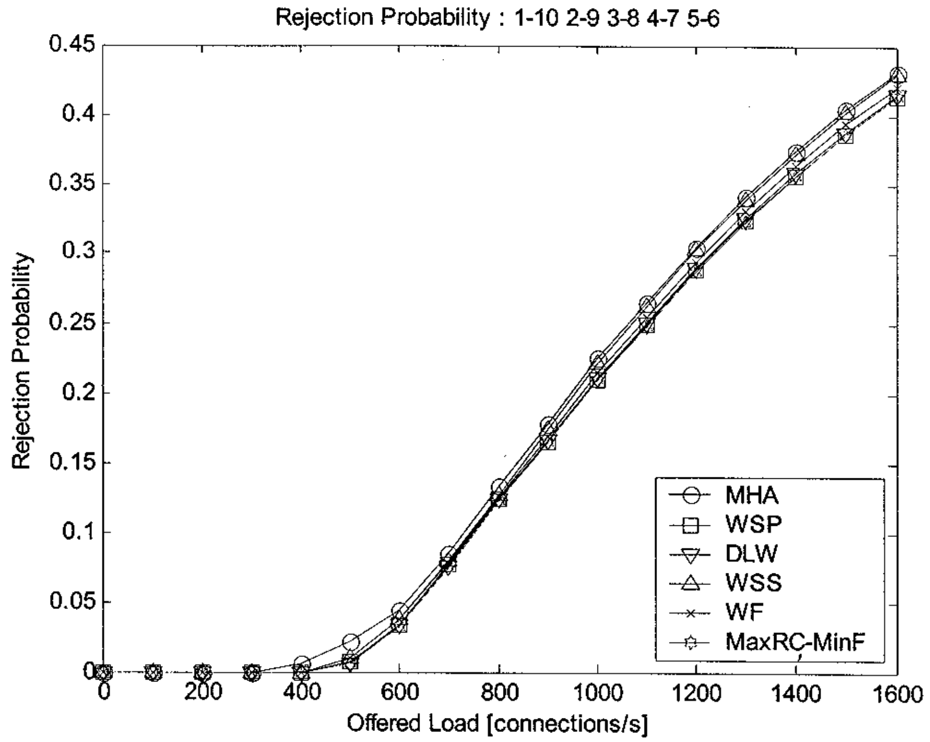
ภาพที่ 42 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3



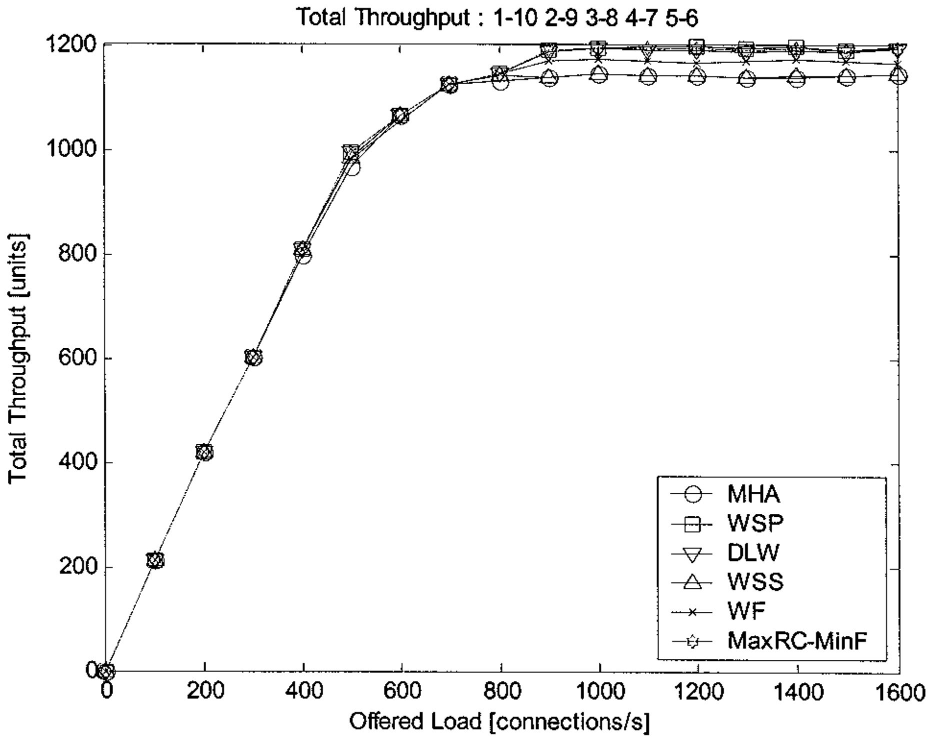
ภาพที่ 43 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3



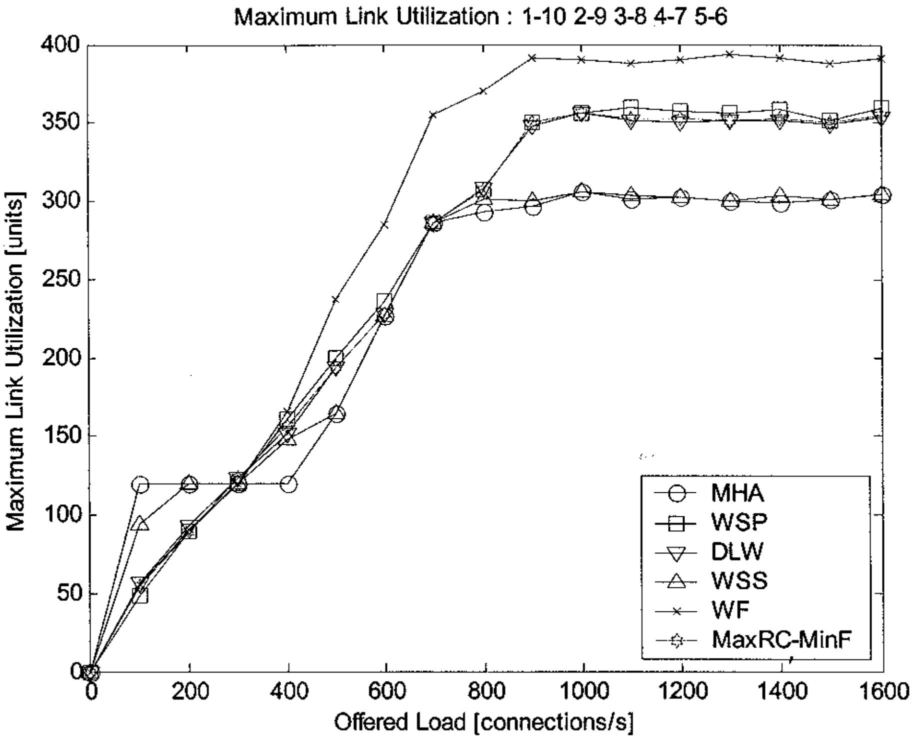
ภาพที่ 44 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3



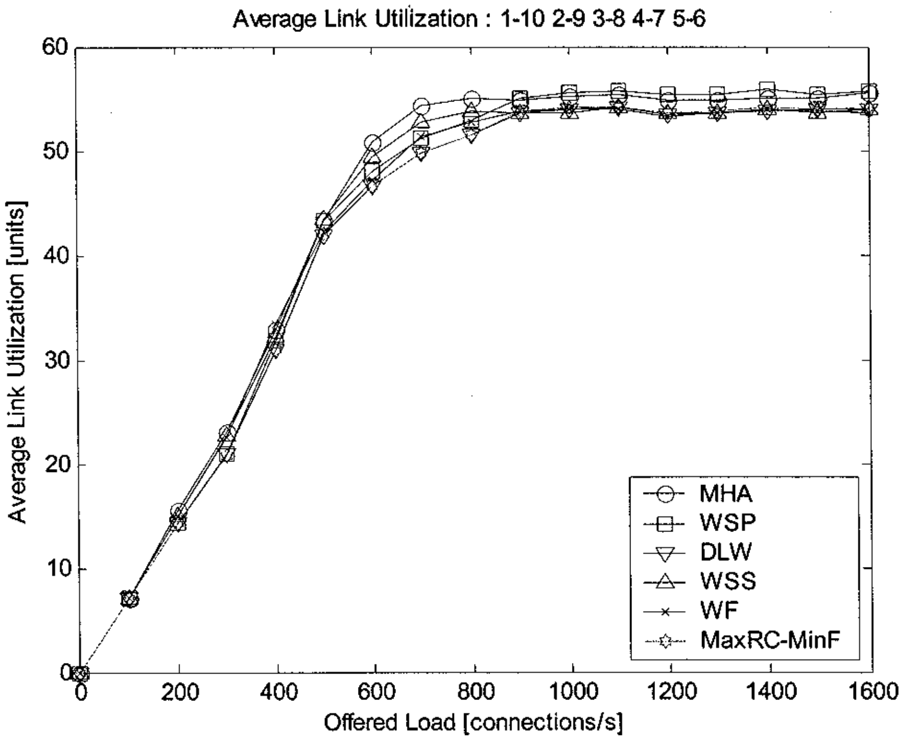
ภาพที่ 45 Rejection Probability ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6



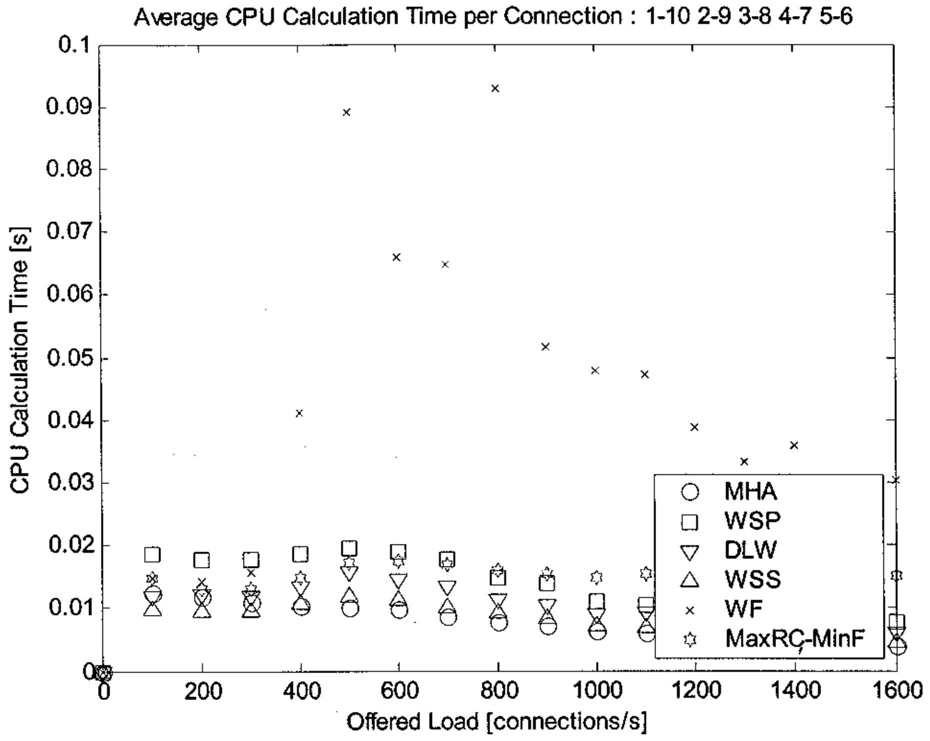
ภาพที่ 46 Total Throughput ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6



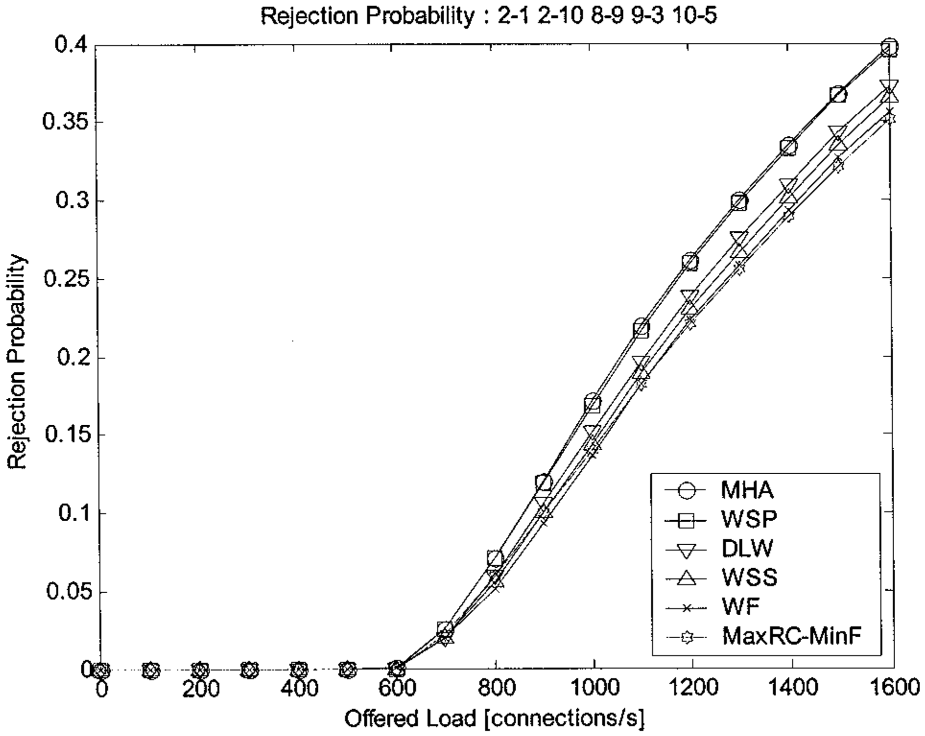
ภาพที่ 47 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6



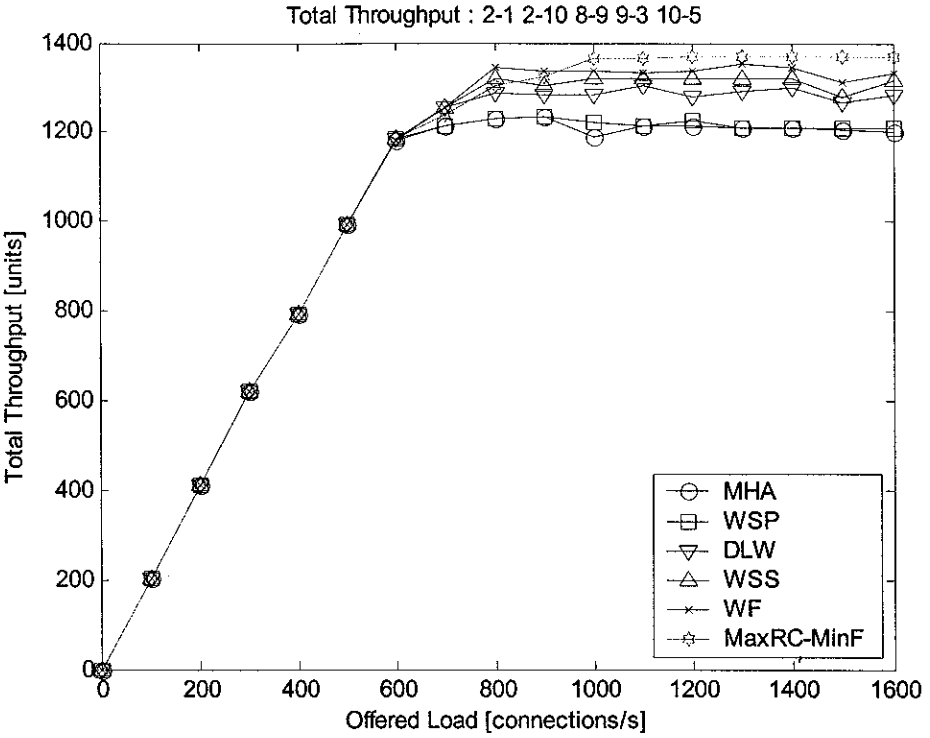
ภาพที่ 48 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6



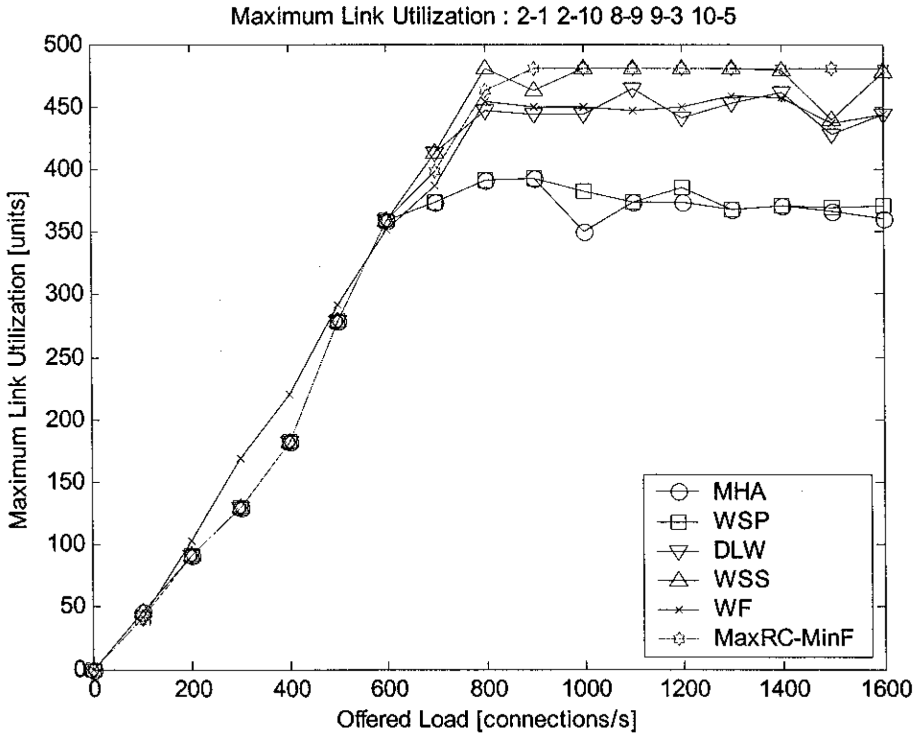
ภาพที่ 49 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6



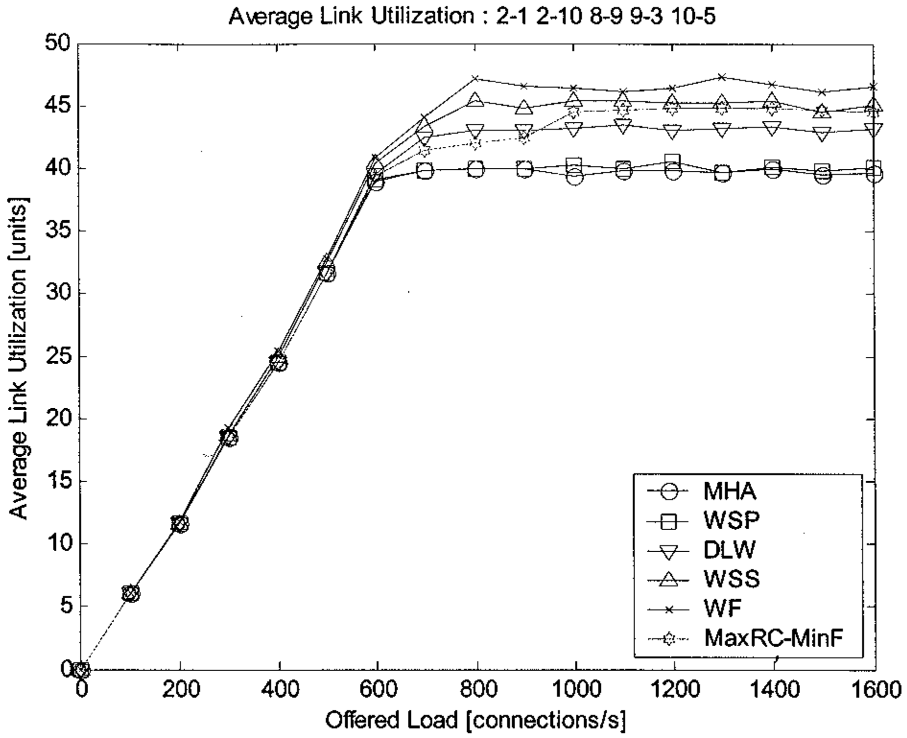
ภาพที่ 50 Rejection Probability ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5



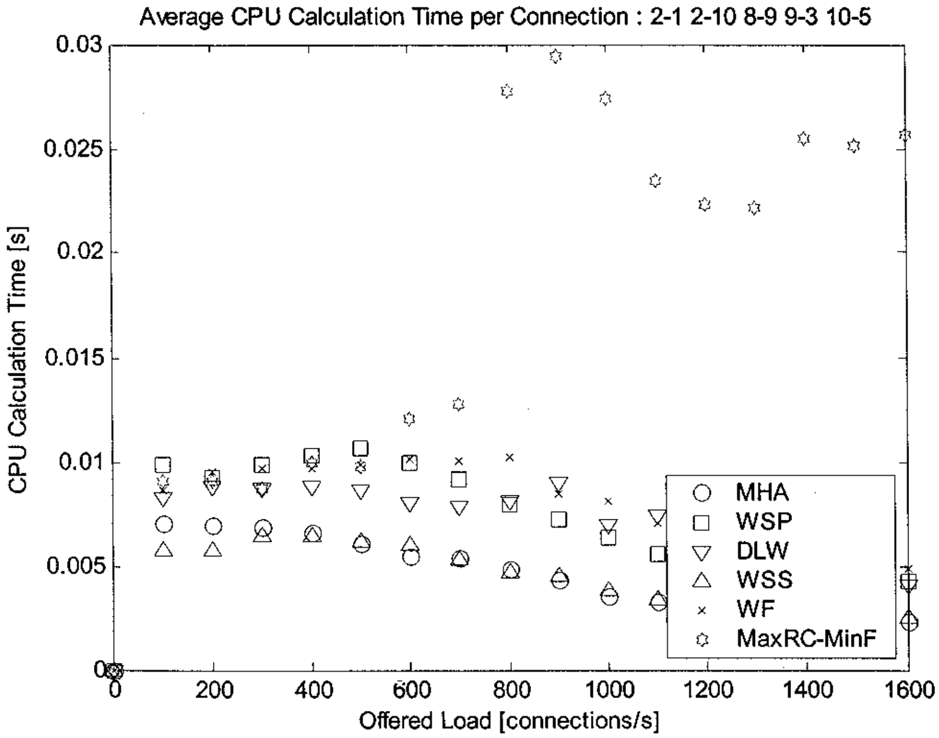
ภาพที่ 51 Total Throughput ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5



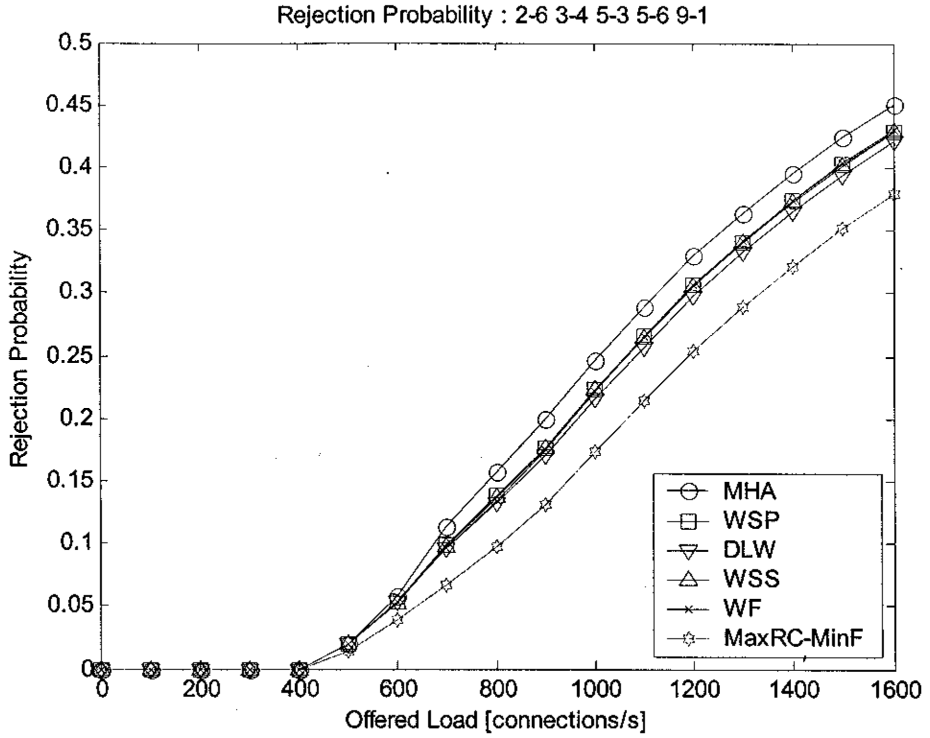
ภาพที่ 52 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5



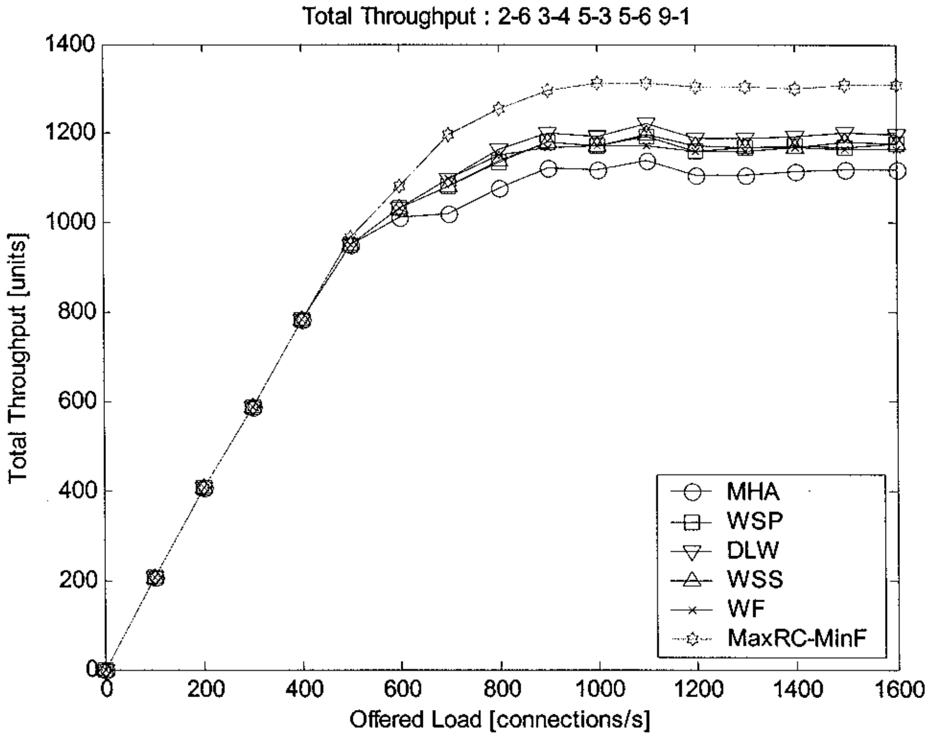
ภาพที่ 53 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5



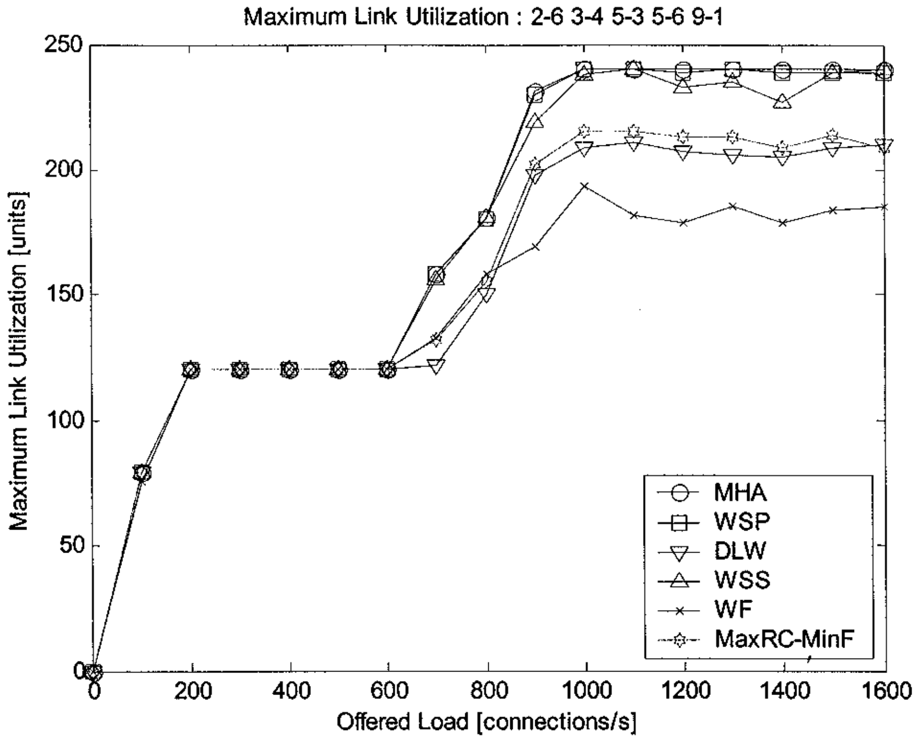
ภาพที่ 54 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5



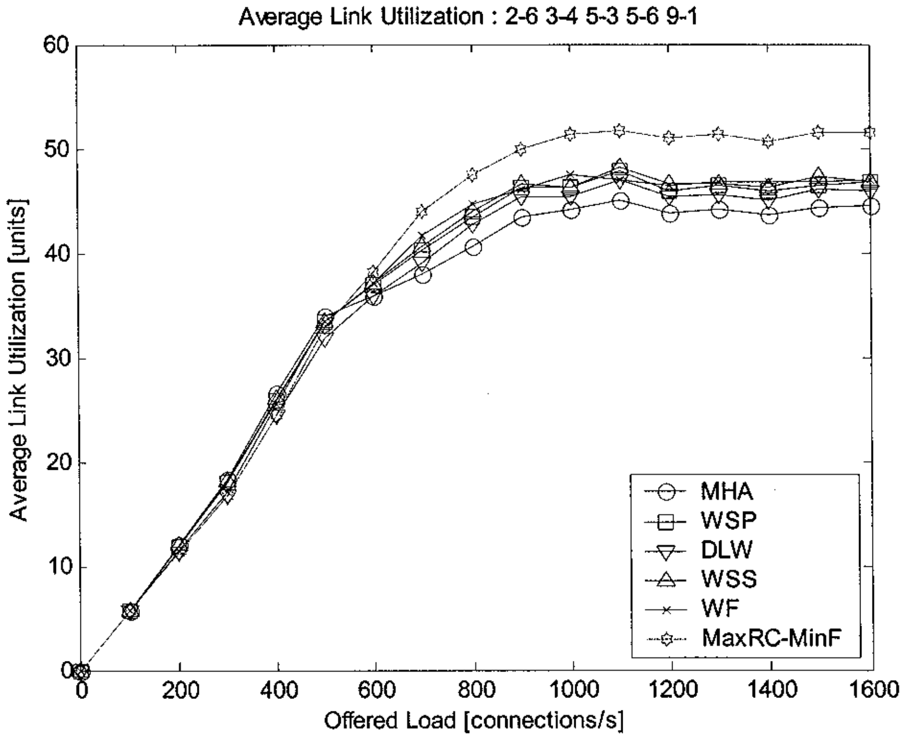
ภาพที่ 55 Rejection Probability ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1



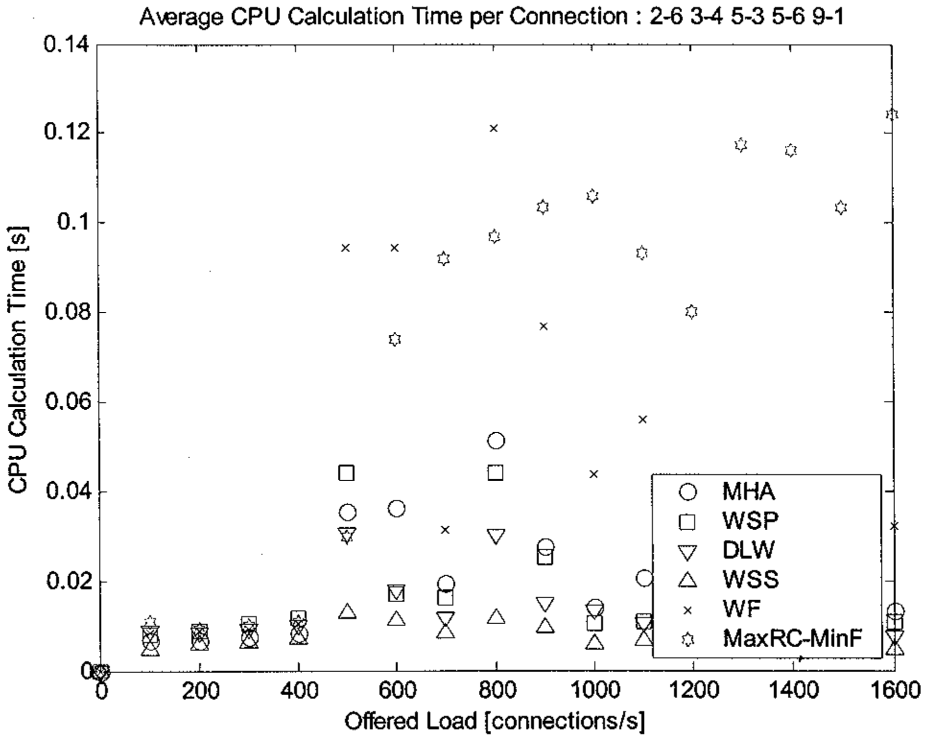
ภาพที่ 56 Total Throughput ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1



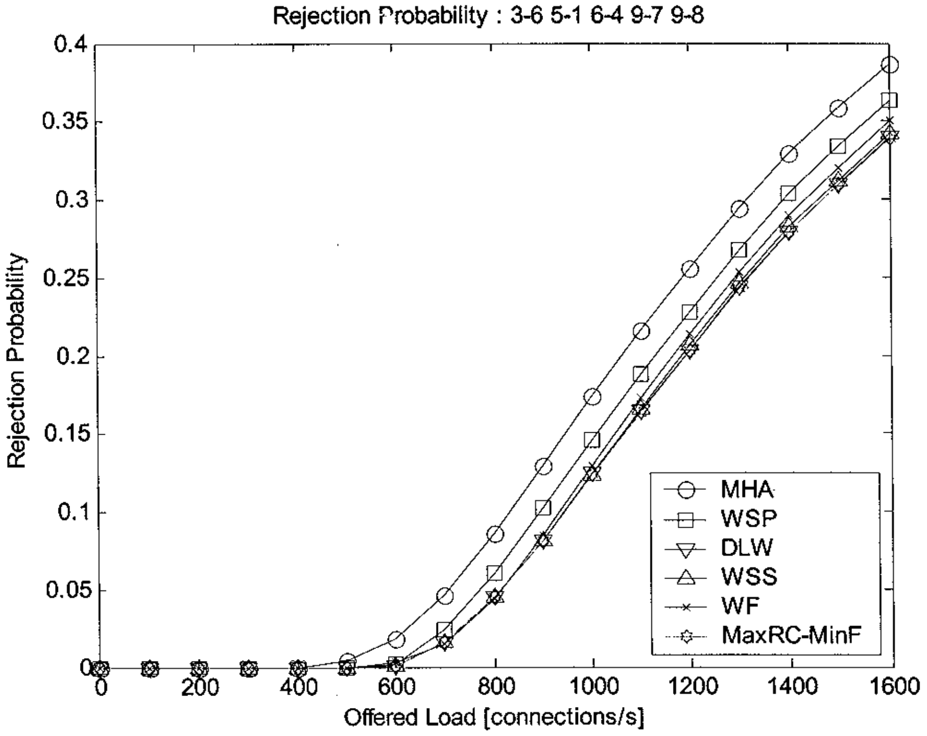
ภาพที่ 57 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1



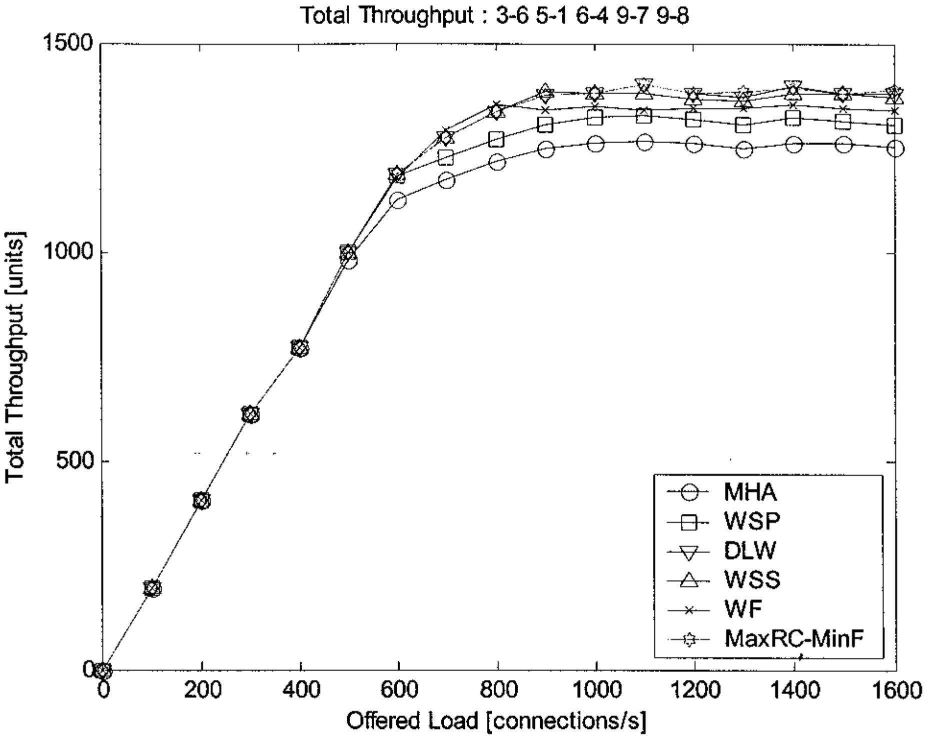
ภาพที่ 58 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1



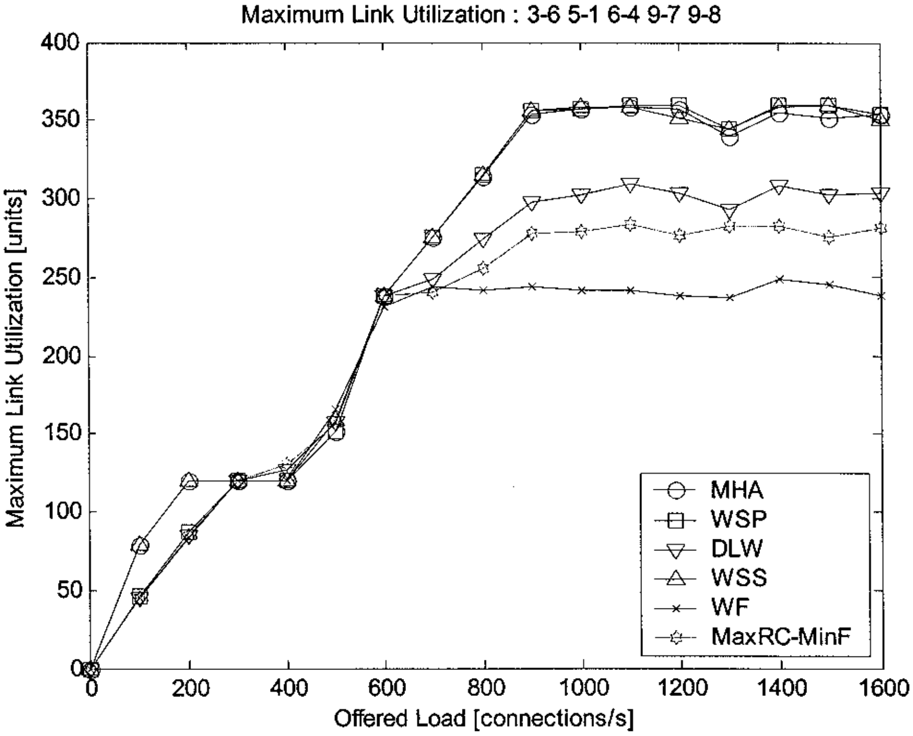
ภาพที่ 59 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1



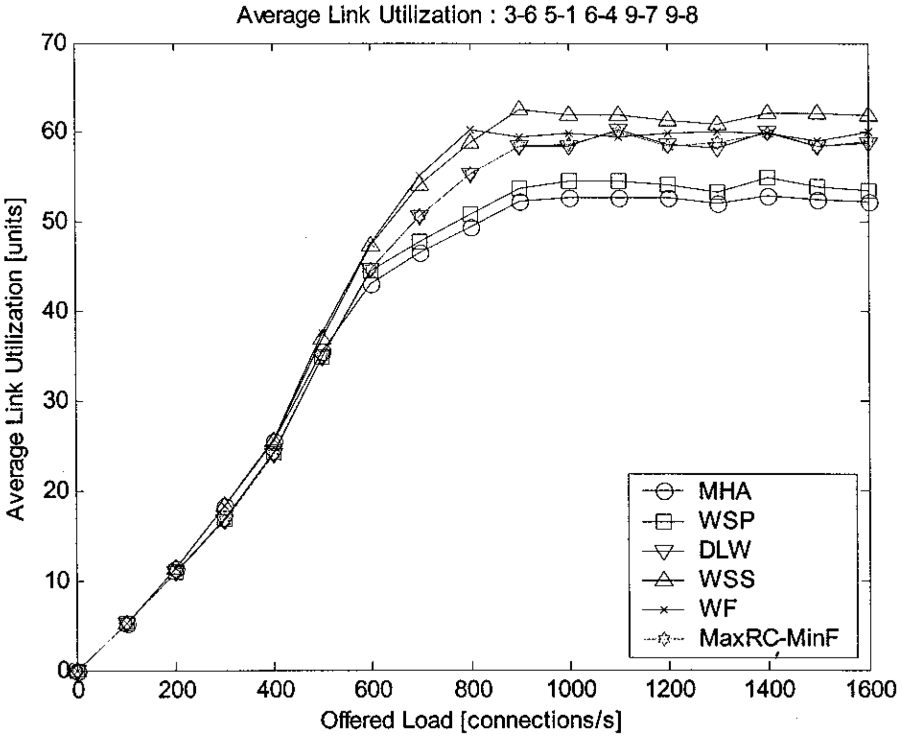
ภาพที่ 60 Rejection Probability ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8



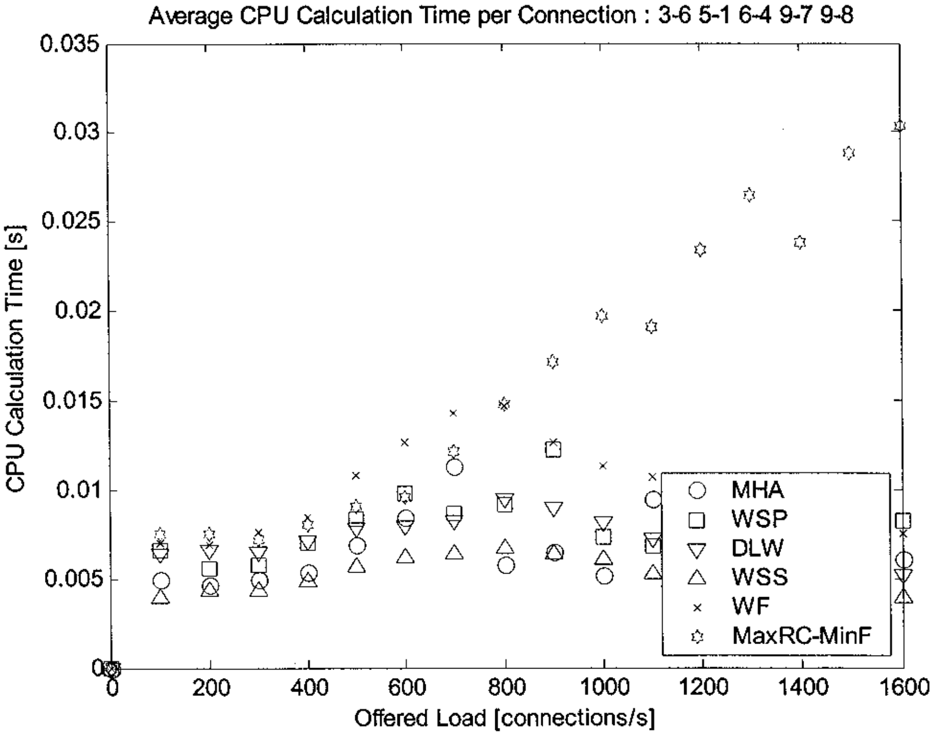
ภาพที่ 61 Total Throughput ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8



ภาพที่ 62 Maximum Link Utilization ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8



ภาพที่ 63 Average Link Utilization ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8



ภาพที่ 64 CPU Calculation Time ของโครงข่าย NET2 คู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8

6. วิเคราะห์ผลการจำลองโครงข่าย NET2

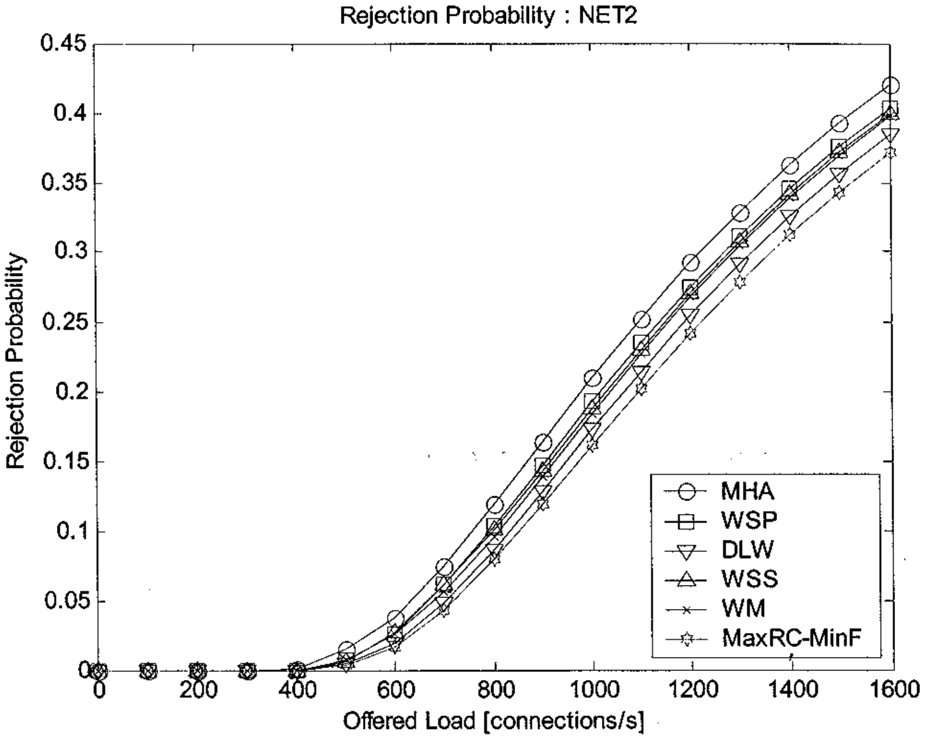
ภาพที่ 40 ถึงภาพที่ 44 แสดงผลการจำลองโครงข่าย NET ระหว่างคู่โหนด 1-6, 5-1, 5-8, 8-3 และ 10-3 ซึ่งจะพบว่าเมื่อโครงข่ายมีขนาดใหญ่ขึ้นวิธีการ MaxRC-MinF และ DLW จะมีค่า Rejection Probability ต่ำที่สุด มีค่า Total Throughput สูงที่สุด และมีค่า Average Link Utilization สูงที่สุด แต่วิธีการ MaxRC-MinF มีค่า CPU Calculation Time สูงสุด

จากภาพที่ 45 ถึงภาพที่ 49 เมื่อทดสอบระหว่างคู่โหนด 1-10, 2-9, 3-8, 4-7 และ 5-6 พบว่าแต่ละวิธีการมีค่า Rejection Probability, Total Throughput และ Average Link Utilization ใกล้เคียงกัน แต่อย่างไรก็ตามวิธีการ MaxRC-MinF มีแนวโน้มที่ดีเมื่อเทียบกับวิธีการอื่นๆ ยกเว้นเวลาในการคำนวณที่ยาวนานกว่าวิธีการอื่นๆ

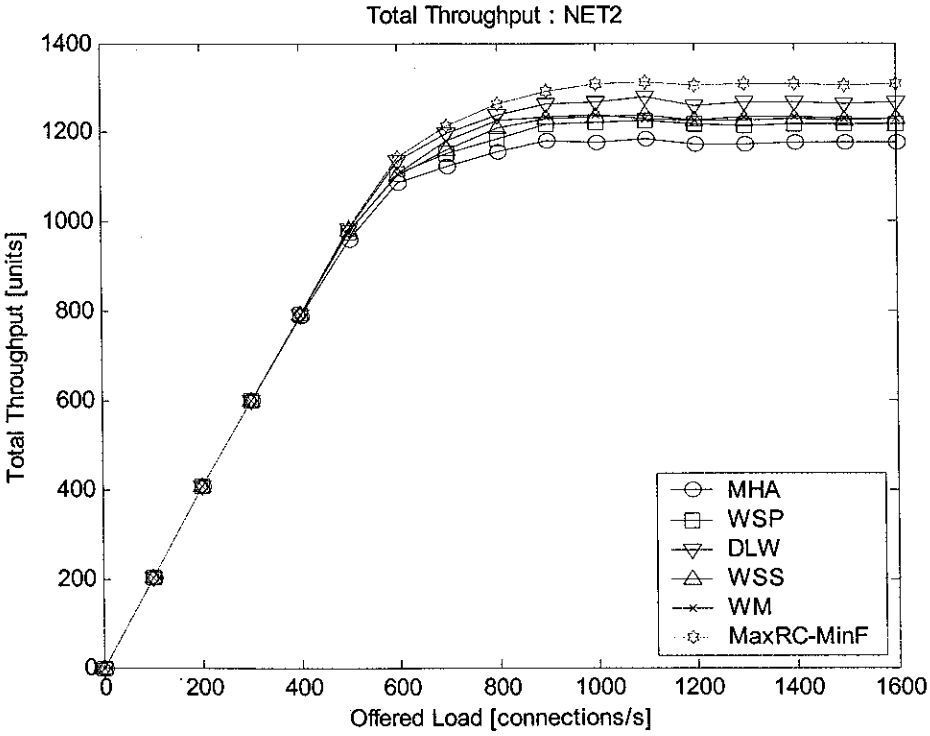
สำหรับผลการทดสอบคู่โหนด 2-1, 2-10, 8-9, 9-3 และ 10-5 พบว่าวิธีการ MaxRC-MinF มีค่า Rejection Probability ต่ำที่สุดในช่วงทราฟฟิก 1200 –1600 connections/s (ภาพที่ 50) มีค่า Total Throughput สูงสุดในช่วงทราฟฟิกโหลด 1000 – 1600 connections/s (ภาพที่ 51) มีค่า Maximum Link Utilization สูงในช่วง 900 – 1600 connections/s (ภาพที่ 52) มีค่า Average Link Utilization ปานกลาง (ภาพที่ 53) และมีค่า CPU Calculation Time per Connection สูงสุด (ภาพที่ 54)

จากผลการทดสอบระหว่างคู่โหนด 2-6, 3-4, 5-3, 5-6 และ 9-1 พบว่าวิธีการ MaxRC-MinF มีค่า Rejection Probability ในระดับต่ำ (ภาพที่ 55) มีค่า Total Throughput สูง (ภาพที่ 56) มีค่า Maximum Link Utilization ปานกลาง (ภาพที่ 57) มีค่า Average Link Utilization ในระดับสูง (ภาพที่ 58) แต่มีค่า CPU Calculation Time ที่สูงสุด (ภาพที่ 59)

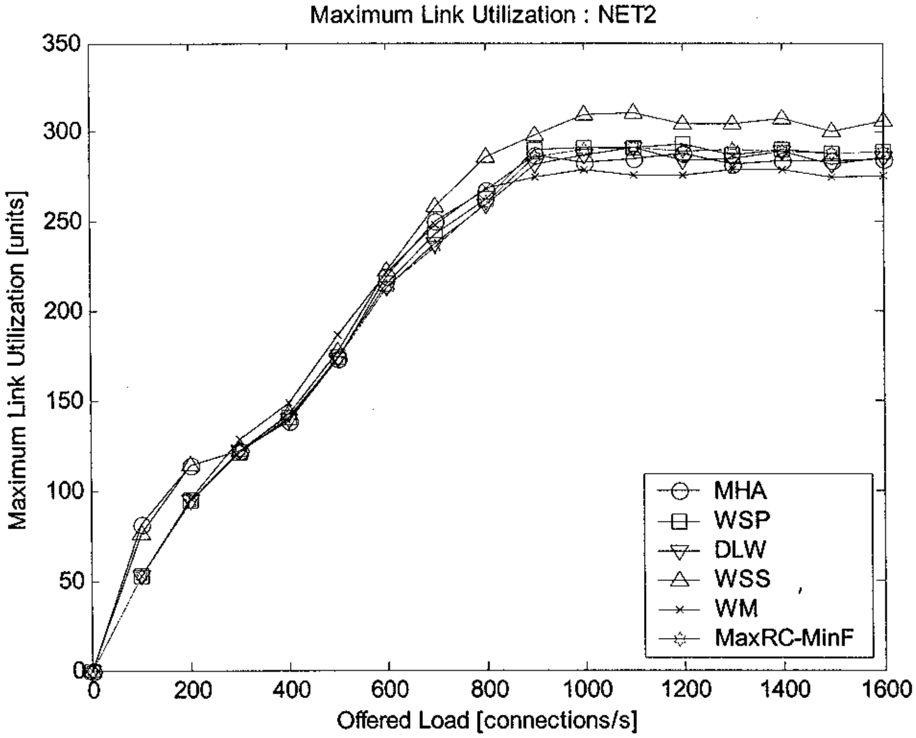
สำหรับการทดสอบสุดท้ายคือคู่โหนด 3-6, 5-1, 6-4, 9-7 และ 9-8 โดยมีผลการจำลองแบบดังภาพที่ 60 ถึงภาพที่ 64 วิธีการที่นำเสนอหรือ MaxRC-MinF มีประสิทธิภาพสูง คือ Rejection Probability ต่ำ Total Throughput สูง มีประสิทธิภาพปานกลางคือ Maximum และ Average Link Utilization และมีประสิทธิภาพด้อยในด้านเวลาในการคำนวณ



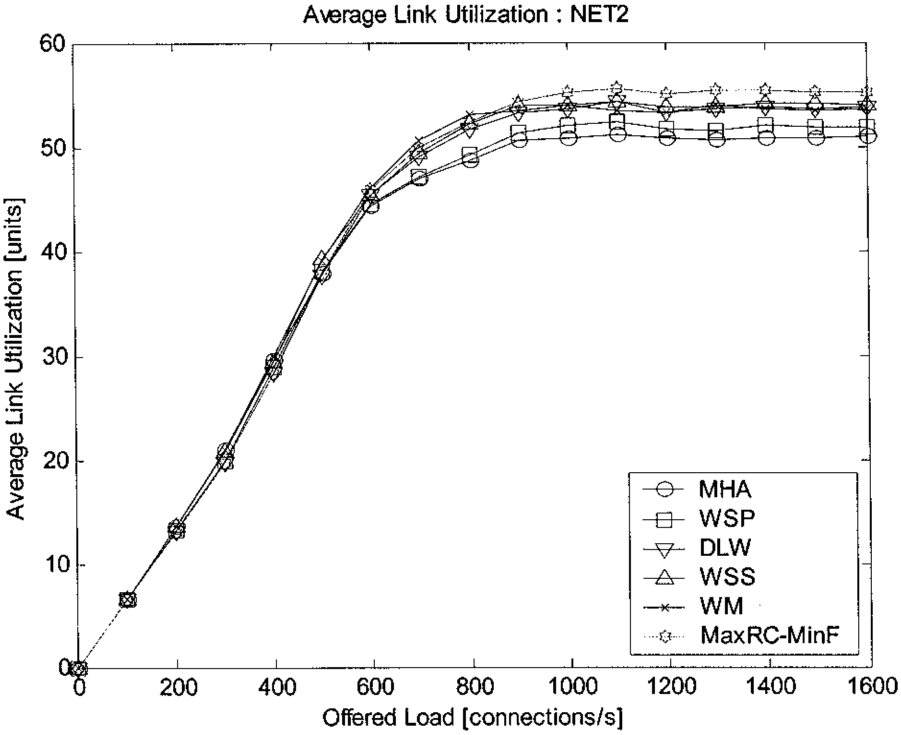
ภาพที่ 65 Rejection Probability โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2



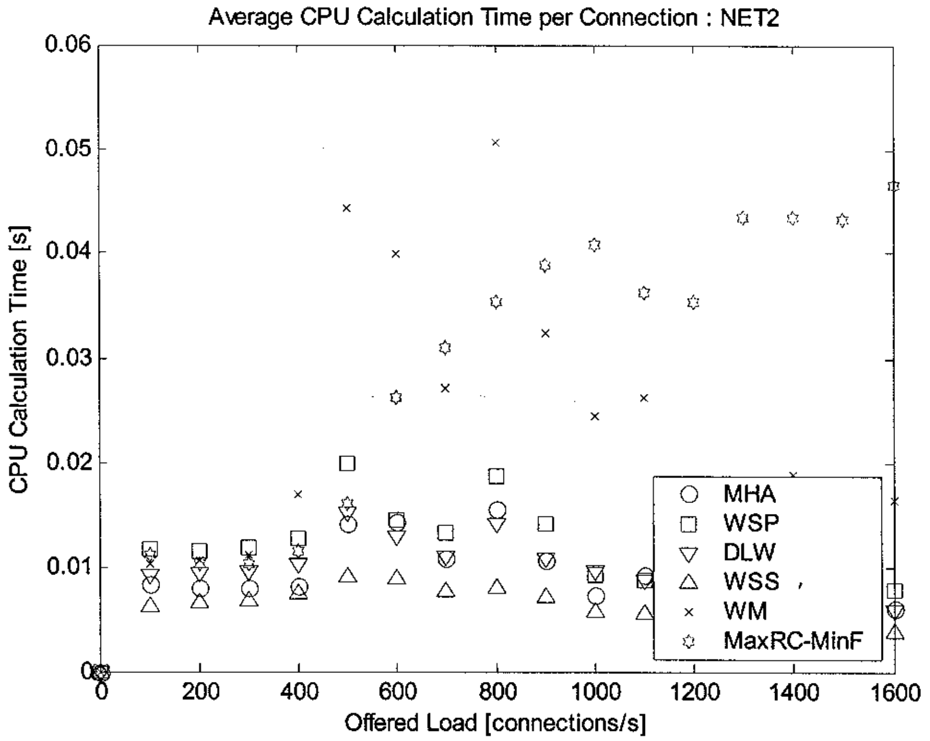
ภาพที่ 66 Total Throughput โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2



ภาพที่ 67 Maximum Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2



ภาพที่ 68 Average Link Utilization โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2



ภาพที่ 69 CPU Calculation Time โดยเฉลี่ยทั้ง 5 คู่โหนดของโครงข่าย NET2

เพื่อให้เห็นผลการจำลองแบบที่ถูกต้องและน่าเชื่อถือมากยิ่งขึ้น จึงได้นำผลการทดสอบทั้ง 5 กรณีมาเฉลี่ย ดังภาพที่ 65 ถึงภาพที่ 69 วิธีการ MaxRC-MinF ให้ค่า Rejection Probability ต่ำที่สุด (ภาพที่ 65) แสดงให้เห็นว่าสามารถทำงานแบบ On-line Routing ได้เป็นอย่างดี ทำให้สามารถรองรับการร้องขอสร้างเส้นทางได้สูงสุด ทำให้มี Total Throughput สูงที่สุด (ภาพที่ 66) คือส่งข้อมูลได้ในปริมาณที่สูงที่สุด นอกจากนี้วิธีการที่นำเสนอยังสามารถใช้งาน Bandwidth ของโครงข่ายได้อย่างมีประสิทธิภาพสูงสุดและมีการกระจายโหลดได้ดีที่สุด สังเกตได้จากค่า Average Link Utilization สูงที่สุด (ภาพที่ 68) ถึงแม้ค่า Maximum Link Utilization จะมีค่าในระดับปานกลางก็ตาม (ภาพที่ 67) สำหรับด้านความซับซ้อนของวิธีการพบว่าวิธีการที่นำเสนอมีความซับซ้อนสูงกล่าวคือมีค่า CPU Calculation Time สูงสุด แต่อย่างไรก็ดีด้วยเทคโนโลยีของ CPU ในปัจจุบันที่ประมวลผลได้รวดเร็วสูงกว่า CPU ที่ใช้ในงานวิจัยมาก (งานวิจัยใช้ CPU Pentium4 1.5 GHz) ค่า CPU Calculation Time ที่ได้นี้จึงมีค่าที่สามารถยอมรับให้ใช้งานได้

หากมองในด้าน Scalability จะพบว่าเมื่อโครงข่ายมีขนาดใหญ่ขึ้น วิธีการที่นำเสนอยังคงให้ประสิทธิภาพที่ดีโดยเฉพาะด้าน Rejection Probability, Total Throughput และ Link Utilization

บทที่ 5
บทสรุป

บทนี้จะกล่าวถึงสรุปผลการออกแบบและผลการทดสอบวิธีการที่นำเสนอ และแสดงข้อเสนอแนะในการพัฒนาต่อไป

1. สรุปผลการวิจัย

วิธีการหาเส้นทางในโครงข่าย MPLS ที่ดีควรมีคุณลักษณะดังนี้

1. มีความน่าจะเป็นในการปฏิเสธการร้องขอเส้นทาง (Rejection Probability) ต่ำ
2. มีค่าวิสัยสามารถรวม (Total Throughput) สูง
3. มีค่าการใช้งานลิงค์สูงสุด (Maximum Link Utilization) สูง
4. มีค่าการใช้งานลิงค์เฉลี่ย (Average Link Utilization) สูง
5. และมีค่าเวลาในการประมวลผล (CPU Calculation Time) ต่ำ

ซึ่งงานวิจัยนี้ได้เสนอวิธีการที่ใช้หลักการหาเส้นทางที่สั้นที่สุด ที่มีจำนวนโฟล (การรบกวน) ต่ำ มีแบนด์วิดท์คงเหลือสูง และมีความจุของลิงค์สูง ชื่อว่า Maximize Residual bandwidth and link Capacity – Minimize total Flow (MaxRC-MinF) routing algorithm ซึ่งสามารถบรรลุความต้องการทั้ง 4 ประการข้างต้น คือ มีความน่าจะเป็นในการปฏิเสธการร้องขอเส้นทางต่ำ มีค่าวิสัยสามารถรวม ค่าการใช้งานลิงค์สูงสุด และค่าการใช้งานลิงค์เฉลี่ยสูง แต่ไม่สามารถบรรลุวัตถุประสงค์ในข้อสุดท้ายได้ คือมีค่าเวลาในการประมวลผลสูงกว่าวิธีการอื่นๆ

2. ข้อเสนอแนะ

ข้อเสนอแนะรวมถึงแนวทางในการพัฒนาในอนาคต ได้แก่

1. ทดสอบกับโครงข่าย และทราฟฟิกหลายๆ ประเภทให้มากขึ้น
2. นำพารามิเตอร์ของคุณภาพการให้บริการ (QoS) อื่นๆ มาคำนวณด้วย เช่น เวลาประวิง (Delay)
3. นำเอาความเสมอภาค (Fairness) เข้ามาพิจารณาด้วย

บรรณานุกรม

[1] Awduche, D. 1999. MPLS and traffic engineering in IP networks (Online). IEEE Commum. 37 (12).

[2] Davie, B. and Rekhter, Y. (2000). MPLS Technology and Application. Academic Press. Morgan Kaufmann Publishers.

[3] Koushik Kar, Murali Kodialam and T.V. Lakshman. 2000. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Application (Online). IEEE Journal on Selected Areas in Communication, Vol.18, No.12.

[4] Cisco System, 2001. MPLS Architecture Overview, Technical Paper.

[5] Rosen, E. Viswanathan, A. and Callon, R. 2001, January. Multiprotocol label switching architecture. RFC3031.

[6] Subhash Suri, Marcel Waldvogel, Daniel Bauer, and Priyank Ramesh Warkhede. 2002. Profile-Based Routing and Traffic Engineering. Elsevier Science Computer Communication, Vol. 25.

[7] Ying-Xiao Xu and Gen-Du Zhang. 2002. Models and Algorithms of QoS-Based Routing with MPLS Traffic Engineering. (Online)," 5th IEEE International Conference on High Speed Networks and Multimedia Communications.

[8] Halimi, A. Hendling, K. Statovei.-Halimi, B. and Bengi, K. 2002. Impact of Different Link Cost Metrics on the Performance of Routing Algorithm," The 8th International Conference on Communication Systems (ICCS).

[9] Yang, Yi. Zhang, Lei. Jogesh K. Muppala and Samuel T. Chansom. 2003. Bandwidth-delay constrained routing Algorithm. Computer Network Journal of Elslvier.

[10] Lim, S.H. Yaacob, M.H. Phang, K.K. and Ling, T.C. 2004, February. Traffic engineering enhancement to QoS-OSPF in DiffServ and MPLS networks (Online). IEE Proc.-Commun, Vol.151, No.1.

[11] Guerin, R. Orda, A. Williams, D. 1997. Qos routing mechanisms ans OSPF Extensions (Online). IEEE Globecom'97, Vol.3.

[12] Huan Pham and Bill Lavery. 2002 New Dynamic Link Weight Algorithm for On-line Calculation of LSPs in MPLS Networks (Online). IEEE Proceeding.

[13] Sa-Ngiamsak, W. Thipchaksurat, S. and Varakulsiripunth, R. 2004. A Bandwidth-Based Constraint Routing Algorithm for Multi-Protocol Label Switching Networks. ICACT.

[14] Fabio Ricciato. 2004. On-line Routing of MPLS Tunnels with Time-Varying Bandwidth Profiles. HPSR. Workshop on High Performance Switching and Routing.

[15]

ประวัติผู้เขียน

ปิยาภรณ์ กระฉอดนอก เกิดเมื่อ 9 กันยายน 2517 สำเร็จการศึกษาวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม จากมหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อปี 2540 และต่อมาได้ศึกษาระดับปริญญาโทต่อด้วยทุนส่งเสริมผู้มีความสามารถพิเศษเป็นอาจารย์ของมหาวิทยาลัยเทคโนโลยีสุรนารี โดยสำเร็จการศึกษาวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า(ไฟฟ้าสื่อสาร) จากจุฬาลงกรณ์มหาวิทยาลัย ปัจจุบันเป็นอาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี งานวิจัยที่สนใจได้แก่การจัดการทราฟฟิกในเครือข่ายการสื่อสาร