การจัดการการจราจรข้อมูลบนเครือข่ายโอเอสพีเอฟ ด้วยวิธี Branch Exchange

กายรัฐ เจริญราษฎร์[1]* และ ทิชากร สำรองทรัพย์[2]

บทคัดย่อ

ระบบเครือข่ายคอมพิวเตอร์และผู้ใช้งานที่มีมากขึ้น ส่งผลให้การสื่อสารผ่านเครือข่ายเพิ่มมากขึ้น และทำให้ประสิทธิภาพของเครือข่ายลดลง ผู้วิจัยจึงได้ศึกษาเกี่ยวกับโพรโทคอลชี้เส้นทางที่ปัจจุบันที่นิยมใช้และแพร่หลายมากที่สุดในระบบเครือข่ายไอพี ซึ่งก็คือ Open Shortest Path First (OSPF) และเพื่อที่จะทำให้เครือข่ายมีประสิทธิภาพที่ดีผู้วิจัยจึงนำวิธี Branch Exchange มาทดลองประยุกต์ใช้ในการหาค่าน้ำหนักที่เหมาะสมของช่องสัญญาณ โดย Branch Exchange นั้นจะแก้ไขปัญหาในเรื่องของประสิทธิภาพและเวลาที่ใช้ประมวลผล ซึ่งพบว่าประสิทธิภาพของวิธี Branch Exchange นั้นขึ้นอยู่กับสัดส่วนของจำนวนสายสัญญาณที่เลือกเพื่อทำการสลับค่าน้ำหนักกับสายสัญญาณเส้นอื่นๆ ต่อจำนวนสายสัญญาณทั้งหมด โดยหากสัดส่วนของสายสัญญาณนี้มีค่ามาก พบว่าประสิทธิภาพการไหลของวิธี Branch Exchange จะใกล้เคียงกับวิธี Linear Programming แต่เวลาในการประมวลผลจะใกล้เคียงหรือมากกว่าวิธี Linear Programming แต่ถ้าหากสัดส่วนของสายสัญญาณมีค่าน้อย ประสิทธิภาพการไหลวิธี Branch Exchange จะลดลง แต่เวลาที่ใช้ในการประมวลผลจะเร็วกว่าวิธี Linear Programming มาก

คำสำคัญ: Branch Exchange โอเอสพีเอฟ ค่าน้ำหนัก

[1] อาจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

[2] นิสิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 0-3435-2853 อีเมล: kairat.j@ku.ac.th

รับเมื่อ 3 มีนาคม 2557 ตอบรับเมื่อ 2 ตุลาคม 2557

OSPF Network Traffic Engineering by Branch Exchange Method

Kairat Jaroenrat[1]* and Tichakorn Samrongsub[2]

Abstract

The increasing communication over networks leads to a greater number of users, and consequently, the poorer network performance. The routing protocol which is commonly used and most widespread in the Internet called Open Shortest Path First (OSPF) is studied. In order to improve the network performance and the processing time, the Branch Exchange method is applied to determine the appropriate weight of the network links. The network performance by the Branch Exchange method is based on the ratio of the number of chosen cables to the overall cables. When the ratio is high, the data flow performance is almost the same as by the Linear Programming with extra processing time. When the ratio is low, the data flow performance is slightly lower from the Linear Programming method with much less processing time.

Keywords: Branch Exchange, OSPF, Weight

1 Lecturer, Division of Computer Engineering, Faculty of Engineering at Kamphaengsaen, Kasetsart University.

2 Student, Division of Computer Engineering, Faculty of Engineering at Kamphaengsaen, Kasetsart University.

* Corresponding Author, 0-3435-2853 E-mail: kairat.j@ku.ac.th

Received 3 March 2014; Accepted 2 October 2014

1. บทนำ

ระบบเครือข่ายคอมพิวเตอร์เข้ามามีบทบาทสำคัญในชีวิตประจำวันของเรามากขึ้น มีการใช้งานคอมพิวเตอร์และต้องการที่จะสื่อสารกันเพิ่มมากขึ้น ส่งผลให้ต้องมีการเชื่อมต่อคอมพิวเตอร์ที่ต้องการสื่อสารเหล่านั้นเข้าด้วยกัน ดังนั้นเพื่อเพิ่มความสามารถของระบบให้มีประสิทธิภาพสูงขึ้นและลดค่า Delay ของระบบโดยรวมลง จำเป็นต้องพิจารณาถึงกลไกการทำงานของโพรโทคอลชี้เส้นทางในเครือข่ายเราเตอร์ ซึ่งโพรโทคอลชี้เส้นทางในปัจจุบันที่นิยมใช้และแพร่หลายมากที่สุดตัวหนึ่งคือ Open Shortest Path First (OSPF) [1] เนื่องจากมีจุดเด่นในหลายด้าน ไม่ว่าจะเป็นการใช้อัลกอริทึมในการค้นหาเส้นทางด้วยตัวของมันเอง การอัพเดทเส้นทางจากโหนดข้างเคียง การอัพเดทเส้นทางแบบ Link State ที่ไม่กิน Bandwidth ของระบบเครือข่ายตลอดเส้นทาง ทำให้โพรโทคอลนี้นิยมใช้งานกันอย่างมาก

ในอดีตที่ผ่านมามีแนวคิดในการนำวิธีการทางคณิตศาสตร์และอัลกอริทึมอื่นๆ มาเพิ่มประสิทธิภาพการไหลของข้อมูลในเครือข่ายไอพี โดยการหาค่าน้ำหนักสายสัญญาณที่เหมาะสมซึ่งก็คือการทำ Traffic Engineering บนเครือข่ายโอเอสพีเอฟ [2]-[4] หรือแม้กระทั่งบน Service Network [5] ที่มีการควบคุมดีเลย์ นอกจากนั้นยังมีงานวิจัยการหาค่าน้ำหนักของเส้นทางที่คุ้มค่าที่สุดในโครงข่ายโอเอสพีเอฟด้วยวิธีโปรแกรมเชิงเส้น โดย กฤษณะ และคณะ [6] ที่เป็นการนำเอาวิธีโปรแกรมเชิงเส้นมาเพื่อใช้กำหนดค่าน้ำหนักเส้นทางเพื่อจัดการจราจรของข้อมูล ให้การไหลของข้อมูลนั้นมีประสิทธิภาพที่ดีนั้น พบว่าเป็นวิธีที่ใช้เวลานานมากในการประมวลผล และไม่สามารถใช้กับเครือข่ายขนาดใหญ่ได้ซึ่งเหมือนกับงานวิจัยของ Holmberg and Yuan [7] ที่ประยุกต์ใช้วิธี Simulated Annealing (SA) นอกจากนี้งานวิจัยการใช้วิธีอาณานิคมมดในการตั้งค่าน้ำหนักให้แก่เครือข่ายโอเอสพีเอฟ โดยเอกวุฒิและกายรัฐ [8] ก็เป็นวิธีที่มีประสิทธิภาพในการจัดการจราจรข้อมูลที่ดีขึ้นกว่าการใช้ค่าน้ำหนักเริ่มต้นเพียงเล็กน้อย แต่สามารถใช้กับเครือข่ายขนาดใหญ่ได้

ผู้วิจัยจึงได้ศึกษาวิธีการ Branch Exchange [9] ซึ่งมีความซับซ้อนน้อยกว่าวิธีโปรแกรมเชิงเส้นและสามารถกำหนดพารามิเตอร์ในการคำนวณเพื่อให้สามารถใช้เวลาประมวลผลในช่วงที่ยอมรับได้ ซึ่งจากการศึกษาโครงงานการวางแผนระบบจำหน่ายกำลังไฟฟ้าที่ดีที่สุด โดยวิธีการ Branch Exchange โดยโสมพันธุ์ [10] ซึ่งเป็นการนำเอาวิธีการ Branch Exchange เข้ามาแก้ปัญหาเกี่ยวกับการวางแผนระบบจำหน่ายกำลังไฟฟ้าให้มีค่าใช้จ่ายที่น้อยลงกว่าเดิมแต่มีประสิทธิภาพที่ดีขึ้น และงานวิจัยของ Ababei และ Kavasseri [11] ซึ่งเป็นการนำวิธี Branch Exchange มาช่วยให้กระบวนการ Reconfiguration เร็วขึ้น พบว่าวิธี Branch Exchange เป็นวิธีที่มีประสิทธิภาพดีและใช้เวลาประมวลผลไม่นาน ผู้วิจัยจึงได้ประยุกต์วิธีการ Branch Exchange และทำการทดสอบและเปรียบเทียบประสิทธิภาพการตั้งค่าน้ำหนักให้แก่เครือข่าย ว่าจะให้ค่าน้ำหนักที่เหมาะสมและสามารถลดดีเลย์ [12] ภายในระบบเครือข่ายได้ดี และใช้เวลาประมวลผลน้อยกว่าวิธีโปรแกรมเชิงเส้นหรือไม่ งานวิจัยนี้จึงเป็นการเปรียบเทียบประสิทธิภาพทั้งด้านดีเลย์และด้านของเวลาการประมวลผลโดยเทียบกับวิธี Linear Programming

2. หลักการพื้นฐาน

ในการวิจัยเรื่องการใช้วิธีการ Branch Exchange นี้ ผู้พัฒนาได้ศึกษาทฤษฎีและหลักการที่เกี่ยวข้องกับการพัฒนางานระบบมาประยุกต์ใช้กับงานวิจัย โดยมีหัวข้อต่างๆ ดังนี้

2.1 Open Shortest Path First

โอเอสพีเอฟเป็นโพรโทคอลชี้เส้นทางตัวหนึ่งนิยมใช้กันอย่างแพร่หลายมากที่สุดในระบบเครือข่าย เนื่องจากมีจุดเด่นในหลายด้าน เช่น การที่เป็นโพรโทคอลที่เส้นทางแบบโพรโทคอลการค้นหาเส้นทางแบบ Link State อัลกอริทึมโอเอสพีเอฟที่สามารถค้นหาเส้นทางได้ด้วยตัวเองนี้ใช้การคำนวณเส้นทางด้วยอัลกอริทึม Dijkstra

โดยคำนวณและเลือกเส้นทางที่มีค่าน้ำหนัก (Cost) รวมน้อยที่สุด [13] ซึ่งโดยปกติแล้วจะกำหนดค่าน้ำหนักเริ่มต้น (Default Cost) ของสายสัญญาณ ด้วยค่าที่คำนวณได้จากสมการ (1)

$$\text{Cost} = \frac{10^8}{Bandwidth} \quad (1)$$

เมื่อ Cost คือค่าน้ำหนักของสายสัญญาณ

Bandwidth คือค่าอัตราการรองรับการไหลของข้อมูล หรือค่าความกว้างของช่องสัญญาณ

2.2 Branch Exchange

Branch Exchange [9] เป็นวิธีการออกแบบแบบกระจาย เริ่มต้นด้วยการเชื่อมสถานีทั้งหมดด้วยรูปแบบการเชื่อมต่อแบบใดแบบหนึ่ง จากนั้นพยายามเพิ่ม (Add) ลด (Drop) สายสัญญาณ หรือลดสายบางเส้นแล้วแทนด้วยสายสัญญาณเส้นใหม่ (Exchange) เพื่อให้ได้โครงข่ายที่มีคุณสมบัติดีขึ้น ซึ่งโดยปกติแล้ววิธี Branch Exchange นั้นสามารถนำมาแก้ปัญหาการออกแบบเครือข่ายด้วยการเพิ่ม ลด และแลกเปลี่ยนสายสัญญาณเข้าไปในเครือข่าย โดยการแลกเปลี่ยนสายสัญญาณนี้ใช้วิธีทางคณิตศาสตร์ที่เรียกว่าวิธีการจัดลำดับ (Permutation) [14] แต่เนื่องจากวัตถุประสงค์ของการทำวิจัยนี้คือต้องการปรับค่า Weight ของเครือข่ายให้ดีขึ้น ผู้วิจัยจึงดัดแปลงวิธีการแลกเปลี่ยนสายสัญญาณไปเป็นการแลกเปลี่ยนค่าน้ำหนักของสายสัญญาณ

การจัดลำดับเพื่อแลกเปลี่ยนที่กล่าวมานั้น คือการจัดสิ่งของทั้งหมดหรือบางส่วนของสิ่งของชุดหนึ่ง ๆ โดยถือว่าลำดับมีความสำคัญ ถ้าจำนวนสายเส้นสัญญาณในเครือข่ายมีความแตกต่างกันทั้งหมด n สิ่ง จะได้คำตอบมากถึง n! ดังนั้นเพื่อลดจำนวนคำตอบและเพิ่มความเร็วในการประมวลผลผู้วิจัยใช้วิธีเลือกสายสัญญาณที่มีปริมาณการไหลของข้อมูลคับคั่งที่สุด มา r อันดับ ($r \le n$) จะได้จำนวนคำตอบดังแสดงในสมการ (2)

$$P_r^n = \frac{n!}{(n-r)!} \quad (2)$$

เมื่อ P_r^n = วิธีการจับลำดับทั้งหมดของสายเส้นสัญญาณที่เป็นไปได้

n = จำนวนสายสัญญาณทั้งหมดของเครือข่าย

r = จำนวนสายสัญญาณที่มีภาระการไหลของข้อมูลมาก r อันดับ

ซึ่งการจัดลำดับของสายสัญญาณนี้ใช้สำหรับหาจำนวนวิธีทั้งหมดที่เป็นไปได้ของเครือข่าย แต่เราสามารถใช้วิธีการที่เรียกว่า การจัดหมู่ (Combination) มาเพื่อใช้คำนวณหาวิธีการทั้งหมดแทนโดยลำดับไม่มีความสำคัญ ซึ่งการจัดหมู่นี้ คือการจัดสิ่งของทั้งหมดหรือบางส่วนโดยถือว่าลำดับไม่มีความสำคัญ [14] โดยถ้าจำนวนสายสัญญาณในเครือข่ายมีความแตกต่างกันทั้งหมด n เส้น และเลือกสายสัญญาณที่มีปริมาณการไหลของข้อมูลคับคั่งที่สุด มา r เส้น จะได้จำนวนวิธีการทั้งหมดดังแสดงในสมการ (3)

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \quad (3)$$

2.3 Objective Function

เมื่อพิจารณากราฟของระบบโครงข่าย G = (N,A) โดยกำหนดอัตราการไหลหรือความกว้างของช่องสายสัญญาณ (C_a) โดยที่ $a \in A$ โดยกำหนด d_{st} แทน Traffic Flow ระหว่าง s และ t และให้ f_a^{st} แสดงปริมาณ Traffic Flow จาก s ไป t บนช่องสายสัญญาณ a

ปริมาณ traffic l_a บน link $a \in A$ ซึ่งเป็นผลรวมของ f_a^{st} ใน [2] นั้นได้เสนอวิธีการวัดประสิทธิภาพของระบบโครงข่ายไว้โดยใช้ฟังก์ชันต้นทุน (Cost Function) ดังสมการ (4)

$$Min \quad \Phi = \sum_{a \in A} \phi_a(l_a, C_a) \quad (4)$$

ในขณะที่ ϕ_a (l_a, C_a) อิงกับค่า Delay ในตัวแบบ M/M/1 ในทฤษฎี Queuing ซึ่งได้จากสมการ (5)

$$\phi_a(l_a, C_a) = l_a / (C_a - l_a) \quad (5)$$

2.4 Optimum Solution

Optimum Solution เป็นการหาคำตอบที่ดีที่สุดโดยหาค่าฟังก์ชันต้นทุน (Cost Function) ที่น้อยที่สุด ดังแสดงในสมการ (6)

$$Min \quad \Phi = \sum_{a \in A} \phi_a(l_a, C_a) \quad (6)$$

โดย

$$\phi_a(l_a, C_a) = \begin{cases} \phi_a \geq l_a \\ \phi_a \geq 3l_a - \frac{2}{3}C_a \\ \phi_a \geq 10l_a - \frac{16}{3}C_a \\ \phi_a \geq 70l_a - \frac{178}{3}C_a \\ \phi_a \geq 500l_a - \frac{1468}{3}C_a \\ \phi_a \geq 5000l_a - \frac{19468}{3}C_a \end{cases}$$

เมื่อ Φ คือค่า Delay Cost ทั้งหมดในเครือข่าย

ϕ_a คือค่า Delay Cost ที่ใช้ในสายสัญญาณ a

l_a คือค่าภาระการไหลของสายสัญญาณ a หรือค่า Load ของสายสัญญาณ a

C_a คือค่าความกว้างของช่องสัญญาณ (Bandwidth) ของสายสัญญาณ a

ภายใต้ Flow Conservation ในสมการ (7)

$$\sum_{u:(u,v)\in A} f_{u,v}^{st} - \sum_{u:(u,v)\in A} f_{v,u}^{st} = \begin{cases} d_{st} \ if \ \ v = t \\ -d_{st} \ if \ \ v = s : v, s, t \in N \\ 0 \ otherwise \end{cases}$$

โดย $f_a^{st} \geq 0 : a \in A, \ t \in N$ (7)

2.5 การวิเคราะห์การไหลของข้อมูล

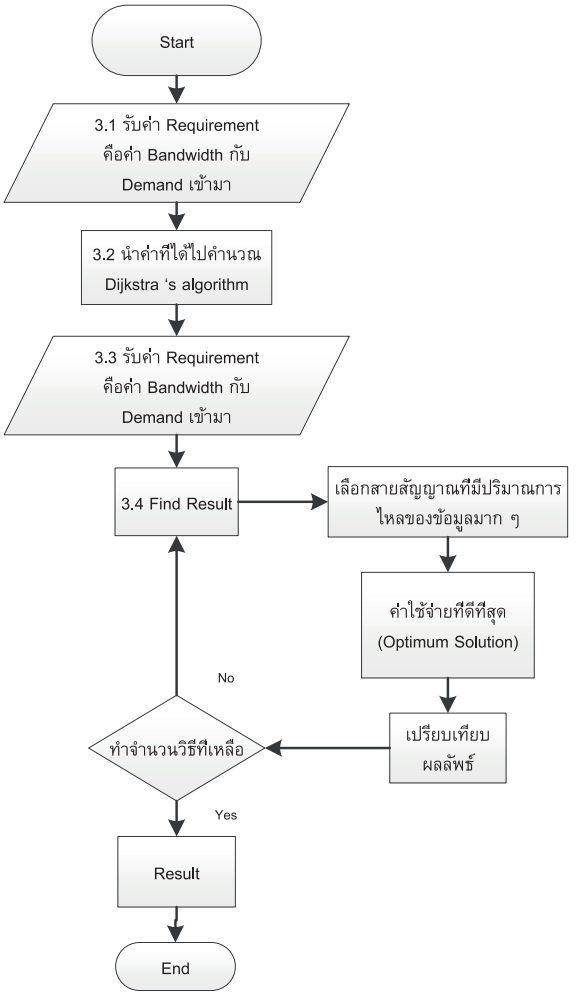
การวิเคราะห์การไหลของข้อมูลเป็นการนำความต้องการเชิงสมรรถนะของการประยุกต์ใช้เครือข่ายมาวิเคราะห์ร่วมกับข้อมูลตำแหน่งของอุปกรณ์ในเครือข่าย อุปกรณ์แม่ข่ายและสถานีปลายทาง การไหลของข้อมูลถือเป็นข้อมูลที่มีความสำคัญมากในการออกแบบระบบเครือข่าย เพราะการไหลของข้อมูลใช้แสดงถึงภาระที่เครือข่ายจะต้องรองรับ การไหลของข้อมูลถือเป็นข้อมูลสำคัญที่ระบุทิศทางของข้อมูลพร้อมกับคุณสมบัติเชิงสมรรถนะของการไหล ในการวิเคราะห์ระบบเพื่อการออกแบบเครือข่ายจะใช้การไหลเป็นตัวแทนความต้องการในการใช้งานเครือข่ายของโปรแกรมหรือการส่งผ่านข้อมูลแต่ละชนิด [6]

3. วิธีการดำเนินการวิจัย

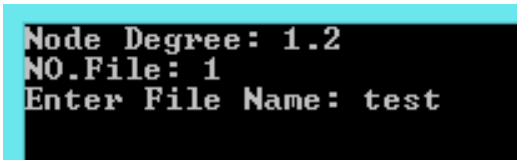
การทดลองนี้ทำบนระบบจำลองเครือข่ายซึ่งผู้วิจัยพัฒนาขึ้นมาด้วยภาษา C++ โดยทำการทดลองกับเครือข่ายที่มีจำนวน 10 โหนด ที่สร้างขึ้นมาแบบสุ่มโดยมีดีกรี 1.0, 1.1, 1.2, 1.3, 1.5, 1.8, 2.0, 2.2, 2.5 ชนิดละ 5 เครือข่าย รวมทั้งสิ้น 45 เครือข่าย เริ่มจากการนำความต้องการในการไหลที่กำหนดขึ้นแบบสองทิศทาง (ความต้องการไหลขาไปและขากลับไม่จำเป็นต้องเท่ากัน) ไปคิดภาระในการไหลข้อมูลในช่องสัญญาณ จากนั้นก็ทำการเลือกสายสัญญาณที่มีค่าภาระการไหลสูงสลับค่าน้ำหนักกับสายเส้นอื่นๆ ด้วยวิธี Branch Exchange ดังแสดงรายละเอียดในรูปที่ 1

กระบวนการหาค่าน้ำหนักสายสัญญาณที่แสดงในรูปที่ 1 มีขั้นตอนดังต่อไปนี้

3.1 รับค่า Requirement ซึ่งประกอบไปด้วยค่า Capacity หรือ Bandwidth คือค่าความกว้างของช่องสัญญาณของสายสัญญาณที่เชื่อมแต่ละคู่โหนด และรับค่า Demand ซึ่งก็คือปริมาณความต้องการการไหลของข้อมูล โดยผู้วิจัยสร้างขึ้นมาด้วยโปรแกรม Generate Network (Gen) [15] ซึ่งพัฒนาด้วยภาษา C++ ดังตัวอย่างแสดงในรูปที่ 2 โดยกำหนดค่า Node Degree ของเครือข่าย และ



รูปที่ 1 ภาพรวมของระบบ



รูปที่ 2 โปรแกรม Generate Network

จำนวนไฟล์หรือก็คือจำนวนเครือข่ายที่จะสร้าง จากนั้นตั้งชื่อเครือข่าย ซึ่งจะได้ผลลัพธ์ออกมาในรูปแบบของ Text File ที่ประกอบไปด้วยค่าของ Demand กับค่าของ Capacity ที่จะใช้ในขั้นตอนถัดไป

3.2 กำหนดค่าน้ำหนักเริ่มต้นให้กับสายสัญญาณแต่ละเส้นตามสมการ (1) แล้วใช้ Dijkstra's Algorithm ในการแก้ปัญหาเพื่อหาเส้นทางที่สั้นที่สุดจากโหนดหนึ่งไปยังโหนดอื่นๆ จากนั้นคำนวณหาค่าภาระการไหลของข้อมูลในสายสัญญาณแต่ละเส้นด้วยการหาผลรวมของการไหลข้อมูลทั้งหมดในสายสัญญาณแต่ละเส้น

3.3 รับค่า r ซึ่งคือจำนวนของสายสัญญาณที่จะมีการสลับค่าน้ำหนักกับสายสัญญาณเส้นอื่นๆ โดยจะเลือกสายสัญญาณที่มีค่าภาระการไหลของข้อมูลมากที่สุด r อันดับ เพราะสายสัญญาณที่มีปริมาณการไหลของข้อมูลที่คับคั่งมากเป็นสายที่ควรเกิดการเปลี่ยนของค่าน้ำหนักเพื่อให้ภาระการไหลของข้อมูลลดลง

3.4 จากนั้นทำการสลับระหว่างค่าน้ำหนักของแต่ละสายสัญญาณในจำนวน r เส้นกับค่าน้ำหนักของสายสัญญาณเส้นอื่นทุกเส้นที่เป็นไปได้ เช่นหากเครือข่ายมีทั้งหมด 4 เส้น และกำหนดค่า r เป็น 2 จะหมายถึงเลือกสายสัญญาณที่มีความคับคั่งของข้อมูลมากที่สุดสองเส้นจากที่คำนวณไว้ในข้อ 3.2 แล้วนำค่าน้ำหนักของแต่ละเส้นในสองเส้นนี้ไปสลับกับค่าน้ำหนักของอีกสามเส้นที่เหลือ และหาค่า Delay Cost ของเครือข่ายในทุกกรณี โดยใช้สมการ (6) ซึ่งจำนวนวิธีทั้งหมดที่เป็นไปได้คำนวณได้จากสมการ (2) นั้นคือ 12 กรณี จากนั้นจึงเปรียบเทียบหากรณีที่ให้ค่า Delay Cost ของเครือข่ายน้อยที่สุด ซึ่งคือกรณีที่ให้ค่าประสิทธิภาพ (Cost Performance) ที่ดีที่สุดนั่นเอง

4. ผลการดำเนินงาน

การทดสอบหาชุดค่าน้ำหนักและค่า Delay Cost ที่เหมาะสม เพื่อให้มีความสามารถในการค้นหาคำตอบได้อย่างมีประสิทธิภาพ ด้วยการคำนวณเปรียบเทียบประสิทธิภาพตามสมการ (8) ได้ผลลัพธ์ดังแสดงในตารางที่ 1 และ 2

$$\text{Cost Performance } (\%) = \frac{Dc \ x \ 100}{c} \qquad (8)$$

Dc คือค่า Delay Cost ในโครงข่ายที่คำนวณค่าน้ำหนักด้วยวิธีอื่นที่ผู้ทดลองสนใจ

c คือค่า Delay Cost ของโครงข่ายที่กำหนดค่าน้ำหนักด้วยวิธีมาตรฐานที่ใช้เป็นตัวเปรียบเทียบ

ตารางที่ 1 ตัวอย่างค่าประสิทธิภาพเฉลี่ยของวิธีโปรแกรมเชิงเส้น เมื่อเทียบกับวิธีการใช้ค่าน้ำหนักเริ่มต้นของเครือข่ายขนาด 10 โหนด ที่มีโหนดดีกรีเท่ากับ 2.5

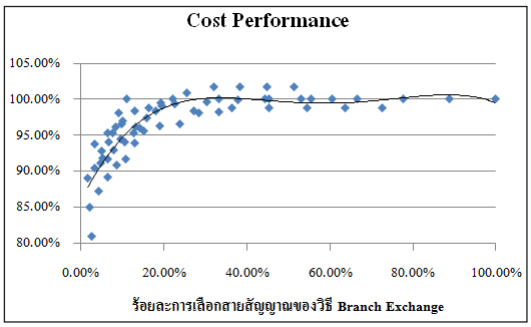
Delay Cost	Time (s)	Cost of Default Method	Cost Performance
2,690.80	2,554.40	3,261.33	121.46 %

ตารางที่ 2 ตัวอย่างค่าประสิทธิภาพและเวลาเฉลี่ยของวิธี Branch Exchange เมื่อเทียบกับวิธีโปรแกรมเชิงเส้น ของเครือข่ายขนาด 10 โหนด ที่มีโหนดดีกรีเท่ากับ 2.5

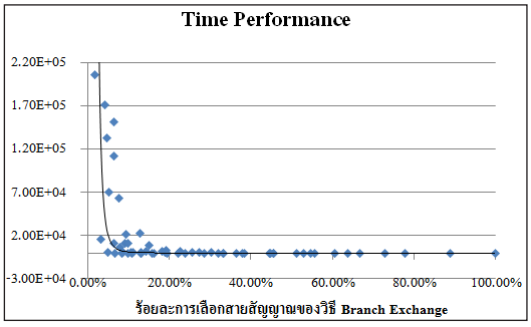
r	Delay Cost	Cost Performance	Time Performance
1	3,025.46	89.04 %	2.06e7 %
2	2,979.20	90.40 %	1.57e6 %
3	2,896.80	92.85 %	9.12e4 %
4	2,860.93	94.02 %	4,08e3 %
5	2,797.06	96.12 %	1.98e2 %
6	2,773.46	96.96 %	9.97e0 %

4.1 ผลการทดลอง

ตารางที่ 1 และตารางที่ 2 เป็นตัวอย่างผลการทดลองของเครือข่ายที่มี Node Degree เท่ากับ 2.5 ที่เป็นค่าเฉลี่ยจาก 5 เครือข่ายที่ทำการทดสอบ โดยตารางที่ 1 เป็นผลเปรียบเทียบค่า Delay Cost ของวิธี Linear Programming กับวิธีที่ใช้ค่าน้ำหนักเริ่มต้น โดยแสดงเวลาที่ใช้ในการคำนวณด้วยวิธี Linear Programming ด้วยส่วนตารางที่ 2 เป็นผลจากวิธีการ Branch Exchange โดยแปรผันจำนวนสายสัญญาณ (r) ที่ต้องการเลือกสลับค่าน้ำหนักด้วยวิธีการ Branch Exchange จากนั้นเปรียบเทียบค่า Delay Cost กับวิธีการคำนวณแบบ Linear Programming โดยคำนวณผลตามสมการ (8) และแสดงผลในคอลัมน์ Cost



รูปที่ 3 ประสิทธิภาพเชิงดีเลย์ของวิธี Branch Exchange เมื่อเทียบกับวิธีโปรแกรมเชิงเส้น



รูปที่ 4 ประสิทธิภาพเชิงเวลาของวิธี Branch Exchange เมื่อเทียบกับวิธีโปรแกรมเชิงเส้น

Performance ของตาราง นอกจากนั้นยังเปรียบเทียบประสิทธิภาพในด้านของเวลาที่ใช้ในการประมวลผลแสดงในคอลัมน์ Time Performance ของตาราง ด้วยวิธีการคำนวณตามสมการ (9) อีกด้วย

$$\text{Time Performance (\%)} = \frac{Dt \ x \ 100}{t} \qquad (9)$$

Dt คือเวลาที่ใช้ในการประมวลผลด้วยวิธีโปรแกรมเชิงเส้น

T คือเวลาที่ใช้ในการประมวลผลด้วยวิธี Branch Exchange

4.2 วิเคราะห์ผลการทดลอง

จากรูปที่ 3 และ 4 แสดงค่าประสิทธิภาพ (Performance) ของวิธี Branch Exchange เทียบกับวิธี Linear Programming

ในแง่ของค่า Delay Cost และแง่เวลา โดยแปรผันค่าร้อยละการเลือกสายสัญญาณ ซึ่งก็คือสัดส่วนของจำนวนสายสัญญาณ (r) ที่ต้องการเลือกสลับค่าน้ำหนักด้วยวิธีการ Branch Exchange ต่อจำนวนสายสัญญาณทั้งหมด

รูปที่ 3 เป็นผลการเปรียบเทียบค่า Delay Cost ของเครือข่ายที่กำหนดค่าน้ำหนักด้วยวิธี Branch Exchange ด้วยการเลือกสายสัญญาณจำนวน r เส้นต่อจำนวนสายทั้งหมด กับค่า Delay Cost ของเครือข่ายที่กำหนดค่าหนักด้วยวิธี Linear Programming พบว่าวิธี Branch Exchange ที่มีสัดส่วนการเลือกสายสัญญาณมาก จะส่งผลให้ประสิทธิภาพของเครือข่ายหรือค่า Delay Cost ดีขึ้นใกล้เคียงเครือข่ายที่ได้จากวิธี Linear Programming เนื่องจากมีการประมวลผลกรณีที่เป็นไปได้จำนวนมากขึ้น ทำให้มีโอกาสเจอกรณีที่ให้ผลลัพธ์ดีได้มากกว่า ส่วนรูปที่ 4 เป็นผลการเปรียบเทียบค่าเวลาที่ใช้ในการประมวลผลของวิธี Branch Exchange กับเวลาประมวลผลของวิธี Linear Programming โดยแปรผันสัดส่วนการเลือกสายสัญญาณจำนวน r เส้นต่อจำนวนสายทั้งหมด พบว่าวิธี Branch Exchange ที่มีสัดส่วนการเลือกสายสัญญาณมากขึ้น จะส่งผลให้เวลาที่ใช้ในการประมวลผลมากขึ้นจนใกล้เคียงกับวิธี Linear Programming ที่สัดส่วนการเลือกสายสัญญาณสูงๆ แต่หากสัดส่วนการเลือกสายสัญญาณมีค่าน้อยลง เวลาในการประมวลผลจะน้อยลง และน้อยลงกว่าวิธี Linear Programming ตั้งแต่ไม่กี่สิบเท่าจนถึงหลายหมื่นเท่า เนื่องจากยิ่งเลือกจำนวนสายสัญญาณน้อยลง ก็ยิ่งจะทำให้จำนวนกรณีที่เป็นไปได้ลดน้อยลง ตามสมการ (2) การประมวลผลจึงสามารถทำได้เร็วขึ้นมาก

5. สรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

การทดสอบนำหลักการของวิธี Branch Exchange ซึ่งนำหลักการจัดลำดับ และการจัดหมู่มาใช้กับการสลับค่าน้ำหนักให้แก่เครือข่ายโอเอสพีเอฟ แล้วนำมาวิเคราะห์การไหลของข้อมูล ประสิทธิภาพและเวลา ซึ่งผลที่ได้ที่มีการสลับค่าน้ำหนักของเส้นทางแล้ว จะมีประสิทธิภาพที่ดีกว่าเครือข่ายเริ่มต้น โดยโปรแกรมจะทำการรับค่าความกว้างของช่องสัญญาณและปริมาณการไหลของข้อมูลมาวิเคราะห์ แล้วรับค่า r คือจำนวนเส้นที่มีภาระการไหลของข้อมูลมากที่สุด r อันดับ โดยพิจารณาจากค่าภาระงาน (Load) ของแต่ละลิงค์ แล้วทำวิธี Branch Exchange คือ การจัดหมู่ (Combination) ก่อน เพื่อให้ได้วิธีการในการจัดหมู่ทั้งหมดก่อน แล้วนำแต่ละวิธีของการจัดหมู่มาทำการจัดลำดับ (Permutation) ใหม่ แล้วหาค่า Delay Cost ที่น้อยที่สุดและเวลาที่ใช้ในการประมวลผล

จากผลการดำเนินงานจะเห็นว่า ยิ่งสัดส่วนของจำนวนสายสัญญาณ (r) มากๆ ยิ่งทำให้ค่า Delay Cost ในเครือข่ายนั้นมีค่าน้อยลง ซึ่งหมายถึงการได้ค่าน้ำหนักที่มีประสิทธิภาพมากขึ้น แต่เวลาที่ใช้ในการประมวลผลจะมากขึ้นตามไปด้วย เมื่อนำไปเปรียบเทียบกับวิธี Linear Programming ปรากฏว่า ประสิทธิภาพที่ได้น้อยกว่าหรือเท่ากับวิธี Linear Programming แต่เวลาที่ใช้ในการประมวลผลเร็วกว่า แต่ถ้าสัดส่วนของจำนวนสายสัญญาณ (r) มีค่ามากๆ ก็ทำให้เวลาประมวลผลนานกว่าวิธี Linear Programming และได้ค่า Delay Cost ที่แย่กว่า

ดังนั้น สรุปได้ว่าการหาค่าน้ำหนักของสายสัญญาณเพื่อใช้ในโครงข่ายโอเอสพีเอฟที่มีประสิทธิภาพสูงที่สุด คือวิธีการ Linear Programming แต่วิธีการนี้เป็นวิธีการที่ใช้เวลานานมากในการประมวลผล จึงไม่เหมาะกับการใช้กับโครงข่ายโอเอสพีเอฟขนาดใหญ่ ดังนั้นผู้วิจัยจึงขอเสนอการใช้วิธี Branch Exchange ในการหาค่าน้ำหนักของสายสัญญาณ ซึ่งเป็นวิธีที่สามารถเลือกได้ว่าจะเลือกประสิทธิภาพในเรื่องของเครือข่าย (Delay Cost) กับเวลาในการประมวลผล ซึ่งสามารถใช้กับเครือข่ายขนาดใหญ่ได้โดยสามารถกำหนดค่า r ให้มีค่าน้อย แต่ก็ยังคงได้ประสิทธิภาพที่ใกล้เคียงกับวิธีการ Linear Programming

5.2 ข้อเสนอแนะ

การคำนวณด้วยวิธี Branch Exchange สามารถคำนวณแยกกันในกรณีแต่ละกรณีที่เป็นไปได้ ดังนั้นหาก

สามารถใช้การคำนวณแบบขนาน (Parallel Processing) จะสามารถได้ผลลัพธ์เร็วขึ้นเป็นอย่างมาก รวมถึงหากสามารถให้โปรแกรมเก็บผลการประมวลผลของแต่ละกรณีไว้ได้ แม้ระหว่างการประมวลผลจะเกิดเหตุไม่คาดฝันขึ้น ไม่ว่าจะเป็นไฟดับ หรือเครื่องหยุดทำงาน ก็จะไม่ต้องคำนวณผลลัพธ์ในกรณีที่เคยคำนวณแล้วใหม่

6. กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนจาก คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

เอกสารอ้างอิง

[1] J. Moy, (April 1998). "*OSPF Version 2*," RFC 2328. [Online]. Available: http://www.ietf. org/rfc/rfc2328.txt

[2] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 519-528.

[3] W. Ben-Ameur, E. Gourdin, B. Liau, and N. Michel, "Designing Internet Networks," in *Proc. DRCN2000*, April 2000, pp. 56-61.

[4] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag*., vol. 40, pp. 118-124, 2002.

[5] M. Phayung, B. Pongsarun, "The optimization and intelligence approach for traffic engineered design of a service network," in *Proc. FGCT*, 2012, pp. 143 -149.

[6] K. Lanthong and K. Jaroenrat, "OSPF weight optimization by linear programming method," in *9th National Kasetsart University KamphaengSaen Conference*, Kasetsart University KamphaengSaen Campus, NakhonPathom Province, 2012 (in Thai).

[7] K. Holmberg and D. Yuan, "Optimization of Internet Protocol Network Design and Routing," *Networks*, vol. 43, no. 1, pp. 39-53, 2004.

[8] E. Chairat and K. Jaroenrat, "Using Ant Colony Algorithm for Weight Setting in OSPF Network," *8th National Conference on Computing and Information Technology*, Pattaya City, 2012 (in Thai).

[9] P. CharnKeitkong, *Computer Network Design*, 1st Edition, Pathumthani: Rangsit University, 2007 (in Thai).

[10] S. Khuantham, "Optimal Distribution System Planning Using Branch Exchange Technique," Master Thesis, Department of Electrical Engineering, Kasetsart University, 2003 (in Thai).

[11] C. Ababei and R. Kavasseri, "Efficient Network Reconfiguration Using Minimum Cost Maximum Flow-Based Branch Exchanges and Random Walks-Based Loss Estimations," *IEEE Trans. on Power Systems*, vol. 26, no. 1, pp. 30-37, Feb 2011.

[12] K. Jaroenrat, P. CharnKeitkong, "On Routing Performance of MENTOR Algorithm," *WSEAS Transaction on Communications*, Issue 9, vol. 5, pp. 1769-1776, 2006.

[13] R. Cahn, *Wide Area Network Design*, San Francisco, CA: Morgan Kaufmann Publisher, 1998.

[14] S. Sinsomboonthong, *Engineering Statistics*, 4th Edition, Bangkok: Jamjuree Product, 2010 (in Thai).

[15] K. Lanthong and K. Jaroenrat, (2012). *Software: Generate Network*. [Online]. Available: http://pirun.ku.acth/~fengkrj/conference/2555/kps2555/gen.rar.