

ฐานข้อมูลแบบสามเล่ม - รูปแบบใหม่ของระบบกระจาย

นางสาว จุติพันธ์ กิจเจริญทรัพย์



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์

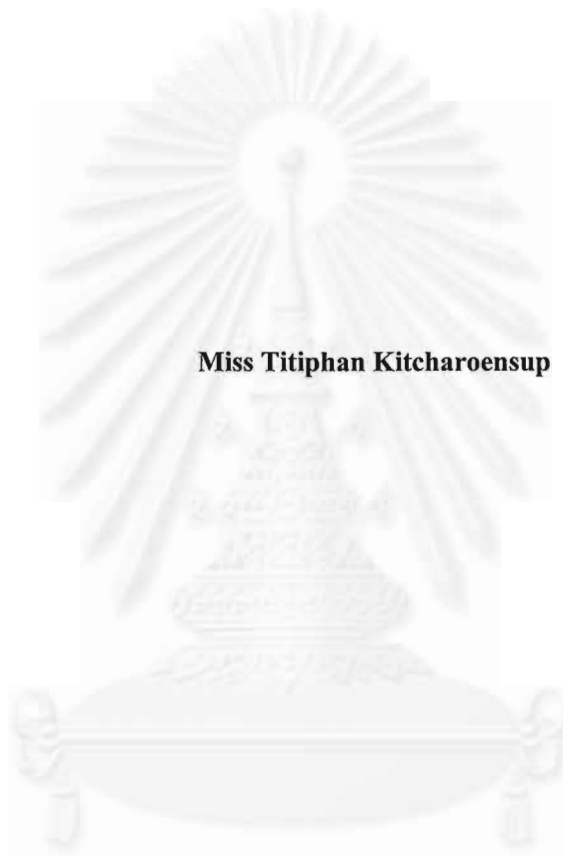
บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2542

ISBN 974-3327-54-2

ลิขสิทธิ์ของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

THREE-TIER DATABASE - A NEW PARADIGM FOR DISTRIBUTED SYSTEM



Miss Titiphan Kitcharoensup

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computational Science**

Department of Mathematics

Graduate School

Chulalongkorn University

Academic Year 1999

ISBN 974-3327-54-2

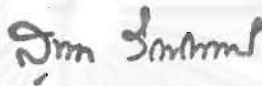
หัวข้อวิทยานิพนธ์ ฐานข้อมูลแบบสามเส้า - รูปแบบใหม่ของระบบกระจาย

โดย นางสาว จูติพันธ์ กิจเจริญทรัพย์

ภาควิชา คณิตศาสตร์

อาจารย์ที่ปรึกษา อาจารย์ ดร.พีระพนธ์ โสฬศสถิตย์

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต



..... คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. สุชาดา กิระนันท์)

คณะกรรมการสอบวิทยานิพนธ์



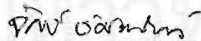
..... ประธานกรรมการ

(รองศาสตราจารย์ ดร. ชิดชนก เหลือสินทรัพย์)



..... อาจารย์ที่ปรึกษา

(อาจารย์ ดร. พีระพนธ์ โสฬศสถิตย์)



..... กรรมการ

(อาจารย์ ดร. จักษ์ อิศวานนท์)

พิมพ์ต้นฉบับบทความวิจัยฉบับวิทยานิพนธ์ภายในกรอบสี่เหลี่ยมนี้เพียงแผ่นเดียว

ผู้พิมพ์ กิจเจริญทรัพย์: ฐานข้อมูลแบบสามเต้า – รูปแบบใหม่ของระบบกระจาย (THREE-TIER DATABASE - A NEW PARADIGM FOR DISTRIBUTED SYSTEM) อ. ที่ปรึกษา: อ. ดร. พีระพนธ์ โสฬศสถิตย์, 71 หน้า. ISBN 974-3327-54-2

ปัจจุบันการบริการด้านฐานข้อมูลได้มีการใช้สถาปัตยกรรมแบบสองเต้า หรือที่เรียกกันโดยทั่วไปว่า Client/Server การทำงานของสถาปัตยกรรมแบบสองเต้า แบ่งเป็น 2 ชั้น คือ ชั้นที่ 1 Client ซึ่งเป็นส่วนที่ผู้ใช้ ใช้ประมวลผลในการติดต่อหรือส่งคำร้องขอผ่านไปยังฐานข้อมูล และชั้นที่ 2 Server เป็นเป็นส่วนของการจัดการฐานข้อมูล การจัดเก็บฐานข้อมูล งานประยุกต์ และการประมวลผลข้อมูลแบบกระจาย การที่จำนวนของข้อมูลบน Server และ จำนวนของ Client ขยายตัวขึ้นเรื่อยๆ จะส่งผลกระทบต่อสมรรถนะในการใช้งาน นั่นคือ ทำให้การทำงานช้าลง และการจัดการฐานข้อมูลมีความซับซ้อนมากขึ้น เพราะ ภาระงานส่วนใหญ่ตกอยู่กับ Server อย่างเดียว จากปัญหาในการขยายตัวของ Client/Server ที่กล่าวมาข้างต้น จึงทำให้เกิดแนวความคิดใหม่ที่จะช่วยแก้ปัญหาดังกล่าว โดยการนำเสนอสถาปัตยกรรมที่เรียกว่า สามเต้า (Three-tier) การทำงานของของสถาปัตยกรรมแบบสามเต้า แบ่งเป็น 3 ชั้น คือ ชั้นที่ 1 Client, ชั้นที่ 2 Application Server และ ชั้นที่ 3 Database Server ซึ่งจะลดหน้าที่ของ Server ให้น้อยลง และชั้นของการบริการที่เพิ่มขึ้นนี้เรียกว่า Application Server ทำหน้าที่ในการจัดการเพื่อให้ Client และ Database Server สามารถติดต่อถึงกันได้ โดย Application Server จะควบคุมการติดต่อสื่อสารระหว่าง Client กับ Database Server และ Application Server ยังเก็บข้อมูลต่างๆ เกี่ยวกับระบบอีกด้วย ซึ่งวิธีการดังกล่าวจะช่วยให้ระบบการประมวลผลมีประสิทธิภาพสูงขึ้น

ในการวิจัยนี้จะทำการเปรียบเทียบการปฏิบัติงานระหว่างสถาปัตยกรรมแบบสองเต้า กับสถาปัตยกรรมแบบสามเต้า โดยวัดจากค่า Delay รวม ที่ได้มาจากการหาเส้นทางที่มีระยะทางที่สั้นที่สุด ซึ่งมีการนำเสนอ Algorithm เพื่อใช้ในการหาเส้นทางที่มีระยะทางที่สั้นที่สุด ที่มีความซับซ้อนของข้อมูลที่ส่งผ่านในเครือข่ายให้เป็น Polynomial และมีการพัฒนาการทำงานของ Algorithm โดยใช้แนวความคิดที่จะเก็บข้อมูลที่เปลี่ยนแปลงในช่วงระยะเวลาที่กำหนด เพื่อนำมาหาค่าประมาณของระยะทางที่เกิดขึ้น โดยอาศัยหลักการของ Fuzzy Logic

จุฬาลงกรณ์มหาวิทยาลัย

คณิตศาสตร์

ภาควิชา
วิทยาการคอมพิวเตอร์
สาขาวิชา
ปีการศึกษา 2542

ลายมือชื่อนิสิต กิจเจริญทรัพย์
ลายมือชื่ออาจารย์ที่ปรึกษา *Perple Sph...*
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

#4072245423 : MAJOR COMPUTATIONAL SCIENCE

KEY WORD: THREE-TIRE / DISTIDUTED COMPUTING

TITIPHAN KITCHAROENSUP : THREE-TIER DATABASE - A NEW PARADIGM FOR DISTRIBUTED SYSTEM. THESIS ADVISOR : PERAPHON SOPHATSATHIT, Ph.D. 71pp. ISBN 974-3327-54-2

Conventional database services are based on two-tier architecture, which is commonly known as Client/Server architecture. The first tier, a front-end client application, queries and instructs the second tier, a back-end database server to retrieve and process client's requests. As data volume and number of clients grow, service performance is degraded and data management becomes increasingly complex and time-consuming. This study presents a new approach for automated service performance optimization and efficient data management operation using a three-tier architecture. In the three-tier architecture, a support tier, called application server, is placed between the client and the server. This application server holds the information about the system, which facilitates administration of the interaction between the client and the server by means of the Distributed Bellman-Ford Algorithm (DBA). The information obtained can then be used to assist the application of DBA in reliable distributed computing.

The purpose of this study is to compare the performance between two-tier architecture and three-tier architecture by measuring total delay from a source node to a destination node, based on the DBA routing algorithm. A proposed empirical method for finding the minimum delay path in polynomial message complexity is presented. Performance improvement of the proposed method is carried out through simulation of this three-tier architecture with the help of fuzzy set approach to model the uncertainty of delay information. Result statistics are collected and analyzed to verify the effectiveness of the proposed method.

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....คณิตศาสตร์.....

สาขาวิชา.....วิทยาการคอมพิวเตอร์.....

ปีการศึกษา.....2542.....

ลายมือชื่อนิสิต.....Titiphan Kitcharoensup.....

ลายมือชื่ออาจารย์ที่ปรึกษา.....Peraphon Sophatsathit.....

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะสำเร็จลุล่วงลงมิได้ หากปราศจากความช่วยเหลือเป็นอย่างดีจากบุคคลและหน่วยงานหลายๆ ฝ่าย ซึ่งบุคคลท่านแรกและผู้เขียนซาบซึ้ง และอยากขอบพระคุณเป็นอย่างยิ่งคือ อาจารย์ ดร.พิระพนธ์ โสพัศสถิตย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ท่านกรุณาให้คำปรึกษา คำแนะนำ และช่วยกลั่นกรองผลงานตั้งแต่เริ่มต้นจนเสร็จสิ้น ด้วยความดูแลเอาใจใส่อย่างใกล้ชิดเสมอมา รวมทั้งสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช) ซึ่งเป็นหน่วยงานราชการที่ให้ความอนุเคราะห์ทางด้านอุปกรณ์การศึกษาและเงินทุนสนับสนุนการค้นคว้าและวิจัย ผู้วิจัยจึงขอขอบพระคุณมา ณ ที่นี้ด้วย

ท้ายนี้ ผู้วิจัยใคร่ขอขอบพระคุณสมาชิกในครอบครัว และเพื่อนๆทุกคน ที่เป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญตาราง	ณ
สารบัญภาพ	ญ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	4
1.3 ขอบเขตของโครงการ.....	4
1.4 สมมติฐานการวิจัย.....	5
1.5 ประโยชน์ที่คาดว่าจะได้รับการวิจัย.....	5
1.6 ลำดับขั้นตอนและวิธีการดำเนินการวิจัย.....	6
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	8
2.1 DISTRIBUTED BELLMAN-FORD ALGORITHM (DBF)	8
2.2 APPROXIMATE DISTRIBUTED BELLMAN-FORD ALGORITHM (AADBF)	9
2.3 A FUZZY SET DELAY REPRESENTATION FOR COMPUTER NETWORK ROUTING (FSDR)	9
บทที่ 3 วิธีดำเนินการวิจัย	11
3.1 การวิเคราะห์ระบบ	11
3.2 การออกแบบโครงสร้าง NETWORK.....	12
3.3 การออกแบบ PROCEDURE	13
บทที่ 4 ผลการวิจัย.....	19
4.1 เครือข่ายแบบสองเสา.....	20
4.2 เครือข่ายแบบสามเสา.....	26

บทที่ 5 สรุปผลการวิเคราะห์ข้อมูล อภิปรายผล และข้อเสนอแนะ	32
5.1 สรุปผลการวิจัย.....	32
5.2 อภิปรายผล	40
5.3 ข้อเสนอแนะ.....	40
รายการอ้างอิง.....	42
ภาคผนวก	43
ภาคผนวก ก.....	43
ภาคผนวก ข.....	57
ประวัติผู้วิจัย.....	71



จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 4 ไฟล์รูปร่างการต่อของเครือข่าย (NXX_YT.TXT).....	19
ตารางที่ 4.1.1 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสองเส้นขนาด 5 โหนด.....	23
ตารางที่ 4.1.2 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสองเส้นขนาด 10 โหนด.....	24
ตารางที่ 4.2.1 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสามเส้นขนาด 5 โหนด.....	29
ตารางที่ 4.2.2 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสามเส้นขนาด 10 โหนด.....	30
ตารางที่ 5.1.1 ตารางเปรียบเทียบค่าคาดหวังของ DELAY	32
ตารางที่ 5.1.2 ตารางตัวอย่างการวิเคราะห์ค่า DELAY สำหรับเครือข่ายขนาด 5 โหนด.....	35
ตารางที่ 5.1.3 ตารางสรุปการวิเคราะห์ค่า DELAY สำหรับเครือข่ายขนาด 5, 10, 15, 20 และ 25 โหนด.....	38
ตารางที่ ก.1 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสองเส้นขนาด 15 โหนด.....	43
ตารางที่ ก.2 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสองเส้นขนาด 20 โหนด.....	46
ตารางที่ ก.3 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสองเส้นขนาด 25 โหนด.....	51
ตารางที่ ข.1 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสามเส้นขนาด 15 โหนด.....	57
ตารางที่ ข.2 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสามเส้นขนาด 20 โหนด.....	60
ตารางที่ ข.3 SHORTEST PATH และ ค่าคาดหวังของ DELAY ของเครือข่ายสามเส้นขนาด 25 โหนด.....	65

สารบัญภาพ

รูปที่ 1.1.1 การติดต่อระหว่างหน้าที่หลัก 3 ส่วน	1
รูปที่ 1.1.2 สถาปัตยกรรม CLIENT/SERVER แบบแรกเริ่ม	2
รูปที่ 1.1.3 สถาปัตยกรรมแบบที่มีการเก็บส่วนของการปฏิบัติงานรวมเข้าไว้ใน CLIENT/SERVER	3
รูปที่ 1.6.1 สถาปัตยกรรมแบบสามเส้า.....	6
รูปที่ 3.2.1 โครงสร้างภายในของ SERVER ในเครือข่ายแบบสองเส้า.....	12
รูปที่ 3.2.2 โครงสร้างภายในของ APPLICATION SERVER และ DATABASE SERVER ในเครือข่ายแบบสามเส้า.....	13
รูปที่ 3.3.1 FLOWCHART ของ SIMULATION	14
รูปที่ 3.3.2 แสดงภาพการเปรียบเทียบ CD(A1,A2,A3) กับ ND(B1,B2,B3).....	16
รูปที่ 4.1.1 รูปร่างของเครือข่ายสองเส้าขนาด 5 โหนด (N05_2T.TXT)	20
รูปที่ 4.1.2 รูปร่างของเครือข่ายสองเส้าขนาด 10 โหนด (N10_2T.TXT).....	21
รูปที่ 4.2.1 รูปร่างของเครือข่ายสามเส้าขนาด 5 โหนด (N05_3T.TXT)	26
รูปที่ 4.2.2 รูปร่างของเครือข่ายสามเส้าขนาด 10 โหนด (N10_3T.TXT).....	27
รูปที่ 5.1.1 กราฟเปรียบเทียบค่าคาดหวังของ DELAY	33
รูปที่ 5.1.2 กราฟสรุปการวิเคราะห์ค่า DELAY สำหรับเครือข่ายขนาด 5, 10, 15, 20 และ 25 โหนด	39



1.1 ความเป็นมาและความสำคัญของปัญหา

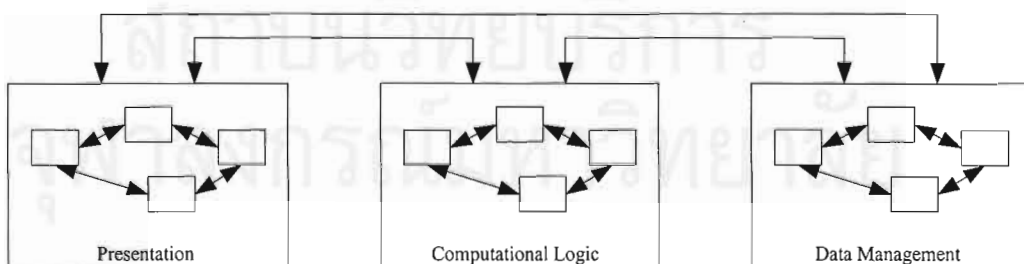
การบริการฐานข้อมูลในปัจจุบันใช้สถาปัตยกรรมแบบสองเส้ (Two-tier) หรือที่เรียกกันโดยทั่วไปว่า Client/Server แบ่งการทำงานออกเป็น 2 ชั้น คือ

- ชั้นที่ 1 Client เป็นส่วนที่ผู้ใช้ ใช้ประมวลผลในการติดต่อหรือส่งคำร้องขอผ่านไปยังฐานข้อมูลเช่น การสอบถามข้อมูล การเพิ่ม การลด และการแก้ไขเปลี่ยนแปลงฐานข้อมูล เป็นต้น
- ชั้นที่ 2 Server เป็นส่วนของการจัดการฐานข้อมูล การจัดเก็บฐานข้อมูล งานประยุกต์ และการประมวลผลข้อมูลแบบกระจาย

การทำงานของสถาปัตยกรรมแบบ Client/Server จะแบ่งหน้าที่การทำงานตามการจัดการข้อมูล ซึ่งสามารถแบ่งเป็น 3 ส่วนหลักดังนี้

- ส่วนของการแสดง (Presentation) คือส่วนที่ใช้ในการติดต่อกับผู้ใช้
- ส่วนของตรรกวิทยา (Computational logic) คือส่วนที่ใช้ในการคิดคำนวณสำหรับการทำงานตามคำร้องขอ
- ส่วนของการจัดการข้อมูล (Data management) คือส่วนที่ใช้ในการเก็บและการจัดการเกี่ยวกับฐานข้อมูล

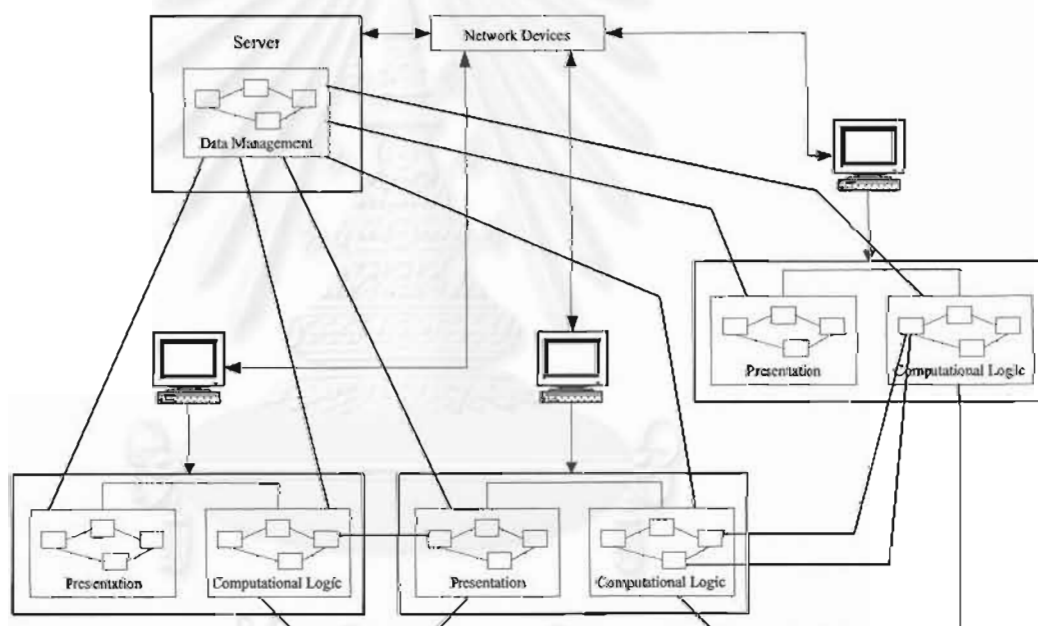
รูปที่ 1.1.1 การติดต่อระหว่างหน้าที่หลัก 3 ส่วน



รูปที่ 1.1.1 แสดงถึงความสัมพันธ์ระหว่างหน้าที่หลักทั้ง 3 ส่วน และการติดต่อภายในแต่ละส่วนจะประกอบด้วย ส่วนของหน่วยการทำงานย่อย (Module) อีกมากมาย โดยเส้นเชื่อมในรูปแสดงถึง การแลกเปลี่ยนข้อมูล ในที่นี้จะหมายถึงข้อมูล 2 ประเภทคือ เนื้อข้อมูล (Data information) และข้อมูลประเภทควบคุม (Control information)

เมื่อเราพิจารณาถึงสถาปัตยกรรม Client/Server แบบแรกเริ่ม (First-generation Client/Server) ได้มีการแยกหน้าที่ในส่วนของการแสดง (Presentation) และตรรกวิทยา (Computational logic) ไปไว้บนเครื่อง PC ซึ่งเป็นระบบงานในส่วนของ Client และนำส่วนของการจัดการข้อมูล (Data management) ไปไว้บนเครื่อง Server ดังรูปที่ 1.1.2

รูปที่ 1.1.2 สถาปัตยกรรม Client/Server แบบแรกเริ่ม
(First-generation Client/Server Architecture)



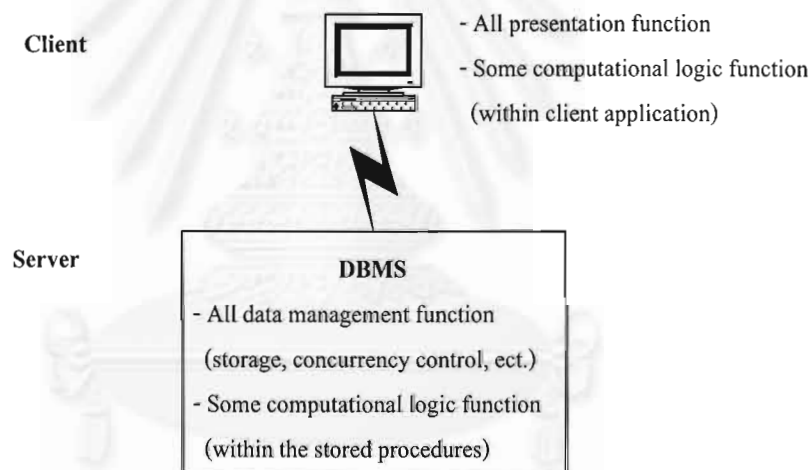
จากรูปที่ 1.1.2 จะเห็นได้ว่าการติดต่อสื่อสารระหว่างเครื่องภายในระบบมีความซับซ้อน และก่อให้เกิดปริมาณข้อมูลที่ส่งผ่านเครือข่ายจำนวนมาก ยิ่งกว่านั้นการติดต่อสื่อสารระหว่างเครื่องภายในระบบ แสดงให้เห็นว่าหน่วยการทำงานย่อย (Module) ที่ต้องมีการติดต่อกับระบบอื่น จำเป็นต้องทราบว่าจะไปที่ไหน และไปอย่างไร เพื่อให้ได้ผลลัพธ์ตามต้องการ เช่น ต้องทราบที่อยู่ทางกายภาพของระบบอื่น (Physical address) และทราบว่าหน่วยงานย่อยอะไรที่ต้องอ้างถึงเป็นต้น ซึ่งส่วนต่างๆ จำเป็นต้องเขียน โปรแกรมที่รัดกุมเพื่อให้ครอบคลุมการทำงานทั้งหมด

จากเหตุผลที่กล่าวมา ถ้าระบบเดิมมีการทำงานภายใต้สภาพแวดล้อมเดียว (Homogeneous environment) เช่น ใช้ UNIX ทั้งหมด หรือใช้ Windows Client ทั้งหมดกับ NetWare Server เป็นต้น เมื่อมีการพัฒนาระบบใหม่ ไม่ว่าจะเป็นการพัฒนาสภาพแวดล้อมให้ระบบมีความทนทาน (Robust) หรือพัฒนาเพื่อให้ใช้ได้กับหลาย platform

(Multiplatform) หรือหลาย application (Multiapplication) ซึ่งหลีกเลี่ยงไม่ได้ที่จะต้องเกิดความล่าช้า เกิดความผิดพลาดจำนวนมากในระยะเริ่มต้น และเกิดค่าใช้จ่ายจำนวนมาก

จากตัวอย่างของระบบที่กล่าวมาข้างต้น ส่วนของตรรกวิทยา (Computational logic) จะถูกรวมเข้าไว้ในเครื่องคอมพิวเตอร์ที่เป็น Client ซึ่งจะทำให้ระบบมีลักษณะของสถาปัตยกรรมแบบ Fat Client การทำงานที่มีการร้องขอข้อมูลจากฐานข้อมูลที่หลากหลาย จะทำให้ภาระการทำงานมาอยู่ที่ desktop อีกทั้งยังทำให้เกิดปัญหาในการบำรุงรักษาหากมีการเปลี่ยนแปลงแก้ไข โปรแกรมซึ่งอยู่ในส่วนของตรรกวิทยา ดังนั้นเพื่อเป็นการแก้ไขข้อบกพร่องของสถาปัตยกรรม Client/Server แบบแรกเริ่ม (First-generation Client/Server) จึงได้มีการเสนอให้มีการเก็บบางส่วนของตรรกวิทยาไว้ในการควบคุมของ Server ดังรูปที่ 1.1.3

รูปที่ 1.1.3 สถาปัตยกรรมแบบที่มีการเก็บส่วนของการปฏิบัติงานรวมเข้าไว้ใน Client/Server
(Stored Procedures Incorporated into the Client/Server Architecture)



การจัดให้ Server มีความสามารถในส่วนของตรรกวิทยา (Computational logic) โดยผ่านทาง การจัดการของ DBMS ทำให้ Server สามารถทำหน้าที่มากกว่าการจัดการข้อมูล เช่น สามารถร้องขอไปยังส่วนของการปฏิบัติงานอื่น หรือทำการตัดสินใจการปฏิบัติงานของ Server ว่าทำสำเร็จหรือไม่สำเร็จ โดยทั้งหมดนี้จะไม่ต้องการความต่อเนื่องข้ามเครือข่ายจาก Client ภายในระบบ แต่ก็ยังเกิดปัญหาที่ว่าหากต้องการทำงานข้ามเครื่องซึ่งต่างระบบกัน ทำให้มีความต้องการหน่วยการทำงานระหว่างฐานของเครือข่าย (Network-based) เพื่อทำการควบคุมแลกเปลี่ยนข้อมูล นอกจากนี้แล้วยังต้องการ DBMS เครื่องมือที่ใช้ในการพัฒนา และโปรโตคอลของเครือข่าย (IPX/SPX, TCP/IP และอื่นๆ) ที่มีความหลากหลาย ยิ่งกว่านั้นหากสถาปัตยกรรมแบบ Client/Server หรือแบบสองเส้านี้ อยู่ภายใต้สิ่งแวดล้อมที่เป็นแบบกระจายที่มีขนาดใหญ่ และข้อมูลมีความโยงใยกันแบบซับซ้อนแล้ว การแก้ปัญหาดังกล่าวนั้นต้องการค่าใช้จ่ายสูงในการแก้ไขเปลี่ยนแปลงโปรแกรม

กล่าวโดยสรุปคือ การที่จำนวนของข้อมูลบน Server และ จำนวนของ Client ขยายตัวขึ้นเรื่อยๆ จะส่งผลกระทบต่อสมรรถนะในการใช้งาน นั่นคือ ทำให้การทำงานช้าลง และการจัดการฐานข้อมูลมีความซับซ้อนมากขึ้น เพราะ ภาระงานส่วนใหญ่ตกอยู่กับ server อย่างเดียว

จากปัญหาในการขยายตัวของ Client/Server ที่กล่าวมาข้างต้น จึงทำให้เกิดแนวความคิดใหม่ที่จะช่วยแก้ปัญหาดังกล่าว โดยการนำเสนอสถาปัตยกรรมที่เรียกว่า สามเสา(Three-tier) ซึ่งจะลดหน้าที่ของ Server ให้น้อยลง และเพิ่มชั้นของการบริการขึ้นมาระหว่างชั้นของ Client และ Database Server ชั้นของการบริการนี้จะเรียกว่า Application Server ทำหน้าที่ในการจัดการเพื่อให้ Client และ Database Server สามารถติดต่อถึงกันได้สะดวก เหมาะสม และรวดเร็วขึ้น การทำงานของระบบสามเสา แบ่งเป็น 3 ชั้น คือ

- **ชั้นที่ 1 Client** จะทำหน้าที่พื้นฐานเหมือนเดิม แต่จะมีความแตกต่างกันที่วิธีการติดต่อสัมพันธ์กับเครือข่ายที่มีอยู่ ลักษณะการติดต่อมีได้ 2 ทางคือ สามารถติดต่อผ่าน Application Server หรือ สามารถติดต่อกับ Database Server โดยตรง
- **ชั้นที่ 2 Application Server** ประกอบด้วยชุดคำสั่งและตารางเก็บค่าต่างๆ ที่จำเป็นต่อการใช้งานเช่น การประมวลผลคำร้องขอจาก Client การส่งผ่านเครือข่าย การแปลงข้อมูล และ การจัดรูปแบบของฐานข้อมูลที่ได้รับมา
- **ชั้นที่ 3 Database Server** เป็นที่เก็บฐานข้อมูล ซึ่งฐานข้อมูลแบบกระจายนี้อาจประกอบด้วย Server เดียวหรือ กลุ่มของ Server ส่งผลให้มีการติดต่อโดยผ่าน DBMS ทั้งที่เหมือน (homogeneous) และ ไม่เหมือน (non-homogeneous) กันได้

1.2 วัตถุประสงค์ของการวิจัย

1. เพิ่มความสามารถในการประมวลผลแบบกระจาย ที่มีประสิทธิภาพและประสิทธิผลเหนือสถาปัตยกรรม Client/Server
2. นำเสนอแนวทางของการประมวลผลแบบกระจายในรูปแบบใหม่
3. ปรับปรุงสมรรถนะ และเพิ่มขีดความสามารถ ทั้งในด้าน ปริมาณ คุณภาพ และความน่าเชื่อถือของการประมวลผลแบบกระจาย

1.3 ขอบเขตของโครงการ

โครงการการวิจัยนี้จะแบ่งงานออกเป็น 3 ขั้นตอน ดังนี้

1. เปรียบเทียบการทำงานของระบบสามเสา กับระบบ Client/Server โดยเน้นการติดต่อประสานงานทั้งสองแบบ ด้วยวิธี Additive Approximate Distributed Bellman-Ford Algorithm (AADBF) ในการกระจายผลลัพธ์
2. วิเคราะห์รายละเอียดของการทำงาน ในแต่ละ โหนดตามวิธีการทำงานของเครือข่ายในชั้นที่ 1 โดยใช้ Fuzzy Set Delay Representation For Computer Network Routing Algorithm ในการกระจายผลลัพธ์

3. ทดลองโดยการเลียนแบบ (Simulation) วิธีการทำงานทั้งสอง โดยประเมินผลการทดลองเพื่อ
- วิเคราะห์หลักการพื้นฐานทางทฤษฎี การประมวลผลแบบกระจายของระบบสามเส้า กับระบบ Client/Server
 - วิเคราะห์เวลาในการให้บริการ และเวลาของการรอคอยในการให้บริการของระบบสามเส้า กับระบบ Client/Server

1.4 สมมติฐานการวิจัย

สถาปัตยกรรมแบบสามเส้า (Three-tier) เป็นลักษณะของการให้บริการการทำงานแบบกระจายภายใต้ระบบการประมวลผลใดๆ ก็ตาม ซึ่งทำให้ Client และ Server สามารถติดต่อสื่อสารและทำงานร่วมกัน โดยมีความสะดวก ยืดหยุ่น และถูกต้องมากที่สุด

เนื่องจากทั้ง Client และ Server ไม่เหมาะกับการประมวลผลแบบกระจาย ดังนั้นจึงเป็นหน้าที่ของ Application Server ซึ่งมีความเหมาะสมสำหรับหน้าที่ดังนี้

- ทำหน้าที่กระจาย Transaction
- ทำหน้าที่ให้บริการเพื่อค้นหาข้อมูลตามคำร้องขอ
- ทำหน้าที่ให้บริการ Client โดยมีหน้าที่เป็นตัวกลางระหว่าง Client และ Server ทำให้ Client ไม่สามารถมองเห็น DBMS และสามารถทำงานกับ DBMS ที่มากกว่าหนึ่ง โดยไม่ต้องเขียนโปรแกรมมาก
- ทำหน้าที่ให้บริการในการส่ง Message ในกรณีที่มีบาง Site ไม่สามารถใช้ได้อย่างกระตือรือร้น ทำให้ Client และ Server สามารถติดต่อสื่อสารซึ่งกันและกัน ด้วยวิธีที่เชื่อถือได้
- ทำหน้าที่ให้บริการในการประมวลผลระหว่างระบบ นั่นคือสามารถประมวลผลโดยอ้างถึงหน่วยประมวลผลข้ามระบบได้
- ทำหน้าที่จัดการระบบการไหลของข้อมูล โดยอาศัยหลักการของการหาระยะทางที่สั้นที่สุดในสายงาน ซึ่งข้อมูลทั้งสองประเภทคือ เนื้อข้อมูล (Data information) และข้อมูลประเภทควบคุม (Control information) จะถูกส่งไปตามเส้นทางที่กำหนดอย่างอัตโนมัติ

จากหน้าที่ของ Application Server ที่กล่าวมาข้างต้น จึงได้กำหนดจุดมุ่งหมายของโครงการวิจัย โดยจะเน้นการวิจัยทางด้าน Routing Algorithms ทั้งนี้เนื่องจากระบบสามเส้า ได้มีการเพิ่มขึ้นของ Application Server ขึ้นมา ทำให้เครือข่ายมีความซับซ้อนมากขึ้น ดังนั้นจึงจำเป็นต้องมีการจัดการทางด้านการจัดส่งข้อมูลที่ดี จึงจะทำให้การเพิ่มสมรรถนะของการประมวลผลแบบกระจายแบบสามเส้าให้มีประสิทธิภาพมากขึ้น ได้

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

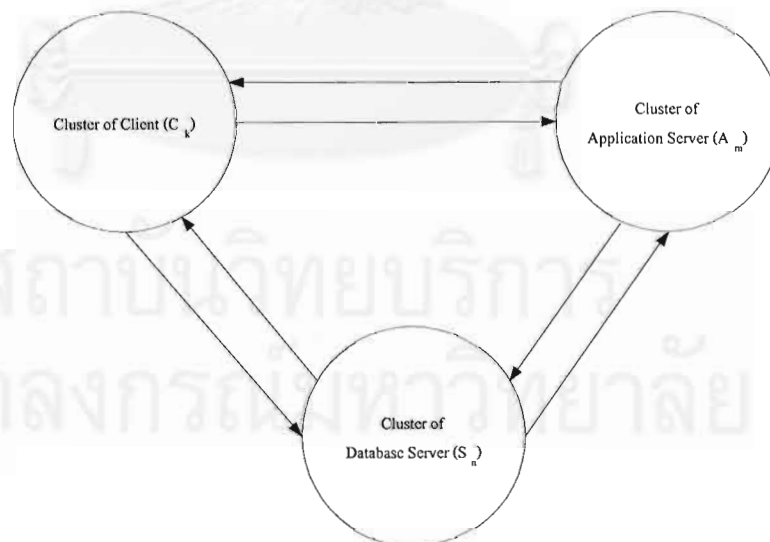
1. แนวทางการประมวลผลกระจายในรูปแบบสามเส้า
2. หลักการและระเบียบวิธีการเพิ่มขีดความสามารถของสถาปัตยกรรมแบบสามเส้าให้มากขึ้น มีความยืดหยุ่นของสถาปัตยกรรมระบบเพื่อรองรับการขยายตัวของจำนวนข้อมูลและจำนวนของ Client

3. หลักการและระเบียบวิธีการเพิ่มสมรรถนะในการประมวลผลแบบกระจายให้มีประสิทธิภาพมากขึ้น ทั้งในด้านของความเร็ว ความถูกต้อง ความน่าเชื่อถือของการประมวลผล และความสามารถของการทำงานในปริมาณที่มากขึ้น

1.6 ลำดับขั้นตอนและวิธีการดำเนินการวิจัย

- ศึกษาหลักการและ ทฤษฎีเกี่ยวกับ
 - Network Topology, Routing Algorithms และ Queuing Theory
 - สถาปัตยกรรมแบบ Client/Server
 - การจัดการ การประมวลผลแบบกระจาย
 - Fuzzy Logic
 - Probability Distribution
- กำหนดสมมุติฐาน, ตัวแปรที่ใช้ในการทดลอง, ตัวแปรควบคุม และเลือกกลุ่มตัวอย่างในการทดลองของสถาปัตยกรรมแบบสามเส้า ดังรูปที่ 1.6.1

รูปที่ 1.6.1 สถาปัตยกรรมแบบสามเส้า
(Three-Tier Architecture)



- กำหนดให้กราฟ $G = (V, E)$ แทนกราฟที่ไม่มี cycles ของเครือข่ายในสถาปัตยกรรมแบบสามเส้า โดย V คือ กลุ่มของจุดในกราฟ $(C_n, A_m, S_n \in V)$ และ E คือ กลุ่มของเส้นเชื่อมระหว่างจุดในกราฟ

- กำหนดให้ C_k แทนกลุ่มของ Client ในกราฟ โดยมีทั้งหมด k ตัว
 - กำหนดให้ A_m แทนกลุ่มของ Application Server ในกราฟ โดยมีทั้งหมด m ตัว
 - กำหนดให้ S_n แทนกลุ่มของ Database Server ในกราฟ โดยมีทั้งหมด n ตัว
 - กำหนดให้กราฟมีจำนวนจุดทั้งหมด N จุด ($N = k + m + n$)
 - กำหนดให้จุด s แทนจุดเริ่มต้น (Source) ในการหาเส้นทางที่สั้นที่สุดจากจุด s ไปยังจุดใดๆ ในกราฟ
 - กำหนดให้ d_{ij} แทนระยะทางระหว่างจุด i และจุด j โดย i, j คือจุดใดๆ ในกราฟ
 - $d_{ij} \geq 0$ ถ้ามีเส้นเชื่อมระหว่างจุด i และจุด j
 - $d_{ij} = -1$ ถ้าไม่มีเส้นเชื่อมระหว่างจุด i และจุด j
 - กำหนดให้ β แทนตัวแปรที่มีค่ามากกว่าหรือเท่ากับ 0 ในที่นี้จะมามีค่าเท่ากับค่าของระยะทางสั้นที่สุดในกราฟ (minimum length)
 - กำหนดให้ h แทนจำนวนสูงสุดของเส้นเชื่อม ขณะทำการหาเส้นทางที่สั้นที่สุดในแต่ละรอบของการทำซ้ำ โดยที่ $h \leq N - 1$
 - กำหนดให้ D_{hi} แทนค่าของระยะทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุด i ภายใต้งเงื่อนไขคือ ต้องมีเส้นเชื่อมไม่มากกว่าค่า h
 - ค่าของระยะทางที่สั้นที่สุดของจุดเริ่มต้น s สำหรับทุกรอบของการทำซ้ำ ($\forall h$) มีค่าเท่ากับ 0

$$D_{hs} = 0, \forall h$$
 - ค่าของระยะทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุด i สำหรับรอบที่ 0 ของการทำซ้ำ ($h=0$) มีค่าเท่ากับ -1

$$D_{0i} = -1, \forall i \neq s$$
3. สร้างแบบจำลองเชิงคณิตศาสตร์ ให้สอดคล้องกับสมมุติฐานที่ตั้งไว้ และทดสอบแบบจำลอง
 4. เก็บข้อมูลของผลการทดลองที่ได้จากแต่ละตัวแบบ และวิเคราะห์ผลลัพธ์ที่ได้โดยการจำลองสถานการณ์ (Simulation)
 5. เปรียบเทียบผลลัพธ์ที่ได้จากแต่ละตัวแบบ สำหรับการทำงานของระบบสามเส้ากับ Client/Server และสรุปผลการทดลองที่ได้ทั้งหมด ในรูปแบบของตาราง และ กราฟ
 6. เขียนวิทยานิพนธ์

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

เนื่องจากจุดมุ่งหมายของโครงการวิจัยเน้นการวิจัยทางด้าน Routing Algorithms ดังนั้นการทดลองโดยการจำลองสถานการณ์ (Simulation) จึงได้ออกแบบ Procedure โดยยึดถือวิธีการคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุดจาก Approximate Distributed Bellman-Ford Algorithm (AADBF) และเพื่อให้ค่าของระยะทางที่สั้นที่สุดที่คำนวณได้มีความแน่นอนมากยิ่งขึ้นจึงได้นำวิธีการของ A Fuzzy Set Delay Representation for Computer Network Routing (FSDR) มาประยุกต์ใช้กับ AADBF ซึ่งทั้งสอง Algorithms มีหลักการพื้นฐานมาจากวิธีการของ Distributed Bellman-Ford Algorithm (DBF) โดยรายละเอียดของแต่ละ Algorithm มีดังนี้

2.1 Distributed Bellman-Ford Algorithm (DBF)

DBF เป็นวิธีการที่ใช้ในการหาระยะทางที่สั้นที่สุดจากโหนดเริ่มต้น (Source node) ไปยังโหนดอื่นๆภายในเครือข่าย โดยกำหนดค่าเริ่มต้นของระยะทางที่สั้นที่สุดของโหนดเริ่มต้นเท่ากับ 0 และ โหนดอื่นๆ เท่ากับค่าอนันต์ และกำหนดให้ระยะทางจากโหนดใดๆ ที่ไม่มีเส้นเชื่อมถึงกันมีค่าเท่ากับอนันต์ การคำนวณจะเริ่มจากการหาระยะทางที่สั้นที่สุดของเส้นเชื่อม 1 เส้นที่ออกจากจุดเริ่มต้น หลังจากนั้นก็จะทำการคำนวณค่าใหม่ โดยเพิ่มจำนวนของเส้นเชื่อมขึ้นทีละ 1 ซึ่งการคำนวณนี้จะวนเพิ่มค่าเช่นนี้เรื่อยๆ จนกระทั่ง ค่าของระยะทางที่สั้นที่สุดจากโหนดเริ่มต้นไปยังแต่ละ โหนดมีค่าเท่ากับค่าที่คำนวณได้ในรอบก่อนหน้าทุกค่า หรือจำนวนรอบในการวนเท่ากับ $n-1$ รอบ ในที่นี้กำหนดให้ n คือจำนวน โหนดภายในเครือข่าย การทำงานของ DBF จะมีกฎที่ใช้อยู่ 2 ลักษณะดังนี้

1. กฎของการปรับค่า (Adapting rule) หมายถึง ทำอย่างไรเมื่อ โหนดนั้นๆ ได้รับข้อความที่ส่งมาจากโหนดข้างเคียง เพื่อใช้ในการปรับค่าระยะทางที่สั้นที่สุด การปรับค่านี้อาจกำหนดให้มีการปรับเป็นช่วงเวลา
2. กฎของการส่งค่า (Sending rule) หมายถึง ต้องทำอะไรเมื่อ โหนดนั้นๆ มีการเปลี่ยนแปลงค่า และจะต้องกระจายค่าอะไรไปยังโหนดข้างเคียง

ข้อดีของ DBF คือ การคำนวณจะใช้ข้อมูลจำนวนน้อยในแต่ละโหนด นั่นคือใช้เฉพาะข้อมูลของโหนดข้างเคียงและ ชื่อของแต่ละโหนดในเครือข่ายเท่านั้น โดยไม่จำเป็นต้องเก็บรายละเอียดลักษณะรวมของเครือข่าย (Network topology) การคำนวณในแต่ละขั้นตอน แต่ละโหนดสามารถทำเป็นแบบขนานได้ (Parallel computing) โดยอาศัยข้อมูลที่ส่งมาจากโหนดข้างเคียงซึ่งทำงานแบบไม่พร้อมกัน (Asynchronous) และการทำเช่นนี้จะทำให้การทำงานมีความถูกต้องโดยไม่จำเป็นต้องเริ่มต้นคำนวณใหม่ทั้งหมดหากสถานะหรือความยาวของเส้นเชื่อมระหว่างโหนด (Link status or Link length) เกิดการเปลี่ยนแปลงในภายหลัง

2.2 Approximate Distributed Bellman-Ford Algorithm (AADBF)

เนื่องจากการทำงานโดยใช้วิธีการของ DBF กับบางระบบที่มีจำนวนโหนดมาก ($n \gg 1$) และมีจำนวนของทางเดินเป็น 2^n สามารถทำให้เกิดความซับซ้อนของข้อมูล (Message complexity) ที่เป็น Exponential นั่นคือมีค่าเท่ากับ $O(2^n)$ ดังนั้นจึงมีการพัฒนาวิธีการทำงานแบบ AADBF ขึ้นมาเพื่อช่วยแก้ปัญหาดังที่กล่าวมาแล้ว โดย AADBF จะเพิ่มตัวแปร β ซึ่งมีค่ามากกว่าหรือเท่ากับ 0 เพื่อใช้ในการหาระยะทางที่สั้นที่สุดที่มีค่ามากที่สุดเท่ากับ $\beta_n + D_i$ และให้ค่าความซับซ้อนของข้อมูลเท่ากับ $nm\Delta / \beta$

ความแตกต่างระหว่าง DBF และ AADBF นั้นจะอยู่ที่กฎของการปรับค่า (Adapting rule) โดย DBF จะทำการปรับค่าก็ต่อเมื่อ ค่าของระยะทางที่สั้นที่สุดที่หาได้ใหม่ มีค่าน้อยกว่าค่าเก่า แต่สำหรับ AADBF นั้นจะทำการปรับค่าก็ต่อเมื่อ ค่าของระยะทางที่สั้นที่สุดที่หาได้ใหม่ มีค่าน้อยกว่าค่าเก่าลบด้วยค่า β

จะเห็นได้ว่าการทำงานของ AADBF ไม่ได้มุ่งที่จะพัฒนาเพื่อลดความซับซ้อนของเวลาในการหาระยะทางที่สั้นที่สุด แต่จะมุ่งเน้นเพื่อลดความซับซ้อนของข้อมูลที่ส่งผ่านในเครือข่ายให้เป็น Polynomial โดยใช้วิธีการคำนวณแบบเดิม ซึ่งระยะทางที่สั้นที่สุดที่ได้จะมีค่าค่อนข้างเหมาะสม (Almost optimal)

2.3 A Fuzzy Set Delay Representation for Computer Network Routing (FSDR)

ประสิทธิภาพของการกระจายค่าในการคำนวณหาทางเดินของข้อมูลในเครือข่าย ขึ้นอยู่กับความสามารถในการปรับค่าข้อมูลในเครือข่ายเมื่อมีการเปลี่ยนแปลงค่าของระยะทางและสถานะของระบบเครือข่าย เมื่อโหนดต่างๆ ในเครือข่ายมีการแลกเปลี่ยนข้อมูลซึ่งกันและกันในช่วงระยะเวลาที่กำหนด ข้อมูลที่ได้รับในขณะนั้นจะเป็นข้อมูลที่เก่าไม่ทันสมัย เนื่องจากระบบมีการเปลี่ยนแปลงอยู่ตลอดเวลา ทำให้ค่าของระยะทางที่สั้นที่สุดที่คำนวณได้อาจจะเกิดความไม่แน่นอน ดังนั้นจึงเกิดแนวความคิดที่จะเก็บข้อมูลที่เปลี่ยนแปลงในช่วงระยะเวลาที่กำหนด เพื่อนำมาหาค่าประมาณของระยะทางที่เกิดขึ้น โดยอาศัยหลักการของ Fuzzy Logic

วิธีการ FSDR จะจัดเก็บข้อมูลที่เกิดขึ้นระหว่างช่วงของการปรับค่าข้อมูล โดยจะเก็บค่าสูงสุด และค่าต่ำสุดของระยะทางที่เปลี่ยนแปลง และนำค่าทั้งหมดมาหาค่าเฉลี่ยน้ำหนัก เพื่อให้ได้ระยะทางปัจจุบัน (current length) ระยะทางที่ต่ำสุด (minimum length) และระยะทางที่สูงที่สุด (maximum length) โดยทุกค่าที่หาได้จะต้องนำไปคูณกับค่า Mean packet transmission delay ด้วย หลังจากนั้นก็จะทำการส่งค่าต่างๆ ไปยังโหนดข้างเคียงเพื่อทำการปรับค่าของระยะทางที่สั้นที่สุดโดยใช้วิธีการของ DBF

ระยะทางที่สั้นที่สุดที่คำนวณได้ใหม่ซึ่งประกอบด้วย minimum delay, weighted mean delay และ maximum delay โดยกำหนดให้เป็น a_1, a_2 และ a_3 ตามลำดับ ส่วนข้อมูลก่อนการปรับค่ากำหนดให้เป็น b_1, b_2 และ b_3 ตามลำดับ ค่าเหล่านี้จะถูกนำมาเปรียบเทียบ โดยจะทำการแปลงค่าทั้งสองให้อยู่ในรูปแบบของ Fuzzy number ซึ่งจะใช้วิธีการ Ordering method ของ Ronald Yager ในการคำนวณ เมื่อได้เซตของ Fuzzy number แล้วก็จะนำมาเปรียบเทียบโดยมีข้อกำหนดคือ ถ้าค่าความแตกต่างระหว่าง a_2 กับ b_2 มีค่ามากกว่า ค่าพิเศษที่กำหนดขึ้น (Threshold) จะ

เลือกค่าที่มีค่าของระยะทางที่ต่ำกว่า แต่ถ้าค่าความแตกต่างระหว่าง a_2 กับ b_2 มีค่าน้อยกว่า ค่าพิเศษที่กำหนดขึ้น จะมีการพิจารณาดังนี้

1. ถ้า $a_1 \leq b_1$ และ $a_3 \leq b_3$ จะเลือกระยะทางที่สั้นที่สุดที่คำนวณได้ใหม่
2. ถ้า $a_1 \leq b_1$ และ $a_3 > b_3$ จะเลือกระยะทางที่สั้นที่สุดเก่า เนื่องจากค่าใหม่ที่คำนวณได้มีความไม่แน่นอนสูง
3. ถ้า $a_1 - b_1 \leq$ ค่าพิเศษที่กำหนดขึ้น และ $a_3 \leq b_3$ จะเลือกระยะทางที่สั้นที่สุดที่คำนวณได้ใหม่
4. ถ้า $a_1 - b_1 \leq$ ค่าพิเศษที่กำหนดขึ้น และ $a_3 - b_3 \leq$ ค่าพิเศษที่กำหนดขึ้น จะเลือกได้ทั้งค่าใหม่และค่าเก่า



จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

วิธีดำเนินการวิจัย

3.1 การวิเคราะห์ระบบ

การวิเคราะห์ระบบเป็นขั้นตอนแรกในการดำเนินงาน เพื่อกำหนดความต้องการของระบบใหม่ที่จะทำขึ้น โดยแบ่งการวิเคราะห์ออกเป็น 3 ส่วนดังนี้

3.1.1 Functional Requirements

เป็นการวิเคราะห์ความต้องการของระบบในแง่ของหน้าที่ ว่าระบบนี้จะต้องทำหน้าที่อะไรบ้าง ซึ่งแบ่งเป็นหัวข้อใหญ่ๆ ได้ดังนี้

- มีการอ่าน input ที่จำเป็น มาจาก text file
- delay ของแต่ละเส้นทางใน Network จะมีการเปลี่ยนแปลงตลอดเวลาด้วยวิธีการ random จำนวน packet ที่ส่งของแต่ละเส้นทาง แล้วคำนวณหาค่า delay ภายในระยะเวลาที่กำหนด
- ระบบสามารถคำนวณหาระยะเวลาในการเดินทางรวมจากต้นทางถึงปลายทางได้
- ระบบสามารถทำการทดลองการส่ง packet ได้หลายครั้ง

3.1.2 Application Requirements

เป็นการวิเคราะห์ความต้องการของระบบในแง่ของอุปกรณ์ ว่าระบบต้องการอุปกรณ์อะไรบ้างทั้งทางด้าน Software และ Hardware ซึ่งความต้องการของระบบในการทดลองนี้คือ

- ใช้ Software ภาษา C
- โปรแกรมทำงานบน Dos และ UNIX platform

3.1.3 Data Requirements

เป็นการวิเคราะห์ความต้องการของระบบในแง่ของข้อมูล ว่าระบบต้องการข้อมูลประเภทใด อะไรบ้าง ซึ่งในการทดลองนี้คือ

- ข้อมูลทดสอบสำหรับเครือข่ายทั้งแบบสองเส้า และแบบสามเส้า
- ข้อมูลทดสอบซึ่งป้อนผ่านแฟ้มข้อมูลในรูปของ Nxx_yT.TXT

โดยที่ xx หมายถึง จำนวน โหนดในเครือข่าย
y หมายถึง รูปแบบของเครือข่าย

- 2 คือแบบ สองเส้า
- 3 คือแบบ สามเส้า

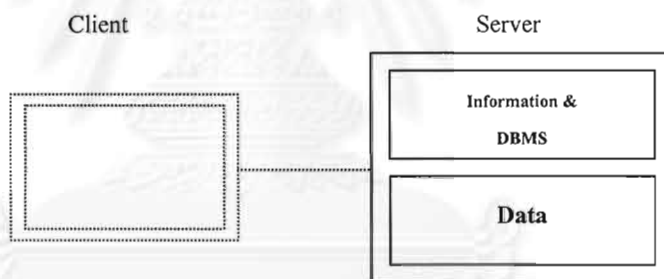
3.2 การออกแบบโครงสร้าง Network

รูปแบบของเครือข่ายที่ใช้ในการทดลอง เป็นเครือข่ายซึ่งมีเส้นเชื่อมแต่ละโหนดเป็นเส้นที่มีทิศทางไปและกลับ (Undirected Graph) โดยโครงสร้างของเครือข่ายที่ใช้ในการทดลองจะได้อาจมาจากโปรแกรม SIMNODE และเก็บรูปแบบที่ได้ในไฟล์ Nxx_yT.TXT

ในการทดลองจะกำหนดให้มีการสืบค้นที่ Primary Site เท่านั้น นั่นคือจะต้องทำการสืบค้นหาข้อมูลตามคำร้องขอของ Client เฉพาะกลุ่มที่มีการติดต่อกันโดยตรง (Direct Connection) ซึ่งมีค่าความน่าจะเป็นที่จะพบข้อมูลใน Server หรือ Application Server แต่ละตัวรวมกันได้เท่ากับ 1

การออกแบบโครงสร้างเครือข่ายแบบสองเส้าและแบบสามเส้า จะมีจำนวนโหนดเท่ากัน และโครงสร้างแต่ละแบบจะมีคุณลักษณะเฉพาะตัว โดยโครงสร้างแบบสองเส้าประกอบด้วย Client ซึ่งแต่ละตัวมีความอิสระต่อกัน (ไม่มีเส้นเชื่อมระหว่างกัน) และ Server ที่แต่ละตัวเก็บข้อมูลอยู่สองประเภทคือ Data และ Information ดังแสดงในรูปที่ 3.2.1 และเส้นเชื่อมภายในระบบสร้างโดยวิธีการสุ่มระหว่างชั้น Client กับ Server และระหว่างชั้น Server กับ Server ดังนั้นการร้องขอข้อมูลของ Client แต่ละตัวจะไม่ว่าข้อมูลที่ต้องการเก็บไว้ใน Server ตัวไหน

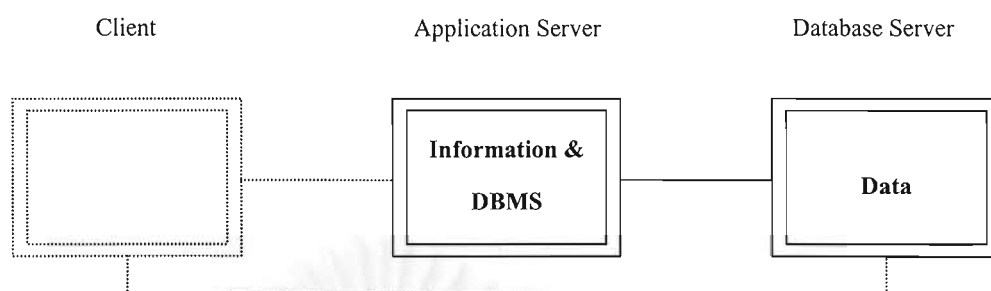
รูปที่ 3.2.1 โครงสร้างภายในของ Server ในเครือข่ายแบบสองเส้า



ส่วนโครงสร้างแบบสามเส้าประกอบด้วย Client, Application Server และ Database โดย Client แต่ละตัวมีความอิสระต่อกัน (ไม่มีเส้นเชื่อมระหว่างกัน) และภายใน Application Server แต่ละตัวจะเก็บข้อมูลประเภท Information ซึ่งจะทราบว่า Data เก็บไว้ที่ Database Server ตัวไหน ดังแสดงในรูปที่ 3.2.2 ดังนั้นการร้องขอข้อมูลของ Client แต่ละตัวจะไม่ว่าข้อมูลที่ต้องการเก็บไว้ใน Application Server หรือ Database Server ตัวไหน จะต้องทำการสืบค้นที่ชั้นของ Application Server ก่อน จึงจะรู้ที่อยู่ของข้อมูลใน Database Server และเส้นเชื่อมภายในระบบสร้างโดย

1. สร้างเส้นเชื่อมด้วยวิธีการสุ่มระหว่างชั้น Client กับ Application Server และระหว่างชั้น Client กับ Database Server
2. สร้างเส้นเชื่อมแบบ Fully-Connected ระหว่างชั้น Application Server กับ Application Server และระหว่างชั้น Application Server กับ Database Server ทั้งนี้เพื่อให้สามารถเลือกเส้นทางในการส่งผ่านข้อมูลได้หลายทาง ไม่ว่าจะในทางตรงหรือทางอ้อม
3. สร้างเส้นเชื่อมแบบ Bus ระหว่างชั้น Database Server กับ Database Server เนื่องจากข้อมูลมีลักษณะการเก็บแบบกระจาย ดังนั้นการเชื่อมต่อแบบ Bus จะช่วยให้การเข้าถึงข้อมูลสะดวกขึ้น

รูปที่ 3.2.2 โครงสร้างภายในของ Application Server และ Database Server ในเครือข่ายแบบสามเส้า

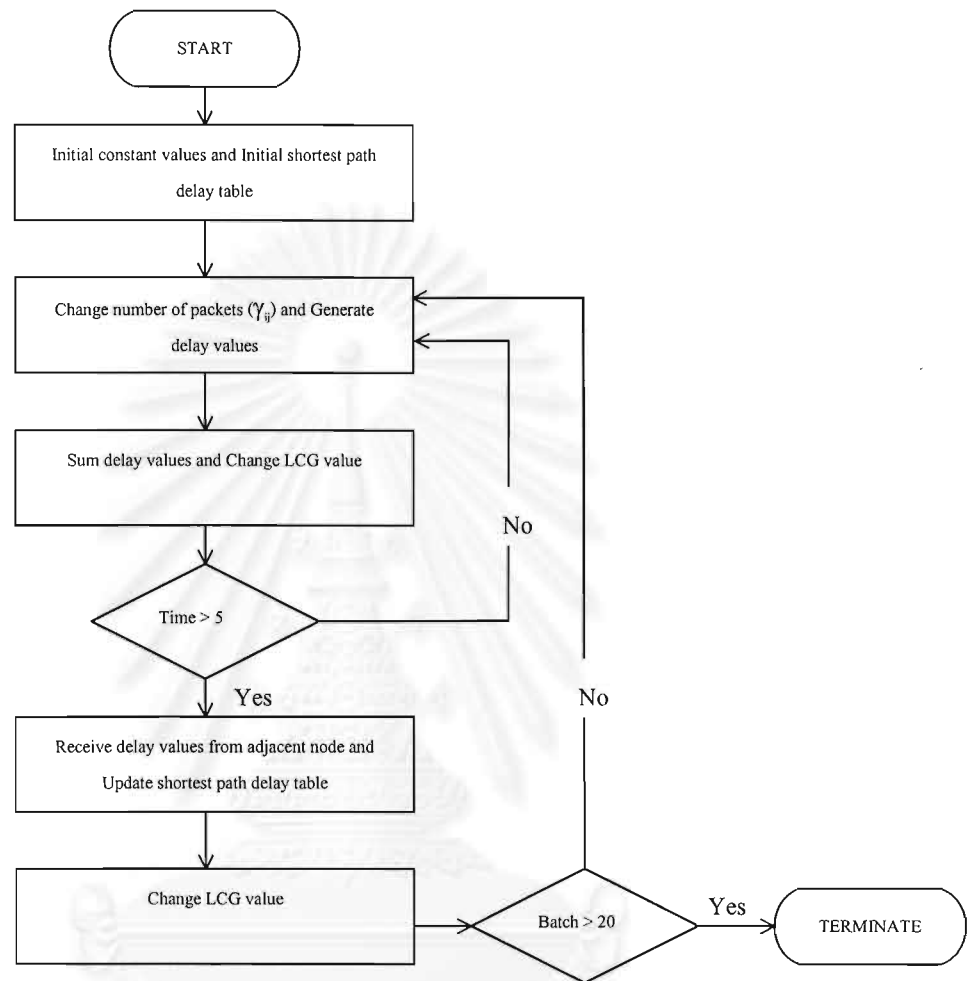


3.3 การออกแบบ Procedure

ในการพัฒนาระบบเพื่อจำลองการทำงานของเครือข่ายแบบสองเส้าและเครือข่ายแบบสามเส้า จะทำการสุ่มจำนวนของ packet ที่จะส่งออกสู่ระบบเครือข่าย แล้วคำนวณหาค่า delay ที่เกิดขึ้นของแต่ละเส้นเชื่อมภายในเครือข่าย โดยจะทำการสุ่มและคำนวณ 5 ครั้ง ต่อการคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุด (Shortest path) จากจุดเริ่มต้นไปยังจุดปลายทางที่ต้องการ 1 ครั้งของการสุ่มทดลอง ทั้งนี้เนื่องจากการทำงานของระบบงานจริงค่าของ delay ในแต่ละเส้นทางจะมีการเปลี่ยนแปลงอยู่ตลอดเวลา การคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุดจะต้องกำหนดช่วงระยะเวลาในการคำนวณ (period time) ดังนั้นการทดลองโดยการจำลองสถานการณ์ (Simulation) จึงกำหนดให้การสุ่มและคำนวณเพื่อหาค่า delay จากการทดลอง 5 ครั้งภายในช่วง period ที่กำหนด การคำนวณก็ทำโดยเก็บค่า delay ที่ได้ทั้งหมดไปหาค่าเฉลี่ย ตามวิธีของ [5] จากนั้นจึงนำค่าเฉลี่ยที่คำนวณได้มาหาเส้นทางที่มีระยะทางที่สั้นที่สุด โดยยึดถือวิธีการคำนวณจาก Algorithm [2], [3] และ [6] เพื่อให้ผลลัพธ์ที่ได้มีความถูกต้องใกล้เคียงกับความจริงมากยิ่งขึ้น เนื่องจากความไม่แน่นอนของชุดข้อมูลที่ได้จากการจำลองสถานการณ์ จึงกำหนดให้มีการทำคำนวณหาเส้นทางที่มีระยะทางที่สั้นที่สุดทั้งหมด 20 ชุด (batch) การทำงานของการจำลองสถานการณ์สามารถแสดงในรูปของ Flowchart ได้ดังรูปที่ 3.3.1

จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.3.1 Flowchart ของ Simulation



จาก Flowchart ในรูปที่ 3.3.1 สามารถอธิบายโดยเขียนเป็น Algorithm ได้ดังนี้

Simulation Algorithm

MAIN

- 1 Initialize();
- 2 Initialtab();
- 3 Send_packet();

Repeat step 3 until iteration equal constant Batch

Initialize()

- 1 Initial constant values
- 2 InputNetwork();

Initialtab()

- 1 set $D_s = 0$ and $D_i = \infty$, for all $i \neq s$
- 2 Generate_random_number();
- 3 find $D_i = \min_{j \in N(i)} [D_j + d_{ij}]$, for all $i \neq s$
Repeat step 3 until none of the delays (costs) changes

Sent_packet()

- 1 Change_numpk();
- 2 Generate_delay();
- 3 Sum_delay();
- 4 Change_LCG1();
Repeat step 1 until iteration equal constant Time
- 5 Average();
- 6 Receive();
- 7 Updatetab();
- 8 Change_LCG2();
- 9 Print Final Report

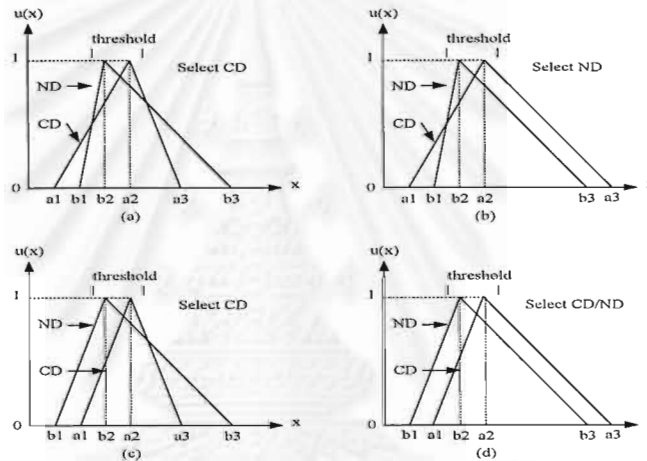
Updatetab()

- 1 set bias to parent(i)
- 2 compare the current delay values (CD(a1, a2, a3)) with the new delay values (ND(b1, b2, b3)) by using fuzzy ordering method
 - if CD is selected, set a2 to be Current Delay(d_{ij})
 - if ND is selected, set b2 to be Current Delay(d_{ij})
- 3 set $D_i = D'_i$ and find $D'_i = \min_{j \in N(i)} [D_j + d_{ij}]$, for all $i \neq s$
- 4 if D'_i is less than $D_i - \beta$, set $D_i = D'_i$
Repeat step 3 for all $i \neq s$

เกณฑ์ในการเปรียบเทียบ $CD(a_1, a_2, a_3)$ กับ $ND(b_1, b_2, b_3)$ ดังรูปที่ 3.3.2

- ถ้า $a_1 \leq b_1$ และ $a_3 \leq b_3$ จะเลือกกระยะทางที่สั้นที่สุดที่คำนวณได้ในครั้งหลัง
- ถ้า $a_1 \leq b_1$ และ $a_3 > b_3$ จะเลือกกระยะทางที่สั้นที่สุดเก่า เนื่องจากค่าใหม่ที่คำนวณได้มีความไม่แน่นอนสูง
- ถ้า $a_1 - b_1 \leq$ ค่าพิเศษที่กำหนดขึ้น และ $a_3 \leq b_3$ จะเลือกกระยะทางที่สั้นที่สุดที่คำนวณได้ในครั้งหลัง
- ถ้า $a_1 - b_1 \leq$ ค่าพิเศษที่กำหนดขึ้น และ $a_3 - b_3 \leq$ ค่าพิเศษที่กำหนดขึ้น จะเลือกได้ทั้งค่าใหม่และค่าเก่า

รูปที่ 3.3.2 แสดงภาพการเปรียบเทียบ $CD(a_1, a_2, a_3)$ กับ $ND(b_1, b_2, b_3)$
(Comparison Criteria of Fuzzy Delay)



รายละเอียดในส่วนต่างๆ ของ Algorithm ที่พัฒนาขึ้นมีดังนี้

MAIN

เป็น module หลักของโปรแกรม จะทำหน้าที่แบ่งงานย่อยๆ ให้แต่ละ module ย่อยภายในโปรแกรมอันได้แก่การกำหนดค่าเริ่มต้น การสร้างตารางเส้นทางเริ่มต้น (Initial Routing Table) และการส่ง packet ออกสู่ระบบเครือข่าย

Initialize()

การกำหนดค่าเริ่มต้น สามารถแบ่งออกได้เป็น 2 function ย่อยคือ

- *Initial constant values* จะทำการกำหนดค่าเริ่มต้นของตัวแปรต่างๆ ที่จำเป็น
- *InputNetwork* ทำหน้าที่รับค่าต่างๆ มาใส่ลงใน data structure ที่เตรียมไว้ เพื่อให้โปรแกรมรู้ถึงรูปร่างการต่อของ Network

Initialtab()

การสร้างตารางเส้นทางเริ่มต้น สามารถแบ่งออกได้เป็น 2 function ย่อยคือ

- กำหนดค่าเริ่มต้นของข้อมูล
 - ค่าของระยะทางที่สั้นที่สุดของจุดเริ่มต้น s สำหรับทุกรอบของการทำซ้ำ ($\forall h$) มีค่าเท่ากับ 0

$$D_{hs} = 0, \forall h$$
 - ค่าของระยะทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุด i สำหรับรอบที่ 0 ของการทำซ้ำ ($h=0$) มีค่าเท่ากับ -1

$$D_{0i} = -1, \forall i \neq s$$
 - การสร้างเลขสุ่มของ delay
- การเลือกเส้นทางที่มีระยะทางที่สั้นที่สุด โดยใช้วิธีการของ Distributed Bellman-Ford Algorithm

$$D_i = \min_{j \in N(G)} [D_j + d_{ij}], \forall i \neq s$$

Sent_packet()

การส่ง packet ออกวิ่ง สามารถแบ่งออกได้เป็น 4 function ย่อยคือ

- การสร้างเลขสุ่มของจำนวน packet ที่จะส่งสำหรับแต่ละโหนด (γ_j)
- การเปลี่ยนองค์ประกอบ LCG
- การคำนวณหาค่า delay
- การเลือกเส้นทางใหม่ที่มีระยะทางที่สั้นที่สุด

Change_numpk()

การสร้างเลขสุ่ม

การสร้างเลขสุ่มของจำนวน packet ที่จะส่งสำหรับแต่ละโหนด (γ_j) จะมีหัวใจหลักคือ LCG ที่เป็นตัวสร้างเลขสุ่มขึ้นมาโดยใช้สูตร

$$RND = (RA * Z_0 + RC) \% RM$$

โดยกำหนดค่าเริ่มต้นของตัวแปรแต่ละตัวดังนี้

- $RA = 79$
- $RC = 53$
- $RM = 1024$
- $Z_0 = 30$ และจะมีการเปลี่ยนแปลงค่าทุกครั้ง

Change_LCG1() และ Change_LCG2()

การเปลี่ยนองค์ประกอบ LCG

เหตุผลที่ต้องมี module นี้คือ เราไม่ต้องการที่จะให้เกิดการซ้ำกันของตัวเลขสุ่มที่สร้างขึ้นของแต่ละ run จึงจัดให้มีการเปลี่ยนค่าคงที่ต่างๆ ของ LCG ทุกครั้งและทุก batch ที่โปรแกรมทำงาน

Generate_delay()

การคำนวณหาค่า delay

ใช้วิธีการคำนวณจาก Algorithm [1] โดยกำหนดให้ฟังก์ชันความน่าจะเป็นของความหนาแน่น (Probability density function) สำหรับขนาดของ packet เท่ากับ $\mu e^{-\mu x}$

- ค่ากลางของขนาด packet $1/\mu = 200$ bits/packet
- ค่า Capacity (C_i) = 40 bits/sec สำหรับเส้นทางที่เชื่อมระหว่าง Client กับ Database Server โดยตรงในสถาปัตยกรรมแบบสามเส้า
- ค่า Capacity (C_i) = 200 bits/sec สำหรับเส้นทางทุกเส้นในสถาปัตยกรรมแบบสองเส้า และ ทุกเส้นยกเว้นเส้นทางที่เชื่อมระหว่าง Client กับ Database Server โดยตรงในสถาปัตยกรรมแบบสามเส้า
- ค่า Arrival rate (λ_i) = $h_{ij} + \gamma_{ij}$ packets/sec
กำหนดให้ h_{ij} คือ จำนวนครั้งของการส่ง packet ผ่านเส้นเชื่อมระหว่าง โหนด i และ โหนด j
 γ_{ij} คือ จำนวนของ packet ที่ส่งผ่านเส้นเชื่อมระหว่าง โหนด i และ โหนด j
- ค่า delay สำหรับแต่ละเส้นเชื่อม เท่ากับ $1 / (\mu C_i - \lambda_i)$

Updatetab()

การเลือกเส้นทางใหม่

เนื่องจากค่าของ delay มีการเปลี่ยนแปลงอยู่ตลอดเวลา จึงได้กำหนดรอบของการ update ตามรอบของ batch ที่วิ่งในโปรแกรม และในแต่ละรอบของ batch จะประกอบด้วยรอบของการเปลี่ยนค่าของ delay 5 ครั้ง ดังนั้นการเลือกเส้นทางใหม่จึงแบ่งออกได้เป็น 4 function ย่อยคือ

- *sum_delay* เมื่อมีการเปลี่ยนค่าของ delay แต่ละครั้งจะต้องมีการบวกค่าของค่าสูงสุด ค่าต่ำสุด และ ค่าปัจจุบันของ delay เอาไว้เพื่อใช้ในการหาค่า Weight Mean ของทั้งสามค่า ในแต่ละ batch
- *Average* เมื่อมีการเปลี่ยนค่าของ delay ครบ 5 ครั้งจะนำค่าที่บวกเก็บไว้มาหาค่า Weight Mean เพื่อใช้ในการเปรียบเทียบกับ ค่า Weight Mean ของรอบ batch ก่อนหน้า โดยอาศัยหลักการของ Ordering fuzzy subsets of unit interval และ หลักการของ A Fuzzy Set Delay Representation for Computer Network Routing (FSDR)
- *Receive* เมื่อได้ค่าเฉลี่ยของ delay แล้ว จะทำการ update ตารางเส้นทางของโหนดอื่นที่ไม่ใช่โหนดเริ่มต้น ทั้งนี้เนื่องจากสมมติฐานที่ว่า โหนดข้างเคียงของโหนดเริ่มต้นจะต้องมีการ update ค่าในตารางเส้นทางก่อนโหนดเริ่มต้น ดังนั้นการทำงานใน module นี้จึงเสมือนเป็นการรับค่าที่เปลี่ยนแปลงแล้วของโหนดข้างเคียง
- *Updatetab* หลังจากรับค่าในตารางเส้นทางจากโหนดข้างเคียงแล้ว ก็จะทำการ update ค่าในตารางของโหนดเริ่มต้น โดยอาศัยหลักการของ Distributed Bellman-Ford และ Additive Approximate Distributed Bellman-Ford Algorithm

บทที่ 4

ผลการวิจัย

การวิเคราะห์ Algorithm ในงานวิจัยนี้ ใช้ข้อมูลตัวอย่าง 10 ชุด ซึ่งแต่ละชุดเก็บในรูปแบบของแฟ้มข้อมูล โดยทำการสร้างไฟล์ Nxx_yT.TXT ทั้งหมด 10 ไฟล์ ประกอบด้วยข้อมูล รูปแบบสองเส้า 5 ไฟล์ และ รูปแบบสามเส้า 5 ไฟล์ แต่ละรูปแบบแบ่งเป็นเครือข่ายที่มีจำนวนโหนดแตกต่างกันอีก คือ 5, 10, 15, 20 และ 25 โหนด ดังแสดงไว้ในตารางที่ 4

ตารางที่ 4 ไฟล์รูปร่างการต่อของเครือข่าย (Nxx_yT.TXT)

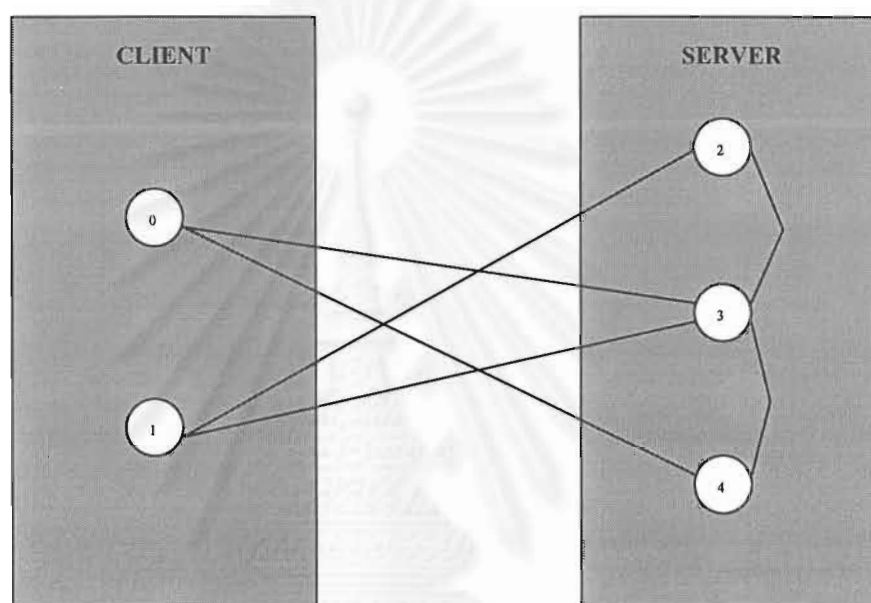
รูปแบบสองเส้า			รูปแบบสามเส้า			
ชื่อไฟล์	Client	Server	ชื่อไฟล์	Client	Application Server	Database Server
N05_2T.TXT	0, 1	2, 3, 4	N05_3T.TXT	0, 1	2	3, 4
N10_2T.TXT	0, 1, 2, 3, 4	5, 6, 7, 8, 9	N10_3T.TXT	0, 1, 2, 3, 4	5, 6	7, 8, 9
N15_2T.TXT	0, 1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 12, 13, 14	N15_3T.TXT	0, 1, 2, 3, 4, 5, 6	7, 8	9, 10, 11, 12, 13, 14
N20_2T.TXT	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10, 11, 12, 13, 14, 15, 16, 17, 18, 19	N20_3T.TXT	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10, 11, 12, 13	14, 15, 16, 17, 18, 19
N25_2T.TXT	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	N25_3T.TXT	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	12, 13, 14, 15	16, 17, 18, 19, 20, 21, 22, 23, 24

จากข้อมูลทั้ง 2 ชุดในตารางที่ 4 จะเห็นว่าสถาปัตยกรรมของเครือข่ายแบบสองเส้าจะแบ่งเป็น โหนดของ Client และ Server ซึ่งจะแสดงการตัวอย่างการเชื่อมต่อแต่ละโหนดของไฟล์ N5_2T.TXT และไฟล์ N10_2T.TXT ในรูปที่ 4.1.1 และในรูปที่ 4.1.2 ตามลำดับ ส่วนสถาปัตยกรรมของเครือข่ายแบบสามเส้าจะแบ่งเป็น โหนดของ Client, Application Server และ Database Server ซึ่งจะแสดงการตัวอย่างการเชื่อมต่อแต่ละโหนดของไฟล์ N5_3T.TXT และไฟล์ N10_3T.TXT ในรูปที่ 4.2.1 และในรูปที่ 4.2.2 ตามลำดับ

4.1 เครือข่ายแบบสองเสา

ในการจำลองสถานการณ์แบบสองเสา มีการกำหนดรูปร่างของเครือข่ายสองเสาที่ใช้ทำการทดลองดังนี้

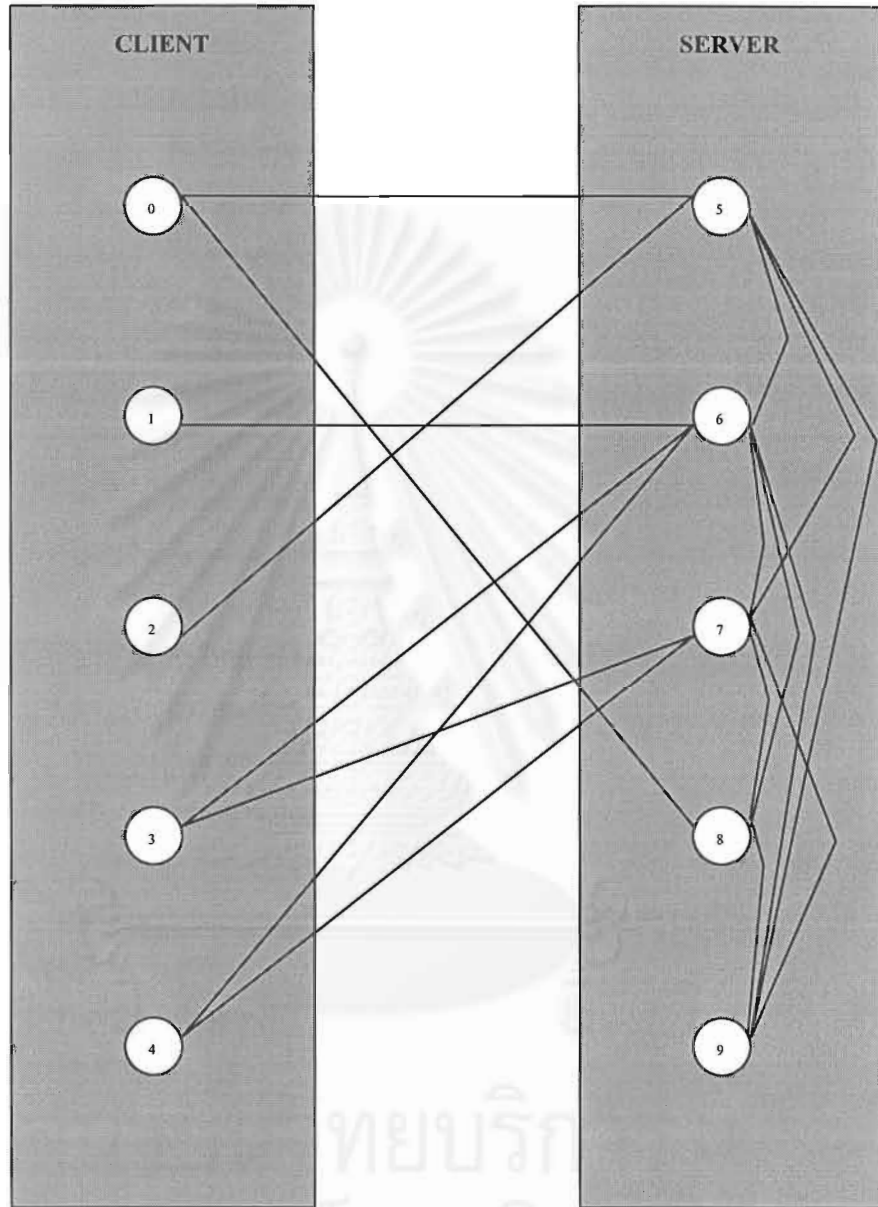
รูปที่ 4.1.1 รูปร่างของเครือข่ายสองเสาขนาด 5 โหนด (N05_2T.TXT)



รูปที่ 4.1.1 เป็นรูปร่างของเครือข่ายสองเสาขนาด 5 โหนด โดยกำหนดให้โหนดที่ 0 และ 1 เป็นโหนดของ Client และให้โหนดที่ 2, 3 และ 4 เป็นโหนดของ Server ซึ่งสร้างเส้นเชื่อมด้วยวิธีการสุ่มระหว่างชั้น Client กับ Server และระหว่างชั้น Server กับ Server

จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4.1.2 รูปร่างของเครือข่ายสองเส้าขนาด 10 โหนด (N10_2T.TXT)



รูปที่ 4.1.2 เป็นรูปร่างของเครือข่ายสองเส้าขนาด 10 โหนด โดยกำหนดให้โหนดที่ 0, 1, 2, 3 และ 4 เป็นโหนดของ Client และให้โหนดที่ 5, 6, 7, 8 และ 9 เป็นโหนดของ Server ซึ่งสร้างเส้นเชื่อมด้วยวิธีการสุ่มระหว่างชั้น Client กับ Server และระหว่างชั้น Server กับ Server

ผลของการทดลองส่ง packet ในเครือข่ายเป็นจำนวน 20 batch, batch ละ 5 ครั้ง เป็นดังนี้
กำหนดให้ X และ Y คือตัวแปรสุ่มที่มีการแจกแจงชี้กำลัง (Exponential Distribution)

$$X, Y \sim \text{Exp}(\mu)$$

$$f(x) = \mu e^{-\mu x}, x \in X$$

$$f(y) = \mu e^{-\mu y}, y \in Y$$

μ คือค่ากลางของการ service rate

S คือตัวแปรสุ่มที่มีการแจกแจงแบร์นูลลี (Bernoulli Distribution)

$$S \sim \text{Ber}(p)$$

$$S = \begin{cases} 1, & \text{ด้วยความน่าจะเป็น } p \\ 0, & \text{ด้วยความน่าจะเป็น } q \end{cases}$$

p คือค่าความน่าจะเป็นที่จะพบข้อมูลใน Server แต่ละตัว

q คือค่าความน่าจะเป็นที่จะไม่พบข้อมูลใน Server แต่ละตัว

n คือจำนวนของ Server ในเครือข่าย

$D \subset \mathbb{N}^2$ คือเซตที่นับได้ของคู่ลำดับ (x,y)

C คือโหนดของ Client

S_i คือโหนดของ Server ตัวที่ i

S_i' คือโหนดของ Server ที่ไม่ใช่ตัวที่ i

TD_i คือค่าของ Delay ของ Server ตัวที่ i

TD คือค่าของ Delay รวมของ Server ทุกตัว

ในการที่ Client ร้องขอข้อมูลจาก Server โดยทำการสืบค้นข้อมูลผ่านชั้นของ Server สามารถแบ่งได้ 2 กรณีคือ

กรณีที่ 1 พบข้อมูลใน Server ตัวที่ i

$$P(S=1), \text{Path}(C-S_i), \text{Delay} = X$$

$$TD_i' = S * X$$

$$= p_i * X$$

กรณีที่ 2 ไม่พบข้อมูลใน Server ตัวที่ i

$$P(S=0), \text{Path}(C-S_i - S_i'), \text{Delay} = Y$$

$$TD_i'' = S * Y$$

$$= q_i * Y$$

∴ ค่าของ Delay ของ Server ตัวที่ i และค่าของ Delay รวมของ Server ทุกตัวคือ

$$TD_i = p_i * X + q_i * Y$$

$$TD = \sum_n TD_i, i = 1, 2, \dots, n$$

ดังนั้นค่าคาดหวังของ Delay ของเครือข่ายสองเส้าคือ

$$\begin{aligned}
 E(TD) &= \sum_n E(TD_i) \\
 &= \sum_n E(p_i * X + q_i * Y) \\
 &= \sum_n [\sum_D (p_i * x + q_i * y) * f(x,y)] \\
 &= \sum_n [p_i * \sum_D x * f(x,y) + q_i * \sum_D y * f(x,y)] \\
 &= \sum_n [p_i * E(X) + q_i * E(Y)]
 \end{aligned}$$

กล่าวโดยสรุปคือ ค่าคาดหวังของ Delay ของเครือข่ายสองเส้า มีค่าเท่ากับผลรวมตามจำนวน Server ของค่าความน่าจะเป็นที่จะพบข้อมูลใน Server ตัวที่ i (p_i) คูณกับค่าคาดหวังของ Delay ซึ่งมีทางเดินจาก Client ไปยัง Server ตัวที่พบข้อมูลโดยตรง ($E(X)$) บวกกับ ค่าความน่าจะเป็นที่จะไม่พบข้อมูลใน Server ตัวที่ i (q_i) คูณกับค่าคาดหวังของ Delay ซึ่งมีทางเดินจาก Client ไป Server ตัวที่ i และต่อไปยัง Server ตัวอื่นที่มีข้อมูล ($E(Y)$)

ผลของการทดลอง Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเส้าขนาด 5 และ 10 โหนด แสดงดังตารางที่ 4.1.1 และ 4.1.2 ตามลำดับ ส่วนผลการทดลองของเครือข่ายสองเส้าขนาด 15, 20 และ 25 โหนด ดูได้จากภาคผนวก ก. ตารางที่ ก.1, ก.2 และ ก.3 ตามลำดับ

ตารางที่ 4.1.1 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเส้าขนาด 5 โหนด

เครือข่ายสองเส้าขนาด 5 โหนด					
Source Node = 0	Delay	P	q	E(TD _i)	
0>3>2	4	0.333	0.667	0.333*E(0)+0.667*E(0)	0
0>3	2	0.334	0.666	0.334*E(4)+0.666*E(8)	0.132988
0>4	2	0.333	0.667	0.333*E(4)+0.667*E(0)	0.016402
E(TD) =					0.149391
Source Node = 1	Delay	p	q	E(TD _i)	
1>2	2	0.333	0.667	0.333*E(4)+0.667*E(0)	0.016402
1>3	2	0.334	0.666	0.334*E(4)+0.666*E(6)	0.084885
1>3>4	3	0.333	0.667	0.334*E(0)+0.667*E(0)	0
E(TD) =					0.101287

ตารางที่ 4.1.2 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเส้าขนาด 10 โหนด

เครือข่ายสองเส้าขนาด 10 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>5	2	0.188	0.812	$0.188 * E(4) + 0.812 * E(14)$	0.415505
0>5>6	3	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
0>5>7	4	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
0>8	2	0.188	0.812	$0.188 * E(4) + 0.812 * E(10)$	0.241307
0>8>9	5	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
E(TD) =					0.656812
Source Node = 1	Delay	p	q	E(TD _i)	
1>6>5	3	0.188	0.812	$0.188 * E(0) + 0.812 * E(0)$	0
1>6	2	0.208	0.792	$0.208 * E(4) + 0.792 * E(30)$	1.674718
1>6>7	5	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
1>6>8	4	0.188	0.812	$0.188 * E(0) + 0.812 * E(0)$	0
1>6>9	3	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
E(TD) =					1.674718
Source Node = 2	Delay	p	q	E(TD _i)	
2>5	2	0.188	0.812	$0.188 * E(4) + 0.812 * E(42)$	3.195202
2>5>6	3	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
2>5>7	4	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
2>5>6>8	9	0.188	0.812	$0.188 * E(0) + 0.812 * E(0)$	0
2>5>6>9	5	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
E(TD) =					3.195202
Source Node = 3	Delay	p	q	E(TD _i)	
3>6>5	3	0.188	0.812	$0.188 * E(0) + 0.812 * E(0)$	0
3>6	2	0.208	0.792	$0.208 * E(4) + 0.792 * E(20)$	0.787144
3>7	1	0.208	0.792	$0.208 * E(2) + 0.792 * E(0)$	0.003094
3>6>8	4	0.188	0.812	$0.188 * E(0) + 0.812 * E(0)$	0
3>6>9	3	0.208	0.792	$0.208 * E(0) + 0.792 * E(0)$	0
E(TD) =					0.790238

เครือข่ายสองเส้าขนาด 10 โหนด (ต่อ)					
Source Node = 4	Delay	p	q	E(TD _i)	
4>6>5	3	0.188	0.812	0.188*E(0) + 0.812*E(0)	0
4>6	2	0.208	0.792	0.208*E(4) + 0.792*E(20)	0.787144
4>7	3	0.208	0.792	0.208*E(6) + 0.792*E(0)	0.021372
4>6>8	4	0.188	0.812	0.188*E(0) + 0.812*E(0)	0
4>6>9	3	0.208	0.792	0.208*E(0) + 0.792*E(0)	0
E(TD) =					0.808516

ตัวอย่างการคำนวณตามตารางที่ 4.1.1 และ 4.1.2 ตามรูปที่ 4.1.2 เช่น Client โหนดที่ 4 (Source Node=4) ของเครือข่ายสองเส้าขนาด 10 โหนด จะเห็นว่าทางเดินจากโหนดที่ 4 ไปยังจุดปลายทางที่เป็น Server แต่ละตัวมี 2 แบบ คือ

- มีทางเดินจาก Client ไปยัง Server ตัวที่พบข้อมูลโดยตรง ได้แก่ 4>6 และ 4>7 เนื่องจากการคำนวณจะคิดค่าของ Delay ในทิศทางไปและกลับ มีผลทำให้ค่า X ของโหนดที่ 6 และ 7 เป็น 4 และ 6 ตามลำดับ ส่วนโหนดของ Server ที่เหลือค่า X จะเท่ากับ 0
- มีทางเดินจาก Client ไป Server ตัวที่ i (ในกรณีนี้ i = 6, 7) และต่อไปยัง Server ตัวอื่นที่มีข้อมูล ได้แก่ 4>6>5, 4>6>8 และ 4>6>9 จะเห็นว่าทางเดินที่เกิดขึ้นเป็นการส่งผ่านโหนดที่ 6 ไปยังโหนดที่ 5, 8 และ 9 และเนื่องจากการคำนวณจะคิดค่าของ Delay ในทิศทางไปและกลับ จึงส่งผลให้ค่า Y ของโหนดที่ 6 มีค่าเท่ากับ 20 ส่วนโหนดของ Server ที่เหลือค่า Y จะเท่ากับ 0

ดังนั้น ค่าคาดหวังของ Delay ของ Server แต่ละโหนดคือ

$$E(TD_5) = 0.188*E(0) + 0.812*E(0) \\ = 0$$

$$E(TD_6) = 0.208*E(4) + 0.792*E(20) \\ = 0.787144$$

$$E(TD_7) = 0.208*E(6) + 0.792*E(0) \\ = 0.021372$$

$$E(TD_8) = 0.188*E(0) + 0.812*E(0) \\ = 0$$

$$E(TD_9) = 0.208*E(0) + 0.792*E(0) \\ = 0$$

∴ ค่าคาดหวังของ Delay รวมของ Server ทุกตัวคือ

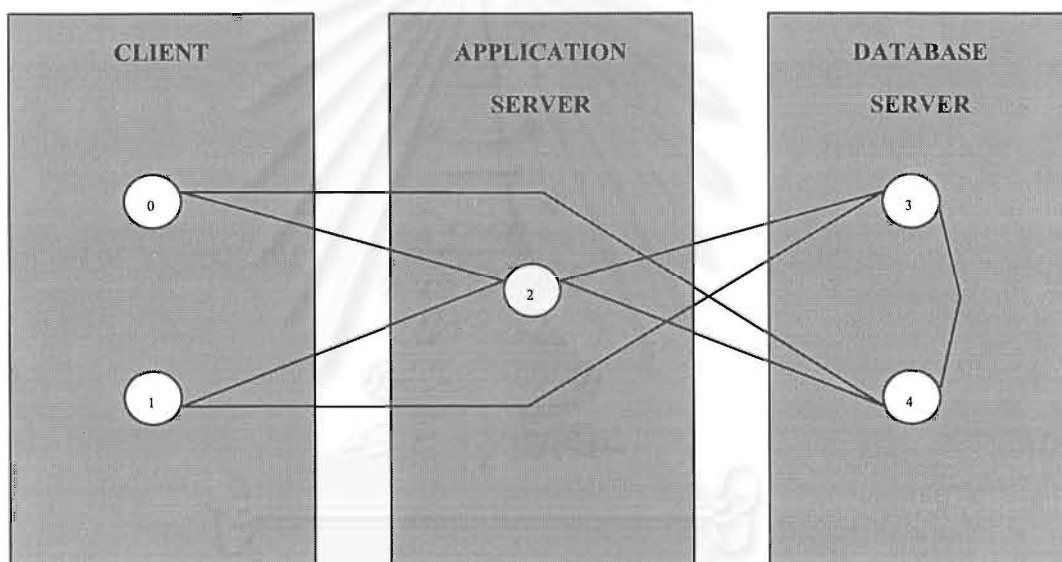
$$E(TD) = 0 + 0.787144 + 0.021372 + 0 + 0 \\ = 0.808516$$

สำหรับผลของการจำลองสถานการณ์เครือข่ายแบบสองเสาที่มีจำนวนโหนดมากกว่า 10 โหนด ได้สรุปไว้ในภาคผนวก ก

4.2 เครือข่ายแบบสามเสา

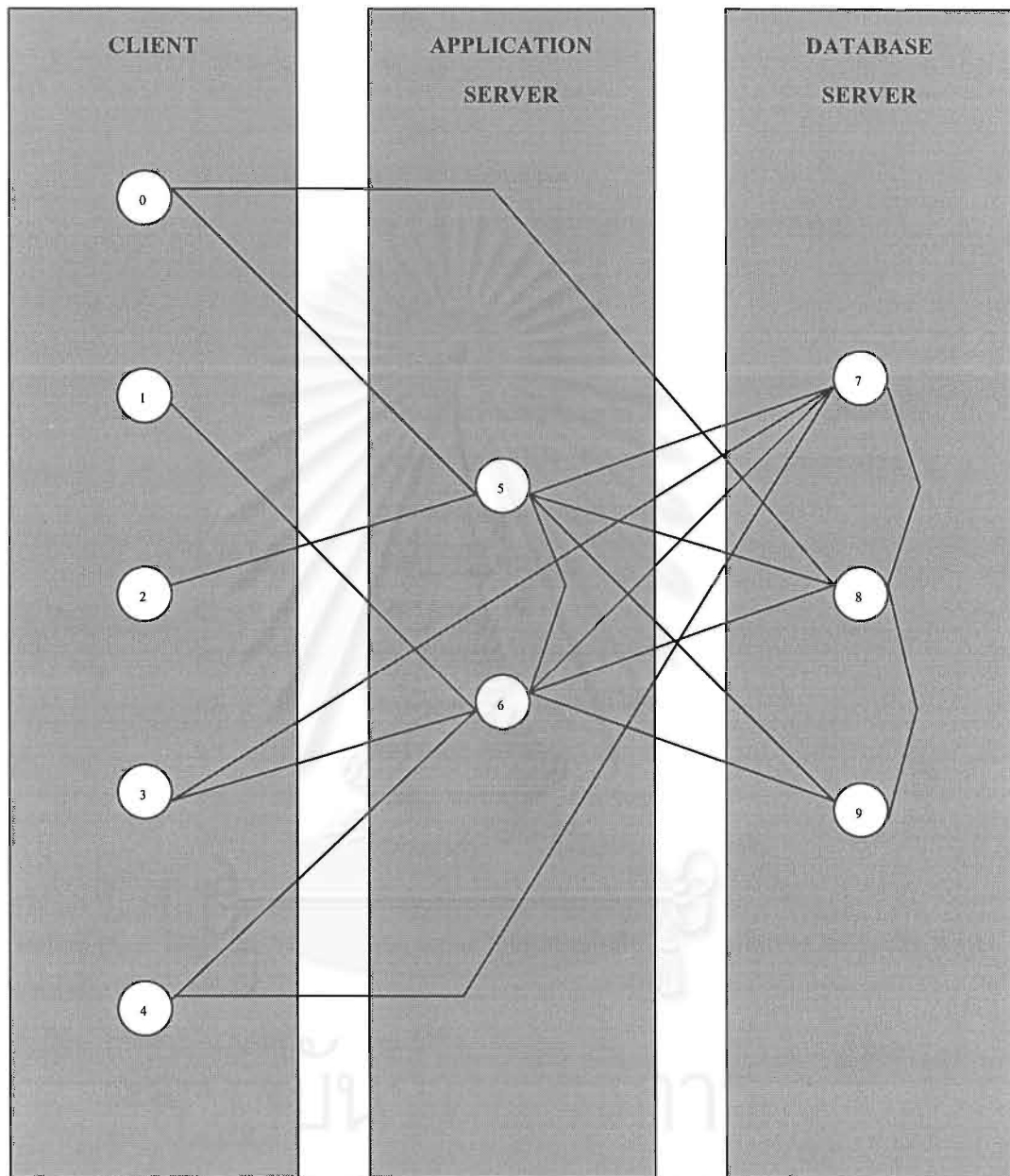
ในการจำลองสถานการณ์แบบสามเสา มีการกำหนดรูปร่างของเครือข่ายสามเสาที่ใช้ทำการทดลองดังนี้

รูปที่ 4.2.1 รูปร่างของเครือข่ายสามเสาขนาด 5 โหนด (N05_3T.TXT)



รูปที่ 4.2.1 เป็นรูปร่างของเครือข่ายสามเสาขนาด 5 โหนด โดยกำหนดให้โหนดที่ 0 และ 1 เป็นโหนดของ Client โหนดที่ 2 เป็นโหนดของ Application Server และโหนดที่ 3 และ 4 เป็นโหนดของ Database Server ซึ่งสร้างเส้นเชื่อมด้วยวิธีการสุ่มระหว่างชั้น Client กับ Application Server และระหว่างชั้น Client กับ Database Server และสร้างเส้นเชื่อมแบบ Fully-Connected ระหว่างชั้น Application Server กับ Application Server และระหว่างชั้น Application Server กับ Database Server ส่วนเส้นเชื่อมระหว่างชั้น Database Server กับ Database Server สร้างเส้นเชื่อมแบบ Bus

รูปที่ 4.2.2 รูปร่างของเครือข่ายสามเส้าขนาด 10 โหนด (N10_3T.TXT)



รูปที่ 4.2.2 เป็นรูปร่างของเครือข่ายสามเส้าขนาด 10 โหนด โดยกำหนดให้โหนดที่ 0, 1, 2, 3 และ 4 เป็นโหนดของ Client โหนดที่ 5 และ 6 เป็นโหนดของ Application Server และโหนดที่ 7, 8 และ 9 เป็นโหนดของ Database Server ซึ่งสร้างเส้นเชื่อมด้วยวิธีการสุ่มระหว่างชั้น Client กับ Application Server และระหว่างชั้น Client กับ Database Server และสร้างเส้นเชื่อมแบบ Fully-Connected ระหว่างชั้น Application Server กับ Application Server และ ระหว่างชั้น Application Server กับ Database Server ส่วนเส้นเชื่อม ระหว่างชั้น Database Server กับ Database Server สร้างเส้นเชื่อมแบบ Bus

ผลของการทดลองส่ง packet ในเครือข่ายเป็นจำนวน 20 batch, batch ละ 5 ครั้ง เป็นดังนี้
กำหนดให้ X และ Y คือตัวแปรสุ่มที่มีการแจกแจงชี้กำลัง (Exponential Distribution)

$$X, Y \sim \text{Exp}(\mu)$$

$$f(x) = \mu e^{-\mu x}, x \in X$$

$$f(y) = \mu e^{-\mu y}, y \in Y$$

μ คือค่ากลางของการ service rate

S คือตัวแปรสุ่มที่มีการแจกแจงแบร์นูลลี (Bernoulli Distribution)

$$S \sim \text{Ber}(p)$$

$$S = \begin{cases} 1, & \text{ด้วยความน่าจะเป็น } p \\ 0, & \text{ด้วยความน่าจะเป็น } q \end{cases}$$

p คือค่าความน่าจะเป็นที่จะพบข้อมูลใน Application Server แต่ละตัว

q คือค่าความน่าจะเป็นที่จะไม่พบข้อมูลใน Application Server แต่ละตัว

n คือจำนวนของ Application Server ในเครือข่าย

$D \subset \mathbb{N}^2$ คือเซตที่นับได้ของคู่ลำดับ (x,y)

C คือ โหนดของ Client

AS_i คือ โหนดของ Application Server ตัวที่ i

AS_i' คือ โหนดของ Application Server ที่ไม่ใช่ตัวที่ i

S คือ โหนดของ Database Server

TD_i คือค่าของ Delay ของ Application Server ตัวที่ i

TD คือค่าของ Delay รวมของ Application Server ทุกตัว

ในการที่ Client ร้องขอข้อมูลจาก Database Server โดยทำการสืบค้นข้อมูลผ่านชั้นของ Application Server สามารถแบ่งได้ 2 กรณีดังนี้

กรณีที่ 1 พบข้อมูลใน Application Server ตัวที่ i

$$P(S=1), \text{Path}(C-AS_i-S), \text{Delay} = X$$

$$TD_i' = S * X$$

$$= p_i * X$$

กรณีที่ 2 ไม่พบข้อมูลใน Application Server ตัวที่ i

$$P(S=0), \text{Path}(C-AS_i-AS_i'-S), \text{Delay} = Y$$

$$TD_i'' = S * Y$$

$$= q_i * Y$$

∴ ค่าของ Delay ของ Application Server ตัวที่ i และค่าของ Delay รวมของ Application Server ทุกตัวคือ

$$TD_i = p_i * X + q_i * Y$$

$$TD = \sum_n TD_i, i = 1, 2, \dots, n$$

ดังนั้นค่าคาดหวังของ Delay ของเครือข่ายสามเส้าคือ

$$\begin{aligned}
 E(TD) &= \sum_n E(TD_i) \\
 &= \sum_n E(p_i * X + q_i * Y) \\
 &= \sum_n [\sum_D (p_i * x + q_i * y) * f(x,y)] \\
 &= \sum_n [p_i * \sum_D x * f(x,y) + q_i * \sum_D y * f(x,y)] \\
 &= \sum_n [p_i * E(X) + q_i * E(Y)]
 \end{aligned}$$

กล่าวโดยสรุปคือ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้า มีค่าเท่ากับผลรวมตามจำนวน Application Server ของ ค่าความน่าจะเป็นที่จะพบข้อมูลใน Application Server ตัวที่ i (p_i) คูณกับค่าคาดหวังของ Delay ซึ่งมีทางเดินจาก Client ไป Application Server ตัวที่พบข้อมูลโดยตรง และส่งต่อไป Database Server ซึ่งเป็นที่เก็บข้อมูล (E(X)) บวกกับ ค่าความน่าจะเป็นที่จะไม่พบข้อมูลใน Application Server ตัวที่ i (q_i) คูณกับค่าคาดหวังของ Delay ซึ่งมีทางเดินจาก Client ไป Application Server ตัวที่ i และต่อไปยัง Application Server ตัวอื่นที่มีข้อมูล และส่งต่อไป Database Server ซึ่งเป็นที่เก็บข้อมูล (E(Y))

ผลของการทดลอง Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้าขนาด 5 และ 10 โหนด แสดงดังตารางที่ 4.2.1 และ 4.2.2 ตามลำดับ ส่วนผลการทดลองของเครือข่ายสองเส้าขนาด 15, 20 และ 25 โหนด ดูได้จากภาคผนวก ข. ตารางที่ ข.1, ข.2 และ ข.3 ตามลำดับ

ตารางที่ 4.2.1 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้าขนาด 5 โหนด

เครือข่ายสามเส้าขนาด 5 โหนด					
Source Node = 0	Delay	P	q	E(TD _i)	
0>2	2	1	0	1 * E(14)	0.500301
0>2>3	4				
0>2>4	3				
E(TD) =					0.500301
Source Node = 1	Delay	P	q	E(TD _i)	
1>2	2	1	0	1 * E(14)	0.500301
1>2>3	4				
1>2>4	3				
E(TD) =					0.500301

ตารางที่ 4.2.2 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้าขนาด 10 โหนด

เครือข่ายสามเส้าขนาด 10 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>5	2	0.499	0.501	$0.499 * E(28) + 0.501 * E(0)$	0.921689
0>5>6	3	0.501	0.499	$0.501 * E(0) + 0.499 * E(0)$	0
0>5>7	4				
0>5>8	5				
0>5>9	5				
E(TD) =					0.921689
Source Node = 1	Delay	p	q	E(TD _i)	
1>6>5	3	0.499	0.501	$0.499 * E(0) + 0.501 * E(0)$	0
1>6	2	0.501	0.499	$0.501 * E(24) + 0.499 * E(0)$	0.692856
1>6>7	4				
1>6>8	4				
1>6>9	4				
E(TD) =					0.692856
Source Node = 2	Delay	p	q	E(TD _i)	
2>5	2	0.499	0.501	$0.499 * E(26) + 0.501 * E(0)$	0.802098
2>5>6	3	0.501	0.499	$0.501 * E(0) + 0.499 * E(0)$	0
2>5>7	4				
2>5>8	4				
2>5>9	5				
E(TD) =					0.802098
Source Node = 3	Delay	p	q	E(TD _i)	
3>6>5	3	0.499	0.501	$0.499 * E(0) + 0.501 * E(0)$	0
3>6	2	0.501	0.499	$0.501 * E(22) + 0.499 * E(0)$	0.588178
3>6>7	4				
3>6>8	3				
3>6>9	4				
E(TD) =					0.588178

เครือข่ายสามเส้าขนาด 10 โหนด (ต่อ)					
Source Node= 4	Delay	p	q	E(TD _i)	
4>6>5	4	0.499	0.501	0.499*E(0) + 0.501*E(0)	0
4>6	3	0.501	0.499	0.501*E(30) + 0.499*E(0)	1.052905
4>6>7	5				
4>6>8	5				
4>6>9	5				
E(TD) =					1.052905

ตัวอย่างการคำนวณตามตารางที่ 4.2.1 และ 4.2.2 ตามรูปที่ 4.2.2 เช่น Client โหนดที่ 4 (Source Node=4) ของเครือข่ายสามเส้าขนาด 10 โหนด จะเห็นว่าทางเดินจากโหนดที่ 4 ไปยังจุดปลายทางที่เป็น Database Server แต่ละตัวมี 2 แบบ คือ

- มีทางเดินจาก Client ไป Application Server ตัวที่พบข้อมูลโดยตรง และส่งต่อไป Database Server ซึ่งเป็นที่เก็บข้อมูล ได้แก่ 4>6>7, 4>6>8 และ 4>6>9 เนื่องจากการคำนวณจะคิดค่าของ Delay ในทิศทางไปและกลับ มีผลทำให้ค่า X ของโหนดที่ 6 เป็น 30 ส่วนโหนดของ Application Server ที่เหลือค่า X จะเท่ากับ 0
- มีทางเดินจาก Client ไป Application Server ตัวที่ i (ในกรณีนี้ i = 6) และต่อไปยัง Application Server ตัวอื่นที่มีข้อมูล และส่งต่อไป Database Server ซึ่งเป็นที่เก็บข้อมูล เช่น 4>6>5>7 เป็นต้น แต่จะเห็นได้ว่าไม่มีทางเดินประเภทนี้จากโหนดที่ 4 ไปยังจุดปลายทางที่เป็น Database Server แต่ละตัว จึงส่งผลให้ค่า Y ของโหนด Application Server ทุกตัวมีค่าเท่ากับ 0

ดังนั้น ค่าคาดหวังของ Delay ของ Application Server แต่ละโหนดคือ

$$E(TD_5) = 0.499*E(0) + 0.501*E(0)$$

$$= 0$$

$$E(TD_6) = 0.501*E(30) + 0.499*E(0)$$

$$= 1.052905$$

∴ ค่าคาดหวังของ Delay รวมของ Application Server ทุกตัวคือ

$$E(TD) = 0 + 1.052905$$

$$= 1.052905$$

สำหรับผลของการจำลองสถานการณ์เครือข่ายแบบสามเส้าที่มีจำนวนโหนดมากกว่า 10 โหนด ได้สรุปไว้ในภาคผนวก ข

บทที่ 5

สรุปผลการวิเคราะห์ข้อมูล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการจำลองสถานการณ์ส่ง packet ในเครือข่ายเป็นจำนวน 100 ครั้ง เพื่อให้ผลลัพธ์ที่ได้มีความถูกต้องใกล้เคียงกับความจริงมากยิ่งขึ้น สามารถสรุปตารางการเปลี่ยนแปลงค่าคาดหวังของ Delay ได้ดังตารางที่ 5.1.1

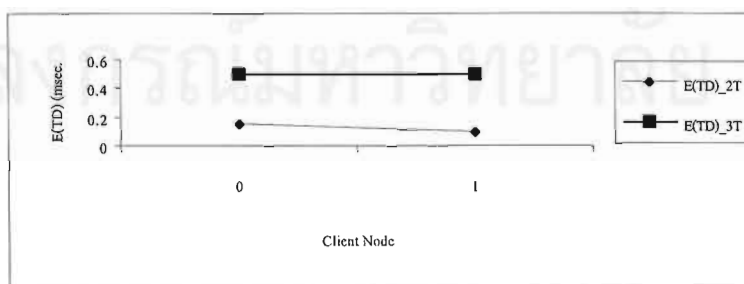
ตารางที่ 5.1.1 ตารางเปรียบเทียบค่าคาดหวังของ Delay

Client Node (1)	E(TD) of Two-Tier (2)	E(TD) of Three-Tier (3)	% of change ((3) – (2) * 100)
เครือข่ายขนาด 5 โหนด			
0	0.149391	0.500301	35.09%
1	0.101287	0.500301	39.90%
เครือข่ายขนาด 10 โหนด			
0	0.656812	0.921689	26.49%
1	1.674718	0.692856	-98.19%
2	3.195202	0.802098	-239.31%
3	0.790238	0.588178	-20.21%
4	0.808516	1.052905	24.44%
เครือข่ายขนาด 15 โหนด			
0	3.241704	0.707804	-253.39%
1	8.404684	0.90754	-749.71%
2	7.333752	0.509771	-682.40%
3	3.448668	0.671447	-277.72%
4	18.26414	1.175269	-1708.89%
5	2.257195	0.422073	-183.51%
6	3.51086	1.320574	-219.03%
เครือข่ายขนาด 20 โหนด			
0	11.20953	1.195717	-1001.38%
1	5.238009	1.656245	-358.18%
2	12.18771	1.195717	-1099.20%
3	4.777414	1.068384	-370.90%
4	4.577338	1.777093	-280.02%
5	6.482951	1.266549	-521.64%
6	16.92434	1.158986	-1576.54%
7	18.61379	1.777093	-1683.67%

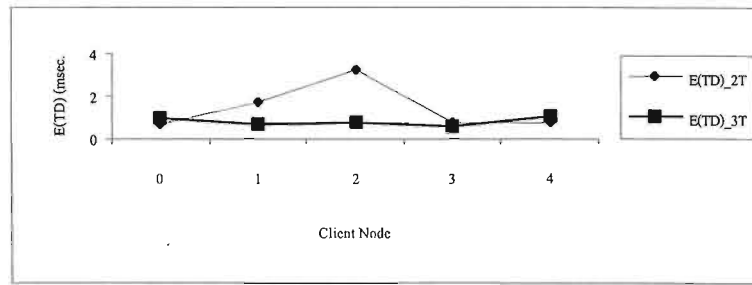
Client Node (1)	E(TD) of Two-Tier (2)	E(TD) of Three-Tier (3)	% of change ((3) – (2) * 100)
8	22.09506	1.656245	-2043.88%
9	16.93313	1.777093	-1515.60%
เครือข่ายขนาด 25 โหนด			
0	16.79708	1.993793	-1480.33%
1	20.69078	2.105672	-1858.51%
2	13.16454	2.386963	-1077.76%
3	17.33144	2.882322	-1444.91%
4	27.58383	1.570121	-2601.37%
5	33.6995	2.246171	-3145.33%
6	28.35925	2.56223	-2579.70%
7	33.3735	2.572179	-3080.13%
8	12.63216	1.708383	-1092.38%
9	32.718	1.636097	-3108.19%
10	34.36716	2.246171	-3212.10%
11	28.83163	3.68647	-2514.52%

เพื่อให้เห็นการเปลี่ยนแปลงค่าคาดหวังของ Delay ที่ชัดเจนยิ่งขึ้น จึงนำค่าในตารางที่ 5.1.1 มาแสดงในรูปของกราฟ ดังรูปที่ 5.1.1 โดยจะแบ่งเป็นรูป (a), (b), (c), (d) และ (e) ซึ่งแสดงถึงค่าคาดหวังของ Delay ที่แบ่งตามขนาดของเครือข่ายคือ ขนาด 5, 10, 15, 20 และ 25 ตามลำดับ จากรูปกราฟจะกำหนดให้แกน x แสดงถึงชื่อโหนดของ Client และแกน y แสดงถึงค่าคาดหวังของ Delay มีหน่วยเป็น milliseconds

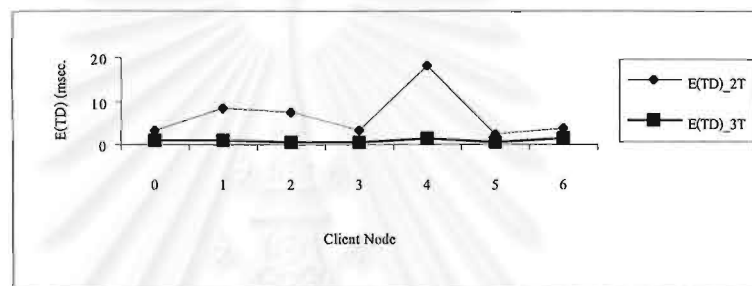
รูปที่ 5.1.1 กราฟเปรียบเทียบค่าคาดหวังของ Delay



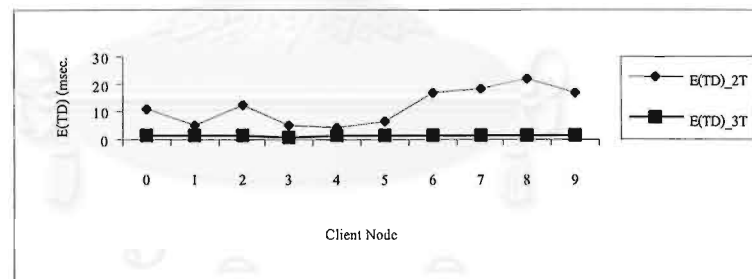
(a)



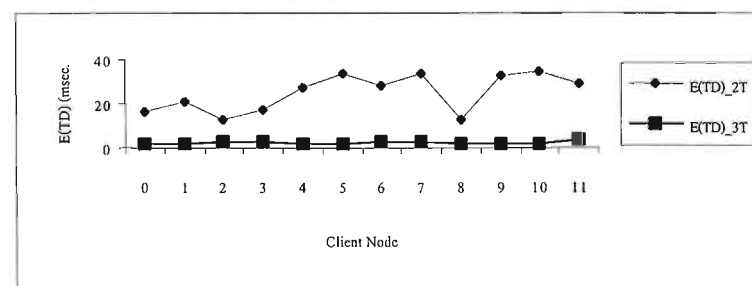
(b)



(c)



(d)



(e)

สำหรับการวิเคราะห์เครือข่ายทั้งแบบสองเส้นและแบบสามเส้น ได้มีการเก็บค่าของข้อมูลที่ใช้ในการคำนวณหาค่า Mean Delay per Line (MDL) และ ค่า Mean Packet Delay (MPD) โดยยึดถือวิธีการคำนวณจาก [1] ดังตัวอย่างการคำนวณแสดงในตารางที่ 5.1.2 และ สรุปผลการคำนวณโดยรวมในตารางที่ 5.1.3 ตามลำดับ

$$\begin{aligned} \text{กำหนดให้} \quad \text{Mean Delay per Line (MDL)} &= \sum(\lambda_i T_i / \lambda) \\ \text{Mean Packet Delay (MPD)} &= \text{MHP} * \text{MDL} \\ \text{Mean number of Hop per Packet (MHP)} &= \lambda / \gamma \\ \text{Queueing and Transmission Time (T}_i) &= 1 / (\mu C_i - \lambda_i) \\ \text{Arrival Rate for channel } i (\lambda_i) &= \sum h_{ij} \gamma_{ij} \\ \lambda &= \sum \lambda_i \\ \gamma &= \sum \gamma_{ij} \\ h_i &= \text{Number of hops in the route from node } i \text{ to node } j \\ \gamma_{ij} &= \text{Traffic between each pair of nodes} \\ \mu &= \text{Mean service rate} \\ C_i &= \text{Capacity of communication channel } i \end{aligned}$$

ตารางที่ 5.1.2 ตารางตัวอย่างการวิเคราะห์ค่า Delay สำหรับเครือข่ายขนาด 5 โหนด
ที่เกิดจากการส่ง packet ออกวิ่ง 100 ครั้ง โดย Source Node = 0

SEQ	Two-Tier					Three-Tier				
	$\Sigma \lambda$	$\Sigma \gamma$	MHP	MDL	MPD	$\Sigma \lambda$	$\Sigma \gamma$	MHP	MDL	MPD
1	140	96	1.46	1.02	1.48	94	60	1.57	1.01	1.59
2	140	112	1.25	1.01	1.27	162	68	2.38	1.03	2.44
3	136	100	1.36	1.02	1.38	162	64	2.53	1.02	2.59
4	88	92	0.96	1.01	0.96	106	64	1.66	1.02	1.69
5	180	100	1.8	1.02	1.83	162	68	2.38	1.03	2.44
6	156	92	1.7	1.02	1.72	162	64	2.53	1.02	2.59
7	172	116	1.48	1.02	1.51	138	72	1.92	1.02	1.95
8	148	100	1.48	1.02	1.51	218	84	2.6	1.03	2.67
9	172	92	1.87	1.02	1.91	162	64	2.53	1.02	2.59
10	124	92	1.35	1.01	1.36	102	60	1.7	1.02	1.73
11	108	96	1.12	1.01	1.14	206	80	2.58	1.03	2.65
12	160	108	1.48	1.02	1.51	162	64	2.53	1.02	2.59
13	164	112	1.46	1.02	1.49	102	60	1.7	1.02	1.73
14	116	84	1.38	1.01	1.4	182	76	2.39	1.03	2.46
15	144	96	1.5	1.02	1.52	158	60	2.63	1.02	2.69

SEQ	Two-Tier					Three-Tier				
	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD
16	156	100	1.56	1.02	1.59	102	60	1.7	1.02	1.73
17	160	100	1.6	1.02	1.63	182	76	2.39	1.03	2.46
18	120	112	1.07	1.01	1.09	166	76	2.18	1.02	2.23
19	120	88	1.36	1.01	1.38	134	68	1.97	1.02	2.01
20	120	96	1.25	1.02	1.27	182	76	2.39	1.03	2.46
21	128	92	1.39	1.01	1.41	142	72	1.97	1.02	2.01
22	128	104	1.23	1.02	1.25	98	56	1.75	1.02	1.78
23	192	116	1.66	1.02	1.69	182	76	2.39	1.03	2.46
24	120	88	1.36	1.02	1.39	142	72	1.97	1.02	2.01
25	152	100	1.52	1.02	1.55	138	60	2.3	1.03	2.36
26	136	84	1.62	1.02	1.65	134	64	2.09	1.02	2.13
27	148	108	1.37	1.02	1.39	142	72	1.97	1.02	2.01
28	140	108	1.3	1.02	1.32	162	68	2.38	1.03	2.44
29	128	100	1.28	1.02	1.3	166	80	2.08	1.02	2.12
30	136	88	1.55	1.02	1.57	142	72	1.97	1.02	2.01
31	104	88	1.18	1.01	1.19	162	68	2.38	1.03	2.44
32	144	104	1.38	1.02	1.41	198	76	2.61	1.03	2.67
33	160	108	1.48	1.02	1.51	94	60	1.57	1.01	1.59
34	136	100	1.36	1.02	1.38	162	68	2.38	1.03	2.44
35	116	92	1.26	1.01	1.28	162	64	2.53	1.02	2.59
36	124	92	1.35	1.02	1.37	106	64	1.66	1.02	1.69
37	152	96	1.58	1.02	1.61	162	68	2.38	1.03	2.44
38	132	112	1.18	1.02	1.2	162	64	2.53	1.02	2.59
39	116	100	1.16	1.01	1.18	138	72	1.92	1.02	1.95
40	124	92	1.35	1.02	1.37	218	84	2.6	1.03	2.67
41	120	88	1.36	1.02	1.38	162	64	2.53	1.02	2.59
42	112	100	1.12	1.01	1.13	102	60	1.7	1.02	1.73
43	164	120	1.37	1.02	1.39	206	80	2.58	1.03	2.65
44	168	100	1.68	1.02	1.71	162	64	2.53	1.02	2.59
45	120	80	1.5	1.02	1.52	102	60	1.7	1.02	1.73
46	128	96	1.33	1.01	1.35	182	76	2.39	1.03	2.46
47	156	104	1.5	1.02	1.52	158	60	2.63	1.02	2.69
48	180	108	1.67	1.02	1.7	102	60	1.7	1.02	1.73
49	128	100	1.28	1.01	1.3	182	76	2.39	1.03	2.46
50	128	88	1.45	1.01	1.48	166	76	2.18	1.02	2.23
51	112	100	1.12	1.01	1.13	134	68	1.97	1.02	2.01
52	140	96	1.46	1.02	1.48	182	76	2.39	1.03	2.46

SEQ	Two-Tier					Three-Tier				
	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD
53	140	112	1.25	1.01	1.27	142	72	1.97	1.02	2.01
54	136	100	1.36	1.02	1.38	98	56	1.75	1.02	1.78
55	88	92	0.96	1.01	0.96	182	76	2.39	1.03	2.46
56	180	100	1.8	1.02	1.83	142	72	1.97	1.02	2.01
57	156	92	1.7	1.02	1.72	138	60	2.3	1.03	2.36
58	172	116	1.48	1.02	1.51	134	64	2.09	1.02	2.13
59	148	100	1.48	1.02	1.51	142	72	1.97	1.02	2.01
60	172	92	1.87	1.02	1.91	162	68	2.38	1.03	2.44
61	124	92	1.35	1.01	1.36	166	80	2.08	1.02	2.12
62	108	96	1.12	1.01	1.14	142	72	1.97	1.02	2.01
63	160	108	1.48	1.02	1.51	162	68	2.38	1.03	2.44
64	164	112	1.46	1.02	1.49	198	76	2.61	1.03	2.67
65	116	84	1.38	1.01	1.4	94	60	1.57	1.01	1.59
66	144	96	1.5	1.02	1.52	162	68	2.38	1.03	2.44
67	156	100	1.56	1.02	1.59	162	64	2.53	1.02	2.59
68	160	100	1.6	1.02	1.63	106	64	1.66	1.02	1.69
69	120	112	1.07	1.01	1.09	162	68	2.38	1.03	2.44
70	120	88	1.36	1.01	1.38	162	64	2.53	1.02	2.59
71	120	96	1.25	1.02	1.27	138	72	1.92	1.02	1.95
72	128	92	1.39	1.01	1.41	218	84	2.6	1.03	2.67
73	128	104	1.23	1.02	1.25	162	64	2.53	1.02	2.59
74	192	116	1.66	1.02	1.69	102	60	1.7	1.02	1.73
75	120	88	1.36	1.02	1.39	206	80	2.58	1.03	2.65
76	152	100	1.52	1.02	1.55	162	64	2.53	1.02	2.59
77	136	84	1.62	1.02	1.65	102	60	1.7	1.02	1.73
78	148	108	1.37	1.02	1.39	182	76	2.39	1.03	2.46
79	140	108	1.3	1.02	1.32	158	60	2.63	1.02	2.69
80	128	100	1.28	1.02	1.3	102	60	1.7	1.02	1.73
81	136	88	1.55	1.02	1.57	182	76	2.39	1.03	2.46
82	104	88	1.18	1.01	1.19	166	76	2.18	1.02	2.23
83	144	104	1.38	1.02	1.41	134	68	1.97	1.02	2.01
84	160	108	1.48	1.02	1.51	182	76	2.39	1.03	2.46
85	136	100	1.36	1.02	1.38	142	72	1.97	1.02	2.01
86	116	92	1.26	1.01	1.28	98	56	1.75	1.02	1.78
87	124	92	1.35	1.02	1.37	182	76	2.39	1.03	2.46
88	152	96	1.58	1.02	1.61	142	72	1.97	1.02	2.01
89	132	112	1.18	1.02	1.2	138	60	2.3	1.03	2.36

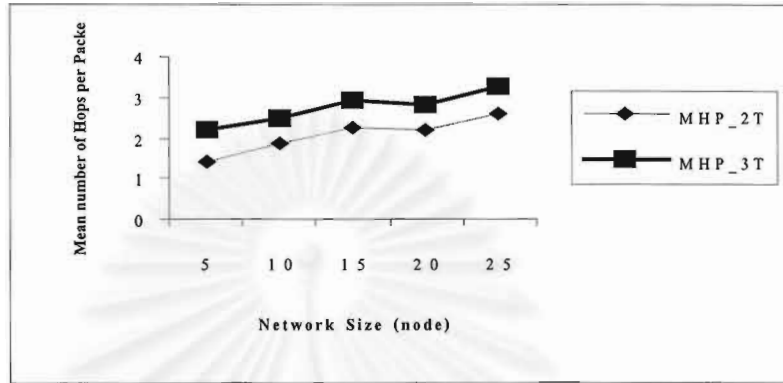
SEQ	Two-Tier					Three-Tier				
	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD	$\Sigma \lambda$	$\Sigma \gamma$	λ/γ	MDL	MPD
90	116	100	1.16	1.01	1.18	134	64	2.09	1.02	2.13
91	124	92	1.35	1.02	1.37	142	72	1.97	1.02	2.01
92	120	88	1.36	1.02	1.38	162	68	2.38	1.03	2.44
93	112	100	1.12	1.01	1.13	166	80	2.08	1.02	2.12
94	164	120	1.37	1.02	1.39	142	72	1.97	1.02	2.01
95	168	100	1.68	1.02	1.71	162	68	2.38	1.03	2.44
96	120	80	1.5	1.02	1.52	198	76	2.61	1.03	2.67
97	128	96	1.33	1.01	1.35	94	60	1.57	1.01	1.59
98	156	104	1.5	1.02	1.52	162	68	2.38	1.03	2.44
99	180	108	1.67	1.02	1.7	162	64	2.53	1.02	2.59
100	128	100	1.28	1.01	1.3	106	64	1.66	1.02	1.69
AVERAGE	138.72	98.92	1.4029	1.017	1.4255	150.8	68.56	2.185	1.0233	2.235

ตารางที่ 5.1.3 ตารางสรุปการวิเคราะห์ค่า Delay สำหรับเครือข่ายขนาด 5, 10, 15, 20 และ 25 โหนด
ที่เกิดจากการส่ง packet ออกวิ่ง 100 ครั้ง โดย Source Node = 0

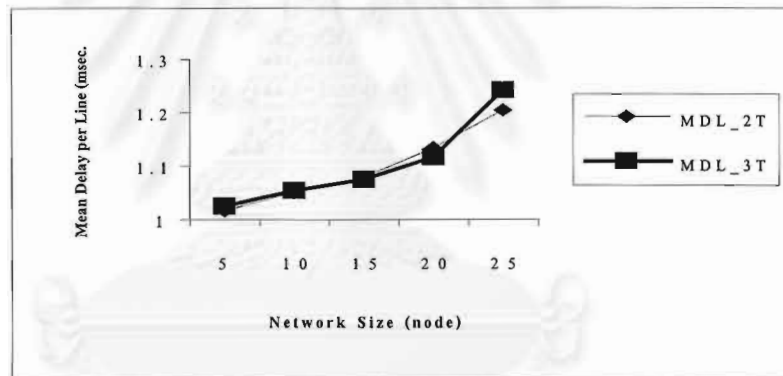
ขนาดเครือข่าย (โหนด)	Mean number of Hops per Packet (MHP)		Mean Delay per Line (MDL)		Mean Packet Delay (MPD)	
	Two-Tier	Three-Tier	Two-Tier	Three-Tier	Two-Tier	Three-Tier
5	1.4029	2.185	1.017	1.0233	1.4255	2.235
10	1.8356	2.4977	1.052	1.0545	1.9327	2.6371
15	2.2739	2.9164	1.0804	1.0759	2.4581	3.1416
20	2.2229	2.7955	1.1346	1.1166	2.5248	3.1245
25	2.6113	3.2922	1.2023	1.2413	3.1438	4.0984

เพื่อให้เห็นการเปรียบเทียบการวิเคราะห์ค่า Delay ของสถาปัตยกรรมแบบสองเส้ากับสถาปัตยกรรมแบบสามเส้าที่ชัดเจนยิ่งขึ้น จึงนำค่าในตารางที่ 5.1.3 มาแสดงในรูปของกราฟ ดังรูปที่ 5.1.2 โดยจะแบ่งเป็นรูป (a), (b) และ (c) ซึ่งแสดงถึงค่า Mean number of Hops per Packet (MHP), ค่า Mean Delay per Line (MDL) และค่า Mean Packet Delay (MPD) ตามลำดับ จากกราฟจะกำหนดให้แกน x แสดงถึงขนาดของเครือข่าย (จำนวนโหนด) และแกน y แสดงถึงค่า MHP, MDL และ MPD ตามลำดับ

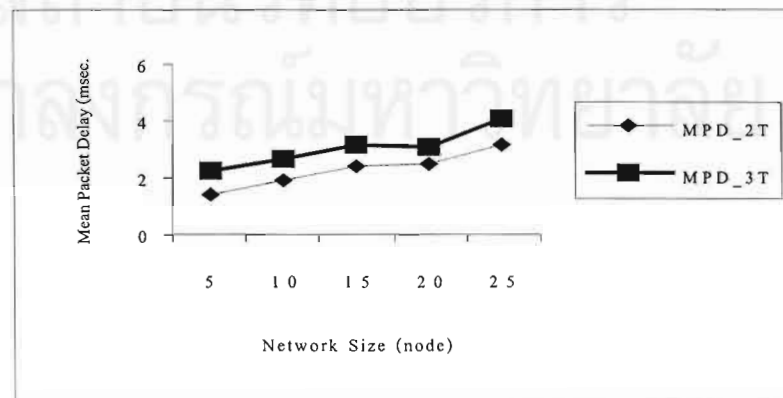
รูปที่ 5.1.2 กราฟสรุปการวิเคราะห์ห้ค่า Delay สำหรับเครือข่ายขนาด 5, 10, 15, 20 และ 25 โหนด
 ที่เกิดจากการส่ง packet ออกวิ่ง 100 ครั้ง โดย Source Node = 0



(a)



(b)



(c)

5.2 อภิปรายผล

เนื่องจากการทดลองได้มีการออกแบบโมเดลแบบง่าย (Simple model) เป็นระบบกระจาย (Distributed System) การเก็บข้อมูลใน Database จะมีลักษณะการกระจาย ดังนั้นค่า Total Delay (TD) ที่ได้จากการคำนวณ เป็นค่าที่เกิดจาก ค่าความล่าช้าในการสืบค้น (Search Delay) บวกกับค่าความล่าช้าในการส่งผ่าน (Transmission Delay) ซึ่งการค้นหาข้อมูลจะคิดเฉพาะ การค้นหาที่ Primary site (Direct connection) โดยไม่รวมการค้นหาที่ Secondary site (Indirect connection)

จุดเด่นของรูปแบบสามเส้าคือ Application Server จะเก็บรายละเอียดของตำแหน่งข้อมูลที่บ้านทึกไว้ในชั้นของ Database Server โดยทำหน้าที่เป็น DBMS การค้นหาข้อมูลจะทำผ่านชั้นของ Application Server เท่านั้น ซึ่งจะมีความรู้เกี่ยวกับ Database Server ที่กระจายอยู่ในเครือข่าย อันเป็นการเพิ่ม Hit ratio สูงขึ้น (Hit ratio เทียบได้กับค่าความน่าจะเป็นของการพบข้อมูลที่ต้องการ คือค่า p นั่นเอง) ดังนั้น Delay ที่เกิดขึ้นหลัง Application Server จะเป็นเพียง delay ที่เกิดจากการส่งผ่านค่า (transmission delay) จาก Database Server ไปที่ Client เท่านั้น ในขณะที่รูปแบบสองเส้า การจัดการฐานข้อมูลและการค้นหาข้อมูลจะทำที่ Server ทั้งหมด ทำให้ภาระงานทั้งหมดตกอยู่ที่ Database Server ตัวเดียว

จากตารางที่ 5.1.1 หรือรูปที่ 5.1.1 จะเห็นได้ว่า ถ้าเครือข่ายมีขนาดใหญ่ขึ้น ค่าคาดหวังของ Total delay จะขึ้นอยู่กับค่าความล่าช้าในการสืบค้น (Search Delay) มีผลทำให้ค่าคาดหวังของ Total delay ของเครือข่ายแบบสามเส้าจะดีกว่าเครือข่ายแบบสองเส้า ทั้งนี้เพราะ เครือข่ายแบบสามเส้ามี Application Server ซึ่งเป็นเสมือน Intelligent Manager เป็นที่เก็บ Information ของ Database Server ดังนั้นการสืบค้นข้อมูลที่ทำผ่านชั้นของ Application Server ในเครือข่ายแบบสามเส้าจะเสียเวลาน้อยกว่า เครือข่ายแบบสองเส้าซึ่งการสืบค้นข้อมูลทำในชั้นของ Server โดยตรง แต่ถ้าเครือข่ายมีขนาดเล็ก ตัวอย่างเช่น เครือข่ายขนาด 5 โหนด ความแตกต่างของจำนวนของ Application Server ในเครือข่ายแบบสามเส้า และจำนวนโหนดของ Server ในเครือข่ายแบบสองเส้ามีไม่มาก ค่าคาดหวังของ Total delay จะขึ้นอยู่กับค่าความล่าช้าในการส่งผ่าน (Transmission Delay) ทำให้ค่าคาดหวังของ Total delay ของเครือข่ายแบบสองเส้าจะดีกว่าเครือข่ายแบบสามเส้า เพราะเครือข่ายแบบสามเส้ามี Overhead สูง ค่าหลักๆ ได้แก่ Mean number of Hops per Packet ซึ่งมากกว่าเครือข่ายแบบสองเส้า ส่งผลทำให้มีค่าความล่าช้าในการส่งผ่านมากกว่า ถึงแม้ว่าจะมีค่า Mean Delay per Line ที่แตกต่างกันน้อยก็ตาม ดังแสดงให้เห็นในตารางที่ 5.1.2 และ 5.1.3 หรือรูปที่ 5.1.2

5.3 ข้อเสนอแนะ

โครงการการวิจัยนี้สามารถขยายงานในอนาคตได้ดังนี้

1. เนื่องจากการวิจัยนี้มีการค้นหาข้อมูลจะคิดเฉพาะ การค้นหาที่ Primary site (Direct connection) ดังนั้นงานที่น่าจะขยายในอนาคตควรเพิ่มการค้นหาที่ Secondary site (Indirect connection) ด้วย
2. ควรขยายงานวิจัยให้รูปแบบของเครือข่ายมีขนาดใหญ่หลายๆ ($n \gg 1$) และมีความซับซ้อนมากกว่าเดิม (Complex Network)

3. ควรพิจารณาปัญหาคอขวด (Bottleneck) ที่อาจเกิดขึ้นเนื่องจากการเพิ่มขึ้นของ Application Server ของเครือข่ายแบบสามเส้า
4. ควรพิจารณาปัญหาการรอคอย (Queueing) ที่เกิดขึ้นในชั้นของ Application Server ของเครือข่ายแบบสามเส้า และเปรียบเทียบกับผลที่เกิดขึ้นในชั้นของ Server ของเครือข่ายแบบสองเส้า



จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Andrew S. Tanenbaum. Computer Networks. Englewood Cliffs New Jersey: Prentice-Hall Inc., 1981.
2. Baruch Awerbuch, Amotz Bar-Noy and Madan Gopal. Approximate Distributed Bellman-Ford Algorithm. IEEE (1991): 1206-1213.
3. D. Bertsekas and R. Gallager. Data Networks. Englewood Cliffs New Jersey: Prentice-Hall Inc., 1987.
4. Edward R. Dougherty. Probability and Statistics for the Engineering, Computing and Physical Sciences. Englewood Cliffs New Jersey: Prentice-Hall Inc., 1990.
5. R.R. Yager. A Procedure for Ordering Fuzzy Subsets of Unit Interval. Information and Sciences 24 (1981): 143-161.
6. Seidhar Pithani and Adarshpal S. Sethi. Fuzzy Set Delay Representation for Computer Network Routing Algorithm. IEEE (1993): 286-293.



จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ภาคผนวก ก.

ผลการทดลองของเครือข่ายแบบสองเสา ที่จำนวนโหนดมากกว่า 10 โหนด

ตารางที่ ก.1 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเสาขนาด 15 โหนด

เครือข่ายสองเสาขนาด 15 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>8>7	5	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
0>8	2	0.129	0.871	$0.129 * E(4) + 0.871 * E(30)$	1.836853
0>8>9	5	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
0>12>10	4	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
0>8>11	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
0>12	2	0.13	0.87	$0.13 * E(4) + 0.87 * E(26)$	1.40485
0>12>13	4	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
0>12>13>14	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
E(TD) =					3.241704
Source Node = 1	Delay	p	q	E(TD _i)	
1>7	2	0.117	0.883	$0.117 * E(4) + 0.883 * E(10)$	0.240254
1>7>8	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
1>14>9	4	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
1>14>13>12>10	8	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
1>14>13>12>11	13	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
1>14>13>12	6	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
1>14>13	3	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
1>14	2	0.129	0.871	$0.129 * E(4) + 0.871 * E(68)$	8.164431
E(TD) =					8.404684

<i>Source Node = 2</i>	Delay	p	q	$E(TD_i)$	
2>8>7	5	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
2>8	2	0.129	0.871	$0.129 * E(4) + 0.871 * E(64)$	7.333752
2>8>9	4	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
2>8>12>10	6	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
2>8>11	4	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
2>8>12	4	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
2>8>13	4	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
2>8>9>14	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
$E(TD) =$					7.333752
<i>Source Node = 3</i>	Delay	p	q	$E(TD_i)$	
3>7	1	0.117	0.883	$0.117 * E(2) + 0.883 * E(38)$	2.882219
3>7>8	4	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
3>7>13>14>9	6	0.13	0.87	$0.13 * E(0) + 0.87 * E(10)$	0
3>12>10	3	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
3>12>11	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
3>12	2	0.13	0.87	$0.13 * E(4) + 0.87 * E(16)$	0.566449
3>7>13	4	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
3>7>13>14	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
$E(TD) =$					3.448668
<i>Source Node = 4</i>	Delay	p	q	$E(TD_i)$	
4>7	3	0.117	0.883	$0.117 * E(6) + 0.883 * E(108)$	18.26414
4>7>8	6	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
4>7>13>14>9	7	0.13	0.87	$0.13 * E(0) + 0.87 * E(10)$	0
4>7>6>10	8	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
4>7>8>11	14	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
4>7>13>12	8	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
4>7>13	5	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
4>7>13>14	6	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
$E(TD) =$					18.26414

<i>Source Node = 5</i>	Delay	p	q	$E(TD_i)$	
5>8>7	4	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
5>8	1	0.129	0.871	$0.129 * E(2) + 0.871 * E(22)$	1.037437
5>13>14>9	4	0.13	0.87	$0.13 * E(0) + 0.87 * E(10)$	0
5>13>10	5	0.117	0.883	$0.117 * E(0) + 0.883 * E(0)$	0
5>8>11	4	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
5>8>12	3	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
5>13	2	0.118	0.882	$0.118 * E(4) + 0.882 * E(24)$	1.219758
5>13>14	3	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
$E(TD) =$					2.257195
<i>Source Node = 6</i>	Delay	p	q	$E(TD_i)$	
6>7	3	0.117	0.883	$0.117 * E(6) + 0.883 * E(12)$	0.342374
6>7>8	6	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
6>10>9	4	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
6>10	1	0.117	0.883	$0.117 * E(2) + 0.883 * E(40)$	3.168486
6>10>11	4	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
6>10>12	3	0.13	0.87	$0.13 * E(0) + 0.87 * E(0)$	0
6>10>13	4	0.118	0.882	$0.118 * E(0) + 0.882 * E(0)$	0
6>10>9>14	5	0.129	0.871	$0.129 * E(0) + 0.871 * E(0)$	0
$E(TD) =$					3.51086

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ก.2 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเล้าขนาด 20 โหนด

เครือข่ายสองเล้าขนาด 20 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>11>10	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
0>11	2	0.1	0.9	$0.1 * E(4) + 0.9 * E(80)$	11.20471
0>11>12	4	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
0>11>13	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
0>11>18>14	5	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
0>11>15	3	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
0>11>15>16	11	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
0>11>17	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
0>11>18	4	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
0>19	2	0.098	0.902	$0.098 * E(4) + 0.902 * E(0)$	0.004827
E(TD) =					11.20953
Source Node = 1	Delay	p	q	E(TD _i)	
1>10	2	0.1	0.9	$0.1 * E(4) + 0.9 * E(18)$	0.728578
1>14>15>11	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
1>14>12	4	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
1>14>13	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
1>14	2	0.101	0.899	$0.101 * E(4) + 0.899 * E(48)$	4.509431
1>14>15	3	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
1>10>16	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
1>14>18>17	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
1>14>18	3	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
1>10>19	4	0.098	0.902	$0.098 * E(0) + 0.902 * E(0)$	0
E(TD) =					5.238009

<i>Source Node = 2</i>	Delay	p	q	$E(TD_i)$	
2>11>10	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
2>11	2	0.1	0.9	$0.1 * E(4) + 0.9 * E(84)$	12.18771
2>11>12	3	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
2>11>13	3	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
2>11>15>14	5	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
2>11>15	3	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
2>11>15>16	11	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
2>11>17	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
2>11>18	3	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
2>11>19	5	0.098	0.902	$0.098 * E(0) + 0.902 * E(0)$	0
$E(TD) =$					12.18771
<i>Source Node = 3</i>	Delay	p	q	$E(TD_i)$	
3>10	1	0.1	0.9	$0.1 * E(2) + 0.9 * E(24)$	1.245146
3>10>11	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
3>14>12	5	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
3>14>13	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
3>14	2	0.101	0.899	$0.101 * E(4) + 0.899 * E(42)$	3.532268
3>14>15	3	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
3>10>16	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
3>14>18>17	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
3>14>18	3	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
3>10>19	4	0.098	0.902	$0.098 * E(0) + 0.902 * E(0)$	0
$E(TD) =$					4.777414

<i>Source Node = 4</i>	Delay	p	q	E(TD _i)	
4>18>19>10	4	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
4>11	3	0.1	0.9	0.1*E(6) + 0.9*E(18)	0.733928
4>11>12	5	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
4>11>13	4	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
4>18>14	2	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
4>18>14>15	4	0.099	0.901	0.099*E(0) + 0.901*E(0)	0
4>18>17>16	8	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
4>18>17	2	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
4>18	1	0.101	0.899	0.101*E(2) + 0.899*E(44)	3.84341
4>18>19	2	0.098	0.902	0.098*E(0) + 0.902*E(0)	0
E(TD) =					4.577338
<i>Source Node = 5</i>	Delay	p	q	E(TD _i)	
5>11>10	5	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
5>11	2	0.1	0.9	0.1*E(4) + 0.9*E(54)	5.588958
5>11>12	4	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
5>11>13	4	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
5>15>14	4	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
5>15	3	0.099	0.901	0.099*E(6) + 0.901*E(20)	0.893993
5>15>16	6	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
5>11>17	5	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
5>11>18	4	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
5>11>19	5	0.098	0.902	0.098*E(0) + 0.902*E(0)	0
E(TD) =					6.482951

<i>Source Node = 6</i>	Delay	p	q	$E(TD_i)$	
6>10	1	0.1	0.9	$0.1 * E(2) + 0.9 * E(102)$	16.92434
6>10>11	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
6>10>11>12	8	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
6>10>11>13	7	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
6>10>19>18>14	6	0.101	0.899	$0.101 * E(4) + 0.899 * E(42)$	0
6>10>15	4	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
5>10>16	4	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
6>10>19>18>17	8	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
6>10>19>18	6	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
6>10>19	4	0.098	0.902	$0.098 * E(0) + 0.902 * E(0)$	0
$E(TD) =$					16.92434
<i>Source Node = 7</i>	Delay	p	q	$E(TD_i)$	
7>11>10	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
7>11	3	0.1	0.9	$0.1 * E(6) + 0.9 * E(108)$	18.61379
7>11>12	5	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
7>11>13	5	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
7>11>15>14	6	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
7>11>15	4	0.099	0.901	$0.099 * E(0) + 0.901 * E(0)$	0
7>11>15>16	12	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
7>11>17	6	0.1	0.9	$0.1 * E(0) + 0.9 * E(0)$	0
7>11>18	5	0.101	0.899	$0.101 * E(0) + 0.899 * E(0)$	0
7>11>19	6	0.098	0.902	$0.098 * E(0) + 0.902 * E(0)$	0
$E(TD) =$					18.61379

<i>Source Node = 8</i>	Delay	p	q	E(TD _i)	
8>10	2	0.1	0.9	0.1*E(4) + 0.9*E(120)	22.09506
8>10>11	5	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
8>10>11>12	9	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
8>10>11>13	8	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
8>10>19>18>14	7	0.101	0.899	0.101*E(4) + 0.899*E(42)	0
8>10>15	5	0.099	0.901	0.099*E(0) + 0.901*E(0)	0
8>10>16	5	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
8>10>19>18>17	9	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
8>10>19>18	7	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
8>10>19	5	0.098	0.902	0.098*E(0) + 0.902*E(0)	0
E(TD) =					22.09506
<i>Source Node = 9</i>	Delay	p	q	E(TD _i)	
9>11>10	5	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
9>11	3	0.1	0.9	0.1*E(6) + 0.9*E(102)	16.93313
9>11>12	4	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
9>11>13	4	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
9>11>15>14	6	0.101	0.899	0.101*E(0) + 0.899*E(0)	0
9>11>15	4	0.099	0.901	0.099*E(0) + 0.901*E(0)	0
9>11>15>16	12	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
9>11>17	6	0.1	0.9	0.1*E(0) + 0.9*E(0)	0
9>11>18	4	0.101	0.899	0.101*E(0) + 0.899*E(10)	0
9>11>19	6	0.098	0.902	0.098*E(0) + 0.902*E(0)	0
E(TD) =					16.93313

ตารางที่ ก.3 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสองเส้าขนาด 25 โหนด

เครือข่ายสองเส้าขนาด 25 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>12	2	0.079	0.921	$0.079 * E(4) + 0.921 * E(40)$	3.306918
0>12>13	5	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
0>20>14	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
0>20>14>15	7	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
0>12>18>16	7	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
0>20>24>23>17	8	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
0>12>18	5	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
0>20>19	4	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
0>20	2	0.08	0.92	$0.08 * E(4) + 0.92 * E(88)$	13.49016
0>12>21	3	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
0>20>14>22	10	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
0>20>24>23	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
0>20>24	5	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					16.79708
Source Node = 1	Delay	p	q	E(TD _i)	
1>12	2	0.079	0.921	$0.079 * E(4) + 0.921 * E(26)$	1.484316
1>19>13	3	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
1>19>20>14	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
1>19>18>16>15	8	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
1>19>18>16	7	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
1>19>20>24>23>17	8	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
1>19>18	5	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
1>19	2	0.071	0.929	$0.071 * E(4) + 0.929 * E(108)$	19.20646
1>19>20	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
1>12>21	3	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
1>12>21>22	10	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
1>19>20>24>23	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
1>19>20>24	5	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					20.69078

<i>Source Node = 2</i>	Delay	p	q	$E(TD_i)$	
2>14>20>12	9	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
2>16>13	4	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
2>14	2	0.08	0.92	$0.08 * E(4) + 0.92 * E(78)$	10.96111
2>16>15	3	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
2>16	1	0.071	0.929	$0.071 * E(2) + 0.929 * E(32)$	2.203431
2>16>17	2	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
2>16>18	3	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
2>14>20>19	7	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
2>14>20	3	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
2>14>20>12>21	12	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
2>14>22	4	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
2>16>17>23	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
2>14>20>24	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					13.16454
<i>Source Node = 3</i>	Delay	p	q	$E(TD_i)$	
3>23>24>20>12	11	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
3>15>16>13	9	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
3>15>14	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
3>15	2	0.079	0.921	$0.079 * E(4) + 0.921 * E(62)$	7.326187
3>15>16	3	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
3>15>16>17	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
3>23>18	4	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
3>23>24>20>19	8	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
3>23>24>20	5	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
3>15>8>21	11	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
3>23>22	5	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
3>23	3	0.08	0.92	$0.08 * E(6) + 0.92 * E(74)$	10.00526
3>23>24	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					17.33144

<i>Source Node = 4</i>	Delay	p	q	E(TD _i)	
4>12	1	0.079	0.921	$0.079 * E(2) + 0.921 * E(136)$	27.58383
4>12>13	4	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
4>12>20>14	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
4>12>18>16>15	7	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
4>12>18>16	6	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
4>12>18>16>17	7	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
4>12>18	4	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
4>12>13>19	6	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
4>12>20	4	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
4>12>21	3	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
4>12>21>22	10	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
4>12>20>24>23	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
4>12>20>24	5	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					27.58383
<i>Source Node = 5</i>	Delay	p	q	E(TD _i)	
5>13>12	5	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
5>13	2	0.071	0.929	$0.071 * E(4) + 0.929 * E(154)$	33.6995
5>13>14	5	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
5>13>16>15	6	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
5>13>16	5	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
5>13>16>17	6	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
5>13>16>18	7	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
5>13>19	4	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
5>13>19>20	7	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
5>13>21	5	0.079	0.921	$0.079 * E(0) + 0.921 * E(0)$	0
5>13>14>22	11	0.071	0.929	$0.071 * E(0) + 0.929 * E(0)$	0
5>13>16>17>23	8	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
5>13>19>20>24	8	0.08	0.92	$0.08 * E(0) + 0.92 * E(0)$	0
E(TD) =					33.6995

<i>Source Node = 6</i>	Delay	p	q	E(TD _i)	
6>24>20>12	10	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
6>24>20>19>13	9	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
6>24>20>14	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
6>15	3	0.079	0.921	0.079*E(6) + 0.921*E(8)	0.169274
6>15>16	4	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
6>24>23>17	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
6>24>23>18	5	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
6>24>23>19	7	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
6>24>20	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
6>24>20>12>21	12	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
6>24>23>22	10	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
6>24>23	3	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
6>24	1	0.08	0.92	0.08*E(2) + 0.92*E(138)	28.18997
E(TD) =					28.35925
<i>Source Node = 7</i>	Delay	p	q	E(TD _i)	
7>14>20>12	11	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
7>14>13	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
7>14	2	0.08	0.92	0.08*E(4) + 0.92*E(154)	33.3735
7>14>15	4	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
7>14>15>16	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
7>14>15>16>17	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
7>14>15>16>18	7	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
7>14>20>19	7	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
7>14>20	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
7>14>20>12>21	12	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
7>14>22	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
7>14>20>24>23	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
7>14>20>24	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
E(TD) =					33.3735

<i>Source Node = 8</i>	Delay	p	q	E(TD _i)	
8>21>12	4	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
8>21>13	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
8>15>14	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
8>15	2	0.079	0.921	0.079*E(4) + 0.921*E(60)	6.909719
8>15>16	4	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
8>15>16>17	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
8>15>16>18	6	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
8>21>13>19	7	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
8>21>20	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
8>21	3	0.079	0.921	0.079*E(6) + 0.921*E(54)	5.722444
8>21>22	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
8>15>16>17>23	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
8>15>24	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
E(TD) =					12.63216
<i>Source Node = 9</i>	Delay	p	q	E(TD _i)	
9>14>20>12	10	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
9>14>13	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
9>14	2	0.08	0.92	0.08*E(4) + 0.92*E(152)	32.718
9>14>15	4	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
9>14>15>16	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
9>14>15>16>17	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
9>14>15>16>18	7	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
9>14>20>19	7	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
9>14>20	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
9>14>20>12>21	12	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
9>14>22	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
9>14>20>24>23	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
9>14>20>24	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
E(TD) =					32.718

<i>Source Node = 10</i>	Delay	p	q	E(TD _i)	
10>13>12	6	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
10>13	3	0.071	0.929	0.071*E(6) + 0.929*E(156)	34.36716
10>13>14	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
10>13>16>15	6	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
10>13>16	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
10>13>16>17	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
10>13>16>18	7	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
10>13>19	4	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
10>13>19>20	7	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
10>13>21	5	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
10>13>14>22	11	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
10>13>16>17>23	8	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
10>13>19>20>24	8	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
E(TD) =					34.36716
<i>Source Node = 11</i>	Delay	p	q	E(TD _i)	
11>14>20>12	9	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
11>14>13	4	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
11>14	2	0.08	0.92	0.08*E(4) + 0.92*E(140)	28.83163
11>14>15	4	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
11>14>15>16	5	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
11>14>15>16>17	6	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
11>14>15>16>18	7	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
11>14>20>19	7	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
11>14>20	3	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
11>14>20>12>21	12	0.079	0.921	0.079*E(0) + 0.921*E(0)	0
11>14>22	4	0.071	0.929	0.071*E(0) + 0.929*E(0)	0
11>14>20>24>23	5	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
11>14>20>24	4	0.08	0.92	0.08*E(0) + 0.92*E(0)	0
E(TD) =					28.83163

ภาคผนวก ข.

ผลการทดลองของเครือข่ายแบบสามเส้า ที่จำนวน โหนดมากกว่า 10 โหนด

ตารางที่ ข.1 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้าขนาด 15 โหนด

เครือข่ายสามเส้าขนาด 15 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>8>7	5	0.526	0.474	$0.526 * E(0) + 0.474 * E(0)$	0
0>8	2	0.474	0.526	$0.474 * E(25) + 0.526 * E(0)$	0.707804
0>8>9	5				
0>8>10	5				
0>8>11	5				
0>8>12	4				
0>8>13	3				
0>8>14	3				
E(TD) =					0.707804
Source Node = 1	Delay	p	q	E(TD _i)	
1>7	2	0.526	0.474	$0.526 * E(27) + 0.474 * E(0)$	0.90754
1>7>8	5	0.474	0.526	$0.474 * E(0) + 0.526 * E(0)$	0
1>7>9	4				
1>7>10	5				
1>7>11	4				
1>7>12	5				
1>7>13	4				
1>7>14	5				
E(TD) =					0.90754

<i>Source Node = 2</i>	Delay	p	q	$E(TD_i)$	
2>8>7	5	0.526	0.474	$0.526 * E(0) + 0.474 * E(0)$	0
2>8	2	0.474	0.526	$0.474 * E(21) + 0.526 * E(0)$	0.509771
2>8>9	4				
2>8>10	4				
2>8>11	4				
2>8>12	3				
2>8>13	3				
2>8>14	3				
E(TD) =					0.509771
<i>Source Node = 3</i>	Delay	p	q	$E(TD_i)$	
3>7	1	0.526	0.474	$0.526 * E(23) + 0.474 * E(0)$	0.671447
3>7>8	4	0.474	0.526	$0.474 * E(0) + 0.526 * E(0)$	0
3>7>9	4				
3>7>10	4				
3>7>11	4				
3>7>12	4				
3>7>13	3				
3>7>14	4				
E(TD) =					0.671447
<i>Source Node = 4</i>	Delay	p	q	$E(TD_i)$	
4>7	3	0.526	0.474	$0.526 * E(31) + 0.474 * E(0)$	1.175269
4>7>8	6	0.474	0.526	$0.474 * E(0) + 0.526 * E(0)$	0
4>7>9	5				
4>7>10	5				
4>7>11	5				
4>7>12	5				
4>7>13	5				
4>7>14	6				
E(TD) =					1.175269

<i>Source Node = 5</i>	Delay	p	q	E(TD _i)	
5>8>7	4	0.526	0.474	$0.526 * E(0) + 0.474 * E(0)$	0
5>8	1	0.474	0.526	$0.474 * E(19) + 0.526 * E(0)$	0.422073
5>8>9	4				
5>8>10	4				
5>8>11	4				
5>8>12	3				
5>8>13	2				
5>8>14	2				
E(TD) =					0.422073
<i>Source Node = 6</i>	Delay	p	q	E(TD _i)	
6>7	3	0.526	0.474	$0.526 * E(33) + 0.474 * E(0)$	1.320574
6>7>8	6	0.474	0.526	$0.474 * E(0) + 0.526 * E(0)$	0
6>7>9	5				
6>7>10	6				
6>7>11	5				
6>7>12	6				
6>7>13	5				
6>7>14	6				
E(TD) =					1.320574

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.2 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้าขนาด 20 โหนด

เครือข่ายสามเส้าขนาด 20 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>11>10	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
0>11	2	0.25	0.75	$0.25 \cdot E(32) + 0.75 \cdot E(18)$	1.195717
0>11>12	3	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
0>11>13	3	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
0>11>15>14	5				
0>11>15	3				
0>11>13>16	5				
0>11>13>17	4				
0>11>18	3				
0>11>19	5				
E(TD) =					1.195717
Source Node = 1	Delay	p	q	E(TD _i)	
1>10	2	0.25	0.75	$0.25 \cdot E(56) + 0.75 \cdot E(0)$	1.656245
1>10>11	5	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
1>10>12	5	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
1>10>13	5	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
1>10>14	4				
1>10>15	5				
1>10>16	5				
1>10>17	5				
1>10>18	5				
1>10>19	4				
E(TD) =					1.656245

<i>Source Node = 2</i>	Delay	p	q	$E(TD_i)$	
2>11>10	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
2>11	2	0.25	0.75	$0.25 * E(32) + 0.75 * E(18)$	1.195717
2>11>12	3	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
2>11>13	3	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
2>11>15>14	5				
2>11>15	3				
2>11>13>16	5				
2>11>13>17	4				
2>11>18	3				
2>11>19	5				
$E(TD) =$					1.195717
<i>Source Node = 3</i>	Delay	p	q	$E(TD_i)$	
3>10	1	0.25	0.75	$0.25 * E(44) + 0.75 * E(0)$	1.068384
3>10>11	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
3>10>12	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
3>10>13	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
3>10>14	3				
3>10>15	4				
3>10>16	4				
3>10>17	4				
3>10>18	4				
3>10>19	3				
$E(TD) =$					1.068384

<i>Source Node = 4</i>	Delay	p	q	$E(TD_i)$	
4>11>10	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
4>11	3	0.25	0.75	$0.25 * E(40) + 0.75 * E(22)$	1.777093
4>11>12	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
4>11>13	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
4>11>15>14	6				
4>11>15	4				
4>11>13>16	6				
4>11>13>17	5				
4>11>18	4				
4>11>19	6				
$E(TD) =$					1.777093
<i>Source Node = 5</i>	Delay	p	q	$E(TD_i)$	
5>11>10	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
5>11	2	0.25	0.75	$0.25 * E(34) + 0.75 * E(18)$	1.266549
5>11>12	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
5>11>13	3	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
5>11>15>14	5				
5>11>15	3				
5>11>13>16	5				
5>11>13>17	4				
5>11>18	4				
5>11>19	5				
$E(TD) =$					1.266549

<i>Source Node = 6</i>	Delay	p	q	E(TD _i)	
6>10	1	0.25	0.75	$0.25 \cdot E(46) + 0.75 \cdot E(0)$	1.158986
6>10>11	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
6>10>12	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
6>10>13	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
6>10>14	3				
6>10>15	4				
5>10>16	4				
6>10>17	4				
6>10>18	4				
6>10>19	4				
E(TD) =					1.158986
<i>Source Node = 7</i>	Delay	p	q	E(TD _i)	
7>11>10	5	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
7>11	3	0.25	0.75	$0.25 \cdot E(40) + 0.75 \cdot E(22)$	1.777093
7>11>12	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
7>11>13	4	0.25	0.75	$0.25 \cdot E(0) + 0.75 \cdot E(0)$	0
7>11>15>14	6				
7>11>15	4				
7>11>13>16	6				
7>11>13>17	5				
7>11>18	4				
7>11>19	6				
E(TD) =					1.777093

<i>Source Node = 8</i>	Delay	p	q	$E(TD_i)$	
8>10	2	0.25	0.75	$0.25 * E(56) + 0.75 * E(0)$	1.656245
8>10>11	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
8>10>12	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
8>10>13	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
4>10>14	4				
8>10>15	5				
8>10>16	5				
8>10>17	5				
8>10>18	5				
8>10>19	4				
$E(TD) =$					1.656245
<i>Source Node = 9</i>	Delay	p	q	$E(TD_i)$	
9>11>10	5	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
9>11	3	0.25	0.75	$0.25 * E(40) + 0.75 * E(22)$	1.777093
9>11>12	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
9>11>13	4	0.25	0.75	$0.25 * E(0) + 0.75 * E(0)$	0
9>11>15>14	6				
9>11>15	4				
9>11>13>16	6				
9>11>13>17	5				
9>11>18	4				
9>11>19	6				
$E(TD) =$					1.777093

ตารางที่ ข.3 Shortest path และ ค่าคาดหวังของ Delay ของเครือข่ายสามเส้นทางขนาด 25 โหนด

เครือข่ายสามเส้นทางขนาด 25 โหนด					
Source Node = 0	Delay	p	q	E(TD _i)	
0>12	2	0.237	0.763	$0.237 * E(64) + 0.763 * E(0)$	1.993793
0>12>13	4	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
0>12>16>14	5	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
0>12>15	4	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
0>12>16	3				
0>12>17	3				
0>12>19>18	4				
0>12>19	3				
0>12>20	3				
0>12>20>21	4				
0>12>22	3				
0>12>23	4				
0>12>24	5				
E(TD) =					1.993793
Source Node = 1	Delay	p	q	E(TD _i)	
1>12	2	0.237	0.763	$0.237 * E(66) + 0.763 * E(0)$	2.105672
1>12>13	4	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
1>12>16>14	5	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
1>12>15	4	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
1>12>16	3				
1>12>17	3				
1>12>19>18	5				
1>12>19	3				
1>12>20	3				
1>12>20>21	4				
1>12>22	3				
1>12>23	4				
1>12>24	5				
E(TD) =					2.105672

Source Node = 2	Delay	p	q	E(TD _i)	
2>14>16>12	5	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
2>14>13	3	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
2>14	2	0.26	0.74	$0.26 * E(42) + 0.74 * E(28)$	2.386963
2>14>15	5	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
2>14>16	3				
2>14>17	3				
2>14>15>18	5				
2>14>19	5				
2>14>20	3				
2>14>21	4				
2>14>22	3				
2>14>15>23	4				
2>14>15>24	5				
E(TD) =					2.386963
Source Node = 3	Delay	p	q	E(TD _i)	
3>15>12	5	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
3>15>14>13	6	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
3>15>14	6	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
3>15	3	0.236	0.764	$0.236 * E(64) + 0.764 * E(22)$	2.882322
3>15>16	5				
3>15>14>17	5				
3>15>18	4				
3>15>19	5				
3>15>14>20	6				
3>15>21	5				
3>15>22	5				
3>15>23	4				
3>15>24	4				
E(TD) =					2.882322

<i>Source Node = 4</i>	Delay	p	q	$E(TD_i)$	
4>12	1	0.237	0.763	$0.237 * E(56) + 0.763 * E(0)$	1.570121
4>12>13	3	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
4>12>16>14	4	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
4>12>15	3	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
4>12>16	3				
4>12>17	3				
4>12>19>18	4				
4>12>19	3				
4>12>20	2				
4>12>20>21	3				
4>12>22	2				
4>12>23	4				
4>12>24	4				
E(TD) =					1.570121
<i>Source Node = 5</i>	Delay	p	q	$E(TD_i)$	
5>13>12	3	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
5>13	2	0.267	0.733	$0.267 * E(64) + 0.733 * E(0)$	2.246171
5>13>14	3	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
5>13>14>15	4	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
5>13>16	3				
5>13>17	3				
5>13>18	3				
5>13>18>19	4				
5>13>20	4				
5>13>21	4				
5>13>22	3				
5>13>23	4				
5>13>24	4				
E(TD) =					2.246171

<i>Source Node = 6</i>	Delay	p	q	E(TD _i)	
6>15>12	5	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
6>15>14>13	6	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
6>15>14	5	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
6>15	3	0.236	0.764	$0.236 * E(58) + 0.764 * E(22)$	2.56223
6>15>16	4				
6>15>14>17	5				
6>15>18	4				
6>15>19	4				
6>15>14>20	6				
6>15>21	4				
6>15>22	5				
6>15>23	4				
6>15>24	4				
E(TD) =					2.56223
<i>Source Node = 7</i>	Delay	p	q	E(TD _i)	
7>14>16>12	5	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
7>14>13	4	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
7>14	2	0.26	0.74	$0.26 * E(46) + 0.74 * E(28)$	2.572179
7>14>15	5	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
7>14>16	4				
7>14>17	3				
7>14>15>18	5				
7>14>19	5				
7>14>20	3				
7>14>21	4				
7>14>22	4				
7>14>15>23	4				
7>14>15>24	5				
E(TD) =					2.572179

Source Node = 8	Delay	p	q	E(TD _i)	
8>15>12	4	0.237	0.763	0.237*E(0) + 0.763*E(0)	0
8>15>14>13	5	0.267	0.733	0.267*E(0) + 0.733*E(0)	0
8>15>14	5	0.26	0.74	0.26*E(0) + 0.74*E(0)	0
8>15	2	0.236	0.764	0.236*E(46) + 0.764*E(18)	1.708383
8>15>16	4				
8>15>14>17	4				
8>15>18	3				
8>15>19	3				
8>15>14>20	5				
8>15>21	3				
8>15>22	4				
8>15>23	3				
8>15>24	3				
E(TD) =					1.708383
Source Node = 9	Delay	p	q	E(TD _i)	
9>14>16>12	4	0.237	0.763	0.237*E(0) + 0.763*E(0)	0
9>14>13	3	0.267	0.733	0.267*E(0) + 0.733*E(0)	0
9>14	1	0.26	0.74	0.26*E(36) + 0.74*E(22)	1.636097
9>14>15	4	0.236	0.764	0.236*E(0) + 0.764*E(0)	0
9>14>16	3				
9>14>17	2				
9>14>15>18	4				
9>14>19	4				
9>14>20	3				
9>14>21	3				
9>14>22	3				
9>14>15>23	3				
9>14>15>24	4				
E(TD) =					1.636097

<i>Source Node = 10</i>	Delay	p	q	E(TD _i)	
10>13>12	3	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
10>13	2	0.267	0.733	$0.267 * E(64) + 0.733 * E(0)$	2.246171
10>13>14	3	0.26	0.74	$0.26 * E(0) + 0.74 * E(0)$	0
10>13>14>15	4	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
10>13>16	3				
10>13>17	3				
10>13>18	3				
10>13>18>19	4				
10>13>20	4				
10>13>21	4				
10>13>22	3				
10>13>23	4				
10>13>24	4				
E(TD) =					2.246171
<i>Source Node = 11</i>	Delay	p	q	E(TD _i)	
11>14>16>12	6	0.237	0.763	$0.237 * E(0) + 0.763 * E(0)$	0
11>14>13	4	0.267	0.733	$0.267 * E(0) + 0.733 * E(0)$	0
11>14	3	0.26	0.74	$0.26 * E(56) + 0.74 * E(34)$	3.68647
11>14>15	6	0.236	0.764	$0.236 * E(0) + 0.764 * E(0)$	0
11>14>16	4				
11>14>17	4				
11>14>15>18	6				
11>14>19	6				
11>14>20	4				
11>14>21	5				
11>14>22	5				
11>14>15>23	5				
11>14>15>24	6				
E(TD) =					3.68647

ประวัติผู้วิจัย

นางสาวฐิติพันธ์ กิจเจริญทรัพย์ เกิดวันที่ 5 พฤษภาคม พ.ศ. 2515 ภูมิลำเนาเดิมอยู่ที่อำเภอเมือง จังหวัด เชียงราย เข้ารับการศึกษาใน โรงเรียนประจำจังหวัดจนจบระดับมัธยมศึกษาปีที่ 6 จากนั้นจึงเข้ามาศึกษาระดับอุดมศึกษาที่ มหาวิทยาลัยรังสิต จนสำเร็จการศึกษาระดับปริญญาตรี เกียรตินิยมอันดับสอง คณะบริหารธุรกิจ สาขา ธุรกิจคอมพิวเตอร์ ปีการศึกษา 2534 และได้ทำงานที่ บริษัทกรุงไทยคอมพิวเตอร์เซอร์วิสจำกัด (KCS) เป็นเวลา 3 ปี จึงได้เข้ามาศึกษาต่อระดับปริญญาโท คณะวิทยาศาสตร์ ภาควิชาคณิตศาสตร์ สาขาวิชาวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีพ.ศ. 2540 จนถึงปัจจุบัน



จุฬาลงกรณ์มหาวิทยาลัย