

การเลือกเส้นทางที่มีต้นทุนต่ำสำหรับโครงข่ายไอพี  
โดยใช้อัลกอริทึมเอสตาร์

MINIMUM COST-PATH FINDING FOR IP NETWORK  
USING A-STAR ALGORITHM

สุริยะ สัตย์ศรี  
SURIVA SATSRI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-230-023

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเลือกเส้นทางที่มีต้นทุนต่ำสำหรับโครงข่ายไอพี  
โดยใช้อัลกอริทึมเอสตาร์

MINIMUM COST - PATH FINDING FOR IP NETWORK  
USING A - STAR ALGORITHM



T107461

สุริยะ สัตย์ศรี

SURIYA SATSRI

เลขหมู่.....  
เลขทะเบียน **107461**  
วันเดือนปี **29 ส.ค. 2553**

b.....
j.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-230-028

**MINIMUM COST - PATH FINDING FOR IP NETWORK  
USING A - STAR ALGORITHM**

**SURIYA SATSRI**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF ENGINEERING IN INFORMATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2009**

**KMITL-2009-EN-M-230-028**

**COPYRIGHT 2009**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**คณะวิศวกรรมศาสตร์**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ใบรับรองวิทยานิพนธ์**

---

**หัวข้อวิทยานิพนธ์** การเลือกเส้นทางที่มีต้นทุนต่ำสำหรับโครงข่ายไอพี โดยใช้อัลกอริทึม เอสตาร์

**Thesis Title** Minimum Cost-Path Finding for IP Network Using A – Star Algorithm

**นักศึกษา** นายสุริยะ สัตย์ศรี

**รหัสประจำตัว** 47061135

**ปริญญา** วิศวกรรมศาสตรมหาบัณฑิต

**สาขาวิชา** วิศวกรรมสารสนเทศ

**อาจารย์ที่ปรึกษาวิทยานิพนธ์** รศ.อรलग แสงอรุณ

**หมายเลขวิทยานิพนธ์** KMITL-2009-EN-M-230-028

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.กนก	เจนจิระพงศ์เวช	
รศ.ดร.ชวลิต	เบญจางคประเสริฐ	
ผศ.ดร.พิทักษ์	ธรรมวาริน	
รศ.นภพินท์	อนันตรศิริชัย	
รศ.อรलग	แสงอรุณ	

วัน/เดือน/ปี ที่สอบ วันศุกร์ที่ 1 พฤษภาคม พ.ศ. 2552 เวลา 10.00-12.00 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 2

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 1 พฤษภาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	การเลือกเส้นทางที่มีต้นทุนต่ำสำหรับโครงข่ายไอพี โดยใช้อัลกอริทึมเอสตาร์
ชื่อนักศึกษา	นายสุริยะ สัตย์ศรี
รหัสประจำตัว	47061135
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ. อรลาภ แสงอรุณ

## บทคัดย่อ

เอสตาร์เป็นอัลกอริทึมที่ดีที่สุด สำหรับการค้นหาเส้นทางเดินข้อมูลที่มีค่าต้นทุนต่ำใน เกมคอมพิวเตอร์ โดยใช้ฟังก์ชันการตัดสินใจ ด้วยค่าฮิวริสติกที่ได้จากการวัดระยะขจัดจากโหนด ปัจจุบันถึงโหนดเริ่มต้น กับค่าต้นทุนที่ประมาณว่าเหมาะสมสำหรับเส้นทางนั้น ในบทวิจัยนี้ ประยุกต์ใช้อัลกอริทึมเอสตาร์ เพื่อหาเส้นทางที่มีต้นทุนต่ำในโครงข่ายไอพี โดยนำผลการพยากรณ์ ความคับคั่งของทราฟฟิกในแต่ละเส้นทางมาเป็นตัวแปรเสริม รวมเป็นค่าต้นทุนร่วมกับค่าระยะทาง และ ค่าการสูญเสียที่เกิดจากการส่งข่าวสารผ่านเส้นใยแก้วนำแสง เพื่อใช้เป็นสมมติฐานสำหรับการกำหนดค่าถ่วงน้ำหนักในการตัดสินใจเลือกเส้นทางเดินที่มีค่าต้นทุนต่ำ ค่าถ่วงน้ำหนักนี้จะมีการเปลี่ยนแปลงตลอดเวลา บนพื้นฐานของความคับคั่งของ ทราฟฟิกในเส้นทางนั้น ๆ

ในส่วนของการพยากรณ์ความคับคั่งของทราฟฟิก ในแต่ละเส้นทางของการสื่อสารข้อมูล ผ่านโครงข่ายไอพี ในบทวิจัยนี้เลือกใช้โมเดลการพยากรณ์ ด้วยอัลกอริทึมของโครงข่ายประสาทเทียม แบบแพร่ย้อนกลับ ข้อมูลที่ใช้ในการศึกษาเป็นการวัดปริมาณทราฟฟิกที่ส่งผ่าน โครงข่ายไอพี ของ บ.ทีโอที จำกัด (มหาชน) ซึ่งประกอบด้วยโหนดหลัก 13 โหนด เชื่อมโยงกันเป็นโครงข่าย แบบไฮแมงมุม 28 เส้นทางเดิน ผลการวิจัยพบว่า การประยุกต์ใช้อัลกอริทึมของ เอสตาร์ ในการหาเส้นทางที่ซับซ้อนของโครงข่ายไอพีดังกล่าว ในบางครั้งของการส่งทราฟฟิกของเราเตอร์ไปตามเส้นทางที่อัลกอริทึมเอสตาร์เลือกนั้น ไม่ได้เป็นเส้นทางที่มีต้นทุนต่ำที่สุดเสมอไป งานวิจัยนี้จึงปรับ ขบวนการในอัลกอริทึมเอสตาร์ เพื่อให้สามารถส่งทราฟฟิกไปในเส้นทางที่มีต้นทุนต่ำในทุกครั้ง ของการส่งขณะที่โครงข่ายไม่มีเหตุขัดข้อง ผลจากการปรับอัลกอริทึมเป็นไปตามวัตถุประสงค์

<b>Thesis Title</b>	Minimum Cost-Path Finding for IP Network Using A - Star Algorithm
<b>Student</b>	Mr. Suriya Satsri
<b>Student ID</b>	47061135
<b>Degree</b>	Master of Engineering
<b>Program</b>	Information Engineering
<b>Year</b>	2009
<b>Thesis Adviser</b>	Assoc. Prof. Ornlarp Sangaroon

### **ABSTRACT**

A-star algorithm is the optimal of graph search algorithm in computer games for finding the least cost-path from a given initial node to one goal node. It uses a distance-plus-cost heuristic function to determine the suitable search of visits nodes in the graph. The distance-plus-cost heuristic is a sum of two functions: the distance is the heuristic estimate distance to the goal and the cost – path function must be an admissible cost from starting node to current node. This research described the minimum cost – path finding for IP network using A-star algorithm. By using the results of IP traffic congestion prediction in each hop with parameter of cost plus – distance and loss in optical transmission line, given heuristic weighting value for finding the least cost-path. These weighting values will be varies as IP traffic congestion in that hop.

Traffic congestion prediction in this thesis using Neural Network with back propagation algorithm is chosen to find out the best model. The data used in the prediction is accumulated at different node in 13 nodes 28 IP main routes of TOT Public company limited in Thailand from 2006 to 2008. The results shown that A-star not guaranteed the minimum cost-path for complexity network with many nodes. Modifications of A-star algorithm are described. The procedure of applications can be adapted under the given circumstance with good solution.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ จัดทำได้สำเร็จลุล่วงไปได้ด้วยดีจากความกรุณาของท่านอาจารย์ที่  
ปรีกษา รศ.อรุณกานต์ แสงอรุณ ผู้ให้ความช่วยเหลือ ชี้แนะ ในการแก้ปัญหาต่าง ๆ ตลอดจนให้ความรู้  
และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ ตลอดจนข้อคิดต่างๆ อัน  
ก่อให้เกิดประโยชน์ต่อการศึกษาค้นคว้า และเป็นแนวทางในการจัดทำวิทยานิพนธ์จนสำเร็จ

ขอขอบคุณ บริษัท ทีโอที จำกัด (มหาชน) และพนักงานที่ให้โอกาสด้านการศึกษา และ  
ข้อมูลในการจัดทำวิทยานิพนธ์

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา  
ซึ่งเป็นที่รัก และเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน

สุริยะ สัตย์ศรี

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	IX
สารบัญรูป.....	X
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 แนวคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตของการศึกษา .....	2
1.6 ขั้นตอนของการศึกษา.....	3
บทที่ 2 โครงข่ายไอพี .....	4
2.1 อุปกรณ์เชื่อมต่อเครือข่าย.....	4
2.1.1 เราเตอร์.....	4
2.1.2 สวิตช์.....	4
2.1.3 ฮับ .....	5
2.1.4 โปรแกรมที่ใช้ในเครือข่าย .....	5
2.2 โครงสร้างตามลำดับชั้น (Hierarchical Design) ของโครงข่าย ไอพี .....	5
2.2.1 ชั้นรวมศูนย์ (Core Layer).....	6
2.2.2 ชั้นกระจาย ศูนย์ (Distribution Layer) .....	7
2.2.3 ชั้นเข้าถึงผู้ใช้ (Access Layer) .....	7
2.3 สถาปัตยกรรมโครงข่ายแบบ TCP/IP .....	7
2.3.1 Encapsulation/Demultiplexing.....	9
2.3.2 รูปแบบระบบ เครือข่าย TCP/IP .....	10

# สารบัญ (ต่อ)

	หน้า
2.3.3 ชั้นโฮสต์-เครือข่าย (Host – to - Network Layer) .....	11
2.3.4 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer) .....	11
2.3.5 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) .....	14
2.3.6 ชั้นสื่อสารการประยุกต์ (Application Layer).....	16
2.3.7 หลักการทำงานของโปรโตคอล TCP/IP.....	16
2.4 การเลือกเส้นทางเดินของข้อมูลและโปรโตคอล.....	18
2.4.1 อัลกอริทึมสำหรับการเลือกทางเดินข้อมูล (Routing Algorithms) .....	19
2.4.2 การเลือกทางเดินแบบสถิตย์ (Static routing Algorithms) .....	19
2.4.3 เร้าตติ้งโปรโตคอล (Routing Protocol).....	22
2.4.3.1 เร้าตติ้งโปรโตคอล แบ่งตามขอบเขตการใช้งาน .....	22
2.4.3.2 เร้าตติ้งโปรโตคอล แบ่งตามพฤติกรรมการทำงาน.....	23
2.4.3.3 เร้าตติ้งโปรโตคอล แบ่งตามความสัมพันธ์กับ Subnet Mask .....	23
2.4.4 ตารางการจัดเส้นทาง (Routing Table) .....	23
2.4.5 การอ่าน Routing Table ของRouter .....	24
2.4.6 วิธีการสร้าง Static Router.....	24
2.4.7 วิธีการทำ Enable Dynamic Route .....	25
2.4.8 Administrative Distance (AD) .....	26
2.4.9 ค่าเชื่อมโยงหรือค่าต้นทุน (Cost) ในการหาเส้นทาง .....	27
2.5 ไอพีทราฟฟิก (IP Traffics).....	27
2.6 ระบบบริหารจัดการ โครงข่าย (Network Management System).....	28
บทที่ 3 การหาเส้นทางที่มีต้นทุนต่ำที่สุด .....	30
3.1 อัลกอริทึมการหาเส้นทางที่เกี่ยวข้อง .....	30
3.1.1 อัลกอริทึมดijkstra (Dijkstra Algorithm) .....	30
3.1.2 เหตุผลความรู้พื้นฐาน (Knowledge Base Reasoning) .....	30
3.1.3 โครงข่ายประสาทเทียม (Neural Network).....	31
3.1.4 อัลกอริทึมจینیติก (Genetic Algorithm) .....	31
3.1.5 อัลกอริทึมฟัซซี่ (Fuzzy Algorithm) .....	31

# สารบัญ (ต่อ)

	หน้า
3.1.6 อัลกอริทึมเวกเตอร์ระยะทาง (Distance-Vector Algorithm) .....	31
3.1.7 อัลกอริทึมสถานการณืเชื่อมโยง (Link-State Algorithm).....	32
3.2 การประยุกต์ใช้อัลกอริทึม เอสตาร์ ในการหาเส้นทางที่มีต้นทุนต่ำ.....	32
3.2.1 หลักการของอัลกอริทึมเอสตาร์.....	33
3.2.2 ตัวอย่าง การพิจารณาอัลกอริทึมเอสตาร์.....	34
3.2.3 การปรับอัลกอริทึมของเอสตาร์ให้เหมาะสมกับโครงข่ายไอพีของ ทีไอที.....	37
<b>บทที่ 4 ทฤษฎีโครงข่ายประสาทเทียมที่เกี่ยวข้องในการวิจัย.....</b>	<b>40</b>
4.1 ทฤษฎีโครงข่ายประสาทเทียมชีวภาพ .....	40
4.1.1 ลักษณะทั่วไปของโครงข่ายประสาทเทียม .....	42
4.1.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม .....	42
4.1.2.1 โครงข่ายชั้นเดียว .....	43
4.1.2.2 โครงข่ายแบบหลายชั้น .....	43
4.2 ฟังก์ชันกระตุ้นความสนใจของโครงข่ายประสาทเทียม.....	44
4.2.1 ฟังก์ชันเทรสโไฮด .....	44
4.2.2 ฟังก์ชันเชิงเส้นอิมตัว .....	44
4.2.3 ฟังก์ชันซิกมอยด์.....	44
4.3 กระบวนการการเรียนรู้ของโครงข่ายประสาทเทียม.....	45
4.3.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning) .....	45
4.3.2 การเรียนรู้แบบไม่มีผู้สอน .....	45
4.4 อัลกอริทึมโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Neural Networks).....	45
4.5 ขั้นตอนการคำนวณ ของอัลกอริทึม โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.....	46
4.6 การประยุกต์ใช้งาน โครงข่ายประสาทเทียม .....	48
4.7 การใช้โครงข่ายประสาทเทียมในการพยากรณ์ .....	49
4.7.1 การเตรียมข้อมูล .....	49
4.7.2 การกำหนดโครงสร้างของตัวแบบที่ใช้พยากรณ์.....	49
4.7.3 การนอร์มอลไลเซชัน (Normalization) .....	50

## สารบัญ (ต่อ)

	หน้า
4.7.4 การฝึกสอนโครงข่ายประสาทเทียม.....	50
4.7.5 การทดสอบเพื่อหาโมเดลที่ดีในการพยากรณ์.....	50
4.7.6 การตรวจสอบความถูกต้องของการพยากรณ์ที่ใช้ในการทดสอบ.....	50
4.7.7 การวิเคราะห์ผล และการนำไปใช้.....	51
<b>บทที่ 5 วิธีดำเนินการวิจัยและผลการวิจัย.....</b>	<b>52</b>
5.1 โครงข่ายและเส้นทางที่ทำการวิจัย.....	52
5.1.1 โครงข่ายที่ใช้ในการวิจัย.....	52
5.1.2 การเก็บข้อมูลกราฟฟิคเพื่อการวิจัย.....	56
5.2 การประมาณค่ากราฟฟิคโดยใช้ตัวแบบโครงข่ายประสาทเทียมในการพยากรณ์.....	61
5.2.1 ผลการกำหนดโครงสร้าง (Topology) ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.....	61
5.2.2 ผลของการพยากรณ์กราฟฟิคตัวแบบที่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.....	62
5.3 การประยุกต์ใช้อัลกอริทึม เอสตาร์ ในการหาเส้นทางที่สั้นที่สุด.....	65
5.3.1 การหาค่าต้นทุนต่ำในแต่ละเส้นทาง.....	65
5.3.2 การหาค่าฮิวริสติก (Heuristic).....	68
5.4 ตัวอย่างการหาเส้นทางด้วยอัลกอริทึมเอสตาร์.....	69
5.5 ผลการทดลองหาเส้นทางด้วยอัลกอริทึมเอสตาร์ กรณีค่าถ่วงน้ำหนักเท่ากัน.....	71
5.5.1 เส้นทางจากเชียงใหม่ – หาดใหญ่.....	71
5.5.2 เส้นทางเชียงใหม่ – ชลบุรี.....	73
5.5.3 เส้นทางเชียงใหม่ – ขอนแก่น.....	74
5.6 ผลของการหาเส้นทางด้วยอัลกอริทึมเอสตาร์ กรณีที่ให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิค.....	75
5.6.1 เส้นทางเชียงใหม่ – หาดใหญ่.....	75
5.6.2 เส้นทางเชียงใหม่ – ชลบุรี.....	76
5.6.3 เส้นทางเชียงใหม่ – ขอนแก่น.....	77

## สารบัญ (ต่อ)

	หน้า
5.7 ผลของการหาเส้นทางด้วยอัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักเท่ากัน.....	78
5.7.1 เส้นทางเชียงใหม่ – หาดใหญ่ .....	78
5.7.2 เส้นทางเชียงใหม่ – ชลบุรี .....	79
5.7.3 เส้นทางเชียงใหม่ – ขอนแก่น .....	80
5.8 ผลของการหาเส้นทางด้วยอัลกอริทึม เอสตาร์ ที่ปรับปรุง โดยให้ค่าถ่วงน้ำหนักมากไปทางทราฟฟิก.....	81
5.8.1 เส้นทางเชียงใหม่ – หาดใหญ่ .....	81
5.8.2 เส้นทางเชียงใหม่ – ชลบุรี .....	82
5.8.3 เส้นทางเชียงใหม่ – ขอนแก่น .....	83
5.9 เปรียบเทียบอัลกอริทึม เอสตาร์ และอัลกอริทึมเอสตาร์ที่ปรับปรุง .....	84
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ .....	88
6.1 บทสรุปผลการวิจัย .....	88
6.2 ข้อเสนอแนะและแนวทางในการพัฒนา.....	88
เอกสารอ้างอิง .....	90
ภาคผนวก.....	92
ผลงานทางวิชาการที่ได้รับการตีพิมพ์.....	93
ประวัติผู้เขียน.....	94

# สารบัญตาราง

ตารางที่	หน้า
2.1 Port Number ในส่วนของ TCP.....	17
2.2 Port Number ในส่วนของ UDP.....	18
2.3 ค่าความน่าเชื่อถือของเราที่ดึง โพร โทคอล.....	26
5.1 โหนดต่าง ๆ และชื่อย่อของโครงข่ายไอพีที่ใช้ในการวิจัย.....	55
5.2 ปริมาณกราฟฟิกของโครงข่ายไอพีที่บันทึก ณ Hop ที่ 1 - 7..	57
5.3 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ Hop ที่ 8 - 14.....	58
5.4 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ Hop ที่ 15 - 21.....	59
5.5 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ Hop ที่ 22 - 28.....	60
5.6 หาจำนวนรอบที่เหมาะสม.....	61
5.7 ค่าโมเมนต์ัม และ ค่าอัตราการเรียนรู้ที่เหมาะสม.....	62
5.8 หาค่าต้นทุนค่าที่ให้ค่าน้ำหนักเท่ากัน.....	67
5.9 หาค่าต้นทุนค่าที่ให้ค่าน้ำหนักมากไปทางกราฟฟิก.....	68
5.10 ค่าฮิวริสติกของแต่ละโหนดในแต่ละเส้นทาง.....	69
5.11 พิจารณาที่โหนดเชียงใหม่.....	69
5.12 พิจารณาที่โหนดอยุธยา.....	70
5.13 พิจารณาที่โหนดหลักสี่.....	70
5.14 พิจารณาที่โหนดพระโขนง.....	70
5.15 พิจารณาที่โหนดลาดหญ้า.....	71

# สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างพื้นฐานของโครงข่ายไอพี.....	6
2.2 ขั้นตอนการ Encapsulation และ Demultiplexing.....	9
2.3 โครงสร้าง TCP/IP.....	10
2.4 IP Packet Contents.....	12
2.5 ICMP Header.....	14
2.6 UDP Header.....	14
2.7 TCP Header.....	15
2.8 การติดต่อสื่อสารระหว่างคอมพิวเตอร์ที่ใช้โปรโตคอล TCP/IP.....	17
2.9 การเปรียบเทียบ Subnet Mask ของ เราเตอร์.....	21
2.10 การใช้คำสั่ง netstat -r และ route.....	21
2.11 เราท์ติ้งโปรโตคอลแบ่งตามขอบเขตการใช้งาน.....	22
2.12 การทำ Static Route.....	25
2.13 การต่อ NMSกับระบบโครงข่าย.....	28
3.1 ความสัมพันธ์ของสมการ อัลกอริทึม เอสตาร์.....	34
3.2 กราฟโครงข่าย.....	34
3.3 กราฟที่มีค่าถ่วงน้ำหนักและค่าฮิวริสติก.....	35
3.4 การหาเส้นทางที่ได้รับการปรับปรุงให้เหมาะกับโครงข่าย.....	38
3.5 ขั้นตอนการทำงานของ อัลกอริทึม เอสตาร์ ที่ปรับสำหรับโครงข่ายไอพีที่ใช้ในการทดลอง.....	39
4.1 โครงสร้างและส่วนประกอบของเซลล์ประสาทชีวภาพ.....	41
4.2 ลักษณะของโหนดภายในโครงสร้างของเซลล์ประสาทเทียม.....	42
4.3 ลักษณะโครงสร้างของโครงข่ายแบบชั้นเดียว.....	43
4.4 แสดงลักษณะโครงสร้างของโครงข่ายแบบหลายชั้น.....	43
4.5 ฟังก์ชันกระตุ้น(Activation Function).....	44
4.6 โครงสร้างของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.....	46
4.7 การคำนวณของอัลกอริทึมโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ.....	47
5.1 โครงข่ายไอพีหลักของ TOT.....	53
5.2 โครงข่ายเมโทรแลน.....	54

## สารบัญญรูป (ต่อ)

รูปที่	หน้า
5.3 ปรับโครงข่ายเพื่อใช้ในการวิจัย .....	54
5.4 โครงข่ายไอพีที่ใช้ในการวิจัย.....	55
5.5 กราฟตัวอย่างปริมาณกราฟฟิกจากการมอนิเตอร์ .....	56
5.6 ปริมาณกราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ Hop ที่ 1 – 7.....	57
5.7 ปริมาณกราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ Hop ที่ 8 – 14.....	58
5.8 ปริมาณกราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ Hop ที่ 15 – 21.....	59
5.9 ปริมาณกราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ Hop ที่ 22 – 28.....	60
5.10 ตัวแบบโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่เหมาะสมเพื่อการพยากรณ์.....	62
5.11 ผลการพยากรณ์กราฟฟิกของฮอปที่ 1 – 7 .....	63
5.12 ผลการพยากรณ์กราฟฟิกของฮอปที่ 8 – 14 .....	63
5.13 ผลการพยากรณ์กราฟฟิกของฮอปที่ 15 – 21 .....	64
5.14 ผลการพยากรณ์กราฟฟิกของฮอปที่ 22 – 28 .....	64
5.15 หาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ให้ค่าถ่วงน้ำหนักเท่ากัน เส้นทางเชียงใหม่ – หาดใหญ่.72	
5.16 หาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ให้ค่าถ่วงน้ำหนักเท่ากัน เส้นทางเชียงใหม่ – ชลบุรี .....73	
5.17 หาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ให้ค่าถ่วงน้ำหนักเท่ากัน เส้นทางเชียงใหม่ – ขอนแก่น.74	
5.18 หาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิกเส้นทาง เชียงใหม่ – หาดใหญ่.....	75
5.19 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิกเส้นทาง เชียงใหม่ – ชลบุรี .....	76
5.20 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิกเส้นทาง เชียงใหม่ – ขอนแก่น.....	77
5.21 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ปรับปรุงกรณีที่ให้ค่าถ่วงน้ำหนักเท่ากันเส้นทาง เชียงใหม่ – หาดใหญ่.....	78
5.22 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ปรับปรุง กรณีที่ให้น้ำหนักเท่ากันเส้นทาง เชียงใหม่ – ชลบุรี .....	79
5.23 หาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักเท่ากันเส้นทาง เชียงใหม่ – ขอนแก่น.....	80

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.24 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักมากไปทาง ทราฟฟิกเส้นทางเชียงใหม่ – หาดใหญ่.....	81
5.25 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักมากไปทาง ทราฟฟิกเส้นทางเชียงใหม่ – ชลบุรี .....	82
5.26 หาเส้นทางด้วย เอ สตาร์อัลกอริทึมที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักมากไปทาง ทราฟฟิกเส้นทางเชียงใหม่ – ขอนแก่น .....	83
5.27 เปรียบเทียบ ค่าต้นทุนของ เอ สตาร์เดิม กับ เอ สตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายไม่มีเหตุขัดข้อง.....	84
5.28 เปรียบเทียบ ค่าต้นทุนของ เอ สตาร์เดิม กับ เอ สตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายเกิดเหตุขัดข้อง 1 เส้นทาง .....	84
5.29 เปรียบเทียบ ค่าต้นทุนของ เอ สตาร์เดิม กับ เอ สตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายเกิดเหตุขัดข้อง 2 เส้นทาง .....	85
5.30 เปรียบเทียบ ค่าต้นทุนของ เอ สตาร์เดิม กับ เอ สตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายเกิดเหตุขัดข้อง 3 เส้นทาง .....	85
5.31 เปรียบเทียบจำนวน Hop ในแต่ละเส้นทางระหว่างการคำนวณด้วยเอสตาร์เดิม กรณีกำหนดค่าถ่วงน้ำหนักเท่ากัน และ กำหนดค่า ถ่วง น้ำหนักมากไปทางทราฟฟิก.....	86
5.32 เปรียบเทียบจำนวน Hop ในแต่ละเส้นทางระหว่างการคำนวณด้วยเอสตาร์ที่ปรับปรุง กรณี กำหนดค่าถ่วงน้ำหนักเท่ากัน และ กำหนดค่า ถ่วง น้ำหนักมากไปทางทราฟฟิก.....	86
5.33 กราฟเปรียบเทียบระหว่าง อัลกอริทึม เอ สตาร์เดิม กับอัลกอริทึม เอ สตาร์ที่ปรับปรุง .....	87

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการสื่อสารข้อมูลทางอินเทอร์เน็ต มีการขยายตัวอย่างรวดเร็ว ส่งผลให้โครงข่ายไอพีซึ่งรองรับการให้บริการอินเทอร์เน็ตต้องขยายตัวตามด้วย ผู้ให้บริการโครงข่ายต่างต้องพยายามใช้งานโครงข่ายที่มีอยู่เดิมให้มีประสิทธิภาพเต็มที่ โดยให้สามารถบริการครอบคลุมพื้นที่มากที่สุด ด้วยปริมาณความจุมากพอที่จะรองรับปริมาณผู้ใช้ที่เพิ่มขึ้นอย่างรวดเร็ว และง่ายต่อการบำรุงรักษา ปัญหาอย่างหนึ่งของโครงข่ายที่ต้องคำนึงถึง คือ เมื่อโครงข่ายมีเหตุขัดข้อง เราเตอร์จะต้องเรียนรู้เส้นทางใหม่จากเราเตอร์ข้างเคียง เพื่อหาเส้นทางที่ดีที่สุดในการส่งผ่านทราฟฟิก ซึ่งในบางครั้งต้องใช้เวลามากในการค้นหา ถ้าโครงข่ายมีขนาดใหญ่ โอกาสที่เราเตอร์อีกตัวหนึ่งในระบบจะเกิดเหตุขัดข้องในเวลาใกล้เคียงกันมีมาก ส่งผลให้เราเตอร์ไม่สามารถส่งผ่านทราฟฟิกไปได้ เหล่านี้ล้วนเป็นสาเหตุสำคัญที่ทำให้คุณภาพของการบริการลดลง วิทยานิพนธ์นี้ จึงนำเสนอการหาเส้นทางสำรองเพื่อให้สามารถส่งผ่านทราฟฟิกไปได้ โดยประยุกต์ใช้อัลกอริทึมเอสตาร์ และปรับขบวนการในอัลกอริทึมดังกล่าว เพื่อใช้งานได้อย่างเหมาะสมกับการหาเส้นทางที่มีต้นทุนต่ำในโครงข่ายไอพีของ บริษัท ทีโอที จำกัด (มหาชน) เพื่อให้ได้เส้นทางที่มีต้นทุนต่ำที่สุด และนำเสนอใช้เป็นเส้นทางสำรองในกรณีเกิดเหตุขัดข้องกับโครงข่าย เพื่อให้โครงข่ายสามารถใช้งานได้อย่างมีประสิทธิภาพ

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อหาโมเดลการพยากรณ์ความคับคั่งของทราฟฟิกของโครงข่ายไอพี โดยใช้วิธีโครงข่ายประสาทเทียม เพื่อประโยชน์ในการบริหารจัดการโครงข่ายในอนาคต
2. เพื่อหาเส้นทางสำรอง ต้นทุนต่ำ กรณีเส้นทางขัดข้อง สำหรับโครงข่ายไอพี ของ บริษัท ทีโอที จำกัด(มหาชน)โดยประยุกต์ใช้ อัลกอริทึมเอสตาร์ (A Star Algorithm)
4. เพื่อเปรียบเทียบองค์ประกอบในการหาต้นทุนอย่างเหมาะสม
5. เพื่อปรับปรุงขบวนการในอัลกอริทึมเอสตาร์ให้เหมาะสมกับโครงข่าย ไอพีที่ซับซ้อน
6. เปรียบเทียบประสิทธิภาพของอัลกอริทึมเอสตาร์ และอัลกอริทึมเอสตาร์ที่ปรับขบวนการในการคำนวณแล้ว

### 1.3 สมมติฐานของการศึกษา

อัลกอริทึมเอสตาร์ เป็นอัลกอริทึมที่นิยมใช้ในการหาเส้นทางของเกมคอมพิวเตอร์[1] โดยวิทยานิพนธ์นี้จะพิจารณาอัลกอริทึมนี้สำหรับการเลือกทางเดินให้กับทราฟฟิกในโครงข่ายไอพีซึ่งเป็นโครงข่ายที่ให้บริการสื่อสารข้อมูล โดยคาดหวังไว้ว่าวิทยานิพนธ์นี้จะสามารถแก้ปัญหาในการหาเส้นทางหลัก และเส้นทางสำรองให้กับโครงข่ายไอพี ซึ่งเป็นโครงข่ายที่ติดตั้งใช้งานจริงโดยที่ระบบจะเลือกให้เป็นโครงข่ายที่มีต้นทุนต่ำเสมอ แม้ว่าเกิดเหตุขัดข้องขึ้นกับเส้นทาง หรือ ฮอป (Hop) ใด ๆ ในโครงข่ายนั้นก็ตาม นอกจากการวิเคราะห์เส้นทางต้นทุนต่ำดังกล่าวแล้ว งานวิจัยนี้ยังเป็นการสำรวจการเพิ่มขึ้นหรือลดลงของปริมาณทราฟฟิกแต่ละฮอปในโครงข่าย เพื่อประโยชน์ในการวางแผนโครงข่ายไอพีที่จะขยายต่อไปในอนาคต ให้สามารถรองรับการให้บริการที่เพิ่มขึ้นได้อย่างมีประสิทธิภาพ

### 1.4 แนวความคิดที่ใช้ในการวิจัย

เนื่องจากผู้ใช้บริการ โครงข่ายมีการใช้งานหลากหลายบริการ และมีปริมาณการใช้งานที่เพิ่มขึ้นเป็นอย่างมาก ผู้ให้บริการโครงข่ายจำเป็นต้องปรับปรุงให้สมรรถนะของโครงข่ายให้เหมาะสม และสามารถให้บริการเพื่อตอบสนองความต้องการของผู้ใช้บริการได้มากที่สุด การหาเส้นทางที่มีต้นทุนต่ำที่สุดในบทยนี้ จึงเริ่มจากการหาแนวโน้มการเปลี่ยนแปลงของการใช้ทราฟฟิกในอนาคดล่วงหน้า 1 ปี โดยการพยากรณ์ด้วยวิธีโครงข่ายประสาทเทียม แล้วนำผลการพยากรณ์มาประกอบการพิจารณาหาเส้นทางที่มีต้นทุนต่ำที่สุดในโครงข่ายหลัก ซึ่งเป็นโครงข่ายที่ใช้งานอยู่ในปัจจุบัน และให้บริการครอบคลุมทั่วประเทศ เพื่อใช้เป็นเส้นทางหลัก และ เส้นทางสำรอง โดยใช้ อัลกอริทึมเอสตาร์ ซึ่งเป็นการนำความรู้ด้านปัญญาประดิษฐ์มาประยุกต์ใช้หาเส้นทางที่มีต้นทุนต่ำในการส่งผ่านทราฟฟิกจากต้นทางไปยังปลายทางของโครงข่าย โดยพิจารณาจากโครงข่ายที่อยู่ในสถานะใช้งานได้ปกติ และสถานะที่เกิดเหตุขัดข้องมากกว่า 1 ฮอป เพื่อให้ได้เส้นทางที่มีต้นทุนต่ำเสมอ แม้ว่าจะเกิดเหตุขัดข้องในโครงข่ายก็ตาม

### 1.5 ขอบเขตของการศึกษา

พิจารณาโครงข่ายหลักที่ทำหน้าที่เป็นชั้นรวมศูนย์ (Core Layer) กระจายอยู่ทั่วประเทศ จำนวน 13 โหนด 28 ฮอป ซึ่งเชื่อมโยงอยู่กับชั้นกระจายศูนย์ (Distribution Layer) และชั้นเข้าถึงผู้ใช้ (Access Layer) ที่ให้บริการครอบคลุมพื้นที่ทั่วประเทศ แล้วพิจารณาการหาเส้นทางที่มีต้นทุนต่ำที่สุด ด้วยอัลกอริทึมเอสตาร์ โดยแต่ละฮอปจะพิจารณาค่าต้นทุนรวมที่ประกอบด้วยพารามิเตอร์เสริมคือ ค่าทราฟฟิกที่ได้จากการพยากรณ์ของอัลกอริทึมโครงข่ายประสาทเทียม ระยะทางระหว่าง

โหนด และค่าการสูญเสียในใยแก้วนำแสง แล้วเปรียบเทียบการคำนวณจากค่าถ่วงน้ำหนัก โดยให้ค่าถ่วงน้ำหนักเท่ากัน และให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิก จากนั้นพิจารณาหาเส้นทางที่มีต้นทุนต่ำที่สุดของแต่ละฮอป โดยยกตัวอย่างข้อมูลจาก 3 ฮอป คือ เชียงใหม่-หาดใหญ่ เชียงใหม่-ชลบุรี และ เชียงใหม่ - ขอนแก่น จากนั้นจำลองการเกิดเหตุขัดข้องมากกว่าหนึ่งครั้ง และปรับขบวนการภายในอัลกอริทึมเอสตาร์ ให้เหมาะสมกับโครงข่ายที่พิจารณา

## 1.6 ขั้นตอนของการศึกษา

ในวิทยานิพนธ์นี้กำหนดขั้นตอนศึกษาดังนี้

1. กำหนดโครงสร้างทางกายภาพของโครงข่ายไอพี
2. กำหนดพารามิเตอร์ที่สำคัญ เช่น จำนวนเราเตอร์ จำนวนเส้นทางระหว่างเราเตอร์ (Hop) และวัดค่าต่าง ๆ ในโครงข่าย เช่น ปริมาณกราฟฟิก ระยะทางระหว่างเราเตอร์ และคำนวณค่าสูญเสียในสายใยแก้วนำแสง (Loss) แต่ละฮอป
3. กระจายข่าวสารบนโครงข่ายไปยังเราเตอร์อื่น ๆ และรับข่าวสารจากเราเตอร์อื่น ๆ โดยทุก ๆ เราเตอร์สามารถรู้โครงสร้าง และสถานะของโครงข่าย
4. ใช้อัลกอริทึมที่เหมาะสมในการหาเส้นทางที่มีต้นทุนต่ำที่สุด ในวิทยานิพนธ์นี้ใช้อัลกอริทึมเอสตาร์ (A Star Algorithm) ในการหาเส้นทาง ซึ่งขั้นตอนของอัลกอริทึมนี้คือจะมีการสร้างกราฟการเชื่อมโยงระหว่างโหนด (Node) ทุก ๆ การเชื่อมโยงจะมีการกำหนดค่าถ่วงน้ำหนักหรือ ค่าต้นทุน (Cost) ซึ่งเป็นฟังก์ชันของ ปริมาณกราฟฟิก ค่าระยะทางระหว่างโหนด และค่าการสูญเสียในการเชื่อมโยง ระหว่างโหนด จากนั้นเราเตอร์จะทำการเลือกเส้นทางที่มีค่าต้นทุนน้อยที่สุดในการส่งกราฟฟิก
5. วิเคราะห์ผล และปรับอัลกอริทึมให้เหมาะสม กับโครงข่าย โดยแยกวิเคราะห์เป็น 2 กรณี คือ กรณีที่ค่าถ่วงน้ำหนักเท่ากัน และกรณีที่ค่าถ่วงน้ำหนักมากทางกราฟฟิก ซึ่งเป็นค่าที่มีการเปลี่ยนแปลงตลอดเวลา และเป็นสาเหตุที่สำคัญทำให้คุณภาพการบริการลดลง ในส่วนของการปรับอัลกอริทึม กำหนดให้มีการเปรียบเทียบต้นทุนก่อนทุกเส้นทาง

## บทที่ 2

# โครงข่าย ไอพี

ในบทนี้ จะกล่าวถึง โครงสร้างพื้นฐานของโครงข่ายไอพี ซึ่งเป็นหัวข้อการวิจัย เนื้อหาประกอบด้วย อุปกรณ์เชื่อมต่อเครือข่าย โครงสร้างตามลำดับชั้น สถาปัตยกรรมโครงข่ายแบบ TCP/IP (Transmission Control Protocol/Internet Protocol) การจัดเส้นทางเดินข้อมูล (Routing) การเก็บข้อมูลทราฟฟิกในโครงข่ายไอพี (IP Traffic) ของบริษัท ทีโอที จำกัด (มหาชน) และสุดท้ายกล่าวถึงการทำงานของระบบบริหารจัดการโครงข่าย ดังรายละเอียดในหัวข้อต่อไป

### 2.1 อุปกรณ์เชื่อมต่อเครือข่าย

อุปกรณ์ที่ใช้ในการเชื่อมต่อโครงข่ายและอุปกรณ์ที่ทำหน้าที่ในการรับส่งข้อมูลระหว่างโครงข่ายมีหลายประเภทด้วยกัน อุปกรณ์แต่ละชนิดมีขีดความสามารถแตกต่างกันออกไป อุปกรณ์ที่นิยมใช้ในการเชื่อมต่อโครงข่ายหลักทั้งเครือข่ายท้องถิ่น (LAN) และเครือข่ายแบบกว้าง (WAN) ประกอบด้วยเราเตอร์ (Router) สวิตช์ (Switch) และ ฮับ (Hub)

#### 2.1.1 เราเตอร์ (Router)

เราเตอร์ทำหน้าที่เป็นอุปกรณ์รับข้อมูลข่าวสารที่ส่งมา โดยมีลักษณะเป็นแพ็กเก็ต (Packet) เพื่อตรวจสอบหาตำแหน่งที่อยู่ (Address) ปลายทาง จากนั้นนำมาเปรียบเทียบกับตารางเส้นทางที่ได้โปรแกรมไว้ เพื่อหาเส้นทางที่จะส่งต่อไป หากเส้นทางที่ส่งมาจากอินเทอร์เน็ต และ จะส่งต่อออกทางพอร์ตทางออก (Port WAN) ของเครือข่าย WAN ที่เป็นแบบจุดต่อจุด ก็จะมีการปรับปรุงรูปแบบสัญญาณให้เข้ากับมาตรฐานใหม่ เพื่อส่งไปยังเครือข่าย WAN อื่น ๆ ได้ ปัจจุบันอุปกรณ์เราเตอร์ได้รับการพัฒนาอย่างกว้างขวาง จึงทำให้สามารถใช้งานเราเตอร์ได้อย่างมีประสิทธิภาพ โดยเฉพาะเมื่อเชื่อมอุปกรณ์เราเตอร์หลาย ๆ ตัวเข้าด้วยกันเป็นโครงข่ายขนาดใหญ่ ประสิทธิภาพของเราเตอร์จะ เพิ่มมากขึ้น โดยสามารถหาเส้นทางเดินข้อมูลที่สั้นที่สุดหรือสามารถเลือกตามความเหมาะสม และแก้ปัญหาที่เกิดขึ้นเองได้

#### 2.1.2 สวิตช์ (Switch)

อุปกรณ์สวิตช์มีหลายแบบ ถ้าแบ่งกลุ่มข้อมูลเป็นแพ็กเก็ต เล็ก ๆ หลายแพ็กเก็ต เรียกว่า เซลล์ (Cell) กลายเป็น เซลล์สวิตช์ (Cell Switch) หรือ ที่รู้จักกันในนามของ เอทีเอ็มสวิตช์ (ATM Switch) และ ถ้าเป็นสวิตช์ข้อมูลในระดับเฟรมของอินเทอร์เน็ตเรียกว่า อินเทอร์เน็ตสวิตช์ (Ethernet Switch) ถ้าเป็นสวิตช์ตามมาตรฐานเฟรมข้อมูลที่เป็กลางที่สามารถนำข้อมูลอื่นมาประกอบภายใน

ได้ก็เรียกว่า เฟรมรีเลย์ (Frame Relay) อุปกรณ์สวิตซ์จึงเป็นอุปกรณ์ที่ใช้เทคโนโลยีใหม่ และมีแนวโน้มที่จะพัฒนาให้ใช้กับความเร็วยของการรับส่งข้อมูลจำนวนมาก เช่น เฟรมรีเลย์ (Frame Relay) และ เอทีเอ็ม สวิตซ์ (ATM Switch) สามารถสวิตซ์ข้อมูลขนาดหลายร้อยล้านบิตต่อวินาทีได้ เทคโนโลยีนี้จึงเป็นเทคโนโลยีที่กำลังได้รับความนิยม

### 2.1.3 ฮับ (Hub)

ฮับเป็นอุปกรณ์ที่ใช้เชื่อมต่อกลุ่มของคอมพิวเตอร์หรือเครือข่ายคอมพิวเตอร์ มีหน้าที่รับส่งเฟรมข้อมูลทุกเฟรมที่ได้รับจากพอร์ตใดพอร์ตหนึ่งไปยังทุก ๆ พอร์ตที่เหลือ คอมพิวเตอร์ที่เชื่อมต่อเข้ากับฮับจะแชร์แบนด์วิดท์หรืออัตราการส่งข้อมูลของโครงข่าย ฉะนั้นยังมีคอมพิวเตอร์มาเชื่อมต่อเข้ากับฮับมากเท่าใดยิ่งทำให้แบนด์วิดท์ต่อคอมพิวเตอร์แต่ละเครื่องลดลง ซึ่งปัจจุบันมีฮับหลายชนิดจากหลายบริษัท ข้อแตกต่างระหว่างฮับเหล่านี้ก็จะเป็นจำนวนพอร์ต สายสัญญาณที่ใช้ ประเภทของเครือข่าย และอัตราข้อมูลที่ฮับสามารถรองรับได้

### 2.1.4 โปรแกรมที่ใช้ในเครือข่าย

โปรแกรมที่ใช้ในเครือข่ายประกอบด้วย โปรโตคอล (Protocol) เพื่อควบคุมขบวนการสื่อสารระหว่างกัน โดยการสื่อสารที่โปรโตคอลกำหนดอาจเป็นได้ทั้งแบบต่อเนื่องหรือแบบมีการเชื่อมต่อ (Connection-Oriented Services) และเครือข่ายแบบไม่มีการเชื่อมต่อ (Connectionless) โดยปกติเครือข่ายส่วนใหญ่สนับสนุนการใช้ร่วมกันของโปรโตคอลได้หลายแบบในลักษณะของชั้นสื่อสาร ซึ่งโปรโตคอลที่ซ้อนกันอยู่จะมีการสื่อสารระหว่างชั้นที่อยู่ติดกันด้วยรูปแบบมาตรฐานที่กำหนดไว้ โปรโตคอลต่างชั้นกันจึงสื่อสารกันได้ ชั้นของโปรโตคอลที่ทำงานซ้อนกันอยู่มักเลียนแบบโครงสร้างมาตรฐาน OSI (Open System Interconnection) หรือ TCP/IP

## 2.2 โครงสร้างตามลำดับชั้น (Hierarchical Design) ของโครงข่าย ไอพี

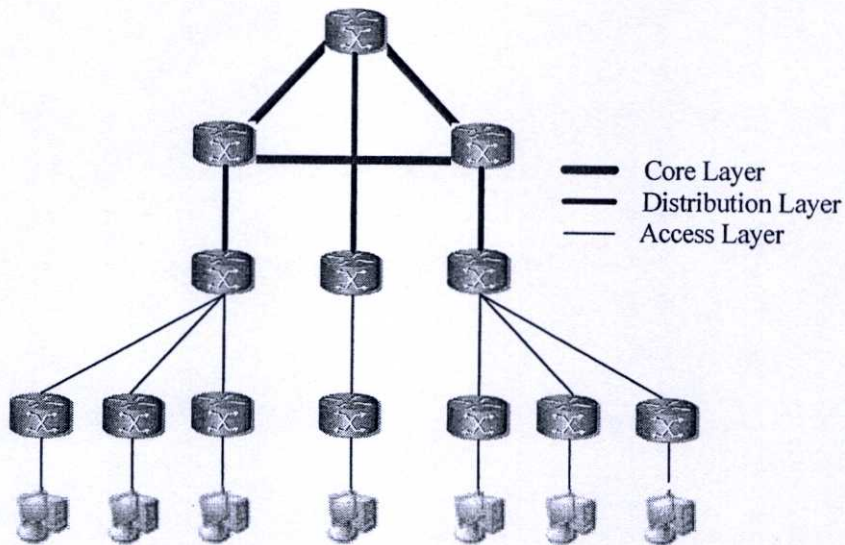
โครงข่ายไอพีเป็นโครงข่ายการสื่อสารข้อมูล เพื่อรองรับปริมาณการใช้งานอินเทอร์เน็ตเป็นหลัก ซึ่งปัจจุบันมีการขยายตัวเพิ่มขึ้นอย่างรวดเร็ว ปัจจุบันนี้ทั่วโลกได้ปรับเปลี่ยนโครงข่ายโทรคมนาคมแบบเดิมเป็นโครงข่ายไอพี เนื่องจากอุปกรณ์โครงข่ายเดิมมีต้นทุนต่อหน่วยสูง และผู้ผลิตหลายรายเลิกผลิตหันมาผลิตอุปกรณ์สำหรับโครงข่ายไอพีแทน เพราะโครงข่ายไอพี ถือเป็นโครงข่ายที่มีประสิทธิภาพสูงในการให้บริการโทรคมนาคม โดยเฉพาะการให้บริการอินเทอร์เน็ตที่สามารถรับส่งข้อมูลภาพ และเสียง ได้สะดวกรวดเร็ว ขณะที่ผู้ประกอบการสามารถลดต้นทุนประหยัดพลังงานที่ใช้ และสะดวกในการบำรุงรักษา

โครงสร้างของระบบประกอบขึ้นจากเครือข่ายคอมพิวเตอร์ขององค์กรต่าง ๆ จำนวนมากที่ได้รับการเชื่อมต่อเข้าด้วยกันเป็นโครงข่ายขนาดใหญ่ที่สามารถรองรับความต้องการ การ

ติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ทุกเครื่องที่อาจอยู่ใกล้หรือไกลกันได้อย่างไร้ขอบเขต และมีประสิทธิภาพสูง โครงข่ายไอพีโดยทั่วไปถูกออกแบบในลักษณะเป็นโครงสร้างตามลำดับชั้น (Hierarchical Design) เพื่อให้ง่ายในการจัดการโครงข่าย การเลือกใช้อุปกรณ์เครือข่าย และการติดตั้งเพิ่มเติมในอนาคต โดยทั่วไปแล้วจำนวนของลำดับชั้น และหน้าที่ของแต่ละลำดับชั้นในแต่ละเครือข่ายนั้น อาจมีจำนวนชั้น และหน้าที่ที่แตกต่างกันไป แต่ที่สำคัญก็คือลำดับชั้นล่างจะให้บริการลำดับชั้นที่สูงกว่า สำหรับโครงข่ายไอพีของ บริษัทที่ไอที จำกัด (มหาชน) ที่จะกล่าวถึงต่อไปนี้ โครงสร้างตามลำดับชั้นของระบบ ประกอบไปด้วยชั้นที่เรียกว่าชั้นรวมศูนย์ (Core Layer) ชั้นกระจายศูนย์ (Distribution Layer) และชั้นเข้าถึงผู้ใช้ (Access Layer) ตามลำดับ [2] ดังแสดงในรูปที่ 2.1

### 2.2.1 ชั้นรวมศูนย์ (Core Layer)

Core Layer เป็นจุดศูนย์กลาง และเป็นหัวใจหลักของโครงข่าย โดยปกติจะมีเครื่องคอมพิวเตอร์ขนาดใหญ่ มีเครือข่ายเชื่อมโยงจำนวนมาก ทำหน้าที่เชื่อมต่อกับลำดับชั้นรองลงมาที่เรียกว่า ชั้นกระจาย ศูนย์ (Distribution Layer) หลาย ๆ จุดเข้าไว้ด้วยกัน สำหรับชั้นนี้ต้องสามารถรับส่ง แพ็กเก็ตของข้อมูล ข่าวสารต่าง ๆ ได้อย่างรวดเร็วมาก แต่อย่างไรก็ดี ในโครงข่ายบางโครงข่าย อุปกรณ์ที่ทำงานในส่วนของ Core Layer นี้ ก็กับ Distribution Layer อาจเป็นตัวเดียวกันก็ได้ คือ มีอุปกรณ์เราเตอร์ (Router) ตัวหลักตัวหนึ่ง ทำหน้าที่เป็น Core Router และมี เราเตอร์ปลายทางหลาย ๆ ตัวทำหน้าที่เป็น Access Router



รูปที่ 2.1 โครงสร้างพื้นฐานของโครงข่ายไอพี

## 2.2.2 ชั้นกระจาย ศูนย์ (Distribution Layer )

Distribution Layer เป็นชั้นที่รองรับการเชื่อมต่อจาก Access Layer หลายๆ จุดเข้าด้วยกัน และผ่านต่อไปยัง Core Layer สำหรับ อุปกรณ์ที่ทำงานอยู่ในชั้นนี้มักเป็นเราเตอร์ที่มีประสิทธิภาพสูง และมีจำนวนพอร์ตมากพอสำหรับรองรับการเชื่อมต่อไปยังเราเตอร์ที่ทำงานอยู่ในชั้นเข้าถึงผู้ใช้ (Access Layer) เราเตอร์ในชั้นนี้ถือได้ว่าเป็นเสมือนจุดศูนย์กลางรวม ของเราเตอร์ต่าง ๆ ที่อยู่ในชั้น Access Layer เพื่อให้ผู้ใช้ที่เชื่อมต่อกับเราเตอร์ใน Access Layer ได้ และภายในเราเตอร์ที่อยู่ใน ชั้น Distribution Layer นี้ควรมีการใช้กับอุปกรณ์สนับสนุน (Implement Feature) เช่น Inter VLAN Routing, Access Control Lists (ACL) รวมถึงคุณภาพของการบริการ (quality of service: Qos) และ วิถีทาง (Policy) ต่าง ๆ ในการใช้งานของโครงข่ายด้วย

## 2.2.3 ชั้นเข้าถึงผู้ใช้ (Access Layer)

Access Layer จะเป็นชั้นที่ใกล้ชิดกับผู้ใช้งานมากที่สุด เป็นจุดที่นำเครื่องคอมพิวเตอร์ของ ผู้ใช้เข้าสู่ระบบเครือข่าย สำหรับ LAN และ Campus Network และ อุปกรณ์โครงข่ายที่ทำงานอยู่ใน ชั้นนี้มักเป็นเราเตอร์ ตัวเล็ก ๆ ที่มีจำนวนพอร์ตที่พอเพียงต่อการรองรับการเชื่อมต่อจากเครื่อง คอมพิวเตอร์ของผู้ใช้ผ่านทางสายเคเบิล เช่น UTP เป็นต้น เราเตอร์ ดังกล่าวนี้อาจจำเป็นต้องมีพอร์ต Up Link เพื่อเชื่อมโยงขึ้นไปยังเราเตอร์ที่อยู่ใน Distribution Layer หรือระดับ Core Layer ทั้งนี้ ขึ้นอยู่กับการออกแบบ ความเหมาะสมของอุปกรณ์ที่ทำงานในชั้นนี้คือ ควรมีต้นทุนของอุปกรณ์ที่ ต่ำโดย ยังไม่จำเป็นต้องมีอุปกรณ์สนับสนุนการใช้งานขั้นสูงมากนัก และที่สำคัญควรติดตั้งได้ง่าย

จากโครงสร้างคร่าว ๆ ของระบบ จะเห็นว่าภาพรวมของระบบมีความคล้ายคลึงกับระบบ โทรศัพทที่ใช้กันอยู่ทั่วไป โดยผู้ใช้แต่ละคนมีความเป็นอิสระในการส่งข่าวสารไปหาผู้อื่นได้ แต่ต้องทราบหมายเลขของผู้ใช้รายนั้นก่อนเสมอ อาจกล่าวได้ว่าโครงข่ายไอพีนี้ มีไว้เพื่อเข้าสู่ ระบบอินเทอร์เน็ตซึ่งมีรูปแบบการใช้งานทั่วไปดังนี้

1. การติดต่อข้อมูลข่าวสารระยะไกล
2. การสื่อสารระหว่างบุคคลต่อบุคคล
3. ความบันเทิงส่วนบุคคล
4. การพาณิชย์อิเล็กทรอนิกส์ และห้องสมุดอิเล็กทรอนิกส์

## 2.3 สถาปัตยกรรมโครงข่ายแบบ TCP/IP

ในหัวข้อนี้จะกล่าวถึงสถาปัตยกรรมโครงข่ายแบบ TCP/IP ซึ่งใช้ในระบบอินเทอร์เน็ตโดย TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นมาตรฐานของการรับส่งข้อมูล ระหว่างคอมพิวเตอร์ ซึ่งเริ่มต้นทดลองในปี ค.ศ. 1969 โดยกระทรวงกลาโหมสหรัฐฯ หรือ Department Of Defense (DOD) เพื่อเชื่อมโยงคอมพิวเตอร์ทางทหารของแต่ละหน่วย ซึ่งเป็น

คอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกันได้ (File Transfer) และสามารถใช้บริการอื่น ๆ ได้อีก เช่นการรับส่งจดหมายอิเล็กทรอนิกส์ (E-Mail) เป็นต้น จุดประสงค์ของโครงการนี้เพื่อสร้างระบบเครือข่ายคอมพิวเตอร์ให้สามารถส่งข้อมูลกันได้ แม้ว่าสายส่งข้อมูลบางส่วนหรือคอมพิวเตอร์บางเครื่องในเครือข่ายจะถูกทำลายเสียหายไปก็ตาม โครงการนี้มีชื่อว่า ARPANET (Advance Research Projects Agency Network) นับเป็นโครงการที่ประสบความสำเร็จอย่างสูงจนใช้งานได้อย่างจริงจังในปี ค.ศ.1975

ในปี ค.ศ. 1983 กฎการสื่อสารมาตรฐาน TCP/IP (TCP/IP Protocol) ถูกกำหนดให้เป็นมาตรฐานการรับส่งข้อมูลของกระทรวงกลาโหมสหรัฐฯ และได้ถูกรวมให้เป็นส่วนหนึ่งของระบบปฏิบัติการ UNIX โดยได้รับทุนสนับสนุนจากกระทรวงกลาโหมสหรัฐฯเช่นเดียวกัน ทำให้ TCP/IP ก็ยังมีบทบาทในการสื่อสารมากขึ้นเรื่อย ๆ โดยมีวัตถุประสงค์เพื่อให้สามารถสื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้อย่างอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นเพื่อสามารถส่งผ่านข้อมูลไปให้ถึงปลายทางได้ ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวาง จนกระทั่งกลายเป็นระบบอินเทอร์เน็ตในปัจจุบัน โดยมีจุดประสงค์ของการสื่อสารตามมาตรฐาน สามประการคือ

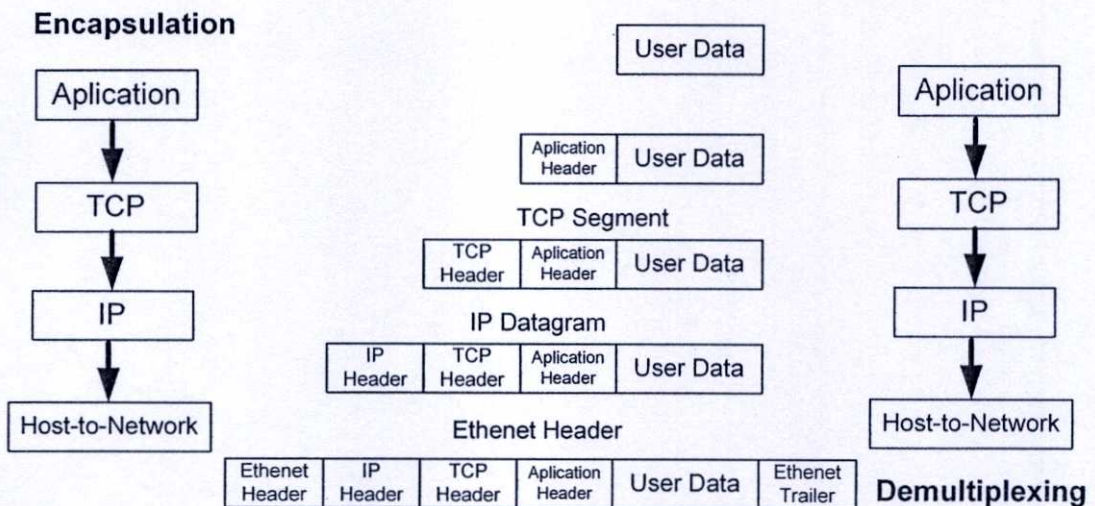
1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วย รับ-ส่ง เกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ Real-Time การสื่อสารในรูปแบบของเสียง (Voice) และข้อมูล (Data)

ซอฟต์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลของ ARPANET ประกอบด้วยส่วน 2 ส่วนหลัก ๆ คือ Transmission Control Protocol (TCP) และ Internet Protocol (IP) [3] ซึ่ง TCP มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับ และผู้ส่ง ให้ได้รับข้อมูลถูกต้องครบถ้วน หากข้อมูลเกิดการสูญหายก็จะแจ้งให้ต้นทางส่งข้อมูลมาใหม่ ส่วน IP จะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบตำแหน่ง (Address) ของผู้รับโดยใช้ข้อมูลขนาด 4 ไบต์ นอกจากนี้ยังมีบทบาทสำคัญในการทำงานในเครือข่ายอินเทอร์เน็ต เนื่องจากเมื่อโปรโตคอลอื่น ๆ ต้องการส่งผ่านข้อมูลข้ามเครือข่ายในอินเทอร์เน็ตนั้น จะต้องอาศัยการผนึกข้อมูล (Encapsulation) ไปด้วย IP ที่มีกลไกการระบุเส้นทางบริการ (Route Service) ผ่าน Gateway หรือ Router เพื่อนำข้อมูลไปยังโครงข่าย และเครื่องปลายทางที่ต้องการ เนื่องจากกลไกการระบุเส้นทางจะทำงานโดย

IP เท่านั้น จึงเรียก IP ว่าเป็นโปรโตคอลที่มีความสามารถในการระบุเส้นทางการส่งผ่านข้อมูลได้ (Routable)

### 2.3.1 Encapsulation/Demultiplexing

การส่งข้อมูลผ่านในแต่ชั้นสื่อสาร แต่ละชั้นสื่อสารจะทำการประกอบข้อมูลที่ได้รับมาเข้ากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลที่เรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือ โปรโตคอลเดียวกัน ทางด้านผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลผลทำให้ทราบว่าข้อมูลที่ตามมามีลักษณะเป็นอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



รูปที่ 2.2 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละชั้นสื่อสารมีชื่อเรียกแตกต่างกัน ดังนี้

1. User Data : ข้อมูลที่มาจากผู้ใช้ (User) หรือข้อมูลที่ผู้ใช้เป็นผู้ป้อนให้กับ Application เรียกว่าข้อมูลผู้ใช้ (User Data)
2. Application Data : เมื่อ Application ได้รับข้อมูลจาก User ก็จะนำมาประกอบกับส่วนหัวของ Application เรียกว่า Application Data และส่งต่อไปยัง โปรโตคอล TCP

3. TCP Segment : เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP

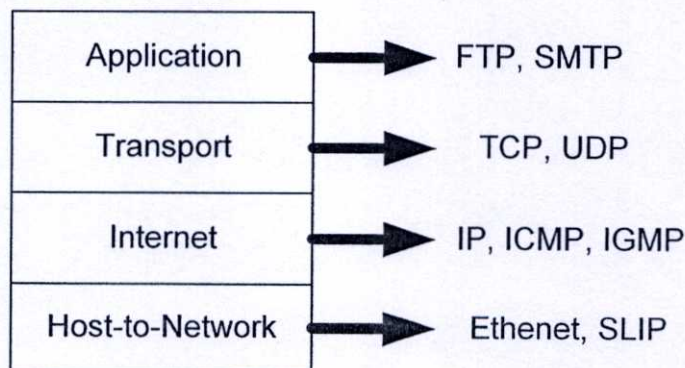
4. IP Datagram : เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปชั้น Host-to-Network Layer

5. Ethernet Frame : ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ Flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมต่อต่อไป

ตัวอย่างของกลุ่มโปรโตคอลในชุดของ TCP/IP ที่พบ และใช้งานบ่อย ๆ นั้น ส่วนใหญ่จะไม่ใช้งานโดยตรง แต่ใช้งานผ่านแอปพลิเคชันต่าง ๆ เช่น Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), Transport Control Protocol (TCP), Simple Mail Transfer Protocol (SMTP) และ Domain Name System (DNS) เป็นต้น [15]

### 2.3.2 รูปแบบระบบ เครือข่าย TCP/IP

โครงสร้างแบบ ทีซีพี/ไอพี (TCP/IP Model) เป็นมาตรฐานที่ทำให้คอมพิวเตอร์ภายในระบบเครือข่ายสามารถเชื่อมต่อเข้าหากัน และ สามารถแลกเปลี่ยนข้อมูลกันได้ โดยใช้แนวคิดของการแบ่งชั้นสื่อสารโปรโตคอล สถาปัตยกรรมระบบเครือข่าย TCP/IP มีชั้นสื่อสารที่น้อยกว่าโครงสร้างของชั้นสื่อสารระบบมาตรฐาน OSI โดยระบบมาตรฐาน OSI มีลำดับชั้นสื่อสารของโปรโตคอลทั้งหมด 7 ชั้น แต่ในระบบ TCP/IP มีทั้งหมดเพียง 4 ชั้นเท่านั้น ในแต่ละชั้นสื่อสารของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่ 2.3 โครงสร้าง TCP/IP

### 2.3.2.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นสื่อสารนี้ เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสารอินเทอร์เน็ต (IP) มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูล ทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสารสามารถเทียบได้กับชั้นที่ 1 และ 2 ในโครงสร้างมาตรฐาน OSI เป็นชั้นสื่อสารที่ไม่ได้เกี่ยวข้องกับระบบอินเทอร์เน็ตโดยตรง แต่เป็นระบบพื้นฐานของการเชื่อมต่อที่ระบบอินเทอร์เน็ตใช้ส่งข้อมูลภายในเครือข่าย หน้าที่ของชั้นนี้ใช้สำหรับการส่งข้อมูล เนื่องจากตัวแบบ TCP/IP ไม่ได้กำหนดมาตรฐานในชั้นนี้ โดยกำหนดไว้เพียงว่าให้สามารถส่งข้อมูลสู่เครือข่ายได้เท่านั้น ทำให้ไม่สามารถระบุเนื้อหา หน้าที่ที่ชัดเจนได้ ดังนั้นจึงอาจจะยกระบบของโครงสร้างมาตรฐาน OSI ทั้งสองชั้นแรกมา ซึ่งได้แก่การจัดเตรียมข้อมูลเพื่อให้เหมาะสมก่อนที่จะส่งไปตามสายส่งไปยังที่หมายปลายทาง ได้แก่ การจัดเตรียม Packet Header การควบคุมระบบฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการจัดส่ง เช่น การเชื่อมต่อกับ Network card และการใช้งาน Device Driver หน้าที่สำหรับการรับข้อมูลคือ คอยรับกรอบของข้อมูล แล้วนำข้อมูลส่วนหัวออกมา และจัดเตรียมข้อมูลเพื่อส่งให้กับชั้นต่อไป

### 2.3.2.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ชั้นสื่อสารอินเทอร์เน็ตนี้ เทียบได้กับชั้นสื่อสารชั้นที่ 3 (Network Layer) ในโครงสร้างแบบมาตรฐาน OSI เป็นชั้นสื่อสารที่มีหน้าที่ส่งข้อมูลจากจุดเริ่มต้นไปยังปลายทาง โดยหาเส้นทางที่ข้อมูลจะใช้เดินทางผ่านเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่งจนกระทั่งถึงปลายทาง โปรโตคอลที่ใช้ในชั้นนี้คือ อินเทอร์เน็ตโปรโตคอล หรือ ไอพี ซึ่งทำหน้าที่เปรียบเสมือนของจดหมายซึ่งระบุที่อยู่ของต้นทาง และปลายทาง และมีบุรุษไปรษณีย์ทำหน้าที่ส่งจดหมายนั้นผ่านกรมการไปรษณีย์ในพื้นที่ต่าง ๆ จนถึงจุดหมายปลายทาง ที่อยู่บนซองจดหมายในอินเทอร์เน็ตโปรโตคอลเรียกว่า หมายเลขไอพี ที่ทำการไปรษณีย์คือ เราเตอร์ ซึ่งมีหน้าที่ค้นหาเส้นทางที่เหมาะสมเพื่อส่งข้อมูลไปตามสายส่งจนกระทั่งถึงปลายทาง ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับ แพ็กเก็ต (Packet-Switching Network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่าง ๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ ถ้ามีการส่งแพ็กเก็ตออกมาเป็นชุด โดยมีจุดหมายปลายทางเดียวกัน ในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะไปอิสระแก่กันและกัน ดังนั้น แพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่ปฏิบัติตามลำดับก็ได้

ไอพีแพ็กเก็ต หรือ Datagram ของชั้นสื่อสารชั้นที่ 3 จะเป็นข้อมูลสำหรับชั้นสื่อสารชั้นที่ 2 ซึ่งจะถูกรหัสเข้าไว้เป็นเฟรมดังที่ได้อธิบายไว้ก่อนหน้านี้ ในทำนองเดียวกัน ไอพีแพ็กเก็ตดังแสดงในรูปที่ 2.5 ประกอบด้วยข้อมูลจากชั้นสื่อสารชั้นบนรวมกับ IP header ซึ่งมีรายละเอียดดังนี้

1. Version (VERS) เป็นข้อมูลขนาด 4 บิตซึ่งเป็นตัวบอกลำดับเลขรุ่นของโปรโตคอล IP ที่กำลังใช้งานอยู่อุปกรณ์ทั้งหมดจะต้องใช้ IP ที่เป็นรุ่นเดียวกัน มิฉะนั้นอุปกรณ์ที่ใช้โปรโตคอล IP รุ่นที่แตกต่างออกไปจะปฏิเสธแพ็กเก็ตนั้น

0	4	8	16	19	24	31
---	---	---	----	----	----	----

VERS	HLEN	Service Type	Total Length		
Identification			Flags	Fragment Offset	
Time-To-Live		Protocol	Header Checksum		
Source IP Address					
Destination IP Address					
IP Options (If Any)					Padding
Data					
...					

รูปที่ 2.4 IP Packet Contents

2. IP Header Length (HLEN) เป็นข้อมูลขนาด 4 บิตที่ระบุความยาวของ Datagram Header ที่มีหน่วยเป็น 32-Bit words ตัวเลขนี้คือความยาวทั้งหมดของข้อมูล Header

3. Type of Service (TOS) เป็นข้อมูลขนาด 8 บิตที่กำหนดระดับความสำคัญที่ได้กำหนดให้โดยโปรโตคอลในชั้นสื่อสารระดับบน

Total Length-เขตข้อมูลขนาด 16 บิตที่กำหนดความยาวทั้งหมดของแพ็กเก็ต (รวมทั้ง header) มีหน่วยเป็นไบต์เมื่อนำค่า HLEN มาลบออกจาก Total Length ก็จะได้ขนาดของข้อมูลในแพ็กเก็ตนั้น

4. Identification เป็นข้อมูลขนาด 16 บิต ที่บรรจุด้วยเลขจำนวนเต็มที่ใช้ในการระบุตัวตนของ Datagram หมายเลขนี้เป็นหมายเลขลำดับ (Sequence Number)

5. Flag เป็นข้อมูลขนาด 3 บิต ซึ่งเลข 2 บิตล่างจะควบคุมการทำ Fragmentation บิตแรกจะกำหนดว่าแพ็กเก็ตนั้นสามารถถูกแบ่งออกเป็น Fragment ขนาดเล็กได้หรือไม่ บิตที่ 2 จะกำหนดว่าแพ็กเก็ตนั้นคือ Fragment ส่วนสุดท้ายของทั้งหมดหรือไม่

6. Fragment Offset เป็นข้อมูลขนาด 13 บิต ที่นำมาใช้ในการประกอบ Datagram Fragment ต่าง ๆ เข้าด้วยกัน

7. Time to Live (TTL) เป็นข้อมูลขนาด 8 บิต ที่เก็บรักษาตัวนับจำนวนที่ซึ่งจะลดค่าลงเรื่อยๆ จนมีค่าเป็น "0" จากนั้นข้อมูลใน Datagram จะถูกลบทิ้งไป ซึ่งจะเป็นการป้องกันไม่ให้เกิดการเกิดคั่งค้างอยู่ในระบบเครือข่ายไม่รู้จบ ตัวอย่างเช่น Hop count

8. Protocol เขตข้อมูลขนาด 8 บิตที่ระบุโปรโตคอลในชั้นสื่อสารระดับบนที่จะรับแพ็กเก็ตข้อมูลไปภายหลังจากที่การประมวลผลโดยใช้โปรโตคอล IP เสร็จสิ้นแล้ว ตัวอย่างเช่น TCP หรือ UDP

9. Header Checksum เป็นข้อมูลขนาด 16 บิต ที่ช่วยให้โปรโตคอล IP ตรวจสอบความถูกต้องของข้อมูล

10. Source IP Address เป็นข้อมูลขนาด 32 บิต ที่ระบุหมายเลข IP Address ของผู้ส่งข้อมูล

11. Destination IP Address เป็นข้อมูลขนาด 32 บิต ที่ระบุหมายเลข IP Address ของผู้รับข้อมูล

12. Options เป็นข้อมูลที่มีความยาวไม่คงที่ที่ช่วยโปรโตคอล IP สนับสนุนทางเลือกในการปฏิบัติงานต่างๆ เช่นการรักษาความปลอดภัย

13. Padding ตัวเลข "0" จำนวนหนึ่งที่ถูกเพิ่มเติมเข้ามาในแพ็กเก็ต เพื่อให้แน่ใจว่า IP header นั้น มีขนาดเป็นจำนวนเท่าของ 32 บิตเสมอ

14. Data ส่งข้อมูลที่มีความยาวไม่คงที่ซึ่งมีขนาดสูงสุด 64 กิโลไบต์ ที่บรรจุข้อมูลสำหรับชั้นสื่อสารในระดับบน

ICMP (Internet Control Message Protocol) เป็นโปรโตคอลที่ใช้ในการตรวจสอบ และรายงานสถานภาพของค้ำแกรม (Datagram) ในกรณีที่เกิดปัญหาเกี่ยวกับค้ำแกรม เช่น เราเตอร์ไม่สามารถส่งค้ำแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต์ต้นทาง เพื่อรายงานข้อผิดพลาดที่เกิดขึ้น แต่อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่ เมื่อมีการส่งค้ำแกรมออกไปแล้วไม่มี ICMP Message ฟ้องข้อผิดพลาดกลับมา แสดงความหมายได้สองกรณี คือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหาในการสื่อสารซึ่งอาจเป็นไปได้ทั้งการส่งค้ำแกรม และ ICMP Message ที่ส่งกลับมา แล้วเกิดมีปัญหาระหว่างทาง ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Unreliable) ซึ่งจะเป็นหน้าที่ของโปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้น ๆ มีความน่าเชื่อถือ ในส่วนของ ICMP Message จะประกอบด้วย บิต Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม ขนาดบิต Type และ Code ดังรูป 2.5

8-bit Type	8-bit Code	16-bit Checksum
ICMP Content		

รูปที่ 2.5 ICMP Header

### 2.3.2.3 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

ชั้นสื่อสารนำส่งข้อมูลเป็นชั้นสื่อสารที่อยู่เหนือชั้น IP เทียบได้กับชั้นที่ 4 ในโครงสร้างมาตรฐานแบบ OSI เป็นชั้นที่มีหน้าที่ควบคุมการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองเครื่องที่ติดต่อกัน ซึ่งอาจแบ่งได้สองลักษณะคือ บริการส่งข้อมูลแบบมีการเชื่อมต่อ และบริการส่งข้อมูลแบบไม่มีการเชื่อมต่อ และ จัดส่งข้อมูลไปยัง Application ที่ต้องการข้อมูล โปรโตคอลที่นิยมใช้ในชั้นนี้ได้แก่ UDP (User Datagram Protocol) และ TCP (Transmission Control Protocol)

UDP (User Datagram Protocol) เป็นโปรโตคอลที่อยู่ในชั้นสื่อสารนำส่งข้อมูล Transport เมื่อเทียบกับโมเดลมาตรฐาน OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่าง Datagram และไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล กลไกการตรวจสอบโดยการ Checksum ของ UDP นั้นเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะรู้ว่ามีความผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึงหรือเวลาผิดพลาด ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด โดยมีรายละเอียดดังนี้

16-bit Source	16-bit Destination
ICMP Content	Checksum
Data	

รูปที่ 2.6 UDP Header

1. Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งค่าตัวแกรมนี้
2. Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับค่าตัวแกรม
3. UDP Length : ความยาวของค่าตัวแกรม ทั้งส่วน Header และ Data นั้นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
4. Checksum : เป็นตัวตรวจสอบความถูกต้องของ UDP Datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

TCP (Transmission Control Protocol) อยู่ในชั้นสื่อสารนำส่งข้อมูล เช่นเดียวกัน โดยทำหน้าที่จัดการ และควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถ และรายละเอียดมากกว่า UDP โดยค่าตัวแกรมของ TCP จะมีความสัมพันธ์ต่อกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นทางการ (Connection - Oriented) มีรายละเอียดดังนี้

16-bit Source Port Number				16-bit Source Destination Port				
32-bit Sequence Number								
32-bit Acknowledge Number								
Header Length	6-bit Reserved	URG	ACK	PUSH	RESET	SYN	FIN	16-bit Windows Size
16-bit TCP Checksum				16-bit Urgent Pointer				
TCP Option								
Data								

รูปที่ 2.7 TCP Header

1. Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งค่าตัวแกรมนี้
2. Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับค่าตัวแกรม
3. Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
4. Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
5. Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์

Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้น ๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP Segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน Flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment Number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใด ๆ ซึ่งถ้าไม่มี Flag เป็นตัวกำหนด อาจจะมีการนำข้อมูลมาใช้และก่อให้เกิดความผิดพลาดได้

#### 2.3.2.4. ชั้นสื่อสารการประยุกต์ (Application Layer)

ชั้นสื่อสารการประยุกต์ (Application layer) เทียบได้กับชั้นที่ 5 ถึง 7 ในโครงสร้างแบบมาตรฐาน OSI ซึ่งจะครอบคลุมบริการที่เกี่ยวข้องกับการรักษาความปลอดภัย การเข้ารหัส การเชื่อมต่อระหว่างโปรแกรมประยุกต์ และเป็นชั้นที่โปรแกรมประยุกต์ใช้งานโดยตรง โดยโปรโตคอลที่อยู่บนชั้นนี้จะถูกออกแบบให้เหมาะสมสำหรับประเภทของโปรแกรมประยุกต์เฉพาะทาง เช่น โปรแกรมอีเมลใช้โปรโตคอล SMTP (Simple Mail Transfer Protocol) ในการส่งอีเมล ใช้โปรโตคอล POP3 (Post Office Protocol) ในการรับ และเรียกดูอีเมล ส่วนโปรแกรมเว็บเบราว์เซอร์ใช้โปรโตคอล HTTP (Hypertext Transfer Protocol) ในการเรียกดูเว็บเพจ เป็นต้น

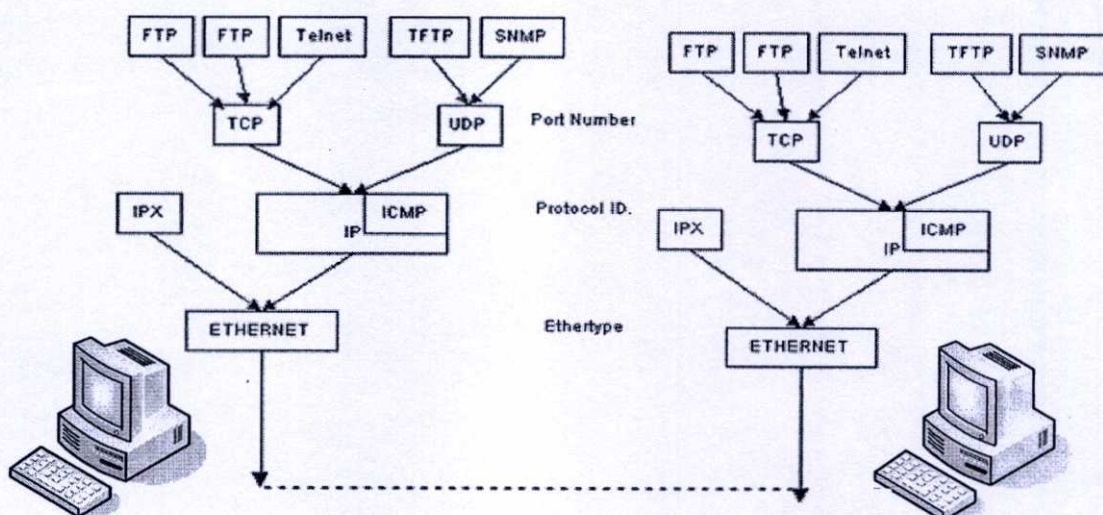
ชั้นสื่อสารการประยุกต์ มีโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โปรโตคอลสำหรับการจัดการเพิ่มข้อมูล เรียกว่า FTP (File Transfer Protocol) และโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP โดยโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่อง Host ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่อง Host นั้น โปรโตคอลสำหรับการจัดการเพิ่มข้อมูลช่วยในการคัดลอกเพิ่มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาเพิ่มข้อมูลไปยังเครื่องใด ๆ ก็ได้ โปรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

#### 2.3.2.5 หลักการทำงานของโปรโตคอล TCP/IP

จากรูปโปรโตคอล TCP/IP ประกอบด้วยโปรโตคอลย่อยหลายโปรโตคอลซึ่งทำงานในระดับต่าง ๆ กัน โดยที่โปรโตคอลย่อยที่ทำงานในระดับ Application Layer ได้แก่ FTP (File Transfer Protocol), Telnet, SNMP (Simple Network Management Protocol) โปรโตคอลเหล่านี้จะเรียกใช้โปรโตคอลย่อยในระดับ Transport ด้านล่างแบบใดแบบหนึ่งระหว่าง TCP (Transmission Control Protocol) หรือ UDP (User Datagram Protocol) และโปรโตคอลย่อยทั้งสองจะเรียกใช้

โปรโตคอลระดับ Network ต่อไป ซึ่งได้แก่ IP หรือ ICMP (Internet Communication Message Protocol)

สำหรับการทำงานในระดับ Transport ที่มีโปรโตคอลย่อย TCP หรือ UDP นั้น โปรแกรมที่อยู่ด้านบนจะเรียกใช้ผ่านช่องทางที่เป็นตัวเลขที่เรียกว่า Port Number [4] โดยหมายเลขนี้เป็นเลขที่มาตรฐานในโปรโตคอลแบบ TCP/IP มาตรฐานของ Port Number ทั้ง TCP และ UDP ได้แสดงไว้บางส่วนในตารางที่ 2.1-2.2



รูปที่ 2.8 การติดต่อสื่อสารระหว่างคอมพิวเตอร์ที่ใช้โปรโตคอล TCP/IP

ตารางที่ 2.1 Port Number ในส่วนของ TCP

TCP Port No.	Service Name	Application Layer Services
5	RJE	Remote Job Entry
7	Echo	Echo
20	FTP	File Transfer Protocol (Data)
21	FTP	File Transfer Protocol(Control)
23	Telnet	Terminal connection
25	SMTP	Simple Mail Transfer Protocol
37	Time	Time
53	Domain	Domain Name Server(DNS)
66	SQL*NET	Oracle SQL*NET
80	HTTP	Hypertext Transfer Protocol
110	POP3	Post Office Protocol

## ตารางที่ 2.2 Port Number ในส่วนของ UDP

UDP Port No.	Service Name	Application Layer Interface
7	Echo	Echo
13	Day Time	Day Time
37	Time	Time
69	TFTP	Trivial File Transfer Protocol
70	Gopher	Gopher
107	R Telnet	Remote Telnet
161	SNMP	Simple Network Management Protocol

## 2.4 การเลือกเส้นทางเดินของข้อมูลและโปรโตคอล

จุดประสงค์หลักของการเลือกเส้นทางเดิน คือ จัดเตรียมการเชื่อมโยงที่มีประสิทธิภาพให้เกิดการเชื่อมต่อระหว่าง โครงข่าย รวมถึงการรักษาความปลอดภัย ในส่วนของระบบเครือข่าย อินเทอร์เน็ต เราเตอร์จะมีหน้าที่รับ-ส่งข้อมูลผ่านชุดการเชื่อมต่อระหว่างกันบนโครงข่าย โดยเราเตอร์แต่ละเราเตอร์จะกำหนดการตัดสินใจเลือกเส้นทางเดินโดยใช้พื้นฐานความรู้เรื่องกราฟโครงข่าย (Network Topology) โดยอาศัยข้อมูลหมายเลข IP และเงื่อนไขต่าง ๆ ของเครือข่าย เช่น เวลาที่รอคอยของแพ็กเก็ต และความสมดุลในการกระจายปริมาณข้อมูล เป็นต้น และสำหรับเครือข่ายที่มีโครงสร้างแบบไม่ซับซ้อนมากนัก การกำหนดแบบแผนหรืออัลกอริทึมการหาเส้นทางเป็นไปได้ง่าย ส่วนในเครือข่ายที่ซับซ้อนมากขึ้น จำเป็นต้องพิจารณาความไม่หยุดนิ่งของการสื่อสารระหว่างเส้นทางนั้น ด้วย อีกทั้งเราเตอร์เองก็ต้องพยายามหลีกเลี่ยงส่วนของความผิดพลาดที่เกิดขึ้นบนโครงข่าย และ ส่วนของความคับคั่งของ ทราฟฟิกในแต่ละเส้นทาง

ฟังก์ชันของการกำหนดเส้นทาง (Routing Function) สามารถจำแนกเป็น 2 ส่วน คือ Routing Information ซึ่งเป็น Information ที่เกี่ยวกับ Topology ของโครงข่าย และการล่าช้าของระบบ Internet และ Routing Algorithm ซึ่งเป็น ขั้นตอนที่ใช้สร้างเส้นทางการตัดสินใจเพื่อ Datagram โดยเฉพาะ และการวางรากฐานการกำหนดเส้นทางโดยทั่วไปของ Information

ชั้นควบคุมเครือข่ายซึ่งมีหน้าที่ในการจัดส่งแพ็กเก็ตจากเครื่องผู้ส่งไปยังเครื่องผู้รับ สำหรับระบบเครือข่ายย่อย (Subnet) แพ็กเก็ตจะถูก “รับแล้วส่งต่อ (Hop)” หลายครั้งกว่าจะถึงปลายทาง ยกเว้นชั้นควบคุมเครือข่ายของระบบที่ใช้การส่งข้อมูลแบบกระจายอยู่ในเครือข่ายเดียวกันเท่านั้นที่ไม่ต้องจัดการปัญหานี้ แต่ถ้าผู้ส่งและผู้รับข้อมูลไม่ได้อยู่ในเครือข่ายเดียวกัน ก็

ยังคงเกิดปัญหาขึ้นได้ ดังนั้นกฎการ รับ-ส่ง ข้อมูล อัลกอริทึม และ โครงสร้างข้อมูลที่ทำหน้าที่เลือกเส้นทางเดินข้อมูลจึงเป็นส่วนประกอบที่สำคัญที่ได้รับการพัฒนาอย่างต่อเนื่อง

#### 2.4.1 อัลกอริทึมสำหรับการเลือกทางเดินข้อมูล (Routing Algorithms)

Routing Algorithms จะทำหน้าที่กำหนดการตัดสินใจของเราเตอร์ว่า แพ็กเก็ตที่รับเข้ามา ควรจะส่งต่อไปให้เราเตอร์ที่โหนดใดเป็นลำดับต่อไป โดยกำหนดข้อมูลที่ดีที่สุดในการสร้างตารางการเลือกเส้นทาง (Algorithm Table) Algorithms แต่ละ Routing Algorithms จะทำการตีความหมายของค่าว่าข้อมูลที่ดีที่สุด ของตัวเอง และสร้างตัวเลขขึ้นมาเรียกว่า ค่าใช้จ่าย (Cost) หรือเรียกว่า เมตริก (Metric) ของแต่ละเส้นทางในโครงข่าย โดยปกติ ค่าที่น้อยกว่าจะหมายถึงเส้นทางที่ดีกว่า และเส้นทางที่ส่งผ่านข้อมูลจะได้รับการกำหนดขึ้นมาด้วยอัลกอริทึมใด ๆ จะต้อง มีคุณลักษณะ 6 ประการคือ ความถูกต้อง (Correctness) ความง่าย (Simplicity) ความคงทน (Robustness) ความแน่นอน (Stability) ความเป็นธรรม (Fairness) และความเหมาะสม (Optimality) [5]

การเลือกเส้นทางเดินข้อมูลแบ่งออกเป็นสองประเภทคือประเภท ปรับตัวได้ (Adaptive Algorithms) และประเภทที่ไม่มีการปรับตัว (Non-Adaptive Algorithms) ประเภทที่ไม่มีการปรับตัว จะไม่นำสถานะของเครือข่าย เช่น ปริมาณข้อมูลในระบบหรือรูปแบบ โครงสร้างเครือข่ายเข้ามาใช้ในการพิจารณาเลือกทางเดินข้อมูล ทุกทางเดินข้อมูลจะมีการวางแผนไว้ล่วงหน้า และจัดส่งแผนทางเดินนี้ไปให้กับทุกโหนดในระบบ บางครั้งเรียกทางเดินนี้ว่า การเลือกทางเดินแบบสถิตย์ (Static Routing Algorithms) สำหรับการเลือกทางเดินประเภทมีการปรับตัว จะมีการเปลี่ยนแปลงการตัดสินใจเพื่อให้เหมาะสมกับสถานะแวดล้อมของระบบในขณะที่จะส่งข้อมูล อัลกอริทึมประเภทนี้จะแตกต่างกันที่วิธีการให้ได้ว่าซึ่งข้อมูลเมื่อมีการเปลี่ยนเส้นทาง และกรรมวิธีที่ใช้ในการวัดความเหมาะสมของแต่ละเส้นทาง บางครั้งเรียกว่า การเลือกทางเดินแบบพลวัต (Dynamic Routing Algorithms) ในบทนี้จะกล่าวถึงเฉพาะการเลือกทางเดินแบบสถิตย์ที่ใช้ในการวิจัยเท่านั้น

#### 2.4.2 การเลือกทางเดินแบบสถิตย์ (static routing algorithms)

การเลือกทางเดินแบบสถิตย์ ส่วนมากถูกนำมาใช้ในการหาเส้นทางสำรอง สามารถสรุปการทำงานได้ดังนี้

1. ตั้งค่ากำหนดของเส้นทางนั้น
2. เราเตอร์ติดตั้งเส้นทางนั้นเข้าไปในตารางเส้นทางทางเดินข้อมูล (Routing table)

ถ้าเครือข่ายยังสามารถเชื่อมต่อได้

3. แพ็กเก็ตถูกส่งออกไปตามเส้นทางที่กำหนด

สำหรับการรับส่งข้อมูลระหว่างคอมพิวเตอร์ที่อยู่ภายใน Subnet Address เดียวกัน คืออยู่ในวง LAN เดียวกัน คอมพิวเตอร์จะส่งแพ็กเก็ตไปได้ทันที แต่ถ้าเป็นการรับส่งระหว่างคอมพิวเตอร์ที่

อยู่ต่าง Subnet Address กัน เครือข่ายจำเป็นจะต้องมีอุปกรณ์ เครือข่าย ที่ทำงานอยู่ใน Layer 3 ซึ่งได้แก่ Router หรือ Switch (Layer 3) ซึ่งอุปกรณ์เหล่านี้จะทำหน้าที่ในการจัดหาเส้นทาง เพื่อให้คอมพิวเตอร์ที่อยู่ต่าง Subnet สามารถติดต่อสื่อสารกันได้นั่นเอง ในการหาเส้นทาง เราเตอร์จะไม่ใช้วิธีในการจำ IP Address ของคอมพิวเตอร์ปลายทางทุก ๆ ตัว เพราะจะทำให้เกิดปัญหาเรื่องความจำ และที่สำคัญมากก็คือเราเตอร์จะทำงานได้ช้าลง แต่เราเตอร์จะจำ IP Address ของแต่ละ Subnet โดยเราเตอร์สามารถตรวจสอบได้ว่า IP Address ปลายทางจะส่งไปอยู่ใน Subnet เดียวกันหรือไม่ โดยการนำ IP Address ปลายทาง มาทำการ Logical AND กับ Subnet Mask ของตัวเอง ผลลัพธ์ที่ได้ก็คือ Subnet Address ของ Router ปลายทาง (Destination Subnet Address) เมื่อได้รับ IP Address ปลายทางแล้ว ต่อมาก็จะจัดการทำ Logical AND IP Address ต้นทาง กับ Subnet Mask ของตัวเอง ผลลัพธ์ที่ออกมาก็คือ Subnet Address ต้นทาง (Source Subnet Address) ค่าที่คำนวณมาได้จากการ Logical AND ทั้ง 2 ครั้งเป็นค่า Network Address ในแต่ละ Subnet จากนั้นนำค่า Destination Subnet Address และ Source Subnet Address มาเปรียบเทียบกัน ถ้าผลลัพธ์ที่ออกมา เท่ากัน แสดงว่า อยู่ใน Subnet เดียวกัน ก็จะส่ง แพ็กเก็ต ได้โดยทำการ ARP (Address Resolution Protocol) เพื่อตรวจสอบ เลขประจำตัวอุปกรณ์ (MAC Address) ของเครื่องปลายทาง จากนั้นจึงส่งแพ็กเก็ตออกไป ถ้าผลลัพธ์ที่ออกมาไม่เท่ากัน แสดงว่าเครื่องปลายทางอยู่ต่าง Subnet ก็จะทำการ ARP อีกครั้ง โดยครั้งนี้ไม่ได้ตรวจสอบ MAC Address ของเครื่องคอมพิวเตอร์ แต่เป็นการตรวจสอบ MAC Address ของเราเตอร์เมื่อเจอก็จะ Encapsulate Packet IP ลงไปใน Ethernet Frame โดยระบุ MAC Address ของเราเตอร์ปลายทางเพื่อค้นหาเส้นทางต่อไป การทำ ARP เป็นกระบวนการที่คอมพิวเตอร์จะ Broadcast ไปยังคอมพิวเตอร์ทุกเครื่องในโครงข่ายโดยการส่ง IP Address แล้วตรวจสอบ MAC Address ถ้า IP Address ตรงกับปลายทางคอมพิวเตอร์ปลายทางก็จะตอบรับ

ตัวอย่าง การจัดหาเส้นทาง เพื่อให้คอมพิวเตอร์ที่อยู่ต่าง Subnet สามารถติดต่อสื่อสารกันได้ โดยกำหนดให้เครื่องคอมพิวเตอร์ต้นทาง IP Address 168.108.2.10 Subnet Mask 255.255.255.0 ต้องการ Ping ไปที่โหนดปลายทางหมายเลข IP 168.108.3.30 มีวิธีทำดังนี้คือ

1. จัดการ Logical AND IP Address ปลายทางกับ Subnet Mask ต้นทางจะได้

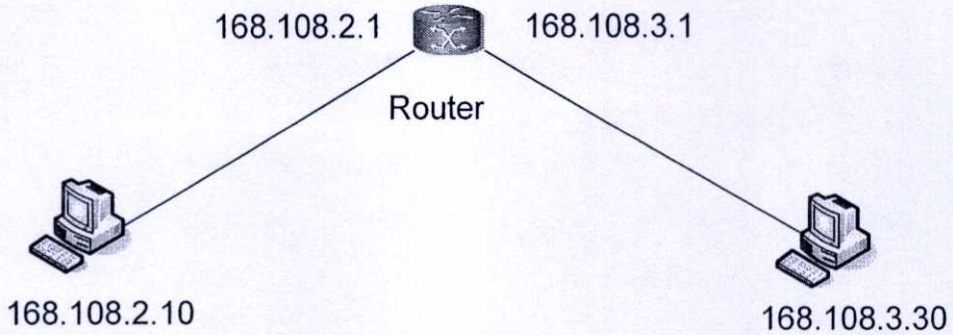
10101000.01101100.00000011.00011110

11111111.11111111.11111111.00000000

10101000.01101100.00000011.00000000

และได้ Subnet Address ปลายทางเป็น 168.108.3.0

2. จัดการ Logical AND IP Address ต้นทางกับ Subnet Mask ของต้นทางเอง จะได้ Subnet Address ต้นทางเป็น 168.108.2.0



รูปที่ 2.9 การเปรียบเทียบ Subnet Mask ของเราเตอร์

3. นำข้อผลลัพธ์ในข้อ 1 กับข้อ 2 มาเปรียบเทียบกัน ผลที่ได้คือไม่เท่ากัน ก็แสดงว่าเครื่องต้นทางกับปลายทางอยู่คนละ Subnet กัน และที่คอมพิวเตอร์ IP Address 168.108.2.10 จะ ping ไปที่ 168.108.3.30 ได้นั้น IP Address 168.108.2.10 เอง จะต้องได้ข้อมูลของ Subnet Address ของ 168.108.3.0 ใน Routing Entry ของ Routing Table เสียก่อน ซึ่งจะสามารถดูได้โดยเข้าไปที่ Command Line แล้วพิมพ์คำว่า netstat -r หรือ route print

โดยทั่วไปโครงสร้างของ Routing Entry คือ < Destination Subnet Address > < Subnet Mask > จากตัวอย่างรูปที่ 2.10 จะได้ 168.108.3.0 255.255.255.0 และ 168.108.2.1 โดยปกติแล้ว คอมพิวเตอร์ที่ใช้จะมีเราเตอร์ 1 ตัว ดังนั้นจึงทำให้เกิดคำว่า “Default Route” ขึ้นมา ซึ่งเมื่อคอมพิวเตอร์ต้องการจะส่ง แพ็กเก็ต ไปต่าง Subnet ก็จะไปส่งออกทาง Default Route ได้ทันที แล้วเราเตอร์จะจัดการหาเส้นทางตาม Routing Table ที่มีอยู่ในตัวเราเตอร์นั้น

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>netstat -r

Route Table
=====
Interface List
0x1 ..... MS TCP Loopback interface
=====
Active Routes:
Network Destination  Netmask          Gateway          Interface        Metric
127.0.0.0             255.0.0.0        127.0.0.1        127.0.0.1         1
=====
Persistent Routes:
None

C:\Documents and Settings\Administrator>route print

Interface List
0x1 ..... MS TCP Loopback interface
=====
Active Routes:
Network Destination  Netmask          Gateway          Interface        Metric
127.0.0.0             255.0.0.0        127.0.0.1        127.0.0.1         1
=====
Persistent Routes:
None

C:\Documents and Settings\Administrator>
  
```

รูปที่ 2.10 การใช้คำสั่ง netstat - r และ route print

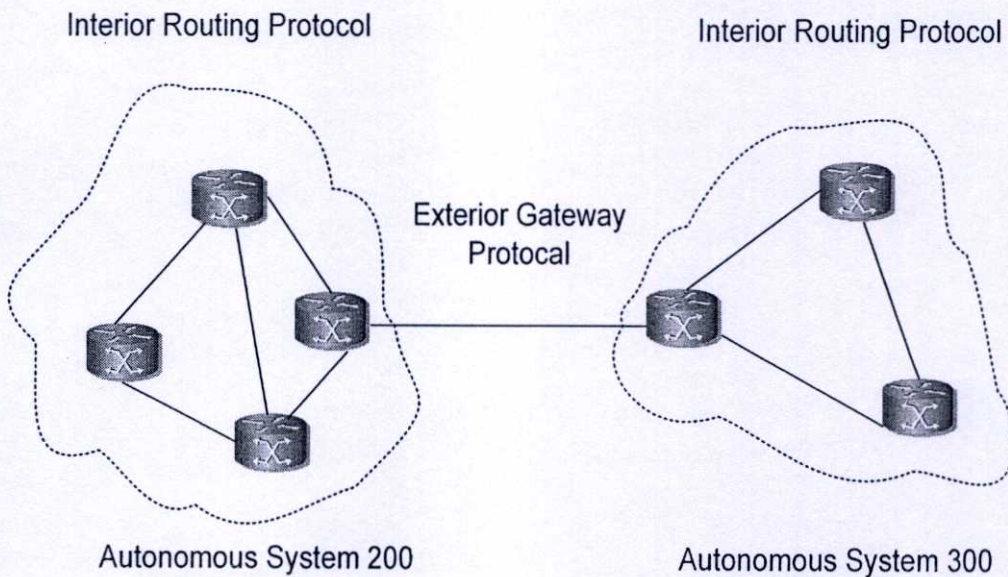
### 2.4.3 เราติงโปรโตคอล (Routing Protocol)

เราติงโปรโตคอล เป็นโปรโตคอลในชั้นสื่อสารควบคุมเครือข่าย ซึ่งหมายถึงการสื่อสารระหว่างเราเตอร์ เช่น ช่วยให้เราเตอร์แต่ละตัวสามารถใช้ข้อมูลร่วมกันได้ โดยข้อมูลที่ได้รับจะถูกนำไปใช้ในการปรับปรุงข้อมูลในตารางเส้นทางเดินข้อมูลของเราเตอร์ เราติงโปรโตคอล สามารถแบ่งออกเป็นหลายประเภท โดยแบ่งตามขอบเขตการใช้งาน แบ่งตามพฤติกรรมการทำงาน แบ่งตามความสัมพันธ์กับ Subnet Mask ในหัวข้อนี้จะกล่าวถึงโดยย่อเฉพาะโปรโตคอลที่สำคัญที่มีการใช้ในระบบ TCP/IP

#### 2.4.3.1 เราติงโปรโตคอลแบ่งตามขอบเขตการใช้งานได้

เราติงโปรโตคอลแบ่งตามขอบเขตการใช้งาน สามารถแบ่งออกเป็น 2 ประเภทคือ

1. Interior Protocol เป็นโปรโตคอลที่ใช้แลกเปลี่ยนฐานความรู้ระหว่างเราเตอร์ภายในองค์กรเดียวกัน ซึ่งใช้ในระบบ TCP/IP ได้แก่ Routing information Protocol (RIP) Interior Gateway Routing Protocol (IGRP) Enhanced Interior Gateway Routing Protocol (EIGRP) Open Shortest Path First (OSPF)
2. Exterior Protocol เป็นโปรโตคอลที่ใช้แลกเปลี่ยนฐานความรู้ต่างองค์กรกันหรือความน่าเชื่อถือต่างกัน ซึ่งได้แก่ BGP และ EGP



รูปที่ 2.11 เราติงโปรโตคอลแบ่งตามขอบเขตการใช้งาน

### 2.4.3.2 เราตั้งโปรโตคอลแบ่งตามพฤติกรรมการทำงาน

เราตั้งโปรโตคอล แบ่งตามพฤติกรรมการทำงานได้ 2 ประเภทคือ

1. Distance Vector หมายถึงการที่เราเตอร์ทำการแลกเปลี่ยน Routing Table กับเราเตอร์เพื่อนบ้านนั่นเอง ซึ่งเราเตอร์เพื่อนบ้านนั้นจะเชื่อตาม Routing Table ที่ส่งมาทันที จากนั้นก็จะทำการ Update Routing Table ของเดิมทันที

2. Link State คือการที่เราเตอร์จะเรียนรู้สถานการณ์ทำงานของแต่ละ Interface ของตัวเอง แล้วส่งไปให้ เราเตอร์เพื่อนบ้านตัดสินใจเลือกจัดการเอง

### 2.4.3.3 เราตั้งโปรโตคอลแบ่งตามความสัมพันธ์กับ Subnet Mask

เราตั้งโปรโตคอลสามารถแบ่งตามความสัมพันธ์กับ Subnet Mask ออกเป็น 2 ประเภทคือ

1. Classless คือ การที่ทุก ๆ Subnet จะต้องมีการมี Subnet Mask เหมือนกันทั้ง Network
2. Classful คือการที่ทุก ๆ Subnet จะต้องมีการมี Subnet Mask โดยในแต่ละโครงข่าย อาจจะมีการมี Subnet Mask ไม่เหมือนกัน

### 2.4.4 ตารางการจัดเส้นทาง (Routing Table)

Routing Table จะประกอบไปด้วยบันทึกหรือฐานข้อมูล (Record) ต่างๆ ซึ่งเรียกว่า Routing Entry ในวิชาเน็ตเวิร์ก และภายใน Routing Entry จะประกอบด้วยบันทึกหรือฐานข้อมูล (Record) ต่าง ๆ ในแต่ละกลุ่ม เช่น เป้าหมายปลายทาง ตัวเลขบอกลำดับความสำคัญ (Administrative Distance, AD) ดังตัวอย่างของคำสั่งที่อยู่ใน Routing Table ดังต่อไปนี้

```
Router1# show ip route
```

```
Code : C - Connected, S - Static, I - IGRP, R - RIP, M-Mobile, B - BGP,
```

```
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF internal area
```

```
C 10. 34. 0. 0 is directly connect, Serial 1/2
```

```
D 10. 36. 0. 0 [90/20537600] via 10. 34. 0. 3 00: 02: 40 Serial 1/2
```

มีความหมายดังต่อไปนี้

1. D หมายถึงแหล่งที่มาของ Routing entry ว่า มาจาก Protocol แบบใด ซึ่งจะสามารถดูได้จาก Code ในที่นี้ D ก็คือ Protocol แบบ EIGRP

2. 10.36.0.0 หมายถึง Subnet Address ปลายทาง

3. 90 หมายถึงค่า Administrative Distance (AD) ซึ่งใช้ในการบอกความน่าเชื่อถือของ Routing Entry นี้

4. 20537600 หมายถึงค่าต้นทุนของเส้นทางนี้ แต่ละ Protocol จะมีค่าแตกต่างกันไป

5. via 10.34.0.3 หมายถึง IP Addressของเราเตอร์ตัวถัดไป (Next Hop Router) ที่เราเตอร์จะส่งแพ็กเก็ตเกิดไปให้เพื่อส่งต่อไปยังปลายทาง
6. 00:02:40 หมายถึงระยะเวลาเริ่มแรกที่ Routing Entry นี้ถูกสร้างขึ้น
7. Serial 1/2 หมายถึง Interface ขาออก ของเราเตอร์ปัจจุบันที่แพ็กเก็ตจะถูกส่งออกไปให้กับเราเตอร์ตัวถัดไป

#### 2.4.5 การอ่าน Routing Table ของ Router

เมื่อรับแพ็กเก็ตเข้ามาหนึ่งแพ็กเก็ตแล้ว เราเตอร์จะทำการตรวจสอบว่า Subnet Address ตรงกับ Interface ที่กำหนดไว้ที่ตัวเราเตอร์เองหรือไม่ เช่น สมมุติเราใช้คำสั่ง Show IP Route จะเห็นว่ามี C 192.168.10.0 is directly connected , Fast Ethernet 0/1 ซึ่งจะเห็นได้ว่า Subnet Address 192.168.10.0 ถูกเชื่อมต่ออยู่ตรง ๆ กับ Fast Ethernet 0/1 โดยที่คอมพิวเตอร์ปลายทางจะถูกเชื่อมต่ออยู่กับ Subnet Address เดียวกับเราเตอร์ แล้วเราเตอร์ จึงทำการส่งแพ็กเก็ตไปยังคอมพิวเตอร์ปลายทางโดยการทำ ARP แต่ถ้ารับแพ็กเก็ตเข้ามาแล้วไม่ตรงกับ Subnet Address ที่ Interface กับตัวเอง เราเตอร์ก็จะทำการค้นหา Routing Entry ที่ตรงกับ Subnet Address ปลายทางที่เรารู้อยู่ แล้วอ่านค่า Next Hop Address ใน Routing Entry ว่าชี้ไปที่เราเตอร์ตัวใด จากนั้นจึงส่งแพ็กเก็ตออกไปทาง Interface ขาออก ซึ่งก่อนจะทำการส่ง จะทำ ARP เพื่อหา MAC Address ของ เราเตอร์ ปลายทาง และห่อหุ้ม (Encapsulate) Packet IP Ethernet Frame ก่อนแล้วส่งแพ็กเก็ตให้เราเตอร์ตัวถัดไป เพื่อหาเส้นทางต่อไป

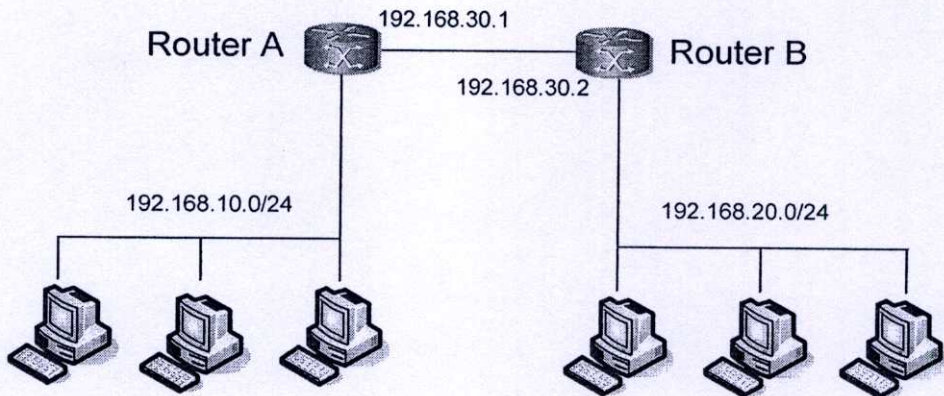
#### 2.4.6 วิธีการสร้าง Static Router

สามารถทำได้โดยการเพิ่ม Static Route เข้าไปในเราเตอร์ Static Route เป็นเรตติ้งเอ็นทรีที่ผู้ดูแลระบบเพิ่มเข้าไปเอง เพื่อบอกให้เราเตอร์ทราบว่า หากเราเตอร์ต้องการส่งแพ็กเก็ตไปยัง Subnet Address ต่าง ๆ เราเตอร์จะต้องส่งแพ็กเก็ตไปหาเราเตอร์ตัวถัดไป (Next Hop Address) ตัวใดตัวหนึ่ง หรือบอกว่าเราเตอร์จะต้องส่งไปทางอินเตอร์เฟซใด ในการสร้าง Static Route จะต้องกระทำสองทิศทางคือ เราเตอร์ A จะต้องชี้ Static Route ไปหาเราเตอร์ B และเราเตอร์ B จะต้องชี้ Static Route ไปหาเราเตอร์ A ด้วย แพ็กเก็ตจึงจะสามารถส่งไป และกลับได้ แสดงดังรูปที่ 2.12 ซึ่งโครงสร้างของ Static Route เป็นดังนี้

ip route < Subnet Mask ปลายทาง > < ip interface ของ router ที่จะออก >

โดยที่เราเตอร์ A ทำ IP Route เพื่อให้เห็น Subnet Address 192.168.20.0 สามารถสร้าง IP Route ได้โดย IP Route 192.168.20.0 255.255.255.0 192.168.30.2 ที่เราเตอร์ B ต้องทำ IP Route เพื่อให้เห็น Subnet Address 192.168.10.0 สามารถสร้าง IP Route ได้โดย IP Route 192.168.10.0 255.255.255.0 192.168.30.1 นอกจากนั้นยังสามารถใส่ค่า Parameter AD (Administrative Distance) ซึ่งเป็นค่าที่บ่งบอกถึงความน่าเชื่อถือของ Routing Entry ที่เราสร้างขึ้น ในกรณีนี้จะใช้

ถ้าสำหรับทำ Link Backup ของการสร้าง Routing Table แบบ Dynamic Route เช่น Dynamic Route ใช้ EIGRP Protocol มีค่า Cost เท่ากับ 90 ถ้าเราจะสร้างเส้นทางสำรองโดย Static Route เราจะต้องใส่ค่า AD เท่ากับ 91 ซึ่งในการหาเส้นทางที่ดีที่สุดจะพิจารณาค่า AD ที่มีค่าน้อยก่อน ดังนั้นถ้า Link ที่ทำแบบ Dynamic Route เกิดเหตุขัดข้อง Link ที่เป็น Static Route จะทำงานแทนทันที



รูปที่ 2.12 การทำ Static Route

#### 2.4.7 วิธีการทำ Enable Dynamic Route

Dynamic Route เป็นการเรียนรู้ Subnet Address ต่าง ๆ แบบอัตโนมัติ และมีการยกเลิกการใช้งาน Subnet Address อัตโนมัติ เมื่อ Subnet Address เกิดเหตุขัดข้อง โดยเราตั้งโปรโตคอลที่ใช้แต่ละโปรโตคอลอาจมีขั้นตอนในการปฏิบัติ ต่างกันหรือเหมือนกันบ้างกับโปรโตคอลอื่น ๆ เราตั้งโปรโตคอลที่ใช้งานในปัจจุบันได้แก่ RIP IGRP EIGRP OSPF IS-IS (Intermediate System-Intermediate System) และ BGP (Border Gateway Protocol) ซึ่งการทำ Enable Dynamic Route สามารถทำได้โดยการรันเราตั้งโปรโตคอลขึ้นมา จากนั้นเราเตอร์แต่ละตัวจะเรียนรู้ฐานข้อมูลจากตารางเราตั้งของเราเตอร์ข้างเคียง โดยเราตั้งโปรโตคอลจะมีความสามารถในการตรวจจบการเปลี่ยนแปลงต่าง ๆ ในโครงข่าย และ ปรับปรุงตารางเราตั้งให้สอดคล้องกับสภาพความเป็นจริงของโครงข่ายในขณะนั้นอย่างอัตโนมัติ จะเห็นได้ว่า การใช้ Dynamic Route จะช่วยให้ผู้บริหารจัดการระบบโครงข่าย สามารถลดภาระในการจัดการระบบลงไปได้อย่างมาก

### 2.4.8 Administrative Distance (AD)

ในการเรียนรู้ Subnet Address ของ Router จะมีการเรียนรู้มากกว่าหนึ่งแหล่งที่มา ดังนั้นเราเตอร์ จะพิจารณาค่าความน่าเชื่อถือว่า แหล่งไหนมีความน่าเชื่อถือมากกว่ากัน เพื่อตัดสินใจ ในการเลือกเส้นทางซึ่งค่าความน่าเชื่อถือต่าง ๆ แสดงดังตารางที่ 2.3 [6]

ตารางที่ 2.3 ค่าความน่าเชื่อถือของเราติง โปร โทคอล

ลำดับที่	แหล่ง ที่ให้ข้อมูล ( Routing Protocol )	ค่า AD Default
1	Directly connected interface	0
2	Static route ที่ออกไปทาง interface ตัวเอง	0
3	Static route ที่ออกไปทาง router เพื่อนบ้าน	1
4	EIGRP Summary Route	5
5	External BGP	20
6	Internal EIGRP	90
7	IGRP	100
8	OSPF	110
9	IS-IS	115
10	RIP ( V1 and V2 )	120
11	EGP	140
12	External EIGRP	170
13	Internal BGP	200
14	Unknown	255

ข้อดีของ Static Route คือ

1. ไม่เป็นการเพิ่มภาระทำงานหรือ โหลด (Load) ของเราเตอร์ในการ Update Routing Table หรือข้อมูลเกี่ยวกับสถานะของเราเตอร์เป็นระยะ ๆ และไม่ต้องคำนวณเราติงเทเบิลใหม่อยู่บ่อย ๆ เมื่อโครงข่ายมีเหตุขัดข้อง

2. ไม่เพิ่ม ทราฟฟิก บนแบนด์วิดท์ ของโครงข่าย

3. มีความปลอดภัยมากกว่า Dynamic Route เพราะ Dynamic Route เมื่อมีผู้อื่นมาเชื่อมต่ออุปกรณ์ก็สามารถจะใช้งานได้เลย ไม่ต้องผ่านผู้ดูแลระบบซึ่งอาจจะทำให้ระบบเสียหายได้

4. Static Route จะนิยมใช้ในการสร้างเส้นทางสำรองมากกว่าการสร้างเส้นทางหลัก

ข้อเสียของ Static Route คือ

1. จะต้องทำ Routing Entry ทุก Subnet Address จึงทำให้เสียเวลา

2. Static Route ไม่สามารถตรวจสอบสถานะของเราเตอร์ตัวถัดไปได้ เช่น

ถ้าเราเตอร์ตัวถัดไปเกิดเหตุขัดข้อง ก็ยังส่งแพ็กเก็ตไปให้อยู่ตลอด จนกว่าผู้ดูแลระบบจะแก้ไข

### 2.4.9 ค่าเชื่อมโยงหรือค่าต้นทุน (Cost) ในการหาเส้นทาง

ถ้าเราเตอร์มีการเชื่อมต่อกันหลาย ๆ เส้นทาง เราเตอร์จะเลือกเส้นทางที่ดีที่สุดในการรับส่งข้อมูล ซึ่งจะต้องเลือกค่า Cost ที่น้อยที่สุด โดยเราตั้งโปรโตคอลแต่ละ Protocol จะมีค่า Cost ตามมาตรฐานของตัวเอง เช่น

RIP Version 1 และ Version 2 ใช้ค่า Cost ซึ่งได้มาจากการนับจำนวน Hop ใช้อัลกอริทึมในการคำนวณหาต้นทุนแบบ Bellman-Ford โดยสามารถมีโหนดได้สูงสุด 15 โหนด

IGRP ใช้ค่าต้นทุนจาก Composite (แบนด์วิดท์ และ Delay Time) ใช้อัลกอริทึมในการคำนวณหาต้นทุนแบบ Bellman-Ford โดยสามารถมีโหนดได้สูงสุด 100 โหนด

EIGRP ใช้ค่าต้นทุนจาก Composite (Delay, Bandwidth, Reliability, Loading และ MTU) ใช้อัลกอริทึมในการคำนวณหาต้นทุนแบบ DUAL (Diffusing Update Algorithm) โดยสามารถมีโหนดได้สูงสุด 100 โหนด และสามารถปรับปรุงได้ถึง 250 โหนด

OSPF ใช้ค่าต้นทุนจากค่า แบนด์วิดท์ ของโครงข่าย ใช้อัลกอริทึมในการคำนวณหาต้นทุนแบบ Dijkstra โดยสามารถมีโหนดได้โดยไม่จำกัด [7]

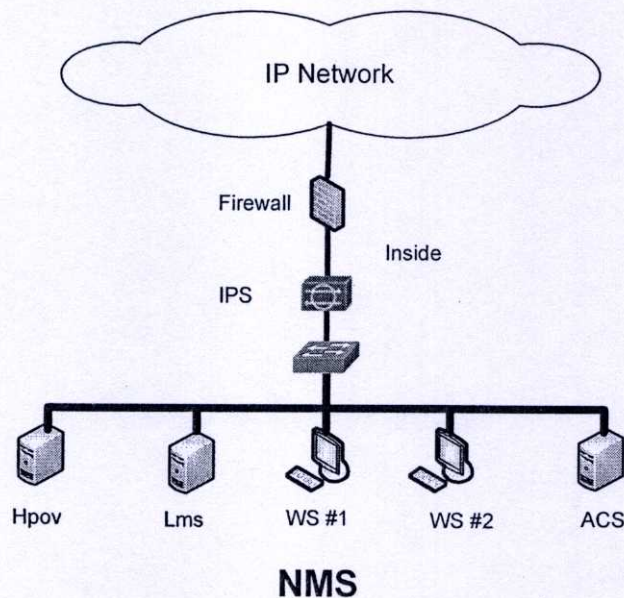
## 2.5 ไอพีทราฟฟิก (IP Traffics)

ความคับคั่งของข้อมูล (Congestion) เป็นสาเหตุที่สำคัญที่จะทำให้ประสิทธิภาพโดยรวมของระบบลดลง โดยสามารถเกิดได้จากหลายสาเหตุ เช่น แพ็กเก็ต ของข้อมูลถูกส่งมาจากหลายเส้นทางพร้อม ๆ กัน และจะต้องส่งออกไปทางเส้นทางเดียวกัน เราเตอร์จึงต้องเก็บข้อมูลรอจำนวนมาก จนหน่วยความจำไม่สามารถรองรับได้ ข้อมูลจะเกิดการสูญหายไป การเพิ่มขนาดของหน่วยความจำอาจช่วยแก้ปัญหานี้ได้ แต่เราเตอร์ที่มีหน่วยความจำไม่จำกัด ความคับคั่งจะยิ่งมากขึ้น [Nagle, 1987] เนื่องจากเงื่อนไขของเวลา (Time Out) ผู้ใช้จะส่งข้อมูลซ้ำเข้ามาอีกเป็นการเพิ่มความคับคั่งมากขึ้น นอกจากนี้หน่วยประมวลผลของเราเตอร์ที่ทำงานช้า และการใช้สายความเร็วต่ำ ก็เป็นอีกสาเหตุหนึ่งของความคับคั่งดังกล่าว ในบทวิจัยนี้ จึงทำการตรวจสอบ ปริมาณของไอพี ทราฟฟิก (Traffic IP) และทำนายปริมาณ ความความคับคั่งข้อมูลแต่ละโหนด เพื่อ หาเส้นทางหลีกเลี่ยงจุดที่เกิดปัญหา และใช้เป็นองค์ประกอบในการหาเส้นทางที่เหมาะสมในการกระจายข้อมูลสู่เราเตอร์ในหลายเส้นทางได้ แนวทางที่นำมาใช้คือกำหนดหลักเกณฑ์บางประการ ในอัลกอริทึมสำหรับเลือกทางเดินในบทต่อไป

สำหรับรายละเอียดในการเก็บข้อมูลทราฟฟิกที่ต้องมีการเฝ้าระวัง คือมีการเก็บข้อมูลตลอดเวลา [8, 9] และการคาดการณ์ปริมาณการใช้ทราฟฟิกในอนาคตเพื่อประโยชน์ในการบริหารจัดการ การจัดเส้นทางสำรอง และบริหารจัดการโครงข่าย ในอนาคต ได้นำเสนอไว้ในบทที่ 4

## 2.6 ระบบบริหารจัดการโครงข่าย (Network Management System)

ระบบบริหารจัดการโครงข่ายหรือ NMS (Network Management System) ทำหน้าที่เป็นศูนย์กลางการควบคุม และเฝ้ามองโครงข่าย [10] มีระบบเตือนเมื่อมีส่วนหนึ่งส่วนใดของโครงข่ายทำงานผิดพลาดหรือเกิดเหตุขัดข้องทำให้ผู้ดูแลระบบทราบได้ทันที และเข้าไปทำการแก้ไขได้อย่างรวดเร็ว หน้าที่หลักของ NMS คือการตรวจสอบโครงข่ายตลอดเวลา ทำรายงานสถิติการใช้โครงข่าย เช่น สถิติของปริมาณข้อมูล ปริมาณผู้ใช้ โดยสามารถเขียนเป็นกราฟเพื่อให้ผู้ดูแลระบบนำไปวิเคราะห์ และวางแผนขยายเครือข่ายในอนาคต นอกจากนี้ผู้ดูแลระบบยังสามารถตรวจสอบและแก้ไขระบบ รวมถึงการติดตั้งซอฟต์แวร์ และการตั้งค่าระบบให้กับอุปกรณ์เครือข่ายที่อยู่ห่างไกลได้ จากจุดศูนย์กลาง NMS จึงเป็นอุปกรณ์ที่ระบบโครงข่ายขนาดใหญ่หรือผู้ให้บริการโครงข่ายแบบสาธารณะที่มีผู้ใช้บริการจำนวนมากจำเป็นต้องมี เพราะเป็นอุปกรณ์ที่ทำให้การเฝ้ามองระบบเป็นไปอย่างมีประสิทธิภาพ การที่ระบบบริหารจัดการโครงข่ายจะมีประสิทธิภาพหรือไม่ นั้น ขึ้นอยู่กับระบบซอฟต์แวร์ที่ต้องมีอยู่ในตัวอุปกรณ์โครงข่าย (Agent) เป็นส่วนสำคัญ ส่วนของเอเจนต์ยังมีการเก็บข้อมูลไว้ในฐานข้อมูลสารสนเทศ เพื่อการจัดการ MIB (Management Information Base) รูปตัวอย่างการต่อ NMS กับระบบโครงข่าย แสดงดังรูปที่ 2.13



รูปที่ 2.13 รูปตัวอย่างการต่อ NMS กับระบบโครงข่าย

การทำงานของอุปกรณ์ต่าง ๆ บนโครงข่ายจะมีส่วนข้อมูลของตัวเองเก็บไว้ที่ MIB ดังนั้น NMS จึงส่งคำถามไปยังเอเจนต์ของโหนดปลายทาง การส่งคำถาม และเอเจนต์ส่งข้อมูลคำตอบนี้ ย่อมเป็นไปตามมาตรฐานโปรโตคอลที่กำหนด เช่น ลักษณะคำถามคำตอบของ SNMP ที่สอบถาม

กันเป็นระบบ และเป็นมาตรฐานสากล ข้อมูลในฐานะข้อมูลที่เก็บในเอเจนต์ของแต่ละอุปกรณ์ ประกอบด้วย ข้อมูลชื่ออุปกรณ์ รหัสอุปกรณ์ หมายเลขแอดเดรสบนโครงข่าย ตารางกำหนดเส้นทาง ปริมาณข้อมูลที่รับส่ง ข้อผิดพลาดที่ปรากฏ ฯลฯ ดังนั้นระบบ NMS จึงได้ข้อมูลของทุกอุปกรณ์ที่มีเอเจนต์อยู่ และนำข้อมูลเหล่านั้นมาแสดงผลในเชิงวิเคราะห์ต่าง ๆ ได้แก่ ไดอะแกรมรูปภาพของโครงข่ายทางกายภาพ การนำข้อมูลมาแสดงผลที่ NMS การส่งคำถามไปยังอุปกรณ์ที่ โหนดต่าง ๆ เป็นระยะ ๆ และรับคำตอบมาปรับปรุงข้อมูล หากส่งคำถามไปยังตัวอุปกรณ์ที่มีในระบบ และไม่ได้รับคำตอบก็จะมีวิธีการตรวจสอบอย่างอื่นประกอบ เช่น อุปกรณ์นั้นมีปัญหาอย่างไร หรือไม่ หากพบปัญหา NMS จะแสดงปัญหาเพื่อให้ผู้ดูแลระบบทราบ ระบบบริหารจัดการโครงข่ายจึงเป็นซอฟต์แวร์ที่นำข้อมูลจากเอเจนต์ต่าง ๆ มาแสดงผล และติดต่อกับผู้ดูแลระบบ ดังนั้นจึงมีผู้พัฒนาระบบ NMS ในรูปแบบต่าง ๆ กันมากขึ้น พัฒนาการของ NMS ได้ก้าวหน้าขึ้นเป็นลำดับ การสร้างความเชื่อมั่นให้กับผู้ใช้งานโครงข่ายเป็นสิ่งสำคัญ NMS จึงเป็นสิ่งที่มีความจำเป็น และเข้ามามีบทบาทสำคัญในเทคโนโลยีโครงข่ายไอพี

## บทที่ 3

# การหาเส้นทางที่มีต้นทุนต่ำที่สุด

อัลกอริทึมในการหาเส้นทางถูกพัฒนาอย่างต่อเนื่องทั้งแบบสถิต และแบบพลวัต ตัวอย่างการหาเส้นทางแบบสถิตได้แก่วิธีการหาเส้นทางที่สั้นที่สุด วิธีพลัดตั้ง และวิธีกำหนดการไหลของข้อมูล เป็นต้น ส่วนวิธีแบบพลวัตได้แก่วิธีตารางระยะทาง และวิธีกำหนดสถานเชื่อมต่อ นอกจากนี้ยังมีวิธีสำคัญอื่น ๆ เช่น กรณีที่ระบบมีจำนวนเราเตอร์เพิ่มมากขึ้นจนเราเตอร์บางส่วนไม่สามารถทำงานได้ทัน จะใช้อัลกอริทึมในการหาเส้นทางแบบลำดับชั้นเป็นส่วนใหญ่ สำหรับในบทวิจัยนี้ได้ศึกษาอัลกอริทึมการหาเส้นทางที่มีต้นทุนต่ำที่สุดที่เกี่ยวข้อง และ ประยุกต์ใช้อัลกอริทึมเอสตาร์ในการหาเส้นทางที่ดีที่สุด จากนั้นได้ทำการปรับขบวนการในอัลกอริทึมเอสตาร์ ให้เหมาะสมกับการใช้งานในโครงข่ายไอพี ของ บริษัท ทีโอที จำกัด (มหาชน) เพื่อจัดเส้นทางสำรองในกรณีที่ระบบเกิดเหตุขัดข้อง ดังรายละเอียดในหัวข้อต่อไป

### 3.1 อัลกอริทึมการหาเส้นทางที่เกี่ยวข้อง

#### 3.1.1 อัลกอริทึมดิจทรา (Dijkstra Algorithm)

Dijkstra Algorithm เป็นอัลกอริทึมที่หาเส้นทางที่สั้นที่สุดที่นิยมใช้ แต่จากการศึกษาพบว่าต้องใช้เวลาในการค้นหาเยอะ เพราะต้องมีการค้นหาทั้งฟังก์ชัน  $d(v)$  และ  $Pred(v)$  ซึ่งทั้งสองฟังก์ชัน เป็นฟังก์ชันที่จะบอกว่าระยะทางที่สั้นที่สุดจากจุดเริ่มต้นถึงโหนด  $v$  ต้องเดินทางเป็นระยะทางอย่างน้อยเท่ากับ  $d(v)$  และต้องเดินทางผ่านโหนด  $Pred(v)$  อีกด้วย ประสิทธิภาพของอัลกอริทึมอยู่ในรูปของฟังก์ชันกำลังสองของฟังก์ชัน  $O(n^2)$  อัตราการสูญเสียข้อมมากกว่า  $O(n)$  แต่นิยมนำมาประยุกต์ใช้เนื่องจากอัลกอริทึม Dijkstra เขียนโปรแกรมได้ง่าย ไม่จำเป็นต้องจองเนื้อที่หน่วยความจำมาก Bing Liu และ Siew Hwse ได้นำเสนอวิธีการของ Dijkstra มาใช้กับการหาเส้นทางของแผนที่ถนนที่มีทั้งหาทิศทาง และระยะทาง [11]

#### 3.1.2 เหตุผลความรู้พื้นฐาน (Knowledge Base Reasoning)

Knowledge Base Reasoning เป็นอัลกอริทึมที่สามารถประยุกต์ใช้กับงานที่ต้องการให้มีการเรียนรู้ แต่ผู้ใช้สามารถออกแบบเพิ่มเติมได้ ขึ้นอยู่กับความต้องการของงานที่จะนำไปใช้ Bing Liu และ Tay J [12] ได้พัฒนาต่อจากผลการวิจัยของ Bing Liu และ Siew Hwse [11] ผลการวิจัยเป็นที่ยอมรับ แต่เนื่องจากขนาดของกลุ่มข้อมูลไม่แน่นอน ผลลัพธ์ที่ได้จึงไม่แน่นอนเสมอไปเมื่อเทียบกับ โมเดลทางสถิติ

### 3.1.3 โครงข่ายประสาทเทียม (Neural Network)

การใช้โครงข่ายประสาทเทียมในการหาเส้นทาง เป็นที่น่าสนใจเนื่องจากปัจจุบันนี้มีการวิจัยและนำโครงข่ายประสาทเทียมมาประยุกต์ใช้กันอย่างกว้างขวาง แต่ไม่เป็นที่นิยมในการนำมาใช้หาเส้นทางเท่าที่ควร เนื่องจากการเลือกตัวแปรในการตัดสินใจนั้นทำได้ยากกว่า ฉะนั้นเมื่อพิจารณาถึงประสิทธิภาพของอัลกอริทึม ซึ่งอยู่ในรูปของฟังก์ชัน  $O(n)$  แล้วจะให้ผลที่ไม่ดีนัก Kamoun F และ Mehmet Ali [13] นำเสนอการใช้โครงข่ายประสาทเทียมเพื่อประยุกต์กับงานเครือข่ายโดยกล่าวถึงเส้นทางเลือกที่เหมาะสมมากกว่า แต่ถ้ระบบโครงข่ายซับซ้อนมากขึ้นการหาเส้นทางจะทำได้ยากขึ้น นอกจากนี้ผลของการทดลองยังไม่มีการเสนอเปรียบเทียบกับงานวิจัยอื่น และไม่ได้กล่าวถึงผลเท่าไรนัก เป็นเพียงการเสนอแนวคิดในการประยุกต์มากกว่า

### 3.1.4 อัลกอริทึมจีนิติก (Genetic Algorithm)

อัลกอริทึมนี้เป็นการจัดข้อมูลให้เข้าคู่กัน ดังนั้นการหาเส้นทางที่สั้นที่สุดของระบบทำให้ไม่ค่อยได้ผลดี เนื่องจากต้องคอยปรับเปลี่ยนข้อมูลเพื่อให้ข้อมูลเข้ากลุ่มกัน นักวิจัยชื่อ Gen M. และ Runwei Cheng [14] เสนอการหาเส้นทางที่สั้นที่สุดโดยใช้อัลกอริทึมจีนิติก ซึ่งทำการทดลองไว้ แต่ไม่ได้เปรียบเทียบข้อมูลกับงานวิจัยอื่น ๆ เนื่องจากผลของการทดลองยังไม่สมบูรณ์ สามารถทดสอบได้เพียง 6-10 โหนดเท่านั้น ถ้าคิดเป็นเปอร์เซ็นต์ความน่าเชื่อถือจะได้เส้นทางที่สั้นที่สุดที่เชื่อถือได้ ประมาณ 92 % เท่านั้น

### 3.1.5 อัลกอริทึมฟัซซี่ (Fuzzy Algorithm)

อัลกอริทึมนี้เป็นการจัดการข้อมูลให้เป็นกลุ่ม มีการเปลี่ยนหรือสลับข้อมูล เป็นการค้นหาเส้นทางที่สั้นที่สุดโดยตรง ต่างกับวิธีการค้นหาแบบลึก (Depth-First Search) งานวิจัยของ Chandwani M และ Chaudhari N.S [15] เสนอวิธีการประยุกต์ฟัซซี่ในการหาเส้นทางที่สั้นที่สุด แต่ไม่ได้เปรียบเทียบข้อมูลกับงานวิจัยอื่น ๆ เช่นกัน

### 3.1.6 อัลกอริทึมเวกเตอร์ระยะทาง (Distance-Vector Algorithm)

อัลกอริทึมนี้เป็นการหาเส้นทางที่มีการใช้จำนวนโหนดที่วิ่งผ่าน (Hop Count) น้อยที่สุด ดังนั้นอัลกอริทึมนี้อาจไม่เหมาะสมในการส่งข้อมูล เนื่องจากถ้าเส้นทางนั้นถูกใช้ส่งข้อมูลจำนวนมากโดยตลอด เส้นทางนั้นอาจจะเกิดการสูญเสียข้อมูลจำนวนมาก งานวิจัยของ Shree Murthy และ J.J Garcia-Luna-Aceves [16] เป็นการนำมาซึ่งการทำงานแบบ RIP (Routing Information Protocol) แต่วิธีการดังกล่าวเป็นอัลกอริทึมที่ง่ายต่อการเลือกเส้นทาง แต่เส้นทางที่ได้ไม่จำเป็นต้องได้เส้นทางที่เหมาะสม และวิธีการดังกล่าวนี้มีข้อจำกัดเรื่องจำนวนโหนด เพราะอัลกอริทึมสามารถบริการได้เพียง 15 โหนดเท่านั้น

### 3.1.7 อัลกอริทึมสถานะการเชื่อมต่อ (Link-State Algorithm)

อัลกอริทึมนี้ใช้ในการหาเส้นทางที่มีค่าน้ำหนักที่น้อยที่สุด โดยที่ค่าน้ำหนักที่มีการเปลี่ยนแปลง (Dynamic Routing) ตลอดเวลา กระบวนการพิจารณาจึงสามารถนำค่าตัวแปรหลายค่าที่มีความสำคัญกับค่าน้ำหนักของเส้นทางมาคิดหาเส้นทางที่สั้นที่สุด เช่น ค่าความจุของเส้นทาง หรือ ค่าเวลา เป็นต้น งานวิจัยของ Paolo Narvaez และ Kai-Yeung Siu [17] ได้พยายามทำการเลือกเส้นทางที่สั้นที่สุดเป็นผลสำเร็จ ซึ่งการวิจัยนี้เป็นการนำมาสู่การหาเส้นทางแบบ OSPF (Open Shortest Path First) แต่การพิจารณายังคงต้องใช้วิธีการค้นหาทุกเส้นทาง ซึ่งทำให้เสียเวลา และ ถ้ากรณีเกิดเหตุขัดข้องของการเชื่อมต่อ จะพบว่าเส้นทางที่ได้ในครั้งใหม่อาจเป็นเส้นทางที่ไม่สั้นที่สุดเสมอตามต้องการ

อัลกอริทึมในการเลือกกำหนดเส้นทางหลาย ๆ อัลกอริทึมที่กล่าวแล้วนั้น ต่างมีพื้นฐานในการหาเส้นทางที่มีต้นทุนต่ำที่สุด ซึ่งใช้ค่าองค์ประกอบในการหาต้นทุนในการพิจารณา แตกต่างกัน ตัวอย่างโครงข่ายที่ประกอบด้วยค่าองค์ประกอบในการหาต้นทุน โดยที่ค่าองค์ประกอบในการหาต้นทุนนั้นจะเป็นค่าที่นำมาใช้พิจารณาตามแต่วัตถุประสงค์ที่เหมาะสมในแต่ละโครงข่าย โดยอาจจะไม่ได้หมายความว่าเส้นทางที่สั้นที่สุด ตัวอย่างเช่น

- ค่าเชื่อมต่อที่เป็นส่วนกลับของค่าความจุของเส้นทาง โดยกำหนดให้ค่าเชื่อมต่อมีค่ามากในเส้นทางที่มีค่าความจุน้อย โดยจุดประสงค์ต้องการส่งข้อมูล ไปยังเส้นทางที่มีค่าความจุมากก่อน ถ้าแต่ละเส้นทางมีค่าเชื่อมต่อเท่ากันก็จะพิจารณาที่จำนวน Hop ที่น้อยกว่าแทน

- ค่าเชื่อมต่อเป็นสัดส่วนตรงกับค่าเฉลี่ยของความล่าช้า (Average Packet Delay) ซึ่งรวมถึงการล่าช้าในการเข้าคิวใน Switch Buffer และความล่าช้าในการส่งผ่านข้อมูลของแต่ละเส้นทาง เพราะเส้นทางที่สั้นที่สุดก็ต้องเป็นเส้นทางที่มีความเร็วในการส่งจากต้นทาง ไปยังปลายทางน้อยที่สุดด้วย

- ค่าเชื่อมต่อเป็นสัดส่วนตรงกับค่าความคับคั่ง (Congestion) เช่น ค่าทราฟฟิก (Traffic) [18] ดังนั้นเส้นทางที่สั้นที่สุดก็ย่อมหลีกเลี่ยงเส้นทางที่คับคั่งเช่นกัน

## 3.2 การประยุกต์ใช้อัลกอริทึมเอสตาร์ในการหาเส้นทางที่มีต้นทุนต่ำ

บนพื้นฐานในการหาเส้นทางที่มีต้นทุนต่ำที่สุด ซึ่งใช้ค่าองค์ประกอบในการหาต้นทุนในการพิจารณาตามหัวข้อ 3.1 บทวิจัยนี้ประยุกต์ใช้ อัลกอริทึม เอสตาร์ ในการหาเส้นทางที่มีต้นทุนต่ำ เพื่อเป็นเส้นทางสำรอง สำหรับโครงข่ายไอพีของของ บริษัท ทีโอที จำกัด (มหาชน) ดังรายละเอียดในหัวข้อต่อไปนี้

### 3.2.1 หลักการ ของ อัลกอริทึมเอสตาร์

อัลกอริทึม เอสตาร์ เป็น Graph Search Algorithm โดย Peter Hart, Nils Nilsson และ Bertram Raphael เป็นผู้คิดค้นเป็นคนแรกในปี ค.ศ. 1968 อัลกอริทึม เอสตาร์ เป็นที่นิยมใช้ในการหาทางเดินของระบบปัญญาประดิษฐ์ (Artificial Intelligence, AI) สำหรับการหาทางเดินในเกมต่าง ๆ โดยใช้ในการเขียน โปรแกรมเพื่อค้นหาคู่ต่อสู้ในแผนที่ของเกม ซึ่งใช้ค่าการตัดสินใจ (Heuristic) ที่มีการเปลี่ยนแปลงตลอดเวลาบนเส้นทางที่น่าจะเป็นไปได้ หรือใช้ระยะทางเป็นเกณฑ์ในการกำหนดค่าของต้นทุนในการค้นหาศัตรูหรือหาคู่ต่อสู้ โดยอัลกอริทึมนี้จะประกอบด้วยสามส่วน [19] ที่ทำให้เส้นทางต่าง ๆ นั้นถูกตัดสินใจเลือกใช้ หรือเลือกไม่ใช้ สามส่วนนั้นประกอบด้วย

1. Heuristic คือ ค่าสำหรับการตัดสินใจในการเดินทางผ่านจุดใด ๆ โดยกำหนดให้เกณฑ์นั้นแสดงค่า เป็นตัวเลข
2. Cost คือ ค่าใด ๆ ที่บอกค่าใช้จ่าย หรือระยะทางที่ใช้ เวลาในการเดินทางผ่านหรือค่าใด ๆ ที่ทำให้เส้นทางนั้นเหมาะสมต่อการเลือกใช้หรือไม่ โดยให้เกณฑ์เป็นค่าตัวเลข
3. Priority คือ ค่าที่ได้จากค่าการตัดสินใจรวมกับค่าใช้จ่าย (Heuristic + Cost) จะได้เป็นค่า Priority ออกมา โดยจะเป็นตัวบอกว่าเส้นทางดังกล่าวนี้ เหมาะสมที่จะผ่านหรือไม่ โดยวัดจากตัวเลขที่แสดง

การเลือกเส้นทางที่มีต้นทุนต่ำ ซึ่งมีพื้นฐาน คือ สมมุติว่ามีสมการ  $f(s)$  ค่าของ  $f(s)$  ที่น้อยที่สุดจะถือว่าเป็น โหนดที่ดีที่สุด สมการนี้จะขึ้นอยู่กับข้อมูลซึ่งมาจากขอบเขต (Domain) ของปัญหา โดยค่าที่น้อยที่สุดมักจะถูกเลือก เพื่อดำเนินการค้นหาเป้าหมาย เมื่อพบเป้าหมาย หรือ โหนดที่มีค่าน้อยสุดแล้ว ให้เลือกเพียง โหนดเดียวกระจายต่อไป และหยุดการค้นหาเมื่อเจอเป้าหมายนั้น ๆ อัลกอริทึมนี้เป็นลักษณะหนึ่งของการค้นหาข้อมูลแบบกราฟ (Graph search) ชนิด Best-First-Search ซึ่งเป็นการค้นหาเส้นทางที่นำเอาข้อดีของทั้ง Depth First และ Breadth First มารวมกันเป็นวิธีการเดียวโดยที่แต่ละขั้นของการค้นหาเส้นทางนั้นจะเลือกเอาโหนดที่ดีที่สุด (Most Promising) โดยอาศัยฮิวริสติกฟังก์ชัน สำหรับอัลกอริทึมเอสตาร์มีความสัมพันธ์ของ 2 ฟังก์ชัน ในรูปที่ 3.1 คือ

$$f(s) = g(s) + h(s) \quad (3.1)$$

โดยในโครงข่ายนี้ กำหนดให้

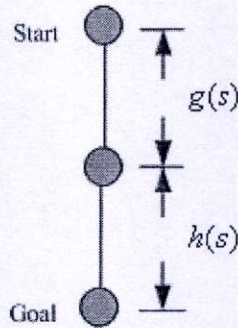
$f(s)$  คือ ค่าต้นทุนที่ต่ำที่สุด ที่ถูกกำหนดให้อยู่ในแต่ละขอบในโครงข่าย

$h(s)$  คือ ค่าที่พิจารณาตั้งแต่โหนดปัจจุบัน ไปยังโหนดปลายทาง ได้จากค่าการวัดระยะทางจากโหนดปัจจุบัน ไปยังโหนดปลายทาง

$g(s)$  คือ ค่าที่พิจารณาตั้งแต่โหนดต้นทาง มายังโหนดปัจจุบัน โดยพิจารณาจากค่าผลรวมของดังประกอบเสริมที่กำหนดหน้าหน้าแล้ว ดังสมการ 3.2

$$g(s) = (T * W_1) + (D * W_2) + (L * W_3) \quad (3.2)$$

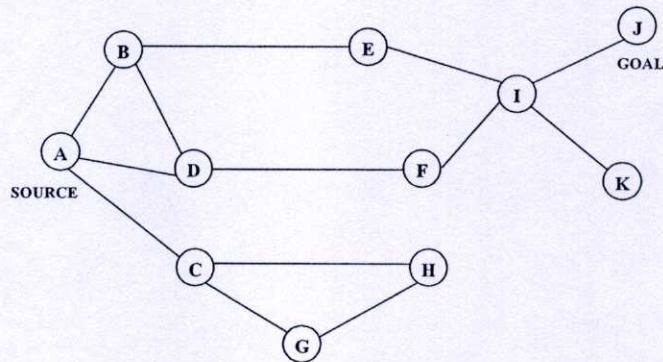
เมื่อ  $W_1, W_2$  และ  $W_3$  คือ ค่าถ่วงน้ำหนัก มีค่าไม่เท่ากันขึ้นอยู่กับการ  
 $T$  คือ ค่าปริมาณ ทราฟฟิก ของโครงข่ายไอพี (Mbps)  
 $D$  คือ ระยะทางความยาวของเส้นใยแก้วนำแสง (Km)  
 $L$  คือ ค่าการสูญเสียในใยแก้วนำแสง (dB)



รูปที่ 3.1 รูปแบบความสัมพันธ์ของสมการ อัลกอริทึม เอสตาร์

### 3.2.2 ตัวอย่าง การพิจารณา อัลกอริทึม เอสตาร์

ตัวอย่าง จากกราฟ รูปที่ 3.2 ที่นำมาพิจารณาหาเส้นทางที่ต้นทุนต่ำที่สุดตามอัลกอริทึม เอสตาร์



รูปที่ 3.2 ตัวอย่างกราฟโครงข่าย

สมมติให้

$N$  คือ จำนวนโหนด

$W_i$  คือ น้ำหนักของ เส้นเชื่อมโง (Hop) ที่  $i$  โดย  $i$  มีค่าตั้งแต่  $1 \sim M$

$P$  คือ จำนวนเส้นทางเดินของข้อมูล

ขั้นตอนการหาเส้นทางที่ต้นทุนต่ำที่สุดตาม อัลกอริทึม เอสตาร์ แสดงตามลำดับดังนี้

1. ใส่ค่าถ่วงน้ำหนักในเมตริกซ์ที่มีขนาด  $= N * N$

2. พิจารณา ค่าต้นทุนของเส้นทางตามลำดับ

โดยกำหนดให้  $g(s) + h(s)$  เป็นต้นทุนของ อัลกอริทึม เอ- สตาร์ และ แทนด้วย  $W(\delta)$  สมมุติให้  $P = 2$  ดังนั้นค่า  $\delta$  ของเส้นทาง จะเป็นลำดับที่ 1 และ 2 คือ

จากโหนด A ถึง โหนด B มีค่า เป็น  $\delta_1 = P_{AB}$  มีค่าน้ำหนักหรือค่าต้นทุนเป็น  $W(\delta_1)$

จากโหนด A ถึง โหนด D มีค่า เป็น  $\delta_2 = P_{AD}$  มีค่าน้ำหนักหรือค่าต้นทุน เป็น  $W(\delta_2)$

3. เปรียบเทียบค่า  $W(\delta_1)$  และ  $W(\delta_2)$  ที่กำหนดไว้ในข้อ 2 ค่าที่น้อยกว่าจะเป็น ทางเดินของข้อมูลที่ต้องการ ดังนั้น

$$f(s) = \text{Min}W(\delta) \text{ ซึ่งอาจมีค่าเป็น } W(\delta_1) \text{ หรือ } W(\delta_2) \quad (3.3)$$

4. ให้ทำการเก็บค่า  $W(\delta)$  ที่ไม่ถูกเลือกไว้ในตาราง Database โดยเรียงตามสมการ 3.4

$$f(s)_{old} = \text{Min}(W(\delta_{noselect})) \quad (3.4)$$

5. กรณีไม่พบโหนด ปลายทาง ให้ พิจารณาดังนี้

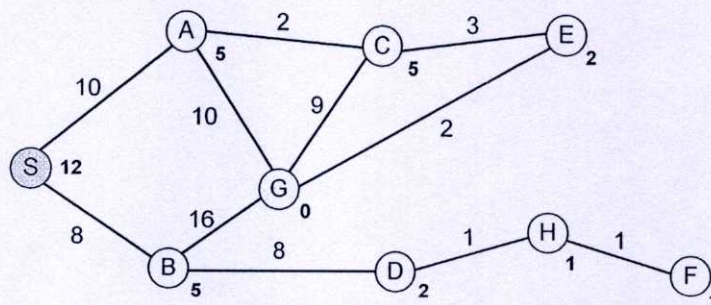
ก. พิจารณาเส้นทางจากต้นทางถึงปลายทาง

$$f(s_i)_{new} = f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_n)) \quad (3.5)$$

ข. เลือกเส้นทางที่มีค่าต่ำที่สุด

$$f(s) = \text{Min}[f(s)_{old}, f(s)_{new}] \quad (3.6)$$

ตัวอย่าง 3.2 พิจารณาการหาเส้นทางที่มีต้นทุนต่ำที่สุดตามรูปที่ 3.3 ด้วย อัลกอริทึม เอสตาร์ มีขั้นตอนดำเนินการดังนี้



รูปที่ 3.3 ตัวอย่างกราฟที่มีค่าถ่วงน้ำหนัก และค่าฮิวริสติก

1. เส้นทางที่พิจารณาได้ เริ่มต้นมีเพียง 2 เส้นทาง คือ SA และ SB

$$\begin{aligned}
 2. \text{ หาค่า } f(s) &= \text{Min}[W(\delta_1), W(\delta_2), \dots, W(\delta_z)] \\
 &= \text{Min}[W(SA), W(SB)] \\
 &= [H5C10F15, H5C8F13] \\
 &= 13
 \end{aligned}$$

3. เก็บค่า  $f(s)_{old}$  ใน List  $f(s)_{old}$

$$f(s)_{old} = W(SA) = 15$$

ถ้า Situation ยังไม่ถึงโหนดปลายทาง ให้ดำเนินการต่อไป

4. หาค่า  $f(s_i)_{new} = f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_z))$

$$\begin{aligned}
 \text{ก. หา } f(s_i)_{new} &= 13 + \text{Min}[W(BG), W(BD)] \\
 &= 13 + [H0C16F16, H2C8F10] \\
 &= 13 + 10 \\
 &= 23
 \end{aligned}$$

$$\begin{aligned}
 \text{ข. หา } f(s) &= \text{Min}[f(s)_{old}, f(s)_{new}] \\
 &= \text{Min}[15, 23] \\
 &= 15
 \end{aligned}$$

5. หาค่า  $f(s_i)_{new} = f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_z))$

$$\begin{aligned}
 \text{ก. พิจารณา } f(s_i)_{new} &= f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_z)) \\
 f(s_i)_{new} &= 15 + \text{Min}[W(AG), W(AC)] \\
 &= 15 + [H0C10F10, H3C2F5] \\
 &= 15 + 5 \\
 &= 20
 \end{aligned}$$

$$\begin{aligned}
 \text{ข. พิจารณา } f(s) &= \text{Min}[f(s)_{old}, f(s)_{new}] \\
 &= [23, 20] \\
 &= 20
 \end{aligned}$$

$$\begin{aligned}
 \text{ค. พิจารณา } f(s_i)_{new} &= f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_z)) \\
 f(s_i)_{new} &= 20 + \text{Min}[W(CG), W(CE)] \\
 &= 20 + [H0C9F9, H2C3F5] \\
 &= 20 + 5 \\
 &= 25
 \end{aligned}$$

$$\begin{aligned} \text{ง. พิจารณา} \quad f(s) &= \text{Min}[f(s)_{old}, f(s)_{new}] \\ &= [30, 25] \\ &= 25 \end{aligned}$$

$$\begin{aligned} \text{จ. พิจารณาต่อ} \quad f(s_i)_{new} &= f(s_i) + \text{Min}(W(\delta_1), W(\delta_2), \dots, W(\delta_z)) \\ f(s_i)_{new} &= 6 + \text{Min}[W(EG)] \\ &= 25 + [H0C2F2] \\ &= 25 + 2 \\ &= 27 \end{aligned}$$

ดังนั้นเส้นทางที่ต้นทุนต่ำที่สุดคือ  $S \rightarrow A \rightarrow C \rightarrow E \rightarrow G$

### 3.2.3 การปรับอัลกอริทึมเอสตาร์ให้เหมาะสมกับโครงข่ายไอพีที่ใช้ทดลอง

ในการหาเส้นทางโดยการใช้ อัลกอริทึมเอสตาร์ นั้น ในบางครั้งจะไม่ได้เส้นทางที่ต่ำที่สุดเสมอไป ในวิทยานิพนธ์นี้จึงได้ปรับวิธีการหาเส้นทางของ อัลกอริทึมเอสตาร์ เพื่อให้สามารถใช้กับโครงข่ายที่ใช้ในการทดลองนี้ได้อย่างเหมาะสม เนื่องจากโครงข่ายที่ถูกต่อใช้งานเป็นโครงข่ายแบบไฮแมงมุม จึงมีบางเส้นทางเป็นทางเดินลัด ในบทนี้ได้กำหนดเงื่อนไขใหม่ โดยไม่ให้เกิดการหาเส้นทางซ้ำทางเดิมที่ทราฟฟิกเคยเดินทางผ่านไปแล้วตามขั้นตอนของอัลกอริทึมเอสตาร์ และกำหนดให้มีการเปรียบเทียบต้นทุนของเส้นทางระหว่างโหนดต้นทางและปลายทางก่อนตัดสินใจในทุก ๆ เส้นทางที่ทราฟฟิกจะสามารถเดินทางผ่านไปได้ ยกตัวอย่างโครงข่ายอย่างง่ายในรูปที่ 3.4 ซึ่งประกอบด้วย 3 โหนด โดยกำหนดให้

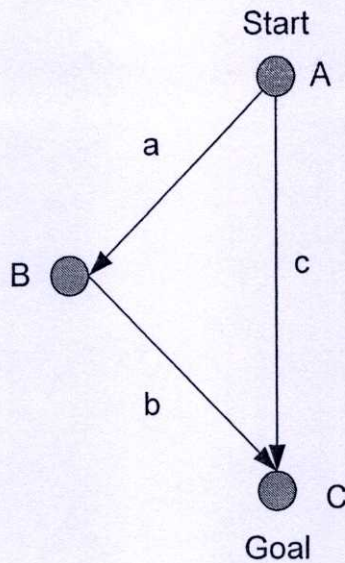
A เป็นโหนด ต้นทาง

C เป็นโหนด ปลายทาง

B เป็นโหนด ใดที่ทราฟฟิกสามารถเดินทางผ่าน

A C เป็นเส้นทางเดินลำดับที่ 1

A B C เป็นเส้นทางเดินลำดับที่ 2



รูปที่ 3.4 โครงข่ายตัวอย่างที่แสดงการปรับอัลกอริทึมเอสตาร์

จากรูปที่ 3.4 ปรับอัลกอริทึมเอสตาร์โดยกำหนดให้มีการเปรียบเทียบค่าต้นทุนของทั้ง 2 เส้นทาง ได้ค่า  $g(s)$  ดังสมการที่ 3.7

$$g(s) = (a+b) - c \quad (3.7)$$

$g(s)$  คือ ค่าต้นทุนรวมจากต้นทางถึงปลายทาง

$a, b, c$  คือ ค่าต้นทุนของแต่ละเส้นทาง

โดยให้เป็นไปตามเงื่อนไขดังนี้

ถ้า  $a+b = c$

$$g(s) = \overline{AB} \text{ และ } \overline{BC} \text{ หรือ } \overline{AC}$$

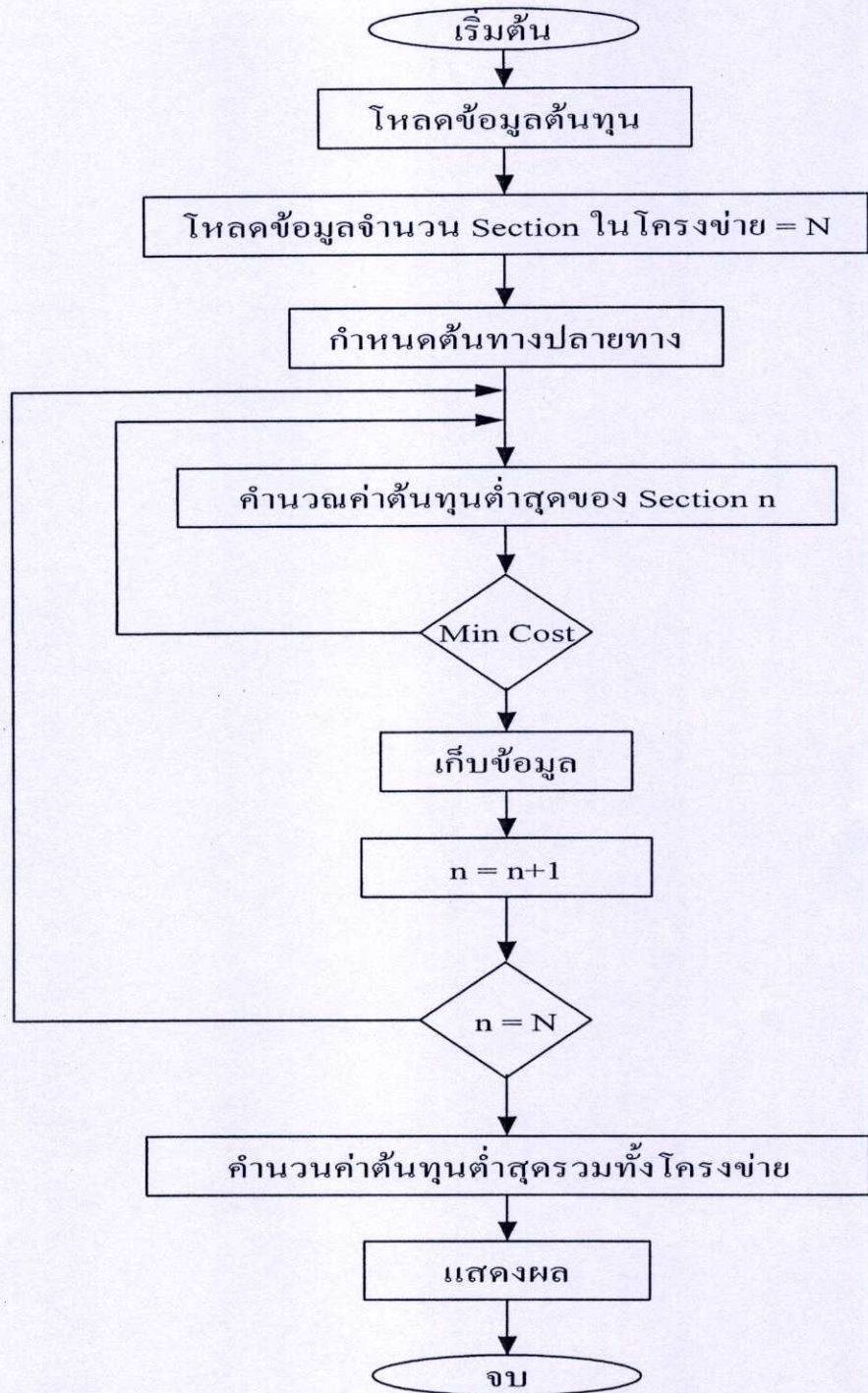
ถ้า  $a+b > c$

$$g(s) = \overline{AC}$$

ถ้า  $a+b < c$

$$g(s) = \overline{AB} \text{ และ } \overline{BC}$$

สำหรับขั้นตอนการทำงานของ อัลกอริทึม เอสตาร์ที่ปรับแล้วแสดงดังไดอะแกรมในรูปที่ 3.5



รูปที่ 3.5 ไดอะแกรมแสดงการทำงานของ อัลกอริทึมเอสตาร์ที่ปรับปรุง

## บทที่ 4

# ทฤษฎีโครงข่ายประสาทเทียมที่เกี่ยวข้องในการวิจัย

ความคับคั่ง (Congestion) ของข้อมูล หรือ ทราฟฟิก (Traffics) เป็นสาเหตุที่สำคัญที่จะทำให้ประสิทธิภาพโดยรวมของระบบลดลง ในบทวิจัยนี้ จึงทำการตรวจสอบปริมาณทราฟฟิก และทำการพยากรณ์ค่าความคับคั่งข้อมูลทราฟฟิกแต่ละโหนด เพื่อนำไปใช้เป็นองค์ประกอบของค่าต้นทุน (Cost) ในการกระจายข้อมูลสู่เราเตอร์ ขั้นตอนในการพยากรณ์ เริ่มตั้งแต่เก็บรวบรวมข้อมูลทราฟฟิกแต่ละโหนด การหาตัวแบบการพยากรณ์ที่เหมาะสม และการวิเคราะห์ผล รายละเอียดของแต่ละขั้นตอนแสดงในบทที่ 5 ตัวแบบการพยากรณ์ในบทวิจัยนี้ ใช้อัลกอริทึมของโครงข่ายประสาทเทียม แบบแพร่ย้อนกลับ (Backpropagation Neural Networks) ดังจะได้กล่าวถึงทฤษฎีโครงข่ายประสาทเทียมโดยย่อในหัวข้อต่อไปนี้

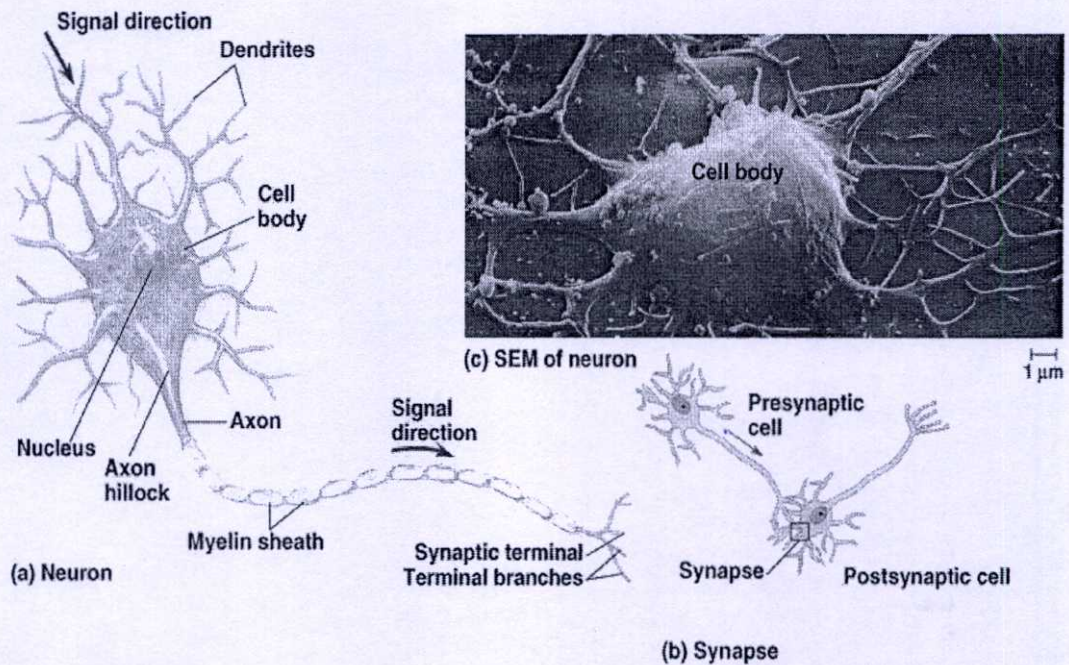
### 4.1 ทฤษฎีโครงข่ายประสาทชีวภาพ

โครงข่ายประสาทเทียม (Artificial Neural Networks) เป็นแขนงหนึ่งของระบบปัญญาประดิษฐ์ (Artificial Intelligence) โดยมีโครงสร้างและการทำงานคล้ายกับระบบประสาทของมนุษย์ แนวความคิดทฤษฎีโครงข่ายประสาทเทียม ได้แก่ การนำข้อดีของระบบประสาทมาใช้ในการทำงานร่วมกับความสามารถของคอมพิวเตอร์ เช่น ความสามารถเรียนรู้จากประสบการณ์ การจำแนกลักษณะสิ่งของที่มีคุณลักษณะใกล้เคียงกัน และการแปลความหมายของสัญลักษณ์และภาพ และงานอีกประเภทหนึ่งที่น่าวิธีโครงข่ายประสาทเทียมมาใช้ คือ งานด้านการพยากรณ์ Delurgio (1998) กล่าวว่า วิธีโครงข่ายประสาทเทียมสามารถเรียนรู้รูปแบบของระบบที่มีความซับซ้อนสูงและหาค่าพยากรณ์จากข้อมูลในอดีตได้ดีกว่าวิธีการพยากรณ์แบบดั้งเดิม (Traditional Statistical Method)

โดยธรรมชาติเซลล์ประสาทของสิ่งมีชีวิตแบ่งเป็นหลายประเภท โดยแบ่งตามตำแหน่งและหน้าที่ เช่น เซลล์ประสาทกล้ามเนื้อ เซลล์ประสาทในสมอง เซลล์ประสาทที่ลิ้น และจมูก เป็นต้น โครงสร้าง และ ส่วนประกอบของเซลล์ประสาทโดยทั่วไปแสดงได้ดังรูปที่ 4.1 โดยรูป (a) แสดงโครงสร้างโครงข่ายประสาทชีวภาพ รูป (b) แสดงส่วนที่ติดต่อกับเซลล์อื่นๆ (Synapse) และรูป (c) แสดงรูปโครงข่ายประสาทผ่านกล้องจุลทรรศน์อิเล็กตรอนแบบสแกน (Scanning Electron Microscope) โครงสร้างเซลล์ประสาทชีวภาพประกอบด้วย 4 ส่วนคือ ตัวเซลล์ประสาทหรือนิวรอล ซึ่งมีนิวเคลียสอยู่ตรงกลาง รอบ ๆ เซลล์ประสาทมียื่นออกไปเพื่อรับ และส่งสัญญาณจากเซลล์ประสาทอื่น ๆ เรียกว่า แอกซอน (Axon) และที่ปลายกิ่งจะแตกออกเป็นก้านย่อย ๆ เรียกว่าเดน

ไดรต์ (Dendrite) รอยต่อระหว่างก้านของเซลล์ประสาทที่ต่างกันเรียกว่าไซแนป (Synapse) ซึ่งสามารถเปลี่ยนค่าความต้านทานทางไฟฟ้าได้ตามสัญญาณที่ส่งระหว่างเซลล์ประสาท สำหรับการส่งสัญญาณระหว่างเซลล์ประสาททำได้โดยการถ่ายทอดสารประกอบโซเดียม และ โพตัสเซียม

นักวิจัย ชื่อ ฮอดกิน (Hodkin) และ ฮักลีย์ (Huxley) ซึ่งได้รับรางวัลโนเบลทางชีววิทยาได้ค้นพบว่าการไหลของสารประกอบโซเดียม และ โพตัสเซียมของเซลล์ประสาทของปลาหมึกทำให้เกิดความต่างศักย์อยู่ระหว่าง 50 ถึง 70 มิลลิโวลต์ จากผลการศึกษาดังกล่าวทำให้เราสามารถจำลองการทำงานของเซลล์ประสาทโดยอาศัยวงจรอิเล็กทรอนิกส์ได้ [20]

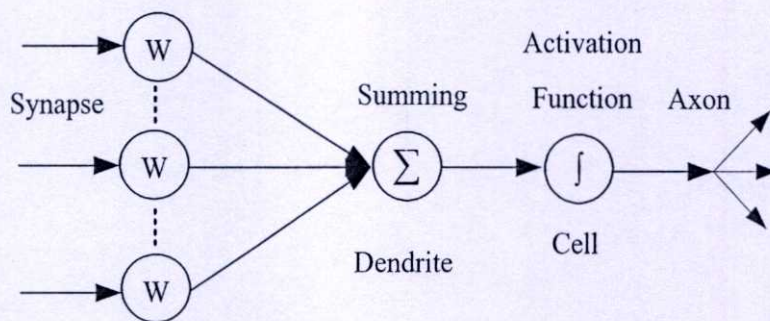


รูปที่ 4.1 โครงสร้างและส่วนประกอบของเซลล์ประสาทชีวภาพ

การทำงานของเซลล์ประสาท จะเริ่มจากเซลล์นิวรอลรับอินพุต (Input) จากเซลล์นิวรอลตัวอื่น ผ่านจุดเชื่อมโยงที่เรียกว่า ไซแนป (Synapse) สัญญาณข้อมูลจาก ไซแนป จะถูกส่งผ่านเข้าทาง เดนไดรต์ (Dendrite) ซึ่งเป็นส่วนที่ทำหน้าที่เป็นตัวรับข้อมูลอินพุตเข้าสู่ตัวเซลล์ (Cell Body) สัญญาณอินพุตจะถูกประมวลผลบางประการตามกระบวนการที่เกิดขึ้นภายในเซลล์ และส่งสัญญาณเอาต์พุต (Output) ออกทางส่วนของเซลล์ที่เรียกว่า แอกซอน (Axon) สัญญาณดังกล่าวจะผ่านข้ามไซแนป ด้วยเงื่อนไขบางประการ ซึ่งเป็นส่วนอินพุตของเซลล์อื่นต่อไป โครงข่ายประสาทเทียมมีคุณลักษณะคล้ายกับสมองในแง่ที่สามารถรวบรวมความรู้ (Knowledge) ได้โดยการผ่านกระบวนการเรียนรู้ (Learning Process) และความรู้เหล่านั้นจะถูกจัดเก็บอยู่ในโครงข่ายด้วยค่าน้ำหนัก (Weight) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่

#### 4.1.1 ลักษณะทั่วไปของโครงข่ายประสาทเทียม

1. การประมวลผลต่าง ๆ เกิดขึ้นในหน่วยประมวลผลย่อยเรียกว่า โหนด (Node) ซึ่งจำลองลักษณะการทำงานมาจากตัวเซลล์
2. การส่งสัญญาณ (Signal) ต่าง ๆ ระหว่างโหนด โดยส่วนที่เชื่อมต่อกัน (Connection Link) จำลองมาจากการเชื่อมต่อกันของเดนไดรต์ และแอกซอนในระบบประสาทของมนุษย์
3. แต่ละการเชื่อมต้อมีค่าน้ำหนัก (Weight) ที่ต่างกัน ขึ้นอยู่กับอิทธิพลที่โหนดจะได้รับจากโหนดอื่น ๆ ซึ่งจำลองมาจากไซแนปส์ ค่าน้ำหนักที่ได้จะทำหน้าที่เปรียบเสมือนความรู้ที่รวบรวมไว้ใช้ในการแก้ปัญหาเฉพาะอย่างของมนุษย์
4. ภายในโหนดจะมีฟังก์ชันกำหนดสัญญาณเอาต์พุต ซึ่งถูกเรียกว่า Activation Function หรือ Transfer Function ซึ่งทำหน้าที่เปรียบเสมือนกระบวนการทำงานในตัวเซลล์ประสาท ลักษณะของโหนดภายในโครงสร้างของเซลล์ประสาทเทียมสามารถสร้างแบบจำลองทางคณิตศาสตร์ได้ แสดงดังรูปที่ 4.2



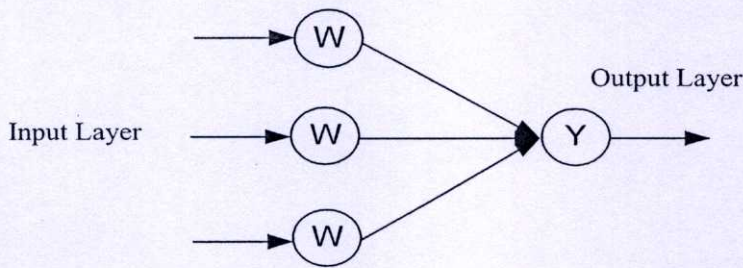
รูปที่ 4.2 ลักษณะของ โหนดภายใน โครงสร้างของเซลล์ประสาทเทียม

#### 4.1.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมประกอบด้วยเซลล์ประสาทเทียมหรือโหนดจำนวนมากที่เชื่อมต่อกัน ซึ่งการเชื่อมต้อมีแนวโน้มที่จะแบ่งออกเป็นกลุ่มย่อยเรียกว่า ชั้น (Layer) ในชั้นแรกจะเป็นชั้นนำข้อมูลเข้า เรียกว่าชั้นอินพุต (Input Layer) ส่วนชั้นสุดท้ายเรียกว่าชั้นเอาต์พุต (Output Layer) และชั้นที่อยู่ระหว่างชั้นอินพุต และชั้นเอาต์พุต เรียกว่า ชั้นฮิดเดน (Hidden Layer) ซึ่งชั้นฮิดเดนจะมีมากกว่า 1 ชั้นก็ได้ โครงข่ายประสาทเทียมสามารถแบ่งประเภทตามจำนวนชั้นของโครงข่ายได้ 2 แบบคือ โครงข่ายชั้นเดียว (Single Layer) และโครงข่ายหลายชั้น (Multi Layer) การนับชั้นของโครงข่ายจะนับเฉพาะชั้นที่มีการประมวลผลเท่านั้น หรืออาจกล่าวได้ว่า จำนวนชั้นของโครงข่ายประสาทเทียมคือ จำนวนชั้นฮิดเดนรวมกับจำนวนชั้นเอาต์พุต [21]

#### 4.1.2.1 โครงข่ายชั้นเดียว

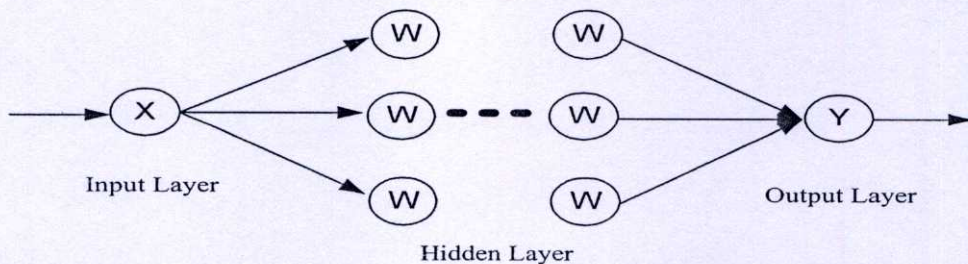
เป็นโครงข่ายประสาทเทียมอย่างง่ายที่มีเพียงชั้นอินพุต และชั้นเอาต์พุตเท่านั้น โหนดในชั้นอินพุตจะทำหน้าที่ในการรับข้อมูลอินพุต (Input Value) แล้วส่งข้อมูลอินพุตผ่านเส้นเชื่อมโยงต่าง ๆ ไปให้โหนดในชั้นเอาต์พุต ความเข้มของสัญญาณหรือปริมาณข้อมูลที่นำเข้าสู่โหนดในชั้นเอาต์พุตจะขึ้นอยู่กับค่าน้ำหนักที่อยู่บนเส้นเชื่อมโยง โหนดในชั้นเอาต์พุตทำหน้าที่นำข้อมูลที่รับมา มาทำการคำนวณโดยใช้ฟังก์ชันทางคณิตศาสตร์ที่เรียกว่า ทรานเฟอร์ฟังก์ชัน ที่เหมาะสมกับปัญหา แล้วส่งผลลัพธ์ที่ได้ออกมาเป็นข้อมูลเอาต์พุต ตัวอย่างโครงข่ายแบบชั้นเดียว เช่น เพอเซปตรอนอย่างง่าย (Simple Perceptron) โครงข่ายโฮปฟิลด์ (Hopfield Networks) เป็นต้น ลักษณะโครงสร้างของโครงข่ายแบบชั้นเดียวแสดงดังรูปที่ 4.3



รูปที่ 4.3 ลักษณะโครงสร้างของโครงข่ายแบบชั้นเดียว

#### 4.1.2.2 โครงข่ายแบบหลายชั้น

โครงข่ายแบบหลายชั้นเป็นโครงข่ายที่มีชั้นฮิดเดนตั้งแต่ 1 ชั้นขึ้นไป ส่วนใหญ่จะถูกนำไปใช้ในกรณีที่มีปัญหาซับซ้อนสูง ซึ่งโครงข่ายแบบชั้นเดียวไม่เพียงพอในการแก้ปัญหาได้ จึงเพิ่มจำนวนของโหนดที่มีการคำนวณหรือชั้นฮิดเดนให้กับโครงข่าย ตัวอย่างของโครงข่ายแบบหลายชั้น เช่น โครงข่ายการแพร่แบบย้อนกลับ (Backpropagation) เซลฟออร์แกนไนซิงแมปส์ (Self Organizing Maps) เคาน์เตอร์พรอพเพเกชัน (Counterpropagation) เป็นต้น ลักษณะโครงสร้างของโครงข่ายแบบหลายชั้นแสดงดังรูปที่ 4.4



รูปที่ 4.4 ลักษณะโครงสร้างของโครงข่ายแบบหลายชั้น

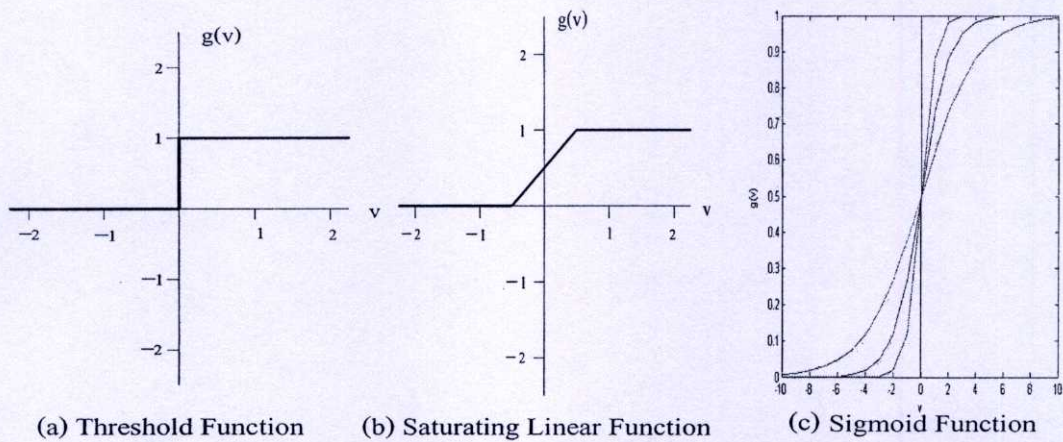
## 4.2 ฟังก์ชันกระตุ้นความสนใจของโครงข่ายประสาทเทียม

ในการคำนวณค่าเอาต์พุตของโครงข่ายประสาทเทียมจะต้องมีปัจจัยสำคัญอีกอย่างหนึ่ง คือ ฟังก์ชันกระตุ้นความสนใจ (Activation Function) ซึ่งจะเป็นตัวกำหนดค่าเอาต์พุตที่ได้สามารถแบ่งฟังก์ชันกระตุ้นพื้นฐานออกเป็น 3 ชนิดคือ

4.2.1. ฟังก์ชันเทรลโสด์ (Threshold Function หรือ Hard limit Function หรือ Step Function) แสดงตัวอย่างดังรูปที่ 4.5 (a)

4.2.2. ฟังก์ชันเชิงเส้นอิ่มตัว (Saturating Linear Function หรือ Piecewise-Linear Function) ตัวอย่างของฟังก์ชันแบบนี้ ดังแสดงในรูปที่ 4.5(b)

4.2.3. ฟังก์ชันซิกมอยด์ (Sigmoid function) เป็นฟังก์ชันที่ได้รับความนิยมนำมาใช้เป็นฟังก์ชันแอคทิเวชันของโครงข่ายประสาทเทียมมาก เนื่องจากเป็นฟังก์ชันที่เพิ่มขึ้นอย่างสม่ำเสมอ (Smoothness) ฟังก์ชันซิกมอยด์ จัดเป็นฟังก์ชันแอคทิเวชันที่ไม่เป็นเชิงเส้น [22] ตัวอย่างของฟังก์ชันซิกมอยด์ คือ ฟังก์ชันลอจิสติก (Logistic Function) และฟังก์ชันแทนเจนไฮเพอร์โบลิก (Hyperbolic Tangent Function) แสดงในรูปที่ 4.5 (c)



รูปที่ 4.5 ฟังก์ชันกระตุ้น (Activation Function)

นอกจากฟังก์ชันแอคทิเวชันพื้นฐานทั้ง 3 ชนิด ที่กล่าวข้างต้นในรูปที่ 4.5 แล้วยังมีฟังก์ชันแอคทิเวชัน อื่น ๆ อีกมากที่สามารถนำมาพิจารณาใช้ อย่างไรก็ตามการพิจารณาว่าจะใช้ฟังก์ชันแอคทิเวชัน แบบใดนั้นควรพิจารณาปัจจัยประกอบดังต่อไปนี้

1. ความเป็นเชิงเส้น หรือไม่เชิงเส้น

2. ลักษณะของข้อมูลป้อนเข้า
3. ขอบเขตที่ต้องการและลักษณะของข้อมูลส่งออก
4. ปัจจัยอื่นๆ ที่มีนัยสำคัญต่อผลตอบสนองของนิวรอลหรือโครงข่ายประสาทโดยรวม เช่น ในบางครั้งอาจเลือกใช้ฟังก์ชันสโตซาสติก (Stochastic Function) เป็นฟังก์ชันแบบเชิงเส้น และไม่เป็นเชิงเส้นในโครงข่ายประสาทเดียวกัน ตัวอย่างเช่น โครงข่ายประสาทแบบ RBF (Radial Basis Function) ซึ่งใช้ฟังก์ชันแอกทิเวชันแบบไม่เป็นเชิงเส้นในชั้นซ่อน (Hidden Layer) ของโครงข่ายประสาท ขณะที่ชั้นข้อมูลส่งออก (Output Layer) ใช้แอกทิเวชันแบบเป็นเชิงเส้น

#### 4.3 กระบวนการการเรียนรู้ของโครงข่ายประสาทเทียม

การเรียนรู้ของโครงข่าย คือ การปรับค่าถ่วงน้ำหนักให้สอดคล้องกับอินพุตเพื่อให้ได้เอาต์พุตที่ต้องการ อัลกอริทึมการเรียนรู้ถูกจัดเป็น 2 ประเภทคือ

##### 4.3.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

ข้อมูลจะต้องประกอบด้วยตัวอย่างข้อมูลที่ต้องการปรับสอน และผลลัพธ์ที่ต้องการให้โครงข่ายสร้าง เมื่อมีการนำข้อมูลในลักษณะเดียวกันมาเป็นข้อมูลอินพุตขณะสอน โครงข่ายชนิดนี้ จะกำหนดค่าผลลัพธ์เป้าหมายให้กับข้อมูลอินพุตแต่ละตัว โครงข่ายจะนำค่าผิดพลาดระหว่างค่าเป้าหมายกับค่าผลลัพธ์ที่ได้มาใช้ในการปรับค่าน้ำหนัก เพื่อให้ค่าผลลัพธ์ที่ใกล้เคียงกับค่าเป้าหมายมากที่สุด ตัวอย่างแบบจำลองนี้ได้แก่ การแพร่แบบย้อนกลับ (Backpropagation) อะดาไลน์ (Adaline) และเพอเซปตรอน (Perceptron) เป็นต้น

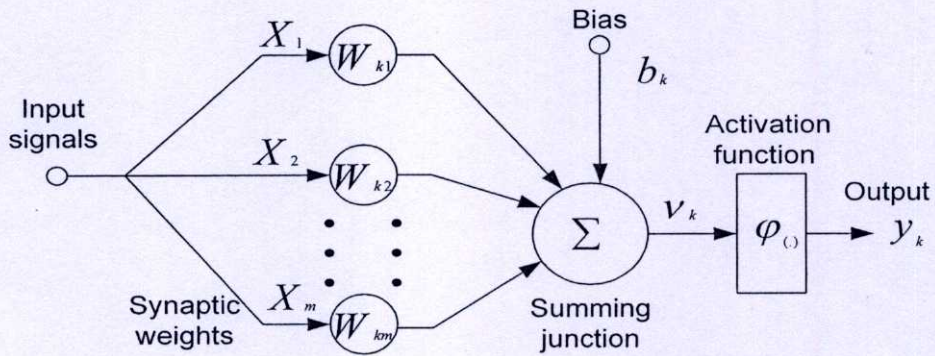
##### 4.3.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

จะสอนโครงข่ายโดยการป้อนข้อมูลอินพุตอย่างต่อเนื่องเพียงอย่างเดียว ไม่มีการส่งค่าผลลัพธ์เป้าหมายให้กับข้อมูลอินพุตแต่ละตัว การปรับน้ำหนักใช้ข้อมูลที่นำมาสอนเป็นตัวปรับค่า โดยค่าน้ำหนักจะปรับตามกลุ่มที่ข้อมูลเข้ามีรูปแบบคล้าย ๆ กัน ตัวอย่างแบบจำลองนี้ได้แก่เคาเตอร์พรอพเกชัน (Counter Propagation: CPN) แบบจำลองอะแดปทีฟรีโซแนนซ์ทฤษฎี (Adaptive Resonance Theory Neural Networks:ART) เป็นต้น [23]

#### 4.4 อัลกอริทึมโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ (Backpropagation Neural Networks)

โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ เป็นรูปแบบโครงข่ายที่ได้รับความนิยมใช้ในการพยากรณ์ เนื่องจากสามารถแก้ปัญหาในลักษณะเชิงเส้น (Linear) และไม่เชิงเส้น (Nonlinear) ได้ โดยมีกระบวนการเรียนรู้ข้อมูลจากอดีต โดยเริ่มจากนำค่าอินพุตมาคำนวณ หาผลรวม

(Combination Function) ที่ได้รับแต่ละโหนด แล้วส่งผลลัพธ์ไปยังขั้นต่อไป โดยผ่านฟังก์ชันการแปลง (Transfer Function) ซึ่งใช้ ฟังก์ชันซิกมอยด์ เพื่อให้ค่าผลลัพธ์อยู่ระหว่าง 0 กับ 1 ทิศทางการไหลข้อมูลเคลื่อนไปข้างหน้า ไปสู่เอาต์พุต จากนั้นใช้วิธีเรียนรู้แบบมีผู้สอน โดยนำผลที่ได้จากโครงข่ายในแต่ละครั้งของชุดข้อมูลที่ป้อนให้กับโครงข่ายนั้นมาเปรียบเทียบกับเอาต์พุตเป้าหมาย (Target Output) เพื่อหาค่าความผิดพลาดในแต่ละครั้งแล้วนำค่าความผิดพลาดที่ได้นั้นถอยหลังกลับไปยังชั้นฮิดเดน แล้วส่งต่อไปยังโหนดต่าง ๆ ของชั้นภายในโครงข่าย และนำค่าผิดพลาดที่ได้มาใช้ในการคำนวณปรับค่าถ่วงน้ำหนัก ซึ่งค่าถ่วงน้ำหนักจะเปลี่ยนมากหรือน้อยนั้นขึ้นอยู่กับค่าผิดพลาดมากหรือน้อยที่ได้รับ โดยที่ปรับค่าถ่วงน้ำหนักจะมีการคำนวณในทิศทางย้อนกลับจากทางด้านเอาต์พุตผ่านชั้นฮิดเดนจนปรับค่าถ่วงน้ำหนักมาถึงชั้นแรกที่อยู่ก่อนชั้นฮิดเดน จากนั้นคำนวณหาค่าของเอาต์พุตใหม่ กระบวนการเรียนรู้จะมีการคำนวณเช่นนี้ไปต่อไป จนกระทั่งค่าความผิดพลาดมีน้อยที่สุดเมื่อเทียบกับค่า Target โครงสร้างโครงข่ายแสดงดังรูปที่ 4.6



รูปที่ 4.6 โครงสร้างของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

#### 4.5 ขั้นตอนการคำนวณของอัลกอริทึม โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

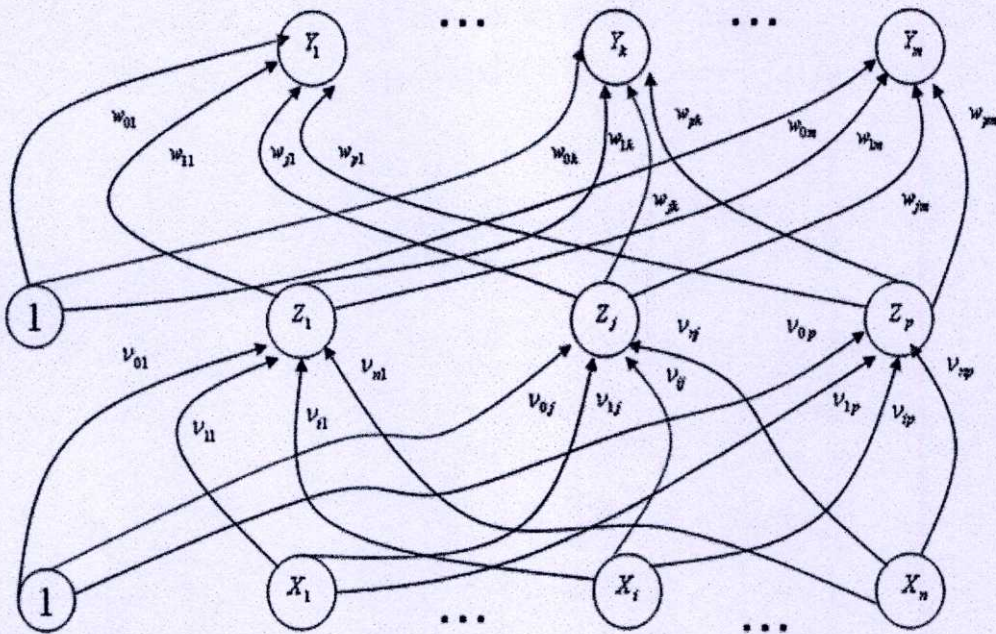
จากโครงสร้างของโครงข่าย ตามตัวอย่างในรูปที่ 4.7 สามารถสรุปขั้นตอนการคำนวณ ของอัลกอริทึม โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับได้ดังนี้

1. ป้อน อินพุต  $x = 1, 2, 3, \dots, n$  ให้แก่โหนดอินพุต กำหนดค่าถ่วงน้ำหนักเริ่มต้น (Initial Weight) โดยการสุ่ม ให้ค่าอยู่ในช่วง  $(-1, 1)$  แล้วส่งผ่านไปยังชั้นฮิดเดนและส่งต่อ ไปยังเอาต์พุต โดยคำนวณหาผลรวม (Combination Function ) แต่ละโหนด และส่งไปยังขั้นต่อไปจากนั้นผ่านฟังก์ชันซิกมอยด์ โดยฟังก์ชันเอาต์พุต คือ

$$y_k = f(y_{in_k}) \quad (4.1)$$

$$\text{โดยที่ } y_{in_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (4.2)$$

เมื่อ  $y_{in_k}$  คือ ข้อมูลอินพุต ที่ของ โหนด k ในชั้นเอาต์พุต  
 $w_{jk}$  คือ ค่าน้ำหนักบนเส้นเชื่อมโยงจากโหนดที่ j ไปยังโหนดที่ k  
 $z_j$  คือ ข้อมูลอินพุต ของโหนด ที่ j ในชั้นซ่อนหรือ ชั้น ฮิดเดน  
 $w_{ok}$  คือ ค่า Bias on Output ที่โหนด k  
 $y_k$  คือ เอาต์พุต ที่โหนด k ในชั้นเอาต์พุต



รูปที่ 4.7 การคำนวณของ อัลกอริทึม โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

2. เมื่อได้ผลลัพธ์แล้วนำค่าผลลัพธ์ดังกล่าวมาคำนวณหาค่าความผิดพลาด (Error) เทียบกับค่าที่ต้องการของแต่ละ โหนดในชั้นเอาต์พุต

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (4.3)$$

โดยที่  $t_k$  คือ ค่าเอาต์พุต ที่ต้องการ (Target Value)

$\delta_k$  คือ ค่าน้ำหนักที่ถูกปรับแก้แล้ว เนื่องจาก เอาต์พุต ที่โหนด k ในชั้นเอาต์พุต ที่ ถูกป้อนกลับไปชั้นฮิดเดน และส่งกลับไปชั้น เอาต์พุต ที่โหนด k อีก

3. จากนั้นจะคำนวณหาค่าความผิดพลาดของแต่ละ โหนดในชั้นฮิดเดน จากสมการ 4.4-4.5

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (4.4)$$

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (4.5)$$

$\delta_j$  คือค่าหนัก  $v_{ij}$  ที่ปรับแก้แล้วเนื่องจาก เอาต์พุต ที่โหนด  $j$  ในชั้นฮิดเดน  $Z$

4. เมื่อได้ค่าความผิดพลาดแล้วจะนำค่าดังกล่าวมาคำนวณหาค่าถ่วงน้ำหนักที่จะปรับเปลี่ยนในแต่ละชั้นดังสมการ 4.6 และ 4.7

การคำนวณค่าถ่วงน้ำหนักในชั้นเอาต์พุต

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (4.6)$$

การคำนวณค่าถ่วงน้ำหนักในชั้นซ่อน

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (4.7)$$

โดยที่  $\alpha$  คือ อัตราการเรียนรู้ของโครงข่ายประสาทเทียม

5. ปรับปรุงค่าถ่วงน้ำหนักที่คำนวณได้จากขั้นตอนที่ 5 โดยจะปรับค่าน้ำหนักที่ชั้นเอาต์พุตก่อนแล้วจึงปรับค่าถ่วงน้ำหนักที่ชั้นซ่อนถัดไป

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \quad (4.8)$$

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \quad (4.9)$$

6. ทำการวนซ้ำจนกว่าค่าความผิดพลาดที่คำนวณได้นั้นจะมีค่าใกล้เคียงกับศูนย์หรือเป็นค่าที่สามารถยอมรับได้ จึงจะหยุดการฝึกหัดโครงข่าย [24]

#### 4.6 การประยุกต์ใช้งานโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมสามารถนำมาประยุกต์ใช้งานต่างๆ ได้ดังต่อไปนี้

1. การจำแนกรูปแบบ (Pattern Recognition) เช่น การมองเห็นวัตถุ และการวิเคราะห์เสียงพูดเพื่อแปลความหมาย ซึ่งได้แก่ การสื่อสารผ่านเครือข่ายต่าง ๆ ระบบรักษาความปลอดภัยด้วยเสียงและระบบตอบรับโทรศัพท์อัตโนมัติ เป็นต้น

2. การทำนาย (Prediction) หรือการพยากรณ์ (Forecast ion) เช่น การทำนายหุ้น การพยากรณ์อัตราการไหลของน้ำ การพยากรณ์ราคาสินค้า เป็นต้น

3. การควบคุม (Control) เช่น การควบคุมระบบของเครื่องปรับอากาศ การควบคุมระบบเครื่องยนต์ และการควบคุมหุ่นยนต์ เป็นต้น
4. การหาความเหมาะสม (Optimization) เช่น การเลือกกระยะทางที่ใกล้ที่สุดในการเดินทาง เป็นต้น
5. การจัดกลุ่ม (Clustering) และการจัดหมู่ (Categorization)
6. การบีบอัดข้อมูล (Data Compression)
7. การจดจำตำแหน่ง (Content Addressable Memory)

#### 4.7 การใช้โครงข่ายประสาทเทียมในการพยากรณ์

อัลกอริทึมโครงข่ายประสาทเทียมที่นิยมใช้ในการพยากรณ์ คือ อัลกอริทึม โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ สำหรับในหัวข้อนี้ จะกล่าวถึงขั้นตอนในการพยากรณ์ ความซับซ้อนของข้อมูลในโครงข่ายไอพี ตามขั้นตอนดังนี้คือ

##### 4.7.1 การเตรียมข้อมูล

สำหรับการเตรียมข้อมูลที่จะใช้ในการพยากรณ์ เป็นขั้นตอนการดำเนินงานที่มีความสำคัญอย่างยิ่ง เนื่องจากข้อมูลที่ใช้มีความคลาดเคลื่อน จะมีผลทำให้ผลการพยากรณ์ลดความแม่นยำลง ข้อมูลที่นำมาใช้ในระบบจะต้องเป็นข้อมูลที่มีลักษณะแบบอนุกรมเวลา (Time Series) คือเป็นข้อมูลที่มีความต่อเนื่องโดยที่ข้อมูล ณ ช่วงเวลาหนึ่งจะมีผลมาจากความสัมพันธ์ของข้อมูลในอดีต ซึ่งควรพิจารณา 2 หัวข้อหลักคือ

1. ข้อมูลรบกวน (Noisy Data) จะต้องพิจารณาว่าข้อมูลที่ผ่านมาการคัดเลือกนั้นมีค่าในช่วงที่เหมาะสมไม่เป็นข้อมูลที่มากหรือน้อยเกินไปกับข้อมูลทั้งหมด โดยเฉลี่ย
2. ข้อมูลขาดหายไป (Missing Value) จะต้องพิจารณาว่าข้อมูลที่ผ่านมาการคัดเลือกนั้นต้องมีความต่อเนื่องไม่มีส่วนใดที่ขาดหายไป

##### 4.7.2 การกำหนดโครงสร้างของตัวแบบที่ใช้พยากรณ์

เมื่อได้ทำการเตรียมข้อมูลแล้ว ก่อนที่จะเริ่มทำการพยากรณ์โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับนั้น ต้องกำหนดโครงสร้างของโครงข่ายประสาทเทียมที่จะใช้ ซึ่งในการกำหนดโครงสร้างของโครงข่ายประสาทเทียมต้องกำหนด 4 ส่วน ด้วยกันคือ

1. กำหนดจำนวน โหนดในชั้น อินพุต เพื่อใช้กำหนดจำนวนอินพุต ในแต่ละรอบ
2. กำหนดจำนวนชั้นของ ฮิดเดน
3. กำหนดจำนวน โหนดในแต่ละชั้นของ ฮิดเดน
4. กำหนดจำนวน โหนดในชั้น เอาต์พุต เพื่อใช้กำหนด ผลของเอาต์พุต ในแต่ละรอบ

เนื่องจากในการกำหนดโครงสร้างของโครงข่ายประสาทเทียมมีผลต่อการเรียนรู้ของโครงข่ายประสาทเทียม จึงควรกำหนดโครงสร้างให้มีความเหมาะสมกับข้อมูลที่ได้อีกที่เตรียมไว้

#### 4.7.3 การนอร์มอลไลเซชัน (Normalization)

เพื่อหาค่าถ่วงน้ำหนักเริ่มต้น โดยการนำข้อมูลที่ได้อีกจากการจัดเตรียมข้อมูลมาทำการปรับเปลี่ยนค่าของข้อมูลให้อยู่ในระหว่าง 0 ถึง 1 เพื่อให้ข้อมูลมีความเหมาะสมในการที่จะป้อนอินพุตให้กับโครงข่ายประสาทเทียมที่เลือกใช้ในการพยากรณ์กราฟฟิก ซึ่งกระบวนการในการทำนอร์มอลไลเซชัน คำนวณได้จาก

$$xNorm = \frac{x_i}{2(\mu + \sigma)} \quad (4.10)$$

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i \quad (4.11)$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2 \quad (4.12)$$

เมื่อ  $x_i$  คือข้อมูล Traffic  $\mu$  คือค่าเฉลี่ยของข้อมูล (Mean)  $\sigma$  คือค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) และ  $N$  คือจำนวนข้อมูลทั้งหมด

#### 4.7.4 การฝึกสอนโครงข่ายประสาทเทียม

เพื่อเตรียมโครงข่ายให้พร้อมต่อการพยากรณ์ข้อมูล โดยให้โครงข่ายประสาทเทียมเรียนรู้รูปแบบและความสัมพันธ์ของข้อมูล และทำการปรับแต่งโครงข่าย โดยใช้วิธีการแพร่ย้อนกลับไปแก้ไขค่าน้ำหนักของโครงข่าย เพื่อให้ค่าความผิดพลาดที่จะเกิดในการพยากรณ์มีน้อยที่สุด

#### 4.7.5 การทดสอบเพื่อหาโมเดลที่ดีในการพยากรณ์

ในกระบวนการทดสอบข้อมูล จะมีวิธีการคำนวณเพื่อให้ได้เอาต์พุตในลักษณะเช่นเดียวกับการฝึกสอนในส่วนของ Forward Propagation ทุกประการ แต่แตกต่างกันตรงที่ข้อมูลที่ใช้เป็นข้อมูลทดสอบ และเอาต์พุตที่ได้ คือ ผลลัพธ์ของการตรวจสอบ ซึ่งการตรวจสอบนั้นจะใช้เวลาน้อยกว่ากระบวนการปรับสอนมากเพราะการคำนวณเป็นเพียง Forward Propagation เพียงรอบเดียว โดยนำค่าถ่วงน้ำหนักที่ได้จากการปรับสอนมาใช้ [25]

#### 4.7.6 การตรวจสอบความถูกต้องของการพยากรณ์ที่ใช้ในการทดสอบ

วิธีที่ใช้ในการตรวจสอบความถูกต้องของผลการพยากรณ์ซึ่งได้จากการพยากรณ์ก่อนหน้าหลาย ๆ โมเดลที่ใช้ในการทดลองเพื่อให้ได้โมเดลที่ดีที่สุด การพยากรณ์กราฟฟิกในแต่ละรอบของ

โครงข่ายไอพี ซึ่งมีการตรวจสอบความถูกต้องของการพยากรณ์อยู่หลายวิธีที่นิยมกัน เช่น ค่าเบี่ยงเบนสัมบูรณ์เฉลี่ย (Mean Absolute Deviation: MAD) ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Square Error: MSE) เปอร์เซ็นต์ค่าความคลาดเคลื่อนเฉลี่ย (Mean Percentage error: MPE) เปอร์เซ็นต์ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error: MAPE) ในวิทยานิพนธ์นี้ใช้ค่าเบี่ยงเบนสัมบูรณ์เฉลี่ย เพื่อหาค่าความผิดพลาด โดยค่าความผิดพลาดหาได้จากผลต่างระหว่างค่ากราฟฟิคในอดีตของแต่ละเดือนกับค่าที่พยากรณ์ได้ประจำเดือนนั้น ๆ ซึ่งแสดงการคำนวณได้จากสูตร

$$MAD = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4.13)$$

โดยที่  $n$  คือจำนวนของข้อมูลเอาต์พุต  $e_i$  คือค่าความผิดพลาดที่ระยะเวลา  $i$  และ  $i = 1, 2, 3 \dots n$  สำหรับการพิจารณาแบบจำลองพยากรณ์ที่เหมาะสม (Optional Forecasting Models) จะพิจารณาจากแบบจำลองที่มีค่า MAD มีค่าน้อยที่สุด

#### 4.7.7 การวิเคราะห์ผล และการนำไปใช้

การพยากรณ์หมายถึงการคาดคะเนหรือการทำนายเหตุการณ์ หรือสภาพการต่าง ๆ ในอนาคต โดยอาศัยข้อมูล ประสบการณ์ ความรู้ความสามารถของผู้พยากรณ์ที่เกิดขึ้นในอดีต มาทำการศึกษาถึงแนวโน้มหรือรูปแบบของการเกิดเหตุการณ์ในอนาคต ในวิทยานิพนธ์นี้ได้ศึกษาการพยากรณ์กราฟฟิคของโครงข่ายไอพี โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ซึ่งใช้ชั้นอินพุต 1 โหนด ชั้นซ่อน 3 โหนด ชั้นเอาต์พุต 1 โหนด ค่าโมเมนตัม 0.1 ค่าอัตราการเรียนรู้ 0.2 มาทำการพยากรณ์โครงข่ายไอพี 13 โหนดหลัก 28 ฮอป แล้วนำผลของการพยากรณ์มาใช้เป็นส่วนประกอบของค่าถ่วงน้ำหนักในการหาเส้นทางที่มีต้นทุนต่ำที่สุด

## บทที่ 5

# วิธีดำเนินการวิจัยและผลการวิจัย

ขั้นตอนการดำเนินการศึกษาวิจัย มีดังนี้

1. กำหนดโครงข่าย และกำหนดเส้นทางที่จะทำการวิจัย
2. ประมาณค่ากราฟฟิก (Traffic) ในแต่ละโหนด เพื่อเป็นองค์ประกอบการหาต้นทุนต่ำสุดในการค้นหาเส้นทางด้วยอัลกอริทึมเอสตาร์ โดยหาตัวแบบพยากรณ์ปริมาณกราฟฟิก ในอนาคต โดยใช้โครงข่ายประสาทเทียมที่เหมาะสมที่ได้พิจารณาไว้ในบทที่ 4
3. ประยุกต์ใช้ อัลกอริทึมของเอสตาร์ในการหาเส้นทางที่สั้นที่สุด โดยใช้องค์ประกอบของ ปริมาณกราฟฟิก ระยะทาง และค่าการสูญเสียในสายใยแก้วนำแสง
4. ปรับปรุงอัลกอริทึมของเอสตาร์ วิเคราะห์เปรียบเทียบ
5. สรุปผลการวิจัย

### 5.1 โครงข่ายและเส้นทางที่ทำวิจัย

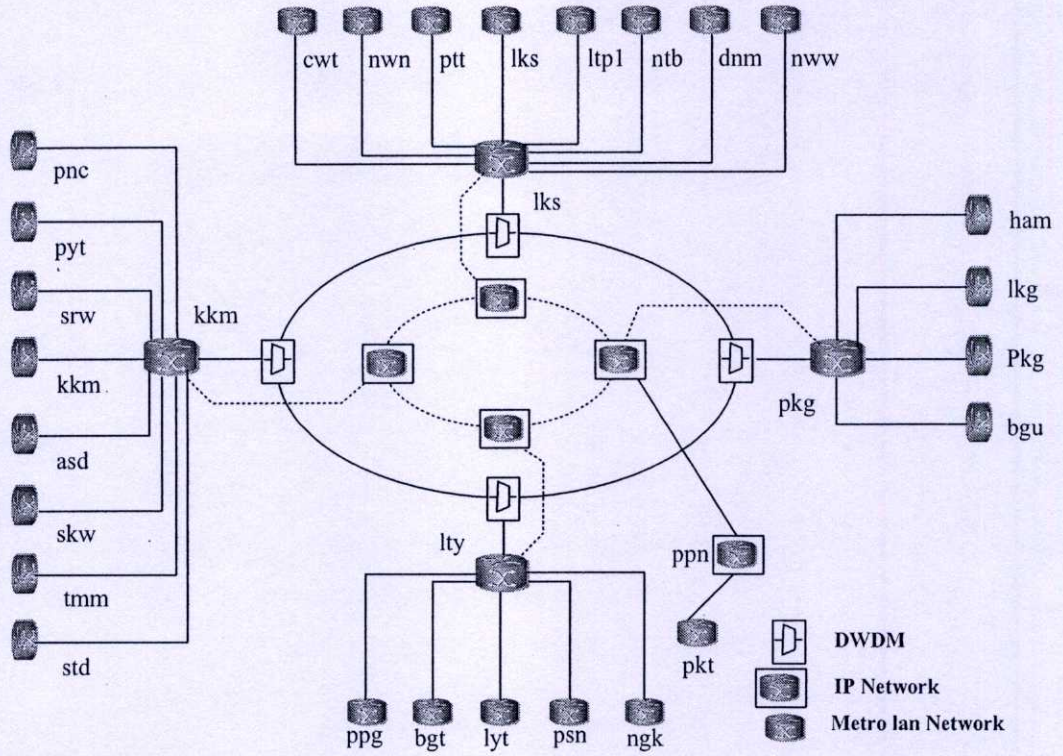
ในการคำนวณค้นหาเส้นทางที่มีต้นทุนต่ำที่สุด ในวิทยานิพนธ์นี้ดำเนินการโดยใช้ข้อมูลจริง บนโครงข่ายไอพีของ บริษัท ทีโอที จำกัด (มหาชน) ดังแสดงในรูปที่ 5.1 และ 5.2 ซึ่งเป็นโครงข่ายหลักที่ให้บริการในการรับ และส่งอินเทอร์เน็ตทั่วประเทศ ประกอบด้วย 2 โครงการด้วยกันคือ โครงข่ายไอพีไอพี และ โครงข่ายไอพี เมโทรแลน เพื่อประโยชน์ในการนำไปใช้งานจริงในการหาเส้นทางสำรอง กรณีเกิดเหตุขัดข้องในบางเส้นทางได้ต่อไป

#### 5.1.1 โครงข่ายที่ใช้ในการวิจัย

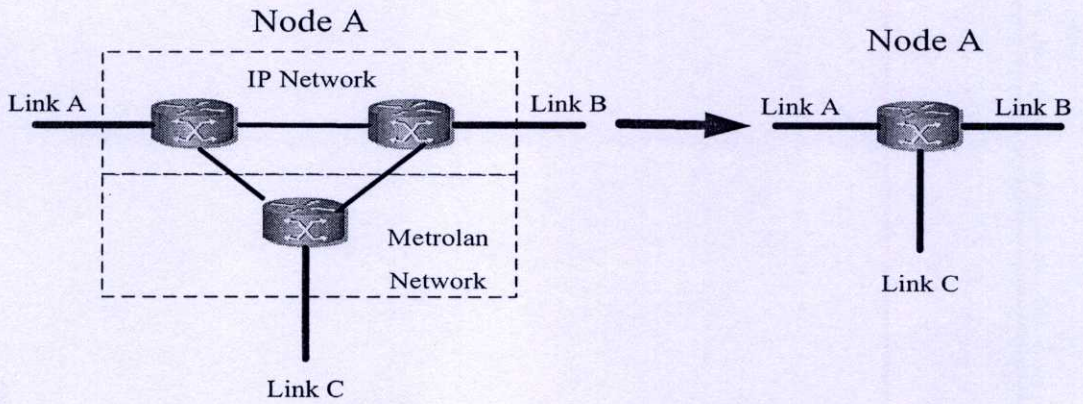
โครงข่ายไอพี ของบริษัท ทีโอที จำกัด (มหาชน) ที่ใช้ในการวิจัย ประกอบด้วยโหนดหลัก 13 โหนด กระจายอยู่ทั่วประเทศ โดยแบ่งเป็นส่วนภูมิภาค 9 โหนด และในนครหลวงอีก 4 โหนด โครงข่ายถูกออกแบบเป็นวง (Ring) จำนวน 5 วง เชื่อมต่อกัน เพื่อป้องกันโครงข่ายเกิดเหตุขัดข้องวงที่อยู่ในเขตนครหลวงมีความจุถึง 10 Gbps ขณะที่ในส่วนภูมิภาคมีความจุเพียง 2.5 Gbps แต่ละโหนดในนครหลวงใช้เราเตอร์ทำหน้าที่เป็นคอร์เลเยอร์ 2 ยูนิท เพื่อกระจายกราฟฟิกไปสู่ผู้ใช้งานได้ทั่วถึง และใช้เป็นเราเตอร์สำรอง (Back Up) ซึ่งกันและกันได้อีกด้วย

สำหรับการวิจัยนี้สมมุติให้ในแต่ละโหนดมีเพียง 1 เราเตอร์ เนื่องจากเราเตอร์ที่อยู่ในโหนดเดียวกัน มีค่าการสูญเสียของสายใยแก้วนำแสงน้อยมาก จึงไม่นำมาคำนวณ จะคำนวณเฉพาะระยะทาง และค่าการสูญเสียของสายใยแก้วที่อยู่ระหว่างโหนดหลักเท่านั้น นอกจากนี้ยังนำโครงข่ายเมโทรแลนในรูปที่ 5.2 มาพิจารณาร่วมด้วย

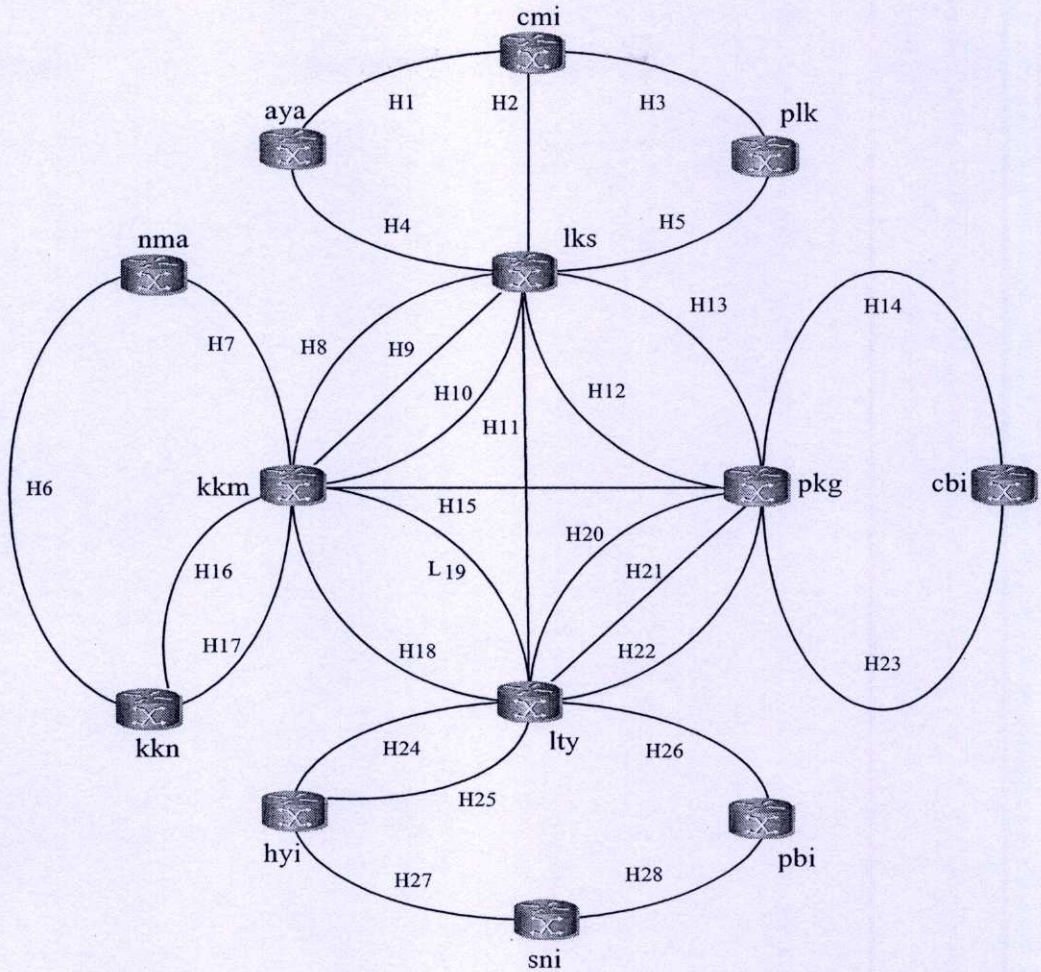




รูปที่ 5.2 โครงข่ายเมโทรแลน



รูปที่ 5.3 ปรับ โครงข่ายเพื่อใช้ในการวิจัย



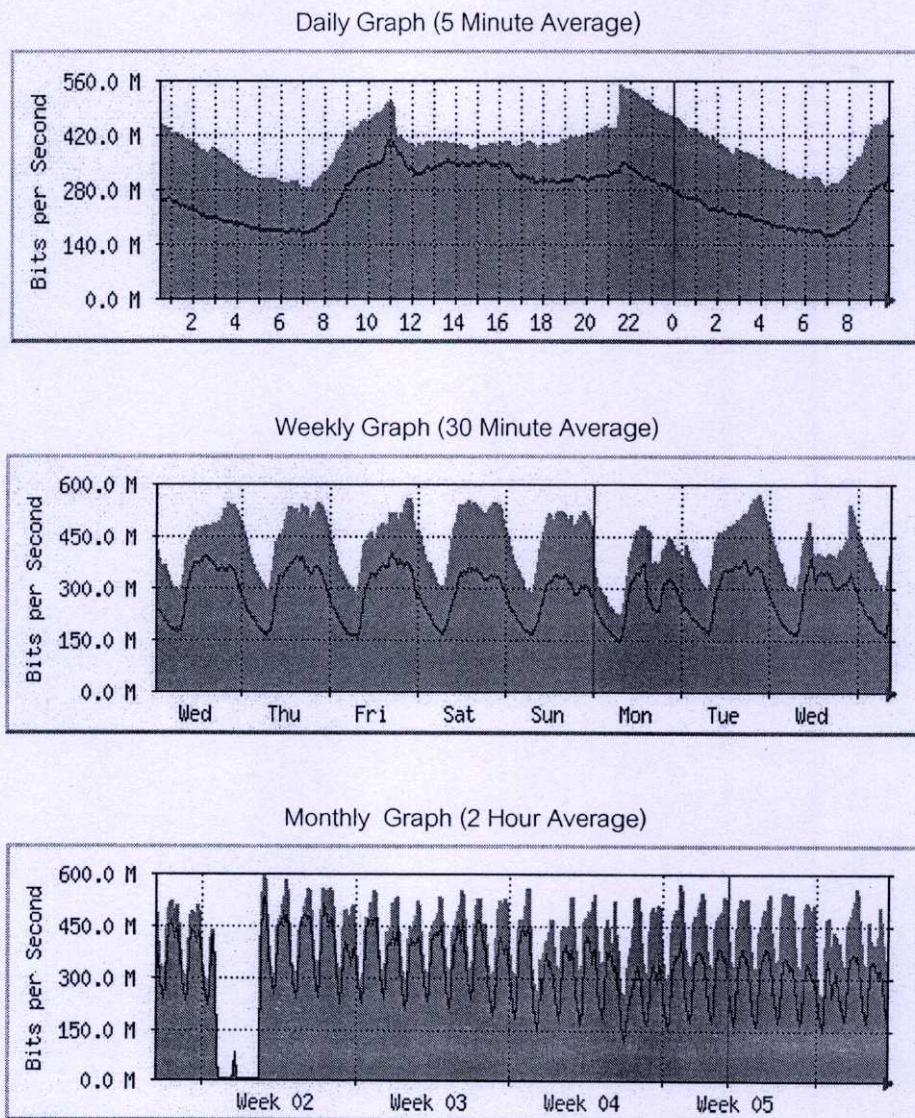
รูปที่ 5.4 โครงข่ายไอพีที่ใช้ในการวิจัย

ตารางที่ 5.1 โหนดต่าง ๆ และชื่อย่อของโครงข่ายไอพีที่ใช้ในการวิจัย

โหนดลำดับที่	ชื่อโหนด	ชื่อย่อโหนด
1	เชียงใหม่	cmi
2	อยุธยา	aya
3	พิษณุโลก	plk
4	นครราชสีมา	nma
5	หลักลี	lks
6	กรุงเทพฯ	kkm
7	พระโขนง	pkg
8	ชลบุรี	cbi
9	ขอนแก่น	kkn
10	ลาดหญ้า	lty
11	หาดใหญ่	hyi
12	เพชรบุรี	pbi
13	สุราษฎร์ธานี	sni

### 5.1.2 การเก็บข้อมูล ทราฟฟิก เพื่อการวิจัย

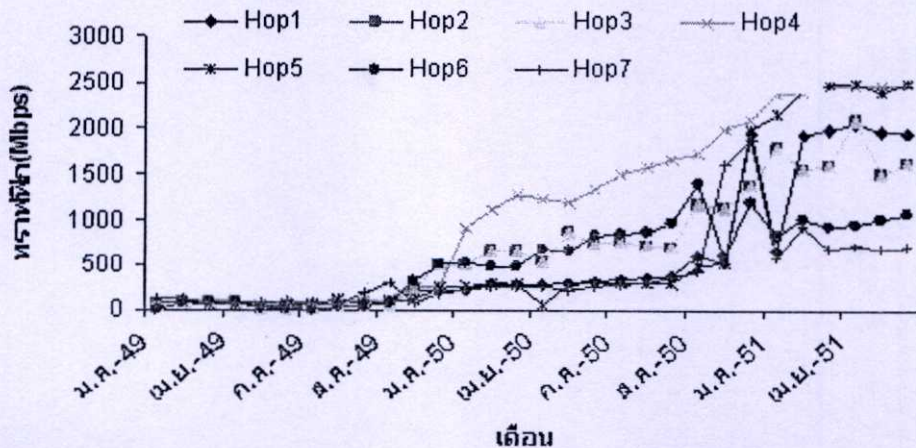
ในส่วนของการพยากรณ์ปริมาณทราฟฟิกในบทวิจยนี้ ได้จากการวัดทราฟฟิกในเวลาจริง (Real Time) จากโครงข่ายไอพี ของ บริษัท ทีโอที จำกัด (มหาชน) (TOT Public Company Limited) ณ ทุก ๆ โหนดหลัก 28 เส้นทาง (Hop: H) ตั้งแต่เดือน มกราคม 2549 ถึงเดือนมิถุนายน 2551 โดยระบบบริหารจัดการโครงข่ายหรือ NMS (Network Management System) ทำหน้าที่เป็นศูนย์กลางการควบคุม และ ฝ้ามองโครงข่ายตลอดเวลา รูปที่ 5.5 เป็นตัวอย่างการเปลี่ยนแปลงปริมาณทราฟฟิก ในแต่ละช่วงเวลา ซึ่งมีหน่วยเป็น บิตต่อวินาที (Bits/Second) โดยเก็บค่าเฉลี่ยทราฟฟิกสูงสุด ทุก 5 นาที ในรอบหนึ่งวัน ทุก 30 นาที ในรอบ 1 สัปดาห์ และทุก 2 ชั่วโมง ในรอบ 1 เดือน



รูปที่ 5.5 กราฟตัวอย่างปริมาณ ทราฟฟิกจากการมอนิเตอร์

ตารางที่ 5.2 ปริมาณทราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ สอบที่ 1-7

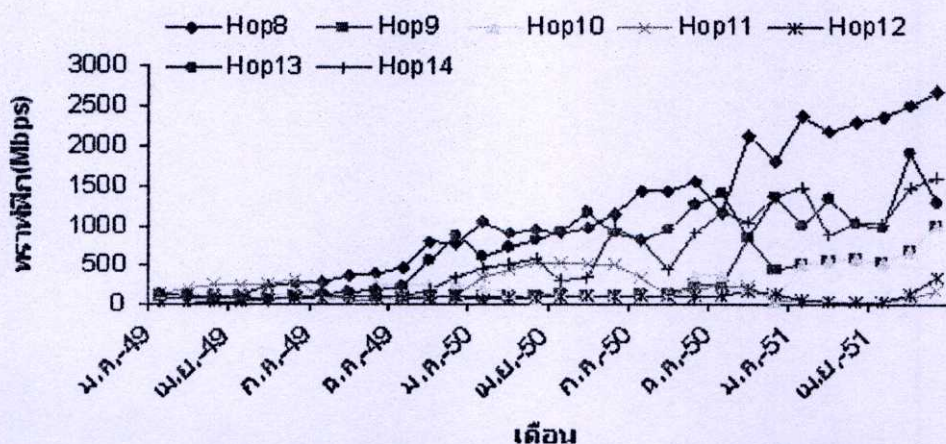
เดือน	ทราฟฟิกสอบต่าง ๆ ของโครงข่ายไอพี (Mbps)						
	Hop1	Hop2	Hop3	Hop4	Hop5	Hop6	Hop7
ม.ค.-49	50.100	58.100	58.100	75.000	78.900	9.000	112.300
ก.พ.-49	78.400	79.200	79.200	103.600	79.500	75.600	111.800
มี.ค.-49	82.300	83.900	83.900	105.600	69.000	77.100	89.800
เม.ย.-49	82.300	83.900	83.900	105.600	69.000	77.100	89.800
พ.ค.-49	45.800	24.430	24.430	54.400	46.050	23.840	89.690
มิ.ย.-49	50.900	29.080	29.080	71.680	45.000	24.330	98.890
ก.ค.-49	50.710	28.830	28.830	73.740	50.820	1.890	80.520
ส.ค.-49	78.000	54.180	54.180	105.900	128.300	37.000	66.900
ก.ย.-49	67.270	55.910	55.910	97.940	64.460	44.850	186.630
ต.ค.-49	84.800	54.810	54.810	121.930	99.610	73.060	301.190
พ.ย.-49	213.200	244.300	244.300	253.600	127.700	331.900	71.100
ธ.ค.-49	220.700	510.200	510.200	253.600	235.800	506.200	182.000
ม.ค.-50	222.000	516.100	516.100	910.000	260.300	528.900	214.000
ก.พ.-50	298.100	663.700	663.700	1117.600	284.800	488.800	263.200
มี.ค.-50	286.100	673.600	673.600	1271.200	291.300	488.800	263.200
เม.ย.-50	292.500	540.500	540.500	1237.000	69.600	696.600	258.800
พ.ค.-50	310.300	863.800	863.800	1200.700	286.400	662.600	224.700
มิ.ย.-50	315.300	751.000	751.000	1340.800	309.800	832.600	269.600
ก.ค.-50	345.300	761.000	761.000	1507.100	309.800	850.600	291.700
ส.ค.-50	371.400	723.800	723.800	1577.800	301.700	868.200	295.400
ก.ย.-50	393.300	701.300	701.300	1670.400	300.500	964.600	361.000
ต.ค.-50	603.100	1170.400	1170.400	1720.200	487.200	1395.600	427.800
พ.ย.-50	525.471	1131.176	1131.176	1983.776	526.978	615.617	1595.659
ธ.ค.-50	1948.000	1376.000	1376.000	2095.000	1995.000	1192.000	1865.000
ม.ค.-51	679.000	1779.000	1779.000	2379.000	2141.000	826.000	581.000
ก.พ.-51	1922.000	1567.000	1567.000	2402.000	2441.000	1011.000	938.000
มี.ค.-51	1990.000	1601.000	1601.000	2479.000	2479.000	926.000	682.000
เม.ย.-51	2056.000	2080.000	2080.000	2451.000	2486.000	946.000	723.000
พ.ค.-51	1961.000	1498.000	1498.000	2470.000	2394.000	1004.000	680.000
มิ.ย.-51	1944.000	1625.000	1625.000	2482.000	2484.000	1078.000	713.000



รูปที่ 5.6 ปริมาณทราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ สอบที่ 1-7

ตารางที่ 5.3 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ สอบที่ 8-14

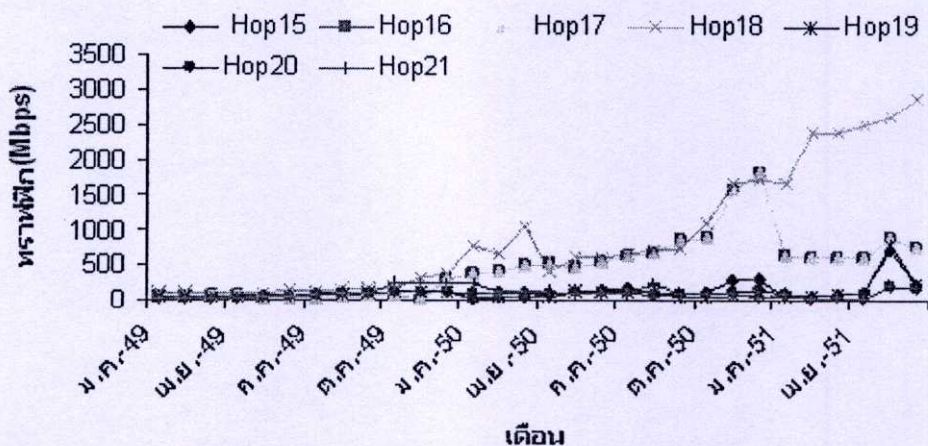
เดือน	กราฟฟิกสอบ ต่าง ๆ ของโครงข่ายไอพี (Mbps)						
	Hop8	Hop9	Hop10	Hop11	Hop12	Hop13	Hop14
ม.ค.-49	99.200	98.000	92.000	120.000	9.000	110.000	51.300
ก.พ.-49	72.300	107.100	104.800	183.460	10.000	95.700	67.300
มี.ค.-49	108.800	92.700	107.800	227.840	32.600	51.400	72.800
เม.ย.-49	108.800	99.700	107.800	227.840	32.600	74.200	72.800
พ.ค.-49	212.240	107.500	109.300	236.000	47.900	52.540	52.300
มิ.ย.-49	264.160	100.800	108.800	320.000	64.600	72.240	61.410
ก.ค.-49	262.660	106.300	108.400	81.000	75.200	118.220	95.470
ส.ค.-49	353.000	100.800	106.500	58.000	102.400	149.530	77.200
ก.ย.-49	387.880	101.400	239.200	36.000	100.500	158.610	91.270
ต.ค.-49	452.090	99.400	246.300	95.500	75.200	220.710	100.750
พ.ย.-49	779.000	101.200	250.000	97.000	82.000	539.500	166.900
ธ.ค.-49	751.400	103.500	255.000	102.000	87.000	880.800	343.900
ม.ค.-50	1051.200	88.700	191.400	337.780	54.385	599.100	460.900
ก.พ.-50	913.000	95.100	134.700	439.780	77.000	713.200	493.700
มี.ค.-50	947.100	102.000	79.400	524.930	80.700	798.100	572.000
เม.ย.-50	899.000	120.000	120.000	520.000	91.700	922.600	282.000
พ.ค.-50	968.600	99.800	126.600	511.000	93.500	1157.100	327.400
มิ.ย.-50	1131.200	99.800	126.600	506.000	96.500	899.800	978.830
ก.ค.-50	1422.000	109.800	126.600	330.000	98.500	815.800	808.600
ส.ค.-50	1436.000	126.600	126.600	101.000	100.500	941.900	436.300
ก.ย.-50	1549.800	248.800	350.000	197.300	100.300	1271.900	908.100
ต.ค.-50	1171.300	248.800	350.000	207.300	100.800	1404.800	1160.500
พ.ย.-50	2122.282	840.800	176.130	225.400	156.620	858.527	1049.925
ธ.ค.-50	1820.000	417.450	46.000	38.150	124.950	1358.527	1356.000
ม.ค.-51	2351.000	511.570	511.570	51.390	42.120	1009.000	1474.000
ก.พ.-51	2170.000	554.100	554.100	19.950	29.210	1341.000	879.000
มี.ค.-51	2280.000	576.820	576.820	25.480	29.970	1019.000	1052.000
เม.ย.-51	2360.000	513.740	513.740	28.720	27.260	970.000	1026.000
พ.ค.-51	2500.000	664.170	664.170	78.090	120.100	1898.000	1478.000
มิ.ย.-51	2670.000	1009.110	1009.110	159.350	336.230	1282.000	1603.000



รูปที่ 5.7 ปริมาณ กราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ สอบที่ 8-14

ตารางที่ 5.4 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ สอบที่ 15-21

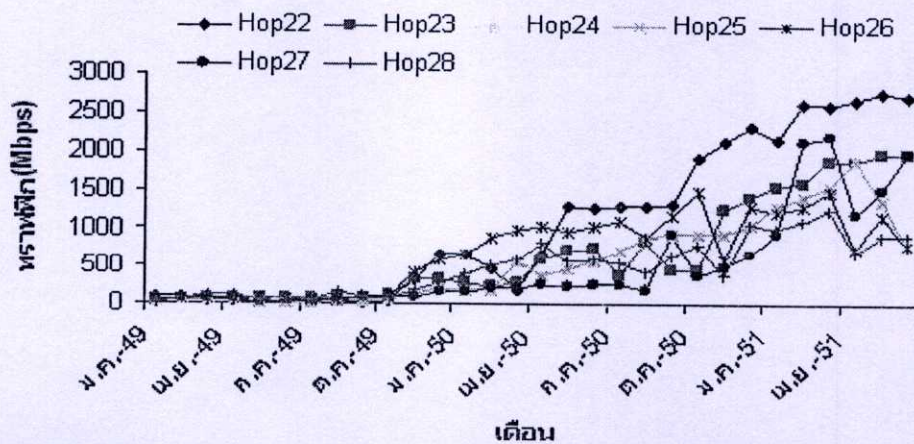
เดือน	กราฟฟิกสอบ ต่าง ๆ ของโครงข่ายไอพี (Mbps)						
	Hop15	Hop16	Hop17	Hop18	Hop19	Hop20	Hop21
ม.ค.-49	50.000	60.000	60.000	112.000	50.000	20.000	18.000
ก.พ.-49	64.700	65.700	65.700	99.700	52.100	21.600	19.900
มี.ค.-49	79.900	76.300	76.300	103.900	54.700	38.700	40.000
เม.ย.-49	79.900	76.300	76.300	107.000	54.700	38.700	40.000
พ.ค.-49	73.800	27.930	27.930	122.280	63.500	52.400	47.700
มิ.ย.-49	84.600	37.040	37.040	157.330	70.600	65.900	60.700
ก.ค.-49	101.900	35.460	35.460	131.280	79.400	90.000	87.200
ส.ค.-49	101.900	54.600	54.600	165.600	94.300	102.800	104.000
ก.ย.-49	122.600	78.510	78.510	158.650	98.100	100.500	100.800
ต.ค.-49	129.600	95.080	95.080	201.650	146.400	131.000	236.500
พ.ย.-49	132.000	16.100	16.100	337.400	149.000	133.000	242.000
ธ.ค.-49	135.000	314.100	314.100	391.600	152.000	135.300	257.000
ม.ค.-50	40.600	390.400	390.400	779.400	58.800	107.700	263.800
ก.พ.-50	45.000	426.600	426.600	667.400	61.500	107.700	130.800
มี.ค.-50	64.100	513.300	513.300	1043.000	72.900	117.700	130.800
เม.ย.-50	106.100	528.800	528.800	407.300	65.700	97.400	128.400
พ.ค.-50	135.100	484.200	484.200	600.300	136.000	128.100	122.600
มิ.ย.-50	164.100	556.400	556.400	600.300	136.000	128.100	122.600
ก.ค.-50	207.100	630.700	630.700	663.700	136.000	128.100	122.600
ส.ค.-50	144.000	676.300	676.300	724.400	136.000	95.140	258.250
ก.ย.-50	123.970	848.900	848.900	745.500	61.180	105.100	114.600
ต.ค.-50	130.970	878.300	878.300	1121.900	61.180	118.100	114.600
พ.ย.-50	310.040	1564.907	1564.907	1677.236	70.270	127.100	114.600
ธ.ค.-50	295.560	1842.000	1842.000	1722.000	71.910	156.900	49.410
ม.ค.-51	93.900	635.000	635.000	1654.000	68.890	97.660	97.660
ก.พ.-51	80.980	623.000	623.000	2386.000	76.130	25.260	25.260
มี.ค.-51	118.590	604.000	604.000	2378.000	105.610	50.770	50.770
เม.ย.-51	125.110	599.000	599.000	2500.000	120.210	27.260	27.260
พ.ค.-51	708.830	899.000	899.000	2600.000	815.530	221.370	221.370
มิ.ย.-51	261.130	751.000	751.000	2860.000	243.370	206.310	206.310



รูปที่ 5.8 ปริมาณ กราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ สอบที่ 15 - 21

ตารางที่ 5.5 ปริมาณกราฟฟิกของโครงข่ายไอพี ที่บันทึก ณ สอบที่ 22-28

เดือน	กราฟฟิกสอบ ต่าง ๆ ของโครงข่ายไอพี (Mbps)						
	Hop22	Hop23	Hop24	Hop25	Hop26	Hop27	Hop28
ม.ค.-49	106.800	54.900	42.500	42.500	48.700	51.500	24.000
ก.พ.-49	102.600	54.500	65.900	65.900	76.700	60.700	75.700
มี.ค.-49	51.500	76.200	61.000	61.000	82.900	69.400	108.900
เม.ย.-49	54.300	76.200	61.000	61.000	82.900	69.400	108.900
พ.ค.-49	30.900	59.950	15.380	15.380	87.200	33.200	58.700
มิ.ย.-49	32.770	62.860	13.180	13.180	73.100	45.800	99.210
ก.ค.-49	31.560	82.110	10.390	10.390	49.990	50.550	48.310
ส.ค.-49	36.000	77.200	43.770	43.770	43.700	72.500	160.900
ก.ย.-49	34.370	60.820	57.440	57.440	83.770	85.930	98.070
ต.ค.-49	42.500	108.650	67.280	67.280	95.540	125.530	63.840
พ.ย.-49	337.900	336.100	136.900	136.900	439.800	88.700	192.500
ธ.ค.-49	670.600	336.100	215.300	215.300	594.600	165.200	275.000
ม.ค.-50	675.800	282.600	197.400	197.400	672.900	175.200	411.000
ก.พ.-50	468.800	228.000	197.100	197.100	883.600	224.400	534.100
มี.ค.-50	188.800	282.000	575.400	575.400	970.600	224.400	589.400
เม.ย.-50	667.600	600.600	388.100	388.100	1027.100	257.800	803.500
พ.ค.-50	1296.300	714.500	472.500	472.500	945.700	236.000	561.900
มิ.ย.-50	1263.900	745.100	569.500	569.500	1014.700	257.600	598.500
ก.ค.-50	1275.500	392.300	720.800	720.800	1095.700	270.600	555.900
ส.ค.-50	1290.300	829.000	880.900	880.900	889.400	196.000	425.900
ก.ย.-50	1305.400	454.200	917.400	917.400	1162.600	917.400	651.800
ต.ค.-50	1900.600	454.200	923.400	923.400	1481.800	374.300	778.100
พ.ย.-50	2121.990	1246.793	930.402	930.402	587.737	507.636	369.887
ธ.ค.-50	2307.000	1409.000	1045.000	1045.000	1286.000	664.000	1042.000
ม.ค.-51	2145.000	1549.000	1296.000	1296.000	1222.000	928.000	1011.000
ก.พ.-51	2629.000	1594.000	1412.000	1412.000	1286.000	2120.000	1091.000
มี.ค.-51	2600.000	1886.000	1545.000	1545.000	1472.000	2182.000	1231.000
เม.ย.-51	2670.000	1859.000	1890.000	1890.000	745.000	1184.000	694.000
พ.ค.-51	2750.000	1983.000	1362.000	1362.000	1165.000	1492.000	908.000
มิ.ย.-51	2720.000	1969.000	788.000	788.000	774.000	1984.000	905.000



รูปที่ 5.9 ปริมาณ กราฟฟิกเฉลี่ยสูงสุดรายเดือน ณ สอบที่ 22-28

## 5.2 การประมาณค่ากราฟฟิคโดยใช้ตัวแบบโครงข่ายประสาทเทียมในการพยากรณ์

ขั้นตอนในการพยากรณ์ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ดำเนินการตามตามขั้นตอนในหัวข้อ 4.7.2 ถึง 4.7.6 แสดงผลในหัวข้อต่อไปนี้

### 5.2.1 ผลการกำหนดโครงสร้าง (Topology) ของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ

ในบทนี้นำเสนอการหาตัวแบบ (Model) ที่ดีที่สุดของโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับเพื่อการพยากรณ์ปริมาณกราฟฟิค ของแต่ละ สอบ ทั้ง 28 สอบ ตามขั้นตอนในหัวข้อ 4.7.2 ซึ่งมีองค์ประกอบหลักที่ต้องทดสอบ คือ ค่าจำนวนรอบสูงสุดในการคำนวณ ค่าโมเมนต์ดัม และค่าอัตราการเรียนรู้ที่เหมาะสม เพื่อให้ได้ค่าเอาต์พุตที่ใกล้เคียงกับค่าเป้าหมาย (Target) มากที่สุด

การทดลองหาจำนวนรอบแบ่งเป็นสองช่วงด้วยกันคือ 1 ถึง 10 รอบ และ 10 ถึง 300 รอบ ได้ผล SSE (Sum Square Error) น้อยที่สุดอยู่ที่ 180 รอบ ดังแสดงในตารางที่ 5.6

ตารางที่ 5.6 หาจำนวนรอบที่เหมาะสม

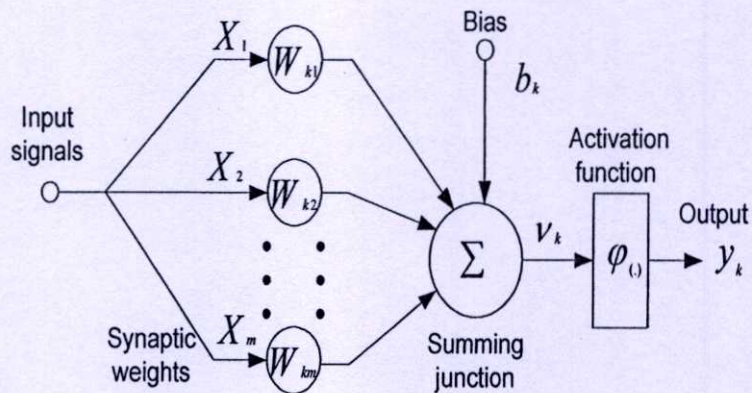
หาจำนวนรอบสูงสุดในการรันโปรแกรม			
โดยให้ชั้นอินพุต = 1, ชั้นซ่อน = 3, ชั้นเอาต์พุต = 1			
U=0.1,LR=0.9			
จำนวนรอบ	SSE	จำนวนรอบ	SSE
1	0.0050	120	0.0039
2	0.0050	130	0.0035
3	0.0046	140	0.0039
4	0.0047	150	0.0037
5	0.0046	160	0.0040
6	0.0039	170	0.0035
7	0.0046	<b>180</b>	<b>0.0034</b>
8	0.0040	190	0.0038
9	0.0047	200	0.0037
10	0.0036	210	0.0038
20	0.0042	220	0.0037
30	0.0036	230	0.0036
40	0.0037	240	0.0038
50	0.0038	250	0.0036
60	0.0040	260	0.0038
70	0.0037	270	0.0037
80	0.0035	280	0.0037
90	0.0039	290	0.0037
100	0.0038	300	0.0038
110	0.0037		

จากนั้นหาค่า โมเมนต์ดัม ( $\mu$ ) และ ค่าอัตราการเรียนรู้ ( $\alpha$ ) โดยทำการปรับเปลี่ยนค่า โมเมนต์ดัม และ ค่าอัตราการเรียนรู้ ตั้งแต่ 0.1–0.9 ปรับจนได้ค่า SSE น้อยที่สุดได้ค่าโมเมนต์ดัม เท่ากับ 0.1 และ อัตราการเรียนรู้เท่ากับ 0.2 ดังแสดงในตารางที่ 5.7

ตารางที่ 5.7 ค่าโมเมนต์ดัม และค่าอัตราการเรียนรู้ที่เหมาะสม

(รอบ=180 LR=0.9)		(รอบ=180 $\mu=0.1$ )	
หาค่า	SSE	หาค่า LR	SSE
0.1	0.0035	0.1	0.0025
0.2	0.0038	0.2	0.0023
0.3	0.0037	0.3	0.0028
0.4	0.0039	0.4	0.0026
0.5	0.0038	0.5	0.0029
0.6	0.0037	0.6	0.0032
0.7	0.0039	0.7	0.0032
0.8	0.0036	0.8	0.0034
0.9	0.0036	0.9	0.0040

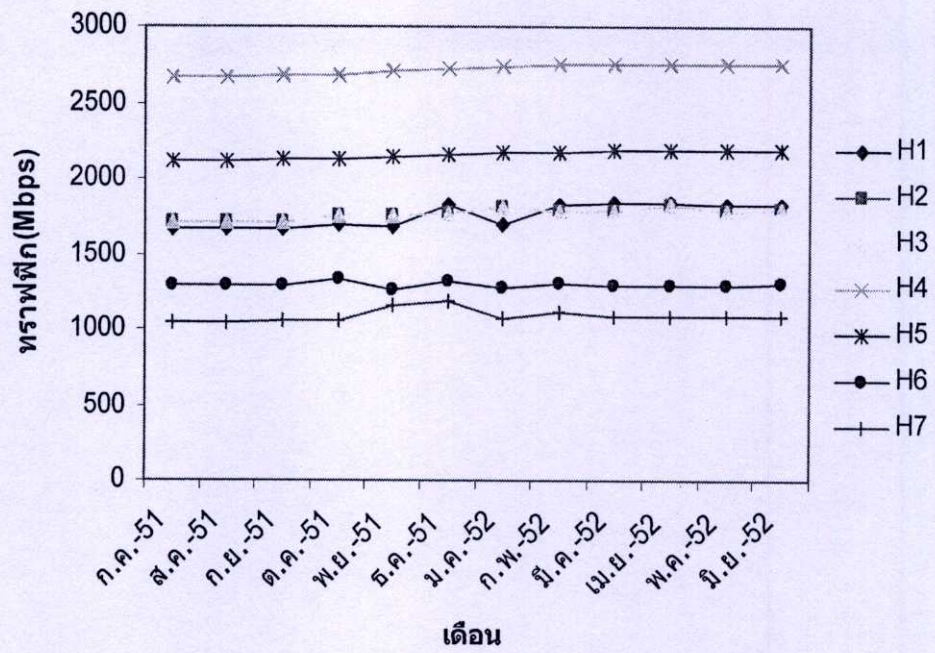
สำหรับตัวแบบที่โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่เหมาะสมที่ได้จากการ ทดสอบ แสดง ตามรูปที่ 5.10



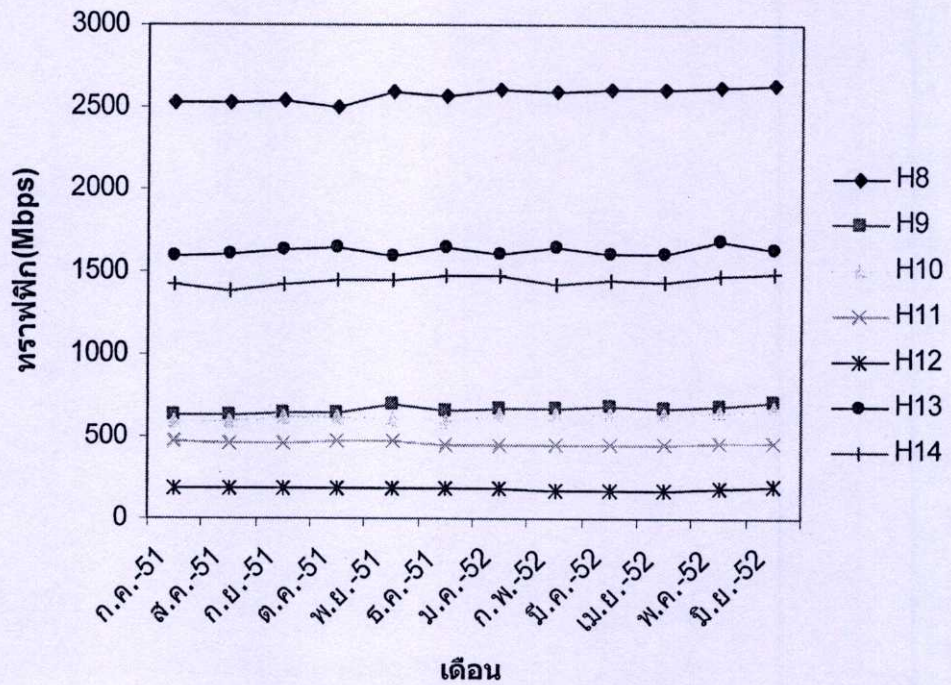
รูปที่ 5.10 ตัวแบบโครงข่ายประสาทเทียมแบบแพร่ย้อนกลับที่เหมาะสมเพื่อการพยากรณ์

### 5. 2. 2 ผลของการพยากรณ์ ทราฟฟิกตัวแบบที่โครงข่ายประสาทเทียม แบบแพร่ย้อนกลับ

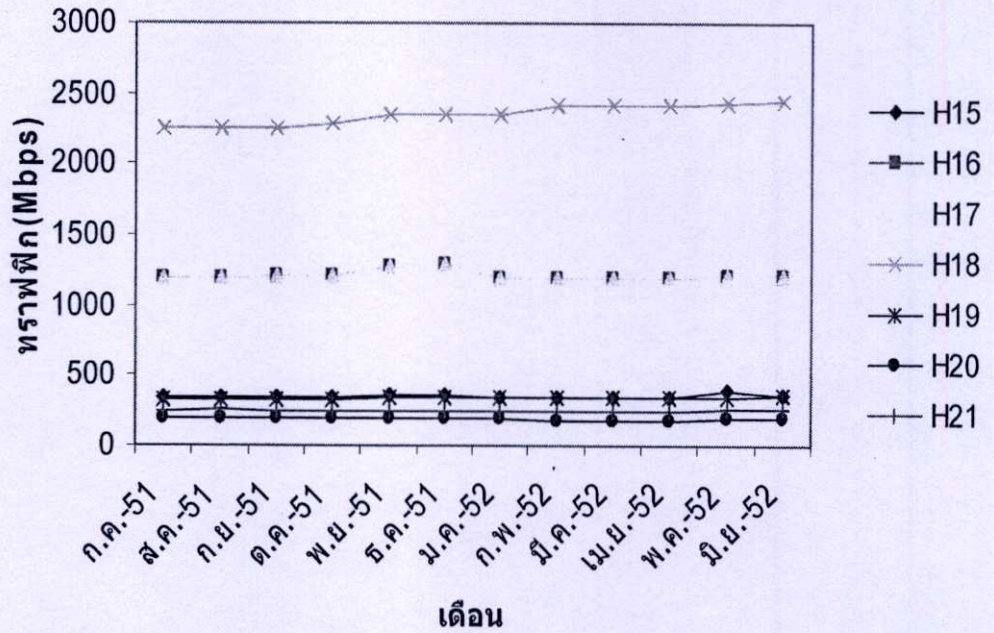
ในการพยากรณ์ทราฟฟิกโดยใช้โครงข่ายประสาทเทียมนี้ ใช้ข้อมูลทราฟฟิกเฉลี่ยของโครงข่ายไอพี ซึ่งเก็บข้อมูล ตั้งแต่เดือน มกราคม 2549 ถึง เดือน มิถุนายน 2551 รวมเวลา 30 เดือน เป็นข้อมูลในการเรียนรู้ของโครงข่าย แล้วทำการพยากรณ์ล่วงหน้าซึ่งได้ผลการพยากรณ์แต่ละขอบ ดังแสดงในรูปที่ 5.11–5.14



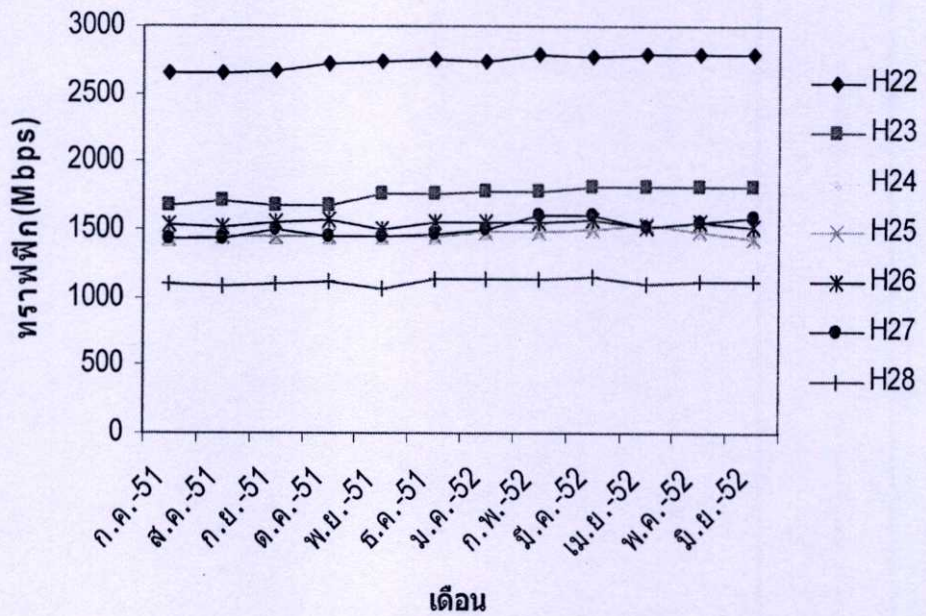
รูปที่ 5.11 ผลการพยากรณ์ทราฟฟิกของ ฮอปที่ 1-7



รูปที่ 5.12 ผลการพยากรณ์ ทราฟฟิกของ ฮอปที่ 8-14



รูปที่ 5.13 ผลการพยากรณ์ทราฟฟิกของ ฮอปที่ 15-21



รูปที่ 5.14 ผลการพยากรณ์ทราฟฟิกของ ฮอปที่ 22-28

### 5.3 การประยุกต์ใช้ อัลกอริทึมของเอสตาร์ในการหาเส้นทางที่สั้นที่สุด

จากผลของการพยากรณ์กราฟฟิกในแต่ละเดือนในช่วงเวลา 1 ปี (12 เดือน ตั้งแต่ กรกฎาคม 2551 ถึง มิถุนายน 2552) ขั้นตอนต่อไปในบทวิจัยนี้คือ เลือกค่า ทราฟฟิกของเดือนที่มีค่าทราฟฟิก สูงที่สุดของแต่ละขอบ เพื่อนำค่าทราฟฟิกที่ได้จากการพยากรณ์ดังกล่าวมาใช้เป็นองค์ประกอบร่วมกับค่าระยะทาง และค่าการสูญเสียในสายใยแก้วนำแสงของแต่ละขอบ เพื่อหาค่าต้นทุนที่ต่ำสุดของแต่ละ ขอบ และนำไปใช้ ในการกำหนดเส้นทางด้วย อัลกอริทึม เอสตาร์ โดยกำหนดเส้นทางที่จะส่งแพ็คเก็ต ในหัวข้อนี้ได้ยกตัวอย่างเส้นทางหลัก 3 เส้นทางแสดง ตามรูปที่ 5.15-5.26 คือ เชียงใหม่-หาดใหญ่ เชียงใหม่-ชลบุรี และเชียงใหม่-ขอนแก่น เป็นเส้นทางทดสอบ

#### 5.3.1 การหาค่าต้นทุน ต่ำสุด (Cost) ในแต่ละเส้นทาง

ในการคำนวณหาเส้นทางที่มีค่าต้นทุนต่ำที่สุด (Cost) เพื่อใช้กับอัลกอริทึม เอสตาร์ ในบทวิจัยนี้ กำหนดเงื่อนไขในการพิจารณา คัดค่าต้นทุนซึ่งประกอบด้วย ปริมาณ ทราฟฟิก ระยะทาง และ ค่าการสูญเสียในสายใยแก้วนำแสง เป็น 2 กรณี คือ

1. คัดค่าต้นทุน โดยวิธีให้น้ำหนักของพารามิเตอร์ 3 ตัว มีค่าเท่ากัน และ
2. คัดค่าต้นทุน โดยให้น้ำหนักมากไปทาง ค่าของทราฟฟิก

โดยได้ทดสอบหาค่าต้นทุนใน 3 เส้นทาง คือ เชียงใหม่-หาดใหญ่ เชียงใหม่-ชลบุรี และ เชียงใหม่-ขอนแก่น ตามโครงสร้างของโครงข่ายหลักตามรูปที่ 5.16 แบ่งเงื่อนไขในการทดสอบ เป็น 2 กรณี คือ กรณีโครงข่ายสมบูรณ์ไม่มีเหตุขัดข้อง และกรณีโครงข่ายเกิดเหตุขัดข้องบางเส้นทาง โดยกำหนดให้สามารถขัดข้องได้สูงสุด 3 เส้นทาง

สำหรับค่าของทราฟฟิก และค่าระยะทางระหว่างโหนดได้จากการวัดจริงในโครงข่ายไอพี ส่วนค่าการสูญเสียในสายของใยแก้วนำแสงได้จากการหาค่าเฉลี่ยของโครงข่ายที่ใช้งานจริง โดยไม่รวมค่าชดเชย (Margin)

สูตรการคำนวณหาค่า Link Cost แสดง ดังสมการ 5.1-5.6

$$\text{Link Cost} = A + B + C \quad (5.1)$$

$$\text{โดยที่} \quad A = T * W_1 \quad (5.2)$$

$$B = D * W_2 \quad (5.3)$$

$$C = L * W_3 \quad (5.4)$$

แทนค่าสมการ (5.2) - (5.4) ในสมการที่ (5.1) จะได้

$$\text{Link Cost} = (T * W_1) + (D * W_2) + (L * W_3) \quad (5.5)$$

เมื่อ  $T$  คือ ปริมาณ ทราฟฟิก (Mbps)

$D$  คือ ระยะทางในแต่ละช่วง (Km)

$L$  คือ ค่าการสูญเสียในเส้นใยแก้วนำแสง (dB)

$W_1$  คือ ค่าถ่วงน้ำหนักในเทอม A (ปริมาณทราฟฟิก)

$W_2$  คือ ค่าถ่วงน้ำหนักในเทอม B (ระยะทาง)

$W_3$  คือ ค่าถ่วงน้ำหนักในเทอม C (ค่า Losses)

โดยกำหนดให้

$$W_1 + W_2 + W_3 = 1 \quad (5.6)$$

เมื่อ เทอม A เป็นตัวแปรของปริมาณทราฟฟิก เทอม B เป็นตัวแปรของระยะทางระหว่างจุดที่เชื่อมโยงกับอุปกรณ์เราเตอร์ด้านต้นทางกับด้านปลายทาง เทอม C ในที่นี้พิจารณาการสูญเสียในสายโดยรวม (Total Losses) ได้มาจากการลดทอนกำลังงานในเส้นใยแก้วนำแสงเฉลี่ยของโครงข่ายที่ใช้งานจริง (Optical Fiber Attenuation: 0.2525 dB/km) และเพื่อให้ได้ค่าเชื่อมโยงที่น้อย และเหมาะสมที่สุดจึงได้พิจารณาที่ค่าถ่วงน้ำหนักที่แตกต่างกัน

จากการเฝ้าดูความเปลี่ยนแปลงของทราฟฟิก จะเห็นได้ว่า ความคับคั่งของทราฟฟิก เป็นสาเหตุที่สำคัญของการเรียกไม่ผ่าน ฉะนั้นในงานวิจัยนี้จึงกำหนดเงื่อนไขในการหาต้นทุนต่ำสุดไว้ 2 กรณี คือ

1. หาค่าต้นทุนต่ำ โดย กำหนดให้ค่าน้ำหนักเท่ากัน

จากสมการที่ (5.4) ผลรวมของ  $W_1 + W_2 + W_3 = 1$  จึงกำหนดให้ค่าทั้งสามมีค่าเท่ากันคือเท่ากับ  $1/3$  นั่นคือ  $W_1 = 0.33$ ,  $W_2 = 0.33$  และ  $W_3 = 0.33$  เมื่อแทนค่าในสมการที่ (5.5) สามารถคำนวณหาค่าเชื่อมโยงได้ตามตารางที่ 5.8

2. หาค่าต้นทุนต่ำ โดยกำหนดให้ค่าน้ำหนักมากไปทางค่า ทราฟฟิก ในที่นี้ กำหนดให้ค่าถ่วงน้ำหนักมีค่า  $W_1 = 0.50$   $W_2 = 0.25$  และ  $W_3 = 0.25$  เมื่อนำไปแทนค่าในสมการที่ (5.5) สามารถคำนวณหาค่าต้นทุนได้ตามตารางที่ 5.9

ตารางที่ 5.8 หาค่าต้นทุนต่ำที่ให้ค่าน้ำหนักเท่ากัน

Hop No.	ต้น ทง	ปลาย ทง	กราฟฟิก (Mbps)	ระยะทาง (Km)	Loss (dB)	Weight			A	B	C	Cost
						w1	w2	w3				
1	cmi	aya	1840.193	745.39	188.21	0.33	0.33	0.33	607.26	245.98	62.11	915.352
2	cmi	lks	1829.781	1061.32	267.98	0.33	0.33	0.33	603.83	350.24	88.43	1042.498
3	cmi	plk	1829.781	820.23	207.11	0.33	0.33	0.33	603.83	270.68	68.35	942.849
4	aya	lks	2753.394	62.11	15.68	0.33	0.33	0.33	908.62	20.50	5.18	934.292
5	plk	lks	2184.692	854.90	215.86	0.33	0.33	0.33	720.95	282.12	71.23	1074.300
6	kkn	nma	1331.154	199.64	50.41	0.33	0.33	0.33	439.28	65.88	16.64	521.797
7	nma	kkm	1181.643	285.59	72.11	0.33	0.33	0.33	389.94	94.24	23.80	507.984
8	lks	kkm	2636.967	35.18	8.88	0.33	0.33	0.33	870.20	11.61	2.93	884.740
9	lks	kkm	713.134	24.10	6.09	0.33	0.33	0.33	235.33	7.95	2.01	245.295
10	lks	kkm	670.940	22.20	1.19	0.33	0.33	0.33	221.41	7.33	0.39	229.128
11	lks	lty	479.368	29.50	7.45	0.33	0.33	0.33	158.19	9.74	2.46	170.385
12	lks	pkg	202.665	35.40	8.94	0.33	0.33	0.33	66.88	11.68	2.95	81.511
13	lks	pkg	1688.643	61.80	15.60	0.33	0.33	0.33	557.25	20.39	5.15	582.796
14	cbi	pkg	1490.051	77.05	19.46	0.33	0.33	0.33	491.72	25.43	6.42	523.564
15	kkm	pkg	388.416	13.60	3.43	0.33	0.33	0.33	128.18	4.49	1.13	133.799
16	kkn	kkm	1291.743	331.68	3.43	0.33	0.33	0.33	426.28	109.45	1.13	536.863
17	kkn	kkm	1291.743	485.23	122.52	0.33	0.33	0.33	426.28	160.13	40.43	626.833
18	kkm	lty	2447.713	6.60	1.67	0.33	0.33	0.33	807.75	2.18	0.55	810.473
19	kkm	lty	351.888	11.90	3.01	0.33	0.33	0.33	116.12	3.93	0.99	121.042
20	pkg	lty	202.885	24.40	6.16	0.33	0.33	0.33	66.95	8.05	2.03	77.037
21	pkg	lty	263.972	21.70	5.48	0.33	0.33	0.33	87.11	7.16	1.81	96.080
22	pkg	lty	2793.952	20.80	5.25	0.33	0.33	0.33	922.00	6.86	1.73	930.601
23	cbi	pkg	1818.466	77.05	19.46	0.33	0.33	0.33	600.09	25.43	6.42	631.940
24	lty	hyi	1529.292	1076.59	271.84	0.33	0.33	0.33	504.67	355.27	89.71	949.648
25	lty	hyi	1529.292	1104.75	278.95	0.33	0.33	0.33	504.67	364.57	92.05	961.287
26	lty	pbi	1577.074	112.93	28.51	0.33	0.33	0.33	520.43	37.27	9.41	567.111
27	sni	hyi	1603.958	372.27	94.00	0.33	0.33	0.33	529.31	122.85	31.02	683.175
28	pbi	sni	1152.137	563.80	142.36	0.33	0.33	0.33	380.21	186.05	46.98	613.238

จากตาราง ที่ 5.8 จะเห็นได้ว่าเส้นทางที่มีการใช้กราฟฟิกหนาแน่น จะเกิดต้นทุนสูงที่สุดเสมอ ฉะนั้น ปริมาณการเพิ่มของผู้ใช้ หรือปริมาณกราฟฟิกที่เพิ่มขึ้นอย่างรวดเร็วจึงเป็นสาเหตุที่สำคัญที่ต้องนำมาพิจารณาต่อไป

ตารางที่ 5.9 ค่าต้นทุนต่ำที่ให้ค่าน้ำหนักมากไปทางกราฟฟิก

Hop No	ต้น ทาง	ปลาย ทาง	กราฟฟิก (Mbps)	ระยะทาง (Km)	Loss (dB)	Weight			A	B	C	Cost
						w1	w2	w3				
1	cmi	aya	1840.193	745.39	188.211	0.50	0.25	0.25	920.10	186.35	47.05	1153.497
2	cmi	lks	1829.781	1061.32	267.983	0.50	0.25	0.25	914.89	265.33	67.00	1247.216
3	cmi	plk	1829.781	820.23	207.108	0.50	0.25	0.25	914.89	205.06	51.78	1171.725
4	aya	lks	2753.394	62.11	15.683	0.50	0.25	0.25	1376.70	15.53	3.92	1396.145
5	plk	lks	2184.692	854.90	215.862	0.50	0.25	0.25	1092.35	213.73	53.97	1360.037
6	kkn	nma	1331.154	199.64	50.409	0.50	0.25	0.25	665.58	49.91	12.60	728.089
7	nma	kkm	1181.643	285.59	72.111	0.50	0.25	0.25	590.82	71.40	18.03	680.247
8	lks	kkm	2636.967	35.18	8.883	0.50	0.25	0.25	1318.48	8.80	2.22	1329.499
9	lks	kkm	713.134	24.10	6.085	0.50	0.25	0.25	356.57	6.03	1.52	364.113
10	lks	kkm	670.940	22.20	1.187	0.50	0.25	0.25	335.47	5.55	0.30	341.317
11	lks	lty	479.368	29.50	7.449	0.50	0.25	0.25	239.68	7.38	1.86	248.921
12	lks	pkg	202.665	35.40	8.939	0.50	0.25	0.25	101.33	8.85	2.23	112.417
13	lks	pkg	1688.643	61.80	15.605	0.50	0.25	0.25	844.32	15.45	3.90	863.673
14	cbi	pkg	1490.051	77.05	19.455	0.50	0.25	0.25	745.03	19.26	4.86	769.152
15	kkm	pkg	388.416	13.60	3.434	0.50	0.25	0.25	194.21	3.40	0.86	198.467
16	kkn	kkm	1291.743	331.680	3.434	0.50	0.25	0.25	645.87	82.92	0.86	729.650
17	kkn	kkm	1291.743	485.230	122.521	0.50	0.25	0.25	645.87	121.31	30.63	797.809
18	kkm	lty	2447.713	6.60	1.667	0.50	0.25	0.25	1223.86	1.65	0.42	1225.923
19	kkm	lty	351.888	11.90	3.005	0.50	0.25	0.25	175.94	2.98	0.75	179.670
20	pkg	lty	202.885	24.40	6.161	0.50	0.25	0.25	101.44	6.10	1.54	109.083
21	pkg	lty	263.972	21.70	5.479	0.50	0.25	0.25	131.99	5.43	1.37	138.781
22	pkg	lty	2793.952	20.80	5.252	0.50	0.25	0.25	1396.98	5.20	1.31	1403.489
23	cbi	pkg	1818.466	77.05	19.455	0.50	0.25	0.25	909.23	19.26	4.86	933.359
24	lty	hyi	1529.292	1076.59	271.839	0.50	0.25	0.25	764.65	269.15	67.96	1101.753
25	lty	hyi	1529.292	1104.75	278.949	0.50	0.25	0.25	764.65	276.19	69.74	1110.571
26	lty	pbi	1577.074	112.93	28.515	0.50	0.25	0.25	788.54	28.23	7.13	823.898
27	sni	hyi	1603.958	372.27	93.998	0.50	0.25	0.25	801.98	93.07	23.50	918.546
28	pbi	sni	1152.137	563.80	142.360	0.50	0.25	0.25	576.07	140.95	35.59	752.608

### 5.3.2 การหาค่าฮิวริสติก (Heuristic)

ค่าฮิวริสติกมีส่วนสำคัญมากสำหรับวิธีการหาเส้นทางที่มีต้นทุนต่ำด้วย อัลกอริทึม เอสตาร์ เพราะจะเป็นตัวช่วยในการหาเส้นทางเป้าหมายได้แม่นยำมากขึ้น ในวิทยานิพนธ์นี้ใช้วิธีหาค่าฮิวริสติกของแต่ละโหนดที่พิจารณาโดยการวัดค่าระยะทางจากโหนดต้นทาง กับโหนดเป้าหมาย (Goal) ซึ่งทำการวัดโดยใช้ โปรแกรม Google Earth ได้ผลการวัดตามตารางที่ 5.10

ตารางที่ 5.10 ค่าฮิวริสติกของแต่ละเส้นทาง

Heuristic Straight-Line Distance(Km)				
Node No.	Name	cmi – hyi	cmi-cbi	cmi-kkn
N1	cmi	1326.75	644.20	481.38
N2	plk	1114.11	393.75	273.64
N3	lks	760.93	71.47	373.12
N4	aya	815.39	112.68	335.88
N5	nma	905.67	218.06	180.21
N6	kkm	746.47	66.77	388.38
N7	kkn	1065.93	401.83	0.00
N8	pkg	740.73	56.33	385.59
N9	lty	742.29	67.18	391.05
N10	cbi	708.52	0.00	394.76
N11	pbi	677.91	113.76	481.10
N12	sni	274.70	506.06	902.36
N13	hyi	0.00	703.33	1073.23

#### 5.4 ตัวอย่างการหาเส้นทางด้วย อัลกอริทึม เอสตาร์

จากโครงข่ายหลักของ บริษัท ทีโอที จำกัด(มหาชน) ตามรูปที่ 5.4 กำหนดเส้นทางจาก เชียงใหม่ – หาดใหญ่ และ จากค่าต้นทุนกรณีกำหนดให้ค่าถ่วงน้ำหนักเท่ากันตามตารางที่ 5.8 และ ค่าฮิวริสติกตามตารางที่ 5.10 ทำการหาเส้นทางด้วย อัลกอริทึม เอสตาร์ โดยพิจารณา กรณี โครงข่ายสมบูรณ์ ตามขั้นตอนต่อไปนี้

พิจารณาเส้นทางจาก เชียงใหม่-หลักสี่ มีเส้นทางที่กราฟฟิกสามารถวิ่งผ่านได้สามเส้นทางด้วยกันคือ ผ่าน H1, H2 และ H3 ซึ่งสามารถนำค่าต้นทุนต่ำจาก ตารางที่ 5.8 และค่าฮิวริสติก จาก ตาราง 5.10 มาแทนใน อัลกอริทึมของ เอสตาร์ ตามสมการที่ 4. 1 ได้ค่าต้นทุนที่ต่ำที่สุด  $f(s)$  ตาม ตารางที่ 5.11 ซึ่งจะเห็นได้ว่า ค่าต้นทุนต่ำสุดเกิดที่ H1 กราฟฟิก จะวิ่งผ่าน H1

ตารางที่ 5.11 พิจารณาที่โหนดเชียงใหม่

cmi	$h(s)$	$g(s)$	$f(s)$	Low Cost
H1	815.39	915.352	1730.742	1730.742
H2	760.93	1042.498	1803.428	
H3	1114.11	942.849	2056.959	

ปลายทาง H1 คือ โหนดคอซุชยา (aya) ซึ่ง ที่โหนดคอซุชยา จะมีการพิจารณาค่าต้นทุนต่ำสุด ในเส้นทางที่เกี่ยวข้องอีกครั้ง ได้ค่าตามตารางที่ 5.12 ซึ่งจะเห็นได้ว่ามีเพียงหนึ่งเส้นทางเท่านั้นที่จะ ส่งผ่านกราฟฟิกไปยัง โหนด หลักสี่ (lks) ตามเส้นทาง H4

ตารางที่ 5.12 พิจารณาที่โหนดอยุธยา

aya	$h(s)$	$g(s)$	$f(s)$	Low Cost
H4	760.93	934.292	1695.222	1695.222
H2	760.93	1042.498	1803.428	
H3	1114.11	942.849	2056.959	

โหนดหลักสี่ มีเส้นทางที่ต้องเลือก 6 เส้นทาง ในการส่งผ่านกราฟฟิกคือ H8, H9, H10, H11, H12, และ H13 จำนวน เส้นทางที่มีต้นทุนต่ำสุดได้ ตามตารางที่ 5.13 เส้นทางที่มีต้นทุนต่ำที่สุด คือ H12 กราฟฟิกจะวิ่งผ่านไปยังปลายทาง คือโหนด พระโขนง (pkg)

ตารางที่ 5.13 พิจารณาที่โหนดหลักสี่

lks	$h(s)$	$g(s)$	$f(s)$	Low Cost
H8	746.47	884.740	1631.210	
H9	746.47	245.295	991.765	
H10	746.47	229.128	975.598	
H11	742.29	170.385	912.675	
H12	740.73	81.511	822.241	822.241
H13	740.73	582.796	1323.526	
H2	760.93	1042.498	1803.428	
H5	760.93	1074.300	1835.230	

และ ที่พระโขนง มีเส้นทางเลือกอยู่ 10 เส้นทางคือ H14, H15, H20, H21, H22, H23 และ เส้นทางเดิมที่อาจจะส่งผ่านกราฟฟิกได้คือ H8, H9, H10 และ H 11 จำนวนค่าต้นทุนต่ำสุดได้ ตามตารางที่ 5.14 โดยเส้นทางที่มีต้นทุนน้อยที่สุดคือ H20 กราฟฟิกจะวิ่งผ่าน H 20 ไปหาปลายทาง ลาดหญ้า (lty)

ตารางที่ 5.14 พิจารณาที่โหนดพระโขนง

pkg	$h(s)$	$g(s)$	$f(s)$	Low Cost
H14	708.52	523.564	1232.084	
H15	740.73	133.799	874.529	
H20	742.29	77.037	819.327	819.327
H21	742.29	96.080	838.370	
H22	742.29	930.601	1672.891	
H23	708.52	631.940	1340.460	
H8	746.47	884.740	1631.210	
H9	746.47	245.295	991.765	
H10	746.47	229.128	975.598	
H11	742.29	170.385	912.675	

จากโหนด ลาดหญ้า มีเส้นทางเลือก 3 ทางคือ H24, H25, และ H26 ซึ่งมีค่าต้นทุนต่ำสุด ตามตารางที่ 5.15 เส้นทาง H25 จะถูกเลือกใช้ในการส่งผ่านทราฟฟิกไปยังโหนดหาดใหญ่ (hyi) ซึ่งเป็นโหนดปลายทาง

ตารางที่ 5.15 พิจารณาที่โหนดลาดหญ้า

ity	$h(s)$	$g(s)$	$f(s)$	Low Cost
H24	0.00	961.287	961.287	
H25	0.00	949.648	949.648	949.648
H26	677.91	567.111	1245.021	

จากผลการทดสอบ จะเห็นได้ว่า เส้นทางการวิ่งผ่านของทราฟฟิกคือ H1→H4→H12→H20→H25 แสดงในรูปที่ 5.16

แต่เนื่องจากโครงข่ายถูกต่อใช้งานเป็นโครงข่ายแบบโยแมงมุม จึงมีบางเส้นทางเป็นทางเดินลัด ในบทความวิทยานิพนธ์นี้จึงได้กำหนดเงื่อนไขใหม่ โดยไม่ให้เกิดการหาเส้นทางซ้ำทางเดิมที่เคยเดินทางผ่านไป แล้ว และให้มีการเปรียบเทียบเส้นทางก่อนระหว่างทางลัดกับทางปกติที่มีปลายทางที่โหนดเดียวกันตามรายละเอียดในบทที่ 3 รูปที่ 3.4 และ สมการที่ 3.6

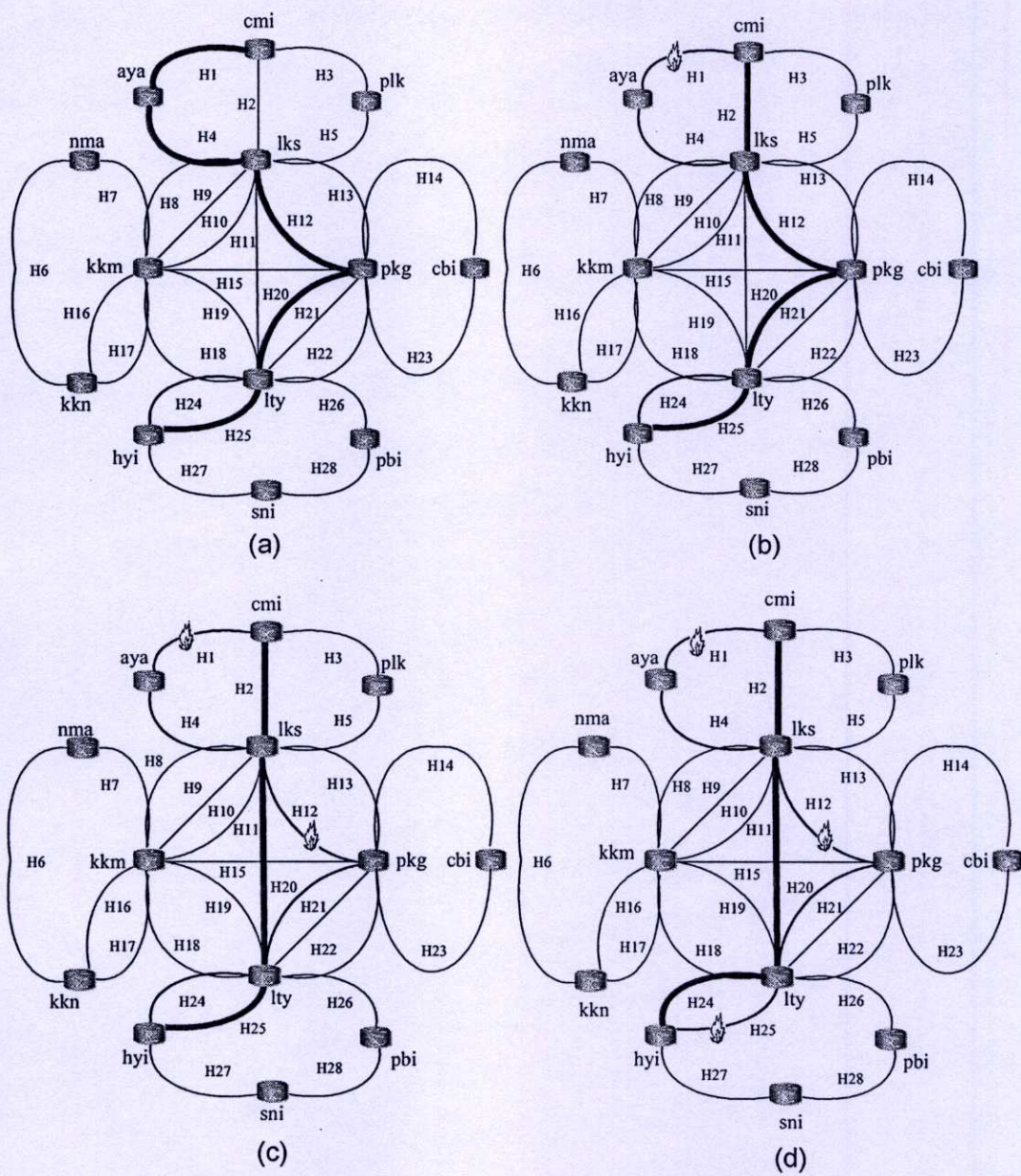
สำหรับ เส้นทาง อื่น ๆ เอสตาร์อัลกอริทึม จะสามารถหาได้ทำนองเดียวกัน โดยการนำไปเขียนโปรแกรมจำลองการทำงาน แสดงผลในหัวข้อต่อไป

## 5. 5 ผลการทดลองหาเส้นทางด้วย อัลกอริทึม เอสตาร์ กรณี ค่าถ่วงน้ำหนักเท่ากัน

ในหัวข้อนี้นำเสนอผลการหาเส้นทางด้วย อัลกอริทึมเอสตาร์ ตามเงื่อนไขที่กำหนดในตอนต้น แสดงตามรูปที่ 5.15-5.26 โดยกรณีโครงข่ายสมบูรณ์รูป (a) และกรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอป (Hop No.1) เส้นทางส่งผ่านของทราฟฟิก แสดงดังรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอป เส้นทางส่งผ่านของทราฟฟิก แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอป เส้นทางส่งผ่านของทราฟฟิก แสดงในรูป (d) ตามลำดับ

### 5.5.1 เส้นทางจาก เชียงใหม่ – หาดใหญ่

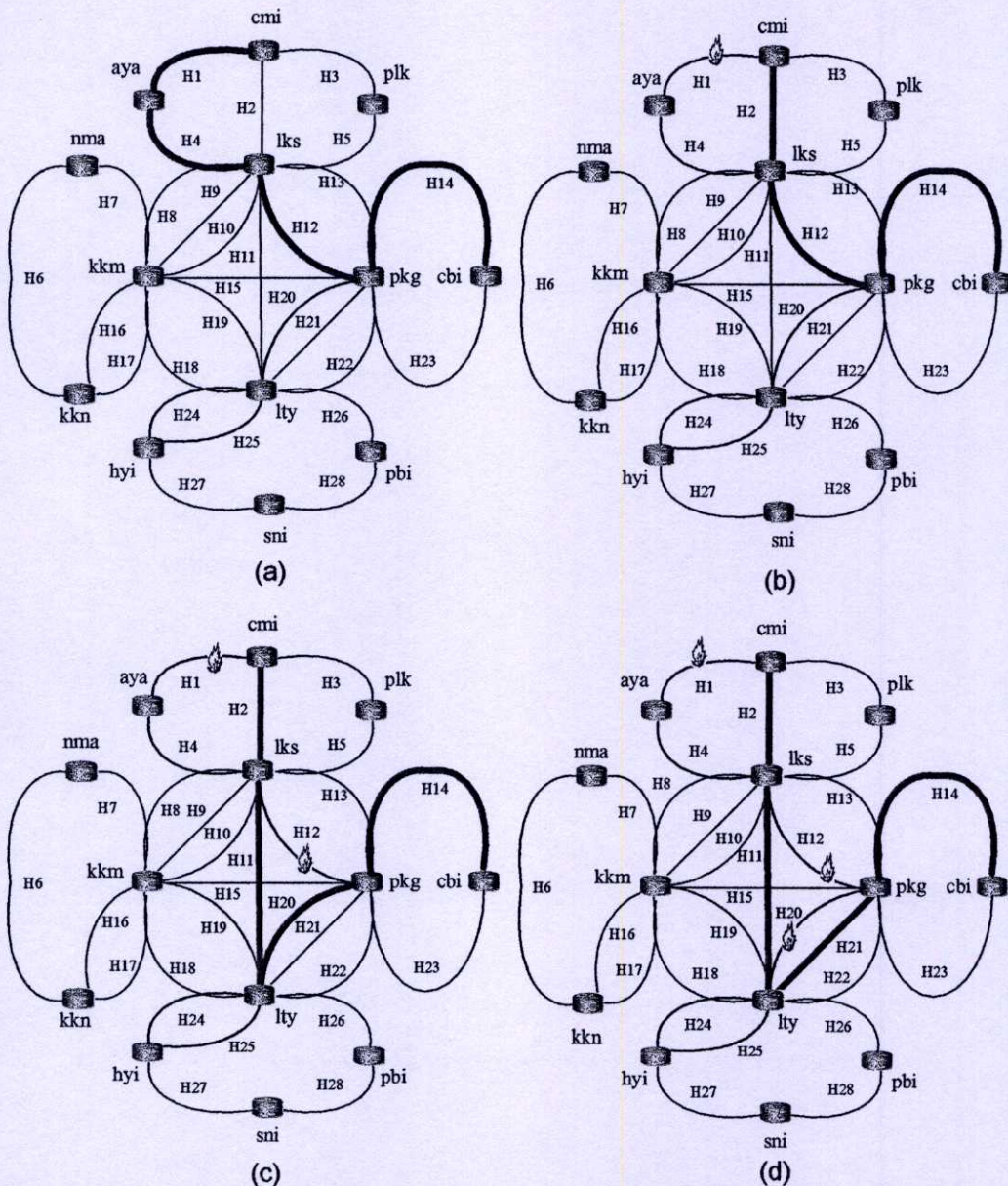
กรณีโครงข่ายสมบูรณ์ ไม่มีเหตุขัดข้องใด ๆ เส้นทางส่งผ่านของทราฟฟิก จะเดินทางผ่านฮอป H1→H4→H12→H20→H25 แสดงในรูปที่ 5.15 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอป (Hop No.1) เส้นทางส่งผ่านของทราฟฟิกจะผ่าน ฮอป H2→H12→H20→H25 แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอป (H1, H12) เส้นทางส่งผ่านของทราฟฟิกจะผ่าน ฮอป H2→H11→H25 แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอป (H1, H12, H25) เส้นทางส่งผ่านของทราฟฟิกจะผ่าน ฮอป H2→H11→H24 แสดงในรูป (d)



รูปที่ 5.15 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ให้ค่าถ่วงน้ำหนักเท่ากัน  
เส้นทาง เชียงใหม่-หาดใหญ่

### 5.5.2 เส้นทาง เชียงใหม่-ชลบุรี

กรณีโครงข่ายสมบูรณ์เส้นทาง การส่งผ่านของกราฟฟิคจะผ่านฮอป  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H14$  แสดงในรูปที่ 5.16 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอป (Hop No.1) เส้นทาง การส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H2 \rightarrow H12 \rightarrow H14$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอป (H1, H12) เส้นทาง การส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H2 \rightarrow H11 \rightarrow H20 \rightarrow H14$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอป (H1, H12, H20) เส้นทาง การส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H2 \rightarrow H11 \rightarrow H21 \rightarrow H14$  แสดงในรูป (d)



รูปที่ 5.16 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ให้ค่าดวงน้ำหนักเท่ากัน  
เส้นทาง เชียงใหม่-ชลบุรี

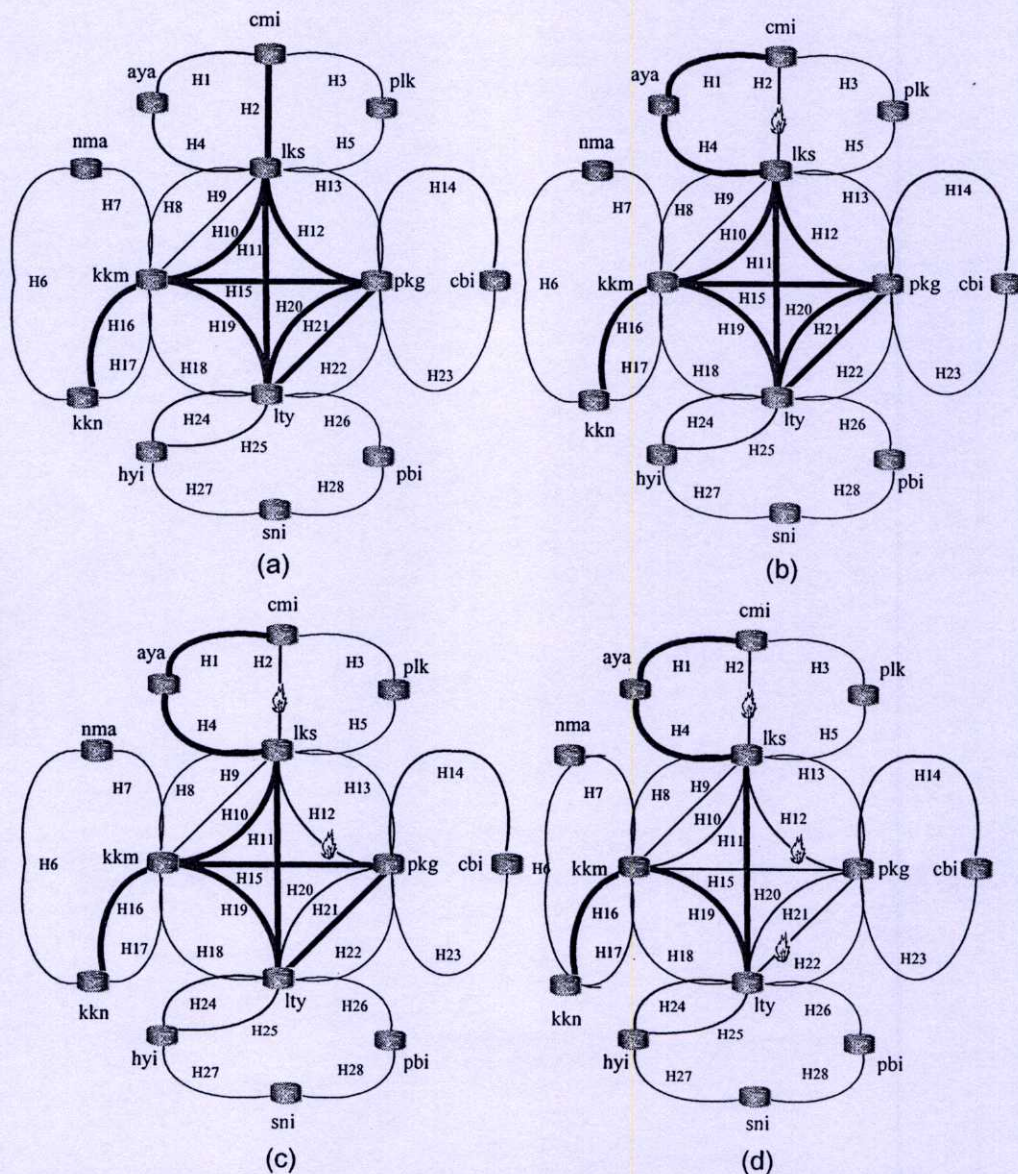






### 5.6.3 เส้นทาง เชียงใหม่-ขอนแก่น

กรณีโครงข่ายสมบูรณ์เส้นทางของการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H2 \rightarrow H12 \rightarrow H20 \rightarrow H21 \rightarrow H15 \rightarrow H19 \rightarrow H11 \rightarrow H10 \rightarrow H16$  แสดงในรูปที่ 5.20 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ ( $H2$ ) เส้นทางของการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H20 \rightarrow H21 \rightarrow H15 \rightarrow H19 \rightarrow H11 \rightarrow H10 \rightarrow H16$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ ( $H2, H12$ ) เส้นทางของการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H20 \rightarrow H21 \rightarrow H15 \rightarrow H19 \rightarrow H11 \rightarrow H10 \rightarrow H16$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ ( $H2, H12, H21$ ) เส้นทางของการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H20 \rightarrow H19 \rightarrow H15 \rightarrow H16$  แสดงในรูป (d)

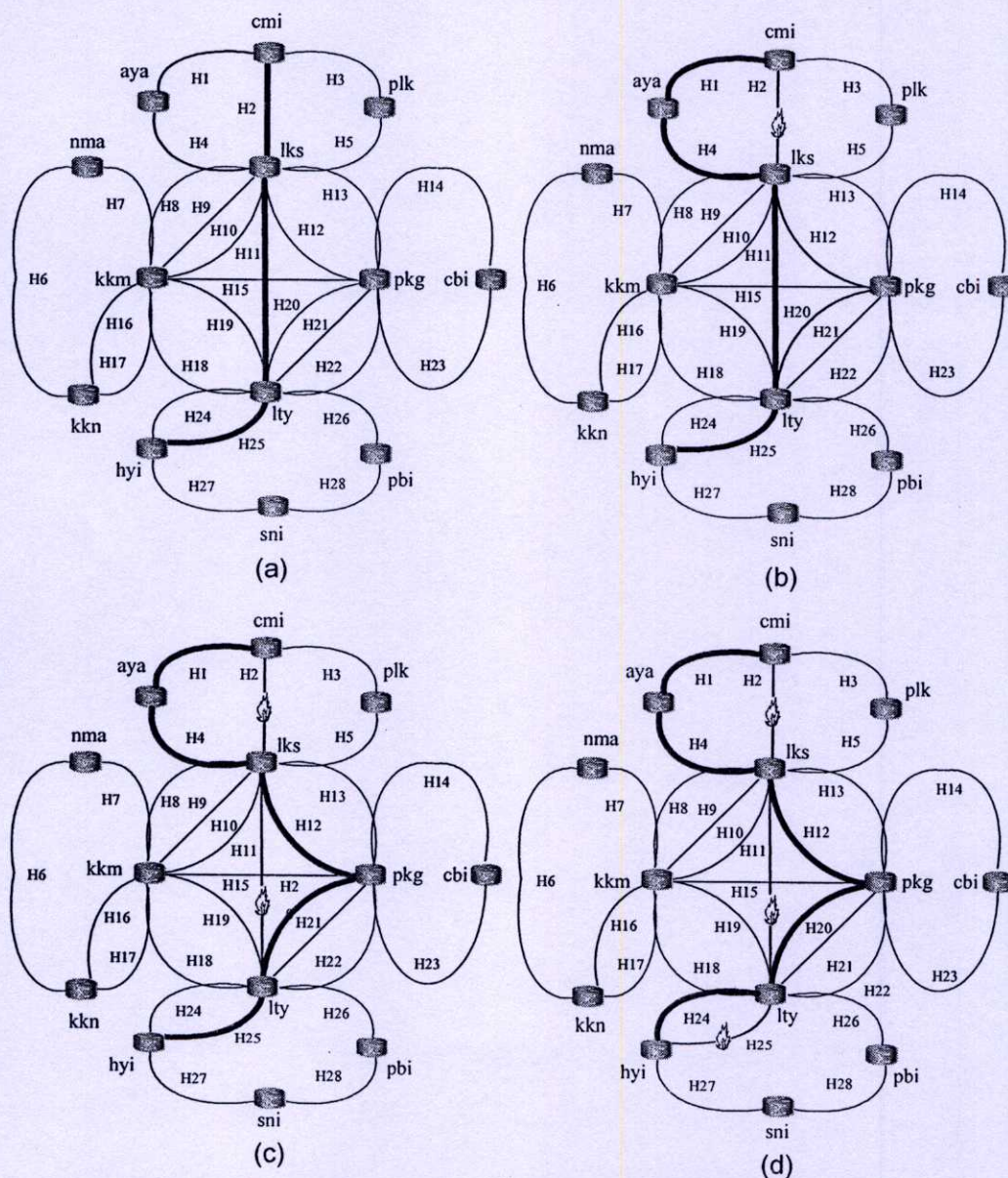


รูปที่ 5.20 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ให้ค่าถ่วงน้ำหนักมาก ไปทางกราฟฟิก  
เส้นทาง เชียงใหม่-ขอนแก่น

## 5.7 ผลของการหาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง โดยให้น้ำหนักเท่ากัน

### 5.7.1 เส้นทางเชียงใหม่ – หาดใหญ่

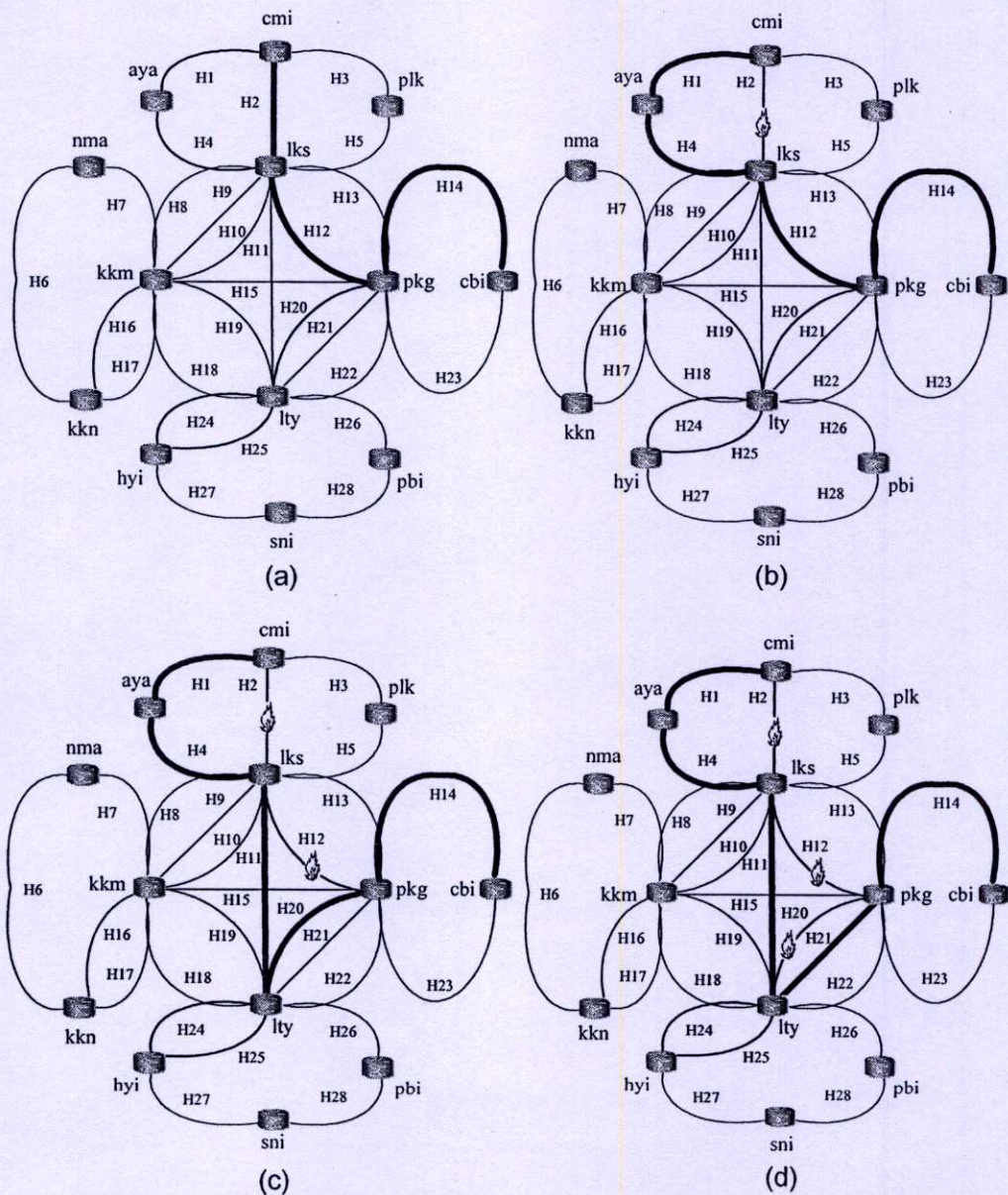
กรณีโครงข่ายสมบูรณ์เส้นทางการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H2 \rightarrow H11 \rightarrow H25$  แสดงในรูปที่ 5.21 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ ( $H2$ ) เส้นทางการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H25$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ ( $H2, H11$ ) เส้นทางการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H20 \rightarrow H25$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ ( $H2, H11, H25$ ) เส้นทางการส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H20 \rightarrow H24$  แสดงในรูป (d)



รูปที่ 5.21 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุงกรณีที่มีค่าถ่วงน้ำหนักเท่ากัน  
เส้นทาง เชียงใหม่-หาดใหญ่

### 5.7.2 เส้นทางเชียงใหม่-ชลบุรี

กรณีโครงข่ายสมบูรณ์เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H2 \rightarrow H12 \rightarrow H14$  แสดงในรูปที่ 5.22 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอป ( $H2$ ) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H14$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอป ( $H2, H12$ ) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H20 \rightarrow H14$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอป ( $H2, H12, H20$ ) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอป  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H21 \rightarrow H14$  แสดงในรูป(d)

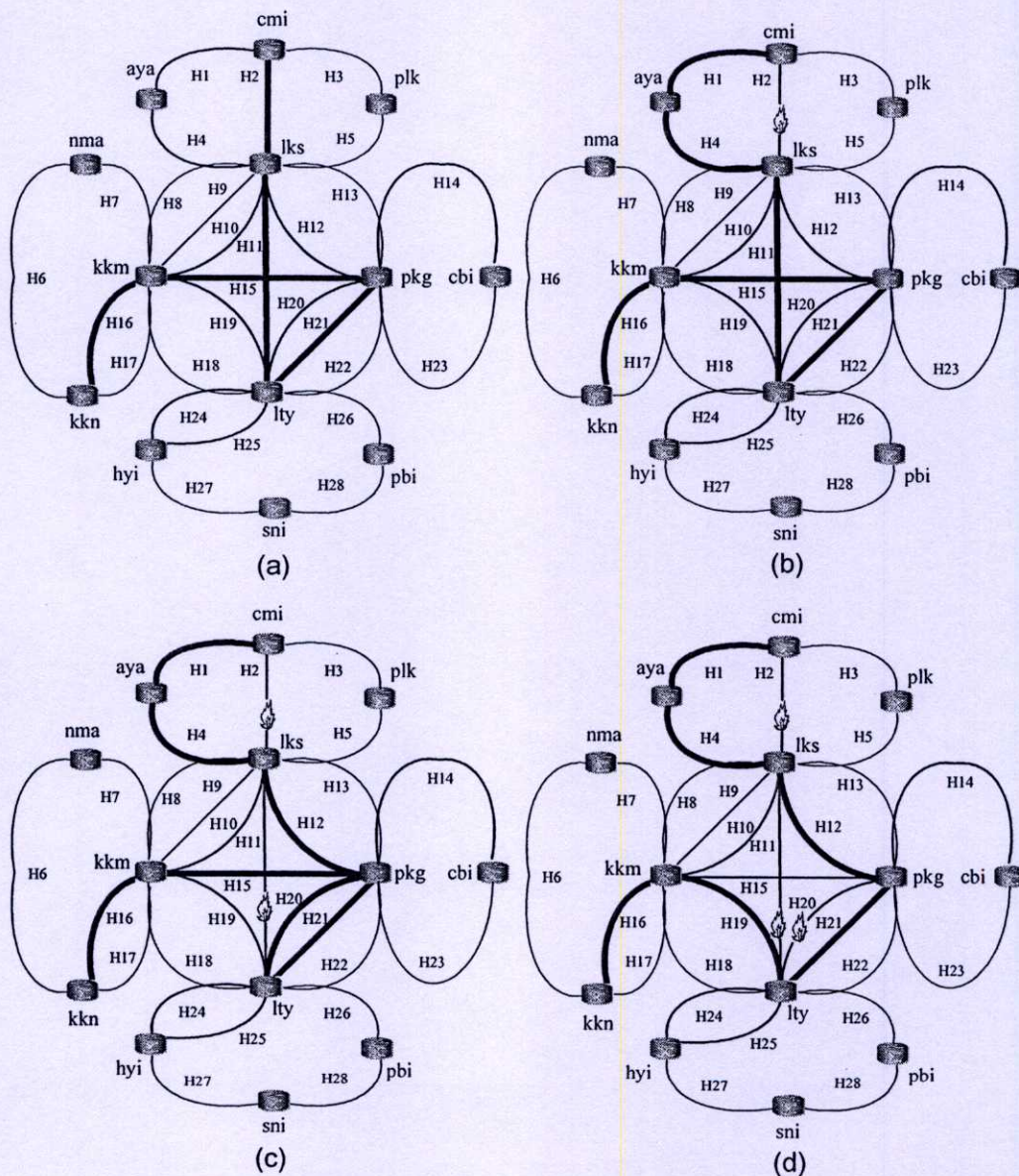


รูปที่ 5.22 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักเท่ากัน

เส้นทาง เชียงใหม่-ชลบุรี

5.7.3 เส้นทาง เชียงใหม่-ขอนแก่น

กรณีโครงข่ายสมบูรณ์เส้นทางของการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H2→H11→H21→H15→H16 แสดงในรูปที่ 5.23 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ (H2) เส้นทางของการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H11→H21→H15→H16 แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ (H2, H11) เส้นทางของการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H12→H20→H21→H15→H16 แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ (H2, H11, H20) เส้นทางของการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H12→H21→H19→H16 แสดงในรูป (d)

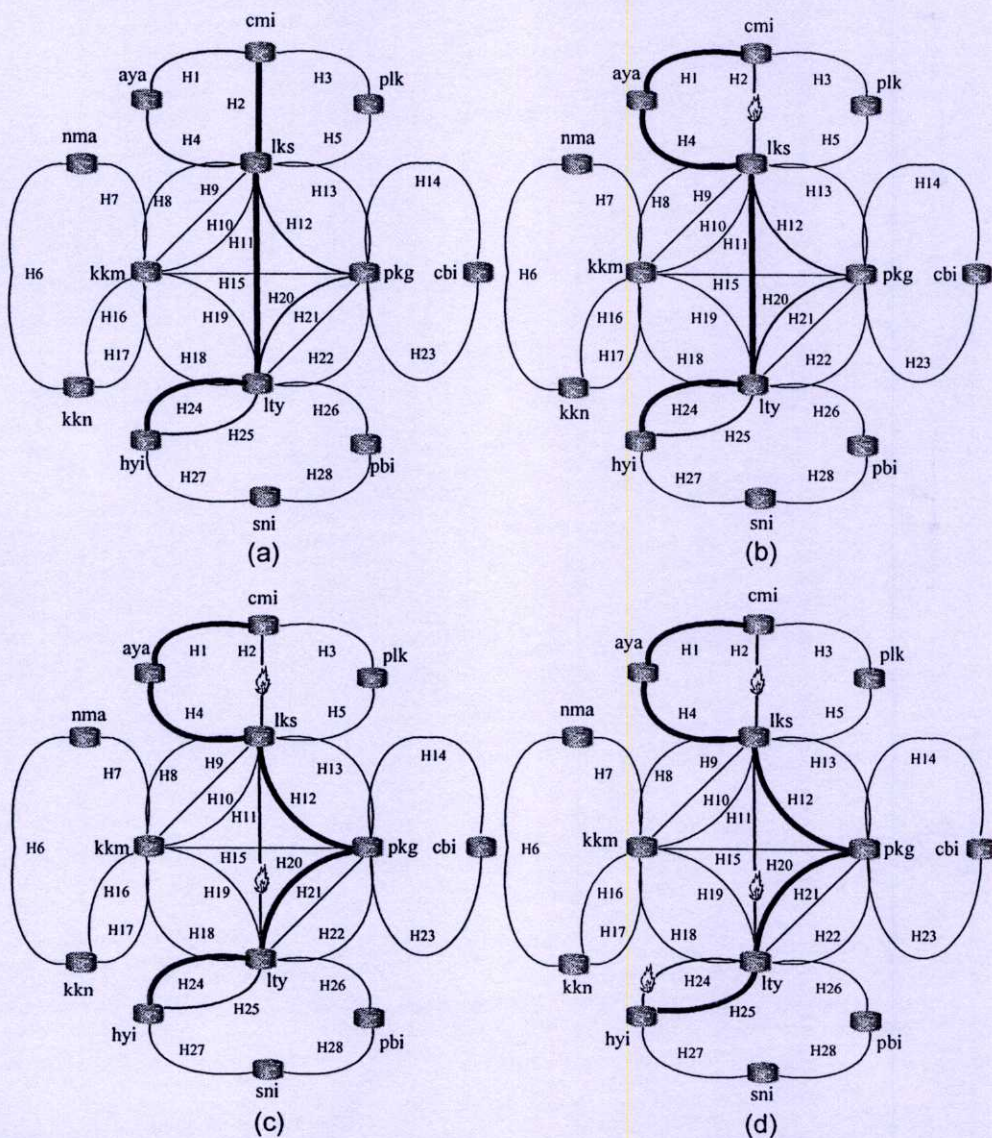


รูปที่ 5.23 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักเท่ากัน  
เส้นทาง เชียงใหม่-ขอนแก่น

## 5.8 ผลของการหาเส้นทางด้วยอัลกอริทึมเอสตาร์ที่ปรับปรุงโดยให้ค่าถ่วงน้ำหนักมากทางทราฟฟิก

### 5.8.1 เส้นทาง เชียงใหม่-หาดใหญ่

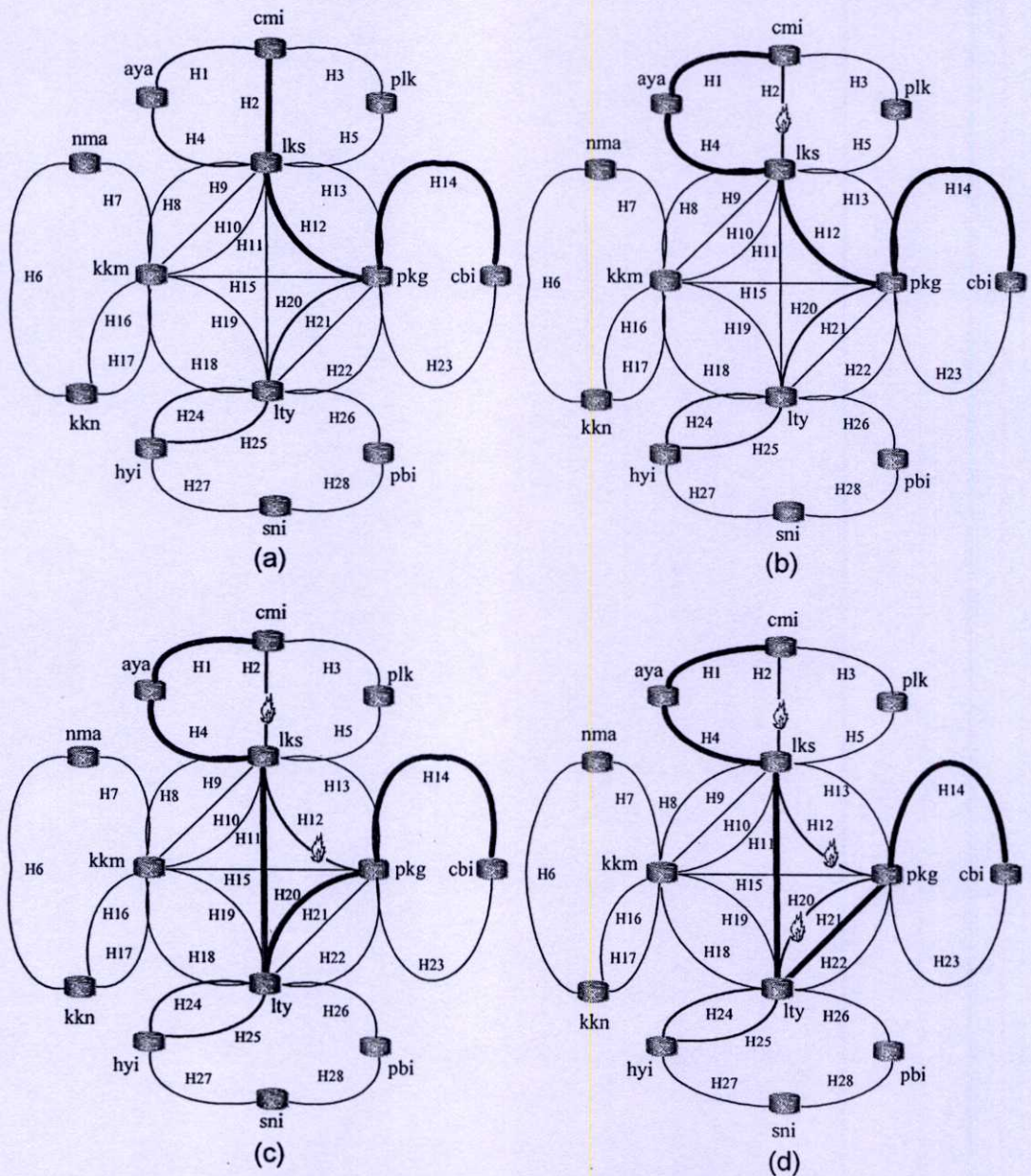
กรณีโครงข่ายสมบูรณ์เส้นทางที่ส่งผ่านของทราฟฟิกจะผ่าน ฮอบ  $H2 \rightarrow H11 \rightarrow H24$  แสดงในรูปที่ 5.24 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ ( $H2$ ) เส้นทางที่ส่งผ่านของทราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H24$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ ( $H2, H11$ ) เส้นทางที่ส่งผ่านของทราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H20 \rightarrow H24$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ ( $H2, H11, H24$ ) เส้นทางที่ส่งผ่านของทราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H20 \rightarrow H25$  แสดงในรูป (d)



รูปที่ 5.24 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนักมากไปทางทราฟฟิก เส้นทาง เชียงใหม่-หาดใหญ่

### 5.8.2 เส้นทาง เชียงใหม่-ชลบุรี

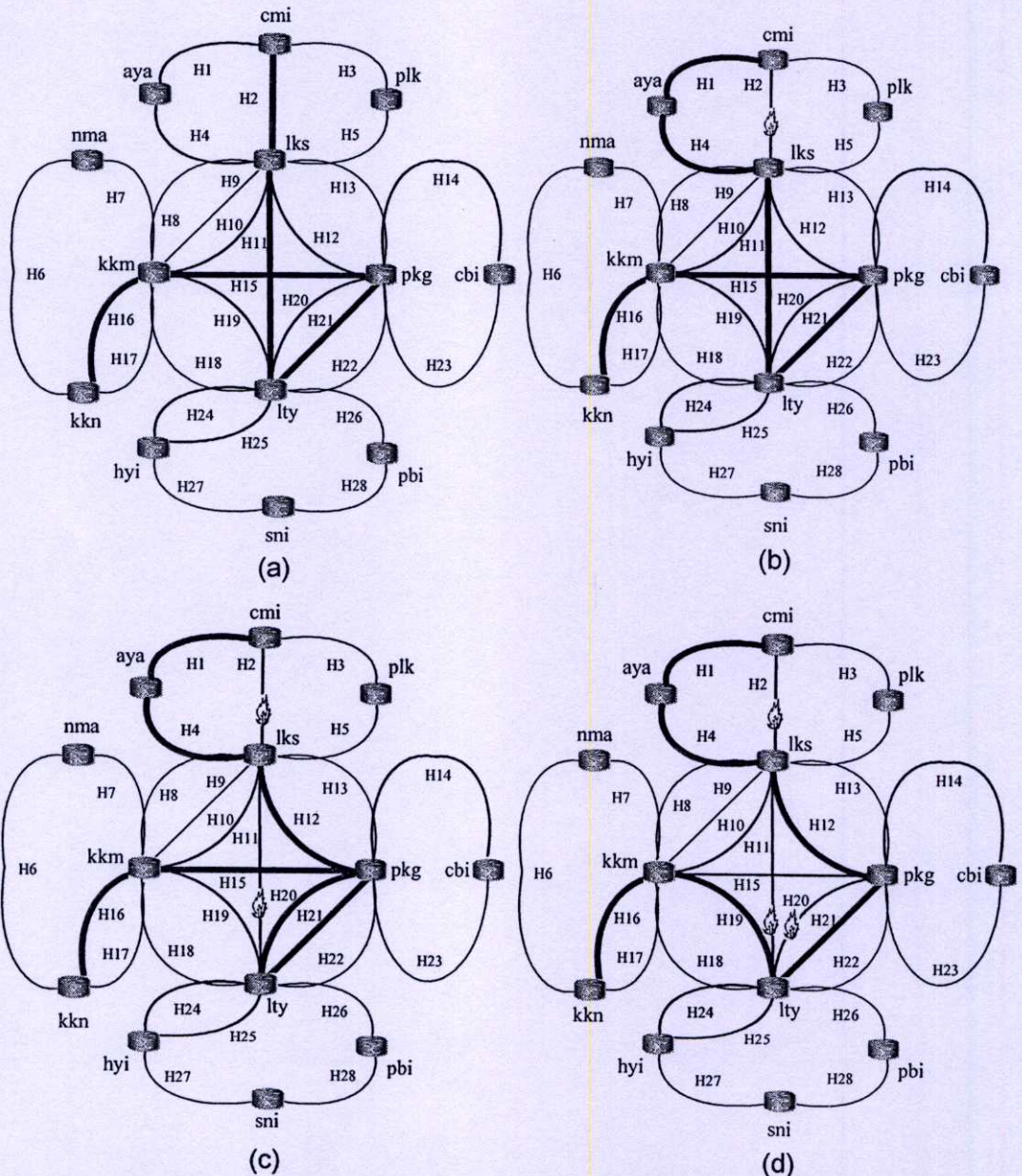
กรณีโครงข่ายสมบูรณ์เส้นทาง การส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H2 \rightarrow H12 \rightarrow H14$  แสดงในรูปที่ 5.25 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ ( $H2$ ) เส้นทาง การส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H12 \rightarrow H14$  แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ ( $H2, H12$ ) เส้นทาง การส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H20 \rightarrow H14$  แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ ( $H2, H12, H20$ ) เส้นทาง การส่งผ่านของกราฟฟิกจะผ่าน ฮอบ  $H1 \rightarrow H4 \rightarrow H11 \rightarrow H21 \rightarrow H14$  แสดงในรูป(d)



รูปที่ 5.25 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ให้ค่าถ่วงน้ำหนัก มากไปทางกราฟฟิก เส้นทาง เชียงใหม่-ชลบุรี

5.8.3 เส้นทาง เชียงใหม่-ขอนแก่น

กรณีโครงข่ายสมบูรณ์เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H2→H11→H21→H15→H16 แสดงในรูปที่ 5.26 (a) กรณีโครงข่ายมีเหตุขัดข้อง 1 ฮอบ (H2) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H11→H21→H15→H16 แสดงในรูป (b) กรณีโครงข่ายมีเหตุขัดข้อง 2 ฮอบ (H2, H11) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H12→H20→H21→H15→H16 แสดงในรูป (c) และ กรณีโครงข่ายมีเหตุขัดข้อง 3 ฮอบ (H2, H11, H20) เส้นทางการส่งผ่านของกราฟฟิคจะผ่าน ฮอบ H1→H4→H12→H21→H19→H16 แสดงในรูป (d)

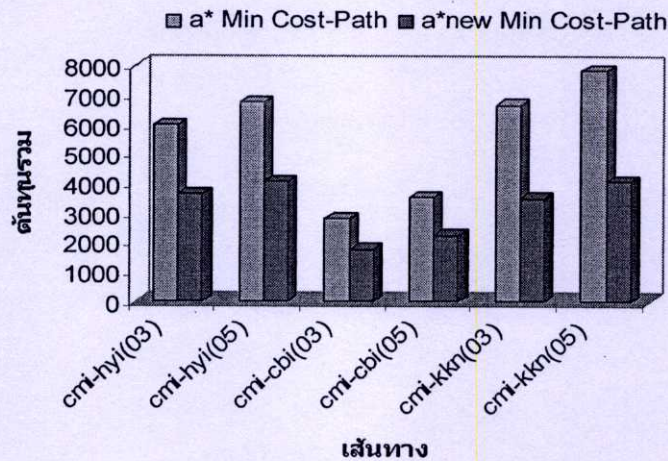


รูปที่ 5.26 หาเส้นทางด้วย อัลกอริทึม เอสตาร์ ที่ปรับปรุง กรณีที่ทำให้ค่าถ่วงน้ำหนักมากไปทางกราฟฟิค เส้นทาง เชียงใหม่-ขอนแก่น

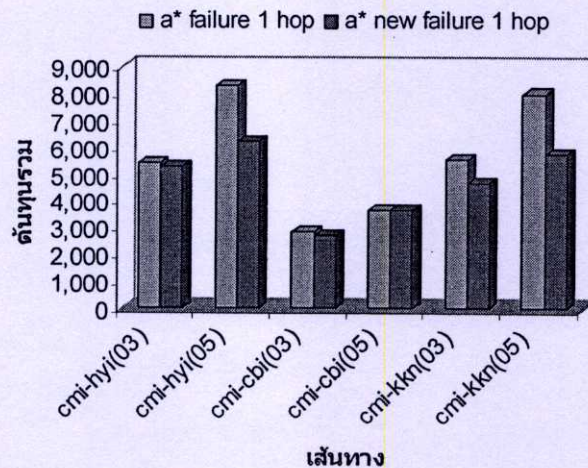
## 5.9 เปรียบเทียบอัลกอริทึม เอสตาร์ และอัลกอริทึม เอสตาร์ที่ปรับปรุง

ในหัวข้อนี้ทำการการเปรียบเทียบอัลกอริทึม เอสตาร์ และ อัลกอริทึม เอสตาร์ที่ได้รับการปรับปรุง โดยเปรียบเทียบประสิทธิภาพของระบบเป็น 2 แบบคือ เปรียบเทียบค่าต้นทุน กับ เปรียบเทียบ จำนวน ฮอป (Hop Count) ในแต่ละเส้นทางจากต้นทางถึงปลายทาง โดยกำหนดให้ กราฟแท่ง (03) แทนเส้นทางที่มีต้นทุนต่ำ กรณีที่กำหนดค่าถ่วงน้ำหนักเท่ากัน และ กราฟแท่ง (05) แทนเส้นทางที่มีต้นทุนต่ำ กรณีที่กำหนดให้ค่าถ่วงน้ำหนักมีน้ำหนักมาก ไปทางค่ากราฟฟิิก

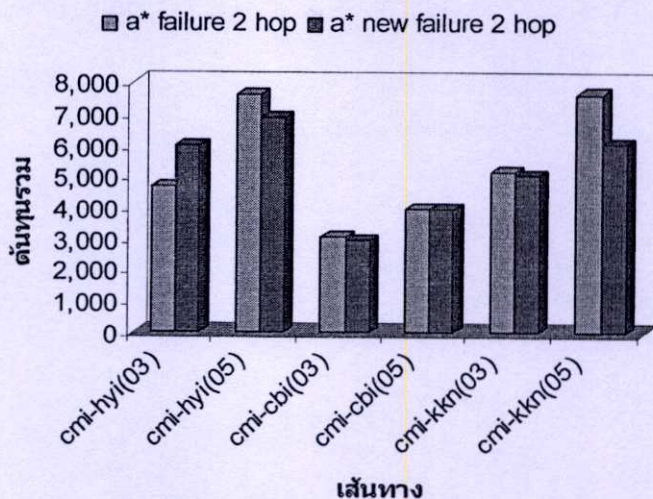
ผลการเปรียบเทียบค่าต้นทุนที่ต่ำที่สุด แสดงดังกราฟรูปที่ 5.27-5.30 ผลการเปรียบเทียบ จำนวนฮอป แสดงดังกราฟรูปที่ 5.31-5.32



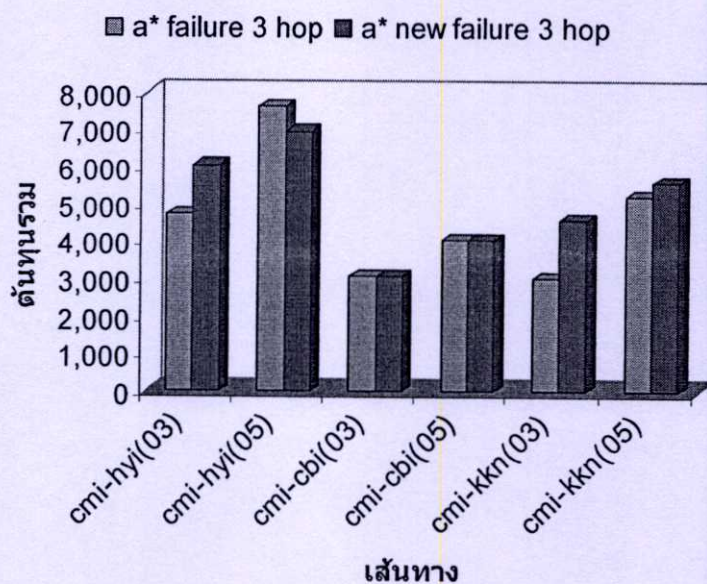
รูปที่ 5.27 เปรียบเทียบ ค่าต้นทุนของ เอสตาร์ กับ เอสตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายไม่มีเหตุขัดข้อง



รูปที่ 5.28 เปรียบเทียบ ค่าต้นทุนของ เอสตาร์ กับ เอสตาร์ที่ปรับปรุง ในแต่ละเส้นทาง กรณีโครงข่ายเกิดเหตุขัดข้อง 1 เส้นทาง



รูปที่ 5.29 เปรียบเทียบ ค่าต้นทุนของ เอสตาร์ กับ เอสตาร์ ที่ปรับปรุง ในแต่ละเส้นทาง กรณี โครงข่ายเกิดเหตุขัดข้อง 2 เส้นทาง

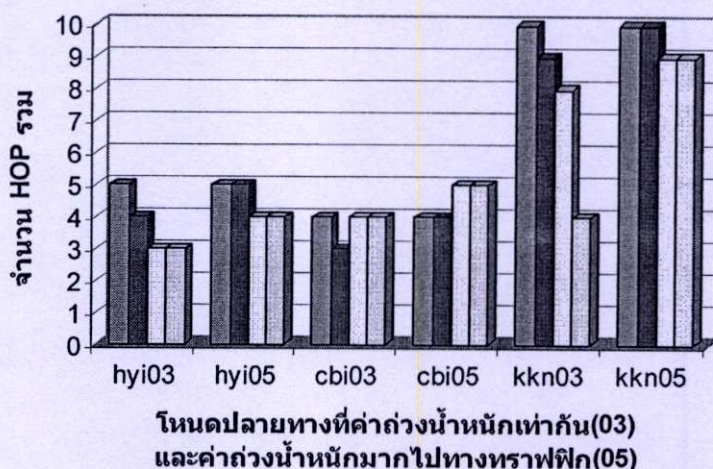


รูปที่ 5.30 เปรียบเทียบ ค่าต้นทุนของ เอสตาร์ กับ เอสตาร์ ที่ปรับปรุง ในแต่ละเส้นทาง กรณี โครงข่ายเกิดเหตุขัดข้อง 3 เส้นทาง

จากกราฟการเปรียบเทียบ จะเห็นได้ว่าในขณะที่โครงข่ายไม่มีเหตุขัดข้อง ผลของ อัลกอริทึม เอสตาร์ ที่ปรับปรุงใหม่ จะได้เส้นทางที่มีค่าต้นทุนต่ำกว่าอัลกอริทึม เอสตาร์ เสมอ และประสิทธิภาพจะลดลงเรื่อยๆ เมื่อมีเหตุขัดข้องตั้งแต่ 2 ฮอป ขึ้นไป

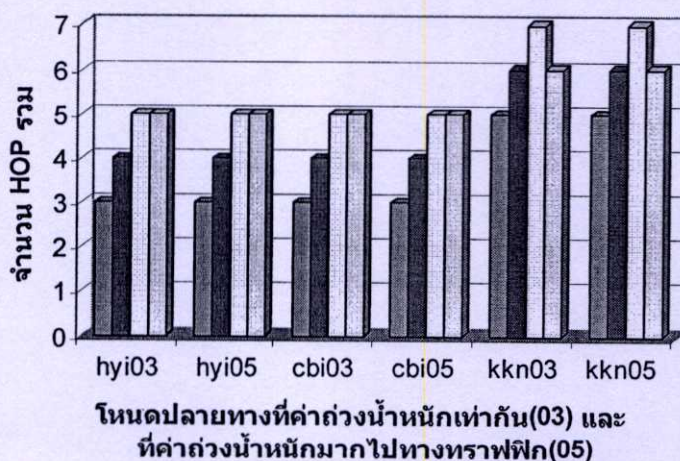
ในส่วนของการเปรียบเทียบจำนวน ฮอปที่แสดงใน รูปที่ 5.31 จะเห็นได้ว่า อัลกอริทึม เอสตาร์ นั้น เส้นทางที่มี ค่าต้นทุนต่ำไม่จำเป็นต้องมี จำนวน ฮอป น้อยเสมอไป ในขณะที่ อัลกอริทึม เอสตาร์ ที่ปรับปรุงใหม่ จะให้ผลตรงข้ามในกรณีที่โครงข่ายไม่มีเหตุขัดข้อง และประสิทธิภาพจะ ลดลงเรื่อย ๆ เมื่อมีเหตุขัดข้องตั้งแต่ 2 ฮอป เป็นต้นไป ดังแสดงใน รูปที่ 5.32

■ Min Cost-Path ■ Failure 1 Hop □ Failure 2 Hop □ Failure 3 Hop



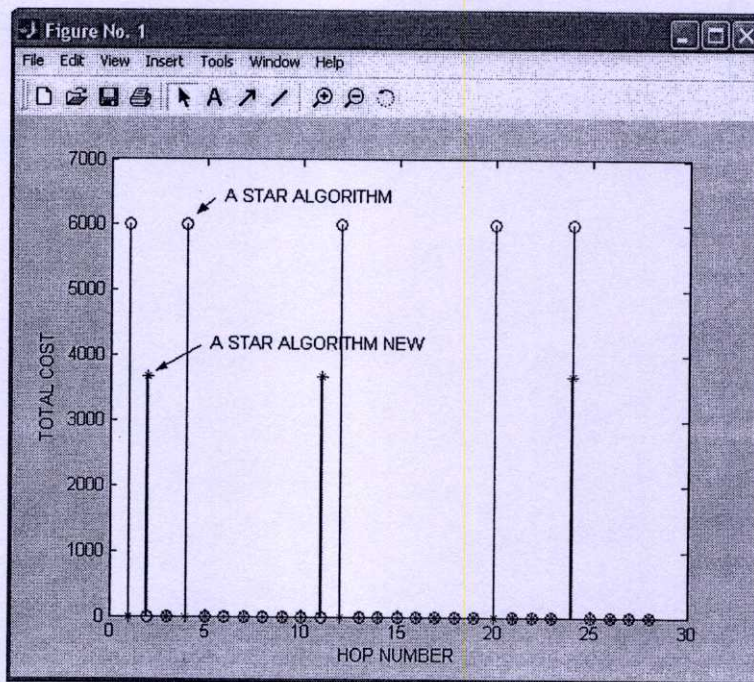
รูปที่ 5.31 เปรียบเทียบจำนวน ฮอป ในแต่ละเส้นทางระหว่างการคำนวณด้วย เอสตาร์ กรณี กำหนดค่าถ่วงน้ำหนักเท่ากัน และ กำหนดค่า ถ่วง น้ำหนักมาก ไปทางทราฟฟิก

■ Min Cost-Path ■ Failure 1 Hop □ Failure 2 Hop □ Failure 3 Hop



รูปที่ 5.32 เปรียบเทียบจำนวน ฮอป ในแต่ละเส้นทางระหว่างการคำนวณด้วย เอสตาร์ ที่ปรับปรุง กรณี กำหนดค่าถ่วงน้ำหนักเท่ากัน และกำหนดค่าถ่วงน้ำหนักมากไปทางทราฟฟิก

โดย รูปที่ 5.33 แสดงค่ากราฟเปรียบเทียบความสัมพันธ์ระหว่างค่าต้นทุน และจำนวน ฮอป ระหว่างอัลกอริทึมเอสตาร์ กับอัลกอริทึมเอสตาร์ที่ปรับปรุงแล้ว โดยการจำลองการทำงานของ โปรแกรม MATLAB ยกตัวอย่าง เส้นทาง เชียงใหม่-หาดใหญ่ ขณะที่โครงข่ายทำงานปกติไม่มี เหตุขัดข้องที่ฮอปใด ๆ โดยพิจารณาในกรณีที่ให้ค่าถ่วงน้ำหนักขององค์ประกอบในการหาต้นทุนที่ เท่ากัน



รูปที่ 5.33 กราฟเปรียบเทียบระหว่าง อัลกอริทึม เอสตาร์ กับอัลกอริทึม เอสตาร์ ที่ปรับปรุง

จากรูปที่ 5.33 แสดงให้เห็นว่า อัลกอริทึม เอสตาร์ มีค่าต้นทุนรวม 6017.180 ได้จำนวน ฮอป 5 ฮอป(H1, H4, H12, H20 และH24) ส่วนอัลกอริทึมเอสตาร์ ที่ได้รับการปรับปรุง มีค่าต้นทุน รวม 3665.750 ได้จำนวนฮอป 3 ฮอป(H2, H11 และH25) จะเห็นได้ว่าอัลกอริทึมเอสตาร์ ที่ได้รับการปรับปรุงแล้ว ให้ค่าต้นทุนรวม และจำนวนฮอปน้อยกว่าวิธีการของอัลกอริทึม เอสตาร์เดิม

## สรุปผลการวิจัยและข้อเสนอแนะ

### 6.1 บทสรุปผลการวิจัย

การวางแผนพัฒนาโครงข่ายในปัจจุบันมีความจำเป็นมากจะต้องใช้การพยากรณ์ ทั้งนี้เพื่อความเหมาะสมทำให้โครงข่ายมีประสิทธิภาพใช้งานได้ดีในอนาคตด้วย ประกอบกับการพยากรณ์มีการพัฒนาเทคนิคการพยากรณ์ที่เหมาะสมกับการใช้งาน ได้อย่างมีประสิทธิภาพมากขึ้น

งานวิจัยนี้ได้พิจารณา การพยากรณ์โอพี โดยใช้โครงข่ายประสาทเทียมแบบแพร่ย้อนกลับ ได้ทำการหาโมเดลที่เหมาะสมที่ใช้ในการทดสอบคือ 1 อินพุตเลเยอร์ 3 ฮิดเดนเลเยอร์ และ 1 เอาต์พุตเลเยอร์ แล้วใช้จำนวนรอบสูงสุดในการวนลูปคือ 180 ค่าโมเมนตัม 0.1 และค่าอัตราการเรียนรู้ 0.2 จะได้ค่าผิดพลาด (SSE) 0.023 ผลของการพยากรณ์แต่ละสอบเป็นเวลา 1 ปี (กรกฎาคม 51 – มิถุนายน 52) มีแนวโน้มของแต่ละเดือนเพิ่มขึ้น เมื่อได้ผลการพยากรณ์แล้ว นำผลการพยากรณ์แต่ละเส้นทาง ระยะทาง และค่าการสูญเสียในสายใยแก้วนำแสง มาเป็นองประกอบของค่าต้นทุน เพื่อหาเส้นทางที่มีค่าต้นทุนต่ำที่สุดโดยใช้อัลกอริทึม เอ สตาร์ ซึ่งค่าต้นทุนของอัลกอริทึมดังกล่าว ประกอบด้วยสองส่วนคือ ค่าคอร์ด กับค่าฮิวริสติก ทั้งนี้เพื่อนำเสนอใช้เป็นโครงข่ายสำรอง กรณีโครงข่ายเกิดเหตุขัดข้องแล้วไม่สามารถฟื้นตัวได้ตามเวลาที่กำหนด ทำให้ขาดประสิทธิภาพในการให้บริการ งานวิจัยนี้ยังได้ปรับปรุงอัลกอริทึมเอสตาร์ให้เหมาะสมกับโครงข่ายที่ใช้ในการวิจัยคือ การเพิ่มเงื่อนไขเพื่อให้เกิดการวนกลับ (Loop Free) และพิจารณาเส้นทางที่มีค่าต้นทุนต่ำที่สุดกรณีโครงข่ายปกติ เทียบกับกรณีที่โครงข่ายเกิดเหตุขัดข้อง 1 - 3 ครั้ง ของ 3 เส้นทางคือ เชียงใหม่ – หาดใหญ่ เชียงใหม่ - ชลบุรี และเชียงใหม่ – ขอนแก่น ซึ่งให้ผลค่อนข้างดีทั้งในด้านค่าต้นทุนต่ำที่สุด และจำนวนสอบ (Hop Count)

### 6.2 ข้อเสนอแนะและแนวทางในการพัฒนา

6.2.1 งานวิจัยนี้ได้ทำการศึกษาปริมาณกราฟฟิคโอพี ซึ่งเป็น โครงข่ายที่เกิดขึ้นใหม่ ทำให้การเก็บข้อมูลปริมาณกราฟฟิคได้น้อย ส่งผลให้การพยากรณ์ต้องอาศัยข้อมูลเฉลี่ยของกราฟฟิคในระบบโครงข่ายมาเป็นข้อมูลในการเรียนรู้ให้กับ โครงข่ายประสาทเทียม ถ้ามีข้อมูลในการเรียนรู้นั้นมากขึ้นความแม่นยำในการพยากรณ์ก็จะมากขึ้นด้วย

6.2.2 การใช้เวลาเฉลี่ยในการเก็บข้อมูลของโปรแกรม MRTG ในบางครั้ง บางสอบ มีการเปลี่ยนแปลงไม่เท่ากัน ทำให้เวลาเฉลี่ยของระบบโครงข่ายไม่เท่ากัน ส่งผลให้ปริมาณกราฟฟิคคลาดเคลื่อนตามไปด้วย

6.2.3 ค่าการสูญเสียในสายใยแก้ว ซึ่งนำมาใช้เป็นองค์ประกอบของค่าถ่วงน้ำหนักในการหาเส้นทาง ด้วยอัลกอริทึมเอสตาร์นั้น ได้จากการคำนวณค่าเฉลี่ยต่อระยะทาง งานวิจัยครั้งต่อไป เพื่อให้ได้ค่าที่ถูกต้อง อาจหาได้จากการวัดจริง

6.2.4 ในงานวิจัยครั้งต่อไปอาจใช้องค์ประกอบของค่าถ่วงน้ำหนักในการหาเส้นทาง ของอัลกอริทึมเอสตาร์ เพิ่มขึ้น เช่น ค่าความจุของระบบ ค่าเสถียรภาพของแต่ละฮอป จำนวนอุปกรณ์ ทวนสัญญาณ เป็นต้น

## เอกสารอ้างอิง

- [1] บุญเจริญ ศิริเนาวกุล “ปัญญาประดิษฐ์ Artificial Intelligence” กรุงเทพฯ สำนักพิมพ์ท็อป พ.ศ. 2551
- [2] Allan Reid (Cisco Academy), **WAN Technologies CCNA 4 Companion Guide**, Indianapolis USA, pp. 60 – 64, Aug. 2006.
- [3] ลัญฉกร วุฒิสัทติกุลกิจ “โครงข่ายอินเทอร์เน็ตและโพรโทคอลที่ซีพี/ไอพี”  
ห้องปฏิบัติการวิจัยโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์  
จุฬาลงกรณ์มหาวิทยาลัย พ.ศ. 2545
- [4] อาณัติ รัตนกรกุล “ก้าวสู่อาชีพผู้ดูแลระบบเครือข่ายคอมพิวเตอร์ในองค์กร(ภาคปฏิบัติ)”  
กรุงเทพฯ ซีเอ็ดดูเคชั่น พ.ศ. 2549
- [5] สัตยุทธิ์ สว่างวรรณ “เครือข่ายคอมพิวเตอร์ Computer Networks” กรุงเทพฯ เพียร์สันเอ็ดดูเคชั่น อินโดไชน่า พ.ศ. 2547
- [6] [http://www.cisco.com/en/US/tech/tk365/technologies\\_tech\\_note09186a0080094195.shtml](http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094195.shtml)
- [7] เอกสิทธิ์ วิริยะจารี “เรียนรู้ระบบเน็ตเวิร์กจากอุปกรณ์ของ Cisco ภาคปฏิบัติ”  
กรุงเทพฯ ซีเอ็ดดูเคชั่น พ.ศ. 2548
- [8] The Multi Router Traffic Grapher(On Line) <http://oss.oetiker.ch/mrtg>
- [9] Douglas R. Mauro and Kevin J. Schmidt **Essential SNMP**, United States of America, pp. 301-315, Sep. 2005.
- [10] จตุชัย แพงจันทร์ “เจาะระบบ Network 2 th Edition” นนทบุรี ไอดีซีฯ พ.ศ. 2551
- [11] Bing, L. and Siew-iiwse, C., “Finding the shortest route using case, knowledge, and Dijkstra’s algorithm,” The Proceeding of the 10<sup>Th</sup> Conference on Artificial Intelligence for Applications (CALA’94), Los Alamitos, pp. 149-155, Oct. 1994.
- [12] Bing, L. and Tay, J., “Using Knowledge about the road network for route finding,” The Proceeding of the 11<sup>Th</sup> Conference on Artificial Intelligence for Applications, pp. 306-312, Oct. 1995.
- [13] Kamoun, F.,and Mehmet Ali, M.K., “Aneural network shortest path algorithm for optimal routing in packet-switch communication networks,” Global Telecommunications Conference 1991, Vol. 1, pp. 120-124, Dec. 1991.

- [14] Gen, M., Runwei, C. and Dingwei, W., "**Genetic algorithm for solving shortest path problems,**" IEEE International Conference on Evolutionary Computation, pp. 401-406, Apr. 1997.
- [15] Chandwani, M. and Chaudhari, N.S., "**An algorithm for fuzzy control bases on shortest path framework,**" International Conference on Industrial Electronics, Vol.1, pp. 254-257, Nov. 1993.
- [16] Murthy, S. and Garcia-Luna-Aceves, J.J., "**Loop-Free Internet routing using hierarchical routing trees,**" Sixteenth Annual Joint Conference of the IEEE computer and Communication Societies, Driving the information Revolution, Vol.1 pp. 101-108, Apr. 1997.
- [17] Narvaez, P., Kai-Yeung, S. and Hong-Yi, T., "**Local restoration algorithm for link-state routing protocol,**" Proceeding of the Eight International Conference on Computer communication and networks, pp. 352-357, Oct. 1999.
- [18] Author of cisco Internetworking & Troubleshooting, **IP network design** Printed in the United States of America, pp.17-58, 2001.
- [19] Kiattisin S., Suwannapong T., and Chamnongthai K., "**A model of resource management In ATM network within effective bandwidth using random paths algorithm.,**" Accepted for The Fifth IFIP Conference on Intelligence in Network, pp. 473-486, Nov. 1999.
- [20] Simon Haykin, **Second Edition Neural Network a Comprehensive Foundation**, New Jersey, pp. 1-6, 1999.
- [21] Juan R. Rabunal and Julian Dorado, "**Artificial neural network in real-life applications,**" Yurchak Printing Inc., pp. 52-53, 2006.
- [22] Fredric M. Ham, Ivica Kostanic, **Principles of neurocomputing for Science & Engineering**, New York, pp. 27-35, 2001.
- [23] Robert J. Schalkoff, **Artificial Neural Networks**, Singapore, pp. 308 - 309, 1997
- [24] Fausett, Laurene V., **Fundamentals of neural network: architectures, algorithm, and application**, New Jersey, pp. 289-296, 1994.
- [25] T. Paungma, M. Sukkasem, W. Sangkhawijit, "**Utilizing the neural network model for traffic prediction at heavily loaded area of GSM system,**" CITA 01, pp. 280-288, 17-19 Oct. 2001.

ภาคผนวก

## ภาคผนวก

### บทความที่ได้รับการตีพิมพ์ในวิทยานิตน์

- [1] S. Satsri, S. Ardhan, V. Chutchavong and O. Sangaroon, “ANN based NGN IP Traffic Prediction in Thailand”, International Conference on Control, Automation and Systems, Seoul, Korea, pp. 2154-2157 Oct. 17-20, 2007.
- [2] S. Satsri, S. Ardhan, V. Chutchavong and O. Sangaroon, “IP Traffic Prediction in IP-based NGN in THAILAND by Neural Network”, International Conference on Engineering, Applied Sciences, and Technology, Bangkok, Thailand, pp.138–141, Nov. 21-23, 2007.
- [3] S. Ardhan, S. Satsri, V. Chutchavong and O. Sangaroon, “Improved Model for Traffic Fluctuation Prediction by Neural network” , International Conference on Control, Automation and Systems, Seoul, Korea, pp. 122-125, Oct. 17-20, 2007.

## ประวัติผู้เขียน

นายสุริยะ สัตย์ศรี เกิดเมื่อวันที่ 23 พฤศจิกายน พ.ศ.2511 ที่จังหวัดนครศรีธรรมราช สำเร็จการศึกษาปริญญารัฐประศาสนศาสตรบัณฑิต จากมหาวิทยาลัยสุโขทัยธรรมมาธิราช 2543 ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม จากสถาบันเทคโนโลยีราชมงคล (ธัญบุรี) ในปี พ.ศ. 2547 และปฏิบัติงานที่ บริษัท ทีโอที จำกัด (มหาชน) ปัจจุบันดำรงตำแหน่ง วิศวกร 5