สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# การตรวจหาข้อมูลประเภท P2P

# P2P TRAFFIC IDENTIFICATION

นายพงษ์ศักดิ์ เรืองดำ
นายรุ่งโรจน์ รัตนบุรี
นายสิริชัย แซ่ตั้ง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เลขหมู่.....
เลขทะเบียน.....103085
วัน,เดือน,ปี.....27 ส.ค. 2552
b.....12100456
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจหาข้อมูลประเภท P2P

P2P TRAFFIC IDENTIFICATION


ผู้จัดทำ

1. นายพงษ์ศักดิ์ เรืองดำ รหัสนักศึกษา 49015290
2. นายรุ่งโรจน์ รัตนบุรี รหัสนักศึกษา 49015298
3. นายสิริชัย แซ่ตั้ง รหัสนักศึกษา 47015340

อาจารย์ที่ปรึกษา

(อ. อำนาจ ขาวเน)


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การตรวจหาข้อมูลประเภท P2P

นายพงษ์ศักดิ์ เรืองดำ 49015290
นายรุ่งโรจน์ รัตนบุรี 49015298
นายสิริชัย แซ่ตั้ง 47015340
อ. อำนาจ ขาวเน อาจารย์ที่ปรึกษา
ปีการศึกษา 2551

## บทคัดย่อ

ในปัจจุบันมีการใช้งาน โปรโตคอลประเภท P2P กันอย่างแพร่หลาย ซึ่งทำให้เกิดความหนาแน่นในการรับส่งข้อมูลบนระบบเครือข่ายเป็นจำนวนมาก และส่งผลกระทบต่อระบบเครือข่ายหลายองค์กรจึงเริ่มให้ความสนใจในการตรวจหาข้อมูลของโปรโตคอลประเภท P2P โดยใช้วิธีการต่างๆ เพื่อเป็นแนวทางในการแก้ไขปัญหาที่เกิดจากโปรโตคอลประเภทP2P

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อวัตถุประสงค์ในการศึกษาและพัฒนาซอฟต์แวร์ที่ใช้ในการแยกประเภทข้อมูล ของโปรโตคอล P2P บนระบบเครือข่าย โดยใช้เทคนิคการเรียนรู้โดยใช้โครงข่ายประสาทเทียม เพื่อเรียนรู้ลักษณะของโปรโตคอลประเภท P2P เพื่อให้สามารถแยกโปรโตคอลประเภท P2P ออกจากโปรโตคอลประเภทอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# P2P TRAFFIC IDENTIFICATION

Pongsak Ruangdom 49015290
Rungrod Rutanaburee 49015298
Sirichai Saetang 47015340
Amnach Khawne Advisor
Academic Year 2008

## ABSTRACT

In now a day, Protocol P2P has been popularly used, which causes a huge of the traffic in transferring and receiving data on the network. It affects the network of organizations. Therefore, many organizations have paid attention in searching for the data of Protocol P2P by using several methods in order to be the problem solutions caused by Protocol P2P.

This thesis aims to study and develop the software which is used in classifying the data of Protocol P2P on the network. Beside, artificial nerve tactic is one of several ways to learn the appearance of Protocol P2P in order to classify Protocol P2P from the others.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ในการจัดทำโครงงานเรื่องการตรวจหาข้อมูลประเภท P2P นี้สามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำต้องขอบพระคุณอาจารย์ อำนาจ ขาวเน อาจารย์ที่ปรึกษา ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆจนสำเร็จลุล่วง

นอกจากนี้คณะผู้จัดทำขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนในด้านต่างๆจนการทำงานสำเร็จด้วยดี รวมทั้งขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ ไว้ ณ ที่นี้

พงษ์ศักดิ์ เรืองดำ
รุ่งโรจน์ รัตนบุรี
สิริชัย แซ่ตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... I

บทคัดย่อภาษาอังกฤษ ..... II

กิตติกรรมประกาศ ..... III

สารบัญ ..... IV

สารบัญตาราง ..... VIII

สารบัญรูปภาพ ..... IX

บทที่ 1 บทนำ

1.1 ความเป็นมาของปัญหา ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ประโยชน์ที่คาดว่าจะได้รับ ..... 1

1.4 ขอบเขตของโครงงาน ..... 2

1.5 วิธีการดำเนินงาน ..... 2

บทที่ 2 Internet Protocol (IP)

2.1 Internet Protocol (IP) ..... 3

2.1.1 Internet Protocol (IP) ..... 3

2.1.2 IP Header ..... 3

2.1.3 Internet Control Message Protocol (ICMP) ..... 5

2.1.4 ICMP Header ..... 6

2.2 TCP Protocol ..... 8

2.2.1 TCP Protocol ..... 8

2.2.2 TCP Header ..... 9

2.3 UDP Protocol ..... 11

2.3.1 UDP Protocol ..... 11

2.3.2 UDP Header ..... 11

2.4 Network Programming ..... 12

2.4.1 Raw Sockets ..... 12

2.4.2 ICMP Protocol ..... 13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


บทที่ 3 Client-Server Model and P2P

3.1 Client-Server Model ..... 17

3.2 P2P Model ..... 20

3.2.1 Bittorrent Protocol ..... 24

3.2.2 องค์ประกอบของโปรโตคอล Bittorrent ..... 24

3.2.2.1 Torrent client ..... 24

3.2.2.2 Tracker server ..... 24

3.2.2.3 Torrent file ..... 24

บทที่ 4 โครงข่ายประสาทเทียม (Artificial Neural Network)

4.1 โครงข่ายประสาทเทียม (Artificial Neural Network) ..... 26

4.2 ฟังก์ชันกระตุ้นความสนใจ (Activation Function) ..... 28

4.2.1 สเตปฟังก์ชัน (Step function) ..... 28

4.2.2 ซิกมอยด์ฟังก์ชัน (Sigmoid function) ..... 29

4.2.3 ไฮเปอร์โบลิกแทนเจนฟังก์ชัน (Hyperbolic tangent function) ..... 30

4.3 การฝึกสอนให้กับโครงข่ายประสาทเทียม(Training of Artificial Neural Networks)30

4.4 วัตถุประสงค์ของการฝึกฝน (Objective of Training) ..... 31

4.4.1 การฝึกฝนแบบควบคุม (Supervised Training) ..... 31

4.4.2 การฝึกฝนแบบอิสระ(Unsupervised Training) ..... 31

4.5 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Artificial Neural Networks) .. 32

4.6 โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Artificial Neural Networks)..... 33

4.7 เพอร์เซ็ปตรอน (Perceptrons) ..... 34

4.8 แบคพรอพเกชัน ..... 35

4.8.1 การพัฒนาประสิทธิภาพของแบคพรอพเกชัน ..... 37

4.9 Recurrent Networks ..... 37

4.10 Hopfield Network ..... 38

4.11 Bidirectional Associative Memories (BAMs) ..... 39

4.12 Kohonen Maps ..... 39



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า

4.13 ข้อดีของโครงข่ายประสาทเทียม ..... 40
4.14 ข้อเสียของโครงข่ายประสาทเทียม ..... 40
4.15 การพัฒนาของโครงข่ายประสาทเทียม ..... 40

บทที่ 5 การออกแบบและพัฒนา

5.1 การออกแบบ ..... 41
5.2 การเขียนโปรแกรมดักจับPacket ..... 42
5.2.1 การพัฒนา Program โดยใช้ RAW Socket ..... 42
5.2.2 WinPcap Library ..... 42
5.3 การหาค่า Parameter ..... 42
5.3.1 โครงสร้างของ ตัวแปรที่ใช้เก็บค่าข้อมูล ..... 42
5.3.2 นำค่าข้อมูลที่ได้จากการ Capture มาเก็บในโครงสร้างตัวแปรที่สร้างขึ้น ..... 42
5.3.3 นำค่าที่อยู่ในตัวแปรมาคำนวณหาค่า Parameter ที่ใช้ในการ Train ..... 42
5.4 Neural network Training ..... 42
5.5 Traffic Classification ..... 43
5.6 การจัดเก็บข้อมูล ..... 43
5.6.1 ตาราง TRAFFIC ..... 43
5.6.2 ตาราง WEEK ..... 45
5.7 การแสดงผล ..... 46
5.7.1 การแสดงผล CURRENT HOURS ..... 47
5.7.2 การแสดงผล CURRENT DAY ..... 48
5.7.3 การแสดงผล PREVIOUS 7 DAY ..... 49
5.7.4 การแสดงผล PREVIOUS 30 DAY ..... 50
5.7.5 การแสดงผล PREVIOUS 12 MONTHS ..... 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า

บทที่ 6 การทดลองและผลการทดลอง

6.1 การตั้งอุปกรณ์ดักจับ Packet ..... 52

6.2 Capture Packet ..... 55

6.3 Parameter ที่ได้จากการทำงาน ..... 55

6.4 Neural Network Training ..... 56

6.5 Traffic Classification ..... 57

6.6 ผลการทดลอง ..... 57

บทที่ 7 สรุปผลการทดลอง

7.1 บทสรุป ..... 59

7.2 ปัญหาอุปสรรคและแนวทางแก้ไข ..... 59

7.3 แนวทางการพัฒนาต่อ ..... 60

บรรณานุกรม ..... 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หน้า

ตารางที่ 2.1 อธิบายส่วนต่างๆของ IP Head ..... 4
ตารางที่ 2.2 ค่า TYPE ของ ICMP ..... 6
ตารางที่ 2.3 ค่า TYPE ของ ICMP ..... 7
ตารางที่ 3.1 ตัวอย่างของหมายเลข Port มาตรฐาน ..... 18
ตารางที่ 3.2 P2P Protocol ..... 21
ตารางที่ 5.1 เปรียบเทียบความเร็วในการประมวลผล ..... 44
ตารางที่ 5.2 เปรียบเทียบความเร็วในการประมวลผล ..... 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

หน้า


รูปที่ 2.1 IP Header ..... 3
รูปที่ 2.2 ICMP Header ..... 6
รูปที่ 2.3 TCP Header ..... 9
รูปที่ 2.4 Flag ..... 9
รูปที่ 2.5 Three-way handshaking ..... 10
รูปที่ 2.6 Modified three-way handshake for closing a TCP connection ..... 10
รูปที่ 2.7 UDP Header ..... 11
รูปที่ 3.1 Client-Server Model ..... 19
รูปที่ 3.2 P2P Model ..... 20
รูปที่ 4.1 โครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ ..... 26
รูปที่ 4.2 ไดอะแกรมนิวรอลที่สร้างขึ้น (Artificial Neuron) ..... 27
รูปที่ 4.3 กราฟที่ได้จากฟังก์ชันของสเตปฟังก์ชัน ..... 28
รูปที่ 4.4 กราฟที่ได้จากฟังก์ชันของซิกมอยด์ฟังก์ชัน ..... 29
รูปที่ 4.5 กราฟที่ได้จากฟังก์ชันของไฮเปอร์โบลิกแทนเจนฟังก์ชัน ..... 30
รูปที่ 4.6 ลักษณะโครงข่ายประสาทเทียมแบบชั้นเดียว(Single-Layer Neural Networks) ..... 33
รูปที่ 4.7 โครงข่ายแบบฟีดฟอร์เวิร์ด (Feed-forward network) ..... 34
รูปที่ 4.8 แผนภาพของเพอร์เซ็ปตรอน ..... 34
รูปที่ 4.9 แผนภาพของโครงข่ายแบคพรอพเกชันแบบสองชั้น ..... 35
รูปที่ 5.1 โครงสร้างภาพรวมของโปรแกรม ..... 41
รูปที่ 5.2 ER ไดอะแกรมตารางTRAFFIC ..... 44
รูปที่ 5.3 ER ไดอะแกรมตารางWEEK ..... 45
รูปที่ 5.4 ตัวอย่างการแสดงผลผ่าน WEB BROWSER ..... 46
รูปที่ 5.5 ตัวอย่างการแสดงผล CURRENT HOURS IN ..... 47
รูปที่ 5.6 ตัวอย่างการแสดงผล CURRENT HOURS OUT ..... 47
รูปที่ 5.7 ตัวอย่างการแสดงผล CURRENT DAY IN ..... 48
รูปที่ 5.8 ตัวอย่างการแสดงผล CURRENT DAY OUT ..... 48
รูปที่ 5.9 ตัวอย่างการแสดงผล PREVIOUS 7 DAY IN ..... 49
รูปที่ 5.10 ตัวอย่างการแสดงผล PREVIOUS 7 DAY OUT ..... 49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 5.11 ตัวอย่างการแสดงผล PREVIOUS 30 DAY IN ..... 50
รูปที่ 5.12 ตัวอย่างการแสดงผล PREVIOUS 30 DAY OUT ..... 50
รูปที่ 5.13 ตัวอย่างการแสดงผล PREVIOUS 12 MONTHS IN ..... 51
รูปที่ 5.14 ตัวอย่างการแสดงผล PREVIOUS 12 MONTHS OUT ..... 51
รูปที่ 6.1 Test Base Model ..... 52
รูปที่ 6.2 การติดตั้ง Bridge Connection 1 ..... 53
รูปที่ 6.3 การติดตั้ง Bridge Connection 2 ..... 53
รูปที่6.4 การติดตั้ง Bridge Connection 3 ..... 54
รูปที่ 6.5 การติดตั้ง Bridge Connection 4 ..... 54
รูปที่ 6.6 การ Run Program Capture Packet ..... 55
รูปที่ 6.7 หน้าจอเมื่อ Train เสร็จสิ้น ..... 56
รูปที่ 6.8 กราฟที่ใช้แสดงผล Traffic out ..... 58
รูปที่ 6.9 กราฟที่ใช้แสดงผล Traffic in ..... 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มา

ปัจจุบันอินเทอร์เน็ต (Internet) ได้เข้ามามีบทบาทสำคัญในชีวิตประจำวัน สืบเนื่องมาจากอินเทอร์เน็ตนั้นเปรียบเสมือนคลังความรู้ขนาดใหญ่ที่รวบรวมข้อมูลต่างๆ ทั่วทุกมุมโลก โดยที่ผู้ใช้ไม่ต้องเสียเวลาเดินทางไปเสาะหาข้อมูลด้วยตนเอง ผู้ใช้เพียงแค่ต่ออินเทอร์เน็ตเพียงเท่านี้ก็สามารถท่องโลกกว้างได้อย่างสะดวกสบายด้วยเหตุนี้จึงทำให้จำนวนผู้ใช้งานอินเทอร์เน็ตมีจำนวนเพิ่มมากขึ้น แต่เนื่องจากแบนด์วิดท์ (Bandwidth) ของแต่ละองค์กรนั้นมีจำนวนจำกัดและไม่ได้มีการจัดสรรการใช้งานอย่างมีประสิทธิภาพ จึงทำให้เกิดปัญหาการแย่งกันใช้แบนด์วิดท์ โดยเฉพาะในปัจจุบันมีการพัฒนาโปรโตคอลประเภท P2P ขึ้นซึ่งมีผลก่อให้เกิดความหนาแน่นในการรับส่งข้อมูลบนเครือข่ายเป็นจำนวนมากและอาจมีผลกระทบทำให้การรับส่งข้อมูลใหม่ที่เกิดขึ้น ทำได้ช้าและไม่คล่องตัวและส่งผลกระทบให้ผู้ใช้คนอื่นๆ ภายในองค์กรไม่สามารถเข้าถึงข้อมูลในระบบอินเทอร์เน็ตได้ หรือเข้าได้อย่างยากลำบาก

## 1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการทำงานของระบบเครือข่ายและโปรโตคอลที่ใช้ในการติดต่อสื่อสาร

1.2.2 เพื่อศึกษาการทำงานของโปรโตคอลประเภท P2P

1.2.3 เพื่อศึกษาการแยกประเภทของข้อมูลในเครือข่ายด้วยวิธีการต่างๆ

1.2.4 เพื่อศึกษาการทำงานของโครงข่ายประสาทเทียม

1.2.5 เพื่อพัฒนาซอฟต์แวร์ ที่ใช้ในการแยกประเภท โปรโตคอลประเภท P2P

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 เข้าใจการทำงานของระบบเครือข่ายและโปรโตคอลที่ใช้ในการติดต่อสื่อสาร

1.3.2 เข้าใจการทำงานของโปรโตคอลประเภท P2P

1.3.3 เข้าใจการแยกประเภทของข้อมูลในเครือข่ายด้วยวิธีการต่างๆ

1.3.4 เข้าใจการทำงานของโครงข่ายประสาทเทียม

1.3.5 สามารถพัฒนาซอฟต์แวร์ที่ทำงานบนระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขอบเขตของงานวิจัย

โครงงานนี้จะทำการศึกษาและพัฒนาซอฟต์แวร์สำหรับการแยกประเภทของข้อมูลในชั้นโปรแกรมประยุกต์ (Application Layer) โดยใช้โครงข่ายประสาทเทียม ในการแยกแยะประเภทของข้อมูลที่ผ่านเข้าออกในระบบเครือข่าย โดยสามารถแสดงผลชนิดและปริมาณของข้อมูลที่ได้ผ่านการแยกประเภทแล้ว

## 1.5 วิธีการดำเนินงาน

1.5.1 โครงงานนี้เริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่างๆที่เกี่ยวข้อง

1.5.2 ออกแบบและพัฒนาระบบ

1.5.3 ทดสอบและวิเคราะห์การทำงานของระบบ

1.5.4 สรุปผลการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# Internet Protocol (IP)

## 2.1 Internet Protocol (IP)

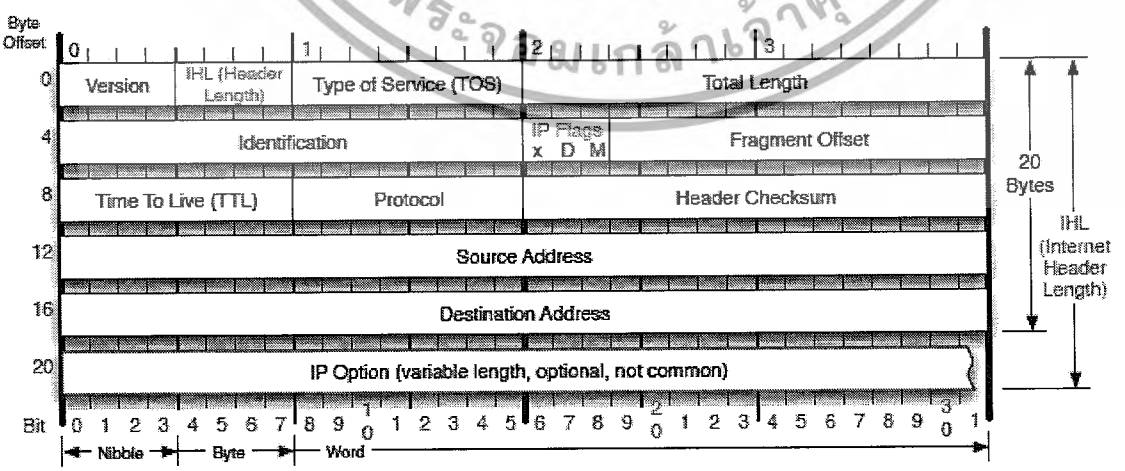
### 2.1.1 Internet Protocol (IP)

IP เป็นโปรโตคอลที่ทำหน้าที่รับภาระในการนำข้อมูลไปส่งยังผู้รับ ที่เชื่อมต่ออยู่ในระบบ Network ซึ่งทั้งสองฝั่งอาจอยู่คนละ Network กันก็ได้ โปรโตคอลอื่นๆ ในระดับ Network Layer ขึ้นไปทั้งTCP, UDP, ICMP ต่างก็ต้องอาศัยโปรโตคอล IP ในการรับส่งข้อมูลทั้งสิ้น

โปรโตคอล IP มีความสามารถในการค้นหาเส้นทางจากผู้รับไปยังผู้ส่ง มีกลไกที่ชาญฉลาดในการค้นหาเส้นทาง สามารถค้นหาเส้นทางได้ไปถึงผู้รับได้เอง หากมีเส้นทางที่สามารถไปได้ แต่ไม่ได้ติดต่อระหว่างผู้รับกับผู้ส่งโดยตรง และไม่มีการยืนยันว่า ข้อมูลถึงผู้รับจริงหรือไม่ ทั้งนี้อาจเกิดจากหลายสาเหตุ เช่น ที่อยู่ของผู้รับไม่มีการเชื่อมต่ออยู่ในระบบ Internet กล่าวได้ว่าโปรโตคอล IP มีหน้าที่ในการค้นหาเส้นทางเท่านั้น ไม่มีการยืนยันผลสำเร็จในการส่งข้อมูล หากเกิดข้อผิดพลาดในการส่งข้อมูล แม้ว่าจะมีการส่ง ICMP massage กลับมารายงานข้อผิดพลาด แต่ก็รับประกันไม่ได้อยู่ดีว่า ICMP message จะกลับมาถึงเรียบร้อยหรือไม่ ด้วยเหตุนี้ จึงถือว่า IP เป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (reliable)

### 2.1.2 IP Header

เมื่อข้อมูลถูกส่งลงมาจากชั้น Transport Layer สู่ชั้นNetwork Layer กระบวนการ Encapsulate ของ IP Protocol จะทำหารเพิ่มส่วน Header ลงไป Header ของ IP datagram มีขนาด 20-32 ไบต์ มีส่วนประกอบต่างๆ ดังแสดงในรูปที่ 2.1




**รูปที่ 2.1 IP Header**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1 อธิบายส่วนต่างๆของ IP Header**

| ตำแหน่ง | ชื่อ | อธิบาย |
|---|---|---|
| 0-3 | version | 4 bits แรกใช้บอกเลข version ของ Internet Protocol ใช้เพื่อตรวจสอบ Format ของ Protocol ระหว่างผู้รับ, ผู้ส่ง และ router ต่าง ๆ ใน IP Software ต่าง ๆ จะทำการตรวจสอบหมายเลข version ของใน IP headerนี้ก่อนจะดำเนินอื่น ๆ ทั้งนี้เพื่อป้ องกันความผิดพลาดในการนำข้อมูลใน packet ออกมาใช้งาน โดยหากหมายเลข version มีความแตกต่างไปจาก version ที่ software นั้นรองรับ IP Software นั้นจะทำการปฏิเสธpacket นั้น |
| 4-7 | Length | ใช้บอกความยาวของ Header มีขนาด 4 bits จะบอกขนาดเป็นจำนวนเท่าของ 32 bits |
| 8-15 | Type of Service | เป็นข้อมูลขนาด 8 บิต ปัจจุบันไม่ได้ใช้งานแล้ว |
| 16-31 | Total length | เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP Datagram ด้วยขนาด 16 บิตทำให้ Datagram มีขนาดสูงสุดไม่เกิน 65535 ไบต์ และมีขนาดเล็กสุดไม่ต่ำกว่า 512 ไบต์ |
| 32-47 | Identification | มีขนาด 16 bits เก็บตัวเลขเฉพาะ (unique) ซึ่งใช้เพื่อระบุ packet หาก router ทำการแยกส่วน data packet แต่ละส่วนที่แยกออกมาเป็น packet ใหม่นั้น router จะทำการคัดลอก header ของ packetเดิมไปใส่ให้กับ packet ใหม่ที่แยกส่วนออกมานั้น และเมื่อต้องการจะรวม packet ที่ปลายทางจะใช้ field นี้เพื่อระบุว่า packet ใดคือ packet ที่จะต้องนำมารวมกัน |
| 48-50 | Flag | มีขนาด 3 bits แบ่งเป็น 2 ส่วนคือ 1 bit แรกจะใช้เพื่อกำหนดว่า packet นั้นจะแยกส่วนได้หรือไม่โดยถ้ากำหนดให้เป็น 1 คือ do not fragment ส่วนอีก 2 bit ที่เหลือใช้เพื่อควบคุมส่วนที่ถูกแยก |
| 51-63 | fragment offset | มีขนาด 13 bits เป็น field ซึ่งใช้ระบุว่า packet ที่แยกส่วนนั้นอยู่ส่วนใดของ packet<br>ดั้งเดิม |
| 64-71 | Time to Live (TTL) | เป็นจำนวนครั้งสูงสุดที่ดาต้าแกรมนี้จะถูกส่งผ่านครือข่ายไปยังปลายทางได้ เพื่อ ป้องกันไม่ให้ดาต้าแกรมถูกเราต์ไปเรื่อยๆอย่างไม่สิ้นสุด ปกติค่านี้จะเริ่มต้นที่ 32 และจะถูกลดค่าลงทีละ 1 เมื่อมีการเราต์ จนค่านี้มีค่าเป็น 0 ก็จะไม่ถูกเราต์อีกต่อไป |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1 อธิบายส่วนต่างๆของ IP Header (ต่อ)**

| | | |
|---|---|---|
| 72-79 | Protocol | มีขนาด 8 bits ใช้เพื่อระบุ format ของ data ซึ่งก็คือ high-level protocol ซึ่งใช้ในการขนส่งข้อมูลบน IP Protocol |
| 80-95 | Header Checksum | มีขนาด 16 bits เป็นส่วนตรวจสอบความถูกต้องของข้อมลใน Header โดยไม่เกี่ยวกับส่วนข้อมูลที่อยู่ภายใน payload ค่านี้จะถูกคำนวนใหม่ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลใน Header (เช่น TTL ที่มีการเปลี่ยนแปลงทุกครั้งที่ IP datagram ถูกส่งผ่านเราเตอร์) |
| 86-127 | Source IP Address | มีขนาด 32 bits เป็นตัวเลขขนาด 32 bits ใช้เพื่อระบุผู้ส่ง |
| ไม่แน่นอน | option | มีขนาดข้อมูลไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน |
| ขึ้นอยู่กับ Option | Padding | ใช้เติม header ให้เต็มจำนวนเท่าของ 32 byte |

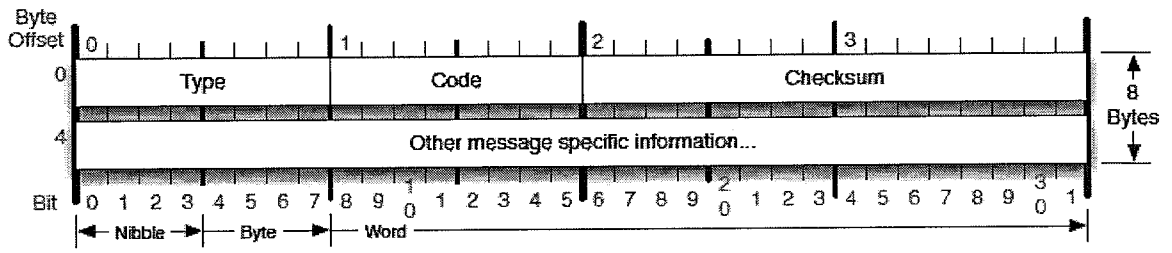
### 2.1.3 Internet Control Message Protocol (ICMP)

เนื่องจากการทำงานของเครือข่ายที่ใช้ IP ในปัจจุบันมีขนาดใหญ่ และการทำงานของอุปกรณ์กำหนดเส้นทางการส่งข้อมูล (Router) ในเครือข่ายนั้นทำงานอย่างอิสระ (มีลักษณะเป็น Autonomous)ไม่ขึ้นกับผู้ส่งข้อมูล หรือมีต้องการความร่วมมือจากผู้ส่งข้อมูลในการกำหนดเส้นทางการส่งข้อมูลแต่อย่างใด ซึ่งหากอุปกรณ์เหล่านี้ทำงานอย่างปกติก็จะปราศจากปัญหา แต่ในเครือข่ายขนาดใหญ่อย่าง internet นั้นเป็นการยากที่อุปกรณ์ในเครือข่ายจะทำงานได้อย่างปกติทุก ๆ อุปกรณ์ ซึ่งเมื่ออุปกรณ์เครือข่ายดังกล่าวทำงานผิดปกติก็จำเป็นจะต้องมีเครื่องมือเพื่อใช้ตรวจสอบ แต่โดยกลไกของ Protocol อย่าง IP นั้นไม่มีส่วนที่สนับสนุนการตรวจสอบความผิดปกติ ดังนั้นจึงต้องมีการสร้าง Protocol ใหม่ขึ้นมาเพื่อใช้ในการตรวจสอบรวมไปถึงทดสอบการสื่อสารนั่นก็คือ ICMP

ข้อความที่โปรโตคอล ICMP ส่งนั้น แบ่งออกได้เป็น 2 แบบ คือ ICMP Error Message หรือข้อความแจ้งข้อผิดพลาด และ ICMP Query หรือข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของโปรโตคอล ICMP เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่ Router จะส่งข้อความแจ้งเป็น ICMP Message ที่ชื่อ Destination Unreachable ให้กับผู้ส่งข้อมูลนั้น นอกจากนี้ตัวข้อมูลที่แจ้งข้อความ ก็จะมีส่วนของข้อมูล IP Datagram ที่เกิดปัญหาด้วย ดังนั้น เมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้ว ก็จะทราบได้ว่า จุดที่เกิดปัญหานั้นอยู่ที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4 ICMP Header




**รูปที่ 2.2 ICMP Header**

ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code Header ของ ICMP จะเริ่มต้นหลังจาก bit ที่160 ของ IP header ซึ่งจะมีรายละเอียดดังนี้

- Type - ICMP type คือประเภทของ message
- Code - จะใช้คู่กับ Type เพื่อบ่งบอกประเภท message เช่น an ICMP Destination Unreachable might have this field set to 1 through 15 each bearing different meaning.
- Checksum - เก็บ error checking ซึ่งจะคำนวณมาจาก ICMP header+data
- ID - เก็บ ID value จะ return มาในกรณี ECHO REPLY
- Sequence - เก็บ sequence value จะ return มาในกรณี ECHO REPLY เช่นกัน

โดยค่า TYPE ที่เป็นไปได้มีดังตารางที่ 2.2 สำหรับค่าของ CODE นั้นการแปลความความจะสัมพันธ์กับค่า TYPE ดังแสดงไว้ในตารางที่ 2.3

**ตารางที่ 2.2 ค่า TYPE ของ ICMP**

| TYPE Field | ICMP Message Type | TYPE Field | ICMP Message Type |
|---|---|---|---|
| 0 | Echo Reply | 12 | Parameter Problem on a Datagram |
| 3 | Destination Unreachable | 13 | Timestamp Request |
| 4 | Source Quench | 14 | Timestamp Reply |
| 5 | Redirect (Change a route) | 15 | Information Request (Obsolete) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.2 ค่า TYPE ของ ICMP (ต่อ)**

| TYPE Field | ICMP Message Type | TYPE Field | ICMP Message Type |
|---|---|---|---|
| 8 | Echo Request | 16 | Information Reply (Obsolete) |
| 9 | Router Advertisement | 17 | Address Mask Request |
| 10 | Router Solicitation | 18 | Address Mask Reply |
| 11 | Time Exceeded for Datagram | | |

**ตารางที่ 2.3 ค่า TYPE ของ ICMP**

| TYPE | CODE | Meaning |
|---|---|---|
| **3** | **0** | Network Unreachable |
| | **1** | Host Unreachable |
| | **2** | Protocol Unreachable |
| | **3** | Port Unreachable |
| | **4** | Fragmentation needed and DF s |
| | **5** | Source route failed |
| | **6** | Destination Network Unknown |
| | **7** | Destination Host Unknown |
| | **8** | Source Host Isolated |
| | **9** | Communication with Destination Network Administratively Prohibited |
| | **10** | Communication with Destination Host Administratively Prohibited |
| | **11** | Network Unreachable for Type Of Service |
| | **12** | Host Unreachable for Type Of Service |
| **5** | **0** | Redirect Datagram for the Net (Obsolete |
| | **1** | Redirect Datagram for the Host |
| | **2** | Redirect Datagram for the Type of Service and Net |
| | **3** | Redirect Datagram for the Type of Service and Host |
| **11** | **0** | Time-to-live Count Exceeded |
| | **1** | Fragment reassembly time exceeded |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 TCP Protocol

### 2.2.1 TCP Protocol

TCP เป็น Protocol ที่เป็นเอกเทศ (Independent) และเอนกประสงค์ซึ่งสามารถนำไปประยุกต์ใช้งานได้กับระบบทั่ว ๆ ไป TCP เป็น Protocol ที่เป็นที่นิยมอย่างมากเนื่องจากมีข้อจำกัดไม่มากนักในการใช้งานร่วมกับProtocol ในระดับชั้นล่าง ๆ ลงไป ยกตัวอย่างเช่นใช้กับ Ethernet หรือ ใช้กับเน็ตเวิร์คซึ่งมีความซับซ้อนมากกว่าอย่าง ATM นอกจากนั้น Protocol ซึ่งเป็นมาตรฐานเปิดชื่อ TP-4 ของ International Organization for Standardization ก็พัฒนาโดยใช้ TCP เป็นพื้นฐาน

ในการสื่อสารข้อมูลที่ระดับชั้นล่างสุด การรับส่งข้อมูลต่าง ๆ อาจเกิดการสูญหาย (Data lost) เกิดการซ้ำกันของข้อมูล (Data duplication) หรือข้อมูลผิดลำดับ (Out of order) ซึ่งอาจเกิดขึ้นจากหลายปัจจัยเช่น ความผิดพลาดอันเนื่องมาจากอุปกรณ์ฮาร์ดแวร์ หรือโครงข่ายสื่อสารมีการใช้งานหนาแน่น (heavily loaded) เป็นต้น

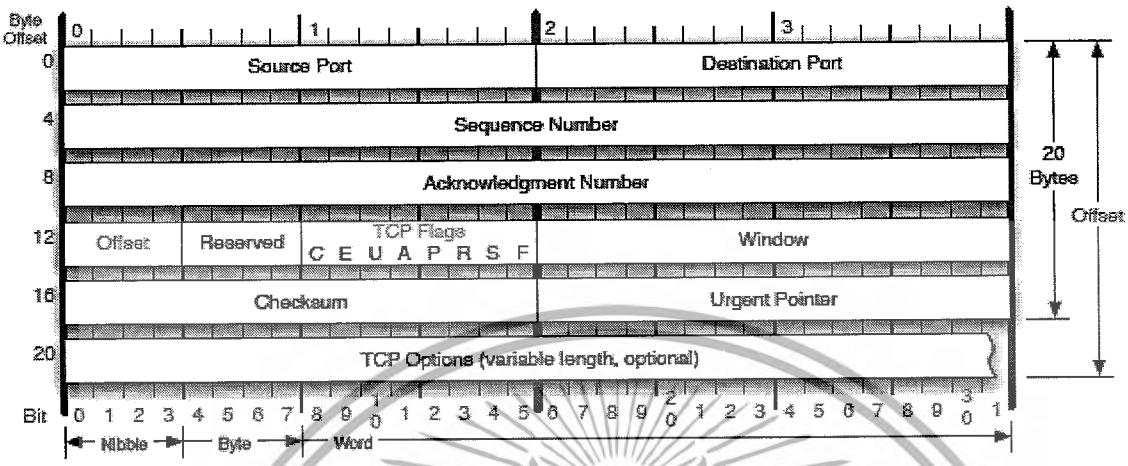
โดยปกติการสื่อสารข้อมูลในระดับชั้นล่างสุด (Lowest level) จะมีเป้ าหมายอยู่ที่การเพิ่มประสิทธิภาพในการส่งข้อมูลผ่านโครงข่ายให้สูงที่สุด แต่การทำงานของการสื่อสารในระดับชั้นที่สูงขึ้นไป (ในระดับชั้นของโปรแกรมประยุกต์) จะมีเป้ าหมายอยู่ที่การส่งข้อมูลขนาดใหญ่ (Large volume) ซึ่งจะพบว่าข้อมูลขนาดใหญ่นั้นเมื่อผ่านมาถึงการสื่อสารระดับชั้นล่างสุดจะถูกแบ่งออกเป็น packet ขนาดเล็กลงเพื่อประสิทธิภาพและตามที่ได้กล่าวมาข้างต้นการสื่อสารข้อมูลในระดับล่างสุดนั้น unreliable ซึ่งหากปราศจาก Protocol ที่เหมาะสมแล้วจะทำให้เกิดปัญหาในการสื่อสารข้อมูลอย่างมาก (ปัญหา duplication, data lost, delay) ซึ่งแต่ละโปรแกรมประยุกต์ที่พัฒนาขึ้นจะต้องมีส่วนของโปรแกรมซึ่งสามารถตรวจสอบความผิดพลาดต่าง ๆ (Error detection)

เพื่อแก้ปัญหาดังกล่าวจึงมีงานวิจัยเพื่อสร้าง Protocol เอนกประสงค์ ซึ่งสามารถเตรียมการสื่อสารในลักษณะของ stream ของข้อมูลที่เชื่อถือได้ ซึ่งวิธีการนี้จะทำให้สามารถแยกส่วนโปรแกรมประยุกต์กับรายละเอียดทางด้านการสื่อสารข้อมูลผ่านโครงข่ายออกจากกัน ทำให้ง่ายที่จะกำหนดรูปแบบการเชื่อมต่อ (interface) ในรูปแบบเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

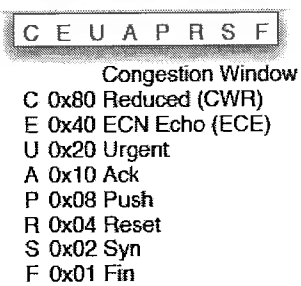
### 2.2.2 TCP Header

ส่วนหัว (Header) ของ packet ข้อมูลแบบ TCP จะมีลักษณะดังนี้




**รูปที่2.3 TCP Header**

- **Source Port Number** : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- **Destination Port Number** : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- **Sequence Number** : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
- **Acknowledgment Number** : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
- **Header Length** : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์
- **Flag** : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

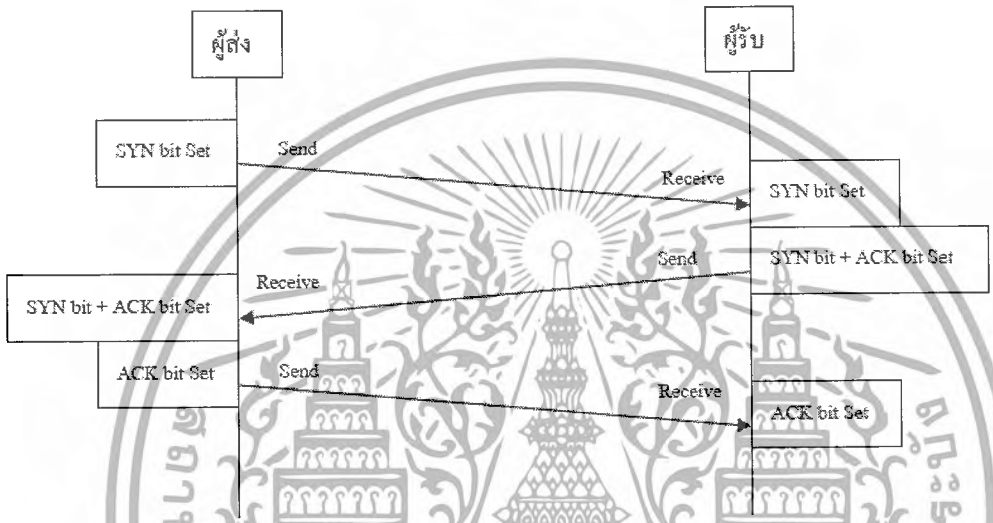



**รูปที่ 2.4 Flag**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

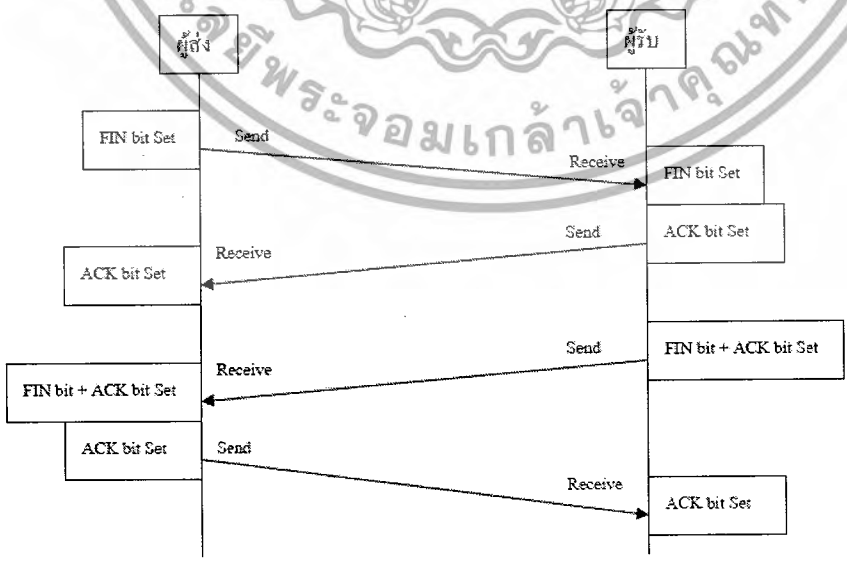
- **Checksum** เก็บค่าสำหรับการตรวจสอบความผิดพลาด(Error-checking) ของ header+ data
- **Urgent Point** ใช้ระบุตำแหน่งของข้อมูลเร่งด่วน (urgent data)
- **TCP Options** ใช้ระบุ options พิเศษ ซึ่งอาจจะมีหรือไม่มีก็ได้ ขึ้นอยู่กับข้อมูลหรือเหตุการณ์ระหว่างการสื่อสาร

ในการเริ่มต้นการเชื่อมต่อแบบ TCP จะใช้กลไก “Three-way handshake” ซึ่งมีลำดับการทำงานดังแสดงในรูปที่ 2.5




**รูปที่ 2.5 Three-way handshaking**

สำหรับการจบการเชื่อมต่อจะมีลำดับการทำงานดังแสดงในรูปที่ 2.6




**รูปที่ 2.6 Modified three-way handshake for closing a TCP connection**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 UDP Protocol

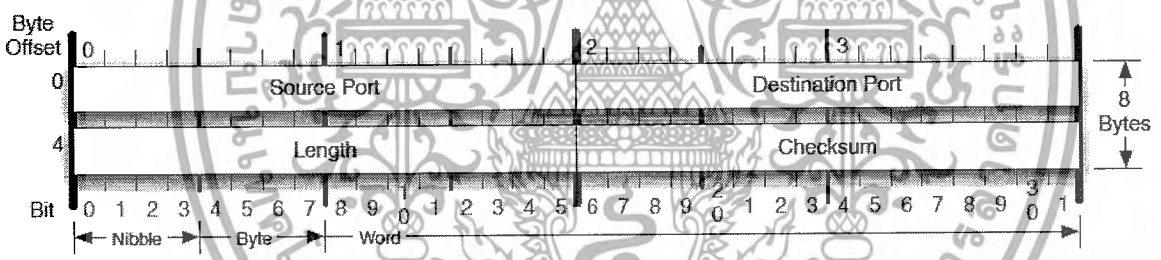
### 2.3.1 UDP Protocol

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

### 2.3.2 UDP Header

ส่วนหัว (Header) ของ packet ข้อมูลแบบ UDP จะมีลักษณะดังนี้




**รูปที่ 2.7 UDP Headers**

- **UDP Source Port**: field ข้อมูลนี้มีขนาด 16 bits และมีลักษณะเป็น Optional คือจะใส่ข้อมูล port หรือไม่ใส่ก็ได้
- **UDP Destination Port**: เป็น field ที่มีขนาด 16 bits ใช้ระบุหมายเลข port ปลายทาง
- **UDP Message Length**: เป็น field ที่มีขนาด 16 bits ใช้ระบุความยาวของข้อมูลที่จะส่ง
- **UDP Checksum**: มีขนาด 16 bits ใช้สำหรับใส่ค่า checksum โดยการคำนวณ checksum นั้นจะนำทั้งส่วน header และ ส่วน data มาทำการคำนวณ หากไม่ใช้จะใส่ค่าใน field นี้เป็น 0 การทำให้สามารถเลือกที่จะไม่ใช้ checksum ก็เพื่อที่จะลดการคำนวณลงในกรณีที่การสื่อสารนั้นกระทำบนเครือข่ายที่มีความน่าเชื่อถือสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 Network Programming

### 2.4.1 Raw Sockets

ในการส่งข้อมูลผ่านระบบเครือข่ายจะพบว่าโดยส่วนใหญ่จะอาศัยการทำงานของโปรโตคอล TCP หรือ UDP บน IP แต่ในงานบางงานอาจต้องการการทำงานกับโปรโตคอลชนิดอื่นซึ่งไม่ใช่ทั้ง TCP และ UDP ดังนั้น ในส่วนพื้นฐานของการเชื่อมต่อเช่น Socket จึงต้องพร้อมที่จะรองรับงานประเภทดังกล่าวด้วย ดังนั้นจึงต้องมี Socket อีกประเภทนอกเหนือจาก Stream และ Datagram เพื่อรองรับการสื่อสารข้อมูลโดยโปรโตคอลอื่น ๆ นอกเหนือจาก TCP และ UDP นี้ ซึ่ง Socket ประเภทดังกล่าวก็คือ Raw Socket นั่นเอง

โดยปกติในการทำงานกับ Stream หรือ Datagram Socket นั้น เมื่อ Socket ได้รับข้อมูลที่ส่งมาถึงก็จะทำการแบ่งแยกข้อมูลที่ได้รับออกเป็น 2 ส่วนได้แก่ Header และ Data โดยการแบ่งข้อมูลดังกล่าวจะเป็นอัตโนมัติ ซึ่งทำให้ Application program ไม่ต้องรองรับการแบ่งส่วนของข้อมูล แต่สำหรับ Raw Socket นี้ข้อมูลซึ่งได้รับมาจะถูกส่งให้กับ Application program โดยที่ไม่มีการแบ่งแยก Header และ Data ใด ๆ ดังนั้นในการใช้งาน Socketประเภทนี้ Application Program จะต้องมีฟังก์ชั่นซึ่งรองรับการแบ่งแยก Header และ Data รวมทั้งฟังก์ชั่นพิเศษอื่น ๆ ตามที่โปรโตคอลนั้น ๆ ต้องการเพื่อที่จะทำงานได้เช่น ฟังก์ชั่นสำหรับคำนวณ Check Sum หรือ ฟังก์ชั่นสำหรับทำ Error Correction เป็นต้น

สำหรับการส่งข้อมูลผ่าน Raw Socket นั้นจะมีขั้นตอนที่ยุ่งยากน้อยกว่าการรับข้อมูลเล็กน้อยตรงที่ Application Program ไม่ต้องรับผิดชอบการสร้าง IP Header หากแต่ Application Program รับผิดชอบเพียงการสร้าง Header ของโปรโตคอลในระดับถัด ๆ ไป

Raw Socket สามารถเรียกใช้คำสั่งดังนี้

```
Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Raw,
ProtocolType.xxxx);
```

การส่งข้อมูลผ่าน Raw Socket:

```
sock.SendTo(Bytes(), packetsize, SocketFlags.None, iep);
```

การรับข้อมูล:

```
recv = sock.ReceiveFrom(data, ref ep);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะพบว่าในการสร้าง Raw Socket นั้นใช้ลักษณะวิธีการเช่นเดียวกันกับการสร้าง Stream และDatagram Socket เพียงเปลี่ยน parameter ในส่วนของ Socket Type เป็น SocketType.Raw สำหรับ parameter ตัวสุดท้ายนั้นจะเป็นการระบุประเภทของโปรโตคอลที่จะใช้งานกับ Socket นี้ ซึ่งจะมีผลในการใส่ประเภทของโปรโตคอลในชั้นถัดไปใน Header ของ IP ในตอนที่จะทำการส่งเท่านั้น

ในส่วนการรับและส่งข้อมูลนั้น สังเกตได้ว่าใช้คำสั่งรวมทั้ง Parameters แบบเดียวกันกับการรับ/ส่งข้อมูลกับ Datagram Socket ทุกประการ

**2.4.2 ICMP Protocol**

Internet Control Message Protocol หรือ ICMP นี้เป็นตัวอย่างหนึ่งของโปรโตคอลซึ่งทำงานบน IP แต่ไม่ได้ทำงานบน TCP หรือ UDP ดังนั้นจึงไม่สามารถใช้ Stream หรือ Datagram Socket ได้โปรแกรมตัวอย่างต่อไปนี้จะพบว่ามีทั้งส่วนที่ใช้เพื่ออ่าน/เขียน Header ของ ICMP โดยโปรแกรมที่ยกมาเป็นตัวอย่างนี้คือโปรแกรม PING นั่นเอง

ตัวอย่าง Class สำหรับใช้ เพื่ออ่าน Header ต่าง ๆ ของ ICMP Packet

```
using System;
using System.Net;
using System.Text;
class ICMP
{
        public byte Type;
        public byte Code;
        public UInt16 Checksum;
        public int MessageSize;
        public byte[] Message = new byte[1024];
        public ICMP()
        {
        }
        public ICMP(byte[] data, int size)
        {
                Type = data[20];
                Code = data[21];
                Checksum = BitConverter.ToUInt16(data, 22);
                MessageSize = size - 24;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Buffer.BlockCopy(data, 24, Message, 0, MessageSize);
    }
    public byte[] getBytes()
    {
            byte[] data = new byte[MessageSize + 9];
            Buffer.BlockCopy(BitConverter.GetBytes(Type), 0, data, 0, 1);
            Buffer.BlockCopy(BitConverter.GetBytes(Code), 0, data, 1, 1);
            Buffer.BlockCopy(BitConverter.GetBytes(Checksum), 0, data, 2, 2);
            Buffer.BlockCopy(Message, 0, data, 4, MessageSize);
            return data;
    }
    public UInt16 getChecksum()
    {
            UInt32 chcksm = 0;
            byte[] data = getBytes();
            int packetsize = MessageSize + 8;
            int index = 0;
            while ( index < packetsize)
            {
                chcksm += Convert.ToUInt32(BitConverter.ToUInt16(data, index));
                index += 2;
            }
            chcksm = (chcksm >> 16) + (chcksm & 0xffff);
            chcksm += (chcksm >> 16);
            return (UInt16)(~chcksm); } }
```

ตัวอย่างโปรแกรมซึ่งใช้งาน Raw Socket – Simple Ping

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class SimplePing
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    public static void Main(string[] argv)
    {
        byte[] data = new byte[1024];
        int recv;
        Socket host = new Socket(AddressFamily.InterNetwork, SocketType.Raw,
        ProtocolType.Icmp);
        IPEndPoint iep = new IPEndPoint(IPAddress.Parse(argv[0]), 0);
        EndPoint ep = (EndPoint)iep;
        ICMP packet = new ICMP();
        packet.Type = 0x08;
        packet.Code = 0x00;
        packet.Checksum = 0;
        Buffer.BlockCopy(BitConverter.GetBytes((short)1), 0, packet.Message, 0, 2);
        Buffer.BlockCopy(BitConverter.GetBytes((short)1), 0, packet.Message, 2, 2);
        data = Encoding.ASCII.GetBytes("test packet");
        Buffer.BlockCopy(data, 0, packet.Message, 4, data.Length);
        packet.MessageSize = data.Length + 4;
        int packetsize = packet.MessageSize + 4;
        UInt16 chcksum = packet.getChecksum();
        packet.Checksum = chcksum;
        host.SetSocketOption(SocketOptionLevel.Socket,
        SocketOptionName.ReceiveTimeout, 3000);
        host.SendTo(packet.getBytes(), packetsize, SocketFlags.None, iep);
        try
        {
            data = new byte[1024];
            recv = host.ReceiveFrom(data, ref ep);
        } catch (SocketException)
        {
            Console.WriteLine("No response from remote host");
            return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        ICMP response = new ICMP(data, recv);
        Console.WriteLine("response from: {0}", ep.ToString());
        Console.WriteLine(" Type {0}", response.Type);
        Console.WriteLine(" Code: {0}", response.Code);
        int Identifier = BitConverter.ToInt16(response.Message, 0);
        int Sequence = BitConverter.ToInt16(response.Message, 2);
        Console.WriteLine(" Identifier: {0}", Identifier);
        Console.WriteLine(" Sequence: {0}", Sequence);
        string stringData = Encoding.ASCII.GetString(response.Message, 4,
        response.MessageSize - 4);
        Console.WriteLine(" data: {0}", stringData);
        host.Close();
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# บทที่ 3

# Client Server Model and P2P Model

## 3.1 Client-Server Model

ในการสื่อสารข้อมูลจะพบว่า Socket นั้นเป็นช่องทางที่โปรแกรมใช้ในการติดต่อกับโครงข่ายสื่อสาร ซึ่งจะช่วยในเรื่องของการจัดการข้อมูลเช่น การเตรียม buffer สำหรับข้อมูลเข้าและออก การรับ และส่งข้อมูล เป็นต้น ส่วน Protocolที่ใช้ ยกตัวอย่างเช่น TCP/IP มีหน้าที่เพียงกำหนดวิธีการส่งผ่านข้อมูลในการสื่อสารเท่านั้น ส่วนบทบาทของผู้รับและผู้ส่งหรือ จังหวะในการรับส่งข้อมูล นั้นมิได้ถูกกำหนดไว้ทั้งใน Specification ของทั้ง Socket และ Protocol

เพื่อให้การสื่อสารข้อมูลทางคอมพิวเตอร์เป็นไปโดยราบรื่นและมีประสิทธิภาพ ได้มีการจัดการจังหวะของการสื่อสารด้วยแนวคิดที่เรียกว่า "Client-Server Paradigm"

ประเด็นหลักที่ผลักดันให้เกิด "Client-Server Paradigm" คือปัญหาที่เรียกว่า Rendezvous ปัญหา Rendezvousเกิดขึ้นจากจังหวะการทำงานที่ไม่สอดคล้องกันของโปรแกรมที่สื่อสารกัน ยกตัวอย่างเช่น มีโปรแกรม 2 โปรแกรมต้องการสื่อสารข้อมูลผ่านโครงข่ายสื่อสาร หาก 2 โปรแกรมนั้นถูกผู้ใช้งานเรียกใช้งาน (run) พร้อม ๆ กัน โปรแกรมทั้ง 2 นั้นก็จะสามารถสื่อสารกันได้ หากถ้าโปรแกรมทั้งสองนั้นถูกเรียกใช้งานไม่พร้อมกัน โปรแกรมที่ทำงานก่อนเมื่อเริ่มการสื่อสารก็จะพบว่าโปรแกรมที่จะสื่อสารด้วยนั้นไม่มีก็จะจบการทำงานไป ในขณะที่โปรแกรมที่เริ่มทำงานทีหลังพอเริ่มต้นการสื่อสารก็จะพบว่าโปรแกรมที่จะสื่อสารด้วยนั้นจบการทำงานไปแล้ว เป็นต้น Client-Server Model ได้ทำการแก้ไขปัญหานี้โดยการกำหนดบทบาทและจังหวะในการสื่อสาร โดยกำหนดให้ในการสื่อสารระหว่างกันของคู่โปรแกรมนั้นจะต้องมีฝ่ายหนึ่งเมื่อเริ่มต้นทำงานแล้วจะต้องรอจนกว่าอีกฝ่ายหนึ่งจะติดต่อเข้ามาก่อนจึงจะเริ่มต้นกระบวนการสื่อสาร (รับ-ส่งข้อมูล)ได้

Client-Server Paradigm จะแบ่งประเภทของโปรแกรมที่สื่อสารกันออกเป็น 2 ประเภท โดยอาศัยการเริ่มต้นการสื่อสาร (Direction of initiation) เป็นเกณฑ์ในการแบ่งประเภท โดยทั่วไปโปรแกรมซึ่งเป็นผู้เริ่มต้นกระบวนการสื่อสารข้อมูลจะถูกเรียกว่า "Client" ในขณะที่โปรแกรมซึ่งรอให้โปรแกรมอื่นร้องขอการเชื่อมต่อจะเรียกว่า "Server" กระบวนการในการทำงานโดยทั่วไป Client ซึ่งส่วนใหญ่จะอยู่ในรูปของโปรแกรมประยุกต์เช่น web browserเป็นต้น เมื่อโปรแกรมประยุกต์ดังกล่าวถูกเรียกขึ้นมาใช้งาน ก็จะทำการติดต่อไปยัง Server, ส่งการร้องขอบริการ (Service-request) ไปยัง Server จากนั้นก็จะรอการตอบสนองจาก Server ทางฝั่ง Server เมื่อได้รับการร้องขอบริการก็จะทำการประมวลผลตามที่ Client ร้องขอ เมื่อได้ผลลัพธ์แล้วก็จะส่งผลลัพธ์นั้นไปยัง Clientโปรแกรม Client โดยทั่วไปอาจมองได้เป็น 2 ลักษณะคือ Standard Client Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

103085

และ Nonstandard Client Software ความแตกต่างระหว่างโปรแกรม Client ทั้ง 2 ลักษณะคือ Standard Client Software นั้นจะเป็นโปรแกรมที่ทำการติดต่อสื่อสารกับ Service ที่เป็นมาตรฐาน และมี port ในการเชื่อมต่อซึ่งเป็นที่รู้กันโดยทั่วไป ตัวอย่างของ Service ที่เป็นมาตรฐานได้แก่ WWW, FTP, SMTP เป็นต้น โดยปกติหมายเลข port ของ Service มาตรฐานจะมีค่าน้อยกว่า 1024 สำหรับNonstandard Client Software นั้นจะเป็นโปรแกรมที่ทำการติดต่อกับ Service ซึ่งให้บริการเป็นการเฉพาะ ใช้หมายเลข port ซึ่งกำหนดขึ้นมาเอง โดยมากหมายเลข port จะมีค่ามากกว่า 1024

**ตารางที่ 3.1 ตัวอย่างของหมายเลข Port มาตรฐาน**

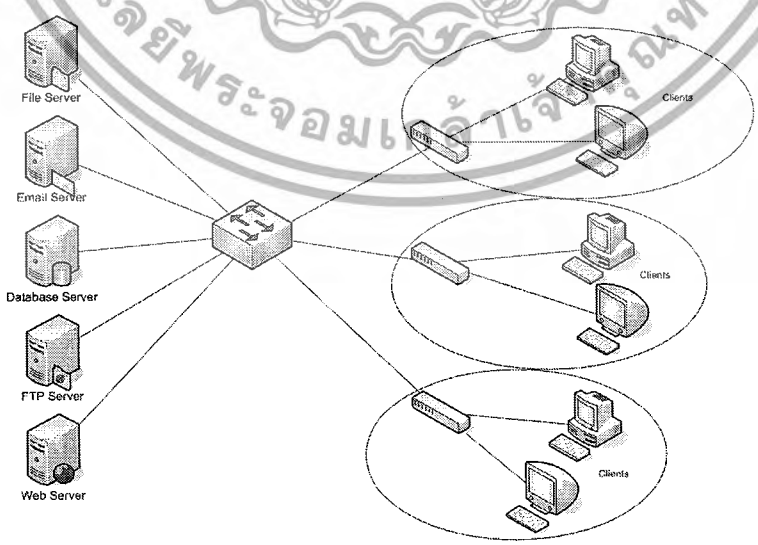
| Port | Service/Description | Port | Service/Description |
|---|---|---|---|
| 20,21 | FTP | 514 | Shell |
| 22 | Secure Shell | 515 | Print Service |
| 23 | Telnet | 517 | Talk |
| 43 | WHOIS | 518 | NTalk |
| 52,54,56 | XNS | 520 | Routing (RIP) |
| 53 | Domain Name System | 530 | RPC |
| 70 | Gopher | 546,547 | DHCPv6 |
| 79 | Finger | 554 | Real Time Streaming Protocol |
| 80 | HTTP (WEB) | 631 | Internet Printing Protocol |
| 88 | Kerberos | 636 | Secured LDAP |
| 110 | POP3 | 666 | DOOM (1st online, Game) |
| 111 | Sun RPC | 691 | MS Exchange |
| 118, 156 | SQL | 711 | Cisco TDP |
| 123 | Network Time Protocol | 860 | iSCSI |
| 137,138,139 | NetBIOS | 873 | rsync |
| 143 | IMAP | 989,990 | Secured FTP |
| 389 | LDAP | 992 | Secured Telnet |
| 443 | Secured HTTP | 993 | Secured IMAP |
| 445 | MS-SMB file sharing | 995 | Secured POP3 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นโปรแกรม Server ยังอาจมีรูปแบบการสร้างแบ่งเป็น 2 ประเภท โดยแบ่งตามลักษณะการจัดการข้อมูลสถานะของการ interaction ระหว่าง Client และ Server ในระหว่างการสื่อสาร Server ซึ่งไม่มีการจัดเก็บข้อมูลสถานะระหว่างการสื่อสารนั้นจะเรียกว่า Stateless Servers ส่วน Server ที่มีการจัดเก็บข้อมูลสถานะระหว่างการสื่อสารจะเรียกว่า Stateful Servers

เพื่อให้เกิดความเข้าใจในเรื่องของ Stateless และ Stateful Servers จะยกตัวอย่างโปรแกรมซึ่งทำหน้าที่เป็น Fileserver หาก File Server นี้สร้างแบบ Stateless จะได้ว่า Server นี้จะไม่มีการเก็บข้อมูลสถานะระหว่างการเชื่อมต่อกับClients ดังนั้นในการร้องขอบริการเกี่ยวกับไฟล์ของ Clients ทุกครั้งจะต้องมีการระบุรายละเอียดสถานะด้วย เช่นในการร้องขอ Copy file จาก Server นั้น โปรแกรม Client จะต้องระบุชื่อไฟล์โดยสมบูรณ์ (Complete File Name) ระบุตำแหน่งในไฟล์ซึ่งจะให้ Server อ่านข้อมูลเพื่อส่งไปให้ Client ทำการ Copy และจะต้องทำเช่นนี้ทุกครั้งที่มีการร้องขอให้ Server ส่งข้อมูลใน Block ถัดไปเพื่อทำการ copy จนกว่าจะจบไฟล์ ในขณะที่หาก File Server นั้นเป็น Stateful Server ตัว Server ก็จะมีการจัดเก็บสถานะการทำงานระหว่างการสื่อสารข้อมูลกับ Client ดังนั้นในการร้องขอ Copy file จาก Server ก็ไม่จำเป็นที่จะต้องระบุชื่อไฟล์โดยสมบูรณ์และตำแหน่งในไฟล์ในทุก ๆ การร้องขอ ทั้งนี้เพราะ Server ได้ทำการจัดเก็บข้อมูลเหล่านั้นไว้แล้วเมื่อแรกเริ่มที่ Client ร้องขอการ Copy ไฟล์

จากตัวอย่างจะพบว่า Stateful Servers นั้นจะให้ประสิทธิภาพที่ดีกว่า Stateless Servers แต่ในทางปฏิบัติ Stateful Server จะมีความยากและซับซ้อนในการสร้าง นอกจากนั้นปัญหาที่เกิดจากการสื่อสารเช่น การล่าช้าของข้อมูล (Data Delaying), การซ้ำกันของข้อมูล (Data Duplication) และลำดับของข้อมูลผิดพลาด (Out of Order) นั้นก็อาจเป็นสาเหตุในสถานะ (State) ซึ่ง Server เก็บไว้นั้นผิดพลาดได้



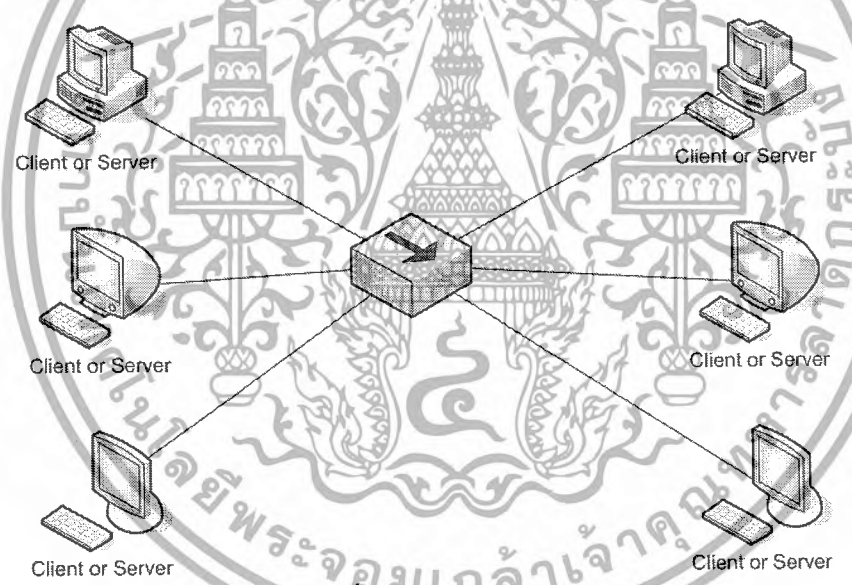

**รูปที่ 3.1 Client-Server Model**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 P2P Model

P2P หรือเต็มๆ Peer to Peer อาจจะมีคำอื่นอีกเช่น People to People, Point to Point ซึ่งมีความหมายคล้ายคลึงกับ Peer to Peer คือ

- ระบบที่อนุญาตให้ผู้ใช้สามารถแลกเปลี่ยนไฟล์หรือใช้ทรัพยากรร่วมกันผ่านระบบเครือข่าย โดยแต่ละเครื่องเป็นผู้กำหนดว่าข้อมูลหรือทรัพยากรใดจะแชร์กับคนอื่น
- เครือข่ายประเภทไม่มีเครื่องเซิร์ฟเวอร์ โดยเป็นระบบการสื่อสารจากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่งโดยตรง
- PC ทุกเครื่องมีสิทธิเท่าเทียมกัน หรือ เพียร์ (Peer) แต่เดิมนั้นเมื่อเราต้องการดาวน์โหลดไฟล์จากแม่ข่ายโดยใช้ FTP, HTTP นั้น หากมีคนที่ต้องการไฟล์เดียวกับเรา 500 คนมาดาวน์โหลดบนแม่ข่ายเดียวกันพร้อมๆ กันโดยแต่ละคนมี Bandwidth คนละ 256kbps ถ้าจะให้ทุกคนนั้นได้ความเร็วในการดาวน์โหลดสูงสุดตัวแม่ข่ายจะต้องมี Bandwidth เท่ากับ 256kbps * 500 (125mbit) เลยทีเดียว




**รูปที่ 3.2 P2P Model**

ซึ่งไม่ใช่เพียงแต่จะเปลือง Bandwidth เท่านั้น แต่ยังจะต้องใช้แม่ข่ายที่มีความเร็วในการประมวลผลสูงเพื่อรองรับความต้องการที่เพิ่มขึ้นอีกด้วย จากปัญหาดังกล่าวนี้เองทำให้โปรแกรมแชร์ไฟล์ P2P ถูกพัฒนาขึ้นมารองรับความต้องการที่เพิ่มขึ้นโดยใช้ทรัพยากรของแม่ข่ายให้น้อยลงนั้นเอง ระบบ P2P นั้นถูกพัฒนาเพื่อใช้ในการแลกเปลี่ยนไฟล์ระหว่างกันโดยที่ไม่พึ่งแม่ข่ายในการแจกจ่ายไฟล์และทำให้สามารถหาไฟล์ที่ต้องการได้ง่ายขึ้น จากความหมายนี้เองทำให้เราเรียกโปรแกรมที่ใช้ในการแลกเปลี่ยนไฟล์ระหว่างกันโดยไม่ได้ร้องขอไฟล์จากแม่ข่ายว่า "P2P File Sharing" โปรแกรมประเภทนี้มีอยู่มากมายหลายตัวด้วยกันดังตัวอย่างในตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2 P2P Protocol**

| Protocol | Use | Applications |
|---|---|---|
| Ares | File sharing | Ares Galaxy, Warez P2P |
| BitTorrent | File sharing/Software distribution/Media distribution | ABC, AllPeers, Azureus, BitComet, BitLord, BitTornado, BitTorrent, Burst!, Deluge, FlashGet, G3 Torrent, Halite, KTorrent, LimeWire, MLDonkey, Opera, QTorrent, rTorrent, TorrentFlux, Transmission, Tribler, µTorrent, Thunder, Shareaza |
| CSpace | File sharing, text chat, remote desktop | a peer-to-peer based communications system |
| Direct Connect | File sharing | DC++, NeoModus Direct Connect, SababaDC, BCDC++, ApexDC++, StrongDC++ |
| Direct Connect | File sharing | DC++, NeoModus Direct Connect, SababaDC, BCDC++, ApexDC++, StrongDC++ |
| Domain Name System | Internet information retrieval | See Comparison of DNS server software |
| eDonkey | File sharing | aMule, eDonkey2000 (discontinued), eMule, eMule Plus, FlashGet, iMesh, Jubster, lMule, MLDonkey, Morpheus, Pruna, xMule, Shareaza |
| FastTrack | File sharing | giFT, Grokster, iMesh (and its variants stripped of adware including iMesh Light), Kazaa (and its variants stripped of adware such as Kazaa Lite), KCeasy, Mammoth, MLDonkey, Poisoned |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2 P2P Protocol (ต่อ)**

| Protocol | Use | Applications |
|---|---|---|
| Freenet | Distributed data store | Entropy (on its own network), Freenet |
| G3 | File sharing | FilesWire (Web based P2P), DeskWire |
| GNUnet | File sharing, chat | GNUnet, (GNUnet-gtk) |
| Gnutella | File sharing | Acquisition, BearShare, Cabos, FrostWire, Gnucleus, Grokster, gtk-gnutella, iMesh, Kiwi Alpha, LimeWire, MLDonkey, Morpheus, Poisoned, Swapper, XoloX, Shareaza |
| Gnutella2 | File sharing | Adagio, Gnucleus, iMesh, Kiwi Alpha, MLDonkey, Morpheus, TrustyFiles, Shareaza |
| Kad Network | File sharing | aMule, eMule, MLDonkey |
| JXTA | Peer applications | Collanos Workplace (Teamwork software), Sixearch |
| Krawler | Social network | Krawler[x] |
| MANOLITO/MP2P | File sharing | Blubster, Piolet |
| MFPnet | File sharing | amiciPhone (no longer available) |
| Napster | File sharing | Napigator, Napster |
| NeoEdge | File sharing | MostFun Game Player, NeoARM game delivery |
| OpenNap | File sharing | WinMX, Utatane, XNap, Napster |
| P2PTV | Video stream or file sharing | TVUPlayer, Joost, CoolStreaming, Cybersky-TV, TVants, PPLive, Kontiki, LiveStation |
| PDTP | Streaming media or file sharing | PDTP |
| Peercasting | Multicasting streams | PeerCast, IceShare, FreeCast, Rawflow |
| Pichat | Chat, Collaboration | a peer-to-peer chat platform |
| Usenet | Distributed discussion | See list of news clients |
| WPNP | File sharing | WinMX |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.2 P2P Protocol (ต่อ)**

| Protocol | Use | Applications |
|---|---|---|
| Windows Peer-to-Peer | Distributed peer application development, collaboration [9] | Shipped with Advanced Networking Pack for Windows XP [10], Windows XP SP2, Windows Vista. This is a Windows component that runs only over IPv6 and provides a 'meta' peer-to-peer network that applications can utilize. It does not have file sharing support but third-parties can develop one |
| Direct Connect | File sharing | DC++, NeoModus Direct Connect, SababaDC, BCDC++, ApexDC++, StrongDC++ |
| Domain Name System | Internet information retrieval | See Comparison of DNS server software |
| eDonkey | File sharing | aMule, eDonkey2000 (discontinued), eMule, eMule Plus, FlashGet, iMesh, Jubster, lMule, MLDonkey, Morpheus, Pruna, xMule, Shareaza |
| FastTrack | File sharing | giFT, Grokster, iMesh (and its variants stripped of adware including iMesh Light), Kazaa (and its variants stripped of adware such as Kazaa Lite), KCeasy, Mammoth, MLDonkey, Poisoned |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.1 Bittorrent Protocol**

เป็น P2P ที่เป็นที่นิยมในปัจจุบันมีหลักการทำงานคร่าวๆก็คือ

1.เชื่อมต่อไปยังแม่ข่าย เพื่อยืนยันตัวตนและส่ง สารบัญไฟล์ที่เราแชร์ไว้ไปด้วย

2.หากเราต้องการหาไฟล์สักไฟล์หนึ่งเมื่อเราระบุ Keyword โปรแกรมจะส่งคำร้องไปยังแม่ข่าย จากนั้นแม่ข่ายจะส่งรายชื่อไฟล์พร้อมข้อมูลตัวตนของคนที่มีไฟล์ที่ตรงกับ Keyword ที่เราระบุกลับมา

3.เมื่อเราพบไฟล์ที่ต้องการแล้วตัวโปรแกรมจะใช้ข้อมูลที่ได้จากแม่ข่ายติดต่อไปยังคนนั้นๆโดยตรงเพื่อร้องขอไฟล์

จากการทำงาน จะเห็นได้ว่าตัวแม่ข่ายนั้นไม่ได้เป็นคนเก็บไฟล์จริงๆ มีไว้เพียงแค่เก็บสารบัญเท่านั้น และถึงแม้ว่า P2P จะแก้ปัญหาจุดอ่อนเรื่อง Bandwidth และทรัพยากรเครื่องแม่ข่ายที่สูงได้ แต่ถึงกระนั้นความเร็วที่ได้ก็ไม่ค่อยจะดีซักเท่าไหร่เนื่องจากความเร็วที่ได้นั้น จะได้จากจากเชื่อมต่อผู้ใช้ด้วยกันเอง ซึ่งโดยทั่วไปแล้วความเร็ว Internet ในการรับส่ง ของผู้ใช้ตามบ้านนั้นไม่ได้เร็วเหมือนเครื่องที่ใช้เป็นแม่ข่ายโดยซึ่งเฉพาะที่จะตั้งอยู่ใน ISP ซึ่ง Bandwidth อย่างต่ำๆ นั้นเป็น 100 Mbit ทำให้โปรแกรมประเภท P2P ที่กล่าวมานั้นความเร็วที่ได้จะขึ้นอยู่กับว่าคนที่มีไฟล์ที่เรากำลังโหลดอยู่นั้นมีความเร็วในการส่งข้อมูลให้เรามากแค่ไหน รวมถึงมีคนที่มีไฟล์เดียวกับเรานั้นมีมากแค่ไหน ซึ่งจะเห็นได้ว่าความเร็วที่ได้นั้นมีเงื่อนไขหลายอย่างประกอบกันมากพอสมควร ระบบนี้คิดโดย "Bram Cohen"

**3.2.2 องค์ประกอบของโปรโตคอล Bittorrent**

**3.2.2.1 Torrent client**

Torrent client หรือที่เรียกกันติดปากว่า Program torrent ซึ่งมีให้เลือกกันลายตัวทีเดียว เช่นBitcomet ,Bit Tornado ,Azureus ,Torrent Storm ฯลฯซึ่งแต่ละตัวนั้นก็มีข้อแตกต่างกันนิดๆหน่อยในเรื่องความสามารถที่ผู้เขียนแต่ละคนเขียนเพิ่มลงไป แต่โดยพื้นฐานก็อยู่บน Protocal Bittorrent เดียวกันตัวProgram เหล่านี้ ใช้ในการดาวน์โหลด ไฟล์ ด้วย Protocal Torrent นั่นเอง

**3.2.2.2 Tracker server**

เรียกสั้นๆว่า Tracker ทำหน้าที่เป็นแม่ข่ายกลางระหว่าง Torrent client ด้วยกันและโดยส่วนมากแล้วจะเก็บไฟล์ .torrent ไว้ด้วยแต่ก็มีคนชอบมีคนสับสนว่าตัว Torrent portal เช่น พวก supernova ที่ชี้ไปยังไฟล์ Torrent ของ Tracker ที่อื่นนั้นเป็น Tracker สะเอง

**3.2.2.3 Torrent file**

ไฟล์นี้เป็นไฟล์ที่สำคัญมากๆ เพราะจะเก็บข้อมูลหลายอย่างเพื่อใช้ในการดาวน์โหลดไฟล์ ด้วย Protocol torrent นี้ไฟล์นี้จะถูกใช้โดย Torrent client เพื่ออ่านข้อมูลที่บรรจุอยู่ เช่น ที่อยู่ของ Tracker, Check sum ของไฟล์ที่เก็บไว้ หลังจากเมื่อเราใช้ Torrent client เปิด file .torrent แล้ว ตัว Program จะอ่านค่าที่อยู่ของ Tracker server ในไฟล์ที่เปิด แล้วติดต่อไปหาเพื่อทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเกี่ยวกับไฟล์ที่ต้องการไปหา จากนั้น ทาง Tracker server จะตรวจสอบว่ามีไฟล์ที่ขอมามีการลงทะเบียนไว้ในระบบหรือไม่ ถ้ามีจะตรวจดูว่ามีคนเข้ามาโหลดไฟล์นี้อยู่เท่าไหร่แล้วก็จะทำการส่งรายการ IP Address ของคนที่โหลดอยู่กลับไปรวมทั้งเก็บ IP Address ของเราไว้ด้วย เมื่อ Torrent client ได้ IP Address แล้วก็จะทำการติดต่อไปยัง IP Address ที่ได้มาโดยจะส่งคำถามไปว่า มีส่วนไหนอยู่บ้าง ทางปลายทางจะตอบกลับมา และตัว Torrent client จะตรวจว่าเรายังขาดส่วนไหนแล้วส่งคำร้องขอส่วนที่ต้องการไปให้ ทางปลายทางก็จะส่งส่วนที่เราขอกลับมาให้ขณะเดียวกันนั้นเองตัว Torrent client ก็จะเปิดรับคำร้องจากคนอื่นๆในแบบเดียวกัน ตรงนี้จะมีกฎอยู่เล็กน้อยคือเมื่อเราส่งคำร้องไปแล้วทางปลายทางจะทดสอบเราโดยการส่งคำร้องให้เราส่งส่วนที่เรามีกลับไปหาด้วยถ้าเราส่งได้ช้าหรือไม่ส่งเลยตัว Torrent client ปลายทางจะทำการปฏิเสทคำร้องจากเราหรือไม่ก็ส่งให้เราช้าๆ เหมือนที่เราส่งกลับไปหาเขาระบบนี้เป็นระบบต้อง แบ่งปัน "ให้มากได้มากให้น้อยได้น้อย" นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

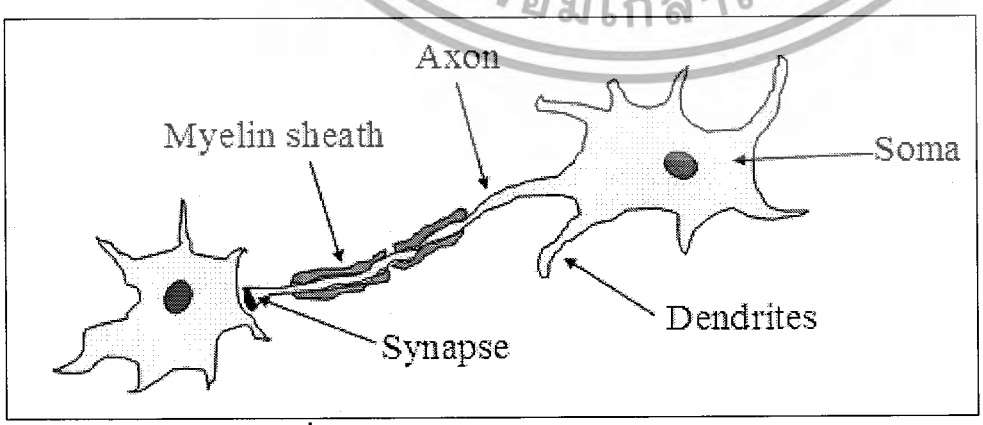
# บทที่ 4

# โครงข่ายประสาทเทียม (Artificial Neural Network)

## 4.1 โครงข่ายประสาทเทียม (Artificial Neural Network)

หรือที่มักจะเรียกสั้น ๆ ว่า ข่ายงานประสาท (Neural Network หรือ Neural Net) คือ แบบจำลองทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (Connectionist) เพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำแบบรูป (Pattern Recognition) และการอุปมานความรู้ (Knowledge Deduction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาข่ายงานไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ "นิวรอน" (Neurons) และ จุดประสานประสาท (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า "เดนไดรท์" (Dendrite) ซึ่งเป็นอินพุทเข้ามายัง "โซมา" (Soma) ซึ่งเป็นส่วนประกอบหลักของเซลล์ประสาท ส่งออกไปยังปลายในการส่งกระแสประสาทเรียกว่า "แอคซอน" (Axon) ซึ่งเป็นเหมือนเอาท์พุทของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาท (Action Potential) จะวิ่งผ่านเดนไดรท์เข้าสู่โซมาซึ่งจะเป็นตัวตัดสินว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอคซอนของมันไปยังเดนไดรท์ของเซลล์ถัดไป ซึ่งข้อดีของสมองมนุษย์นั้นคือสามารถเรียนรู้ได้ จึงเป็นแรงจูงใจในการพัฒนาโครงข่ายประสาทเทียม

ตามโมเดลนี้ข่ายงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาท จนเป็นโครงข่ายที่ทำงานร่วมกัน

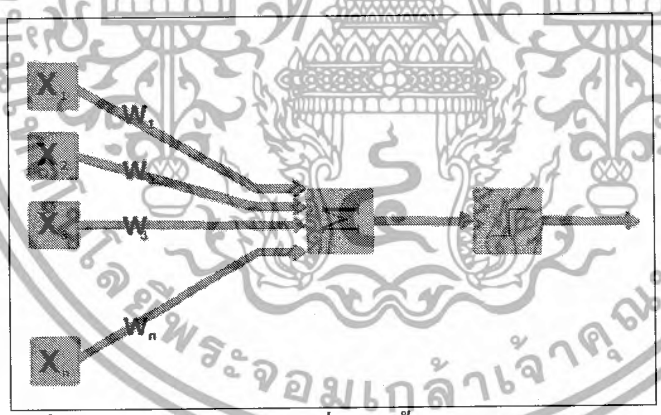



**รูปที่ 4.1 โครงสร้างตัวอย่างของเซลล์ประสาทชีวภาพ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบสร้างโครงข่ายประสาทเทียมนั้นมีสมมติฐานขั้นแรกจากคุณสมบัติของระบบประสาทชีวภาพ ดังที่กล่าวมา กล่าวคือ ชุดรับสัญญาณข้อมูล อินพุทของเซลล์ประสาทหนึ่งได้จากสัญญาณเอาท์พุทของเซลล์ประสาทอื่นๆ ผ่านทางจุดประสานประสาทและเดนไดรท์ ข้อมูลแต่ละค่าที่รับมาจะถูกลดขนาดด้วยจุดประสานประสาท ซึ่งภายในประกอบด้วยสารเคมีประเภท $K^{+}$ ,$Ca^{++}$ ,$Na^{+}$ ,$Cl^{-}$ ซึ่งจะมีลักษณะทางความนำพัลส์ (Pulse) สัญญาณไฟฟ้าเคมีที่แตกต่างกันด้วยเหตุนี้ โมดุลโครงข่ายประสาทเทียมที่สร้างขึ้น จะต้องมีการถ่วงน้ำหนักให้กับโมดุลก่อนที่จะนำเข้าสู่โมดุลประสาทเทียม จุดนี้เรียกว่า ซินแนปติกส์เวจท์ ปริมาณของข้อมูลที่เข้าสู่นิวรอลจะถูกนำมารวมกัน และตัดสินใจด้วยระดับความสนใจของนิวรอล (Activation level) แล้วจะส่งเป็นเอาท์พุทออกที่แอกซอนไปยังนิวรอลอื่นๆ

โครงข่ายประสาทเทียมมีโครงสร้างแตกต่างจากข่ายงานในสมอง แต่ก็ยังเหมือนสมอง ในแง่ที่ว่าโครงข่ายงานประสาทเทียม การรวมกลุ่มแบบขนานของหน่วยประมวลผลย่อย ๆ และการเชื่อมต่อนี้เป็นส่วนสำคัญที่ทำให้เกิดสติปัญญาของข่ายงาน เมื่อพิจารณาขนาดแล้วสมองมีขนาดใหญ่กว่าข่ายงานประสาทเทียมอย่างมาก รวมทั้งเซลล์ประสาทยังมีความซับซ้อนกว่าหน่วยย่อยของข่ายงาน อย่างไรก็ดีหน้าที่สำคัญของสมอง เช่น การเรียนรู้ยังคงสามารถถูกจำลองขึ้นอย่างง่ายด้วยโครงข่ายประสาทนี้




**รูปที่ 4.2 ไดอะแกรมนิวรอลที่สร้างขึ้น (Artificial Neuron)**

จากภาพแสดงถึงโมดุลที่สร้างขึ้นโดยแนวความคิดจากเซลล์สมองชีวภาพ สัญญาณอินพุทคือ $X_1, X_2, \ldots, X_n$ จะถูกป้อนเข้าไปยังนิวรอลที่สร้างขึ้น ซึ่งเปรียบเทียบได้กับสัญญาณที่ป้อนเข้ายังซินแนปส์ของนิวรอลชีวภาพ สัญญาณอินพุทนี้จะนำไปคูณกับค่าซินแนปติกส์เวจท์ที่มีค่าตั้งแต่ 0-1 $W_1, W_2, \ldots, W_n$ ก่อนที่จะเข้าสู่บล็อกซัมเมชั่น ซึ่งค่าถ่วงน้ำหนักนี้จะสอดคล้องกับค่าสเตรงท์ (Strength) ของจุดต่อซินแนปส์ชีวภาพแต่ละจุด (Single biological synaptic connection) บล็อกซัมเมชั่นนี้ก็จะทำหน้าที่สอดคล้องคล้ายกับตัวเซลล์สมองชีวภาพ ผลรวมทางคณิตศาสตร์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

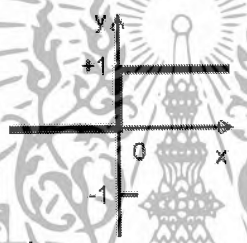
อินพุตและเวจท์จะได้เป็นเอาท์พุท เราเรียกว่า เน็ต (NET) ซึ่งเราจะรวมกันในรูปของเวกเตอร์ได้ดังนี้

$$NET = X_1 W_1 + X_2 W_2 + ... + X_n W_n \quad (4.1)$$

## 4.2 ฟังก์ชันกระตุ้นความสนใจ (Activation Function)

เมื่อได้สัญญาณ NET แล้วกระบวนการต่อมาที่นิวรอลต้องทำคือตัดสินใจ เราจึงต้องกำหนดฟังก์ชันการตัดสินใจ เพื่อใช้เป็นระดับของการตัดสินใจให้กับนิวรอล เพื่อให้ได้สัญญาณเอาท์พุทของนิวรอลออกมา ซึ่งเชื่อมต่อไปยังนิวรอลตัวอื่นๆ เป็นโครงข่าย OUT

**4.2.1 สเตปฟังก์ชัน (Step function)** หรือลีเนียร์เทรชโฮลฟังก์ชัน (linear threshold function)




**รูปที่ 4.3 กราฟที่ได้จากฟังก์ชันของสเตปฟังก์ชัน**

สเตปฟังก์ชัน หรือลีเนียร์เทรชโฮลฟังก์ชันนั้นจะนำสัญญาณที่เข้ามาคูณกับค่าถ่วงน้ำหนัก (Weight) ค่าผลรวมจะมาเทียบกับค่าเทรชโฮล หากผลรวมมากกว่าค่าเทรชโฮลแล้วค่าระดับความสนใจของนิวรอลเท่ากับ +1 ถ้าค่าน้อยกว่าเทรชโฮลนั้นค่าระดับความสนใจของนิวรอลจะเป็น -1 (อาจแทนด้วย 0 ในบางระบบ)

$$X = \sum_{i=1}^{n} w_i x_i \quad (4.2)$$

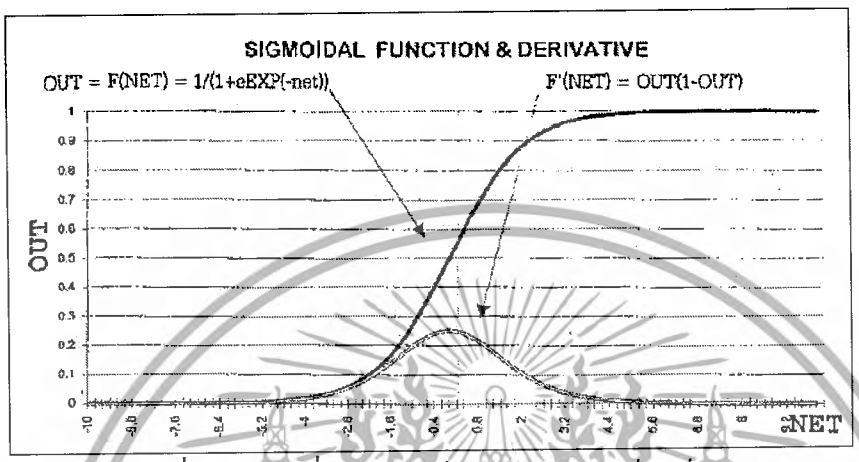
โดย X เป็นค่าผลรวมระหว่างค่าถ่วงน้ำหนักกับอินพุต

$$Y = \begin{cases} +1 \; for \; X > t \\ 0 \; for \; X \le t \end{cases} \quad (4.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Y เป็นค่าระดับความสนใจของนิวรอล และ t เป็นค่าเทรสโฮลคงที่ หรืออาจเป็นฟังก์ชันอื่นๆ ที่เลื่อนแบบคุณสมบัติที่ไม่เป็นเชิงเส้นของเซลล์ประสาทชีวภาพได้อย่างใกล้เคียงกว่า และใช้เป็นฟังก์ชันให้กับโครงข่ายทั่วไปได้

**4.2.2 ซิกมอยด์ฟังก์ชัน (Sigmoid function)**



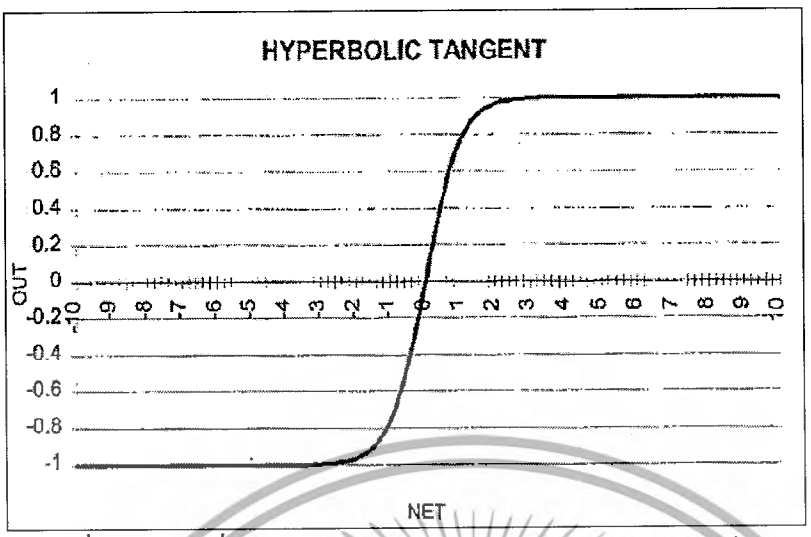

**รูปที่ 4.4 กราฟที่ได้จากฟังก์ชันของซิกมอยด์ฟังก์ชัน**

ลักษณะของเทรชโฮลฟังก์ชันมีลักษณะเป็นนอลลีเนียร์ฟังก์ชัน (Non-linear function ) เช่น เอสเคิร์ฟว (S-Curve) เราจะได้ค่าเอาท์พุทที่มีความไวต่อสัญญาณอินพุทที่มีขนาดเล็กๆ และเฉลี่ยต่อสัญญาณแรงๆ ซึ่งสัญญาณอ่อนๆไปทางบวกเพียงเล็กน้อยก็จะให้ผลใกล้เคียง "1" กระตุ้นหรือสัญญาณอ่อนๆ ทางลบเพียงเล็กน้อยก็จะทำให้ผลใกล้เคียง "0" (ยับยั้ง) ขณะที่สัญญาณแรงๆ ทางบวกก็ยังคงให้ผลใกล้เคียง "1" และสัญญาณทางลบแรงๆ ก็ยังคงให้ผลใกล้เคียง "0" เช่นกัน 00 จะมีอัตราขยายแบบนอลลีเนียร์ (non- linear gain) ซึ่งคุณลักษณะแบบนี้สามารถแก้ปัญหานอยซ์แซททูเรชันดิเลมมา (Noise-saturation dilemma) ได้ และทำให้นิวรอลเทียมที่สร้างขึ้นสามารถทำงานกับขนาดของอินพุทได้กว้างมากขึ้น โดยครอบคลุมอินพุตในช่วง $-\infty$ ถึง $+\infty$ โดยฟังก์ชันของซิกมอยด์ฟังก์ชันคือ

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{4.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 ไฮเปอร์โบลิกแทนเจนฟังก์ชัน (Hyperbolic tangent function)




**รูปที่ 4.5 กราฟที่ได้จากฟังก์ชันของไฮเปอร์โบลิกแทนเจนฟังก์ชัน**

มีลักษณะคล้ายกับซิกมอยด์ฟังก์ชันแต่เนื่องจากมันจะมีความสมมาตรจึงให้ผลอยู่ระหว่าง "-1" ถึง "1" ผลจะเป็น "0" เมื่อเน็ตเป็น "0" และผลเข้าใกล้ "1" เมื่ออินพุทไปทางบวกและเข้าใกล้ "-1" เมื่อ อินพุทมีทิศทางไปทางลบ โดยฟังก์ชันของไฮเปอร์โบลิกแทนเจนฟังก์ชันคือ

$$\text{Tanh}(x) \qquad (4.5)$$

## 4.3 การฝึกสอนให้โครงข่ายประสาทเทียม (Training of Artificial Neural Networks)

ค่าถ่วงน้ำหนัก มีความสัมพันธ์กับอะไร เปลี่ยนแปลงอย่างไร นั้นก็เช่นเดียวกับเด็กที่คลอดออกมาก็มีสมองแล้วแต่สมองยังไม่เจริญเติบโตเพียงพอ และยังไม่ได้รับการฝึกสอน และเรียนรู้ เด็กจึงไม่สามารถทำกิจกรรมใดๆด้วยตนเอง เว้นแต่กิจกรรมที่ธรรมชาติสร้างมาพร้อมกับการกำเนิดที่เรียกว่า "สัญชาติญาณ" ซึ่งธรรมชาติใส่คุณลักษณะบางอย่างให้เซลล์สมองบางส่วนตั้งแต่ทารก เจริญเติบโตอยู่ในครรภ์มารดา เช่น ระบบควบคุมการหายใจ, ความรู้สึก, การเรียกร้องเมื่อหิว, การตอบสนองต่อสิ่งเร้า ฯลฯ เด็กจะพัฒนาการเรียนรู้ไปตามขั้นตอน หลังจากนั้นสมองของเขาจะได้รับการฝึกสอน และเจริญเติบโตไปพร้อมๆกัน เซลล์สมองจะได้รับการปรับคุณลักษณะสอดคล้องกับการฝึกสอน และจะเจริญเป็นโครงข่ายสอดคล้องกัน

โครงข่ายประสาทเทียมที่สร้างขึ้นมีลักษณะเช่นเดียวกัน คือ เมื่อสร้างเสร็จ แต่ละเซลล์ประสาทที่สร้างขึ้นมานั้น จะยังไม่มีคุณลักษณะใดเลย เนื่องจากยังไม่มีการกำหนดค่าซินแนปติกส์เวจท์ที่เหมาะสมกับงานที่ต้องการให้กับมัน จึงต้องมีการฝึกสอนเพื่อให้โครงข่ายที่สร้างขึ้นมีคุณลักษณะตามที่ต้องการ การฝึกสอนของโครงข่ายประสาทเทียม จะกระทำโดยการปรับเปลี่ยนค่าซินแนปติกส์เวจท์เพื่อให้โครงข่ายจดจำรูปแบบความสัมพันธ์ระหว่างอินพุทกับเอาท์พุทได้ โดยใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นแรกกำหนดเป็นค่าสุ่มใดๆ (Random weight) ก่อนแล้วถึงปรับเปลี่ยนค่าถ่วงน้ำหนักไปตามอัลกอริทึมสมมติฐานหลายๆรอบจนกว่าจะได้ผลลัพธ์ของโครงข่าย เหมือนกับผลลัพธ์ที่ต้องการในเงื่อนไขความผิดพลาดที่ยอมรับได้

## 4.4 วัตถุประสงค์ของการฝึกฝน (Objective of Training)

เนื่องจากค่าถ่วงน้ำหนักที่ให้เป็นค่าสุ่มใดๆ โครงข่ายจึงไม่แสดงคุณลักษณะใดออกมา การฝึกสอน (Training) ให้โครงข่ายก็คือการปรับค่าค่าถ่วงน้ำหนักทุกๆจุดให้สอดคล้องกับอินพุทหลายๆแบบ เพื่อให้ได้เอาท์พุทตามต้องการนั่นเอง การฝึกสอนโครงข่าย จะต้องบรรลุถึงกระบวนการเข้าใจพื้นฐานเสียก่อน คือการเรียนรู้ในโครงข่ายประสาทเทียมนั้นมีขีดจำกัด ปัญหาต่างๆ ผู้ใช้คงต้องแก้ไขให้มันก่อน แล้วนำผลนั้นไปอ้างอิงสำหรับการปรับปรุงค่าค่าถ่วงน้ำหนัก หลังจากปรับค่าถ่วงน้ำหนักจนได้ค่าผิดพลาดที่เอาท์พุทเทียบกับเป้าหมายน้อยลงเป็นที่พอใจแล้ว โครงข่ายประสาทเทียมนั้นก็พร้อมที่จะวิเคราะห์อินพุทและให้เอาท์พุทตามลักษณะตัวอย่างที่มันเคยเรียนรู้มาการเรียนรู้จะมีการปรับค่าถ่วงน้ำหนักหลายๆรอบ จนค่าถ่วงน้ำหนักสอดคล้องกับตัวอย่างหลายๆตัวอย่าง และให้เอาท์พุทตามต้องการ พบว่าโครงข่ายได้ตัวอย่างสำหรับการฝึกฝนมากๆ โครงข่ายก็จะมีความแม่นยำสูงขึ้น แต่ก็จะใช้เวลาในการฝึกฝนเพิ่มขึ้นเช่นกัน หากพิจารณาต่อไปจะพบว่า โครงข่ายประสาทเทียมที่สร้างขึ้นจะมีพฤติกรรมคล้ายกับระบบการเรียนรู้ของมนุษย์มากเป็นเพราะมีต้นแบบมาจากระบบประสาทชีวภาพนั่นเอง

### 4.4.1 การฝึกฝนแบบควบคุม (Supervised Training)

วิธีการฝึกฝนถูกจัดเป็น 2 ประเภท คือ แบบควบคุม และแบบอิสระ (Unsupervised training) โดย การฝึกฝนแบบควบคุม จะต้องการคู่ของการฝึกฝนระหว่างอินพุทกับเป้าหมายที่ต้องการที่เรียกว่า คู่การฝึกฝน (Training pairs) โครงข่ายจะถูกเทรนไปตามจำนวนของคู่ที่ฝึกฝน (จำนวนคู่ของอินพุทกับเอาท์พุทที่ต้องการให้โครงข่ายรู้จัก) เอาท์พุทที่คำนวณได้จากโครงข่ายจะถูกเปรียบเทียบกับความสอดคล้องกับเป้าหมาย ค่าผิดพลาดที่เกิดขึ้นจะถูกป้อนกลับไปยังโครงข่ายและเปลี่ยนแปลงค่าถ่วงน้ำหนักให้สอดคล้องกับวิธีการ ที่ทำให้แนวโน้มของค่าผิดพลาดที่เกิดขึ้นระหว่างผลลัพธ์กับเป้าหมายโดยเฉลี่ยมีค่าลดต่ำลง ตัวอย่างการฝึกฝนแบบนี้ ได้แก่ การฝึกฝนแบบแพร่กลับ (Back-propagation) ซึ่งการฝึกฝนแบบควบคุมนั้นจะต้องทำการฝึกฝนโครงข่ายก่อนที่จะนำข้อมูลที่ไม่เคยจำแนกในโครงข่าย

### 4.4.2 การฝึกฝนแบบอิสระ

ถึงแม้ว่าวิธีการแบบควบคุมสามารถจะประยุกต์ใช้เพื่อปรับคุณลักษณะของโครงข่ายได้สำเร็จ แต่ก็ยังมีข้อวิจารณ์อยู่ คือ มันเป็นไปอย่างชีวภาพไม่ได้ และยากที่จะเชื่อได้ว่า กลไกการฝึกฝนของสมองจะต้องการ การเปรียบเทียบระหว่างค่าที่ต้องการกับผลลัพธ์จริง โดยกระบวนการป้อนกลับไปแก้ไขคุณลักษณะของโครงข่าย และถ้าสมมติว่า ถ้าสมองมีกลไกเช่นนี้ ต้องมีผู้หา

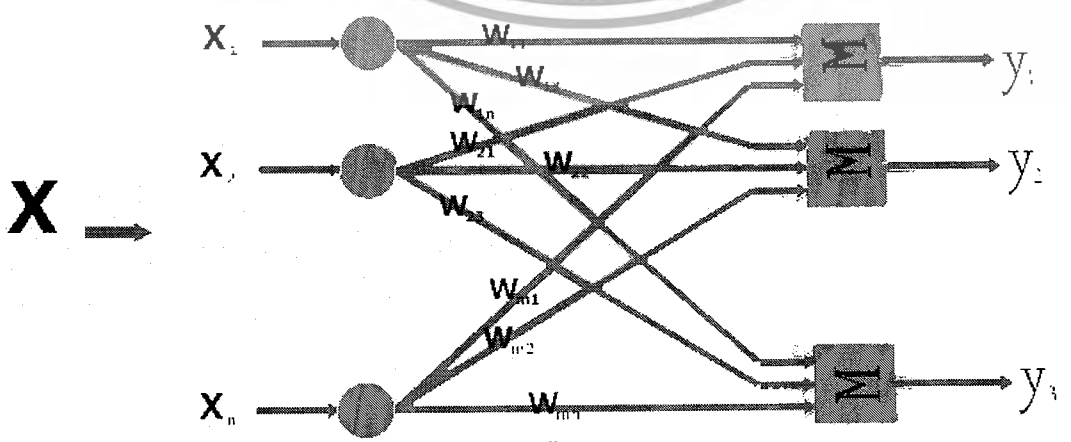
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ต้องการเพื่อนำมาเป็นเป้าหมายตลอดเวลา และจะเอามาจากที่ใดสรุปคือ ต้องมีผู้คิดเป้าหมายให้กับโครงข่ายก่อน โครงข่ายไม่สามารถคิดและปรับคุณลักษณะได้ก่อนด้วยตนเอง ในทางตรงกันข้ามหากพิจารณาทารกแรกเกิดสมองของเขาสามารถจัดระบบเองได้อย่างไร การฝึกฝนแบบอิสระที่สร้างขึ้นคงยังห่างไกลความเป็นไปได้ ที่จะมีลักษณะการฝึกฝนแบบระบบของสมอง จนกระทั่งมีการพัฒนาการฝึกฝนแบบอิสระนี้ขึ้นปี 1984 ได้เสนอแนวคิดที่เป็นการฝึกฝนแบบไม่ต้องการเป้าหมายไม่มีการตัดสินใจด้วยเหตุผลในอุดมคติมาก่อน

ชุดข้อมูลของการฝึกฝน จะมีเพียงอินพุทเวกเตอร์เท่านั้น ฝึกฝนอัลกอริทึมจะเปลี่ยนแปลงค่าถ่วงน้ำหนักของโครงข่าย เพื่อสร้างเอาท์พุทที่มีความคงที่ ยกตัวอย่างเช่น หากให้โครงข่ายรู้จำภาพหน้าคนหนึ่ง หากภาพหน้าคนคนนั้นเปลี่ยนแปลงไปเล็กน้อย (ภาพอาจมีสัญญาณรบกวนรวมอยู่บ้าง) โครงข่ายนั้นก็ยังสามารถบอกได้ว่าคนคนนั้นเป็นคนเดิม เป็นต้น การฝึกฝนจะไม่มีการตัดสินใจมาก่อน ไม่มีการกำหนดแบบเอาท์พุทมาก่อน (อาจกล่าวได้ว่าแบบเอาท์พุทจะถูกกำหนดโดยอินพุทเวกเตอร์นั่นเอง) ดังนั้น เอาท์พุทของโครงข่ายก็เช่นกัน ส่วนใหญ่จะถูกแปรรูปซึ่งจะเข้าใจได้ภายหลังกระบวนการฝึกฝน ดังนั้นจึงไม่สามารถแก้ปัญหาที่เคร่งครัดสำคัญได้ แต่มักนิยมใช้โครงข่ายแบบนี้กับงานง่ายๆ ประเภทการเปรียบเทียบเอกลักษณ์ รูปแบบที่สัมพันธ์กันระหว่างอินพุท-เอาท์พุท ที่ถูกกำหนดโดยโครงข่าย

## 4.5 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Artificial Neural Networks)

ที่กล่าวมาจะถึงจุดนี้ เป็นการกล่าวถึงหลักการและเหตุผลในการสร้างเซลล์ประสาทเทียมเพียงหนึ่งเซลล์ โดยใช้แนวความคิดจากเซลล์ประสาทชีวภาพ การจะนำเซลล์ประสาทเทียมมาใช้งานได้นั้นต้องใช้เซลล์ประสาทเทียมที่มีคุณลักษณะต่างๆกัน (ค่าถ่วงน้ำหนักจะทำให้คุณสมบัติของเซลล์ประสาทเทียมแต่ละเซลล์มีคุณลักษณะแตกต่างกันไป) มาเชื่อมโยงเป็นโครงข่ายในลักษณะเดียวกับเซลล์สมองชีวภาพเสียก่อน ซึ่งลักษณะการเชื่อมโยงมีหลายชนิด แต่ละชนิดก็มีคุณลักษณะเด่นที่แตกต่างกันไป




**รูปที่ 4.6 ลักษณะโครงข่ายประสาทเทียมแบบชั้นเดียว (Single-Layer Neural Networks)**

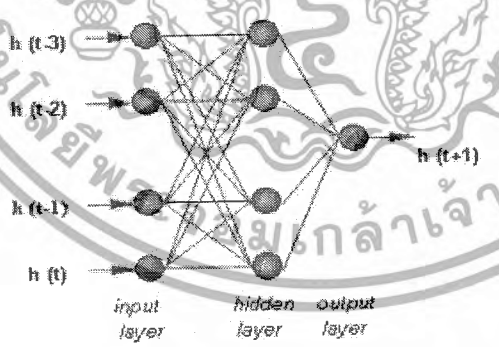
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป เป็นโครงข่ายประสาทเทียมแบบชั้นเดียว ที่ประกอบด้วยเซลล์ประสาทเทียมง่ายๆ หลายชุด ความสามารถในการคำนวณของโครงข่ายประสาทเทียมได้มาจากลักษณะการเชื่อมต่อเป็นโครงข่ายประสาทเทียมโครงข่ายง่ายๆเป็นกลุ่มโมดุลประสาทเทียมที่เชื่อมต่อกันเป็นชั้นๆ (Layer)

อย่างไรก็ดีลักษณะการเชื่อมโยงระหว่างโครงข่ายไม่ได้มีแบบเดียว การเชื่อมโยงระหว่างชั้นอาจมีการเชื่อมโยงย้อนกลับมาที่ชั้นอินพุทอีก ซึ่งโครงข่ายประสาทชีวภาพก็มีลักษณะดังกล่าวเช่นกัน สำหรับค่าน้ำหนัก มีวิธีการพิจารณาในรูปของ เวจท์เมตริก (Weight matrix) ซึ่งหากโครงข่ายมีหลายชั้น จะช่วยให้ระบุค่าถ่วงน้ำหนักได้ง่ายขึ้น และเพื่อหลีกเลี่ยงความสับสนจะกำหนดเป็นมิติ (Dimensions) ของเมตริก โดยให้ m แทนจำนวนแถว หรือจำนวนของอินพุท และ n แทน จำนวนของนิวรอล ที่สร้างขึ้น ตัวอย่างเช่น ค่าถ่วงน้ำหนักที่เชื่อมระหว่างอินพุทตัวที่ 4 กับนิวรอลตัวที่ 2 คือ $W_{4,2}$

## 4.6 โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Artificial Neural Networks)

โครงข่ายที่ซับซ้อนจะมีความสามารถในการคำนวณที่ดีขึ้นมันจะเป็นโครงข่ายที่มีโครงสร้างเป็นจินตนาการที่น่าเป็นไปได้โดยการจัดการเชื่อมโยงนิวรอลมีโครงสร้างเป็นชั้นๆ คล้ายส่วนหนึ่งของสมองและมีการพัฒนาวิธีการเกี่ยวกับการฝึกสอนให้โครงข่ายแบบหลายชั้นทำงานได้ ตามความต้องการแล้วเมื่อไม่นานมานี้โครงข่ายแบบหลายชั้นอาจจะสร้างจากกลุ่มของโครงข่ายแบบชั้นเดียวเอาท์พุทของชั้นหนึ่ง จะใช้เป็นอินพุทของชั้นถัดไป



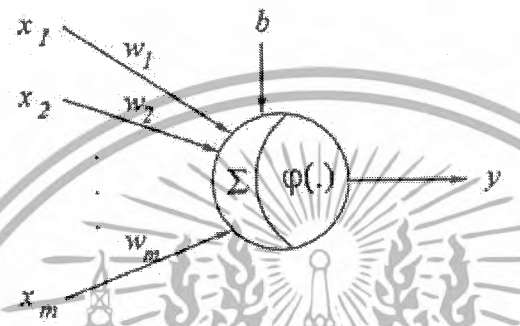

**รูปที่ 4.7 โครงข่ายแบบฟีดฟอร์เวิร์ด (Feed-forward network)**

จากภาพเป็นโครงข่ายแบบฟีดฟอร์เวิร์ดโดยชั้นแรกเป็นอินพุตเลเยอร์ (Input layer) แต่ละโหนดในชั้นนี้ได้รับสัญญาณจากภายนอก ซึ่งชั้นนี้อาจไม่ใช่นิวรอลก็ได้ แล้วส่งสัญญาณนี้ไปยังชั้นถัดไปคือ ฮิดดันเลเยอร์ (Hidden layer) ในชั้นนี้ประกอบด้วยนิวรอล สัญญาณที่รับเข้ามาจะถูกส่งไปยังเอาท์พุตเลเยอร์ (Output layer) เพื่อส่งสัญญาณที่ได้ออกไป เมื่อโครงข่ายแบบฟีดฟอร์เวิร์ดผ่านการฝึกฝนแล้วสถานะจะไม่เปลี่ยนถ้าข้อมูลใหม่เข้ามา กล่าวคือระบบนี้ไม่มีหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาเหตุที่เรียกว่าฟีดฟอร์ดเวิร์ดเนื่องจากข้อมูลที่ได้รับมาจากอินพุตโหนด (Input node) จะถูกส่งไปยังเอาท์พุตโหนด (Output node) โดยไม่มีการส่งข้อมูลบางส่วนของเอาท์พุตโหนด กลับมายังอินพุตโหนดระบบที่มีการส่งข้อมูลบางส่วนของเอาท์พุตโหนดกลับมานั้นเรียกว่า recurrent network

## 4.7 เพอร์เซ็ปตรอน (Perceptrons)




**รูปที่ 4.8 แผนภาพของเพอร์เซ็ปตรอน**

เพอร์เซ็ปตรอนเป็นโมเดลแรกของโครงข่ายประสาทเทียมที่เป็นแบบควบคุม (Rosenblatt, 1958) ซึ่งเป็นสามารถแยกข้อมูลที่รับมาแบ่งได้เป็นสองประเภท โดยสามารถนำไปใช้เพื่อจำแนกรูปและในงานจำพวกการรู้จำต่างๆ ได้

เพอร์เซ็ปตรอนใช้สเตปฟังก์ชัน ถ้าค่าผลลัพธ์เป็น +1 แสดงว่าผลรวมระหว่างการคูณของข้อมูลที่รับมาและค่าถ่วงน้ำหนักมีค่ามากกว่า เทรชโฮล และในทางกลับกันถ้าค่าน้อยกว่าค่าผลลัพธ์จะเป็น -1 ฟังก์ชันของ step สามารถเขียนได้ดังนี้

$$Step(X) = \begin{cases} +1 \; for \; X > t \\ -1 \; for \; X \le t \end{cases} \tag{4.6}$$

ฟังก์ชันกระตุ้นความสนใจของเพอร์เซ็ปตรอนสามารถเขียนได้ดังนี้

$$Y = Step\left(\sum_{i=0}^{n} w_i x_i\right) \tag{4.7}$$

การสอนเพอร์เซ็ปตรอนนั้นสามารถสังเกตจากผลลัพธ์ที่ได้จากการแบ่งแยก ถ้าผลลัพธ์ไม่ถูกต้องต้องทำการเปลี่ยนแปลงค่าถ่วงน้ำหนักเพื่อทำให้ใกล้กับข้อมูลที่รับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w_i \longleftarrow w_i + (a \times x_i \times e) \tag{4.8}$$

เมื่อ e คือ ค่าความผิดพลาดของผลลัพธ์

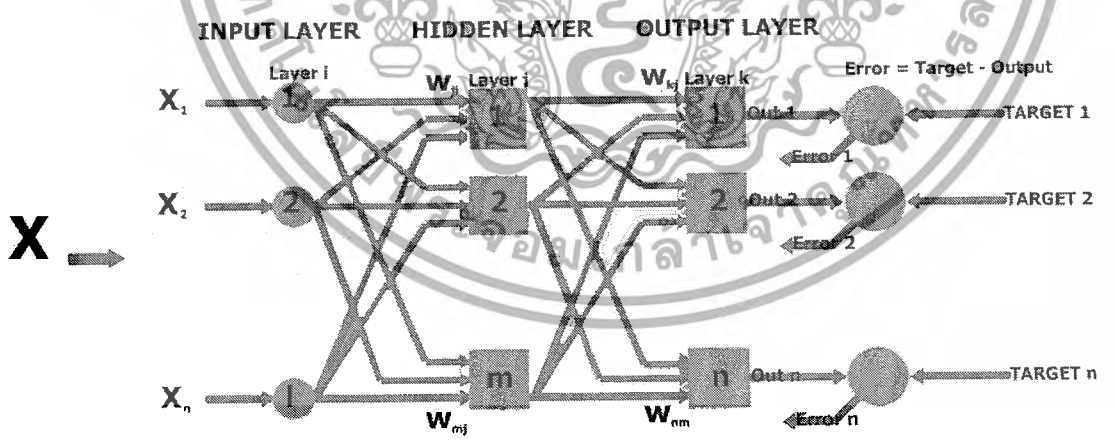
a คือ ค่าอัตราการเรียนรู้ ( 0 < a < 1)

ถ้าค่าความผิดพลาดของผลลัพธ์เป็นศูนย์แสดงว่าผลลัพธ์นั้นถูกต้อง ถ้าผลลัพธ์มีค่ามากจะทำการลดค่าถ่วงน้ำหนักสำหรับค่าของข้อมูลที่รับมาที่มีค่าเป็นบวก และการสอนถ้าทำในครั้งแรกไม่ถูกต้องก็จะใช้วิธีเดิมสอนจนกระทั่งได้ผลที่ถูกต้อง เรียกวิธีว่า epoch

## 4.8 แบคพรอพเกชัน

แบคพรอพเกชันเป็นโครข่ายที่มีโครงสร้างแบบมัลติเลเยอร์ (Multilayer network) การเรียนรู้ควบคุมโดยแต่ละนิวรอลมีค่าถ่วงน้ำหนักช่วยกับอินพุต และค่าถ่วงน้ำหนักจะถูกปรับเปลี่ยนเมื่อมีค่าความผิดพลาดจากฝึกฝน โดยส่วนใหญ่ใน multilayer back-propagation network จะนิยมใช้ซิกมอยด์ฟังก์ชัน

วิธีการแบคพรอพเกชันจะกำหนดค่าเริ่มต้นของค่าถ่วงน้ำหนักโดยการสุ่มค่า จากนั้นจะนำข้อมูลเข้าโครงข่ายจนกระทั่งได้ผลลัพธ์ แล้วนำค่าความผิดพลาดจากผลลัพธ์ส่งกลับเป็นข้อมูลทำให้ค่าถ่วงน้ำหนักเปลี่ยนไป ทำวิธีการนี้จนกระทั่งได้ผลลัพธ์ใกล้ค่าตามที่ต้องการ หรือจนกระทั่งค่าความผิดพลาดมีค่าน้อยมากๆ




**รูปที่ 4.9 แผนภาพของโครงข่ายแบคพรอพเกชันแบบสองชั้น**

เราพิจารณาทั้งสามชั้นของโครงข่ายโดยให้ i แทนโหนดในอินพุตเลเยอร์ j แทนโหนดในฮิดดันเลเยอร์ และ k แทนโหนดในเอาท์พุตเลเยอร์ $w_{ij}$ แทนค่าถ่วงน้ำหนักระหว่างโหนดในอินพุตเลเยอร์ และโหนดในฮิดดันเลเยอร์

ฟังก์ชันที่ใช้หาค่าผลลัพธ์จากโหนด j ในโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_j = \sum_{i=1}^{n} x_i . w_{ij} - \theta_j \tag{4.9}$$

$$Y_j = \frac{1}{1+e^{-X_j}} \tag{4.10}$$

เมื่อ n แทนจำนวนของข้อมูลในโหนด j $w_{ij}$ แทนค่าถ่วงน้ำหนักของจุดเชื่อมต่อระหว่างโหนด i และ j $\theta_j$ แทนค่าเทรชโฮล ที่ใช้ในโหนด j โดยการสุ่มค่าระหว่าง 0 ถึง 1 $x_j$ แทนข้อมูลจากอินพุตโหนด I และ $y_j$ แทนผลที่ได้จากโหนด j

ข้อมูลที่ส่งเข้าโครงข่ายจนถึงผลลัพธ์ ค่าความผิดพลาด (Error gradient) จะถูกคำนวณจากแต่ละโหนด k ในเอาท์พุตเลเยอร์โดยค่าความผิดพลาดถูกกำหนดจากผลต่างระหว่างค่าที่ต้องการและค่าที่ได้จริง

$$e_k = d_k - y_k \tag{4.11}$$

ค่าความผิดพลาดของอาท์พุตโหนด k ถูกกำหนดจากค่าความผิดพลาดจากโหนดคูณกับค่าที่ได้จากฟังก์ชันกระตุ้นความสนใจ

$$\delta_k = \frac{\partial y_k}{\partial x_k} . e_k = y_k . (1 - y_j) . e_k \tag{4.12}$$

เมื่อ $x_k$ แทนผลรวมของค่าอินพุตในโหนด k

ค่าความผิดพลาดสำหรับแต่ละโหนด j ในฮิดดันเลเยอร์จาก

$$\delta_j = y_j . (1 - y_j) \sum_{k=1}^{n} w_{jk} \delta_k \tag{4.13}$$

เมื่อ n เป็นจำนวนของโหนดในเอาท์พุตเลเยอร์โดยที่จำนวนของผลลัพธ์ได้มาจากแต่ละโหนดในฮิดดันเลเยอร์

ค่าถ่วงน้ำหนักในโครงข่ายจะถูกทำให้เป็นข้อมูลที่ถูกต้องดังสูตร

$$w_{ij} \longleftarrow w_{ij} + \alpha . x_i . \delta_j \tag{4.14}$$

$$w_{jk} \longleftarrow w_{jk} + \alpha . y_i . \delta_k \tag{4.15}$$

เมื่อ $x_i$ เป็นข้อมูลจากอินพุตโหนด i และ $\alpha$ เป็นอัตราการเรียนรู้ ซึ่งควรเป็นค่าบวกและมีค่าไม่เกิน 1 อีกทั้งไม่ควรเป็นค่าที่มากจนเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.8.1 การพัฒนาประสิทธิภาพของแบคพรอพเกชัน**

เนื่องจากวิธีการแบคพรอพเกชันเมื่อนำไปแก้ปัญหานั้นพบว่ามีแนวโน้มค่อนข้างช้า ในบางปัญหาอาจทำร้อยหรือพันครั้งของ epoch จึงทำให้ได้ค่าความผิดพลาดที่เป็นที่พอใจได้ จึงได้มีการรวมแรงกระตุ้น (Momentum) ในสูตรเพื่อช่วยแก้ค่าถ่วงน้ำหนัก โดยคิดเฉพาะค่าถ่วงน้ำหนักที่เปลี่ยนในรอบที่ผ่านมา เมื่อ t แทนรอบปัจจุบันและ t-1 แทนรอบที่ผ่านมา เราสามารถเขียนกฎการเรียนรู้ได้ดังนี้

$$\Delta w_{ij}(t) = \alpha.x_i.\delta_j + \beta\Delta w_{ij}(t-1) \tag{4.16}$$

$$\Delta w_{jk}(t) = \alpha.y_i.\delta_k + \beta\Delta w_{jk}(t-1) \tag{4.17}$$

เมื่อ $\Delta w_{ij}(t)$ เป็นผลรวมการเพิ่มค่าถ่วงน้ำหนักของการเชื่อมต่อระหว่างโหนด i และ j

$\beta$ แทนค่าแรงกระตุ้นมีค่าระหว่าง 0 ถึง 1 หากค่าเป็น 0 แสดงว่าไม่มีการใช้แรงกระตุ้น

กฎที่มีการเพิ่มค่าแรงกระตุ้นเรียกว่า generalized delta rule และการเพิ่มแรงกระตุ้นนั้นยังทำให้วิธีการแบคพรอพเกชันหลีกเลี่ยงจาก local minima และทำให้เคลื่อนผ่านพื้นที่ที่มีค่าความผิดพลาดที่เกือบไม่มีการเปลี่ยนแปลงได้เร็วขึ้น

อาจมีการใช้ไฮเปอร์โบลิกแทนเจนฟังก์ชันแทนซิกมอยด์ฟังก์ชันเพื่อให้เพอร์เซ็ปตรอนทำงานได้เร็วยิ่งขึ้น สมการของไฮเปอร์โบลิกแทนเจนฟังก์ชัน คือ

$$\tanh(x) = \frac{2a}{1+e^{-bx}} - a \tag{4.18}$$

เมื่อ a และ b เป็นค่าคงที่ เช่น a = 1.7 และ b = 0.7

## 4.9 Recurrent Networks

recurrent network เป็นการเรียนรู้แบบควบคุมสามารถนำข้อมูลย้อนกลับจากผลลัพธ์ไปยังอินพุตโหนด และสถานะสามารถเปลี่ยนแปลงจากข้อมูลที่ได้รับ กล่าวคือระบบมีหน่วยความจำ ในการแก้ปัญหาไม่ว่าสถานการณ์ใดๆนั้นปัจจัยจะขึ้นกลับข้อมูลก่อนหน้าทั้งหมด

recurrent network นำข้อมูลผ่านเข้าระบบ รวมถึงนำข้อมูลย้อนกลับจากผลลัพธ์ไปยังอินพุต และจะทำกระบวนการนี้ซ้ำจนกระทั่งผลลัพธ์ไม่เปลี่ยน เรียกสถานะนี้ว่า equilibrium หรือ stability และในบางกรณี recurrent network ไม่สามารถไปสถานะที่คงที่ได้ เนื่องจากค่าต่างของผลลัพธ์ที่เปลี่ยนแปลงตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.10 Hopfield Network

โดยมากจะใช้ sign activation function เป็นฟังก์ชันกระตุ้นความสนใจโดยมีสมการดังนี้

$$Sign(X) = \begin{cases} +1 \; for \; X > 0 \\ -1 \; for \; X < 0 \end{cases} \tag{4.19}$$

ในฟังก์ชันกระตุ้นความสนใจจะไม่กำหนด Sign(0) เนื่องจากถ้าโครงข่ายได้รับข้อมูล 0 หมายถึง สถานะเดิมกล่าวคือ ระบบคงที่แล้ว

การทำงานของ hopfield network จะใช้เครื่องหมายเมทริคซ ( matrix arithmetric) โดยค่าถ่วงน้ำหนักจะแทนด้วย matrix W ดังนี้

$$W = \sum X_i X_i' - NI \tag{4.20}$$

เมื่อ $X_i$ แทน vector ของข้อมูล

$X_i'$ แทน matrix transposition

I แทน เมตริกซ์เอกลักษณ์ขนาด m x m

N แทน จำนวนของสถานะ ( $X_i$ ) ที่สามารถเรียนรู้ได้เวกเตอร์ผลลัพธ์ถูกกำหนด

โดย $Y_i = Sign(WX_i - \theta)$ เมื่อ $\theta$ แทนค่า threshold matrix

ระยะการพัฒนาการใช้ hopfield network แบ่งได้ 3 ระยะคือ

1. ใช้สำหรับฝึกฝนโครงข่ายให้สามารถจดจำได้
2. ทดสอบโครงข่าย โดยการใส่ข้อมูลแล้วตรวจสอบผลลัพธ์ว่าข้อมูลถูกต้องหรือไม่
3. นำระบบไปใช้จริง

Hamming distance ใช้วัดจำนวนของสอง vector ที่แตกต่างกัน เช่นถ้าวัดระหว่าง vector X และ Y แทนด้วย $\|X, Y\|$

Hopfield network มีหน่วยความจำสำหรับนำ vector ที่รับมาใส่ใน vector หน่วยความจำของ hamming distance จาก vector ที่ได้รับที่ค่าน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.11 Bidirectional Associative Memories (BAMs)

โครงสร้างคล้ายกับ hopfield network เป็นการเรียนรู้แบบควบคุม และสามารถใช้ associate item จากกลุ่มหนึ่งไปใช้ในกลุ่มอื่นได้ โดยโครงข่ายประกอบด้วย 2 ชั้น แต่ละโหนดในชั้นแรกจะเชื่อมต่อกับทุกโหนดในชั้นอื่น (fully connected) ซึ่งต่างจาก hopfield network ที่จะเชื่อมทุกโหนดภายในชั้นเดียวกันเท่านั้น แต่ในและใช้ sign activation function เหมือนกับ hopfield network

weights matrix ถูกกำหนดโดย

$$W = \sum_{i}^{n} X_i Y_i' \tag{4.21}$$

เมื่อ X, Y เป็นกลุ่มข้อมูลที่จะทำการเรียนรู้

การทดสอบโครงข่ายเมื่อแทนข้อมูล $X_1$ โครงข่ายจะได้ผลลัพธ์ตาม vector ($Y_1$) นี้ $Sign(W'X_1)$ และในทางกลับกันสามารถแทน $Y_1$ แล้วได้ $X_1$ ดังนี้ $Sign(WY_1)$ ดังนั้น BAM จะรับประกันความคงที่ของผลลัพธ์ไม่ว่าจะใส่ข้อมูลใดก็ตาม แล้วทำการฝึกฝนข้อมูลนั้น

## 4.12 Kohonen Maps

kohonen map หรือ self-organizing feature map โดย kohonen map ใช้วิธีการ winner – take –all เป็นการเรียนรู้แบบอิสระอาจเรียกว่า competitive learning ซึ่งวิธีการ winner – take –all นั้นใช้หลักการที่ใช้หน่วยประสาทเดียวจัดการผลลัพธ์ของโครงข่ายในการตอบสนองให้ข้อมูลที่ได้รับ

จุดประสงค์ของ kohonen map เป็นการรวมกลุ่มของข้อมูลที่ได้รับเข้ากับกลุ่มของการรวมกลุ่ม เช่นถ้า kohonen map ได้รับกลุ่มหัวข้อใหม่จะนำหัวข้อนั้นไปรวมกับประเภทของหัวเรื่อง kohonen map จะไม่บอกว่าอยู่กลุ่มใดแต่จะบอกว่าเป็นของส่วนไหน kohonen map จะมีประโยชน์ในการจัดกลุ่มข้อมูลที่ไม่รู้จัก

kohonen map ประกอบด้วย 2 ชั้นคืออินพุตเลเยอร์ และcluster layer โดย cluster layer จัดการเหมือนเอาท์พุตเลเยอร์แต่ละอินพุตเลเยอร์ จะเชื่อมต่อกับทุกๆ โหนดใน cluster layer และโหนดใน cluster layer ถูกจัดเตรียม grid information ถึงแม้ไม่มีความจำเป็นก็ตาม

การฝึกฝนตาม kohonen map เริ่มจากทุกค่าถ่วงน้ำหนักนั้นทำการสุ่มเป็นค่าน้อยๆ ค่าอัตราการเรียนรู้สุ่มเป็นค่าบวกเลขน้อยๆ จากนั้นแทนอินพุตเวกเตอร์เข้าไปในอินพุตเลเยอร์ของแผนที่ในอินพุตเลเยอร์ส่งข้อมูลที่ได้รับไปยัง cluster layer หน่วยประสาทใน cluster layer ที่เข้ากับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่รับมากที่สุดจะเป็นผู้ชนะ โดยหน่วยประสาทจะจัดการผลที่ได้จากการจัดกลุ่มของแผนที่ แล้วค่าถ่วงน้ำหนักจะเปลี่ยนแปลง

ในการพิจารณาว่าหน่วยประสาทใดจะชนะนั้น เวกเตอร์ที่ได้จะถูกเปรียบเทียบกับอินพุตเวกเตอร์ โดยค่าหน่วยประสาทใดมีค่าใกล้เคียงอินพุตเวกเตอร์จะเป็นผู้ชนะ

Euclidean distance ($d_i$) จาก input vector x ของหน่วยประสาทด้วย weight vector $w_i$ คำนวณโดย

$$d_i = \sqrt{\sum_{j=1}^{n}\left(w_{ij} - x_j\right)^2} \tag{4.22}$$

เมื่อ n เป็นจำนวนของหน่วยประสาทในอินพุตเลเยอร์

ในโครงข่ายที่มีค่า $d_i$ มีค่าน้อยที่สุดจะเป็นผู้ชนะ และหน่วยประสาทนั้นจะปรับปรุงค่าถ่วงน้ำหนักดังนี้ $w_{ij} = w_{ij} + \alpha(x_j - w_{ij})$ ในทางปฏิบัติแล้วนั้นเมื่อหน่วยประสาทที่ชนะปรับปรุงค่าถ่วงน้ำหนักแล้วนั้น หน่วยประสาทที่อยู่ใกล้หน่วยประสาทที่ชนะทั้งสองแนวก็จะปรับปรุงค่าด้วย การฝึกฝนจะสิ้นสุดเมื่อการปรับปรุงค่าถ่วงน้ำหนักมีค่าน้อยมากๆจากกลุ่มของหน่วยประสาท

## 4.13 ข้อดีของโครงข่ายประสาทเทียม

1.เซลล์ของมนุษย์หรือสิ่งมีชีวิตทั้งหลายโครงข่ายประสาทชีวภาพซึ่งทำงานแบบขนาน คือ ส่วนต่าง ๆ ของข่ายประสาททำงานพร้อม ๆ กันในเวลาเดียวกัน

2. เกิดความสามารถในการเรียนรู้ขึ้นได้

## 4.14 ข้อเสียของโครงข่ายประสาทเทียม

1. โครงข่ายประสาทเป็นวิธีที่ยากต่อการทำความเข้าใจในแบบที่ถูกผลิตออกมา

2. โครงข่ายประสาทมีคุณสมบัติที่ไวต่อรูปแบบของข้อมูลที่รับมา ถ้าเราแทนข้อมูลด้วยรูปแบบที่ แตกต่างกันก็จะสามารถผลิตผลลัพธ์ที่แตกต่างกันออกมา ดังนั้นการกำหนดค่าเริ่มต้นให้กับ ข้อมูลจึงเป็นส่วนที่มีความสำคัญส่วนหนึ่ง

## 4.15 การพัฒนาของโครงข่ายประสาทเทียม

การนำ Genetic algorithm นำมาหาค่าเริ่มต้นของค่าถ่วงน้ำหนักสำหรับโครงข่าย เพื่อลดปัญหาเกี่ยวกับวิธีการค้นหา อีกทั้งช่วยให้จำนวนของหน่วยประสาท และการเชื่อมต่อระหว่างหน่วยประสาทมีจำนวนที่เหมาะสมได้

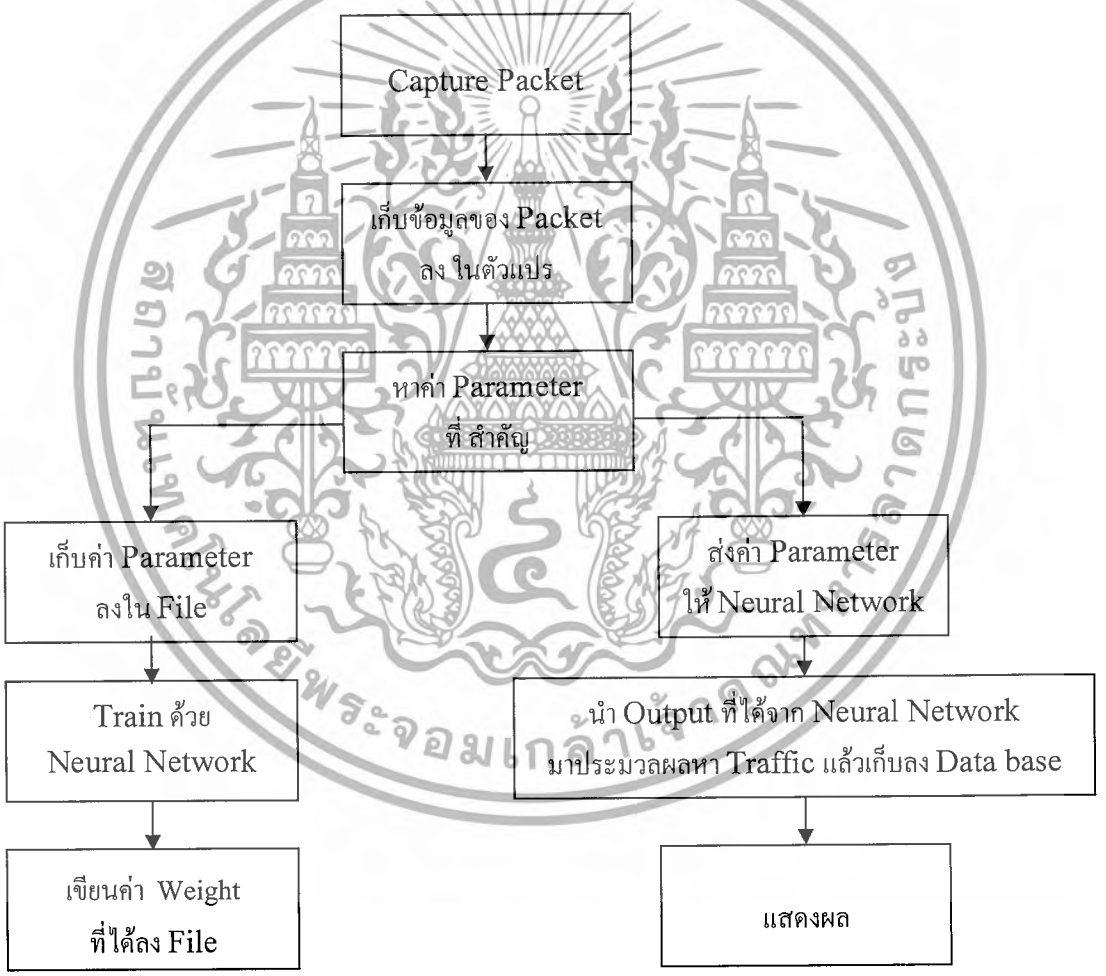
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# การออกแบบและพัฒนา

## 5.1 การออกแบบ

ขั้นตอนการทำงานของซอฟต์แวร์จะแบ่งออกเป็นสามส่วนหลักๆคือ

- การ TRAIN ข้อมูลด้วย NEURAL NETWORK
- การแยกประเภทข้อมูล NEURAL NETWORK
- การเก็บข้อมูลและนำไปแสดงผล




**รูปที่ 5.1 โครงสร้างภาพรวมของโปรแกรม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 การเขียนโปรแกรมดักจับPacket

### 5.2.1 การพัฒนา Program โดยใช้ RAW Socket

การจับ Packet โดยใช้ Socket จะใช้รูปแบบ RAW Socket โดยจะตั้งค่า IP Address เป็น IP Address ของ Ethernet Card ที่จะรับ Packet และตั้งค่าให้รับเฉพาะ Packet ที่เป็น IP Protocol ซึ่งจะรับทุก Packet ที่ผ่าน Ethernet Card

### 5.2.2 WinPcap Library

WinPcap คือ Library ที่เป็น open source โดยจะนำ ฟังก์ชั่น มาใช้บางส่วนคือ

- ฟังก์ชั่น pcap_findalldevs() ใช้ค้นหา Ethernet Card ทั้งหมดบนเครื่อง
- ฟังก์ชั่น pcap_open_live() ใช้เลือก Ethernet Card ที่ใช้จับ packet
- ฟังก์ชั่น pcap_next_ex() ใช้รับ packet

## 5.3 การหาค่า Parameter

### 5.3.1โครงสร้างของ ตัวแปรที่ใช้เก็บค่าข้อมูล

เราสร้างตัวแปรแบบ Class เพื่อใช้ในการเก็บค่าต่างๆ ขึ้นมา 3 ตัว

- ctpt เป็นตัวแปรที่ใช้เก็บข้อมูล ต่างๆของ host ที่สนใจ เช่น traffic รวมทั้งหมดที่เกิดขึ้นทั้งขาเข้าและขาออก traffic ที่เกิดขึ้นในช่วง 1 segment เวลา start เวลา stop จำนวน Connection ทั้งหมดที่เกิดขึ้นของ host ที่เราสนใจ จำนวน IP ที่ host ติดต่อด้วย
- ctipp เก็บจำนวน connection ที่ใช้ในการติดต่อระหว่าง host กับแต่ละ Public IP
- parameter เป็นตัวแปรที่ใช้เก็บข้อมูล ต่างๆของ connection ที่สนใจ เช่น traffic รวมทั้งหมดที่เกิดขึ้นทั้งขาเข้าและขาออก traffic ที่เกิดขึ้นในช่วง 1 segment เวลา start เวลา stop ทั้งหมดที่เกิดขึ้น และสถานะที่บ่งชี้ว่าเป็น P2P หรือไม่

### 5.3.2 นำค่าข้อมูลที่ได้จากการ Capture มาเก็บในโครงสร้างตัวแปรที่สร้างขึ้น

### 5.3.3 นำค่าที่อยู่ในตัวแปรมาคำนวณหาค่า Parameter ที่ใช้ในการ Train ด้วย Neural Network

ซึ่งค่า Parameter ต่างๆที่ได้จะถูกเขียนลงใน file เป็น block block ละ128 byte

## 5.4 Neural network Training

เราเลือกชนิดของ Neural Network ที่ใช้คือ back propagation การทำงานเริ่มจากอ่าน parameter จาก File เป็น block block ละ128 byte เพื่อสร้าง Target ขึ้นจาก ค่า Status ใน parameter จากนั้นนำ parameter ไป Train ด้วย Neural Network โดยใช้ฟังก์ชั่น feed Forward() แล้ว Output Node ที่ได้จาก feed Forward () จะนำไปคำนวณกับ Target จริงๆ เพื่อจะนำไปปรับค่า weight ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกต้องแล้วกลับไป Train ต่อไปหลังจากนั้นเมื่อ Train เสร็จก็จะนำ weight ที่ได้จากการ Train มาทดสอบว่า weight ที่ได้นั้นเมื่อนำมาใช้งานจะมีความถูกต้องเป็นกี่ เปอร์เซ็นต์

## 5.5 Traffic Classification

หลังจากได้ค่า Weight ที่เหมาะสมแล้ว เรานำค่า Weight ที่ได้ไปใช้ในการแยกประเภทข้อมูลด้วยโปรแกรมที่พัฒนาขึ้นในส่วนของการแยกประเภทข้อมูล ในขั้นตอนนี้จะใช้โปรแกรมที่พัฒนาคล้ายกับหัวข้อที่ 5.2 จนถึง 5.3 แต่ไม่ได้เขียนค่า parameter ลงใน File หลังจากได้ค่า parameter มาเรานำค่า parameter ป้อนเป็น input ให้กับ Neural Network ซึ่งในขณะนี้ได้เรียนรู้ค่า weight เหมาะสมแล้ว Neural Network จะทำการแยกประเภทแล้วให้ค่า output ออกมาว่า parameter ที่เป็น Input นั้นเป็น P2P หรือไม่

ขั้นตอนต่อมาเราสร้างตัวแปรขึ้น 4 ตัวเพื่อเก็บค่า Traffic ที่ได้จากการแยกประเภทโดยมีรายละเอียดดังนี้

- All in เก็บค่า Traffic ขาเข้าทั้งหมด
- All out เก็บค่า Traffic ขาออกทั้งหมด
- P2P in เก็บค่า Traffic ที่เป็น P2P ขาเข้าทั้งหมด
- P2P out เก็บค่า Traffic ที่เป็น P2P ขาออกทั้งหมด

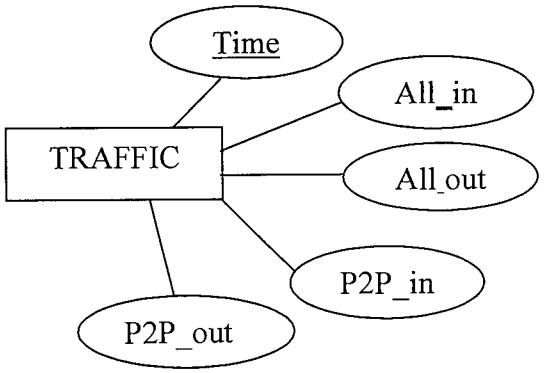
ตัวแปรทั้ง 4 ตัวนี้จะถูกประมวลผลในช่วงระยะเวลาทุกๆ 10 วินาที จากนั้นจะส่งค่าที่ได้ไปเก็บลง DATA BASE ในขั้นตอนถัดไป

## 5.6 การจัดเก็บข้อมูล

### 5.5.1 ตาราง TRAFFIC

ข้อมูลที่ได้จากการวิเคราะห์ด้วยโครงข่ายประสาทเทียมแล้ว จะถูกส่งมาเขียนลงบน DATABASE โดย DATABASE ที่เลือกใช้คือ MY SQL ในครั้งแรกที่ดีออกแบบฐานข้อมูลออกแบบมาเพียงตารางเดียวคือตารางTRAFFICซึ่งจะเก็บข้อมูลเพื่อแสดงผลของทั้งหมดโดยจะเก็บข้อมูลทุกๆ 10 วินาทีต่อการเขียนข้อมูลลง 1 ROW ตารางTRAFFICที่ออกแบบมามีดังนี้ที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




TRAFFIC

| TIME | All_in | All_out | P2P_in | P2P_out |
|---|---|---|---|---|

**รูปที่ 5.2 ER ไดอะแกรมตารางTRAFFIC**

แต่เมื่อเราใช้ฐานข้อมูลเพียงตารางเดียวหาเรียกหน้าที่ดูย้อนหลังมากๆทำให้ดึงข้อมูลจากฐานข้อมูลจำนวนมากทำให้เกิดความล่าช้าในการประมวลผลดังนี้

**ตารางที่ 5.1 เปรียบเทียบความเร็วในการประมวลผล**

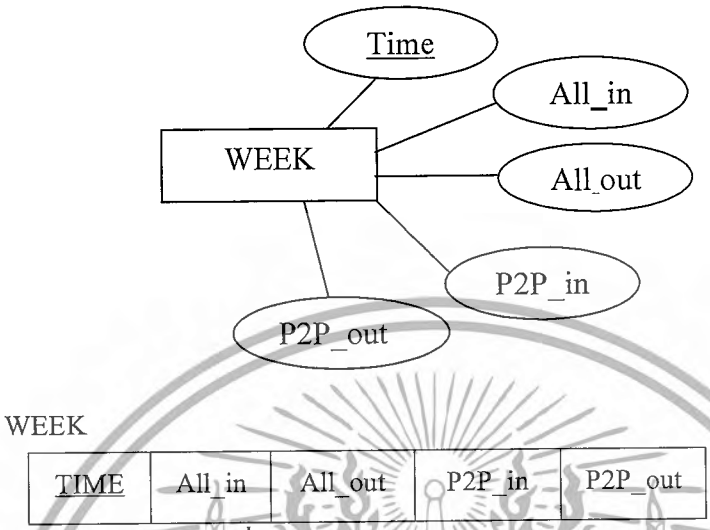
| Type | row | Speed |
|---|---|---|
| Hours | 360 | Quick |
| 24Hours | 8,640 | Normal |
| 7Days | 60,480 | Slow |
| 30Days | 259,200 | Very slow |
| 30 Months | 3,110,400 | Bad |

จากตาราง จะเห็นได้ว่ายิ่งดูย้อนหลังมากเท่าไหร่จะใช้จำนวนข้อมูลที่มากขึ้นและทำให้ถึงขั้นแสดงผลไม่ได้ในบางกรณีจึงได้หาวิธีแก้ไขจนในที่สุดก็พบว่าการเพิ่มตารางเป็นตารางย้อนหลังเป็นวิธีที่แก้ไขปัญหาได้และน่าจะเป็นวิธีที่เหมาะสมที่สุดจึงได้ให้ข้อมูลในตาราง TRAFFIC เดิมแสดงผลเฉพาะช่วง 1 ชั่วโมงกับช่วง 1วัน และ เพิ่มตาราง WEEK เก็บข้อมูลเพื่อแสดงผลของสัปดาห์ เดือน และ ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.5.2 ตาราง WEEK**

ตาราง WEEK และใช้ข้อมูลในตารางโดยมีลักษณะเหมือนตารางTRAFFIC ดังนี้




**รูปที่ 5.3 ER ไดอะแกรมตารางWEEK**

โดยค่าที่ได้จากการวิเคราะห์ด้วยโครงข่ายประสาทเทียมจะเก็บไว้ที่ตารางTRAFFIC เหมือนเดิมและตารางWEEKจะได้จากผลรวมของทุกๆ168Rowที่เป็นค่าเก่ากว่า8640Rowล่าสุดในตารางTRAFFICโดยทุกๆ 168Row ในตารางTRAFFICรวมกันจะได้ 1Row บันทึกลงในตาราง WEEK โดยจำนวนRowที่จะเหลืออยู่ในตาราง TRAFFIC คือ

จำนวนRowที่เหลือในตารางTRAFFIC = 8640 + ((จำนวน Row ทั้งหมด – 8640) % 168))

และจำนวน Row ที่เกิดขึ้นในตารางWEEKคือ

จำนวน Row ที่เกิดขึ้นในตารางWEEK = ((จำนวน Row ทั้งหมด – 8640) / 168)

โดยกระบวนการย้ายรวมข้อมูลในตารางดังที่กล่าวมาเพื่อไม่ให้เป็นภาระของตัวใดตัวหนึ่ง จึงให้กระบวนการดังกล่าวฝังอยู่ทุกๆหน้าของเว็บเมื่อแก้ไขโดยวิธีนี้สามารถลดข้อบกพร่องของการประมวลผลได้ดังนี้

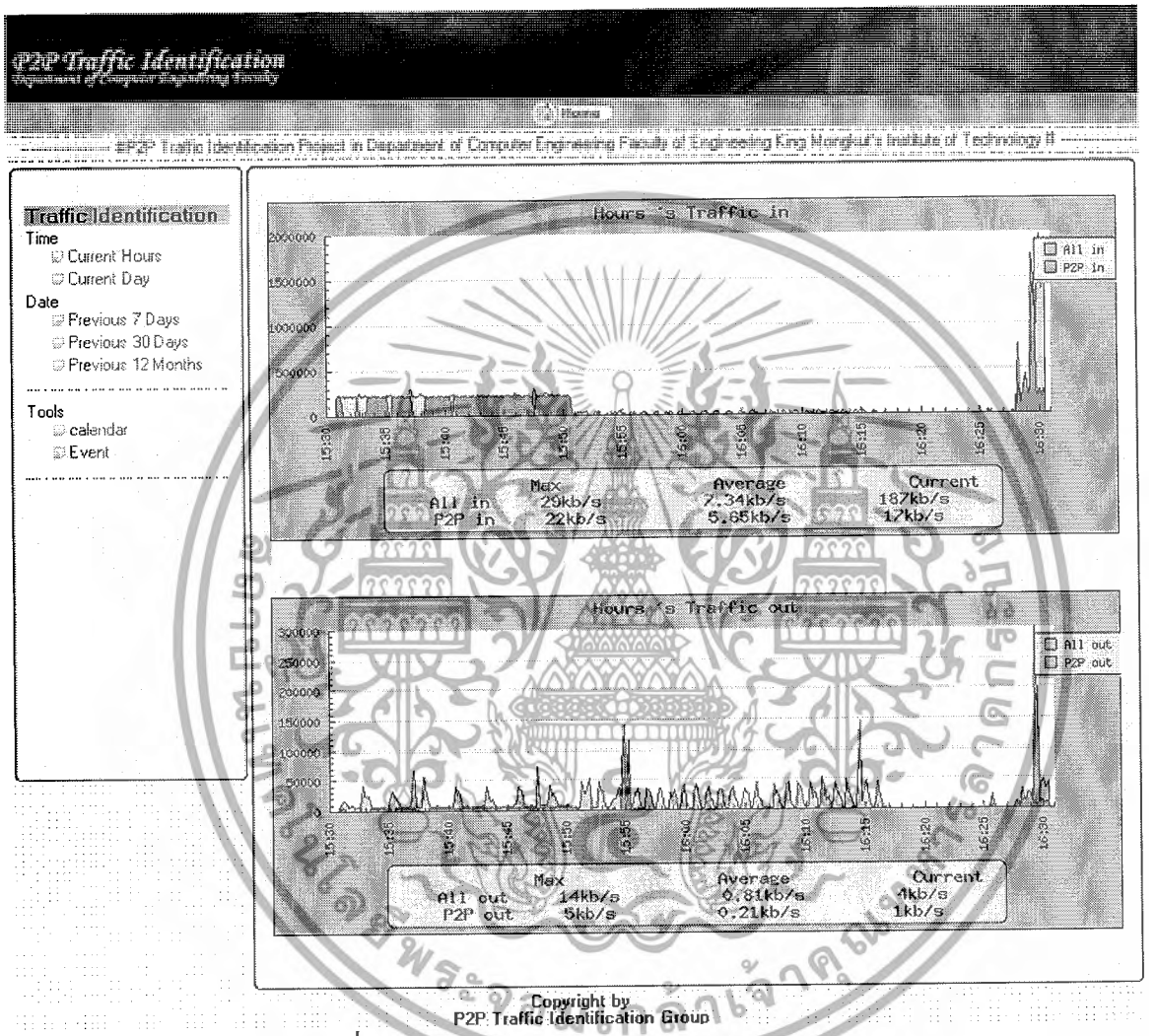
**ตารางที่ 5.2 เปรียบเทียบความเร็วในการประมวลผล**

| Type | row | Speed |
|---|---|---|
| Hours | 360 | Quick |
| 24Hours | 8,640 | Normal |
| 7Days | 360 | Quick |
| 30Days | 1,542 | Normal |
| 30 Months | 18,504 | Normal |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.7 การแสดงผล

ในส่วนของการแสดงผลจะแสดงผลผ่าน WEB BROWSER โดยใช้ภาษา SQL อ่านข้อมูลใน MY SQL ออกมาแสดงเป็นกราฟ แยกประเภท TRAFFIC ทั่วไป กับ TRAFFIC ประเภท P2P โดยแสดงทั้ง TRAFFIC ขาเข้า และ TRAFFIC ขาออก

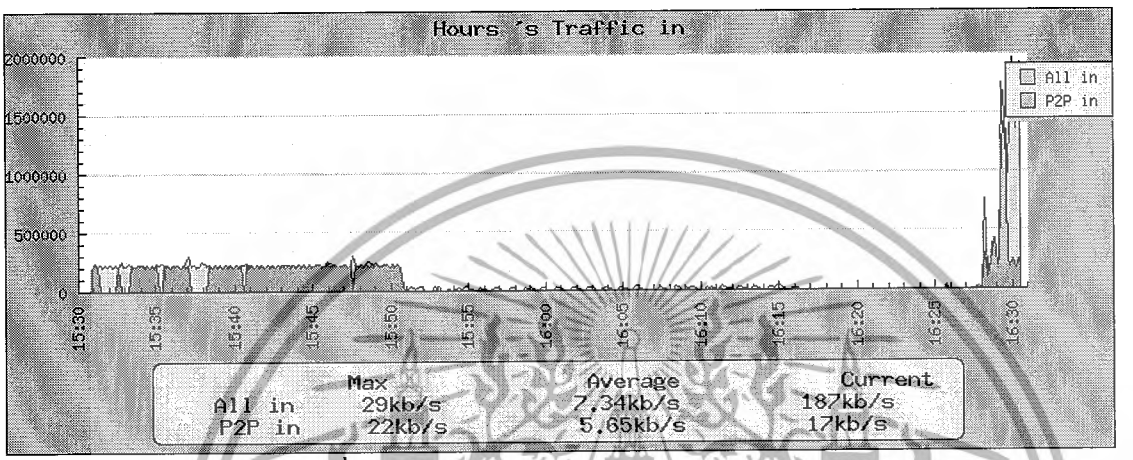



รูปที่ 5.4 ตัวอย่างการแสดงผลผ่าน WEB BROWSER

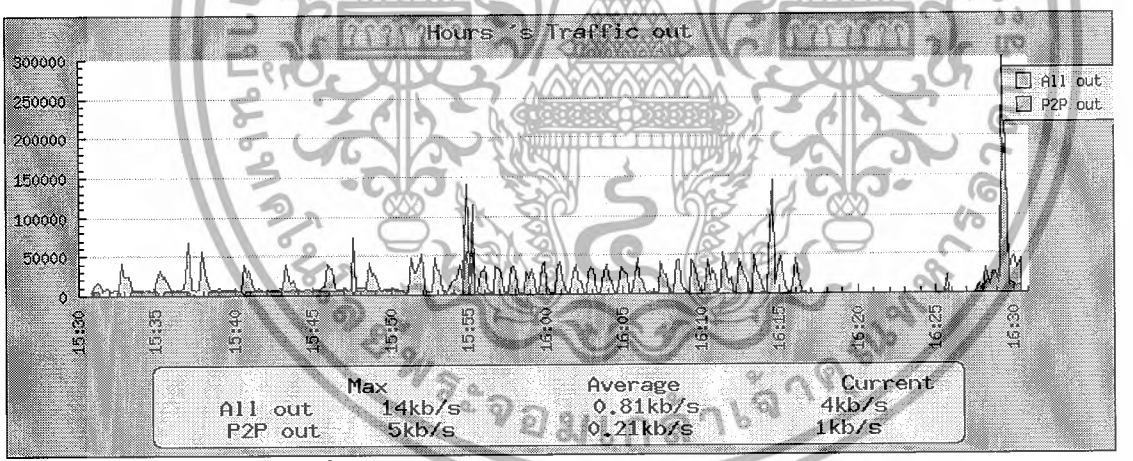
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.1 การแสดงผล CURRENT HOURS

การแสดงผล CURRENT HOURS ใช้ข้อมูลจากตาราง TRAFFIC จำนวน 360 Row แสดงผล TRAFFIC IN และ TRAFFIC OUT ทั้ง P2P และ NON P2P โดยจะมีการแสดงค่าสูงสุดของ Bandwidth ที่เกิดขึ้น ค่าเฉลี่ยของ Bandwidth ที่เกิดขึ้นภายในชั่วโมงปัจจุบันและค่าของ Bandwidth ที่เกิดขึ้น ณ เวลาปัจจุบัน




**รูปที่ 5.5 ตัวอย่างการแสดงผล CURRENT HOURS IN**

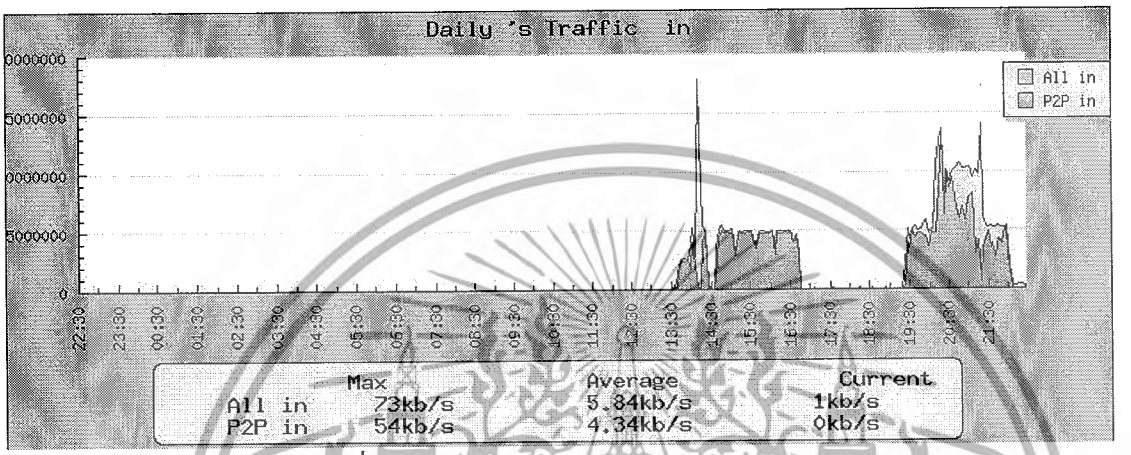



**รูปที่ 5.6 ตัวอย่างการแสดงผล CURRENT HOURS OUT**

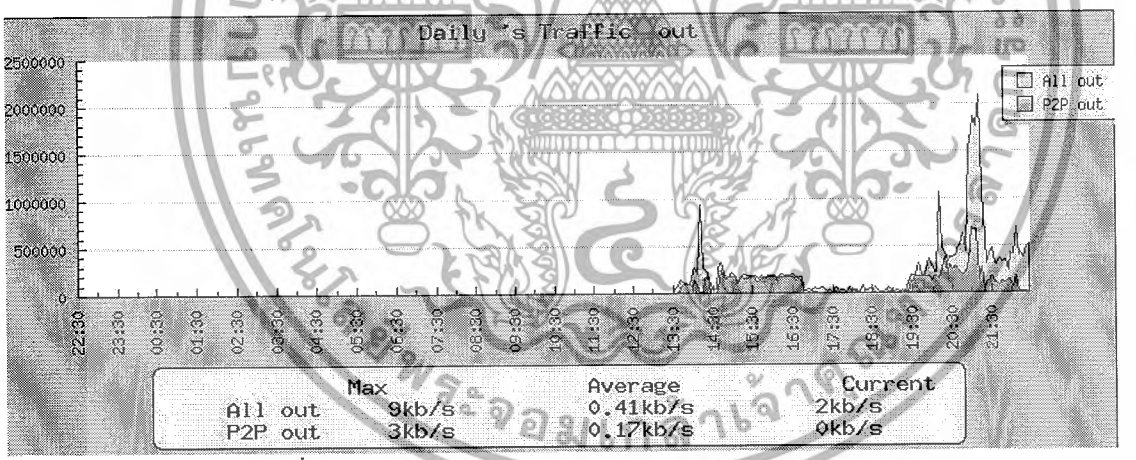
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.2 การแสดงผล CURRENT DAY

การแสดงผล CURRENT DAY ใช้ข้อมูลจากตาราง TRAFFIC จำนวน 8640 Row แสดงผล TRAFFIC IN และ TRAFFIC OUT ทั้ง P2P และ NON P2P โดยจะมีการแสดงค่าสูงสุดของ Bandwidth ที่เกิดขึ้น ค่าเฉลี่ยของ Bandwidth ที่เกิดขึ้นภายในชั่วโมงปัจจุบันและค่าของ Bandwidth ที่เกิดขึ้น ณ เวลาปัจจุบัน




**รูปที่ 5.7 ตัวอย่างการแสดงผล CURRENT DAY IN**

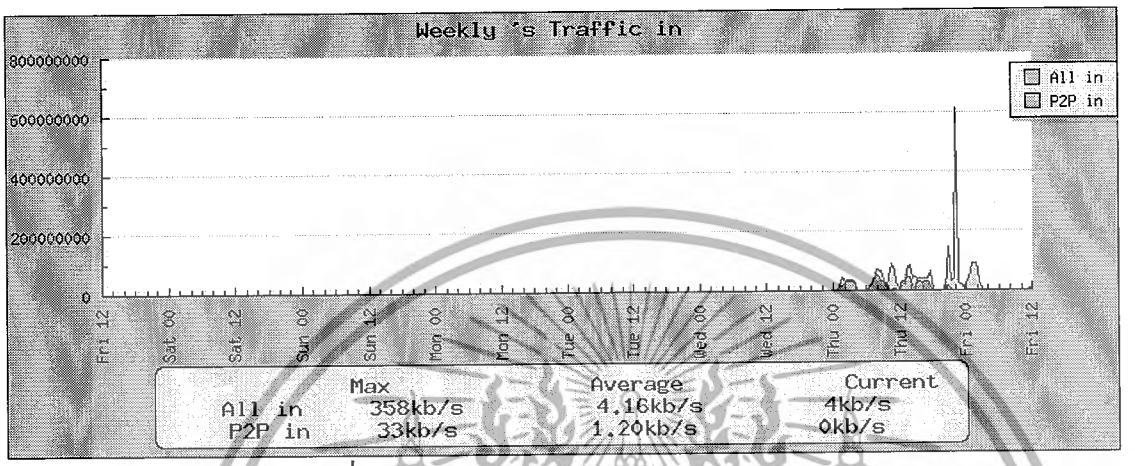



**รูปที่ 5.8 ตัวอย่างการแสดงผล CURRENT DAY OUT**

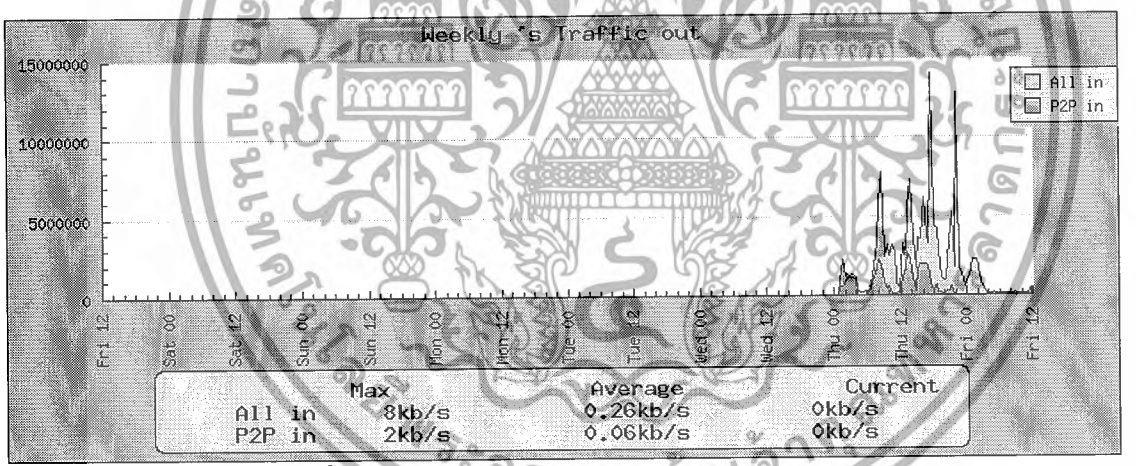
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.3 การแสดงผล PREVIOUS 7 DAY

การแสดงผล PREVIOUS 7 DAY ใช้ข้อมูลจากตาราง WEEK จำนวน 360 Row แสดงผล TRAFFIC IN และ TRAFFIC OUT ทั้ง P2P และ NON P2P โดยจะมีการแสดงค่าสูงสุดของ Bandwidth ที่เกิดขึ้น ค่าเฉลี่ยของ Bandwidth ที่เกิดขึ้นย้อนหลังไป 7 วัน




**รูปที่ 5.9 ตัวอย่างการแสดงผล PREVIOUS 7 DAY IN**

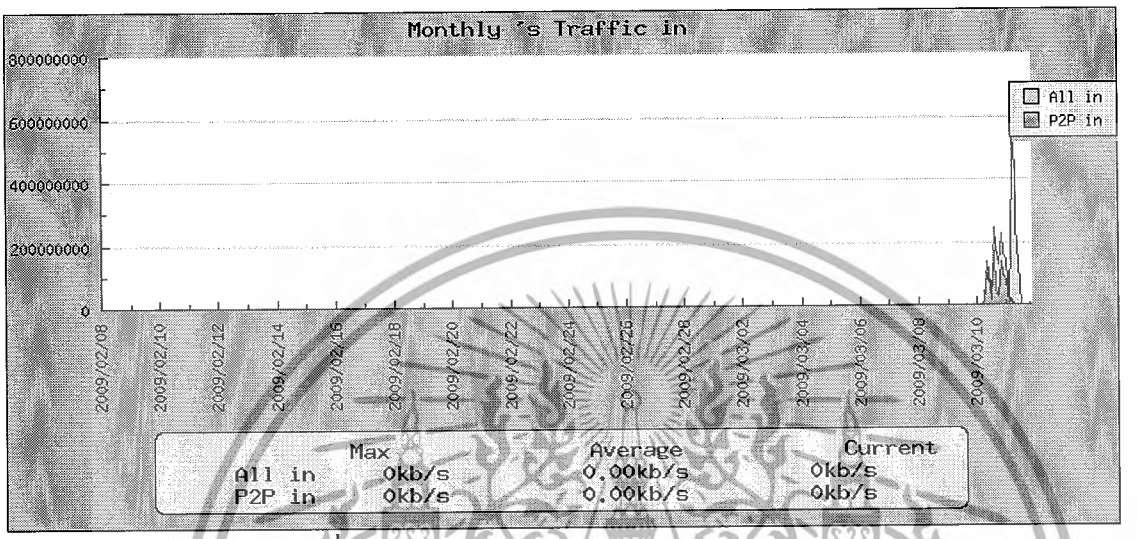



**รูปที่ 5.10 ตัวอย่างการแสดงผล PREVIOUS 7 DAY OUT**

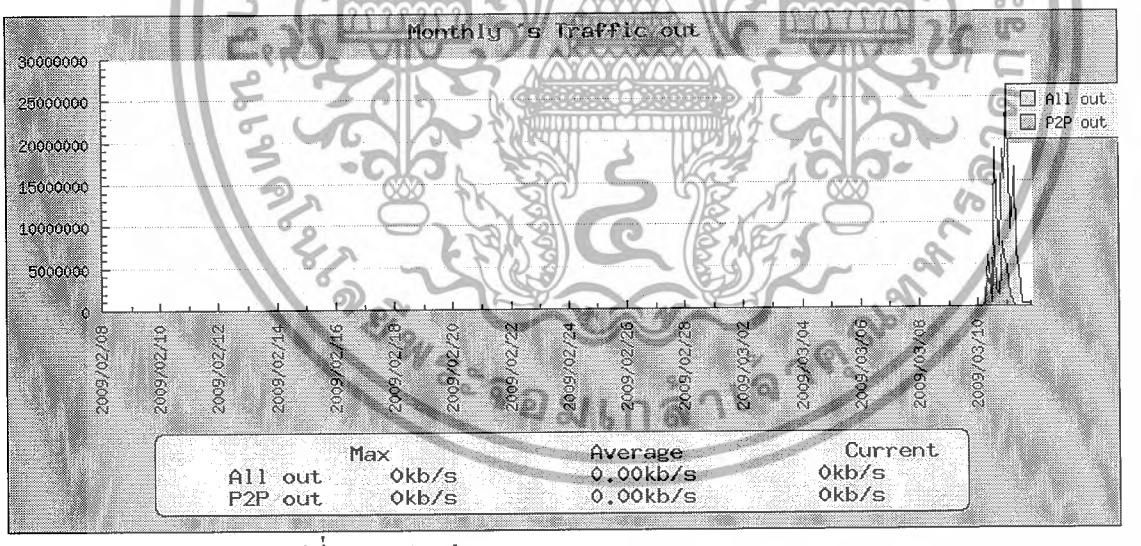
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.4 การแสดงผล PREVIOUS 30 DAY

การแสดงผล PREVIOUS 30 DAY ใช้ข้อมูลจากตาราง WEEK จำนวน 1530 Row แสดงผล TRAFFIC IN และ TRAFFIC OUT ทั้ง P2P และ NON P2P โดยจะมีการแสดงค่าสูงสุดของ Bandwidth ที่เกิดขึ้น ค่าเฉลี่ยของ Bandwidth ที่เกิดขึ้นย้อนหลัง30 วัน




**รูปที่ 5.11 ตัวอย่างการแสดงผล PREVIOUS 30 DAY IN**

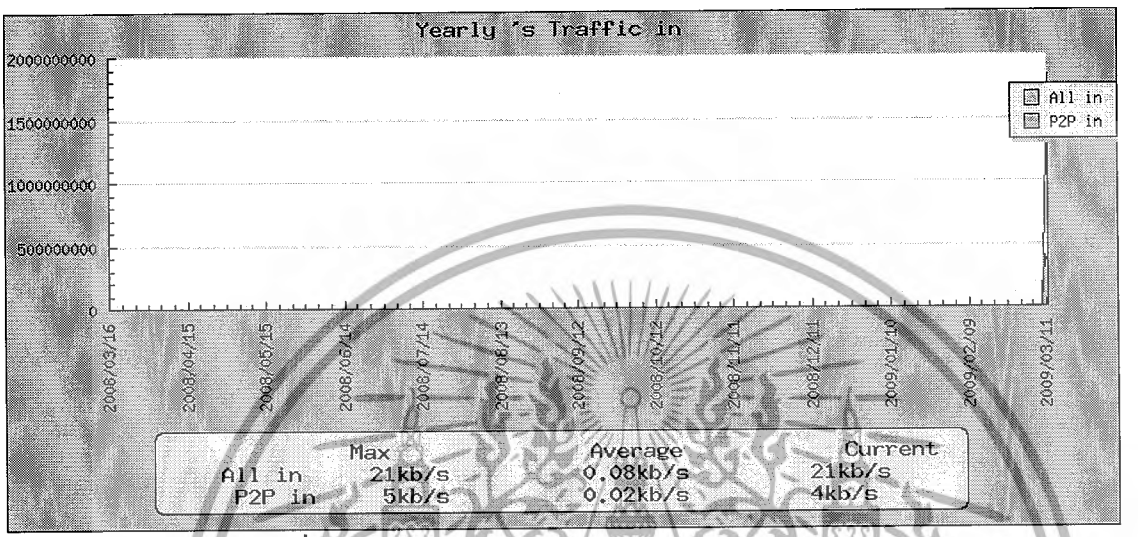



**รูปที่ 5.12 ตัวอย่างการแสดงผล PREVIOUS 30 DAY OUT**

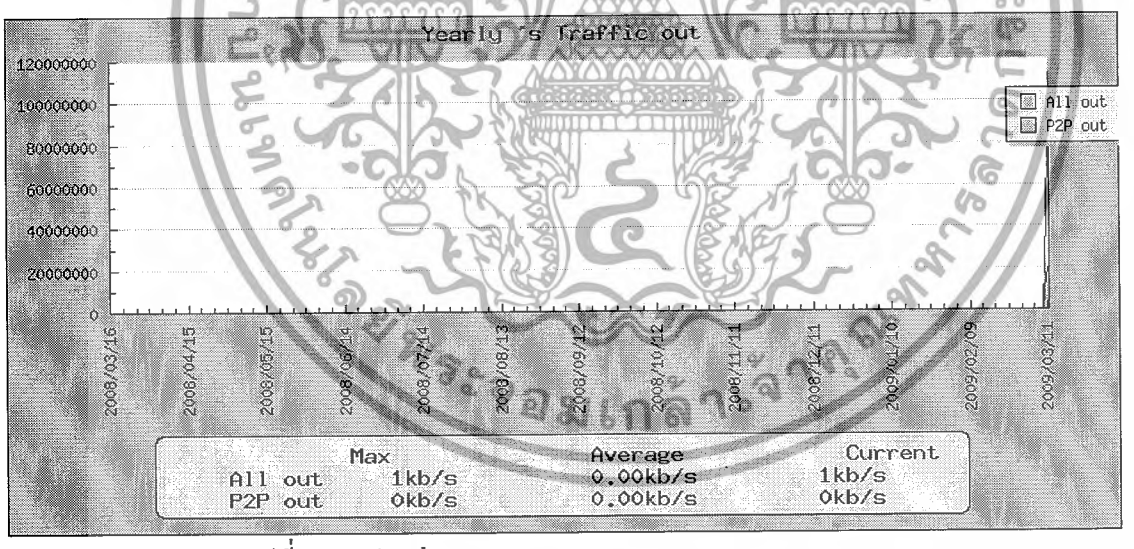
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.7.5 การแสดงผล PREVIOUS 12 MONTHS

การแสดงผล PREVIOUS 12 MONTHS ใช้ข้อมูลจากตาราง WEEK จำนวน 188615 Row แสดงผล TRAFFIC IN และ TRAFFIC OUT ทั้ง P2P และ NON P2P โดยจะมีการแสดงค่าสูงสุดของ Bandwidth ที่เกิดขึ้น ค่าเฉลี่ยของ ที่เกิดขึ้นย้อนหลัง12 เดือน




**รูปที่ 5.13 ตัวอย่างการแสดงผล PREVIOUS 12 MONTHS IN**



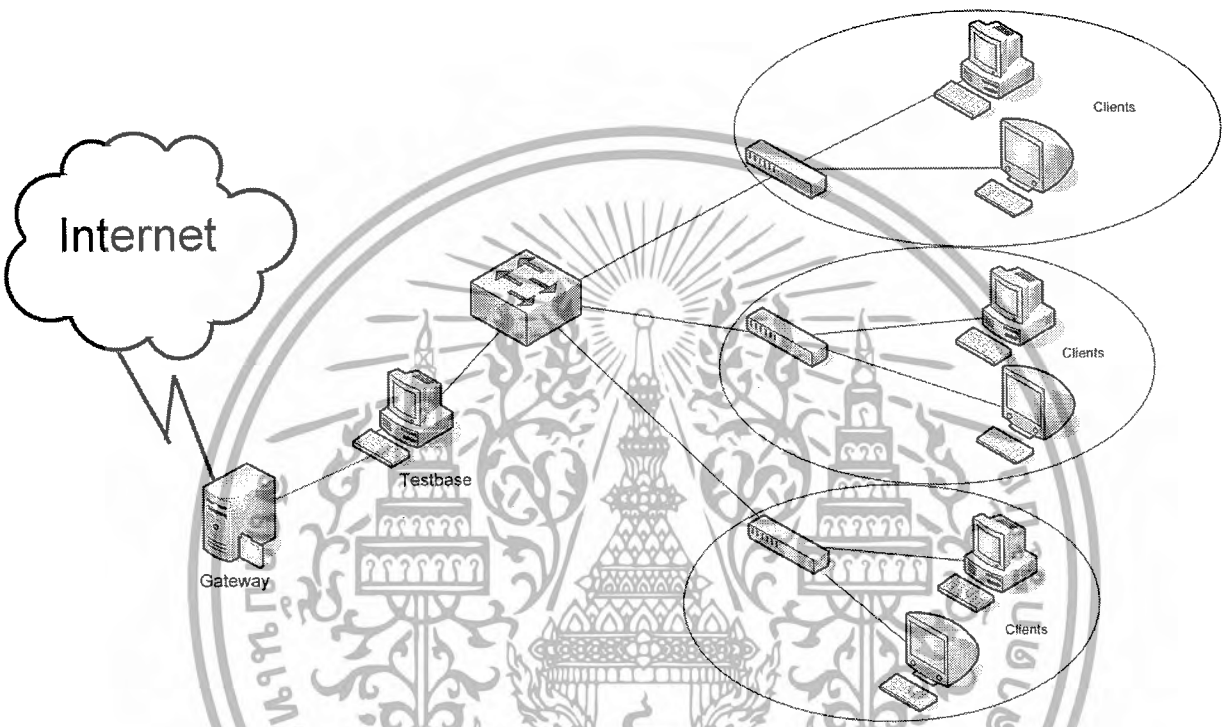

**รูปที่ 5.14 ตัวอย่างการแสดงผล PREVIOUS 12 MONTHS OUT**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6

# การทดลองและผลการทดลอง

## 6.1 การตั้งอุปกรณ์ดักจับ Packet

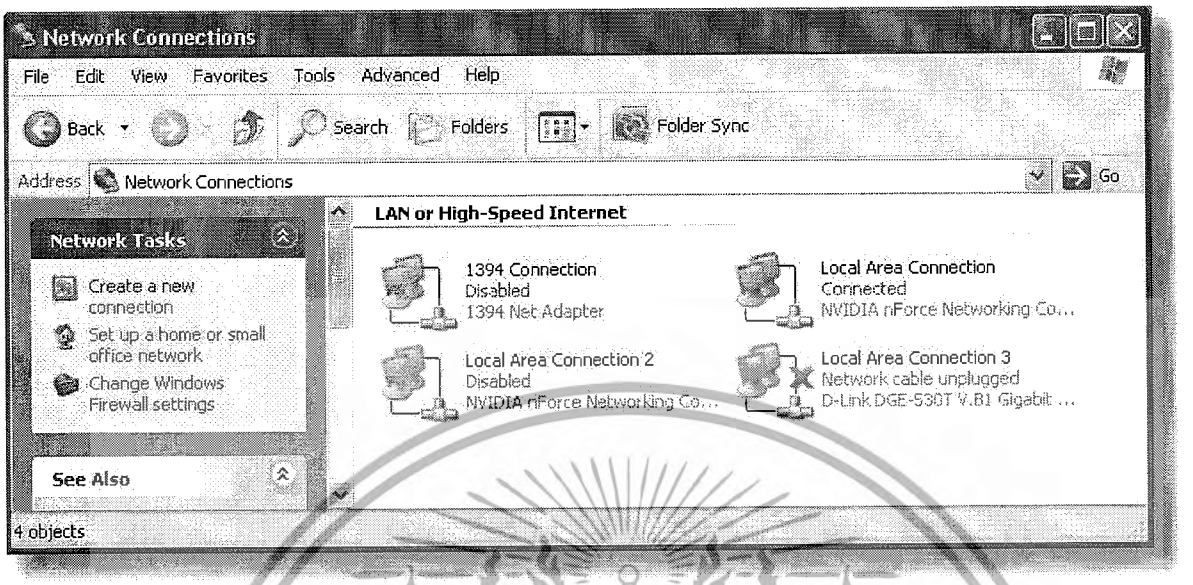



**รูปที่ 6.1 Test Base Model**

ทดลองดักจับ Packet ด้านหลัง Gateway โดยคอมพิวเตอร์ที่เป็น Test Base นั้นจะทำหน้าที่เป็น Bridge ซึ่งจะต้องใช้การ์ด LAN สองตัว เครื่องคอมพิวเตอร์ที่เป็น Test Base นั้นจะต่อขั้นระหว่าง Gateway และระบบเครือข่าย โดยมีวิธีการตั้งค่า Network คอมพิวเตอร์ให้เป็น Bridge ดังนี้

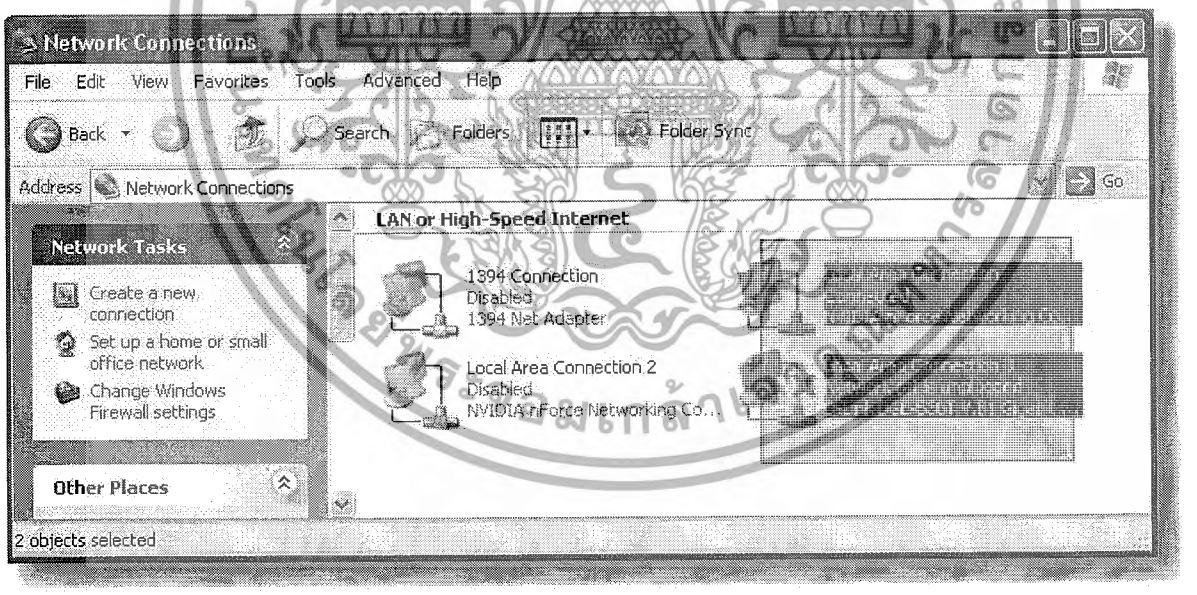
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ไปที่ Control Panel -> Network Connection




**รูปที่ 6.2 การติดตั้ง Bridge Connection 1**

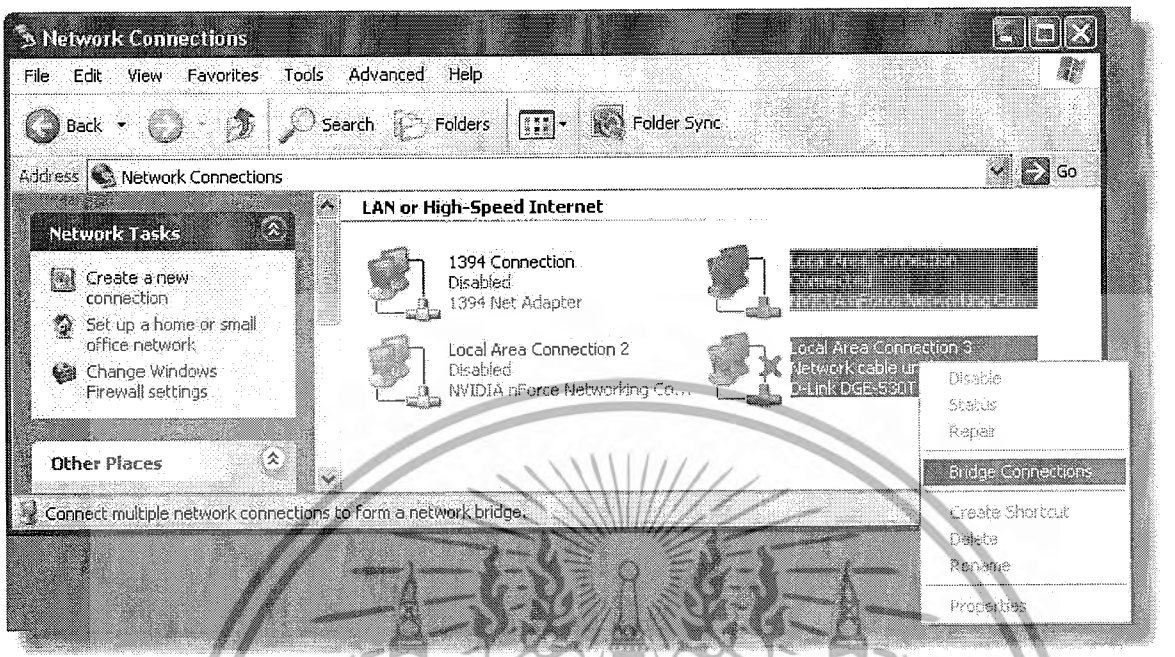
2. ลากเมาส์คลุม Eternet Cardที่จะใช้ทำ Bride




**รูปที่ 6.3 การติดตั้ง Bridge Connection 2**

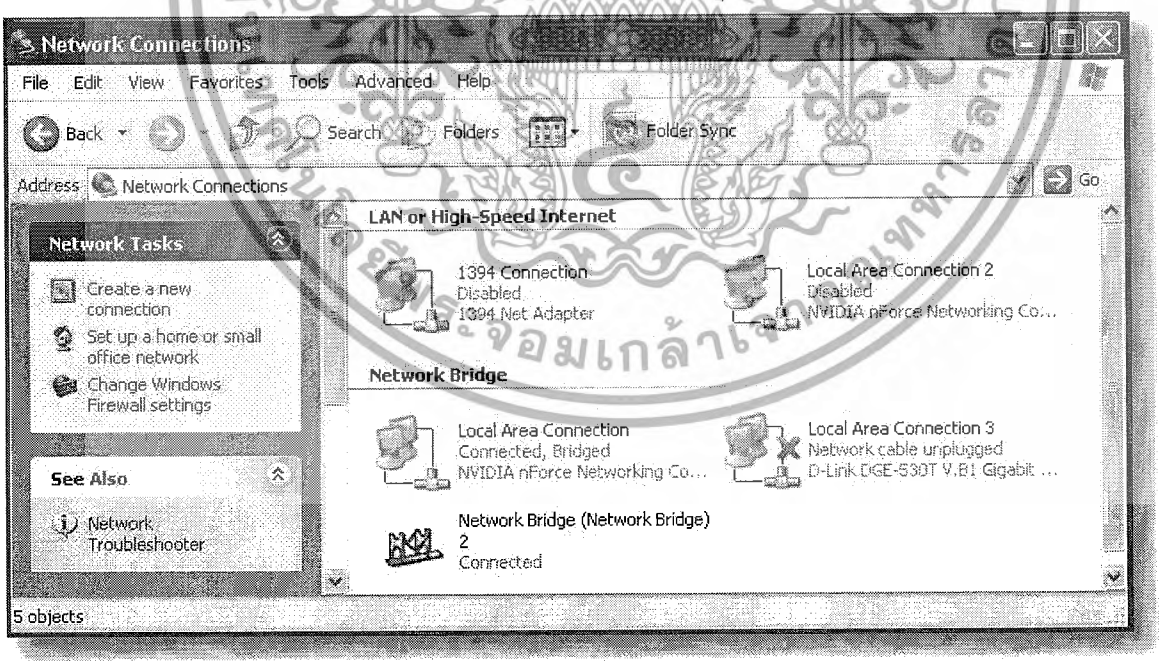
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกขวาเลือก Bridge Connections




**รูปที่6.4 การติดตั้ง Bridge Connection 3**

4. จะได้Network Bridgeดังรูปที่ 5.5 จากนั้นจึงตั้งค่า IP Adress ตามปกติ




**รูปที่ 6.5 การติดตั้ง Bridge Connection 4**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องที่ใช้ในการทดลอง

1. Laptop Fujistu

Cpu Pentium 900

Ram 256 Mbyte

2. Desktop

Cpu AMD X2 4000+

Ram 2Gbyte

3. Desktop

Cpu Pentium 4 2.66

Ram 1Gbyte

## 6.2 Capture Packet

จากรูปที่ 6.6 เป็นการรันโปรแกรมดักจับ Packet เพื่อนำไปสร้าง Parameter สำหรับการ Train โดยจะมีเมนูเพื่อให้เลือก Interface Card

**รูปที่ 6.6 การ Run Program Capture Packet**

ในการทดลองเราใช้งานเครื่องภายในระบบ โดยทำงานในรูปแบบต่างๆ เช่น Bit torrent, HTTP, FTP, VDO Stream, Audio Stream, MSN ฯลฯ

## 6.3 Parameter ที่ได้จากการทำงาน

หลังจากการทำงานเสร็จสิ้นแล้วโปรแกรมจะเขียน Parameter ทั้งหมดที่จะนำไปใช้งานต่อลงใน Fileซึ่งแบ่งข้อมูลออกเป็น block แต่ละ block มีขนาด 128 byte จากการทดลองเป็นเวลา 4 ชั่วโมง เราได้ File ขนาด ประมาณ 20 MByte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.4 Neural Network Training

ในการ Train เราใช้ข้อมูลจาก File block ละ 128 Byte ในการ Training

จากรูปที่ 6.7 เป็นตัวอย่างผลของการTrain โดยที่

- Epoch : จะบอกรอบที่ทำการTraining
- TSet ACC : คือความถูกต้องของOutput ขณะ Training
- MSE : ค่าเฉลี่ยของoutput node ในการ Train แต่ละรอบ
- GSet ACC : เป็นเปอร์เซ็นความถูกต้องของการทดสอบแยกแยะแพ็กเกจจริงๆที่ไม่ใช่การTrain

**รูปที่ 6.7 หน้าจอเมื่อ Train เสร็จสิ้น**

ผลที่ได้จากการเทรนครั้งนี้คือได้ความถูกต้องในการแยกแยะที่เราดักจับไว้ 93.876% เนื่องจากเราทำการตั้งค่าไว้ให้หยุดหาก TSet ACC และ GSet ACC มีค่ามากกว่า 90% และการ Train รอบแรกจะทำการบันทึก Weights อัตโนมัติและหารอบต่อไป Train ได้ TSet ACC และ GSet ACC สูงกว่าเดิมก็จะบันทึกค่า Weights ที่ดีกว่าทับลงไป หลังจากการทดลองเราได้ File ที่บันทึกค่า Weights ขนาดประมาณ 27 Kbyte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.5 Traffic Classification

หลังจากได้ค่า weight ที่เหมาะสมแล้ว เรานำค่า weight ที่ได้ไปใช้ เราส่ง INPUT ให้ Neural Network ทำการแยกประเภท การทำงานเป็นแบบ Real Time OUTPUT ที่ได้ถูกเขียนลงใน DATA BASE จากการทดลอง ใช้งานเครื่องที่อยู่ในระบบ ด้วยการทำงานประเภทต่างๆ เช่น Bit torrent, HTTP, FTP, VDO Stream, Audio Stream, MSN ฯลฯ

การทดลองรูปแบบที่ 1

เครื่องที่ 1 ทำงาน Web site และ MSN

เครื่องที่ 2 ทำงาน Web site และ MSN

เครื่องที่ 3 ทำงาน Web site และ MSN

ผลที่ได้สามารถระบุได้ว่า Traffic ที่เกิดขึ้น เป็นชนิด Non P2P โดยมีค่าความถูกต้องอยู่ที่ประมาณ 80 เปอร์เซ็นต์

การทดลองรูปแบบที่ 2

เครื่องที่ 1 ทำงาน Download bit torrent

เครื่องที่ 2 ทำงาน Download bit torrent

เครื่องที่ 3 ทำงาน Download bit torrent

ผลที่ได้สามารถระบุได้ว่า Traffic ที่เกิดขึ้น เป็นชนิด P2P โดยมีค่าความถูกต้องอยู่ที่ประมาณ 80 เปอร์เซ็นต์

การทดลองรูปแบบที่ 3

เครื่องที่ 1 ทำงาน Web site และ MSN

เครื่องที่ 2 ทำงาน Download bit torrent

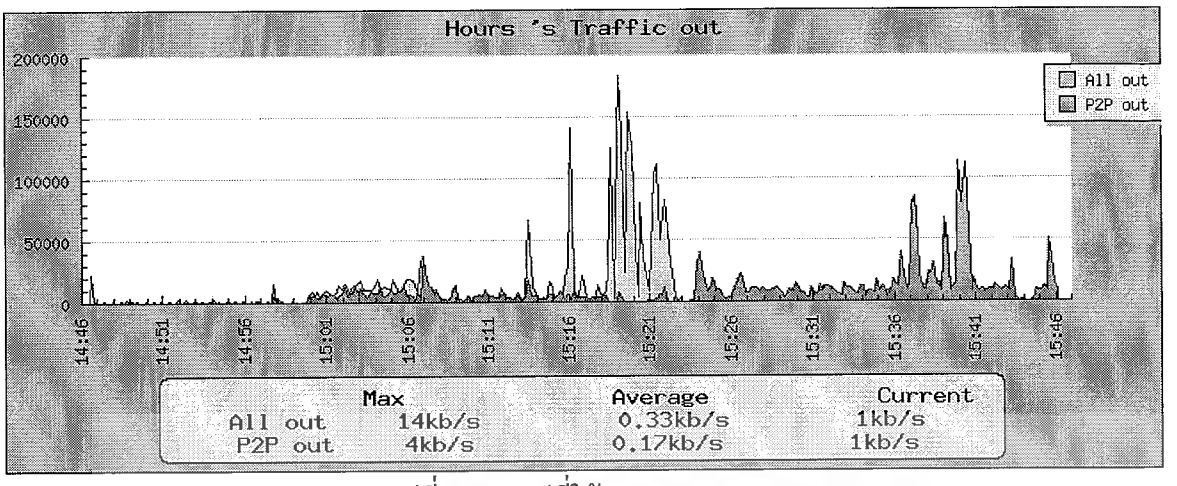
เครื่องที่ 3 ทำงาน Download bit torrent, download ไฟล์จาก FTP, เปิด Web site

ผลที่ได้สามารถระบุได้ว่า Traffic ที่เกิดขึ้น เป็นชนิด P2P และ Non P2P โดยมีค่าความถูกต้องอยู่ที่ประมาณ 63 เปอร์เซ็นต์

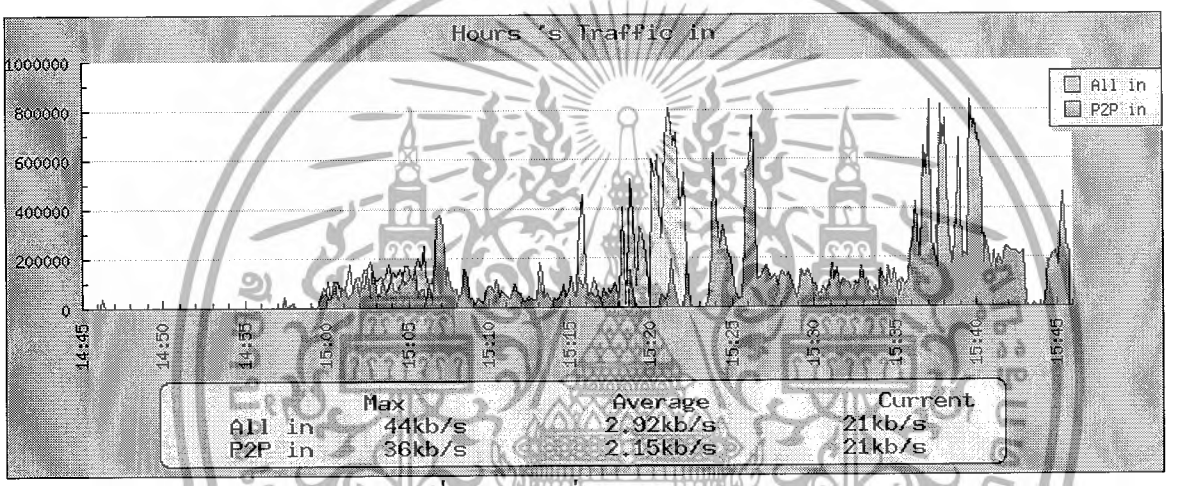
## 6.6 ผลการทดลอง

กราฟที่ได้จากการทดลองเป็นกราฟที่แสดงผลของ Traffic ใน 1 ชั่วโมงสามารถ refresh เพื่อดู Traffic ปัจจุบันได้ โดยกราฟจะแยก ประเภทของ Traffic ออกเป็นสองชนิด คือแบบ P2P และ All Traffic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.8 กราฟที่ใช้แสดงผล Traffic out**




**รูปที่ 6.9 กราฟที่ใช้แสดงผล Traffic in**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7

# บทสรุป

## 7.1 บทสรุป

จากการศึกษาและทดลอง ทำให้เข้าใจการทำงานของโปรโตคอลที่ใช้ติดต่อสื่อสารในระบบเน็ตเวิร์ค สามารถพัฒนาโปรแกรมดักจับข้อมูลที่อยู่ใน Buffer ของ Network Card ได้เห็นรูปแบบของ Frame ข้อมูลที่ถูกส่งในระบบเน็ตเวิร์ค รวมถึงเข้าใจหลักการทำงานของ Neural network สามารถใช้Neural network Train หาค่า Weights และนำค่า Weights ที่ได้ไปแยกประเภทของ Traffic ที่ใช้งานจริงบนระบบ Network ว่าเป็น P2P หรือ Non P2P ได้

## 7.2 ปัญหาอุปสรรคและแนวทางแก้ไข

- ความผิดพลาด ในขั้นตอนการเขียนข้อมูลลงใน File ด้วยค่า 0D ระบบจะเกิดค่า 0A ขึ้นต่อท้ายโดยอัตโนมัติ *การแก้ไข* ต้องใช้ HEX EDITER ช่วยลบค่า 0A ซึ่งต่อท้าย 0D ทิ้งไป
- ความผิดพลาด ในการอ่านข้อมูลจาก File ด้วย Library File Stream คือ Function return ค่ากลับแล้วว่าจบ File แต่เนื้อหาที่ได้มายังไม่ครบถ้วน *การแก้ไข* เปลี่ยน Library ที่ใช้อ่าน File ใหม่เป็น system.dll
- ในเทอมแรกวิธีการป้อน Input ให้กับ Neuron Network ยังไม่เหมาะสม เราใช้ข้อมูลดิบที่ได้จากการ Capture Packet เป็น Input ซึ่งทำให้ Input มีขนาดใหญ่และผลที่ได้ก็มีความถูกต้องน้อย *การแก้ไข* เราเปลี่ยนมาใช้วิธีการหาค่า Parameter ที่จำเป็นจาก Packet ที่ Capture เข้ามาเพื่อใช้เป็น Input ให้กับ Neuron Network ผลที่ได้คือพื้นที่ในการเก็บข้อมูลลดลง การทำงานเร็วขึ้น และมีความถูกต้องมากขึ้นอีกด้วย
- ความผิดพลาดจากการประมวลผลที่เกี่ยวข้องกับตัวแปรชนิด Double ในการเรียกใช้งานหรือนำค่าไปประมวลผลด้วยตัวแปรชนิด Double บางครั้งเกิดการ error ที่ข้อมูลในตัวแปรและให้ค่าที่ผิดพลาดกลับมาทั้งๆที่คำสั่งและรูปแบบไวยากรณ์ ถูกต้องตามปกติ *การแก้ไข* หลังจากการทดลองแก้ไขอยู่นานได้พบว่าหากเรียกใช้คำสั่งหรือ Function แบบเดิมซ้ำอีกครั้งจะไม่เกิด error ขึ้นทั้งๆที่เป็นการทำงานแบบเดียวกันและอยู่ในตำแหน่งที่ติดกัน
- ความสามารถของเครื่องที่ใช้ทดลองยังมีประสิทธิภาพไม่พอในการประมวลผล Connection จำนวนมากๆ และSoftware ยังไม่ Optimize พอ *การแก้ไข* ปรับปรุง Software ให้ทำงานแบบ multi thread ทำการ Optimize Functionvต่างๆให้มีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.3 แนวทางการพัฒนาต่อ

- Support ให้สามารถรับการทำงานกับจำนวน Connection ในปริมาณมากๆได้
- พัฒนาให้สามารถระบุชนิดของ โปรโตคอลแต่ละตัวได้
- นำไปประยุกต์ใช้ในงานด้านต่างๆ เช่น การจัดการบริหารระบบ Network
- พัฒนาให้สามารถระบุ Traffic ที่เกิดขึ้นจากแต่ละ IP Address ได้
- พัฒนาให้สามารถระบุช่วงเวลาที่ต้องการดูข้อมูลให้มีความหลากหลายมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

นิรุธ อำนวยศิลป์. 2548. Network And protocols Programming using C/C++ .กรุงเทพฯ: เจริญการพิมพ์.

ไพศาล โมลิสกุลมงคล. 2538. พัฒนา Web database ด้วย PHP. กรุงเทพฯ : ไทยเจริญการพิมพ์.

ศุภมิตร จิตตะยโสธร. 2536. DataBase Design & SQL. กรุงเทพฯ : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Subhabrata Sen, Analyzing peer-to-peer traffic across large networks AT&T Labs–Research 180 Park Ave., Florham Park, NJ

Aditus Consulting [Online]. 2000-2007.
Available: http://www.aditus.nu/jpgraph/.

Bearshare[Online].
Available: http://www.bearshare.com.

Direct Connect [Online].
Available: http://www.neo-modus.com.

Gnutella hosts [Online].
Available: http://www.gnutellahosts.com.

Grokster[Online].
Available: http://www.grokster.com.

InSecure [Online]. 2000-2007.
Available: http://nmap.org/book/tcpip-ref.html/.

KaZaA[Online].
Available: http://www.kazaa.com.

Limewire[Online].
Available: http://www.limewire.com.

Morpheus[Online].
Available: http://www.musiccity.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้