

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การตรวจสอบสถานะของเครือข่ายระยะไกลผ่านอินเทอร์เน็ต (เวิร์ลไวด์เว็บ)

REMOTE NETWORK MONITORING ON INTERNET (WWW)



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่.....ปีการศึกษา 2541  
เลขที่ฉบับนี้.....  
วันที่.....เดือน.....ปี.....  
ฉบับนี้ให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจสอบสถานะของเครือข่ายระยะไกลผ่านอินเทอร์เน็ต (เวิร์ลไวด์เว็บ)

REMOTE NETWORK MORNITORING ON INTERNET (WWW)

ผู้จัดทำ

1. นายกิตติเทพ พิพัฒน์กุลชัย รหัส 39013229

2. นายชูศักดิ์ ภัทรคุชฎีกุล รหัส 39013235



อาจารย์ที่ปรึกษา

(อาจารย์รัชชัย สุทธิทศธรรม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การตรวจสอบสถานะของเครือข่ายระยะไกลผ่านอินเทอร์เน็ต (วีรล์ไวด์เว็บ)

นายกิตติเทพ พิพัฒน์กุลชัย

นายชูศักดิ์ ภัทรคุณภูิกุล

อาจารย์รัชชัย สุทธิพิศธรรม อาจารย์ที่ปรึกษา

ปีการศึกษา 2541

### บทคัดย่อ

ในปัจจุบันการตรวจสอบสถานะของระบบเครือข่าย เช่นการแสดงผลรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับระบบ การเชื่อมต่อของอุปกรณ์ต่าง ๆ จำนวนข้อมูลของความผิดพลาด และชนิดของแพ็กเกจ ตารางการหาเส้นทาง และอีกหลาย ๆ อย่างที่ใช้ในการจัดการระบบเครือข่าย ซึ่งกระทำได้โดยต้องมีเครื่อง management station ในเครื่องนี้จะมีโปรแกรมที่ช่วยให้ผู้ใช้ทำการจัดการส่วนต่าง ๆ ที่เกี่ยวข้องกับระบบ และเครื่องที่เป็น agent (ต้องมีการติดตั้งโปรแกรม agent ก่อน) สำหรับการจัดการของเครื่อง manager เครื่องที่ไม่เป็นเครื่อง manager นั้น ไม่สามารถมองเห็นสถานะภาพและสถานะของระบบเครือข่ายของตนได้เลย

โครงการนี้ทำการตรวจสอบสถานะของระบบเครือข่าย โดยใช้ภาษาจาวาในการนำเสนอ โดยการเขียนโปรแกรมภาษาจาวาขึ้นมาเพื่อติดต่อกับ MIB แล้วนำข้อมูลที่ได้มาเก็บไว้ในฐานข้อมูล จากนั้นเขียนโปรแกรมอีกตัวขึ้นมาเพื่อติดต่อกับฐานข้อมูลแล้วนำเสนอออกทางเว็บไซต์ (www) เพื่อที่จะให้เครื่องที่ไม่เป็น manager ก็สามารคมองเห็นสถานะของระบบเครือข่ายของตนได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Remote Network Monitoring on INTERNET (WWW)

Kittithep Pipattanakulchai

Chusak Pattaradusadekul

Tawatchai Suttitosatum Advisor

Year of Academic 1998

### ABSTRACT

Recently, When administrator want to monitor network such as serveral description on network, interface of devices, number of error packets, type of packets and so on for management network. He can get its by management station. This machine is contained software tool for administrator managing network. Exectly, agent machine (installed agent program already) is always respond data if management station request. Many machine which not management station can't monitor your network

This project is monitor network. First, created java language for request to Management Information Base (MIB). Then, receive data from MIB and keep its to database. Created another java language for query data from database and present it on web site (WWW). Finally, many machine which not management station can monitor your network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กิตติกรรมประกาศ

วิทยานิพนธ์นี้ เสร็จสมบูรณ์ได้จากความช่วยเหลือ และความร่วมมือของทุกคน เริ่มจากอาจารย์รัชชัย สุทธิทศธรรม อาจารย์ที่ปรึกษาที่คอยให้คำแนะนำ ควบคุม เอาใจใส่ และช่วยเหลือในทุก ๆ ส่วนในวิทยานิพนธ์ชิ้นนี้ จึงขอขอบพระคุณ

ขอขอบพระคุณบุคคลที่ให้ข้าพเจ้าได้มีโอกาสมาถึงจุดนี้ บิดา มารดา อันเป็นที่เคารพรักรยิ่ง ให้กำลังใจ และคอยดูแลเอาใจใส่ข้าพเจ้าเสมอมา ผู้ทำวิทยานิพนธ์นี้ร่วมกับข้าพเจ้า ที่ไม่ทอดทิ้ง เคียงบ่าเคียงไหล่ฝ่าอุปสรรคมาด้วยกัน และเพื่อนอีกหลาย ๆ คนที่คอยให้คำปรึกษาแก้ไขวิกฤตการณ์ครั้งแล้วครั้งเล่า จนวิทยานิพนธ์นี้สำเร็จได้ด้วยดี ข้าพเจ้าดีใจ ข้าพเจ้าจะรำลึกน้ำใจของทุก ๆ คนนี้ไว้ตลอดไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	2
1.2 ขอบเขตการทำงาน	2
1.3 วิธีการดำเนินงาน	2
บทที่ 2 Simple Network Management Protocol	3
2.1 สถาปัตยกรรมของการบริหารระบบเครือข่าย	3
2.2 สถาปัตยกรรมของ SNMP	4
2.3 Trap-Directed Polling	5
2.4 Proxies	5
2.5 Operations Supported by SNMP	6
2.6 รูปแบบของ SNMP	8
2.7 ลักษณะของ PDU	10
2.8 ข้อจำกัดของ SNMP	13
2.9 การพัฒนา SNMP	14
บทที่ 3 Management Information Base	15
3.1 โครงสร้างของข้อมูลที่ใช้ในการบริหาร	15
3.2 Practical issues	19
3.3 มีมาตรฐาน	20
บทที่ 4 Remote Monitoring	23
4.1 Basic concept	24
4.2 Textual Conventions	24
4.3 The RMON MIB	27
4.4 Network Inventory	38
4.5 Hardware Platform	39
บทที่ 5 ภาษาจาวา	41
5.1 ความสามารถของภาษาจาวา	41
5.2 การประยุกต์ใช้งาน	42

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	5.3 Java Virtual Machine	43
	5.4 หลักการทำงานของภาษาจาวา	44
	5.5 Web Browser support Java	48
	5.6 Java Tool	49
	5.7 ความแตกต่างระหว่าง Java Applet และ Java Application	54
	5.8 ความแตกต่างระหว่าง Java และ Java Script	55
	5.9 ทิศทางของจาวา	55
บทที่	6 การใช้ภาษาจาวา	57
	6.1 ความหมายของ JDBC	59
	6.2 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC	59
	6.3 โครงสร้างของ JDBC	61
	6.4 รูปแบบของ JDBC Driver	62
	6.5 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล	69
บทที่	7 การสร้างและผลการทดลอง	72
	7.1 การนำเอาข้อมูลเก็บในฐานข้อมูล	72
	7.2 การนำเอาข้อมูลจากฐานข้อมูลแสดงบนเว็บเพจ	79
บทที่	8 บทวิจารณ์และสรุป	83
	8.1 สรุปผลการดำเนินงาน	83
	8.2 แนวทางการพัฒนาต่อ	83
หนังสืออ้างอิง		84
ภาคผนวก		85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

2.1 ตารางความหมายของ error ใน SNMP	9
3.1 ตารางอุปกรณ์ใน Layer ของ OSI	21
4.1 ตารางควบคุม	26
4.2 ตารางข้อมูล	26
5.1 ตารางเปรียบเทียบเครื่องมือในการพัฒนาภาษาจาวา	54
5.2 ตารางเปรียบเทียบ Java Script และ Java	55
6.1 ตารางเปรียบเทียบ JDBC Driver	71



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ

2.1 message ของ SNMP	4
3.1 โครงสร้างของ MIB	16
4.1 RMON MIB	27
4.2 RMON statistics Group	28
4.3 RMON history Group	29
4.4 RMON Alarm Group	30
4.5 ตัวอย่าง RMON Configuration	31
4.6 RMON host Group	31
4.7 RMON hostTopN Group	32
4.8 RMON matrix Group	33
4.9 RMON Filter Group	34
4.10 RMON Packet capture Group	36
4.11 RMON Event Group	37
5.1 การประมวลผลของโปรแกรมทั่วไปกับระบบปฏิบัติการที่ต่างกัน	45
5.2 การประมวลผลของโปรแกรมภาษาจาวา	45
5.3 โปรแกรม Microsoft Visual J++	51
5.4 โปรแกรม Symantic Visual Café	52
6.1 Two-Tier Model	60
6.2 Three-Tier Model	61
6.3 ระดับการเชื่อมต่อของ JDBC API	61
6.4 JDBC driver types	62
6.5 การติดต่อ JDBC โดยรูปแบบที่ 1 JDBC-ODBC Bridge	64
6.6 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API, partly Java driver	65
6.7 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol,all-Java driver	
ยกตัวอย่างของ dbAnywhere, SequelLink	66
6.8 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol,all-Java driver	
ยกตัวอย่างของ WebLogic	67
6.9 การติดต่อ JDBC โดยรูปแบบที่ 4 Native-protocol,all-Java driver	68
7.1 การนำเอาข้อมูลจาก RMON MIB ไปเก็บใน DB	72
7.2 ข้อมูลที่ส่งไปขอ MIB	73
7.3 ค่าที่ได้จากฐานข้อมูล MIB	74
7.4 ผลลัพธ์แยกข้อมูลต่าง ๆ ที่ได้รับจากฐานข้อมูล MIB	77

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 ผลข้อมูลจาก MIB ที่เก็บในฐานข้อมูล MS SQL Server	78
7.6 เว็บเพจที่ใช้ในการมอนิเตอร์	79
7.7 เว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ MIB	80
7.8 เว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ RMON ในลักษณะกราฟแท่ง	81
7.9 เว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ RMON ในลักษณะกราฟวงกลม	82



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

โครงการนี้ได้แบ่งการทำงานออกเป็น 2 ส่วนด้วยกันคือ ส่วนติดต่อกับ MIB แล้วนำค่าข้อมูลที่  
ได้เก็บไว้ฐานข้อมูล อีกส่วนหนึ่ง นำค่าจากฐานข้อมูลไปแสดงผลที่เว็บไซต์โดยภาษาจาวา

ในส่วนแรก การที่จะติดต่อกับ MIB ได้นั้น เราจำเป็นต้องมีความรู้ และเข้าใจเกี่ยวกับเรื่องของ  
TCP/IP ว่าการส่งข้อมูลไประหว่างเครื่องต้นทางจนถึงเครื่องปลายทางจะมีขั้นตอนอย่างไรบ้าง เรื่องของ  
SNMP ก็คือการจัดการระบบเครือข่ายเป็นอย่างไร manager สามารถติดต่อกับ agent ได้ใน  
ลักษณะใด MIB และ RMON MIB คืออะไร เกี่ยวข้องกับ agent และจะช่วยให้ SNMP จัดการระบบเครือ  
ข่ายได้อย่างไร Winsock ที่จะมามีบทบาทในส่วนการเขียนโปรแกรมติดต่อกับ SNMP เมื่อมีความรู้ในเรื่อง  
เหล่านี้ เราก็สามารถเขียนโปรแกรมขึ้นมาติดต่อกับโปรโตคอล SNMP ไปขอข้อมูลต่าง ๆ ที่อยู่ใน MIB  
และ RMON MIB นำมาเก็บในฐานข้อมูลได้

ในปัจจุบันระบบเครือข่ายที่มีความนิยมใช้กันอย่างกว้างขวาง และเป็นที่ยอมรับกันมาในขณะนี้คง  
จะหนีไม่พ้น ระบบเครือข่ายอินเทอร์เน็ต (internet) ซึ่งเป็นระบบเครือข่ายที่มีขนาดหรือความกว้างของ  
เครือข่ายใหญ่ที่สุดในโลกในขณะนี้ โดยจะมีการเชื่อมต่อเครือข่ายกันโดยผ่านสายโทรศัพท์ เป็นวิธีที่  
สะดวกที่สุดในการเชื่อมต่อเข้ากับระบบเครือข่ายนี้ และนิยมใช้กันมากทั่วโลก บนเครือข่ายอินเทอร์เน็ต  
นี้ได้มีการนำเสนอข้อมูลความรู้ ผลงานต่าง ๆ รวมถึงการแสวงหาผลกำไรได้อีกด้วยการโฆษณาต่าง  
ๆ มากมาย ทำให้บนเครือข่ายมีข้อมูลต่าง ๆ มากมายสามารถค้นหาได้ง่ายและสะดวกอีกด้วย เช่น เว็บ  
บราวเซอร์ (web browser) เป็นการนำเสนอข้อมูลข่าวสารต่าง ๆ ในรูปแบบของโฮมเพจ (home page) ,  
ไออาซี (IRC) เป็นบริการหนึ่งบนเครือข่ายอินเทอร์เน็ตใช้ในการติดต่อสื่อสารกันในระยะทางไกล, ทั่วโลก  
โดยการสนทนาในปัจจุบันที่นิยมใช้กันมากทำได้โดยการพิมพ์ข้อความ บริการเหล่านี้เป็นการให้บริการ  
บนเครือข่ายอินเทอร์เน็ต เป็นต้น

โครงการนี้ต้องการที่จะนำเสนอการมอนิเตอร์ (monitor) ต่าง ๆ โดยผ่านทางระบบเครือข่าย  
อินเทอร์เน็ตนี้ เพราะจะทำให้มีความสะดวกสบายในการใช้งานกว่าเนื่องจาก ซอฟต์แวร์ (software) ที่ใช้  
ในการตรวจสอบสถานะการทำงานของอุปกรณ์ต่าง ๆ ที่อยู่ภายในเครือข่าย จำเป็นจะต้องมีการติดตั้งไว้ที่  
เครื่องใดเครื่องหนึ่งที่อยู่ภายในเครือข่าย ขอบเขตของโครงการนี้จะเป็นการเรียนแบบการทำงานต่าง ๆ  
ของซอฟต์แวร์อันนี้แต่จะมีการเพิ่มเติมความสามารถในการทำงาน คือไม่จำเป็นต้องมีการติดตั้งตัว  
ซอฟต์แวร์ที่เครื่องสำหรับเครื่องที่จะทำการตรวจสอบสถานะการทำงานของอุปกรณ์ต่าง ๆ ภายในเครือ  
ข่าย โดยกระทำผ่านเครือข่ายอินเทอร์เน็ตซึ่งเป็นที่ทราบกันดีอยู่แล้วว่ามีเครือข่ายโยงใยทั่วโลก ดังนั้นแม้  
ว่าจะอยู่ที่ใดในโลกนี้ที่สามารถติดต่อกับเครือข่ายอินเทอร์เน็ต ก็สามารถที่จะตรวจสอบสถานะการ  
ทำงานต่างๆ ของเครือข่ายได้เช่นกัน การเพิ่มความสามารถเหล่านี้โดยการใช้บริการบราวเซอร์หรือที่เรียก  
กันโดยทั่วไปว่า “โฮมเพจ”

การสร้าง “โฮมเพจ” โดยทั่วไปแล้วเป็นที่สนใจของนักเล่นอินเทอร์เน็ตอยู่แล้ว ส่วนใหญ่ก็จะมุ่ง  
หวังไปที่ต้องการให้ผู้คนที่เข้ามาเยี่ยมชม การสร้างตัว “โฮมเพจ” นี้เฉพาะตัวบราวเซอร์เอง (   
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือที่รู้จักกันในภาษา “HTML” ) อยู่หลายอย่างเช่นกัน เช่นการสร้างภาพเคลื่อนไหว,การจัดรูปแบบเหล่านี้เป็นต้น ดังนั้นจึงได้มีผู้คิดค้นภาษาใหม่ขึ้นมาเพื่อเป็นตัวที่ใช้ในการสร้างตัว “โฮมเพจ” นี้ขึ้นมา

ได้แก่ “ภาษา จาวา” เป็นการรวบรวมเอาความสามารถต่าง ๆ จากโปรแกรมของภาษาอื่น ๆ เข้ามารวมไว้ด้วย ภาษาจาวาสามารถติดต่อระหว่างเครือข่ายได้อีกด้วย แต่โดยหลัก ๆ แล้วภาษานี้ต้องการสร้างขึ้นมานำมาใช้ในการเขียน หรือสร้าง “โฮมเพจ” เป็นหลัก ภาษาจาวาสามารถทำการติดต่อกับฐานข้อมูลได้โดยตรงหรืออาจจะมีการผ่านตัวจัดการฐานข้อมูลก็ได้เช่นเดียวกัน นี่ก็เป็นส่วนหนึ่งของความสามารถของภาษาจาวาเอง

วัตถุประสงค์ของโครงการนี้ก็เพื่อต้องการที่จะทำให้ตัวจาวาเองทำการติดต่อกับฐานข้อมูลที่สามารถจัดเก็บ สถานะสถานะการต่าง ๆ ของอุปกรณ์ภายในเครือข่าย เพื่อดึงเอาข้อมูลส่วนนี้มาแสดงผลที่โฮมเพจ

### 1.1 วัตถุประสงค์

- 1.สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ต่าง ๆ ภายในเครือข่ายได้
- 2.สามารถนำผลการตรวจสอบที่ได้มานำเสนอในรูปแบบของเว็บไซต์

### 1.2 ขอบเขตการทำงาน

- 1.ศึกษาการทำงานของ HP Openview ในการจัดการระบบเครือข่าย
- 2.ศึกษาการทำงานของ SNMP โพรโทคอล เพื่อติดต่อกับ RMON MIB แล้วเก็บในฐานข้อมูล
- 3.ศึกษาการทำงานของภาษาจาวาเพื่อติดต่อกับฐานข้อมูล
- 4.เขียน โปรแกรมภาษาจาวาติดต่อกับข้อมูลกับฐานข้อมูล MIB และ RMON
- 5.นำข้อมูลที่ได้เก็บในฐานข้อมูลที่สร้างขึ้น (MS SQL Server)
- 6.เขียน โปรแกรมภาษาจาวานำข้อมูลจากฐานข้อมูลนำเสนอบนเว็บไซต์

### 1.3 วิธีการดำเนินงาน

โครงการนี้จะต้องเริ่มศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับเครือข่าย การส่งผ่านข้อมูลในเครือข่าย การบริหารระบบเครือข่ายก่อน ซึ่งมีอยู่หลายเรื่อง ในวิทยานิพนธ์นี้จะอธิบายในเรื่องหลัก ๆ เท่านั้น บทที่ 2,3 และ 4 เป็นเรื่องเกี่ยวกับระบบเครือข่าย บทที่ 5 และ 6 เป็นเรื่องของภาษาและวิธีใช้ภาษานั้นติดต่อกับฐานข้อมูล บทที่ 7 ผลการทำงาน ซึ่งในบทที่ 2 เป็นเรื่องของ โพรโทคอลที่ใช้ในการส่งผ่านไปขอข้อมูลจากฐานข้อมูล MIB มีวิธีการหลายวิธี ในโปรแกรมที่เขียนขึ้นนี้ใช้วิธี Get request เท่านั้น ในบทที่ 3 และ 4 เรื่องของฐานข้อมูล MIB และ RMON ที่มีข้อมูลเกี่ยวกับสถานะของเครื่องนั้น ๆ

บทที่ 5 เป็นความรู้เบื้องต้นในเรื่องของภาษาจาวา และบทที่ 6 เป็นภาษาจาวาในการติดต่อกับฐานข้อมูลทั้งการนำข้อมูลเข้าไปเก็บและการสอบถามนำข้อมูลออกมาแสดงบนเว็บ รวมถึงการใช้คำสั่งภาษา SQL ทั้ง Insert, Delete และ Update และบทที่ 7 จะเป็นผลของการทำงานที่โครงการนี้ได้ทำมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### Simple Network Management Protocol (SNMP)

ในการดูแลจัดการเครือข่ายที่มีขนาดใหญ่ มีความซับซ้อน ซึ่งประกอบไปด้วยอุปกรณ์หลากหลายชนิดนั้นไม่สามารถกระทำโดยมนุษย์ตามลำพังได้ แต่จะต้องใช้เครื่องมือสำหรับบริหารเครือข่าย (Network Management Tool) ต่าง ๆ มาช่วยในการดูแลจัดการ และ Simple Network Management Protocol (SNMP) ก็เป็นเครื่องมือในการบริหารเครือข่ายที่มีลักษณะดังกล่าวข้างต้น

Simple network management protocol (SNMP) ถูกพัฒนาโดยยึดหลักความง่ายและเป็นเครื่องมือที่ถูกสร้างมาในยุคต้นๆ โดยใช้ทำงานกับ TCP/IP ซึ่งเป็น Protocol ที่มีการใช้งานกันอย่างแพร่หลายกันมากโดย SNMP ได้ถูกกำหนดให้เป็นมาตรฐานหนึ่งของ ISO (The International Organization for Standardization) และ CITT (The International Telegraph and Telephone Consultative Committee)

#### 2.1 สถาปัตยกรรมของการบริหารระบบเครือข่าย

model ของ network management ที่ใช้สำหรับระบบเครือข่าย TCP/IP ประกอบด้วยส่วนต่างๆ ดังนี้

- Management station
- Management agent
- Management information base (MIB)
- Network-management protocol

ตัว management station เป็นเครื่องที่ Administrator หรือ network manager ใช้ในการบริหารระบบเครือข่าย (หรืออาจจะกล่าวว่าเป็นเครื่องที่ใช้ interface กับ user นั้นเอง) โดย management station อาจจะประกอบด้วยส่วนต่างๆหรือคุณสมบัติ เหล่านี้เป็นอย่างน้อย

- Management application (หรือเรียกสั้นๆว่า manager )สำหรับใช้วิเคราะห์ข้อมูล, ใช้ทำการ fault recovery หรืออื่นๆ เป็นต้น
- ส่วนของ Interface ทั้งที่เป็น monitor และ control ระบบเครือข่าย
- สามารถจะนำเอาข้อมูลจาก MIB ของอุปกรณ์ต่างๆ ในระบบเครือข่ายมาใช้งานหรือสั่งงานได้
- สามารถที่จะทำการตรวจสอบหรือควบคุมส่วนต่างๆ ในระบบเครือข่ายได้

ส่วนอุปกรณ์อื่นๆ ในระบบ network management เราเรียกว่า management agent เช่น host, bridge, router และ hub เป็นต้น โดย agent เหล่านี้สามารถถูกบริหารโดย management station หรือ manager โดยแต่ละ agent จะถูกมองเป็น object แต่ละ object จะต้องประกอบด้วยฐานข้อมูลภายในหรือ MIB ซึ่งในการตรวจสอบตัว manager จะมาอ่านข้อมูลใน MIB ส่วนการกำหนดค่าหรือการทำงานต่างๆ ก็ทำโดยการเปลี่ยนค่าหรือเซตค่าต่างๆ ใน MIB ของตัว agent ที่ต้องการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

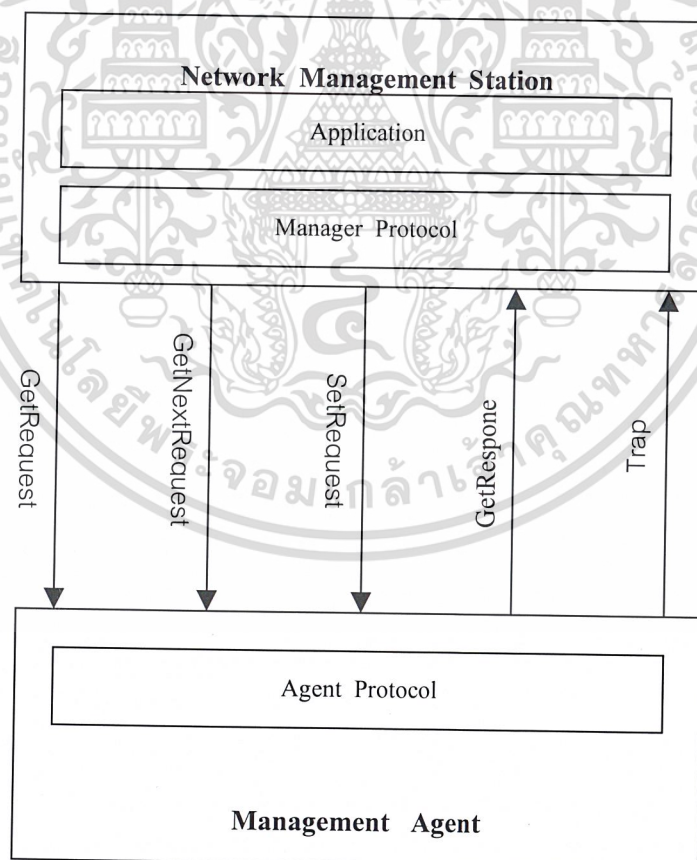
ตัว manager และ agent ถูกเชื่อมโยงโดย network-management protocol โดย protocol ที่ใช้ในการบริหาร network ของ TCP/IP คือ SNMP ซึ่งมีความสามารถดังนี้

- Get : อนุญาตให้ manager ทำการ retrieve ค่าของข้อมูลของ object ใน agent ได้
- Set : อนุญาตให้ manager ทำการเซตค่าข้อมูลของ object ใน agent ได้
- Trap : อนุญาตให้ agent ส่ง message ไปยัง Manager ในกรณีเกิดเหตุสำคัญ

## 2.2 สถาปัตยกรรมของ SNMP (Network Management Protocol Architecture)

SNMP ถูกออกแบบให้เป็น Protocol ในระดับ application level โดยจะทำงานบน user datagram protocol (UDP) ทำงานของ Protocol แบบ SNMP สำหรับ stand-alone management station โดย manager process control ทำการเข้าถึงข้อมูลส่วนกลาง (Central MIB) ที่ management station หรือ manager แล้วนำไปแสดงต่อ network manager โดย manager process ใช้ SNMP ซึ่งทำงานบน UDP และ IP และ Protocol ที่ใช้ในแต่ละระบบเครือข่าย (เช่น Ethernet, FDDI, X.25)

การติดต่อแลกเปลี่ยนข้อมูล (Message) ของ SNMP ระหว่าง management station กับ SNMP Agent โดยจะมี SNMP message อยู่ 5 ประเภทดังนี้



รูปที่ 2.1 แสดง message ของ SNMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Get Request เป็นข้อความที่ส่งโดย management application เพื่อร้องขออ่านข้อมูลของอุปกรณ์ต่างๆ จากโปรแกรม agent
- Get Next Request เป็นข้อความที่ส่งโดย management application เพื่อร้องขออ่านข้อมูลเช่นกันแต่ต่างกับแบบแรกที่เป็นการร้องขอข้อมูลเป็นชุด เช่น ขออ่านข้อมูลของตาราง เป็นต้น
- Get Response เป็นข้อความที่โปรแกรม agent ส่งค่าข้อมูลตอบกลับไปยังโปรแกรม management application หลังจากที่ได้อ่าน GetRequest หรือ GetNextRequest มา
- Set Request ตัว management application ส่ง message นี้เพื่อกำหนดค่าข้อมูลตัวแปรของ MIB
- Trap โปรแกรม agent ส่ง message นี้เพื่อรายงานเหตุการณ์หรือปัญหาที่เกิดขึ้นให้แก่ management application ทราบ

### 2.3 Trap-Directed Polling

ถ้า management station หรือ manager ต้องมีความรับผิดชอบ agent จำนวนมากๆ และในแต่ละ agent ยังประกอบไปด้วย object จำนวนมากด้วยแล้วยอมทำให้ management station ไม่สามารถที่จะทำการ poll เพื่อไปอ่านข้อมูลของ agent ทั้งหมดได้ ซึ่งจะต้องมีการนำเอาเทคนิคที่เรียกว่า Trap-Directed Polling มาใช้

โดยเทคนิคนี้จะอาศัยวิธีการโดยตัว management station จะทำการ poll ไปยัง agent เพื่อไปร้องขอเฉพาะข้อมูลที่ต้องการเท่านั้นเช่น คุณสมบัติของ interface, ข้อมูลค่าเฉลี่ยของจำนวน packet ที่ได้รับได้ส่งในช่วงเวลาหนึ่งๆ เป็นต้น ทั้งนี้ที่ baseline นี้ ถูกสร้างขึ้น management station จะหยุดการ poll จากนั้น agent แต่ละ agent จะทำหน้าที่ในการแจ้ง management station ให้ทราบถึงเหตุการณ์ผิดปกติต่างๆ ที่เกิดขึ้น

ทันทีที่ management station ได้รับ trap มันอาจจะเลือกที่จะทำ action บางอย่าง ณ จุดนี้ management station อาจจะส่ง poll ตรงไปยัง agent ที่ส่งรายงานเหตุการณ์ให้มันทราบโดยตรง และอาจจะส่งไปยัง agent อื่น ๆ ที่อยู่ใกล้เคียงด้วย เพื่อวินิจฉัยปัญหาที่เกิดขึ้นและขอข้อมูลเกี่ยวกับเหตุการณ์ที่เกิดขึ้นเพิ่มเติม

### 2.4 Proxies

ในการใช้ SNMP นั้นอุปกรณ์ในระบบเครือข่ายต่างๆ นั้นจะต้องสนับสนุนการใช้งาน Protocol แบบ SNMP ด้วยนอกจากนี้จะสนับสนุน Protocol แบบ UDP และ IP ด้วย ซึ่งอุปกรณ์บางตัวไม่มีการสนับสนุน protocol ดังกล่าวเช่น bridge และ modem บางตัว นอกจากนี้ยังมีระบบขนาดเล็กอีกเป็นจำนวนมากเช่น PCs, workstation, programmable controllers) เมื่อมีการนำมาใช้ในระบบเครือข่ายที่มีการใช้งานเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SNMP ก็จะเป็นอย่างยิ่งที่จะต้องมีส่วนที่เรียกว่า Proxy โดยการทำงานตัว management station จะทำการส่ง query ไปยังอุปกรณ์ที่เป็น proxy agent แทนการส่งไปยัง

agent นั้นตรงๆ โดยตัว proxy agent จะทำหน้าที่ในการแปลงแต่ละ query ที่ได้รับมาไปให้ agent อื่นที่หนึ่ง จากนั้นเมื่อ agent มีการส่งข้อมูลหรือ message ไปยัง management station ก็ต้องผ่านตัว proxy agent ก่อนเพื่อทำการแปลงก่อนเช่นกัน

## 2.5 Operations Supported by SNMP

ใน SNMP จะสนับสนุนการดำเนินการที่เป็น general-purpose operation ที่ทำงานกับ scalar Object คือ

1. Get: คือ management station ทำการ retrieve ค่าของ scalar-object จาก agent
2. Set: คือ management station ทำการ update ค่าของ scalar-object ใน agent
3. Trap: คือ agent ทำการส่งค่า scalar-object ให้กับ management station โดยที่ไม่ได้ทำการร้องขอ

### 2.5.1 Communities and Community Names

การใช้ SNMP network management ตัว application ที่เครื่อง management-station และที่ managed-station (agent) ต้องสนับสนุนโปรโตคอลแบบ SNMP

SNMP network management มีคุณสมบัติหลายอย่างที่ไม่อยู่ใน distributed application ทั่วไป รวมไปถึงคุณสมบัติความสัมพันธ์แบบ one-to-many relationship ระหว่างตัว management station และกลุ่มของ agent คือเครื่อง management station สามารถที่จะ get และ set object ใน agent และสามารถจะรับ trap จาก agent หลายตัวได้ ดังนั้นในการดำเนินการ operation หรือในการทำ control จะมอง management station โดยภายในจะมี agent เป็น subset โดย subset เหล่านี้สามารถจะ overlap กันได้ โดยใน agent จะมี MIB เป็นของตนเองซึ่งเราสามารถจะ control agent ต่างๆ โดยการใช้ management station มา set ค่าต่างๆ ของ MIB ของ agent นั้นๆ ซึ่งในการ control จะมีอยู่ 3 ลักษณะ

1. Authentication service: agent อาจจะต้องการให้มีการจำกัดการเข้าถึง MIB ของแต่ละ management station ไม่เหมือนกัน (authorized management station)
2. Access policy: agent อาจต้องการให้มีการกำหนดสิทธิ์ในการเข้าถึงที่แตกต่างกันของแต่ละ management station
3. Proxy service : เป็นการที่ agent มีการทำงานกับ proxy

จาก 3 ลักษณะการทำงานนี้จะสัมพันธ์กับ security concern ในภาวะแวดล้อมที่มีการแบ่ง network component ออกเป็นย่อยๆ เช่น การมี administrative หลาย entity agent ต้องการจะป้องกัน MIB ของตนเองจาก unwanted/unauthorized access

SNMP community คือความสัมพันธ์ระหว่าง SNMP agent และ set ของ SNMP manager ที่ได้กำหนด authentication, access-control และ proxy characteristics โดยหลักการของ community คือหนึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งกำหนดให้หนึ่ง agent โดย agent จะกำหนดหรือสร้าง 1 community สำหรับแต่ละความต้องการของการทำ authentication, access-control และ proxy characteristics

แต่ละ community จะมี community name ที่เป็น unique (ไม่ซ้ำกัน) ภายใน agent โดย management station จะได้รับ community name ในการดำเนินการ get และ set ตัว agent อาจจะมีการสร้างจำนวนของ community หลาย community ในกรณีที่เป็นการทำงานแบบ overlapping management station membership

ในการทำ community เป็นการทำภายใน agent ดังนั้น อาจจะมี community name ชื่อเดียวกันใน agent อื่นๆ ซึ่งในการกำหนดชื่อจะไม่เกี่ยวข้องหรือแสดงถึงการทำ community ดังนั้นตัว management station จะต้องทำการเก็บ trace ของ community name ของในแต่ละ agent ที่มีต้องการจะ access ด้วย

### 2.5.1.1 Authentication Service

Authentication Service จะเกี่ยวข้องกับการรับประกันว่า การสื่อสารหนึ่ง ๆ นั้นเชื่อถือได้ ในกรณีของ SNMP message ฟังก์ชันของ authentication service จะเป็นการทำให้ผู้รับมั่นใจว่า message นั้นมาจากแหล่งกำเนิดที่มันอ้างถึงจริง โดยมีวิธีการตามที่ได้นิยามไว้ใน RFC1157 ดังนั้นคือ ทุก ๆ message (get หรือ put request) จาก management station ไปยัง agent จะมี community name รวมอยู่ด้วย community name จะทำหน้าที่เหมือนเป็น password และ message จะถูกสมมติว่าเชื่อถือได้ถ้าผู้ส่งทราบ password

### 2.5.1.2 Access Policy

agent จะจำกัดการ access MIB ของมันไว้ให้กับเซตของ management station ที่ถูกเลือกโดยการนิยาม community ขึ้นมา community หนึ่ง agent สามารถกำหนดประเภทของการ access MIB ที่แตกต่างกันให้กับ management station ที่แตกต่างกันได้ การควบคุมการ access นี้มีอยู่ 2 ด้านคือ

2.5.1.2.1 SNMP MIB view : สับเซตของ object ต่าง ๆ ภายใน MIB แต่ละ community สามารถมี MIB view ที่แตกต่างกันได้ เซตของ object ในมุมมอง (view) หนึ่ง ไม่จำเป็นจะต้องเป็นของ subtree กิ่งเดียวของ MIB

2.5.1.2.2 SNMP access mode : คือสมาชิกของเซต {READ-ONLY, READ-WRITE} access mode จะถูกกำหนดให้สำหรับแต่ละ community

คอมบินชันของ MIB view และ access mode ถูกเรียกว่า SNMP community profile ดังนั้น community profile จะประกอบด้วยสับเซตของ MIB ซึ่งถูกนิยามที่ agent และ access mode สำหรับ object เหล่านั้น SNMP access mode จะถูกนำไปใช้กับ object ทั้งหมดที่อยู่ใน MIB view

community profile จะสัมพันธ์กับแต่ละ community ที่ถูกนิยามโดย agent คอมบินชันของ SNMP community และ SNMP community profile ถูกเรียกว่า SNMP access policy

### 2.5.1.2.3 Proxy Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

proxy คือ SNMP agent ซึ่งทำหน้าที่ในนามของอุปกรณ์อื่น ๆ ซึ่งไม่สนับสนุน TCP/IP และ SNMP ในบางกรณี (proxied device) และระบบการบริหารเครือข่าย (network management systems) ลงให้น้อยที่สุด

สำหรับแต่ละอุปกรณ์ที่ proxy system เป็นตัวแทนนั้นจะมี SNMP access policy เก็บไว้ ดังนั้น proxy จะทราบว่า MIB object ใดที่สามารถนำมาใช้ในการจัดการ proxied system (the MIB view) ได้ รวมทั้ง access mode ของ MIB object ด้วย

## 2.6 รูปแบบของ SNMP

โครงสร้างของข้อมูลในระดับบิตเป็นโครงสร้างซึ่งสามารถจะส่งลงบนบัสไปยังอุปกรณ์ปลายทางได้ โครงสร้างดังกล่าวมีชื่อว่า Type-Length-Value ประกอบด้วย 3 ส่วนย่อยดังนี้

1. Type field เป็นส่วนที่ถูกส่งไปเพื่อบอกชนิดของข้อมูลที่จะถูกส่งตามมา
2. Length field เป็น field ที่อยู่ต่อจาก Type field ใช้บอกจำนวน octet ที่อยู่ใน Value field
3. Value field ประกอบด้วย octet จำนวนตั้งแต่ศูนย์ขึ้นไป ซึ่งจะบรรจุข้อมูลชนิดต่างๆ เช่น integer , ASCII character หรือ OBJECT IDENTIFIER ไว้

### 2.6.1 ส่วนประกอบของ SNMP Protocol Data Units (PDUs)

Frame หรือหน่วยของข้อมูลซึ่งพร้อมจะถูกถ่ายโอนใน Network ถูกแสดงในรูปที่ 2.2 ประกอบด้วย 2 ส่วนดังนี้

2.6.1.1 Local Network Header และ Trailer จะถูกกำหนดขึ้นตามแต่นโยบายของ WAN หรือ LAN protocol

2.6.1.2 ส่วนที่เป็น data จริงที่จะถูกถ่ายโอนเรียกว่า internet protocol (IP) datagram ในกรณีที่ IP datagram มีขนาดยาวเกินกว่าจะใส่ไว้ใน 1 Frame จะต้องแบ่งการส่งเป็นหลาย Frame แล้วทยอยส่งไป

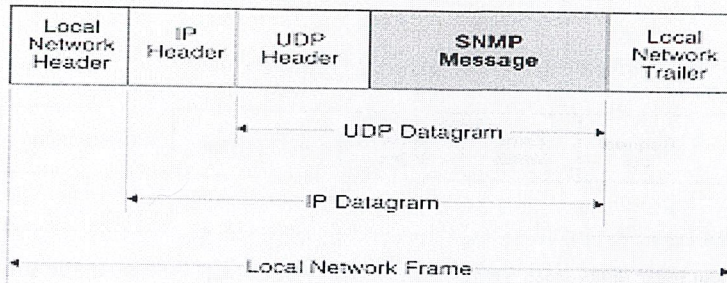
IP datagram ประกอบด้วย

2.6.1.2.1 IP address ระบุชื่อผู้รับ

2.6.1.2.2 user datagram protocol (UDP) header ใช้ตรวจสอบความผิดพลาดของข้อมูลที่อาจเกิดขึ้น ในระหว่างการส่งและใช้ระบุ protocol ที่จะใช้ในการประมวลผล datagram ใน layer ที่สูงขึ้นไป

2.6.1.2.3 SNMP message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงรูปแบบข้อมูลที่จะใช้ถ่ายโอนในเครือข่าย

### SNMP message มีส่วนย่อย 2 ส่วนคือ

1. Version + Community name บางครั้งเราเรียกส่วนนี้ว่า SNMP authentication header

1.1 Version number จะบอกเวอร์ชันของโปรโตคอล SNMP ที่ใช้อยู่ โดยถ้า message ที่ถูกส่งไปมาระหว่าง manager และ agent บรรจุ Version number ไม่ตรงกัน ก็จะไม่มีการประมวลผลใดๆเกิดขึ้นทั้งสิ้น

1.2 Community name จะเปรียบเสมือนเครื่องทดสอบ manager ว่าเป็นตัวจริงหรือไม่ โดยถ้า Community name ที่ส่งมาใน Frame ตรงกับที่เก็บอยู่ใน agent การประมวลผล SNMP PDU จะสามารถเริ่มขึ้นได้

2. PDU มีอยู่ 5 ชนิดคือ GetRequest, GetNextRequest, GetResponse, SetRequest และ Trap

#### รูปแบบของ Get ,Set และ Response PDU

1. PDU Type ใช้บอกชนิดของ PDU
2. Request ID ใช้บอกคู่ของการ request และ response ตัวอย่างเช่น ถ้า manager ส่ง RequestPDU ที่มี Request ID =3 ไป agent จะสนองตอบความต้องการของ Request อันนี้โดยส่ง Response PDU ที่มี Request ID=3 กลับมาเพื่อให้รู้ว่าการตอบสนองอันที่ส่งมาเป็นของการร้องขออันไหน
3. ค่าของส่วน Error Status มีความหมายดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 แสดงความหมายของ error ใน SNMP

Error	Value	Meaning
NoError	0	ไม่มีความผิดพลาดใดๆเกิดขึ้น
TooBig	1	ขนาดของ GetResponse ใหญ่เกินกว่าที่กำหนด
NoSuchName	2	ไม่มีชื่อของข้อมูลที่ต้องการใน MIB
BadValue	3	ส่วนของ Type Length และ Value ในส่วนของข้อมูลใน SetResponse ขัดแย้งกัน
ReadOnly	4	ไม่นิยาม
GenErr	5	ความผิดพลาดอื่นเกิดขึ้น

4. Error Index จะเป็นตัวบ่งชี้ว่าส่วนใดใน Variable Bindings ที่เป็นสาเหตุทำให้ error เกิดขึ้น

5. VarBindList ประกอบด้วย

5.1 Object จะบรรจุ object identifier ไว้เพื่อบอกว่าค่าที่ตามมานั้นเป็นค่าอะไร สำหรับ GetRequest และ GetNextRequest PDU ส่วน Value ใน list จะเป็น NULL

5.2 Value คือค่าของข้อมูลเช่น ค่า speed ของการถ่ายโอนข้อมูล เป็นต้น สำหรับ GetRequest และ GetNextRequest PDU ส่วนของ Value จะมีค่าเป็น NULL

## 2.7 ลักษณะของ PDU

### 2.7.1 GetRequest PDU

GetRequest PDU จะถูกส่งออกมาโดย SNMP entity ในนามของ network management station application entity ที่เป็นฝ่ายส่งจะรวมเอาฟิลด์ดังต่อไปนี้เอาไว้ใน PDU

2.7.1.1 PDU type : แสดงให้ทราบว่านี่คือ GetRequest PDU

2.7.1.2 request-id : entity ฝ่ายส่งจะกำหนดตัวเลขเฉพาะสำหรับแต่ละการร้องขอไปยัง agent หนึ่ง ๆ request-id จะทำให้ SNMP application สามารถสร้างความสัมพันธ์ระหว่างการตอบสนองที่ได้รับ (incoming responses) กับการร้องขอที่ยังค้างอยู่ (outstanding requests) ได้และยังทำให้ SNMP entity สามารถจัดการกับ PDU ที่ซ้ำ ๆ กันซึ่งถูกสร้างโดย transport service ที่เชื่อถือไม่ได้ด้วย

2.7.1.3 variablebindings : เป็นรายการของ object instance ที่ต้องการทราบค่า

SNMP entity ที่เป็นฝ่ายรับจะตอบสนองต่อ GetRequest PDU ด้วย GetResponse PDU ซึ่งบรรจุค่า request-id ที่เหมือนกัน

GetRequest operation มีคุณสมบัติเป็น atomic หมายความว่า ค่าทั้งหมดจะถูกดึงออกมาหรือไม่ก็ไม่มีเลยกล่าวคือ ถ้า entity ที่เป็นฝ่ายตอบสนองสามารถให้ค่าตัวแปรทั้งหมดที่อยู่ใน variablebindings list ที่รับเข้ามาได้แล้ว GetResponse PDU ก็จะมี variablebindings field แยกด้วย แต่ถ้าค่าของตัวแปรใดตัวแปรเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อญาติเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งไม่สามารถถูกจัดหาให้ได้แล้วก็จะไม่มีค่าใด ๆ ถูกส่งกลับมาเลยซึ่งอาจเกิดขึ้นเนื่องจาก error condition ดังต่อไปนี้คือ

- Object named ใน variablebindings field อาจจะไม่ตรงกับ object identifier ใดใน MIB view ที่เกี่ยวข้องเลยหรืออาจจะเป็นชื่อของ aggregate type ดังนั้นจึงไม่มี instance value ที่สัมพันธ์กัน ในกรณีดังกล่าว GetResponse PDU จะมี error-status เป็น noSuchName และค่าใน error-index field เป็นค่าของ index ของ object ที่มีปัญหาใน variablebindings field ดังนั้นถ้าตัวแปรตัวที่สามใน list ของ variablebindings field ไม่สามารถให้ค่ากับ get operation ได้แล้ว error-index field จะมีค่าเป็น 3
- ขนาดของ GetResponse PDU อาจจะทำให้ใหญ่เกินไป ในกรณีนี้ entity ที่ทำหน้าที่ตอบสนองจะส่ง GetResponse PDU ที่มี error-status เป็น tooBig
- entity ที่เป็นฝ่ายตอบรับ (responding entity) อาจจะไม่สามารถให้ค่ากับ object ด้วยสาเหตุอื่น ในกรณีนี้ entity ที่เป็นฝ่ายตอบรับจะส่ง GetResponse PDU ที่มี error-status เป็น genErr และค่าใน error-index field เป็นค่าของ index ของ object ที่มีปัญหาใน variablebindings field

ลอจิกดังกล่าวข้างต้นสามารถสรุปเป็น procedure ได้ดังนี้คือ

```
procedure receive-getrequest;
```

```
begin
```

```
  if object not available for get then
```

```
    issue getresponse (noSuchName, index)
```

```
  else if generated PDU too big then
```

```
    issue getresponse (tooBig)
```

```
  else if value not retrievable for some other reason then
```

```
    issue getresponse (genErr, index)
```

```
  else issue getresponse (variablebindings)
```

```
end;
```

### 2.7.2 GetNextRequest PDU

GetNextRequest PDU เกือบจะเหมือนกับ GetRequest PDU ทุกประการ มันมีวิธีการแลกเปลี่ยน PDU ซึ่งเหมือนกับ GetRequest PDU สิ่งที่แตกต่างกันคือ สำหรับ GetRequest PDU แต่ละตัวแปรใน variablebindings list จะอ้างถึงค่าของ object instance ว่าเป็นลำดับถัดไปของมันใน lexicographical order

Lexicographical order สามารถสร้างได้โดยการ traverse tree ของ object identifier ใน MIB ซึ่ง child node ของ parent node ถูกเรียงลำดับจากน้อยไปหามากเสมอ (object identifier คือลำดับของเลขจำนวนเต็มซึ่งสะท้อนให้เห็นถึงโครงสร้างแบบทรีของ object ต่าง ๆ ใน MIB

เช่นเดียวกับ GetRequest GetNextRequest มีคุณสมบัติเป็น atomic ลอจิกดังกล่าวข้างต้นสามารถสรุปเป็น procedure ได้ดังนี้คือ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure receive-getnextrequest;
```

```
begin
    if no next object not available for get then
        issue getresponse (noSuchName, index)
    else if generated PDU too big then
        issue getresponse (tooBig)
    else if value not retrievable for some other reason then
        issue getresponse (genErr, index)
    else issue getresponse (variablebindings)
end;
```

ความแตกต่างระหว่าง GetRequest และ GetNextRequest ทำให้ network management station สามารถค้นพบโครงสร้างของ MIB view ได้ และยังให้กลไกในการค้นหา table ซึ่งเราไม่รู้ค่า entries ของมันได้อย่างมีประสิทธิภาพอีกด้วย

### 2.7.3 SetRequest PDU

SetRequest PDU ถูกส่งออกมาโดย SNMP entity ในนามของ network management station application SetRequest PDU มีวิธีการแลกเปลี่ยนและรูปแบบ message ที่เหมือนกันกับ GetRequest PDU SetRequest จะใช้สำหรับเขียนค่าของ object ดังนั้นใน variablebindings list ของ SetRequest PDU จะมีทั้ง object instance identifier และค่าที่ถูกกำหนดให้กับแต่ละ object instance ใน list

SNMP entity ที่เป็นฝ่ายรับจะตอบสนองค่า SetRequest PDU ด้วย GetResponse PDU ที่บรรจุค่า request-id ที่เหมือนกัน

SetRequest operation มีคุณสมบัติเป็น atomic หมายความว่า ตัวแปรทั้งหมดจะถูกอัปเดต หรือไม่ก็ไม่มีตัวแปรใดถูกอัปเดตเลย ถ้า entity ที่เป็นฝ่ายตอบรับ (responding entity) สามารถเซตค่าให้กับตัวแปรทุกตัวใน variablebindings list ที่รับเข้ามาได้แล้ว GetResponse PDU จะมี variablebindings field ซึ่งมีค่าที่ให้กับตัวแปรแต่ละตัวรวมอยู่ด้วย ถ้าตัวแปรตัวใดตัวหนึ่งไม่สามารถถูกเซตค่าได้แล้วจะไม่มีค่าใดถูกส่งกลับมาและไม่มีค่าใดถูกอัปเดต error condition ที่เหมือนกับในกรณีของ GetRequest อาจจะถูกส่งกลับมา (noSuchName, tooBig, genErr) error condition อีกอันหนึ่งที่อาจถูกส่งมาก็คือ badValue error condition นี้จะถูกส่งมาถ้า SetRequest มีคู่ของ variable name และ value อย่างน้อยที่สุดหนึ่งคู่ที่ไม่สอดคล้องกันในชนิด (type), หรือความยาว (length), หรือค่าที่แท้จริงของค่าที่ถูกกำหนดให้

ลอจิกดังกล่าวข้างต้นสามารถสรุปเป็น procedure ได้ดังนี้คือ

```
procedure receive-setrequest;
```

```
begin
    if object not available for set then
        issue getresponse (noSuchName, index)
```

```
    else if generated PDU too big then
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        issue getresponse (tooBig)
    else if value not retrievable for some other reason then
        issue getresponse (genErr, index)
    else issue getresponse (variablebindings)
end;

```

#### 2.7.4 Trap PDU

Trap PDU จะถูกส่งออกมาโดย SNMP entity ในนามของ network management agent application Trap PDU ถูกใช้สำหรับการแจ้งให้ management station ทราบว่า มีเหตุการณ์ที่สำคัญบางอย่างเกิดขึ้น รูปแบบของ Trap PDU ก่อนข้างที่จะแตกต่างกับ SNMP PDUs ตัวอื่น Trap PDU จะประกอบด้วยฟิลด์ต่าง ๆ ดังนี้คือ

- PDU type : แสดงให้เห็นว่าเป็น Trap PDU
- enterprise : ใช้ระบุ network management subsystem ที่สร้าง trap ขึ้นมา
- agent-addr : IP address ของ object ที่สร้าง trap
- generic-trap : predefined trap type ชนิดหนึ่ง
- specific-trap : รหัสที่แสดงให้เห็นถึงธรรมชาติของ trap โดยเฉพาะเจาะจงมากกว่า
- time-stamp : เวลาระหว่างการเริ่มต้น ((re)initialization) ของ network entity ครั้งสุดท้ายซึ่งมีการส่ง trap ออกมากับการเกิดขึ้นของ trap
- variablebindings : ข้อมูลเพิ่มเติมที่เกี่ยวข้องกับ trap
- generic-trap field อาจจะมีค่าเป็นค่าใดค่าหนึ่งใน 7 ค่าต่อไปนี้
- ColdStart(0) : SNMP entity ที่เป็นฝ่ายส่งกำลังทำการ reinitialize ตัวเองเพื่อให้โครงสร้างของ agent (agent's configuration) หรือ protocol entity implementation ถูกเปลี่ยนแปลงได้
- WarmStart(1) : SNMP entity ที่เป็นฝ่ายส่งกำลังทำการ reinitialize ตัวเองเพื่อให้ทั้ง agent's configuration และ protocol entity implementation ไม่ถูกเปลี่ยนแปลง โดยทั่วไปนี่คือ routine restart
- linkDown(2) : บอกให้ทราบว่า failure เกิดขึ้นในส่วนเชื่อมต่อการสื่อสาร (communication link) หนึ่งของ agent
- linkUp(3) : บอกให้ทราบว่าส่วนเชื่อมต่อการสื่อสารหนึ่งของ agent ได้เกิดขึ้นแล้ว
- authenticationFailure(4) : บอกให้ทราบว่า protocol entity ที่เป็นฝ่ายส่งได้รับ protocol message ซึ่งไม่ผ่าน authentication
- egpNeighborLoss(5) : บอกให้ทราบว่า EGP neighbor ซึ่งมี protocol entity ที่เป็นฝ่ายส่ง (sending protocol entity) เป็น EGP peer ได้ถูก marked down แล้วและ peer relationship จะไม่มีอีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- enterpriseSpecific(6) : แสดงให้เห็นว่า protocol entities ที่เป็นฝ่ายส่ง (sending protocol entity) พบว่ามี enterprise-specific event บางอย่างเกิดขึ้น specific-trap field จะแสดงชนิดของ trap

## 2.8 ข้อจำกัดของ SNMP

1. ด้านความปลอดภัย เนื่องจาก SNMP มี concept ในการรักษาความปลอดภัยโดยใช้ Community Name เพียงอย่างเดียวและการใช้ Community Name ก็มีจุดอ่อนดังได้กล่าวไปแล้ว ดังนั้นจะเห็นได้ว่าในเรื่องความปลอดภัยยังถือเป็นข้อด้อยสำคัญของ SNMP
2. การถ่ายโอนข้อมูลจำนวนมาก การถ่ายโอนข้อมูลเป็นจำนวนมากๆ โดย SNMP เช่น การถ่ายโอนข้อมูลที่เป็นตารางนั้น ต้องทำการถ่ายโอนทีละแถวไปเรื่อยๆจนหมดตาราง ไม่สามารถทำการถ่ายโอนข้อมูลทั้งหมดทีเดียวได้ ทำให้เสียเวลามาก
3. การบริหารอย่างเป็นลำดับขั้น คงต้องยอมรับกันว่าอุปกรณ์ในระบบนั้นมีเป็นจำนวนร้อยหรืออาจเป็นพันทีเดียว ดังนั้นการที่จะใช้ NMS เพียงตัวเดียวในการบริหารเครือข่ายย่อมทำให้ NMS ทำงานหนักมาก ความคิดที่จะกระจายจุดรวบรวมข้อมูลให้มีหลายจุดแล้วค่อยส่งข้อมูลที่รวบรวมได้นั้นส่งให้ NMS จึงเกิดขึ้น ซึ่งก็เป็นความคิดที่ดี แต่ใน SNMP ไม่มีการจัดให้มีการติดต่อระหว่างจุดย่อยดังกล่าวกับ NMS เนื่องจากถือว่าจุดย่อยนั้นเปรียบเสมือน NMS ตัวหนึ่ง ความคิดนี้จึงไม่สามารถทำได้ในทางปฏิบัติ
4. การรายงานความผิดพลาด เมื่อ NMS ส่ง message ไปขอข้อมูลจากอุปกรณ์แต่อุปกรณ์ไม่มีข้อมูลดังกล่าว NMS จะไม่สามารถรู้ได้เลยว่าเกิดเหตุการณ์นี้ขึ้น ดังนั้น NMS จะทำการวนขอข้อมูลไปเรื่อยๆ แทนที่จะสามารถใช้เวลาที่เสียไปนี้ไปทำงานอย่างอื่นได้

## 2.9 การพัฒนา SNMP

### Secure SNMP

ในครั้งแรกที่ SNMP ถูกนำไปเผยแพร่ นั้น จุดอ่อนของ SNMP ยังไม่ได้ถูกค้นพบทั้งหมด มีเพียงจุดอ่อนด้านความปลอดภัยเท่านั้นที่ทุกคนมองเห็น Secure SNMP ถูกพัฒนาขึ้นมาเพื่อแก้ไขจุดอ่อนนี้ แต่ก็ไม่ได้ได้รับความนิยมเนื่องจาก

1. แก้ไขจุดอ่อนเพียงอันเดียวของ SNMP
2. ระบบรักษาความปลอดภัยที่สร้างขึ้น มีผลกระทบให้ต้องมีการส่ง message จำนวนมากขึ้นกว่าเดิมในการทำงานแต่ละครั้ง
3. Secure SNMP ใช้ Data Encryption Standard (DES) Algorithm ในการเข้ารหัสและตรวจสอบรหัส ทำให้การเผยแพร่ Algorithm นี้ต้องทำภายใต้การควบคุมของกระทรวงพาณิชย์ของสหรัฐอเมริกา ดังนั้นจึงไม่สามารถนำ Algorithm นี้มาใช้ในระบบเครือข่ายข้ามชาติได้

### The Simple Management Protocol (SMP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า มีลักษณะสำคัญดังนี้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. SMP ถูกออกแบบเป็น Management Protocol ดังนั้นจึงสามารถนำไปใช้ในการบริหารของระบบได้ ไม่เพียงแต่การบริหารเครือข่ายเท่านั้น
2. สามารถใช้ร่วมกับ TCP/IP
3. ได้ออกแบบระบบรักษาความปลอดภัยเพื่อแก้ไขจุดอ่อนของ SNMP
4. ถูกออกแบบมาให้สามารถรองรับ SNMP ได้

### SNMPv2

ถูกสร้างขึ้นในช่วงเวลาระหว่างการพัฒนา SMP และ Secure SNMP ซึ่งเป็นรูปแบบที่ได้รับความนิยมในปัจจุบัน ใน SNMPv2 มีหัวข้อที่ถูกปรับปรุงใหม่ดังนี้

1. ด้านความปลอดภัย SNMPv2 ได้ทำการปรับปรุงระบบรักษาความปลอดภัยของ Secure SNMP โดยตั้งวัตถุประสงค์เพื่อลดจำนวน message ที่ต้องใช้ในการทำงานแต่ละครั้งลงและเพื่อให้สามารถนำเครื่องมือนี้ มาใช้ในระบบเครือข่ายข้ามชาติได้ ในการบรรลุดังกล่าวนี้ จึงมีการปรับปรุงเครื่องมือที่ใช้ในการเข้ารหัส ซึ่งแต่เดิมใช้ DES Algorithm ขึ้นมา โดย Algorithm ใหม่ นี้ถูกนำมาใช้กับเครือข่ายข้ามชาติ ทำให้ทุกเครือข่ายในปัจจุบันสามารถใช้ Management Protocol แบบเดียวกันทั้งหมด
2. เพิ่มความสามารถในการทำงานบริหาร ทำให้จำนวนงานที่ถูกทำในการบริหารเครือข่ายภายใน 1 วินาที มากขึ้นกว่าเดิม
3. การรายงานความผิดพลาด มีการเพิ่มชนิดของ message ที่ถูกส่งระหว่าง agent และ manager ขึ้นมาเพื่อให้ agent สามารถแจ้งแก่ manager ได้เมื่อไม่มีข้อมูลที่ manager เรียกว่า อีกสิ่งให้ agent สามารถยืนยันการเซตค่ากลับมาให้ manager ทราบได้อีกด้วย
4. การถ่ายโอนข้อมูลจำนวนมาก มีการนิยาม GetBulk PDU เพิ่มขึ้นมา เพื่อให้สามารถถ่ายโอนข้อมูลจำนวนมากได้ภายในครั้งเดียวเช่น ในการเรียกข้อมูลที่เป็นตาราง GetBulk สามารถทำให้เกิดการส่งข้อมูลทั้งหมดได้พร้อมกัน ไม่ต้องส่งทีละแถวเหมือนเมื่อใช้ GetNextRequest อีกต่อไป
5. การปฏิบัติการภายใต้โปรโตคอลอื่น แต่เดิมนั้น SNMP ปฏิบัติการภายใต้ UDP ได้เพียงอย่างเดียวเท่านั้น ใน SNMPv2 จึงทำการปรับปรุงให้ SNMP สามารถปฏิบัติการภายใต้โปรโตคอลอื่นๆได้ด้วย
6. ความสามารถในการติดต่อระหว่าง Manager กับ Manager คงยังจำกันได้ว่ามีความคิดในการช่วยลดภาระในการทำงานของ NMS ลงโดยการกระจายจุดรวบรวมข้อมูลออกไป เพื่อให้ทำการรวบรวมข้อมูลแล้วจึงค่อยส่งข้อมูลนั้นต่อมาให้ NMS ซึ่งใน SNMP แบบเก่านั้นไม่สามารถทำตามความคิดนี้ได้ เนื่องจากไม่มีการเปิดให้เกิดการส่งข้อมูลจากจุดย่อยซึ่งก็คือ NMS ย่อย มายัง NMS ศูนย์กลาง ดังนั้นใน SNMPv2 จึงมีการพัฒนาให้ NMS สามารถทำหน้าที่เป็นทั้ง agent และ manager ได้ โดยเมื่อ NMS ย่อยกำลังทำหน้าที่รวบรวมข้อมูลจากอุปกรณ์ มันจะทำหน้าที่เป็น manager และในขณะที่ทำการส่งข้อมูลให้กับ NMS ศูนย์กลาง มันจะทำหน้าที่เป็น agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### Management Information Base

การบริหารระบบเครือข่ายที่ใช้โปรโตคอลเป็นแบบ IPC/IP จะมีส่วนของฐานข้อมูล (Database) ที่เป็นตัวเก็บข้อมูลเกี่ยวกับ element ต่างๆ ที่ใช้ในการบริหารระบบ โดยจะเรียกฐานข้อมูลดังกล่าวว่า MIB หรือ Management Information Base ซึ่งภายใน MIB จะเก็บข้อมูลของแต่ละ resource โดยจะถูกมองให้เป็น Object โดยลักษณะภายในของ MIB จะมีลักษณะเป็น Tree ขนาดใหญ่ที่บอกให้ทราบถึงสถานที่เก็บข้อมูลข่าวสารต่างๆ โดยเราสามารถจะตรวจสอบหรือดูสถานะต่างๆ ของ Resource ได้โดยการอ่านค่าข้อมูลจาก MIB หรือจะทำการควบคุมการทำงานของ Resource โดยการทำการแก้ไขข้อมูลภายใน MIB

#### 3.1 โครงสร้างของข้อมูลที่จะใช้ในการบริหาร (Structure of management information = SMI)

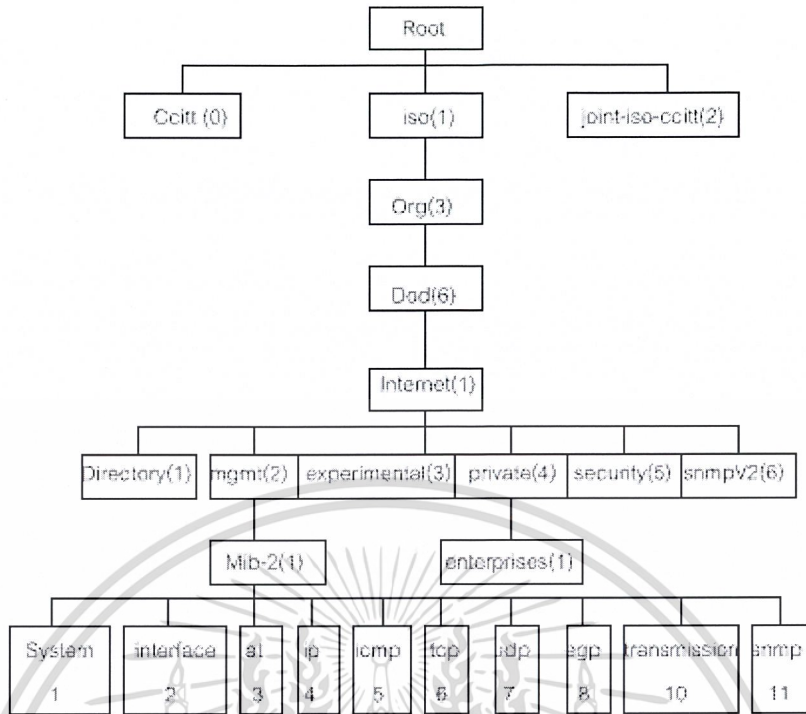
โครงสร้างของข้อมูลหรือ SMI นี้ได้มีการกำหนดชนิดของข้อมูลที่ไม่ซับซ้อนที่สามารถที่จะใช้ใน MIB และหลักการตั้งชื่อให้กับ resource ต่างๆ MIB สามารถที่จะใช้เก็บได้เฉพาะชนิดข้อมูลง่ายๆ เท่านั้น เช่น ข้อมูลแบบ Scalar และแบบ Array 2 มิติ ซึ่งเป็นเหตุผลว่าทำไม SNMP จึงสามารถที่จะเรียกดูข้อมูลได้เฉพาะแบบ Scalar และแบบตาราง SMI จะไม่สนับสนุนให้มีการสร้างหรือเรียกดูข้อมูลที่มีลักษณะโครงสร้างที่ซับซ้อน โดยเราจะใช้ SMI ในการ

- กำหนดโครงสร้างของ MIB
- กำหนด Object รวมไปถึง syntax และค่าของแต่ละ Object
- การ Encode ค่าของ Object

##### 3.1.1 MIB Structure

ในการกำหนด Object ใน MIB จะกำหนดโดยใช้ภาษา ASN.1 (Abstract Syntax Notation One) โดยจะใช้ OBJECT IDENTIFIER ในการกำหนด Object โดยการระบุชื่อของ Object โดยโครงสร้างภายใน MIB จะประกอบไปด้วย Object ในลักษณะโครงสร้างที่เป็น Tree ดังที่ได้กล่าวไว้ก่อนหน้านี้ซึ่งจะแสดงได้ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 โครงสร้างของ MIB

สมมติว่า node ที่มีชื่อว่า system มีลูกต่อไปอีกหลายตัว ได้แก่ sysDescr, sysContact, sysName และ sysLocation โดยโหนดเหล่านี้เป็น leaf node (คือ ไม่มีลูกต่อไปอีก) เราจะเรียกโหนดเหล่านี้ว่า Object ซึ่ง Object เหล่านี้จะไม่มีการมี value อยู่ภายใน แต่ Object จะกำหนดชนิด (type) ของ value และ value จะอยู่ภายใน Object instance

ความแตกต่างระหว่าง Object กับ instance จะเห็นได้ชัดเจนขึ้นเมื่อพิจารณาการอ้างถึงข้อมูลที่อยู่ภายใน MIB ดังนี้

ถ้าเราต้องการอ้างถึง Object sysDescr เราจะใช้เส้นทางว่า

.iso.org.dod.internet.mgmt.mib-2.sysDescr

แต่ถ้าเราต้องการอ้างถึง value ของ sysContact เราจะใช้เส้นทางว่า

.iso.org.dod.internet.mgmt.mib-2.sysDescr.0

เราต้องมีจุดหน้า node แรก (iso) เพื่อระบุว่าเส้นทางนี้เริ่มต้นจาก root ของ MIB และตัวเลขที่อยู่ภายในวงเล็บที่ตามหลังชื่อ node ทุก node เพื่อการระบุเส้นทางแบบย่อยๆ ตัวอย่างเช่น ถ้าเราต้องการอ้างถึง node ที่มีชื่อว่า system (ที่ถูกควรเรียกว่า group ที่มีชื่อว่า system เนื่องจาก system เป็นพ่อของ system leaf node ทุกตัว) เราสามารถใช้เส้นทางว่า

1.3.6.1.2.1.1

**แทนการใช้**

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.iso.org.dod.internet.mgmt.mib-2.system

ได้

โดยตัวเลขเหล่านี้จะเรียงลำดับตามตำแหน่งของกิ่ง (branch) ของ tree concept ที่สำคัญของ MIB คือชุดของตัวเลขที่ถูกใช้แทนชื่อข้อมูลที่ถูกใช้ไปแล้วจะไม่ถูกนำกลับมาใช้อีกถึงแม้ว่านิยามของข้อมูลอันนั้นจะถูกลบทิ้งไปแล้วก็ตาม และเมื่อมีการเพิ่มข้อมูลใหม่ลงใน MIB นิยามของข้อมูลดังกล่าวจะถูกเขียนเป็น ASCII text โดยใช้ ASN.1 BER ไฟล์ที่เขียนขึ้นนี้จะถูกนำมา compile โดยใช้ ASN.1 Compiler (เป็นโปรแกรมที่มีอยู่ใน NMS) เมื่อไฟล์ถูก compile แล้ว NMS ก็จะสามารถมองเห็นข้อมูลใหม่ได้โดยอัตโนมัติ

ส่วนการ allocate node หรือ object เราสามารถจะใช้คำสั่งดังนี้

```
internet OBJECT IDENTIFIER ::= {iso org(3) dod(6) 1}
```

ซึ่งก็คือสั่งให้ทำการ allocate Object internet โดยไปเป็น node ลูกของ dod โดยเป็นกิ่งที่ 1 ดังในรูปที่ 3.1

### 3.1.2 Object Syntax

SMI Syntax เป็นส่วนย่อยของ Syntax มาตรฐาน ISO ที่ชื่อว่า *Abstract Syntax Notation 1 (ASN.1)* ที่มีความซับซ้อนกว่ามาก ชนิดของข้อมูลใน ASN.1 มี 3 ชนิดคือ

1. ชนิดทั่วไป (Universal Types) ซึ่งประกอบด้วย INTEGER, OCTET STRING, NULL, OBJECT IDENTIFIER, SEQUENCE, SEQUENCE OF
2. Application Type เป็นชนิดข้อมูลที่เพิ่มเติมเข้ามา ซึ่งประกอบด้วย
  - Network Address ข้อมูลชนิดนี้สร้างโดยใช้คำสั่ง CHOICE
  - Ip Address เป็นชนิดข้อมูลที่มีขนาด 32 บิตใช้ในการระบุ IP
  - Counter เป็น non-negative integer ที่สามารถทำการ increase ได้แต่ไม่สามารถที่จะทำการ decrease ได้โดยมีขนาด 32 บิต
  - Gauge เป็น non-negative integer ที่สามารถทำการ increase และ decrease ได้โดยมีขนาด 32 บิต
  - Time Ticks เป็น non-negative integer ใช้สำหรับ Object ที่มีการอ้างอิงถึง epoch
  - Opaque สนับสนุนการส่งผ่านข้อมูลชนิดใดก็ได้

### 3.1.3 ชนิดข้อมูลแบบ Object (Defining Objects)

เราสามารถที่จะกำหนดชนิดข้อมูลของ Object ให้มีชนิดเป็นแบบ Object ได้โดยทำการเขียน Macro โดยที่ Macro ที่ใช้ในการกำหนดได้แสดงไว้ดังนี้

```
IMPORTS ObjectName, ObjectSyntax FROM RFC-1155-SMI
```

```
OBJECT-TYPE MACRO ::=
```

```
BEGIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TYPE NOTATION ::= “SYNTAX”      type(TYPE ObjectSyntax)
                    “ACCESS”      Access
                    DescrPart
                    ReferPart
                    IndexPart
                    DefValPart

VALUE NOTATION ::=value{VALUE ObjectName}

Access ::= “read-only” | “read-write” | “write-only” | “not-accessible”

Status ::= “mandatory” | “optional” | “obsolete” | “deprecated”

DescrPart ::= “DESCRIPTION” value (description DisplayString) | empty

ReferPart ::= “REFERENCE” value {reference DisplayString} | empty

IndexPart ::= “INDEX” “{ “IndexTypes”}”

IndexTypes ::= IndexType | IndexTypes “,” IndexType

IndexType ::= value{indexobject ObjectName} --if indexobject,use the SYNTAX
--value of the correspondent
--OBJECT-TYPE invocation
| type (indextype) --otherwise use named SMI type;
--must conform to IndexSyntax
--below

DefValPart ::= “DEFVAL” “{ “value (defvalue ObjectSyntax) “}” | empty

DisplayString ::= OCTET STRING SIZE (0..255)

END

IndexSyntax ::= CHOICE { number INTEGER (0..MAX),
                        string OCTET STRING,
                        object OBJECT IDENTIFIER,
                        address NetworkAddress,
                        IpAddress IpAddress }

```

### 3.1.4 Encoding

Object ใน MIB ได้มีการทำ encode โดยใช้วิธี basic encoding rules (BER) ที่เหมาะสมใช้กับ ASN.1 ซึ่งเป็นวิธีการ encode ที่เป็น standardized ที่ใช้กันทั่วไป

#### 3.1.4.1 Encoding Rules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า เป็นกฎเกณฑ์ในการเขียน MACRO นิยามข้อมูล จุดสำคัญใน MACRO ที่ควรระวังมีดังนี้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วน Syntax นิยามชนิดของข้อมูล
- ชื่อของข้อมูลที่อ้างอิงจาก MIB
- วิธีการ access ข้อมูลได้แก่ อ่านได้อย่างเดียว , อ่านและเขียนได้ , เขียนได้อย่างเดียว และ อนุญาตให้ access ใดๆทั้งสิ้น
- สถานะของข้อมูลได้แก่ เป็นข้อมูลที่ต้องทำ (เป็นคำสั่ง) , สามารถเลือกทำหรือไม่ก็ได้ ข้อมูล
- ถูกลบทิ้งไปแล้ว และ เป็นข้อมูลที่ล้าสมัย

### 3.2 Practical issues

#### 3.2.1 Measurement

จากตอนต้นที่ได้อธิบายถึง 2 หน้าที่หลักๆของการทำ network-management system ก็คือการทำ monitoring และการ control ซึ่งการทำ monitoring ก็เพื่อทำการวัดและทำรายงานผลการทำ management system ใน SNMP จะมี MIB ที่สนับสนุนการทำ monitoring โดยใช้ค่า scalar ซึ่งอาจจะเป็นการ counter และค่าที่ได้จากการประมาณ ข้อเสียใน SNMP ก็คือการออกแบบการทำ network-management อย่างไรจึงจะทำให้ manager มั่นใจได้กับผลของรายงานที่ได้

บางครั้งการนี้อาจจะเป็นปัญหาเล็กน้อย เช่น พิจารณาอุปกรณ์อย่างเช่น bridge หรือ router ซึ่งมีหน้าที่เป็นสื่อกลาง (intermediate system) ที่คอยทำการจับ (capture) และ forward packet ที่วิ่งจาก host หนึ่งไปยัง host หนึ่งในระบบ internet ตัว manager จะเป็นผู้กำหนดขนาดของการออกแบบ(มีจำนวน bridge หรือ router เท่าไร) เพื่อที่จะทำ monitor จำนวนของ traffic ที่ผ่านมาที่ bridge หรือ router ซึ่งเป็นเรื่องที่ยากและแน่นอนว่าไม่ว่า network-management package ใดก็สามารถที่จะนับและรายงานจำนวนของ packet ที่เข้าออกในแต่ละอุปกรณ์ได้อย่างถูกต้องแม่นยำ ได้มีการทดสอบโดยกลุ่มนักวิจัยของ AT&T ได้ผลว่ามีปัญหาบางอย่างเกิดขึ้น โดยในการทดสอบได้มีการลอง run ทดสอบบน SNMP-managed bridge หลายยี่ห้อ โดยได้กำหนดชุดของข้อมูลและรายงานโดยตัว agent ที่แน่นอน โดยในการทดลองได้มีการจำลองเป็นการใช้งานของ user ในองค์กรขนาดใหญ่โดยมี network-management center และมีอุปกรณ์ต่างๆ หลายชนิดหลายยี่ห้อมาใช้เป็น SNMP-manage โดยประกอบเป็น Ethernet Lan 2 วง ที่มีอุปกรณ์

- Bridge : เป็นตัวเชื่อมต่อ Lan 2 วงโดยใช้ bridge 1 ตัวโดยมีอยู่ 7 ยี่ห้อมาใช้ในการทดสอบโดยแต่ละยี่ห้อจะมี SNMP agent software เป็นของตนเอง
- Network-management station : ในการทดสอบจะใช้ 2 network management station ที่ต่างยี่ห้อกัน
- Lan analyzer : ใช้สำหรับทำการ generate test packet address สำหรับ off-Lan destination (MAC และ IP address) และทำการจับและวิเคราะห์ message ที่ได้จาก SNMP agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 Private MIBs

ใน SNMP ที่จะมีการใช้ MIB ที่ถูกออกแบบมาเพื่อที่จะสามารถเพิ่ม Object เข้าไปได้ ซึ่งเราจะอ้างถึง Private extension ที่สามารถจะเพิ่ม private subtree โดยจะอนุญาตให้ vendor ทำการสร้าง object เพื่อที่จะระบุ manage entity บน product ของตนและ object นี้จะถูก management station เห็นเพราะว่าได้มีการใช้ standardized SMI และ standardized object identifier ในการออกแบบ จึงเป็นไปได้ว่าเราจะสามารถ manage private object โดย management station ที่มี vendor ต่างกันได้

ใน SNMP ตัว management station สามารถจะ access ได้เฉพาะข้อมูลที่มีรู้จักและรู้ว่าจะเรียกขอข้อมูลอย่างไร ซึ่งเราจะกล่าวถึงการออกแบบ SNMP เพื่อป้องกันการที่ตัว management station ไปเรียกขอข้อมูลจาก agent ไม่ได้ โดยธรรมดาจะไม่เป็นปัญหาหาก management station และ agent station มาจาก vendor เดียวกันแต่สำหรับจากต่าง vendor กันก็อาจจะมีปัญหาได้ โดยส่วนมากแล้ว format ของ MIB ที่มีโดยทั่วไปจะมีอยู่ 3 format คือ

- เป็น Original SNMP SMI ที่ระบุโดย RFC 1155
- Concise MIB Format ระบุโดย RFC 1212
- OSI SMI specification

ในการ convert จาก MIB หนึ่งไปเป็น MIB format หนึ่งจำเป็นต้องใช้คนที่มีความชำนาญและจะต้องมีความรู้ใน MIB ทั้ง 2 format ซึ่ง RFC 1212 ได้มี guideline ที่เรียกว่า deOSIfying เป็นการ convert จาก OSI SMI ไปเป็น SNMP MIB format และอธิบายวิธีการต่างๆ

### 3.3 มิบบมาตรฐาน (MIB Standard)

เป็นมิบที่ถูกกำหนดให้เป็นมาตรฐานร่วมกันระหว่างผู้ผลิตต่าง ๆ จากรูปที่ 1 มิบบมาตรฐานจะอยู่ที่ โทหนด iso.org.dod.internet.mgmt.mib2 ซึ่งมีทั้งหมด 11 กลุ่ม

#### 3.3.1 System Group

System group เป็นกลุ่มที่แสดงข้อมูลทั่วไปเกี่ยวกับคอนฟิกเกอร์เวชันต่าง ๆ ของอุปกรณ์เช่น sysObjectID เป็นข้อมูลที่บอกถึง Product Identifier ซึ่งจะบอกถึงผู้ผลิตของอุปกรณ์ซึ่ง Product Identifier จะอยู่ภายใต้ โทหนด private.enterprise ใน MIB tree และ sysService จะมีค่าที่แสดงโดย Code 7 bit โดยที่แต่ละ bit จะสัมพันธ์กับ layer ใน TCP/IP หรือ OSI architecture ซึ่ง bit ต่ำจะสัมพันธ์กับ layer 1 ถ้าระบบมีการบริการที่ layer ไคแล้วก็จะทำการ set bit โดยค่าที่ set สามารถจะแสดงได้ดังนี้

$$\text{SysService} = \sum_{L \in S} 2^{L-1}$$

โดย S = เซตของจำนวนของ Layer ที่บริการ ยกตัวอย่างเช่น node ของ host ที่มี application service จะมีการเซตค่าเป็น binary 1001000 หรือค่าฐานสิบเท่ากับ 72 ( $2^{(4-1)} + 2^{(7-1)}$ ) ใน TCP/IP protocol layer ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 แสดงอุปกรณ์ใน Layer ของ OSI

Layer	Functionality
1	Physical (เช่น repeater)
2	Data-link/subnetwork (เช่น bridge)
3	Internet (เช่น IP router)
4	End-to-end (เช่น IP host)
7	Application

ส่วน sysUpTime object ใช้แสดงเวลาดังแต่ network-management portion ของระบบจนถึงการ reinitialized ครั้งหลังสุด โดย Object นี้สามารถจะใช้สำหรับเป็น fault monitoring manager หรือผู้บริหารระบบสามารถจะทำการ poll ไปยังแต่ละ agent เป็นช่วงๆ ด้วยค่านี้ ถ้าค่า current value ของ agent น้อยกว่าค่าล่าสุด (recent value) แล้วตัว agent จะต้อง restarted นับตั้งแต่ทำ last poll

### 3.3.2 Interface Group

เป็นกลุ่มที่แสดงถึงข้อมูลต่างๆ ไปเกี่ยวกับ physical interface ของ entity รวมไปถึง configuration information และข้อมูลทางสถิติของเหตุการณ์ที่เกิดขึ้นในแต่ละ interface

ในกลุ่มนี้มี Object ifNumber ซึ่งจะมีข้อมูลของจำนวนของ network interface ทั้งหมดโดยไม่คำนึงถึงสถานะปัจจุบัน ส่วนที่เหลือจะเป็น Object ifTable ซึ่งใน 1 แถวของแต่ละ interface โดยจะมี ifIndex เป็น index โดยจะมีค่าเป็น Integer อยู่ในช่วง 1-ค่าใน ifNumber Object ifType จะเป็น record ของชนิดของ interface interface ต่อไปนี้ได้ถูก assigned ไว้ใน MIB-II

IfPhyAddress เป็น Object ที่ใช้สำหรับเก็บค่า Physical Address โดยที่ค่านี้จะขึ้นอยู่กับชนิดของ interface ยกตัวอย่างเช่น IEEE LANs และ MANs และ FDDI ข้อมูลที่ ifPhyAddress เก็บจะเป็นค่า MAC (Medium access control) address ของ interface

2 Object ในกลุ่มนี้ที่มีความสัมพันธ์กับสถานะของ interface คือ ifAdminStatus Object ซึ่งสามารถ read-write และอนุญาตให้ manager ระบุการดำเนินการกับสถานะของ interface ifOperStatus Object เป็น Object ที่ทำการ read ได้เท่านั้นเป็น Object ที่ใช้แสดงถึง current operational status ของ interface ถ้าทั้ง 2 Object นี้มีค่าเป็น down(2) แล้ว interface จะถูก shut off โดย manager ถ้า ifAdminStatus มีค่าเป็น up(1) โดยที่ ifOperStatus มีค่าเป็น down(2) แล้ว interface จะ failed หรือ turn off

Object ifSpeed เป็น read-only Object จะเก็บค่า current capacity ของ interface เป็น bits per second

### 3.3.3 AT Group

address-translation group จะประกอบด้วยตาราง 1 ตารางโดยที่แต่ละแถวในตารางจะสัมพันธ์กับ

1 physical interface ของระบบในแถวจะแสดงการ mapping จาก network address เป็น physical address โดย network address ก็คือ IP address ของระบบของ interface นั้นๆ ส่วน physical address จะขึ้นอยู่กับ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

subnetwork ยกตัวอย่าง ถ้าเป็น interface ใน lan แล้ว physical address ก็คือ MAC address ของ interface นั้นๆ แต่ถ้า subnetwork เป็น x.25 packet-switching network แล้ว physical address อาจจะเป็น x.121 address

index ของตารางจะอยู่ที่ atIfIndex โดยค่าของ IfIndex จะ match กับค่าหนึ่งใน interface group

### 3.3.4 IP Group

เป็น network protocol ที่ให้บริการในการส่ง datagram จาก source computer ไปยัง destination computer ซึ่งใน IP group นี้จะให้ข้อมูลเกี่ยวกับ IP ซึ่งจะแบ่งชนิดของข้อมูลเหล่านี้ได้ดังนี้

- Object ที่ให้ข้อมูลเกี่ยวกับ errors และชนิดของ IP packet
- ตารางของข้อมูลที่เกี่ยวข้องกับ IP address
- IP routing table
- ข้อมูลที่เป็นการ map IP address ไปเป็น address ของ protocol อื่น ๆ

### 3.3.5 ICMP Group

เป็น Internet Control Message Protocol ที่เป็นส่วนประกอบที่สำคัญของ IP เช่นเมื่อ IP เกิดปัญหาในการส่ง datagrams message ของ ICMP จะถูกส่งกลับไปยังแหล่งที่ส่ง datagram นั้นมาเป็นต้น

### 3.3.6 TCP Group

เป็น transport protocol ที่ทำให้การส่งข้อมูลผ่าน application ที่มีการเชื่อมต่อกันมีประสิทธิภาพ TCP Protocol Data Unit จะถูกเรียกว่า segment หรืออาจจะกล่าวได้ว่า segment นี้คือ TCP message ซึ่งประกอบไปด้วย TCP header และ application data TCP เป็น protocol ที่อยู่ใน layer 4 ซึ่ง TCP segment จะถูกใส่ลงใน IP datagrams เพื่อการส่งผ่านระบบเครือข่าย โดยส่วนที่เรียกว่า protocol field ใน IP header จะถูก set ให้มีค่าเป็น TCP ใช้การส่งผ่านข้อมูลแบบ connection-oriented ซึ่งสามารถควบคุมการส่งข้อมูลไม่ให้เกิดการส่งข้อมูลในจำนวนที่มากกว่าที่ buffer ของ receiver สามารถรองรับได้ และ TCP จะมีการส่งข้อมูลซ้ำอีกครั้งถ้า TCP ไม่ได้รับ acknowledge segment ในเวลาที่กำหนด

### 3.3.7 UDP Group

เป็นข้อมูลที่สัมพันธ์กับการ implement และการ execution experience ของ udp บนระบบ

### 3.3.8 EGP Group

เป็นข้อมูลที่สัมพันธ์กับการ implement และการ execution experience ของ egp บนระบบ

### 3.3.10 Transmission

เป็นข้อมูลเกี่ยวกับการ transmission schemes และ access protocols ของแต่ละ system interface

### 3.3.11 SNMP Group

เป็นข้อมูลเกี่ยวกับเอสเอ็นเอ็มพีโพรโทคอล เช่นจำนวน get, set, .... Message ที่เข้าและออกจากอุปกรณ์, จำนวนความผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### Remote Monitoring (RMON)

ด้วยโครงสร้างการเชื่อมต่อระบบเครือข่ายมีขนาดใหญ่ และมีความสลับซับซ้อนมากยิ่งขึ้นในปัจจุบัน ทำให้ระบบการทำงานของมาตรฐานโพรโทคอล SNMP ไม่สามารถจะรองรับการทำงานบริหารและควบคุมระบบเครือข่ายได้ดีพอ เพราะพื้นฐานการทำงานของมาตรฐานโพรโทคอล SNMP มีองค์ประกอบอยู่สองส่วนคือฟังก์ชัน SNMP Agent และฟังก์ชัน SMP Server เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น SNMP Server จะคอยส่งสัญญาณตรวจสอบ (Polling) และขอข้อมูลจาก SNMP Agent นั้นมาแสดงผลเป็นรูปลักษณะต่าง ๆ บนหน้าจอมอนิเตอร์ของเครื่องคอมพิวเตอร์นั้น ๆ และด้วยเหตุที่ฟังก์ชัน SNMP Server มีการทำงานในลักษณะคอยส่งสัญญาณตรวจสอบ (Polling) ก่อนที่จะได้ข้อมูลจาก SNMP Agent นั้นมาแสดงผล ดังนั้นถ้าต้องการข้อมูลของระบบเครือข่ายอย่างถูกต้องและรวดเร็วเวลาทุก ๆ นาทีหรือทุก ๆ ชั่วโมง ก็จะต้องสั่งกำหนดให้ SNMP Server ส่งสัญญาณตรวจสอบออกไปในระบบเครือข่ายทุก ๆ นาทีหรือทุก ๆ ชั่วโมง ซึ่งการทำงานในลักษณะเช่นนี้ทำให้ไม่สามารถที่จะรองรับการบริหารและควบคุมระบบเครือข่ายได้ดีพอเพราะการที่ SNMP Server ส่งสัญญาณตรวจสอบออกไปในระบบเครือข่ายเป็นจำนวนมากนั้นจะทำให้ประสิทธิภาพการสื่อสารข้อมูลในระบบเครือข่ายลดลงได้ และการที่ระบบมาตรฐานโพรโทคอลออกแบบให้ SNMP Server เป็นฟังก์ชันที่รวบรวมข้อมูลทุกอย่างจาก SNMP Agent ของทุกเครื่องแล้วคำนวณวิเคราะห์และแสดงผลโดยเครื่องคอมพิวเตอร์เดี่ยวนั้นทำให้เมื่อระบบเครือข่ายมีขนาดใหญ่ประกอบด้วยเซกเมนต์เป็นจำนวนมากและถ้าจะต้องคำนวณวิเคราะห์และแสดงผลของระบบเครือข่ายหลาย ๆ เซกเมนต์หรือทั้งหมดพร้อม ๆ กันนั้น เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น SNMP Server จะไม่สามารถรองรับการทำงานได้ดีพอและจะต้องใช้เครื่องคอมพิวเตอร์ที่มีขนาดใหญ่มาก ๆ ทั้งฮาร์ดดิสก์ (Harddisk) และหน่วยความจำ (Memory) ซึ่งจะทำให้ราคาของเครื่องคอมพิวเตอร์เครื่องนี้มีราคาที่สูงมาก

ด้วยเหตุผลของการขยายตัวของระบบเครือข่ายทำให้มีการคิดค้นมาตรฐานใหม่สำหรับการบริหารและควบคุมระบบเครือข่ายเกิดขึ้น ซึ่งในปี ค.ศ.1991 สถาบัน IETF (Internet Engineering Task Force) ได้ประกาศมาตรฐานที่เรียกว่าโพรโทคอล RMON (Remote Monitoring) โดยมาตรฐานนี้มีโครงสร้างของฐานข้อมูล MIB เช่นเดียวกับมาตรฐาน SNMP ซึ่ง RMON ได้แบ่งกลุ่มของฐานข้อมูลออกเป็นทั้งหมด 9 กลุ่มสำหรับระบบอีเทอร์เน็ต (Ethernet) และ 10 กลุ่มสำหรับระบบโทเคนริงค์ (Token Ring) องค์ประกอบการทำงานของ RMON มีองค์ประกอบคล้ายกับโพรโทคอล SNMP คือประกอบด้วยฟังก์ชัน RMON Server และฟังก์ชัน RMON Client โดยฟังก์ชัน RMON Server จะติดตั้งอยู่ในอุปกรณ์เครือข่ายต่าง ๆ ซึ่งในระบบโพรโทคอล SNMP จะทำงานด้วยฟังก์ชัน SNMP Agent และในทำนองเดียวกันเครื่องคอมพิวเตอร์ที่ผู้บริหารเครือข่ายใช้บริหารและควบคุมระบบเครือข่ายนั้นจะทำงานด้วยฟังก์ชัน RMON Client แทน โดยสำหรับระบบโพรโทคอล SNMP ก็จะทำงานฟังก์ชัน SNMP Server ดังนั้นเครื่องไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ที่ทำงานด้วยฟังก์ชัน RMON Client นี้จำเป็นจะต้องเป็นเครื่องคอมพิวเตอร์ที่มีคุณสมบัติต่าง ๆ สูงเหมือนกับเครื่องคอมพิวเตอร์ที่ทำงานด้วยฟังก์ชัน SNMP Server และยังสามารถที่จะมีเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น RMON Client ได้หลายเครื่องในระบบเครือข่ายเดียวกันได้ โดยที่ในแต่ละเครื่องอาจกำหนดให้แสดงผลของแต่ละกลุ่มของเครือข่าย หรือชนิดของอุปกรณ์เครือข่ายที่ต้องควบคุม โดยเฉพาะงานนั้น ๆ ได้

หลังจาก 3 ปีให้หลังที่ได้มีการประกาศใช้มาตรฐานโปรโตคอล RMON ก็ได้มีการริเริ่มพัฒนาระบบมาตรฐานใหม่คือ ได้เพิ่มประสิทธิภาพการทำงานของโปรโตคอล RMON ให้มากยิ่งขึ้นอีก ซึ่งได้เป็นมาตรฐานใหม่ที่เรียกว่า มาตรฐานโปรโตคอล RMON2

โปรโตคอล RMON2 เป็นโปรโตคอลที่มีการทำงานด้วยฟังก์ชันการบริหารและควบคุมระบบเครือข่ายในระดับชั้นฟังก์ชันเลเยอร์ (OSI-Layer) ที่ใกล้เคียงกับระบบซอฟต์แวร์มากยิ่งขึ้น คือจากเดิมที่โปรโตคอล RMON จะทำงานในระดับของสารสัญญาณ (Physical Layer) และชั้นดาตalink (Datalink Layer) เท่านั้น สำหรับโปรโตคอล RMON2 จะมีการทำงานในระดับตั้งแต่ชั้นเน็ตเวิร์กเลเยอร์ (Network Layer) ขึ้นไป

#### 4.1 Basic concepts

อุปกรณ์ตรวจสอบระบบเครือข่ายระยะไกล เรียกว่า “monitor” หรือ “probe” ซึ่งเป็นเครื่องมือที่ใช้สำหรับบริหารระบบเครือข่ายทำให้ผู้บริหารระบบเครือข่ายสามารถรู้การจราจรบนระบบเครือข่ายและอุปกรณ์ต่าง ๆ ที่ใช้บนระบบเครือข่าย ซึ่งเป็นการวิเคราะห์และตรวจสอบ (probes) โดยจะทำการสำรวจไปทั่วทุกที่เรียกว่า “Promiscuous mode” โดยจะสำรวจทุก ๆ packet บนระบบเครือข่าย หลังจากนั้นจะสรุปผลข้อมูลที่สำรวจได้เป็น ค่าสถิติความผิดพลาด (error statistics) เช่นจำนวนการชน (collisions) ของ packets และตัวเลขความมีเสถียรภาพเช่น จำนวนของการรับส่ง packets ต่อวินาทีและขนาดของ packet ในการแตกกระจาย

สำหรับการบริหารระบบเครือข่ายต้องการหนึ่ง monitor ต่อหนึ่งระบบเครือข่ายย่อย (subnetwork) ซึ่ง monitor อาจจะเป็นเพียงอุปกรณ์ตัวเดียวที่ใช้จับ (capture) และวิเคราะห์การจราจร ระบบเครือข่ายจะต้องมีการติดต่อระหว่าง RMON Agent และ RMON management application โดยการทำงานทั้งหมดนี้คือ remote monitors

#### 4.2 Textual Conventions

RMON MIB จะเพิ่มชนิดข้อมูลขึ้นมาใหม่ 2 ชนิดคือ OwnerString และ EntryStatus กำหนดใน ASN.1 ดังนี้

```
OwnerString ::= DisplayString
```

จะแสดงเจ้าของ ๆ object ต่าง ๆ

```
EntryStatus ::= INTEGER {valid(1), createRequest(2), underCreation(3), invalid(4)}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้แสดงสถานะ (status) ของแถว (row) ที่เก็บ object instance ซึ่งมี 4 สถานะทั้งหมดนี้ใช้ในการสร้าง (creation), ปรับปรุง (modification) และการลบ (deletion) ของแถวในตาราง ตัวอย่างโครงสร้างการใช้ทุก ๆ ตารางควบคุมและตารางข้อมูลใน RMON MIB (กำหนดใน ASN.1)

#### rmlControlTable OBJECT-TYPE

SYNTAX SEQUENCE OF RmlControlEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 “A control table”  
 ::= { ex1 1 }

#### rmlControlIndex OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION “The value of this object uniquely identifies this rmlControl entry”  
 ::= { rmlControlEntry 1 }

#### rmlControlParameter OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION “The value of this Object characterizes data table rows Associated with this entry.”  
 ::= { rmlControlEntry 2 }

#### rmlControlIndex OBJECT-TYPE

SYNTAX OwnerString  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 “The entity that configured this entry”  
 ::= { rmlControlEntry 3 }

#### rmlControlStatus OBJECT-TYPE

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYNTAX           EntryStatus

ACCESS           read-write

STATUS           mandatory

DESCRIPTION

“The status of this rmlControl entry”

::= { rmlControlEntry 4 }

**RmlControlEntry ::= SEQUENCE {**

    rmlControlIndex        INTEGER,

    rmlControlParameter   Counter,

    rmlControlOwner       OwnerString,

    rmlControlStatus      RowStatus }

ตารางที่ 4.1 ตารางควบคุม

rmlControlTable

rmlControlIndex	rmlControlParameter	rmlControlOwner	rmlControlStatus
1	5	monitor	Valid(1)
2	26	Manager alpha	Valid(1)

ตารางที่ 4.2 ตารางข้อมูล

rmlDataTable

rmlDataControlIndex	rmlDataIndex	rmlDataVale
1	1	46
2	1	96
2	2	85
2	3	77

ส่วนประกอบของ rmlControlTable (ตารางควบคุม) สามารถอ่าน-เขียนได้

- rmlControlIndex : เก็บ integer แสดงลำดับแถวทั้งหมดของ rmlcontrolTable
- rmlControlParameter : เก็บ integer แสดงลักษณะของ object ในแถวของตารางข้อมูล
- rmlControlOwner : เก็บ OwnerString แสดงเจ้าของ ๆ แถวนั้น
- rmlControlStatus : เก็บ EntryStatus สถานะของแถวนั้น

ส่วนประกอบของ rmlDataTable (ตารางข้อมูล) สามารถอ่านได้อย่างเดียว

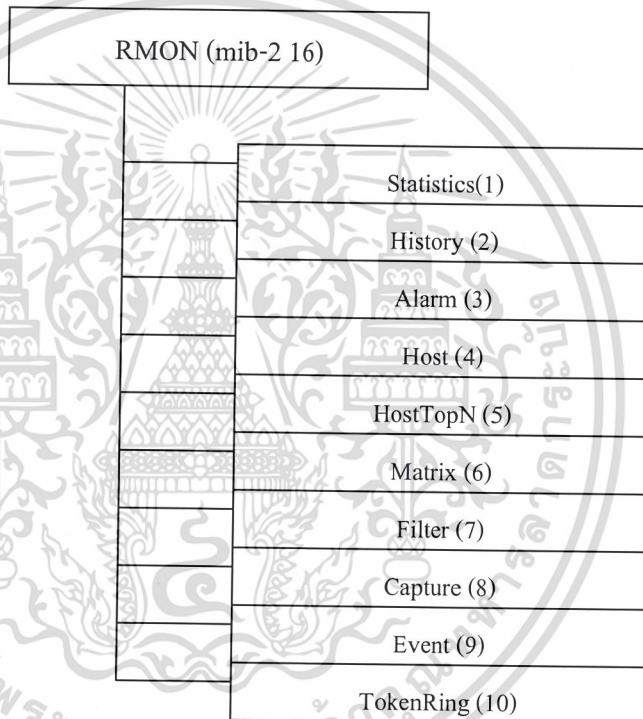
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- rmlDataControlIndex : เก็บ integer ถ้าค่าที่เก็บมีค่าเหมือนกับค่าใดใน rmlControlIndex แสดงว่าถูกควบคุมด้วยแถวนั้น ตัวอย่างในตารางด้านบน แถวของ manager alpha ควบคุมทุกแถวใน rmlDataTable ที่มีค่าของ rmlDataControlIndex เท่ากับ 2
  - rmlDataIndex : เก็บ integer แสดงลำดับแถวทั้งหมดของ rmlDataTable
  - rmlDataValue : เก็บ Counter แสดงข้อมูลข่าวสารที่ถูก monitor บนระบบเครือข่าย
- จากตัวอย่างของตารางควบคุมและตารางข้อมูลที่ได้จากการกำหนดใน ASN.1 ตารางควบคุมมี 2 แถว แถวแรกมี “monitor” เป็นเจ้าของส่วนแถวที่สองมี “manager alpha” เป็นเจ้าของ

**4.3 The RMON MIB**

RMON MIB มีทั้งหมด 10 กลุ่มดังนี้คือ



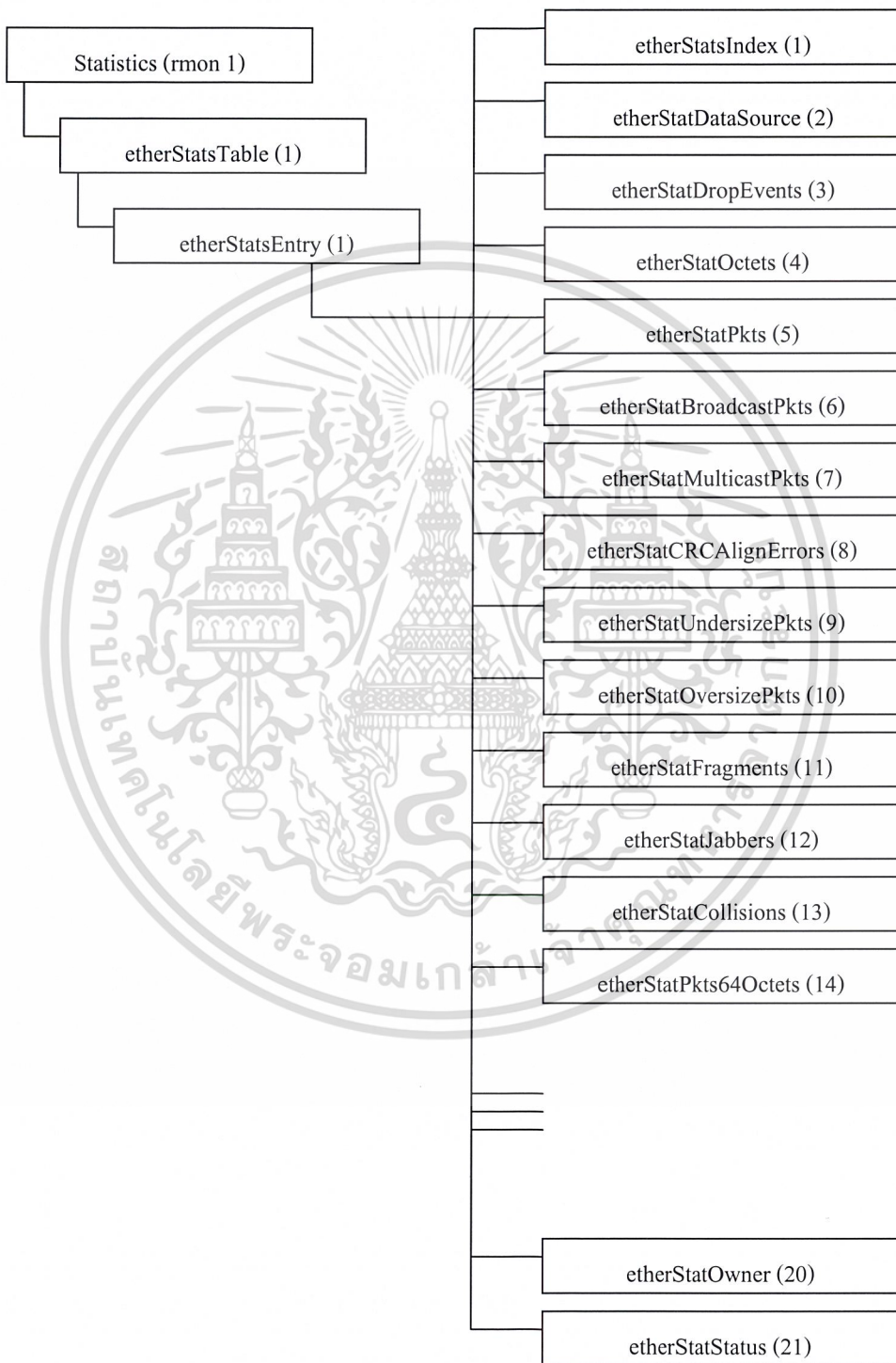
รูปที่ 4.1 แสดง RMON MIB

**4.3.1 Statistics Group**

Statistics Group หรือ Ethernet Statistics Group กลุ่มนี้จะเก็บสถิติของแต่ละระบบเครือข่ายที่ถูกตรวจสอบ (probe) สถิติที่เก็บจะเป็นข้อมูลชนิด counter ที่เริ่มจากศูนย์เมื่ออยู่ในสถานะ valid โดยสถิติต่างๆ จะเก็บไว้ใน etherStatsTable เช่น จำนวน unicast packet ที่เข้าสู่พอร์ท, จำนวน non-unicast packet ได้แก่ boardcast และ multicast packet ที่เข้าสู่พอร์ท, จำนวนแพ็กเก็ตที่เข้าสู่พอร์ทและถูกละทิ้งเนื่องจากมีความผิดพลาด (error), จำนวนแพ็กเก็ตที่เข้าสู่พอร์ทและถูกละทิ้งถึงแม้แพ็กเก็ตนั้นจะไม่มีผิดพลาดเช่น บัฟเฟอร์ของหน่วยความจำขณะนั้นไม่มีที่ว่าง, จำนวนของแพ็กเก็ตที่เข้าสู่พอร์ทและถูกละทิ้งเนื่องจากมันถูกส่งไปยังโปรโตคอลที่ไม่ได้สนับสนุน, จำนวน unicast packet ที่ออกจากพอร์ท,

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนของ non-unicast packet ที่ออกจากพอร์ท, จำนวนแพ็กเก็ตที่ส่งออกแต่ไม่ได้ส่งเนื่องจากแพ็กเก็ตมีความผิดพลาด

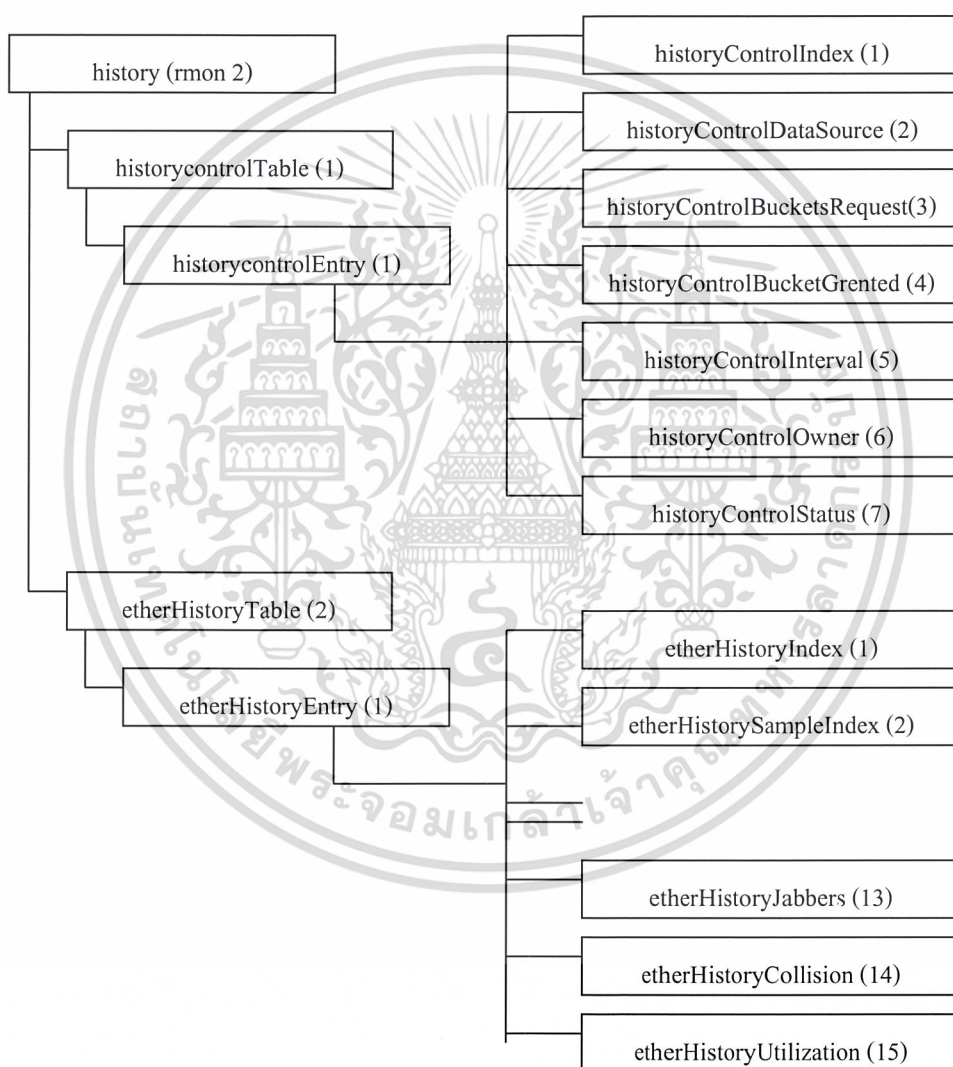


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรู๊ปที่ทางบริษัทได้มอบหมายให้ไปดูแลภาคหน้าไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 History Group

history group หรือ Ethernet History Group กลุ่มนี้จะเก็บตัวอย่างสถิติต่าง ๆ เป็นระยะ ๆ บนระบบเครือข่ายอีเทอร์เน็ตโดยจะเก็บสถิติที่ได้รับมาล่าสุดไว้ ยกตัวอย่าง monitor จะทำการสุ่มตัวอย่างต่าง ๆ ทุก ๆ 30 นาที object ที่ได้จากการสุ่มจะเก็บไว้ใน etherHistoryTable จำนวน 50 แถวเป็นต้น object ที่เก็บจะเป็นชนิด counter ยกเว้น object ใน etherHistoryUtilization จะเป็น integer ที่แสดงประสิทธิภาพของระบบเครือข่ายซึ่งจะต้องใช้ object ใน etherStatsOctets และ etherStatsPkts มาใช้ในการคำนวณจากสูตร

$$\text{ประสิทธิภาพ} = \frac{(\text{Packets} * (96 + 64) + (\text{Octets} * 8))}{\text{Interval} * 10^7} * 100\%$$



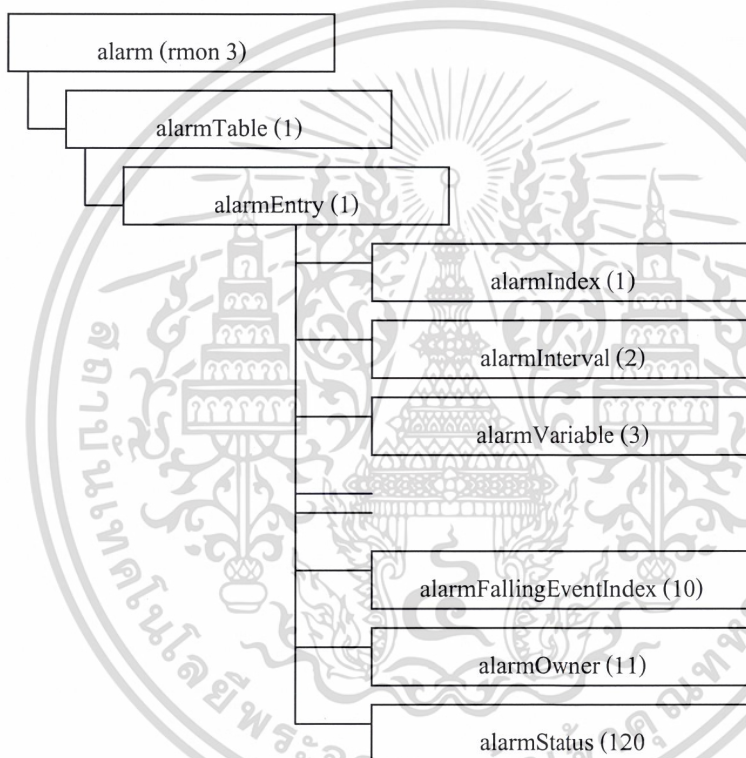
รูปที่ 4.3 แสดง RMON history Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.3 Alarm Group

Alarm Group ใช้ในการแสดงถึงการกำหนดกลุ่มหรือค่าของจุดวิกฤติขึ้นมา (set of threshold) สำหรับการดำเนินงานของเน็ตเวิร์ก (Network performance) ถ้าจุดวิกฤติที่กำหนดขึ้นนั้นเกิดการขัดแย้ง (Crossed) ขึ้นกับทิศทางที่ควรจะเป็น alarm ก็จะเกิดขึ้นและถูกส่งไปที่ส่วนควบคุมส่วนกลาง (control console) เช่น alarm จะเกิดขึ้นได้เมื่อมีค่ามากกว่า 500 CRCerrors (ซึ่งค่านี้เป็น threshold) ในทุก ๆ 5 นาที (เวลาในการสุ่มค่า)

Alarm Group ประกอบด้วย alarmTable ซึ่งประกอบด้วยหลาย entry และแต่ละ entry จะแสดงรายละเอียดเฉพาะอย่างที่ได้มีการตรวจดูมา (monitored) การสุ่มตัวอย่าง (sampling) ใน entry แต่ละครั้งนั้นเมื่อเสร็จสิ้นการ sampling จะมีการเปลี่ยนแปลงค่าของ parameter threshold ที่ได้ใหม่นั้นลงไปแทนค่าเก่าที่เก็บอยู่ทันที



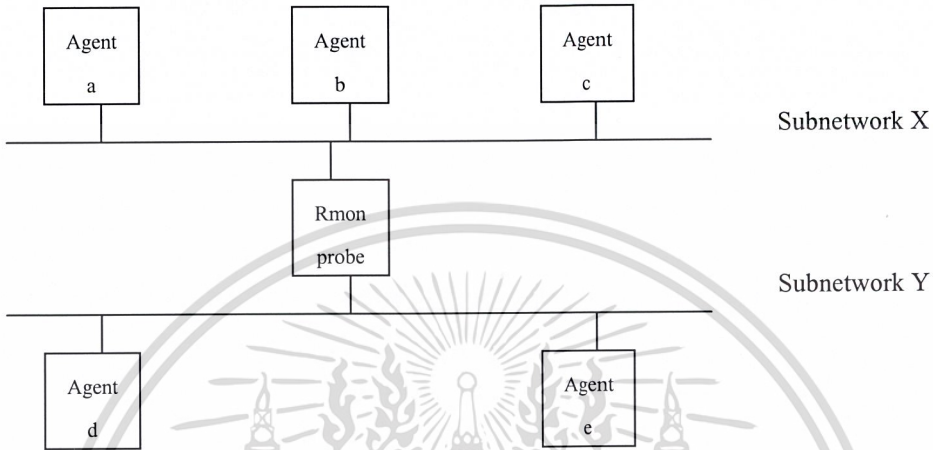
รูปที่ 4.4 แสดง RMON Alarm Group

Alarm มีรูปแบบการทำงานโดยตัว monitor หรือ management station นั้นสามารถที่จะกำหนด alarm แบบใหม่ออกมาได้โดยสร้างที่แฉวใหม่ใน alarmTable ซึ่งรวมถึงตัวแปร, การสุ่มตัวอย่างของค่าที่แตกต่างกันในกลุ่ม และ threshold parameter จะเป็นลักษณะพิเศษที่จัดอยู่ในแฉวเดียวกัน Threshold นั้นประกอบด้วย 2 แบบคือ Rising threshold และ Falling threshold ในแบบ Rising threshold นั้นจะเกิด crossed ขึ้นก็ต่อเมื่อค่าที่ทำการ sampling ออกมามีค่ามากกว่าหรือเท่ากับ rising threshold และค่าที่มาจากการ sampling ครั้งสุดท้ายนั้นน้อยกว่า threshold ในทำนองเดียวกัน falling threshold จะเกิด crossed ก็ต่อเมื่อค่าที่ sampling ออกมานั้นน้อยกว่าหรือเท่ากับ falling threshold และค่าที่ได้จาก sampling interval ครั้งสุดท้ายต้องมากกว่า threshold

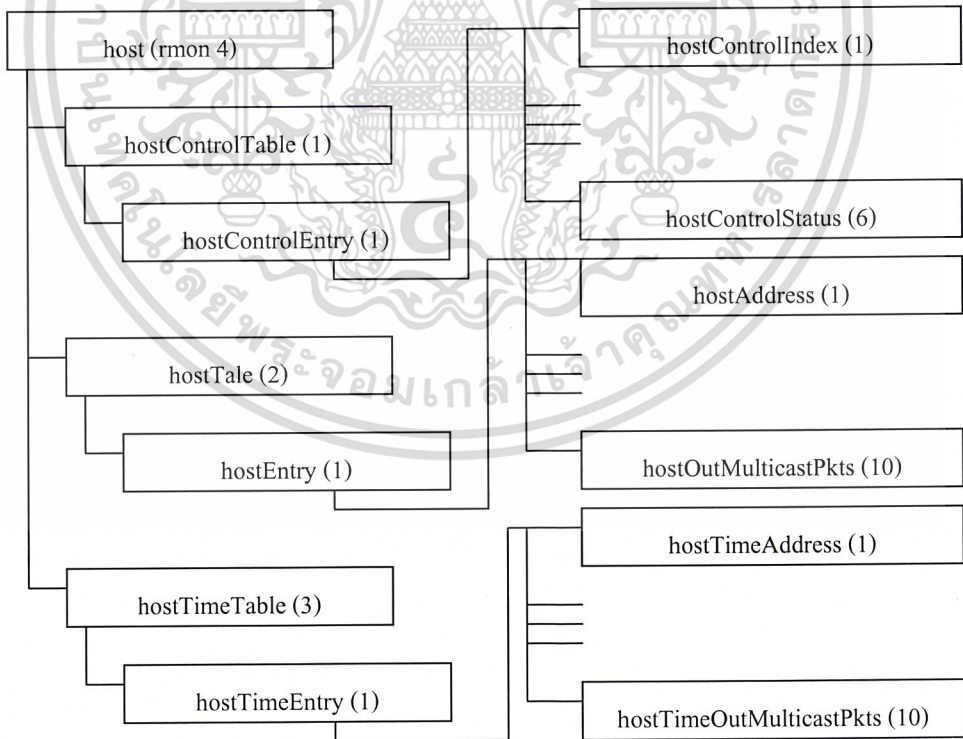
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 Host Group

host Group จะเก็บสถิติของแต่ละ host ที่ถูกค้นพบบนระบบเครือข่าย กลุ่มนี้จะค้นพบ host บนระบบเครือข่ายโดยการเก็บรายชื่อของ MAC Address ต้นทางและปลายทางที่ไปกับ good packets รูปที่ 7 แสดง RMON probe ก็กับการเชื่อมต่อของ 2 subnetwork, subnetwork X(interface#1 ; hostControlIndex =1) มี 3 host ซึ่ง monitor จะรู้จักทั้ง 3 host ส่วน subnetwork Y(interface#2 ; hostControlIndex =2) มี 2 host



รูปที่ 4.5 แสดงตัวอย่าง RMON Configuration

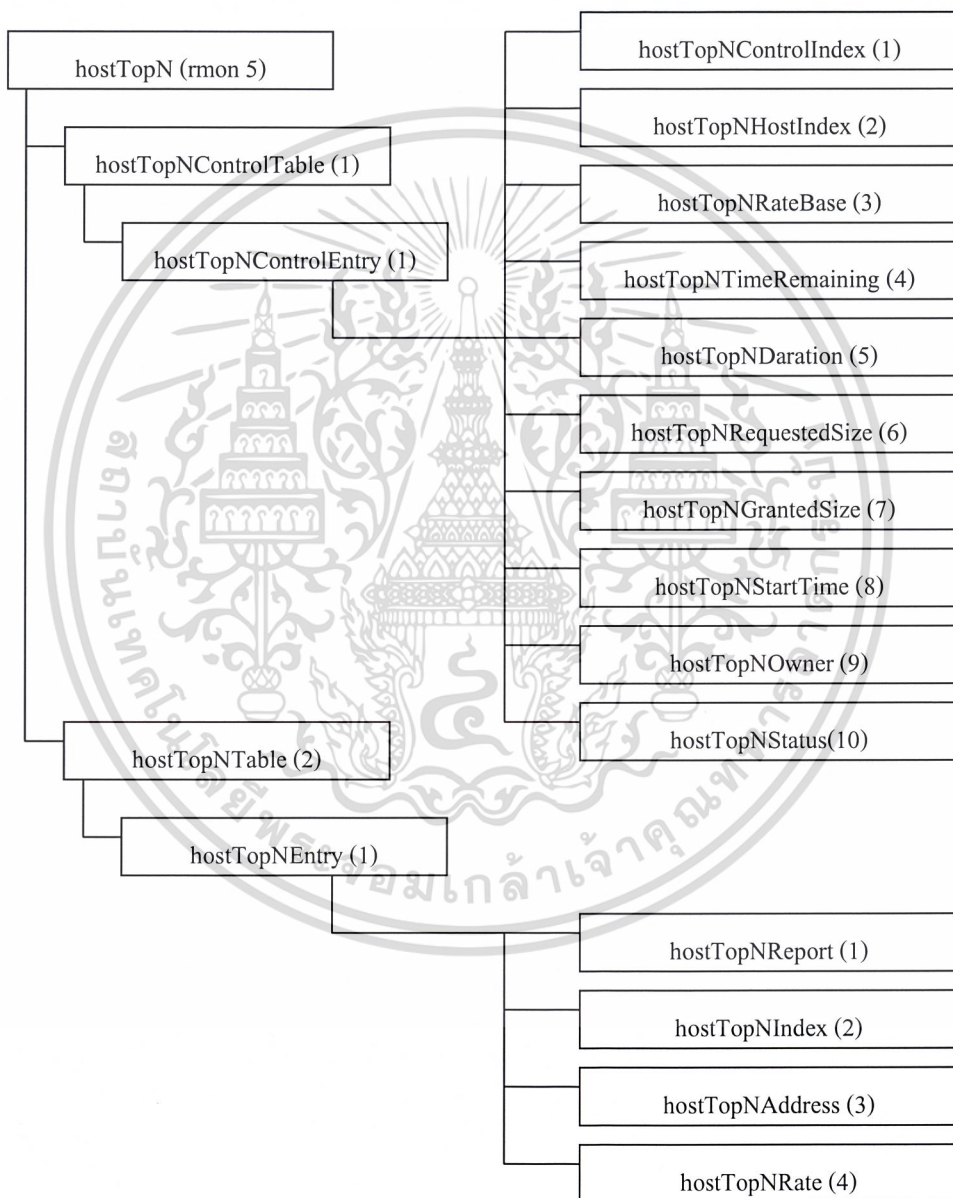


รูปที่ 4.6 แสดง RMON host Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.5 HostTopN Group

hostTopN Group ใช้ในการเก็บสถิติต่าง ๆ ที่สูงสุดตามลำดับของแต่ละ hosts บนระบบเครือข่าย ยกตัวอย่างเช่น เก็บสถิติของ 10 hosts ที่มีการส่งข้อมูลมากที่สุดในแต่ละวัน สถิติที่นำมาเปรียบเทียบกับกันได้มาจาก host group โดยจะทำการสุ่มตัวอย่างของสถิติในแต่ละ host ออกมาเก็บไว้เป็น report ซึ่งแต่ละ report จะเก็บผลลัพธ์เป็นตัวแปร (variable) ตัวแปรจะแสดงจำนวนของการเปลี่ยนแปลงใน host group, hostTopN group ประกอบด้วย หนึ่งตารางควบคุม (hostTopNControl Table) และหนึ่งตารางข้อมูล (hostTopNTable)



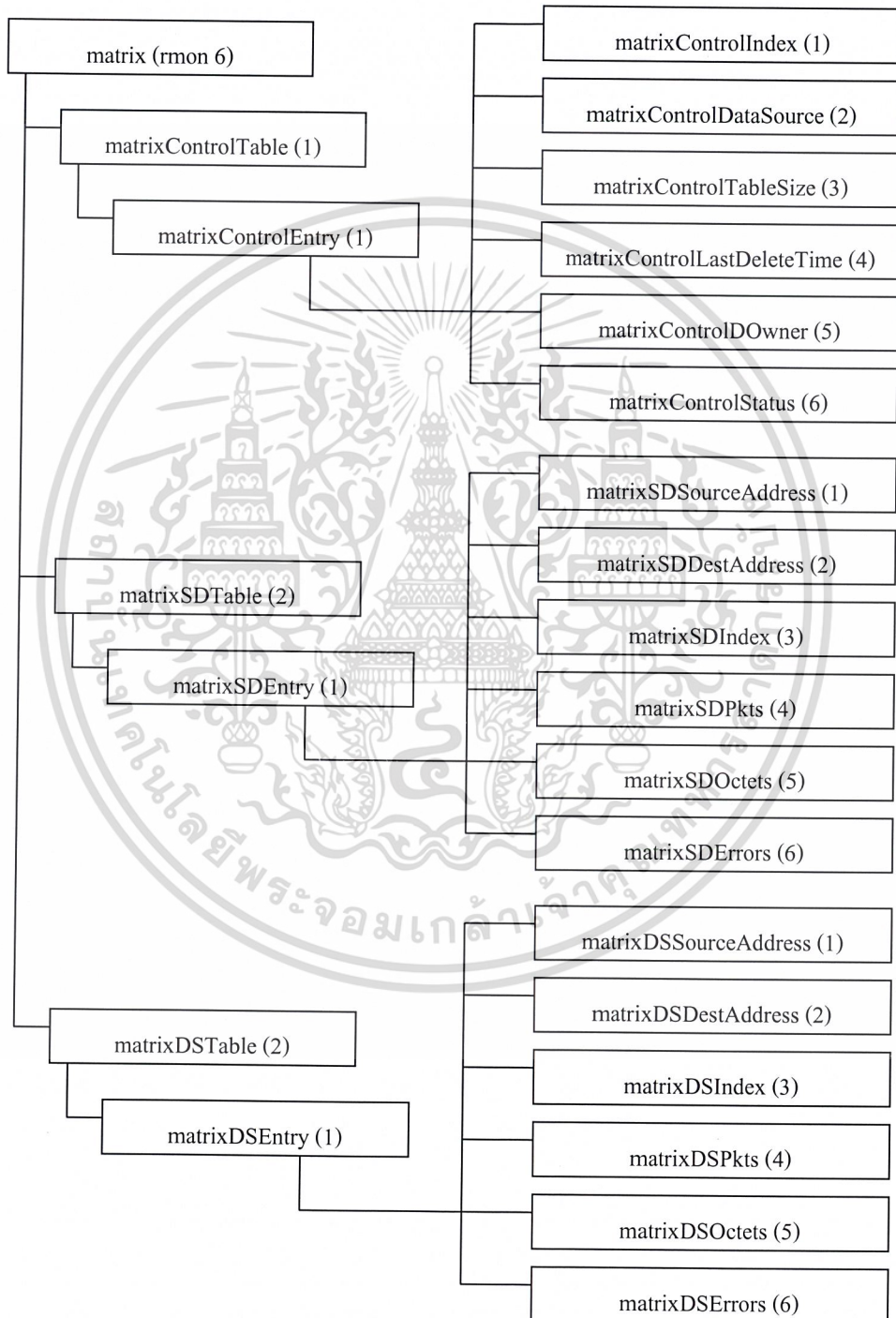
รูปที่ 4.7 RMON hostTopN Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.6 Matrix Group

Matrix Group ใช้เก็บข้อมูลข่าวสารเกี่ยวกับการจราจรระหว่างคู่ของ host บนระบบเครือข่าย โดยข้อมูลข่าวสารนี้ถูกเก็บอยู่ในรูปแบบของ matrix

กลุ่ม matrix ประกอบด้วย 3 ตารางคือ matrixControlTable (ตารางควบคุม), matrixSDTable และ matrixDSTable (ตารางข้อมูล)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 4.8 RMON matrix Group  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

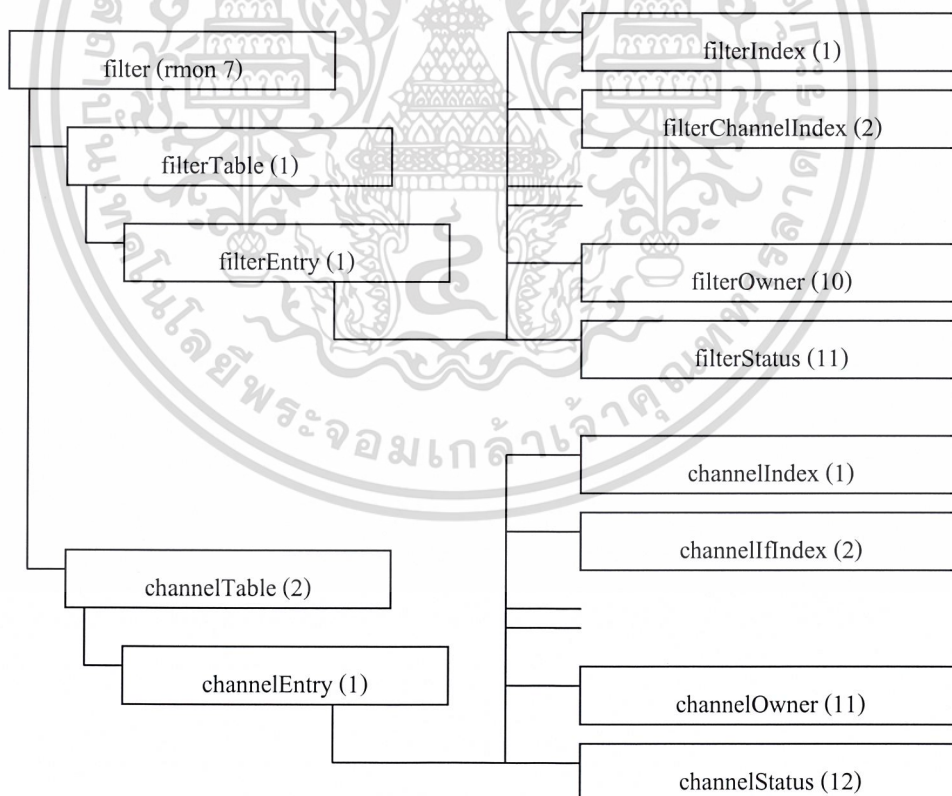
### 4.3.7 Filter Group

Filter Group เป็นการกำหนดให้ management station สามารถที่จะ monitor ในการดูและเลือก packet ใน subnetwork นั้น ๆ ซึ่ง filter group นี้จะมีการแบ่งออกเป็น 2 ชนิดคือ data filter และ status filter

- Data filter จะสามารถใช้ monitor ในการมองดู packet ในรูปแบบของ bit pattarn ซึ่งเป็นส่วนของ packet match
- Status filter จะสามารถใช้ monitor ในการมองดู packet ในรูปแบบของ status ต่าง ๆ เช่น valid, CRCerror

Filter สามารถที่จะใช้ร่วมกับลอจิก AND และ OR ในการทำ complex test ซึ่งสามารถที่จะประยุกต์เพิ่มเข้าไปใน packet การไหลของ packet จะผ่านการทดสอบโดยอ้างอิงกับ channel และจะมีการนับ packet นั้นไปเรื่อย ๆ การที่จะเพิ่ม event เข้าไปตัว channel นั้นจะมีโครงสร้างที่จะทำให้เกิด event ใด ๆ ซึ่งจะกำหนดอยู่ใน event group เมื่อ packet ผ่านมาถึง channel และ channel อยู่ในสถานะ enable และเมื่อ packet ผ่านมาที่ channel ที่ channel นั้นสามารถที่จะดึง packet นั้นมาได้ถ้า mechanism ถูกกำหนดไว้ใน capture group

Filter Group ประกอบด้วย 2 control table คือ chanelTable และ filterTable ซึ่ง channel แต่ละ channel นั้นจะมีลักษณะการทำงานเฉพาะอย่างและสามารถที่จะเกี่ยวข้องกับ filter ได้หลาย filter ที่อยู่ใน filterTable



รูปที่ 4.9 แสดง RMON Filter Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.8 Capture Group

Packet capture group สามารถที่จะใช้ในการ setup ตัว buffering scheme สำหรับการที่จะทำ capturing packet โดยมี 1 channel ที่อยู่ใน filter group ซึ่ง packet capture ประกอบด้วย 2 table คือ bufferControlTable ประกอบด้วยรายละเอียดของ buffering function และมี captureBufferTable โดยเป็น buffer ของ data

ในหนึ่ง row ที่มีอยู่ใน bufferControlTable กำหนดให้เป็น 1 buffer ซึ่งไว้ใช้ในการ capture และ store packet จาก 1 channel table นั้นประกอบด้วย object ต่าง ๆ ที่มีหน้าที่การทำงานที่แตกต่างกัน

เกี่ยวกับการ set ค่าของ parameter ส่วนมากของ packet จะถูกเก็บใน buffer และการส่งที่หาได้ส่วนมากก็คือการส่งถึง Management stations ใน 1 SNMP Get of GetNext request และเพื่อความสะดวกจึงมีการใช้ตัวย่อกันซึ่งมีดังนี้

CS = bufferControlCaptureSliceSize : จำนวนสูงสุดของ octets ในแต่ละ packet นั้นจะ save ไว้ใน Capture buffer

DS = bufferControlDownloadSliceSize : จำนวนสูงสุดของ octets ในแต่ละ packet ใน buffer นี้จะทำการส่งค่าใน SNMP ที่ดึงมาจาก packet กลับไป

DO = bufferControlDownloadOffset : offset ของ first octets ในแต่ละ packet ใน buffer นี้จะทำการส่งค่าใน SNMP ที่ดึงมาจาก packet กลับไป

PL = captureBufferPacketLength : ความยาวจริงของ packet

PDL = ความยาวของ captureBufferPacketData : การเก็บ packet data จริงสำหรับแกลวนั้น โดย captureBufferTable

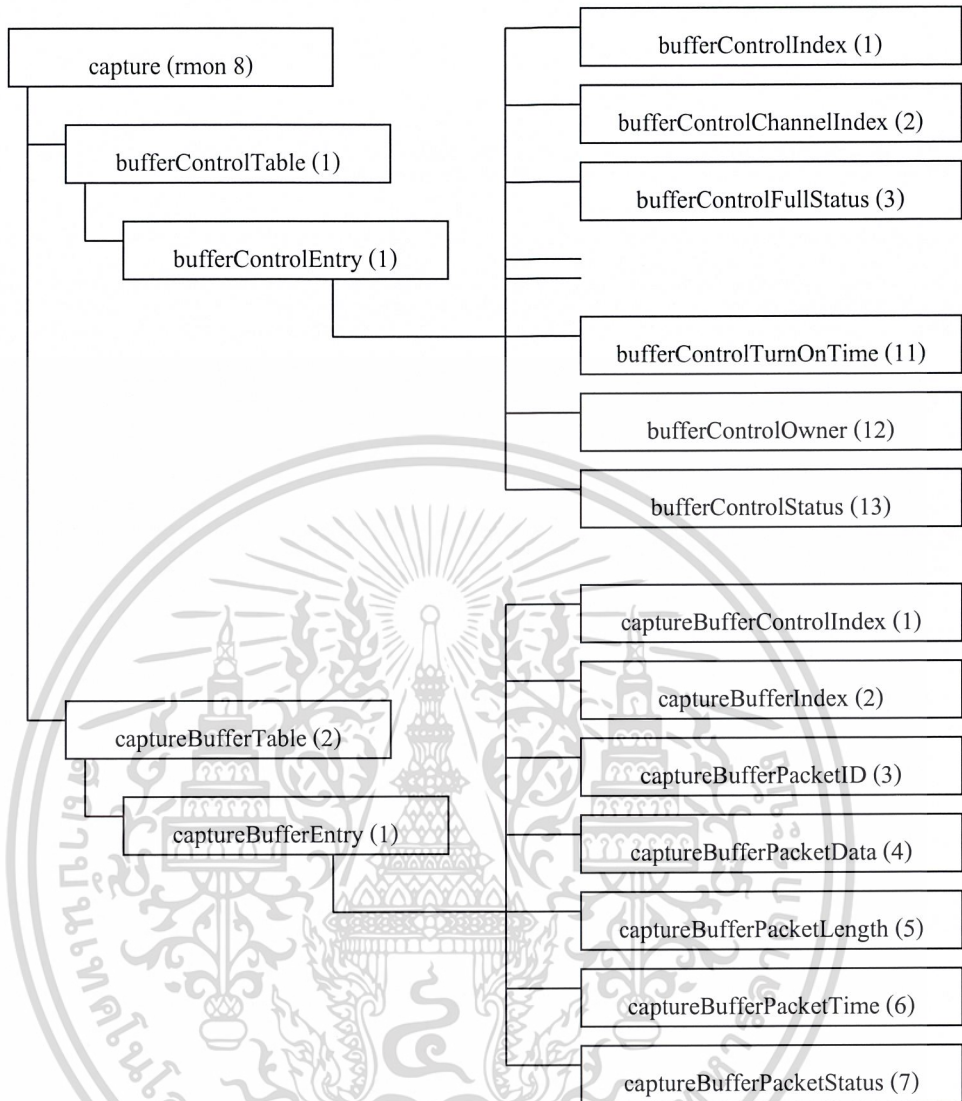
สมการแสดงการหาความยาวของ captureBufferPacketData (PDL)

$$PDL = \text{MIN} [PL, CS]$$

Packet (ถ้า  $PL < \text{หรือ} = CS$ ) หรือ packet slice (ถ้า  $PL > CS$ ) จะเป็นการเก็บ OCTET STRING อย่างเดียว 1 แกลวของ captureBufferTable อย่างไรก็ตาม OCTET STRING นี้จะดีถ้ามีความยาวพอ ๆ กับ SNMP message เดียวตัวพารามิเตอร์ DO และ DS เป็น tool ในการดึงขึ้นมาของ captured packet เป็นส่วน ๆ ถ้ามีการ set DO เป็น 0 และ DS เป็น 100, จะได้รับ captureBufferPacketData, จะได้รับ octets 100..MIN (actualStoreData 1,199); ถ้าที่ station มีข้อความขึ้นว่า “off the end of the packet” แสดงว่าจะได้รับความยาวของสตริงเป็น 0

ยกตัวอย่าง ตัว Management station สามารถที่จะทำการ set DO เป็น 0, DS เป็น 100 หรืออื่น ๆ คัดค้านั้นสามารถที่จะทำให้มีการผ่านของ packet ได้ตลอดทั้ง table, ทั้งที่ได้รับมาตลอด PL, ทั้ง 100 byte แรกของ packet และบางที่เป็น PacketStatus และส่งออกไป ดังนั้น station สามารถที่จะทำการ set ให้ DO ได้ถึง 100 และทำการส่งผ่านจนกระทั่งทั้งหมดของ capture data นั้นที่ทำการดึง packet นั้นเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดง RMON Packet capture Group

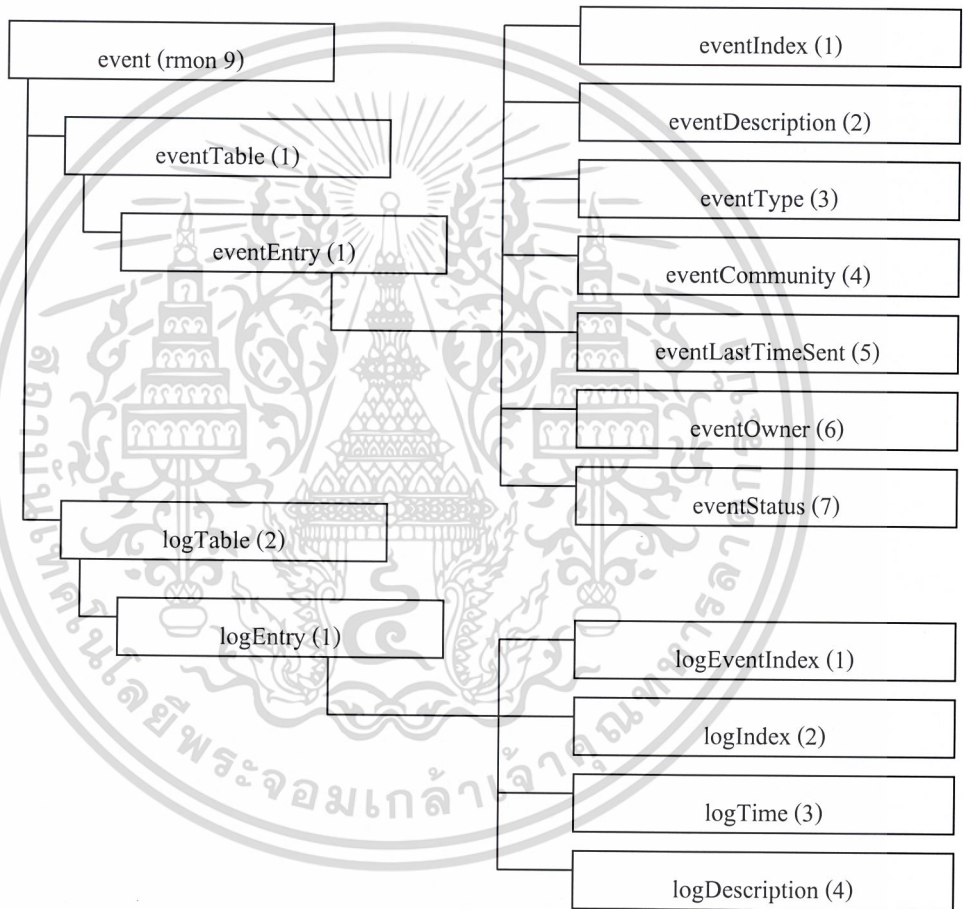
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.9 Event Group

Event Group จะรองรับต่อการกำหนดของ events ต่าง ๆ ที่เกิดขึ้น Event จะถูก trigger โดยเงื่อนไขที่ตั้งขึ้นใน MIB, event สามารถที่จะ trigger การกระทำต่าง ๆ โดยกำหนดไว้ใน MIB และ event จะเป็นตัวทำให้เกิด information ต่าง ๆ เข้ามาใน group นี้และเมื่อเกิด SNMP trap ตัว message ก็จะถูกส่งออกมา

ดังนั้น Event ที่ถูกกำหนดอยู่ใน group นี้สามารถที่จะใช้ trigger ให้ทำงานใน group อื่น ๆ ได้ event สามารถที่จะ trigger การเปลี่ยน on, off ของ channel ได้

Event group ประกอบด้วย control table และ data table ประกอบด้วย event ที่กำหนดขึ้น แต่ละแถวของ table ประกอบด้วย parameter ของ event ที่เกิดขึ้นเมื่อแน่ใจใน condition ที่พบใน event table นั้น ประกอบด้วย object หลายแบบ



รูปที่ 4.11 แสดง RMON Event Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.10 Token Ring Group

RFC 1513 เพิ่มกลุ่มใหม่ไปยัง RMON MIB คือกลุ่ม tokenRing ซึ่งกลุ่มนี้ประกอบไปด้วย 4 กลุ่มดังนี้

Token Ring Ring Station Group

Token Ring Ring Station Order Group

Token Ring Ring Station Configuratio Group

Token Ring Ring Source Routing Group

##### 4.3.10.1 Token Ring Ring Station Group

Token Ring Ring Station Group เก็บสถิติและสถานะของข้อมูลข่าวสาร ตั้งแต่เริ่มการ monitor ที่สัมพันธ์กับแต่ละระบบเครือข่าย token ring โดยกลุ่มนี้จะจัดหาสถานะของข้อมูลข่าวสารสำหรับ แต่ละวงแหวน (ring) ที่ถูก monitor กลุ่มนี้ประกอบไปด้วย 2 ตารางคือ ringStationControlTable และ ringStationTable

##### 4.3.10.2 Token Ring Ring Station Order Group

Token Ring Ring Station Order Group จัดลำดับของระบบเครือข่าย token ring ที่ถูก monitor กลุ่มนี้ประกอบด้วย 1 ตารางคือ ringStationOrderTable และหนึ่งทางเข้าสำหรับแต่ละระบบเครือข่าย token ring

##### 4.3.10.3 Token Ring Ring Station Configuration Group

Token Ring Ring Station Configuration Group จะบริหารระบบเครือข่าย token ring ตลอดจนการทำงานต่าง ๆ กลุ่มนี้มี 2 ตารางคือ ตารางควบคุม (ringStationConfigControlTable) และตารางข้อมูล (ringStationConfigTable)

##### 4.3.10.4 Token Ring Ring Source Routing Group

Token Ring Ring Source Routing Group จะเก็บการใช้ประโยชน์สถิติที่ได้รับมาจากการหาเส้นทางของข้อมูลข่าวสาร กลุ่มนี้มี 1 ตารางคือ sourceRoutingStatsTable

#### 4.4 Network Inventory

SNMP จะไม่สามารถที่จะใช้งานได้อย่างสมบูรณ์หรือเต็มประสิทธิภาพในการจัดเก็บทุก ๆ device ลงบน Internet ได้ในการที่จะเพิ่ม device ลงไปใน RMON จะเตรียมช่องทางที่สะดวกและดูแลรายการของ device ทั้งหมดใน network ไม่ว่าจะเป็นการ sending หรือ receiving packets of data การเฝ้าดูความหนาแน่นในการส่งหรือรับข้อมูลใน subnetwork ต่าง ๆ RMON monitor สามารถที่จะดูรายการที่มีอยู่ใน network ได้ทันทีในทุก ๆ device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 Hardware Platform

Platform ต่าง ๆ เป็นสิ่งจำเป็นที่ใช้ใน RMON monitor ซึ่ง SNMP ก็รองรับด้วย ดังนั้นจะมี logic ที่สนับสนุนในทั้งหมดของ RMON functionality และจะมีความสำคัญในตอนที่ทำการเพิ่ม device เข้าไป platform นั้นมีให้เลือกอย่างกว้างขวาง สามารถที่จะเป็น personal computer หรือ workstation ใดๆ อย่างหนึ่งก็สามารถที่จะเข้าถึง RMON function ได้ อีกทางหนึ่งคือ สามารถที่จะใช้ nondedicated host computer ในการติดต่อซึ่งกันและกันกับ device อย่างเช่น bridge หรือ router, หรือจะเป็น network management station ระดับเดียวกันได้

การที่จะเลือก platform เป็นแบบ dedicated หรือ nondedicated นั้นจะขึ้นอยู่กับขนาดและความซับซ้อนของ subnetwork ที่จะส่งไปให้ สำหรับ subnetwork ที่มี traffic ที่น้อยกว่า subnetwork อื่น ๆ และไม่ต้องการที่จะทำงานตลอดเวลา สามารถเลือก platform แบบ nondedicated ก็พอแล้วและยังช่วยประหยัดเงินอีกด้วย ตัวอย่างก็คือระบบ LAN สำหรับ network ทั่ว ๆ ทางเลือกที่ดีสำหรับ platform ที่สามารถติดต่อซึ่งกันและกันกับ device ก็คือ hub, router หรือ bridge ส่วน high-traffic subnetwork จะใช้ FDDI backbone และจะใช้ platform แบบ dedicated



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### ภาษาจาวา

ภาษาจาวา หรือ Java Programming Language ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ เริ่มแรกกันได้คิดค้นภาษาจาวาขึ้นมาเพื่อใช้กับระบบปฏิบัติการ Set Top Box คือโครงการของระบบตลาดอิเล็กทรอนิกส์ ที่ต้องการให้ระบบสามารถทำการควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพลตฟอร์ม (Platform) ที่ใช้ คือการพัฒนาระบบปฏิบัติการหรือ ซอฟต์แวร์ให้สามารถทำงานได้ โดยไม่สนใจว่าไมโครโพรเซสเซอร์ที่ใช้อยู่จะเป็นรุ่นใด ยี่ห้อไหน ฮาร์ดแวร์เป็นของอะไร ทั้งยังได้รวมความสามารถไปถึงการเชื่อมต่อ การเปลี่ยนข้อมูลข่าวสารต่างๆ ไปและทางด้านธุรกิจ ผ่านทางระบบเครือข่ายของคอมพิวเตอร์อีกด้วย

ทางทีมงานซึ่งพัฒนาโปรแกรมนี้ได้เล็งเห็นถึงการเจริญเติบโตและความสามารถในการแลกเปลี่ยนข่าวสารบนอินเทอร์เน็ตจึงได้วางแผนและพัฒนา ระบบ Interface แบบใหม่เพื่อใช้กับอินเทอร์เน็ต ขึ้น ด้วยสาเหตุนี้เองภาษาจาวาจึงได้บังเกิดขึ้น รวมทั้งยัง ได้มีการเปิดตัวเว็บเบราว์เซอร์ (Web Browser) ที่ชื่อว่า Hotjava เพื่อใช้สำหรับแสดงความสามารถของภาษาจาวา

ภาษาจาวาเป็นภาษาที่พัฒนาขึ้นมาเพื่อสามารถนำเอามาใช้งานในเครื่องชนิดอื่น ๆ เช่น เมื่อนำมาคอมไพล์ในเครื่องชนิดหนึ่ง จะได้ผลลัพธ์เป็น ไบต์โค้ด จากนั้นในการทำงานของภาษาจาวา จะใช้ตัวรัน ซึ่งมีชื่อเรียกย่อว่า JVM (Java virtual machine) จะทำหน้าที่ในการแปลงคำสั่งในลักษณะไบต์โค้ดให้กลายเป็นภาษาเครื่องของแต่ละเครื่องนั้น ดังนั้นในการทำงานต่างเครื่องคือ เมื่อมีการคอมไพล์เป็นไบต์โค้ดอีกเครื่องในการนำไปรันอีกเครื่องจำเป็นจะต้องมีตัวแปลงภาษา ให้เป็นภาษาเครื่องชนิดนั้น ๆ ด้วย

#### 5.1 ความสามารถของภาษาจาวา

ลักษณะของภาษา Java จะเป็นการเขียนโปรแกรมอ้างอิงเชิงวัตถุ OOP (Object Oriented Programming) สามารถสรุปความสามารถการทำงานได้ดังต่อไปนี้

1. การทำงานที่ไม่ขึ้นอยู่กับแพลตฟอร์ม โดยใช้วิธีการแปลงข้อมูลจากตัวคอมไพล์เลอร์ให้เป็นแปลงจาก source code ไปเป็น แบบ byte code ที่ภาษาจาวารู้จักสามารถทำงานได้
2. การใช้ Methods จัดการกับข้อมูลแทนการใช้ฟังก์ชัน
3. ความสามารถในการโต้ตอบแบบ InterActive กับงานด้านมัลติมีเดีย
4. ความสามารถในการดึงเอาคลาสไลบรารีจากที่ต่างๆ ผ่านทาง HTTP และ FTP
5. ความสามารถในการเล่นไฟล์เสียง ไฟล์วิดีโอและไฟล์ภาพกราฟิกเคลื่อนไหว
6. การโต้ตอบแบบทันต่อเหตุการณ์
7. เป็นระบบ แบ่งงานออกเป็นที่ทำออกเป็นส่วน ๆ คือสามารถทำงานได้หลายงานพร้อมกันใน

เวลาเดียว  
เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ไม่มีตัวแปรประเภทพอยน์เตอร์จึงหมดปัญหาเกี่ยวกับการขอใช้หน่วยความจำโดยตรง และสามารถทำการ คือค่าให้แก่หน่วยความจำได้อีกด้วยในกรณีที่ใช้งานตัวแปรนั้น เรียบร้อยแล้ว
9. มีความปลอดภัยสูง ด้วยความสามารถที่ไม่ให้ผู้ใช้เข้าไปยุ่งเกี่ยวกับหน่วยความจำของระบบ จึงปลอดภัยจากไวรัส
10. การดาวน์โหลดซอฟต์แวร์ใหม่ๆ และอัปเดตเวอร์ชันใหม่ได้อย่างง่ายดายโดยผ่านทางเว็บไซต์

## 5.2 การประยุกต์ใช้งาน

จากความสามารถดังที่กล่าวผ่านมาข้างต้นคงพอจะเป็นข้อสรุปได้แล้วว่า Java เป็นภาษาโปรแกรมที่มีความสามารถรอบด้าน อีกทั้งยังมีความยืดหยุ่นภายในตัวสูง การที่จะนำมาสร้างและพัฒนางานใดๆ มีประสิทธิภาพคงจะทำได้ไม่ยากนัก เราสามารถใช้ Java สร้างและพัฒนาแอปพลิเคชันต่างๆ โดยยกตัวอย่างพอสังเขปได้ดังนี้

### Client

Java สามารถเป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้าน Client ได้เพียงพอโดยเฉพาะด้านกราฟิก และระบบติดต่อกับผู้ใช้ เพราะปัจจุบัน โปรแกรมที่เขียนขึ้นจากภาษา Java จะทำงานที่ฝั่ง Client เป็นส่วนมากจะเห็นได้จาก Java Applet ต่างๆบน Web Browser แต่จะให้ดีและสมบูรณ์ยิ่งขึ้นต่อความเป็น Client Software ที่ดีก็คือ การติดต่อบริหารข้อมูลจาก Client ไปเป็น Server แม้ว่าในปัจจุบัน Java จะสามารถทำได้แล้วก็ตามแต่ก็ต้องใช้เครื่องมือชนิดอื่นเป็นตัวช่วย เช่น การเชื่อมต่อกับฐานข้อมูลแบบ ODBC (Object-Oriented Database Connectivity) ซึ่งการทำงานของทางฝั่ง Server ต้องใช้ภาษาคอมพิวเตอร์ชนิดอื่นช่วยเพื่อการติดต่อกับฐานข้อมูล

### Server

Java เมื่อใช้สร้างโปรแกรมประยุกต์ที่ติดตั้งบน Server นั้น Java จำเป็นที่จะต้องได้รับการปรับปรุงการทำงานอยู่สามอย่างคือ การเชื่อมต่อกับฐานข้อมูลโดยตรง การทำงานให้สอดคล้องกับอินเทอร์เน็ต และความเร็วของการรันต้องเพิ่มขึ้นเท่ากับโปรแกรมประยุกต์ทั่วไป

ระบบ JDBC (Java Database Connectivity) ได้เข้ามาแก้ปัญหาในส่วนของการทำงานติดต่อกับฐานข้อมูลโดยตรง โดยการร่วมมือกันของบริษัทซอฟต์แวร์ที่มีชื่อเสียงหลายบริษัท ได้สร้างมาตรฐานการเชื่อมต่อกับฐานข้อมูลของ Java ขึ้น แต่ใจขณะนี้ยังไม่เป็นที่รู้จักแพร่หลายนักสำหรับ JDBC คาดว่าในอนาคตอันใกล้นี้การเชื่อมต่อกับฐานข้อมูลจะทำได้ง่ายขึ้นจาก Client สู่ Server โดยใช้ Java

### Security

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Java ได้สร้างสภาพแวดล้อมเสมือน “Java Virtual Machine” ซึ่งจะอนุญาตให้เฉพาะ Java Applet หรือ Java Application ซึ่งเป็น Byte Code เท่านั้นที่สามารถรันได้โดยที่ Byte Code จะปราศจากไวรัสหรือส่วนที่จะทำอันตรายต่อระบบ อันเนื่องจาก Java Virtual Machine จะจำกัดสิทธิ์การเข้าใช้ทรัพยากรของ Byte Code ในอนาคตการรักษาความปลอดภัยอีกระบบหนึ่งที่กำลังถูกนำมาปรับใช้คือ “การเข้ารหัสข้อมูล” เพิ่มความปลอดภัยของข้อมูล

การใช้จาวาประยุกต์สร้างและพัฒนางานยังสามารถแบ่งเป็นประเภทตามลักษณะของกลุ่มงานได้ดังนี้

- งานด้านการศึกษา โดยจัดทำเป็นลักษณะสั่งการสอนคล้ายกับ CAI มีความสามารถในการโต้ตอบระหว่างผู้ใช้งานกับคอมพิวเตอร์
- ตอบสนองงานทางด้าน Web Page ทำให้งานสร้าง Web Page คุมีชีวิตชีวาขึ้น โดยการนำเอาความสามารถทางด้านมัลติมีเดียมาใช้อย่างเต็มที่
- สร้างสรรค์เครื่องมือพัฒนา (Software Developer tool kits)
- สร้างแอปพลิเคชันทางธุรกิจ มีความสามารถในการประมวลผลข้อมูลด้วยตัวเอง
- สร้างตัวจัดการบนอินเทอร์เน็ต มีความสามารถในการเรียกหาข้อมูลผ่านทางระบบเครือข่าย
- พัฒนาเกม จะมีความเป็นมัลติ คือสามารถเล่นได้หลายๆ คนพร้อมกันผ่านทางระบบเครือข่ายหลากหลายพื้นที่ไม่ว่าจะเล่นด้วยกัน ณ ตำแหน่งใด ทั่วทุกมุมโลก

### 5.3 Java Virtual Machine

Java Virtual Machine เป็นหลักการสร้างคอมพิวเตอร์จำลองของจาวา โดยการสมมติให้มีคอมพิวเตอร์อีกเครื่องหนึ่งขึ้นมาโดยเครื่องนี้จะใช้ในการคอมไพล์โปรแกรมภาษาจาวาทุกโปรแกรม เมื่อต้องการให้โปรแกรมภาษาจาวาไปทำงานบนคอมพิวเตอร์จริงๆ เครื่องใด เครื่องใด เราก็เพียงแค่สร้างตัวอินเทอร์พรีเตอร์ (Interpreter) ของคอมพิวเตอร์จำลองตัวนั้นๆ เครื่องนั้นๆ ภาษาจาวาทุกโปรแกรมก็จะสามารถทำงานบนระบบคอมพิวเตอร์นั้นๆ ได้ตามต้องการ ด้วยหลักการนี้ก็เป็นที่มาของคุณสมบัติของจาวา ที่ไม่ขึ้นกับแพลตฟอร์มและฮาร์ดแวร์ใดๆ หรือพอร์ตเทเบิล (Portable) เราอาจจะเรียก Java Virtual Machine สั้นๆ ว่า JVM ก็ได้

สำหรับเหตุผลที่ต้องมี JVM นี้ จริงๆ แล้วเป็นการกำหนดค่าขึ้นมา เพื่อเป็นคำจำกัดความเฉพาะในความคิดเท่านั้น เพื่อให้ นักพัฒนาจะได้ไม่ถูกบังคับให้ต้องสร้างตัวอินเทอร์พรีเตอร์ตามแนวทางแบบใดแบบหนึ่งโดยเฉพาะ แต่ตัวอินเทอร์พรีเตอร์ที่สร้างขึ้นตามข้อกำหนดดังกล่าว ไม่ว่าจะอยู่บนแพลตฟอร์มใด จะสามารถรันโปรแกรมที่เขียนขึ้นในภาษาจาวาได้ โดยให้ผลลัพธ์ออกมาเหมือนกัน

แต่ในข้อกำหนดของ JVM ก็มีกาให้นิยามที่ชัดเจนมากๆ เกี่ยวกับการออกแบบอินเทอร์พรีเตอร์ในหลายๆ ส่วน โดยเฉพาะอย่างยิ่งส่วนที่เกี่ยวข้องกับการกระจายโค้ดของจาวาไปใส่ไว้ในรูปแบบที่กำหนด ข้อกำหนดนี้ได้แก่ไวยากรณ์ของออบเจกต์และโอเปอเรนด์ พร้อมกับค่าประจำตัว การจัดโครงสร้างของโค้ด การจัดวางรูปแบบของออบเจกต์ของจาวา ข้อกำหนดต่างๆ เหล่านี้จะทำให้นักพัฒนาอินเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์พีเตอร์ทั้งหลาย สามารถจะสร้างอินเตอร์พีเตอร์ขึ้นมาใหม่บนแพลตฟอร์มใดๆ ก็ได้ ดังนั้น การพัฒนาภาษาจาวา จึงไม่ได้ถูกปิดกั้นอยู่กับระบบของซันซึ่งเป็นผู้คิดค้นภาษาจาวานี้ขึ้นมาเท่านั้น

จากที่กล่าวมาข้างต้นถึงคุณสมบัติของจาวาที่ไม่ยึดติดอยู่กับแพลตฟอร์มใดๆ ทำให้การทำงานกับระบบคอมพิวเตอร์ที่ทำงานแบบกระจาย (Distributed Computing) ได้รับความตอบสนองอย่างเหมาะสม นั่นคือ ทำให้เกิดความพอร์ดเทเบิลนั่นเอง

อย่างไรก็ตาม JVM ก็ยังมีข้อจำกัดอยู่ ข้อจำกัดของ JVM นั้นจะอยู่ที่ข้อจำกัดของการออกแบบตัวอินเตอร์พีเตอร์แทน เช่น การจำกัดค่าของโอเปอเรนด์ และขนาดของสแต็ค เป็นต้น ข้อกำหนดเหล่านี้หมายถึงว่า JVM สามารถจะอ้างถึงหน่วยความจำได้เฉพาะในห้วงแอดเดรสเท่าที่มีอยู่เท่านั้น

ข้อจำกัดภายในของตัว JVM เองนั้นมีห้วงแอดเดรสที่ใช้ได้ถึง 4 GB เพราะขนาดของความกว้างของการอ้างแอดเดรสเป็น 32 บิต Method ต่างๆ ของจาวามีขนาดได้เพียง 32 KB เพราะมีข้อจำกัดของการใช้คำสั่งกระโดด เป็นแบบ 16 บิต (โดยมีบิตแรกเป็นตัวบอกว่ากระโดดไปข้างหน้า หรือข้างหลัง และบิตต่อไป บอกระยะทางการกระโดดจากจุดที่ล่่านี่มีขนาดเพียง 8 บิต นอกจากนั้น จำนวนของค่าคงที่ที่อยู่ในส่วนกลาง ยังถูกจำกัดด้วยขนาดของดัชนี 16 บิต ทำให้มีจำนวนได้เพียง 32,000 ค่าในแต่ละ method

การที่เราถือว่า การมีค่าเหล่านี้ เป็นข้อจำกัด ก็อาจจะเป็นการมองการณ์ไกลไปสักหน่อย เนื่องจากปัจจุบันเครื่องคอมพิวเตอร์ส่วนใหญ่มีหน่วยความจำจำกัดเพียง 16 หรือ 32 MB กันเท่านั้น หน่วยความจำขนาด 4 GB จึงยังเป็นเรื่องที่ไม่ต้องคิดหนักในตอนี้ ในขณะที่ขนาดของ method ที่จำกัดที่ 32 KB ก็เป็นเพียง method เดียวเท่านั้น

#### 5.4 หลักการทำงานของภาษาจาวา

ในที่นี้ขออธิบายการทำงานของภาษาจาวาเริ่มตั้งแต่การคอมไพล์ตัวโปรแกรมจนกระทั่งเราเรียกใช้งานในเว็บเพจ เป็นดังนี้

##### การคอมไพล์

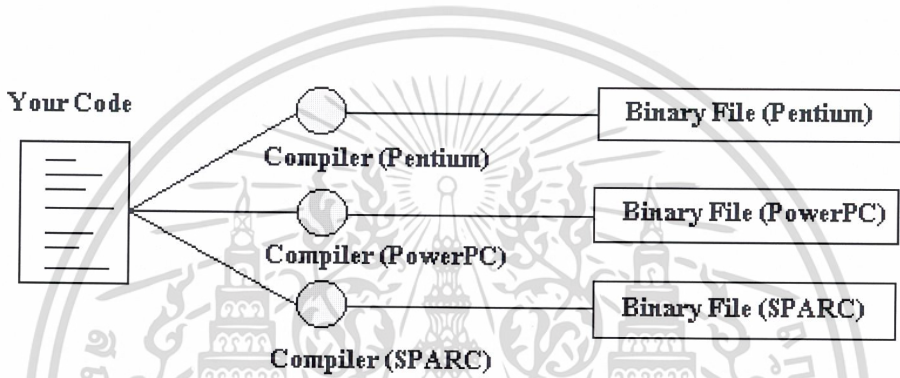
คอมไพเลอร์ของภาษาจาวาก็เช่นเดียวกับคอมไพเลอร์ในภาษาอื่นๆ นั่นคือ มันจะสร้างรหัสภาษาเครื่อง (Machine Code หรือ Assembler Code) จากภาษาในระดับที่สูงกว่าเพื่อให้ซีพียู (CPU : Central Processing Unit) สามารถนำไปใช้งานได้

แต่ข้อแตกต่างที่สำคัญระหว่างคอมไพเลอร์ของภาษาจาวากับภาษาอื่นๆ คือ ซีพียูหรือโปรเซสเซอร์ที่จะทำหน้าที่ในการปฏิบัติตามคำสั่งที่ได้จากการคอมไพล์ภาษาจาวานั้นไม่มีอยู่จริง เป็นเพียงสิ่งที่สมมติขึ้นมาที่เรียกว่า Java Virtual Machine นอกจากนี้การอ้างถึงส่วนต่างๆ ของโปรแกรมที่คอมไพล์ด้วยคอมไพเลอร์ของภาษาจาวาก็จะมีวิธีการที่แตกต่างออกไป

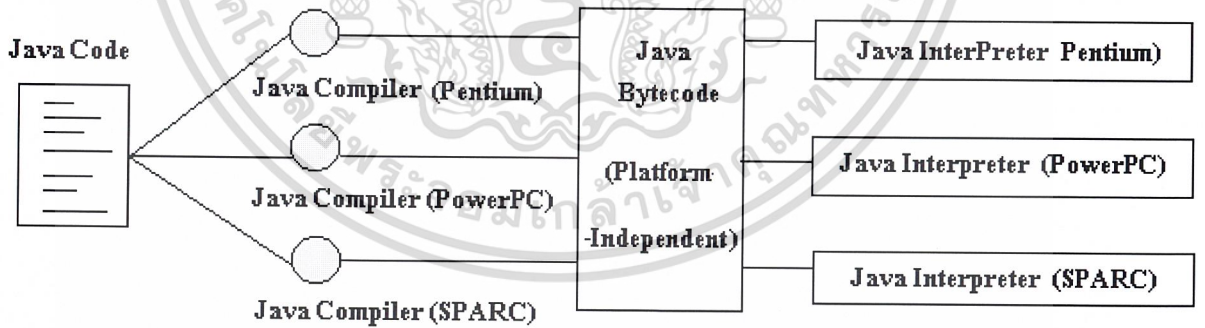
คอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนการอ้างถึงส่วนของโปรแกรมจากการ ใช้ชื่อแบบในภาษาสูง ไปเป็นตัวเลขเหมือนคอมไพเลอร์ภาษาอื่นๆ ทำกัน และคอมไพเลอร์ภาษาจาวาก็จะไม่มีการสร้างแผนที่ของการจัดวางโปรแกรมบนหน่วย ความจำขึ้นมาในระหว่างการคอมไพล์ด้วยเหตุผลที่สำคัญคือ เพื่อเป็นการสร้างความพอร์ดเทเบิลให้กับตัวโปรแกรมเพราะการจัดวางตำแหน่งของโปรแกรมจะต้องไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นอยู่กับลักษณะการทำงานของโปรเซสเซอร์ตัวใดตัวหนึ่ง การยังไม่จัดวางตำแหน่งช่วยให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลาง สามารถนำไปใช้บนคอมพิวเตอร์แพลตฟอร์มอื่นๆ ได้ นอกจากนี้ยังทำให้เกิดความปลอดภัยอีกด้วย

สิ่งที่ได้จากการคอมไพล์ในภาษาจาวาเราเรียกว่า ไบท์โค้ด (ByteCode)



รูปที่ 5.1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน



รูปที่ 5. 2 การประมวลผลของโปรแกรมภาษา Java

**การวางตำแหน่งในหน่วยความจำ**

ในภาษาจาวาจะไม่มีกรลดรูปแบบการอ้างถึงส่วนต่างๆ ของโปรแกรมจากการเรียกเป็นชื่อให้เหลือเพียงตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งของโปรแกรมลงในหน่วยความจำ คอมไพเลอร์ภาษาจาวาจะทิ้งชื่อของแต่ละส่วนของโปรแกรม (โดยเฉพาะ method) เอาไว้ในตัวโปรแกรมการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สร้างขึ้น เมื่อโปรแกรมทำงาน จะเป็นหน้าที่ของตัว อินเทอร์เน็ตที่คอยเปิดตารางค้นหาที่อยู่ของ method ที่ต้องการเรียกใช้งาน โดยก่อนที่จะเริ่มทำงานจริง อินเทอร์เน็ตจะต้องสร้างแผนที่ในการจัดวางสิ่งต่างๆ ลงในหน่วยความจำขึ้นมาแล้วจึงสร้างตารางขึ้นมา เพื่อช่วยหาตำแหน่งของ method เมื่อมีการเรียกใช้งาน โดยใช้ชื่อของ method

### การรันโปรแกรม

การรันโค้ดที่คอมไพล์เอาไว้สำหรับ Java Virtual Machine เป็นหน้าที่ของตัวอินเทอร์เน็ต การรันโปรแกรมจะแบ่งได้เป็น 3 ขั้นตอนหลักๆ คือ การอ่าน การตรวจสอบความถูกต้อง และการทำตามโค้ด หน้าที่ในการอ่านโค้ดเข้าสู่ระบบจะเป็นของ Class Loader หน้าที่การทำงานในส่วนนี้จะไม่ได้เข้ามาเฉพาะๆ ปล่อยให้กำลังจะเรียกใช้เท่านั้น แต่จะอ่านคลาสที่มีการอ้างอิงและคลาสที่มีการ inherited มาโดยคลาสที่อ้างอิง เมื่อผ่านขั้นตอนนี้แล้ว โค้ดทั้งหมดก็จะถูกส่งผ่านตัวตรวจสอบไบนารีโค้ดเพื่อให้แน่ใจว่าโค้ดที่ส่งมามีความถูกต้องตามมาตรฐานของจาวา และจะไม่รบกวนเสถียรภาพของระบบ เมื่อผ่านการตรวจสอบแล้วโค้ดก็จะถูกส่งต่อไปยังระบบรันใหม่ (Run-Time System) ซึ่งจะสั่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่งซึ่งหลักการทำงานในแต่ละขั้นตอนที่กล่าวมาข้างต้นเป็นดังนี้

### Class Loader

ตัวคลาสโหลดเดอร์ ทำหน้าที่ดึงโค้ดทั้งหมดที่จำเป็นในการทำงานของแอปพลิเคชัน ไม่ว่าจะ เป็นคลาสที่ถูก Inherited มาหรือคลาสอื่นๆ ที่มีการเรียกใช้ เมื่อโหลดเดอร์ดึงคลาสใดเข้ามาแล้วก็จะจัด คลาสนั้นๆ ใส่เข้าไปใน namespace ของมันเอง โดยการเก็บ จะใช้ชื่อของคลาสเป็นสำคัญ ไม่ได้ใช้การ อ้างอิงเป็นตัวเลข หลักการนี้ก็เหมือนกันกับการทำงานของ Virtual Machine ที่โอเอส (OS : Operating System) สร้างขึ้นให้แอปพลิเคชันแต่ละตัวทำงาน ถ้าไม่ได้มีการเจาะจงเรียกใช้คลาสที่อยู่นอก namespace นี้ การเรียกใช้ชื่อต่างๆ ในคลาส ก็จะไม่มีการรบกวนกันระหว่างคลาสเลย

คลาสทั้งหมดที่อยู่บนเครื่องโลคอลเอง จะได้รับห้วงแอดเดรส (Address Space) เป็นของตัวเอง ส่วนคลาสต่างๆ ที่ดึงมาจากภายนอกจะได้รับ namespace เป็นของตัวเอง การทำงานลักษณะนี้จะช่วยให้ คลาสที่อยู่บนโลคอลทำงานได้ประสิทธิภาพดีขึ้น เพราะใช้ namespace ร่วมกันได้ แต่ก็ยังมีการป้องกัน ความผิดพลาดที่อาจจะเกิดจากคลาสที่ดึงเข้ามาจากภายนอก และในทางกลับกัน คลาสที่อิมพอร์ตเข้ามา ก็ ปลอดภัยจากความผิดพลาดที่อาจจะเกิดขึ้นจากคลาสโลคอลด้วย

เมื่อคลาสทั้งหมดที่เกี่ยวข้องกับการทำงานถูกอิมพอร์ตเข้ามาเรียบร้อยแล้ว การจัดวางหน่วย ความจำสำหรับเริ่มต้นการทำงานก็จะเกิดขึ้นได้ การเรียกชื่อต่างๆ ก็จะสามารถจับคู่กับแอดเดรสจริงๆ ของหน่วยความจำได้ แล้วตัวโหลดเดอร์จะสร้างตารางสำหรับค้นหาที่อยู่เสี่ยงจากการทำงานผิดพลาดของ ซุปเปอร์คลาส (Super class) และการอ้างอิงแอดเดรสที่ไม่ถูกต้องได้

### ตรวจสอบไบนารีโค้ด

เมื่อโค้ดเดินทางมาจนถึงขั้นตอนการสร้างตารางจับคู่ชื่อกับแอดเดรสแล้ว ก็ยังไม่สามารถแน่ใจ ได้ว่าโค้ดที่อ่านเข้ามาจะมีความปลอดภัย ดังนั้นจึงต้องมีตัว Verifier หรือตัวตรวจสอบไบนารีโค้ด ทำหน้าที่ ตรวจสอบความถูกต้องที่ละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวา และสอดคล้องกับการทำงานของ ตัวโปรแกรมหรือไม่ การตรวจสอบไบนารีโค้ดในเชิงทฤษฎีจะสร้างค้นหาปัญหาต่างๆ ได้หลายอย่าง เช่น ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะไม่มีการสร้างพอยต์เตอร์ที่เกินกว่าหน่วยความจำจริง ไม่มีคำสั่งใดสามารถละเมิดสิทธิการทำงานของตัวโปรแกรมได้ ไม่มีการจับคู่อบเจ็กต์ผิด จะไม่มีการให้โอเปอร์เรนด์มากหรือน้อยเกินไป การกำหนดค่าต่างๆ สำหรับไบท์โค้ดจะต้องถูกต้องครบถ้วน และจะไม่มีการแปลงข้อมูลผิดรูปแบบ

การใช้ตัวตรวจสอบตอบสนองจุดประสงค์ 2 ประการสำคัญ คือ สิ่งต่างๆ ดังที่กล่าวมาแล้วจะถูกตรวจสอบก่อนทำให้ตัวอินเทอร์พรีเตอร์มั่นใจได้ว่า ไบท์โค้ดที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบ และจุดประสงค์ที่สองก็คือ ตัวอินเทอร์พรีเตอร์จะทำงานตามไบท์โค้ดได้รวดเร็วกว่า เพราะไม่ต้องคอยระวังว่าจะมีปัญหาก่อขึ้น และไม่ต้องหยุดเป็นช่วงๆ เมื่อพบปัญหาและต้องแก้ไข

ในการทำงาน ไบท์โค้ดจะถูกตรวจสอบเพียงครั้งเดียวเท่านั้น และจะทำงานไปได้ตลอดไม่ต้องมีการตรวจสอบซ้ำอีกเมื่อมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

### การทำงานตามโค้ด

เมื่อตัวโหนดเคอร์ได้รวบรวมโค้ดเข้ามาสู่ระบบทำการจัดวางในหน่วยความจำ และตัวตรวจสอบได้ทำการตรวจสอบความถูกต้องแล้ว โค้ดก็จะถูกส่งต่อไปยังตัวอินเทอร์พรีเตอร์เพื่อทำงานตามคำสั่ง การทำงานตามคำสั่งของโค้ดก็คือการเปลี่ยนโค้ดให้กลายเป็นคำสั่ง การทำงานจริงที่ตัวระบบโคลเอนท์ที่รันโค้ดนี้สามารถทำงานได้ ซึ่งวิธีการที่ทำได้ก็มีอยู่ 2 วิธีด้วยกันคือ ตัวอินเทอร์พรีเตอร์ทำการคอมไพล์โค้ดเหล่านี้ให้กลายเป็นเนทีฟโค้ดที่ตัวเครื่องโคลเอนท์เข้าใจ แล้วค่อยทำงาน เพื่อให้ได้ความเร็วสูงสุดในการทำงาน กับอีกวิธีหนึ่งก็คือ ตัวอินเทอร์พรีเตอร์อ่านโค้ดเข้ามาแล้วตีความทำงานไปที่ละคำสั่ง และทำการตีความไปเรื่อยๆ ตลอดเวลาที่มีการทำงาน

โดยปกติแล้ว ผู้สร้างตัวอินเทอร์พรีเตอร์มักจะเลือกใช้วิธีการหลัง รูปแบบของไบท์โค้ดในภาษาจาวามีความยืดหยุ่นเพียงพอที่จะสามารถเปลี่ยนไปทำงานบนเครื่องโคลเอนท์แบบต่างๆ ได้โดยไม่มีผลกระทบต่อโอเวอร์เฮดมากมายเกินความจำเป็น อย่างไรก็ตามโคลเอนท์ของจาวาบางระบบจะมีความสามารถในการทำงานได้ทั้งสองวิธี คือ โปรแกรมเมอร์สามารถจะเลือกใช้วิธีการคอมไพล์กับงานที่เน้นการคำนวณมากๆ เพื่อเป็นการเพิ่มสมรรถนะในการทำงานให้ได้เต็มที่ ซึ่งโคลเอนท์แบบนี้จะให้ทั้งความพอร์ตเทเบิล และสมรรถนะที่ดี

การสร้างระบบรันไทม์ที่ดีจะต้องถ่วงดุลย์ความสำคัญ 3 ประการให้ได้พอเหมาะ นั่นคือ ความพอร์ตเทเบิล ความปลอดภัย และสมรรถนะ เรื่องของความพอร์ตเทเบิลทำได้โดยการใช้รูปแบบของไบท์โค้ดที่มีความเป็นกลางเพียงพอ สามารถนำไปใช้รันบนเครื่องคอมพิวเตอร์แบบต่างๆ ได้โดยง่าย นอกจากนั้น การที่ตัวอินเทอร์พรีเตอร์ทำการกำหนดการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์ (แทนที่จะเป็นระหว่างการคอมไพล์เหมือนภาษาอื่นๆ) ก็เป็นการเพิ่มความแน่นอนว่า คลาสต่างๆ ที่อิมพอร์ตเข้ามาจะยังคงใช้ได้อยู่ตลอดเวลา เรื่องของความปลอดภัยนั้น เป็นสิ่งที่มีการคำนึงถึงอยู่ตลอดกระบวนการทำงานของระบบรันไทม์ โดยเฉพาะอย่างยิ่งในส่วนของตัวตรวจสอบไบท์โค้ด ที่ทำให้แน่ใจได้ว่าโปรแกรมจะทำงานได้ถูกต้องตามข้อกำหนดของจาวา ส่วนเรื่องของสมรรถนะนั้นก็สามารถจัดการได้ในสองระยะ คือ พยายามเอาโอเวอร์เฮดทั้งหลายไปใส่ไว้ที่ตอนเริ่มต้น โหนดโปรแกรมเข้ามาสู่ระบบ หรือไม่ก็กำหนดให้ทำงานเป็นแบบแบ็กกราวนด์ (Back-Ground Thread)

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยสิ่งต่างๆ เหล่านี้ ทำให้จาวาสามารถปล่อยสมรรถนะระดับที่น่าพอใจออกมาได้โดยที่ยังคงไว้ซึ่งความพอร์ตเทเบิล และสภาพแวดล้อมที่ปลอดภัย นอกจากนี้ ยังสามารถจะดึงสมรรถนะระดับสูงสุดออกมาใช้ได้ทันทีเมื่อต้องการ

### การสร้างและรันภาษาจาวา

การสร้างโปรแกรมจาก Java ก็เหมือนกับภาษา Programming อื่นๆ คือเขียน Code โปรแกรมจาก Editor ใดๆ ก็ได้ บันทึกลงในนามสกุล .java จากนั้นจึงนำไปคอมไพล์ด้วยคอมไพเลอร์ของ Java จะได้ไฟล์ใหม่ในนามสกุล .class ที่เก็บรหัสของการคอมไพล์ไว้จาวาเรียกรูปแบบข้อมูลที่อยู่ในไฟล์ใหม่นี้ว่า Byte Code ไฟล์ที่ได้คือแอปเพล็ตนั่นเองแอปเพล็ตยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่เรารู้จักกันเพราะข้อมูลแบบ Byte-Code จะมีรูปแบบข้อมูลที่อยู่ที่กึ่งกลางระหว่างโค้ดโปรแกรม (Source Code) กับโค้ดที่คอมพิวเตอร์อ่านแล้วนำไปปฏิบัติงานได้ทันที (Machine Code) หากจะรันแอปเพล็ตบนระบบใดๆ จะต้องใช้อินเตอร์พรีเตอร์จาวาของระบบนั้นๆ เพื่อตรวจสอบความถูกต้องของ Byte-Code แล้วแปลให้เป็นรหัสภาษาเครื่องส่งให้ระบบปฏิบัติการนำไปรันต่อไปขั้นตอนการทำงาน

## 5.5 WEB BROWSER SUPPORT JAVA

HTML (Hypertext Markup Language) เป็นภาษาที่มีรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบน Internet เป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งของคอมพิวเตอร์มีลักษณะของโปรแกรมเป็นไฟล์ตัวอักษรในมาตรฐานของรหัสแอสกี ประกอบด้วย Reserve Markup Directive หรือ คำสั่งต่างๆ ซึ่งมีลักษณะจะเป็นคำสั่งที่ง่ายต่อการทำความเข้าใจ เพื่อให้การสร้างเว็บเพจมีความสะดวกและง่ายมากยิ่งขึ้น ทั้งนี้เพื่อส่งผ่าน โครงสร้างของข้อมูลระหว่างผู้ใช้

ในปัจจุบันภาษา HTML ได้ถูกกำหนดมาตรฐานขึ้นมาเป็นรุ่นที่ 3.0 HyperText ที่ถูกสร้างขึ้นมา จะอยู่ในรูปของแฟ้มเอกสาร HTML (Document File) ที่มีการกำหนดคุณสมบัติ (markup) ของเว็บเพจเข้าไป ยังไม่สามารถแสดงผลข้อมูลออกมาให้ใช้งานได้โดยตรง ถ้าต้องการดูผลลัพธ์ที่เกิดจากการสร้างต้องผ่านโปรแกรมที่ทำหน้าที่แปลคำสั่งนั้นก่อน เราเรียกโปรแกรมที่ทำหน้าที่นี้ว่า “โปรแกรมเว็บเบราว์เซอร์” (Web Browser Program) หน้าที่หลักของโปรแกรมนี้อ็คือเป็นตัวแปลคำสั่งของ HyperText แล้วแสดงผลออกมาเป็นรูปภาพ เสียงข่าวสารและข้อมูล และคุณสมบัติอื่นๆ ของโปรแกรมเว็บเบราว์เซอร์ ไม่ว่าจะเป็นการดาวน์โหลดไฟล์ การดึงรูปภาพมาใช้งาน การพิมพ์เอกสาร HTML ออกทางเครื่องพิมพ์ การส่งจดหมาย (e-mail) และความสามารถด้านอื่นๆ อีกมากมาย ทำให้เราสามารถใช้งาน Internet หรือ Intranet ได้อย่างสะดวก รวดเร็ว และสนุกสนาน ซึ่งทีมพัฒนาภาษา JAVA ได้วางแผนและพัฒนาระบบ Interface โดยเล็งเห็นถึงการเจริญเติบโตและความสามารถของภาษา JAVA จึงได้เปิดตัว WebBrowser มีชื่อว่า Hotjava เพื่อใช้สำหรับแสดงศักยภาพของ Java ที่มีอยู่อย่างเต็มที่ราวปี 2538 เพราะว่ในขณะนั้นยังไม่มี Browser ตัวใดที่รู้จักและสามารถทำงานร่วมกับภาษา JAVA ได้เลย และแน่นอนว่า Browser ที่ได้รับการพัฒนาต่อๆ มาย่อมจะมีการเพิ่มเติมความสามารถที่ดีกว่า HotJava อย่างแน่นอนคือ Netscape Navigator เป็น Browser ที่บริษัท Netscape Communications ได้สร้างและเป็นบริษัทแรกที่ได้ลิขสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติเท่านั้น ไม่สามารถนำไปใช้เพื่อการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา JAVA จากบริษัท Sun Web Browser โดย Netscape เวอร์ชันแรกที่ Support คือ Netscape 3.0 เพื่อใช้สำหรับการทำงานบน Windows3.X, Windows95, WindowsNT, Solaris และ Apple Macintosh และยังมี Web Browser ของอีกบริษัทหนึ่งที่มีชื่อเสียงไม่แพ้ Netscape คือบริษัท Microsoft ซึ่งได้ออก Web Browser ที่ Support Java ในชื่อ Internet Explorer 3.0 เพื่อให้ Web Browser ที่บริษัทพัฒนาเป็นเวอร์ชันล่าสุดในขณะนั้น สามารถรองรับการออกแบบ Homepage ที่ถูกออกแบบด้วยภาษาจาวา

ปัจจุบันนี้มีบริษัทที่สร้าง Web Browser อื่นๆ อีกเช่น AOL (America online), IBM ซึ่งแน่นอนว่าปัจจุบันนี้ Web Browser เหล่านั้นสามารถ Support กับ Homepage ในเครือข่าย Internet ที่ใช้ภาษา Java แน่แน่นอนเพราะปัจจุบันได้มีการนำ Java ไปประยุกต์ใช้งานในงานด้านต่างๆ ทางธุรกิจ การศึกษา บันเทิง ฯลฯ อย่างมากมาย

## 5.6 Java Tool

การศึกษาและพัฒนาโปรแกรมภาษาจาวานั้นในช่วงเริ่มแรกนั้นมีลักษณะที่เรียกว่า command line driven compiler เครื่องมือแรกที่ใช้ในการประยุกต์เพื่อสร้าง applet คือ Sun's Java Development Kit (JDK) ซึ่ง applets ที่ได้มานั้นมีลักษณะที่มีการเคลื่อนไหวไม่มากนักหรือ applet ที่มีความยุ่งยากไม่มากนัก และการประยุกต์เพื่อสร้าง standalone Java applications นั้นก็เช่นกัน นักพัฒนาโปรแกรมก็ยังคงต้องการเครื่องมือที่ใช้ช่วยในการศึกษาและพัฒนาโปรแกรมด้วย ในปัจจุบันนี้มีเครื่องมือที่ช่วยในการพัฒนาโปรแกรมภาษา Java ซึ่งต่างได้รวมเอาคุณสมบัติของ GUI (Graphic User Interface) ไว้ในการพัฒนา Java Application โดย Java Tool เหล่านั้น Compatible กับ Java Language Standards. เครื่องมือที่ใช้ศึกษาและพัฒนาโปรแกรมภาษา Java จึงมีความง่ายและสะดวกกว่าในการที่จะศึกษาและใช้งานภาษา Java มากขึ้น เครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวานั้นมีหลายบริษัทที่พัฒนาเครื่องมือเหล่านั้นออกมามากตัวอย่าง โปรแกรมที่มีชื่อเสียงพอสังเขปดังต่อไปนี้

### JDK หรือ Java Development Kit

JDK เป็นเครื่องมือสำหรับช่วยในการสร้างและพัฒนา Java ประกอบไปด้วยไฟล์ต่างๆ ดังต่อไปนี้

1. ไฟล์ appletviewer.exe ใช้สำหรับทดสอบ Java Applet เพื่อ View ผลการทำงาน โดยไม่ต้องสั่ง Run ผ่านทาง Browser
2. ไฟล์ java.exe เป็น Interpreter ใช้ทดสอบการทำงานของ Java Application
3. ไฟล์ javac.exe มาจากคำว่า Java Compiler ใช้สำหรับแปล Source Program ที่เราเขียนขึ้นแบบ ASCII Code (.java) เป็น Byte Code (.class)
4. ไฟล์ javadoc.exe ใช้สำหรับสร้างไฟล์เอกสาร Document จาก Source code ของ Java ในรูปแบบของ HTML ภายในจะเป็นการบอกรายละเอียดของคลาสต่างๆ ที่ผู้ใช้เขียนขึ้น กฎการเขียนที่ควรรู้ไว้กรณีที่มีผู้สนใจอยากจะทำคือ ผู้ใช้จะต้องใส่คำสั่ง `/**` เข้าไปด้วยเสมอ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงส่วนบนของโปรแกรมก่อนการประกาศคลาส ลักษณะการใช้งานจะเหมือนกับคำสั่ง // และ /\* \*/ (สำหรับใส่ Comment คำอธิบายต่างๆ ไป)

5. ไฟล์ javap.exe ใช้แปลงคลาสไฟล์ของ Java ที่เป็นภาษาแอสเซมบลี กลับมาเป็น Source code

### Sun's Java WorkShop

Java WorkShop 1.0 เป็นผลิตภัณฑ์ที่สามารถใช้ในการเขียนโปรแกรม Java และ HTML ที่สามารถแสดงผลได้ในรูปแบบหรือลักษณะที่คล้ายกับ Browser ซึ่งมาจาก Sun's HotJava Web browser. ทำให้ Java components ที่ใช้รันภายใต้ Microsoft Windows 95, Windows NT, and Sun Solaris 2.4 ซึ่งใช้ Platform เดียวกันคือ Java virtual machine (VM) และมี compiler components บางตัวที่มาจาก Sun Java Development Kit (JDK). ซึ่งผลิตภัณฑ์นี้ต้องการใช้ Hardware ในการพัฒนาที่มีประสิทธิภาพค่อนข้างมาก เช่น ต้องการ RAM มากถึง 32MB ที่จะทำให้ประสิทธิภาพในการทำงานนั้นเป็นที่ยอมรับได้

### Microsoft Visual J++, Professional Edition

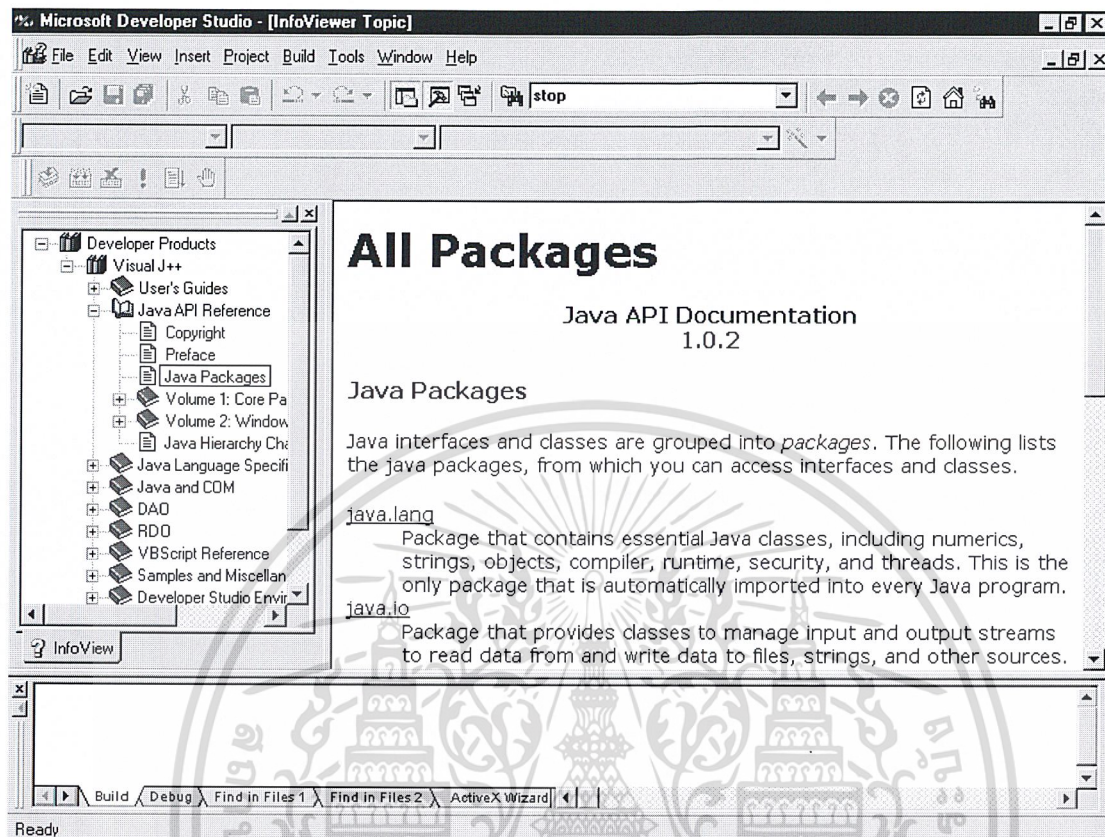
Visual J++ คือเครื่องมือที่ใช้พัฒนาการเขียน Java applets และ applications โดยบริษัท ไมโครซอฟท์. Visual J++ ยังมีคุณสมบัติอีกอย่างทีใน Java Language ของ Sun ไม่มีคือในส่วนที่เรียกว่า **Component Object Model (COM)** และ Type Library Wizard ที่ทำงานโดย type libraries ของ COM objects (รวมถึง ActiveX controls) ที่ User สามารถที่จะ import มาใช้ได้. ด้วยเหตุนี้เอง User จึงสามารถใช้ Methods และกำหนด Properties ได้อย่างง่ายดาย

Sun JDK 1.1 ได้ถูกรวมอยู่ใน Visual J++ ตั้งแต่ version 1.0 ซึ่งปัจจุบันมีถึง version 1.1 รวมทั้ง International และ UNICODE features, Digital signing ของ applets, the JAR packaging capability (our .CAB files), Java Native Method Interface capabilities, Java Beans หรือตลอดจน Support ActiveX รวมทั้งการติดต่อกับ Database ด้วยวิธี RDO (Remote Access Object) หรือ DAO (Data Access Object)

Visual J++ ยังมีความสามารถในด้าน visually create Java forms โดย resource editor ขอมให้ user ใช้ visually layout ในการ interface application ของ user เอง. resource editor ก็เหมือนกับ editor โดยทั่วไปที่ใช้กันอยู่ใน Microsoft Visual C++ ซึ่ง User ยังสามารถที่จะ convert เป็นรูปแบบของ Visual Basic และ Visual C++

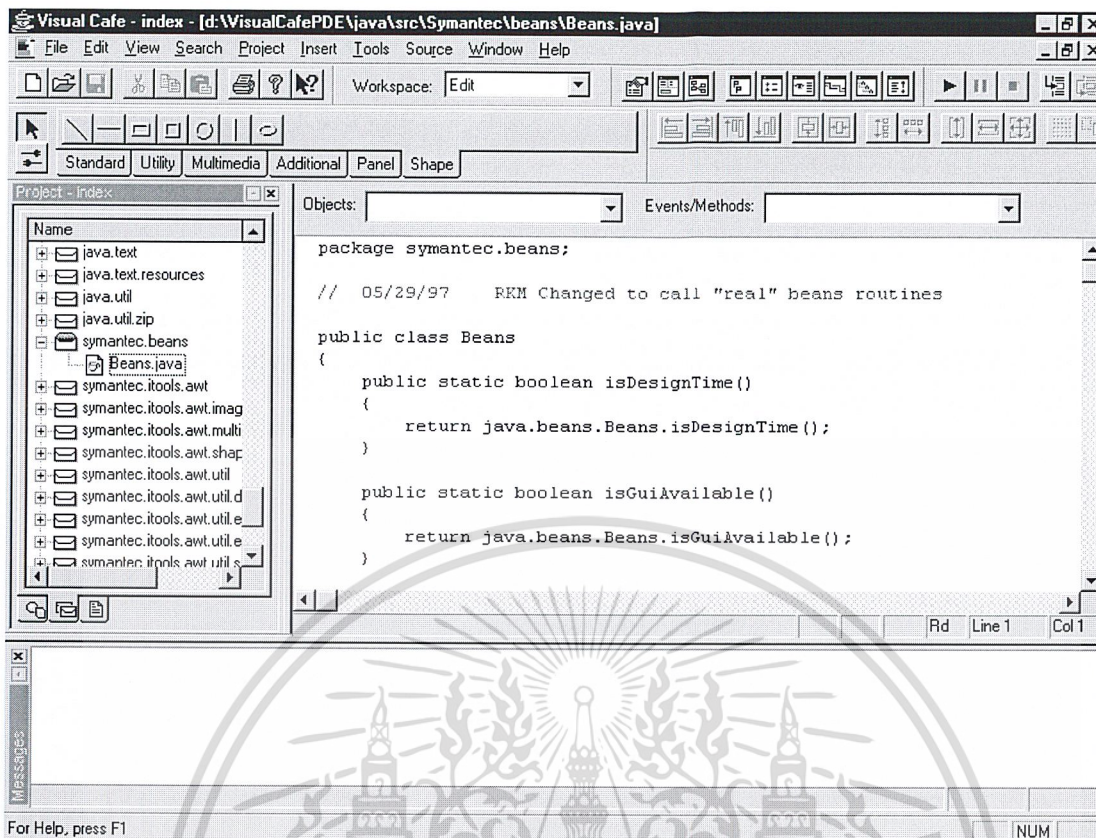
### Symantec Visual Cafe

Visual Cafe ก็เป็นเครื่องมืออีกชนิดหนึ่งที่มีรูปแบบการ Interface คล้ายกับ tools ตัวอื่นๆ แต่จะมี wizard ใหม่สำหรับช่วยทางด้าน component-based programming ซึ่งเรียกว่า RAD (Rapid Application Development) Tool ยกตัวอย่างเช่น Borland Delphi หรือ Microsoft Visual Basic. และ Visual Cafe ยังมี JavaBeans specification. ซึ่งนักพัฒนาโปรแกรมสามารถเขียนโปรแกรมด้วย Components ที่ถูกสร้างขึ้นมาแล้ว dragging มายังรูปแบบที่ต้องการได้ สามารถการ setting properties และ events ได้โดยตรง. ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-3 โปรแกรม Microsoft Visual J++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-4 โปรแกรม Symantec Visual Café

สามารถสรุปเป็นตารางเปรียบเทียบความสามารถของเครื่องมือในการพัฒนาภาษา Java ได้ดังต่อไปนี้

Summary of features	Java 1.0	WorkShop 1.0	Microsoft Visual J++ 1.0	Visual Café 1.0
y YES n NO				
Products listed in alphabetical order	Y	Y	Y	Y
Development Environment				
Context-sensitive help	Y	y	y	Y
Wizards or equivalents	Y	y	y	Y
Cut-and-paste from online help	N		y	Y
User can customize and reuse desktop	Y		y	Y
Third-party program launch from IDE	Y		y	Y
Include JIT compiler	N		y	Y

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Supports JDK tools	Y	y	Y
<b>Editor</b>			
Syntax highlighting	n	y	Y
Remappable keys	n	y	Y
Macros	n	y	Y
Brief compatibility	n	y	y
Epsilon Compatibility	y	y	y
<b>Programming Tools</b>			
Wizard can generate applets	y	y	y
Wizard can generate applications	y	y	y
Wizard can add control events	y	n	y
Wizard adds animation automatically	n	y	y
Class browser	y	y	y
Can browse uncompiled projects	y	y	y
Graphical class-designer tool	n	n	y
GUI resource designer	y	y	y
Attaches code to controls in resource editor	y	n	y
Supports .BMP/.GIF/.JPG graphics files	yyy	yyy	yyy
Can import Windows resource script files	n	y	y
HTML editor included	n	n	n
Supports ActiveX components	n	y	n
ActiveX creation in Java	N/A	y	N/A
Supports JavaBeans	n	n	y
Can target reusable components	y	y	y
Provides classes for database connectivity	n	y	n
Provides resolution-independent form design	y	y	y
<b>Debugging</b>			
Integrated debugger	y	y	y

Includes browser with debugging hooks	y	y	y
Debugger catches Java exceptions	y	y	y
Breakpoints	y	y	y
Watch variables	y	y	y
Support multithreaded debugging	y	y	y
Supports remote debugging	n	n	y
Integrated source control	y	y	n
Disassembles Java byte code	n	y	n
Applet Viewer	y	y	y
JavaScript/HTML debugging	nn	nn	nn

ตารางที่ 5.1 ตารางเปรียบเทียบเครื่องมือในการพัฒนาภาษาจาวา

### 5.7 ความแตกต่างระหว่าง Java Applet และ Java Application

Java Applet กับ Java Application เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อใช้งานบนอินเทอร์เน็ตเหมือนกัน ทำงานอยู่บนโฮมเพจร่วมกับภาษา HTML ที่ใช้สร้าง โฮมเพจเหมือนกันลักษณะที่แตกต่างกันก็คือ

- Java Applet ไม่สามารถทำงานเดี่ยวๆ ด้วยตนเองได้จะทำงานร่วมกันกับไฟล์ .html โดยเราจะเรียกใช้ไฟล์ที่เขียนจาก Java นี้ผ่านทาง Tag คำสั่ง `<APPLET>...</APPLET>` ภายในไฟล์ HTML จากนั้นก็จะเรียกใช้ไฟล์โปรแกรม javac.exe มาทำการคอมไพล์เพื่อที่จะได้ไฟล์ที่มีนามสกุล .class ไว้สำหรับให้ไฟล์ HTML เรียกผ่านวิธีแสดงผลการทำงานของ Java Applet เราจะใช้โปรแกรม appletviewer.exe หรือ Web Browser เช่น Netscape, Hotjava, Internet Explorer ฯลฯ โดยเลือกใช้ตัวใดตัวหนึ่งก็ได้
- Java Application ตัวที่ 2 นี้มันจะสามารถปฏิบัติงานเดี่ยวๆ ได้ด้วยตัวเองคล้ายกับการเขียนโค้ดในภาษาโปรแกรมทั่วไป สิ่งที่เราควรจดจำไว้ให้ดีในการเขียน Java Application ก็คือ เราจะต้องมี Method ของ Main(string args[]) อย่างน้อยๆ 1 Method เพื่อใช้เป็นจุดเริ่มต้นของการทำงานของโปรแกรม อันนี้ถ้าหากสังเกตให้ดีจะเหมือนกับการเขียนโปรแกรมภาษาซีทุกอย่างไป วิธีการคอมไพล์ก็จะใช้โปรแกรม javac.exe เช่นเดียวกับ Java Applet ส่วนการแสดงผลเราจะใช้โปรแกรม java.exe แทน ซึ่งจะแตกต่างจาก Java Applet ตรงจุดนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.8 ความแตกต่างระหว่าง Java และ JavaScript

สามารถสรุปเป็นตารางแสดงการเปรียบเทียบระหว่าง JavaScript และ Java ได้ดังนี้

JavaScript	Java
1. ไม่มีการถูก compile ก่อน Run แต่จะ Run ได้เลยโดยใช้ interpreter บน client	1. ถูก compile ที่ Server ก่อนรันบน Client
2. Object-based, รหัสใช้ object ภายในซึ่งมีอยู่แล้วและไม่มีการประกาศ class หรือการสืบทอด class	2. Object-oriented, Applet มีการสร้าง Object และสืบทอดจาก class
3. รหัสรวมอยู่ในเนื้อ Script HTML	3. Applet จะแยกต่างหากจาก Script HTML
4. ไม่มีการประกาศชนิดของตัวแปร	4. ต้องมีการประกาศชนิดของตัวแปร
5. ตรวจสอบความถูกต้องในขณะที่รัน ถ้ามีข้อผิดพลาดเกิดขึ้น interpreter จะแจ้งให้ทราบในขณะที่ Run ไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้	5. ต้องถูกตรวจสอบความถูกต้องก่อนการ Run คือในขณะที่ compile ถ้า compile ไม่ผ่านไม่สามารถรันได้ไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้

ตารางที่ 5. 2 ตารางเปรียบเทียบ Java Script และ Java

## 5.9 ทิศทางของ JAVA

JAVA ได้เริ่มต้นจากการเป็นภาษาคอมพิวเตอร์ง่ายๆ สำหรับสร้าง Applet หรือโปรแกรมเล็กๆ ซึ่งสามารถทำงานบนเครื่องคอมพิวเตอร์ใดๆ ก็ได้ โดยไม่จำเป็นต้องสร้างโปรแกรมที่ใหญ่โตมากนักขึ้นมา ปัจจุบันนี้กระแสความต้องการและความนิยมของจาวากำลังสูงขึ้นอย่างรวดเร็วส่งผลให้เกิดพัฒนาโปรแกรมประยุกต์ หรือ Application ใหม่ๆ โดยใช้ภาษา JAVA และ JAVA ไม่ได้เป็นเพียงแค่ Applet เล็กๆ อีกต่อไปแล้วเพราะเราสามารถใช้ JAVA เพื่อเป็นเครื่องมือสำหรับสร้างซอฟต์แวร์ขนาดใหญ่ ซึ่งผู้ที่กำหนดทิศทางหรือมองภาพของ JAVA ได้ดีในอนาคตอันใกล้ก็คือผู้อยู่เบื้องหลังความสำเร็จของ JAVA ซึ่งบทความต่อไปนี้เป็นารสรุปทศวรรษกับบุคคล 2 ท่านซึ่งบุคคลแรกคือผู้ให้กำเนิด JAVA คือ Jame Gosling และบุคคลที่สองคือผู้ที่ทำให้ชาวโลกได้รู้จักกับ JAVA โดยเข้ามาพัฒนาระบบการตลาดและสนับสนุนโปรดักชันต่างๆ ของเทคโนโลยี JAVA ให้ออกไปสู่ลูกค้าทั่วโลก คือ Dr.Alan Baratz ประธานบริษัท JavaSoft ซึ่งบทความนี้ได้เรียบเรียงสรุปจากวารสารทางอินเทอร์เน็ตที่ชื่อ JavaWorld ดังต่อไปนี้

ในปี 2539 ที่ผ่านมาเป็นปีแห่งการพัฒนาซอฟต์แวร์ประเภท Client หลายต่อหลายฝ่าย ต่างพุ่งเป้าความสนใจไปที่เครื่องคอมพิวเตอร์ประเภทโต้ตอบกับผู้ใช้ในฝั่ง Client กับ Application ต่างๆ ที่สำหรับใช้ภายในองค์กร ในปี 2540 จะเป็นปีทองที่ Software จะก้าวไปไกลมากกว่า Client โดยจะเคลื่อนเอกสารที่เข้าไปใกล้ชิดกับผู้ใช้งานยิ่งขึ้นในรูปแบบของอุปกรณ์ส่วนบุคคลต่างๆ ภายใต้การใช้ JAVA ควบคู่กัน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ไฟฟ้า และเครื่องจักรขนาดเล็ก เช่น PDA (Personal Digital Assistant) หรือ เครื่องมือช่วยระบบดิจิทัลส่วนบุคคล, อุปกรณ์ Set-top Boxes (กล่องควบคุมเนกประสงค์) Smart Phones, ระบบฝังตัวในอุปกรณ์ต่างๆ เช่นใน เครื่องวิทยุติดตามตัว, เครื่องพิมพ์, เครื่องส่งแฟกซ์, , โทรศัพท์เว็บ, โทรศัพท์สนับเว็บ, บราวเซอร์, รวมไปถึงเครื่องใช้ไฟฟ้าต่างๆ ซึ่งอุปกรณ์เหล่านี้จะถูกเชื่อมการติดต่อกับระบบ Smart Cards นั้นหมายถึงว่าผู้บริโภคสามารถที่จะตั้งโปรแกรมควบคุมระยะไกล กับสวิทซ์ไฟฟ้าผ่านทางอุปกรณ์ช่วย (PDAs) เล็กๆ เหล่านี้ ซึ่งขณะนี้มียักษ์ประชิดจำนวนมากต่างก็พยายามที่จะสร้างสรรค์อุปกรณ์แปลกๆ ใหม่ๆ ขึ้นมา ซึ่งจะทำให้ JAVA มีศักยภาพสูงสุด ในการเป็นเครื่องมือสนับสนุนอุปกรณ์เหล่านี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### การใช้ภาษาจาวาติดต่อกับฐานข้อมูล

ภาษาจาวาเป็นภาษาคอมพิวเตอร์ที่ได้รับการพัฒนาอย่างรวดเร็วและต่อเนื่องนอกเหนือจากการสร้างกราฟิกหรือองค์ประกอบที่ใช้ในการประดับโฮมเพจ (Homepage) ยังมีการพัฒนาองค์ประกอบใหม่ ๆ ในการปฏิบัติงานในด้านต่างๆ ได้ถูกเพิ่มเติมประสิทธิภาพให้แก่จาวาคือ JDBC (Java Database Connectivity) ซึ่งทำหน้าที่อำนวยความสะดวกในการเชื่อมต่อและสื่อสารกับระบบฐานข้อมูล โดยพื้นฐานของภาษาจาวาที่ได้รับการออกแบบมาเพื่อปฏิบัติงานกับระบบเครือข่ายคอมพิวเตอร์โดยตรง องค์ประกอบนี้จึงทำให้จาวาเหมาะสมเป็นอย่างยิ่งในการเป็นเครื่องมือสำหรับสร้างโปรแกรมประยุกต์เพื่อเข้าถึงข้อมูลแบบไคลเอนท์ที่เซิร์ฟเวอร์ โดยมี JDBC เป็นองค์ประกอบในการเชื่อมในการถ่ายทอดคำสั่งการเรียกค้นข้อมูลจากไคลเอนท์ที่ ไปสู่ เซิร์ฟเวอร์ และรับข้อมูลผลลัพธ์จาก เซิร์ฟเวอร์ กลับคืนไปสู่ไคลเอนท์

ระบบไคลเอนท์ที่เซิร์ฟเวอร์ (Client-Server) ในปัจจุบันมีการเพิ่มขึ้นของค่าใช้จ่ายของบุคลากรที่ มาทำหน้าที่ดูแลบำรุงรักษา Database Server ซึ่งได้แก่บุคลากรด้านการจัดการและด้านบริการสนับสนุน สำหรับ Network ขนาดเล็ก (โดยทั่วไปน้อยกว่า 20 ผู้ใช้) โดยปกติผู้จัดการ Network เพียงคนเดียวสามารถ รับภาระหน้าที่ดูแลบำรุงรักษา Database Server ควบคุมการเข้าใช้ Database Server จากบรรดาผู้ใช้ทั้งหลาย และให้บริการสนับสนุน Front End เมื่อจำนวนผู้ใช้งานข้อมูลเพิ่มจำนวนขึ้น หรือฐานข้อมูลด้วย ตัวของมันเองมีขนาดใหญ่ขึ้น โดยปกติแล้วก็เริ่มจำเป็นที่ต้องจ้างผู้จัดการฐานข้อมูลเพิ่มอีกคนเพื่อมาทำ หน้าที่รันระบบจัดการฐานข้อมูลและให้บริการสนับสนุนในส่วน Front End ด้วย นอกจากนี้แล้วในกรณี ที่ระบบจัดการฐานข้อมูลรันอยู่บนระบบปฏิบัติการที่บุคลากรด้านบริการสนับสนุนไม่คุ้นเคยนั้นก็ต้องมีการอบรม ซึ่งเป็นค่าใช้จ่ายที่ต้องเพิ่มเข้าไปในค่าใช้จ่ายเริ่มแรกด้วย

นอกจากนี้แล้วค่าใช้จ่ายด้านฮาร์ดแวร์ก็เพิ่มขึ้นด้วย ในขณะที่มีฐานข้อมูล ไคลเอนท์ที่เซิร์ฟเวอร์ เป็นจำนวนมากที่รันอยู่ภายใต้ระบบปฏิบัติการที่นิยมใช้กันเช่น Netware OS2 และ Unix เป็นต้น และ บริษัทผู้ขายส่วนใหญ่ก็ประกาศว่าระบบจัดการฐานข้อมูลของตนเองนั้นสามารถรันอยู่บนฮาร์ดแวร์เดียวกับที่ File Server Software รันอยู่ได้ก็ตาม แต่ว่าโดยปกติแล้วถ้าต้องการสมรรถนะที่สูงและความถูกต้อง ของข้อมูลที่สูงก็สมควรให้ Database Server แยกไปรันบนคอมพิวเตอร์เฉพาะกิจของ Database เอง สิ่งนี้ หมายความว่าต้องทำการซื้อคอมพิวเตอร์เพิ่มอีกเครื่อง เพื่อมาทำเป็น Database Server โดยเฉพาะ คอมพิวเตอร์นี้ต้องมีกำลังและประสิทธิภาพสูง มีแรมเป็นจำนวนมากและมีฮาร์ดดิสก์ขนาดใหญ่ พร้อม ทั้งอุปกรณ์สนับสนุนเช่น เครื่องป้องกันไฟฟ้า (UPS) เพื่อป้องกัน เซิร์ฟเวอร์ เสียหายจากเหตุการณ์ไฟตก หรือไฟดับ

ค่าใช้จ่ายโดยรวมของ Software ของระบบไคลเอนท์ที่เซิร์ฟเวอร์ ปกติแล้วสูงกว่าระบบจัดการ ฐานข้อมูลแบบหลายผู้ใช้งานพีซี ค่าใช้จ่ายต่อหนึ่ง เซิร์ฟเวอร์ สำหรับฐานข้อมูลไคลเอนท์ที่เซิร์ฟเวอร์ ที่ สามารถบริการผู้ใช้ได้แบบไม่จำกัดจำนวน มีราคาสูง บวกกับค่าใช้จ่ายแยกออกไปของ front-end หรือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือพัฒนา software พร้อมทั้งค่าใช้จ่ายบุคลากรสำหรับการฝึกอบรมโปรแกรมเมอร์ในระบบใหม่ กล่าวได้ว่าค่าใช้จ่ายของระบบไคลเอนท์ที่เซอร์เวอร์ สูงกว่าของระบบจัดการฐานข้อมูลบนพีซีมาก

ยังมีเรื่องที่น่ายุ่งยากอีกเรื่องหนึ่งคือ ระบบไคลเอนท์ที่เซอร์เวอร์ นั้นมีส่วนประกอบต่างๆ เป็นจำนวนมากตามหลักที่ไปที่กล่าวว่า ยิ่งระบบใดมีส่วนประกอบเป็นจำนวนมากโอกาสที่ส่วนประกอบต่างๆ ในระบบนั้นจะเสื่อกก็ยิ่งมากตามไปด้วย เมื่อเกิดเหตุการณ์ร้ายแรงและระบบเกิดพังเสียหายก็เป็นการยากที่จะชี้ชัดถึงปัญหาที่แท้จริงได้ ระยะเวลาที่ใช้ในการทำให้ทุกสิ่งทุกอย่างกลับสู่สภาพปกติและเริ่มทำงานได้อีกก็เป็นเวลาที่ยาวนาน สิ่งต่างๆ ที่เกิดขึ้นเช่นนี้มาจากการขาดประสบการณ์และความชำนาญของบุคลากรด้านบริการสนับสนุนและโปรแกรมเมอร์ ทั้งนี้เนื่องจากไคลเอนท์ที่เซอร์เวอร์ เป็นเทคโนโลยีใหม่ เมื่อเวลาผ่านไปและระบบไคลเอนท์ที่เซอร์เวอร์ เริ่มเป็นที่นิยมใช้มากขึ้นปัญหาที่ได้กล่าวมาแล้วก็จะลดน้อยลง

องค์ประกอบทางด้าน Hardware และ Software รวมกัน ซึ่งระบบจัดการฐานข้อมูลไคลเอนท์ที่เซอร์เวอร์ รันอยู่ใน Platform ต่างๆ เช่น พีซี, Unix Workstation, Super Server, Minicomputer, Mainframe ซึ่งพีซีเป็น Platform ที่นิยมใช้กันมากที่สุดสำหรับไคลเอนท์ที่เซอร์เวอร์

บรรดากระบวนฮาร์ดแวร์ต่างๆ ทั้งหลายก็มีความแตกต่างทั้งในเรื่องคุณลักษณะและความสามารถ อย่างไรก็ตามก็มีคุณลักษณะร่วมที่จำเป็นสำหรับระบบปฏิบัติการ โดยปกติแล้ว Application ถูกเขียนให้รันภายใต้ระบบปฏิบัติการใดระบบหนึ่งโดยเฉพาะ ซอฟต์แวร์ระบบปฏิบัติการที่นิยมใช้กัน เช่น MS-Dos, PC-Dos, OS/2, Unix, VMS ของ Digital, MVS/XA ที่รันบน IBM Mainframe เป็นต้น

คุณลักษณะเบื้องต้นของระบบปฏิบัติการที่จำเป็นสำหรับระบบจัดการฐานข้อมูลไคลเอนท์ที่เซอร์เวอร์ คือ ความสามารถในการทำ Multitasking หรือการรัน Application หลาย Application พร้อมกัน Multitasking ทำให้ซอฟต์แวร์ระบบจัดการฐานข้อมูลสามารถบริการการสอบถามต่างๆ และคำขอต่างๆ ของบรรดาผู้ใช้ทั้งหลายได้อย่างถูกต้อง โดยปราศจากการรบกวนซึ่งกันและกัน multitasking ทำงานโดยแบ่งเวลาประมวลผลของซีพียูออกเป็นช่วงๆ (time Slices) ให้แก่บรรดา Task ต่างๆ และบรรดา Process ต่างๆ ระบบปฏิบัติการ Multitasking ทั้งที่เป็นแบบ Preemptive และแบบ Non-Preemptive

นอกจากนี้บางระบบปฏิบัติการยังสามารถเป็นระบบ Multi-user ได้ (คือยอมให้มีผู้ใช้หลายคนทำงานที่แตกต่างกันได้พร้อมกัน) โดยเฉพาะเมื่อมีการใช้เทอร์มินอลธรรมดาทั้งหลายเป็นทางเข้าสู่ฐานข้อมูลไอเอสแบบ multi-user ไม่ได้ ก่อให้เกิดข้อดีหรือข้อเสียโดยเฉพาะเมื่อถูกใช้เป็น Platform สำหรับระบบจัดการฐานข้อมูลไคลเอนท์ที่เซอร์เวอร์

การเพิ่มความสามารถด้าน Multithread เข้าไปในซอฟต์แวร์ไอเอส Multitask แบบ Preemptive ความสามารถ Multithread ทำให้ Application ทำ Multitask ได้ภายในตัวเองได้ ตัวอย่างเช่น ระบบจัดการฐานข้อมูลผู้ใช้คนเดียวแบบ Multithread สามารถทำการ Start ใหม่ (Process หรือ Task) ให้ทำงานออกภายนอกที่ซบซ้อนโดยทำอยู่ใน Background ในขณะที่ผู้ใช้กำลังทำการสอบถาม ซึ่งทำอยู่ใน foreground Multithread เป็นปัจจัยที่สำคัญสำหรับการออกแบบระบบจัดการฐานข้อมูลไคลเอนท์ที่เซอร์เวอร์ ที่ซบซ้อน Multithread ทำให้ Application สามารถควบคุมให้ Task ใหม่เริ่มทำงานหรือหยุดทำงานได้ Application สามารถถูกออกแบบให้มีความชาญฉลาดภายในที่ตัดสินใจได้ว่า Process ใดหรือ Task ใดมีไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับความสำคัญและควรได้รับเวลา ซึ่พียูมาก ตัวอย่างเช่น ระบบจัดการฐานข้อมูลให้การแก้ไขข้อมูลมีลำดับความสำคัญสูงกว่าการสอบถามข้อมูล เป็นต้น

เนื่องจากจุดด้อยในเรื่องการไม่มี Platform ใดที่สามารถสนองความต้องการได้ทุกประการหรือเหมาะสมในทุกสถานการณ์ในระบบไคลเอนท์เซิร์ฟเวอร์ ในปัจจุบันมีการประยุกต์ใช้ระบบที่เรียกว่า “Network Computing” เพื่อช่วยแก้ไขในปัญหาของระบบไคลเอนท์เซิร์ฟเวอร์ ในเรื่องการใช้บุคลากรในการดูแลระบบน้อยลงคือจะไม่ต้องไปดูแลในส่วนของไคลเอนท์ ที่ในอนาคตจะมีจำนวนเพิ่มขึ้นเรื่อยๆ การง่ายในการปรับปรุงระบบไม่ต้องคอยตามแก้ไขที่ทุกเครื่องไคลเอนท์ และจากปัญหาการทำงานไม่ครบทุก Platform ในปัจจุบันมีภาษาคอมพิวเตอร์หนึ่งที่สามารถเขียนโปรแกรมคอมพิวเตอร์โดยไม่คำนึงถึง Platform ประกอบกับความสามารถทางด้านระบบ Network Computing ที่ภาษานี้สามารถนำมาประยุกต์ใช้ได้อย่างดี ภาษาดังกล่าวคือ ภาษาจาวา

## 6.1 ความหมายของ JDBC

JDBC (Java DataBase Connectivity) ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystems ซึ่งก็คือฟังก์ชันมาตรฐานหรือ Java Application Programming Interface (API) สำหรับการเชื่อมต่อกับระบบฐานข้อมูล. นักพัฒนาสามารถใช้ JDBC API และยังสามารถประกอบด้วย packages อื่นๆ ด้วย ซึ่งนำเสนอในรูปแบบ function พิเศษหรือ higher level API เพื่อเขียนโปรแกรมอิสระในการกำหนด DBMS หรือ database connectivity mechanism ในการเริ่มใช้งาน. โดยทั่วไปการใช้ SQL database ในการติดต่อ (Interface) กับ framework เพื่อที่จะจัดมาตรฐานในการติดต่อในส่วนบนสุดของชนิดต่างๆ ของ database connectivity modules ซึ่งก็คือมาตรฐานของ ANSI SQL-2 Entry level database. เพราะว่า relational databases เกือบจะทั้งหมดในปัจจุบันใช้มาตรฐานของ SQL-2 Entry level

JDBC สร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC DataBase Connectivity ของบริษัท Microsoft) ซึ่งในปัจจุบันได้ถือว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูลสำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) และระบบเครือข่ายท้องถิ่น (LAN) แต่ตามหลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบ X-Windows และ JDBC driver ต้องเข้ากันได้กับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้น

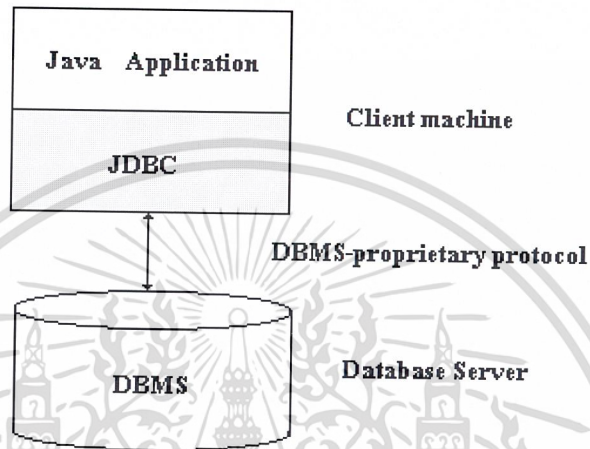
## 6.2 รูปแบบการเชื่อมต่อฐานข้อมูลของ JDBC

JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-Tier model และ Three-Tier model

### Two-Tier model

จาวาแอปเพล็ต (Applet) หรือจาวาแอปพลิเคชัน (Application) จะติดต่อกับฐานข้อมูลโดยตรง จึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC ไดรฟ์เวอร์พิเศษที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลชนิดนั้นได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 3-1 คำสั่งในการเรียกค้นข้อมูลในรูปแบบของภาษา SQL ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Structured Query Language) จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูล ก็จะถูกส่งกลับมาสู่ผู้ใช้ฐานข้อมูลดังกล่าวนี้ส่วนมากจะติดตั้งอยู่ต่างเครื่องคอมพิวเตอร์ โดยผู้ใช้สามารถเชื่อมต่อ ผ่านระบบเน็ตเวิร์ก (Network) รูปแบบ Two Tier นี้ใช้หลักการทำงานเช่นเดียวกันกับรูปแบบไคลเอนท์/เซิร์ฟเวอร์ ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือไคลเอนท์และเครื่องคอมพิวเตอร์ ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two-Tier มักจะเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

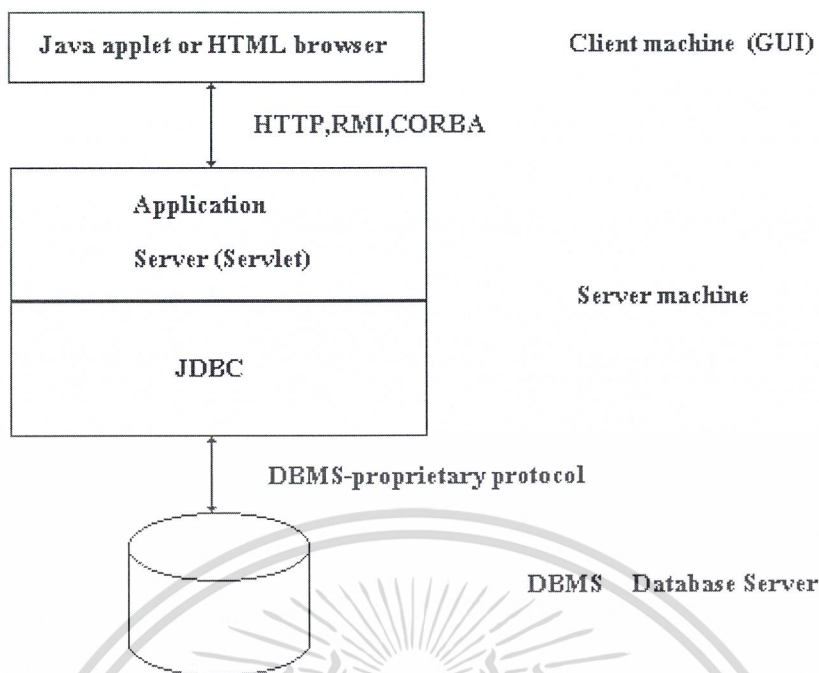


รูปที่ 6.1 Two-Tier Model

### Three-Tier model

คำสั่งเรียกค้นต่างๆ จากผู้ใช้จะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านี้ให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้ก็จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่งต่อไปให้ผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักจะพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ (Web-Server) ซึ่งทำหน้าที่เป็น Middle Tier ก็จะเป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งไคลเอนท์ และเซิร์ฟเวอร์ฐานข้อมูลสามารถพูดคุยกัน ได้การปรับเปลี่ยนระบบคอมพิวเตอร์ของไคลเอนท์และเซิร์ฟเวอร์ หรือแม้กระทั่งการเปลี่ยนฐานข้อมูลตัวใหม่จะไม่มีผลกระทบซึ่งกันและกันเกิดขึ้น

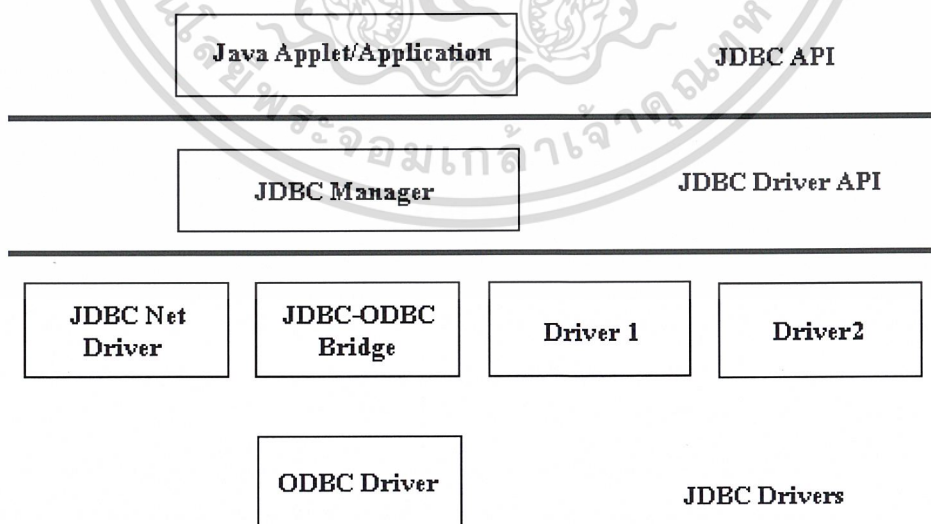
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6. 2 Three-Tier Model

### 6.3 โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 3-3 ระดับบนสุด JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง JDBC Driver (มีไดร์ฟเวอร์ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม

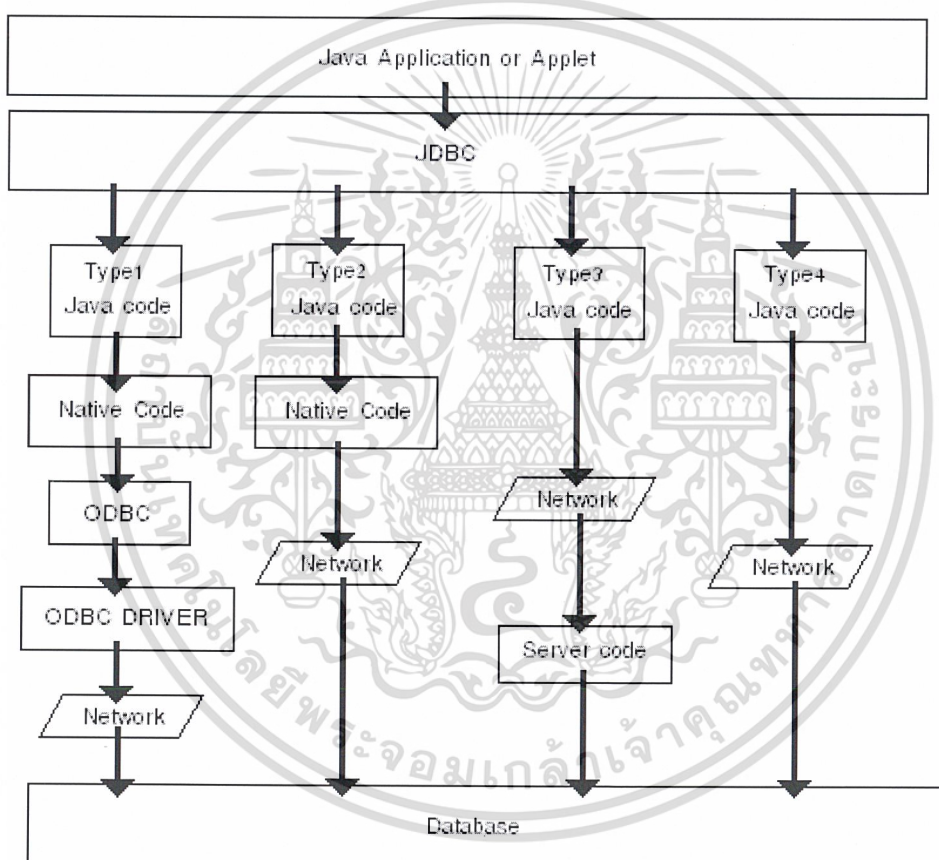


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 6.3 ระดับการเชื่อมต่อของ JDBC API าดให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 6.4 รูปแบบของ JDBC ไดรฟ์เวอร์ (Driver)

JDBC ไดรฟ์เวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่างๆ ของ JDBC ถูกจัดแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

1. JDBC-ODBC Bridge
2. Native-API (Partly-Java) Drivers
3. Net-Protocol (All-Java) Drivers
4. Native-Protocol (All-Java) Drivers



รูปที่ 6.4 JDBC driver types

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการทำงานของไดรฟ์เวอร์แต่ละชนิดอธิบายได้ดังนี้

### JDBC/ODBC bridge

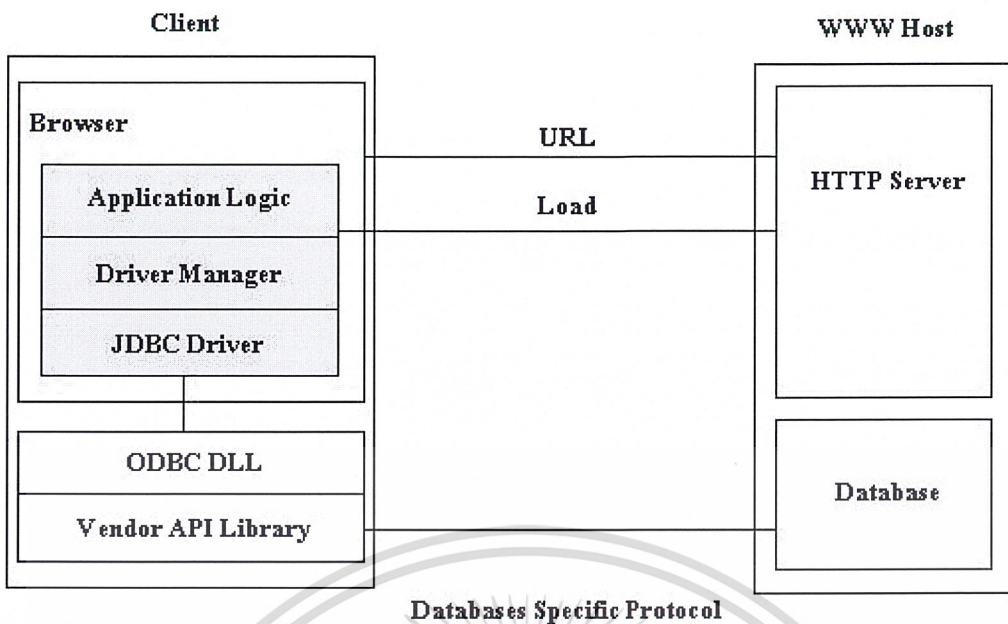
JDBC/ODBC Bridge ถูกพัฒนาโดย JavaSoft และ Intersolv ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลได้โดยผ่านการทำงานของ ODBC โดยนำข้อดีของ ODBC-enabled data sources ที่มีใช้อยู่โดยทั่วไปอย่างมากมาย. ฟังก์ชันไคลเอนท์ Java applet หรือ application จะถูกเขียนโดยใช้ JDBC API. Bridge จะทำการแปลงการเรียกใช้ JDBC ไปยัง ODBC และส่งส่งค่า ODBC driver ที่เหมาะสมสำหรับ back-end database

ข้อดีของ bridge ทำให้ application สามารถติดต่อกับ databases ได้อย่างง่ายดายโดยจาก vendors ที่มากมายโดยเลือก ODBC driver ที่เหมาะสม อย่างไรก็ตามการติดต่อ database ประเภทนี้ต้องพิจารณาค่าใช้จ่าย (Overhead) และ ความซับซ้อน (Complexity) เพราะว่าการเรียกใช้จะมีลำดับการเรียกใช้ดังนี้คือ

JDBC  $\leftrightarrow$  bridge  $\leftrightarrow$  ODBC  $\leftrightarrow$  Native client-API  $\leftrightarrow$  database

การติดต่อโดยใช้ประเภทนี้ชนิดของ Driver จะไม่อนุญาตสำหรับการย้ายข้อมูลแบบ just-in-time ของ Java applet . Native code ต้องมีการเตรียมการติดตั้งบนเครื่องไคลเอนท์ โดยตรงเพราะว่า JDBC-ODBC Bridge เป็นไดรฟ์เวอร์ที่ถูกเขียนขึ้นมาจากภาษา C/C++ ในบางส่วนจึงทำให้ไม่สามารถดาวน์โหลด (Download) ข้ามเครือข่ายอินเทอร์เน็ตและแปล (Interpret) บนเว็บเบราว์เซอร์ (Webbrowser) ได้ เพราะภาษา C/C++ ต้องรันเฉพาะแต่ละระบบเท่านั้น ดังนั้น ODBC ไดรฟ์เวอร์ในรูปแบบของไลบรารีไฟล์หรือ DLL (Dynamic Link Library) และ JDBC-ODBC Bridge ไดรฟ์เวอร์ต้องได้รับการติดตั้งบนไคลเอนท์แต่ละเครื่องเพื่อให้สามารถติดต่อกับฐานข้อมูลที่อยู่ได้ มีความต้องการในการเตรียมการติดตั้ง software โดย Administrative รูปแบบเหมือนไคลเอนท์-server application ดังนั้น JDBC/ODBC Bridge จะไม่สามารถแก้ไขปัญหาของไคลเอนท์ที่ program ดังกล่าวได้จึงเหมาะสำหรับการใช้งานกับระบบ Three-Tier โดยไดรฟ์เวอร์ทั้งสองจะได้รับการติดตั้งอยู่ที่ Middle-Tier หรือเครื่องคอมพิวเตอร์ตัวกลาง เช่น Web-Server เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 การติดต่อ JDBC โดยรูปแบบที่ 1 JDBC-ODBC Bridge

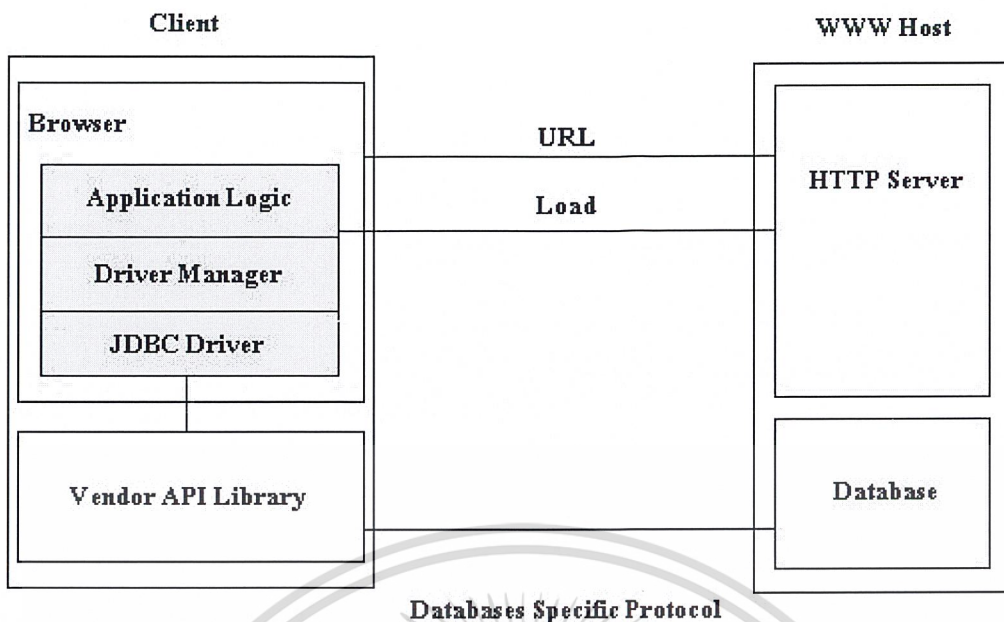
**Native-API, partly Java driver**

Native-API, partly Java driver เป็น two-tier ที่ซึ่ง JDBC driver ต้องการ Vendor Library ในการแปลง JDBC function ไปยังคุณลักษณะของการใช้ภาษาในการ Query เรียกใช้ DBMS ยกตัวอย่างเช่น library สำหรับ Sybase คือ dblib , สำหรับ Oracle คือ ocilib ซึ่ง driver เหล่านี้จะถูกใช้เพื่อเขียนในการรวมของภาษา Java และ C/C++ การติดต่อโดย JDBC ประเภทนี้จะเหมือนกับ JDBC/ODBC Bridge คือต้องการ code ซึ่งก็คือ vendor library เพื่อที่จะใช้ในการติดตั้งบนเครื่องไคลเอนท์ ดังนั้นต้องมีการบำรุงรักษา (maintenance) ปัญหาเหมือนกับ bridge อย่างไรก็ตาม Driver ประเภทนี้จะเร็วกว่าประเภทแรก เพราะว่ามี Layer พิเศษของการแปลงเป็น ODBC ถูกกำจัดออกไป

JDBC ↔ Native client-API ↔ database

ข้อดีของไคลเอนท์ชนิดนี้คือ สามารถใช้ Native-API Driver ติดต่อกับฐานข้อมูลนั้นๆ โดยตรง โดยผ่านโปรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะสมกับการเชื่อมต่อกับฐานข้อมูลแบบ Two-Tier Client/Server แต่ก็มีข้อเสียที่คล้ายคลึงกับ JDBC-ODBC Bridge คือต้องติดตั้งไคลเอนท์ (DLL) และโปรโตคอลไคลเอนท์เช่น SQL-NET, INET, Open Client, ฯลฯ บนเครื่องคอมพิวเตอร์นั้นๆ ที่สำคัญส่วนหนึ่งของ Native-API Driver ถูกเขียนขึ้นจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้และไม่สามารถเชื่อมต่อกับฐานข้อมูลข้ามชนิดกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 การติดต่อ JDBC โดยรูปแบบที่ 2 Native-API, partly Java driver

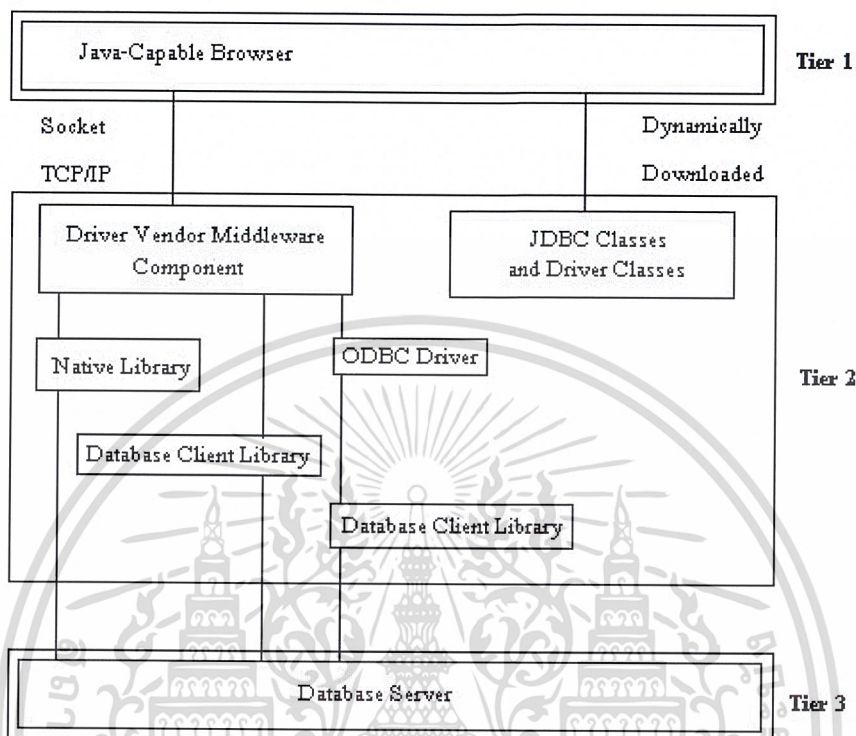
**Network-protocol, all-Java driver**

Network-protocol, all-Java ประเภทนี้ driver จะทำการแปลงการเรียกใช้ JDBC ให้อยู่ในรูปของเน็ตเวิร์คโปรโตคอลร่วม (DBMS-independent network Protocol) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปเฉพาะของแต่ละฐานข้อมูล (database-specific API) บนเซิร์ฟเวอร์นั้นๆ Middle-tier server รูปแบบการเชื่อมต่อฐานข้อมูลจะเป็นลักษณะ three tier : JDBC driver (ปรกติ 200 KB หรือน้อยกว่านั้น) ทำการ execute บนไคลเอนท์ และถูกใช้เครื่องมือทาง logic ส่งคำสั่ง SQL ไปยัง Network และไปยัง JDBC เซอร์เวอร์, เมื่อได้รับข้อมูลกลับมาจาก เซอร์เวอร์ และถูกจัดการ connection . Driver ประเภทนี้อ่อนุญาตสำหรับการใช้ Just-in-time บนเครื่องไคลเอนท์

JDBC เซอร์เวอร์จะจัดการ connection ที่มีมาไปยัง database ซึ่งดีเหมือน exception และ status events resulting จาก SQL execution . Packages ของ data สำหรับการติดต่อข้าม network ไปยัง JDBC ไคลเอนท์

Middleware server component สามารถอธิบายเหมือน native component หรือถูกเขียนด้วย Java. การอธิบาย Native ในการ connect ไปยัง database server โดยใช้ library ของ vendor หรือ ODBC . Symantec’s dbAnywhere และ Intersolv’s SequeLink ทั้งคู่ถูกจัดให้อยู่ในประเภทนี้, SequeLink ไม่ต้องการ database client libraries ที่จะใช้ในการติดตั้งลงบน เซอร์เวอร์ แต่จะถูกใช้ Libraries ของตนเอง. เซอร์เวอร์ ต้องมีการติดตั้ง Configuration สำหรับ database ที่จะใช้ติดต่อเช่น ติดตั้งหมายเลขของ Port, ตัวแปรของ database specific environment สำหรับ instance, DSQuery กับ Sybase, Database specific parameters เช่น logging , translation และ parameter อื่นๆ ที่ซึ่ง เซอร์เวอร์ อาจจะต้องการ ถ้า Middleware เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

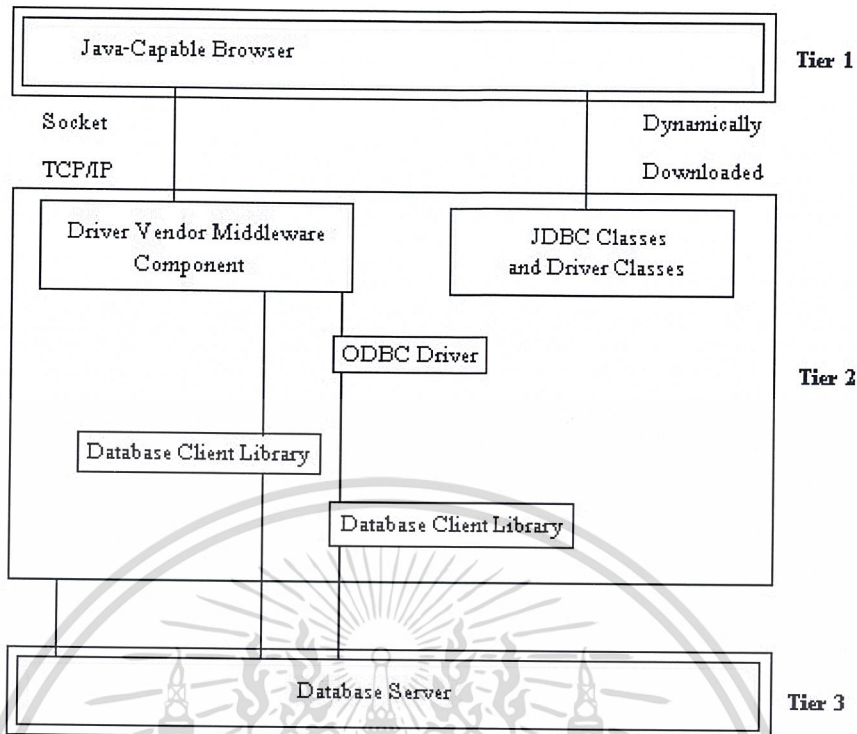
server ถูกเขียนด้วยภาษา Java สามารถใช้ JDBC –compliant driver ในการติดต่อกับ DBMS โดย Database Proprietary Protocol ของ vendor ไม่ต้องการที่จะ configure ที่ เซอร์เวอร์ สำหรับ database



รูปที่ 6.7 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver ยกตัวอย่างของ dbAnywhere, SequelLink

สำหรับคุณสมบัติเหมือนการติดต่อกับ Caching, จำนวนของ listener threads , และ อื่นๆ , ความต้องการค่าคงที่ที่ต้องมีการติดตั้ง. Weblogic's T3 server ถูกจัดให้อยู่ในประเภทนี้. JDBC T3 driver จะถูก executing บนไคลเอนท์ โดยจะนำชื่อของ driver ใช้ในการติดต่อสื่อสารกับ DBMS ซึ่งดีเหมือนกับคุณสมบัติ ที่ต้องการโดย driver เช่น username, password, database name, และ อื่นๆ ดังนั้นการส่งข้อมูลไปยัง T3 server จะ Load ผ่านไปยัง driver class และใช้ในการติดต่อกับ database . Server ยังต้องมีการ load ด้วย database vendor libraries และ/หรือ ODBC driver ในกรณีนี้การส่งค่า driver class name ในประเภทที่ 1 หรือประเภทที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.8 การติดต่อ JDBC โดยรูปแบบที่ 3 Network-protocol, all-Java driver ยกตัวอย่างของ WebLogic

Driver ประเภทนี้ถือว่าเหมาะสมมากที่สุดสำหรับ Internet/intranet -based, multi-user data-intensive application, ที่ซึ่งมีจำนวนของ concurrent data operations มากๆ เพราะมีความยืดหยุ่นคล่องตัวที่สุด เพราะว่าเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไดรฟ์เวอร์ร่วมที่เขียนขึ้นจากภาษาอื่น ซึ่งต้องติดตั้งเฉพาะบนไคลเอนต์นั้น และสามารถรันบนระบบใดก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine) การปฏิบัติงานทุกรูปแบบทั้งด้านโหลดข้ามเครือข่ายหรือรันบนเครื่องเดี่ยวๆ สามารถกระทำได้หมดเช่นการ queries searches และอื่นๆ และยังสามารถทางด้าน scalability และประสิทธิภาพซึ่งถือเป็นส่วนประกอบหลัก เซอร์เวอร์ สามารถจัดการ multiplexing management ระหว่าง database ที่หลากหลาย, สามารถจัดหา logging และ administration facilities สามารถ load balancing features, และยังสามารถรองรับ catalog และการ Query caches. Three-tier web database application ส่วนใหญ่มี security, firewalls, และ proxies

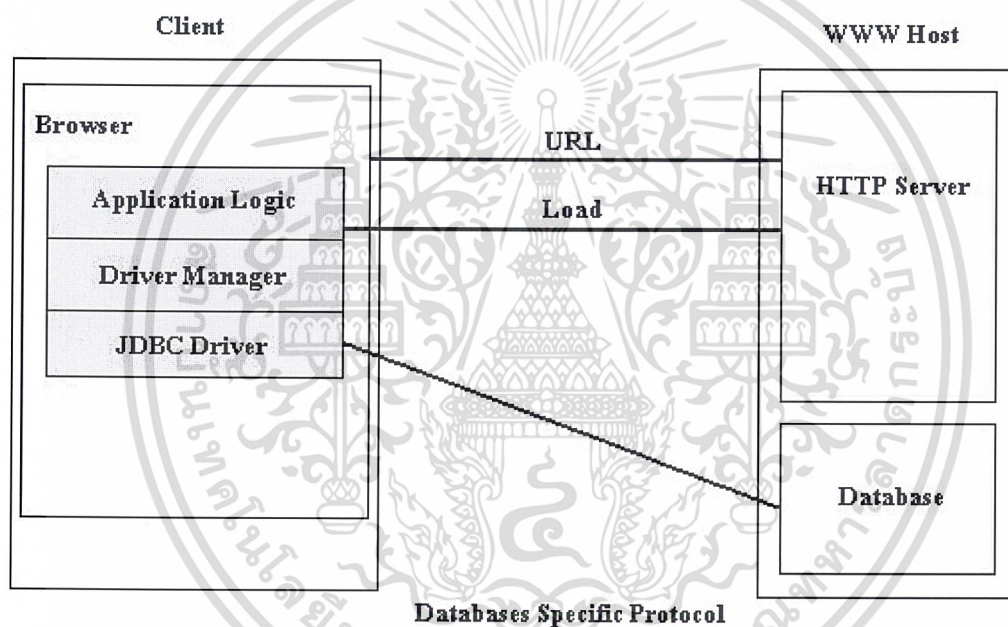
ข้อดีของ network-centric drivers คือองค์ประกอบของ เซอร์เวอร์ คือ proprietary middleware; ตัวแทนจำหน่ายแต่ละรายใช้ middleware ของตนเองสำหรับการติดต่อสื่อสารข้ามผ่าน network ดังนั้น enterprise users จึงมีความสามารถที่จะจัดการได้โดย CORBA products และการ upgrade paths ในอนาคต

Openjdbc จาก I-Kinetics คือ Network-centric driver ที่ใช้ CORBA คือ middleware จะเชื่อมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้า JDBC ไคลเอนท์ และ เซอร์เวอร์. เครื่องมือชนิดนี้มีข้อดีของ CORBA's distributed computing ไม่วากกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการแก้ไข

infrastructure คือวงการอุตสาหกรรมยอมรับเป็นมาตรฐานในการ Open ที่จะติดต่อไปยังการสื่อสารอื่นๆ Driver มาเป็น assembly ของ cooperating objects ที่ค่อนข้างจะมากกว่าหนึ่ง เซอร์เวอร์ .

#### Native-protocol, all-Java driver

Native-protocol, all-Java drivers จะแปลงการเรียกใช้คำสั่งของ JDBC ให้อยู่ในรูปแบบของเน็ตเวิร์คโปรโตคอลเฉพาะของฐานข้อมูลนั้นโดยตรงไป โดยคุณสมบัติของตัวแทนจำหน่าย database. Driver เหล่านี้สามารถเขียนในภาษา Java และสามารถติดต่อกับ applet แบบ just-in-time เพราะว่า drivers เหล่านี้จะแปลง JDBC ตรงไปยัง native protocol โดยปราศจากการใช้ ODBC หรือ Native APIs ซึ่งสามารถจัดหาสำหรับการติดต่อ database ที่มีประสิทธิภาพสูง. Driver เหล่านี้สามารถสร้างจากตัวแทนจำหน่าย DBMS เท่านั้นเนื่องจากความจริงที่ซึ่งความรู้ของ Protocol lies กับตัวแทนจำหน่าย. การนำการติดต่อประเภทนี้มาใช้งานมีอยู่เล็กน้อยมากในปัจจุบัน แต่ในอนาคตคาดว่าจะเพิ่มจำนวนขึ้นเรื่อยๆ



รูปที่ 6.9 การติดต่อ JDBC โดยรูปแบบที่ 4 Native-protocol, all-Java driver

ข้อดีของไดร์ฟเวอร์ชนิดนี้คือไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กร และไม่มีคามจำเป็นที่ต้องติดตั้งไดร์ฟเวอร์ตัวกลาง เช่น ODBC บนไคลเอนท์ที่แต่ละเครื่องจึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลแบบ Two-Tier สำหรับเครือข่ายอินทราเน็ตภายในองค์กร ไดร์ฟเวอร์ชนิดนี้มีข้อเสียคือไม่เหมาะที่จะนำไปใช้บนเครือข่ายอินเทอร์เน็ต เพราะปัญหาเรื่องความปลอดภัย การเชื่อมต่อฐานข้อมูลไม่สามารถกระทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไดร์ฟเวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JDBC ไดรฟ์เวอร์ชนิดที่ 3 และ 4 คือไดรฟ์เวอร์ที่คาดว่าจะเป็นที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องมาก กว่า เช่นมีการติดตั้งไดรฟ์เวอร์โดยอัตโนมัติเมื่อมีการดาวน์โหลดจาวาแอปพริท สำหรับไดรฟ์เวอร์ชนิดที่ 1 และ 2 คือไดรฟ์เวอร์ เพื่อพัฒนาขึ้นมาเพื่อใช้งานร่วมกับไดรฟ์เวอร์ดั้งเดิม (ODBC) หรือเน็ตเวิร์คโปรโตคอลเฉพาะฐานข้อมูล ที่มีใช้อยู่แล้วในปัจจุบันไปหลายๆ ก่อน ซึ่งแน่นอนว่าไดรฟ์เวอร์ชนิดที่ 1 และ 2 ชนิดนี้มีข้อจำกัดในการใช้งานมากกว่า 2 ชนิดแรกโดยเฉพาะเรื่องความปลอดภัยและ Bridge ร่วมกับ ODBC ไดรฟ์เวอร์บน Window 95/NT ผลปรากฏว่าไม่ยุ่งยากและให้ประสิทธิภาพดีพอสมควร เหมาะสมในการเป็นเครื่องมือในการทดลองและศึกษาขั้นตอนในการเชื่อมต่อกับฐานข้อมูลจนตั้งแต่ขั้นพื้นฐานจนถึงขั้นตอนให้ปฏิบัติงานได้จริง

## 6.5 ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล

1. เปิดการเชื่อมต่อกับฐานข้อมูล
2. ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
3. จัดการกับผลลัพธ์ที่ได้รับ
4. ปิดการเชื่อมต่อ

และเพื่อให้เข้าใจการทำงานของ JDBC ยิ่งขึ้นตัวอย่างโปรแกรมบางส่วนต่อไปนี้จะแสดงให้เห็นถึงขั้นตอนทั้ง 4 ในการติดต่อกับฐานข้อมูลผ่าน ODBC ไดรฟ์เวอร์โดยใช้ JDBC-ODBC Bridge (ฟังก์ชัน API ต่างๆ ของ JDBC ที่มาพร้อมกับ JDK เวอร์ชัน 1.1 ขึ้นไปถูกเก็บรวบรวมไว้ในแพ็คเกจ java.sql.\*)

// (1) เปิดการเชื่อมต่อกับฐานข้อมูล

จากตัวอย่างสิ่งแรกๆ ที่ทำคือเลือกชนิดของ JDBC ไดรฟ์เวอร์ (JDBC-ODBC Bridge) ระบุชื่อฐานข้อมูล (DSN : Data Source Name) รวมถึงชื่อผู้ใช้ (Login) และรหัสผ่าน (Password) จากนั้นเป็นการสร้างอ็อบเจกต์ stmt (คลาส Statement) สำหรับการเรียกค้นข้อมูล และจัดส่งคำสั่ง SQL ไปในรูปแบบสตริง (String) ผ่านเม็ททอด executeQuery() ผลลัพธ์ที่ได้จากเม็ททอดนี้คือผลลัพธ์จากการประมวลผลของฐานข้อมูลซึ่งจะเก็บอยู่ในอ็อบเจกต์ rs (คลาส ResultSet) จากนั้นข้อมูลที่ได้รับนี้จะถูกถ่ายทอด (Mapping) ให้อยู่ในชนิดข้อมูลที่จาวารู้จัก เช่น int, String, float, ฯลฯ) เพื่อนำไปใช้ประโยชน์ต่อไป สุดท้ายคือการปิดการเชื่อมต่อกับฐานข้อมูล

ภาษาจาวาและ JDBC คือเครื่องมือที่จะทำให้การเชื่อมต่อสื่อสารกับระบบฐานข้อมูลเป็นไปอย่างสะดวกรวดเร็วและเป็นหนึ่งเดียว (Database-neutral Communication) อย่างไรก็ตามพื้นฐานที่จะขาดเสียไม่ได้สำหรับการสร้างระบบเชื่อมต่อฐานข้อมูลนั้นก็คือความรู้ความเข้าใจเกี่ยวกับหลักการงานหลักการออกแบบของฐานข้อมูลชนิดสัมพันธ์ (Relational Database) ระบบจัดการฐานข้อมูล (DBMS : Database Management Systems) และภาษา SQL จากชนิดของ JDBC Drivers สามารถแสดงตัวแทนจำหน่ายและฐานข้อมูลที่สามารถปฏิบัติการแสดงดังตารางได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vendor	Type	DBMSs supported
Agave Software Design	3	Oracle, Sybase, Informix, others via ODBC
Asgard Software	3	Unisys A series DMSII database
Borland	4	InterBase 4.0
Caribou Lake Software	3	Ingres
Connect Software	4	Sybase, MS SQL Server
DataRamp	3	Several dozen through ODBC drivers.*
Ensodex, Inc.	3	Several through ODBC.*
IBM	2,3	IBM DB2 Version 2
IBM	4	DB2 for OS/400
GWE Technologies	4	mysql
IDS Software	3	Oracle, Sybase, MS SQL Server, MS Access, Informix, Watcom, and others via ODBC
I-Kinetics, Inc.	3	Oracle, Informix, Sybase, and others via ODBC
Imaginary	4	mSQL
InterSoft	3	Essentia
Intersolv	3	DB2, Ingres, Informix, Oracle, Microsoft SQL Server, Sybase 10/11
JavaSoft	1	Several dozen through ODBC drivers.*
KonaSoft, Inc.	3,4	Sybase, Oracle, Informix
NetAway	3	Oracle, Informix, Sybase, MS SQL Server, DB2, others via ODBC
OpenLink	3	Oracle, Informix, Sybase, MS SQL Server, CA-Ingres, Progress, Unify, PostgreSQL, Solid, and others via ODBC.
Oracle Corporation	2	Oracle
SAS Institute Inc.	3,4	SAS, and via SAS/ACCESS, Oracle, Informix, Ingres, and ADABAS
SCO	3	Informix, Oracle, Ingres, Sybase, Interbase
StormCloud Development	3	Any DBMS that is accessible via ODBC
Sybase, Inc	3,4	Sybase SQL Server, SQL Anywhere, Sybase IQ, Replication Server and more than 25 enterprise and legacy data servers via Sybase OmniCONNECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symantec	3	Oracle, Sybase, MS SQL Server, MS Access, Watcom and others via ODBC
Trifox, Inc.	3	ADABAS, DB2, Informix, Ingres, Oracle, Rdb, SQL Server, Sybase, and legacy systems via GENESIS.
Visigenic	3	Several dozen through ODBC drivers.*
WebLogic	2	Oracle, Sybase, MS SQL Server
WebLogic	3	Several dozen through ODBC drivers.*
XDB Systems, Inc.	1,3	Many databases through ODBC.*
Yard Software GmbH	4	YARD-SQL Database

ตารางที่ 6.1 เปรียบเทียบ JDBC Driver

\* รายการของ ODBC drivers สำหรับการเชื่อมต่อด้วย JDBC-ODBC Bridge หรือการเชื่อมต่อด้วยชนิดที่ 3 drivers สำหรับตารางแสดงให้เห็นถึงผู้แทนจำหน่าย DBMS หรือ ODBC vendors ยกตัวอย่างเช่น Intersolv, Visigenic, SCO, และ DataRamp. JDBC drivers ต่างๆ สามารถตรวจสอบได้ที่ <http://splash.javasoft.com/jdbc/jdbc.vendors.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

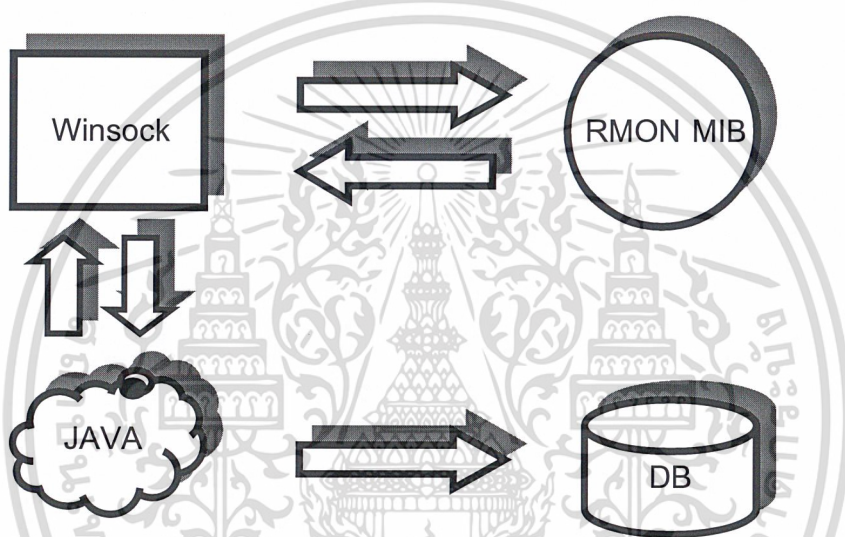
## บทที่ 7

### การสร้างและผลการทดลอง

โครงงานนี้แบ่งการทำงานออกเป็น 2 ส่วนคือส่วนการนำเอาข้อมูลมาเก็บในฐานข้อมูล และอีกส่วนก็จะนำเอาข้อมูลในฐานข้อมูลมาแสดงบนเว็บ

#### 7.1 การนำเอาข้อมูลมาเก็บในฐานข้อมูล

ส่วนนี้จะเริ่มโดยการเขียนโปรแกรมภาษาจาวา (แอปพลิเคชัน) ผ่านระบบเครือข่ายทาง winsock เพื่อไปเอาข้อมูลจากฐานข้อมูล MIB และ RMON ในเครื่องอาเจนท์ที่ต้องการ เมื่ออาเจนท์ส่งข้อมูลกลับคือมา ก็จะนำข้อมูลที่ได้นั้นไปเก็บไว้ในฐานข้อมูล (ในโครงงานนี้ใช้ฐานข้อมูล MS SQL Server)



รูปที่ 7.1 แสดงการนำเอาข้อมูลจาก RMON MIB เก็บใน DB

โปรแกรมในส่วนของการส่งแพคเกจไปขอข้อมูลจาก MIB

```
protected void Sendpacket()
```

```
throws IOException {
```

```
int i;
```

```
byte Temp[] = {48,116,2,1,0,4,6,112,117,98,108,105,99,
```

```
                -95,103,2,2,112,0,2,1,0,2,1,0,48,91, // ส่งขอทั้ง System Group
```

```
                48,11,6,7,43,6,1,2,1,1,1,5,0, //SysDescription
```

```
                48,11,6,7,43,6,1,2,1,1,2,5,0, //SysObjectID
```

```
                48,11,6,7,43,6,1,2,1,1,3,5,0, //SysUpTime
```

```
                48,11,6,7,43,6,1,2,1,1,4,5,0, //SysContact
```

```
                48,11,6,7,43,6,1,2,1,1,5,5,0, //SysName
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

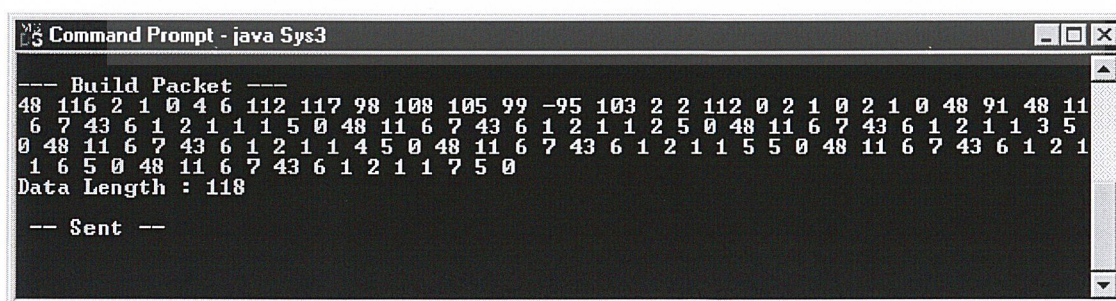
48,11,6,7,43,6,1,2,1,1,6,5,0,           //SysLocation
48,11,6,7,43,6,1,2,1,1,7,5,0};         //SysService

String PDUTemp = "";
byte PDUTempP[] = new byte[1024];
for (i = 0;i<118;i++) {
    PDUTemp = PDUTemp + String.valueOf(Temp[i]);           // การ buildPacket ต้องการ
    PDUTempP[i]= Temp[i];                                   // ค่า string จึงต้องแปลงก่อน
}

int port = 161; int Length = 118;String host = 161.246.4.3; // เลือกพอร์ทและชื่อโฮสต์
DatagramPacket PDUMessage = buildPacket(PDUTempP,InetAddress.getByName
(host),port,Length);
socket.send(PDUMessage); // ส่งแพคเกจ
System.out.println("Sent Packet.");
}

protected DatagramPacket buildPacket(byte message[],InetAddress host,int port,int Length)
throws IOException {
System.out.println("--- Build Packet ---");
byte data[] = new byte[1024];
for (int I=0 ;I<118;I++)
{
data[I] = message[I];
System.out.print(data[I]);}
return new DatagramPacket(data,Length,host,port);}

```



```

Command Prompt - java Sys3
--- Build Packet ---
48 116 2 1 0 4 6 112 117 98 108 105 99 -95 103 2 2 112 0 2 1 0 2 1 0 48 91 48 11
6 7 43 6 1 2 1 1 1 5 0 48 11 6 7 43 6 1 2 1 1 2 5 0 48 11 6 7 43 6 1 2 1 1 3 5
0 48 11 6 7 43 6 1 2 1 1 4 5 0 48 11 6 7 43 6 1 2 1 1 5 0 48 11 6 7 43 6 1 2 1
1 6 5 0 48 11 6 7 43 6 1 2 1 1 7 5 0
Data Length : 118
--- Sent ---

```

รูปที่ 7.2 แสดงข้อมูลที่ส่งไปขอ MIB

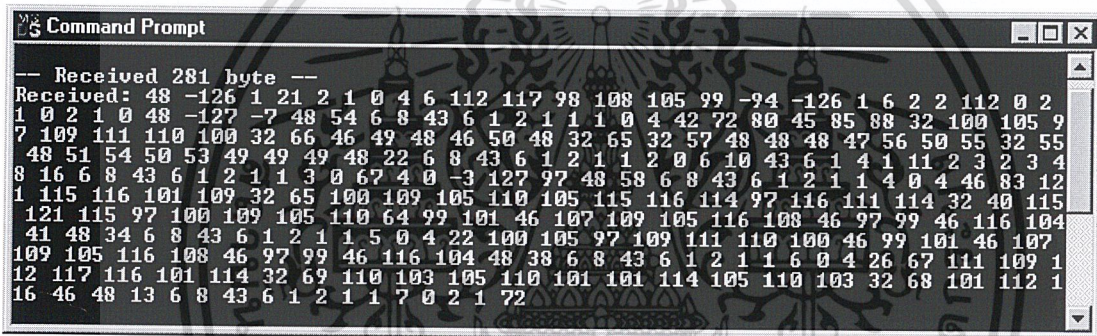
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมในส่วนของการรองรับแพคเกจจากฐานข้อมูล MIB

```
protected void receive(String namesql) throws IOException {
    int NumberPort,I,Lport;

    byte buffer[] = new byte[1024]; // สร้างบัฟเฟอร์ขึ้นมาเพื่อ
    InetAddress IPAdd ; // รองรับแพคเกจ
    String result="";

    DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
    socket.receive(packet); // รับแพคเกจ
    byte[] Tbuffer = new byte[packet.getLength()]; // เปลี่ยนให้อยู่ในรูป string
    Tbuffer = packet.getData(); // เพื่อใช้แสดงผล
    for (I=0;I<packet.getLength();I++)
        result = result + String.valueOf(Tbuffer[I]) + “ ”;
```



```
Command Prompt
-- Received 281 byte --
Received: 48 -126 1 21 2 1 0 4 6 112 117 98 108 105 99 -94 -126 1 6 2 2 112 0 2
1 0 2 1 0 48 -127 -7 48 54 6 8 43 6 1 2 1 1 0 4 42 72 80 45 85 88 32 100 105 9
7 109 111 110 100 32 66 46 49 48 46 50 48 32 65 32 57 48 48 48 47 56 50 55 32 55
48 51 54 50 53 49 49 48 22 6 8 43 6 1 2 1 1 2 0 6 10 43 6 1 4 1 11 2 3 2 3 4
8 16 6 8 43 6 1 2 1 1 3 0 67 4 0 -3 127 97 48 58 6 8 43 6 1 2 1 1 4 0 4 46 83 12
1 115 116 101 109 32 65 100 109 105 110 105 115 116 114 97 116 111 114 32 40 115
121 115 97 100 109 105 110 64 99 101 46 107 109 105 116 108 46 97 99 46 116 104
41 48 34 6 8 43 6 1 2 1 1 5 0 4 22 100 105 97 109 111 110 100 46 99 101 46 107
109 105 116 108 46 97 99 46 116 104 48 38 6 8 43 6 1 2 1 1 6 0 4 26 67 111 109 1
12 117 116 101 114 32 69 110 103 105 110 101 101 114 105 110 103 32 68 101 112 1
16 46 48 13 6 8 43 6 1 2 1 1 7 0 2 1 72
```

รูปที่ 7.3 แสดงค่าที่ได้จากฐานข้อมูล MIB

## โปรแกรมในส่วนการแยกข้อมูลต่างๆ ที่ได้รับมาจากฐานข้อมูล MIB

```
StringBuffer strb=new StringBuffer();
StringBuffer desc=new StringBuffer();
String objid="";
String uptime="";
StringBuffer contact=new StringBuffer();
StringBuffer sysname=new StringBuffer();
StringBuffer location=new StringBuffer();
int service = 0;
for (i=0;i<packet.getLength();i++){ // ข้อมูลทั้งหมดที่จะ
    strb.append((char)Tbuffer[i]); // นำมาแยก
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
int posdesc;  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int numdesc;
if (Tbuffer[3]==2) {
posdesc=44;numdesc = Tbuffer[43];
else if((Tbuffer[4]==2)&&(Tbuffer[36]==6)) {
posdesc=48;numdesc = Tbuffer[47];
else {
posdesc=46;numdesc = Tbuffer[45];
    for (i=posdesc;i<(numdesc+posdesc);i++){
        desc.append((char)Tbuffer[i]);          // ข้อมูลที่แยกในส่วน sysDesc
int posobjid;
int numobjid;
posobjid = numdesc+posdesc+14;
numobjid = Tbuffer[numdesc+posdesc+14-1];
    for (i=posobjid+1;i<(posobjid+numobjid);i++){
        objid = objid + String.valueOf("."+Tbuffer[i]);
        objid = "1.3" + objid;          // ข้อมูลที่แยกในส่วน sysObjID
int posuptime;
int numuptime=0;
posuptime = posobjid+numobjid+14;
numuptime = Tbuffer[posobjid+numobjid+14-1];
byte[] timebuffer = new byte[numuptime];
    for (i=posuptime;i<(posuptime+numuptime);i++){
        timebuffer[i-posuptime] =Tbuffer[i];
    }
int shiftcount=0;
long shifted;
long tenuptime=0;
int c,cc;
long day,hour,min,sec,temptime;
for (c=numuptime-1;c>=0;c--)
{
    //loop for shift byte at buffer[c]
    shifted=timebuffer[c] & 255;
    for (cc=0;cc<shiftcount;cc++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        shifted=shifted << 8;
    }
    shiftcount++;
    tenuptime=tenuptime | shifted;
}
day = tenuptime / 8640000;
temptime = tenuptime % 8640000;
hour = temptime / 360000;
temptime = temptime % 360000;
min = temptime / 6000;
temptime = temptime % 6000;
sec = temptime / 100;
uptime = String.valueOf(day)+" days "+String.valueOf(hour)+" hours "+
String.valueOf(min)+" minutes "+String.valueOf(sec)+" seconds";
// ข้อมูลที่แยกในส่วน sysUpTime
int poscontact;
int numcontact;
poscontact = posuptime+numuptime+14;
numcontact = Tbuffer[posuptime+numuptime+14-1];
for (i=poscontact;i<(poscontact+numcontact);i++){
    contact.append((char)Tbuffer[i]); // ข้อมูลที่แยกในส่วน sysContact
int possysname;
int numsysname;
possysname = poscontact+numcontact+14;
numsysname = Tbuffer[poscontact+numcontact+14-1];
for (i=possysname;i<(possysname+numsysname);i++){
    sysname.append((char)Tbuffer[i]); // ข้อมูลที่แยกในส่วน sysName
int poslocation;
int numlocation;
poslocation = possysname+numsysname+14;
numlocation = Tbuffer[possysname+numsysname+14-1];
for (i=poslocation;i<(poslocation+numlocation);i++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int posservice;

int numservice;

posservice = poslocation+numlocation+14;

numservice = Tbuffer[poslocation+numlocation+14-1];

    for (i=posservice;i<(posservice+numservice);i++){

        service = Tbuffer[i];}                // ข้อมูลที่แยกในส่วน sysService

    NumberPort = packet.getPort();           // หมายเลข Port

    Lport = socket.getLocalPort();           // หมายเลข Local Port

    IPAdd = packet.getAddress();             // หมายเลข IP

    String IP = IPAdd.getHostAddress();

    System.out.println("SysDescription: "+desc);

    System.out.println("SysObjectID:  "+objid);

    System.out.println("SysUpTime:     "+uptime);

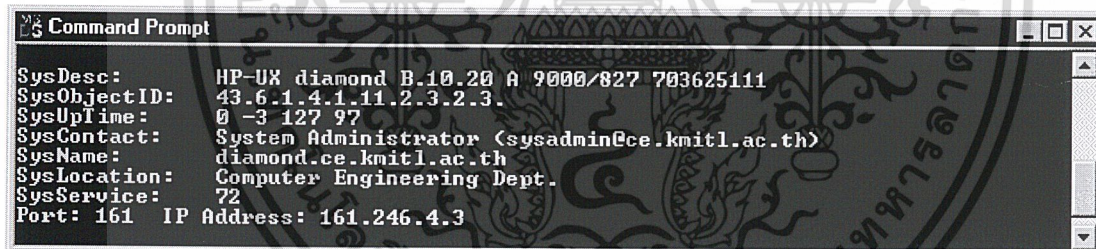
    System.out.println("SysContact:    "+contact);

    System.out.println("SysName:       "+sysname);

    System.out.println("SysLocation:   "+location);

    System.out.println("SysService:    "+service);

```



```

Command Prompt
SysDesc:      HP-UX diamond B.10.20 A 9000/827 703625111
SysObjectID: 43.6.1.4.1.11.2.3.2.3.
SysUpTime:   0 -3 127 97
SysContact:  System Administrator (sysadmin@ce.kmitl.ac.th)
SysName:     diamond.ce.kmitl.ac.th
SysLocation: Computer Engineering Dept.
SysService:  72
Port: 161 IP Address: 161.246.4.3

```

รูปที่ 7.4 แสดงผลการแยกข้อมูลต่างๆ ที่ได้รับจากฐานข้อมูล MIB

โปรแกรมในส่วนของการนำข้อมูลที่แยกได้เก็บไว้ในฐานข้อมูล MS SQL Server

```

String url = "jdbc:odbc:snmp";

    Connection con;

    Statement stmt;

    String query = "select IP, SysDesc from SystemMIB";

    try {

        Class.forName("jdbc.odbc.JdbcOdbcDriver");}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ (java.lang.ClassNotFoundException) { อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

System.err.print("ClassNotFoundException: ");
System.err.println(e.getMessage());
}
try {
    con = DriverManager.getConnection(url,"sa", "navy");
    stmt = con.createStatement();
    stmt.executeUpdate("update SystemMIB set SysDesc=" + desc+
        "", SysObjectID=" +objid+
        "", SysUpTime=" +uptime+
        "", SysContact=" +contact+
        "", SysName=" +sysname+
        "", SysLocation=" +location+
        "", SysService=" +service+ " where name =" +namesql+ "");
    ResultSet rs = stmt.executeQuery(query);
    System.out.println("IP and Description:");
    while (rs.next()) {
        String ip = rs.getString("IP");
        String de = rs.getString("SysDesc");
        System.out.println(ip + " " + de);
    }
    stmt.close();con.close();
}
catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
}

```

```

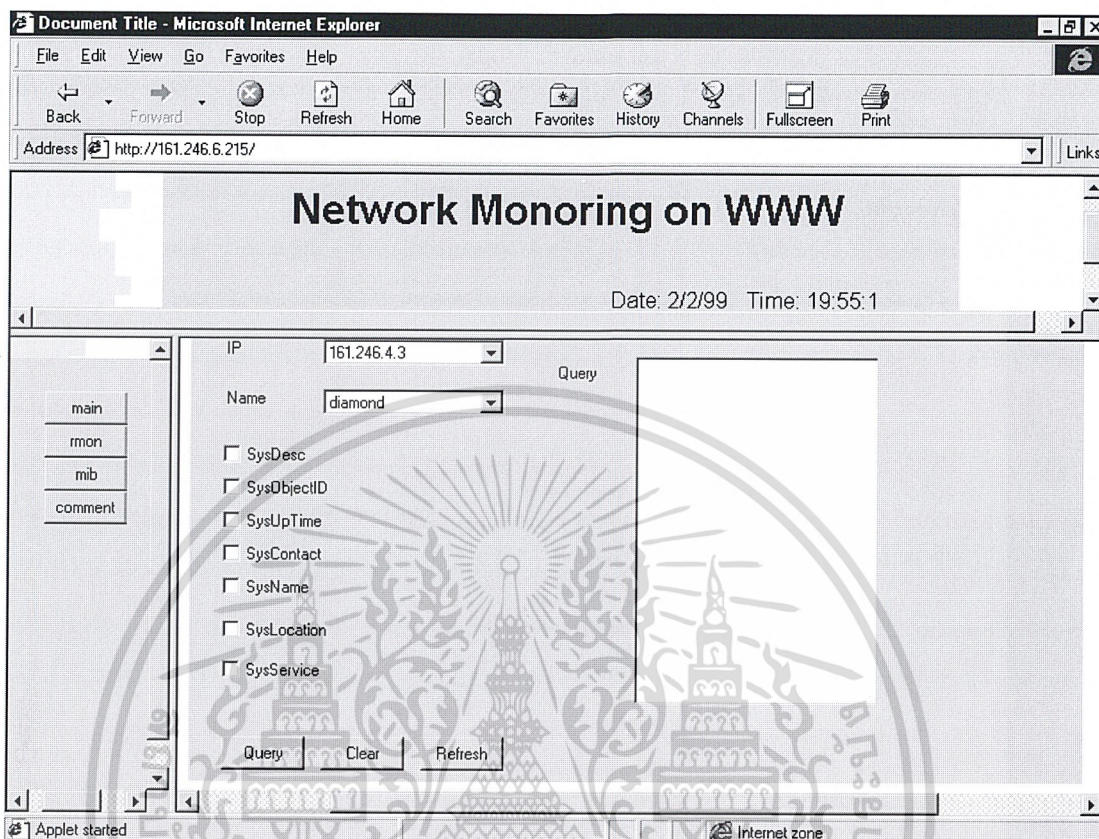
cmd - java Sys5
IP and Description:
161.246.29.1 NetWare 386 TCP/IP
161.246.10.22 SUNW,SPARCcenter-2000
161.246.10.21 SUNW,SPARCcenter-2000
161.246.5.95 3Com SuperStackII Switch 1000, SW Version:2.00
161.246.4.3 HP-UX diamond B.10.20 A 9000/827 703625111

```

รูปที่ 7.5 แสดงผลข้อมูลจาก MIB ที่เก็บในฐานข้อมูล MS SQL Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

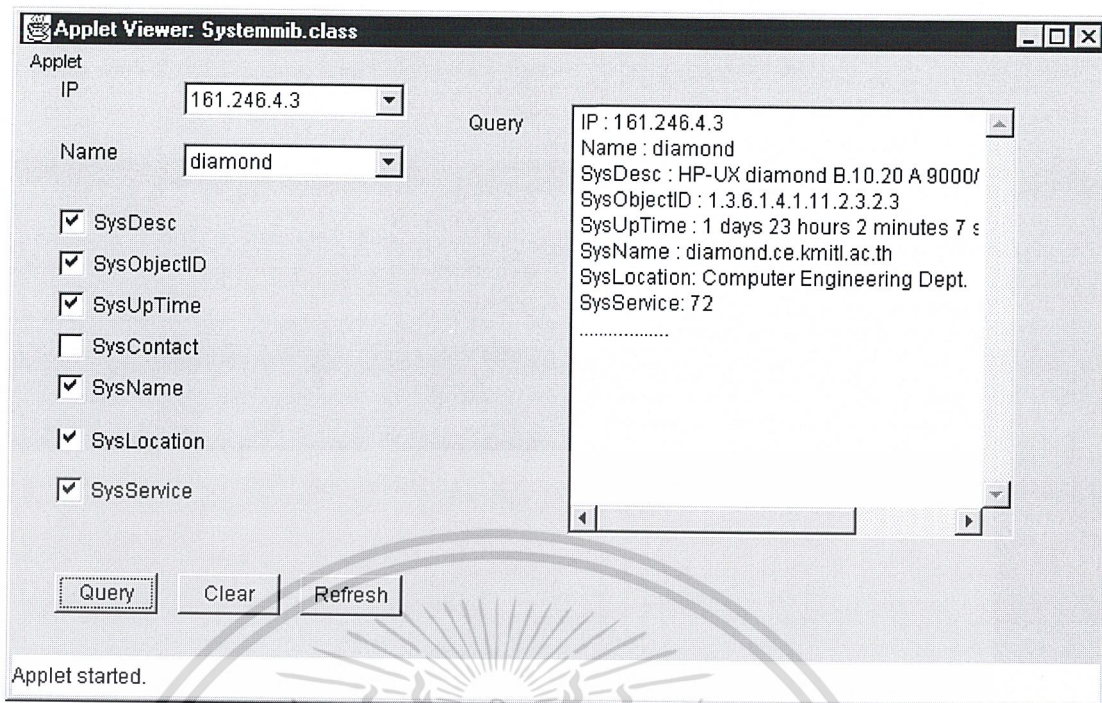
ส่วนที่ใช้ในการมอนิเตอร์คือนำเอาผลที่ได้จากการไปนำเอาค่าที่ใช้เก็บไว้ในฐานข้อมูลที่กำหนดไว้ ออกมาแสดงผลที่ เว็บเพจดังรูป



รูปที่ 7.6 แสดงเว็บเพจที่ใช้ในการมอนิเตอร์

เป็นหน้าตาของเว็บเพจที่มีการบริการเป็น web server ซึ่งจะทำการเปิดบริการทิ้งไว้เพื่อคอยทำการตรวจจับหรือมอนิเตอร์การบริการที่เกี่ยวกับทางด้าน SNMP ทั้ง rmon และ mib นอกจากนี้ยังจะมีการเปิดไว้บริการเพื่อขอข้อเสนอแนะ หรือคำติชมต่างๆ เกี่ยวกับบริการของ เว็บเพจสำหรับโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

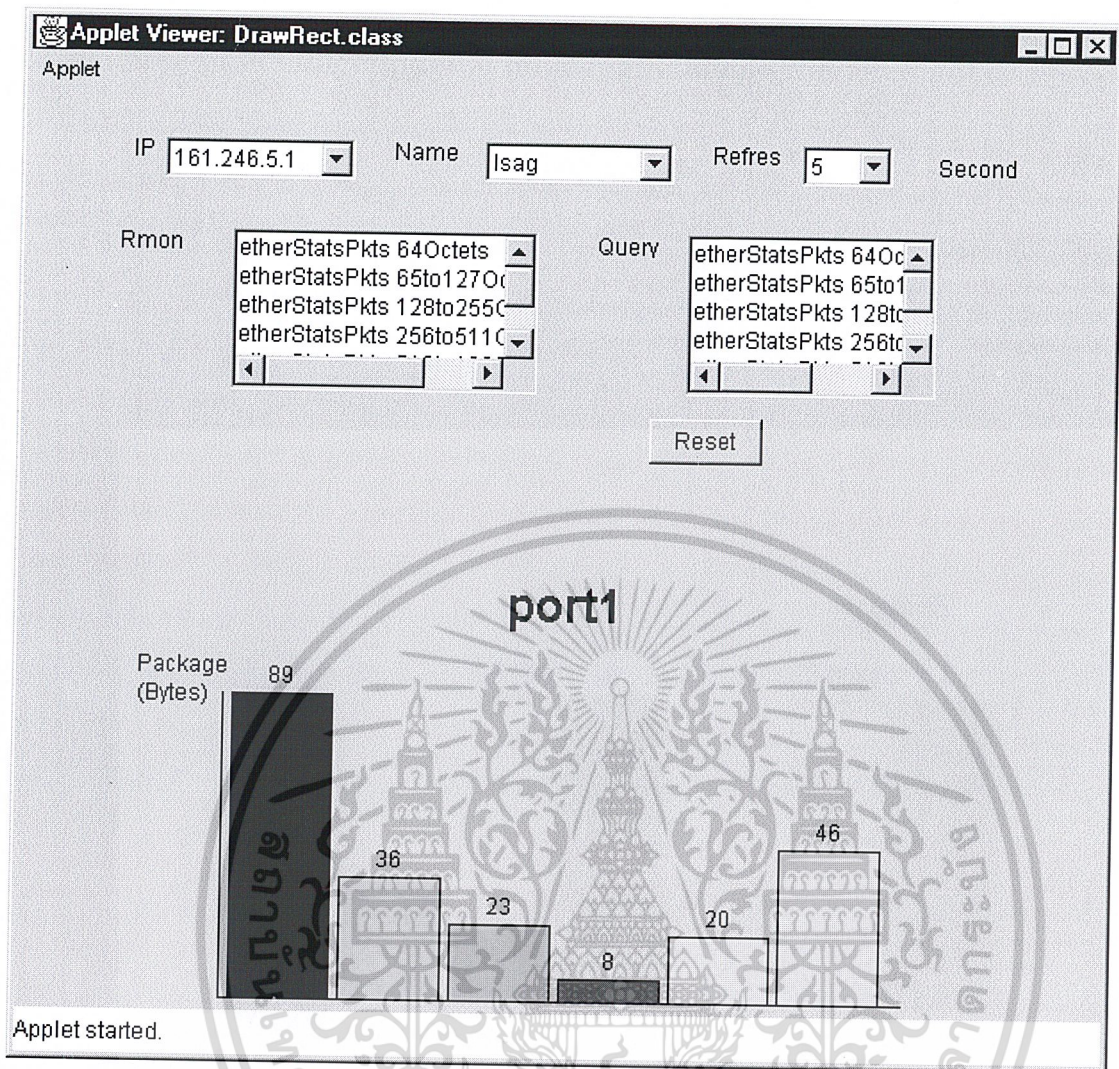


รูปที่ 7.7 แสดงเว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ mib

เป็นหน้าตาผลที่ได้ทำการมอนิเตอร์ในฐานะข้อมูล mib โดยทำการดึงค่าจากฐานข้อมูลออกมาแสดงผลตามผู้ใช้งานต้องการ โดยค่าที่อยู่ในฐานข้อมูลที่ได้ทำการออกแบบไว้ดังนี้

MIB
SysDesc
SysObjectID
SysUpTime
SysContact
SysName
SysLocation
SysService

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

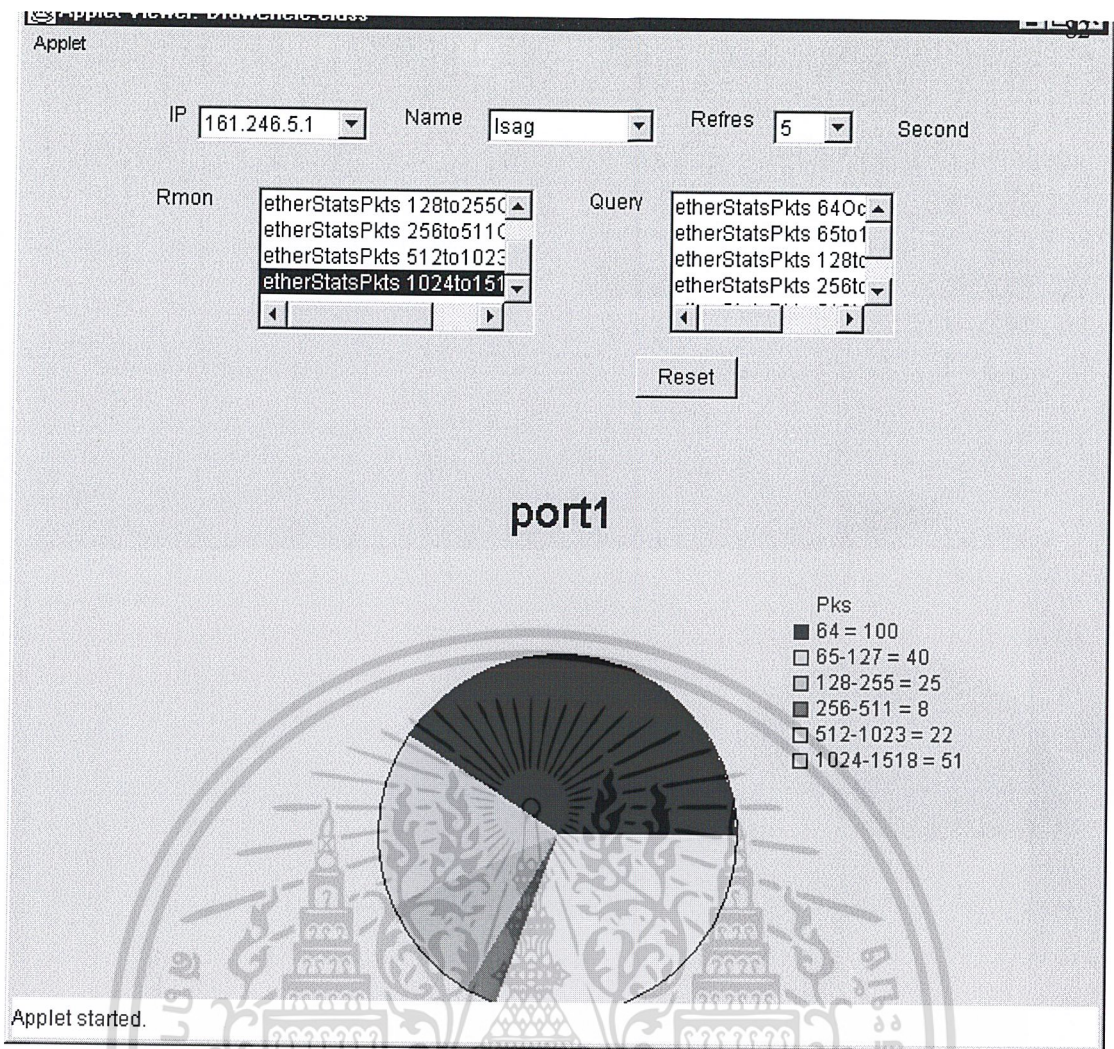


รูปที่ 7 แสดงเว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ rmon ในลักษณะการแฟ้ม

เป็นหน้าตาผลที่ได้ทำการมอนิเตอร์ในฐานข้อมูล rmon โดยทำการดึงค่าจากฐานข้อมูลออกมาแสดงผลตามผู้ใช้งานต้องการ โดยค่าที่อยู่ในฐานข้อมูลที่ได้ทำการออกแบบไว้ดังนี้

rmon
EtherstatsPktsat64Octate
EtherstatsPktsat65to127Octate
EtherstatsPktsat128to255O Octate
EtherstatsPktsat256to511O Octate
EtherstatsPktsat512to1023O Octate
EtherstatsPktsat1024to1518 Octate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงพาณิชย์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 แสดงเว็บเพจที่ใช้ในการมอนิเตอร์ในส่วนของ rmon ในรูปกราฟ วงกลม

เป็นหน้าตาผลที่ได้ทำการมอนิเตอร์ในฐานข้อมูล rmon โดยทำการดึงค่าจากฐานข้อมูลออกมาแสดงผลตามผู้ใช้งานต้องการ โดยค่าที่อยู่ในฐานข้อมูลที่ได้ทำการออกแบบไว้ดังนี้

rmon
EtherstatsPktsat64Octate
EtherstatsPktsat65to127Octate
EtherstatsPktsat128to255O Octate
EtherstatsPktsat256to511O Octate
EtherstatsPktsat512to1023O Octate
EtherstatsPktsat1024to1518 Octate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### บทวิจารณ์และสรุป

#### 8.1 สรุปผลการดำเนินงาน

โครงการการตรวจสอบสถานะระบบเครือข่าย โดยการใ้ภาษาจาวาจำลองตัวเองคล้ายกับ management station ในการติดต่อฐานข้อมูล MIB และ RMON จากอุปกรณ์ agent โดยใช้โปรโตคอล SNMP เพื่อนำเอาค่าสถานะของระบบเครือข่ายที่ได้รับจากอุปกรณ์ agent นั้นมาแล้วแยกข้อมูลต่าง ๆ ที่ได้ออกจากกัน แล้วจึงนำมาเก็บไว้ในฐานข้อมูล

ข้อมูลที่มีในฐานข้อมูล MIB และ RMON นี้มีมากมายหลายชนิดมาก ซึ่งทั้งหมดนั้นใช้สำหรับการจัดการระบบเครือข่ายทั้งสิ้น ในโครงการนี้เป็นวิธีการติดต่อฐานข้อมูลดังกล่าวให้ได้เท่านั้น ไม่ได้เก็บข้อมูลที่มีทั้งหมดของ MIB และ RMON ไว้

เมื่อนำเอาค่าที่ได้มาแล้วไปใส่ไว้ในฐานข้อมูลเป็นที่เรียบร้อยแล้วจึงจะทำการ มอนิเตอร์โดยการใช้ภาษาจาวาเพื่อทำการติดต่อกับฐานข้อมูล จากนั้นจึงทำการนำเอาค่าที่ได้ออกมาแสดงผลที่เว็บเพจ ดังที่ได้ทราบกันคืออยู่แล้วว่าโดยตัวภาษาจาวาเองนั้น สามารถทำงานบนเว็บเพจได้อยู่แล้วจึงสามารถทำให้ง่ายในการทำงานบน เว็บเบราว์เซอร์ ซึ่งตรงตามวัตถุประสงค์ของโครงการ คือต้องการให้มีการมอนิเตอร์ผ่านเว็บเพจ ดังที่ทราบกันโดยทั่วไปแล้วโดยบริการทางอินเทอร์เน็ตเองนั้น มีบริการของ WWW(World Wide Web) ซึ่งเป็นเครือข่ายที่กว้างไกลที่สุดในโลกในขณะนี้ และกำลังเป็นที่สนใจของนักเล่นคอมพิวเตอร์โดยทั่ว ๆ ไปในขณะนี้ อย่างไรก็ตามภาษาจาวาโดยตัวของมันเองไม่สามารถรันได้โดยตัวของมันเอง จำเป็นจะต้องอาศัยตัวที่ช่วยในการรันไม่ว่าจะเป็น จาวาแอปพลิเคชัน หรือ จาวาแอปเพลท นอกจากนี้จาวายังเป็นภาษาใหม่ต่างจากภาษาอื่น ๆ ได้แก่ Delphi, Visual C, Visual Basic ดังนั้นในการรันของภาษาจาวาจะช้ามากเมื่อเทียบกับภาษาอื่น ๆ เพราะมีการรันแบบอ่านคำสั่งทีละคำมาแปลเป็นภาษาเครื่อง

ข้อดีของโครงการนี้เพื่อที่จะช่วยให้ administrator หรือผู้ใช้อื่น ๆ ที่สนใจในระบบเครือข่ายสามารถที่จะตรวจสอบสถานะระบบเครือข่ายของตนได้ และกรณีที่โครงการนี้สร้างฐานข้อมูล SQL Server ขึ้นมาเก็บข้อมูลจากฐานข้อมูล RMON และ MIB ก็เพื่อที่จะได้จัดการข้อมูลในฐานข้อมูล SQL Server ได้ (ไม่สามารถจัดการฐานข้อมูล RMON และ MIB ได้) แต่ข้อเสียคือฐานข้อมูล RMON และ MIB นั้นมีอยู่มากมาย ซึ่งในฐานข้อมูล SQL Server ในโครงการไม่ได้มีแนวทางที่จะต้องการเก็บฐานข้อมูล RMON และ MIB ทั้งหมด ดังนั้น ผู้ใช้จึงสามารถตรวจสอบสถานะภาพของเครือข่ายได้แค่เฉพาะที่มีอยู่ในฐานข้อมูล SQL Server เท่านั้นไม่สามารถที่จะเห็นฐานข้อมูล RMON และ MIB ทั้งหมดได้

#### 8.2 แนวทางการพัฒนาต่อ

ในปัจจุบันได้มีหลาย ๆ บริษัทเสนอตัวโปรแกรมที่ใช้ในการพัฒนาภาษาจาวาแบบต่าง ๆ ขึ้นมากมายดังที่ได้กล่าวมา จึงทำให้เป็นการง่ายในการพัฒนาตัวภาษาจาวาเอง โครงการที่ได้สร้างขึ้นมาครั้งนี้จะมีส่วนช่วยในการตรวจสอบสถานะของระบบเครือข่าย การเดินทางของแพ็กเก็ตต่างภายในระบบซึ่งเกิดขึ้นในลักษณะนั้นทางคณะผู้จัดทำคาดหวังไว้ว่าโครงการนี้จะเป็นประโยชน์อย่างยิ่ง

## หนังสืออ้างอิง

### หนังสืออ้างอิงหนังสือภาษาต่างประเทศ

1. core java second editon gary cornll,cay s,horstamann
- 2.Designing TCP/IP Internetworks, Geoff Bennett, A Division of International Thomson Publishing Inc.
3. Managing internetworks with SNMP : the definitive guide to the Simple Network Management Protocol (SNMP) and SNMP version 2, Mark A.Miller, Imprint New York: M&T Books
4. William Stallings : SNMP, SNMPv2, and RMON Practical Network Management, Addison Wesley Publishing Company
5. Rose, M.T., The Simple Book : An Introduction to Management of TCP/IP based Internets, Prentice-Hall, Engwood Cleffs, NJ., 1990
6. William Stallings : SNMP, SNMPv2, and CMIP The Practical Guide to Network Management Standards. Fifth printing, 1994

### หนังสืออ้างอิงภาษาไทย

1. เรียนรู้การสร้างโฮมเพจด้วย HTML ปิยวิฑ เจนกิจจาไพมูลย์
2. วารสารไมโครคอมพิวเตอร์, Intelligence Network ฉบับที่ 142
3. ปริชญานิพนธ์ เรื่องการบริหารเครือข่ายโดยใช้ HP Open View Network Node Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### IDS Server

IDS Server คือ Server ที่ใช้ในการติดต่อกับ Internet Database ซึ่งทำงานได้ทั้ง HTML และ Java ในการเขียน Web Pages เพื่อติดต่อกับฐานข้อมูลสำหรับโปรแกรมประยุกต์ในอินเทอร์เน็ตและอินทราเน็ต และ IDS Server นั้นสามารถติดต่อผ่าน ODBC บน Windows NT และ Windows 95. ฐานข้อมูลนั้นสามารถที่จะนำไปใช้ได้รวมถึง Oracle, Sybase, SQL Server และระบบฐานข้อมูลที่โด่งดังอื่นๆ ด้วย

IDS Server มี API ของจาวาที่ใช้ติดต่อกับ JDBC ด้วยคือ IDS JDBC Driver. โปรแกรมประยุกต์ใช้ภาษาจาวาสามารถใช้ JDBC API นี้ในการติดต่อกับฐานข้อมูลโดยผ่าน IDS Server. IDS JDBC Driver นี้มีคุณสมบัติมากมายทั้งยังมีขนาดเล็กมีประสิทธิภาพสูง โดยใช้ Type-3 JDBC driver. เพราะว่า JDBC driver นี้สามารถทำงานได้ทั้ง Java applets and Java programs. และยังสามารถใช้กับบราวเซอร์ทุกประเภทที่สามารถทำงานกับภาษาจาวาได้ (All Java WebBrowser, JDK ทุกรุ่น (Version), Microsoft SDK สำหรับภาษา Java

คุณสมบัติอื่นๆ ที่ถือว่าเป็นจุดเด่นหนึ่งของ IDS Server ที่ถือว่ามีเสน่ห์ น่าสนใจ พอสรุปได้ดังต่อไปนี้

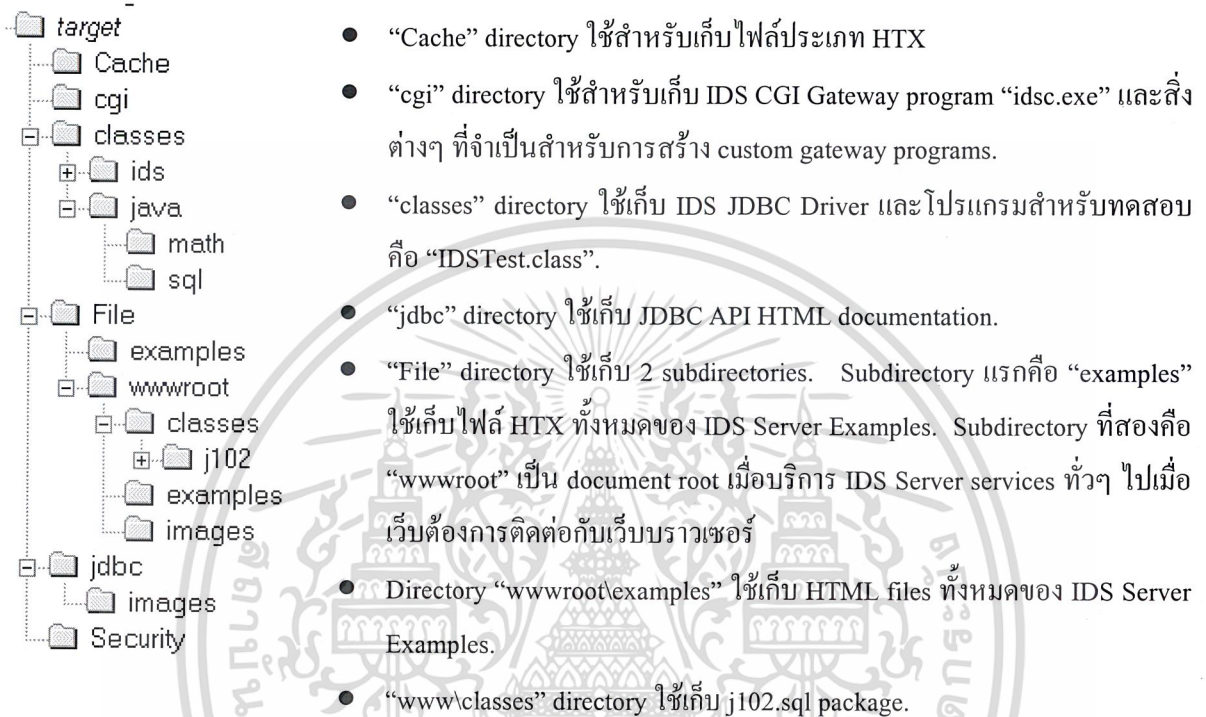
- Define/prepare/execute any SQL statement and stored procedures
- Seamless HTML table integration
- Create master/detail reports
- View results in small groups
- Easy onbreak processing
- Conditional HTML output
- Dynamic SQL query
- Flexible variable handling
- Use JavaScript to process results
- Heterogeneous database access
- Transaction control
- Integrated IDS JDBC Driver
- Public-Key and Secret-Key Secure JDBC
- Complete firewall support for JDBC applications
- JDBC ResultSet Caching
- Supports Zero Installation Applet Deployment

#### การติดตั้ง IDS Server

IDS Server เป็นไฟล์ที่ขยายตัวเองโดยอัตโนมัติ (one self-extracting executable file). โดยเรียกใช้คำสั่ง “IDSSxxx.EXE” ในการติดตั้ง IDS Server, ดับเบิลคลิกกับไอคอนและทำงานตามขั้นตอนทีละขั้นตอน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาไปใช้ประโยชน์ด้านการค้าตามหน้าจอ. ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. Directory Hierarchy

ในการติดตั้ง IDS Server จะสร้างลำดับไดเรกทอรี (directory hierarchy) ภายใต้ไดเรกทอรีที่ผู้ติดตั้งเป็นผู้เลือก (Target Directory) ดังต่อไปนี้



รูปที่ 1 – Directory Hierarchy

## 2. Install Databases

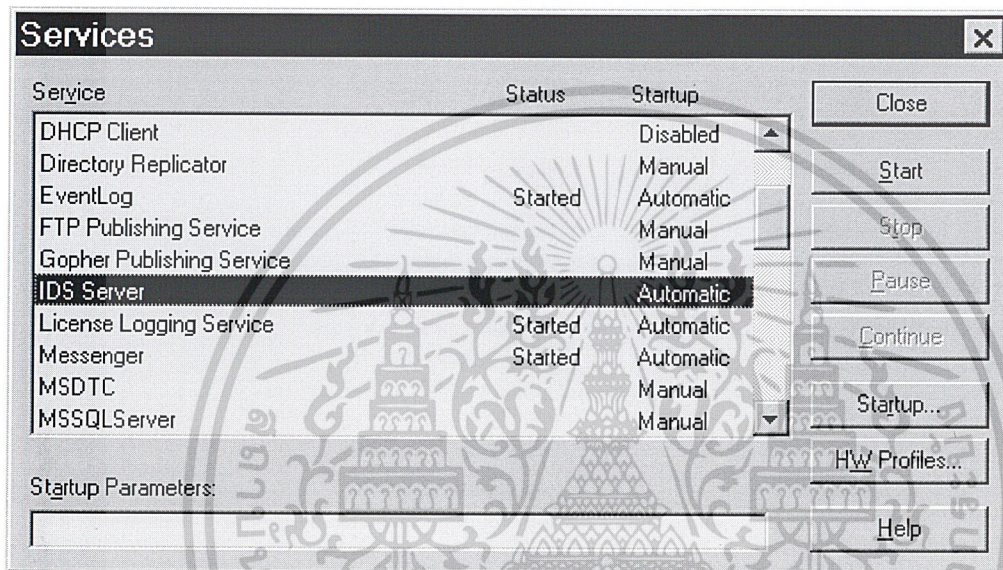
ก่อนการทดสอบโปรแกรม ผู้ใช้ควรจะต้องติดตั้ง ODBC system data source ตามขั้นตอนต่อไปนี้

- 1) เปิด Control Panel และคลิกที่ไอคอน 32-bit ODBC icon.
- 2) คลิกบนปุ่ม “System DSN” button at the bottom.
- 3) คลิกบนปุ่ม “Add”
- 4) เลือก Driver ยกตัวอย่างเช่น “Microsoft Access Driver (\*.mdb)” คลิก OK
- 5) คลิกบนปุ่ม “Select...” แล้วใช้ dialog เพื่อหาไฟล์ Database
- 6) ที่ฟิลด์ “Data Source Name:” ให้ใส่ชื่อเพื่ออ้างอิงถึง ODBC Driver นี้
- 7) คลิกปุ่ม “OK” และคลิกปุ่ม “Close”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. Running IDS Server

สำหรับ Windows NT<sup>1</sup>, IDS Server จะทำงานบน NT service. ใน Windows NT 4.0, การติดตั้งนั้นใน “Start” menu – “Programs” folder – “IDS Server” folder, a shortcut called “Shortcut to Services”. ให้เลือก shortcut นี้บริการที่ใช้ในการควบคุม window จะปรากฏขึ้น. ใน Windows NT 3.51, ผู้ใช้ยังต้องค้นหาใน Control Panel ใน Main program group และคลิกที่ไอคอน “Services” icon เพื่อที่จะให้บริการในการควบคุม window นั้นทำงาน



รูปที่ ก – 2 Running IDS Server บน Windows NT

ใน main list box, highlight “IDS Server”. ให้คลิกเมาส์ข้างขวามุม “Start” เพื่อที่จะเริ่มต้นการทำงานของ server . ในการติดตั้ง IDS Server “Automatic” ให้คลิกไปที่ปุ่ม “Startup...” เพื่อเข้าไปแก้ไขยัง option ต่อไป.

สำหรับ Windows 95<sup>1</sup>, IDS Server จะทำงานด้วย console program. การติดตั้งจะถูกสร้างใน “Start” menu – “Programs” folder – “IDS Server” folder, a shortcut called “IDS Server”. ถ้าเลือก shortcut จะเริ่มต้นการทำงานของ IDS Server ใน console window. ผู้ใช้สามารถจะสั่งเริ่มต้นการทำงานด้วย command line ก็ได้.

```
C:\>installation_path\idss -con ↵
```

```
IDS Server Version 2.5 Lite Copyright (c) 1997 IDS Software.
```

```
IDS Server started...
```

ในการจบการทำงานของ IDS Server ใน Windows 95<sup>1</sup> นั้น, ผู้ใช้ก็เพียงแค่กดปุ่ม Ctrl-C ใน console window หรือปิด console window ลงเท่านั้นaltogether.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กระบวนการของ JDBC

กระบวนการของ IDS Server ในการติดต่อฐานข้อมูลสำหรับโปรแกรมภาษาจาวานั้นคือใช้ IDS JDBC Driver. Driver นี้มีพื้นฐานมาจาก JDBC API ล่าสุดใน JDK 1.1.

### 1. Overview

JavaSoft, เป็นผู้สร้าง JDBC API, ประเภทของ JDBC drivers แบ่งเป็น 4 ประเภทดังต่อไปนี้

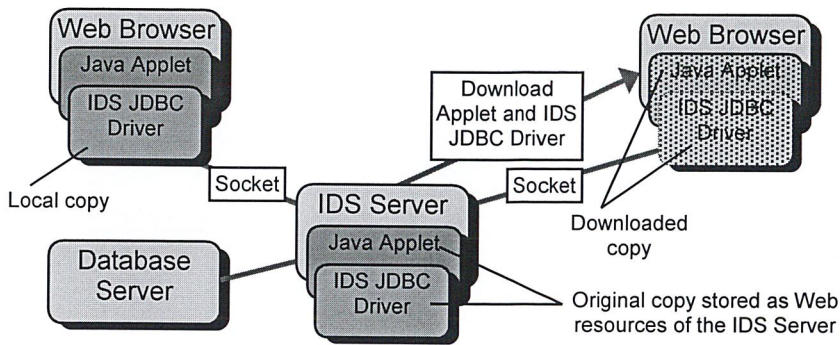
- 1) JDBC-ODBC bridge
- 2) native-API partly-Java driver
- 3) net-protocol all-Java driver
- 4) native-protocol all-Java driver.

JDBC Driver ของ IDS Server ถือว่าเป็น Driver ชนิดที่ 3 “3. *net-protocol all-Java driver* ในการแปลงการเรียกใช้ JDBC ไปยัง DBMS โดย protocol อีกระยะที่ซึ่งการแปลงไปสู่ DBMS protocol นั้นทำโดย server. Net server middleware สามารถที่จะติดต่อไปยัง Java clients และได้ในหลายๆ ฐานข้อมูล. ข้อกำหนดของ Protocol ในการใช้ขึ้นอยู่กับตัวแทนจำหน่าย (vendor) โดยทั่วไปแล้วทางเลือกของ JDBC นั้นจะยืดหยุ่นมาก. มันจึงดูเหมือนตัวที่ตัวแทนจำหน่ายจะจัดการให้ผลิตภัณฑ์ของตนนั้นเหมาะสมสำหรับการใช้งานบนเครือข่าย ซึ่งยังต้องเพิ่มความต้องการสำหรับความปลอดภัย (security) , เพื่อใช้ในการผ่าน ระบบรักษาความปลอดภัย (firewalls), และอื่นๆ อีกมากมาย

### 2. IDS JDBC Driver

IDS JDBC Driver เป็น Driver ประเภทที่ 3. การใช้ net server middleware โดย IDS JDBC Driver คือ IDS Server, ที่ซึ่งถือว่าเป็นที่ทำงานที่แท้จริงหลังจาก IDS JDBC Driver. การติดต่อไปยังฐานข้อมูลจะทำโดย driver เพื่อที่จะส่งกลับไปมาระหว่าง driver และ IDS Server. JDBC driver จำทำการตัดสินใจเลือก Java Socket ในการเชื่อมต่อ (connection) กลับไปยัง IDS Server. รูปที่ ก – 3 แสดงให้เห็นถึงรูปแบบของ network ในการติดต่อไปยัง IDS Server, database server และ 2 เว็บเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 รูปแบบ Network ของ IDS Server

IDS JDBC Driver คือ Java software package ที่ใช้อธิบายถึง JDBC API. Driver จะถูกเขียนด้วยภาษาจาวาเพียงอย่างเดียว ดังนั้น Platform ของ Drivers จึงเป็นอิสระและสามารถกระจายโดย Web servers หรือติดตั้งบน local Java package ใน client computers ทั่วไป. ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข

### Open Database Connectivity (ODBC)

#### ODBC คืออะไร

Open Database Connectivity คือวิธีการติดต่อและเข้าถึงจากแอปพลิเคชันสู่ระบบจัดการฐานข้อมูล โดยใช้ภาษา SQL เป็นมาตรฐานการเข้าถึงฐานข้อมูล ความสามารถในการต่อเชื่อมแบบนี้ทำให้แอปพลิเคชันสามารถเข้าถึงฐานข้อมูลได้หลายรูปแบบ ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนาโปรแกรมไปได้โดยไม่ต้องทำการระบุชนิดของระบบจัดการฐานข้อมูล

แต่เดิมนั้นการพัฒนาโปรแกรมประยุกต์ที่ใช้งานเกี่ยวกับฐานข้อมูล การเข้าใช้ฐานข้อมูลของโปรแกรมเหล่านี้จะทำการเรียกใช้ SQL แบบฝังตัว (Embedded SQL) ซึ่งในขณะนั้นวิถีทางแบบนี้ก็จะไปได้ดีทีเดียว เพราะว่าตัวโปรแกรมสามารถทำการเปลี่ยนรูปแบบของระบบไม่ว่าจะเป็นทางด้านฮาร์ดแวร์หรือซอฟต์แวร์ได้หลายรูปแบบ รวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มีการย้ายระบบ)

อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่ความแตกต่างกัน เช่นการเรียกใช้ข้อมูลของออราเคิลจากไมโครซอฟท์เอ็กเซล (Microsoft Excel) วิธีการเข้าถึงข้อมูลแบบเดิมนั้นจะต้องทำการพรีคอมไพล์โค้ดของเอ็กเซลและออราเคิล โดยใช้ IBM precompiler และ Oracle precompiler ตามลำดับ ซึ่งจะเห็นว่าเป็นการยุ่งยากมากทีเดียว

วิธีการต่อเชื่อมแบบ ODBC จะให้ความสะดวกในการติดต่อข้อมูลมากกว่าวิธีการดั้งเดิมโดยการกำหนดมาตรฐานการต่อเชื่อมของข้อมูล (Data protocol, DBMS capability) และแนวทางนี้ได้ทำให้เกิดความคิดที่จะสร้างไครีเฟอว์สำหรับการติดต่อกับของฐานข้อมูลขึ้นมา (DLL)

#### ทฤษฎีการทำงานของ ODBC

ส่วนติดต่อของการเชื่อมต่อฐานข้อมูลเปิด (ODBC : Open Database Connectivity) เป็นตัวทำให้โปรแกรมประยุกต์สามารถที่จะเข้าถึงข้อมูลในระบบจัดการฐานข้อมูล (DBMS : Database Management System) โดยใช้ภาษา SQL (Structure Query Language) เป็นมาตรฐานหลักในการเข้าถึงข้อมูล

#### ข้อดีของการติดต่อโดยใช้ ODBC

1. ฟังก์ชันของ ODBC อนุญาตให้แอปพลิเคชันติดต่อกับระบบจัดการฐานข้อมูลได้โดยสะดวก (การทำคำสั่ง SQL และการรับผลลัพธ์)
2. ใช้ภาษา SQL ตามมาตรฐาน SQL CAE, X/Open และ SQL Access Group (SAGX)
3. มีการกำหนดการส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตรฐานเดียวกัน
4. เป็นวิธีการมาตรฐานในการติดต่อกับระบบจัดการฐานข้อมูล
5. มีการกำหนดชนิดข้อมูล (Data Type) เป็นมาตรฐาน
6. ชุดคำสั่ง SQL สามารถกำหนดได้แม้ในขณะที่ Runtime

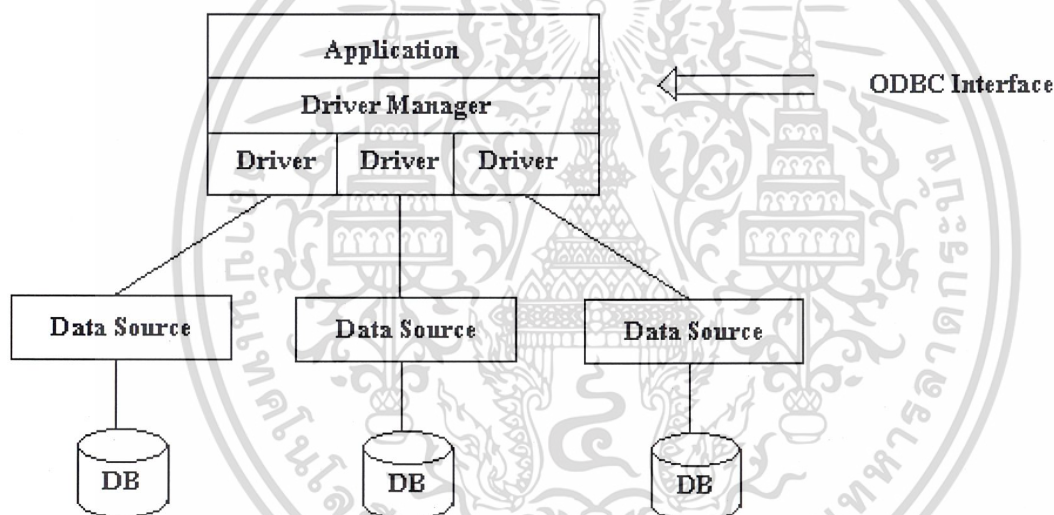
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
7. สามารถเขียนโปรแกรมชุดเดียว แต่สามารถเข้าใช้ระบบจัดการฐานข้อมูลได้หลายตัว  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ตัวโปรแกรมไม่ต้องรับผิดชอบในการดูแลการติดต่อข้อมูลกับระบบจัดการฐานข้อมูล,
9. ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

### องค์ประกอบของ ODBC

สถาปัตยกรรมของ ODBC ประกอบด้วย 4 ส่วนสำคัญ

1. แอปพลิเคชัน ทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของ ODBC ตามคำสั่งภาษา SQL พร้อมทั้งทำการรับผลลัพธ์ด้วย
2. ตัวจัดการไดรฟ์เวอร์ (Driver Manager) ทำหน้าที่โหลดไดรฟ์เวอร์เชื่อมต่อกับแหล่งข้อมูล
3. ไดรฟ์เวอร์ (Driver) ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของ ODBC ,ส่งคำสั่ง SQL ไปสู่แหล่งข้อมูลที่ถูกต้อง และทำการแปลงคำสั่งที่ส่งมาให้อยู่ในรูปแบบที่สนับสนุนคยระบบจัดการฐานข้อมูลแต่ละชนิดอีกด้วย
4. แหล่งข้อมูล (Data Source) เป็นแหล่งข้อมูลที่ใช้ต้องการเข้าถึง



รูปที่ ข-1 องค์ประกอบของ ODBC

### แอปพลิเคชัน

ตัวโปรแกรมจะเรียกใช้การต่อเชื่อม ODBC ในการทำงานต่อไปนี้

1. ร้องขอการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำสั่ง SQL สู่มแหล่งข้อมูล
3. กำหนดพื้นที่การจัดเก็บและรูปแบบของข้อมูลที่เป็นผลลัพธ์จาก SQL Request ร้องขอผลลัพธ์
4. ร้องขอผลลัพธ์
5. ประมวลผลและจัดการกับข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 6. รายงานผลให้กับผู้ใช้ (ถ้าจำเป็น)  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ร้องขอการ Commit หรือ Rollback สำหรับควบคุมการประมวลผล Transaction
8. ยกเลิกการติดต่อกับแหล่งข้อมูล

### ตัวจัดการไดรฟ์เวอร์

ตัวจัดการไดรฟ์เวอร์ คือ DLL (Dynamic Link Library) และไลบรารีอื่น ๆ หน้าที่หลักของตัวจัดการไดรฟ์เวอร์ก็คือการโหลดไดรฟ์เวอร์ ส่วนหน้าที่อื่น ๆ ก็มีดังนี้

1. เรียกใช้ไฟล์ ODBC.INI เพื่อกำหนดชื่อของแหล่งข้อมูล (Data Source name) ให้กับไดรฟ์เวอร์ DLL
2. ทำการประมวลผลการเริ่มต้นการเชื่อมต่อของ ODBC
3. เป็นจุดต่อเชื่อมระหว่างฟังก์ชันของ ODBC กับไดรฟ์เวอร์แต่ละตัว
4. ทำการตรวจสอบพารามิเตอร์และลำดับการเรียกใช้ ODBC

### ไดรฟ์เวอร์

ไดรฟ์เวอร์คือ DLL ที่สร้างฟังก์ชันของ ODBC และทำการโต้ตอบกับแหล่งข้อมูลไดรฟ์เวอร์ทำการตอบสนองการเรียกใช้ฟังก์ชันของ ODBC โดยจะทำงานต่อไปนี้

1. สร้างการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำขอร้องให้กับแหล่งข้อมูล
3. แปลงข้อมูลจากรูปแบบหนึ่งสู่อีกรูปแบบหนึ่ง
4. ส่งผลลัพธ์กลับให้แอปพลิเคชัน
5. จัดการส่งข้อมูลความคิดพลาตให้อยู่ในรูปแบบรหัสมาตรฐานแล้วส่งกลับไปให้แอปพลิเคชัน
6. ทำหน้าที่จัดการและดูแลเคอร์เซอร์ (Cursor)

### แหล่งข้อมูล (Data Source)

แหล่งข้อมูล หมายถึงการรวมกันของระบบจัดการฐานข้อมูล, ระบบปฏิบัติการ และระบบเครือข่าย โดยมีการชี้เฉพาะชนิด และประเภทลงไป หรืออีกนัยหนึ่งก็หมายความว่าการทำงานที่แอปพลิเคชันทำการติดต่อกับระบบจัดการฐานข้อมูล ยี่ห่อหนึ่งบนระบบปฏิบัติการหนึ่งและเข้าถึงโดยระบบเครือข่ายชนิดหนึ่ง เช่น ออรา เกล็ทที่วิ่งบนระบบปฏิบัติการ OS/2 โดยใช้ระบบเครือข่ายของ Novell Netware เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### SQL Syntax

#### SELECT Statement

```
SELECT [ALL | DISTINCT] select_list
    [INTO [[database.]owner.]table_name]
    [FROM
        [[database.]owner.]{table_name | view_name} [HOLDLOCK]
        [, [[database.]owner.]{table_name | view_name} [HOLDLOCK]]...]
    [WHERE search_conditions]
    [GROUP BY [ALL]
        aggregate_free_expression
        [, aggregate_free_expression]...]
    [HAVING search_conditions]
    [ORDER BY
        {[[[database.]owner.]{table_name. | view_name.}] column_name |
        select_list_number | expression [ASC | DESC]
        [, [[[database.]owner.]{table_name. | view_name.}] column_name |
        select_list_number | expression} [ASC | DESC]]...]
    [COMPUTE
        row_aggregate(column_name)
        [, row_aggregate(column_name)...]
    [BY column_name [, column_name]...]
    [FOR BROWSE]
```

#### INSERT Statement

```
INSERT [INTO]
    [[database.]owner.]{table_name | view_name}
    [(column_list)]
    {VALUES (constant_expression [, constant_expression]..) | select_statement}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## UPDATE Statement

```
UPDATE [[database.]owner.]{table_name | view_name}
SET [[[[database.]owner.]{table_name. | view_name.}]
    column_name1 = {expression1 | NULL | (select_statement)}
    [,column_name2 = {expression2 | NULL | (select_statement)}...]
[WHERE search_conditions]
```

## DELETE Statement

```
DELETE [FROM] [[database.]owner.]{table_name | view_name}
[WHERE search_conditions]
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ง

### Source Code

```
import java.net.*;
import java.io.*;
import java.sql.*;
import Alarm;
public class Sys5 extends Thread{

protected DatagramSocket socket;

public Sys5() throws IOException {
    //socket = new DatagramSocket();
}

public void execute() throws IOException {
    int numhost=1;int n=1;
    String host,sh_host;
    while(true) {
        try
        { Thread.sleep(1000);}
        catch(InterruptedException e) {}
        if(numhost==1) {host="161.246.4.3";sh_host="diamond";numhost++;}
        else if(numhost==2) {host="161.246.10.21";sh_host="chaokhun";numhost++;}
        else if(numhost==3) {host="161.246.5.95";sh_host="isag-hub";numhost++;}
        else if(numhost==4) {host="161.246.5.1";sh_host="isag01";numhost++;}
        else if(numhost==5) {host="161.246.6.1";sh_host="compnet";numhost++;}
        else if(numhost==6) {host="161.246.29.1";sh_host="elec1";numhost++;}
        else {host="161.246.10.22";sh_host="khaesad";numhost=1;}

        socket = new DatagramSocket();
        Sendpacket(host);
        receive(sh_host);
        socket.close();
        System.out.println("n : "+n);n=n+1;
    }
}

protected void Sendpacket(String host)
throws IOException {
    int i;
    byte Temp[] = {
        48,116,2,1,0,4,6,112,117,98,108,105,99, //13
        -95,103,2,2,112,0,2,1,0,2,1,0,48,91, //14
        48,11,6,7,43,6,1,2,1,1,1,5,0, //13
        48,11,6,7,43,6,1,2,1,1,2,5,0, //13
        48,11,6,7,43,6,1,2,1,1,3,5,0, //13
        48,11,6,7,43,6,1,2,1,1,4,5,0, //13
        48,11,6,7,43,6,1,2,1,1,5,5,0, //13
        48,11,6,7,43,6,1,2,1,1,6,5,0,
        48,11,6,7,43,6,1,2,1,1,7,5,0}; //13

    String PDUTemp = "";
    byte PDUTempP[] = new byte[1024];
    for (i = 0;i<118;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        PDUTemp = PDUTemp + String.valueOf(Temp[i]);
        PDUTempP[i]= Temp[i];
    }

    int port = 161; int Length = 118;
    DatagramPacket PDUMessage =
    buildPacket(PDUTempP,InetAddress.getByName(host),port,Length);
    socket.send(PDUMessage);
        alarm = new Alarm ();
        alarm.start();
        System.out.println("Port " +String.valueOf(port) + " IP Address: " + InetAddress.getByName
    (host).toString());
        System.out.println("");
        System.out.println("");
    }

```

```

public Alarm alarm;
protected void receive(String sh_host) throws IOException {
    int NumberPort,i,Lport;
    byte buffer[] = new byte[1024];
    InetAddress IPAdd ;
    String result="";

    DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
    socket.receive(packet);
    alarm.stop();
    // get Data
    byte[] Tbuffer = new byte[packet.getLength()];
    Tbuffer = packet.getData();
    for (i=0;i<packet.getLength();i++)
        result = result + String.valueOf(Tbuffer[i]) + " ";

    StringBuffer strb=new StringBuffer();
    StringBuffer desc=new StringBuffer();
    String objid="";
    String uptime="";
    StringBuffer contact=new StringBuffer();
    StringBuffer sysname=new StringBuffer();
    StringBuffer location=new StringBuffer();
    int service = 0;
        for (i=0;i<packet.getLength();i++){
            strb.append((char)Tbuffer[i]);} //strb
    int posdesc;
    int numdesc;
    if (Tbuffer[3]==2) {
        posdesc=44;numdesc = Tbuffer[43];} // for normal host
    else if((Tbuffer[4]==2)&&(Tbuffer[38]==2)) {
        posdesc=45;numdesc = Tbuffer[44];} //for isag01
    else if((Tbuffer[4]==2)&&(Tbuffer[39]==2)) {
        posdesc=46;numdesc = Tbuffer[45];} //for diamond
    else if((Tbuffer[4]==2)&&(Tbuffer[40]==2)) {
        posdesc=47;numdesc = Tbuffer[46];}
    else if((Tbuffer[4]==2)&&(Tbuffer[41]==2)) {
        posdesc=48;numdesc = Tbuffer[47];} //for malib
    else {
        posdesc=1;numdesc = Tbuffer[0];}
    for (i=posdesc;i<(numdesc+posdesc);i++){
        desc.append((char)Tbuffer[i]);} //sysDesc
}

```

เอกสารนี้เป็นเอกสาร

ของสำนักงานคณะกรรมการการอุดมศึกษาและทบวงมหาวิทยาลัย ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ หักสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int posobjid;
int numobjid;
posobjid = numdesc+posdesc+14;
numobjid = Tbuffer[numdesc+posdesc+14-1];
    for (i=posobjid+1;i<(posobjid+numobjid);i++){
        objid = objid + String.valueOf("."+Tbuffer[i]);}
    objid = "1.3" + objid;//sysObjID

int posuptime;
int numuptime=0;
posuptime = posobjid+numobjid+14;
numuptime = Tbuffer[posobjid+numobjid+14-1];
byte[] timebuffer = new byte[numuptime];
    for (i=posuptime;i<(posuptime+numuptime);i++){
        timebuffer[i-posuptime] =Tbuffer[i];
    }

int shiftcount=0;
long shifted;
long tenuptime=0;
int c,cc;
long day,hour,min,sec,temptime;
for (c=numuptime-1;c>=0;c--){
    {
        //loop for shift byte at buffer[c]
        shifted=timebuffer[c] & 255;
        for (cc=0;cc<shiftcount;cc++)
        {
            shifted=shifted << 8;
        }
        shiftcount++;
        tenuptime=tenuptime | shifted;
    }
}
day = tenuptime / 8640000;
temptime = tenuptime % 8640000;
hour = temptime / 360000;
temptime = temptime % 360000;
min = temptime / 6000;
temptime = temptime % 6000;
sec = temptime / 100;
uptime = String.valueOf(day)+" days "+String.valueOf(hour)+" hours "+
String.valueOf(min)+" minutes "+String.valueOf(sec)+" seconds"; //sysUpTime

```

```

int poscontact;
int numcontact;
poscontact = posuptime+numuptime+14;
numcontact = Tbuffer[posuptime+numuptime+14-1];
    for (i=poscontact;i<(poscontact+numcontact);i++){
        contact.append((char)Tbuffer[i]);}
//sysContact

int possysname;
int numsysname;
possysname = poscontact+numcontact+14;
numsysname = Tbuffer[poscontact+numcontact+14-1];
    for (i=possysname;i<(possysname+numsysname);i++){
        sysname.append((char)Tbuffer[i]);}
//sysName

int poslocation;
int numlocation;

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (i=poslocation;i<(poslocation+numlocation);i++){
            location.append((char)Tbuffer[i]);} //sysLocation
int posservice;
int numservice;
posservice = poslocation+numlocation+14;
numservice = Tbuffer[poslocation+numlocation+14-1];
    for (i=posservice;i<(posservice+numservice);i++){
        service = Tbuffer[i];} //sysService
// Get Source Port
NumberPort = packet.getPort();

// Get Source Local Port
Lport = socket.getLocalPort();

// Get Source IP
IPAdd = packet.getAddress();
String IP = IPAdd.getHostAddress();

// Show Packet
System.out.println("");
System.out.println("-- Received "+packet.getLength()+" byte --");
System.out.println("SysDescription: "+desc);
System.out.println("SysObjectID: "+objid);
System.out.println("SysUpTime: "+uptime);
System.out.println("SysContact: "+contact);
System.out.println("SysName: "+sysname);
System.out.println("SysLocation: "+location);
System.out.println("SysService: "+service);
//System.out.println("Received: "+result);
System.out.println("Port: "+ String.valueOf(NumberPort) + " IP Address: " + IP);
System.out.println("Local Port: "+Lport);

String url = "jdbc:odbc:snmp";
Connection con;
Statement stmt;
String query = "select IP, SysDesc from mib";
try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(java.lang.ClassNotFoundException e)
{
    System.err.print("ClassNotFoundException: ");
    System.err.println(e.getMessage());
}
try
{
    con = DriverManager.getConnection(url,"sa", "");
    stmt = con.createStatement();
    stmt.executeUpdate("update mib set SysDesc=" + desc+
        ", SysObjectID=" + objid+
        ", SysUpTime=" + uptime+
        ", SysContact=" + contact+
        ", SysName=" + sysname+
        ", SysLocation=" + location+
        ", SysService=" + service+
        " where name ="+sh_host+""");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและดัดแปลงข้อมูลของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ResultSet rs = stmt.executeQuery(query);
        System.out.println("IP and Description:");
        while (rs.next())
        {
            String ip = rs.getString("IP");
            String de = rs.getString("SysDesc");
            System.out.println(ip + " " + de);
        }
        stmt.close();
        con.close();
    }

    catch(SQLException ex)
    {
        System.err.println("SQLException: " + ex.getMessage());
    }
}

public static void main(String args[]) throws IOException {
    Sys5 echo = new Sys5();
    echo.execute();
}

protected DatagramPacket buildPacket(byte message[], InetAddress host, int port, int Length)
throws IOException {
    System.out.println("");
    byte data[] = new byte[1024];
    for (int i=0 ;i<118;i++)
    {
        data[i] = message[i];
        System.out.print(" ");
    }
    System.out.println("");
    System.out.println("");
    return new DatagramPacket(data, Length, host, port);
}

class Alarm extends Thread {
    public Alarm (){}
    public void run() {
        try {
            Thread.sleep(5000);
            System.out.println("resend packet");
            Sys5 echo = new Sys5();
            echo.execute();
        }
        catch(InterruptedException ex) {
            ex.printStackTrace();
        }

        catch(IOException eio){
            eio.printStackTrace();
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้