

การเฝ้าดูระบบเครือข่าย เพื่อการจัดการด้านเครือข่าย แอน / แวน
NETWORK MONITORING FOR LAN/WAN MANAGEMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-144-4

การเฝ้าดูระบบเครือข่าย เพื่อการจัดการสำหรับเครือข่าย แลน/แวน

NETWORK MONITORING FOR LAN/WAN MANAGEMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่.....
เลขที่.....
ปี 18 ค.ศ. 2544

พ.ศ. 2544

ISBN 974-648-144-4

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NETWORK MONITORING FOR LAN/WAN MANAGEMENT



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2001

ISBN 974-648-144-4



COPYRIGHT 2001

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การเฝ้าดูระบบเครือข่ายเพื่อการจัดการสำหรับเครือข่าย แลน/แวน
NETWORK MONITORING FOR LAN/WAN MANAGEMENT
ชื่อนักศึกษา นายปิยพงศ์ ผดุงพจน์
รหัสประจำตัว 37061152
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมไฟฟ้า
อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.สมพล โกศลวิตร

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
ศ.มบุญ	สุขเกษม
ผศ.ดร.ไกรสิน	สงวัฒนา
รศ.สมยศ	จุนณะปิยะ
รศ.ดร.สุวิพล	สิทธิชีวกาศ
ผศ.สมพล	โกศลวิตร

วัน/เดือน/ปี ที่สอบ 9 เมษายน 2544 เวลา 12.00 - 13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)



วันที่.....1.....เดือน.....พฤษภาคม.....พ.ศ. 2544.....

หัวข้อวิทยานิพนธ์	การเฝ้าดูระบบเครือข่าย เพื่อการจัดการสำหรับเครือข่าย แลน/แวน
นักศึกษา	นายปิยพงศ์ ผดุงพจน์
รหัสประจำตัว	37061152
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. สมผล โกศลวิตร

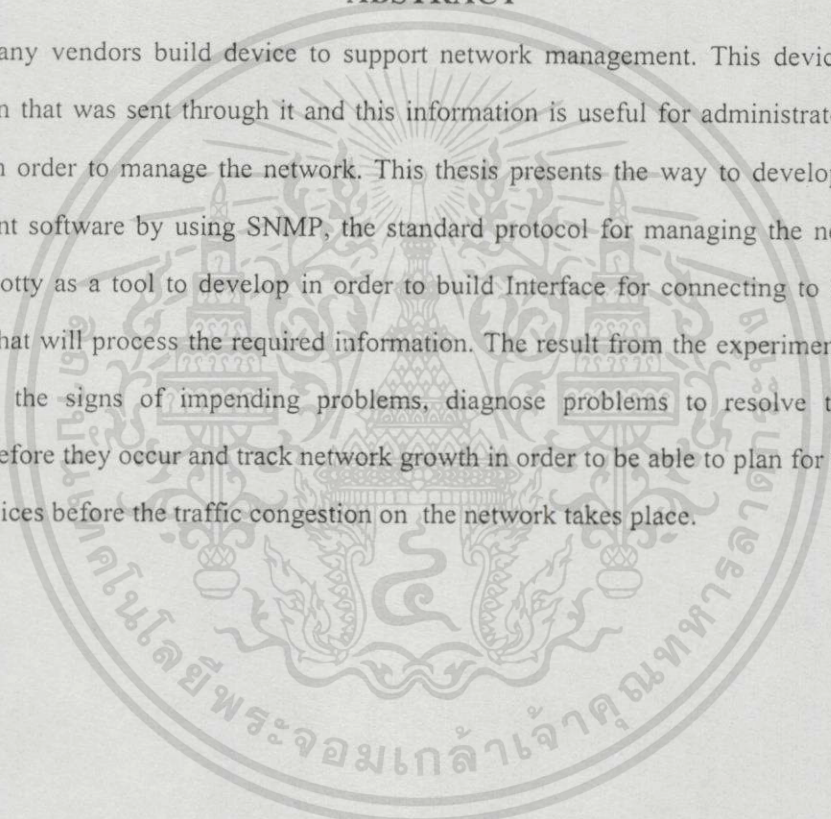
บทคัดย่อ

ผู้ผลิตอุปกรณ์เครือข่ายสมัยใหม่ ส่วนใหญ่จะทำการสร้างอุปกรณ์ให้สนับสนุนการจัดการ (Network Management) โดยอุปกรณ์เหล่านี้มีความสามารถในการรวบรวมข้อมูลที่ส่งผ่านอุปกรณ์ ข้อมูลต่าง ๆ นี้มีประโยชน์สำหรับผู้ดูแลระบบหรือวิศวกรระบบเพื่อนำไปใช้ในการจัดการเครือข่าย วิทยานิพนธ์นี้ได้นำเสนอแนวทางในการพัฒนาซอฟต์แวร์จัดการเครือข่าย (Network Management Software) โดยใช้หลักการของโปรโตคอลเอสเอ็นเอ็มพี ซึ่งเป็นโปรโตคอลมาตรฐานที่ถูกใช้อย่างแพร่หลาย มาใช้จัดการกับอุปกรณ์เครือข่าย และใช้ Scotty เป็นเครื่องมือช่วยในการพัฒนา เพื่อทำการสร้างอินเทอร์เฟซติดต่อกับ SNMP Scotty Extension เพื่อทำการประมวลผลข้อมูลที่ต้องการ ซึ่งผลที่ได้จากการทดสอบสามารถนำมาวิเคราะห์ปัญหาที่กำลังจะเกิดขึ้น ตรวจสอบปัญหาเพื่อทำการแก้ไข ป้องกันปัญหา ก่อนที่ปัญหาจะเกิด และติดตามความเจริญเติบโตของเครือข่ายเพื่อการวางแผนขยายเครือข่าย หรือบริการใหม่ ก่อนที่การจราจรจะคับคั่ง

Thesis Title	Network Monitoring for LAN/WAN Management
Student	Mr. Piyapong Phadungpoj
Student ID.	37061152
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2001
Thesis Advisor	Asst.Prof. Sompol Kosulvit

ABSTRACT

Many vendors build device to support network management. This device collects the information that was sent through it and this information is useful for administrator and system engineer in order to manage the network. This thesis presents the way to develop the network management software by using SNMP, the standard protocol for managing the network device and use Scotty as a tool to develop in order to build Interface for connecting to SNMP Scotty extension that will process the required information. The result from the experiment can be used to analyze the signs of impending problems, diagnose problems to resolve them, prevent problems before they occur and track network growth in order to be able to plan for the expansion or new services before the traffic congestion on the network takes place.



กิตติกรรมประกาศ

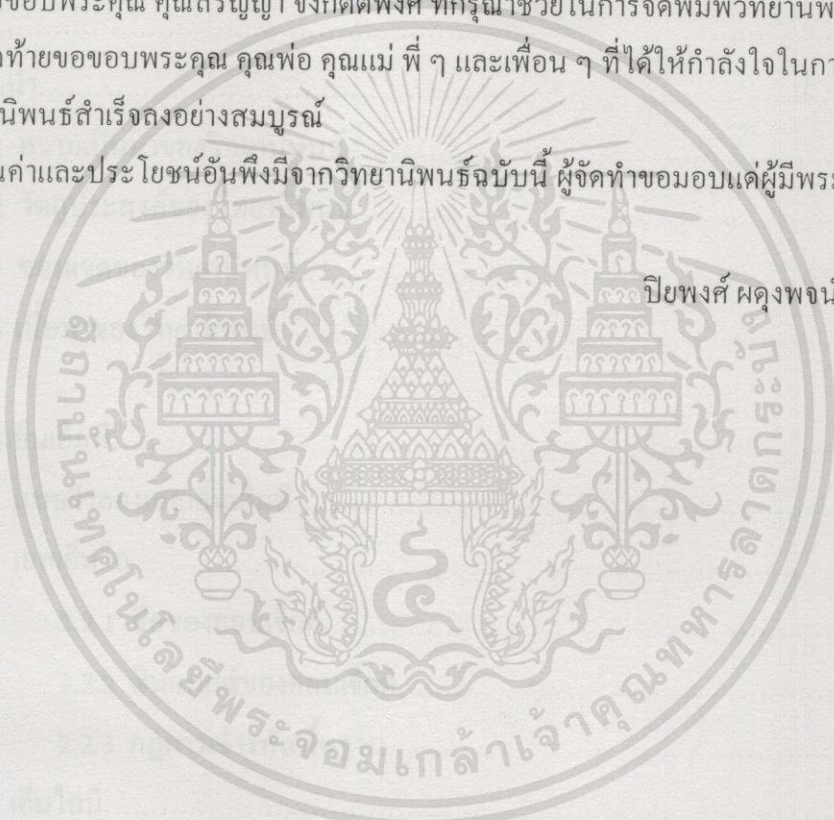
วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาที่ดีจาก ผศ. สมผล โภคสวัสดิ์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์และขอกราบขอบพระคุณเป็นอย่างสูงที่ได้ให้การสนับสนุนและคำปรึกษาในงานวิจัยจนสำเร็จตามวัตถุประสงค์

ขอขอบพระคุณ คุณจินดา สันติกรภาพ คุณวีระพล ภาสะประหาส ที่ให้คำปรึกษาด้านการทดลอง และผู้อำนวยการสำนักกำกับและอนุรักษ์พลังงาน ที่ให้ความอนุเคราะห์สถานที่ในการทดสอบระบบเครือข่าย

ขอขอบพระคุณ คุณสรัญญา จงกิตติพงศ์ ที่กรุณาช่วยในการจัดพิมพ์วิทยานิพนธ์
สุดท้ายขอขอบพระคุณ คุณพ่อ คุณแม่ พี่ ๆ และเพื่อน ๆ ที่ได้ให้กำลังใจในการทำวิจัยเสมอมาจนวิทยานิพนธ์สำเร็จลงอย่างสมบูรณ์

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

ปิยพงศ์ ผดุงพจน์



สารบัญ(ต่อ)

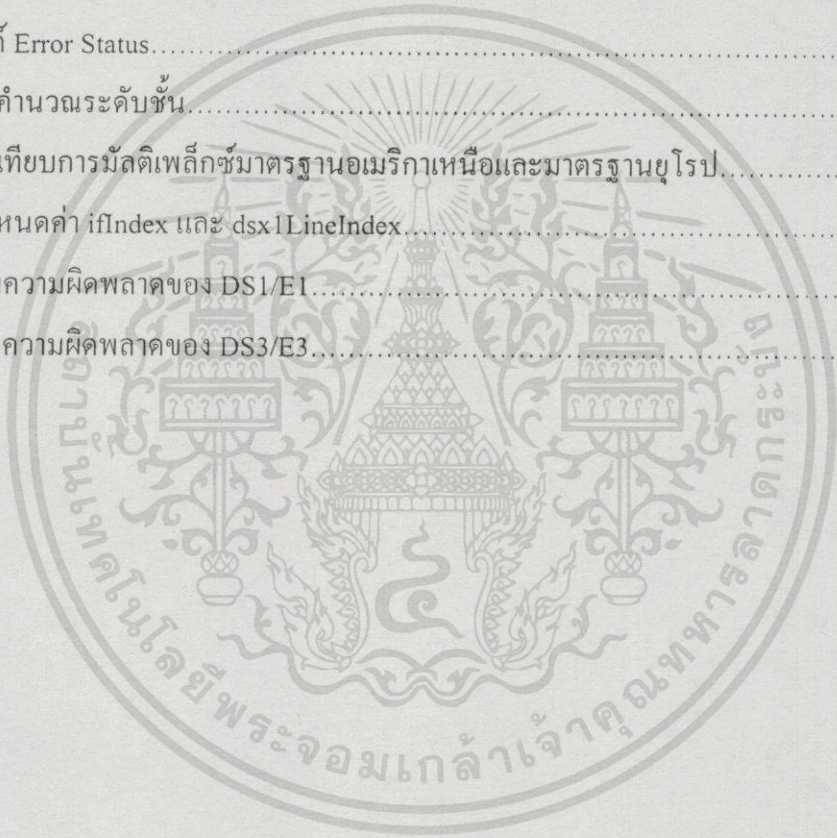
	หน้า
3.1 กลุ่มระบบ.....	22
3.2 กลุ่มอินเทอร์เฟซ.....	23
3.3 กลุ่มเอที.....	27
3.4 กลุ่มไอพี.....	27
3.5 กลุ่มไอซีเอ็มพี.....	33
บทที่ 4 การจัดการข่ายคอมพิวเตอร์เฉพาะบริเวณ (แลน).....	35
4.1 อีเทอร์เน็ต-โลคัลแลน.....	35
4.2 IEEE 802.3 รีพีตเตอร์.....	37
4.3 IEEE 802.3 Medium Attachment Units.....	42
4.5 บริดจ์.....	44
บทที่ 5 การจัดการข่ายคอมพิวเตอร์บริเวณกว้าง (แวน).....	53
5.1 หลักการทั่วไป.....	53
5.2 กลุ่มดีเอสหนึ่ง.....	58
5.3 กลุ่มดีเอสสาม.....	65
บทที่ 6 การพัฒนาโปรแกรม.....	70
6.1 สถานีจัดการเครือข่าย.....	70
6.2 การร้องขอของเอ็นเอ็มเอส.....	71
6.3 การตอบสนองของเอเจนต์.....	72
6.4 การพัฒนาโปรแกรม.....	73
6.4.1 คำสั่ง SNMP.....	73
6.4.2 คำสั่ง MIB.....	82
6.5 โครงสร้างโปรแกรมหลัก.....	85
บทที่ 7 การทดลองและผลการทดลอง.....	87

สารบัญ(ต่อ)

	หน้า
7.1 การเรียกใช้โปรแกรม.....	87
7.2 การทดลองและการวิเคราะห์ข้อมูล.....	88
7.2.1 การทดสอบแสดงผลข้อมูลระบบ.....	88
7.2.2 การทดสอบแสดงผลข้อมูลอินเทอร์เน็ตเฟช.....	89
7.2.3 การทดสอบแสดงผลข้อมูลไอพี.....	93
7.2.4 การทดสอบแสดงผลข้อมูลไอซีเอ็มพี.....	95
7.2.5 การทดสอบแสดงผลข้อมูลของอุปกรณ์แลนและแวน.....	97
บทที่ 8 สรุปผลการวิจัยและข้อเสนอแนะ.....	99
8.1 สรุปผลการวิจัย.....	99
8.2 ข้อเสนอแนะและแนวทางการพัฒนา.....	99
เอกสารอ้างอิง.....	101
ภาคผนวก โปรแกรมต้นฉบับ.....	103
ประวัติผู้เขียน.....	145

สารบัญตาราง

ตารางที่	หน้า
2.1 ชนิดของคลาส.....	9
2.2 ชนิดของรูปแบบ.....	9
2.3 ชนิดของยูนิเวอร์แซลคลาสแท้ก.....	10
2.4 ชนิดของแอปพลิเคชันไวด์คลาสแท้ก.....	10
2.5 ชนิดของคอนเท็กซ์สเปซิฟิกคลาสแท้ก.....	10
2.6 ค่าฟิลด์ Error Status.....	16
3.1 ค่าการคำนวณระดับชั้น.....	22
5.1 เปรียบเทียบการมัลติเพิล็กซ์มาตรฐานอเมริกาเหนือและมาตรฐานยุโรป.....	54
5.2 การกำหนดค่า ifIndex และ dsx1LineIndex.....	56
5.3 การนับความผิดพลาดของ DS1/E1.....	63
5.4 การนับความผิดพลาดของ DS3/E3.....	68



สารบัญรูป

รูปที่	หน้า
2.1 แบบจำลองเอสเอ็นเอ็มพี.....	4
2.2 OBJECT IDENTIFIER (OID) ตามโครงสร้างรูปต้นไม้.....	6
2.3 รูปแบบ BER.....	8
2.4 การเข้ารหัสข้อมูลแบบ TLV.....	8
2.5 ส่วนประกอบไทม์ฟิลด์ (แท็ก=0-30).....	9
2.6 การเข้ารหัสไทม์ฟิลด์ (แท็ก > 30).....	11
2.7 การเข้ารหัสฟิลด์ความยาว.....	11
2.8 เอสเอ็นเอ็มพีเมสเซจ.....	15
2.9 รูปแบบพีดียูที่ใช้ร่วมกัน.....	16
2.10 การส่งผ่าน GetRequest/GetResponse PDU.....	17
2.11 การส่งผ่าน GetNextRequest/GetResponse PDU.....	18
2.12 การส่งผ่าน SetRequest/GetResponse PDU.....	19
2.13 โครงสร้างแตรปพีดียู.....	20
5.1 แสดงหน้าที่ของ CSUs/DSUs.....	55
6.1 แสดงแบบจำลองของสถานีจัดการเครือข่าย.....	70
6.2 แสดงขั้นตอนการร้องขอข้อมูล.....	71
6.3 แสดงขั้นตอนการตอบสนองข้อมูล.....	72
6.4 ขั้นตอนการเตรียมพร้อมก่อนใช้งาน.....	85
6.5 ขั้นตอนการทำงานของโปรแกรมหลัก.....	86
7.1 แสดงเมนูหลักของโปรแกรมและเครือข่ายจริงที่ใช้ทดสอบ.....	87
7.2 แสดงข้อมูลระบบ.....	88
7.3 รายละเอียดของอินเทอร์เฟซ.....	89
7.4 ข้อมูลทางสถิติของอินเทอร์เฟซ.....	90
7.5 สมรรถนะของอินเทอร์เฟซ.....	90
7.6 ค่าเปอร์เซ็นต์การใช้ประโยชน์, ค่าเปอร์เซ็นต์ความผิดพลาด และค่าเปอร์เซ็นต์แพ็กเก็ต ยกเลิกของอินเทอร์เฟซ.....	91
7.7 รายงานการแจ้งเตือน.....	92

สารบัญรูป(ต่อ)

รูปที่	หน้า
7.8 แสดงข้อมูลไอพี.....	93
7.9 แสดงข้อมูลทางสถิติของไอพี.....	94
7.10 แสดงข้อมูลไอซีเอ็มพี.....	95
7.11 แสดงข้อมูลสถิติการรับส่งเฟรมของอีเทอร์เน็ต.....	97
7.12 แสดงข้อมูลของอุปกรณ์ประเภทเลนและแวนอินเทอร์เฟซ.....	98



บทที่ 1

บทนำ

1.1 ความเป็นมาของวิทยานิพนธ์

อุปกรณ์เครือข่ายฮาร์ดแวร์ เช่น เราเตอร์, การ์ดเครือข่าย และเซิร์ฟเวอร์ มักถูกติดตั้งอยู่ในสถานที่ต่าง ๆ ห่างไกลจากผู้ดูแลระบบ เมื่อเครือข่ายเกิดปัญหาขึ้นจึงเป็นการยากลำบากและใช้เวลานานในการค้นหาปัญหา ดังนั้นผู้ดูแลระบบหรือวิศวกรระบบควรมีเครื่องมือไว้คอยช่วยเหลือในการวิเคราะห์ปัญหาของเครือข่าย แต่ก่อนการตรวจสอบแพ็กเก็ตจะกระทำการผ่านแพ็กเก็ตไดรเวอร์ [1] ซึ่งเป็นโปรแกรมเล็ก ๆ ที่ฝังตัวลงในหน่วยความจำ (Resident) ทำงานในชั้นดาต้าลิงก์ ทำหน้าที่เป็นตัวกลางในการจัดการติดต่อผ่านไปยังการ์ดเครือข่าย วิธีการนี้มีข้อจำกัดในด้านความเข้ากันได้ของแพ็กเก็ตไดรเวอร์ที่ใช้กับระบบเครือข่ายต่างชนิดกัน ซึ่งอาจต้องมีการแก้ไขโปรแกรมใหม่ รวมถึงขีดความสามารถด้านการขยายการใช้งาน ซึ่งส่วนใหญ่แล้วแพ็กเก็ตไดรเวอร์จะทำงานบนระบบปฏิบัติการดอส ซึ่งไม่ค่อยนิยมใช้กันแล้ว ส่วนวิธีการที่นิยมใช้กันในปัจจุบันจะอาศัยหลักการของเอสเอ็นเอ็มพี (SNMP: Simple Network Management Protocol) [2] ซึ่งเป็นโปรโตคอลที่ใช้กันอย่างแพร่หลายสำหรับจัดการเครือข่าย ทำงานในระดับชั้นแอปพลิเคชัน ขีดความสามารถด้านการขยายการใช้งานเป็นไปอย่างกว้างขวาง เพราะปัจจุบันการพัฒนาสร้างเอเจนต์จัดการมีราคาถูก มีผลทำให้ผู้ผลิตอุปกรณ์ส่วนใหญ่สร้างเอเจนต์แถมมาให้ เพื่อสนับสนุนการทำงานของเอสเอ็นเอ็มพี ดังนั้นจึงเกิดแนวคิดของวิทยานิพนธ์ฉบับนี้ ซึ่งได้ทำการค้นคว้าเพื่อเสนอแนวทางในการพัฒนาสร้างซอฟต์แวร์จัดการเครือข่าย โดยใช้หลักการของโปรโตคอลเอสเอ็นเอ็มพี เพื่อช่วยเหลือผู้ดูแลระบบในการวิเคราะห์ และแก้ไขปัญหาที่เกิดขึ้นบนระบบเครือข่าย

1.2 วัตถุประสงค์ของการทำวิทยานิพนธ์

วัตถุประสงค์การทำวิทยานิพนธ์ มีดังต่อไปนี้

- เพื่อศึกษารูปแบบการทำงานของชุดโปรโตคอลเอสเอ็นเอ็มพี ในส่วนประกอบหลักคือเอสเอ็มไอ, เอ็มไอบี และโปรโตคอลเอสเอ็นเอ็มพี
- เพื่อศึกษารายละเอียดของอบเจกต์ในกลุ่มเอ็มไอบี II และเอ็มไอบีในกลุ่มอื่นที่เกี่ยวข้องในการนำไปใช้ประโยชน์เพื่อจัดการกับอุปกรณ์เครือข่ายที่สนใจ
- เพื่อสามารถนำซอฟต์แวร์ที่พัฒนาขึ้นมาไปใช้ประโยชน์ในการวิเคราะห์ หรือแก้ปัญหาของเครือข่ายได้
- เพื่อเป็นแนวทางในการพัฒนาซอฟต์แวร์จัดการเครือข่าย ที่มีประสิทธิภาพต่อไป

1.3 ขอบเขตของวิทยานิพนธ์

ขอบเขตการศึกษาของวิทยานิพนธ์มีดังต่อไปนี้

- ทำการศึกษาเฉพาะส่วนของซอฟต์แวร์ที่สถานีจัดการ (Network Management Station) ไม่ครอบคลุมส่วนของซอฟต์แวร์เอเจนต์ ซึ่งจะไม่มีการอัปเดต, อัปเดต หรือแก้ไขเอเจนต์
- ซอฟต์แวร์ที่พัฒนาใช้จัดการกับอุปกรณ์เครือข่ายที่เป็นฮาร์ดแวร์ หรือสามารถจัดการที่ชั้นอินเทอร์เน็ตโปรโตคอลลงมา เช่น เราเตอร์, เซิร์ฟเวอร์ หรือการ์ดเครือข่าย แต่ไม่ครอบคลุมการจัดการที่เกี่ยวข้องกับซอฟต์แวร์ เช่น การจัดการแอปพลิเคชัน รวมถึงซอฟต์แวร์แพคเกจที่เกี่ยวข้องกับการจัดการ
- ออบเจกต์ที่ทำการศึกษายกจะเกี่ยวข้องกับการจัดการในชั้นอินเทอร์เน็ตลงมา ไม่ครอบคลุมชั้นโฮสต์ทูโฮสต์ และชั้นแอปพลิเคชัน ของชุดโปรโตคอลทีซีพี/ไอพี
- การวิเคราะห์ข้อมูลตามหน้าที่จัดการ (Functional Areas) จะเน้นการวิเคราะห์ในส่วนการจัดการความบกพร่อง (Fault Management) การจัดการคอนฟิกูเรชัน (Configuration Management) และการจัดการสมรรถนะ (Performance Management) ไม่ครอบคลุมการจัดการบัญชี (Accounting Management) และการจัดการความปลอดภัย (Security Management)

1.4 เนื้อหาของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 8 บท ดังต่อไปนี้

บทที่ 1 กล่าวถึงความจำเป็นและความสำคัญของปัญหา, วัตถุประสงค์การศึกษา, ขอบเขตการศึกษา และเนื้อหาของในแต่ละบทของวิทยานิพนธ์

บทที่ 2 กล่าวถึงรายละเอียด และโครงสร้างการทำงานของ เอสเอ็มไอ, เอ็มไอบี และชุดโปรโตคอลเอสเอ็นเอ็มพี

บทที่ 3 กล่าวถึงรายละเอียดของออบเจกต์ที่เกี่ยวข้องกับการจัดการอินเทอร์เน็ตเฟสที่ระดับชั้นอินเทอร์เน็ตลงมา ซึ่งเป็นออบเจกต์ที่อยู่ในกลุ่ม MIB-II เป็นหลัก

บทที่ 4 กล่าวถึงรายละเอียดของออบเจกต์ที่เกี่ยวข้องกับการจัดการเล่นอินเทอร์เน็ตเฟส ตามมาตรฐาน IEEE 802.3 / อีเทอร์เน็ต และการจัดการอุปกรณ์เล่น ประเภทบริดจ์, รีพีตเตอร์ และอุปกรณ์ประเภท MAU (Medium Attachment Unit)

บทที่ 5 กล่าวถึงรายละเอียดของออบเจกต์ที่เกี่ยวข้องกับการจัดการแวนอินเทอร์เน็ตเฟส 2 ชนิด คือ DS1 และ DS3 ซึ่ง DS1 และ DS3 เป็นมาตรฐานที่ใช้กันอยู่ในอเมริกาเหนือ สมมูลกับ E1 และ E3 ที่เป็นมาตรฐานใช้กันอยู่ในยุโรป ดังนั้นการจัดการออบเจกต์ DS1 สามารถประยุกต์ใช้ได้ทั้งใน DS1 และ E1 เช่นเดียวกับ DS3 สามารถประยุกต์ใช้ได้ทั้งใน DS3 และ E3

บทที่ 6 กล่าวถึงหลักการดำเนินงาน และรูปแบบของคำสั่งที่ใช้ในการพัฒนาโปรแกรม รวมถึง
 โฟลว์ชาร์ตการทำงานหลักของโปรแกรม

บทที่ 7 กล่าวถึงการเข้าใช้งานโปรแกรม การทดสอบโปรแกรมที่พัฒนาขึ้นมา รวมถึงการ
 วิเคราะห์ผลข้อมูลที่ได้จากการทดสอบ

บทที่ 8 กล่าวถึงสรุปผลการวิจัย ปัญหาที่พบ การนำไปใช้ประโยชน์ รวมถึงข้อเสนอแนะ
 และแนวทางในการพัฒนาต่อ



บทที่ 2

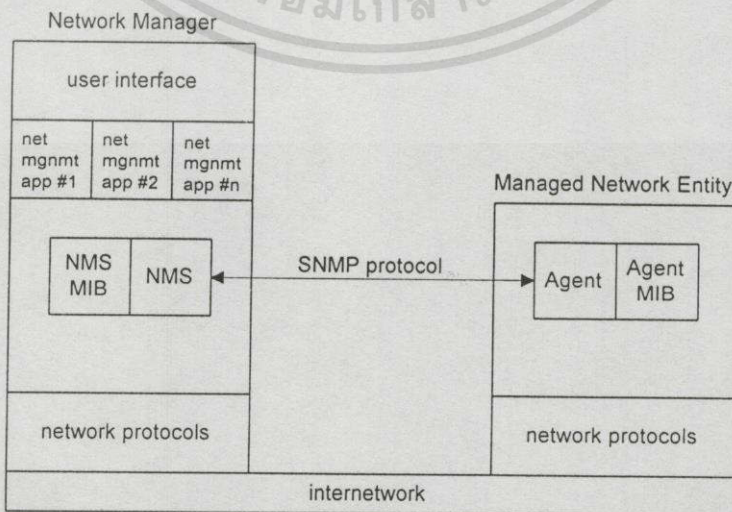
เอสเอ็นเอ็มพี

เอสเอ็นเอ็มพี ถูกพัฒนาโดย IETF (Internet Engineering Task Force) เป็นโปรโตคอลมาตรฐานสำหรับจัดการเครือข่ายที่นิยมใช้อย่างแพร่หลาย ถูกออกแบบให้สามารถทำงานได้อย่างอิสระ สามารถทำงานได้ดีกับเครือข่ายขนาดใหญ่ที่ใช้โปรโตคอลทีซีพี/ไอพี และยังสามารถใช้งานผ่านไอพีเอ็กซ์, แอปพลิเคชัน, โอเอสไอ และโปรโตคอลระดับชั้นขนส่งอื่นได้อีก ชุดโปรโตคอลในตระกูลของเอสเอ็นเอ็มพี มีส่วนประกอบหลักที่สัมพันธ์กันอยู่ 3 ส่วนคือ เอสเอ็มไอ, เอ็มไอบี และโปรโตคอลเอสเอ็นเอ็มพี

อุปกรณ์เครือข่ายทั้งหมดที่จัดการด้วยเอสเอ็นเอ็มพี จะประกอบด้วยเอเจนต์ และสถานีจัดการ (Management Station) เอเจนต์จะรายงานสถานะของอุปกรณ์ไปยังสถานีจัดการ โดยทำการรวบรวมข้อมูลทางสถิติและเก็บไว้ในเอ็มไอบี เอเจนต์อาจจะเป็นซอฟต์แวร์หรือส่วนของโปรแกรมถาวร (Firmware) ที่ติดตั้งอยู่ในอุปกรณ์ ซึ่งเรียกว่าอุปกรณ์ที่จัดการ เช่น เราเตอร์, และเซิร์ฟเวอร์ ส่วนสถานีจัดการจะทำการสอบถามไปยังเอเจนต์แต่ละตัวและเริ่มต้นการส่งข้อมูลในเอ็มไอบีไปยังสถานีจัดการ ตามปกติสถานีจัดการจะทำงานบนวินโดวส์ เพราะมีอินเทอร์เฟซเป็นแบบกราฟิก ซึ่งทำให้ง่ายต่อผู้ใช้

2.1 แบบจำลองเอสเอ็นเอ็มพี

แบบจำลองเอสเอ็นเอ็มพี [2] ดังรูปที่ 2.1 ประกอบด้วย 4 ส่วนหลักคือ อินเทอร์เน็ตเวิร์ก (Internetwork), โปรโตคอลเครือข่าย (Network Protocol), ส่วนจัดการเครือข่าย (Network Manager) และเอนทิตีเครือข่าย (Managed Network Entity)



รูปที่ 2.1 แบบจำลองเอสเอ็นเอ็มพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ตเวิร์ก

อินเทอร์เน็ตเวิร์ก คือกลุ่มเครือข่ายหนึ่งหรือมากกว่าที่ใช้โปรโตคอลร่วมกัน และเชื่อมต่อกันด้วยเคเบิล เครื่องปลายทางทั้งสองสามารถสื่อสารกันได้ถ้ามีปัจจัยที่เหมาะสม เช่น แอคเซสโปรโตคอลที่มาตรฐาน และเส้นทางที่เหมาะสม

โปรโตคอลเครือข่าย

โปรโตคอลเครือข่าย คือกฎระเบียบที่ก่อให้เกิดการสื่อสารระหว่างเครือข่าย ซึ่งส่วนใหญ่แล้วใช้โปรโตคอลทีซีพี/ไอพี เป็นหลัก ถึงแม้ว่าเอสเอ็นเอ็มพีสามารถทำงานผ่านโปรโตคอลขนส่งอื่นได้ แต่ถ้าเอ็นเอ็มเอสมีการดำเนินงานผ่านชุดโปรโตคอลทีซีพี/ไอพี เอสเอ็นเอ็มพีจำเป็นที่ต้องใช้ยูติพีในการรับส่งเมสเสจ หรือรับแพคเกจ ซึ่งจำเป็นต้องใช้ไอพีโปรโตคอลในการเข้าถึงเครือข่าย

ส่วนจัดการเครือข่าย

ส่วนจัดการเครือข่าย คือส่วนที่จัดการโปรโตคอลเครือข่าย และโปรโตคอลเอสเอ็นเอ็มพีที่อุปกรณ์เครือข่ายใช้สื่อสารผ่านอินเทอร์เน็ตเวิร์กไปยังเอนทิตีเครือข่าย มีส่วนประกอบ 4 ส่วนคือ ส่วนที่หนึ่งสถานีจัดการเครือข่าย (Network Management Station) หรือเอ็นเอ็มเอส คือกระบวนการเอนทิตีที่ทำการเฝ้าดู (Monitor) และควบคุม (Control) เอเจนต์ นอกจากนี้ยังสามารถอ่านและเขียนออบเจกต์เอ็มไอบี สามารถเก็บอินฟอร์เมชันของแต่ละเอเจนต์เข้าไว้ในฐานข้อมูลของตัวเองได้ ส่วนที่สอง เอ็นเอ็มเอสเอ็มไอบี คือฐานข้อมูลสารสนเทศของเอ็นเอ็มเอส ซึ่งบรรจุรายการหลักของเอ็มไอบีจากเอเจนต์ทั้งหมด เอ็นเอ็มเอสสามารถควบคุมค่าในเอ็มไอบีของแต่ละเอเจนต์ได้ ถ้าค่านั้นถูกแสดงอยู่ในฐานข้อมูล การบำรุงรักษาเอ็นเอ็มเอสบ่อยๆ จะเป็นการปรับปรุงเพื่อเพิ่มรายการของสารสนเทศเข้าไว้ในฐานข้อมูล ส่วนที่สาม แอปพลิเคชันจัดการเครือข่าย (Network Management Applications) คือส่วนของโปรแกรมที่นำสารสนเทศของเอสเอ็นเอ็มพีไปยังผู้ใช้ โปรแกรมส่วนนี้จะมีความสามารถในการโพลล์ (Poll) ไปยังเอเจนต์, มีความสามารถในการ get request และ set request รวมถึงการจัดการแพคเกจ (Trap Message) ส่วนที่สี่ ยูสเซอร์อินเทอร์เฟซ (User Interface) คือส่วนโปรแกรมของผู้ใช้ปลายทาง เพื่อแสดงผลข้อมูลค่าออบเจกต์ของอุปกรณ์เครือข่ายที่สนใจ โดยส่วนมากจะแสดงผลอยู่ในรูปแบบของ GUI (Graphical User Interface) ที่นิยมใช้กันมากในปัจจุบันก็คือวินโดวส์

เอนทิตีเครือข่าย (Managed Network Entity)

เอนทิตีเครือข่ายที่ถูกจัดการ ประกอบด้วย 2 ส่วนคือ เอเจนต์ และเอเจนต์เอ็มไอบี เอเจนต์คือกระบวนการเอนทิตีที่รับการร้องขอจากเอ็นเอ็มเอส และส่งค่าที่เหมาะสมเพื่อรายงานกลับไปยังเอ็นเอ็มเอส นอกจากนี้เอเจนต์ยังสามารถเซตให้มีการส่งแพคเกจเมสเสจเพื่อรายงานเหตุการณ์ไปยังเอ็นเอ็มเอสได้ ส่วนเอเจนต์เอ็มไอบี คือฐานข้อมูลสารสนเทศของเอเจนต์ ซึ่งเก็บรวบรวมค่าตัวแปรที่น่าสนใจไว้

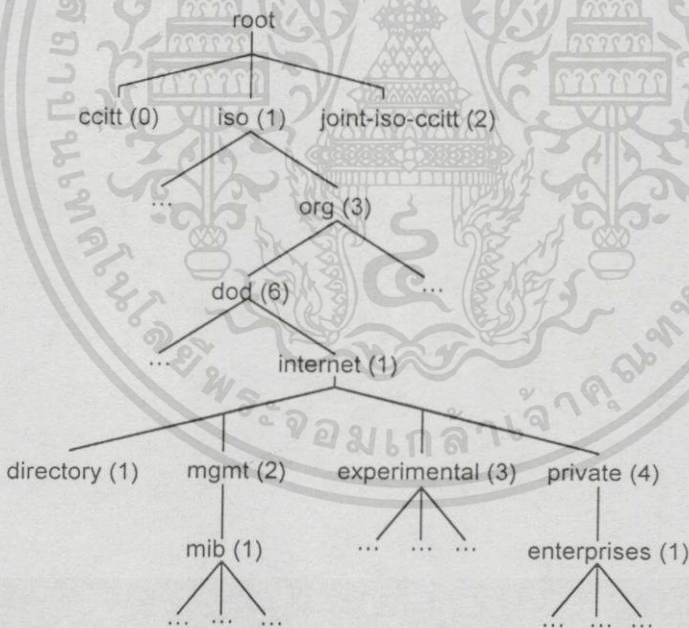
2.2 เอสเอ็มไอ (SMI- Structure of Management Information)

โครงสร้างสารสนเทศการจัดการ (Structure of Management Information) หรือ เอสเอ็มไอ [2,3] ใช้สำหรับกำหนดค่าของออบเจกต์ ซึ่งการกำหนดค่าออบเจกต์จะมีลักษณะเฉพาะ (Attribute) 3 ประการคือ ชื่อ (Name), ซินแทกซ์ (Syntax) และการเข้ารหัส (Encoding)

ออบเจกต์ทุกออบเจกต์จะมีชื่อที่เป็นเอกลักษณ์ไม่ซ้ำกัน เรียกว่า ตัวกำหนดชื่อออบเจกต์ (Object Identifier), ซินแทกซ์ ใช้สำหรับกำหนดโครงสร้างชนิดของข้อมูล เช่น เลขจำนวนเต็ม การกำหนดนี้ใช้ ASN.1 (Abstract Syntax Notation one) เป็นตัวกำหนด ส่วนการเข้ารหัส เป็นการแปลงข้อกำหนดของ ASN.1 เป็นแพทเทิร์นของบิตเพื่อใช้ในการส่งผ่านข้อมูลระหว่างเครื่อง โดยใช้กฎการเข้ารหัสพื้นฐาน BER (Basic Encoding Rules)

2.2.1 ชื่อของออบเจกต์ (Object Name)

ออบเจกต์ทุกออบเจกต์จะมีชื่อเป็นเอกลักษณ์ของตัวเองที่ไม่เหมือนใคร ซึ่งจะเรียกว่าตัวกำหนดชื่อออบเจกต์ (OID- Object Identifier) ชื่อแต่ละชื่อจะมีเลขจำนวนเต็มกำกับอยู่ในแต่ละชั้นเรียงลำดับกันไปอย่างไม่ซ้ำกันตามลักษณะโครงสร้างรูปต้นไม้ (Tree) ตัวกำหนดชื่อของออบเจกต์สามารถเขียนแทนได้ด้วยเลขจำนวนเต็ม โดยแยกแต่ละส่วนด้วยจุดเรียงเป็นลำดับกันไป



รูปที่ 2.2 Object Identifier (OID) ตาม โครงสร้างรูปต้นไม้

จากรูปที่ 2.2 จะเห็นว่าจุดเริ่มต้นของโครงสร้างรูปต้นไม้คือ รุก (Root) ซึ่งประกอบด้วย 3 สาขา (Branch) คือ ccitt(0), iso(1) และ joint-iso-ccitt(2) แต่ละสาขาก็จะมีสาขาย่อย (Sub-Branch) ออกไปอีก การแทนค่าตัวกำหนดชื่อของออบเจกต์สามารถใช้การแทนค่าจากตัวอักษรมาเป็นตัวเลขได้ดังตัวอย่าง เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแทนค่าด้วยตัวอักษร

iso.org.dod.internet.mgmt.mib

หรือการแทนค่าด้วยตัวเลข

1.3.6.1.2.1

2.2.2 ซินแทกซ์ของออบเจกต์ (Object Syntax)

โครงสร้างข้อมูลซินแทกซ์ของออบเจกต์ สำหรับ SNMP จะถูกแบ่งออกเป็น 3 ชนิด คือ ซิมเพิล, ซิมเพิลคอนสตรัค และแอปพลิเคชันไวด์

ชนิดข้อมูลแบบซิมเพิล (Simple Data Type)

ชนิดข้อมูลแบบซิมเพิล เป็นชนิดข้อมูลทั่วไป บางครั้งเรียกชนิดข้อมูลแบบนี้ว่า ปริมิทีฟ (Primitive) ประกอบด้วย

- INTEGER คือค่าที่กำหนดเป็นเลขจำนวนเต็ม ถูกใช้ใน 2 กรณีคือ ชนิดเลขจำนวนเต็มที่ถูกระบุอย่างชัดเจน เช่น 1, 2 หรือ 3 สำหรับ 0 จะไม่นำมาใช้ และชนิดเลขจำนวนเต็มบิตสตริงถูกใช้สำหรับบิตสตริงแบบย่อ เช่น (0...127) ซึ่งแสดงค่าเป็นเลขฐานสิบหก

- OCTET STRING คือค่าที่ถูกระบุตามลำดับของแต่ละ Octet มีค่าอยู่ระหว่าง 0 ถึง 255 SNMP นำไปใช้ใน 3 กรณีคือ DisplayString ซึ่งถึงอักขระแอสกีทั้งหมดที่ใช้พิมพ์ได้ octetBitstring ถูกใช้สำหรับบิตสตริงที่มีความยาวมากกว่า 32 บิต และ PhysAddress ใช้สำหรับแทนค่าแอดเดรสของสื่อในระดับชั้นกายภาพ

- OBJECT IDENTIFIER คือชื่อของออบเจกต์ ถูกแทนด้วยเลขฐานสิบแล้วตามด้วยจุด (dot) ซึ่งการระบุชื่อออบเจกต์ถูกกำหนดตามโครงสร้างรูปต้นไม้ของ MIB

- NULL ปัจจุบันไม่ได้ใช้แล้วสำหรับ SNMP

ชนิดข้อมูลแบบซิมเพิลคอนสตรัค (Simple-Constructed Data Type)

ชนิดข้อมูลแบบซิมเพิลคอนสตรัคใช้แทนรายการและตาราง ซึ่งทั้งรายการและตารางเป็นส่วนสำคัญของคอนสตรัคที่ใช้ใน SNMP ประกอบด้วย

- SEQUENCE ถูกใช้สำหรับรายการ ชนิดของข้อมูลนี้เทียบได้กับคำว่า “โครงสร้าง” ในภาษาการเขียนโปรแกรมทั่วไป SEQUENCE ประกอบด้วยสมาชิกที่อาจเป็นศูนย์หรือมากกว่าแต่ละสมาชิกเป็นชนิดของ ASN.1 ที่แตกต่างกัน รายการของ SEQUENCE เป็นการกำหนดแถวในตารางทั้งหมด แต่ละเอนทรีของ SEQUENCE ก็คือคอลัมน์ในแถวนั้น

- SEQUENCE OF type ถูกใช้สำหรับตาราง ชนิดของข้อมูลนี้เทียบได้กับคำว่า “อาเรย์” ในภาษาการเขียนโปรแกรมทั่วไป ตารางจะประกอบด้วยสมาชิกที่อาจเป็นศูนย์หรือมากกว่า แต่ละสมาชิกเป็นชนิดของ ASN.1 ที่เหมือนกัน SEQUENCE OF type เป็นการกำหนดแถวในตารางเหมือนกันกับ SEQUENCE

ชนิดข้อมูลแบบแอปพลิเคชันไวด์ (Application-Wide Data Type)

SMI ได้กำหนดชนิดข้อมูลแบบแอปพลิเคชันไวด์ไว้ 6 ชนิดด้วยกันคือ

- IpAddress ใช้แทนค่าไอพีแอดเดรส ค่านี้นี้ประกอบด้วย 32 บิต มีความยาว 4 ออกเทท
- Network Address ใช้แทนชนิดของแอดเดรสเครือข่าย โดยใช้พรีมิทีฟ CHOICE
- Counter ใช้สำหรับนับค่าโดยเพิ่มค่าจาก 0 ถึง 4,294,967,295 หรือ 2^{32-1} เมื่อนับค่าไปจนถึงค่าสูงสุดก็จะวนกลับมาที่ศูนย์แล้วนับใหม่

ถึงค่าสูงสุดก็จะวนกลับมาที่ศูนย์แล้วนับใหม่

- Gauge ใช้สำหรับนับค่าโดยการกำหนดค่าเป็นช่วงจาก 0 ถึง 4,294,967,295 หรือ 2^{32-1}

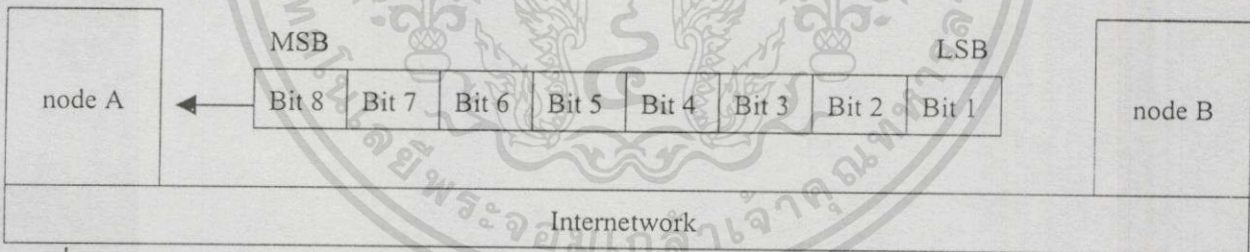
สามารถนับแบบเพิ่มค่าหรือลดค่าก็ได้ เมื่อนับค่าไปจนถึงค่าสูงสุดก็จะหยุด

- TimeTicks ใช้สำหรับบันทึกการนับค่าของเวลาในเศษเสี้ยวของ 100 วินาที มีค่าสูงถึง 4,294,967,295 หรือ 2^{32-1} หน่วย

- Opaque ใช้สำหรับอนุญาตให้ชนิดข้อมูลพิเศษที่ไม่มีกฎเกณฑ์อย่างเป็นทางการของซินแทกซ์ ASN.1 ผ่านได้ ซึ่งปัจจุบันไม่ได้ใช้แล้วสำหรับ SNMP

2.2.3 กฎการเข้ารหัสพื้นฐาน (BER- Basic Encoding Rules)

กฎการเข้ารหัสพื้นฐาน หรือ BER คืออัลกอริทึมที่นำค่าของซินแทกซ์ ASN.1 มาทำการเข้ารหัสบิตให้อยู่ในรูปแบบที่เหมาะสมเพื่อทำการส่งผ่านข้อมูลระหว่างเครือข่าย การเข้ารหัสจะกระทำในระดับบิต บิตที่มีนัยสำคัญมากที่สุด (MSB) คือบิตที่ 8 และจะเป็นบิตที่ถูกส่งข้อมูลก่อน ส่วนบิตที่มีนัยสำคัญน้อยสุด (LSB) คือบิตที่ 1 แสดงดังรูปที่ 2.3



รูปที่ 2.3 รูปแบบ BER

โครงสร้างการเข้ารหัสข้อมูลมีส่วนประกอบ 3 ส่วนคือ ชนิด (Type), ความยาว (Length) และค่า (Value) ซึ่งจะเรียกแผนภาพนี้ว่า TLV แสดงดังรูปที่ 2.4

Tag (or Type Identifier)	Length Identifier	Value (Contents)
--------------------------	-------------------	------------------

รูปที่ 2.4 การเข้ารหัสข้อมูลแบบ TLV

ชนิด (Type Filed)

ส่วนนี้จะประกอบด้วย 3 ส่วนคือ คลาสบิต, เฮฟบิต และแท็กนัมเบอร์บิต ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class Bits	The "F" Bit	Tag Number Bits
------------	-------------	-----------------

รูปที่ 2.5 ส่วนประกอบไทป์ฟิลด์ (แท็ก = 0-30)

- คลาสบิต จะใช้บิต 8 และ 7 เพื่อบ่งบอกชนิดของคลาส ดังตารางที่ 2.1

ตารางที่ 2.1 ชนิดของคลาส

Class Types		
Bits 8-7		Type
0	0	Universal
0	1	Application-wide
1	0	Context-specific
1	1	Private

- เอฟบิต จะใช้บิต 6 เพื่อบ่งบอกรูปแบบว่าชนิดข้อมูลเป็นแบบซิมเพิลหรือคอนสตรัค

ดังตารางที่ 2.2

ตารางที่ 2.2 ชนิดของรูปแบบ

Simple / Constructed Format	
Bit 6	Format Type
0	Simple
1	Constructed

- แท็กนัมเบอร์บิต ใช้บิต 5, 4, 3, 2 และ 1 สำหรับบ่งบอกแท็กนัมเบอร์ ซึ่งจะขึ้นอยู่กับชนิดของคลาสในรูปแบบต่าง ๆ

ตารางที่ 2.3 แสดงค่าแท็กในชนิดข้อมูลแบบยูนิเวอร์แซลคลาส คอถัมน์ S/C จะชี้ถึงชนิดของรูปแบบ ซึ่ง S หมายถึง ซิมเพิล ส่วน C หมายถึง คอนสตรัค

ตารางที่ 2.4 แสดงค่าแท็กในชนิดข้อมูลแบบแอปพลิเคชันไวด์คลาสแท็ก คอถัมน์ S/C จะชี้ถึงชนิดของรูปแบบ ซึ่ง S หมายถึง ซิมเพิล ส่วน C หมายถึง คอนสตรัค

ตารางที่ 2.5 แสดงค่าของแท็กในชนิดข้อมูลแบบคอนเท็กซ์สเปซิฟิกคลาสแท็ก คอถัมน์ S/C จะชี้ถึงชนิดของรูปแบบ ซึ่ง S หมายถึง ซิมเพิล ส่วน C หมายถึง คอนสตรัค

ตารางที่ 2.3 ชนิดของยูนิเวอร์แซลคลาสแท็ก

Universal Class Tags							
Type	Bits 5-4-3-2-1					decimal	S/C
INTEGER	0	0	0	1	0	2	S
OCTEC STRING	0	0	1	0	0	4	S
NULL	0	0	1	0	1	5	S
OBJECT IDENTIFIER	0	0	1	1	0	6	S
SEQUENCE, SEQUENCE OF	1	0	0	0	0	16	C

ตารางที่ 2.4 ชนิดของแอปพลิเคชันไวด์คลาสแท็ก

Application-Wide Class Tags							
Type	Bits 5-4-3-2-1					Decimal	S/C
IpAddress	0	0	0	0	0	0	S
Counter	0	0	0	0	1	1	S
Guage	0	0	0	1	0	2	S
Time Ticks	0	0	0	1	1	3	S
Opaque	0	0	1	0	0	4	S

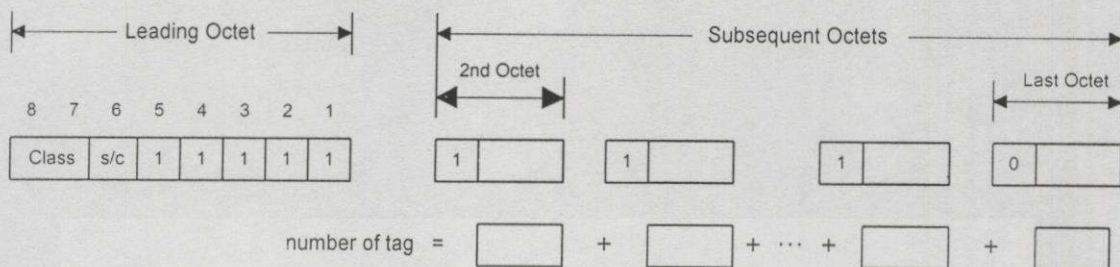
ตารางที่ 2.5 ชนิดของคอนเท็กซ์สเปซิฟิกคลาสแท็ก

Context-Specific Class Tags							
Type	bits 5-4-3-2-1					decimal	S/C
Get-Request PDU	0	0	0	0	0	0	C
Get-Next-Request PDU	0	0	0	0	1	1	C
Get-Response PDU	0	0	0	1	0	2	C
Set-Request PDU	0	0	0	1	1	3	C
Trap PDU	0	0	1	0	0	4	C

กฎการเข้ารหัสของ ASN.1 สำหรับ SNMP ที่กล่าวมาทั้งหมดนี้ใช้สำหรับแท็กนัมเบอร์ที่มีขนาดไม่เกิน 30 ซึ่งจะมีแท็กฟิลด์หนึ่งออกเทท แต่ถ้าแท็กนัมเบอร์ที่มีขนาดตั้งแต่ 31 ขึ้นไปจะมีแท็กฟิลด์หลายออกเททไม่สามารถใช้กฎการเข้ารหัสข้างต้นนี้ได้ สำหรับการเข้ารหัสแท็กนัมเบอร์ที่มีขนาดตั้งแต่ 31 ขึ้นไป ออกเททแรกในฟิลด์ของแท็กนัมเบอร์ 5 บิตแรกถูกเซตค่าเป็น 1 ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกเทตต่อไป ถ้าบิต 8 ถูกเซตค่าเป็น 1 จะชี้ถึงมีออกเทตต่อไป แต่ถ้าบิต 8 ถูกเซตค่าเป็น 0 จะชี้ถึงออกเทตสุดท้าย ส่วนบิต 7 ถึง 1 ของแต่ละออกเทต จะเป็นค่าของแท็กนั้นๆ โดยที่บิต 7 จะเป็นบิตแรกที่จะชี้ถึง MSB

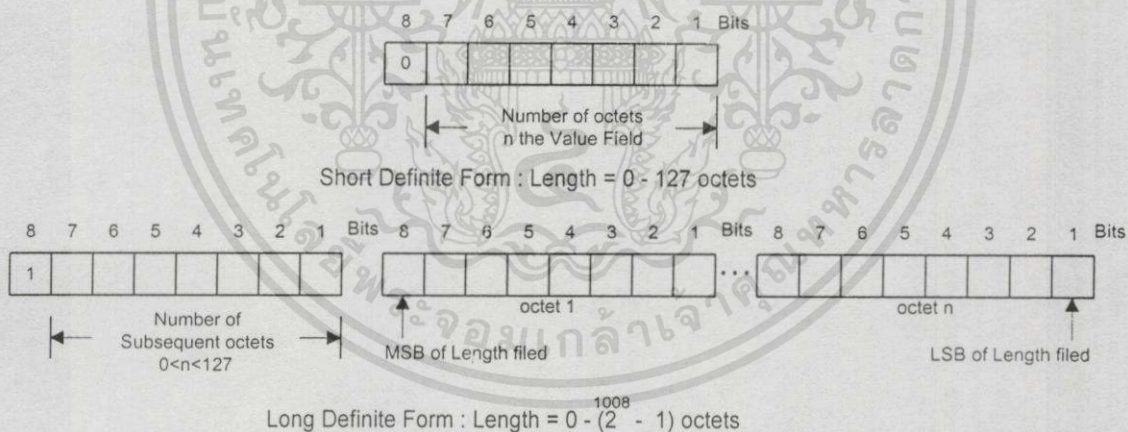


Type Field (tag>30)

รูปที่ 2.6 การเข้ารหัสไทป์ฟิลด์ (แท็ก >30)

ความยาว (Length Field)

ฟิลด์ความยาว จะเป็นส่วนกำหนดจำนวนออกเทตที่บรรจุในฟิลด์ค่า ฟิลด์ความยาวนี้อาจถูกกำหนดในรูปแบบสั้นหรือรูปแบบยาวที่แน่นอน (Definite) โดยมีบิต MSB คือบิต 8 ที่ใช้เป็นตัวกำหนดรูปแบบ แสดงดังรูปที่ 2.7



รูปที่ 2.7 การเข้ารหัสฟิลด์ความยาว

รูปแบบสั้นจะมีความยาวระหว่าง 0 ถึง 127 ออกเทต โดยกำหนดบิต 8 เป็น 0 ส่วน 7 บิตที่เหลือจะเป็นไบนารีที่ชี้ถึงความยาวของคอนเทนต์

รูปแบบยาวจะมีค่า 128 ออกเทตขึ้นไป โดยกำหนดบิต 8 เป็น 1 แต่สามารถมีความยาวที่น้อยกว่านี้ได้ รูปแบบยาวนี้ใช้หลายออกเทตในการแทนความยาวทั้งหมด โดยออกเทตแรก บิต 8 เป็น 1 ส่วนบิตที่เหลือจะเป็นไบนารีที่ชี้ถึงจำนวนออกเทตที่ตามมา มีค่าอยู่ระหว่าง 1 ถึง 126 ส่วนค่า 127 ถูกจองไว้ใช้ในอนาคต ออกเทตต่อมาทั้งหมดจะชี้ถึงความยาวของคอนเทนต์

ค่า (Value Filed)

ฟิลด์ค่า จะบรรจุข้อมูลภายใน (Content) ของออกเทต ซึ่งจะมีค่าของข้อมูลในแต่ละชนิดของ SNMP เช่น ค่าอักขระแอสกี หรือค่าของ OBJECT IDENTIFIER เป็นต้น

2.3 เอ็มไอบี (MIB - Management Information Base)

โครงสร้างชื่อการจัดการเครือข่ายอินเทอร์เน็ต มีตำแหน่ง Object Identifier หรือ OID ที่ {1.3.6.1} ถูกแบ่งย่อยออกเป็น 4 สาขาคือ

- directory(1) เป็นการจองไว้ใช้ในอนาคตของ OSI
- mgmt(2) เป็นส่วนของเอกสารที่ถูกรับรองจากอินเทอร์เน็ต เช่น มาตรฐานเอ็มไอบี ซึ่ง MIB-I และ MIB-II อยู่ภายใต้สาขานี้
- experimental(3) เป็นส่วนที่ใช้สร้างการทดสอบ, วิจัย และการทดลอง โดยมีหน่วยงาน IANA (Internet Assigned Number Authority) เป็นหน่วยงานกำหนดตัวเลขที่ใช้ในเครือข่ายอินเทอร์เน็ต
- private(4) มีสาขาย่อยคือ enterprise(1) เป็นส่วนที่ให้ผู้ขายลงทะเบียนเอ็มไอบีสำหรับอุปกรณ์ของตนได้ โดยมี IANA เป็นผู้กำหนดรหัสเอ็นเทอร์ไพรส์ ปัจจุบันมีมากกว่า 3000 องค์กรที่ลงทะเบียนแล้ว

เอ็มไอบี [3,4,5] หรือ ฐานข้อมูลสารสนเทศการจัดการ (Management Information Base) มีชื่อตามโครงสร้างรูปต้นไม้คือ

iso.org.dod.internet.mgmt.mib

หรือ OID

{ 1.3.6.1.2.1 }

กลุ่มเอ็มไอบีที่ใช้แพร่หลายมากที่สุดคือ MIB-I และ MIB-II เนื่องจากเอเจนต์ส่วนใหญ่ได้ออกแบบให้สนับสนุนออบเจกต์ในกลุ่มนี้ กลุ่ม MIB-I ประกอบด้วยกลุ่ม 8 กลุ่ม 114 ออบเจกต์ ส่วนกลุ่ม MIB-II ซึ่งได้ทำการพัฒนามาจาก MIB-I โดยเพิ่มกลุ่มขึ้นมาอีก 2 กลุ่ม 57 ออบเจกต์ รวมเป็น 10 กลุ่ม 171 ออบเจกต์ ซึ่งในหัวข้อนี้จะกล่าวถึงกลุ่ม MIB-II พอสังเขป ส่วนรายละเอียดของออบเจกต์ในกลุ่ม MIB-II และกลุ่ม MIB อื่นที่มีประโยชน์และเกี่ยวข้องกับการจัดการเครือข่ายแลนและแวน จะกล่าวถึงในบทต่อไป

กลุ่มระบบ (System Group)

กลุ่มระบบมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.1} ประกอบด้วยออบเจกต์ 7 ออบเจกต์ กลุ่มนี้บรรจุรายละเอียดคุณลักษณะ และคอนฟิกูเรชันโดยทั่วไปของอุปกรณ์

เช่น ชื่อโหนด, รายละเอียดโหนด, ตำแหน่งโหนด, OID โหนด และเวลาที่โหนดถูกเปิดใช้มานานเท่าไร ทุกออบเจกต์ในกลุ่มระบบมีสถานะเป็น Mandatory

กลุ่มอินเทอร์เฟซ (Interface Group)

กลุ่มอินเทอร์เฟซมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.2} ประกอบด้วย 2 รายการหลัก 22 ออบเจกต์ กลุ่มนี้ได้จัดเตรียมสารสนเทศอินเทอร์เฟซของฮาร์ดแวร์ รายการแรก ifNumber จะชี้ถึงจำนวนอินเทอร์เฟซทั้งหมด รายการสอง ifTable อยู่ในรูปแบบของตาราง บรรจุสารสนเทศทั้งหมดของอินเทอร์เฟซ แถวแต่ละแถวในตารางก็คือแต่ละอินเทอร์เฟซ แต่ละอินเทอร์เฟซมี 22 คอลัมน์ แต่ละคอลัมน์ได้จัดเตรียมสารสนเทศของอุปกรณ์ไว้ เช่นความเร็วของอินเทอร์เฟซ, ตำแหน่งทางกายภาพของฮาร์ดแวร์, สถานะการทำงานของอินเทอร์เฟซ และสถิติของการจราจร ทุกออบเจกต์ในกลุ่มอินเทอร์เฟซนี้มีสถานะเป็น Mandatory

กลุ่มเอที (Address Translation Group)

กลุ่มเอทีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.3} ประกอบด้วย 1 ตาราง 3 คอลัมน์ กลุ่มเอทีได้จัดเตรียมตารางการแปลงระหว่างไอพีแอดเดรสและแอดเดรสกายภาพ ทุกออบเจกต์ในกลุ่มเอทีนี้มีสถานะเป็น Deprecated ซึ่งสถานะนี้จะถูกยกเลิกในรุ่นต่อไป เนื่องจากมีออบเจกต์อื่นที่มีคุณสมบัติดีกว่ามาแทน แต่ปัจจุบันยังคงใช้อยู่

กลุ่มไอพี (Internet Protocol Group)

กลุ่มไอพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.4} กลุ่มนี้ได้จัดเตรียมสารสนเทศของโฮสต์และเราเตอร์บนฐานการทำงานของไอพี เช่นตารางเส้นทาง (Routing Table), ตารางการแปลงแอดเดรส (Address Conversion Table) และสถิติของดาต้าแกรมที่เกี่ยวข้องกับไอพี กลุ่มนี้ประกอบด้วยออบเจกต์ และตาราง 3 ตาราง คือตารางแอดเดรส ipAddrTable เป็นตารางที่นำไปสู่ค่าของแต่ละไอพีแอดเดรส, ตารางเน็ตเวิร์กทูมีเดีย ipNetToMediaTable เป็นตารางที่ใช้แมปค่าไอพีแอดเดรสไปยังแอดเดรสกายภาพ และตารางไอพีฟอร์เวิร์ด ipForwardTable ตารางนี้ถูกนำมาแทนที่ตารางไอพีเรดิง ipRoutingTable ซึ่งถูกเลิกใช้ (Obsolete) ใน MIB-II ทุกออบเจกต์ในกลุ่มไอพีนี้มีสถานะเป็น Mandatory

กลุ่มไอซีเอ็มพี (Internet Control Message Protocol Group)

กลุ่มไอซีเอ็มพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.5} ประกอบด้วย ออบเจกต์ 26 ออบเจกต์ กลุ่มนี้ใช้สำหรับการบำรุงรักษาทางสถิติและการนับของไอซีเอ็มพีเมสเสจ สำหรับการนับจะทำการนับจำนวนที่ส่ง, จำนวนที่รับ และจำนวนที่ผิดพลาดของไอซีเอ็มพีเมสเสจ ทุกออบเจกต์ในกลุ่มไอซีเอ็มพีนี้มีสถานะเป็น Mandatory

กลุ่มทีซีพี (Transmission Control Protocol Group)

กลุ่มทีซีพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.6} กลุ่มนี้จะทำการรวบรวมข้อมูลทางสถิติ และพารามิเตอร์ที่เกี่ยวข้องกับการเชื่อมต่อของทีซีพี เช่น จำนวนการเชื่อมต่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อของทีซีพี และจำนวนเซกเมนต์ที่ส่งผ่านทั้งหมด กลุ่มนี้ประกอบด้วย 14 ออบเจกต์ 1 ตาราง คือ ตารางการเชื่อมต่อ tcpConnTable ซึ่งบรรจุสารสนเทศที่เกี่ยวข้องกับการเชื่อมต่อของทีซีพี ทุก ออบเจกต์ในกลุ่มทีซีพีนี้มีสถานะเป็น Mandatory

กลุ่มยูดีพี (User Datagram Protocol Group)

กลุ่มยูดีพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.7} กลุ่มนี้จะเก็บสถิติ และสารสนเทศของยูดีพี การทำงานของยูดีพีจะทำงานแบบไม่มีการเชื่อมต่อก่อน (Connectionless) ซึ่งแตกต่างจากทีซีพีที่มีการทำงานแบบมีการเชื่อมต่อก่อน (Connection-Oriented) ดังนั้นยูดีพีจะไม่ มีการยืนยันการเชื่อมต่อก่อน (Establish) กลุ่มนี้ประกอบด้วย 4 ออบเจกต์ 1 ตาราง ออบเจกต์จะทำการดูแลรักษาข้อมูลทางสถิติ และนับจำนวนที่ผิดพลาดของยูดีพี ส่วนตารางยูดีพีจะเก็บการติดตาม (Track) ของไอพีแอดเดรสและหมายเลขพอร์ต ทุกออบเจกต์ในกลุ่มยูดีพีนี้มีสถานะเป็น Mandatory

กลุ่มอีจีพี (Exterior Gateway Protocol Group)

กลุ่มอีจีพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.8} อีจีพีทำการสื่อสารกันระหว่างระบบออโตโนมัส (Autonomous System) เรียกว่า อีจีพีเนเบอร์ (EGP Neighbors) แต่ละ อีจีพีเนเบอร์จะคุยกันเป็นระยะเพื่อตรวจสอบว่าเนเบอร์อื่นยังคงทำงานอยู่หรือไม่ เมื่อใดที่ระบบ ทำการสอบถามถึงกัน (Query) จะเรียกว่า อีจีพีโพลล์ (EGP Poll) กลุ่มอีจีพีนี้ประกอบด้วย 5 ออบเจกต์ 1 ตาราง ออบเจกต์จะทำการดูแลรักษาข่าวสารทางสถิติที่เกี่ยวข้องกับอีจีพี ส่วนตาราง egpNeighTable จะบรรจุสารสนเทศของอีจีพีเนเบอร์ที่เป็นประโยชน์ ทุกออบเจกต์ในกลุ่มอีจีพีนี้มี สถานะเป็น Mandatory

กลุ่มทรานสมิสชัน (Transmission Group)

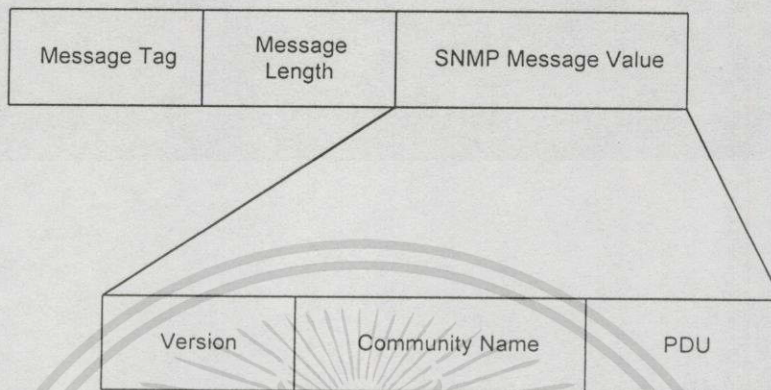
กลุ่มทรานสมิสชันมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.10} กลุ่มนี้ บรรจุออบเจกต์ที่เกี่ยวข้องกับการส่งผ่านข้อมูลของอินเทอร์เน็ต ภายใต้วงศ์ทรานสมิสชันจะมีกลุ่มย่อยของ MIB อยู่ กลุ่มย่อยนี้เริ่มแรกจะอยู่ในกลุ่มการทดลองก่อน (Experimental) ที่ตำแหน่ง OID {1.3.6.1.3} จนกระทั่งมีการยอมรับในการทดลองแล้ว (Proven) จึงถูกจัดสรรหมายเลขเข้าในกลุ่มนี้ ปัจจุบันมีกลุ่ม MIB ภายใต้วงศ์ทรานสมิสชัน 16 กลุ่ม

กลุ่มเอสเอ็นเอ็มพี (Simple Network Management Protocol Group)

กลุ่มเอสเอ็นเอ็มพีมีตำแหน่ง Object Identifier หรือ OID อยู่ที่ {1.3.6.1.2.1.11} ประกอบด้วยออบเจกต์ 30 ออบเจกต์ กลุ่มนี้ได้จัดเตรียมสารสนเทศสำหรับเอสเอ็นเอ็มพีออบเจกต์ รวมถึง เก็บการติดตาม (Track), รวบรวมการนับ, สถิติของเมสเสจ, เงื่อนไขของสถานะ และการตรวจสอบข้อผิดพลาด ทุกออบเจกต์ในกลุ่มเอสเอ็นเอ็มพีนี้มีสถานะเป็น Mandatory

2.4 โพรโทคอลเอสเอ็นเอ็มพี (SNMP- Simple Network Management Protocol)

การแลกเปลี่ยนสารสนเทศระหว่างสถานีจัดการ (Management Station) และเอเจนต์ จะอาศัยเอสเอ็นเอ็มพีเมสเซจ [2,3,4] ในการนำพาข่าวสาร แสดงดังรูปที่ 2.8



รูปที่ 2.8 เอสเอ็นเอ็มพีเมสเซจ

เอสเอ็นเอ็มพีเมสเซจ ประกอบด้วย 3 필ด์คือ

- เวอร์ชันฟิลด์ (Version) หมายถึง ความเข้ากันได้ของสถานีจัดการและเอเจนต์ ที่มีการใช้โปรโตคอลเอสเอ็นเอ็มพีเวอร์ชันเดียวกัน ถ้าเมสเซจของสถานีจัดการและเอเจนต์ใช้ต่างเวอร์ชันกัน เมสเซจนั้นจะถูกยกเลิกการทำงาน เวอร์ชันฟิลด์มีชนิดข้อมูลแบบ INTEGER

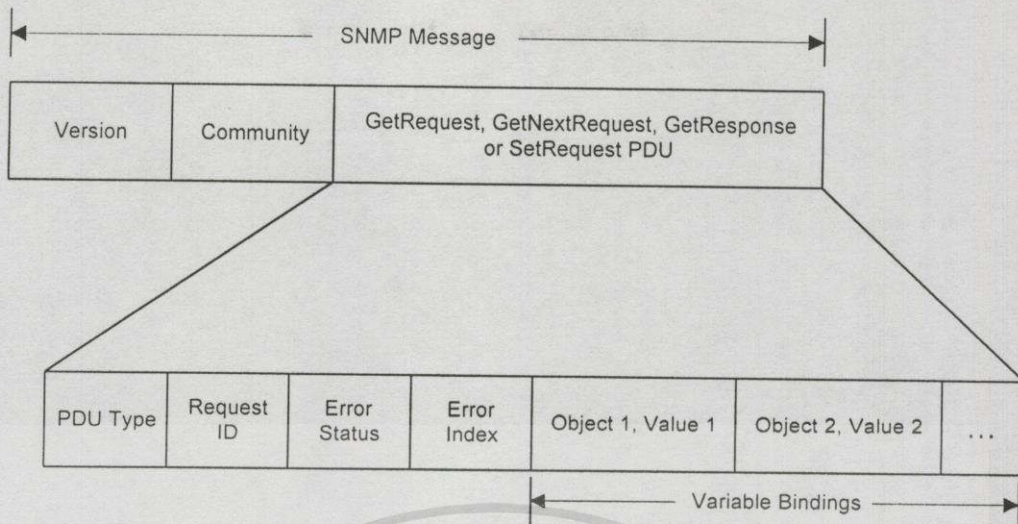
- คอมมิวนิตี้นามฟิลด์ (Community Name) หมายถึง กระบวนการในการรับรองสิทธิ (Authentication) ของสถานีจัดการก่อนการเข้าถึงเอเจนต์ เอเจนต์จะเก็บรายการคอมมิวนิตี้นามที่ใช้งานได้ และตรวจสอบคอมมิวนิตี้นามที่ตรงของเอสเอ็นเอ็มพีเมสเซจที่รับเพื่อทำการเปรียบเทียบกัน ถ้าตรงกัน (Match) ก็จะเริ่มกระบวนการทำงาน ถ้าไม่ตรงกันเมสเซจจะถูกยกเลิก เอเจนต์สามารถทำการคอนฟิกเพื่อให้ทำการส่งแทลป authenticationFailure ไปยังสถานีจัดการถ้าคอมมิวนิตี้นามไม่ตรงกัน คอมมิวนิตี้นามฟิลด์มีชนิดข้อมูลแบบ OCTET STRING

- พีดียูฟิลด์ (PDU- Protocol Data Unit) หมายถึง คำสั่งที่ใช้กระทำการต่างๆ ประกอบด้วย GetRequest, GetNextRequest, GetResponse, SetRequest และ Trap

2.4.1 รูปแบบพีดียู Get, Set และ Response

เอสเอ็นเอ็มพีมีรูปแบบพีดียู [3] ที่ใช้ร่วมกันคือ GetRequest, GetNextRequest, GetResponse และ SetRequest แสดงดังรูปที่ 2.9

- PDU Type ใช้ระบุชนิดของพีดียู คือ GetRequest(0), GetNextRequest(1), GetResponse(2), SetRequest(3) และ Trap(4)



รูปที่ 2.9 รูปแบบพีดิวที่ใช้ร่วมกัน

- Request ID พีดิวนี้มีชนิดข้อมูลแบบ INTEGER เป็นหมายเลขที่ใช้ส่งการร้องขอจากสถานีจัดการไปยังเอเจนต์ และหมายเลขนี้จะต้องแมตช์กันกับการตอบรับจากเอเจนต์ นอกจากนี้ยังสามารถจัดการเงื่อนไขความผิดพลาดการตอบรับค่า Request ID ที่ซ้ำกันซึ่งสถานีจัดการสามารถใช้ตัวจับเวลาไว้เพื่อไม่ตอบสนองค่า Request ID ที่ซ้ำกัน ส่วนกรณีที่ไม่รู้จักค่า Request ID สถานีจัดการจะทำการยกเลิกเมสเซนจ์นั้น

- Error Status พีดิวนี้มีชนิดข้อมูลแบบ enumerated INTEGER ใช้บ่งชี้สถานะว่าปกติ หรือมีเงื่อนไขความผิดพลาดเกิดขึ้น แสดงค่าดังตารางที่ 2.6

ตารางที่ 2.6 ค่าพีดิว Error Status

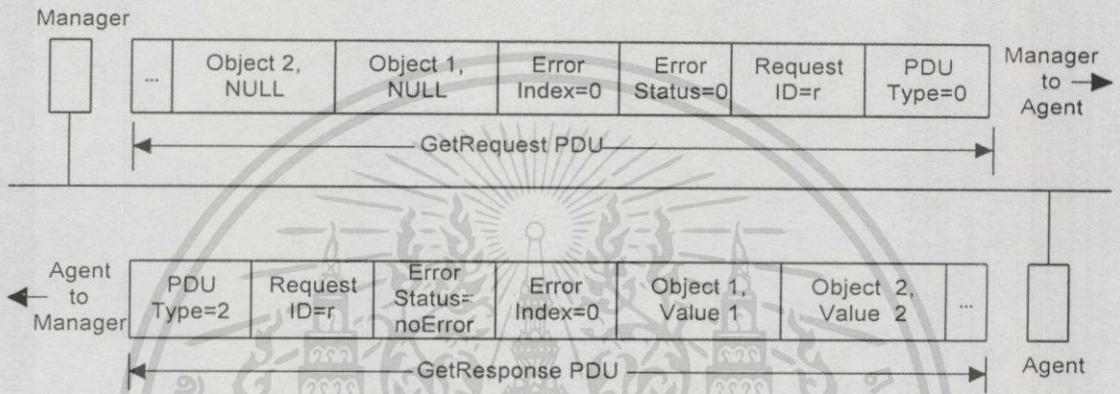
Status Name	Status Value
noError	0
tooBig	1
noSuchName	2
badValue	3
readOnly	4
genError	5

- Error Index พีดิวนี้ใช้ระบุเหตุการณ์ภายใน Variable Binding List ในกรณีพีดิว Error Status มีเงื่อนไขความผิดพลาดเกิดขึ้น เช่นถ้ามีความผิดพลาด readOnly เกิดขึ้นจะรีเทิร์นค่าของ Error Index = 4

- Variable Binding ใช้ชื่อว่า VarBindList ฟิวด์นี้จะมีรายการเป็นคู่ ภายในฟิวด์แต่ละคู่จะประกอบด้วยออบเจกต์ (Object) และค่า (Value) แต่ละฟิวด์ของ VarBindList จะบรรจุชนิดของออบเจกต์ OID และค่าที่ระบุ (Instance) ของ OID นั้น

2.4.2 GetRequest PDU

สถานีจัดการใช้คำสั่ง GetRequest PDU ในการเรียกคืนค่า (Retrieve) ออบเจกต์จากเอเจนต์ที่ต้องการส่วนใหญ่เป็นสกาลาร์ออบเจกต์(ไม่มีมิติ) ส่วนด้านเอเจนต์จะตอบสนองค่าที่ต้องการกลับมายังสถานีจัดการด้วยคำสั่ง GetResponse PDU แสดงดังรูป 2.10



รูปที่ 2.10 การส่งผ่าน GetRequest/GetResponse PDU

การสร้าง GetRequest PDU สถานีจัดการจะกำหนดค่า PDU Type เป็น 0 กำหนดค่า Request ID เซตค่า Error Status และ Error Index เป็น 0 ส่วน VarBindList บรรจุชื่อตัวแปรและค่าเป็น NULL และถ้าไม่มีเงื่อนไขความผิดพลาดอะไร ทางด้านเอเจนต์ก็จะทำการสร้าง GetResponse PDU กำหนดค่า PDU Type เป็น 2 เซตค่า Request ID ที่ตรงกัน เซต Error Status = noError เซต Error Index = 0 และ VarBindList ค่าที่บรรจุตัวแปรอยู่ขณะนั้นได้มาจากกระบวนการ GetRequest PDU ก่อนหน้านั้น

กรณีที่มีเงื่อนไขความผิดพลาดที่เป็นไปได้คือ

- ถ้าตัวแปรใน VarBindList มีออบเจกต์ไม่ตรงกัน เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = noSuchName และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา
- ถ้าตัวแปรเป็นชนิด Aggregate เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = noSuchName และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา
- ถ้าขนาดของ GetResponse PDU มากเกินกว่าข้อจำกัดทางด้านโพลิต เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = tooBig และ Error Index = 0

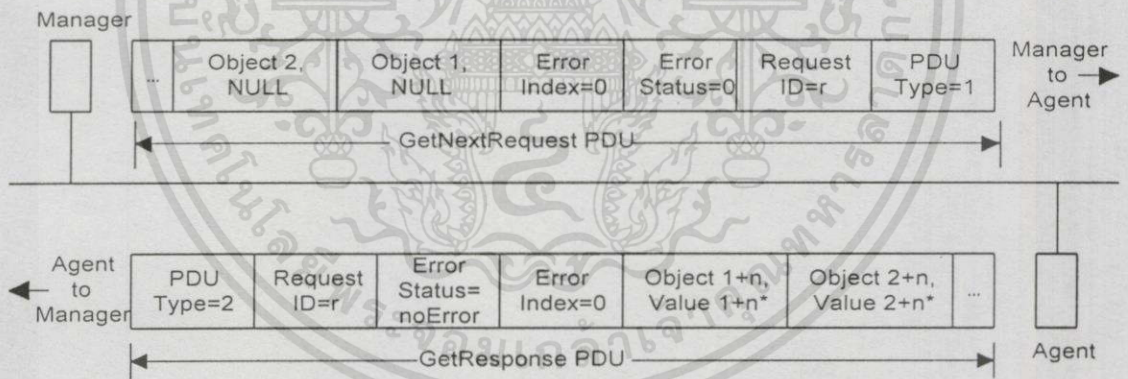
- ถ้าค่าของตัวแปรที่ถูกร้องขอไม่สามารถเรียกคืนค่ากลับได้ ไม่ว่าจะเป็นอย่างใดก็ตาม เอเจนต์จะทำการรีเทริน GetResponse PDU ด้วย Error Status = genErr และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา

2.4.3 GetNextRequest PDU

สถานีจัดการใช้คำสั่ง GetNextRequest PDU ในการเรียกคืนค่าของออบเจกต์จากเอเจนต์ที่ต้องการ โดยส่วนใหญ่จะเป็นคอลัมน์เนียบออบเจกต์ที่มีค่าภายในตาราง (มีมิติ) ส่วนด้านเอเจนต์จะตอบสนองค่าที่ต้องการกลับมายังสถานีจัดการด้วยคำสั่ง GetResponse PDU แสดงดังรูป 2.11

การสร้าง GetNextRequest PDU สถานีจัดการจะกำหนดค่า PDU Type เป็น 1 กำหนดค่า Request ID เขตค่า Error Status และ Error Index เป็น 0 ส่วน VarBindList บรรจุชื่อตัวแปรและค่าเป็น NULL และถ้าไม่มีเงื่อนไขความผิดพลาดอะไร ทางด้านเอเจนต์ก็จะทำการสร้าง GetResponse PDU กำหนดค่า PDU Type เป็น 2 เขตค่า Request ID ที่ตรงกัน เขต Error Status = noError เขต Error Index = 0 และ VarBindList ซึ่งบรรจุชื่อตัวแปรและค่ารวมเข้ากับลำดับ Lexicographical ที่ได้จากระบวนการ GetNextRequest PDU ก่อนหน้านั้น

ส่วนที่แตกต่างกันของ GetRequest และ GetNextRequest PDU คือ Lexicographical ซึ่งหมายถึง การเรียกคืนค่า (Retrieve) ของออบเจกต์ถัดไปที่มีขนาดใหญ่กว่า



*Object 1+n, Value 1+n แทนการสืบทอดลำดับ lexicographical

รูปที่ 2.11 การส่งผ่าน GetNextRequest/GetResponse PDU

กรณีที่มีเงื่อนไขความผิดพลาดที่เป็นไปได้คือ

- ถ้าตัวแปรใน VarBindList ไม่สามารถหาชื่อ Lexicographical ของออบเจกต์ได้ เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = noSuchName และ Error Index ชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา

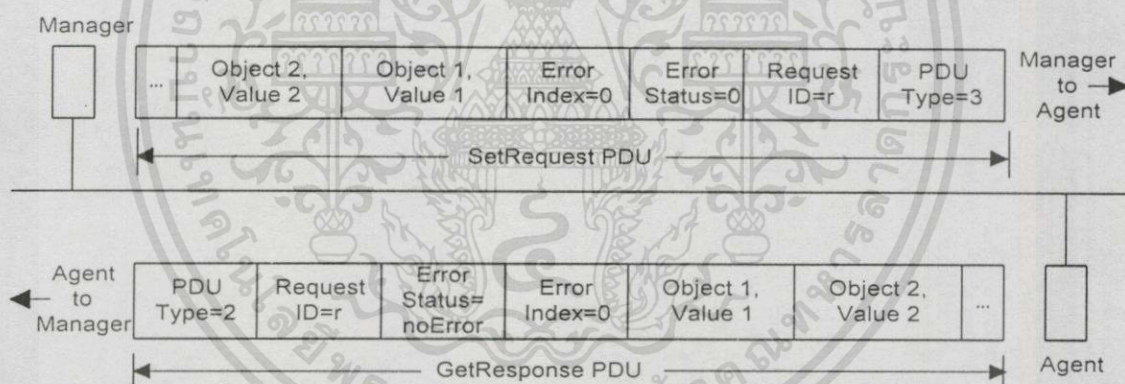
- ถ้าขนาดของ GetResponse PDU มากเกินกว่าข้อจำกัดทางด้านโพลิต เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = tooBig และ Error Index = 0

- ถ้าค่าของ Lexicographical ใน VarBindList ผิด ไม่สามารถเรียกคืนค่ากลับได้ไม่ว่าจะด้วยเหตุผลใด เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = genErr และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา

2.4.4 SetRequest PDU

สถานีจัดการจะใช้คำสั่ง SetRequest PDU ในการเซตเพื่อกำหนดค่าภายในของออบเจกต์ ส่วนด้านเอเจนต์จะตอบสนองค่าที่ต้องการกลับมายังสถานีจัดการด้วยคำสั่ง GetResponse PDU แสดงคังรูป 2.12

การสร้าง SetRequest PDU สถานีจัดการจะกำหนดค่า PDU Type เป็น 3 กำหนดค่า Request ID จากโพลิต เซตค่า Error Status และ Error Index เป็น 0 ส่วน VarBindList จะบรรจุชื่อตัวแปรและค่าที่ตรงกัน เมื่อเอเจนต์รับ SetRequest PDU มันจะเปลี่ยนค่าชื่อออบเจกต์ให้แก่ค่าใน VarBindList และถ้าไม่มีเงื่อนไขความผิดพลาดอะไร ทางด้านเอเจนต์จะทำการสร้าง GetResponse PDU กำหนดค่า PDU Type เป็น 2 เซต Error Status = noError เซต Error Index = 0



รูปที่ 2.12 การส่งผ่าน SetRequest/GetResponse PDU

กรณีที่มีเงื่อนไขความผิดพลาดที่เป็นไปได้คือ

- ถ้าตัวแปรใน VarBindList ไม่สามารถเซตการทำงานในเอ็มไอบีวีว ซึ่งเป็นเซตย่อยของเอ็มไอบีวออบเจกต์ได้ เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = noSuchName และ Error Index จะชี้ไปยังอินเด็กซ์ของออบเจกต์ที่เป็นปัญหา (บางครั้งเอเจนต์จะรีเทริน Error Status = readOnly ถ้าออบเจกต์คงอยู่ และตัวแปรจะต้องมีสิทธิอ่านอย่างเดียว Access = read-only)

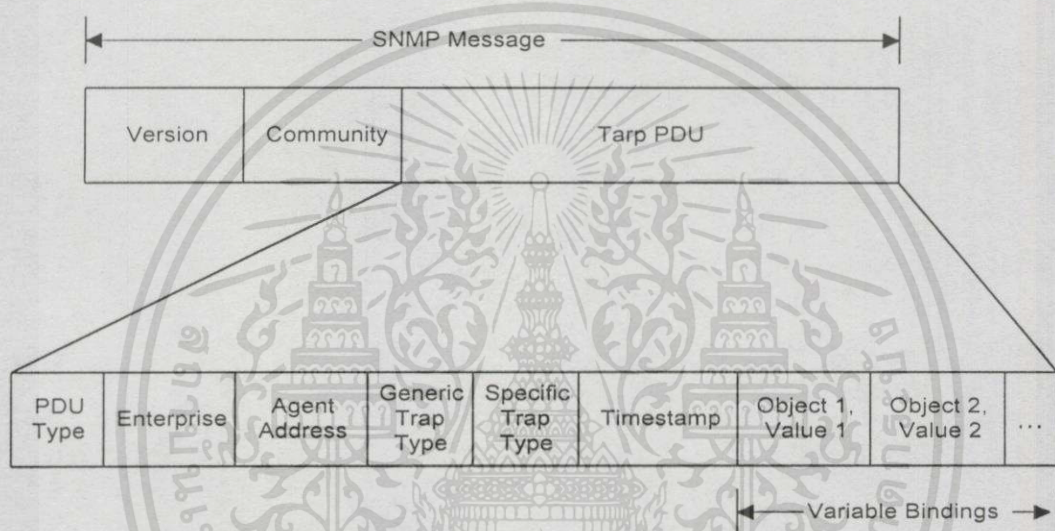
- ถ้าค่าของชื่อตัวแปรใน VarBindList ไม่ตรงกันกับชนิด, ความยาว และค่าของ ASN.1 เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status=badValue และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา

- ถ้าขนาดของ GetResponse PDU มากเกินกว่าข้อจำกัดทางด้าน โคลล เอเจนต์จะรีเทริน GetResponse PDU ด้วย Error Status = tooBig และ Error Index = 0

- ถ้าค่าของตัวแปรไม่สามารถทำการเปลี่ยนค่าได้ไม่ว่าจะเป็นด้วยเหตุผลใด เอเจนต์จะทำการรีเทริน GetResponse PDU ด้วย Error Status = genErr และ Error Index จะชี้ไปยังอินเด็กซ์ของตัวแปรที่เป็นปัญหา

2.4.5 แทรปพีดียู (Trap PDU)

เอเจนต์ใช้แทรปพีดียู ในการแจ้งเตือน ไปยังสถานีจัดการ เพื่อรายงานเหตุการณ์สำคัญที่เกิดขึ้น แทรปพีดียูจะมีโครงสร้างที่แตกต่างจากพีดียูทั้ง 4 ชนิดที่กล่าวมา แสดงดังรูปที่ 2.13



รูปที่ 2.13 โครงสร้างของแทรปพีดียู

- 필ด์ PDU Type ใช้ระบุถึงชนิดของแทรปพีดียู ซึ่งมีค่า PDU Type = 4
 - 필ด์ Enterprise บรรจุ OID ของอุปกรณ์ที่ถูกลงทะเบียนในกลุ่มของเอ็นเทอร์ไพรส์เพื่อใช้ในการกำเนิดแทรปของอุปกรณ์
 - 필ด์ Agent Address บรรจุไอพีแอดเดรสของเอเจนต์ เพื่อเตรียมการกำหนดชื่อต่อไป
 - 필ด์ Generic Trap Type เตรียมการสำหรับสารสนเทศ เพื่อรายงานเหตุการณ์ต่าง ๆ
- ฟิลด์นี้กำหนดค่าเป็นชนิดของเลขจำนวนเต็มที่ถูกระบุค่า โดยกำหนดเป็น 7 ค่าคือ

coldStart(0) : เป็นการส่งโปรโตคอลเอนทิตี ที่มีการเริ่มต้นใหม่ด้วยตัวมันเอง ซึ่งชี้ถึงคอนฟิกูเรชันของเอเจนต์หรือโปรโตคอลเอนทิตีมีการเปลี่ยนแปลง แต่ไม่ถึงกับรีสตาร์ท เนื่องจากการขัดข้อง (Crash) หรือการผิดพลาดหลัก

warmStart(1) : เป็นการส่งโปรโตคอลเอนทิตี ที่มีการเริ่มต้นใหม่ด้วยตัวมันเอง ซึ่งชี้ถึงคอนฟิกูเรชันของเอเจนต์และโปรโตคอลเอนทิตีไม่ถูกเปลี่ยนแปลง ซึ่งคือรูทีนการรีสตาร์ท

linkUp(3) : เป็นสัญญาณที่บ่งบอกถึงข่ายเชื่อมโยงการสื่อสารใช้งานได้ การอ้างอิงถึงอินเทอร์เฟซถูกระบุจากองค์ประกอบแรกในฟิลด์ VarBindList ซึ่งก็คือชื่อและค่าของ ifIndex อินสแตนซ์

authenticationFailure(4) : เอเจนต์รับเอสเอ็มพีเมสเสจ ที่มีการรับรองสิทธิ (Authentication) ไม่ถูกต้องจากสถานีจัดการ นั่นคือคอมมิวนิตี้นาม (Community name) ไม่ถูกต้อง

egpNeighborLoss(5) : สัญญาณของอีจีพีพีเพียนเบอร์ (EGP Peer Neighbors) หยุดการทำงาน

enterpriseSpecific(6) : เป็นการเกิดแทรปที่ไม่ใช่ Generic Trap ซึ่งถูกระบุโดยฟิลด์ Enterprise และฟิลด์ Specific Trap Type

- ฟิลด์ Specific Trap Type เป็นฟิลด์ที่ผู้ผลิตบรรจุค่าแทรปที่กำหนดขึ้นมาเป็นพิเศษกับอุปกรณ์เหล่านั้น

- ฟิลด์ Time Stamp บรรจุเวลาล่วงผ่านระหว่างการเริ่มต้นใหม่ที่สุดของเอเจนต์ และการกำเนิดแทรป

- ฟิลด์ Variable Binding List บรรจุสารสนเทศการนำไปใช้งานเพิ่มเติมที่เกี่ยวข้องกับแทรปพีดียูทั้ง 5 ที่มีรูปแบบของ VarBindList เดียวกัน

การจัดการอินเทอร์เน็ตเฟซ

การจัดการอินเทอร์เน็ตเฟซ หมายถึงการจัดการที่ชั้นอินเทอร์เน็ตเนตลงมา ครอบคลุมรูปแบบการจัดการทั้งหมดของอินเทอร์เน็ตเฟซ ชั้นอินเทอร์เน็ตเนตลงมาจะอ้างอิงถึงชุดอินเทอร์เน็ตโพรโตคอล (ไอพี) เป็นหลัก บทนี้จะอธิบายถึงโครงสร้างทั้งหมดของสารสนเทศ และความหมายของออบเจ็กต์พิเศษที่เกี่ยวข้องกับการจัดการอุปกรณ์อินเทอร์เน็ตเฟซเครือข่าย [6]

3.1 กลุ่มระบบ (System Group)

กลุ่มระบบ [7] กำหนดใน MIB-II ตามเอกสาร RFC 1213 ออบเจ็กต์กลุ่มนี้ประกอบด้วย

- sysDescr (DisplayString) : หมายถึง รายละเอียดของอุปกรณ์ เช่น ชื่อเต็ม, การระบุรุ่นชนิดของฮาร์ดแวร์, รุ่นของซอฟต์แวร์ระบบปฏิบัติการ และรุ่นของซอฟต์แวร์เครือข่าย
- sysObjectID (OBJECT IDENTIFIER) : หมายถึง การระบุหมายเลขออบเจ็กต์เพื่อบ่งบอกถึงโรงงานผู้ผลิต ค่าหมายเลขถูกกำหนดใน SMI เอ็นเทอร์ไพรส์ซันทรี (1.3.6.1.4.1)
- sysUpTime (TimeTicks) : หมายถึง เวลาที่บ่งบอกถึงอุปกรณ์เริ่มต้นทำงานมานานเท่าไร ค่าที่ได้มีหน่วยเป็น 1/100 วินาที
- sysContact (DisplayString) : หมายถึง การระบุชื่อบุคคลที่ทำการติดต่อเมื่อมีปัญหา
- sysName (DisplayString) : หมายถึง การบ่งบอกถึงชื่อของอุปกรณ์
- sysLocation (DisplayString) : หมายถึง การบ่งบอกถึงตำแหน่งทางกายภาพของอุปกรณ์
- sysServices (INTEGER sum) : หมายถึง การบ่งบอกถึงระดับชั้นโพรโตคอลที่ให้บริการอุปกรณ์ ซึ่งจะเป็นการรวมค่าแต่ละชั้นตามสูตร $2^{(L-1)}$ โดย L คือหมายเลขชั้นโพรโตคอล ค่าระดับชั้นต่าง ๆ ที่ใช้คำนวณแสดงดังตาราง 3.1

ตารางที่ 3.1 ค่าการคำนวณระดับชั้น

ชั้น	หน้าที่	2^{L-1}
1	ฟิสิคัล (เช่น รีพิตเตอร์)	1
2	ดาต้าลิงก์ (เช่น บริดจ์)	2
3	อินเทอร์เน็ต (เช่น ไอพี เกตเวย์)	4
4	เอ็นค็พยูเอ็นค็ (เช่น ไอพี โสศค)	8
7	แอปพลิเคชัน (เช่น อี-เมล)	64

เช่น ถ้าอุปกรณ์มีค่า 78 หมายถึงอุปกรณ์นั้นทำงานให้บริการหลักชั้นแอปพลิเคชัน (ชั้น7), ชั้นเอ็นด์ทูเอ็นด์ (ชั้น4), ชั้นอินเทอร์เน็ต (ชั้น3) และชั้นดาต้าลิงก์ (ชั้น2) ซึ่งเมื่อรวมค่าแล้วได้ 78 คือ $2^{(7-1)} + 2^{(4-1)} + 2^{(3-1)} + 2^{(2-1)}$

3.2 กลุ่มอินเทอร์เฟซ (Interface Group)

กลุ่มอินเทอร์เฟซ [7] บรรจุสารสนเทศทั่วไปบนเอนทิตีชั้นอินเทอร์เฟซ กลุ่มนี้ต้นแบบถูกกำหนดใน MIB-II ตามเอกสาร RFC 1213 และกลุ่ม Evolution ถูกกำหนดเพิ่มเติม ตามเอกสาร RFC 1573 อย่างไรก็ตาม เอนเจ็ตส่วนใหญ่ยังไม่ปรับให้สนับสนุนกลุ่ม Evolution ยังคงสนับสนุนกลุ่มตามข้อกำหนดของ MIB-II

กลุ่มอินเทอร์เฟซตามข้อกำหนดของ MIB-II ประกอบด้วยหนึ่งสกาลาร์ออบเจกต์ (ไม่มีมิติ) และหนึ่งคอตมันน์เนียบออบเจกต์ (มีมิติในรูปแบบตาราง)

สกาลาร์ออบเจกต์ คือ

ifNumber (INTEGER) : หมายถึง จำนวนของอินเทอร์เฟซเครือข่ายที่ถูกเชื่อมต่อไปยังอุปกรณ์ (ไม่คำนึงถึงสถานะปัจจุบันของอินเทอร์เฟซ)

ออบเจกต์ตาราง คือ

ifTable จะบรรจุรายการเอนทรี (Entry) ของอินเทอร์เฟซ จำนวนเอนทรีจะเป็นไปตามค่าของ ifNumber ซึ่งหมายถึงจำนวนแถวในตารางนั่นเอง แถวแต่ละแถวคือแต่ละอินเทอร์เฟซของอุปกรณ์ การกำหนดรูปแบบ (Template) ของแถวจะใช้ ifEntry ซึ่งบรรจุรายการออบเจกต์สำหรับอินเทอร์เฟซโดยเฉพาะ และค่าของ ifIndex ใช้สำหรับระบุแถวของแต่ละอินเทอร์เฟซ ค่านี้เป็นเลขจำนวนเต็มที่ไม่ซ้ำกันมีค่าอยู่ระหว่าง 1 และค่าของ ifNumber (ไม่จำเป็นต้องเรียงลำดับต่อเนื่องกัน)

ตาราง ifTable ใช้หมายเลขอินเทอร์เฟซ เป็นคอลลัมน์เพื่อระบุรูปแบบแถวในตาราง

INDEX { ifIndex }

ออบเจกต์ในตาราง ifTable ที่เกี่ยวข้องกับการนับจำนวน (Counter) มีดังนี้

ifInOctets (Counter) : หมายถึง จำนวนทั้งหมดของออกแพทที่รับบนอินเทอร์เฟซ

ifInUcastPkts (Counter) : หมายถึง จำนวนของอินพุตยูนิคาสต์แพ็กเก็ต ที่ถูกส่งไปยังชั้นที่สูงกว่า

ifInNUcastPkts (Counter) : หมายถึง จำนวนของอินพุตบรอดคาสต์/มัลติคาสต์แพ็กเก็ต ที่ถูกส่งไปยังชั้นที่สูงกว่า

ifInDiscards (Counter) : หมายถึง จำนวนของอินพุตแพ็กเก็ตที่ถูกยกเลิก เนื่องจากจํากัดของทรัพยากร เช่น บัฟเฟอร์ล้น (Buffer Overflow)

ifInErrors (Counter) : หมายถึง จำนวนของอินพุตแพ็กเก็ตที่ถูกยกเลิก เนื่องจากเกิดความผิดพลาด

ifInUnknownProtos (Counter) : หมายถึง จำนวนของอินพุตแพ็กเก็ตที่ถูกยกเลิก เนื่องจากไม่รู้จักโปรโตคอล หรือไม่สนับสนุนโปรโตคอลนั้น ๆ

ifOutOctets (Counter) : หมายถึง จำนวนทั้งหมดของออกเทตที่ถูกส่งบนอินเทอร์เฟซ

ifOutUcastPkts (Counter) : หมายถึง จำนวนทั้งหมดของเอาต์พุตยูนิคาสต์แพ็กเก็ต จากชั้นที่สูงกว่า

ifOutNUcastPkts (Counter) : หมายถึง จำนวนทั้งหมดของเอาต์พุตบรอดคาสต์/มัลติคาสต์แพ็กเก็ต จากชั้นที่สูงกว่า

ifOutDiscards (Counter) : หมายถึง จำนวนของเอาต์พุตแพ็กเก็ตที่ถูกยกเลิก เนื่องจากการจำกัดของทรัพยากร เช่น บัฟเฟอร์ล้น (Buffer Overflow)

ifOutErrors (Counter) : หมายถึง จำนวนของเอาต์พุตแพ็กเก็ตที่ถูกยกเลิก เนื่องจากเกิดความผิดพลาด

ออบเจกต์อื่น ๆ ในตาราง ifTable มีดังนี้

ifIndex (INTEGER) : หมายถึง หมายเลขอินเทอร์เฟซ เป็นค่าเลขจำนวนเต็มที่ไม่ซ้ำกันของแต่ละอินเทอร์เฟซ

ifDescr (DisplayString) : หมายถึง รายละเอียดของอินเทอร์เฟซ เช่น ชื่อของผลิตภัณฑ์, ชื่อของผู้ผลิต และรุ่นของอินเทอร์เฟซ

ifType (INTEGER) : หมายถึง ชนิดของอินเทอร์เฟซ ซึ่งจะถูกแบ่งแยกตามโปรโตคอลกายภาพหรือการเชื่อมโยง (Physical/Link)

ifMtu (INTEGER) : หมายถึง ขนาดหน่วยการส่งข้อมูลสูงสุด (Maximum Transmission Unit ; MTU) ของออกเทตที่สามารถรับหรือส่งบนอินเทอร์เฟซ

ifSpeed (Gauge32) : หมายถึง อัตราการส่งผ่านข้อมูล มีหน่วยเป็นบิต/วินาที

ifPhysAddress (PhysAddress) : หมายถึง แอดเดรสที่ระดับชั้นอินเทอร์เฟซ

ifAdminStatus (enumerated INTEGER) : หมายถึง สถานะของอินเทอร์เฟซที่ต้องการ คือ ทำงาน (Up), หยุดทำงาน (Down) และทดสอบ (Testing)

ifOperStatus (enumerated INTEGER) : หมายถึง สถานะของอินเทอร์เฟซในปัจจุบัน คือ ทำงาน (Up), หยุดทำงาน (Down) และทดสอบ (Testing)

ifLastChange (TimeStamp) : หมายถึง เวลาที่อินเทอร์เฟซถูกเปลี่ยนสถานะครั้งสุดท้าย

ifOutQLen (Gauge32) : หมายถึง ความยาวของคิวแพ็กเก็ตด้านเอาต์พุต

ifSpecific (OBJECT IDENTIFIER) : หมายถึง การอ้างอิงถึงเอ็มไอบีโมดูล เพื่อใช้ในการระบุไปยังชื่อของอินเทอร์เฟซ

ออบเจ็กต์กลุ่มอินเทอร์เฟซนี้ ได้บรรจุสารสนเทศพื้นฐานที่มีประโยชน์ ในด้านการเฝ้าดูประสิทธิภาพ หรือการควบคุมความผิดพลาดของเครือข่าย ดังตัวอย่างของการนำไปใช้งาน คือ การตรวจสอบความแอ็ดของข้อมูลในเครือข่าย, การวัดขนาดของออกเทททั้งทางด้านอินพุตและด้านเอาต์พุตของระบบ, การวัดความยาวของคิวสำหรับเอาต์พุต และการชี้เตือนถึงปัญหาที่จะเกิดขึ้นในเร็วๆ นี้

กลุ่มอินเทอร์เฟซ Evolution [8] กำหนดเพิ่มเติมตามเอกสาร RFC 1573 จะมีการปรับปรุงเพิ่มเติมออบเจ็กต์ในตาราง ifTable และเพิ่มตารางใหม่อีก 4 ตาราง คือ

ตาราง ifStackTable จัดเตรียมความหมายเพื่อใช้อ้างความสัมพันธ์ระหว่างสองชั้นย่อย
รูปแบบแถวในตาราง ifStackTable ถูกระบุโดย 2 INTEGER

INDEX { ifStackHighLayer, ifStackLowerLayer }

โดยออบเจ็กต์ ifStackHighLayer มีค่า ifIndex ของชั้นย่อยที่สูงกว่า และออบเจ็กต์ ifStack-LowerLayer มีค่า ifIndex ของชั้นย่อยที่ต่ำกว่า

ภายในตารางบรรจุออบเจ็กต์

ifStackStatus (enumerated RowStatus) : หมายถึง สถานะของแถว สนับสนุนเพียงแต่ค่า active และ ไม่อนุญาตให้มีการเขียนออบเจ็กต์

ตาราง ifXTable กำหนดออบเจ็กต์ใหม่เสริมจากตาราง ifTable

รูปแบบแถวในตาราง ifXTable ถูกระบุโดย

AUGMENTS { ifEntry }

ภายในตารางบรรจุออบเจ็กต์ใหม่ที่เกี่ยวข้องกับการนับ 4 ออบเจ็กต์แบบ 32 บิต คือ

ifInMulticastPkts (Counter32) : หมายถึง จำนวนของอินพุตมัลติคาสท์แพ็กเก็ต ที่ส่งไปยังชั้นที่สูงกว่า

ifInBroadcastPkts (Counter32) : หมายถึง จำนวนของอินพุตบรอดคาสท์แพ็กเก็ต ที่ส่งไปยังชั้นที่สูงกว่า

ifOutMulticastPkts (Counter32) : หมายถึง จำนวนของเอาต์พุตมัลติคาสท์แพ็กเก็ต ที่ส่งจากชั้นที่สูงกว่า

ifOutBroadcastPkts (Counter32) : หมายถึง จำนวนของเอาต์พุตบรอดคาสท์แพ็กเก็ต ที่ส่งจากชั้นที่สูงกว่า

ออบเจ็กต์ที่กำหนดใช้สำหรับอินเทอร์เฟซความเร็วสูง 8 ออบเจ็กต์แบบ 64 บิต คือ

ifHCInOctets (Counter64)

ifHCOctets (Counter64)

ifHCInUcastPkts (Counter64)

ifHCInMulticastPkts (Counter64)

ifHCInBroadcastPkts (Counter64)

ifHCOUcastPkts (Counter64)

ifHCOUmulticastPkts (Counter64)

ifHCOUbroadcastPkts (Counter64)

ความหมายของออบเจกต์ทั้ง 8 จะเหมือนกับออบเจกต์ข้างต้นแบบ 32 บิตที่ผ่านมา ยกเว้น Syntax ที่ขยายเป็น 64 บิต

ออบเจกต์ใหม่อื่น ๆ 5 ออบเจกต์คือ

ifName (DisplayString) : หมายถึง ชื่อของอินเทอร์เฟซ ซึ่งถูกกำหนดโดยอุปกรณ์โลคอล เช่น "le0" หรือหมายเลขพอร์ต เช่น "1"

ifLinkUpDownTrapEnable (enumerated INTEGER) : หมายถึง การชี้ถึงแทรป LinkUp/LinkDown ถูก enable(1) หรือ disable(2)

ifHighSpeed (Gauge32) : หมายถึง ความเร็วของอินเทอร์เฟซ มีหน่วยเป็น Mb/s

ifPromiscuousMode (TruthValue) : หมายถึง อินเทอร์เฟซอยู่ในโหมด Promiscuous ถ้าค่า true(1) ชี้ถึงสถานียอมรับทุกแพ็กเก็ต/เฟรม ที่ถูกส่งบนสื่อ แต่ถ้าค่า false(2) ชี้ว่าอินเทอร์เฟซเพียงแค่ออมรับแพ็กเก็ต/เฟรม จากแอดเดรสของสถานีนั้น

ifConnectorPresent (TruthValue) : หมายถึง อินเทอร์เฟซมีการเชื่อมต่อทางกายภาพ ถ้ามีค่า true(1) แต่ถ้าไม่มีการเชื่อมต่อจะมีค่า false(2)

ตาราง ifTestTable กำหนดสำหรับการวินิจฉัยหรือการทดสอบอื่น ๆ เช่น ลูปแบ็ก รูปแบบแถวในตาราง ifTestTable ถูกระบุโดย

AUGMENTS { ifEntry }

ภายในตารางประกอบด้วยออบเจกต์ คือ

ifTestId (TestAndIncr) : หมายถึง การอ้างหมายเลขทดสอบของอินเทอร์เฟซปัจจุบัน

ifTestStatus (enumerated INTEGER) : หมายถึง การทดสอบดำเนินงานต่อไปในปัจจุบัน

ifTestType (AutonomousType) : หมายถึง ชนิดการทดสอบที่ใช้ดำเนินการ

ifTestResult (enumerated INTEGER) : หมายถึง ผลการทดสอบมาตรฐาน คือ none(1), success(2), inProgress(3), notSupported(4), unableToRun(5), aborted(6), failed(7)

ifTestCode (OBJECT IDENTIFIER) : หมายถึง รหัสที่ใช้แสดงผลการทดสอบพิเศษ เช่น รหัสความผิดพลาดหลังจากการทดสอบขัดข้อง

ifTestOwner (OwnerString) : หมายถึง ใครที่ถูกอ้างในการทดสอบปัจจุบัน

ตาราง ifRcvAddressTable ใช้สำหรับรับแพ็กเก็ตเกิด/เฟรม ที่มากกว่าหนึ่งแอดเดรสอาจเป็น บรอดคาสต์, มัลติคาสต์ หรือยูนิคาสต์

รูปแบบแถวในตาราง ifRcvAddressTable ถูกระบุโดย

INDEX { ifIndex, IfRcvAddressAddress }

ภายในตารางประกอบด้วยออบเจกต์ คือ

ifRcvAddressStatus (RowStatus) : หมายถึง สถานะปัจจุบันของแถว สามารถสร้าง และลบ แถวในตาราง ifRcvAddressTable

ifRcvAddressType (enumerated INTEGER) : หมายถึง การชี้ถึงแอดเดรสที่ยังคงถูกเก็บ รักษาไว้หลังจากรีเซ็ตระบบใหม่ มีค่าคือ other(1), volatile(2), nonVolatile(3)

3.3 กลุ่มเอที (Address Translation Group)

กลุ่มเอที [7] ถูกกำหนดใน MIB-II ตามเอกสาร RFC 1213 กลุ่มนี้ประกอบด้วยหนึ่งตาราง คือตาราง atTable ใช้สำหรับการแปลงค่า (Mapping) ของแอดเดรสเครือข่ายไปยังแอดเดรสกายภาพ ตาราง atTable ใช้หมายเลขอินเทอร์เน็ตเฟสและไอพีแอดเดรสของการแมปปีง เป็นครรชนิเพื่อ ระบุรูปแบบแถวในตาราง

INDEX { atIfIndex, atNetAddress }

แถวแต่ละแถวของตารางจะบรรจุด้วย 3 คอลัมน์ คือ

atIfIndex (INTEGER) : หมายถึง หมายเลขอินเทอร์เน็ตเฟส ซึ่งเป็นผลที่ได้จากการแมปปีง

atPhyAddress (PhyaAddress) : หมายถึง แอดเดรสกายภาพของการแมปปีง

atNetAddress (NetworkAddress) : หมายถึง ไอพีแอดเดรสของการแมปปีง

กลุ่มเอที มีสถานะในปัจจุบันเป็น Deprecated คือสถานะนี้อาจจะถูกยกเลิกในรุ่นต่อไป เพราะมีออบเจกต์อื่นที่มีคุณสมบัติที่ดีกว่า แต่ขณะนี้ยังคงใช้อยู่

3.4 กลุ่มไอพี (Internet Protocol)

กลุ่มไอพี [7] ถูกกำหนดใน MIB-II ตามเอกสาร RFC 1213 กลุ่มนี้บรรจุสารสนเทศที่ เกี่ยวข้องกับการนำไปใช้งานของอุปกรณ์ที่สนับสนุนไอพี เช่น โฮสต์ และเราเตอร์

ออบเจกต์กลุ่มนี้ ประกอบด้วยสกาลาร์ออบเจกต์ และตาราง 4 ตาราง โดย 3 ตารางแรก กำหนดตามเอกสาร RFC 1213 และ 1 ตารางกำหนดตามเอกสาร RFC 1354

สกาลาร์ออบเจกต์

สกาลาร์ออบเจกต์ ที่เกี่ยวข้องกับคานับจำนวน (Counter) คือ

ipInReceives (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมทั้งหมดที่ได้รับจากอินเทอร์เน็ตเฟซ รวมถึงการรับดาต้าแกรมที่มีความผิดพลาดด้วย

ipInHdrErrors (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมที่ถูกยกเลิก (Discard) เนื่องจากเกิดความผิดพลาดขึ้นในไอพีเฮดเดอร์

ipInAddrErrors (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมที่ถูกยกเลิก (Discard) เนื่องจากการขนส่งที่ผิดพลาด (Misdelivery) ในฟิลด์ของไอพีแอดเดรส

ipForwDatagrams (Counter) : หมายถึง จำนวนดาต้าแกรมที่ถูกส่งต่อ (Forward) เนื่องจากไม่ใช่ไอพีปลายทางสุดท้าย

ipInUnknownProtos (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมที่ได้รับเป็นผลสำเร็จ แต่ถูกยกเลิก เนื่องจากไม่รู้จักโปรโตคอล หรือไม่สนับสนุนโปรโตคอลนั้น

ipInDiscards (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมที่ถูกยกเลิก เนื่องจากทรัพยากรจำกัด เช่น จำนวนบัฟเฟอร์ไม่เพียงพอ

ipInDelivers (Counter) : หมายถึง จำนวนอินพุตดาต้าแกรมที่ถูกขนส่งเป็นผลสำเร็จไปยังชั้นที่สูงกว่า

ipOutRequests (Counter) : หมายถึง จำนวนเอาต์พุตดาต้าแกรมที่ทำการร้องขอ ซึ่งถูกรับมาจากชั้นที่สูงกว่า

ipOutDiscards (Counter) : หมายถึง จำนวนเอาต์พุตดาต้าแกรมที่ถูกยกเลิก เนื่องจากทรัพยากรไม่เพียงพอ

ipOutNoRoutes (Counter) : หมายถึง จำนวนเอาต์พุตดาต้าแกรมที่ถูกยกเลิก เนื่องจากการหาเส้นทางจากเราดิงอินฟอร์เมชันไม่พบ

ipReasmReqds (Counter) : หมายถึง จำนวนเฟรกเมนต์ที่รับ ซึ่งมีความจำเป็นต้องทำการรีแอสเซมบลีใหม่

ipReasmOKs (Counter) : หมายถึง จำนวนดาต้าแกรมที่เป็นผลสำเร็จ ซึ่งได้มาจากการรีแอสเซมบลี

ipReasmFails (Counter) : หมายถึง จำนวนที่ทำการรีแอสเซมบลีแล้วเกิดการล้มเหลว (Failure)

ipFragOKs (Counter) : หมายถึง จำนวนดาต้าแกรมที่เป็นผลสำเร็จจากการเฟรกเมนต์

ipFragFails (Counter) : หมายถึง จำนวนดาต้าแกรมที่ถูกยกเลิก เนื่องจากจำเป็นต้องทำการเฟรกเมนต์ แต่แพ็กเฟรกเมนต์ไม่ถูกเซตให้ใช้งาน

ipFragCreates (Counter) : หมายถึง จำนวนของเฟรกเมนต์ที่ถูกสร้าง เนื่องมาจากการทำเฟรกเมนต์เดชัน

ipRoutingDiscards (Counter) : หมายถึง จำนวนของเส้นทางที่ถูกยกเลิก ถึงแม้ว่าเส้นทางนั้นจะใช้ได้

สกาลาร์ออบเจกต์ อื่น ๆ ในกลุ่มนี้ คือ

ipForwarding (INTEGER) : หมายถึง การซึ่งถึงอุปกรณ์นั้นกระทำหน้าที่เหมือนกับเกตเวย์ ถ้าค่าเป็น forwarding(1) แต่ถ้าไม่กระทำหน้าที่เหมือนเกตเวย์จะมีค่าเป็น not-forwarding(2)

ipDefaultTTL (INTEGER) : หมายถึง ค่าดีฟอลต์ที่แทรกในฟิลด์ TTL (Time-to-Live) ของไอพีเฮดเดอร์ของดาต้าแกรม

ipReasmTimeout (INTEGER) : หมายถึง ค่าเวลาสูงสุด (Timeout) มีค่าเป็นวินาที ที่รับเฟรกเมนต์เพื่อเข้าคิวรอในกระบวนการรีแอสเซมบลี

การนับเฟรกเมนต์เดชันของ ipReasmReqs, ipFragCreate, ipReasmOKs และ ipFragOKs ซึ่งถึงจำนวนของเฟรกเมนต์เดชันที่เกิดขึ้นมากเท่าไร ส่วน ipReasmFails และ ipFragFails ซึ่งถึงเฟรกเมนต์ส่วนที่เสีย ซึ่งบ่งบอกถึงประสิทธิภาพของเครือข่ายว่ามีมากเท่าไร ถ้าเฟรกเมนต์มีส่วนเสียมากอาจแก้ไขโดยอัปเดตระบบที่กระทำเฟรกเมนต์เดชันในส่วนของ MTU แต่ถ้าไม่สามารถอัปเดตระบบได้อาจจะลองทำการคอนฟิกค่าของ ipReasmTimeOut ให้มากขึ้นกว่าเดิม

ตารางไอพีแอดเดรส

ตารางไอพีแอดเดรส หรือ ipAddrTable จะเก็บการติดตาม (Track) ของไอพีแอดเดรสที่เกี่ยวข้องกับ โหนด

ตาราง ipAddrTable ใช้ ipAdEntAddr เป็นคีย์เพื่อระบุรูปแบบแถวในตาราง

INDEX {ipAdEntAddr }

สารสนเทศของตารางนี้มีประโยชน์ในด้านการมอนิเตอร์คอนฟิกูเรชัน

แถวแต่ละแถวของตารางประกอบด้วยออบเจกต์ ดังนี้

ipAdEntAddr (IpAddress) : หมายถึง ไอพีแอดเดรสของแถว

ipAdEntIfIndex (INTEGER) : หมายถึง ค่าหมายเลขอินเทอร์เฟซ ที่ระบุไอพีแอดเดรสของอินเทอร์เฟซนั้น

ipAdEntNetMask (IpAddress) : หมายถึง ซับเน็ตมาส์คของไอพีแอดเดรส มีค่า 1 สำหรับส่วนที่เป็น Network ID และค่า 0 สำหรับส่วนที่เป็น Host ID

ipAdEntBcastAddr (INTEGER) : หมายถึง ค่าของบิตที่มีนัยสำคัญน้อยที่สุดของไอพีบรอดคาสท์แอดเดรสที่ใช้ส่งดาต้าแกรมบนลอจิคัลอินเทอร์เฟซร่วมกับไอพีแอดเดรส

ipAdEntReasmMaxSize (INTEGER) : หมายถึง ขนาดที่ใหญ่ที่สุดของไอพีดาต้าแกรม ที่สามารถทำการรีแอสเซมบลีได้

ตารางไอพีเรดิง

ตารางไอพีเรดิง หรือ ipRouteTable จะเก็บการติดตาม (Track) ของเส้นทางไอพีเข้าไปในตารางไอพีเรดิง

ตาราง ipRouteTable ใช้ ipRouteDest เป็นคีย์เพื่อระบุรูปแบบแถวในตาราง

INDEX { ipRouteDest }

สารสนเทศของตารางนี้มีประโยชน์ในด้านการมอนิเตอร์คอนฟิกูเรชัน และการแยกแยะความผิดพลาดของเครือข่าย

แถวแต่ละแถวของตารางประกอบด้วยออบเจกต์ ดังนี้

ipRouteDest (IpAddress) : หมายถึง ไอพีแอดเดรสปลายทางของการเรา ค่า 0.0.0.0 คือ ดีฟอลต์เรา

ipRouteIfIndex (INTEGER) : หมายถึง ค่าอินเด็กซ์ที่ระบุ โคลออินเทอร์เฟซผ่านฮอปถัดไปของเส้นทาง

ipRouteMetric1 (INTEGER) : หมายถึง เรดิงเมตริกหลักสำหรับเส้นทางที่ 1 เมตริกถูกกำหนดโดยการเรียนรู้เส้นทางผ่านเรดิงโปรโตคอล ถ้าเมตริกไม่ใช่ค่าจะถูกเซตเป็น -1

ipRouteMetric2 (INTEGER) : หมายถึง เรดิงเมตริกทางเลือก สำหรับเส้นทางที่ 2 กำหนดโดยการเรียนรู้เส้นทางผ่านเรดิงโปรโตคอล ถ้าเมตริกไม่ใช่ค่าจะถูกเซตเป็น -1

ipRouteMetric3 (INTEGER) : หมายถึง เรดิงเมตริกทางเลือก สำหรับเส้นทางที่ 3 กำหนดโดยการเรียนรู้เส้นทางผ่านเรดิงโปรโตคอล ถ้าเมตริกไม่ใช่ค่าจะถูกเซตเป็น -1

ipRouteMetric4 (INTEGER) : หมายถึง เรดิงเมตริกทางเลือก สำหรับเส้นทางที่ 4 กำหนดโดยการเรียนรู้เส้นทางผ่านเรดิงโปรโตคอล ถ้าเมตริกไม่ใช่ค่าจะถูกเซตเป็น -1

ipRouteMetric5 (INTEGER) : หมายถึง เรดิงเมตริกทางเลือก สำหรับเส้นทางที่ 5 กำหนดโดยการเรียนรู้เส้นทางผ่านเรดิงโปรโตคอล ถ้าเมตริกไม่ใช่ค่าจะถูกเซตเป็น -1

ipRouteNextHop (IpAddress) : หมายถึง ไอพีแอดเดรสฮอปถัดไปของเส้นทาง

ipRouteType (enumerated INTEGER) : หมายถึง ชนิดของเส้นทางคือ other(1), invalid(2), direct(3), indirect(4)

ipRouteProto (enumerated INTEGER) : หมายถึง กลไกเรดิงผ่านเส้นทางการเรียนรู้

ipRouteAge (INTEGER) : หมายถึง ช่วงเวลามีค่าเป็นวินาที ตั้งแต่เส้นทางถูกอัปเดตหรือถูกตรวจทาน

ipRouteMask (IpAddress) : หมายถึง ชั้นเน็ตมาส์ของแอดเดรสปลายทาง เปรียบเทียบกับค่าในฟิลด์ ipRouteDest ถ้าระบบไม่สนับสนุนชั้นเน็ตมาส์ เอเจนต์จะสร้างค่า ipRouteMask โดยกำหนดค่าฟิลด์ ipRouteDest ตามคลาส A, B และ C

ipRouteInfo (OBJECT IDENTIFIER) : หมายถึง การอ้างอิงถึงเอ็มไอบีโมดูล เพื่อใช้ในการระบุไปยังเราติงโปรโตคอล

การสร้างเส้นทางใหม่ในตารางอาจเป็นการเซตเพื่อสร้างแถวใหม่ เรียกว่าสแตติกเรา หรือการเรียนรู้ผ่านเราติงโปรโตคอล แถวที่สร้างได้มาจากหมายเลขของ ipRouteProto ค่าที่กำหนดคือ other (1), local (2), netmgt (3), icmp (4), egp (5), ggp (6), hello (7), rip (8), is-is (9), es-is (10), ciscoigrp (11), bbnSpfIgp (12), ospf (13) และ bgp (14) การลบเส้นทางสามารถทำได้โดยการเซตค่า ipRouteType ให้เป็น อินแวลิด (Invalid) อย่างไรก็ตาม เส้นทางที่สร้างจากการเรียนรู้จะทำการสร้างเส้นทางอย่างทันทีทันใดเมื่อการเรียนรู้นั้นเหมือนกันกับเราติงโปรโตคอล ในกรณีนี้เอเจนต์เท่านั้นที่ขอมให้สแตติกเราถูกลบได้

ตารางไอพีแอดเดรสทรานสเลชัน

ตารางไอพีแอดเดรสทรานสเลชัน หรือ ipNetToMediaTable จะเก็บการติดตาม (Track) ของการแปลงค่า (Mapping) ของแอดเดรสเครือข่ายไปยังแอดเดรสกายภาพ ตารางนี้นำไปใช้แทนตาราง atTable ในกลุ่มเอทีทีจะถูกลบเลิกในเวอร์ชันถัดไป

ตาราง ipNetToMediaTable ใช้หมายเลขอินเทอร์เฟซและไอพีแอดเดรสของการแมปปิงเป็นครรชนีเพื่อระบุรูปแบบแถวในตาราง

INDEX { ipNetToMediaIfIndex , ipNetToMediaNetAddress }

แถวแต่ละแถวของตารางประกอบด้วยออบเจกต์ดังนี้

ipNetToMediaIfIndex (INTEGER) : หมายถึง หมายเลขอินเทอร์เฟซ ที่ถูกระบุโดยค่าของ ifIndex

ipNetToMediaPhysAddress (PhysAddress) : หมายถึง แอดเดรสกายภาพของการแมปปิง

ipNetToMediaNetAddress (IpAddress) : หมายถึง ไอพีแอดเดรสของการแมปปิง

ipNetToMediaType (INTEGER) : หมายถึง ชนิดของการแมปปิง คือ other(1), invalid(2), dynamic(3), static(4)

ตารางไอพีฟอร์เวิร์ดติง

ตารางไอพีฟอร์เวิร์ดติง [9] หรือ ipForwardTable ถูกกำหนดตามเอกสาร RFC 1354 ตารางนี้ได้ทำการปรับปรุงเพิ่มเติมรวมถึงแก้ไขอัลกอริทึมที่ดีกว่าตาราง ipRouteTable ดังนั้นตาราง ipForwardTable จึงถูกนำมาใช้แทนตาราง ipRouteTable ที่จะถูกลบเลิกการใช้ในเวอร์ชันต่อไป

คอลัมน์ต่าง ๆ ในตาราง โดยส่วนใหญ่จะเหมือนกับ ipRouteTable แต่จะมีความยืดหยุ่นมากกว่า และมีออบเจกต์เพิ่มเติมขึ้นมาบางส่วน

ตารางไอพีฟอร์เวิร์ดคิง บรรจุนออบเจกต์บนสุด 2 ระดับ คือ

ipForwardNumber (Gauge32) : หมายถึง จำนวนเอนทรีในตารางเราดิง

และตาราง ipForwardTable ซึ่งบรรจุสารสนเทศของการเรา ตาราง ipForwardTable ใช้ ipForwardDest, ipForwardProto, ipForwardPolicy, ipForwardNextHop เป็นคีย์เพื่อระบุรูปแบบแถวในตาราง ซึ่งจะเห็นว่าการใช้หลายเส้นทาง (Multiple Route) จะให้การจัดการที่ดีกว่า การใช้เส้นทางเดียว (Single Route)

INDEX { ipForwardDest, ipForwardProto, ipForwardPolicy, ipForwardNextHop }

แถวแต่ละแถวของตารางประกอบด้วยออบเจกต์ ดังนี้

ipForwardDest (IpAddress) : เหมือนกับ ipRouteDest ของตาราง ipRouteTable

ipForwardMask (IpAddress) : เหมือนกับ ipRouteMask ของตาราง ipRouteTable

ipForwardPolicy (INTEGER) : หมายถึง นโยบายที่เกี่ยวข้องกับการเลือกใช้เส้นทาง ในกรณีที่มีหลายเส้นทางในการไปถึงเป้าหมาย โดยดีฟอลต์การอ้างนโยบายจะอ้างถึงไอพีในฟิลด์ชนิดของบริการ (Type of Service) ซึ่งจะขึ้นถึงคุณภาพบริการที่ต้องการของดาต้าแกรม เช่น การทำก่อน (Precedence), การหน่วง, ปริมาณงาน (Throughput) และความน่าเชื่อถือ

ipForwardNextHop (IpAddress) : เหมือนกับ ipRouteNextHop ของตาราง ipRouteTable

ipForwardIfIndex (INTEGER) : เหมือนกับ ipRouteIndex ของตาราง ipRouteTable

ipForwardType (INTEGER) : เหมือนกับ ipRouteType ของตาราง ipRouteTable

ipForwardProto (INTEGER) : เหมือนกับ ipRouteProto ของตาราง ipRouteTable แต่จะมีการระบุเราดิงโปรโตคอลที่มากกว่า

ipForwardAge (INTEGER) : เหมือนกับ ipRouteAge ของตาราง ipRouteTable

ipForwardInfo (OBJECT IDENTIFIER) : เหมือนกับ ipRouteInfo ของตาราง ipRouteTable

ipForwardNextHopAS (INTEGER) : หมายถึง จำนวนระบบบอโตโนมัส (Autonomous System) ของฮอปถัดไป ซึ่งค่าเหล่านี้จะมีประโยชน์สำหรับผู้บริหารเครือข่ายภูมิภาค

ipForwardMetric1 (INTEGER) : เหมือนกับ ipRouteMetric1 ของตาราง ipRouteTable

ipForwardMetric2 (INTEGER) : เหมือนกับ ipRouteMetric2 ของตาราง ipRouteTable

ipForwardMetric3 (INTEGER) : เหมือนกับ ipRouteMetric3 ของตาราง ipRouteTable

ipForwardMetric4 (INTEGER) : เหมือนกับ ipRouteMetric4 ของตาราง ipRouteTable

ipForwardMetric5 (INTEGER) : เหมือนกับ ipRouteMetric5 ของตาราง ipRouteTable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 กลุ่มไอซีเอ็มพี (Internet Control Message Protocol Group)

กลุ่มไอซีเอ็มพี [7] กำหนดใน MIB-II ตามเอกสาร RFC 1213 กลุ่มนี้บรรจุสารสนเทศที่เกี่ยวข้องกับการนับจำนวน (Counter) ของการส่งและการรับไอซีเอ็มพีเมสเสจ

กลุ่มนี้ประกอบด้วยออบเจกต์ 26 ออบเจกต์ คือ

icmpInMsgs (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจทั้งหมด รวมถึงการรับไอซีเอ็มพีเมสเสจที่มีความผิดพลาดด้วย

icmpInErrors (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีความผิดพลาด เช่น ผลรวมตรวจสอบผิด หรือความยาวผิด

icmpInDestUnreachs (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่ไม่สามารถระบุปลายทางได้

icmpInTimeExcds (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีเวลามากเกิน

icmpInparmProbs (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่พารามิเตอร์มีปัญหา

icmpInSrcQuenchs (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่แหล่งต้นทางถูกระงับ (Source Quench)

icmpInRedirects (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการรีไคเร็ค

icmpInEchos (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการร้องขอการสะท้อนกลับ

icmpInEchoReps (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการตอบรับการสะท้อนกลับ (Echo-Reply)

icmpInTimestamps (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการร้องขอตราประทับเวลา หรือ ไทม์แสตมป์

icmpInTimestampReps (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการตอบรับตราประทับเวลา หรือ ไทม์แสตมป์

icmpInAddrMasks (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการร้องขอแอดเดรสมาสก์

icmpInAddrMaskReps (Counter) : หมายถึง จำนวนการรับไอซีเอ็มพีเมสเสจที่มีการตอบรับแอดเดรสมาสก์

icmpOutMsgs (Counter) : หมายถึง จำนวนการพยายามส่งไอซีเอ็มพีเมสเสจทั้งหมด รวมถึง icmpOutErrors

icmpOutErrors (Counter) : หมายถึง จำนวนการไม่ส่งไอซีเอ็มพีเมสเซจ เนื่องจากเกิดปัญหาภายในไอซีเอ็มพี เช่น บัฟเฟอร์ขาดแคลน ค่านี้จะไม่รวมถึงความผิดพลาดภายนอกชั้นไอซีเอ็มพี เช่น การเราไอพีค้ำแกรม

icmpOutDestUnreachs(Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่ไม่สามารถระบุปลายทางได้

icmpOutTimeExcds(Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีเวลามากเกิน

icmpOutParmProbs (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่พารามิเตอร์มีปัญหา

icmpOutSrcQuenchs(Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่แหล่งต้นทางถูกระงับ

icmpOutRedirects (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการรีไคเร็ค

icmpOutEchos (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการร้องขอการสะท้อนกลับ

icmpOutEchoReps (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการตอบรับการสะท้อนกลับ (Echo-Reply)

icmpOutTimestamps (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการร้องขอตราประทับเวลา หรือไทม์แสตมป์

icmpOutTimestampReps (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการตอบรับตราประทับเวลา หรือไทม์แสตมป์

icmpOutAddrMasks (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการร้องขอแอดเดรสมาสก์

icmpOutAddrMaskReps (Counter) : หมายถึง จำนวนการส่งไอซีเอ็มพีเมสเซจที่มีการตอบรับแอดเดรสมาสก์

ออบเจ็กต์กลุ่มไอซีเอ็มพีสามารถสรุปได้สั้น ๆ คือ

- ไอซีเอ็มพีเมสเซจแต่ละชนิดมีการนับอยู่ 2 แบบคือการนับจำนวนของเมสเซจที่ถูกสร้างโดยโลคอลไอซีเอ็มพีเอนทิตี และการนับจำนวนของเมสเซจที่ถูกรับโดยโลคอลไอซีเอ็มพีเอนทิตี
- การนับที่เกี่ยวข้องกับการเก็บแทรค (Track) จำนวนทั้งหมดของการรับ, การส่ง, การรับความผิดพลาด หรือการไม่ส่งเนื่องจากมีความผิดพลาดของไอซีเอ็มพีเมสเซจ

บทที่ 4

การจัดการข่ายคอมพิวเตอร์เฉพาะบริเวณ (แลน)

บทนี้กล่าวถึงการจัดการข่ายคอมพิวเตอร์เฉพาะบริเวณ (Local Area Network) หรือแลน ครอบคลุมถึงมาตรฐานแลนที่ได้รับความนิยมในปัจจุบันคือ IEEE 802.3 หรืออีเทอร์เน็ต การจัดการสารสนเทศครอบคลุมถึงอุปกรณ์ที่ใช้บนแลน คือ อุปกรณ์บริดจ์ที่ระดับชั้น MAC (Media Access Control), รีพีตเตอร์ ที่ระดับชั้นกายภาพ และอุปกรณ์ประเภท MAU (Medium Attachment Unit)

4.1 อีเทอร์เน็ต-ไลค์ แลน (Ethernet-Like LAN)

อีเทอร์เน็ต-ไลค์ [10] เป็นโพรโตคอลที่ใช้ควบคุมการเข้าถึงสื่อแบบ CSMA/CD (Carrier Sense Multiple Access/Collision Detection) สื่อของอีเทอร์เน็ต-ไลค์ ถูกแยกแยะโดยค่า 3 ค่าของ ออบเจกต์ ifType คือ ethernet-csmacd (6) , iso88023-csmacd (7) และ starLan (11) กลุ่มนี้กำหนดตามเอกสาร RFC 1398 มีค่า OBJECT IDENTIFIER อยู่ที่ {transmission 7} ออบเจกต์ทั้งหมดในกลุ่มเริ่มต้นด้วย dot3 การอ้างอิงสื่ออยู่บนพื้นฐานตามมาตรฐานของ IEEE 802.3

กลุ่มนี้ประกอบด้วยกลุ่มสถิติ (Statistics) ให้สารสนเทศทางสถิติโดยทั่วไป และกลุ่มสถิติการชนกัน (Collision Statistics) ให้รายละเอียดสารสนเทศการชนกันของข้อมูล

กลุ่มสถิติประกอบด้วย 1 ตาราง คือตาราง dot3StatsTable มี dot3StatsEntry เป็นรูปแบบแถว มี dot3StatsIndex เป็นคอลัมน์

```
INDEX { dot3StatsIndex }
```

ตารางประกอบด้วยออบเจกต์ที่เกี่ยวข้องกับการนับ 12 ออบเจกต์ 7 ออบเจกต์ใช้เกี่ยวกับการส่งเฟรม, 4 ออบเจกต์ใช้เกี่ยวกับการรับเฟรม และ 1 ออบเจกต์สำหรับ SQE Test

กลุ่มออบเจกต์ที่เกี่ยวข้องกับการส่งคือ

dot3StatsSingleCollisionFrames (Counter) : หมายถึง จำนวนเฟรมที่ส่งเป็นผลสำเร็จบนอินเทอร์เฟซ และจะหยุดนับเมื่อเกิดการชนกันหนึ่งเฟรม

dot3StatsMultipleCollisionFrames (Counter) : หมายถึง จำนวนเฟรมที่ส่งเป็นผลสำเร็จบนอินเทอร์เฟซ และจะหยุดนับเมื่อเกิดการชนกันมากกว่าหนึ่งเฟรม

dot3StatsDeferredTransmissions (Counter) : หมายถึง จำนวนเฟรมที่พยายามส่งเป็นอันดับแรกบนอินเทอร์เฟซเกิดการดีเลย์ อาจเนื่องจากสื่อไม่ว่าง

dot3StatsLateCollisions (Counter) : หมายถึง ช่วงเวลาที่ตรวจสอบพบการชนกันบนอินเทอร์เฟซ ที่ช้ากว่า 512 บิตไทม์ หรือ 51.2 ms. บนระบบ 10 Mbit/s

dot3StatsExcessiveCollisions (Counter) : หมายถึง จำนวนเฟรมที่ส่งบนอินเทอร์เฟซเกิดล้มเหลว อาจเนื่องมาจากเกิดการชนกันมากเกินไป

dot3StatsInternalMacTransmitErrors (Counter) : หมายถึง จำนวนเฟรมที่ส่งบนอินเทอร์เฟซเกิดล้มเหลว อาจเนื่องมาจากเกิดความผิดพลาดภายใน MAC (Media Access Control)

dot3StatsCarrierSenseErrors (Counter) : หมายถึง ช่วงเวลาในการฟังว่าสัญญาณว่างหรือไม่ (Carrier Sense) บนเงื่อนไขการสูญเสียหรือการไม่ยืนยันในการส่งเฟรมบนอินเทอร์เฟซ กลุ่มออบเจกต์ที่เกี่ยวข้องกับการรับคือ

dot3StatsAlignmentErrors (Counter) : หมายถึง จำนวนเฟรมที่รับบนอินเทอร์เฟซ มีรูปแบบเฟรมที่ใช้ไม่ได้และไม่ผ่านการตรวจสอบ FCS (Frame Check Sequence)

dot3StatsFCSErrors (Counter) : หมายถึง จำนวนเฟรมที่รับบนอินเทอร์เฟซ มีรูปแบบเฟรมที่ใช้ได้แต่ไม่ผ่านการตรวจสอบ FCS

dot3StatsFrameTooLongs (Counter) : หมายถึง จำนวนเฟรมที่รับบนอินเทอร์เฟซ มีขนาดเฟรมมากเกินไปกว่าขนาดที่อนุญาตสูงสุด

dot3StatsInternalMacReceiveErrors (Counter) : หมายถึง จำนวนการรับเฟรมล้มเหลว อาจเนื่องมาจากเกิดความผิดพลาดภายใน MAC

ออบเจกต์สุดท้ายที่เกี่ยวข้องกับการนับคือ

dot3StatsSQETestErrors (Counter) : หมายถึง จำนวนความผิดพลาด SQE Test (Signal Quality Error) ที่ถูกตรวจพบบนอินเทอร์เฟซ

กลุ่มสถิติการชนกัน (Collision Statistics) ของอีเทอร์เน็ต-10 ประกอบด้วย 1 ตาราง คือ ตาราง dot3CollTable แต่ละอินเทอร์เฟซถูกแทนด้วยฮิสโตแกรมที่ชนกัน แต่ละฮิสโตแกรมมี 16 เซลล์ แต่ละเซลล์ถูกแทนด้วยจำนวนการชนกัน (ต่อเฟรม) แต่ละแถวในตารางแทน 1 เซลล์ในฮิสโตแกรม

ตารางนี้มี dot3CollEntry เป็นรูปแบบแถว มี dot3CollIndex, dot3CollCount เป็นคอลัมน์

INDEX { dot3CollIndex, dot3CollCount }

ภายในตารางประกอบด้วยออบเจกต์คือ

dot3CollIndex (INTEGER) หมายถึง ค่าอินเด็กซ์ที่ระบุอินเทอร์เฟซที่เกี่ยวข้องกับฮิสโตแกรมที่ชนกัน ค่าอินเด็กซ์ที่ระบุนี้เหมือนกับค่าของ ifIndex

dot3CollCount (INTEGER) หมายถึง จำนวนการชนกันต่อเฟรมของเซลล์ในฮิสโตแกรม ถูกแทนด้วยความถี่ มีค่าอยู่ระหว่าง 1 - 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dot3CollFrequencies (Counter) : หมายถึง จำนวนเฟรมที่ส่งผ่านบนอินเทอร์เฟซ ที่มีจำนวนการชนกันต่อเฟรมส่งมาด้วย

การนับเฟรมที่มีการชนกัน 16 จะเทียบเท่ากับ dot3StatsExcessiveCollisions การนับเฟรมที่มีการชนกัน 1 จะเทียบเท่ากับ dot3StatsSingleCollisionFrames และผลรวมขององค์ประกอบที่เหลืออยู่ในฮิสโตแกรม จะเทียบเท่ากับ dot3StatsMultipleCollisionFrames

4.2 IEEE 802.3 รีพีตเตอร์

เอ็มไอบีกำหนด IEEE 802.3 10 megabit/second baseband repeater [11] หรือที่เรียกว่า “ฮับ” เพื่อใช้จัดการกับอุปกรณ์ประเภทรีพีตเตอร์ กลุ่มนี้ถูกกำหนดตามเอกสาร RFC 1516 มีค่า OBJECT IDENTIFIER อยู่ที่ {mib-2 22}

กลุ่มรีพีตเตอร์เอ็มไอบี ประกอบด้วยเอ็มไอบี 3 กลุ่มหลักคือ

กลุ่มพื้นฐาน (Basic Group) : เป็นกลุ่ม Mandatory บรรจุนโยบายที่เกี่ยวกับสถานะ, พารามิเตอร์ และควบคุม ใช้ได้กับรีพีตเตอร์ทั้งหมด, กลุ่มพอร์ตภายในรีพีตเตอร์ และพอร์ตเฉพาะ

กลุ่มมอนิเตอร์ (Monitor Group) : เป็นกลุ่ม Optional บรรจุนโยบายที่เกี่ยวกับสถิติการมอนิเตอร์ ใช้ได้กับรีพีตเตอร์ทั้งหมด, กลุ่มพอร์ตเฉพาะ และพอร์ตเฉพาะ

กลุ่มติดตามแอดเดรส (Address Tracking Group) : เป็นกลุ่ม Optional บรรจุนโยบายที่เกี่ยวกับการติดตาม MAC แอดเดรสของอุปกรณ์รับส่งข้อมูลปลายทาง (DTE) ที่ถูกเชื่อมต่อกับพอร์ตของรีพีตเตอร์

กลุ่มพื้นฐาน (Basic Group)

กลุ่มนี้ประกอบด้วย 6 สกาลาร์ 2 ตาราง คือตารางกลุ่มพอร์ต และตารางพอร์ต

สกาลาร์ออบเจกต์ประกอบด้วย

rptrGroupCapacity (INTEGER) : หมายถึง จำนวนสูงสุดของกลุ่มพอร์ตที่สามารถบรรจุอยู่ในรีพีตเตอร์ มีค่าระหว่าง 1 ถึง 1024 กลุ่มพอร์ตบางกลุ่มอาจจะไม่ถูกแสดง ในกรณีนี้จำนวนกลุ่มพอร์ตที่ถูกแสดงจะน้อยกว่า rptrGroupCapacity แต่จะไม่มากกว่า rptrGroupCapacity

rptrOperStatus (enumerated INTEGER) : หมายถึง สถานะการทำงานของรีพีตเตอร์ในปัจจุบัน มีค่าเป็นเลขจำนวนเต็มคือ

other (1) : ซึ่งถึงสถานะที่ไม่ถูกกำหนดหรือไม่รู้

ok (2) : ซึ่งถึงข้อขัดข้องที่ไม่รู้

rptrFailure (3) : ซึ่งถึงข้อขัดข้องที่เกี่ยวข้องกับรีพีตเตอร์

groupFailure (4) : ซึ่งถึงข้อขัดข้องที่เกี่ยวข้องกับกลุ่มพอร์ต

portFailure (5) : ซึ่งถึงข้อขัดข้องที่เกี่ยวข้องกับพอร์ต

generalFailure (6) : ซึ่งถึงข้อขัดข้องที่ไม่ระบุชนิด

rptrHealthText (DisplayString) : หมายถึง ข้อความที่บรรยายรายละเอียดถึงสภาวะการทำงานของรีพีตเตอร์

rptrReset (enumerated INTEGER) : หมายถึง ออบเจ็กต์ทำการ reset(2) กรณีที่รีพีตเตอร์เกิดรีเซตในสภาวะเริ่มต้นด้วยตัวของมันเอง (self-test) ส่วนค่า noReset(1) จะไม่ส่งผลใด ๆ เพราะเอเจนต์จะรีเทรินค่า noReset(1) เสมอเมื่อทำการอ่านออบเจ็กต์

rptrNonDisruptTest (enumerated INTEGER) : หมายถึง ออบเจ็กต์อาจทำการ selfTest(2) ในกรณีที่รีพีตเตอร์กระทำการทดสอบตัวเองแบบไม่ทำลาย (non- disruptive self-test) ส่วนค่าของ noSelfTest(1) จะไม่ส่งผลใด ๆ เพราะเอเจนต์จะรีเทรินค่า noSelfTest(1) เสมอเมื่ออ่านออบเจ็กต์

rptrTotalPartitionedPorts (Gauge32) : หมายถึง จำนวนพอร์ตทั้งหมดในรีพีตเตอร์ ที่มีออบเจ็กต์อยู่ในสภาวะปัจจุบัน คำนี้นี้ rptrPortOperStatus ไม่มีค่า notPresent, rptrPortAdminStatus มีค่า enabled , และ rptrPortAutoPartitionState มีค่า autoPartitioned

ตารางกลุ่มพอร์ตหรือ rptrGroupTable มี rptrGroupEntry เป็นรูปแบบแถวมี rptrGroupIndex เป็นคอลัมน์

INDEX { rptrGroupIndex }

ภายในตารางประกอบด้วยออบเจ็กต์คือ

rptrGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มภายในรีพีตเตอร์

rptrGroupDescr (DisplayString) : หมายถึง ข้อความบรรยายรายละเอียดของกลุ่มพอร์ต รวมถึงชื่อเต็มและการระบุเวอร์ชันของกลุ่ม เช่น Barney Rubble 10BASE-T 4-port SIMM socket Version 2.1

rptrGroupObjectID (OBJECT IDENTIFIER) : หมายถึง การกำหนดชื่ออย่างเป็นทางการของชนิดกลุ่มพอร์ตของผู้ผลิต คำนี้นี้จะถูกจัดสรรอยู่ในอินเทอร์เน็ตโพรโทคอล { 1.3.6.1.4.1 }

rptrGroupOperStatus (enumerated INTEGER) : หมายถึง สถานะการทำงานของกลุ่มพอร์ตมีค่าคือ other(1), operational(2), malfunctioning(3), notPresent(4), underTest(5), resetInProgress(6)

rptrGroupLastOperStatusChange (TimeTicks) : หมายถึง ค่าเวลาของ sysUpTime เมื่อค่าของ rptrGroupOperStatus ถูกเปลี่ยนแปลงครั้งสุดท้าย ถ้าค่านี้เป็นศูนย์จะชี้ถึงสถานะการทำงานของกลุ่มที่ไม่เปลี่ยนแปลงนับแต่เอเจนต์ถูกเริ่มทำงานใหม่ครั้งสุดท้าย

rptrGroupPortCapacity (INTEGER) : หมายถึง จำนวนพอร์ตที่บรรจุภายในกลุ่มพอร์ต มีค่าระหว่าง 1 ถึง 1024

ตารางพอร์ตหรือ rptrPortTable มี rptrPortEntry เป็นรูปแบบแถวมี rptrPortGroupIndex, rptrPortIndex เป็นคอลัมน์

INDEX { rptrPortGroupIndex, rptrPortIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

rptrPortGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มพอร์ตที่บรรจุพอร์ต

rptrPortIndex (INTEGER) : หมายถึง การระบุพอร์ตภายในกลุ่มพอร์ต

rptrPortAdminStatus (enumerated INTEGER) : หมายถึง ออบเจกต์ทำการเซตค่าพอร์ต เป็น enabled(1) หรือ disabld(2) ถ้าพอร์ตดีสเอเบิลค่า rptrPortAutoPartitionState จะหยุดจนกระทั่ง พอร์ตถูกอีนาเบิล เมื่อพอร์ตอีนาเบิล rptrPortAutoPartitionState จะเซตเป็น notAutoPartitioned(1)

rptrPortAutoPartitionState (enumerated INTEGER) : หมายถึง พอร์ตถูกพาร์ติชันใช้อยู่ ในปัจจุบันมีค่าคือ notAutoPartitioned(1) และ autoPartitioned(2)

rptrPortOperStatus (enumerated INTEGER) : หมายถึง สถานะการทำงานของพอร์ตคือ operational(1), notOperational(2) และ notPresent(3)

กลุ่มมอนิเตอร์ (Monitor Group)

กลุ่มมอนิเตอร์ประกอบด้วยเคาน์เตอร์ที่ใช้ได้กับรีพีตเตอร์ทั้งหมด, รีพีตเตอร์แต่ละพอร์ต และรีพีตเตอร์แต่ละกลุ่มพอร์ต ให้สถิติด้านประสิทธิภาพและความผิดพลาดของอุปกรณ์ที่เชื่อมต่อ ไปยังรีพีตเตอร์

เคาน์เตอร์ที่สามารถใช้ได้กับรีพีตเตอร์ทั้งหมดคือ

rptrMonitorTransmitCollisions (Counter) : หมายถึง จำนวนการชนกันที่เกิดขึ้นภายใน รีพีตเตอร์ เนื่องจากเกิดสภาวะการชนกันสองหรือมากกว่าเข้ามายังพอร์ตพร้อมกัน เวลาต่ำสุดโดย ประมาณสำหรับ ไรลด์โอเวอร์ของเคาน์เตอร์คือ 16 ชั่วโมง

ตารางพอร์ตหรือ rptrMonitorPortGroupTable มี rptrMonitorPortEntry เป็นรูปแบบแถว มี rptrMonitorPortGroupIndex, rptrMonitorPortIndex เป็นคอลัมน์

INDEX { rptrMonitorPortGroupIndex, rptrMonitorPortIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

rptrMonitorPortGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มที่บรรจุพอร์ต

rptrMonitorPortIndex (INTEGER) : หมายถึง การระบุพอร์ตภายในกลุ่ม

rptrMonitorPortReadableFrames (Counter) : หมายถึง จำนวนการรับเฟรมที่ใช้ได้บน พอร์ต เวลาต่ำสุดโดยประมาณสำหรับ ไรลด์โอเวอร์ของเคาน์เตอร์คือ 80 ชั่วโมง

rptrMonitorPortReadableOctets(Counter) : หมายถึง จำนวนการรับออกเททที่ใช้ได้บน พอร์ต เวลาต่ำสุดโดยประมาณสำหรับ ไรลด์โอเวอร์ของเคาน์เตอร์คือ 58 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

`rptrMonitorPortFCSErrors` (Counter) : หมายถึง จำนวนเฟรมที่รับบนพอร์ต ซึ่งมีความผิดพลาดของ FCS แต่ไม่มีความผิดพลาดของเฟรม เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 80 ชั่วโมง

`rptrMonitorPortAlignmentErrors` (Counter) : หมายถึง จำนวนเฟรมที่รับบนพอร์ต ซึ่งมีความผิดพลาดของ FCS และความผิดพลาดของเฟรม เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 80 ชั่วโมง

`rptrMonitorPortFrameTooLongs` (Counter) : หมายถึง จำนวนเฟรมที่รับบนพอร์ต ซึ่งมีความยาวมากกว่าขนาดสูงสุดของเฟรม เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 61 วัน

`rptrMonitorPortRunts` (Counter) : หมายถึง จำนวนการรับ Runts บนพอร์ต (Runts คือ การชนกันของเฟรมเมนต์ ซึ่งเป็นเหตุการณ์ปกติของเครือข่ายไม่ใช่ความผิดพลาด) เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 16 ชั่วโมง

`rptrMonitorPortShortEvents` (Counter) : หมายถึง จำนวนเฟรมที่รับสั้นกว่า Runts ซึ่งการสั้นกว่านี้ชี้ถึงการชนกันของเฟรมเมนต์ เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 16 ชั่วโมง

`rptrMonitorPortCollisions` (Counter) : หมายถึง จำนวนการชนกันที่เกิดขึ้นในรีพีตเตอร์ เนื่องจากมีกิจกรรมเข้ามาบนพอร์ตพร้อมกันมากกว่าหนึ่ง เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 16 ชั่วโมง

`rptrMonitorPortLateEvents` (Counter) : หมายถึง จำนวนการชนกันที่ถูกตรวจพบหลังการรับเฟรมที่มีจำนวนน้อยที่สุดของออกเทท เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์เคาน์เตอร์คือ 81 ชั่วโมง

`rptrMonitorPortVeryLongEvents` (Counter) : หมายถึง จำนวนเฟรมซึ่งมีเวลาที่ยาวกว่าเวลาไทม์เอาต์ของ MAU Jabber Lockup Protection

`rptrMonitorPortDataRateMismatches` : (Counter) หมายถึง จำนวนเฟรมที่รับบนพอร์ต ซึ่งเป็นเฟรมที่ใช้ได้ แต่อัตราข้อมูลไม่ตรงกันกับการส่งที่โลกอล

`rptrMonitorPortAutoPartitions` (Counter) : หมายถึง ช่วงเวลาที่พอร์ตถูกพาร์ติชันอย่างอัตโนมัติ

`rptrMonitorPortTotalErrors` (Counter): หมายถึง จำนวนความผิดพลาดทั้งหมดที่เกิดขึ้นบนพอร์ต ค่านี้จะเป็นผลรวมของค่าความผิดพลาดจากเคาน์เตอร์อื่นบนพอร์ตเดียวกัน ดังต่อไปนี้

`rptrMonitorPortFCSErrors`, `rptrMonitorPortAlignmentErrors`, `rptrMonitorPortFrameTooLongs`, `rptrMonitorPortShortEvents`, `rptrMonitorPortLateEvents`, `rptrMonitorPortVeryLongEvents`, `rptrMonitorPortDataRateMismatches`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางกลุ่มพอร์ตหรือ rptrMonitorGroupTable มี rptrMonitorGroupEntry เป็นรูปแบบแถวมี rptrMonitorGroupIndex เป็นดัชนี

INDEX { rptrMonitorGroupIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

rptrMonitorGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มภายในรีพีตเตอร์

rptrMonitorGroupTotalFrames (Counter) : หมายถึง จำนวนการรับเฟรมที่ใช้ได้ทั้งหมดบนพอร์ตในกลุ่มพอร์ต เคาน์เตอร์จะทำการรวมค่าของ rptrMonitorPortReadableFrames ทุกพอร์ตในกลุ่มพอร์ต เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 80 ชั่วโมง

rptrMonitorGroupTotalOctets(Counter) : หมายถึง จำนวนการรับออกเททที่ใช้ได้ทั้งหมดบนพอร์ตในกลุ่มพอร์ต เคาน์เตอร์จะทำการรวมค่าของ rptrMonitorPortReadableOctets ทุกพอร์ตในกลุ่มพอร์ต เวลาต่ำสุดโดยประมาณสำหรับโรลล์โอเวอร์ของเคาน์เตอร์คือ 58 นาที

rptrMonitorGroupTotalErrors (Counter) : หมายถึง จำนวนความผิดพลาดทั้งหมดที่เกิดขึ้นกับทุกพอร์ตในกลุ่มพอร์ต เคาน์เตอร์จะทำการรวมค่าของ rptrMonitorPortTotalErrors ทุกพอร์ตในกลุ่มพอร์ต

กลุ่มติดตามแอดเดรส (Address Tracking Group)

กลุ่มนี้จัดเตรียมความน่าจะเป็นของอุปกรณ์ที่แอกทีฟอยู่บนพอร์ตของรีพีตเตอร์ ซึ่งจะทำการเก็บแอดเดรสที่ใหม่และทันสมัยมากที่สุด เพื่อนำมาสร้างรายการของอุปกรณ์ที่แอกทีฟอยู่บนพอร์ตได้

กลุ่มนี้ประกอบด้วย 1 ตารางคือ rptrAddrTrackPortTable มี rptrAddrTrackEntry เป็นรูปแบบแถวมี rptrAddrTrackGroupIndex, rptrAddrTrackPortIndex เป็นดัชนี

INDEX { rptrAddrTrackGroupIndex, rptrAddrTrackPortIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

rptrAddrTrackGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มที่บรรจุพอร์ต

rptrAddrTrackPortIndex (INTEGER) : หมายถึง การระบุพอร์ตภายในกลุ่ม

rptrAddrTrackLastSourceAddress (MacAddress) : หมายถึง แหล่ง MAC แอดเดรสที่รับเฟรมการอ่านครั้งสุดท้ายของพอร์ต ออบเจกต์นี้มีสถานะ Deprecated เพราะค่าจะไม่ถูกกำหนดเมื่อไม่มีเฟรมให้เห็นบนพอร์ต การแทนที่ของออบเจกต์นี้คือ rptrAddrTrackNewLastSrcAddress

rptrAddrTrackSourceAddrChanges (Counter32) : หมายถึง ช่วงเวลาที่ออบเจกต์ rptrAddrTrackLastSourceAddress หรือ rptrAddrTrackNewLastSrcAddress ของพอร์ตมีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rpMauGroupIndex, rpMauPortIndex, rpMauIndex เป็นครรขนี้

4.3 IEEE 802.3 Medium Attachment Units (MAUs)

เอ็มไอบีกำหนด IEEE 802.3 Medium Attachment Units (MAUs) [12] ให้สามารถเชื่อมต่อไปยัง IEEE 802.3 รีพีตเตอร์ หรือ 802.3 อินเทอร์เฟซ เรียกว่า รีพีตเตอร์ MAU หรือ อินเทอร์เฟซ MAU กลุ่มนี้ถูกกำหนดตามเอกสาร RFC 1515 มีค่า OBJECT IDENTIFIER อยู่ที่ {mib-2 26}

กลุ่มนี้ประกอบด้วยเอ็มไอบี 3 กลุ่มแต่ละกลุ่มมี 1 ตารางคือ

กลุ่มรีพีตเตอร์ MAU : กลุ่มนี้บรรจุออบเจกต์ที่เกี่ยวข้องกับคอนฟิกูเรชัน, สถานะ และการควบคุม นำไปใช้สำหรับ MAUs ที่เชื่อมต่อไปยังรีพีตเตอร์

กลุ่มอินเทอร์เฟซ MAU : กลุ่มนี้บรรจุออบเจกต์ที่เกี่ยวข้องกับคอนฟิกูเรชัน, สถานะ และการควบคุม นำไปใช้สำหรับ MAUs ที่เชื่อมต่อไปยังอินเทอร์เฟซ

กลุ่มบรอดแบนด์ MAU : กลุ่มนี้บรรจุออบเจกต์ที่เกี่ยวข้องกับคอนฟิกูเรชัน นำไปใช้สำหรับ 10BROAD36 MAUs

กลุ่มรีพีตเตอร์ MAU

กลุ่มนี้ประกอบด้วย 1 ตารางคือ rpMauTable มี rpMauEntry เป็นรูปแบบแถวมี rpMauGroupIndex, rpMauPortIndex, rpMauIndex เป็นครรขนี้

INDEX { rpMauGroupIndex, rpMauPortIndex, rpMauIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

rpMauGroupIndex (INTEGER) : หมายถึง การระบุกลุ่มรีพีตเตอร์ที่บรรจุพอร์ต

rpMauPortIndex (INTEGER) : หมายถึง การระบุพอร์ตรีพีตเตอร์ภายในกลุ่ม

rpMauIndex (INTEGER) : หมายถึง การระบุ MAU ที่เชื่อมพอร์ตรีพีตเตอร์ภายในกลุ่ม

rpMauType (OBJECT IDENTIFIER) : หมายถึง การระบุชนิด MAU บนพื้นฐานของ 10

Mb/s baseband MAU

rpMauStatus (INTEGER) : หมายถึง สภาวะปัจจุบันของ MAU มีค่าคือ

other(1) : ชี้ถึงการส่งค่ากลับ (Return) ถ้า MAU อยู่ในสภาวะอื่น

unknown(2) : ชี้ถึงการส่งค่ากลับ (Return) เมื่อไม่รู้สภาวะจริงของ MAU

operational(3) : ชี้ถึงสภาวะการทำงานที่สมบูรณ์ในการเชื่อมต่อกับพอร์ตของ

DTE หรือ รีพีตเตอร์

standby(4) : ชี้ถึงสภาวะว่างที่พร้อมจะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

shutdown(5) : ซึ่งถึงสภาวะหยุดทำงาน อาจเป็นการปิดเครื่อง

reset(6) : ซึ่งถึงสภาวะการเริ่มต้นใหม่ (Reset)

rpMauMediaAvailable (INTEGER) : หมายถึง การซึ่งถึงสภาวะการเข้าถึงสื่อของ MAU

other(1) : ซึ่งถึงการส่งค่ากลับ (Return) ถ้า MAU อยู่ในสภาวะอื่น

unknown(2) : ซึ่งถึงการส่งค่ากลับ (Return) เมื่อไม่รู้สภาวะจริงของ MAU

available(3) : ซึ่งถึงสภาวะปกติของการเชื่อมต่อ, แสง และการรูปแบ็ก

notAvailable(4) : ซึ่งถึงปัญหาของการสูญเสียการเชื่อมต่อ, แสงอ่อน และการ

รูปแบ็ก

remoteFault(5) : ซึ่งถึงการตรวจสอบพบข้อบกพร่องของการเชื่อมต่อที่อุปกรณ์

รีโมท

invalidSignal(6) : ซึ่งถึงการรับสัญญาณจากอุปกรณ์ปลายทางอื่นใช้ไม่ได้

rpMauMediaAvailableStateExits (Counter32) : หมายถึง ช่วงเวลาของ MAU ที่ rpMauMediaAvailable เปลี่ยนไปยังสภาวะ available(3)

rpMauJabberState (INTEGER) : หมายถึง การซึ่งถึงสภาวะแจ๊พเพอร์ของ MAU

other(1) : ซึ่งถึงการส่งค่ากลับ (Return) ถ้าแจ๊พเพอร์อยู่ในสภาวะอื่น ตามปกติ

ใช้สำหรับ MAU ชนิด AUI

unknown(2) : ซึ่งถึงการส่งค่ากลับ (Return) เมื่อไม่รู้สภาวะจริงของ MAU

noJabber(3) : ซึ่งถึงสภาวะปกติ

jabbering(4) : ซึ่งถึงสภาวะผิดพลาด

rpMauJabberingStateEnters (Counter32) : หมายถึง ช่วงเวลาของ rpMauJabberState ที่เข้าสู่สภาวะ jabbering(4)

กลุ่มอินเทอร์เฟซ MAU

ประกอบด้วย 1 ตารางคือ ifMauTable มี ifMauEntry เป็นรูปแบบแถวมี ifMauIfIndex,

ifMauIndex เป็นดรรชนี

INDEX { ifMauIfIndex, ifMauIndex }

ความหมายของออบเจ็กต์ในตารางนี้จะเหมือนกับตาราง rpMauTable โดยการแทนที่จากรีพีตเตอร์ MAU เป็นอินเทอร์เฟซ MAU

ภายในตารางประกอบด้วยออบเจ็กต์คือ

ifMauIfIndex : เหมือนกับค่าของ rpMauPortIndex ในตาราง rpMauTable

ifMauIndex : เหมือนกับค่าของ rpMauIndex ในตาราง rpMauTable

ifMauType : เหมือนกับค่าของ rpMauType ในตาราง rpMauTable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ifMauStatus : เหมือนกับค่าของ rpMauStatus ในตาราง rpMauTable

ifMauMediaAvailable : เหมือนกับค่าของ rpMauMediaAvailable ในตาราง rpMauTable

ifMauMediaAvailableStateExits : เหมือนกับค่าของ rpMauMediaAvailableStateExits ในตาราง rpMauTable

ifMauJabberState : เหมือนกับค่าของ rpMauJabberState ในตาราง rpMauTable

ifMauJabberingStateEnters : เหมือนกับค่าของ rpMauJabberingStateEnters ในตาราง rpMauTable

กลุ่มบรอดแบนด์ MAU

IEEE ไม่มีการกำหนดมาตรฐานสำหรับกลุ่มบรอดแบนด์รีพีตเตอร์โดยตรง แต่กลุ่มนี้ถูกประยุกต์ขึ้นมาจาก 10BROAD36 MAUs ที่ใช้เชื่อมต่อไปยังอินเทอร์เน็ตเฟซของโหนด และเป็นส่วนเสริมให้กับกลุ่มอินเทอร์เน็ตเฟซ MAU

กลุ่มนี้ประกอบด้วย 1 ตาราง คือ broadMauBasicTable มี broadMauBasicEntry เป็นรูปแบบแถวมี broadMauIfIndex, broadMauIndex เป็นคอลัมน์

INDEX { broadMauIfIndex, broadMauIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

broadMauIfIndex (INTEGER) : หมายถึง การระบุอินเทอร์เน็ตเฟซของ MAU

broadMauIndex (INTEGER) : หมายถึง การระบุ MAU ที่เชื่อมต่อไปยังอินเทอร์เน็ตเฟซ

broadMauXmtRcvSplitType (INTEGER) : หมายถึง ชนิดความถี่ของระบบมัลติเพล็กซ์ หรือเคเบิลที่ใช้แยกการส่งและการรับ สำหรับ 10BROAD36 MAUs มีค่าคือ

other(1) : ซึ่งถึงการส่งค่ากลับ (Return) ถ้าการแยกชนิดไม่ใช่เดี่ยวและคู่

single(2) : ซึ่งถึงระบบเคเบิลเดี่ยว

dual(3) : ซึ่งถึงระบบเคเบิลคู่ ตามปกติออฟเซตจะมีค่าเป็นศูนย์

broadMauXmtCarrierFreq (INTEGER) : หมายถึง ความถี่คลื่นพาห์ที่ใช้ในการส่งของ 10BROAD36 MAUs ที่ MHz./4 คือ 250 KHz.

BroadMauTranslationFreq (INTEGER): หมายถึง ความถี่ออฟเซตที่ใช้ในกระบวนการแปลงของ 10BROAD36 MAUs ที่ MHz./4 คือ 250 KHz.

4.4 บริดจ์ (Bridges)

บริดจ์ [13] เป็นอุปกรณ์ที่ใช้เชื่อมต่อแลนเซกเมนต์เข้าด้วยกัน บริดจ์จะทำงานที่ชั้นต่ำกว่าชั้นเครือข่าย การกำหนดออบเจกต์สำหรับบริดจ์ตั้งอยู่บนพื้นฐานของ IEEE 802.1d กลุ่มนี้กำหนดตามเอกสาร RFC 1493 มี OBJECT IDENTIFIER อยู่ที่ {mib-2 17} ประกอบด้วย 5 กลุ่มคือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มพื้นฐาน (Base Group): เป็นกลุ่ม Mandatory บรรจอบอเบ็จ็ค้พื้นฐานที่ใช้ได้กับบริดจ์ทุกชนิด

กลุ่ม STP (Spanning Tree Protocol Group) : กลุ่มออบเบ็จ็ค้ที่แสดงถึงสถานะของสแปนนิงทรีโพรโตคอล

กลุ่ม TP (Transparent Group) : กลุ่มออบเบ็จ็ค้ที่แสดงถึงสถานะของทรานส์พาเรนบริดจ์

กลุ่มสแตติก(Static Group): กลุ่มออบเบ็จ็ค้ที่แสดงถึงสถานะการกรองแอดเดรสปลายทาง

กลุ่ม SR (Source Routing Group) : กลุ่มออบเบ็จ็ค้ที่แสดงถึงสถานะของซอร์สเราดิง ราย

ละเอียดของเอกสารกลุ่มนี้ถูกแยกต่างหากจากสี่กลุ่ม

กลุ่มพื้นฐาน (Base Group)

ออบเบ็จ็ค้กลุ่มนี้ประกอบด้วย 3 สกาลาร์ 1 ตาราง

สกาลาร์ออบเบ็จ็ค้คือ

dot1dBaseBridgeAddress (MacAddress) : หมายถึง MAC แอดเดรสที่บริดจ์ใช้

dot1dBaseNumPorts (INTEGER) : หมายถึง จำนวนพอร์ตที่บริดจ์ควบคุมอยู่

dot1dBaseType (INTEGER) : หมายถึง ชนิดของบริดจ์ ที่บริดจ์สามารถกระทำได้ คือ

unknown(1) : ซึ่งถึงการไม่รู้ชนิดของบริดจ์

transparent-only(2) : ซึ่งถึงทรานส์พาเรนบริดจ์เท่านั้น

sourceroute-only(3) : ซึ่งถึงซอร์สเราดิงเท่านั้น

srt(4) : ซึ่งถึงเฮสซอร์ที่บริดจ์

ตารางพอร์ต คือ ตาราง dot1dBasePortTable มี dot1dBasePortEntry เป็นรูปแบบแถวมี dot1dBasePort เป็นคอลัมน์ แถวแต่ละแถวคือพอร์ตแต่ละพอร์ตของบริดจ์

INDEX { dot1dBasePort }

ภายในตารางประกอบด้วยออบเบ็จ็ค้คือ

dot1dBasePort (INTEGER) : หมายถึง หมายเลขที่ใช้ระบุพอร์ต

dot1dBasePortIfIndex (INTEGER) : หมายถึง ค่าของ ifIndex ตรงกันกับอินเทอร์เฟซ

dot1dBasePortCircuit (OBJECT IDENTIFIER) : หมายถึง เมื่อใดก็ตามที่พอร์ตมีค่าของ dot1dBaseIfIndex เหมือนกันหลายพอร์ต ออบเบ็จ็ค้จะบรรจุชื่ออินสแตนซ์พิเศษของพอร์ต ในกรณีนี้ ออบเบ็จ็ค้สามารถมีค่า {0.0}

dot1dBasePortDelayExceededDiscards (Counter32) : หมายถึง จำนวนเฟรมที่ถูกยกเลิกเนื่องจากการส่งผ่านบริดจ์เกิดดีเลย์มากเกินไป

dot1dBasePortMtuExceededDiscards (Counter32) : หมายถึง จำนวนเฟรมที่ถูกยกเลิกเนื่องจากมีขนาดมากกว่าขนาดสูงสุดที่เชื่อมต่อเซกเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่ม STP (Spanning Tree Protocol Group)

ออบเจกต์กลุ่มนี้ประกอบด้วย 14 สกาลาร์ 1 ตาราง

สกาลาร์ออบเจกต์คือ

dot1dStpProtocolSpecification (INTEGER) : หมายถึง เวอร์ชันสเปกนิฟิเคชันโปรโตคอลที่กำลังดำเนินการอยู่คือ

unknown(1) : ซึ่ถึงไม่รู้เวอร์ชัน

decLb100(2) : ซึ่ถึง DEC แลนบริดจ์ 100

ieee8021d(3) : ซึ่ถึง IEEE มาตรฐาน

dot1dStpPriority (INTEGER) : หมายถึง ค่าลำดับความสำคัญของบริดจ์ที่สามารถเซตให้เปลี่ยนแปลงค่าได้ ซึ่งค่านี้เป็น 2 ออกเททแรกของบริดจ์ไอดี

dot1dStpTimeSinceTopologyChange (TimeTicks) : หมายถึง เวลาที่บริดจ์ตรวจสอบพบการเปลี่ยนแปลงโทโปโลยีครั้งสุดท้าย

dot1dStpTopChanges(Counter32) : หมายถึง จำนวนโทโปโลยีทั้งหมดที่บริดจ์ตรวจสอบพบการเปลี่ยนแปลง

dot1dStpDesignatedRoot (BridgeId) : หมายถึง บริดจ์ไอดีของรูทของสเปกนิฟิเคชันที่กำหนดโดยบริดจ์

dot1dStpRootCost (INTEGER) : หมายถึง รูทพาทคอสต์ที่เห็นจากบริดจ์นั้น

dot1dStpRootPort (INTEGER) : หมายถึง พอร์ตของบริดจ์ซึ่งนำไปยังรูทบริดจ์

dot1dStpMaxAge (Timeout) : หมายถึง ช่วงเวลา (Age) สูงสุด ซึ่ง STP สามารถเรียนรู้สารสนเทศจาก BPDUs (Bridge Protocol Data Units) ก่อนที่จะถูกยกเลิก

dot1dStpHelloTime (Timeout) : หมายถึง ช่วงเวลาระหว่างการส่งผ่าน BPDUs ของบริดจ์เมื่อเป็นรูทหรือการพยายามเป็นรูท

dot1dStpHoldTime (INTEGER) : หมายถึง ช่วงเวลาระหว่างที่บริดจ์ทำการส่ง BPDUs เมื่อไม่ได้เป็นรูท

dot1dStpForwardDelay (Timeout) : หมายถึง ช่วงเวลาที่พอร์ต รอสภาวะการฟังและการเรียนรู้ ก่อนที่จะส่งไปยังสภาวะฟอร์เวิร์ดดิ้ง ค่านี้ถูกใช้เมื่อมีการตรวจสอบพบการเปลี่ยนแปลงโทโปโลยี ซึ่งช่วงเวลาใดนามิกทั้งหมดจะนำเข้าสู่ฐานข้อมูลฟอร์เวิร์ดดิ้ง

dot1dStpBridgeMaxAge (Timeout) : หมายถึง ค่าที่ใช้สำหรับ dot1dStpMaxAge เมื่อบริดจ์กระทำหน้าที่เป็นรูท

dot1dStpBridgeHelloTime (Timeout) : หมายถึง ค่าที่ใช้สำหรับ dot1dStpHelloTime เมื่อบริดจ์กระทำหน้าที่เป็นรูท

dot1dStpBridgeForwardDelay (Timeout) : หมายถึง ค่าที่ใช้สำหรับ dot1dStpForward-Delay เมื่อบริดจ์กระทำหน้าที่เป็นรูท

ตารางพอร์ต คือ ตาราง dot1dStpPortTable มี dot1dStpPortEntry เป็นรูปแบบแถวมี dot1StpPort เป็นคอลัมน์ แถวแต่ละแถวคือพอร์ตแต่ละพอร์ตของบริดจ์

INDEX { dot1dStpPort }

ภายในตารางประกอบด้วยออบเจกต์คือ

dot1dStpPort (INTEGER) : หมายถึง หมายเลขที่ใช้ระบุพอร์ต

dot1dStpPortPriority (INTEGER) : หมายถึง ค่าลำดับความสำคัญของบริดจ์ที่สามารถเซตให้เปลี่ยนแปลงค่าได้ ซึ่งค่านี้เป็น 2 ออกเททแรกของพอร์ตไอดี

dot1dStpPortState (INTEGER) : หมายถึง สถานะปัจจุบันของพอร์ต คือ

disabled(1) : ชี้ถึงการดิสเอเบิล

blocking(2) : ชี้ถึงการรับ BPDUs ทั้งฟอร์เวิร์ดดิ้งและการเรียนรู้ถูกดิสเอเบิล

listening (3) : ชี้ถึงการแลกเปลี่ยน BPDUs ทั้งฟอร์เวิร์ดดิ้งและการเรียนรู้ถูก

ดิสเอเบิล

learning (4) : ชี้ถึงการแลกเปลี่ยน BPDUs พอร์เวิร์ดดิ้งถูกดิสเอเบิลส่วนการ

เรียนรู้ถูกอีนาเบิล

forwarding (5) : ชี้ถึงการแลกเปลี่ยน BPDUs ทั้งฟอร์เวิร์ดดิ้งและการเรียนรู้ถูก

อีนาเบิล

broken (6) : ชี้ถึงการเกิดความผิดพลาดของเครื่อง

dot1dStpPortEnable (INTEGER) : หมายถึง การอีนาเบิลหรือดิสเอเบิล สถานะของพอร์ต

dot1dStpPortPathCost (INTEGER) : หมายถึง พาราคอสต์ของพอร์ต ซึ่งเป็นพารที่ใช้ไป

ยังรูท

dot1dStpPortDesignatedBridge (BridgeId) : หมายถึง บริดจ์ไอดีที่พิจารณาจากดีไซค์เนทบริดจ์ของเลนที่ถูกเชื่อมต่อไปยังพอร์ตนั้น

dot1dStpPortDesignatedRoot(BridgeId):หมายถึง บริดจ์ไอดีของรูทที่ชี้ถึงการรับ BPDUs จากดีไซค์เนทบริดจ์ของเลนที่ถูกเชื่อมต่อไปยังพอร์ตนั้น

dot1dStpPortDesignatePort(OCTET STRING): หมายถึง พอร์ตไอดีของดีไซค์เนทพอร์ตของเลนที่ถูกเชื่อมต่อไปยังพอร์ตนั้น

dot1dStpPortDesignateCost (INTEGER) : หมายถึง พาราคอสต์ของดีไซค์เนทพอร์ตของเลนที่ถูกเชื่อมต่อไปยังพอร์ตนั้น

dot1dStpPortForwardTransitions (INTEGER) : หมายถึง ช่วงเวลาที่พอร์ตเปลี่ยนสถานะการเรียนรู้ไปยังสถานะฟอร์เวิร์ดคิง

กลุ่ม TP (Transparent Group)

ออบเจ็กต์กลุ่มนี้ประกอบด้วย 2 สกอลาร์ 2 ตาราง

สกอลาร์ออบเจ็กต์คือ

dot1dTpLearnedEntryDiscards (Counter32) : หมายถึง จำนวนการบันทึกข้อมูลทั้งหมดที่มีการเรียนรู้หรืออาจมีการเรียนรู้ในฐานข้อมูลฟอร์เวิร์ดคิง แต่ถูกยกเลิกเนื่องจากมีพื้นที่ไม่เพียงพอในฐานข้อมูลฟอร์เวิร์ดคิง

dot1dTpAgingTime (INTEGER) : หมายถึง ช่วงเวลาใหม่เฮดส์สำหรับการฟอร์เวิร์ดคิงอินฟอร์เมชันที่ถูกเรียนรู้อย่างไดนามิก ค่าดีฟอลต์คือ 300 วินาที

ตารางแรกคือ dot1dTpFdbTable มี dot1dTpFdbEntry เป็นรูปแบบแถวมี dot1dTpFdbAddress เป็นคอลัมน์

INDEX { dot1dTpFdbAddress }

ภายในตารางประกอบด้วยออบเจ็กต์คือ

dot1dTpFdbAddress(MacAddress) : หมายถึง MAC แอดเดรสที่บริดจ์มีการฟอร์เวิร์ดคิงและหรือการกรองอินฟอร์เมชัน

dot1dTpFdbPort (INTEGER) : หมายถึง บริดจ์มีการฟอร์เวิร์ดคิงและหรือการกรองอินฟอร์เมชันของ MAC แอดเดรส ทั้งหมายเลขพอร์ตที่ถูกเรียนรู้และหมายเลขพอร์ตที่ไม่ถูกเรียนรู้ที่ชี้ค่า "0"

dot1dTpFdbStatus (enumerated INTEGER) : หมายถึง สถานะของแถวมีค่าดังนี้

other(1) : ชี้ถึงไม่มีค่าตามรายการข้างล่าง

invalid(2) : ชี้ถึงรายการที่ใช้ไม่ได้ เช่น การเรียนรู้ที่หมดอายุ (Aged-Out)

learned(3) : ชี้ถึงค่าการเรียนรู้ที่ใช้ได้

self(4) : ชี้ถึงค่าแอดเดรสของบริดจ์

mgmt(5) : ชี้ถึงค่าที่บรรจุอยู่ในตาราง dot1dStaticTable

ตารางที่สองคือ dot1dTpPortTable มี dot1dTpPortEntry เป็นรูปแบบแถวมี dot1dTpPort เป็นคอลัมน์ บรรจุอินฟอร์เมชันในแต่ละพอร์ตของบริดจ์

INDEX { dot1dTpPort }

ภายในตารางประกอบด้วยออบเจ็กต์คือ

dot1dTpPort (INTEGER) : หมายถึง การระบุพอร์ตที่บรรจุทรานส์พาเรนบริดจ์

dot1dTpPortMaxInfo(INTEGER): หมายถึง ขนาดใหญ่สุดของข้อมูลใน MAC เฟรมของพอร์ตนั้น

dot1dTpPortInFrames (Counter32) : หมายถึง จำนวนเฟรมที่ถูกรับบนพอร์ต

dot1dTpPortOutFrames (Counter32) : หมายถึง จำนวนเฟรมที่ถูกส่งบนพอร์ต

dot1dTpPortInDiscards (Counter32) : หมายถึง จำนวนการรับเฟรมที่ใช้ได้ แต่ถูกยกเลิกเนื่องจากถูกรองโดยกระบวนการฟอร์เวิร์ดดิ้งของบริดจ์

กลุ่มสแตติก (Static Group)

กลุ่มนี้ประกอบด้วย 1 ตารางคือ dot1dStaticTable มี dot1dStaticEntry เป็นรูปแบบแถวมี dot1dStaticAddress, dot1dStaticReceivePort เป็นคอลัมน์ ได้จัดเตรียมการสร้างแบบสแตติกเพื่อนำเข้าสู่ฐานข้อมูลฟอร์เวิร์ดดิ้งของบริดจ์

INDEX { dot1dStaticAddress, dot1dStaticReceivePort }

ภายในตารางประกอบด้วยออบเจกต์คือ

dot1dStaticAddress (MacAddress) : หมายถึง การระบุ MAC แอดเดรสปลายทาง ค่านี้เป็นได้ทั้งยูนิคาสต์แอดเดรส, กลุ่มแอดเดรส หรือบรอดคาสต์แอดเดรส

dot1dStaticReceivePort (INTEGER) : หมายถึง การระบุพอร์ตที่รับ ถ้าค่าหมายเลขพอร์ตเป็น "0" ซึ่งถึงพอร์ตทั้งหมดของบริดจ์สามารถใช้ได้ ถึงแม้ว่าพอร์ตนั้นจะไม่ถูกคอนฟิกก็ตาม

dot1dStaticAllowedToGoTo (OCTET STRING) : หมายถึง การเซตของพอร์ตที่อนุญาตให้ฟอร์เวิร์ดเฟรม ค่าออกเทตแต่ละบิตจะแทนที่พอร์ต บิตที่มีค่า "1" จะยอมให้ฟอร์เวิร์ดเฟรม ออกเทตแรกจะเป็นการระบุพอร์ต 1-8 บิตที่มีนัยสำคัญมากที่สุดคือพอร์ตที่มีหมายเลขต่ำสุด ซึ่งก็คือพอร์ตหนึ่ง ส่วนบิตที่มีนัยสำคัญน้อยที่สุดคือพอร์ตที่มีหมายเลขสูงสุด ส่วนออกเทตที่สองจะเป็นการระบุพอร์ต 9-16 และเป็นอย่างนี้ต่อไป

dot1dStaticStatus (enumerated INTEGER) : หมายถึง สถานะของแถว ค่าดีฟอลต์คือ permanent (3) มีค่าคือ

other(1) : ซึ่งถึงไม่มีค่าดังรายการข้างล่าง

invalid(2) : ซึ่งถึงการลบรายการที่ตรงกัน

permanent(3) : ซึ่งถึงการคงค่าเช่นนี้หลังจากรีเซตบริดจ์

deleteOnReset(4) : ซึ่งถึงการคงค่าเช่นนี้จนกระทั่งรีเซตบริดจ์

deleteOnTimeout(5) : ซึ่งถึงการคงค่าเช่นนี้จนกระทั่งหมดอายุ

กลุ่ม SR (Source Routing Group)

กลุ่มซอร์สเรติงบริดจ์ [14] ถูกกำหนดตามเอกสาร RFC 1525 ซอร์สเรติงแบ่งออกเป็น

2 กลุ่มหลัก คือกลุ่มซอร์สเรติง และกลุ่มพอร์ตแพร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มซอร์สเรา (Source Route Group) ประยุกต์ใช้สำหรับซอร์สเราติงบริดจ์ทุกชนิด กลุ่มนี้ประกอบด้วย 1 สกอลาร์ 1 ตาราง

สกอลาร์ออบเจกต์คือ

dot1dSrBridgeLfMode (enumerated INTEGER) : หมายถึง การสร้างหรือการคาดหมายการรับ RIFs วิธีเก่าใช้ความยาวสูงสุดของซัพฟิลด์ 3 บิต (Mode3) หรือวิธีใหม่ใช้ความยาวสูงสุดของซัพฟิลด์ 6 บิต (Mode6) ซึ่งบริดจ์สนับสนุนความยาวทั้ง 2 โหมด

ตารางพอร์ตคือ dot1dSrPortTable มี dot1dSrPortEntry เป็นรูปแบบแถวมี dot1dSrPort เป็นคอลัมน์

INDEX { dot1dSrPort }

ภายในตารางประกอบด้วยออบเจกต์คือ

dot1dSrPort (INTEGER) : หมายถึง การระบุพอร์ตที่บรรจุซอร์สเราติงบริดจ์

dot1dSrPortHopCount (INTEGER) : หมายถึง จำนวนสูงสุดของ RDs ที่ถูกอนุญาตในออลพาร์เอ็กซ์พลอเรอร์เฟรมและสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรม

dot1dSrPortLocalSegment (INTEGER) : หมายถึง ริงนัมเบอร์ของเซกเมนต์ที่เชื่อมต่อพอร์ตอยู่ ค่า 65535 ซึ่งไม่มีหมายเลขเซกเมนต์ถูกกำหนดบนพอร์ต

dot1dSrPortTargetSegment (INTEGER) : หมายถึง ริงนัมเบอร์ของเซกเมนต์เป้าหมาย (ปกติจะเป็นริงเสมือนภายในบริดจ์)ที่เชื่อมต่อพอร์ตอยู่ ค่า 65535 ซึ่งไม่มีเซกเมนต์เป้าหมายถูกกำหนดบนพอร์ต

dot1dSrPortBridgeNum (INTEGER) : หมายถึง หมายเลขบริดจ์ที่ใช้ระหว่างโลกออลเซกเมนต์และเซกเมนต์เป้าหมาย ค่า 65535 ซึ่งไม่มีหมายเลขบริดจ์ถูกกำหนดบนพอร์ต

dot1dSrPortLargestFrame (INTEGER) : หมายถึง ขนาดสูงสุดของฟิลด์ข้อมูลของ MAC เฟรมบนพอร์ต ค่าที่ใช้ได้มี 64 ค่า (ระหว่าง 516-65535 ออกเทท) กำหนดโดย IEEE แต่บริดจ์ไม่ทุกชนิดที่ยอมรับค่าทั้งหมดนี้

dot1dSrPortSTESpanMode (INTEGER) : หมายถึง วิธีการประมวลผลของสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรมคือ

disabled(1) : ซึ่งถึงพอร์ต ไม่ยอมรับหรือส่งสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรม

forced(2) : ซึ่งถึงพอร์ตยอมรับและกระจายสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรม

auto-span(3) : ซึ่งถึงเมื่อใดที่ STP ถูกอินาเบลพอร์ต สแปนนิ่งทรีเอ็กซ์พลอเรอร์

เฟรมจะถูกประมวลผลตามค่าของ dot1dStpPortState

dot1dSrPortSpecInFrames (Counter32) : หมายถึง จำนวนซอร์สเราเฟรมที่รับจากพอร์ต

dot1dSrPortSpecOutFrames(Counter32):หมายถึง จำนวนซอร์สเราเฟรมที่ส่งไปยังพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dot1dSrPortApeInFrames (Counter32) : หมายถึง จำนวนออกพาสเอ็กซ์พลอเรอร์เฟรมที่รับจากพอร์ต

dot1dSrPortApeOutFrames (Counter32) : หมายถึง จำนวนออกพาสเอ็กซ์พลอเรอร์เฟรมที่ส่งไปยังพอร์ต

dot1dSrPortSteInFrames(Counter32) : หมายถึง จำนวนสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรมที่รับจากพอร์ต

dot1dSrPortSteOutFrames(Counter32):หมายถึง จำนวนสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรมที่ส่งไปยังพอร์ต

dot1dSrPortSegmentMismatchDiscards (Counter32) : หมายถึง จำนวนเอ็กซ์พลอเรอร์เฟรมที่รับจากพอร์ต แต่ถูกยกเลิกเนื่องจากฟิลด์ RD บรรจุนับเบอร์ประชิดที่ใช้งานไม่ได้

dot1dSrPortDuplicateSegmentDiscards (Counter32) : หมายถึง จำนวนเฟรมที่รับจากพอร์ต แต่ถูกยกเลิกเนื่องจากฟิลด์ RD บรรจุนับเบอร์ที่ซ้ำซ้อนกัน

dot1dSrPortHopCountExceededDiscards (Counter32) : หมายถึง จำนวนเอ็กซ์พลอเรอร์เฟรมที่รับจากพอร์ต แต่ยกเลิกเนื่องจาก RIF มากเกินหรือน่าจะมากกว่าจำนวนสูงสุดของ RDs

dot1dSrPortDupLanIdOrTreeErrors (Counter32) : หมายถึง จำนวนซ้ำซ้อนกันของแลนไอดีหรือความผิดพลาดทรี ใช้ตรวจสอบปัญหาเครือข่ายซอร์สเรดิ้งบริดจ์ของ IBM รุ่นก่อน

dot1dSrPortLanIdMismatches(Counter32):หมายถึง จำนวนออกพาสเอ็กซ์พลอเรอร์เฟรมและสแปนนิ่งทรีเอ็กซ์พลอเรอร์เฟรมที่รับจากพอร์ต แต่ยกเลิกเนื่องจากริงนับเบอร์ของ RD ท้ายสุดไม่เท่ากับโลคอลริงนับเบอร์ของพอร์ต

กลุ่มพอร์ตแพร์ (Port-Pair Group) ประยุกต์ใช้สำหรับบริดจ์ที่สนับสนุนไคเร็กมัลติพอร์ตของซอร์สเรดิ้งบริดจ์ กลุ่มนี้ประกอบด้วย 1 สกาลาร์ 1 ตาราง

สกาลาร์ออบเจกต์คือ

dot1dPortPairTableSize (Gauge32) : หมายถึง จำนวนรายการทั้งหมดในฐานข้อมูลพอร์ตแพร์ของบริดจ์

ตารางพอร์ต คือตาราง dot1dPortPairTable มี dot1dPortPairEntry เป็นรูปแบบแถวมี dot1dPortPairLowPort, dot1dPortPairHighPort เป็นคอลลัมน์

INDEX { dot1dPortPairLowPort, dot1dPortPairHighPort }

ตาราง dot1dPortPairTable แทนฐานข้อมูลพอร์ต-แพร์ แถวแต่ละแถวคือแต่ละแพร์ของพอร์ต แต่ละแถวจะแทนด้วยชุดสาม (ริงนับเบอร์, บริดจ์นับเบอร์, ริงนับเบอร์) ซึ่งพอร์ตหนึ่งของแพร์จะเชื่อมต่อไปยังริงนับเบอร์หนึ่งและพอร์ตอื่นก็จะเชื่อมต่อไปยังริงนับเบอร์อื่น

ภายในตารางประกอบด้วยออบเจกต์คือ

dotIdPortPairLowPort (INTEGER) : หมายถึง การระบุพอร์ตของแพร์ที่มีหมายเลขต่ำกว่า

dotIdPortPairHighPort (INTEGER): หมายถึง การระบุพอร์ตของแพร์ที่มีหมายเลขสูงกว่า

dotIdPortPairBridgeNum (INTEGER) : หมายถึง หมายเลขบริดจ์ชุดสาม (Triplte) ที่เกี่ยวข้องกัน (ริงนัมเบอร์, บริดจ์นัมเบอร์, ริงนัมเบอร์)

dotIdPortPairBridgeState (enumerated INTEGER) : หมายถึง สถานะการเชื่อมต่อกันของพอร์ตแพร์บริดจ์ คือ

invalid(1) : ชี้ถึงการลบพอร์ตแพร์

enbled (2) : ชี้ถึงสถานะการอินาเบิล

disabled (3) : ชี้ถึงสถานะการดิสเอเบิล



บทที่ 5

การจัดการข่ายคอมพิวเตอร์บริเวณกว้าง (แวน)

การจัดการข่ายคอมพิวเตอร์บริเวณกว้าง หรือแวน (Wide Area Network : WAN) คือการเชื่อมต่อเครือข่ายสองเครือข่ายหรือมากกว่าเข้าด้วยกันจนกลายเป็นเน็ตเวิร์กขนาดใหญ่ เครือข่ายอาจอยู่ในสถานที่เดียวกันหรืออยู่คนละเมืองกันก็ได้ ระบบแวนที่กระจายกันอยู่สามารถเชื่อมต่อกันได้โดยผ่านสายโทรศัพท์ทางโมเด็ม หรือถ้าต้องการความเร็วสูงค่าใช้จ่ายก็แพงตามไปด้วยสามารถเชื่อมต่อกันด้วยอุปกรณ์พิเศษคล้ายโมเด็มที่เรียกว่าหน่วยบริการข้อมูลหรือ DSUs (Data Service Units) โดยผ่านสายโทรศัพท์แบบชั้นคุณภาพข้อมูล (Data Grade) เช่น T1 และ T3 ที่พบได้ในผู้ให้บริการที่เป็นบริษัทโทรศัพท์ขนาดใหญ่

บทนี้จะครอบคลุมถึงการจัดการสารสนเทศทางกายภาพของสื่อ 2 ชนิดคือ DS1 และ DS3 อินเทอร์เน็ต สาขา DS1 และ DS3 เป็นมาตรฐานที่ใช้กันอยู่ในอเมริกาเหนือ เริ่มแรกถูกกำหนดโดย AT&T และต่อมาโดย ANSI (American National Standard Institute) ซึ่งจะสมมูลกับ E1 และ E3 ที่เป็นมาตรฐานใช้กันอยู่ในยุโรป ซึ่งถูกกำหนดโดย ITU (International Telecommunication Union) มาตรฐานทั้งสองมีความแตกต่างกันทางด้านความเร็วของคู่สาย

DS1 MIB สามารถประยุกต์ใช้ได้ทั้งใน DS1 และ E1 เช่นเดียวกับ DS3 MIB สามารถประยุกต์ใช้ได้ทั้งใน DS3 และ E3 [6]

5.1 หลักการทั่วไป

กล่าวถึงหลักการและกฎเกณฑ์โดยทั่วไป

การมัลติเพล็กซ์ DS1/DS3

มาตรฐานที่ใช้กันอยู่ในอเมริกาเหนือ หน่วยพื้นฐานสำหรับช่อง DS0 ของการมัลติเพล็กซ์คือ 64 กิโลบิตต่อวินาที ช่อง DS0 หรือที่ถูกเรียกอ้างอิงถึงบ่อย ๆ คือ ช่องสัญญาณชั้นคุณภาพเสียง (Voice Grade) ซึ่งเข้ารหัสเสียงแบบพีซีเอ็ม (PCM : Pulse Code Modulation) ส่วน DS1 และ DS3 ใช้ทีดีเอ็ม (TDM: Time Division Multiplexing) ในการนำพาสัญญาณไปบนหลายช่องสัญญาณ DS0 ทีดีเอ็มเป็นเทคนิคที่ใช้ในการส่งสัญญาณหลายสัญญาณผ่านช่องสื่อสารช่องเดียว โดยวิธีการแบ่งเวลาออกเป็นช่วงเวลาสั้น ๆ (Time Slot) สำหรับสัญญาณแต่ละสัญญาณ

DS1 สามารถนำพาช่องเสียง DS0 จำนวน 24 ช่อง ที่อัตราการสุ่มตัวอย่าง 8000 เฮิรตซ์ แต่ละช่องสัญญาณประกอบด้วย 8 บิต และบิตสำหรับซิงโครไนซ์ซึ่งเพิ่มอีก 1 บิต ดังนั้น 1 เฟรมในระบบจะมี 193 บิต มีอัตราการส่ง (Bit Rate) เท่ากับ 1.544 เมกะบิตต่อวินาที

$$((24*8)+1)*8000 = 1.544 \text{ Mb/s}$$

DS3 สามารถนำพาช่องเสียง DS1 จำนวน 28 ช่อง หรือ DS0 $(28*24) = 672$ ช่อง ที่อัตราการส่ง (Bit Rate) เท่ากับ 44.736 เมกะบิตต่อวินาที ส่วน DS2 และ DS4 นั้นไม่ค่อยได้ใช้กัน

สำหรับมาตรฐานที่ใช้กันอยู่ในยุโรป มีอัตราการส่งหรือความเร็วของกลุ่มสายแตกต่างจากมาตรฐานอเมริกาเหนือคือ E1 มีอัตราการส่ง (Bit Rate) เท่ากับ 2.048 เมกะบิตต่อวินาที ประกอบด้วยช่องเสียงขนาด 64 กิโลบิตต่อวินาที จำนวน 30 ช่อง รวมกับ 2 ช่องสำหรับ Signaling และ Framing ส่วน E3 มีอัตราการส่ง (Bit Rate) เท่ากับ 34.368 เมกะบิตต่อวินาที ประกอบด้วยช่องเสียง E1 จำนวน 16 ช่อง

ตารางที่ 5.1 เปรียบเทียบการมัลติเพล็กซ์มาตรฐานอเมริกาเหนือและมาตรฐานยุโรป

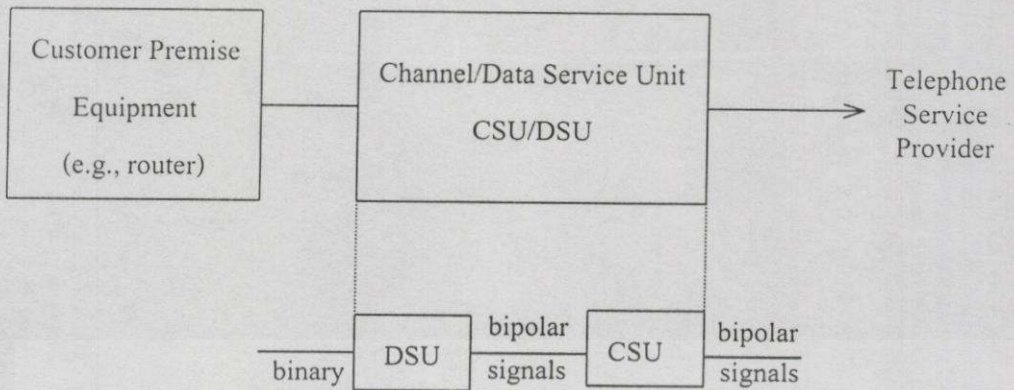
กลุ่มประเทศ	ระดับ				
	DS0	DS1	DS2	DS3	DS4
อเมริกาเหนือ					
จำนวนช่องเสียง	1	24	96	672	4032
อัตราการส่ง	64 Kb/s	1.544 Mb/s	6.312 Mb/s	44.763 Mb/s	274.176 Mb/s
ยุโรป					
จำนวนช่องเสียง	1	30	120	480	1920
อัตราการส่ง	64 Kb/s	2.048 Mb/s	8.448 Mb/s	34.368 Mb/s	139.864 Mb/s

หน่วยบริการช่องสัญญาณ/หน่วยบริการข้อมูล (CSU/DSU)

การเชื่อมโยง DS1/DS3 เพื่อเชื่อมต่อระหว่างสถานที่ผู้ใช้ไปยังบริษัทผู้ให้บริการโทรศัพท์อย่างน้อยที่สุดจะต้องมีอุปกรณ์ที่เรียกว่า หน่วยบริการช่องสัญญาณ/หน่วยบริการข้อมูล (Channel Service Units/Data Service Units) หรือเรียกว่า CSUs และ DSUs แสดงดังรูปที่ 5.1

อุปกรณ์ CSUs จะสิ้นสุดวงจรที่บริษัทผู้ให้บริการโทรศัพท์ หน้าที่ของ CSUs คือการปรับสภาพสาย (Line Conditioning) และการปรับเท่า (Equalization) ซึ่งใช้เกี่ยวกับการขยายเพื่อชดเชยสัญญาณที่สูญเสีย, การควบคุมความผิดพลาด และความสามารถในการตอบสนองของวงจรแบบเบ็กเพื่อทดสอบคำสั่ง

อุปกรณ์ DSUs ทำหน้าที่จัดเตรียมการซิงโครไนซ์เฟรมและการฟื้นตัวของเวลา (Timing Recovery) รวมถึงแปลงสัญญาณไบนารีจากอุปกรณ์ผู้ใช้เป็นสัญญาณไบโพลาร์ส่งไปยังอุปกรณ์ DS1 ของผู้ให้บริการโทรศัพท์ โดยปกติแล้วทั้ง DSUs และ CSUs จะถูกรวมอยู่ในอุปกรณ์เดียวกัน



รูปที่ 5.1 แสดงหน้าที่ของ CSUs/DSUs

การกำหนดค่าระหว่าง ifIndex และ DS1/DS3 อินเทอร์เฟซ

เมื่อใดที่อุปกรณ์ เช่นเราเตอร์กระทำหน้าที่เป็นพรีอ็อกซีสำหรับโพลอด CSU (สถานีจัดการส่งการร้องขอไปยังเราเตอร์และเราเตอร์ส่งต่อไปยัง CSU) เราเตอร์จำเป็นต้องมีการรู้จำ (Recognize) แต่ละอินเทอร์เฟซของ DS1/DS3 ซึ่งแบ่งเป็น 3 อินสแตนซ์คือ CSU ที่ต่อไปยังอินเทอร์เฟซเราเตอร์ (เช่น RS-232 เชื่อมต่อระหว่างเราเตอร์และ CSU), CSU ด้านเราเตอร์ และ CSU ด้านเครือข่าย การระบุค่าอินเทอร์เฟซที่ใช้ใน DS1/DS3 ไม่เพียงแต่ใช้ค่าของ ifIndex ยังมีค่าของ dsx1LineIndex หรือ dsx3LineIndex เพื่อใช้แบ่งแยก 3 อินสแตนซ์ของเราเตอร์

เมื่อใดที่อุปกรณ์ทำหน้าที่เป็นโพลอด SNMP เอเจนต์

- โพลอดอินเทอร์เฟซมีค่า ifIndex เหมือนกับค่า dsx1LineIndex หรือ dsx3LineIndex

เมื่อใดที่อุปกรณ์ทำหน้าที่เป็นพรีอ็อกซีสำหรับโพลอด CSU

- ค่าจำนวนเลขคู่จะมากกว่าค่าของ ifIndex ที่เลือกจากอุปกรณ์พรีอ็อกซีทั้งหมด ส่วนค่าจำนวนเลขคี่จะมากกว่าอยู่หนึ่ง

- ค่าจำนวนเลขคี่ที่กำหนดใน dsx1LineIndex หรือ dsx3LineIndex จะถูกนำไปแทนที่อินเทอร์เฟซด้านเครือข่ายของ CSU

- ค่าจำนวนเลขคู่ที่กำหนดใน dsx1LineIndex หรือ dsx3LineIndex จะถูกนำไปแทนที่อินเทอร์เฟซด้านเราเตอร์ของ CSU

ตัวอย่างการกำหนด ifIndex และ dsx1LineIndex สำหรับเราเตอร์ที่ทำหน้าที่เป็นพรีอ็อกซี โดยสมมุติเราเตอร์มีหนึ่งอินเทอร์เฟซ ดังนั้นจำนวนอินเทอร์เฟซแสดงให้เห็นดังตารางที่ 5.2

ตารางที่ 5.2 การกำหนด ifIndex และ dsx1LineIndex

	ifIndex	dsx1LineIndex
อินเทอร์เฟซเราเตอร์อื่น	1	-
อินเทอร์เฟซเราเตอร์	2	2
อินเทอร์เฟซด้านเราเตอร์ของ CSU	2	4
อินเทอร์เฟซด้านเครือข่ายของ CSU	2	5

ความหมายของค่า

เป็นการอธิบายความหมายของค่าที่เกี่ยวข้องกับเงื่อนไขความผิดพลาดที่ใช้กับ DS1/DS3 โดยแบ่งกลุ่มไว้ดังนี้

● ความผิดพลาดของ DS1/E1

การนับความผิดพลาดของ DS1 และ E1 [6] แบ่งเป็น 3 กลุ่มคือ

- กลุ่มการนับเหตุการณ์ที่เกิดขึ้นจากชนิดของความผิดพลาด
- กลุ่มการนับช่วงเวลาที่เกิดขึ้นจากชนิดของความผิดพลาด
- กลุ่มการนับช่วงเวลาที่เกิดความผิดพลาดที่รุนแรง

กลุ่มการนับเหตุการณ์ที่เกิดขึ้นจากชนิดของความผิดพลาด ประกอบด้วย

Line Coding Violations (LCV) : หมายถึง ความผิดพลาดที่ถูกตรวจจับภายในเฟรมที่รับ ทั้งการละเมิดไบโพลาร์ และการเกิดบิต 0 ที่ต่อเนื่องกันมากเกินไป

Path Coding Violations (PCV) : หมายถึง ความผิดพลาดของบิตในการซิงโครไนซ์เฟรม เช่น ความผิดพลาด CRC

Controlled Slips (CS) : หมายถึง การทำซ้ำ (Replication) หรือส่งสลิปเพย์โหลดบิต กระทำเมื่อเวลาระหว่างการรับซิงโครไนซ์ของเทอร์มินัลและสัญญาณที่รับมีค่าไม่ตรงกัน

Framing Errors : หมายถึง ความผิดพลาดในการกู้คืนเฟรม (Recovery) ที่รับเข้าของ กระแสบิต (Bit Stream) ความผิดพลาดแบ่งเป็น 2 กรณีคือ กรณีข้อบกพร่อง (Defect) เกิดจากความผิดพลาดของเฟรม ประกอบด้วย OOF (Out Of Frame), LOS (Loss of Signal) และ AIS (Alarm Indication Signal) ส่วนกรณีความล้มเหลว (Failure) เป็นความผิดพลาดของเฟรมหลายเฟรมเกิดขึ้น ติดต่อกัน ประกอบด้วย LOF (Loss Of Frame), LOS และ AIS

กลุ่มการนับช่วงเวลาที่เกิดขึ้นจากชนิดของความผิดพลาด ประกอบด้วย

Errored Seconds (ES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ความผิดพลาดเกิดขึ้น

Controlled Slip Seconds (CSS) : หมายถึง ช่วงเวลาเป็นวินาที ที่ CS เกิดขึ้น

Line Errored Seconds (LES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ LCV เกิดขึ้น

Bursty Errored Seconds (BES) : หมายถึง ช่วงเวลาเป็นวินาที ที่หลาย PCV เกิดขึ้น

กลุ่มการนับช่วงเวลาที่เกิดความผิดพลาดที่รุนแรง (Severely) ระดับความรุนแรงสูงสุด คือ ช่วงเวลาที่สายไม่สามารถใช้งานได้ (Unavailable) ประกอบด้วย

Severely Errored Framing Seconds (SEFS) : หมายถึง ช่วงเวลาเป็นวินาที ที่เกิดหนึ่งหรือหลายความผิดพลาดของการจัดเฟรมเกิดขึ้น

Severely Errored Seconds (SES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ความผิดพลาดเกิดขึ้นในหลายกรณีแปรผันไปตามรูปแบบการจัดเฟรม

Unavailable Seconds (UAS): หมายถึง ช่วงเวลาเป็นวินาที ที่สายไม่สามารถใช้ได้ เกิดจาก

- การเกิด SES ติดต่อกัน 10 ครั้ง
- การเกิดเงื่อนไขความผิดพลาดการจัดเฟรมบ่อย ๆ
- การไม่เกิด SES ติดต่อกัน 10 ครั้ง
- เงื่อนไขความผิดพลาดการจัดเฟรมถูกเคลียร์

Degraded Minutes (DM) : หมายถึง ช่วงเวลาเป็นนาที ซึ่งประมาณอัตราความผิดพลาดบิตที่มากเกิน $1E-6$ แต่ไม่เกิน $1E-3$ ตามมาตรฐาน G.821

● ความผิดพลาดของ DS3/E3

การแบ่งประเภทความผิดพลาดของกลุ่มนี้จะเหมือนกับกลุ่ม DS1/E1 รูปแบบการจัดเฟรมที่ใช้บนสาย DS3/E3 มีรายละเอียดที่แตกต่างกันบ้างเล็กน้อยกับ DS1/E1 อย่างไรก็ตามหลักพื้นฐานที่ใช้ในการจัดการความผิดพลาดจะคล้ายคลึงกัน

กลุ่มการนับเหตุการณ์ที่เกิดขึ้นจากชนิดของความผิดพลาด ประกอบด้วย

Line Coding Violations (LCV) : เหมือนกันกับ DS1/E1

P-bit Coding Violations (PCV) : หมายถึง เหตุการณ์ความผิดพลาดที่ละเมิดการเข้ารหัส (Coding Violation Error) คือ P-Bit พาริตี ที่เกิดขึ้นจากการรับรหัส P-Bit บน M-Frame ที่ระบุไม่ตรงกันกับรหัสท้องถิ่นที่คำนวณได้

C-bit Coding Violations (CCV) : หมายถึง เหตุการณ์ความผิดพลาด C-Bit พาริตี และ SYNTRAN ที่ละเมิดการเข้ารหัส

Framing Errors : เหมือนกันกับ DS1/E1

กลุ่มการนับช่วงเวลาที่เกิดขึ้นจากชนิดของความผิดพลาด ประกอบด้วย

Line Errored Seconds (LES) : เหมือนกันกับ DS1/E1

P-bit Errored Seconds (PES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ PCV เกิดขึ้น

C-bit Errored Seconds (CES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ CCV เกิดขึ้น

กลุ่มการนับช่วงเวลาที่เกิดความผิดพลาดที่รุนแรง (Severely)

Severely Errored Framing Seconds (SEFS) : เหมือนกันกับ DS1/E1

P-bit Severely Errored Seconds (PSES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ PCV เกิดขึ้น
มากเกินไป

C-bit Severely Errored Seconds (CSES) : หมายถึง ช่วงเวลาเป็นวินาที ที่ CCV เกิดขึ้น
มากเกินไป

Unavailable Seconds (UAS):หมายถึง ช่วงเวลาเป็นวินาที ที่สายไม่สามารถใช้ได้ เกิดจาก

- การเกิด PSES ติดต่อกัน 10 ครั้ง
- การเกิดเงื่อนไขความผิดพลาดการจัดเฟรมบ่อย ๆ
- การไม่เกิด PSES ติดต่อกัน 10 ครั้ง
- เงื่อนไขความผิดพลาดการจัดเฟรมถูกเคลียร์

● การทดสอบสาย (Line Testing)

กำหนดการทดสอบลูปแบ็กไว้ 2 ชนิดคือ

Line Loopback : หมายถึง สัญญาณที่รับจากสายไม่ผ่านอุปกรณ์ (Minimum Penetration)
แต่จะถูกลูกลับไปยังสาย

Payload Loopback : หมายถึง สัญญาณที่รับจากสายถูกลูกลับผ่านอุปกรณ์ (Maximum Penetration) และถูกส่งกลับไปยังสาย

5.2 กลุ่มดีเอสหนึ่ง

กลุ่มดีเอสหนึ่ง [15] มีประโยชน์ในการจัดการสื่อพีสิคัลอินเทอร์เน็ตเฟสของ DS1/E1 ชนิดของ DS1/E1 ถูกกำหนดโดย ifType ในกลุ่มอินเทอร์เน็ตเฟสคือ

ds1 (18) คือ T-1 (1.544 Mbps) ที่ใช้ในกลุ่มอเมริกาเหนือ

e1 (19) คือ T-1 (2.048 Mbps) ที่ใช้ในกลุ่มยุโรป

กลุ่มนี้กำหนดตามเอกสาร RFC 1406 มี OBJECT IDENTIFIER อยู่ที่ {transmission 18}
ออกแบบเชิงตรรกะแบ่งออกเป็น 3 กลุ่มคือ

- กลุ่มปลายด้านใกล้ดีเอสหนึ่ง (DS1 Near End Group) กลุ่มนี้ได้จัดเตรียมสารสนเทศด้านคอนฟิกูเรชันและการนับความผิดพลาด สำหรับ DS1/E1 ที่ปลายทางด้านใกล้

- กลุ่มปลายด้านไกลดีเอสหนึ่ง (DS1 Far End Group) กลุ่มนี้ได้จัดเตรียมสารสนเทศการนับความผิดพลาด สำหรับ DS1/E1 ที่ปลายทางด้านไกล (Remote)

- กลุ่มแฟรคชันแนลดีเอสหนึ่ง (DS1 Fractional Group) กลุ่มนี้ได้จัดเตรียมสารสนเทศในการรวมของช่องสัญญาณ DS0 เป็นลจิกคัลคินเทอร์เฟซเดียว

กลุ่มปลายด้านใกล้ดีเอสหนึ่ง (DS1 Near End Group)

กลุ่มนี้เป็น Mandatory ประกอบด้วย 4 ตารางคือ ตารางคอนฟิกูเรชัน 1 ตาราง และตารางการนับความผิดพลาด 3 ตาราง

ตารางคอนฟิกูเรชันคือ dsx1ConfigTable มี dsx1ConfigEntry รูปแบบแถวมี dsx1Line-Index เป็นครรชนี

INDEX { dsx1LineIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

dx1LineIndex (INTEGER) : หมายถึง การระบุอินเทอร์เฟซของ DS1/E1

dsx1IfIndex (INTEGER) : หมายถึง ค่า IfIndex ของอินเทอร์เฟซ

dsx1TimeElapsed (INTEGER) : หมายถึง ช่วงเวลาที่ล่วงผ่านเป็นวินาที เริ่มตั้งแต่คาบเวลาการวัดความผิดพลาดใน 15 นาทีปัจจุบัน

dsx1ValidIntervals (INTEGER) : หมายถึง จำนวนกลุ่มข้อมูลที่ใช้ได้ของช่วงที่ผ่านมา คำนี้อาจน้อยกว่า 96 นับจากอินเทอร์เฟซมีการออนไลน์เมื่อ 24 ชั่วโมงสุดท้ายที่ผ่านมา คำนี้อจะเป็นจำนวนที่สมบูรณ์ของช่วง 15 นาทีตั้งแต่อินเทอร์เฟซมีการออนไลน์

dsx1LineType (enumerated INTEGER) : หมายถึง รูปแบบการจัดเฟรมที่ใช้ในสาย DS1

- other(1) : ซึ่งถึงไม่มีค่าตามรายการข้างล่าง
- dsx1ESF(2) : ซึ่งถึง Extended Superframe DS1
- dsx1D4(3) : ซึ่งถึง AT&T D4 format DS1
- dsx1E1(4) : ซึ่งถึง CCITT Recommendation G.704 (Table 4a)
- dsx1E1-CRC(5) : ซึ่งถึง CCITT Recommendation G.704 (Table 4b)
- dsx1E1-MF(6) : ซึ่งถึง G.704 (Table 4a) ในส่วน TS16 multiframing
- dsx1E1-CRC-MF(7) : ซึ่งถึง G.704 (Table 4b) ในส่วน TS16 multiframing

dsx1LineCoding (enumerated INTEGER) : หมายถึง ชนิดการระงับรหัสศูนย์ (Zero-Code Suppression) ที่ใช้ในสาย

- dsx1JBZS(1) : ซึ่งถึง Jammed Bit Zero Suppression
- dsx1B8ZS(2) : ซึ่งถึง Binary Eight Zero Substitution
- dsx1HDB3(3) : ซึ่งถึงการเชื่อมโยงสำหรับ E1
- dsx1ZBTSI(4) : ซึ่งถึง Zero Byte Time Slot Interchange
- dsx1AMI(5) : ซึ่งถึงการระงับที่ไม่ใช้รหัสศูนย์

- other(6) : ซึ่งถึงไม่มีค่าตามรายการข้างบน

dsx1SendCode (enumerated INTEGER) : หมายถึง ค่าที่ซึ่งถึงชนิดของรหัสหรือแพทเทิร์นที่ส่งผ่านสาย DS1

- dsx1SendNoCode(1) : ซึ่งถึงส่งข้อมูลปกติหรือข้อมูลที่ถูกลูป
- dsx1SendLineCode(2) : ซึ่งถึงส่งการร้องขอสำหรับไลน์ลูปแบ็ก
- dsx1SendPayloadCode(3) : ซึ่งถึงส่งการร้องขอสำหรับเพย์โหลดลูปแบ็ก
- dsx1SendResetCode(4) : ซึ่งถึงส่งการร้องขอการสิ้นสุดลูปแบ็ก
- dsx1SendQRS(5) : ซึ่งถึงส่งการทดสอบแพทเทิร์น QRS (Quasi Random Signal)

จากที่กล่าวถึง

- dsx1Send511Pattern(6) : ซึ่งถึงส่งการทดสอบแพทเทิร์นตายตัว 511 บิต
- dsx1Send3in24Pattern(7) : ซึ่งถึงส่งการทดสอบแพทเทิร์นตายตัวในชุดเซต 3 บิตจาก 24 บิต
- dsx1SendOtherTestPattern(8) : ซึ่งถึงส่งการทดสอบแพทเทิร์นอื่นที่นอกเหนือ

จากที่กล่าวถึง

dsx1CircuitIdentifier (DisplayString) : หมายถึง ค่าที่บรรจุรายละเอียดเฉพาะของสาย DS1/E1 เพื่อส่งผ่านไปยังวงจรของผู้ค้า วัตถุประสงค์เพื่ออำนวยความสะดวกในการตรวจสอบ

dsx1LoopbackConfig (enumerated INTEGER) : หมายถึง คอนฟิกูเรชันในการลูปแบ็กของ DS1 อินเทอร์เฟซ

- dsx1NoLoop(1) : ซึ่งถึงไม่ลูปแบ็ก
- dsx1PayloadLoop(2) : ซึ่งถึงสัญญาณที่รับจากสายถูกลูปผ่านอุปกรณ์ (Maximum Penetration) และถูกส่งกลับไปยังสาย
- dsx1LineLoop(3) : ซึ่งถึงสัญญาณที่รับจากสายไม่ผ่านอุปกรณ์ (Minimum Penetration) แต่จะถูกลูปกลับไปยังสาย
- dsx1OtherLoop(4) : ซึ่งถึงลูปแบ็กชนิดอื่น

dsx1LineStatus (INTEGER): หมายถึง ค่าที่ซึ่งถึงสถานะสายอินเทอร์เฟซ บรรจุมารสนเทศของการลูปแบ็ก, ความขัดข้อง (Failure) และการรับส่งสัญญาณเตือน (Alarm) ตำแหน่งของบิตคือ

- | | | |
|----|------------------|--|
| 1 | dsx1NoAlarm | : ซึ่งถึงไม่แสดงสัญญาณเตือน |
| 2 | dsx1RcvFarEndLOF | : ซึ่งถึง LOF ที่ปลายด้านไกล (สัญญาณเตือนสีเหลือง) |
| 4 | dsx1XmtFarEndLOF | : ซึ่งถึงการส่ง LOF ที่ปลายด้านใกล้ |
| 8 | dsx1RcvAIS | : ซึ่งถึงการส่ง AIS ที่ปลายด้านไกล |
| 16 | dsx1XmtAIS | : ซึ่งถึงการส่ง AIS ที่ปลายด้านใกล้ |
| 32 | dsx1LossOfFrame | : ซึ่งถึง LOF ที่ปลายด้านใกล้ (สัญญาณเตือนสีแดง) |

- 64 dsx1LossOfSignal : ซึ่งถึงการสูญเสียของสัญญาณที่ปลายด้านใกล้
- 128 dsx1LoopbackState : ซึ่งถึงการลูปที่ปลายด้านใกล้
- 256 dsx1T16AIS : ซึ่งถึง E1 TS16 AIS
- 512 dsx1RcvFarEndLOMF : ซึ่งถึงการส่ง TS16 LOMF ที่ปลายด้านไกล
- 1024 dsx1XmtFarEndLOMF : ซึ่งถึงการส่ง TS16 LOMF ที่ปลายด้านใกล้
- 2048 dsx1RcvTestCode : ซึ่งถึงรหัสทดสอบที่ตรวจพบที่ปลายด้านใกล้
- 4096 dsx1OtherFailure : ซึ่งถึงสถานะสายอื่นที่ไม่ถูกกำหนด

dsx1SignalMode (enumerated INTEGER) : หมายถึง โหมดที่ใช้สำหรับการส่งสัญญาณในสาย (Signalling)

- none(1) : ซึ่งถึงไม่มีการส่งสัญญาณบนช่องสัญญาณ
- robbedBit(2) : ซึ่งถึงการส่งสัญญาณแบบ Robbed Bit
- bitOriented(3) : ซึ่งถึงการส่งสัญญาณรวมกับช่องสัญญาณ E1
- messageOriented(4) : ซึ่งถึงการส่งสัญญาณร่วมช่อง (Common Channel)

dsx1TransmitClockSource(enumerated INTEGER) : หมายถึง แหล่งของการส่งสัญญาณนาฬิกา

- loopTiming(1) : ซึ่งถึงรับสัญญาณนาฬิกาที่กู้คืนจากอินเทอร์เฟซของมันเอง
- localTiming(2) : ซึ่งถึงแหล่งสัญญาณนาฬิกาท้องถิ่นที่ใช้
- throughTiming(3) : ซึ่งถึงรับสัญญาณนาฬิกาที่กู้คืนจากอินเทอร์เฟซอื่น

dsx1Fdl (enumerated INTEGER) : หมายถึง รายละเอียดของบิตแมปที่ใช้ในการเชื่อมโยงข้อมูลที่สะดวก (facility data link)

- other(1) : ซึ่งถึงการใช่โปรโตคอลอื่นนอกเหนือจากที่กล่าวถึงข้างล่าง
- dsx1Ansi-T1-403(2) : ซึ่งถึงการอ้างถึง FDL ตามมาตรฐานของ ANSI
- dsx1Att-54016(4) : ซึ่งถึงการอ้างถึง ESF FDL
- dsx1Fdl-none(8) : ซึ่งถึงอุปกรณ์ไม่ใช่ FDL

ตารางการนับความผิดพลาดของกลุ่มปลายด้านใกล้ดีเอสหนึ่ง ประกอบด้วย 3 ตารางคือ ตาราง dsx1CurrentTable บรรจุการรวบรวมทางสถิติที่เกี่ยวข้องกับการนับช่วง (Interval) 15 นาทีปัจจุบัน มี dsx1CurrentEntry เป็นรูปแบบแถวมี dsx1CurrentIndex เป็นดัชนี

INDEX { dsx1CurrentIndex }

ซึ่งค่า dsx1CurrentIndex คือค่าของ dsx1LineIndex สำหรับ DS1/E1 อินเทอร์เฟซ

ตาราง dsx1IntervalIndex บรรจุการรวบรวมทางสถิติที่เกี่ยวข้องกับการนับช่วง (Interval) 15 นาทีที่ผ่านมาใน 24 ชั่วโมง ซึ่งสามารถมีค่าได้สูงสุด 96 ช่วง มี dsx1IntervalEntry เป็นรูปแบบแถวมี dsx1IntervalIndex, dsx1IntervalNumber เป็นดรรชนี

INDEX { dsx1IntervalIndex, dsx1IntervalNumber }

ซึ่งค่า dsx1IntervalIndex คือค่าของ dsx1LineIndex สำหรับ DS1/E1 อินเทอร์เฟซ ส่วนค่า dsx1IntervalNumber คือจำนวนช่วงมีค่าอยู่ระหว่าง 1 ถึง 96 ค่า 1 คือช่วงที่เกิดขึ้นใกล้ปัจจุบันมากที่สุด และค่า 96 คือช่วงที่เกิดขึ้นไกลปัจจุบันมากที่สุด

ตาราง dsx1TotalTable บรรจุผลรวมทางสถิติที่เกี่ยวข้องกับการนับคาบเวลา 24 ชั่วโมงที่ผ่านมา มี dsx1TotalEntry เป็นรูปแบบแถวมี dsx1TotalIndex เป็นดรรชนี

INDEX { dsx1TotalIndex }

ซึ่งค่า dsx1TotalIndex คือค่าของ dsx1LineIndex สำหรับ DS1/E1 อินเทอร์เฟซ ออบเจกต์ทั้งหมดจากตารางการนับความผิดพลาดของกลุ่มปลายด้านไกลดีเอสหนึ่ง ทั้ง 3 ตารางที่กล่าวข้างต้น จะมีผลของการนับความผิดพลาดที่เหมือนกันซึ่งถูกแบ่งออกเป็น 10 รูปแบบสรุปให้เห็นดังตารางที่ 5.3 คอลัมน์แรก ออบเจกต์ทั้งหมดมี Syntax เป็น Gauge32

กลุ่มปลายด้านไกลดีเอสหนึ่ง (DS1 Far End Group)

กลุ่มนี้เป็น Optional ประกอบด้วย 3 ตารางคือ dsx1FarEndCurrentTable, dsx1FarEndIntervalTable และ dsx1FarEndTotalTable ซึ่ง 3 ตารางนี้ความหมายของการนับความผิดพลาดจะตรงกันกับตารางในกลุ่มปลายด้านไกลดีเอสหนึ่ง ยกเว้นค่า LCV ที่ไม่ได้บรรจุในกลุ่มนี้ ออบเจกต์และความหมายของการนับความผิดพลาดแสดงดังตารางที่ 5.3 คอลัมน์ที่สอง

มีออบเจกต์เพิ่มเติมที่ยังไม่ได้แสดงดังตารางที่ 5.3 ซึ่งจะกล่าวถึงดังต่อไปนี้

ตาราง dsx1FarEndCurrentTable มี dsx1FarEndCurrentEntry เป็นรูปแบบแถวมี dsx1FarEndCurrentIndex เป็นดรรชนี

INDEX { dsx1FarEndCurrentIndex }

ซึ่งค่า dsx1FarEndCurrentIndex คือค่าของ dsx1LineIndex สำหรับ DS1/E1 ด้านไกลอินเทอร์เฟซที่ถูกเชื่อมต่อไปยังปลายด้านไกลของ DS1/E1 อินเทอร์เฟซ

ตารางนี้มีออบเจกต์เพิ่มเติมคือ

dsx1FarEndTimeElapsed (INTEGER) : หมายถึง ช่วงเวลาที่ล่วงผ่านเป็นวินาที เริ่มตั้งแต่คาบเวลาการวัดความผิดพลาดใน 15 นาทีปัจจุบันที่ปลายด้านไกลเริ่มต้น

ตารางที่ 5.3 การนับความผิดพลาดของ DS1/E1

ออบเจกต์กลุ่มปลายด้านใกล้	ออบเจกต์กลุ่มปลายด้านไกล	การนับความผิดพลาด
dsx1CurrentESs dsx1IntervalESs dsx1TotalESs	dsx1FarEndCurrentESs dsx1FarEndIntervalESs dsx1FarEndTotalESs	Errored Seconds (ES)
dsx1CurrentSESs dsx1IntervalSESs dsx1TotalSESs	dsx1FarEndCurrentSESs dsx1FarEndIntervalSESs dsx1FarEndTotalSESs	Severely Errored Seconds (SES)
dsx1CurrentSEFSs dsx1IntervalSEFSs dsx1TotalSEFSs	dsx1FarEndCurrentSEFSs dsx1FarEndIntervalSEFSs dsx1FarEndTotalSEFSs	Severely Errored Framing Seconds (SEFS)
dsx1CurrentUASs dsx1IntervalUASs dsx1TotalUASs	dsx1FarEndCurrentUASs dsx1FarEndIntervalUASs dsx1FarEndTotalUASs	Unavailable Seconds (UAS)
dsx1CurrentCSSs dsx1IntervalCSSs dsx1TotalCSSs	dsx1FarEndCurrentCSSs dsx1FarEndIntervalCSSs dsx1FarEndTotalCSSs	Controlled Slip Seconds (CSS)
dsx1CurrentPCVs dsx1IntervalPCVs dsx1TotalPCVs	dsx1FarEndCurrentPCVs dsx1FarEndIntervalPCVs dsx1FarEndTotalPCVs	Path Coding Violations (PCV)
dsx1CurrentLESs dsx1IntervalLESs dsx1TotalLESs	dsx1FarEndCurrentLESs dsx1FarEndIntervalLESs dsx1FarEndTotalLESs	Line Errored Seconds (LES)
dsx1CurrentBESs dsx1IntervalBESs dsx1TotalBESs	dsx1FarEndCurrentBESs dsx1FarEndIntervalBESs dsx1FarEndTotalBESs	Bursty Errored Seconds (BES)
dsx1CurrentDMs dsx1IntervalDMs dsx1TotalDMs	dsx1FarEndCurrentDMs dsx1FarEndIntervalDMs dsx1FarEndTotalDMs	Degraded Minutes (DM)
dsx1CurrentLCVs dsx1IntervalLCVs dsx1TotalLCVs		Line Coding Violations (LCV)

$dsx1FarEndValidIntervals$ (INTEGER) : หมายถึง จำนวนกลุ่มข้อมูลที่สามารใช้ได้ที่
ปลายด้านไกลของช่วงที่ผ่านมา

ตาราง $dsx1FarEndIntervalTable$ มี $dsx1FarEndIntervalEntry$ เป็นรูปแบบแถวมี $dsx1FarEndIntervalIndex$, $dsx1FarEndIntervalNumber$ เป็นคระรชนี

INDEX { $dsx1FarEndIntervalIndex$, $dsx1FarEndIntervalNumber$ }

ซึ่งค่า $dsx1FarEndIntervalIndex$ คือค่าของ $dsx1LineIndex$ สำหรับ DS1/E1 ด้านไกลคอล
อินเทอร์เฟซที่ถูกเชื่อมต่อไปยังปลายด้านไกลของ DS1/E1 อินเทอร์เฟซ ส่วน $dsx1FarEndInterval-$
 $Number$ คือจำนวนช่วง มีค่าอยู่ระหว่าง 1 ถึง 96 ค่า 1 คือช่วงที่เกิดขึ้นใกล้ปัจจุบันมากที่สุด และค่า
96 คือช่วงที่เกิดขึ้นไกลปัจจุบันมากที่สุด

ตาราง $dsx1FarEndTotalTable$ มี $dsx1FarEndTotalEntry$ เป็นรูปแบบแถวมี $dsx1Far-$
 $EndTotalIndex$ เป็นคระรชนี

INDEX { $dsx1FarEndTotalIndex$ }

ซึ่งค่า $dsx1FarEndTotalIndex$ คือค่าของ $dsx1LineIndex$ สำหรับ DS1/E1 ด้านไกลคอล
อินเทอร์เฟซที่ถูกเชื่อมต่อไปยังปลายด้านไกลของ DS1/E1 อินเทอร์เฟซ

กลุ่มแฟรคชันแนลคือเอสหนึ่ง (DS1 Fractional Group)

กลุ่มนี้ใช้สร้างลอจิคัลอินเทอร์เฟซเดียว ที่ได้จากการรวมกันของช่องสัญญาณ DS0 บน
สาย DS1/E1 เพื่อให้ได้แบนด์วิดท์ที่กว้างขึ้นในการบรรจุกระแสข้อมูล (Data Stream) ต่าง ๆ ที่เป็น
ประโยชน์ เช่น ข้อมูลภาพ และเสียง

กลุ่มนี้ประกอบด้วย 1 ตารางคือ $dsx1FracTable$ มี $dsx1FracEntry$ เป็นรูปแบบแถวมี
 $dsx1FracIndex$, $dsx1FracNumber$ เป็นคระรชนี

INDEX { $dsx1FracIndex$, $dsx1FracNumber$ }

ซึ่งค่า $dsx1FracIndex$ คือค่าของ $dsx1LineIndex$ สำหรับ DS1/E1 อินเทอร์เฟซ ส่วน
 $dsx1FracNumber$ คือจำนวนช่องสัญญาณภายใน DS1/E1 อินเทอร์เฟซ

ภายในตารางบรรจุออบเจกต์ 1 ออบเจกต์คือ

$dsx1FracIfIndex$ (INTEGER) : หมายถึง การระบุลอจิคัลอินเทอร์เฟซที่บรรจุในช่อง
สัญญาณโดยค่าของ $ifIndex$ ถ้าค่าศูนย์แสดงว่าไม่มีลอจิคัลอินเทอร์เฟซขณะนั้น และถ้าการเข้า
ครอบครองอินเทอร์เฟซหนึ่งใช้เวลามากกว่าหนึ่งโทมส์ลือต ค่าของ $ifIndex$ ก็จะค้นหาให้หลาย
โทมส์ลือต

5.3 กลุ่มดีเอสสาม

กลุ่มดีเอสสาม [16] มีโครงสร้างและความหมายของคำที่เหมือนกับกลุ่ม DS1/E1 ยกเว้น ออบเจ็กต์เพิ่มเติมในกลุ่มปลายด้านไกลดีเอสสาม และการใช้พารามิเตอร์คอนฟิกูเรชัน, สถานะสาย (Line Status) รวมถึงการนับความผิดพลาด (Error Count)

ชนิดของ DS3/E3 ถูกกำหนดโดย ifType ในกลุ่มอินเทอร์เฟซ

ds3 (30) คือ T-3 (44.736 Mbps)

กลุ่มนี้กำหนดตามเอกสาร RFC 1407 มี OBJECT IDENTIFIER อยู่ที่ {transmission 30} ออบเจ็กต์ถูกแบ่งออกเป็น 3 กลุ่มคือ

- กลุ่มปลายด้านใกล้ดีเอสสาม (DS3 Near End Group)
- กลุ่มปลายด้านไกลดีเอสสาม (DS3 Far End Group)
- กลุ่มแฟรคชันแนลดีเอสสาม (DS3 Fractional Group)

กลุ่มปลายด้านใกล้ดีเอสสาม/อีสาม (DS3/E3 Near End Group)

กลุ่มนี้เป็น Mandatory ของทุกระบบที่เชื่อมต่อไปยังอินเทอร์เฟซ DS3/E3 กลุ่มประกอบด้วย 4 ตาราง คือตารางคอนฟิกูเรชัน 1 ตาราง และตารางการนับความผิดพลาด 3 ตารางดังนี้

ตารางคอนฟิกูเรชันคือ dsx3ConfigTable มี dsx3ConfigEntry รูปแบบแถวมี dsx3Line-Index เป็นดรรชนี

INDEX { dsx3LineIndex }

ภายในตารางประกอบด้วยออบเจ็กต์คือ

dsx3LineIndex (INTEGER) : หมายถึง การระบุอินเทอร์เฟซของ DS3/E3

dsx3IfIndex (INTEGER) : เหมือนกับ dsx1IfIndex ของ DS1/E1

dsx3TimeElapsed (INTEGER) : เหมือนกับ dsx1TimeElapsed ของ DS1/E1

dsx3ValidIntervals (INTEGER) : เหมือนกับ dsx1IValidIntervals ของ DS1/E1

dsx3LineType (INTEGER) : หมายถึง ชนิดอินเทอร์เฟซที่ประยุกต์ใช้บนสาย DS3 C-Bit หรือ E3 มีผลในการแปลสถิติการใช้และสถิติความผิดพลาด

- dsx3other(1) : ซี่ถึง DS3 อื่นนอกเหนือจากที่กล่าว
- dsx3M23(2) : ซี่ถึง ANSI T1.107-1988
- dsx3SYNTRAN(3) : ซี่ถึง ANSI T1.107-1988
- dsx3CbitParity(4) : ซี่ถึง ANSI T1.107a-1989
- dsx3ClearChannel(5) : ซี่ถึง ANSI T1.102-1987
- e3other(6) : ซี่ถึง E3 อื่นนอกเหนือจากที่กล่าว

- e3Framed(7) : ซึ่งถึง CCITT G.751
- e3Plcp(8) : ซึ่งถึง ETSI T/NA(91)18

dsx3LineCoding(INTEGER): หมายถึง ชนิดการระงับรหัสศูนย์ (Zero-Code Suppression)

ที่ใช้ในสาย

- dsx3Other (1) : ซึ่งถึงรูปแบบอื่นนอกเหนือจากที่กล่าว
- dsx3B3ZS (2) : ซึ่งถึง B3ZS บนสาย DS3
- e3HDB3 (3) : ซึ่งถึง HDB3 บนสาย E3

dsx3SendCode (INTEGER) : หมายถึง ค่าที่ซึ่งถึงชนิดรหัสหรือแพทเทิร์นที่ส่งผ่านสาย

DS3/E3

- dsx3SendNoCode (1) : ซึ่งถึงส่งข้อมูลปกติหรือข้อมูลที่ถูกลบ
- dsx3SendLineCode (2) : ซึ่งถึงส่งการร้องขอสำหรับ โลนั้ลูปแบ้ก
- dsx3SendPayloadCode (3) : ซึ่งถึงส่งการร้องขอเพ็้โหลดลูปแบ้ก
- dsx3SendResetCode (4) : ซึ่งถึงส่งการร้องขอสิ้นสุดลูปแบ้ก
- dsx3SendDS1LoopCode (5) : ซึ่งถึงการร้องขอของสัญญาณพิเศษสำหรับการ
- dsx3SendTestPattern (6) : ซึ่งถึงการส่งการทดสอบแพทเทิร์น

ลูปแบ้ก

dsx3CircuitIdentifier (DisplayString) : เหมือนกับ dsx1CircuitIdentifier ของ DS1/E1

dsx3LoopbackConfig (INTEGER) : เหมือนกับ dsx1LoopbackConfig ของ DS1/E1

dsx3LineStatus (INTEGER) : หมายถึง ค่าที่ซึ่งถึงสถานะสายของอินเทอร์เฟซ ซึ่งบรรจุสารสนเทศการลูปแบ้ก, ความขัดข้อง (Failure) และการรับส่งสัญญาณเตือน (Alarm) ตำแหน่งของบิตต่าง ๆ คือ

- | | | |
|-----|-------------------|---|
| 1 | dsx3NoAlarm | : ซึ่งถึง ไม่แสดงสัญญาณเตือน |
| 2 | dsx3RcvRAIFailure | : ซึ่งถึง การรับสัญญาณเตือนสีเหลืองด้านรีโมท |
| 4 | dsx3XmitRAIAlarm | : ซึ่งถึง การส่งสัญญาณเตือนสีเหลืองด้านรีโมท |
| 8 | dsx3RcvAIS | : ซึ่งถึง การรับสถานะที่ขัดข้องของ AIS ที่ปลายด้านไกล |
| 16 | dsx3XmitAIS | : ซึ่งถึง การส่ง AIS ที่ปลายด้านใกล้ |
| 32 | dsx3LOF | : ซึ่งถึง การรับสถานะที่ขัดข้องของ LOF ที่ปลายด้านไกล |
| 64 | dsx3LOS | : ซึ่งถึง การรับสถานะที่ขัดข้องของ LOS ที่ปลายด้านไกล |
| 128 | dsx3LoopbackState | : ซึ่งถึง การลูปที่ปลายด้านใกล้ |
| 256 | dsx3RcvTestCode | : ซึ่งถึง การรับแพทเทิร์นทดสอบที่ปลายด้านใกล้ |
| 512 | dsx3OtherFailure | : ซึ่งถึง สถานะสายอื่นที่ไม่ถูกกำหนด |

dsx3TransmitClockSource (INTEGER) : เหมือนกับ dsx1TransmitClockSource ของ DS1/E1

ตารางการนับความผิดพลาดของกลุ่มปลายด้านไกลดีเอสสาม/อิสสาม (DS3/E3 Near End Error Count) ประกอบด้วย 3 ตารางคือ dsx3CurrentTable, dsx3IntervalTable และ dsx3TotalTable
 ออบเจกต์ทั้งหมดจากตารางการนับความผิดพลาดของกลุ่มปลายด้านไกลดีเอสสาม ทั้ง 3 ตารางมีความหมายที่เหมือนกันเทียบเท่ากับ DS1/E1 ยกเว้นผลการนับความผิดพลาดที่แตกต่างกัน
 สรุปให้เห็นดังตารางที่ 5.4 คอลัมน์แรก ออบเจกต์ทั้งหมดของการนับความผิดพลาดมี Syntax เป็น Gauge32

กลุ่มปลายด้านไกลดีเอสสาม/อิสสาม (DS3/E3 Far End Group)

กลุ่มนี้เป็น Optional ของทุกระบบที่เชื่อมต่อไปยังอินเทอร์เน็ตเฟส DS3/E3 กลุ่มประกอบด้วย 4 ตาราง คือตารางคอนฟิกูเรชัน 1 ตาราง และตารางการนับความผิดพลาด 3 ตารางคือ

ตารางคอนฟิกูเรชันคือ dsx3FarEndConfigTable บรรจุสารสนเทศของอุปกรณ์โกลบอลที่ส่งในพาธการกำหนดชื่อข่าวสาร (Path Identification Messages) มี dsx3FarEndConfigEntry เป็นรูปแบบแถวมี dsx3FarEndLineIndex เป็นคอลัมน์

INDEX { dsx3FarEndLineIndex }

ภายในตารางประกอบด้วยออบเจกต์คือ

dsx3FarEndLineIndex (INTEGER) : หมายถึง ค่าของ dsx3LineIndex ที่ใช้ระบุ DS1/E1 อินเทอร์เน็ตเฟส

dsx3FarEndEquipCode(DisplayString): หมายถึง รหัสการกำหนดชื่ออุปกรณ์ที่ปลายด้านไกล (Equipment Identification) เป็นรายละเอียดจำเพาะแต่ละชนิดของอุปกรณ์

dsx3FarEndLocationIDCode (DisplayString): หมายถึง รหัสการกำหนดตำแหน่งที่ปลายด้านไกล (Location Identification) เป็นรายละเอียดจำเพาะตำแหน่งของอุปกรณ์

dsx3FarEndFrameIDCode (DisplayString) : หมายถึง รหัสการกำหนดเฟรมที่ปลายด้านไกล (Frame Identification) ใช้ระบุการวางตำแหน่งของอุปกรณ์ภายในคิก

dsx3FarEndUnitCode (DisplayString) : หมายถึง รหัสปลายด้านไกลใช้ระบุตำแหน่งอุปกรณ์ภายในเบย์

dsx3FarEndFacilityIDCode (DisplayString) : หมายถึง รหัสจำเพาะที่ใช้ระบุพาธที่ปลายด้านไกลดีเอสสาม

ตารางการนับความผิดพลาดของกลุ่มปลายด้านไกลดีเอสสาม/อิสสาม (DS3/E3 Near End Error Count) ประกอบด้วย 3 ตารางคือ dsx3FarEndCurrentTable, dsx3FarEndIntervalTable และ dsx3FarEndTotalTable

ตารางที่ 5.4 การนับความผิดพลาดของ DS3/E3

ออบเจกต์กลุ่มปลายด้านใกล้	ออบเจกต์กลุ่มปลายด้านไกล	การนับความผิดพลาด
dsx3CurrentPESs dsx3IntervalPESs dsx3TotalPESs		P-bit Errored Seconds (PES)
dsx3CurrentPSESs dsx3IntervalPSESs dsx3TotalPSESs		P-bit Severely Errored Seconds (PSES)
dsx3CurrentSEFSs dsx3IntervalSEFSs dsx3TotalSEFSs		Severely Errored Framing Seconds (SEFS)
dsx3CurrentUASs dsx3IntervalUASs dsx3TotalUASs	dsx3FarEndCurrentUASs dsx3FarEndIntervalUASs dsx3FarEndTotalUASs	Unavailable Seconds (UAS)
dsx3CurrentLCVs dsx3IntervalLCVs dsx3TotalLCVs		Line Coding Violations (LCV)
dsx3CurrentPCVs dsx3IntervalPCVs dsx3TotalPCVs		P-bit Coding Violations (PCV)
dsx3CurrentLESs dsx3IntervalLESs dsx3TotalLESs		Line Errored Seconds (LES)
dsx3CurrentCCVs dsx3IntervalCCVs dsx3TotalCCVs	dsx3FarEndCurrentCCVs dsx3FarEndIntervalCCVs dsx3FarEndTotalCCVs	C-bit Coding Violations (CCV)
dsx3CurrentCESs dsx3IntervalCESs dsx3TotalCESs	dsx3FarEndCurrentCESs dsx3FarEndIntervalCESs dsx3FarEndTotalCESs	C-bit Errored Seconds (CES)
dsx3CurrentCSESs dsx3IntervalCSESs dsx3TotalCSESs	dsx3FarEndCurrentCSESs dsx3FarEndIntervalCSESs dsx3FarEndTotalCSESs	C-bit Severely Errored Seconds (CSES)

ออบเจกต์ทั้งหมดจากตารางการนับความผิดพลาดของกลุ่มปลายด้านไกลดีเอสสาม ทั้ง 3 ตารางมีความหมายที่เหมือนกันเทียบกับ DS1/E1 ยกเว้นผลการนับความผิดพลาดที่แตกต่างกัน สรุปให้เห็นดังตาราง 5.4 คอลัมน์ที่สอง ออบเจกต์ทั้งหมดของการนับความผิดพลาดมี Syntax เป็น Gauge32

กลุ่มแฟรคชันแนลดีเอสสาม (DS3 Fractional Group)

กลุ่มประกอบด้วย 1 ตารางคือ dsx3FracTable ใช้สำหรับระบุช่องสัญญาณ DS3/E3 ที่ถูกรวมกันเพื่อสร้างเป็นลจิสต์อินเทอร์เฟซด้านโลกอล กลุ่มนี้มีความหมายของค่าและ Syntax เทียบเท่ากับกลุ่มแฟรคชันแนลดีเอสหนึ่ง



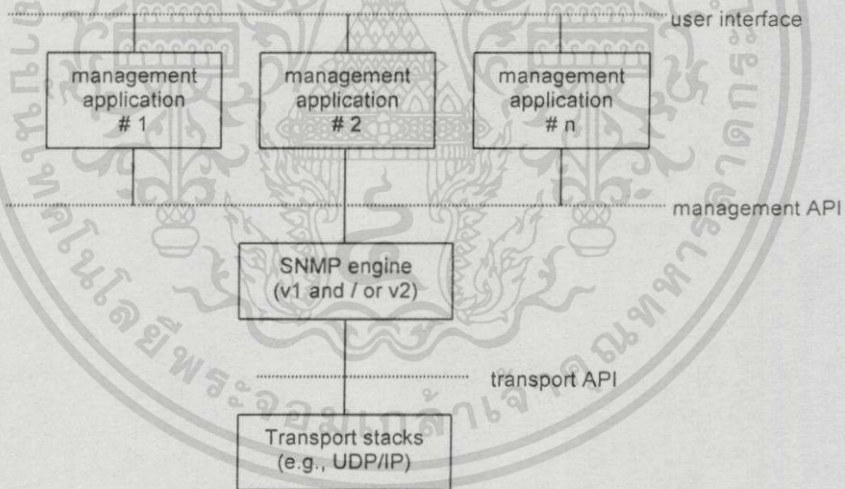
บทที่ 6

การพัฒนาโปรแกรม

การพัฒนาซอฟต์แวร์จัดการเครือข่าย โดยนำหลักการของเอสเอ็นเอ็มพีมาประยุกต์ใช้ จะกระทำการในส่วนของการจัดการเครือข่าย ซึ่งจำเป็นต้องทำความเข้าใจพื้นฐานระบบการทำงานภายในสถานีจัดการเครือข่าย ดังที่จะกล่าวถึงต่อไปนี้

6.1 สถานีจัดการเครือข่าย (Network Management Station)

สถานีจัดการเครือข่าย หรือ เอ็นเอ็มเอส โดยปกติจะเป็นเวิร์กสเตชันหนึ่งในระบบ ที่ดำเนินการรันโปรโตคอลจัดการเครือข่าย (Network Management Protocol) และ แอปพลิเคชันจัดการเครือข่าย (Network Management Applications) ซึ่งโปรโตคอลจัดการเครือข่ายจะเป็นการจัดเตรียมกลไกสำหรับการจัดการ ส่วนแอปพลิเคชันจัดการเครือข่ายจะเป็นการกำหนดนโยบายที่ใช้สำหรับการจัดการ



รูปที่ 6.1 แสดงแบบจำลองของสถานีจัดการเครือข่าย

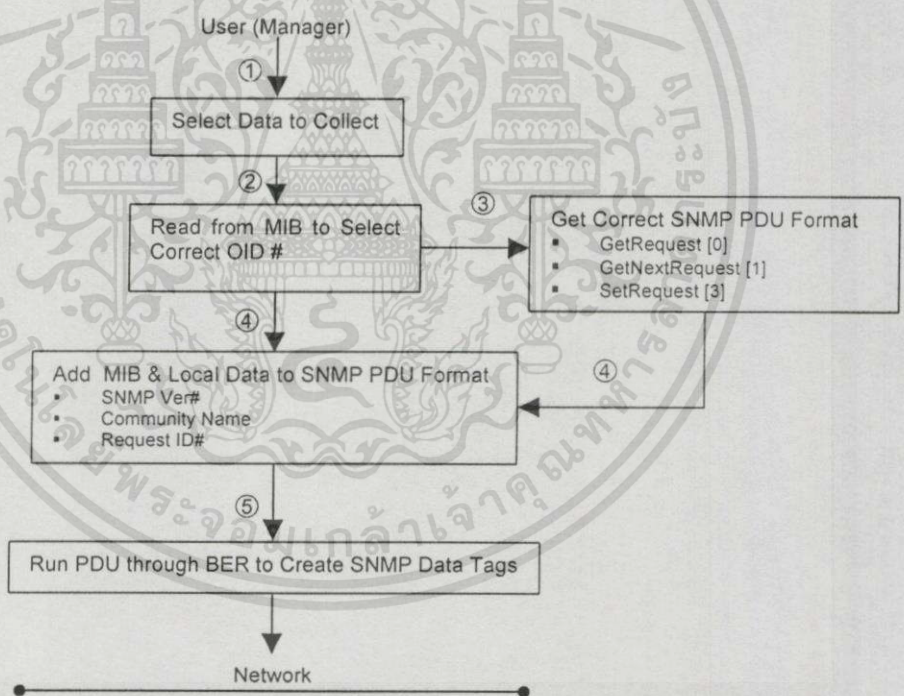
แบบจำลองของสถานีจัดการเครือข่าย [6] แสดงดังรูปที่ 6.1 ส่วนของยูสเซอร์อินเทอร์เฟซเป็นส่วนซอฟต์แวร์ที่พัฒนาขึ้นมาเพื่อแสดงผลที่คอนโซลเวิร์กสเตชันของผู้ใช้ปลายทาง ไปยังแอปพลิเคชันจัดการเครือข่าย ซอฟต์แวร์ส่วนยูสเซอร์อินเทอร์เฟซนี้จะเตรียมความสามารถในการส่งคำสั่งไปยังอุปกรณ์เครือข่าย เพื่อรับข้อมูลในการทำรายงาน หรือเตรียมการแสดงผลต่อไป โดยส่วนมากซอฟต์แวร์จะถูกเขียนให้แสดงผลในรูปแบบของ GUI (Graphical User Interface) ส่วนแอปพลิเคชันจัดการเครือข่าย เป็นซอฟต์แวร์ที่ใช้ได้ต่อกันกับเครื่องประมวลผลเอสเอ็นเอ็มพี

(SNMP Engine) เพื่อนำข้อมูลที่เป็นประโยชน์ที่ได้จากการอ่านฐานข้อมูลของเอ็มไอบี ส่งต่อไปยังยูสเซอร์อินเทอร์เน็ตเฟส ซอฟต์แวร์ในส่วนนี้จะเตรียมความสามารถในการโพลล์ (Poll) ไปยังเอเจนต์เตรียมความสามารถในการ get request และ set request รวมถึงการจัดการกับ Trap Message ส่วนของแอปพลิเคชันสามารถมีได้หลายแอปพลิเคชัน แต่ละแอปพลิเคชันจะมีการเก็บและค้นคืนข้อมูลที่มีรูปแบบเฉพาะตัว ซึ่งส่งผลให้แอปพลิเคชันอื่นที่มี API (Application Programming Interface) แตกต่างกันไปไม่สามารถที่จะอ่านข้อมูลได้ ทำให้ต้องมีการกำหนดชนิดข้อมูลให้อยู่ในรูปแบบทั่วไปผ่านความเข้ากันได้ของ Management API เพื่อให้ทุกแอปพลิเคชันสามารถที่จะทำการเก็บ, ค้นคืนหรือสอบถามข้อมูล เพื่อเข้าถึงข่าวสารของออบเจกต์ได้

6.2 การร้องขอของเอ็นเอ็มเอส (NMS Request)

เมื่อยูสเซอร์มีการร้องขอข้อมูลผ่านโพรโตคอลเอสเอ็นเอ็มพี จะเกิดกิจกรรมต่าง ๆ ดังรูปที่

6.2 แสดงขั้นตอนการร้องขอข้อมูลของเอ็นเอ็มเอส [17]



รูปที่ 6.2 แสดงขั้นตอนการร้องขอข้อมูล

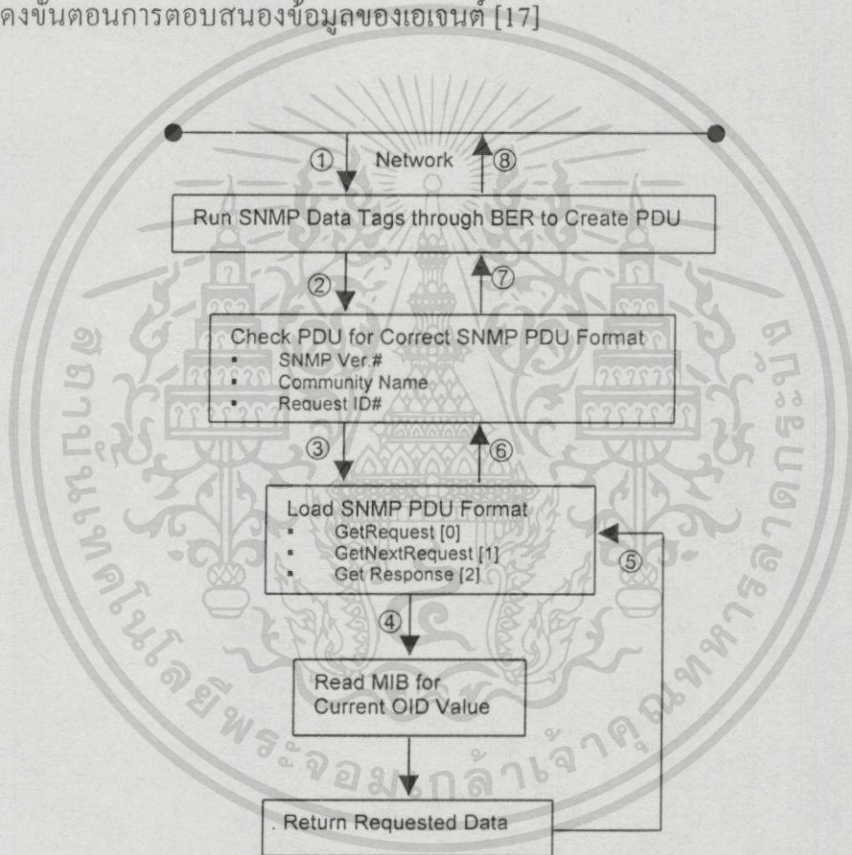
จากรูปที่ 6.2 ยูสเซอร์สั่งการผ่านยูสเซอร์อินเทอร์เน็ตเฟส เพื่อเลือกข้อมูลที่ต้องการร้องขอจากเอ็มไอบี จากนั้นเอ็นเอ็มเอสจะอ่านค่าจากเอ็มไอบีเพื่อเลือกค่าที่ถูกต้องของ OID แล้วเก็บเข้าไปเป็นข้อมูลพื้นฐาน อีกด้านหนึ่งจะเลือกรูปแบบของพิดู (PDU : Protocol Data Unit) ที่ถูกต้องเพื่อใช้จัดการกับชนิดข้อมูลที่ถูกร้องขอ จากนั้นเอ็นเอ็มเอสจะทำการรวมรูปแบบของพิดู และรูปแบบข้อมูลพื้นฐานเข้าเป็นโครงสร้างของเอสเอ็นเอ็มพีเมสเสจ เอสเอ็นเอ็มพีเมสเสจขณะนี้สามารถอ่านได้บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลีกราฟแพลตฟอร์มนี้เท่านั้น ไม่สามารถนำไปอ่านบนแพลตฟอร์มอื่นได้ ซึ่งจะต้องมีการปรับเพื่อให้แพลตฟอร์มอื่นอ่านได้ โดยเอสเอ็นเอ็มพีจะทำการเข้ารหัสเมสเสจให้อยู่ในรูปแบบของ ASN.1 (Abstract Syntax Notation One) ตามกฎการเข้ารหัสพื้นฐาน หรือ BER (Basic Encoding Rules) เพื่อสร้างเป็นเอสเอ็นเอ็มพีแท็ก ที่เรียกว่าโครงสร้าง TLV (Tag, Length, Value) จากนั้นเมสเสจนี้ก็จะถูกส่งลงไปยังยูติลิตี้/ไอพี และชั้นเน็ตเวิร์กแอกเซสผ่านเครือข่ายไปยังเอเจนต์

6.3 การตอบสนองของเอเจนต์ (Agent Response)

การรับเอสเอ็นเอ็มพีเมสเสจผ่านการตอบสนองของเอเจนต์ จะเกิดกิจกรรมต่าง ๆ ขึ้นดังรูปที่ 6.3 แสดงขั้นตอนการตอบสนองข้อมูลของเอเจนต์ [17]



รูปที่ 6.3 แสดงขั้นตอนการตอบสนองข้อมูล

หลังจากเอเจนต์รับการร้องขอข้อมูลจากเอ็นเอ็มเอส ผ่านเครือข่ายชั้นเน็ตเวิร์กแอกเซสและยูติลิตี้/ไอพีขึ้นมาแล้ว เอสเอ็นเอ็มพีแท็กที่ถูกเข้ารหัสจะถูกดำเนินการ โดยผ่านกระบวนการ ASN.1 BER เพื่อแปลงค่ากลับมาเป็นเอสเอ็นเอ็มพีเมสเสจสำหรับใช้บนโพลีกราฟเอเจนต์ จากนั้นเอเจนต์จะทำการแยกโครงสร้างรูปแบบของพีดียู และรูปแบบข้อมูลพื้นฐานออกจากกัน ขั้นตอนนี้จะมีการตรวจทานอินฟอร์เมชันของเอสเอ็นเอ็มพี (เวอร์ชัน, คอมมิวนิตี้นาม, หมายเลขการร้องขอ) และเก็บอินฟอร์เมชันเหล่านี้ไว้เพื่อใช้เปรียบเทียบในขั้นตอนการตอบสนอง จากนั้นเอเจนต์จะทำการแยก

ฟิลด์ข้อมูลของพิดูที่มีกร็องขอ และอ่านค่าฐานข้อมูลในเอ็มไอบีเพื่อรับค่าในแต่ละ OID จากนั้น เอเจนต์จะสร้างกระบวนการตอบสนอง โดยใช้รูปแบบพิดูที่เหมาะสมคือ Get Response พร้อมกับเปรียบเทียบค่าของอินฟอร์เมชันที่เก็บไว้ก่อนหน้านั้น ถ้าตรงกันก็จะทำการสร้างเป็นเอสเอ็นเอ็มพีเมสเสจ จากนั้นเมสเสจนี้ก็จะถูกแปลงให้อยู่ในรูปแบบของ ASN.1 ผ่านกระบวนการเข้ารหัส BER เพื่อสร้างเป็นเอสเอ็นเอ็มพีแพ็กเก็ต หรือโครงสร้าง TLV ส่งผ่านเครือข่ายกลับไปยังเอ็นเอ็มเอส

6.4 การพัฒนาโปรแกรม

โปรแกรมนี้ทำการพัฒนาด้วยภาษา Tcl/Tk (Tool Command Language / Toolkit) เวอร์ชัน 8.2 [18] ซึ่งเป็นภาษาสคริปต์รันภายใต้ระบบปฏิบัติการวินโดวส์ 95 หรือวินโดวส์ 98 และใช้ซอฟต์แวร์ Scotty Version 3.0.0 [19] เป็นเครื่องมือช่วยในการพัฒนา

Scotty เป็นซอฟต์แวร์แพคเกจ ที่ช่วยในการพัฒนาซอฟต์แวร์จัดการเครือข่ายโดยเฉพาะ ซอฟต์แวร์นี้จะทำงานภายใต้ Tcl/Tk มีส่วนประกอบหลัก 2 ส่วนคือ Tnm Tcl Extension ซึ่งจัดการเตรียมการเข้าถึงแหล่งสารสนเทศการจัดการเครือข่าย สนับสนุนหลายโปรโตคอล เช่น SNMPv1, SNMPv2c, SNMPv2u, ICMP, DNS, HTTP, UDP และ SUN RPC ส่วนที่สองคือ Tkined ซึ่งเป็นเอดิเตอร์เครือข่ายเชิงโต้ตอบ มีความสามารถในการสร้าง, การเพิ่ม, การลบ หรือแก้ไขคุณสมบัติของอุปกรณ์บนระบบเครือข่าย และยังทำการบำรุงรักษาแผนผังเครือข่าย (Network Layout)

การพัฒนาซอฟต์แวร์จัดการเครือข่ายโดยใช้ซอฟต์แวร์ Scotty นั้นควรจะต้องทำการศึกษาคำสั่งที่สำคัญภายในแพคเกจ Tnm คือคำสั่ง snmp ใช้สำหรับส่ง SNMP messages ไปยังอุปกรณ์ในระบบเครือข่าย และคำสั่ง mib ใช้สำหรับการโหลดและสอบถามไปยัง SNMP MIB

Tnm 3.0.0 for win32 จะทำงานได้จะต้องทำการติดตั้งซอฟต์แวร์ Tcl 8.2 for win32 ก่อน และไม่สามารถทำงานได้กับ Tcl เวอร์ชันก่อนหน้านี การเรียกใช้คำสั่งของ Tnm จะต้องมีการโหลด Tnm package และอิมพอร์ต namespace เสียก่อน ดังนี้

```
package require Tnm 3.0
namespace import Tnm::*
```

6.4.1 คำสั่ง SNMP

ภายในคำสั่ง SNMP [20] จะมีกลุ่มคำสั่งย่อยที่ถูกเรียกใช้บ่อย ๆ ดังนี้

- การจัดการ SNMP Session

การรับส่งทุก SNMP messages จะเรียกว่าเซสชัน (Session) ในการสร้างเซสชันใหม่เพื่อทำการร้องขอ SNMP จะใช้คำสั่ง “generator” สำหรับเริ่มต้นในการ get, getnext, getbulk หรือ set

snmp generator

⇒ snmp0

เซสชัน generator มีทางเลือกคือ

-address, ไอพีแอดเดรสที่ SNMP ส่งการร้องขอ ค่าอาจเขียนอยู่ในรูปแบบคอด หรือชื่อโฮสต์ ซึ่งสามารถแปลงเป็นไอพีแอดเดรส (ดีพอลต์ 127.0.0.1)

-port , หมายเลขพอร์ต UDP ที่ SNMP ส่งการร้องขอ ค่าอาจเขียนเป็นหมายเลขพอร์ต หรือชื่อบริการ ซึ่งสามารถแปลงเป็นหมายเลขพอร์ต (ดีพอลต์ 161)

-version, เวอร์ชันของ SNMP ที่ใช้ (ดีพอลต์ SNMPv1)

-community, อ่านคอมมิวนิตีส์ตริง ใช้เมื่อมีการอ้างการทำงานของ SNMP (ดีพอลต์ public)

-timeout, เวลาไทม์เอาต์ที่เซสชันรอการตอบสนอง มีหน่วยเป็นวินาที (ดีพอลต์ 5 วินาที)

-retries, จำนวนครั้งที่มีการส่งใหม่ในช่วงเวลาไทม์เอาต์ (ดีพอลต์ 3 ครั้ง)

-delay, การหน่วงเวลาระหว่างการส่ง 2 เมสเซจติดต่อกัน มีหน่วยเป็นมิลลิวินาที (ดีพอลต์ 0 มิลลิวินาที)

-windows, กำหนดจำนวนการร้องขออะซิงโครนัส (อะซิงโครนัส หมายถึงการร้องขอที่ไม่ต้องรอคอยการตอบสนอง) ซึ่งคิวการร้องขอจะต้องไม่มากเกินไปกว่าจำนวนการร้องขออะซิงโครนัสบนสายสัญญาณ (ดีพอลต์ 10)

-alias, การสมนามชื่อ ใช้เพื่อกำหนดชื่อคอนฟิกูเรชันของเซสชันที่ยาวให้สั้นลง สามารถนำไปอ้างอิงใช้ในเซสชันอื่น

-tags, รายการค่าของแท็กสามารถใช้ระบุเซสชัน หรือใช้ค้นหาเซสชันทั้งหมดที่มีค่าของแท็กพร้อมอยู่

ตัวอย่าง การสร้าง SNMP เซสชันเพื่อทำการร้องขอไปยัง ไอพีแอดเดรส 192.168.2.13 ใช้การอ่านคอมมิวนิตีส์ตริง xpublic

```
set s [snmp generator -address 192.168.2.13 -community xpublic ]
```

⇒ snmp1

จากคำสั่งข้างต้นจะเป็นการสร้างชื่อของเซสชันใหม่ และเซสชันใหม่ที่ถูกสร้างสามารถที่จะแสดงค่าคอนฟิกูเรชัน ด้วยคำสั่ง

```
$s configure
```

```
⇒ -address 192.168.2.13 -port 161 -version SNMPv1 -community xpublic
transport udp -timeout 5 -retries 3 -window 10 -delay 0 -tags {}
```

คำสั่งการรีเทรินเพียงค่าเดียว และคำสั่งการเปลี่ยนแปลงคอนฟิกูเรชันของเซสชัน

```
$s cget -address
```

⇒ -address 192.168.2.13

```
$s configure -delay 10 -window 30
```

```
⇒ -address 192.168.2.13 -port 161 -version SNMPv1 -community xpublic
   -transport udp -timeout 5 -retries 3 -window 30 -delay 10 -tags {}
```

การเปลี่ยนแปลงคอนฟิกูเรชันของเซสชัน อาจจะทำให้การร้องขอ SNMP ที่ค้างอยู่เกิดการเปลี่ยนแปลง ดังนั้นการจัดการเซสชันที่สร้างใหม่สำหรับทุกไอพีแอดเดรสและคอมมิวนิตีส์สตริง อาจทำได้ตามรูทีน ดังนี้

```
proc getSession { addr rcomm } {
    set s [snmp find -tags " $addr , $rcomm " ]
    if { [ string length $s ] > 0 } {return [ lindex $s 0 ]}
    return [ snmp generator -address $addr \
              -community $rcomm \
              -tags " $addr , $rcomm " ]
}
proc destroySession { addr rcomm } {
    set list [snmp find -tags " $addr , $rcomm " ]
    foreach job $list {
        $job destroy
    }
}
```

คำสั่งการลบค่าทั้งหมดออกจากเซสชันหรือการยกเลิกค่าที่ค้างในการร้องขออะซิงโครนัส

```
$job destroy ; # job คือชื่อเซสชันที่ทำกรลบค่า
```

จากรูทีนข้างต้นสามารถสร้างเซสชันใหม่ โดยการกำหนดค่าไอพีแอดเดรสและคอมมิวนิตีส์สตริงที่ต้องการ

```
set s [ getSession 192.168.2.1 sun_public ]
```

```
⇒ snmp2
```

```
$s configure ; # แสดงค่าในการสร้างเซสชันใหม่
```

```
⇒ -address 192.168.2.1 -port 161 -version SNMPv1 -community sun_public
   -transport udp -timeout 5 -retries 3 -window 10 -delay 0 -tags {192.168.2.1,
   sun_public}
```

คำสั่งการสร้าง snmp alias ของเซสชันมีรูปแบบคือ

```
snmp alias aliasName "-optionName value ?-optionName value?"
```

ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
snmp alias router1 "-address 192.168.2.13 -timeout 10 "
```

```
snmp alias router2 "-address 192.168.2.14 -windows 10 -delay 20 "
```

จะทำการสร้างสมนามชื่อ router1 และ router2 สามารถนำไปใช้ในคอนฟิกูเรชันเซสชันได้

```
$s configure -alias router1 -alias router2
```

• การสร้าง Varbind Lists

การทำงานของโปรโตคอล SNMP ในการค้นคืนหรือการเซต จะแสดงรายการตัวแปรของ MIB เสมอ รายการนี้จะเรียกว่า Varbind List การระบุตัวแปรสามารถใช้หมายเลข OID (Object Identifier) หรือชื่อ OID (ถ้ามีการโหลด MIB) สำหรับซอฟต์แวร์ Scotty จะมีการโหลด MIB-II อย่างอัตโนมัติ

ตัวอย่างการสร้าง Varbind List ที่ใช้ในการดำเนินการ SNMP get เพื่อค้นคืนค่าตัวแปรของกลุ่มระบบ

```
set vbl [ list sysDescr.0 sysName.0 sysLocation.0 ]
```

ตัวอย่างการสร้าง Varbind List ที่ใช้ในการดำเนินการ SNMP set เพื่อเซตค่าตัวแปรของกลุ่มระบบ

```
#
# สร้าง Varbind โดยใช้ตัวแปร MIB ที่ได้จากการโหลด MIB ไฟล์
#
set vbl [ list [ list sysDescr.0 "Cisco 4500" ] \
               [ list sysName.0 "energy-develop" ] \
               [ list sysLocation.0 "Bangkok Thai, Floor 19" ] ]
```

```
#
```

```
# สร้าง Varbind โดยใช้หมายเลข OID
```

```
#
```

```
set sysDescr 1.3.6.1.2.1.1.1
```

```
set sysName 1.3.6.1.2.1.1.5
```

```
set sysLocation 1.3.6.1.2.1.1.6
```

```
set vbl [ list [ list $sysDescr {OCTET STRING} Cisco 4500 ] \
```

```
               [ list $sysName {OCTET STRING} energy-develop ] \
```

```
               [ list $sysLocation {OCTET STRING} Bangkok Thai, Floor 19 ] ]
```

• การดำเนินการชิงโครนัส และอะชิงโครนัส

การดำเนินการชิงโครนัส สามารถทำได้ทั้งการ get, getnext และ set มีรูปแบบคำสั่งดังนี้

```
snmp# command varbindlist
```

snmp# คือ SNMP เซสชัน, command คือ get, getnext และ set ตัวอย่างการ get เพื่อแสดงผลลัพธ์ของ varbindlist

```
set result [$s get [list sysDescr.0 sysName.0 sysLocation.0]]
=>{1.3.6.1.2.1.1.1.0 {OCTET STRING} {Microsoft Corp. Windows 98.}}
{1.3.6.1.2.1.1.5.0 {OCTET STRING} PENTIUM_100} {1.3.6.1.2.1.1.6.0
{OCTET STRING} Default}
```

```
llength $result
```

```
=>3
```

ถ้ามีความผิดพลาดเกิดขึ้นจะมีการแจ้งเตือนดังตัวอย่าง

```
set result [$s get [list sysDescr.1 sysName.0 sysLocation.0]]
=> noSuchName 0 {1.3.6.1.2.1.1.1.1 NULL {}} {1.3.6.1.2.1.1.5.2 NULL {}}
{1.3.6.1.2.1.1.6.3 NULL {}}
```

จากตัวอย่างข้างต้นการผิดพลาดแจ้งรหัส 0 แสดงถึงความผิดพลาดที่เกิดจาก varbindlist ในตำแหน่งแรก ถ้าแจ้งรหัส 1 แสดงถึงความผิดพลาดที่เกิดจาก varbindlist ในตำแหน่งที่สอง แต่ถ้าแจ้ง noResponse แสดงถึงความผิดพลาดที่เกิดจากเวลา timeout

คำสั่ง catch ใช้เพื่อประเมินค่าสคริปต์และการรีเทิร์น trap ที่อาจจะผิดปกติ

```
catch {set result [$s get $vbl]} err
```

```
=>0
```

ถ้าแจ้งรหัส 0 แสดงถึงการดำเนินการ get ที่เสร็จสมบูรณ์ ถ้าแจ้งรหัส 1 และเซต err แสดงว่าค่าสคริปต์มีความผิดพลาด

ตัวอย่างการดำเนินการ set

```
set vbl [list [list sysName.0 Pentium_III_800_Mhz.]]
=>{sysName.0 Pentium_III_800_Mhz.}
catch {set result [$s set $vbl]} err
=>0
puts $result
=>{1.3.6.1.2.1.1.5.0 {OCTET STRING} Pentium_III_800_Mhz.}
```

จากตัวอย่างข้างต้น ลำดับแรกทำการสร้าง varbindlist หนึ่งรายการ โดยใช้ชื่อ OID และ ค่า ลำดับสองคำสั่ง catch แจ้งค่า 0 แสดงถึงการดำเนินการ set ที่เสร็จสมบูรณ์ สุดท้ายเป็นการรีเทริน เพื่อแสดงค่าของ varbindlist

การดำเนินการอะซิงโครนัส สามารถกระทำคำสั่ง get, getnext และ set ได้เช่นเดียวกันกับ ซิงโครนัส เครื่องประมวลผลโพรโตคอล (Protocol Engine) สามารถเก็บการติดตามการร้องขออะซิงโครนัสและกระบวนการ Callback สคริปต์ที่ตอบรับการร้องขอ การ Callback สคริปต์ จะทำการประเมินค่าไปตามลำดับ % เอสเคปพิเศษ สามารถใช้เพื่อเข้าถึงรายละเอียดที่ถูกบรรจุอยู่ในการตอบสนอง ลำดับ % เอสเคปนี้จะถูกแทนที่ก่อน Callback สคริปต์ถูกประเมินค่า การกำหนด % เอสเคป มีดังนี้

%%	แทนอักขระ % หนึ่งตัว
%V	แทนผลของ varbind list
%R	แทนผลของการร้องขอหมายเลข ID
%S	แทนผลของชื่อเซสชัน
%E	แทนผลของสถานะความผิดพลาด
%I	แทนผลของตัวชี้ความผิดพลาด เริ่มต้นที่ค่า 0
%A	แทนผลของไอพีแอดเดรสของอุปกรณ์ที่ส่ง SNMP แพ็กเก็ต
%P	แทนผลของหมายเลขพอร์ตของอุปกรณ์ที่ส่ง SNMP แพ็กเก็ต
%T	แทนผลของชนิด PDU

ตัวอย่างการ Callback สคริปต์ ลำดับแรกเป็นกระบวนการจัดเตรียมผล SNMP และรายการ ที่ถูกรีเทริน

```
proc GetSessionName {result} { return [lindex $result 0] }
proc GetRequestID {result} { return [lindex $result 1] }
proc GetErrorStatus {result} { return [lindex $result 2] }
proc GetErrorIndex {result} { return [lindex $result 3] }
proc GetVarBindList {result} { return [lindex $result 4] }
proc GetIPAddress {result} { return [lindex $result 5] }
proc GetPortNumber {result} { return [lindex $result 6] }
proc GetPDU {result} { return [lindex $result 7] }
```

ลำดับสองเป็นการ Callback สคริปต์ เพื่อสร้างรายการที่บ่งชี้มาจากผลของอะซิงโครนัสใน ลำดับแรก

```
set nb "nbroutine { %S %R %E %I {%V} %A %P %T } "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

proc nbroutine {result} {
    puts "Session Name = [GetSessionName $result] "
    puts "Request ID = [GetRequestID $result] "
    puts "Error Status = [GetErrorStatus $result] "
    puts "Error Index = [GetErrorIndex $result] "
    puts "VarBindList = [GetVarBindList $result] "
    puts "IP Address = [GetIPAddress $result] "
    puts "Port Number = [GetPortNumber $result] "
    puts "PDU Type = [GetPDU $result] "
}

```

```

catch {set result [$s getnext sysDescr $nb]} err

```

⇒ 0

ผลที่ได้จากคำสั่ง catch เป็น 0 แสดงถึงการกระทำคำสั่ง SNMP getnext ที่เสร็จสมบูรณ์ จากนั้นจะทำการเรียกใช้ routine nbroutine และแสดงผลดังรายการข้างล่าง

⇒ Session Name = snmp1

⇒ Request ID = 18382

⇒ Error Status = noError

⇒ Error Index = -1

⇒ VarBindList = {1.3.6.1.2.1.1.1.0 {OCTET STRING} {Microsoft Corp. Windows 98.}}

⇒ IP Address = 192.168.1.1

⇒ Port Number = 161

⇒ PDU Type = response

กระบวนการรอสำหรับการตอบสนองที่ยังค้างอยู่ สามารถใช้คำสั่ง wait ทำการบดบังจนกระทั่งการร้องขออะซิงโครนัสของเซสชันถูกดำเนินการทั้งหมด ดังตัวอย่าง

```

proc dwait {addr comm} {
    set s [snmp generator -address $addr -community $comm]
    $s get {ifIndex.1 ifInErrors.1 ifInOctets.1} {puts "%V"}
    $s get {ifIndex.2 ifInErrors.2 ifInOctets.2} {puts "%V"}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    $$ get {ifIndex.3 ifInErrors.3 ifInOctets.3} {puts "%V"}

    $$ wait

    puts "Completed!"
}

```

การเรียกใช้รูทีน `dwait` และการแสดงผล (`$addr` แทนค่าไอพีแอดเดรสของอุปกรณ์ที่ตรวจสอบ และ `$comm` แทนค่าของ `community` ที่กำหนด)

```

dwait 192.168.1.1 public
⇒ {1.3.6.1.2.1.2.2.1.1.1 Integer32 1} {1.3.6.1.2.1.2.2.1.14.1 Counter32 0}
{1.3.6.1.2.1.2.2.1.10.1 Counter32 20635}
⇒ {1.3.6.1.2.1.2.2.1.1.2 Integer32 2} {1.3.6.1.2.1.2.2.1.14.2 Counter32 0}
{1.3.6.1.2.1.2.2.1.10.2 Counter32 20742}
⇒ {1.3.6.1.2.1.2.2.1.1.3 Integer32 3} {1.3.6.1.2.1.2.2.1.14.2 Counter32 0}
{1.3.6.1.2.1.2.2.1.10.3 Counter32 20849}
⇒ Completed!

```

• คำสั่ง SNMP WALK

การ walk เป็นการรวบรวมซับทรีจาก MIB ทั้งหมด Scotty ได้จัดเตรียมคำสั่ง walk ไว้ 2 คำสั่ง คำสั่งแรกเป็นการ walk ที่กระทำแบบซิงโครนัส แต่ละ `VarBindList` จะถูกค้นคืนจากเอเจนต์ โดยการประเมินค่าของ Body มีรูปแบบคำสั่งคือ

```
snmp# walk varName vbl body
```

ตัวอย่างการ walk ไปยังไอพีแอดเดรส 192.168.1.1 เพื่อรวบรวมค่าของ `ipRouteDest` และ `ipRouteType` จากตารางไอพีเรดิ้ง ซึ่งจริง ๆ แล้วก่อนการประเมินค่า Body คือ `puts $x` จะเริ่มต้นค่า `VarBindList` จะถูกกำหนดด้วยชื่อตัวแปรคือ `x`

```

set s [snmp generator -address 192.168.1.1 ]
⇒ snmp2
$$ walk x {ipRouteDest ipRouteType} {puts $x}
⇒ 1.3.6.1.2.1.4.21.1.1.127.0.0.0 IPAddress 127.0.0.0}
{1.3.6.1.2.1.4.21.1.8.127.0.0.0 Integer32 direct}

```

```

⇒ {1.3.6.1.2.1.4.21.1.1.192.168.1.0 IpAddress 192.168.1.0}
{1.3.6.1.2.1.4.21.1.8.192.168.1.0 Integer32 direct}
⇒ {1.3.6.1.2.1.4.21.1.1.192.168.1.1 IpAddress 192.168.1.1}
{1.3.6.1.2.1.4.21.1.8.192.168.1.1 Integer32 direct}
⇒ {1.3.6.1.2.1.4.21.1.1.192.168.1.255 IpAddress 192.168.1.255}
{1.3.6.1.2.1.4.21.1.8.192.168.1.255 Integer32 direct}
⇒ {1.3.6.1.2.1.4.21.1.1.224.0.0.0 IpAddress 224.0.0.0}
{1.3.6.1.2.1.4.21.1.8.224.0.0.0 Integer32 direct}
⇒ {1.3.6.1.2.1.4.21.1.1.255.255.255.255 IpAddress 255.255.255.255}
{1.3.6.1.2.1.4.21.1.8.255.255.255.255 Integer32 direct}

```

คำสั่งที่สองเป็นการ walk ที่กระทำแบบอะซิงโครนัส แต่ละ VarBindList จะถูกกั้นคั่นจากเอเจนต์โดยการประเมินค่าของ callback สคริปต์ มีรูปแบบคำสั่งคือ

```
snmp# walk vbl script
```

ตัวอย่างการ walk เพื่อรวบรวมค่า ipRouteDest และ ipRouteType จากตารางไอพีเรดิง โดยดำเนินการแบบอะซิงโครนัส

```

set s [snmp generator -address 192.168.1.1 ]
⇒ snmp2
Ss walk {ipRouteDest ipRouteType} {puts "%E %V"}
⇒ 20573
⇒ noError {1.3.6.1.2.1.4.21.1.1.127.0.0.0 IpAddress 127.0.0.0}
{1.3.6.1.2.1.4.21.1.8.127.0.0.0 Integer32 direct}

```

• การสร้างแทรป (Trap)

การสร้างแทรปของ SNMP ลำดับแรกต้องทำการสร้างเซสชันก่อนด้วยคำสั่ง snmp notifier จากนั้นจึงใช้คำสั่ง trap เพื่อสร้างแทรปเมสเสจ มีรูปแบบคำสั่งคือ

```
snmp# trap snmpTrapOid vbl
```

ตัวอย่าง การส่ง coldStart แทรป ไปยังไอพีแอดเดรส 192.168.1.1 แทรปมี VarBindList คือ {sysDescr.0 "bogus"} และจะมีค่า sysUpTime.0 ถูกเพิ่มเข้าไปอย่างอัตโนมัติในการส่งแทรปเมสเสจ

```
set s [snmp notifier -address 192.168.1.1]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⇒ snmp3

```
$s trap coldStart [list [list sysDescr.0 "bogus"]]
```

ตัวอย่าง การสร้าง Cisco enterprise-specific แทรป Cisco มีหมายเลข enterprise คือ 9 มีหมายเลข generic trap type คือ 6 มีหมายเลข specific trap คือ 100 มีคำสั่งดังนี้

```
$s trap enterprises.9.6.100 [ list [ list ifOperStatus.1.1 ] ]
```

ตัวอย่าง การสร้างคิฟอลต์คอนฟิกูเรชันสำหรับเซสชันโดยใช้ snmp notifier สามารถใช้คำสั่ง configure ได้ดังนี้

```
set trapsession [ snmp notifier ]
⇒ snmp4
$strapsession configure
⇒ -address 127.0.0.1 -port 162 -version SNMPv1 -community public -transport
udp -timeout 5 -retries 3 -window 10 -delay 0 -tags {} -enterprise
1.3.6.1.4.1.1575
```

6.4.2 คำสั่ง MIB

คำสั่ง MIB [20] เป็นส่วนสำคัญมากสำหรับ Tnm 3.0 ภายในคำสั่งจะมีคำส่งย่อยที่มีประโยชน์และถูกเรียกใช้บ่อยๆ ดังนี้

คำสั่ง mib load ใช้สำหรับ โหลด MIB ไฟล์และเข้าถึงข้อกำหนด MIB แพคเกจ Scotty ได้ จัดเตรียม RFC MIB ไฟล์จำนวนมากในไดเรกทอรีของ \$tnm(library)/mibs คำสั่ง mib load จะพยายามโหลดไฟล์อย่างอัตโนมัติจากไดเรกทอรีปัจจุบันก่อน ถ้าไม่มีไฟล์อยู่ในไดเรกทอรีปัจจุบันก็จะพยายามโหลดในไดเรกทอรีของ \$tnm(library)/site และ \$tnm(library)/mibs

MIB ไฟล์สามารถควบคุมให้ทำการโหลดอย่างอัตโนมัติ โดยผ่าน 2 ตัวแปรของ Tcl คือ \$tnm(mibs:core) ซึ่งบรรจุชื่อของ MIB ไฟล์ที่สร้างตามข้อกำหนดของ SNMPv1 และ SNMPv2 และ \$tnm(mibs) ซึ่งบรรจุรายการชื่อไฟล์ของ MIB ที่มีประโยชน์ในแต่ละสถานที่ ผู้ใช้สามารถทำเป็นสคริปต์พิเศษโดยการเซต MIB ไฟล์ใน \$tnm(library)/site/init.tcl

ตัวอย่างการใช้คำสั่ง mib load เพื่อทำการ โหลด MIB ไฟล์ rfc1213-mib และ Ds3-mib

```
mib load rfc1213-mib
```

```
mib load Ds3-mib
```

คำสั่ง mib access ใช้รีเทริน โหมดการเข้าถึงที่ถูกระบุสำหรับ MIB โหนด

mib access ifAdminStatus

⇒ read-write

คำสั่ง mib description ใช้รีเทรินรายละเอียด textual ของ MIB โหนด

mib description ifAdminStatus

⇒ *The desired state of the interface. The testing(3)*

state indicates that no operational packets can be

passed. When a managed system initializes, all

interfaces start with ifAdminStatus in the down(2)

state. As a result of either explicit management

action or per configuration information retained by

the managed system, ifAdminStatus is then changed to

either the up(1) or testing(3) states (or remains in

the down(2) state).

คำสั่ง mib enums ใช้รีเทรินการระบุชนิดของชื่อร่วมกับหมายเลข

mib enums IfAdminStatus

⇒ up 1 down 2 testing 3

คำสั่ง mib exists ใช้ทดสอบโหนดว่ายังคงอยู่หรือไม่ ถ้าผลเป็น 1 แสดงว่ายังคงอยู่ ถ้าเป็น

0 แสดงว่าไม่คงอยู่

mib exists IfAdminStatus

⇒ 1

mib exists dot3Statistics

⇒ 0

คำสั่ง mib file ใช้รีเทรินชื่อไฟล์ที่ใช้สำหรับ MIB โหนด

mib file IfAdminStatus

⇒ C:\PROGRA~1\SCOTTY\lib\tnm3.0.0\mibs\IF-MIB

คำสั่ง mib format ใช้เพื่อแปลงค่าจากเลขจำนวนเต็มเป็นค่าสตริงที่ระบุ

mib format IfAdminStatus 1

⇒ up

mib format IfAdminStatus 2

⇒ down

คำสั่ง mib scan ใช้เพื่อแปลงค่าสตริงที่ระบุกลับค่าเป็นเลขจำนวนเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mib scan ifAdminStatus up

⇒1

mib scan ifAdminStatus down

⇒2

คำสั่ง mib length ใช้รีเทรินความยาวของ OID

mib length 1.3.6.1.2.1.2.2.1.1

⇒10

mib length ifAdminStatus

⇒10

คำสั่ง mib module ใช้รีเทรินชื่อของโมดูล ที่บรรจุ MIB โหนด

mib module ifAdminStatus

⇒IF-MIB

คำสั่ง mib name และ mib type ใช้รีเทรินชื่อเต็มของ โหนด อาจเป็นชื่อโหนดหรือ OID

mib name ifAdminStatus

⇒IF-MIB:ifAdminStatus

mib type 1.3.6.1.2.1.2.2.1.7

⇒IF-MIB:ifAdminStatus

คำสั่ง mib oid ใช้รีเทรินหมายเลขของ OID

mib oid ifAdminStatus

⇒1.3.6.1.2.1.2.2.1.7

คำสั่ง mib subtree ใช้เปรียบเทียบค่า 2 OID ถ้ารีเทรินค่า 1 แสดงว่า OID ลำดับที่ 2 มีซัพทรี ถูกกำหนดอยู่ใน OID ลำดับแรก แต่ถ้ารีเทรินค่า 0 แสดงในทางตรงกันข้าม

mib subtree ifTable ifAdminStatus

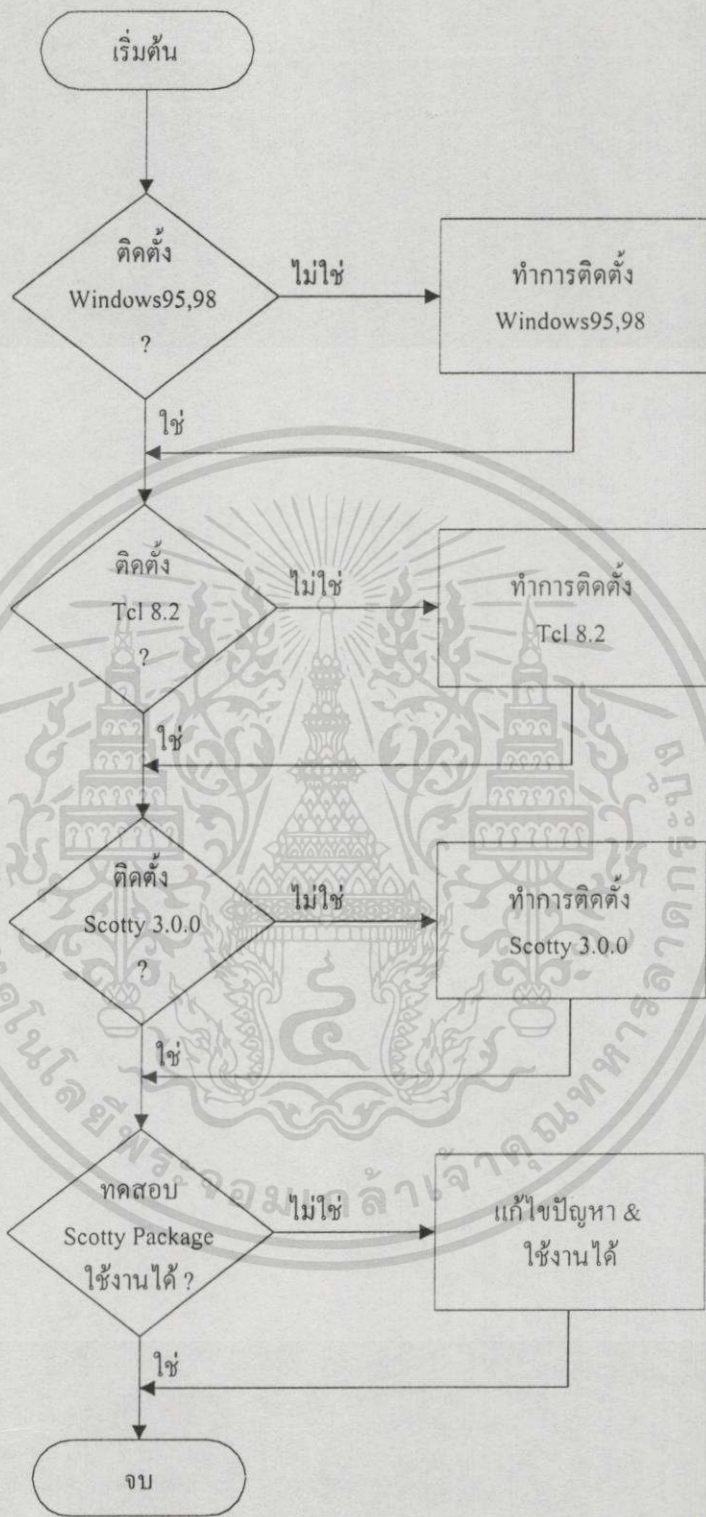
⇒1

mib subtree ifOperStatus ifAdminStatus

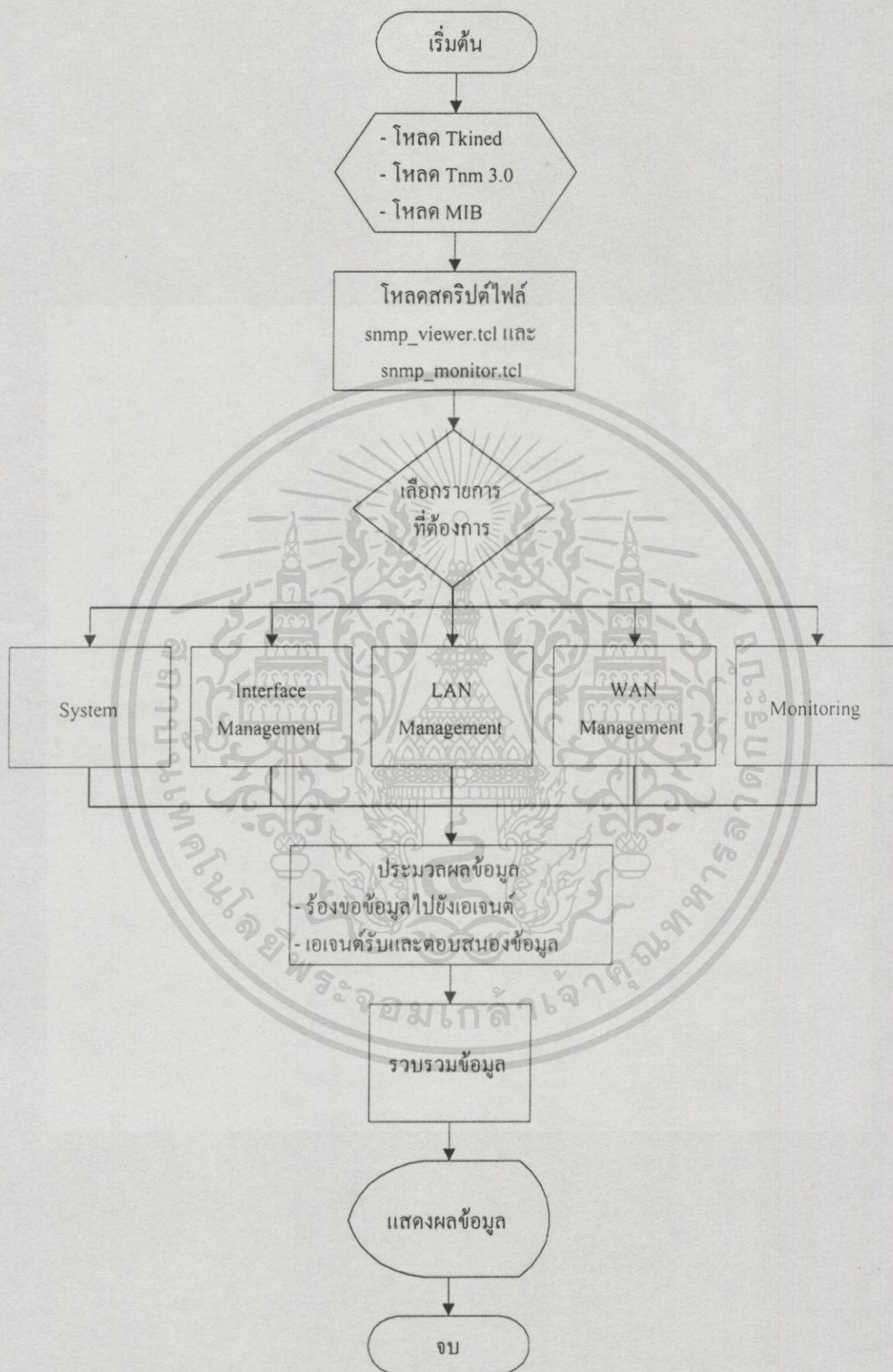
⇒0

6.5 โครงสร้างโปรแกรมหลัก

ในการพัฒนาโปรแกรมได้มีการประยุกต์ใช้คำสั่งต่าง ๆ ที่กล่าวถึงในตอนต้น เพื่อสร้างเป็น สคริปต์ไฟล์ไว้ใช้งาน ขั้นตอนการทำงานแสดงให้เห็นดังรูปต่อไปนี้



รูปที่ 6.4 ขั้นตอนการเตรียมพร้อมก่อนใช้งาน



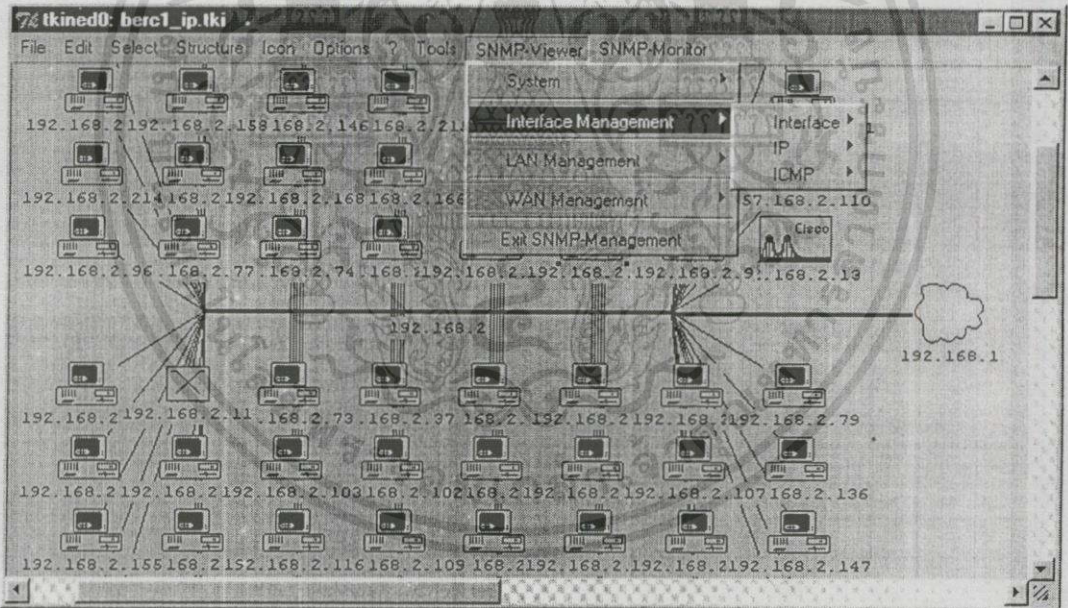
รูปที่ 6.5 ขั้นตอนการทำงานของโปรแกรมหลัก

บทที่ 7

การทดลองและผลการทดลอง

7.1 การเรียกใช้โปรแกรม

โปรแกรมจัดการเครือข่ายของสถานีจัดการที่พัฒนาขึ้นมาทำงานภายใต้ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ 9x ก่อนเริ่มต้นใช้งานต้องทำการติดตั้งซอฟต์แวร์ Tcl/Tk Ver. 8.2 และ Scotty Ver. 3.0.0 ก่อน ซึ่งสามารถดาวน์โหลดได้ที่เว็บไซต์ <http://dev.scriptics.com/software/tcltk/8.2.html> [18] และ <http://wwwhome.cs.utwente.nl/~schoenw/scotty> [19] ตามลำดับ หลังจากติดตั้งซอฟต์แวร์เสร็จแล้ว การเรียกใช้โปรแกรมให้คลิกที่ปุ่ม Start\Programs\Tkinedv1.5.0 โปรแกรม tkined จะถูกโหลดขึ้นมา จากนั้นคลิกเลือกเมนูคำสั่ง Tools\Start Script เลือกไฟล์ snmp_viewer และ snmp_monitor ซึ่งเป็นสคริปต์ไฟล์ที่พัฒนาขึ้นมา จากนั้นคลิกเลือกเมนูคำสั่ง SNMP-Viewer หรือ SNMP-Monitor เพื่อเลือกรายการที่ต้องการตรวจสอบ



รูปที่ 7.1 แสดงเมนูหลักของโปรแกรมและเครือข่ายจริงที่ใช้ทดสอบ

รูปที่ 7.1 แสดงจอภาพเมนูหลักของโปรแกรม ประกอบด้วย 2 ส่วนหลัก จากเมนูบาร์คือ SNMP-Viewer และ SNMP-Monitor

SNMP-Viewer ใช้รายงานผลข้อมูล โดยแสดงผลเป็นตาราง สามารถตรวจสอบข้อมูลทางด้านสถิติการใช้ และตรวจสอบสถานะการทำงานของอุปกรณ์เครือข่าย

SNMP-Monitor ใช้รายงานผลข้อมูล โดยแสดงผลเป็นกราฟ รวมถึงมีความสามารถในการตั้งค่า Threshold เพื่อให้โปรแกรมทำการแจ้งเตือนเมื่อถึงค่าที่ได้กำหนดไว้

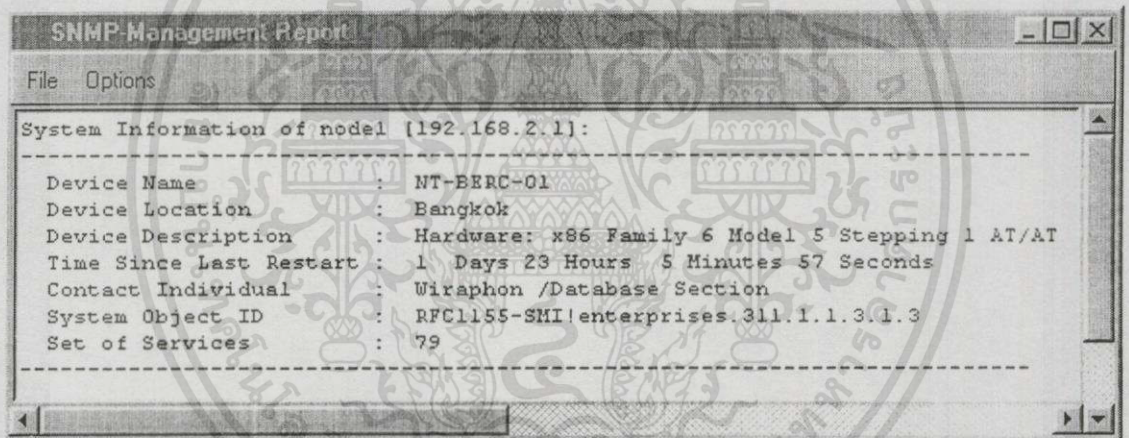
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การทดลองและการวิเคราะห์ข้อมูล

การทดลองผู้วิจัยได้ใช้ระบบเครือข่ายของสำนักกำกับและอนุรักษ์พลังงาน อาคารบางกอกไทยทาวเวอร์ เขตราชเทวี กรุงเทพมหานคร ระบบเครือข่ายใช้เทคโนโลยีตามมาตรฐานอีเทอร์เน็ต มีโปรโตคอล TCP/IP ในการติดต่อสื่อสารกันเป็นหลัก แสดงดังรูปที่ 7.1 อุปกรณ์ในระบบเครือข่ายได้รับแจกไอพีแอดเดรสเป็นคลาส C มีหมายเลขเครือข่าย 192.168.2.0 ซับเน็ตมาสก์ 255.255.255.0 มีอุปกรณ์เราเตอร์ทำหน้าที่เป็นเกตเวย์เชื่อมต่อไปยังเครือข่าย 192.168.1.0 ในการทดสอบสามารถทดสอบอุปกรณ์ใดก็ได้บนระบบเครือข่าย โดยอุปกรณ์นั้นจะต้องถูกติดตั้ง SNMP Agent เสียก่อน ตัวอย่างการทดสอบ ผู้วิจัยได้ทำการทดสอบอุปกรณ์เครือข่ายต่าง ๆ ดังต่อไปนี้

7.2.1 การทดสอบแสดงผลข้อมูลระบบ

วัตถุประสงค์ เพื่อรายงานค่าติดตั้งของอุปกรณ์ โดยทำการตรวจสอบเครื่อง Server ที่ติดตั้งระบบปฏิบัติการ Windows NT Server 4.0 พร้อมติดตั้ง SNMP Agent จากโปรแกรมคลิกเพื่อเลือกเครื่อง Server คลิกเมนูคำสั่ง SNMP-Viewer\System\System Information แสดงผลดังรูปที่ 7.2



รูปที่ 7.2 แสดงข้อมูลระบบ

ผลทดสอบการแสดงผลข้อมูลของระบบ สามารถนำมาวิเคราะห์เพื่อให้ประโยชน์ในด้านการจัดการความบกพร่อง และการจัดการคอนฟิกูเรชัน [21] ซึ่งจะเป็นการค้นคืนค่าของออบเจกต์ในกลุ่ม System ของ MIB-II สามารถที่จะบอกรายละเอียดถึงค่าติดตั้งที่จำเป็นโดยทั่วไปของอุปกรณ์ ส่วนค่าในรายการของ Set of Services จะเป็นการชี้ถึงระดับชั้นของโปรโตคอลที่ให้บริการ [6] ซึ่งจะเป็นการรวมค่าแต่ละชั้นตามสูตร $2^{(L-1)}$ โดย L คือหมายเลขชั้นโปรโตคอล ผลการทดลองที่ได้ 79 หมายถึงอุปกรณ์ให้บริการหลักชั้นแอปพลิเคชัน (ชั้น7), ชั้นเอ็นด์ทูเอ็นด์ (ชั้น4), ชั้นอินเทอร์เน็ต (ชั้น3), ชั้นดาต้าลิงก์ (ชั้น2) และชั้นฟิสิคัล (ชั้น1) ซึ่งรวมค่าได้ $79, 2^{(7-1)} + 2^{(4-1)} + 2^{(3-1)} + 2^{(2-1)} + 2^{(1-1)}$ ข้อมูลเหล่านี้เป็นส่วนประกอบที่มีประโยชน์เพื่อช่วยในการแก้ไขปัญหา เช่น เมื่ออุปกรณ์ไม่ทำงาน จะติดต่อบุคคลใด สถานที่ตั้งอยู่ที่ไหน และอุปกรณ์ถูกเปิดใช้งานมานานเท่าไรแล้ว

7.2.2 การทดสอบแสดงผลข้อมูลอินเทอร์เฟซ

วัตถุประสงค์ เพื่อรายงานสถานะของอุปกรณ์ และปริมาณกราฟฟิกของอินเทอร์เฟซ โดยทำการตรวจสอบอุปกรณ์ฮับ Bay Networks รุ่น BayStack Ethernet ติดตั้ง SNMP NMM Agent V.1.3.0 จากโปรแกรมคลิกเพื่อเลือกอุปกรณ์ฮับ BayStack คลิกเมนูคำสั่ง SNMP-Viewer\Interface Management\Interface\Interface Detail Information, Interface Input Statistics, Interface Output Statistics, Interface Quality แสดงผลดังรูปที่ 7.3, 7.4 และ 7.5

SNMP-Management Report							
File Options							
Interface Detail Information of node0 (203.150.25.19):							
Interface	Description	MTU	Speed	Physical Address	Admin	Oper	Interface Type
1	et01	1514	10 H	00:00:81:0E:12:80	up	up	ethernetCsmacd

รูปที่ 7.3 รายละเอียดของอินเทอร์เฟซ

ผลทดสอบการแสดงผลข้อมูลของอินเทอร์เฟซ สามารถวิเคราะห์เพื่อให้ประโยชน์ในด้านการจัดการความบกพร่อง, การจัดการคอนฟิกูเรชัน และการจัดการสมรรถนะ [21] ซึ่งจะเป็นการค้นคืนค่าออบเจกต์กลุ่ม Interface ของ MIB-II

วิเคราะห์การจัดการความบกพร่อง

ผลการทดลองรูปที่ 7.3 สามารถบ่งบอกถึงจำนวนอินเทอร์เฟซของอุปกรณ์ว่ามีเท่าไร ดูได้จากจำนวนแถวที่มีหมายเลขอินเทอร์เฟซกำกับอยู่ รวมถึงสามารถบ่งบอกสถานะของอินเทอร์เฟซ ซึ่งดูได้จาก Admin และ Oper ดังนี้

- ถ้า Admin และ Oper มีค่า up แสดงว่าอินเทอร์เฟซทำงาน
- ถ้า Admin มีค่า up และ Oper มีค่า down แสดงว่าอินเทอร์เฟซขัดข้อง (Failure)
- ถ้า Admin และ Oper มีค่าเป็น down แสดงว่าอินเทอร์เฟซปิดการบริหาร
- ถ้า Admin และ Oper มีค่าเป็น testing แสดงว่าอินเทอร์เฟซอยู่ในโหมดทดสอบ

แต่ถ้าค่าของออบเจกต์ทั้งสองไม่เป็นไปตามนี้ อาจเป็นไปได้ว่าซอฟต์แวร์ของอุปกรณ์นั้นทำงานไม่ถูกต้อง

วิเคราะห์การจัดการคอนฟิกูเรชัน

คอลัมน์ Description และ InterfaceType บ่งบอกถึงชื่อ และชนิดอินเทอร์เฟซ จากผลการทดลองชื่ออินเทอร์เฟซส่งค่าสตริง "et01" มีชนิด (ifType) ซึ่งจะถูกกำหนดค่าเป็นเลขจำนวนเต็ม หมายเลขนี้จะถูกแปลง (Mapping) เป็นค่าสตริงที่กำหนดในเอ็มไอบีคือ "ethernetCsmacd" คอลัมน์ Speed บอกรหัสความเร็วหรืออัตราการส่งผ่านข้อมูลของอินเทอร์เฟซ มีหน่วยเป็นเมกะบิตต่อวินาที

คอลัมน์ MTU (Maximum Transmission Unit) บ่งบอกถึงขนาดสูงสุดของดาต้าแกรมที่ส่งผ่านบน อินเทอร์เน็ต มีหน่วยเป็น ไบต์

The screenshot shows a window titled "SNMP-Management Report" with a menu bar containing "File" and "Options". The main content area displays two sections of statistics for "node0 [203.150.25.19]".

Interface Input Statistics of node0 [203.150.25.19]:

Interface ID	:	1
Description	:	et01
Bytes Received	:	1327975419
Unicast Packets	:	5
Multicast Packets	:	6161949
Error Packets	:	13522
Discarded Packets	:	21602
Unknown Packets	:	1469212

Interface Output Statistics of node0 [203.150.25.19]:

Interface ID	:	1
Description	:	et01
Bytes Sent	:	3384
Unicast Packets	:	22
Multicast Packets	:	6
Error Packets	:	0
Discarded Packets	:	0

รูปที่ 7.4 ข้อมูลทางสถิติของอินเทอร์เน็ต

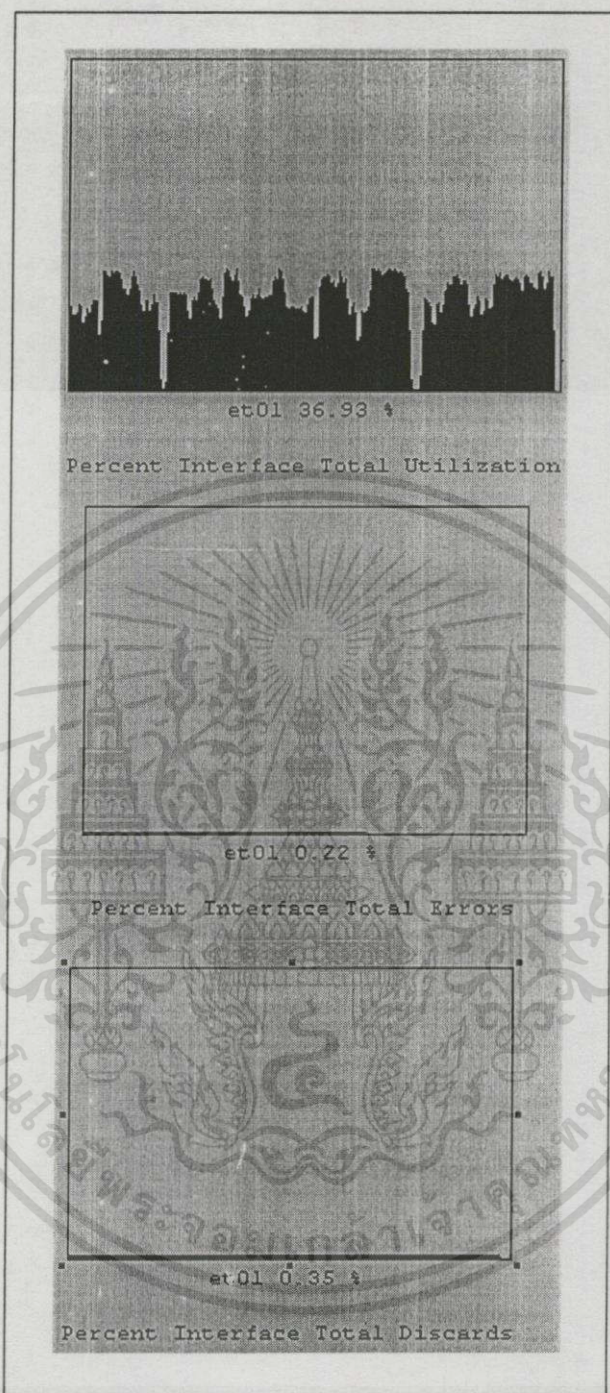
The screenshot shows a window titled "SNMP-Management Report" with a menu bar containing "File" and "Options". The main content area displays "Interface Quality Statistics of node0 [203.150.25.19]".

Interface ID	:	1
Description	:	et01
Input Errors	:	0.22 %
Output Errors	:	0.00 %
Total Errors	:	0.22 %
Input Discards	:	0.35 %
Output Discards	:	0.00 %
Total Discards	:	0.35 %

รูปที่ 7.5 สมรรถนะของอินเทอร์เน็ต

วิเคราะห์การจัดการสมรรถนะ

รูปที่ 7.4 เป็นการรายงานข้อมูลการใช้ทางด้านสถิติของอินเทอร์เน็ต ค่าเหล่านี้สามารถนำมาคำนวณหาค่าเปอร์เซ็นต์การใช้ประโยชน์ (% Utilization) ค่าเปอร์เซ็นต์ความผิดพลาด (% Error) และค่าเปอร์เซ็นต์แพ็กเก็ตยกเลิก (% Discard) แสดงให้เห็นดังรูปที่ 7.5 และกราฟดังรูปที่ 7.6 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6 ค่าเปอร์เซ็นต์การใช้ประโยชน์, ค่าเปอร์เซ็นต์ความผิดพลาด และค่าเปอร์เซ็นต์แพ็กเก็ตขุดเล็กของอินเทอร์เฟซ

การคำนวณค่าเปอร์เซ็นต์การใช้ประโยชน์ของอินเทอร์เฟซ เป็นอัตราส่วนของจำนวนไบต์ที่รับและส่งทั้งหมดคูณ 8 หรืออัตราการส่งผ่านข้อมูลหรือความเร็วอินเทอร์เฟซ ดังสูตรการคำนวณ

$$\% \text{ Utilization} = (\text{total bytes per sec} * 8) / \text{ifSpeed}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{total bytes per sec} = (\text{ifInOctets}_y - \text{ifInOctets}_x) + (\text{ifOutOctets}_y - \text{ifOutOctets}_x) / (y-x)$$

เมื่อ x และ y คือช่วงเวลาการโพลล์มีหน่วยเป็นวินาที

การคำนวณค่าเปอร์เซ็นต์ความผิดพลาดของอินเทอร์เฟซ จะเป็นอัตราส่วนจำนวนแพ็กเก็ตเกิดผิดพลาดต่อจำนวนแพ็กเก็ตทั้งหมด จำนวนแพ็กเก็ตเกิดผิดพลาดประกอบด้วยแพ็กเก็ตผิดพลาดอินพุตรวมกับเอาต์พุต ส่วนจำนวนแพ็กเก็ตทั้งหมดประกอบด้วยแพ็กเก็ตที่รับรวมกับแพ็กเก็ตส่ง ดังสูตรการคำนวณ

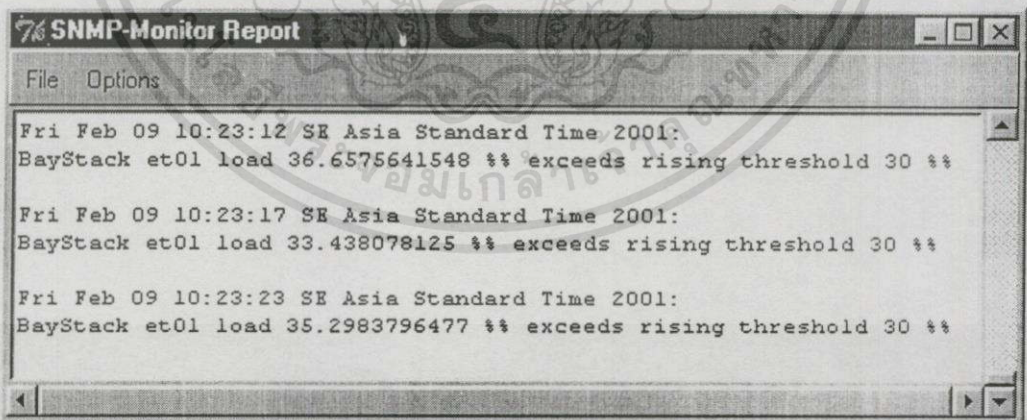
$$\% \text{ Error} = \frac{\text{ifInErrors} + \text{ifOutErrors}}{(\text{ifInUcastPkts} + \text{ifInNUcastPkts}) + (\text{ifOutUcastPkts} + \text{ifOutNUcastPkts})}$$

ถ้าค่าเปอร์เซ็นต์ความผิดพลาดมีค่าสูง แสดงว่าอินเทอร์เฟซมีปัญหาในการรับส่ง สาเหตุอาจเกิดจากสื่อมีปัญหาชั่วคราว

การคำนวณค่าเปอร์เซ็นต์แพ็กเก็ตขบเลิก เป็นอัตราส่วนของจำนวนแพ็กเก็ตที่ถูกขบเลิกต่อจำนวนแพ็กเก็ตทั้งหมด ดังสูตรการคำนวณ

$$\% \text{ Discard} = \frac{\text{ifInDiscards} + \text{ifOutDiscards}}{(\text{ifInUcastPkts} + \text{ifInNUcastPkts}) + (\text{ifOutUcastPkts} + \text{ifOutNUcastPkts})}$$

ถ้าอินเทอร์เฟซขบเลิกแพ็กเก็ตมากขึ้น จำนวนไบต์ทั้งหมดของเอาต์พุตจะลดลง ซึ่งแสดงถึงอินเทอร์เฟซเกิดความแออัด (Congestion)



รูปที่ 7.7 รายงานการแจ้งเตือน

รูปที่ 7.7 เป็นรายงานการแจ้งเตือนเมื่อถึงค่า Threshold ที่ตั้งไว้ ทดสอบโดยการตั้งค่า Threshold ของอัตราการใช้ประโยชน์ไว้ที่ 30 % และเมื่ออัตราการใช้ประโยชน์มีค่าเกินกว่า 30 % โปรแกรมจะทำการแจ้งเตือนขึ้นมาให้ โดยปกติการตั้งค่า Threshold ของอัตราการใช้ประโยชน์ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ที่ 70 % ซึ่งถ้ามีอัตราการใช้เฉลี่ยเกินกว่า 70 % ตลอดเวลา แสดงว่าเครือข่ายอาจเกิดความแออัด ผู้ดูแลระบบจึงควรวางแผนเพื่อทำการขยายเครือข่ายต่อไป

7.2.3 การทดสอบแสดงผลข้อมูลไอพี

วัตถุประสงค์ เพื่อรายงานค่าไอพีแอดเดรส ตาราง เราติง ตาราง ARP (Address Resolution Protocol) และปริมาณทราฟฟิก โดยตรวจสอบอุปกรณ์เราเตอร์ Cisco 4500 ติดตั้ง Agent รุ่น IOS 11.1 จากโปรแกรมคลิกเลือกอุปกรณ์เราเตอร์ คลิกเมนูคำสั่ง SNMP-Viewer/Interface Management /IP/IP Addresses, IP Routing Table, IP ARP Table ดังรูป 7.8 และ IP Statistics ดังรูป 7.9

7% SNMP-Management Report

File Options

IP addresses of node0 [192.168.2.13]:

Interface	Description	IP Address	Subnet Mask	Broadcast	Max Reassemble
1	Ethernet0	192.168.2.13	255.255.255.0	1	18024
4	Serial1	192.168.0.252	255.255.255.248	1	18024

Routing Table of node0 [192.168.2.13]:

Destination	Subnet Mask	Next Hop	Interface	Metric	Type	Protocol	Age
192.168.2.0	255.255.255.0	192.168.2.13	1	0	direct	local	0

ARP Table of node0 [192.168.2.13]:

Interface	MAC Address	IP Address	ARP Type	Vendor
1	00:08:C7:4C:B8:0E	192.168.2.1	dynamic	
1	08:00:20:80:94:B4	192.168.2.2	dynamic	Sun
1	00:00:F8:7C:2B:13	192.168.2.4	dynamic	DEC
1	00:00:8E:DS:7B:89	192.168.2.5	dynamic	Accton Technology Corporation
1	00:20:18:65:6D:4C	192.168.2.8	dynamic	
1	00:00:0C:40:10:D9	192.168.2.13	other	Cisco
1	00:00:F8:7C:2B:13	192.168.2.33	dynamic	DEC
1	00:20:18:8F:6B:94	192.168.2.66	dynamic	

รูปที่ 7.8 แสดงข้อมูลไอพี

ผลการทดสอบการแสดงผลข้อมูลไอพี สามารถนำมาวิเคราะห์เพื่อให้ประโยชน์ในการจัดการความบกพร่อง, การจัดการคอนฟิกูเรชัน และการจัดการสมรรถนะ ซึ่งจะเป็นการกั้นกั้นค่าของออบเจ็กต์กลุ่ม IP ของ MIB-II

การรายงานค่าไอพีแอดเดรส สามารถแสดงรายละเอียดที่เกี่ยวข้องกับค่าของไอพีแอดเดรสค่า Broadcast ที่เป็น 1 แสดงถึง Host ID เป็น 1 หมด (255) ซึ่งคือ IP Broadcast Address นั้นเอง ส่วน Max Reassemble ซึ่งถึงขนาดที่ใหญ่ที่สุดของไอพีค้ำแกรมที่สามารถทำการรีแอสเซมบลีได้

การรายงานค่าตารางเราติง [7] มีประโยชน์ในการติดตามปัญหาของเส้นทาง สามารถแสดงรายละเอียดที่เกี่ยวข้องกับเส้นทาง เช่น ไอพีแอดเดรสปลายทางของเรา ชับเน็ตมาส์กของเรา แอดเดรสของเกตเวย์ (Next Hop) ชนิดของการเรา ถ้ามีค่า direct ซึ่งถึงโลกดไอพีของ Next Hop มี

ค่า Destination บนเครือข่ายไอพีเดียวกัน ถ้าค่า indirect ซึ่งถึง Next Hop มีค่าไอพีเดียวกันกับเราเตอร์ ถ้าค่า invalid จะชี้ถึงเส้นทางที่ถูกลบ ส่วน Protocol จะชี้ถึงกลไกเราเตอร์ที่ใช้ในการเรียนรู้

การรายงานค่า ARP จะให้ข้อมูลที่เกี่ยวข้องกับการแปลงค่า (Mapping) ระหว่าง IP Address และ MAC Address (Media Access Control) โดยปกติชนิดของการแปลงจะเป็นแบบ dynamic แต่ถ้าเป็น invalid จะถือว่าแอดเดรสนั้นเป็นโมฆะ ซึ่งการนำไปใช้ในการแก้ปัญหา อุปกรณ์เป้าหมายจะต้องอยู่ในรายการของ ARP Table ถ้าไม่อยู่ในรายการแสดงว่าอุปกรณ์นั้นอาจจะมีปัญหาในการเชื่อมต่อชั่วคราวบ้าง ถ้ามีการเปลี่ยนแปลงของฮาร์ดแวร์หรือซอฟต์แวร์ใหม่ ค่าใน ARP แคชอาจไม่เป็นปัจจุบันจึงควรมีการลบล้าง ARP แคชใหม่ [7]

IP (Internet Protocol) Statistics of node0 [192.168.2.13]:	
Forwarding Status	: forwarding
Packets Received	: 5071088
Output Requests	: 21856
Delivered Packets	: 58238
Packets Forwarded	: 4104255
Fragments Reassembled	: 0
Fragments Created	: 0
Fragments Received	: 0
Packets Fragments OK	: 10
Reassemble Timeout	: 30
Default Time To Live	: 255
Addressing Errors	: 0
Input Header Errors	: 6
Discarded Packets	: 908398
Reassemble Failures	: 0
Unknown Protocols	: 0
Fragmentation Failures	: 0
Number Route Discards	: 203

รูปที่ 7.9 แสดงข้อมูลทางสถิติของไอพี

รูปที่ 7.9 เป็นการรายงานข้อมูลทางสถิติของไอพี สามารถแบ่งกลุ่มออบเจกต์เป็นกลุ่มที่เกี่ยวข้องกับการนับด้านอินพุต เอาต์พุต และเฟรมเมนต์เดชัน ข้อมูลเหล่านี้มีประโยชน์สามารถนำมาวิเคราะห์เพื่อวัดปริมาณกราฟฟิกของระบบได้ โดยสามารถนำมาหาค่าเปอร์เซ็นต์ความผิดพลาดอินพุต ซึ่งจะเป็นอัตราส่วนจำนวนอินพุตค่าต่ำแกรมที่ผิดพลาดทั้งหมดต่อจำนวนอินพุตค่าต่ำแกรมที่รับ ดังสูตรการคำนวณ

$$\% \text{ ip input errors} = \frac{(\text{ipInDiscards} + \text{ipInHdrErrors} + \text{ipInAddrErrors})}{\text{IpInReceives}}$$

ถ้าค่าเปอร์เซ็นต์ความผิดพลาดของอินพุตมีค่าสูง แสดงว่าสมรรถนะของระบบไม่ดี ถ้าค่าเปอร์เซ็นต์ความผิดพลาดมีค่าน้อย แสดงว่าดาต้าแกรมที่รับมีความผิดพลาดน้อยเมื่อเปรียบเทียบกับดาต้าแกรมที่รับเข้ามาทั้งหมด ซึ่งจะชี้ถึงสมรรถนะที่ดีของระบบ

สถิติการนับจำนวนด้านเอาต์พุต สามารถบ่งบอกถึงสาเหตุของปัญหาบางประการได้ เช่น ถ้าค่า Discarded Packets มีจำนวนเพิ่มมากขึ้นขนาดบัฟเฟอร์ของระบบจะลดน้อยลง ถ้าค่า Number Route Discards เพิ่มขึ้นแสดงว่าดาต้าแกรมไม่สามารถส่งผ่านข้อมูลไปยังปลายทางได้ สาเหตุอาจเกิดจากความบกพร่องของเครือข่ายที่หาเส้นทางไม่พบ

สถิติการนับเฟรมเมนต์เดชันที่เกิดขึ้นจากการรีแอสเซมบลีที่สมบูรณ์ หรือการรีแอสเซมบลีที่ผิดพลาด สามารถบ่งบอกได้ถึงประสิทธิภาพของเครือข่ายว่ามีมากเท่าไร ถ้าพบว่า Reassemble Failures และ Fragmentation Failures มีส่วนเสียมาก อาจลองทำการตั้งค่าของ Reassemble Timeout ให้มากขึ้นเพื่อให้เวลาที่รอในการทำรีแอสเซมบลียาวนานขึ้น

7.2.4 การทดสอบแสดงผลข้อมูลไอซีเอ็มพี

วัตถุประสงค์ เพื่อรายงานค่าไอซีเอ็มพี โดยตรวจสอบอุปกรณ์เราเตอร์ Cisco 4500 ติดตั้ง SNMP Agent รุ่น IOS 11.1 จากโปรแกรมคลิกเพื่อเลือกอุปกรณ์เราเตอร์ คลิกเมนูคำสั่ง SNMP-Viewer\Interface Management\Interface\ICMP\ICMP Statistics แสดงผลดังรูปที่ 7.10

```

SNMP-Management Report
File Options
ICMP (Internet Control Message Protocol) Statistics :
[192.168.2.13:161] [Sat Jun 17 18:16:47 SE Asia Standard Time 2000]:
IP-MIB!icmpInMsgs.0 : 732
IP-MIB!icmpInErrors.0 : 0
IP-MIB!icmpInDestUnreachs.0 : 61
IP-MIB!icmpInTimeExcds.0 : 3
IP-MIB!icmpInParmProbs.0 : 0
IP-MIB!icmpInSrcQuenchs.0 : 0
IP-MIB!icmpInRedirects.0 : 1
IP-MIB!icmpInEchos.0 : 320
IP-MIB!icmpInEchoReps.0 : 5
IP-MIB!icmpInTimestamps.0 : 0
IP-MIB!icmpInTimestampReps.0 : 0
IP-MIB!icmpInAddrMasks.0 : 0
IP-MIB!icmpInAddrMaskReps.0 : 0
IP-MIB!icmpOutMsgs.0 : 177866
IP-MIB!icmpOutErrors.0 : 0
IP-MIB!icmpOutDestUnreachs.0 : 178019
IP-MIB!icmpOutTimeExcds.0 : 18
IP-MIB!icmpOutParmProbs.0 : 0
IP-MIB!icmpOutSrcQuenchs.0 : 0
IP-MIB!icmpOutRedirects.0 : 0
IP-MIB!icmpOutEchos.0 : 45
IP-MIB!icmpOutEchoReps.0 : 320
IP-MIB!icmpOutTimestamps.0 : 0
IP-MIB!icmpOutTimestampReps.0 : 0
IP-MIB!icmpOutAddrMasks.0 : 0
IP-MIB!icmpOutAddrMaskReps.0 : 0
  
```

รูปที่ 7.10 แสดงข้อมูลไอซีเอ็มพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลทดสอบการแสดงผลข้อมูลไอซีเอ็มพี สามารถนำมาวิเคราะห์เพื่อให้ประโยชน์การจัดการด้านสมรรถนะ ซึ่งเป็นการค้นคืนค่าออบเจ็กต์กลุ่ม ICMP ของ MIB-II ไอซีเอ็มพีเป็นโปรโตคอลที่ใช้สำหรับรายงานความผิดพลาดที่เกิดจากการประมวลผลของไอพีดาต้าแกรม ดังนั้นการรับค่าของไอซีเอ็มพีเมสเสจจะส่งผลเสียดต่อสมรรถนะโดยรวมของระบบ เพราะไอซีเอ็มพีเมสเสจที่เกิดขึ้นจำนวนมากขณะนั้นต้องการใช้ทรัพยากรของระบบ ออบเจ็กต์กลุ่มนี้แบ่งการนับออกเป็น 2 กลุ่ม คือกลุ่มการรับเข้าไอซีเอ็มพีเมสเสจ และกลุ่มการส่งออกไอซีเอ็มพีเมสเสจ

ออบเจ็กต์ icmpInMsgs และ icmpOutMsgs ซึ่งถึงจำนวนเมสเสจทั้งหมดของการรับและส่ง รวมถึงเมสเสจที่ผิดพลาด ซึ่งอาจไม่มีผลโดยตรงต่อปัญหาสมรรถนะ แต่เป็นข้อมูลทางสถิติที่ช่วยประกอบในการแก้ปัญหา icmpInErrors และ icmpOutErrors ซึ่งถึงการรับส่งเมสเสจที่ผิดพลาด ซึ่งเกิดจากผลรวมตรวจสอบหรือความยาวที่ผิด (Checksum or Length) ค่านี้ถ้าเกิดขึ้นมากแสดงว่ามีไอพีดาต้าแกรมจำนวนมากผิดพลาด ทำให้เกิดปัญหาสมรรถนะของระบบ icmpInDestUnreachs และ icmpOutDestUnreachs ซึ่งถึงการรับส่งเมสเสจที่ไม่สามารถระบุปลายทางได้ ถ้าเราเตอร์เป็นผู้ส่งหมายถึงเราเตอร์ไม่สามารถทำการส่งแพ็กเก็ตไปยังปลายทางได้ อาจเนื่องมาจากไม่มีแอดเดรสของโฮสต์ปลายทาง หรือเราเตอร์ไม่มีเส้นทางที่จะหาไปยังปลายทางได้ icmpInTimeExcds และ icmpOutTimeExcds ซึ่งถึงการรับส่งเมสเสจที่ใช้เวลานานเกินไป อาจเนื่องจากเวลาดังอยู่ (Time to Live) หมดลง หรือคิวในการรีแอสเซมบลียาวเกินไป icmpInParmProbs และ icmpOutParmProbs ซึ่งถึงการรับส่งเมสเสจที่ไอพีเฮดเดอร์มีปัญหา icmpInSrcQuenchs และ icmpOutSrcQuenchs ซึ่งถึงการรับส่งเมสเสจหยุดลงชั่วคราว ถ้าเกิดมากอาจมีปัญหาคอมพิวเตอร์ของเครือข่าย หรือบัฟเฟอร์ไม่เพียงพอ icmpInRedirects และ icmpOutRedirects ซึ่งถึงการรับส่งเมสเสจที่มีการรีไดเร็กต์ การรีไดเร็กต์เมสเสจเราเตอร์ไปยังโฮสต์เพื่อให้ได้ประสิทธิภาพของเส้นทางที่ดีกว่า โดยเราเตอร์ยังคงส่งผ่านแพ็กเก็ตเดิมไปยังปลายทาง การรีไดเร็กต์จะเป็นการส่งไอพีดาต้าแกรม 2 ครั้งบนสื่อเดียวกัน ซึ่งถ้าเกิดอย่างต่อเนื่องจะส่งผลให้สื่อเกิดความแออัด icmpInEchos, icmpOutEchos และ icmpInEchoReps, icmpOutEchoReps ซึ่งถึงการรับส่ง ICMP echo message ที่มีการร้องขอและตอบรับ ซึ่งการใช้คำสั่ง Ping จะเป็นการส่ง echo-request ไปยังโหนดเป้าหมายบนเครือข่าย และรับ echo-reply ที่ตอบรับจากโหนด การทำเช่นนี้อย่างต่อเนื่องจะเพิ่มจำนวน echo message ขึ้นอย่างรวดเร็ว ส่งผลให้เกิดการสิ้นเปลืองทรัพยากรของระบบ icmpInTimestamps, icmpOutTimestamps, icmpInTimestampReps และ icmpOutTimestampReps ซึ่งถึงการรับส่งเมสเสจที่มีการร้องขอและตอบรับตราประทับเวลา icmpInAddrMasks, icmpOutAddrMasks และ icmpInAddrMaskReps, icmpOutAddrMaskReps ซึ่งถึงการรับส่งเมสเสจที่มีการร้องขอและตอบรับแอดเดรสสมาชิก

ออบเจ็กต์กลุ่มนี้เป็นการนับจำนวนของไอซีเอ็มพีเมสเสจ ดังนั้นการระบุเพื่อหาสาเหตุของปัญหาอาจต้องอาศัยเครื่องวิเคราะห์โปรโตคอลเพื่อตรวจสอบแต่ละแพ็กเก็ตของไอซีเอ็มพีเมสเสจ ดังตัวอย่าง แพ็กเก็ตของเมสเสจที่ไม่สามารถระบุปลายทางได้ (Destination Unreachable) สามารถเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งเป็นรหัสที่แตกต่างกัน 13 รหัส เช่น รหัส 0 ซึ่งถึง Network Unreachable หมายถึงเกิดการขัดข้องในเราตังและการกำหนดตำแหน่งแอดเดรส, รหัส 1 ซึ่งถึง Host Unreachable หมายถึงการส่งขัดข้องอาจเนื่องมาจากซบเน็ตมาสก่ผิด เป็นต้น

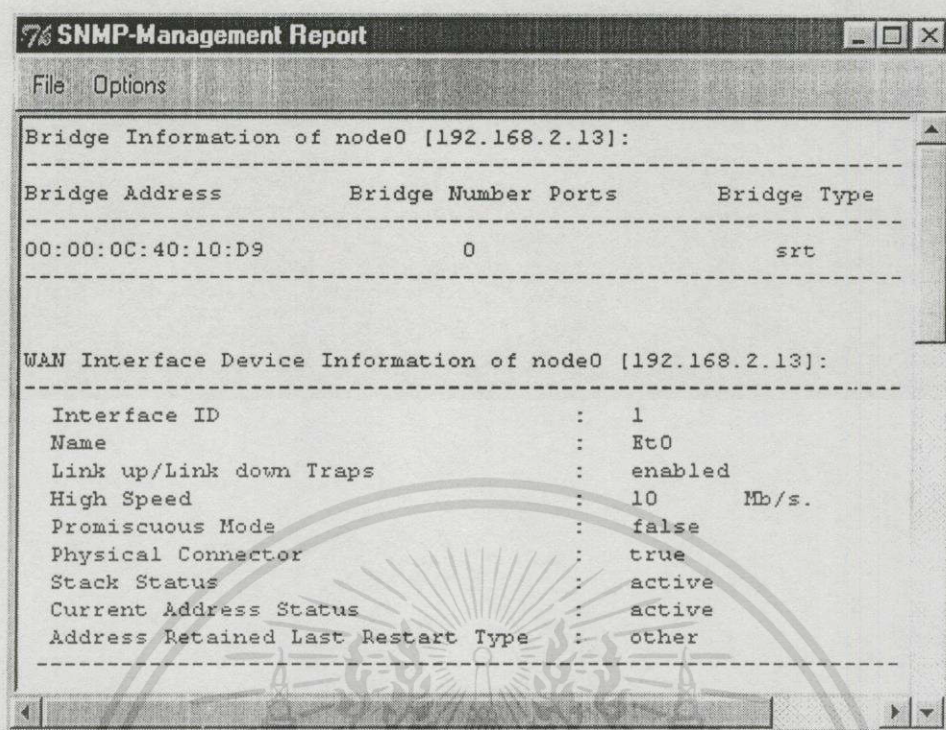
7.2.5 การทดสอบแสดงข้อมูลของอุปกรณ์แลนและแวน

วัตถุประสงค์ เพื่อรายงานค่าทางสถิติ และค่าติดตั้งของอุปกรณ์แลนและแวนอินเทอร์เน็ต โดยตรวจสอบอุปกรณ์เราเตอร์ Cisco 4500 ติดตั้ง Agent รุ่น IOS 11.1 จากโปรแกรมคลิกเพื่อเลือกอุปกรณ์เราเตอร์ คลิกเมนูคำสั่ง SNMP-Viewer\LAN Management\802.3 Ethernet\Ethernet Frame Statistics, \Bridge \ Bridge Information และ \WAN Management \ Information \ WAN Interface Device แสดงดังรูปที่ 7.11 และ 7.12

Ethernet Frame Statistics of node0 [192.168.2.13]:	
Interface ID	: 1
Description	: Ethernet0
Single Collision Frames Sent	: 1999
Multiple Collision Frames Sent	: 3671
Deferred Transmission Frames Sent	: 4555
Time Late Collision Frames Sent	: 0
Excessive Collision Frames Sent	: 0
Internal MAC Transmit Errors	: 0
Carrier Sense Errors	: 0
Alignment Error Frames Received	: 0
FCS Check Errors	: 0
Too Long Frames Received	: 0
Internal MAC Receive Errors	: 0
SQE Test Errors	: 0

รูปที่ 7.11 แสดงข้อมูลสถิติการรับส่งเฟรมของอีเทอร์เน็ต

รูปที่ 7.11 แสดงผลทางด้านสถิติการรับส่งเฟรมของอีเทอร์เน็ต มีประโยชน์ในการจัดการสมรรถนะ ซึ่งเป็นการค้นคืนค่าออบเจ็กต์กลุ่ม Ethernet-Like LAN สามารถบ่งบอกถึงสถิติการนับจำนวนเฟรมการรับส่งที่เป็นผลสำเร็จ จำนวนเฟรมผิดพลาดจากการรับส่งในกรณีต่าง ๆ เช่น เฟรมที่พยายามส่งถูกดีเลย์ เนื่องจากสื่อไม่ว่าง เฟรมที่มีเวลาตรวจสอบพบการชนกัน (ช้ากว่า 512 บิต ไทม์ หรือ 51.2 ms. บนระบบ 10 Mbit/s) เฟรมที่ล้มเหลวจากการรับส่ง เนื่องจากชนกันมากเกินไป หรือจากความผิดพลาดภายใน MAC เฟรมที่ฟังสัญญาณว่าว่างหรือไม่ (Carrier Sense) โดยไม่มีการยืนยันการส่งเฟรม เฟรมที่ไม่ผ่านการตรวจสอบ FCS (Frame Check Sequence) เฟรมที่มีขนาดใหญ่เกินกว่าขนาดที่อนุญาตสูงสุด เฟรมที่ผิดพลาดจาก SQE Test (Signal Quality Error)



รูปที่ 7.12 แสดงข้อมูลของอุปกรณ์ประเภทแลนและแวนอินเทอร์เน็ตเฟซ

รูปที่ 7.12 แสดงผลข้อมูลของบริดจ์ มีประโยชน์ในการจัดการคอนฟิกูเรชัน เป็นการค้นคืนค่าออบเจ็กต์กลุ่ม Bridge สามารถแสดงรายละเอียดของอุปกรณ์ประเภทบริดจ์ เช่น MAC แอดเดรส จำนวนพอร์ต และชนิดของบริดจ์ จากการทดลองคือชนิด srt (Source Routing Transparent)

การแสดงผลสถานะของเราเตอร์ มีประโยชน์ในการจัดการคอนฟิกูเรชัน เป็นการค้นคืนค่าออบเจ็กต์กลุ่ม Evolution สามารถแสดงรายละเอียดที่เกี่ยวข้องกับสถานะของเราเตอร์ เช่น ชื่อของอินเทอร์เน็ตเฟซ สถานะการ LinkUp/LinkDown ความเร็วของอินเทอร์เน็ตเฟซมีหน่วยเป็น Mb/s การใช้หรือไม่ใช้โหมด Promiscuous สถานะการเชื่อมต่อทางกายภาพ สถานะของแอดเดรสปัจจุบัน และการเก็บรักษาสถานะของแอดเดรสไว้แม้มีการเริ่มทำงานใหม่ (Restart)

บทที่ 8

สรุปผลการวิจัยและข้อเสนอแนะ

8.1 สรุปผลการวิจัย

จากการพัฒนาซอฟต์แวร์จัดการเครือข่าย โดยใช้หลักการของเอสเอ็นเอ็มพี ทำให้ได้ศึกษาและเรียนรู้เกี่ยวกับมาตรฐานของออบเจกต์ในกลุ่มต่าง ๆ ของเอ็มไอบี ที่สามารถนำมาใช้ประโยชน์เพื่อการจัดการได้ โดยทำการเขียนซอฟต์แวร์สอบถามไปยังเอเจนต์ของอุปกรณ์เครือข่ายเพื่อติดต่อขอข้อมูลจากออบเจกต์ที่สนใจเพื่อนำมาแสดงผล โดยใช้โปรโตคอลเอสเอ็นเอ็มพีเป็นตัวจัดการซอฟต์แวร์นี้สามารถทำการตรวจสอบอุปกรณ์เครือข่ายที่เป็นฮาร์ดแวร์ หรือจัดการชั้นอินเทอร์เน็ตโปรโตคอลลงมา แต่จะไม่ครอบคลุมการจัดการของชั้นแอปพลิเคชัน รวมถึงซอฟต์แวร์แพคเกจที่เกี่ยวข้องกับการจัดการ ซึ่งถ้าเปรียบเทียบกับซอฟต์แวร์ที่จัดจำหน่ายในเชิงพาณิชย์แล้ว ซอฟต์แวร์ประเภทนี้จะครอบคลุมการจัดการทั้งฮาร์ดแวร์และซอฟต์แวร์ แต่ถ้าต้องการเพียงแค่การจัดการด้านฮาร์ดแวร์ ซอฟต์แวร์ที่พัฒนานี้ก็เพียงพอในการตอบสนองความต้องการนั้นได้

จากการทดลองพบปัญหาบางประการสำหรับการทดสอบซอฟต์แวร์ คือมีออบเจกต์หลายรายการที่ไม่สามารถแสดงผลของข้อมูลออกมาได้ โดยเฉพาะอุปกรณ์เครือข่ายแลนอินเทอร์เฟซและแวนอินเทอร์เฟซ เนื่องจากเอเจนต์ไม่มีรายการของออบเจกต์นั้น ๆ จากการตรวจสอบเอเจนต์พบว่าโดยส่วนใหญ่แล้วอุปกรณ์จะสนับสนุนตามมาตรฐาน MIB-II ทำให้ออบเจกต์อื่นนอกเหนือจากนี้ไม่สามารถทำการตรวจสอบได้ สำหรับผลการทดลองที่ได้สามารถนำมาวิเคราะห์ปัญหาด้านการจัดการเครือข่ายตามมาตรฐาน X.700 ซึ่งกำหนดโดยองค์การมาตรฐานระหว่างประเทศ (ISO) ครอบคลุมปัญหาการจัดการความบกพร่อง, การจัดการคอนฟิกูเรชัน และการจัดการสมรรถนะ แต่ไม่ครอบคลุมการจัดการบัญชี และการจัดการความปลอดภัย ซึ่งผลที่ได้จากการวิเคราะห์นี้จะเป็นประโยชน์เพื่อช่วยเหลือผู้ดูแลระบบหรือวิศวกรระบบ ให้สามารถติดตามและแก้ปัญหาที่เกิดขึ้นบนระบบเครือข่ายได้อย่างรวดเร็ว

8.2 ข้อเสนอแนะและแนวทางการพัฒนา

ซอฟต์แวร์ที่พัฒนาขึ้นมาสามารถทำการตรวจสอบอุปกรณ์ใด ๆ ก็ได้ที่สนใจบนระบบเครือข่าย โดยมีข้อแม้ว่าอุปกรณ์นั้นจะต้องถูกติดตั้งเอเจนต์เสียก่อน ซอฟต์แวร์นี้เหมาะสำหรับใช้ตรวจสอบสถานะของอุปกรณ์ และข้อมูลระบบที่ไม่เปลี่ยนแปลง ส่วนการตรวจสอบข้อมูลทางด้านสถิติจะเป็นการตรวจสอบในช่วงเวลานั้น ๆ ซึ่งการนำผลของข้อมูลที่ได้มาวิเคราะห์ควรจะต้องการเก็บค่าในระยะยาวเพื่อดูแนวโน้ม และทิศทางของข้อมูลว่าเป็นอย่างไร เพื่อการวิเคราะห์จะได้มีความถูกต้องมากยิ่งขึ้น อย่างไรก็ตามวิทยานิพนธ์ฉบับนี้เป็นเพียงแนวทางเพื่อการพัฒนาซอฟต์แวร์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดการเครือข่าย ซึ่งยังคงต้องมีการปรับปรุงและพัฒนาให้มีประสิทธิภาพที่ดียิ่งขึ้นต่อไป ดังนั้น แนวทางการพัฒนาในอนาคตควรเพิ่มการจัดการในชั้นโฮสต์ทูโฮสต์ และชั้นแอปพลิเคชัน เพื่อเพิ่มความสามารถในการจัดการโฮสต์และแอปพลิเคชัน โดยอ้างอิงถึงบริการชั้นขนส่ง เช่น ทีซีพี ยูดีพี และบริการชั้นแอปพลิเคชัน เช่น TELNET, FTP, SMTP และการจัดการสิทธิ ควรเพิ่มเติมการโพลล์ไปยังเอเจนต์ในระยะเวลาอย่างต่อเนื่องเพื่อทำการเก็บค่ารายงานทางสถิติในระยะยาว รวมถึงเพิ่มเติมให้ครอบคลุมหน้าที่การจัดการด้านบัญชี และการจัดการด้านความปลอดภัย



เอกสารอ้างอิง

- [1] นภัทร สระเอี่ยม. “การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี.”
วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2541
- [2] Harnedy, S. J. **Total SNMP : Exploring the Simple Network Management Protocol**. 2nd
ed. New Jersey : Prentice Hall. 1997.
- [3] Miller, M. A. **Managing Internetwork with SNMP: The Definitive Guide to the Simple
Network Management Protocol (SNMP and version2)**. New York : M&T Books.
1993.
- [4] Stallings, W. **SNMP, SNMPv2, and CMIP : the Practical Guide to Network
Management Standards**. Addison-Wesley. 1994.
- [5] Arnett, M.T. et. al. **Inside TCP/IP**. 2nd ed. New Riders. 1995.
- [6] Rose, M. T. and McCloghrie, K. **How to Manage Your Network Using SNMP : The
Networking Management Practicum**. New Jersey : Prentice Hall. 1995.
- [7] Rose, M. T. and McCloghrie, K., Editors. **Management Information Base for Network
Management of TCP/IP-based Internets MIB-II**. Request for Comments 1213.
Hughes LAN Systems. March 1991.
- [8] McCloghrie, K. and Kastenholz, F., Editors. **Evolution of the Interfaces Group of MIB-II**.
Request for Comments 1573. Hughes LAN Systems. January 1994.
- [9] Baker, F., Editors. **The IP Forwarding Table MIB**. Request for Comments 1354.
Advanced Computer Communication. July 1992.
- [10] Kastenholz, F., Editors. **Definitions of Managed Objects for the Ethernet-like Interface
Types**. Request for Comments 1398. FTP Software. January 1993.
- [11] McMaster, D. and McCloghrie, K., Editors. **Definitions of Managed Objects for IEEE
802.3 Repeater Devices**. Request for Comments 1516. SynOptics Communications.
September 1993.
- [12] McMaster, D. et. al., Editors. **Definitions of Managed Objects for IEEE 802.3 Medium
Attachment Units (MAUs)**. Request for Comments 1515. SynOptics Communications.
September 1993.

- [13] Decker, E. et. al., Editors. **Definitions of Managed Objects for Bridges.** Request for Comments 1493. Cisco Systems. July 1993.
- [14] Decker, E. et. al., Editors. **Definitions of Managed Objects for Source Routing Bridges.** Request for Comments 1525. Cisco Systems. September 1993.
- [15] Baker, F. and Watt, J., Editors. **Definitions of Managed Objects for the DS1 and E1 Interface Types.** Request for Comments 1406. Advanced Computer Communications. January 1993.
- [16] Cox, T. and Tesink, K., Editors. **Definitions of Managed Objects for the DS3/E3 Interface Type.** Request for Comments 1407. Bell Communications Research. January 1993.
- [17] Simoneau, P. **SNMP Network Management.** New York : McGraw-Hill. 1999.
- [18] Ajuba Solutions. "Tcl/Tk 8.2." [Online], Available : <http://dev.scriptics.com/software/tcltk/8.2.html>, 1999.
- [19] Schoenwaelder, J. "Scotty-Tcl Extensions for Network Management Applications." [Online], Available : <http://wwwhome.cs.utwente.nl/~schoenw/scotty>, 1999.
- [20] Zeltserman, D. and Puoplo, G. **Building Network Management Tools with Tcl/TK.** New Jersey : Prentice Hall. 1998.
- [21] Leinward, A. and Fang, K. **Network Management A Practical Perspective.** Addison-Wesley. 1993.

ภาคผนวก

โปรแกรมต้นฉบับ

```

# File      : snmp_viewer.tcl
# Purpose:   : Script file for NMS program
# Last Update : 14 Nov 2000.

# Restart using tclsh \
exec tclsh "$0" "$@"
package require Tnm 3.0
package require TnmSnmp $tnm(version)
package require TnmEther $tnm(version)
namespace import Tnm::*
ined size
LoadDefaults snmp
SnmpInit SNMP-Viewer

# Show the system information of MIB II.
proc "System Information" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "System Information of $host \[$ip\]:\n"
        append txt "-----\n"
        try {
            $s walk x "sysName sysLocation sysDescr sysUpTime sysContact
sysObjectID sysServices" {
                set sysName      [lindex [lindex $x 0] 2]
                set sysLocation  [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
                set sysDescr     [lindex [lindex $x 2] 2]
                set sTime        [expr [lindex [lindex $x 3] 2]*0.01]
                set ssTime       [expr fmod($sTime,60)]
                set mTime        [expr $sTime /60]
                set mmTime       [expr fmod($mTime,60)]
                set hTime        [expr $mTime /60]
                set hhTime       [expr fmod($hTime,60)]
                set dTime        [expr $hTime /24]
                set sysUpTime [format "%-2.0f Days %-2.0f Hours %-2.0f Minutes
%-2.0f Seconds" $dTime $hhTime $mmTime $ssTime ]
                set sysContact   [lindex [lindex $x 4] 2]
                set sysObjectID  [lindex [lindex $x 5] 2]
                set sysServices  [lindex [lindex $x 6] 2]
                append txt [format " Device Name : %-25s\n Device Location : %-17s\n
Device Description : %-36s \n Time Since Last Restart : %-s \n Contact
Individual : %-23s \n System Object ID : %-35s\n Set of Services : %-10s \n " \
                $sysName $sysLocation $sysDescr $sysUpTime $sysContact $sysObjectID
                $sysServices]
            }
            append txt "-----\n\n"
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
  } msg {
    append txt "$msg\n"
  }
  writeln $txt
  $$ destroy
}
}

## Start Program for Interface Management.
# Show Interface Detail Information.
proc "Interface Detail Information" {list} {
  ForeachIpNode id ip host $list {
    set s [SnmpOpen $id $ip]
    set txt "Interface Detail Information of $host \[$ip\]:\n"
    append txt "-----\n"
    append txt "Interface Description      MTU  Speed  Physical Address
Admin  Oper  Interface Type\n"
    append txt "----- \n"
    try {
      $$ walk x "ifIndex ifDescr ifMtu ifSpeed ifPhysAddress ifAdminStatus
ifOperStatus ifType" {
        set ifIndex      [lindex [lindex $x 0] 2]
        set ifDescr      [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
        set ifMtu         [lindex [lindex $x 2] 2]
        set ifSpeed       [expr [lindex [lindex $x 3] 2]/1000000]
        set ifPhysAddress [lindex [lindex $x 4] 2]
        set ifAdmin       [lindex [lindex $x 5] 2]
        set ifOper        [lindex [lindex $x 6] 2]
        set ifType        [lindex [lindex $x 7] 2]
        append txt [format "%5s   %-15s %6s %4s M   %18s %6s %6s %s\n" \
          $ifIndex $ifDescr $ifMtu $ifSpeed $ifPhysAddress $ifAdmin $ifOper $ifType]
      } msg {
        append txt "$msg\n"
      }
    }
    append txt "-----\n\n"
    writeln $txt
    $$ destroy
  }
}
}

```

Show interfaces Input Statistics.

```

proc "Interface Input Statistics" {list} {
  ForeachIpNode id ip host $list {
    set s [SnmpOpen $id $ip]
    set txt "Interface Input Statistics of $host \[$ip\]:\n"
    append txt "-----\n"
    try {
      $$ walk x "ifIndex ifDescr ifInOctets ifInUcastPkts ifInNUcastPkts
ifInErrors ifInDiscards ifInUnknownProtos " {

```

```

        set ifIndex          [index [index $x 0] 2]
        set ifDescr         [index [index [index [index $x 1] 2] 0] 0]
        set ifInOct         [index [index $x 2] 2]
        set ifInUOct        [index [index $x 3] 2]
        set ifInNUOct       [index [index $x 4] 2]
        set ifInErrors       [index [index $x 5] 2]
        set ifInDiscard     [index [index $x 6] 2]
        set ifInUnknown     [index [index $x 7] 2]
        append txt [format " Interface ID : %-5s\n Description : %-35s\n Bytes
Received : %-25s\n Unicast Packets : %-25s\n Multicast Packets : %-25s\n Error
Packets : %-25s\n Discarded Packets : %-25s\n Unknown Packets : %-25s\n" \
        $ifIndex $ifDescr $ifInOct $ifInUOct $ifInNUOct $ifInErrors $ifInDiscard
        $ifInUnknown]
        append txt "-----\n" }
    } msg {
        append txt "$msg\n"
    }
    writeln $txt
    $s destroy
}
}

# Show Interface Output Statistics.
proc "Interface Output Statistics" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen Sid $ip]
        set txt "Interface Output Statistics of $host \[$ip]:\n"
        append txt "-----\n"
        try {
            $s walk x "ifIndex ifDescr ifOutOctets ifOutUcastPkts ifOutNUcastPkts
ifOutErrors ifOutDiscards" {
                set ifIndex          [index [index $x 0] 2]
                set ifDescr         [index [index [index [index $x 1] 2] 0] 0]
                set ifOutOct        [index [index $x 2] 2]
                set ifOutUOct       [index [index $x 3] 2]
                set ifOutNUOct      [index [index $x 4] 2]
                set ifOutErrors       [index [index $x 5] 2]
                set ifOutDiscard     [index [index $x 6] 2]
                append txt [format " Interface ID : %-5s\n Description : %-35s\n
Bytes Sent : %-25s\n Unicast Packets : %-25s\n Multicast Packets : %-
25s\n Error Packets : %-25s\n Discarded Packets : %-25s\n" \
                $ifIndex $ifDescr $ifOutOct $ifOutUOct $ifOutNUOct $ifOutErrors $ifOutDiscard ]
                append txt "-----\n" }
            } msg {
                append txt "$msg\n"
            }
            writeln $txt
            $s destroy
        }
    }
}
}

```

```

# Display Interface Quality.
proc "Interface Quality" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen Sid $ip]
        set txt "Interface Quality Statistics of $host \[$ip\]:\n"
        append txt "-----\n"
        try {
            $s walk x "ifIndex ifDescr ifInOctets ifOutOctets ifSpeed ifInUcastPkts
ifInNUcastPkts ifOutUcastPkts ifOutNUcastPkts ifInErrors ifOutErrors ifInDiscards
ifOutDiscards" {
                set ifIndex [lindex [lindex $x 0] 2]
                set ifDescr [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
                set ine [lindex [lindex $x 9] 2]
                set oute [lindex [lindex $x 10] 2]
                set ioe [expr [lindex [lindex $x 9] 2] + [lindex [lindex $x 10] 2]]
                set pkgine [expr [lindex [lindex $x 5] 2] + [lindex [lindex $x 6] 2]]
                set pkgoute [expr [lindex [lindex $x 7] 2] + [lindex [lindex $x 8] 2]]
                set pkge [expr {$pkgine + $pkgoute} ]
                if {$sine == 0 && $pkgine == 0} {
                    set inerr 0
                } else {
                    set inerr [expr {100*double($sine)/$pkgine} ]
                }
                if {$soute == 0 && $pkgoute == 0} {
                    set outerr 0
                } else {
                    set outerr [expr {100*double($soute)/$pkgoute} ]
                }
                if {$sioe == 0 && $pkge == 0} {
                    set err 0
                } else {
                    set err [expr {100*double($sioe)/$pkge} ]
                }
                set ind [lindex [lindex $x 11] 2]
                set outd [lindex [lindex $x 12] 2]
                set iod [expr [lindex [lindex $x 11] 2] + [lindex [lindex $x 12] 2] ]
                if {$sind == 0 && $pkgine == 0} {
                    set indis 0
                } else {
                    set indis [expr {100*double($sind)/$pkgine} ]
                }
                if {$soutd == 0 && $pkgoute == 0} {
                    set outdis 0
                } else {
                    set outdis [expr {100*double($soutd)/$pkgoute} ]
                }
                if {$siod == 0 && $pkge == 0} {
                    set dis 0
                } else {
                    set dis [expr {100*double($siod)/$pkge} ] }
            }
        }
    }
}

```

```

        append txt [format " Interface ID : %-5s\n Description : %-35s\n Input
Errors : %-5.2f %%\n Output Errors : %-5.2f %%\n Total Error : %5.2f %%\n
Input Discards : %-5.2f %%\n Output Discards : %-5.2f %%\n Total Discards
: %-5.2f %%\n " $ifIndex $ifDescr $inerr $outerr $err $indis $outdis $dis]
        append txt "-----\n"
    } msg {
        append txt "$msg\n"
    }
    writeln $txt
    $s destroy
}
}

```

Show the IP Scalars.

```

proc "IP Statistics" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "IP (Internet Protocol) Statistics of $host \[$ip]:\n"
        append txt "-----\n"
        try {
            $s walk x "ipForwarding ipInReceives ipOutRequests ipInDelivers
ipForwDatagrams ipReasmReqds ipFragCreates ipReasmOKs ipFragOKs
ipReasmTimeout ipDefaultTTL ipInAddrErrors ipInHdrErrors ipInDiscards
ipOutDiscards ipReasmFails ipInUnknownProtos ipFragFails ipOutNoRoutes" {
                set ipForwarding [index [index $x 0] 2]
                set ipInReceives [index [index [index [index $x 1] 2] 0] 0]
                set ipOutRequests [index [index $x 2] 2]
                set ipInDelivers [index [index $x 3] 2]
                set ipForwDatagrams [index [index $x 4] 2]
                set ipReasmReqds [index [index $x 5] 2]
                set ipFragCreates [index [index $x 6] 2]
                set ipReasmOKs [index [index $x 7] 2]
                set ipFragOKs [index [index $x 8] 2]
                set ipReasmTimeout [index [index $x 9] 2]
                set ipDefaultTTL [index [index $x 10] 2]
                set ipInAddrErrors [index [index $x 11] 2]
                set ipInHdrErrors [index [index $x 12] 2]
                set ipInDiscards [expr [index [index $x 13] 2]+[index [index $x 14] 2]]
                set ipReasmFails [index [index $x 15] 2]
                set ipInUnknownProtos [index [index $x 16] 2]
                set ipFragFails [index [index $x 17] 2]
                set ipOutNoRoutes [index [index $x 18] 2]
                append txt [format " Forwarding Status : %-25s\n Packets Received
: %-25s\n Output Requests : %-25s\n Delivered Packets : %-25s\n
Packets Forwarded : %-25s\n Fragments Reassembled : %-25s\n Fragments
Created : %-25s\n Fragments Received : %-25s\n Packets Fragments OK
: %-25s\n Reassemble Timeout : %-25s\n Default Time To Live : %-25s\n
Addressing Errors : %-25s\n Input Header Errors : %-25s\n Discarded
Packets : %-25s\n Reassemble Failures : %-25s\n Unknown Protocols
: %-25s\n Fragmentation Failures : %-25s\n Number Route Discards : %-25s\n" \

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SipForwarding SipInReceives SipOutRequests SipInDelivers SipForwDatagrams
SipReasmReqds SipFragCreates SipReasmOKs SipFragOKs SipReasmTimeout
SipDefaultTTL SipInAddrErrors SipInHdrErrors SipInDiscards SipReasmFails
SipInUnknownProtos SipFragFails SipOutNoRoutes ]

```

```

    append txt "-----\n\n" }

```

```

} msg {

```

```

    append txt "$msg\n"

```

```

}

```

```

writeln $txt

```

```

$S destroy

```

```

}

```

```

}

```

```

# Get the IP addresses belonging to this device.

```

```

proc "IP Addresses" {list} {

```

```

    ForeachIpNode id ip host $list {

```

```

        set s [SnmpOpen $id $ip]

```

```

        set txt "IP addresses of $host \[$ip]:\n"

```

```

        append txt "-----\n\n"

```

```

        append txt "Interface Description      IP Address      Subnet Mask

```

```

Broadcast  Max Reassemble\n"

```

```

        append txt "-----\n\n"

```

```

        try {

```

```

            $S walk x "ifIndex ifDescr" {

```

```

                set ifDescr([lindex [lindex $x 0] 2]) [lindex [lindex $x 1] 2] }

```

```

        }

```

```

        try {

```

```

            set list ""

```

```

            $S walk x {ipAdEntAddr ipAdEntNetMask ipAdEntBcastAddr
ipAdEntIfIndex ipAdEntReasmMaxSize} {

```

```

                set ipAdEntAddr      [lindex [lindex $x 0] 2]

```

```

                set ipAdEntNetMask    [lindex [lindex $x 1] 2]

```

```

                set ipAdEntBcastAddr  [lindex [lindex $x 2] 2]

```

```

                set ipAdEntIfIndex    [lindex [lindex $x 3] 2]

```

```

                set ipAdEntReasmMaxSize [lindex [lindex $x 4] 2]

```

```

            if [info exists ifDescr($ipAdEntIfIndex)] {

```

```

                set name $ifDescr($ipAdEntIfIndex)

```

```

            } else {

```

```

                set name ""

```

```

            }

```

```

            lappend list [format "%4s    %-28s %-15s %-15s %6s    %6s" \

```

```

                $ipAdEntIfIndex $name $ipAdEntAddr $ipAdEntNetMask

```

```

                $ipAdEntBcastAddr $ipAdEntReasmMaxSize] }

```

```

            append txt "[join [lsort $list] "\n"]\n"

```

```

        } msg {

```

```

            append txt "$msg\n"

```

```

        }

```

```

        append txt "-----\n\n"

```

```

        writeln $txt

```

```

        $S destroy

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

# Show the routing tables of the agents.
proc "IP Routing Table" {list} {
    foreachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "Routing Table of $host \[$ip\]:\n"
        append txt "-----\n"
        append txt "Destination    Subnet Mask    Next Hop    Interface    Metric\n"
        Type Protocol Age\n"
        append txt "-----\n"
        try {
            $s walk x {ipRouteDest ipRouteMask ipRouteNextHop ipRouteIfIndex
            ipRouteMetric1 ipRouteType ipRouteProto ipRouteAge} {
                set ipRouteDest [lindex [lindex $x 0] 2]
                set ipRouteMask [lindex [lindex $x 1] 2]
                set ipRouteNext [lindex [lindex $x 2] 2]
                set ipRouteIfIndex [lindex [lindex $x 3] 2]
                set ipRouteMetric1 [lindex [lindex $x 4] 2]
                set ipRouteType [lindex [lindex $x 5] 2]
                set ipRouteProto [lindex [lindex $x 6] 2]
                set ipRouteAge [lindex [lindex $x 7] 2]
                append txt [format "%-16s %-16s %-16s %-4s %-5s %-10s%-7s %-8s\n" \
                $ipRouteDest $ipRouteMask $ipRouteNext $ipRouteIfIndex $ipRouteMetric1
                $ipRouteType $ipRouteProto $ipRouteAge]
            }
        } msg {
            append txt "$msg\n"
        }
        append txt "-----\n\n"
        writeln $txt
        $s destroy
    }
}
}

```

```

# Show the routing tables of the agents.
proc "IP ARP Table" {list} {
    global vendor
    foreachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "ARP Table of $host \[$ip\]:\n"
        append txt "-----\n"
        append txt "Interface    MAC Address    IP Address    ARP Type    Vendor\n"
        append txt "-----\n"
        set done 0
        $s walk x {ipNetToMediaIfIndex ipNetToMediaPhysAddress
        ipNetToMediaNetAddress ipNetToMediaType} {
            set ipNetToMediaIfIndex [lindex [lindex $x 0] 2]
            set ipNetToMediaPhysAddress [lindex [lindex $x 1] 2]
            set ipNetToMediaNetAddress [lindex [lindex $x 2] 2]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set ipNetToMediaType          [lindex [lindex $x 3] 2]
set done 1
set vend [TnmEther::GetVendor $ipNetToMediaPhysAddress]
append txt [format "%5s   %-18s %-16s %-10s %s\n" \
$ipNetToMediaIfIndex $ipNetToMediaPhysAddress \
$ipNetToMediaNetAddress $ipNetToMediaType $vend]
}
if { !$done } {
    $s walk x {atIfIndex atPhysAddress atNetAddress} {
        set atIfIndex          [lindex [lindex $x 0] 2]
        set atPhysAddress      [lindex [lindex $x 1] 2]
        set atNetAddress       [lindex [lindex $x 2] 2]
        set vend [TnmEther::GetVendor $atPhysAddress]
        append txt [format "%8s %-18s %-18s %s\n" \
$atIfIndex $atPhysAddress $atNetAddress $vend]
    }
}
append txt "-----\n\n"
writeln $txt
$s destroy
}
}

# Show the IP Quality of the agents.
proc "IP Quality" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmOpen $id $ip]
        set txt "IP(Internet Protocol)Quality Statistics of Shost \[$ip\]:\n"
        append txt "-----\n"
        append txt "Interface      Description          IP Input Errors \n"
        append txt "-----\n"
        try {
            $s walk x {ifIndex ifDescr ipInDiscards ipInHdrErrors ipInAddrErrors
ipInReceives} {
                set ifIndex          [lindex [lindex $x 0] 2]
                set ifDescr          [lindex [lindex $x 1] 2]
                set ipInDiscards     [lindex [lindex $x 2] 2]
                set ipInHdrErrors    [lindex [lindex $x 3] 2]
                set ipInAddrErrors   [lindex [lindex $x 4] 2]
                set ipInReceives     [lindex [lindex $x 5] 2]
                if { $ipInDiscards == 0 && $ipInHdrErrors == 0 && $ipInAddrErrors
== 0 && $ipInReceives == 0 } {
                    set inputerror 0
                } else {
                    set inputerror [expr {100*double($ipInDiscards +
$ipInHdrErrors + $ipInAddrErrors) / $ipInReceives } ]
                }
                append txt [format "  %-10s %-30s %-5.2f %%\n" \
$ifIndex $ifDescr $inputerror]
            }
        }
        append txt "-----\n\n"
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        writeln $txt
        $$ destroy
    }
}

# Show the ICMP Scalars.
proc "ICMP Statistics" {list} {
    writeln "ICMP (Internet Control Message Protocol) Statistics :'"
    ShowScalars $list icmp
}

# Start Program for LAN Management.
# Show the Ether-like LANS for Ethernet Frame Transmission Statistics.
proc "Frame Transmission Statistics" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen Sid $ip]
        set txt "Ethernet Frame Transmission Statistics of $host \[$ip\]:\n"
        append txt "-----\n"
        append txt "ID Description      SColls  MColls  DeferXmt  LateColls
ExColls  InMacXmt  CarSenseErs \n"
        append txt "-----\n"
        try {
            $$ walk x "ifIndex ifDescr dot3StatsSingleCollisionFrames
dot3StatsMultipleCollisionFrames dot3StatsDeferredTransmissions
dot3StatsLateCollisions dot3StatsExcessiveCollisions
dot3StatsInternalMacTransmitErrors dot3StatsCarrierSenseErrors" {
                set ifIndex [lindex [lindex $x 0] 2]
                set ifDescr [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
                set SColls [lindex [lindex $x 2] 2]
                set MColls [lindex [lindex $x 3] 2]
                set DeferXmt [lindex [lindex $x 4] 2]
                set LateColls [lindex [lindex $x 5] 2]
                set ExColls [lindex [lindex $x 6] 2]
                set InMacXmt [lindex [lindex $x 7] 2]
                set CarSenseErs [lindex [lindex $x 8] 2]
                append txt [format "%2s %-12s %10s %10s %10s %10s %10s %10s
%10s\n" $ifIndex $ifDescr $$SColls $$MColls $$DeferXmt $$LateColls $$ExColls
$$InMacXmt $$CarSenseErs ] }
            } msg {
                append txt "$msg\n"
            }
            append txt "-----\n\n"
            writeln $txt
            $$ destroy
        }
    }
}

```

```

# Show the Ether-like LANS for Ethernet Frame Reception Statistics
proc "Frame Reception Statistics" {list} {

```

```

    ForeachIpNode id ip host $list {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set s [SnmOpen $id $ip]
set txt "Ethernet Frame Reception Statistics of $host \[$ip\]:\n"
append txt "-----\n"
append txt "ID Description      AlignErs  FCSErs   TooLong  InMacRecErs
SQETestErs \n"
append txt "-----\n"
try {
    $s walk x "ifIndex ifDescr dot3StatsAlignmentErrors dot3StatsFCSErrors
dot3StatsFrameTooLongs dot3StatsInternalMacReceiveErrors
dot3StatsSQETestErrors" {
        set ifIndex      [lindex [lindex $x 0] 2]
        set ifDescr      [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
        set AlignErs     [lindex [lindex $x 2] 2]
        set FCSErs       [lindex [lindex $x 3] 2]
        set TooLong      [lindex [lindex $x 4] 2]
        set InMacRecErs  [lindex [lindex $x 5] 2]
        set SQETestErs   [lindex [lindex $x 6] 2]
        append txt [format "%2s %-12s %8s %8s %8s %8s %10s \n"
$ifIndex $ifDescr $AlignErs $FCSErs $TooLong $InMacRecErs
$SQETestErs]
    } msg {
        append txt "$msg\n"
    }
    append txt "-----\n\n"
    writeln $txt
    $s destroy
}
}

```

Show the Ether-like LANS for Ethernet Frame Statistics

```

proc "Ethernet Frame Statistics" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmOpen $id $ip]
        set txt "Ethernet Frame Statistics of $host \[$ip\]:\n"
        append txt "-----\n"
        try {
            $s walk x "ifIndex ifDescr dot3StatsSingleCollisionFrames
dot3StatsMultipleCollisionFrames dot3StatsDeferredTransmissions
dot3StatsLateCollisions dot3StatsExcessiveCollisions
dot3StatsInternalMacTransmitErrors dot3StatsCarrierSenseErrors
dot3StatsAlignmentErrors dot3StatsFCSErrors dot3StatsFrameTooLongs
dot3StatsInternalMacReceiveErrors dot3StatsSQETestErrors" {
                set ifIndex      [lindex [lindex $x 0] 2]
                set ifDescr      [lindex [lindex [lindex [lindex $x 1] 2] 0] 0]
                set dot3StatsSingleCollisionFrames      [lindex [lindex $x 2] 2]
                set dot3StatsMultipleCollisionFrames    [lindex [lindex $x 3] 2]
                set dot3StatsDeferredTransmissions      [lindex [lindex $x 4] 2]
                set dot3StatsLateCollisions             [lindex [lindex $x 5] 2]
                set dot3StatsExcessiveCollisions        [lindex [lindex $x 6] 2]
                set dot3StatsInternalMacTransmitErrors  [lindex [lindex $x 7] 2]
            }
        }
    }
}

```

```

set dot3StatsCarrierSenseErrors      [index [index $x 8] 2]
set dot3StatsAlignmentErrors         [index [index $x 9] 2]
set dot3StatsFCSErrors               [index [index $x 10] 2]
set dot3StatsFrameTooLongs          [index [index $x 11] 2]
set dot3StatsInternalMacReceiveErrors [index [index $x 12] 2]
set dot3StatsSQETestErrors           [index [index $x 13] 2]
    append txt [format " Interface ID : %-25s\n Description : %-25s\n Single
Collision Frames Sent : %-25s\n Multiple Collision Frames Sent : %-25s\n
Deferred Transmission Frames Sent : %-25s\n Time Late Collision Frames Sent :
%-25s\n Excessive Collision Frames Sent : %-25s\n Internal MAC Transmit Errors
: %-25s\n Carrier Sense Errors : %-25s\n Alignment Error Frames Received :
%-25s\n FCS Check Errors : %-25s\n Too Long Frames Received : %-25s\n
Internal MAC Receive Errors : %-25s\n SQE Test Errors : %-25s\n " \
    $ifIndex $ifDescr $dot3StatsSingleCollisionFrames
$dot3StatsMultipleCollisionFrames $dot3StatsDeferredTransmissions
$dot3StatsLateCollisions $dot3StatsExcessiveCollisions
$dot3StatsInternalMacTransmitErrors $dot3StatsCarrierSenseErrors
$dot3StatsAlignmentErrors $dot3StatsFCSErrors $dot3StatsFrameTooLongs
$dot3StatsInternalMacReceiveErrors $dot3StatsSQETestErrors ]
    append txt "-----\n\n"
}
} msg {
    append txt "$msg\n"
}
writeln $txt
$S destroy
}
}

# Show the Repeater Information
proc "Repeater Information" {list} {
    writeln "Repeater Information : "
    ShowScalars $list rptrRptrInfo
}

# Show the Repeater Port Status
proc "Repeater Port Status" {list} {
    writeln "Repeater Port Status : "
    ShowTable $list rptrPortTable
}

# Show the Repeater Group Information
proc "Repeater Group Information" {list} {
    writeln "Repeater Group Information : "
    ShowTable $list rptrGroupTable
}

# Show the Repeater Monitor Information
proc "Repeater Monitor Information" {list} {
    writeln "Repeater Monitor Information : "

```

```

ShowScalars $list rpTrMonitorRptrInfo
}

# Show the Repeater Monitor Port Statistics
proc "Repeater Monitor Port Statistics" {list} {
    writeln "Repeater Monitor Port Statistics : "
    ShowTable $list rpTrMonitorPortTable
}

# Show the Repeater Monitor Group Statistics
proc "Repeater Monitor Group Statistics" {list} {
    writeln "Repeater Monitor Group Statistics : "
    ShowTable $list rpTrMonitorGroupTable
}

# Show the Repeater Address Tracking Status
proc "Repeater Address Tracking Status" {list} {
    writeln "Repeater Address Tracking Status : "
    ShowTable $list rpTrAddrTrackTable
}

# Show the Medium Attachment Unit (MAU) for Repeater MAU Status
proc "Repeater MAU Status" {list} {
    writeln "Repeater MAU Status : "
    ShowTable $list rpMauTable
}

# Show the Medium Attachment Unit (MAU) for Interface MAU Status
proc "Interface MAU Status" {list} {
    writeln "Interface MAU Status : "
    ShowTable $list ifMauTable
}

# Show the Medium Attachment Unit (MAU) for Broadband MAU Information
proc "Broadband MAU Information" {list} {
    writeln "Broadband MAU Information : "
    ShowTable $list BroadMauBasicTable
}

# Show the Bridge for Bridge Information (Basic Group)
proc "Bridge Information" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmOpen $id $ip]
        set txt "Bridge Information of $host \[$ip\]:\n"
        append txt "-----\n"
        append txt "Bridge Address      Bridge Number Ports      Bridge Type  \n"
        append txt "-----\n"
        try {
            $$s walk x {dot1dBaseBridgeAddress dot1dBaseNumPorts dot1dBaseType} {
                set dot1dAdd [lindex [lindex $x 0] 2]

```

```

        set dot1Port [lindex [lindex $x 1] 2]
        set dot1Type [lindex [lindex $x 2] 2]
        append txt [format "%16s %12s %20s \n" $dot1Add $dot1Port $dot1Type]
    }
} msg {
    append txt "$msg\n"
}
append txt "-----\n\n"
writeln $txt
$s destroy
}
}

# Show the Bridge for STP(Spanning Tree Bridge) Status Port
proc "STP Status Port" {list} {
    writeln "STP Status Port :"
    ShowTable $list dot1dStpPortTable
}

# Show the Bridge for TP (Transparent)Bridge's Forwarding Database
proc "TP Forwarding Database" {list} {
    writeln "Transparent Bridge Forwarding Database :"
    ShowTable $list dot1dTpFdbTable
}

# Show the Bridge for TP (Transparent)Bridge Port Statistics
proc "TP Port Statistics" {list} {
    writeln "Transparent Bridge Port Statistics :"
    ShowTable $list dot1dTpPortTable
}

# Show the Bridge for Static Group Information
proc "Static Group Information" {list} {
    writeln "Static Information :"
    ShowTable $list dot1dStaticTable
}

# Show the Source Routing Bridge for Source Route Group (Status & Statistics)
proc "Source Route Port Information" {list} {
    writeln "Source Route Port Information :"
    ShowTable $list dot1SrPortTable
}

# Show the Source Routing Bridge for Port-Pair Group
proc "Port-Pair Information" {list} {
    writeln "Port-Pair Information :"
    ShowTable $list dot1dPortPairTable
}
}
}

```

```

# Start Program for WAN Management.
# Show generic information for all types of interface.
proc "Information Device" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "Information Device of $host \[$ip\]:\n"
        append txt "-----\n"
        append txt "ID   Name LinkTrap HighSpeed Promiscuous Connector
StackStatus RcvAddrStatus RcvAddrType   \n"
        append txt "-----\n"
        try {
            $s walk x "ifIndex ifName ifLinkUpDownTrapEnable ifHighSpeed
ifPromiscuousMode ifConnectorPresent ifStackStatus ifRcvAddressStatus
ifRcvAddressType" {
                set ifIndex           [lindex [lindex $x 0] 2]
                set ifName            [lindex [lindex $x 1] 2]
                set ifLink            [lindex [lindex $x 2] 2]
                set ifHighSpeed       [lindex [lindex $x 3] 2]
                set ifPromis          [lindex [lindex $x 4] 2]
                set ifConnect         [lindex [lindex $x 5] 2]
                set ifStack           [lindex [lindex $x 6] 2]
                set ifRcvStatus       [lindex [lindex $x 7] 2]
                set ifRcvType         [lindex [lindex $x 8] 2]
                append txt [format "%2s %6s %10s %8s %12s %12s %12s %12s %12s\n"
SifIndex $ifName $ifLink $ifHighSpeed $ifPromis $ifConnect $ifStack $ifRcvStatus
SifRcvType ] }
            } msg {
                append txt "$msg\n" }
            append txt "-----\n\n"
            writeln $txt
            $s destroy
        }
    }
}

```

```

# Show generic information for WAN Interface Device.
proc "WAN Interface Device" {list} {
    ForeachIpNode id ip host $list {
        set s [SnmpOpen $id $ip]
        set txt "WAN Interface Device Information of $host \[$ip\]:\n"
        append txt "-----\n"
        try {
            $s walk x "ifIndex ifName ifLinkUpDownTrapEnable ifHighSpeed
ifPromiscuousMode ifConnectorPresent ifStackStatus ifRcvAddressStatus
ifRcvAddressType " {
                set ifIndex           [lindex [lindex $x 0] 2]
                set ifName            [lindex [lindex $x 1] 2]
                set ifLinkUpDownTrapEnable [lindex [lindex $x 2] 2]
                set ifHighSpeed       [lindex [lindex $x 3] 2]
                set ifPromiscuousMode [lindex [lindex $x 4] 2]
                set ifConnectorPresent [lindex [lindex $x 5] 2]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set ifStackStatus [index [index $x 6] 2]
set ifRcvAddressStatus [index [index $x 7] 2]
set ifRcvAddressType [index [index $x 8] 2]
append txt [format " Interface ID : %-25s\n Name
: %-25s\n Link up/Link down Traps : %-25s\n High Speed
: %-5s Mb/s.\n Promiscuous Mode : %-25s\n Physical Connector
: %-25s\n Stack Status : %-25s\n Current Address Status :
%-25s\n Address Retained Last Restart Type : %-25s\n \"
$ifIndex $ifName $ifLinkUpDownTrapEnable $ifHighSpeed $ifPromiscuousMode
$ifConnectorPresent $ifStackStatus $ifRcvAddressStatus $ifRcvAddressType ]
append txt "-----\n\n" }
} msg {
append txt "$msg\n"
}
writeln $txt
$S destroy
}
}

# Show the DS1 Near End Group Configuration Information
proc "DS1 Near End" {list} {
writeln "DS1 Near End Configuration Information :\"
ShowTable $list dsx1ConfigTable
}

# Show the DS1 Near End Error Counts
proc "DS1 Near End Errors" {list} {
writeln "DS1 Near End Error Counts :\"
ShowTable $list dsx1CurrentTable
ShowTable $list dsx1IntervalTable
ShowTable $list dsx1TotalTable
}

# Show the DS1 Far End Group Information
proc "DS1 Far End" {list} {
writeln "DS1 Far End Information :\"
ShowTable $list dsx1FarEndCurrentTable
}

# Show the DS1 Far End Error Counts
proc "DS1 Far End Errors" {list} {
writeln "DS1 Far End Error Counts :\"
ShowTable $list dsx1FarEndCurrentTable
ShowTable $list dsx1FarEndIntervalTable
ShowTable $list dsx1FarEndTotalTable
}

# Show the DS1 Fractional Group Information
proc "DS1 Fractional" {list} {

```

```

        writeln "DS1 Fractional Information :"  

        ShowTable $list dsx1FracTable  

    }  

# Show the DS3 Near End Group Configuration Information  

proc "DS3 Near End" {list} {  

    writeln "DS3 Near End Configuration Information :"  

    ShowTable $list dsx3ConfigTable  

}  

# Show the DS3 Near End Error Counts  

proc "DS3 Near End Errors" {list} {  

    writeln "DS3 Near End Error Counts :"  

    ShowTable $list dsx3CurrentTable  

    ShowTable $list dsx3IntervalTable  

    ShowTable $list dsx3TotalTable  

}  

# Show the DS3 Far End Group Information  

proc "DS3 Far End" {list} {  

    writeln "DS3 Far End Information :"  

    ShowTable $list dsx3FarEndCurrentTable  

}  

# Show the DS3 Far End Error Counts  

proc "DS3 Far End Errors" {list} {  

    writeln "DS3 Far End Error Counts :"  

    ShowTable $list dsx3FarEndCurrentTable  

    ShowTable $list dsx3FarEndIntervalTable  

    ShowTable $list dsx3FarEndTotalTable  

}  

# Show the DS3 Fractional Group Information  

proc "DS3 Fractional" {list} {  

    writeln "DS3 Fractional Information :"  

    ShowTable $list dsx3FracTable  

}  

# Exit the menus created by this interpreter.  

proc "Exit SNMP-Viewer" {list} {  

    global menus  

    foreach id $menus { ined delete $id }  

    exit  

}  

set menus [ ined create MENU "SNMP-Viewer" \  

    "System :System Information" "" \  

    "Interface Management:Interface:Interface Detail Information" \  

    "Interface Management:Interface:Interface Input Statistics" \  

    "Interface Management:Interface:Interface Output Statistics" \  

    "Interface Management:Interface:Interface Quality" \  


```

```

"Interface Management:IP:IP Statistics" \
"Interface Management:IP:IP Addresses" \
"Interface Management:IP:IP Routing Table" \
"Interface Management:IP:IP ARP Table" \
"Interface Management:IP:IP Quality" \
"Interface Management:ICMP:ICMP Statistics" "" \
"LAN Management:802.3 Ethernet:Frame Transmission Statistics" \
"LAN Management:802.3 Ethernet:Frame Reception Statistics" \
"LAN Management:802.3 Ethernet:Ethernet Frame Statistics" \
"LAN Management:802.3 Repeater:Repeater Information" \
"LAN Management:802.3 Repeater:Repeater Port Status" \
"LAN Management:802.3 Repeater:Repeater Group Information" \
"LAN Management:802.3 Repeater:Repeater Monitor Information" \
"LAN Management:802.3 Repeater:Repeater Monitor Port Statistics" \
"LAN Management:802.3 Repeater:Repeater Monitor Group Statistics" \
"LAN Management:802.3 Repeater:Repeater Address Tracking Status" \
"LAN Management:802.3 MAU:Repeater MAU Status " \
"LAN Management:802.3 MAU:Interface MAU Status " \
"LAN Management:802.3 MAU:Broadband MAU Information" \
"LAN Management:Bridge:Bridge Information" \
"LAN Management:Bridge:STP Status Port" \
"LAN Management:Bridge:TP Forwarding Database" \
"LAN Management:Bridge:TP Port Statistics" \
"LAN Management:Bridge:Static Group Information" \
"LAN Management:Bridge:Source Route Port Information" \
"LAN Management:Bridge:Port-Pair Information" "" \
"WAN Management:Information:Information Device" \
"WAN Management:Information:WAN Interface Device" \
"WAN Management:DS1:DS1 Near End" \
"WAN Management:DS1:DS1 Near End Errors" \
"WAN Management:DS1:DS1 Far End" \
"WAN Management:DS1:DS1 Far End Errors" \
"WAN Management:DS1:DS1 Fractional" \
"WAN Management:DS3:DS3 Near End" \
"WAN Management:DS3:DS3 Near End Errors" \
"WAN Management:DS3:DS3 Far End" \
"WAN Management:DS3:DS3 Far End Errors" \
"WAN Management:DS3:DS3 Fractional" "" \
"Exit SNMP-Viewer" ]

```

vwait forever

#####

```
# File      : snmp_monitor.tcl
# Purpose:  : Script file for NMS program
# Last Update : 05 Jan 2001.
```

```
# Restarts using tclsh \
exec tclsh "$0" "$@"
package require Tnm 3.0
namespace import Tnm::*
ined size
LoadDefaults snmp monitor
SnmpInit SNMP-monitor
```

```
if {[info exists default(interval)]} {
    set interval $default(interval)
} else {
    set interval 60
}
```

```
if {![info exists default(graph)]} {
    set default(graph) false
}
```

```
# Create a chart.
```

```
proc CreateChart {id x y} {
    global default
    if {$default(graph) != "true"} {
        set id [CloneNode Sid [ined create STRIPCHART] $x $y]
    } else {
        set id [CloneNode Sid [ined create GRAPH]]
    }
    return Sid
}
```

```
# Tell tkined to restart a command when a saved map gets loaded
```

```
proc save { cmd } {
    set cmdList [ined restart]
    if {[lsearch $cmdList $cmd] >= 0} return
    lappend cmdList $cmd
    ined restart $cmdList
}
```

```
# Remove a command from the list of commands saved in the tkined map.
```

```
proc forget { cmd } {
    set cmdList ""
    foreach c [ined restart] {
        if {$c != $cmd} {
            lappend cmdList $c
        }
    }
    ined restart $cmdList
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Restart a monitor job.
proc restart { cmd args } {
    global interval
    if {![catch {expr [lindex $args 0] + 0}]} {
        set itv [lindex $args 0]
        set jid [lindex $args 1]
        set args [lrange $args 2 end]
        set old_interval $interval
        set interval [expr int($itv)]
    }
    foreach id $args {
        catch {ined -noupdate clear $id}
        catch {ined -noupdate clear $id}
    }
    catch {eval $cmd $args}
    if {[info exists old_interval]} {
        set interval $old_interval
    }
}

# Start a new monitoring job.
proc start_snmp_monitor { ip uptime args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$args != ""} {
        set id [lindex $args 0]
        set var [lindex $args 1]
        if {$id != ""} {
            set s [SnmpOpen $id $ip]
            if {[catch {$s get [list $uptime $var]} vbl]} {
                set name [ined -noupdate name $id]
                set addr [ined -noupdate address $id]
                set msg "Unable to restart monitor job on $name"
                if {$addr != ""} {
                    append msg " \[$addr\]"
                }
                writeln "$msg.\nSNMP get failed: $vbl\n"
                $s destroy
                set args [lrange $args 2 end]
                ined delete $id
                continue
            }
            set cx(time,$id) [lindex [lindex $vbl 0] 2]
            set cx(value,$id) [lindex [lindex $vbl 1] 2]
            set cx(snmp,$id) $s
            set cx(name,$id) $var
            set cx(descr,$id) [mib name [mib oid $var]]
            set cx(label,$id) [mib label [mib oid $var]]
        }
    }
}

```

```

lappend ids $id
  inedit -noupdate attribute $id $cx(descr,$id) $cx(descr,$id)
  inedit -noupdate label $id $cx(descr,$id)
}
set args [lrange $args 2 end]
append re_args "\$id $var"
}
if {$sids != ""} {
  set jid [job create -command [list ShowVariable $sids] \
    -interval [expr $interval * 1000]]
  save "restart start_snmp_monitor $interval $jid $ip $uptime $re_args"
}
}

# Prepare a monitoring job.
proc MonitorVariable {list uptime varname} {
  foreach ipnode id ip host $list {
    set s [SnmpOpen $id $ip]
    set syntaxList {
      Gauge Gauge32 Counter Counter32 Counter64
      Unsigned32 INTEGER Integer32
    }
    set err [catch {$s get $uptime} vbl]
    if {$err || [snmp type $vbl 0] == "noSuchInstance"} {
      inedit acknowledge "Time base problem on $ip: $vbl"
      $s destroy
      return ""
    }
    set candidates {}
    set err [catch {$s get $varname} vbl]
    if {!( $err && [snmp type $vbl 0] != "noSuchInstance" )} {
      lappend candidates $vbl
    } else {
      if {[catch {
        $s walk vbl $varname {
          lappend candidates $vbl
        }
      } msg]} {
        inedit acknowledge "Monitor Variable $varname on $ip: $msg"
        $s destroy
        return ""
      }
    }
  }
  set vars {}
  foreach vbl $candidates {
    set oid [snmp oid $vbl 0]
    set type [snmp type $vbl 0]
    if {[lsearch $syntaxList $type] < 0} {
      writeln "$host: [mib name $oid] of base type $type ignored"
      continue
    }
  }
}

```

```

    }
    set enums [mib enums [mib type $oid]]
    if {[string length $enums]} {
        writeln "Shost: [mib name $oid] with enumeration ignored"
        continue
    }
    lappend vars $oid
}
if {$vars == ""} {
    ined acknowledge \
        "$varname is either not available or of wrong type."
    continue
}
set args $ip
lappend args $uptime
set i 0
foreach var $vars {
    set nid [CreateChart $id [expr 30+$i] [expr 30+$i]]
    ined -noupdate attribute $nid "SNMP:Config" [$s configure]
    lappend args $nid
    lappend args $var
    incr i
}
$s destroy
eval start_snmp_monitor $args
}
}

# Program Total Utilization.
proc ShowIfLoadProc {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
        return
    }
    set now [lindex [lindex $vbl 0] 2]
    set ifIn [lindex [lindex $vbl 1] 2]
    set ifOut [lindex [lindex $vbl 2] 2]
    set ifOS [lindex [lindex $vbl 3] 2]
    if {[catch {expr $ifIn - $cx(ifInOctets,$id)} deltaIn]} {
        set deltaIn [expr double($ifIn) - $cx(ifInOctets,$id)]
    }
    if {[catch {expr $ifOut - $cx(ifOutOctets,$id)} deltaOut]} {
        set deltaOut [expr double($ifOut) - $cx(ifOutOctets,$id)]
    }
    if {$cx(duplex,$id)} {
        set delta [expr $deltaIn > $deltaOut ? $deltaIn : $deltaOut]
    } else {
        set delta [expr $deltaIn + $deltaOut]
    }
}

```

```

set speed [ined -noupdate attribute $id SNMP-Monitor:IfSpeed]
if {$speed == ""} {
    set speed $cx(ifSpeed,$id)
}
if {$snow > $cx(sysUpTime,$id) && $speed > 0} {
    set secs [expr ($snow - $cx(sysUpTime,$id)) / 100.0]
    set val [expr ( 8.0 * $delta / $secs ) / $speed * 100]
} else {
    set val 0
}
ined values $id $val
if {$sifOS != "up" && $sifOS != "1"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $sifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id ifLoad \
        [format "$cx(ifDescr,$id) %.2f%%\n \n Percent Interface Total
Utilization " $val]
}
set txt "[ined name $id] $cx(ifDescr,$id) interface load"
MoJoCheckThreshold $id $txt $val "%"
set cx(sysUpTime,$id) $snow
set cx(ifInOctets,$id) $sifIn
set cx(ifOutOctets,$id) $sifOut
set cx(ifOperStatus,$id) $sifOS
}

proc ShowIfLoad { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoad $ids] {
        set i $cx(idx,$id)
        set vbl "sysUpTime.0 ifInOctets.$i ifOutOctets.$i ifOperStatus.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProc $id %E \"%V\""
    }
}

proc start_ifload_monitor { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$sargs != ""} {
        set id [lindex $sargs 0]
        set idx [lindex $sargs 1]
        if {$id != ""} {
            set sh [SnmpOpen $id $ip]
            if {[catch {$sh get [list sysUpTime.0 \
                ifInOctets.$idx ifOutOctets.$idx \
                ifSpeed.$idx ifDescr.$idx ifType.$idx \
                ifOperStatus.$idx]} value]} {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ined -noupdate attribute $id ifLoad [lindex $value 0]
ined -noupdate label $id ifLoad
set args [lrange $args 2 end]
continue
}
set cx(sysUpTime,$id) [lindex [lindex $value 0] 2]
set cx(ifInOctets,$id) [lindex [lindex $value 1] 2]
set cx(ifOutOctets,$id) [lindex [lindex $value 2] 2]
set cx(ifSpeed,$id) [lindex [lindex $value 3] 2]
set cx(ifDescr,$id) [lindex [lindex $value 4] 2]
set cx(ifType,$id) [lindex [lindex $value 5] 2]
set cx(ifOperStatus,$id) [lindex [lindex $value 6] 2]
set cx(snmp,$id) $sh
set cx(idx,$id) $idx
set cx(duplex,$id) 1
lappend ids $id
ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
ined -noupdate label $id ifLoad
}
set args [lrange $args 2 end]
append re_args "$id $idx"
}
if {$sids != ""} {
set jid [job create -command [list ShowIfLoad $sids] \
-interval [expr $interval * 1000]]
save "restart start_ifload_monitor $interval $jid $ip $re_args"
}
}

proc "Interface Total Utilization" { list } {
ForeachIpNode id ip host $list {
set sh [SnmpOpen $id $ip]
set iflist ""
if {[catch {
$sh walk x "ifIndex ifOperStatus ifSpeed ifInOctets ifOutOctets" {
set idx [lindex [lindex $x 0] 2]
set status [lindex [lindex $x 1] 2]
set speed [lindex [lindex $x 2] 2]
if {$speed == 0} {
writeln "Interface $idx on $host \[$ip\] ignored (ifSpeed = $speed)"
continue
}
}
}
}
lappend iflist $idx
}
}}} {
$sh destroy
continue
}
}

```

```

set args $ip
set i 0
foreach if $iflist {
    lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
    lappend args $if
    incr i
}
eval start_ifload_monitor $args
$ssh destroy
}
}

```

Program Input Utilization.

```

proc ShowIfLoadProcInput {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
        return
    }
    set now [lindex [lindex $vbl 0] 2]
    set ifIn [lindex [lindex $vbl 1] 2]
    set ifOut [lindex [lindex $vbl 2] 2]
    set ifOS [lindex [lindex $vbl 3] 2]
    if {[catch {expr $ifIn - $cx(ifInOctets,$id)} deltaIn]} {
        set deltaIn [expr double($ifIn) - $cx(ifInOctets,$id)]
    }
    if {[catch {expr $ifOut - $cx(ifOutOctets,$id)} deltaOut]} {
        set deltaOut [expr double($ifOut) - $cx(ifOutOctets,$id)]
    }
    if {$cx(duplex,$id)} {
        set delta [expr $deltaIn > $deltaOut ? $deltaIn : $deltaOut]
    } else {
        set delta [expr $deltaIn + $deltaOut]
    }
    set speed [ined -noupdate attribute $id SNMP-Monitor:IfSpeed]
    if {$speed == ""} {
        set speed $cx(ifSpeed,$id)
    }
    if {$now > $cx(sysUpTime,$id) && $speed > 0} {
        set secs [expr ($now - $cx(sysUpTime,$id)) / 100.0]
        set val [expr ( 8.0 * $deltaIn / $secs ) / $speed * 100]
    } else {
        set val 0
    }
    ined values $id $val
    if {$ifOS != "up" && $ifOS != "1"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
        ined -noupdate color $id red
    } else {
        ined -noupdate color $id blue
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f %%\n \n Percent Interface Input Utilization " $val]
    }
    set txt "[ined name $id] $cx(ifDescr,$id) interface load"
    MoJoCheckThreshold $id $txt $val "%%"
        set cx(sysUpTime,$id)    $now
        set cx(ifInOctets,$id)    $ifIn
        set cx(ifOutOctets,$id)   $ifOut
        set cx(ifOperStatus,$id)  $ifOS
    }

proc ShowIfLoadInput { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadInput $ids] {
        set i $cx(idx,$id)
        set vbl "sysUpTime.0 ifInOctets.$i ifOutOctets.$i ifOperStatus.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcInput $id %E \"%V\""
    }
}

proc start_ifload_monitor_Input { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while { $sargs != "" } {
        set id [lindex $sargs 0]
        set idx [lindex $sargs 1]
        if { $id != "" } {
            set sh [$SnmpOpen $id $ip]
            if {[catch {$sh get [list sysUpTime.0 \
                ifInOctets.$idx ifOutOctets.$idx \
                ifSpeed.$idx ifDescr.$idx ifType.$idx \
                ifOperStatus.$idx]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $sargs 2 end]
                continue
            }
            set cx(sysUpTime,$id)    [lindex [lindex $value 0] 2]
            set cx(ifInOctets,$id)    [lindex [lindex $value 1] 2]
            set cx(ifOutOctets,$id)   [lindex [lindex $value 2] 2]
            set cx(ifSpeed,$id)       [lindex [lindex $value 3] 2]
            set cx(ifDescr,$id)       [lindex [lindex $value 4] 2]
            set cx(ifType,$id)        [lindex [lindex $value 5] 2]
            set cx(ifOperStatus,$id)  [lindex [lindex $value 6] 2]
            set cx(snmp,$id)          $sh
            set cx(idx,$id)           $idx
            set cx(duplex,$id)        1
        }
    }
    lappend ids $id
}

```

lappend ids \$id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
    set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
    ined -noupdate label $id ifLoad
}
set args [lrange $args 2 end]
append re_args "\$id $idx"
}
if {$sids != ""} {
    set jid [job create -command [list ShowIfLoadInput $sids] \
        -interval [expr $interval * 1000]]
    save "restart start_ifload_monitor_Input $interval $jid $ip $re_args"
}
}

proc "Interface Input Utilization" {list} {
    ForeachIpNode id ip host $list {
        set sh [SnmppOpen $id $ip]
        set iflist ""
        if {[catch {
            $sh walk x "ifIndex ifOperStatus ifSpeed ifInOctets ifOutOctets" {
                set idx [lindex [lindex $x 0] 2]
                set status [lindex [lindex $x 1] 2]
                set speed [lindex [lindex $x 2] 2]
                if {$speed == 0} {
                    writeln "Interface $idx on $host \[$ip\] ignored (ifSpeed = $speed)"
                    continue
                }
                lappend iflist $idx
            }
        ]}] {
            $sh destroy
            continue
        }
        set args $ip
        set i 0
        foreach if $iflist {
            lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
            lappend args $if
            incr i
        }
        eval start_ifload_monitor_Input $args
        $sh destroy
    }
}
}

```

```

# Program Output Utilization
proc ShowIfLoadProcOutput {id status vbl} {
    global cx
    if {$status != "noError"} {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
return
}
set now      [lindex [lindex $vbl 0] 2]
set ifIn     [lindex [lindex $vbl 1] 2]
set ifOut    [lindex [lindex $vbl 2] 2]
set ifOS     [lindex [lindex $vbl 3] 2]
if {[catch {expr $ifIn - $cx(ifInOctets,$id)} deltaIn]} {
    set deltaIn [expr double($ifIn) - $cx(ifInOctets,$id)]
}
if {[catch {expr $ifOut - $cx(ifOutOctets,$id)} deltaOut]} {
    set deltaOut [expr double($ifOut) - $cx(ifOutOctets,$id)]
}
if {$cx(duplex,$id)} {
    set delta [expr $deltaIn > $deltaOut ? $deltaIn : $deltaOut]
} else {
    set delta [expr $deltaIn + $deltaOut]
}
set speed [ined -noupdate attribute $id SNMP-Monitor:IfSpeed]
if {$speed == ""} {
    set speed $cx(ifSpeed,$id)
}
if {$snow > $cx(sysUpTime,$id) && $speed > 0} {
    set secs [expr ($snow - $cx(sysUpTime,$id)) / 100.0]
    set val [expr ( 8.0 * $deltaOut / $secs ) / $speed * 100]
} else {
    set val 0
}
ined values $id $val
if {$ifOS != "up" && $ifOS != "1"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f%%\n\n Percent Interface Output Utilization " $val]
}
set txt      "[ined name $id] $cx(ifDescr,$id) interface load"
MoJoCheckThreshold $id $txt $val "%%"
set cx(sysUpTime,$id) $snow
set cx(ifInOctets,$id) $ifIn
set cx(ifOutOctets,$id) $ifOut
set cx(ifOperStatus,$id) $ifOS
}

proc ShowIfLoadOutput { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadOutput $ids] {
        set i $cx(idx,$id)
        set vbl "sysUpTime.0 ifInOctets.$i ifOutOctets.$i ifOperStatus.$i"

```

```

    $cx(snmplib,$id) get $vbl "ShowIfLoadProcOutput $id %E \"%V\"
}
}

proc start_ifload_monitor_Output { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$args != ""} {
        set id [lindex $args 0]
        set idx [lindex $args 1]
        if {$id != ""} {
            set sh [SnmplibOpen $id $ip]
            if {[catch {$sh get [list sysUpTime.0 \
                ifInOctets.$idx ifOutOctets.$idx \
                ifSpeed.$idx ifDescr.$idx ifType.$idx \
                ifOperStatus.$idx]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $args 2 end]
                continue
            }
            set cx(sysUpTime,$id) [lindex [lindex $value 0] 2]
            set cx(ifInOctets,$id) [lindex [lindex $value 1] 2]
            set cx(ifOutOctets,$id) [lindex [lindex $value 2] 2]
            set cx(ifSpeed,$id) [lindex [lindex $value 3] 2]
            set cx(ifDescr,$id) [lindex [lindex $value 4] 2]
            set cx(ifType,$id) [lindex [lindex $value 5] 2]
            set cx(ifOperStatus,$id) [lindex [lindex $value 6] 2]
            set cx(snmplib,$id) $sh
            set cx(idx,$id) $idx
            set cx(duplex,$id) 1
            lappend ids $id
            ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
            set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
            ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
            ined -noupdate label $id ifLoad
        }
        set args [lrange $args 2 end]
        append re_args " \$$id $idx"
    }
    if {$ids != ""} {
        set jid [job create -command [list ShowIfLoadOutput $ids] \
            -interval [expr $interval * 1000]]
        save "restart start_ifload_monitor_Output $interval $jid $ip $re_args"
    }
}
}

```

```
proc "Interface Output Utilization" {list} {
```

```
    ForeachIpNode id ip host $list {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set sh [SnmOpen Sid $ip]
set iflist ""
if {[catch {
    $sh walk x "ifIndex ifOperStatus ifSpeed ifInOctets ifOutOctets" {
        set idx      [lindex [lindex $x 0] 2]
        set status   [lindex [lindex $x 1] 2]
        set speed    [lindex [lindex $x 2] 2]
        if {$speed == 0} {
            writeln "Interface $idx on $host \[$ip\] ignored (ifSpeed = $speed)"
            continue
        }
        lappend iflist $idx
    }
}]} {
    $sh destroy
    continue
}
set args $ip
set i 0
foreach if $iflist {
    lappend args [CreateChart Sid [expr 30+$i] [expr 30+$i]]
    lappend args $if
    incr i
}
eval start_ifload_monitor_Output $args
$sh destroy
}
}

# Program Input Errors
proc ShowIfLoadProcInErr {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -nouupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
        return
    }

    set ifOS      [lindex [lindex $vbl 0] 2]
    set ifInU     [lindex [lindex $vbl 1] 2]
    set ifInNU    [lindex [lindex $vbl 2] 2]
    set ifOutU    [lindex [lindex $vbl 3] 2]
    set ifOutNU   [lindex [lindex $vbl 4] 2]
    set ifInErr   [lindex [lindex $vbl 5] 2]
    set ifOutErr  [lindex [lindex $vbl 6] 2]
    if {$ifInErr == 0} {
        set val 0
    } else {
        set val [expr {100*double($ifInErr)/($ifInU+$ifInNU)}]
    }
    ined values $id $val
    if {$ifOS != "up" && $ifOS != "1"} {

```

```

ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) SifOS"
ined -noupdate color $id red
} else {
  ined -noupdate color $id blue
  ined -noupdate attribute $id ifLoad \
  [format "$cx(ifDescr,$id) %.2f %%%\n \n Percent Interface Input Errors " $val]
}
}

proc ShowIfLoadInErr { ids } {
  global cx
  foreach id [MoJoCheckIds ShowIfLoadInErr $ids] {
    set i $cx(idx,$id)
    set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInErrors.$i ifOutErrors.$i"
    $cx(snmp,$id) get $vbl "ShowIfLoadProcInErr $id %E \"%V\""
  }
}

proc start_ifload_monitor_InErr { ip args } {
  global cx interval
  set ids ""
  set re_args ""
  while { $args != "" } {
    set id [lindex $args 0]
    set idx [lindex $args 1]
    if { $id != "" } {
      set sh [SnmpOpen $id $ip]
      if {[catch {$sh get [list ifDescr.$idx ifType.$idx ]} value]} {
        ined -noupdate attribute $id ifLoad [lindex $value 0]
        ined -noupdate label $id ifLoad
        set args [lrange $args 2 end]
        continue
      }
      set cx(ifDescr,$id) [lindex [lindex $value 0] 2]
      set cx(ifType,$id) [lindex [lindex $value 1] 2]
      set cx(snmp,$id) $sh
      set cx(idx,$id) $idx
      lappend ids $id
      ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
      set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
      ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
      ined -noupdate label $id ifLoad
    }
    set args [lrange $args 2 end]
    append re_args "\$id $idx"
  }
  if { $ids != "" } {
    set jid [job create -command [list ShowIfLoadInErr $ids] \
-interval [expr $interval * 1000]]
  }
}

```

```

save "restart start_ifload_monitor_InErr $interval $jid $ip $re_args"
}
}

proc "Interface Input Errors" {list} {
    ForeachIpNode id ip host $list {
        set sh [SnmpOpen Sid $ip]
        set iflist ""
        if {[catch {
            $ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInErrors ifOutErrors " {
                set idx          [lindex [lindex $x 0] 2]
                set status       [lindex [lindex $x 1] 2]
                lappend iflist $idx
            }
        ]}] {
            $ssh destroy
            continue
        }
        set args $ip
        set i 0
        foreach if $iflist {
            lappend args [CreateChart Sid [expr 30+$i] [expr 30+$i]]
            lappend args $if
            incr i
        }
        eval start_ifload_monitor_InErr $args
        $ssh destroy
    }
}

# Program Output Errors
proc ShowIfLoadProcOutErr {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -noupdate attribute Sid ifLoad "$cx(ifDescr,$id) $status"
        return
    }

    set ifOS          [lindex [lindex $vbl 0] 2]
    set ifInU         [lindex [lindex $vbl 1] 2]
    set ifInNU        [lindex [lindex $vbl 2] 2]
    set ifOutU        [lindex [lindex $vbl 3] 2]
    set ifOutNU       [lindex [lindex $vbl 4] 2]
    set ifInErr       [lindex [lindex $vbl 5] 2]
    set ifOutErr      [lindex [lindex $vbl 6] 2]
    if {$ifOutErr == 0} {
        set val 0
    } else {
        set val [expr {100*double($ifOutErr)/($ifOutU+$ifOutNU)} ]
    }
}

```

```

ined values $id $val
if {$ifOS != "up" && $ifOS != "1"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f%%\n\n Percent Interface Output Errors " $val]
}
}

proc ShowIfLoadOutErr { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadOutErr $ids] {
        set i $cx(id,$id)
        set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInErrors.$i ifOutErrors.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcOutErr $id %E \"%V\""
    }
}

proc start_ifload_monitor_OutErr { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$sargs != ""} {
        set id [lindex $sargs 0]
        set idx [lindex $sargs 1]
        if {$id != ""} {
            set sh [SnmpOpen $id $ip]
            if {[catch {$sh get [list ifDescr.$idx ifType.$idx ]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $sargs 2 end]
                continue
            }
            set cx(ifDescr,$id) [lindex [lindex $value 0] 2]
            set cx(ifType,$id) [lindex [lindex $value 1] 2]
            set cx(snmp,$id) $sh
            set cx(idx,$id) $idx
            lappend ids $id
            ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
            set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
            ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
            ined -noupdate label $id ifLoad
        }
        set args [lrange $sargs 2 end]
        append re_args " \$$id $idx"
    }
    if {$ids != ""} {

```

```

set jid [job create -command [list ShowIfLoadOutErr $ids] \
    -interval [expr $interval * 1000]]
save "restart start_ifload_monitor_OutErr $interval $jid $ip $re_args"
}
}

proc "Interface Output Errors" {list} {
    ForeachIpNode id ip host $list {
        set sh [SnmpOpen $id $ip]
        set iflist ""
        if {[catch {
            $ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInErrors ifOutErrors " {
                set idx          [lindex [lindex $x 0] 2]
                set status       [lindex [lindex $x 1] 2]
                lappend iflist $idx
            }
        ]}] {
            $ssh destroy
            continue
        }
        set args $ip
        set i 0
        foreach if $iflist {
            lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
            lappend args $if
            incr i
        }
        eval start_ifload_monitor_OutErr $args
        $ssh destroy
    }
}
}

```

```

# Program Total Errors
proc ShowIfLoadProcErr {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
        return
    }
    set ifOS          [lindex [lindex $vbl 0] 2]
    set ifInU         [lindex [lindex $vbl 1] 2]
    set ifInNU        [lindex [lindex $vbl 2] 2]
    set ifOutU        [lindex [lindex $vbl 3] 2]
    set ifOutNU       [lindex [lindex $vbl 4] 2]
    set ifInErr       [lindex [lindex $vbl 5] 2]
    set ifOutErr      [lindex [lindex $vbl 6] 2]
    if {$ifInErr == 0 && $ifOutErr == 0} {
        set val 0
    } else {

```

```

    set val [expr
{100*double($ifInErr+$ifOutErr)/($ifInU+$ifInNU+$ifOutU+$ifOutNU)}]
    }
    ined values $id $val
    if {$ifOS != "up" && $ifOS != "1"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
        ined -noupdate color $id red
    } else {
        ined -noupdate color $id blue
        ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f%%\n \n Percent Interface Total Errors " $val]
    }
}
}

proc ShowIfLoadErr { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadErr $ids] {
        set i $cx(idx,$id)
        set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInErrors.$i ifOutErrors.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcErr $id %E \"%V\""
    }
}

proc start_ifload_monitor_Err { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$sargs != ""} {
        set id [lindex $sargs 0]
        set idx [lindex $sargs 1]
        if {$id != ""} {
            set sh [SnmpOpen $id $ip]
            if {[catch {$sh get [list ifDescr.$idx ifType.$idx ]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $sargs 2 end]
                continue
            }
            set cx(ifDescr,$id) [lindex [lindex $value 0] 2]
            set cx(ifType,$id) [lindex [lindex $value 1] 2]
            set cx(snmp,$id) $sh
            set cx(idx,$id) $idx
            lappend ids $id
            ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
            set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
            ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
            ined -noupdate label $id ifLoad
        }
        set args [lrange $sargs 2 end]
    }
}

```

```

    append re_args "\$Sid $idx"
}
if {$Sids != ""} {
    set jid [job create -command [list ShowIfLoadErr Sids] \
        -interval [expr $interval * 1000]]
    save "restart start_ifload_monitor_Err $interval $jid $ip $re_args"
}
}

proc "Interface Total Errors" {list} {
    ForeachIpNode id ip host $list {
        set sh [SnmppOpen $id $ip]
        set iflist ""
        if {[catch {
            Ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInErrors ifOutErrors" {
                set idx [lindex [lindex $x 0] 2]
                set status [lindex [lindex $x 1] 2]
                lappend iflist $idx
            }
        ]}] {
            Ssh destroy
            continue
        }
        set args $ip
        set i 0
        foreach if $iflist {
            lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
            lappend args $if
            incr i
        }
        eval start_ifload_monitor_Err $args
        Ssh destroy
    }
}
}

```

Program Input Discards

```

proc ShowIfLoadProcInDis {id status vbl} {
    global cx
    if {$status != "noError"} {
        ined -nouupdate attribute $id ifLoad "$cx(ifDescr,$id) $status"
        return
    }

    set ifOS [lindex [lindex $vbl 0] 2]
    set ifInU [lindex [lindex $vbl 1] 2]
    set ifInNU [lindex [lindex $vbl 2] 2]
    set ifOutU [lindex [lindex $vbl 3] 2]
    set ifOutNU [lindex [lindex $vbl 4] 2]
    set ifInDis [lindex [lindex $vbl 5] 2]
    set ifOutDis [lindex [lindex $vbl 6] 2]
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if {$ifInDis == 0} {
    set val 0
} else {
    set val [expr {100*double($ifInDis)/($ifInU+$ifInNU)}]
}
ined values $id $val
if {$ifOS != "up" && $ifOS != "1"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id iiLoad \
[format "$cx(ifDescr,$id) %.2f %%\n \n Percent Interface Input Discards " $val]
}
}

proc ShowIfLoadInDis { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadInDis $ids] {
        set i $cx(idx,$id)
        set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInDiscards.$i ifOutDiscards.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcInDis $id.%E \"%VX\""
    }
}

proc start_ifload_monitor_InDis { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$args != ""} {
        set id [lindex $args 0]
        set idx [lindex $args 1]
        if {$id != ""} {
            set sh [SnmOpen $id $ip]
            if {[catch {$sh get [list ifDescr.$idx ifType.$idx ]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $args 2 end]
                continue
            }
            set cx(ifDescr,$id) [lindex [lindex $value 0] 2]
            set cx(ifType,$id) [lindex [lindex $value 1] 2]
            set cx(snmp,$id) $sh
            set cx(idx,$id) $idx
            lappend ids $id
            ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
            set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
            ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
            ined -noupdate label $id ifLoad
        }
    }
}

```

```

    }
    set args [lrange $args 2 end]
    append re_args "\$Sid $idx"
  }
  if {$Sids != ""} {
    set jid [job create -command [list ShowIfLoadInDis $Sids] \
      -interval [expr $interval * 1000]]
    save "restart start_ifload_monitor_InDis $interval $jid $ip $re_args"
  }
}

```

```

proc "Interface Input Discards" {list} {
  ForeachIpNode id ip host $list {
    set sh [SnmpOpen $id $ip]
    set iflist ""
    if {[catch {
      Ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInDiscards ifOutDiscards " {
        set idx [lindex [lindex $x 0] 2]
        set status [lindex [lindex $x 1] 2]
        lappend iflist $idx
      }
    ]}] {
      Ssh destroy
      continue
    }
    set args $ip
    set i 0
    foreach if $iflist {
      lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
      lappend args $if
      incr i
    }
    eval start_ifload_monitor_InDis $args
    Ssh destroy
  }
}

```

Program Output Discards

```

proc ShowIfLoadProcOutDis {id status vbl} {
  global cx
  if {$Sstatus != "noError"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $Sstatus"
    return
  }

  set ifOS [lindex [lindex $vbl 0] 2]
  set ifInU [lindex [lindex $vbl 1] 2]
  set ifInNU [lindex [lindex $vbl 2] 2]
  set ifOutU [lindex [lindex $vbl 3] 2]
  set ifOutNU [lindex [lindex $vbl 4] 2]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set ifInDis          [index [index $vbl 5] 2]
set ifOutDis        [index [index $vbl 6] 2]
if { $ifOutDis == 0 } {
    set val 0
} else {
    set val [expr {100*double($ifOutDis)/($ifOutU+$ifOutNU)}]
}
ined values $id $val
if { $ifOS != "up" && $ifOS != "1" } {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f%%\n\n Percent Interface Output Discards " $val]
}
}

proc ShowIfLoadOutDis { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadOutDis $ids] {
        set i $cx(idx,$id)
        set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInDiscards.$i ifOutDiscards.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcOutDis $id %E \"%V\""
    }
}

proc start_ifload_monitor_OutDis { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while { $args != "" } {
        set id [index $args 0]
        set idx [index $args 1]
        if { $id != "" } {
            set sh [SnmOpen $id $ip]
            if {[catch { $sh get [list ifDescr.$idx ifType.$idx] } value]} {
                ined -noupdate attribute $id ifLoad [index $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $args 2 end]
                continue
            }
            set cx(ifDescr,$id) [index [index $value 0] 2]
            set cx(ifType,$id) [index [index $value 1] 2]
            set cx(snmp,$id) $sh
            set cx(idx,$id) $idx
            lappend ids $id
            ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
            set cx(ifDescr,$id) [index $cx(ifDescr,$id) 0]

```

```

        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
        ined -noupdate label $id ifLoad
    }
    set args [lrange $args 2 end]
    append re_args " \$$id $idx"
}
if {$Sids != ""} {
    set jid [job create -command [list ShowIfLoadOutDis $Sids] \
        -interval [expr $Sinterval * 1000]]
    save "restart start_ifload_monitor_OutDis $Sinterval $jid $ip $re_args"
}
}

```

```

proc "Interface Output Discards" {list} {
    ForeachIpNode id ip host $list {
        set sh [SnmpOpen $id $ip]
        set iflist ""
        if {[catch {
            $ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInDiscards ifOutDiscards " {
                set idx [lindex [lindex $x 0] 2]
                set status [lindex [lindex $x 1] 2]
                lappend iflist $idx
            }
        ]}] {
            $ssh destroy
            continue
        }
        set args $ip
        set i 0
        foreach if $iflist {
            lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
            lappend args $if
            incr i
        }
        eval start_ifload_monitor_OutDis $args
        $ssh destroy
    }
}

```

Program Total Discards

```

proc ShowIfLoadProcDis {id status vbl} {
    global cx
    if {$Sstatus != "noError"} {
        ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $Sstatus"
        return
    }
    set ifOS [lindex [lindex $vbl 0] 2]
    set ifInU [lindex [lindex $vbl 1] 2]
    set ifInNU [lindex [lindex $vbl 2] 2]

```

```

set ifOutU          [lindex [lindex $vbl 3] 2]
set ifOutNU        [lindex [lindex $vbl 4] 2]
set ifInDis        [lindex [lindex $vbl 5] 2]
set ifOutDis       [lindex [lindex $vbl 6] 2]
if {$ifInDis == 0 && $ifOutDis == 0} {
    set val 0
} else {
set val [expr
{100*double($ifInDis+$ifOutDis)/($ifInU+$ifInNU+$ifOutU+$ifOutNU)}]
}
ined values $id $val
if {$ifOS != "up" && $ifOS != "1"} {
    ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) $ifOS"
    ined -noupdate color $id red
} else {
    ined -noupdate color $id blue
    ined -noupdate attribute $id ifLoad \
[format "$cx(ifDescr,$id) %.2f%%\n\n Percent Interface Total Discards " $val]
}
}

proc ShowIfLoadDis { ids } {
    global cx
    foreach id [MoJoCheckIds ShowIfLoadDis $ids] {
        set i $cx(idx,$id)
        set vbl "ifOperStatus.$i ifInUcastPkts.$i ifInNUcastPkts.$i ifOutUcastPkts.$i
ifOutNUcastPkts.$i ifInDiscards.$i ifOutDiscards.$i"
        $cx(snmp,$id) get $vbl "ShowIfLoadProcDis $id %E \"%V\""
    }
}

proc start_ifload_monitor_Dis { ip args } {
    global cx interval
    set ids ""
    set re_args ""
    while {$sargs != ""} {
        set id [lindex $sargs 0]
        set idx [lindex $sargs 1]
        if {$id != ""} {
            set sh [SnmpOpen $id $ip]
            if {[catch {$sh get [list ifDescr.$idx ifType.$idx ]} value]} {
                ined -noupdate attribute $id ifLoad [lindex $value 0]
                ined -noupdate label $id ifLoad
                set args [lrange $sargs 2 end]
                continue
            }
            set cx(ifDescr,$id) [lindex [lindex $value 0] 2]
            set cx(ifType,$id) [lindex [lindex $value 1] 2]
            set cx(snmp,$id) $sh
            set cx(idx,$id) $idx
        }
    }
}

```

```

lappend ids          $id
ined -noupdate attribute $id ifDescr $cx(ifDescr,$id)
set cx(ifDescr,$id) [lindex $cx(ifDescr,$id) 0]
ined -noupdate attribute $id ifLoad "$cx(ifDescr,$id) waiting"
ined -noupdate label $id ifLoad
}
set args [lrange $args 2 end]
append re_args "$id $idx"
}
if {$sids != ""} {
set jid [job create -command [list ShowIfLoadDis $sids] \
-interval [expr $interval * 1000]]
save "restart start_ifload_monitor_Dis $interval $jid $ip $re_args"
}
}
}

```

```

proc "Interface Total Discards" {list} {
ForeachIpNode id ip host $list {
set sh [SnmppOpen $id $ip]
set iflist ""
if {[catch {
$ssh walk x "ifIndex ifOperStatus ifInUcastPkts ifInNUcastPkts
ifOutUcastPkts ifOutNUcastPkts ifInDiscards ifOutDiscards " {
set idx [lindex [lindex $x 0] 2]
set status [lindex [lindex $x 1] 2]
lappend iflist $idx
}
}]} {
$ssh destroy
continue
}
set args $ip
set i 0
foreach if $iflist {
lappend args [CreateChart $id [expr 30+$i] [expr 30+$i]]
lappend args $if
incr i
}
eval start_ifload_monitor_Dis $args
$ssh destroy
}
}
}

```

```

# Set the Modify Monitor parameters jobs.
proc "Modify Monitor Job" { list } {
MoJoModify }
# Set the default parameters for monitoring jobs.
proc "Set Monitor Parameter" {list} {
MoJoParameter
}

```

Exit the menus created by this interpreter.

```
proc "Exit SNMP-Monitor" {list} {
  global menus
  if {[job find] != ""} {
    set res [ined confirm "Kill running monitoring jobs?" \
      [list "kill & exit" cancel] ]
    if {$res == "cancel"} return
  }
  DeleteClones
  foreach id $menus { ined delete $id }
  exit
}
```

```
set menus [ ined create MENU "SNMP-Monitor" \
  "Monitoring:Display Performance Graphs:Interface Input Utilization" \
  "Monitoring:Display Performance Graphs:Interface Output Utilization" \
  "Monitoring:Display Performance Graphs:Interface Total Utilization" \
  "Monitoring:Display Performance Graphs:Interface Input Errors" \
  "Monitoring:Display Performance Graphs:Interface Output Errors" \
  "Monitoring:Display Performance Graphs:Interface Total Errors" \
  "Monitoring:Display Performance Graphs:Interface Input Discards" \
  "Monitoring:Display Performance Graphs:Interface Output Discards" \
  "Monitoring:Display Performance Graphs:Interface Total Discards" ""\
  "Monitoring:Notification" \
  "Monitoring:Event Log"\
  "Parameter:Set Monitor Parameter" \
  "Parameter:Modify Monitor Job" ""\
  "Exit SNMP-Monitor" ]
```

vwait forever

#####

ประวัติผู้เขียน

นายปิยพงศ์ ผดุงพจน์ เกิดเมื่อวันที่ 23 ตุลาคม 2511 ที่จังหวัด พระนครศรีอยุธยา สำเร็จการศึกษาอุตสาหกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) จากมหาวิทยาลัยสยาม ปีการศึกษา 2536 และบริหารธุรกิจบัณฑิต (วิทยาการจัดการ) จากมหาวิทยาลัยสุโขทัยธรรมมาธิราช ปีการศึกษา 2540

ปี พ.ศ. 2536 เข้าทำงานในตำแหน่ง วิศวกรระบบ บริษัท เบทาโกร ปี พ.ศ. 2540 ทำงานในตำแหน่ง วิศวกร โครงการ บริษัท TIS Consultants และปัจจุบันประกอบอาชีพส่วนตัว

ผลงานวิจัยที่ได้รับการตีพิมพ์

- [1]. ปิยพงศ์ ผดุงพจน์ และ สมผล โกศลวิตร. “การพัฒนาซอฟต์แวร์เฝ้าดูระบบเครือข่ายเพื่อการจัดการสำหรับเครือข่าย แลน/แวน.” วิศวกรรมสาร มก., ฉบับที่ 42, เดือนธันวาคม 2543 - มีนาคม 2544.

