ระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บแบบฝังตัว

นายชนินทร์ มหารักษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2547
ISBN 974-17-5989-4

EMBEDDED WEB-BASED NETWORK MANAGER SYSTEM

Mr. Chanin Maharak

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2004

ISBN 974-17-5989-4

หัวข้อวิทยานิพนธ์	ระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บแบบฝังตัว
โดย	นายชนินทร์ มหารักษ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ บุญชัย โสวรรณวณิชกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ บุญชัย โสวรรณวณิชกุล)

..... กรรมการ
(อาจารย์ ดร.อาทิตย์ ทองทักษ์)

..... กรรมการ
(อาจารย์ ธนา หงษ์สุวรรณ)

ชนินทร์ มหารักษ์ : ระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บแบบฝังตัว. (EMBEDDED WEB-BASED NETWORK MANAGER SYSTEM)
อ. ที่ปรึกษา : ผศ.บุญชัย โสวรรณวณิชกุล, 97หน้า. ISBN 974-17-5989-4.

การจัดการเครือข่ายคอมพิวเตอร์ด้วยเอสเอ็นเอ็มพีได้รับความนิยมอย่างสูง เพราะอุปกรณ์เครือข่ายหลายชนิดสนับสนุนการจัดการผ่านข้อตกลงเอสเอ็นเอ็มพี ในปัจจุบันได้มีการนำเสนอการจัดการเครือข่ายผ่านเว็บเพื่อการใช้งานที่สะดวกยิ่งขึ้นของผู้ดูแลระบบด้วยการพัฒนาเอสเอ็นเอ็มพีเอเจนต์ขึ้นมาใหม่ให้มีความสามารถติดต่อผ่านเอชทีทีพีได้ ส่งผลให้การดูแลจัดการเครือข่ายคอมพิวเตอร์สามารถทำได้จากทุกที่ที่มีเครือข่ายอินเทอร์เน็ต นอกจากนี้ยังสามารถปิดกั้นข้อมูลเอสเอ็นเอ็มพีจากภายนอกเครือข่ายด้วยไฟร์วอลล์โดยไม่กระทบกับการจัดการเครือข่ายคอมพิวเตอร์จากระยะไกล

วิทยานิพนธ์นี้เป็นการพัฒนาระบบจัดการเครือข่ายคอมพิวเตอร์ตามข้อตกลงเอสเอ็นเอ็มพีผ่านเว็บซึ่งทำงานอยู่บนระบบฝังตัวเพื่อให้ผู้ดูแลระบบสามารถจัดการเครือข่ายคอมพิวเตอร์ได้จากเครือข่ายอินเทอร์เน็ต ทั้งนี้ระบบจัดการเครือข่ายที่ได้พัฒนามีระบบรักษาความปลอดภัยโดยอาศัยการพิสูจน์ตนของเอชทีทีพีไดเจสและการเข้ารหัสข้อมูลด้วยเออีเอส ระบบดังกล่าวช่วยเพิ่มความสามารถในการจัดการเครือข่ายกับระบบฝังตัวซึ่งผู้ดูแลระบบสามารถนำระบบฝังตัวเข้าไปใช้งานกับเครือข่ายคอมพิวเตอร์เดิมโดยไม่ต้องเปลี่ยนแปลงอุปกรณ์ใดๆ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2547

4470263721 : MAJOR COMPUTER ENGINEERING

KEY WORD: NETWORK MANAGEMENT / NETWORK MANAGER / SNMP / WEB-BASED NETWORK MANAGEMENT / EMBEDDED SYSTEM / HTTP DIGEST AUTHENTICATION / AES

CHANIN MAHARAK : EMBEDDED WEB-BASED NETWORK MANAGER SYSTEM. THESIS ADVISOR : BOONCHAI SOWANWANICHAKUL, 97 pp. ISBN 974-17-5989-4.

SNMP network management is very popular because of plenty network devices already support SNMP. At the moment, web-based network management, which SNMP agents are able to communicate with HTTP, was proposed for network administrator convenience. As a result, capable to manage network management from everywhere through the Internet. Besides that, SNMP from the outside can be restricted by firewall with no affect on remote network management.

Allowing administrator to manage network via the Internet, this thesis is a development of web-based network manager based on SNMP by using embedded system. A developed system has a security system, which using HTTP digest authentication and AES. This system brings network management to the embedded system, which administrator is able to employ to the existing network with changeless.

Department Computer Engineering Student's signature .....

Field of study Computer Engineering Advisor's signature .....

Academic year 2004

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีจากความกรุณา และความช่วยเหลือของคณาจารย์ทุกๆ ท่าน โดยเฉพาะอย่างยิ่ง ผศ.บุญชัย โสวรรณวณิชกุล ซึ่งนอกจากได้เมตตามาเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์แล้ว ท่านยังได้เสียสละเวลาเพื่อให้คำแนะนำ พร้อมทั้งดูแลเอาใจใส่ ตักเตือน ให้แง่คิด ทั้งในด้านของการทำวิจัยและการใช้ชีวิต

ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา รวมทั้งครอบครัวของข้าพเจ้าที่สนับสนุนการศึกษาของข้าพเจ้า ชี้นำให้เห็นถึงความสำคัญของการศึกษา ตลอดจนคอยเป็นกำลังใจให้ข้าพเจ้าอย่างสม่ำเสมอ

ข้าพเจ้าขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ภายในห้องปฏิบัติการ Digital System Engineering Laboratory เพื่อนๆ นิสิตปริญญาโททุกท่าน และนายเกียรติ ภิรมย์โสภา ที่คอยช่วยเหลือให้คำแนะนำที่เป็นประโยชน์ในการทำวิทยานิพนธ์ คอยช่วยเหลือในเรื่องต่างๆ รวมทั้งช่วยแก้ปัญหาระหว่างการทำวิจัย

ข้าพเจ้าขอขอบคุณ นางสาวหทัยรัตน์ อ้วนสุชาติ ซึ่งคอยให้กำลังใจ และแสดงความคิดเห็นในเรื่องต่างๆ ที่เป็นประโยชน์แก่ข้าพเจ้า

ข้าพเจ้าขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ รวมไปถึงเจ้าหน้าที่ภายในภาควิชาทุกๆ ท่าน ซึ่งเอื้อเฟื้ออุปกรณ์ สถานที่ และโอกาสในการศึกษาของข้าพเจ้า อีกทั้งยังเป็นภาระจัดหาสิ่งขาดตกบกพร่องทั้งหลายแก่ข้าพเจ้าด้วยไมตรีอันอบอุ่น

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญภาพ

ภาพประกอบ หน้า

ภาพประกอบ หน้า

สารบัญตาราง

ตาราง หน้า

บทที่ 1
บทนำ

การจัดการเครือข่ายคอมพิวเตอร์ (Computer Network Management) ได้ถือกำเนิดขึ้นจากความต้องการในการจัดการให้เครื่องคอมพิวเตอร์หลายเครื่องทำงานร่วมกันในระบบเครือข่ายอย่างราบรื่น อีกทั้งการพัฒนาเครือข่ายคอมพิวเตอร์ที่เริ่มมีแนวโน้มในการพัฒนาให้อุปกรณ์ต่างๆ ในอนาคตสามารถเชื่อมต่อกับระบบเครือข่ายคอมพิวเตอร์ ดังนั้นการจัดการเครือข่ายคอมพิวเตอร์จึงเป็นเรื่องจำเป็นในปัจจุบัน เพราะหากมีการจัดการที่ดี ย่อมส่งผลให้สามารถป้องกันปัญหาที่อาจเกิดขึ้นกับเครือข่ายคอมพิวเตอร์ได้ จากเหตุผลดังกล่าว ทำให้ผู้ดูแลระบบเครือข่ายต่างๆ เห็นความสำคัญและมีการใช้งานออกไปในวงกว้าง แต่เนื่องจากการจัดการเครือข่ายคอมพิวเตอร์ในปัจจุบันมีความยุ่งยากในการใช้งาน และสำหรับผู้ดูแลระบบเครือข่ายที่จำเป็นต้องออกนอกสถานที่บ่อยๆ มักพบปัญหาเกี่ยวกับไฟร์วอลล์ปิดกั้นการเข้าถึงเครือข่ายจากภายนอก จึงไม่สามารถจัดการเครือข่ายคอมพิวเตอร์ได้จากภายนอกระบบที่ดูแล ดังนั้นงานวิจัยนี้จึงได้นำเสนอระบบฝังตัวสำหรับการจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บซึ่งสามารถจัดการเครือข่ายได้จากภายนอกระบบที่ดูแล อีกทั้งยังช่วยเพิ่มความสามารถในการจัดการเครือข่ายจากระยะไกลให้เป็นไปได้อย่างสะดวก และปลอดภัย

1.1 ความเป็นมาและความสำคัญของปัญหา

การจัดการเครือข่ายคอมพิวเตอร์ได้ถูกนำมาจัดทำเป็นข้อตกลง (Protocol) ขึ้นหลายแบบด้วยกันอาทิ เอสเอ็นเอ็มพี (SNMP: Simple Network Management Protocol) ซีเอ็มไอพี (CMIP: Common Management Information Protocol) เป็นต้น โดยแต่ละข้อตกลงนั้นมีข้อจำกัดในการใช้งานที่แตกต่างกันออกไป ซึ่งการจัดการเครือข่ายคอมพิวเตอร์ตามข้อตกลงต่างๆ นั้น จำเป็นต้องอาศัยโปรแกรมเฉพาะผลิตภัณฑ์ซึ่งมีข้อเสียในหลายด้าน คือ

- มีความยุ่งยากในการใช้งาน เพราะผู้ใช้ต้องการสั่งงานกับอุปกรณ์ที่ต่างออกไป จำเป็นต้องเรียนรู้วิธีการใช้งานเฉพาะกับอุปกรณ์นั้นๆ เนื่องจากว่าถึงแม้ผลิตภัณฑ์ใดๆ ใช้ข้อตกลงในการจัดการเดียวกัน แต่โปรแกรมการใช้งานเป็นโปรแกรมเฉพาะของผลิตภัณฑ์นั้นๆ ซึ่งการใช้งานย่อมแตกต่างกัน
- โปรแกรมมักขึ้นกับแพลตฟอร์ม (Platform) หากผู้ใช้ต้องการนำโปรแกรมเดียวกันไปใช้กับแพลตฟอร์มที่ต่างออกไป จำเป็นต้องหาโปรแกรมที่ทำงานอยู่บนแพลตฟอร์มนั้นๆ จึงสามารถใช้งานได้

- การกำหนดค่า หรือการตรวจสอบส่วนเอเจนต์ (Agent) มีเงื่อนไขหลายประการ หากผู้ใช้ต้องการเข้าถึงส่วนเอเจนต์จากที่ใดๆ ผ่านเครือข่ายคอมพิวเตอร์ ผู้ใช้อาจได้รับความเสี่ยงที่เกิดจากการส่งผ่านคำสั่งประเภทการจัดการจากเครือข่ายภายนอก

เนื่องจากโปรแกรมเฉพาะด้านในการจัดการซึ่งขึ้นอยู่กับแพลตฟอร์มนั้นมีความยุ่งยากในการใช้งาน ดังนั้นจึงมีการนำเทคโนโลยีเว็บเข้ามาแก้ปัญหา ซึ่งก่อให้เกิดการจัดการเครือข่ายในรูปแบบใหม่ขึ้นเรียกว่า การจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บ (Web-based Network Management) วิธีการดังกล่าวนี้ถูกนำมาใช้เป็นระยะเวลาหนึ่งแล้ว ซึ่งสามารถแก้ปัญหาต่างๆ จากการจัดการแบบเดิมได้ เพราะความยุ่งยากในการใช้งานลดน้อยลงเนื่องจากผู้ใช้คุ้นเคยกับการใช้งานผ่านทางเว็บบราวเซอร์ (Web Browser) มากกว่าการใช้โปรแกรมเฉพาะด้าน การแก้ปัญหาในลักษณะนี้จึงเปรียบเสมือนการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphic User Interface: GUI) ขึ้นบนเว็บเพจ และเนื่องจากโปรแกรมถูกพัฒนาให้ใช้งานผ่านเว็บบราวเซอร์ทำให้การทำงานไม่ขึ้นอยู่กับแพลตฟอร์ม อีกทั้งสามารถเรียกใช้งานจากทุกที่ที่มีเครือข่ายอินเทอร์เน็ต (Internet)

การจัดการเครือข่ายผ่านเว็บได้ถูกออกแบบไว้หลายลักษณะ แต่การออกแบบได้มุ่งเน้นให้ระบบการจัดการทั้งระบบทำงานผ่านเว็บ นั่นคือการติดต่อระหว่างเอเจนต์ไปยังระบบจัดการต้องติดต่อสื่อสารข้อมูลผ่านเว็บ ดังนั้นส่วนเอเจนต์ต้องเพิ่มความสามารถในการทำงานผ่านเว็บ ซึ่งจึงจำเป็นต้องเปลี่ยนแปลงส่วนเอเจนต์ การออกแบบที่มุ่งเน้นให้การทำงานของทั้งส่วนการจัดการ และส่วนเอเจนต์ทำงานผ่านเว็บนั้น ส่งผลให้ส่วนจัดการเครือข่ายผ่านเว็บไม่สามารถทำงานร่วมกับเอเจนต์ในระบบเดิมได้โดยตรง จำเป็นต้องนำส่วนเอเจนต์ใหม่เพิ่มเข้ามาในระบบเครือข่ายอย่างน้อยหนึ่งเอเจนต์ เพื่อให้เป็นส่วนติดต่อไปยังเอเจนต์ในระบบเดิม

ในแง่ของระบบความปลอดภัยนั้น โดยส่วนใหญ่แล้วระบบเครือข่ายที่มีความปลอดภัยจะไม่ยอมให้มีการส่งผ่านคำสั่งด้านการจัดการเครือข่ายเข้าออกไปยังเครือข่ายอินเทอร์เน็ตได้ ดังนั้นการจัดการเครือข่ายจึงต้องกระทำภายในองค์กร หรือหน่วยงานซึ่งไม่สะดวกต่อผู้ดูและระบบเครือข่าย

1.2 แนวทางการวิจัย

วิทยานิพนธ์นี้จึงได้มุ่งเน้นการออกแบบระบบการจัดการเครือข่ายผ่านเว็บเพื่อให้เหมาะสมกับการใช้งานร่วมกับระบบเดิมโดยไม่จำเป็นต้องเปลี่ยนแปลงระบบเดิม เพราะการออก

แบบนี้ไม่กระทบกับส่วนเอเจนต์ อีกทั้งยังได้มีการประยุกต์ให้ส่วนจัดการ (Manager) สามารถทำงานได้บนอุปกรณ์ฝังตัว (Embedded Device) ซึ่งการจำกัดให้ระบบส่วนจัดการทำงานอยู่บนอุปกรณ์ฝังตัวเท่านั้น มีผลทำให้ระบบความปลอดภัยดียิ่งขึ้น เพราะผู้พัฒนาสามารถควบคุมอุปกรณ์ฝังตัวให้ทำหน้าที่เฉพาะเจาะจงต่อการจัดการเท่านั้น การดูแลในส่วนของความปลอดภัยจึงกำหนดขอบเขตได้แคบลง นอกจากนั้นแล้วการติดต่อกับเอเจนต์ใดๆ จากภายนอกเครือข่ายไม่สามารถกระทำโดยตรงได้ จำเป็นต้องติดต่อผ่านอุปกรณ์ฝังตัวเท่านั้น ซึ่งลดโอกาสในการโจมตีเอเจนต์จากภายนอกได้ อุปกรณ์ฝังตัวที่ได้พัฒนายังสามารถเคลื่อนย้ายได้สะดวก ติดตั้งให้เริ่มต้นทำงานได้อย่างรวดเร็ว สามารถตรวจพบอุปกรณ์ที่เป็นเอเจนต์ได้เอง อีกทั้งยังใช้พลังงานน้อยกว่าคอมพิวเตอร์ทั่วไป สามารถเปิดการทำงานได้ตลอดเวลาโดยไม่สิ้นเปลืองพลังงาน เมื่อเครือข่ายเกิดสิ่งผิดปกติขึ้น ส่วนจัดการนี้ยังสามารถจัดการกับสิ่งผิดปกติต่างๆ ได้อย่างรวดเร็ว เพื่อลดความรุนแรงของปัญหาได้

1.3 วัตถุประสงค์ของการวิจัย

เพื่อสร้างระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัว

1.4 ขอบเขตของการวิจัย

- ระบบรองรับการเข้าใช้งานทั้งหมด 5 งานพร้อมๆ กัน
- ระบบสามารถติดต่อกับเอสเอ็นเอ็มพีเอเจนต์ได้พร้อมกัน 3 เอเจนต์
- อุปกรณ์ฝังตัวที่นำมาใช้สามารถติดต่อกับเครือข่ายอีเทอร์เน็ตที่ความเร็ว 10 ล้านบิตต่อวินาที

1.5 ขั้นตอนการทำวิจัย

- พัฒนาส่วนฮาร์ดแวร์สำหรับระบบการจัดการเครือข่าย
- ออกแบบการทำงานของส่วนซอฟต์แวร์ส่วนระบบฝังตัว
- กำหนดวิธีการในการติดต่อระหว่างส่วนระบบฝังตัว กับส่วนเว็บ
- ออกแบบการทำงานของซอฟต์แวร์ส่วนเว็บ
- ทดสอบและปรับปรุงการทำงานระบบจัดการเครือข่ายผ่านเว็บทั้งระบบ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- เป็นแนวทางในการพัฒนาส่วนจัดการ และระบบจัดการเครือข่ายผ่านเว็บที่ดีขึ้น
- สามารถนำอุปกรณ์ฝังตัวที่ได้มาใช้ในการจัดการเครือข่ายได้สะดวก รวดเร็ว
- ป้องกันปัญหาที่อาจเกิดขึ้นกับระบบเครือข่ายได้อย่างทันท่วงที
- สามารถนำอุปกรณ์ฝังตัวที่ได้มาประยุกต์ใช้กับเครือข่ายต่างสถานที่กัน และเข้าถึงได้จากทุกที่ผ่านเครือข่ายอินเทอร์เน็ต
- สามารถเปลี่ยนเว็บเพจเพื่อให้สอดคล้องกับความต้องการในปัจจุบันได้ โดยไม่จำเป็นต้องทำการลงโปรแกรมให้กับอุปกรณ์ฝังตัวใหม่

1.7 ลำดับขั้นตอนในการเสนอผลการวิจัย

งานวิจัยนี้ได้นำเสนอเอกสารและงานวิจัยที่เกี่ยวข้องไว้ในบทที่ 2 จากนั้นนำเสนอการออกแบบระบบจัดการคอมพิวเตอร์ผ่านเว็บแบบฝังตัวในบทที่ 3 ซึ่งกล่าวถึงระบบโดยรวม และอุปกรณ์ฝังตัวที่ได้นำมาใช้ในงานวิจัยนี้ บทที่ 4 กล่าวถึงการพัฒนาส่วนฮาร์ดแวร์ และซอฟต์แวร์ พร้อมทั้งการทำงานร่วมกันของทั้ง 2 ส่วน จากนั้นทดสอบการทำงานของระบบโดยแสดงให้เห็นถึงขีดความสามารถของโปรแกรมในบทที่ 5 และสรุปผลการวิจัย พร้อมทั้งข้อเสนอแนะภายในบทที่ 6

1.8 ผลงานที่ตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการ และนำเสนอในการประชุมวิชาการ คือ

Chanin Maharak, Boonchai Sowanwanichakul, Web-based Network Management using Embedded System, The 4th Information and Computer Engineering Postgraduate Workshop 2004, Jan. 2004, Pages: 63 - 68

Chanin Maharak, Boonchai Sowanwanichakul, Security Methods for Web-based Application on Embedded System, IEEE TENCON 2004, to appear on November 2004

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

การจัดการเครือข่ายคอมพิวเตอร์เป็นงานที่สำคัญ เนื่องจากอุปกรณ์เครือข่ายคอมพิวเตอร์หลายชนิดที่ทำงานร่วมกันจำเป็นต้องมีการเฝ้าสังเกตการทำงานอย่างเหมาะสม ในปัจจุบันจึงมีการนำข้อตกลงสำหรับการจัดการเครือข่ายคอมพิวเตอร์มาใช้ โดยข้อตกลงที่เป็นที่นิยมมากที่สุดคือ ข้อตกลงเอสเอ็นเอ็มพี แต่เนื่องด้วยความก้าวหน้าทางเทคโนโลยี ทำให้การพัฒนาการจัดการเครือข่ายคอมพิวเตอร์เป็นไปอย่างต่อเนื่อง และมีแนวโน้มไปสู่การจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บ (Web-based Network Management)

2.1.1 การจัดการเครือข่ายคอมพิวเตอร์

การจัดการเครือข่ายคอมพิวเตอร์สื่อถึงการเฝ้าดูสถานะปัจจุบันของเครือข่ายและอุปกรณ์ต่างๆ ที่ใช้ภายในเครือข่าย รวมไปถึงการรับทราบเหตุการณ์ต่างๆ เพื่อใช้ตรวจสอบความผิดปกติที่อาจเกิดขึ้นได้ เมื่อระบบการจัดการตรวจพบสิ่งที่อาจทำให้ระบบทำงานผิดพลาด ระบบการจัดการเครือข่ายต้องกำหนดให้อุปกรณ์ต่างๆ เปลี่ยนแปลงการทำงานเพื่อให้เหมาะสมกับสถานการณ์นั้นๆ การจัดการเครือข่ายได้มุ่งเน้นไปที่การจัดการกับอุปกรณ์จำพวกอุปกรณ์จัดเส้นทาง (Router) สวิตช์ (Switch) และฮับ (Hub) ซึ่งได้มีงานวิจัยเกี่ยวกับประเภทของการจัดการ [1] โดยได้แบ่งประเภทของการจัดการตามลักษณะการทำงานไว้ 5 รูปแบบคือ การจัดการกับความผิดพลาด (Fault Management) การจัดการเกี่ยวกับการกำหนดการทำงาน (Configuration Management) การจัดการระบบบัญชีผู้ใช้ (Accounting Management) การจัดการด้านประสิทธิภาพ (Performance Management) และการจัดการด้านความปลอดภัย (Security Management) โดยเรียกการจัดการทั้ง 5 รูปแบบโดยรวมว่า FCAPS [2] ซึ่งการจัดการเครือข่ายจำเป็นต้องใช้รูปแบบหลายรูปแบบผสมผสานกันอย่างเหมาะสม เพื่อให้การจัดการเป็นไปอย่างมีประสิทธิภาพ

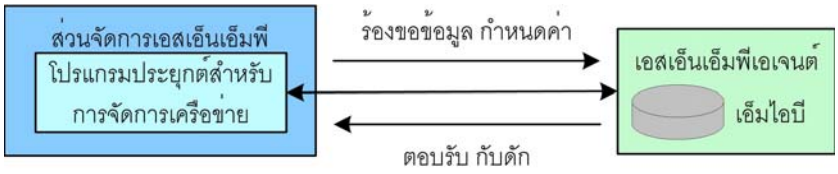
2.1.2 เอสเอ็นเอ็มพี

เอสเอ็นเอ็มพีเป็นข้อตกลงในการจัดการเครือข่ายคอมพิวเตอร์ชนิดหนึ่ง ซึ่งเป็นที่แพร่หลายเนื่องจากความง่ายในการจัดการของข้อตกลง ซึ่งได้แบ่งส่วนการทำงานออกเป็นสองส่วนด้วยกันคือ ส่วนเอเจนต์ (SNMP Agent) และส่วนจัดการ (SNMP Manager) โดยส่วนเอเจนต์

ทำหน้าที่เก็บรวบรวมข้อมูลต่างๆ เกี่ยวกับเครือข่าย เช่น จำนวนข้อมูลที่ไหลผ่านเครือข่าย จำนวนข้อมูลที่เสียหาย ความหนาแน่นของการสื่อสารข้อมูล หรือความเร็วของแต่ละอินเทอร์เฟสเป็นต้น ซึ่งข้อมูลต่างๆ เหล่านี้เมื่อนำมาแปลความหมายแล้ว ทำให้รู้ถึงสภาพหรือสถานะโดยรวมของเครือข่ายที่เอเจนต์ติดต่ออยู่ ณ ขณะนั้น โดยเอเจนต์เก็บข้อมูลต่างๆ ดังกล่าวข้างต้นไว้ในฐานข้อมูลสำหรับเอเจนต์ซึ่งเรียกว่า เอ็มไอบี (MIB: Management Information Base) ซึ่งทุกๆ เอเจนต์ต้องมีเอ็มไอบีอยู่ภายใน ข้อมูลภายในเอ็มไอบีนั้นถูกเก็บในลักษณะของอ็อบเจกต์ (Object) ซึ่งการระบุถึงอ็อบเจกต์แต่ละตัวนั้นใช้หมายเลขโอไอดี (OID: Object Identification) ในการอ้างอิง หมายเลขโอไอดีนั้นเป็นตัวเลขหลายชุดซึ่งแต่ละชุดถูกแบ่งด้วยเครื่องหมายมหัพภาค (.) เช่น 1.3.6.1.2.1.1.1.0

เนื่องจากว่าระบบการทำงานของเอสเอ็นเอ็มพีนั้น มีการทำงานในลักษณะของระบบรับ-ให้บริการ (client-server system) ผู้ดูแลระบบสามารถร้องขอข้อมูลต่างๆ ที่เก็บไว้ในเอเจนต์ใดๆ โดยเรียกใช้ผ่านส่วนจัดการ จากนั้นส่วนจัดการจึงติดต่อไปยังเอเจนต์ผ่านทางช่องทางยูดีพี (UDP Port) หมายเลข 161 การร้องขอข้อมูลนั้นประกอบไปชุดของหมายเลขโอไอดี และคอมมิวนิตี้สตริง (Community string) ซึ่งเปรียบเสมือนรหัสผ่านในการเข้าใช้เอเจนต์ เมื่อเอเจนต์ได้รับการร้องขอจะตรวจสอบสิทธิ์ของคอมมิวนิตี้สตริงนั้น จากนั้นนำข้อมูลที่เก็บอยู่ในเอ็มไอบีส่งกลับไปให้ส่วนจัดการ ผู้ดูแลสามารถกำหนดค่าของข้อมูลต่างๆ ของเอเจนต์ได้โดยสั่งส่วนจัดการให้กำหนดค่าของเอเจนต์ ซึ่งอาจมีผลทำให้ลักษณะการทำงานของเครือข่ายเปลี่ยนไป มีข้อสังเกตอยู่บางประการเกี่ยวกับการรับส่งข้อมูลของข้อตกลงนี้คือ เนื่องจากเอสเอ็นเอ็มพีนั้นถูกออกแบบมาเพื่อความง่ายในการจัดการเครือข่าย ดังนั้นข้อมูลที่ถูกส่งผ่านระหว่างส่วนเอเจนต์ และส่วนจัดการจึงเป็นข้อมูลที่มิได้ถูกเข้ารหัสลับ (Encrypt)

การทำงานตามข้อตกลงของเอสเอ็นเอ็มพีนั้นได้คำนึงถึงเหตุการณ์ผิดปกติที่อาจเกิดขึ้นกับเครือข่ายที่ส่วนเอเจนต์ติดต่อไว้ด้วย ซึ่งส่วนจัดการไม่อาจรับรู้ได้ทันท่วงที จากที่ได้กล่าวมาแล้วในข้างต้น ข้อมูลต่างๆ จึงเกิดจากการร้องขอข้อมูลของส่วนจัดการฝ่ายเดียวเท่านั้น ดังนั้นเอสเอ็นเอ็มพีจึงมีรูปแบบการส่งข้อมูลอีกแบบเรียกว่า เอสเอ็นเอ็มพีแทรพ (SNMP Traps) ซึ่งผู้ดูแลระบบสามารถกำหนดเหตุการณ์ที่ต้องการดักรอได้ตามความเหมาะสมของแต่ละเอเจนต์ เมื่อเอเจนต์ใดพบเหตุการณ์ที่ได้ถูกกำหนดไว้ เอเจนต์จะส่งข้อมูลไปยังส่วนจัดการที่ระบุให้เป็นปลายทางในทันที โดยส่งผ่านทางช่องทางยูดีพีหมายเลข 162 ซึ่งส่วนจัดการต้องนำข้อมูลที่ได้รับมาพิจารณา เพื่อป้องกันปัญหาที่อาจเกิดขึ้นได้จากเหตุการณ์เหล่านั้น การทำงานของเอสเอ็นเอ็มพีได้แสดงไว้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงระบบการจัดการเครือข่ายตามข้อตกลงเอสเอ็นเอ็มพี

จากรูปแสดงให้เห็นชนิดของคำสั่ง 4 ประเภทด้วยกันดังนี้

- คำสั่งประเภทร้องขอข้อมูล (Get) เป็นคำสั่งที่ถูกส่งจากส่วนจัดการไปสู่ส่วนเอเจนต์ ตัวอย่างของคำสั่ง เช่น GetRequest และ GetNextRequest ซึ่งหลังจากเอเจนต์ได้รับคำสั่งจะทำการค้นหาข้อมูลในเอ็มไอบีที่ตรงกับข้อมูลที่ถูกร้องขอมากับคำสั่งและตอบรับกลับด้วยคำสั่งประเภทการตอบรับ
- คำสั่งประเภทกำหนดค่า (Set) เป็นคำสั่งที่ถูกส่งจากส่วนจัดการไปสู่ส่วนเอเจนต์เช่นเดียวกับคำสั่งประเภทแรก แต่มีการกำหนดค่าเป็นข้อมูลไปกับคำสั่งด้วย คำสั่งที่อยู่ในประเภทนี้คือ SetRequest หลังจากเอเจนต์ได้รับคำสั่งจะกำหนดค่าต่างๆ ลงไป และส่งการตอบรับกลับไปยังส่วนจัดการด้วยคำสั่งประเภทการตอบรับ หากการกำหนดค่าไม่สำเร็จต้องกำหนดสถานะผิดพลาด (error-status) กลับไปพร้อมกับการตอบรับด้วย
- คำสั่งประเภทการตอบรับ (Response) เป็นคำสั่งตอบรับที่ส่งจากเอเจนต์มายังส่วนจัดการเพื่อเป็นการตอบรับกับคำสั่งประเภทร้องขอข้อมูล และกำหนดค่า คำสั่งที่อยู่ในประเภทนี้คือ GetResponse
- คำสั่งประเภทเอสเอ็นเอ็มพีแทรพ (Traps) เป็นคำสั่งที่ต่างจากคำสั่งชนิดอื่นเนื่องจากถูกส่งจากส่วนเอเจนต์ไปยังส่วนจัดการเมื่อพบเหตุการณ์ที่กำหนดไว้ ซึ่งมีเหตุการณ์ทั่วๆ ไปที่ระบุไว้ตามข้อตกลงหลายเหตุการณ์ด้วยกัน เช่น coldStart เกิดขึ้นเมื่อมีการเปิดการทำงานของเอเจนต์ warmStart เกิดขึ้นเมื่อมีการสั่งรีเซตเอเจนต์ linkDown เกิดเมื่อการสื่อสารไปยังเอเจนต์ผิดพลาด linkUp เกิดขึ้นเมื่อสามารถสื่อสารกับเอเจนต์ได้ authenticationFailure เมื่อการพิสูจน์ตนกับเอเจนต์ผิดพลาด egpNeighborLoss เกิดเมื่อเอเจนต์เพื่อนบ้านขาดการติดต่อ และ enterpriseSpecific เป็นแทรพพิเศษ

นับตั้งแต่การพัฒนาเอสเอ็นเอ็มพีรุ่นที่ 1 (SNMP [3]) ผู้ดูแลระบบเครือข่ายนิยมใช้กันอย่างแพร่หลายเนื่องจากความสะดวกและเป็นมาตรฐาน แต่เนื่องจากปัญหาด้านความปลอดภัยทำให้มีเอสเอ็นเอ็มพีรุ่นที่สอง (SNMPv2) เกิดขึ้นซึ่งไม่ได้เปลี่ยนแปลงไปจากรุ่นแรกมาก

นัก แต่เอสเอ็นเอ็มพีรุ่นที่ 3 (SNMPv3) ได้เพิ่มระดับความปลอดภัยเข้าไปกับข้อตกลง ซึ่งผู้ใช้สามารถเลือกระดับของความปลอดภัยได้หลายระดับ

2.1.3 ระบบฝังตัว (Embedded System)

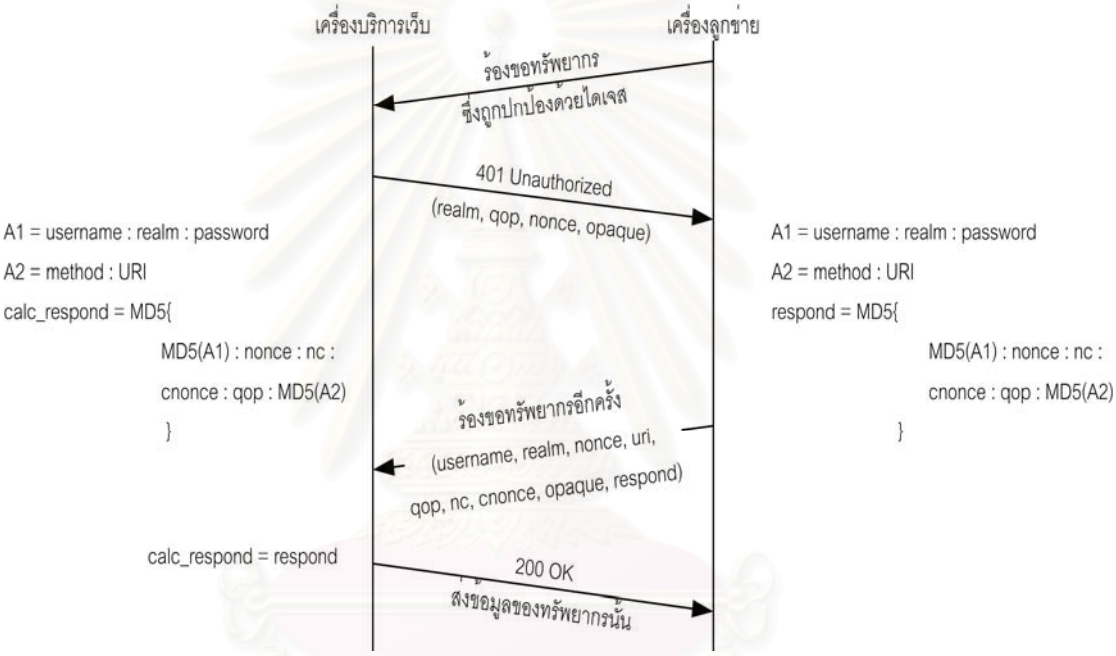
ระบบฝังตัว เป็นระบบที่ถูกออกแบบมาสำหรับการทำงานเฉพาะด้าน ประกอบไปด้วยส่วนของฮาร์ดแวร์ และซอฟต์แวร์เพื่อทำงานร่วมกัน โดยมีขนาดที่เล็กสามารถนำมาใส่โปรแกรมสำหรับการทำงานเฉพาะด้านที่สร้างขึ้นเอง หรือนำไปประกอบการทำงานกับอุปกรณ์อื่นเพื่อให้อุปกรณ์นั้นๆ มีความสามารถในการทำงานที่เพิ่มขึ้นในด้านที่ต้องการ โดยอุปกรณ์ที่ได้ออกแบบบนพื้นฐานของระบบฝังตัวเรียกว่า อุปกรณ์ฝังตัว (Embedded Device) การนำอุปกรณ์ฝังตัวเข้ามาใช้มีประโยชน์ในหลายแง่ ดังนี้

- ออกแบบมาเพื่อใช้งานเฉพาะด้าน ดังนั้นจึงทำงานด้วยความรวดเร็ว และมีประสิทธิภาพในการทำงานสูง
- อายุการใช้งานนานไม่จำเป็นต้องเปลี่ยนฮาร์ดแวร์ หรือซอฟต์แวร์ หากหน้าที่การทำงาน และความสามารถในการทำงานยังเป็นที่พอใจของผู้ใช้ระบบ
- ความปลอดภัยสูงเนื่องจากมีการออกแบบที่เฉพาะเจาะจง ดังนั้นจึงลดความเสี่ยงในการถูกโจมตีลงได้มาก
- ประหยัดพลังงาน สามารถเปิดการทำงานไว้ได้ตลอดเวลาโดยไม่สิ้นเปลือง

2.1.4 การพิสูจน์ตนของเอชทีทีพี

ตามมาตรฐานของเอชทีทีพี 1.1 นั้นมีวิธีการพิสูจน์ตนอยู่ด้วยกัน 2 วิธี คือ การเข้าถึงด้วยการพิสูจน์ตนแบบพื้นฐาน (Basic access authentication) และ การเข้าถึงด้วยการพิสูจน์ตนแบบไดเจส (Digest access authentication) [4] วิธีการพิสูจน์ตนแบบพื้นฐานนั้นเป็นวิธีการที่ง่ายโดยส่งชื่อผู้ใช้และรหัสผ่านในรูปการเข้ารหัสชนิด Base-64 ซึ่งไม่สามารถปกปิดข้อมูลดังกล่าวได้ สำหรับวิธีการพิสูจน์ตนแบบไดเจสนั้น ทรัพยากรที่ถูกป้องกันจะถูกปกป้องด้วยชื่อผู้ใช้ และรหัสผ่านเช่นกัน แต่ทางเครื่องลูกข่าย (Client) มิได้ส่งรหัสผ่านไปในเครือข่ายด้วย เพียงแต่ต้องอาศัยรหัสผ่านซึ่งเป็นข้อมูลลับมาใช้ประกอบกับข้อมูลอื่นๆ เพื่อสร้างตัวแทนข้อมูล (Message Digest) ส่งเป็นค่าตอบกลับ (Respond) ให้เครื่องบริการ (Server) ตรวจสอบ และอนุญาตให้เข้าถึงทรัพยากรนั้นๆ ได้ สำหรับงานวิจัยนี้ได้เลือกใช้การพิสูจน์ตนแบบไดเจสเนื่องจากมีความปลอดภัยที่มากกว่า ซึ่งตัวอย่างของการร้องขอทรัพยากรที่ถูกปกป้องด้วยการพิสูจน์ตนแบบไดเจส โดยการ

เลือกอัลกอริทึมเอ็มดี 5 (MD5) [5] และค่า “qop” เป็น “auth” มีขั้นตอนของการพิสูจน์ตนแบบไดเจสดังรูปที่ 2.2 ส่วนของรายละเอียดการทำงานของการพิสูจน์ตนแบบไดเจสนั้น สามารถศึกษาเพิ่มเติมได้จากการพิสูจน์ตนแบบไดเจสของมาตรฐานเอชทีทีพี 1.1 ซึ่งสังเกตได้ว่าข้อมูลลับที่ทั้งสองฝ่ายจำเป็นต้องรู้ตรงกันนั้นก็คือ รหัสผ่าน (password) ซึ่งในการส่งข้อมูล หรือตอบรับกลับนั้นไม่มีการส่งรหัสผ่านใดๆ ออกไปเลย ดังนั้นรหัสผ่านจึงไม่ถูกเปิดเผยด้วยการดักจับข้อมูล นอกจากนั้นแล้วค่าตอบกลับยังขึ้นกับค่าต่างๆ อีกหลายค่า ซึ่งหากมีค่าใดแม้ค่าเดียวเปลี่ยนไปค่าตอบกลับที่ได้ก็จะไม่เหมือนกันส่งผลให้มีความปลอดภัยที่สูงขึ้น



รูปที่ 2.2 ตัวอย่างวิธีการพิสูจน์ตนแบบไดเจส

2.1.5 เออีเอส (AES: Advance Encryption Standard)

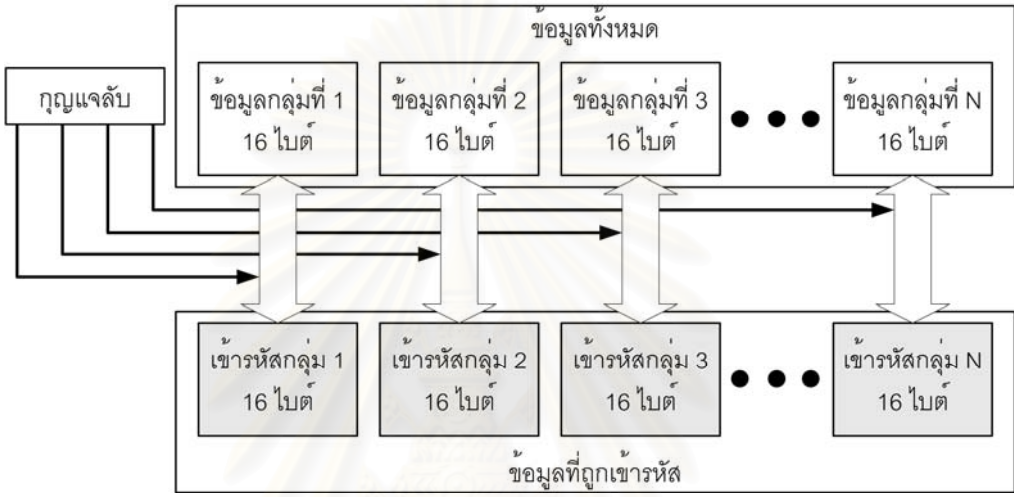
เออีเอสเป็นวิธีการเข้ารหัสข้อมูลชนิดกุญแจสมมาตร (Symmetric Key Encryption) ซึ่งการเข้ารหัสและถอดรหัสนั้นใช้กุญแจลับเดียวกัน ต่างกับการเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric Key Encryption) ซึ่งกุญแจลับสำหรับการเข้ารหัสจะต้องเป็นคู่ซึ่งกันและกันแต่ไม่เหมือนกัน สำหรับการใช้งานสำหรับระบบฝังตัวนั้นไม่นิยมใช้การเข้ารหัสแบบอสมมาตรเนื่องจากใช้ทรัพยากรมาก อีกทั้งยังมีการประมวลผลข้อมูลที่ซับซ้อนไม่เหมาะกับระบบฝังตัวขนาดเล็กซึ่งมีทรัพยากรอยู่อย่างจำกัด และยังต้องมีการแบ่งทรัพยากรการทำงานให้แก่การทำงานส่วนอื่นๆ อีกด้วย เออีเอสนั้นใช้ทรัพยากรที่น้อยกว่าและมีการประมวลผลที่รวดเร็วกว่า ดังนั้นเออีเอสจึงเป็นอัลกอริทึมที่เหมาะสำหรับระบบฝังตัว เออีเอสถือกำเนิดเนื่องจากกุญแจที่ใช้ใน

วิธีการเข้ารหัสแบบเดส (DES) นั้นมีขนาดที่สั้นเกินไปทำให้มีความปลอดภัยต่ำ และในขณะเดียวกันการเข้ารหัสแบบทริปเปิลเดส (3DES) นั้นปลอดภัยมากขึ้นแต่ทำได้ช้า การเข้ารหัสของเออีเอสประกอบด้วย 4 ขั้นตอนด้วยกันดังรูปที่ 2.3

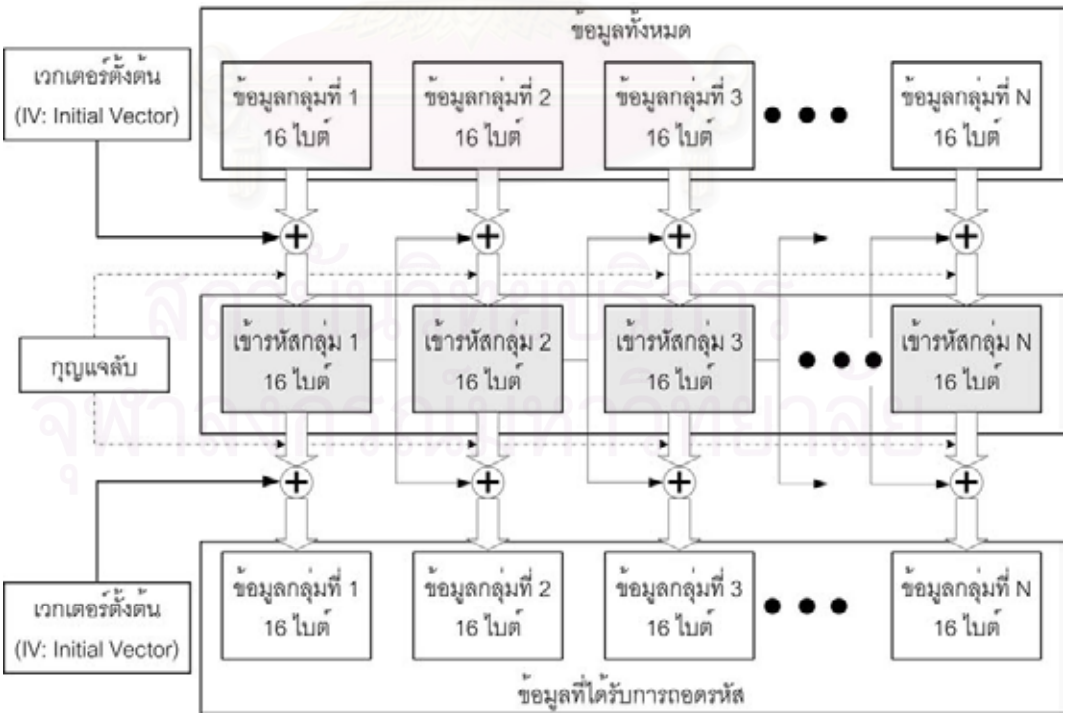
รูปที่ 2.3 วิธีการเข้ารหัสแบบเออีเอส

การเข้ารหัสแบบเออีเอสนั้นจำเป็นต้องใช้ตารางสำหรับการแทนที่ (Substitution) ตารางสำหรับการผสมสดมภ์ (Mix Columns) และกุญแจลับ (Secret Key) ซึ่งเมื่อข้อมูลถูกเข้ารหัสแล้ว สามารถถอดรหัสโดยใช้ตารางสำหรับการแทนที่ผกผัน (Inverse Substitution) ตารางสำหรับการผสมสดมภ์ผกผัน (Inverse Mix Columns) และกุญแจลับที่ตรงกับการเข้ารหัสในครั้งนั้นๆ [6, 7] การเข้ารหัสแบบเออีเอสนั้นเป็นการเข้ารหัสสำหรับชนิดข้อมูลเป็นกลุ่ม (Block) โดยแต่ละกลุ่มข้อมูลมีขนาด 16 ไบต์ ข้อมูลแต่ละกลุ่มสามารถเข้ารหัสแยกจากกันได้โดยไม่ขึ้นต่อกันดังแสดงได้ตามรูป 2.4 อย่างไรก็ตามการเข้ารหัสชุดข้อมูลด้วยวิธีดังกล่าวข้างต้นเป็นวิธีที่มีความปลอดภัยต่ำมาก เนื่องจากหากรู้ถึงข้อมูลกลุ่มใดและผลการเข้ารหัสของข้อมูลกลุ่มนั้น ก็สามารถนำมาลองสุ่มเปลี่ยนกุญแจลับไปเรื่อยๆ จนกว่าข้อมูลที่ถูกเข้ารหัสจะตรงกัน และสามารถเปิดเผยกุญแจลับที่ใช้ในการเข้ารหัสในครั้งนั้นได้ ดังนั้นการเข้ารหัสชุดข้อมูลจึงมีวิธีการอีกหลายวิธีที่มีความปลอดภัยที่สูงขึ้นด้วยการทำให้การเข้ารหัสข้อมูลแต่ละกลุ่มนั้นมีข้อมูลบางส่วนซึ่งไปผสมอยู่กับกุญแจลับทำให้การเข้ารหัสมีผลเปลี่ยนไปถึงแม้กลุ่มข้อมูลที่ตามมาจะมีข้อมูลที่เหมือนกันก็ตาม ยกตัวอย่างเช่น อีซีบี (ECB: Electronic Code Book) ซีบีซี (CBC: Cipher Block Chaining) โอเอฟบี (OFB: Output FeedBack) และซีเอฟบี (CFB: Cipher Feedback) สำหรับอีซีบี และซีบีซีนั้นเป็นวิธีสำหรับการเข้ารหัสกลุ่มข้อมูลโดยแบ่งข้อมูลออกเป็นหลายชุด ซึ่งต้องมีข้อมูลครบทั้งชุดก่อนจึงสามารถทำได้ แต่สำหรับโอเอฟบีและซีเอฟบีเป็นการเข้ารหัสโดยไม่จำเป็นต้องรอข้อมูล

ให้ครบชุดก่อน สำหรับการวิจัยนี้ได้นำวิธีการของซีบีซีมาใช้เนื่องจากระบบฝังตัวมีทรัพยากรเพียงพอสำหรับเก็บข้อมูลให้ครบชุด (16 ไบต์) ดังนั้นจึงไม่จำเป็นต้องใช้โอเอฟบีและซีเอฟบีซึ่งมีความซับซ้อนกว่า การใช้ซีบีซีนั้นมีความปลอดภัยที่สูงกว่าอีซีบีโดยสามารถดูได้จากผลงานวิจัยที่นำรูปภาพมาทดสอบ [8] ผลที่ได้แสดงให้เห็นว่าการใช้อีซีบียังสามารถเห็นภาพโครงร่างของรูปภาพเดิมอยู่ซึ่งไม่ปลอดภัยเท่าซีบีซี โดยการทำงานของซีบีซีสามารถแสดงได้ตามรูปที่ 2.5



รูปที่ 2.4 การเข้ารหัสชุดข้อมูลแบบพื้นฐานด้วยเออีเอส



รูปที่ 2.5 แสดงการทำงานของเออีเอสด้วยวิธีซีบีซี

การเข้ารหัสเออีเอสด้วยวิธีซีบีซีนั้นอาศัยการใช้เวกเตอร์ตั้งต้น (Initial Vector) เพื่อนำมาผสมกับข้อมูลกลุ่มที่ 1 ด้วยวิธีการเอกซคลูซีฟออร์ (exclusive-or) ซึ่งให้ผลของการเข้ารหัสเปลี่ยนไปหากมีค่าของเวกเตอร์ตั้งต้นที่เปลี่ยนไป หลังจากนั้นผลของการเข้ารหัสข้อมูลกลุ่มใดๆ เปรียบเสมือนค่าเวกเตอร์ตั้งต้นของการเข้ารหัสข้อมูลในชุดถัดไป ดังนั้นวิธีการเข้ารหัสเออีเอสด้วยซีบีซีนั้นหากมีค่าเวกเตอร์ตั้งต้นที่เปลี่ยนไป ย่อมส่งให้ผลการเข้ารหัสที่ได้เปลี่ยนไป เวกเตอร์ตั้งต้นเป็นข้อมูลที่สามารถเปิดเผยได้ ดังนั้นหากต้องการให้ข้อมูลที่เกิดจากการเข้ารหัสมีความซับซ้อนมากขึ้น ส่วนโปรแกรมที่ทำหน้าที่เข้ารหัสต้องสร้างเวกเตอร์ตั้งต้นด้วยการสุ่มค่า และส่งเวกเตอร์ตั้งต้นไปด้วยเพื่อใช้เป็นเวกเตอร์ตั้งต้นที่ตรงกันในการถอดรหัส

2.2 งานวิจัยที่เกี่ยวข้อง

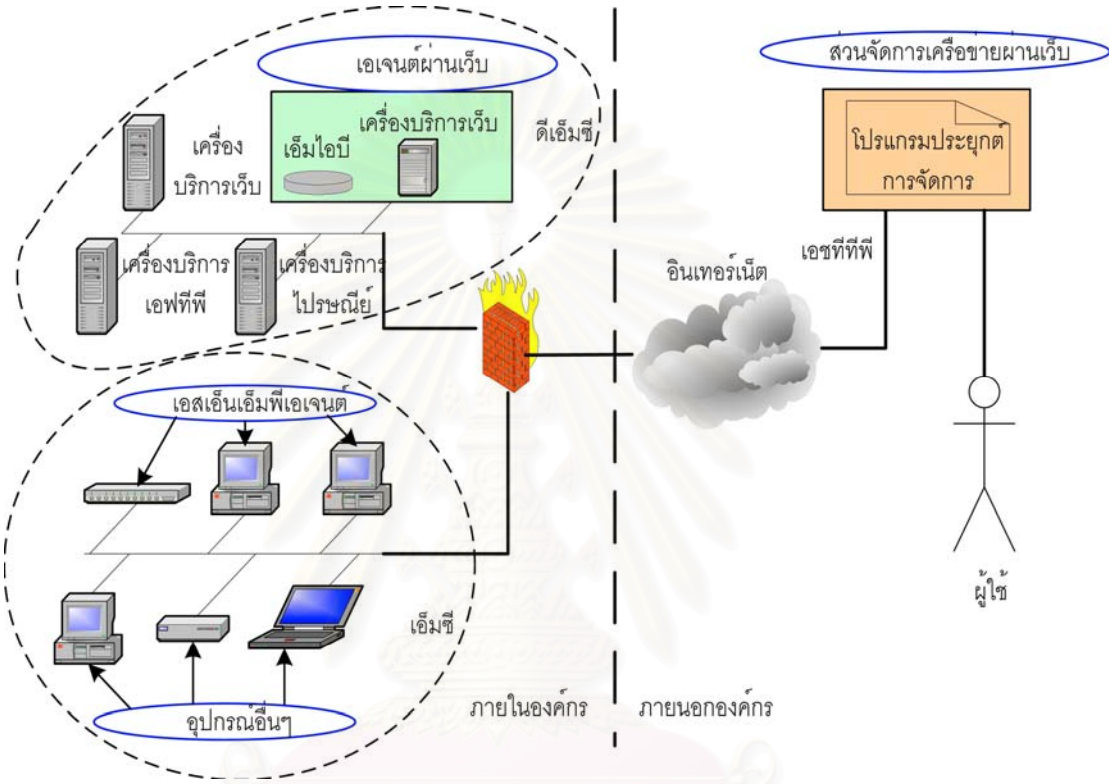
2.2.1 การจัดการเครือข่ายผ่านเว็บ (Web-based Network Management)

การจัดการเครือข่ายคอมพิวเตอร์ได้รับความนิยมอย่างแพร่หลาย แต่การใช้งานยังมีข้อจำกัดอยู่บางประการ ประกอบกับการทำงานผ่านเว็บได้รับความนิยมที่สูงขึ้นเนื่องจากผู้ใช้งานสามารถเข้าถึงได้จากทุกที่ที่มีเครือข่ายอินเทอร์เน็ต ดังนั้นการจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บจึงถูกพัฒนาขึ้น แต่เนื่องจากเอสเอ็นเอ็มพียังเป็นข้อตกลงที่มีผู้ใช้ยอมรับอยู่เป็นจำนวนมาก การพัฒนาให้ทำงานผ่านเว็บได้นั้นต้องสามารถทำงานร่วมกับระบบเอสเอ็นเอ็มพีเดิมได้ด้วย การจัดการเครือข่ายผ่านเว็บนับเป็นการพัฒนาให้ผู้ดูแลระบบสามารถใช้งานได้ง่ายขึ้น ซึ่งได้มีงานวิจัยในแง่ของการจัดการเครือข่ายโดยใช้เอสเอ็นเอ็มพีผ่านเว็บโดยมีการเปลี่ยนแปลงแก้ไขทั้งในส่วนของเอเจนต์และส่วนจัดการ [9] รูปที่ 2.6 แสดงระบบการจัดการคอมพิวเตอร์ผ่านเว็บ ซึ่งเห็นได้ว่าเมื่อต้องการให้ระบบทำงานผ่านเว็บนั้นจำเป็นต้องมีการเปลี่ยนแปลงในส่วนของเอเจนต์ ซึ่งหากในเครือข่ายมีเอเจนต์เดิมอยู่แล้ว และไม่สามารถเปลี่ยนหรือนำเอเจนต์ใหม่มาเพิ่มได้ ทำให้ระบบไม่สนับสนุนการทำงานผ่านเว็บ

นอกจากนี้ ยังมีงานวิจัยสำหรับการตั้งค่าสำหรับการจัดการเครือข่ายผ่านเว็บ [10] ซึ่งได้นำเสนอแบบจำลองสำหรับระบบการจัดการ และได้นำแบบจำลองมาสร้างเป็นระบบที่เรียกว่า WebNCMS (Web-based Network Configuration Management System) โดยส่วนจัดการเครือข่ายมีการทำงานแบบกระจาย (Distributed)

การติดต่อสื่อสารระหว่างเว็บเอสเอ็นเอ็มพีเอเจนต์ และส่วนจัดการเว็บเอสเอ็นเอ็มพีนั้น ต้องสื่อสารกันโดยผ่านเอชทีทีพี (HTTP) ดังนั้นจึงได้มีงานวิจัยที่นำเสนอการติดต่อกันใน

รูปแบบของซีจีไอ (CGI: Common Gateway Interface) มาใช้ประโยชน์ [11] โดยงานวิจัยได้นำเสนอแบบจำลองการจัดการเครือข่ายคอมพิวเตอร์ ซึ่งแบ่งการทำงานออกเป็น 5 ส่วนย่อย และกำหนดความสัมพันธ์ระหว่างแต่ละส่วน ซึ่งระบบโดยรวมทั้งหมดทำงานอยู่บนระบบปฏิบัติการ VxWorks ซึ่งเป็นระบบปฏิบัติการที่ใช้กันมากในการออกแบบระบบฝังตัว



รูปที่ 2.6 ระบบการจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บ

2.2.2 การจัดการเครือข่ายผ่านเว็บบนระบบฝังตัว

ในการนำอุปกรณ์ฝังตัวมาใช้สำหรับการจัดการเครือข่าย จำเป็นต้องมีส่วนบริการเว็บ ซึ่งมีงานวิจัยที่นำมาใช้ประยุกต์กับงานวิจัยนี้ โดยได้พัฒนาส่วนบริการเว็บแบบฝังตัวขึ้นเพื่อใช้กับเอสเอ็นเอ็มพีเอเจนต์ [12] ส่วนบริการเว็บนี้รองรับการทำงานกับข้อมูลทั้งชนิดสถิต (Static) และ พลวัต (Dynamic) ส่วนจัดการที่สามารถติดต่อกับเอสเอ็นเอ็มพีเอเจนต์ เป็นได้ทั้งส่วนจัดการเอสเอ็นเอ็มพี และส่วนจัดการเว็บเอสเอ็นเอ็มพี

นอกจากนี้ยังมีงานวิจัยที่นำการให้บริการเว็บ และเอสเอ็นเอ็มพีมาทำงานบนอุปกรณ์ฝังตัว [13] โดยเสนอแบบจำลองการจัดการแบบบูรณาการ (Integrated Management

Model) ซึ่งอาศัยหลักการทำงานสำคัญของ RapidControl Backplane (RCB) โดยเป็นชั้นของซอฟต์แวร์ที่ทำหน้าที่เปลี่ยนรูปแบบของข้อมูล ระหว่างข้อมูลการจัดการในรูปแบบพิเศษ และรหัสคำสั่งในการจัดการ

2.2.3 ความปลอดภัยบนระบบฝังตัว

การติดต่อสื่อสารระหว่างระบบฝังตัวและระบบอื่นๆ ผ่านทางเครือข่ายอินเทอร์เน็ตนั้น ข้อมูลต่างๆ ถูกส่งผ่านเครือข่ายสาธารณะ ดังนั้นหากไม่มีการปกป้องข้อมูลที่ดีเพียงพออาจเป็นอันตรายต่อการทำงานของระบบได้ จากความสำคัญดังกล่าวจึงได้มีงานวิจัยเกี่ยวกับการนำระบบเข้ารหัสข้อมูลมาประยุกต์ใช้ โดยการเปรียบเทียบการทำงานของเออีเอส และเดส [14] ซึ่งแสดงให้เห็นถึงประสิทธิภาพที่แตกต่างกัน นอกจากนี้แล้วยังมีงานวิจัยซึ่งนำเออีเอสและเดสมาเปรียบเทียบกันโดยใช้การวิเคราะห์เชิงอนุพันธ์ทางระบบรักษาความปลอดภัย (Differential Cryptanalysis) เพื่อศึกษาการกระจายตัวของข้อมูลเปรียบเทียบกับการกระจายตัวมาตรฐาน [15] ซึ่งผลที่ได้นั้นเดสมีค่าการกระจายตัวเฉลี่ยแตกต่างจากการกระจายตัวมาตรฐานอยู่ 28.7% ในขณะที่เออีเอสมีค่าการกระจายตัวเฉลี่ยแตกต่างจากการกระจายตัวมาตรฐานอยู่ 3.6% และงานวิจัยได้สรุปถึงระบบรักษาความปลอดภัยที่เหมาะสมควรมีค่าการกระจายตัวเฉลี่ยแตกต่างจากการกระจายตัวมาตรฐานอยู่ไม่เกิน 10% ดังนั้นเออีเอสจึงเหมาะสมกว่าถึงแม้ว่าการเลือกใช้เออีเอสจะส่งผลให้ระบบฝังตัวใช้ทรัพยากรมากขึ้นก็ตามแต่มากขึ้นในระดับที่ระบบฝังตัวสามารถทำงานได้ โดยหากระบบฝังตัวใช้เดสนั้นจะต้องใช้ทรัพยากรในการเก็บตารางค่าคงที่ประมาณ 1 กิโลไบต์ ในขณะที่หากใช้เออีเอสแล้วจะใช้ทรัพยากรในการเก็บตารางค่าคงที่ประมาณ 8 กิโลไบต์ ซึ่งเมื่อเทียบกับความปลอดภัยที่ได้รับมากขึ้นจากงานวิจัยที่ได้กล่าวมาทั้งหมดนั้นเห็นได้ว่าเหมาะสมในการใช้เออีเอส

บทที่ 3
ระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บแบบฝังตัว

3.1 ข้อกำหนดเบื้องต้น

เนื่องจากปัจจุบันการใช้งานเครือข่ายคอมพิวเตอร์ภายในหน่วยงาน หรือองค์กร มีความนิยมในการใช้งานระบบอีเทอร์เน็ต (Ethernet) เป็นอย่างมาก ดังนั้นงานวิจัยนี้จึงได้มุ่งเน้นไปที่การออกแบบระบบจัดการเครือข่ายที่มีระดับชั้นกายภาพเป็นอีเทอร์เน็ต การใช้งานของเอสเอ็นเอ็มพีนั้นเป็นที่แพร่หลายบนระบบเครือข่าย ซึ่งระบบการจัดการเครือข่ายที่ได้ออกแบบทำงานอยู่บนพื้นฐานของเอสเอ็นเอ็มพี แต่เนื่องด้วยเหตุผลด้านความปลอดภัยเครือข่ายส่วนใหญ่มักไม่อนุญาตให้ข้อมูลเอสเอ็นเอ็มพีไหลผ่านไฟร์วอลล์เข้ามายังหน่วยงานได้ ระบบจัดการเครือข่ายดังกล่าวจึงต้องสามารถทำงานผ่านเว็บเพื่อให้ไหลผ่านไฟร์วอลล์เข้ามาได้ โดยได้เลือกเอสเอ็นเอ็มพีรุ่นที่ 1 ซึ่งมีความนิยมในการใช้งานสูงที่สุด และรองรับการทำงานร่วมกับอุปกรณ์เอสเอ็นเอ็มพีทุกชนิดมาใช้กับระบบ จากการเปิดช่องทางให้ผู้ดูแลระบบสามารถติดต่อเข้ามาผ่านทางเว็บนั้นส่งผลให้ต้องมีระบบรักษาความปลอดภัยโดยต้องไม่สิ้นเปลืองการใช้ทรัพยากรของระบบฝังตัวมากนัก จึงได้เลือกใช้การพิสูจน์ตนแบบเอชทีทีพีไดเจสทำงานร่วมกับการเข้ารหัสข้อมูลด้วยเออีเอส

3.2 ระบบโดยรวม

ระบบจัดการเครือข่ายคอมพิวเตอร์ที่ออกแบบนั้น ต้องสามารถทำงานผ่านเว็บและทำงานร่วมกับระบบเดิมได้ ดังนั้นการออกแบบต้องเสมือนกับมีระบบเดิมอยู่ และเพิ่มหน้าที่การทำงานให้สามารถสั่งงานจัดการเครือข่าย และแสดงผลผ่านเว็บได้ โดยระบบการทำงานทั้งหมดประกอบไปด้วยส่วนประกอบหลักๆ 5 ส่วนดังนี้

- เอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent) เป็นอุปกรณ์หรือเครื่องคอมพิวเตอร์ที่ทำงานเป็นเอเจนต์เหมือนข้อตกลงเอสเอ็นเอ็มพี สามารถติดต่อกับส่วนจัดการผ่านเอสเอ็นเอ็มพี
- ส่วนจัดการเอสเอ็นเอ็มพี (SNMP Manager) เป็นส่วนจัดการเหมือนกับส่วนจัดการของข้อตกลงเอสเอ็นเอ็มพี ติดต่อสื่อสารกับส่วนเอเจนต์ผ่านทางข้อตกลงเอสเอ็นเอ็มพี ซึ่งส่วนนี้อาจมีหรือไม่มีขึ้นอยู่กับความต้องการของผู้พัฒนา
- ส่วนจัดการผ่านเว็บ (Web-based Manager) เป็นส่วนจัดการที่ถูกพัฒนาขึ้นมาใหม่ มีความสามารถในการจัดการเหมือนกับส่วนจัดการเอสเอ็นเอ็มพี แต่เพิ่มความสามารถในการสั่งงานและแสดงผลผ่านทางเว็บบราวเซอร์ด้วย ซึ่งส่วนจัดการผ่านเว็บต้องแปลความ

หมายของข้อมูลที่ถูกส่งมาจากเว็บบราวเซอร์ และสั่งงานไปยังส่วนเอเจนต์ หลังจากส่วนเอเจนต์ทำงานเสร็จต้องทำการแปลงผลที่เกิดขึ้นให้สามารถแสดงผลขึ้นบนเว็บบราวเซอร์ การติดต่อกับส่วนเอเจนต์ยังคงใช้ข้อตกลงเอสเอ็นเอ็มพี ดังนั้นไม่จำเป็นต้องเปลี่ยนแปลงเอสเอ็นเอ็มพีเอเจนต์ใดๆ ทั้งสิ้น

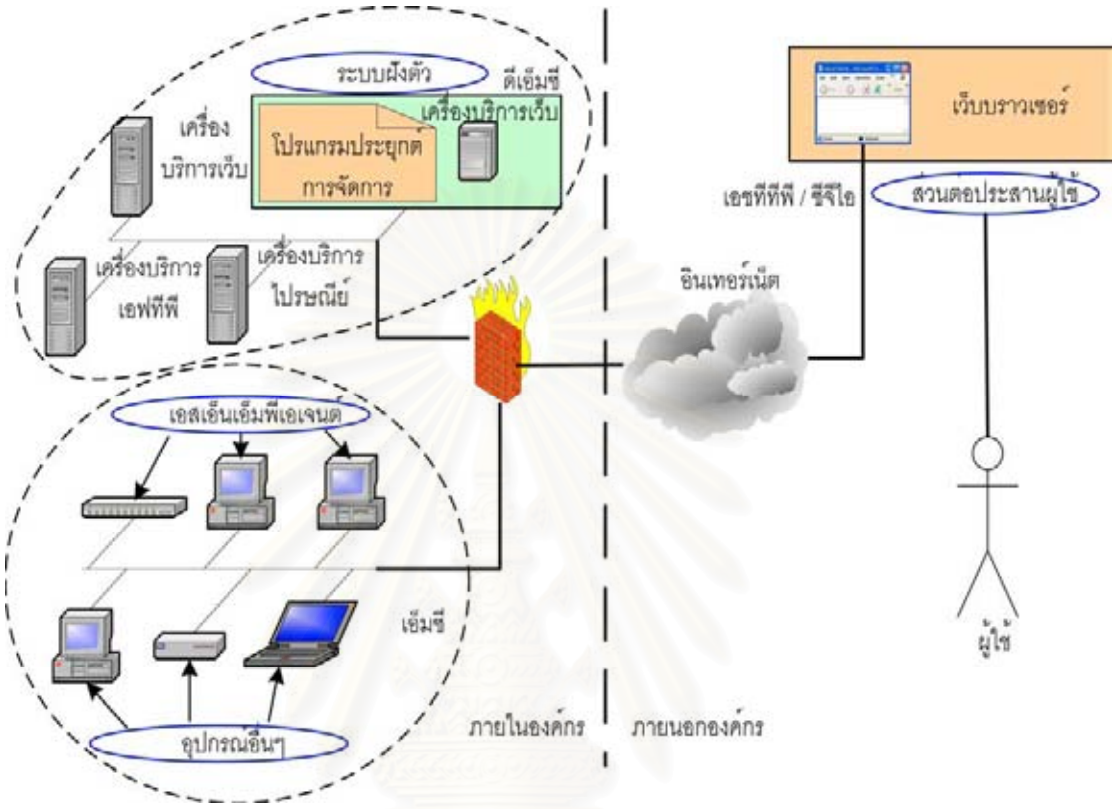
- เว็บบราวเซอร์ มีหน้าที่เปรียบเสมือนเป็นส่วนต่อประสานกับผู้ใช้ (user interface) ระหว่างส่วนจัดการผ่านเว็บ และผู้ใช้งาน หน้าจอของโปรแกรมสั่งงานสามารถส่งคำสั่งร้องขอ และสั่งงานผ่านทางเอชทีทีพีไปยังส่วนจัดการผ่านเว็บ และเมื่อส่วนจัดการผ่านเว็บทำงานเสร็จสิ้นต้องแสดงผลของการทำงานในครั้งนั้น
- ผู้ดูแลระบบ เป็นผู้ที่มีหน้าที่รับผิดชอบกับการดูแลระบบเครือข่ายของหน่วยงาน โดยสามารถดูแลระบบได้โดยผ่านส่วนจัดการเอสเอ็นเอ็มพีภายในหน่วยงาน หรือผ่านทางเว็บบราวเซอร์จากเครือข่ายอินเทอร์เน็ต

ระบบการจัดการนี้ต้องมีลักษณะการทำงานที่สะดวก ดังนั้นจึงได้ออกแบบระบบให้เริ่มต้นทำงานได้โดยง่าย ผู้ดูแลระบบสามารถนำอุปกรณ์ฝังตัวไปเชื่อมต่อกับระบบเครือข่ายอีเทอร์เน็ต จากนั้นอุปกรณ์ฝังตัวจะร้องขอเลขที่อยู่ไอพีจากเครื่องบริการดีเอชซีพี (DHCP Server) ซึ่งหากไม่มีการเปิดการใช้งานเครื่องบริการดีเอชซีพี อุปกรณ์ฝังตัวจะเลือกใช้เลขที่อยู่ไอพีชนิดคงที่ (Static IP Address) ที่ได้กำหนดไว้

รูปที่ 3.1 แสดงตัวอย่างระบบการทำงานโดยรวม ซึ่งเครือข่ายภายในหมายถึงระบบเครือข่ายที่อยู่หลังไฟร์วอลล์ ส่วนเครือข่ายภายนอกเป็นเครือข่ายอินเทอร์เน็ตภายนอกไฟร์วอลล์ เครือข่ายภายในประกอบไปด้วย 2 โซน (Zone) คือ ดีเอ็มซี (DMZ: Demilitarized Zone) และ เอ็มซี (MZ: Militarized Zone) โดยเครื่องจากเครือข่ายภายนอกสามารถติดต่อมายังอุปกรณ์ภายในดีเอ็มซีโซนได้โดยตรงเฉพาะช่องทางที่ไฟร์วอลล์อนุญาตเท่านั้น เหมาะสำหรับการนำเครื่องบริการต่างๆ มาอยู่ภายในโซนนี้ แต่เครื่องจากภายนอกเครือข่ายไม่สามารถติดต่อมายังอุปกรณ์ภายในเอ็มซีโซนได้โดยตรง

ระบบฝังตัวทำหน้าที่เป็นส่วนจัดการผ่านเว็บ โดยทำหน้าที่หลัก 2 หน้าที่ด้วยกัน คือ ทำหน้าที่ดูแลจัดการการทำงานของเอเจนต์ให้เหมาะสมและให้บริการเว็บ ผู้ดูแลระบบสามารถสั่งงานอุปกรณ์ที่เป็นเอสเอ็นเอ็มพีเอเจนต์โดยใช้ส่วนจัดการเอสเอ็นเอ็มพี (หากมีการจัดหาไว้ในระบบ) หรือสั่งงานโดยใช้ส่วนจัดการผ่านเว็บ ผู้ดูแลระบบสามารถดูแลระบบต่างๆ เพียงรู้เลขที่อยู่ไอพี (IP address) ของส่วนจัดการเว็บเท่านั้น นอกจากนั้นแล้วผู้ดูแลระบบต้องมั่นใจว่า

เส้นทางระหว่างส่วนจัดการผ่านเว็บ และอุปกรณ์ที่เป็นเอสเอ็นเอ็มพีเอเจนต์นั้น สามารถส่งผ่านคำสั่งตามข้อตกลงเอสเอ็นเอ็มพีได้



รูปที่ 3.1 ตัวอย่างระบบการจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บแบบฝังตัว

การติดต่อระหว่างระบบฝังตัวกับเครือข่ายภายในนั้นประกอบด้วยการใช้มาตรฐานของเอสเอ็นเอ็มพีติดต่อไปยังเอสเอ็นเอ็มพีเอเจนต์ นอกจากนี้ยังมีการติดต่อไปยังเครื่องบริการดีเอชซีพีเพื่อร้องขอเลขที่อยู่ไอพี และการตรวจสอบสถานะของอุปกรณ์ต่างๆ (ping) ภายในเครือข่าย ซึ่งไฟร์วอลล์ต้องเปิดช่องทางให้เอสเอ็นเอ็มพีสามารถผ่านได้ระหว่างเอ็มซี และดีเอ็มซีโซน สำหรับการติดต่อกับเครือข่ายภายนอกนั้นเปิดให้บริการเพียง 2 บริการเท่านั้น คือการตรวจสอบสถานะของระบบฝังตัว และการให้บริการเว็บผ่านทางช่องทาง TCP/80 โดยไฟร์วอลล์ต้องกำหนดให้เปิดช่องทางเอชทีทีพี และไอซีเอ็มพี (ICMP) ติดต่อกับภายนอกได้ แต่ต้องไม่อนุญาตให้เอสเอ็นเอ็มพีผ่านเข้าไปยังเครือข่ายภายในองค์กรได้ ผู้ดูแลระบบที่ต้องการใช้บริการการจัดการเครือข่ายผ่านเว็บต้องส่งคำสั่งซีจีไอไปยังระบบฝังตัว หากร้องขอทรัพยากรที่มิได้มีการป้องกันด้วยการพิสูจน์ตนแบบเอชทีทีพีไดเจสไว้สามารถร้องขอแบบปกติได้ แต่หากมีการร้องขอทรัพยากรที่ปกป้องด้วยการพิสูจน์ตนแบบเอชทีทีพีไดเจสแล้ว ต้องส่งข้อมูลในรูปแบบตามมาตรฐานเอชทีทีพีไดเจสกำหนดไว้ นอกจากนี้แล้วการติดต่อผ่านเอชทีทีพีของระบบฝังตัวนี้ยังมีความสามารถในการ

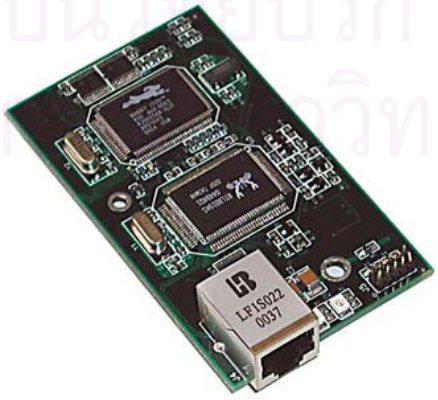
เข้ารหัส และถอดรหัสไว้ด้วย ซึ่งเลือกใช้อัลกอริทึมเออีเอสโดยการเข้ารหัสและถอดรหัสสามารถใช้ได้กับซีจีไอฟังก์ชัน (function) ที่ปกป้องด้วยเอชทีทีพีไดเจสเท่านั้น โดยวิธีการอย่างละเอียดได้อธิบายไว้ในบทถัดไป

ส่วนจัดการเครือข่ายผ่านเว็บที่ได้ออกแบบสามารถติดต่อกับเอสเอ็นเอ็มพีเอเจนต์โดยผ่านข้อตกลงเอสเอ็นเอ็มพี ดังนั้นระบบฝังตัวควรอยู่ในดีเอ็มซีโซน และเอสเอ็นเอ็มพีเอเจนต์ควรอยู่ในเอ็มซีโซนเพื่อป้องกันการติดต่อโดยตรงจากภายนอก และทำการตั้งค่าให้ไฟร์วอลล์ปิดกั้นข้อมูลเอสเอ็นเอ็มพีมิให้ผ่านเข้าออกไปสู่อินเทอร์เน็ตได้

3.3 อุปกรณ์ฝังตัว

อุปกรณ์ฝังตัวที่สามารถนำมาใช้ในการสร้างระบบ จำเป็นต้องปฏิบัติงานได้อย่างรวดเร็ว อีกทั้งยังต้องมีส่วนติดต่อกับระบบอีเทอร์เน็ต ดังนั้นจึงได้เลือกใช้อุปกรณ์อาร์ซีเอ็มรุ่น 2100 (RCM2100) ซึ่งเป็นอุปกรณ์ฝังตัวของบริษัทแรบบิทเซมิคอนดักเตอร์ (Rabbit Semiconductor) โดยมีคุณสมบัติต่างๆ ดังนี้

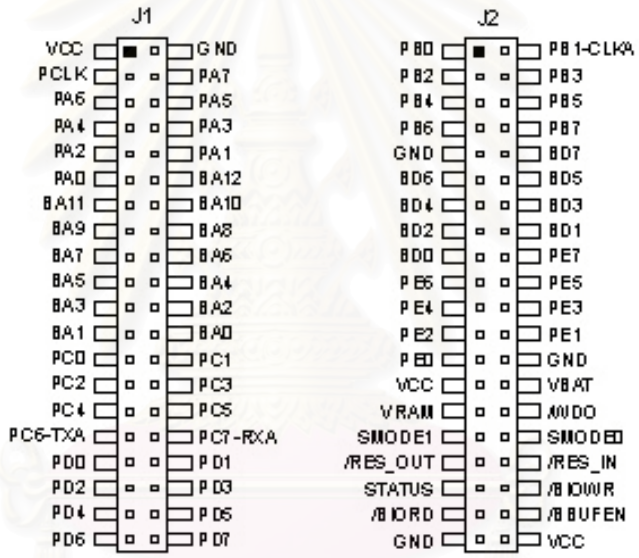
- ใช้หน่วยประมวลผล Rabbit 2000 ซึ่งเป็นหน่วยประมวลผลขนาด 8 บิต
- ทำงานด้วยอัตราสัญญาณนาฬิกา 22.1 ล้านรอบต่อวินาที (MHz)
- มีหน่วยความจำแฟลช (Flash memory) ขนาด 512 กิโลไบต์ (KB) และหน่วยความจำแรมสถิต (Static RAM) ขนาด 512 กิโลไบต์
- ส่วนต่อประสานอีเทอร์เน็ตอาร์เจ 45 (RJ-45) ทำงานที่ความเร็ว 10 ล้านบิตต่อวินาที (Mbps)



รูปที่ 3.2 อุปกรณ์อาร์ซีเอ็มรุ่น 2100

อุปกรณ์อาร์ซีเอ็มรุ่น 2100 เป็นอุปกรณ์ขนาดเล็ก ดังเห็นได้จากรูปที่ 3.2 ดังนั้นในการนำอุปกรณ์ดังกล่าวมาใช้งาน จำเป็นต้องมีอุปกรณ์สำหรับจ่ายพลังงาน อุปกรณ์สำหรับการบรรจุโปรแกรม และอุปกรณ์อื่นๆ ขึ้นมาเอง เพื่อรองรับการใช้งาน ซึ่งจะได้นำเสนอในบทของการออกแบบส่วนฮาร์ดแวร์ และส่วนซอฟต์แวร์

การต่อพ่วงอุปกรณ์ต่างๆ เข้ากับอุปกรณ์อาร์ซีเอ็มนี้สามารถเชื่อมต่อเข้ากับตัวเชื่อมต่อ (Connector) ซึ่งอยู่ด้านล่างของอุปกรณ์อาร์ซีเอ็ม โดยมีด้วยกัน 2 ชุด คือ J1 และ J2 สามารถสังเกตได้โดยตัวเชื่อมต่อ J1 นั้นอยู่ด้านล่างฝั่งที่ใกล้กับหน่วยประมวลผล (Rabbit 2000) มากกว่า J2 โดยขาสัญญาณต่างๆ ด้านล่างของอุปกรณ์อาร์ซีเอ็มแสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 ขาสัญญาณตัวเชื่อมต่อของอุปกรณ์อาร์ซีเอ็ม

บทที่ 4
การพัฒนาส่วนฮาร์ดแวร์ และส่วนซอฟต์แวร์

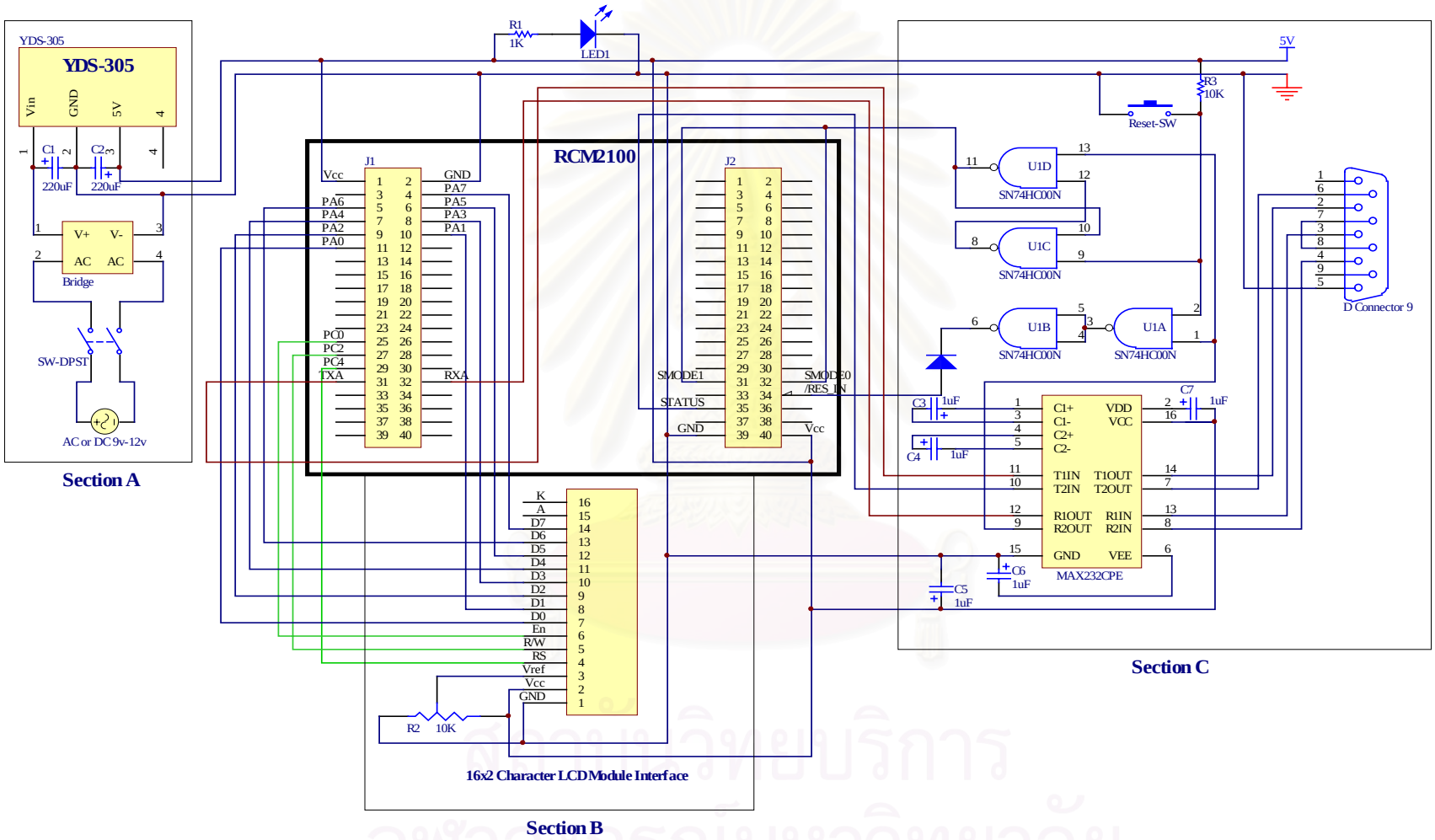
4.1 การพัฒนาส่วนฮาร์ดแวร์

เนื่องจากอุปกรณ์อาร์ซีเอ็มรุ่น 2100 ที่ได้เลือกใช้ในงานวิจัยนี้ ไม่สามารถปฏิบัติงานได้โดยลำพัง ดังนั้นต้องเพิ่มอุปกรณ์ต่อพ่วงเพื่อให้ทำหน้าที่ตามที่ต้องการได้ โดยอุปกรณ์อาร์ซีเอ็มพร้อมทั้งอุปกรณ์อื่นๆ ประกอบอยู่บนแผงวงจร ซึ่งแบ่งส่วนของวงจรออกเป็น 3 ส่วน คือ ส่วนจ่ายไฟ ส่วนต่อประสานการเขียนโปรแกรม และส่วนแสดงผล ซึ่งวงจรโดยรวมทั้งหมดแสดงได้ดังรูป 4.1

4.1.1 ส่วนจ่ายไฟ (Power Supply Part)

ถึงแม้อุปกรณ์อาร์ซีเอ็มรุ่น 2100 จะใช้พลังงานไฟฟ้าน้อยมาก คือใช้กระแสไฟฟ้า 140 มิลลิแอมแปร์ (mA) ที่แรงดันไฟฟ้า 5 โวลต์ (volt) แต่เนื่องจากส่วนจ่ายไฟนี้เป็นส่วนจ่ายไฟเดียวของทั้งวงจร ซึ่งยังต้องมีอุปกรณ์อื่นๆ ที่ต้องทำงานร่วมกัน พร้อมทั้งต้องการให้วงจรทำงานที่แรงดันไฟฟ้าที่คงที่อยู่เสมอ ดังนั้นจึงได้เลือกใช้ตัวคุมค่าแรงดันไฟฟ้า (voltage regulator) ซึ่งเป็นชนิดสวิทซ์ชิ่ง (switching) ของบริษัท YEC รุ่น YDS-305 โดยรองรับการใช้กระแสไฟฟ้าถึง 3 แอมแปร์ ซึ่งเพียงพอสำหรับอุปกรณ์ที่อาจนำมาต่อเพิ่มได้ในภายหลัง

จากรูป 4.1 การต่อวงจรส่วนจ่ายไฟให้กับอุปกรณ์อาร์ซีเอ็มนั้น มีวงจรสำหรับการต่อเพิ่มอยู่ในส่วนเอ (Section A) โดยเริ่มจากการจ่ายไฟซึ่งอาจเป็นได้ทั้งไฟฟ้ากระแสสลับ หรือไฟฟ้ากระแสตรง โดยแรงดันไฟฟ้าที่แนะนำคือ 9 ถึง 12 โวลต์ ไฟฟ้าจะผ่านไปยังสวิทซ์ และบริดจ์ (Bridge) หลังจากนั้นตัวคุมค่าแรงดันไฟฟ้าจะแปลงไฟฟ้าที่ได้ให้อยู่ในระดับแรงดัน 5 โวลต์คงที่เพื่อจ่ายไปให้กับอุปกรณ์อาร์ซีเอ็ม และอุปกรณ์อื่นๆ



รูปที่ 4.1 แสดงวงจรของอุปกรณ์ทั้งหมด

4.1.2 ส่วนแสดงผล (Display Part)

การแสดงผลของอุปกรณ์อาร์ซีเอ็มจำเป็นสำหรับการตรวจสอบการทำงานของอุปกรณ์ ซึ่งส่วนแสดงผลต้องสามารถบ่งบอกถึงเลขที่อยู่ไอพีของอุปกรณ์อาร์ซีเอ็ม สถานะการทำงานของอุปกรณ์ เลขที่อยู่ไอพีของเครื่องที่ติดต่อเข้ามา รวมไปถึงข้อมูลที่จำเป็นสำหรับผู้ดูแลระบบเครือข่าย ดังนั้นจึงได้เลือกส่วนแสดงผลเป็นแอลซีดีชนิดตัวอักษร (character LCD module) ซึ่งเพียงพอต่อการแสดงผลข้อมูลดังที่กล่าวมา โดยสามารถแสดงตัวอักษรได้บรรทัดละ 16 ตัวอักษร 2 บรรทัด (16x2) การทำงานของส่วนแสดงผลแอลซีดีนี้เป็นไปตามมาตรฐาน HD44780 ของบริษัทฮิตาชิ (Hitachi) โดยการต่อวงจรแสดงอยู่ในส่วนบี (Section B) ของรูป 4.1

ส่วนของแอลซีดีนั้นรับส่งข้อมูลผ่านช่องทางข้อมูลอนุกรมเอ (Parallel port A) ซึ่งอยู่บนตัวเชื่อมต่อ J1 สามารถนำแอลซีดีมาตรฐานมาต่อกับส่วนต่อประสานอุปกรณ์แอลซีดีที่ได้ออกแบบไว้ (LCD Module Interface) โดยขาสัญญาณต่างๆ ของแอลซีดีมาตรฐานนั้น ตรงกับขาสัญญาณของส่วนต่อประสานอุปกรณ์แอลซีดี ซึ่งขาสัญญาณเหล่านี้มีหน้าที่ดังตารางที่ 4.1

ตารางที่ 4.1 ขาสัญญาณของอุปกรณ์แอลซีดี

ขาสัญญาณ	ชื่อสัญญาณ	หน้าที่	หมายเหตุ
1	GND	สัญญาณกราวนด์	
2	Vcc	สัญญาณไฟขับวงจร 5 โวลต์	
3	Vref	แรงดันไฟฟ้าอ้างอิงสำหรับความเข้มของจอแอลซีดี	0 – 5 โวลต์
4	RS (Register Select)	สัญญาณเลือกข้อมูล D0-D7 ให้เป็นชุดคำสั่ง หรือเป็นข้อมูล	ชุดคำสั่ง = 0 ข้อมูล = 1
5	R/W	สัญญาณสำหรับอ่านหรือเขียน	R = 1 / W = 0
6	En	สัญญาณสั่งเปิดการทำงาน	
7 - 14	D0 – D7	สัญญาณข้อมูล หรือคำสั่ง	ขนาด 8 บิต
15	A	สัญญาณไฟขับวงจรของไฟส่องหลัง (back light)	ไม่ใช้
16	K	สัญญาณกราวนด์ของไฟส่องหลัง	ไม่ใช้

4.1.3 ส่วนต่อประสานการเขียนโปรแกรม (Programming Interface Part)

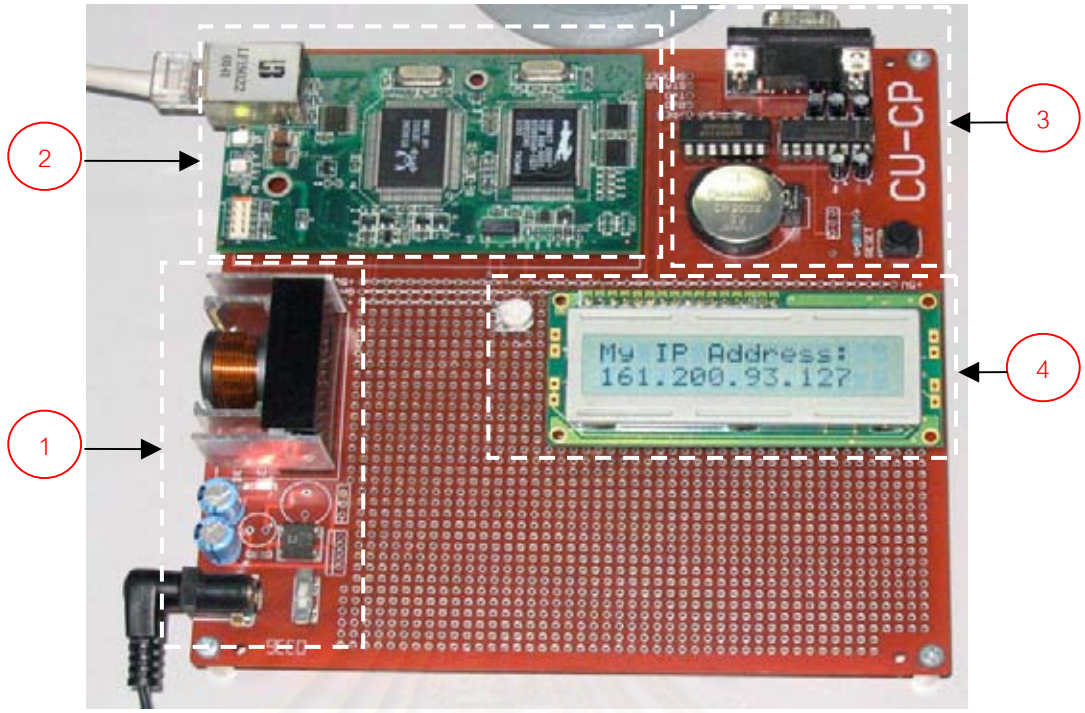
ส่วนต่อประสานการเขียนโปรแกรมสำหรับอุปกรณ์อาร์ซีเอ็มนั้น ติดต่อสื่อสารกับเครื่องคอมพิวเตอร์โดยผ่านทางช่องทางข้อมูลอนุกรม (serial port) โดยรับส่งข้อมูลที่อัตราเร็วสูงสุด 115200 บิตต่อวินาที ผ่านทางมาตรฐานการรับส่งข้อมูลชนิดอาร์เอส-232 (RS-232) จากรูปที่ 4.1 วงจรส่วนต่อประสานการเขียนโปรแกรมอยู่ในส่วนซี (Section C) ของวงจร

การรีเซท (Reset) ใช้การทำงานของสวิตช์ โดยทุกครั้งที่มีการกดสวิตช์ทำให้สัญญาณของขา RES_IN มีแรงดันไฟฟ้าในสถานะต่ำ (low) ส่วนการแปลงสัญญาณระหว่างชนิดทีทีแอล และอาร์เอส-232 นั้น อาศัยการทำงานของ MAX232CPE โดยข้อมูลต่างๆ ถูกส่งผ่านตัวเชื่อมต่อ J1 นอกจากนั้นยังแสดงให้เห็นถึงการกำหนดวิธีการเริ่มต้นการทำงานของอุปกรณ์อาร์ซีเอ็ม โดยการต่อวงจรเข้ากับ SN74HC00N ซึ่งมี NAND Gate อยู่ภายใน 4 ตัวด้วยกัน ซึ่งกำหนดค่าของขาสัญญาณ SMODE0 และ SMODE1 ของอุปกรณ์อาร์ซีเอ็ม ค่าของสัญญาณทั้ง 2 ขานั้นสามารถเปลี่ยนวิธีการเริ่มต้นการทำงาน โดยแสดงได้ดังตารางที่ 4.2

ตารางที่ 4.2 วิธีการเริ่มต้นการทำงาน

SMODE0	SMODE1	**วิธีการเริ่มต้นการทำงาน**
0	0	เริ่มต้นการทำงานกับคำสั่งที่ตำแหน่งที่อยู่ (Address) เป็น 0
0	1	เริ่มต้นการทำงานด้วยช่องทางลูกข่าย (Slave port)
1	0	เริ่มต้นการทำงานด้วยสัญญาณนาฬิกาทางช่องทางข้อมูลอนุกรมเอ (Serial port A)
1	1	เริ่มต้นการทำงานด้วยช่องทางข้อมูลอสมวารเอ (Asynchronous serial port A)

อุปกรณ์ที่เสร็จสมบูรณ์แสดงได้ดังรูปที่ 4.2 ซึ่งประกอบไปด้วยส่วนประกอบ 4 ส่วนด้วยกัน คือ ส่วนที่ 1 เป็นวงจรส่วนจ่ายไฟ ส่วนที่ 2 เป็นอุปกรณ์อาร์ซีเอ็มที่นำมาใช้กับระบบ ส่วนที่ 3 เป็นวงจรส่วนต่อประสานการเขียนโปรแกรม และส่วนสุดท้ายเป็นส่วนแสดงผล

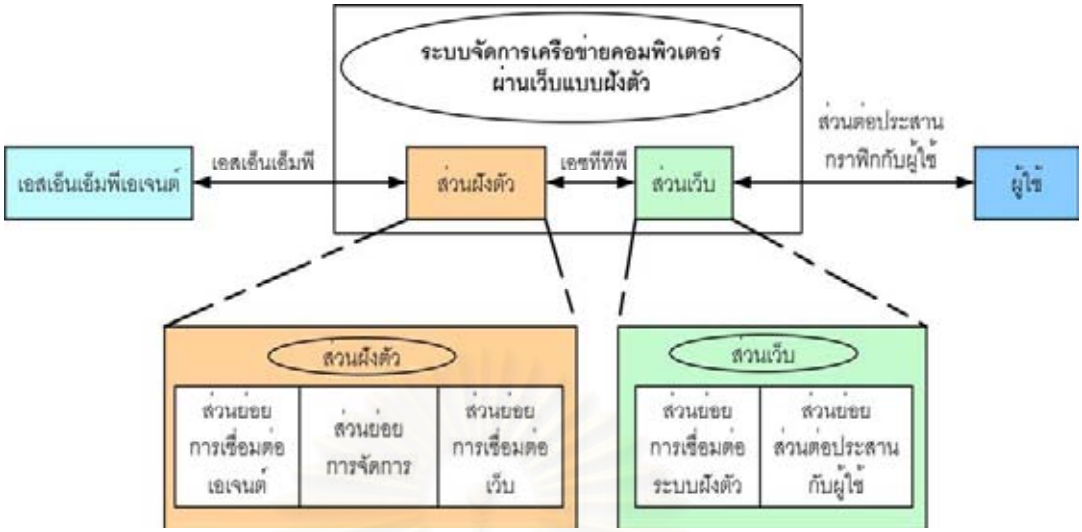


รูปที่ 4.2 แสดงอุปกรณ์ที่เสร็จสมบูรณ์

4.2 การพัฒนาส่วนซอฟต์แวร์

การพัฒนาซอฟต์แวร์สำหรับงานวิจัยนี้ จำเป็นต้องใช้ภาษาในการพัฒนาโปรแกรมให้เหมาะสมกับหน้าที่การทำงานของแต่ละส่วน ดังนั้น จึงได้ใช้ภาษาโปรแกรม 3 ภาษา คือ ภาษาซี (C) เหมาะสำหรับการทำงานด้านการจัดการเครือข่าย และการประมวลผลข้อมูลภายในสำหรับอุปกรณ์อาร์ซีเอ็ม ภาษาแอสเซมบลี (Assembly) เหมาะสำหรับคำสั่งที่ติดต่อกับส่วนของฮาร์ดแวร์โดยตรง และภาษาจาวา (Java) เหมาะสำหรับการติดต่อกับผู้ใช้ผ่านทางเว็บบราวเซอร์ ซึ่งโครงสร้างของซอฟต์แวร์สำหรับงานวิจัยนี้ได้พัฒนาขึ้นจากที่โครงสร้างที่เคยนำเสนอไว้ใน [16] สามารถสรุปรวมการทำงานทั้งหมดโดยแสดงเป็นแบบจำลองการทำงานดังรูป 4.3 ซึ่งแบ่งการทำงานหลักออกได้ 2 ส่วน คือ ส่วนฝังตัว (Embedded part) ซึ่งมีหน้าที่จัดการเอสเอ็นเอ็มพีเอเจนต์โดยตรง และส่วนเว็บ (Web part) ซึ่งเปรียบเสมือนส่วนติดต่อกับผู้ใช้ (user)

การจัดการด้วยส่วนจัดการนี้ประกอบด้วยวิธีในการจัดการ 2 วิธีด้วยกันคือ วิธีการจัดการด้วยมือ (Manual mode) และวิธีการจัดการแบบอัตโนมัติ (Automatic mode) โดยการจัดการทั้ง 2 วิธีนั้น ผู้ดูแลระบบสามารถควบคุมได้จากเว็บบราวเซอร์ ซึ่งเว็บบราวเซอร์จะส่งการควบคุมมายังอุปกรณ์อาร์ซีเอ็มโดยผ่านทางเอชทีทีพีซึ่งมีรูปแบบพิเศษ ดังจะได้กล่าวถึงในหัวข้อ 4.2.3



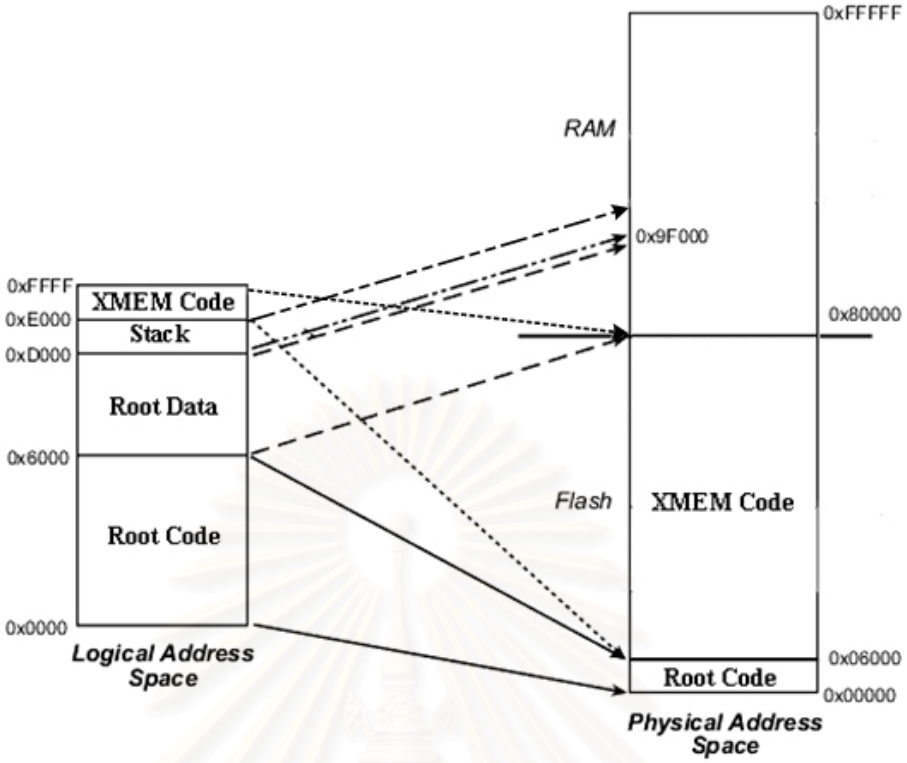
รูปที่ 4.3 แบบจำลองการทำงานของซอฟต์แวร์

แบบจำลองการทำงานของซอฟต์แวร์สามารถแบ่งออกได้เป็น 2 ส่วนคือ ส่วนฝังตัวและส่วนเว็บ โดยแสดงรายละเอียดแต่ละส่วนได้ดังนี้

4.2.1 ส่วนฝังตัว (embedded part)

ซอฟต์แวร์ภายในสำหรับส่วนฝังตัวนี้ประกอบไปด้วยภาษาซี และภาษาแอสเซมบลี ซึ่งการทำงานทั้งหมดเกิดขึ้นภายในอุปกรณ์ฝังตัว โดยสามารถแบ่งการทำงานออกตามหน้าที่การทำงานได้อีก 3 ส่วนย่อย คือ ส่วนย่อยการเชื่อมต่อเอเจนต์ ส่วนย่อยการจัดการ และส่วนย่อยการเชื่อมต่อเว็บ โดยไลบรารี (Library) และฟังก์ชันต่างๆ แสดงไว้ในภาคผนวก ก

สิ่งที่ต้องคำนึงถึงก่อนการออกแบบซอฟต์แวร์ภายในส่วนนี้คือ เลขที่อยู่ของหน่วยความจำทั้งหมดที่ระบบรับรู้ได้นั้นเป็นเลขที่อยู่เชิงกายภาพ (Physical address) ซึ่งมีขนาด 20 บิต อ้างอิงได้ถึง 1 เมกกะไบต์ แต่ระหว่างขั้นตอนการทำงานนั้นชุดคำสั่งสำหรับการประมวลผลข้อมูลสามารถอ้างอิงได้กับเลขที่อยู่เชิงตรรกะ (Logical address) เท่านั้น ไม่สามารถนำเลขที่อยู่เชิงกายภาพมาใช้ในการประมวลผลข้อมูลได้โดยตรง ต้องใช้การย้ายข้อมูลจากตำแหน่งเชิงกายภาพให้อยู่ภายในหน่วยความจำที่อ้างอิงโดยเลขที่อยู่เชิงตรรกะ ซึ่งเลขที่อยู่เชิงตรรกะนั้นมีขนาด 16 บิต อ้างอิงได้เพียง 64 กิโลไบต์ ทำให้ระบบไม่สามารถนำข้อมูลที่มีขนาดใหญ่มาประมวลผลได้พร้อมๆ กัน ดังนั้นการประกาศตัวแปรจึงต้องใช้หน่วยความจำให้ประหยัดและมีประสิทธิภาพที่สุด โดยสามารถจัดการการใช้หน่วยความจำได้หลายวิธี [17] และเพื่อให้เข้าใจถึงความจำเป็นในการใช้ทรัพยากรอย่างประหยัด สามารถแสดงความสัมพันธ์ระหว่างเลขที่อยู่เชิงตรรกะ และเลขที่อยู่เชิงกายภาพได้ดังรูปที่ 4.4



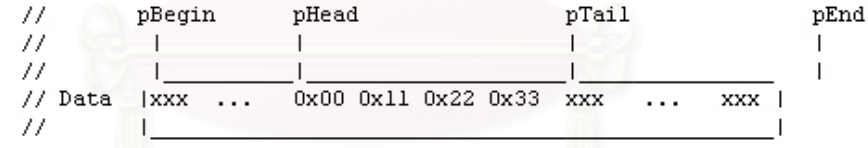
รูปที่ 4.4 แสดงความสัมพันธ์ระหว่างเลขที่อยู่เชิงตรรกะ และเลขที่อยู่เชิงกายภาพ

จากรูปส่วนของโปรแกรมสามารถเก็บอยู่ได้สองส่วนคือ รหัสคำสั่งราก (Root code) และ รหัสคำสั่งหน่วยความจำส่วนเพิ่ม (XMEM code) ซึ่งส่วนของโปรแกรมนี้จัดเก็บอยู่ในแฟลชเพื่อให้โปรแกรมยังคงอยู่ในขณะที่ไม่มีไฟขับวงจร ส่วนของข้อมูลนั้นสามารถเก็บได้ทั้งในข้อมูลราก (Root Data) และ ในหน่วยความจำส่วนเพิ่ม (Extended memory) แต่ระหว่างการทำงานข้อมูลที่นำมาประมวลผลต้องอยู่ภายในข้อมูลรากเท่านั้นจึงจะสามารถนำมาประมวลผลได้ ดังนั้นภายในข้อมูลรากจึงควรเก็บเฉพาะข้อมูลที่มีขนาดเล็ก หรือจำเป็นต้องเรียกใช้บ่อย หากข้อมูลที่ต้องการเก็บมีขนาดใหญ่เกินกว่าที่ข้อมูลรากเก็บได้ ต้องเก็บข้อมูลไว้ในหน่วยความจำส่วนเพิ่ม โดยหากต้องการนำข้อมูลภายในหน่วยความจำส่วนเพิ่มมาประมวลผล สามารถใช้คำสั่งในการแลกเปลี่ยนข้อมูลระหว่างข้อมูลหน่วยความจำส่วนเพิ่ม และข้อมูลราก จากนั้นเมื่อประมวลผลข้อมูลภายในข้อมูลรากเสร็จสิ้น จึงนำข้อมูลภายในข้อมูลรากกลับไปเก็บในหน่วยความจำส่วนเพิ่มเพื่อคืนเนื้อที่ภายในข้อมูลรากไว้สำหรับการประมวลผลข้อมูลอื่นต่อไป การทำงานทั้งหมดของส่วนฝังตัวนี้อาศัยโครงสร้างข้อมูลที่สำคัญ 2 ส่วนด้วยกัน ดังนี้

- my_buff สำหรับเก็บข้อมูลที่ต้องการรับหรือส่งออกจากส่วนฝังตัว ซึ่งเปรียบเสมือนเป็นบัฟเฟอร์ (Buffer) พร้อมด้วยตัวชี้ (Pointer) โดยขอบเขตของบัฟเฟอร์ที่สามารถเก็บข้อมูลได้อยู่ระหว่าง pBegin และ pEnd ส่วนขอบเขตของข้อมูลที่มีอยู่จริงนั้นอยู่

ระหว่าง pHead และ pTail จะสังเกตได้ว่าข้อมูลไม่จำเป็นต้องเริ่มที่ตำแหน่งต้นบัฟเฟอร์เสมอไป เพราะข้อมูลเอสเอ็นเอ็มพีต้องมีการใส่ขนาดของข้อมูลนำหน้าส่วนของข้อมูล จึงจำเป็นต้องสงวนเนื้อที่ไว้ในตอนต้นของข้อมูลด้วย และเนื่องจากภายในโครงสร้างข้อมูลชนิดนี้ประกอบไปบัฟเฟอร์ของข้อมูลซึ่งมีขนาดเท่ากับค่าคงที่ SNMP_MAX_LENGTH โดยได้ประกาศไว้เท่ากับ 512 ไบต์ ดังนั้นเมื่อมีการประกาศตัวแปรชนิด my_buff มากขึ้น ย่อมส่งผลให้หน่วยความจำข้อมูลรากเต็มอย่างรวดเร็ว จึงไม่สามารถสร้างตัวแปรประเภทนี้ให้อยู่ในหน่วยความจำรากได้ตลอดเวลา ต้องอาศัยการพักข้อมูลไว้ในหน่วยความจำส่วนเพิ่ม (Extended Memory) เป็นเหตุให้ต้องมีการกำหนดเขตข้อมูล (field) เพิ่มอีก 2 เขตข้อมูล โดยเป็นเขตข้อมูลที่ลงท้ายด้วยคำว่า “Off” โครงสร้างข้อมูลชนิดนี้ถูกเรียกใช้โดยโครงสร้างข้อมูลชนิด Box เพียงอย่างเดียวเท่านั้น ดังนั้นการเรียกใช้งานสำหรับการจัดการข้อมูลภายใน my_buff จึงเป็นคำสั่งจัดการสำหรับโครงสร้างข้อมูล Box ซึ่งจะได้กล่าวถึงต่อไป สำหรับโครงสร้างข้อมูล my_buff สามารถแสดงได้ดังรูป 4.5

```
typedef struct
{  char *pBegin, *pEnd;       // Border of buffer
   char *pHead, *pTail;       // Data bound in buffer
   int  pHeadOff, pTailOff;   // Offset from Data[] (for storing to XMEM)
   char Data[SNMP_MAX_LENGTH];
} my_buff;
```



รูปที่ 4.5 แสดงโปรแกรมส่วนการประกาศโครงสร้างข้อมูลชนิด my_buff

- Box โครงสร้างข้อมูลชนิดนี้มีไว้สำหรับเก็บข้อมูลสำคัญของการติดต่อจากผู้ใช้ ซึ่งผู้ใช้แต่ละคนเมื่อทำการติดต่อเข้าถึงระบบได้แล้ว ระบบจะสร้างตัวแปรข้อมูลชนิดนี้ให้กับผู้ใช้ นอกจากนี้แล้วโครงสร้างข้อมูลชนิดนี้ยังจำเป็นสำหรับการแจ้งเตือนการเกิดเหตุการณ์ และการตรวจพบข้อมูลเอสเอ็นเอ็มพีแทรพด้วย โดยโครงสร้างข้อมูลสามารถแสดงได้ดังรูป 4.6

```
typedef struct
{  tcp_Socket     *ptcp;   // pointer to TCP socket (Applet)
   udp_Socket     *pudp;   // pointer to UDP socket (SNMP Agent)
   int            num;     // box number;
   char           clientaddr[16];
   unsigned long  ver;
   char           comm[SNMP_MAX_COMMNAME]; // Community
   char           oid[SNMP_MAX_OIDLEN];
   unsigned char  type;
   char           val[SNMP_MAX_OIDLEN];
```

รูปที่ 4.6 แสดงโปรแกรมในส่วนการประกาศโครงสร้างข้อมูลชนิด Box

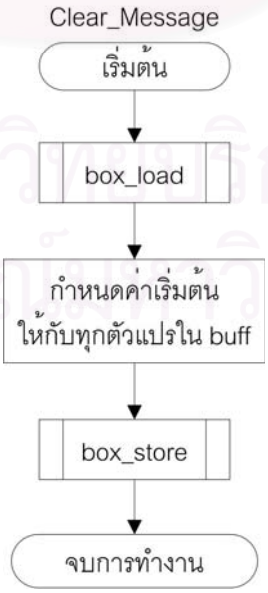
จากการประกาศโครงสร้างข้อมูลตามรูป 4.6 แสดงให้เห็นถึงการประกาศตัวแปร buff โดยเป็นโครงสร้างข้อมูลชนิด my_buff ซึ่งมีขนาดใหญ่ดังที่กล่าวมาแล้ว ดังนั้นตัวแปรใดก็ตามที่เป็นชนิดข้อมูลชนิดนี้จึงเหมาะสำหรับการเก็บภายในหน่วยความจำส่วนเพิ่ม โดยมีคำสั่งสำหรับการแลกเปลี่ยนข้อมูลระหว่างข้อมูลราก และหน่วยความจำส่วนเพิ่มดังรูปที่ 4.7 ซึ่งฟังก์ชันสำหรับการนำข้อมูลจากหน่วยความจำส่วนเพิ่มเข้ามาอยู่ภายในข้อมูลรากคือ box_load และฟังก์ชันสำหรับการนำข้อมูลภายในข้อมูลรากไปเก็บไว้ในหน่วยความจำส่วนเพิ่มคือ box_store

```
_mybox_nodebug void box_load(int boxnum, char *pdata) // load box data from XMEM to pdata
{  auto Box *pbox;

   xmem2root(pdata, boxes_longaddr+(boxnum*MYBOX_BOXSIZE), MYBOX_BOXSIZE);
   pbox = (Box *)pdata;
   pbox->buff.pBegin = pbox->buff.Data;
   pbox->buff.pEnd = pbox->buff.pBegin + sizeof(pbox->buff.Data);
   pbox->buff.pHead = pbox->buff.Data + pbox->buff.pHeadOff;
   pbox->buff.pTail = pbox->buff.Data + pbox->buff.pTailOff;
}

_mybox_nodebug void box_store(int boxnum, char *pdata) // store box data from pdata to XMEM
{  auto Box *pbox;

   pbox = (Box *)pdata;
   pbox->buff.pHeadOff = pbox->buff.pHead - pbox->buff.Data;
   pbox->buff.pTailOff = pbox->buff.pTail - pbox->buff.Data;
   root2xmem(boxes_longaddr+(boxnum*MYBOX_BOXSIZE), pdata, MYBOX_BOXSIZE);
}
```

รูปที่ 4.7 แสดงโปรแกรมย่อยสำหรับการแลกเปลี่ยนข้อมูลชนิด Box ระหว่างข้อมูลราก และหน่วยความจำส่วนเพิ่ม

การพัฒนาได้สร้างฟังก์ชันสำหรับจัดการกับข้อมูลภายในตัวแปร buff ซึ่งอยู่ในโครงสร้างข้อมูลชนิด Box ไว้ 4 ฟังก์ชันด้วยกันดังนี้

- Clear_Message เป็นฟังก์ชันซึ่งทำหน้าที่ลบข้อมูลออกจากส่วนเก็บข้อมูล buff ภายในโครงสร้างข้อมูลชนิด Box โดยผังงานของฟังก์ชันนี้แสดงได้ตามรูปที่ 4.8

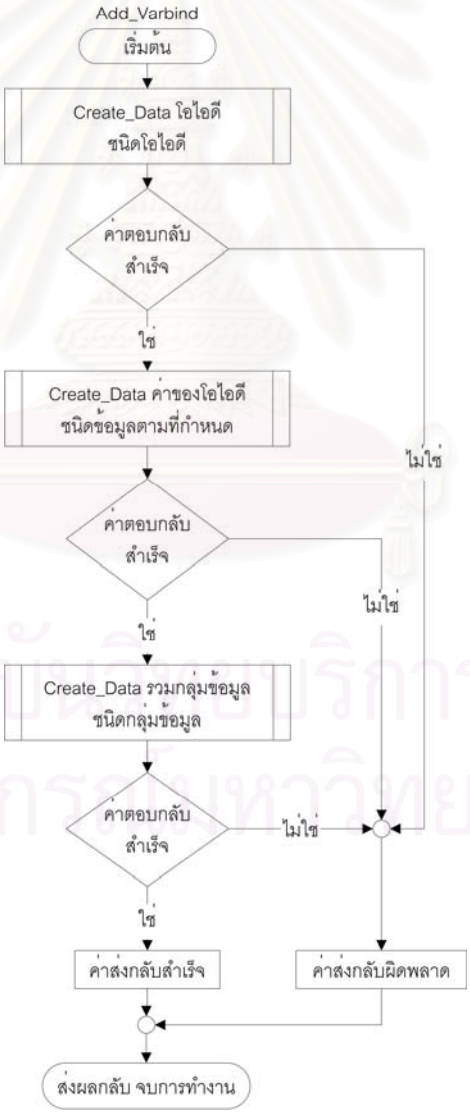


รูปที่ 4.8 ผังงานของฟังก์ชัน Clear_Message

- Create_Data ฟังก์ชันนี้ทำหน้าที่สร้างกลุ่มของข้อมูล 1 ชุดตามรูปแบบบีอีอาร์ (BER: Basic Encoding Rules) ที่ถูกใช้เป็นข้อมูลพื้นฐานของเอสเอ็นเอ็มพีโดยแบ่งเป็น 3 ส่วนคือ ชนิดของข้อมูลขนาด 1 ไบต์ ตามด้วยขนาดความยาวของข้อมูล (length) ซึ่งมีขนาดเท่าใดก็ได้ และสุดท้ายเป็นข้อมูลจริงโดยมีขนาดความยาวเท่ากับความยาวของข้อมูลที่ระบุ อ่านข้อมูลบีอีอาร์เพิ่มเติมได้จากภาคผนวก ข ฟังก์ชันนี้สามารถสร้างกลุ่มของข้อมูลแบบบีอีอาร์ได้ทั้งในส่วนหัวและส่วนข้อมูลของเอสเอ็นเอ็มพี ซึ่งผู้พัฒนาสามารถกำหนดวิธีการเก็บข้อมูลได้ 2 วิธีคือต่อท้ายข้อมูลเดิม หรือแทรกข้อมูลก่อนหน้าข้อมูลเดิมได้วิธีการเรียกใช้ ผู้พัฒนาต้องกำหนดชนิดของข้อมูล ข้อมูล และวิธีการเก็บข้อมูล จากนั้นฟังก์ชันนี้จะรับผิดชอบสร้างชุดข้อมูลบีอีอาร์ให้ 1 ชุดเพื่อเพิ่มข้อมูลนำไปเก็บไว้รวมกับข้อมูลเดิมภายในตัวแปร buff ของโครงสร้างข้อมูลชนิด Box สามารถแสดงผังงานได้ตามรูปที่ 4.9

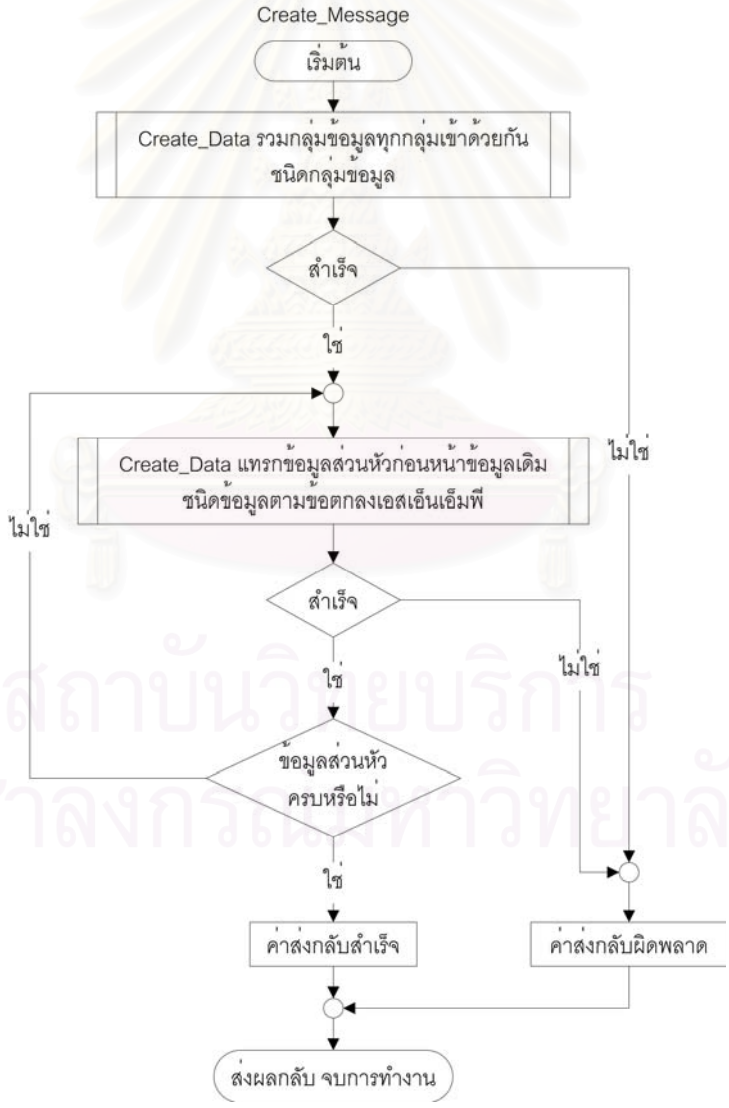
รูปที่ 4.9 ผังงานของฟังก์ชัน Create_Data

- Add_Varbind เป็นฟังก์ชันสำหรับการเพิ่มค่าอ็อบเจกต์ให้กับการกำหนดค่าหรือการร้องขอค่าอ็อบเจกต์จากเอสเอ็นเอ็มพีเอเจนต์ โดยฟังก์ชันจะทำหน้าที่สร้างชุดข้อมูลบีอีอาร์ 2 ชุดเพื่อต่อท้ายข้อมูลเดิม โดยประกอบด้วยโอไอดีและค่าของโอไอดีนั้น จากนั้นรวมชุดข้อมูลทั้ง 2 ชุดนี้ให้กลายเป็นชุดข้อมูลใหม่ตามข้อตกลงในการสร้างชุดค่าของอ็อบเจกต์เอสเอ็นเอ็มพี สำหรับฟังก์ชันนี้ใช้เฉพาะการสร้างส่วนข้อมูลของเอสเอ็นเอ็มพีเท่านั้น ซึ่งหากมีข้อมูลอ็อบเจกต์หลายอ็อบเจกต์ต้องเรียกใช้ฟังก์ชันนี้สำหรับทุกๆ ชุดของอ็อบเจกต์ สำหรับการสร้างส่วนหัวของเอสเอ็นเอ็มพีนั้นต้องเรียกใช้ฟังก์ชัน Create_Message ซึ่งจะได้กล่าวถึงต่อไป ผังงานแสดงได้ดังรูปที่ 4.10



รูปที่ 4.10 ผังงานของฟังก์ชัน Add_Varbind

- Create_Message เป็นฟังก์ชันสำหรับการสร้างส่วนหัวของเอสเอ็นเอ็มพี โดยก่อนการเรียกใช้ฟังก์ชันนี้ผู้พัฒนาควรสร้างส่วนของข้อมูลอ็อบเจกต์ต่างๆ ให้เรียบร้อยเสียก่อน จากนั้นฟังก์ชันนี้จะทำการสร้างส่วนหัวของเอสเอ็นเอ็มพีเพิ่มให้กับข้อมูลดังกล่าว (หลังจากการเรียกใช้ฟังก์ชันนี้แล้วไม่ควรเรียกใช้ฟังก์ชัน Add_Varbind หรือฟังก์ชัน Create_Message ซ้ำซ้อนอีก เพราะจะทำให้โครงสร้างข้อมูลเอสเอ็นเอ็มพีเสียหาย) ผู้พัฒนาสามารถศึกษาโครงสร้างของข้อมูลเอสเอ็นเอ็มพีได้จากภาคผนวก ค สำหรับฟังก์ชันนี้เมื่อเรียกใช้แล้วผู้พัฒนาสามารถส่งข้อมูลนั้นๆ ออกไปได้ทันทีผ่านยูดีพี การทำงานแสดงได้ดังผังงานตามรูปที่ 4.11



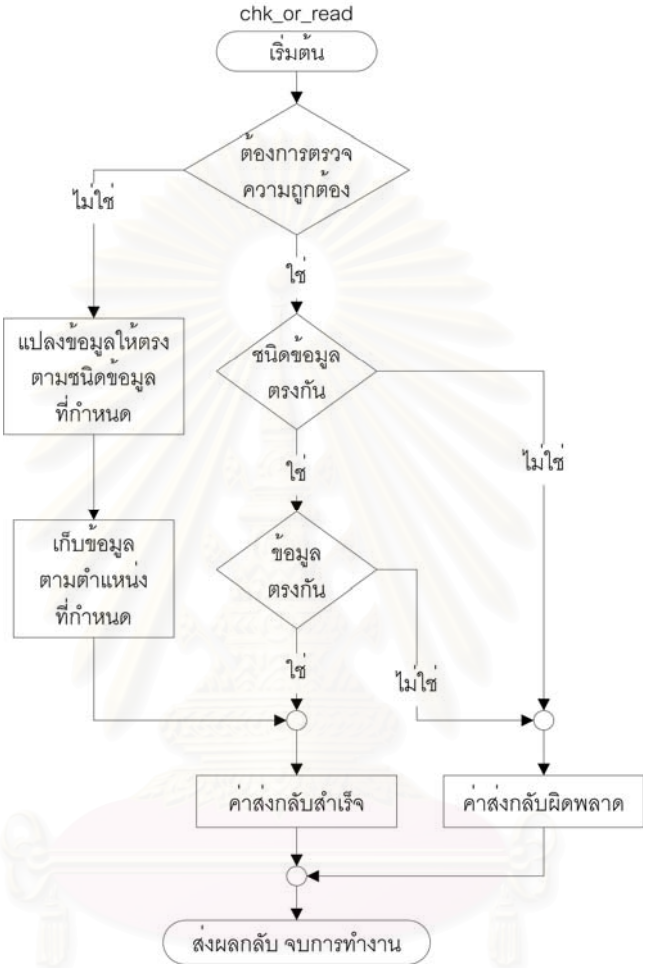
รูปที่ 4.11 ผังงานของฟังก์ชัน Create_Message

สำหรับการจัดการกับข้อมูลภายในตัวแปร buff ของโครงสร้างข้อมูล Box นั้น ผู้พัฒนาควรใช้ฟังก์ชันต่างๆ ตามที่กำหนดไว้เพื่อสะดวกต่อการพัฒนาและสามารถแก้ไขการทำงานได้จากจุดเดียว

4.2.1.1 ส่วนย่อยการเชื่อมต่อเอเจนต์ (Agent connection section)

ส่วนย่อยนี้มีหน้าที่สำหรับการจัดการเชื่อมต่อระหว่างส่วนฝังตัวและเอสเอ็นเอ็มพีเอเจนต์ ซึ่งภายในส่วนย่อยนี้รองรับการติดต่อจากส่วนอื่นโดยผ่านการเรียกใช้ฟังก์ชัน และทำการติดต่อไปยังเอสเอ็นเอ็มพีเอเจนต์ผ่านข้อตกลงเอสเอ็นเอ็มพี โดยกำหนดโครงสร้างข้อมูลให้ตรงตามข้อกำหนดของเอสเอ็นเอ็มพี [18] นอกจากนี้แล้วส่วนย่อยนี้ต้องสามารถรองรับเลขขนาดใหญ่ เนื่องจากภาษาซีที่ใช้พัฒนาในระบบฝังตัวนี้รองรับเลขขนาดความยาวไม่เกิน 4 ไบต์ดังนั้นจำเป็นต้องรองรับตัวเลขขนาดใหญ่ [19] จากเอสเอ็นเอ็มพีเอเจนต์ด้วย ซึ่งมีฟังก์ชันต่างๆ ทั้งหมด 6 ฟังก์ชันด้วยกัน ดังนี้

- chk_or_read สำหรับตรวจสอบความถูกต้องของชุดข้อมูลบีอีอาร์ 1 ชุด หรือแปลงค่าของข้อมูลชุดนั้นเพื่อเก็บเป็นข้อมูลตามชนิดข้อมูลที่ผู้ใช้กำหนดได้ด้วย สามารถแสดงผังการทำงานของฟังก์ชันนี้ได้ตามรูปที่ 4.12

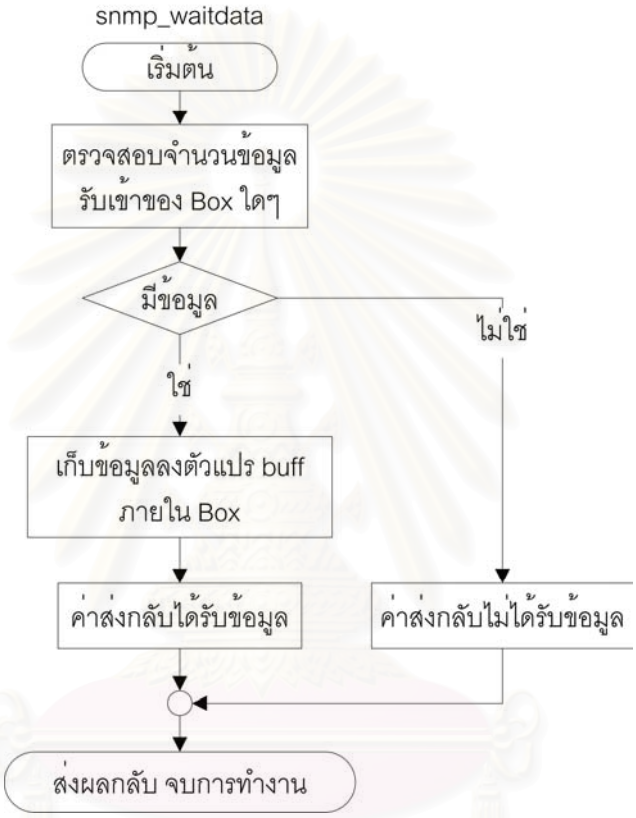


รูปที่ 4.12 ผังงานของฟังก์ชัน chk_and_read

- snmp_send สำหรับการจัดส่งข้อมูลเอสเอ็นเอ็มพีไปยังเอเจนต์ โดยการห่อหุ้มข้อมูลซึ่งบ่งบอกถึงหมายเลขโอไอดีของอ็อบเจกต์ต่างๆ ไว้พร้อมแล้ว จากนั้นนำค่าต่างๆ ที่เก็บอยู่ภายในข้อมูล Box มาสร้างเป็นข้อมูลเอสเอ็นเอ็มพี พร้อมทั้งจัดส่งผ่านยูดีพี โดยแสดงผังงานได้ดังรูป 4.13

รูปที่ 4.13 ผังงานของฟังก์ชัน snmp_send

- snmp_waitdata สำหรับการรอคอยข้อมูลตอบรับกลับ ซึ่งเป็นฟังก์ชันที่ไม่หยุดรอถึงแม้ว่ายังไม่มีข้อมูลมาถึงก็ตาม ดังนั้นจึงไม่เกิดปัญหาเรื่องโปรแกรมหยุดรอไม่สิ้นสุด เมื่อมีข้อมูลเข้ามาฟังก์ชันนี้จะทำหน้าที่คัดลอกข้อมูลเก็บไว้ในส่วนเก็บข้อมูลของตัวแปรประเภท Box ผังงานของฟังก์ชันนี้แสดงได้ตามรูปที่ 4.14

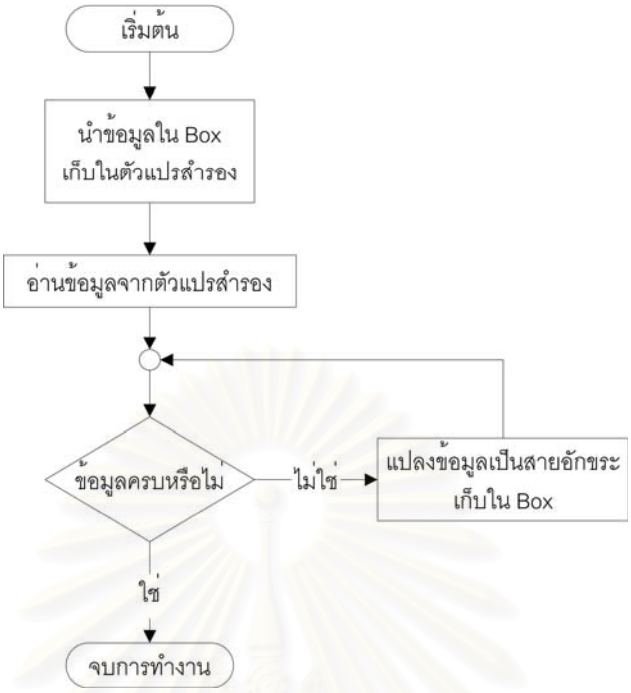


รูปที่ 4.14 ผังงานของฟังก์ชัน snmp_waitdata

- snmp_chkmessage สำหรับการตรวจสอบความถูกต้องของข้อมูลเอสเอ็นเอ็มพี ซึ่งฟังก์ชันนี้ทำหน้าที่ตรวจสอบความถูกต้องเฉพาะส่วนหัวของข้อมูลเอสเอ็นเอ็มพี ด้วยการนำค่าต่างๆ ซึ่งเก็บอยู่ภายในตัวแปรชนิด Box มาเปรียบเทียบกับค่าที่ได้รับกลับมาจากเอสเอ็นเอ็มพีเอเจนต์ ซึ่งหากค่าที่ได้รับกลับมาในส่วนหัวนั้นตรงกันทุกค่า การตรวจสอบจะได้รับค่าตอบกลับที่แสดงว่าสำเร็จ แต่หากมีค่าใดๆ แม้เพียงค่าเดียวที่ได้รับกลับมาไม่ตรงกับค่าที่เก็บไว้ภายในตัวแปรชนิด Box การตรวจสอบครั้งนั้นจะให้ผลที่แสดงว่าไม่สำเร็จ การทำงานของฟังก์ชันนี้แสดงได้ดังรูปที่ 4.15

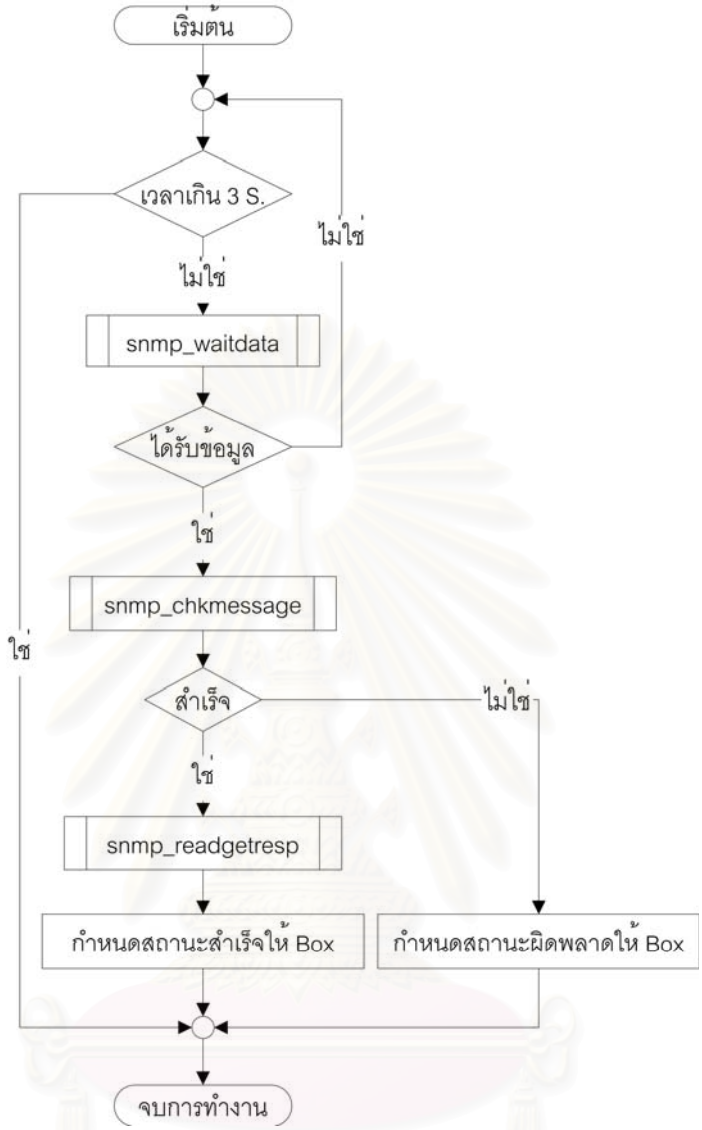
รูปที่ 4.15 ผังงานของฟังก์ชัน snmp_chkmessage

- snmp_readgetresp สำหรับแปลความหมายของข้อมูลที่เก็บอยู่ภายในตัวแปรประเภท Box ให้อยู่ในรูปแบบสายอักขระ (String) โดยนำไปเก็บแทนที่ข้อมูลเดิม การเรียกใช้ฟังก์ชันนี้ควรกระทำหลังจากเสร็จสิ้นการตรวจส่วนหัวของข้อมูลเอสเอ็นเอ็มพีจากฟังก์ชัน snmp_chkmessage เรียบร้อยแล้ว เพราะการแปลความหมายข้อมูลที่ถูกต้องควรแปลความหมายของข้อมูลที่มีส่วนหัวของข้อมูลถูกต้องเท่านั้น ดังนั้นหากการตรวจสอบส่วนหัวของข้อมูลไม่ถูกต้อง การแปลความหมายของข้อมูลก็ไม่จำเป็นต้องถูกเรียกใช้ สามารถแสดงผังงานของฟังก์ชันได้ตามรูปที่ 4.16



รูปที่ 4.16 ผังงานของฟังก์ชัน snmp_readgetresp

- recv_data เป็นฟังก์ชันสำหรับจัดการรับข้อมูลเพื่อให้ผู้ใช้สะดวกยิ่งขึ้น เพราะมีการจัดการตรวจสอบความถูกต้องของข้อมูลทั้งส่วนหัว และส่วนข้อมูลไว้ภายในฟังก์ชัน สามารถแสดงผังงานของฟังก์ชันนี้ได้ตามรูปที่ 4.17

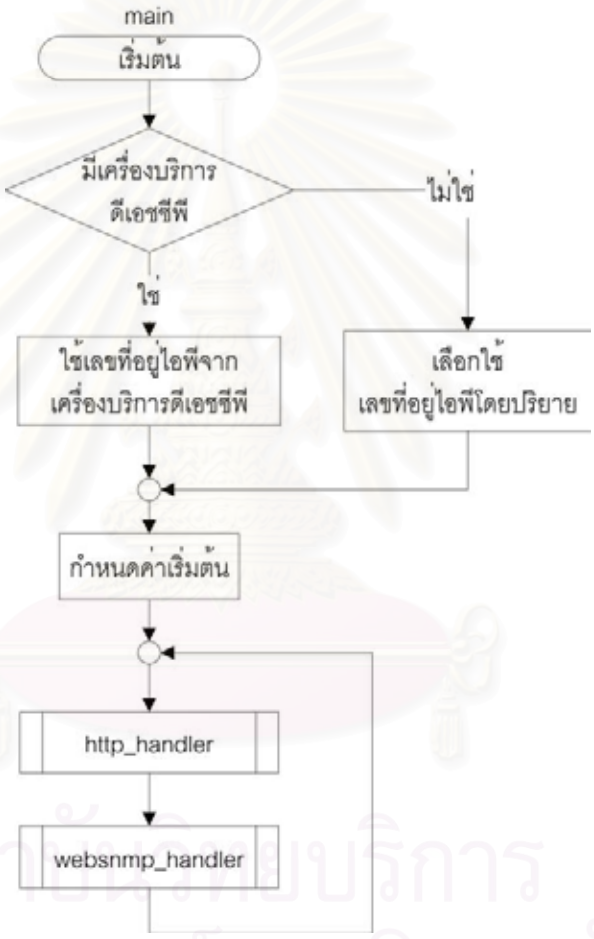


รูปที่ 4.17 ผังงานของฟังก์ชัน recv_data

4.2.1.2 ส่วนย่อยการจัดการ (Management section)

สำหรับส่วนย่อยนี้ เป็นส่วนย่อยหลักที่ทำให้การทำงานทุกอย่างเกิดขึ้น โดยการทำงานทั้งหมดจะเริ่มต้นที่ฟังก์ชัน main และมีการเรียกใช้งานฟังก์ชันอื่นๆ การสั่งงานจากผู้ใช้งานผ่านเว็บกระทำโดยผ่านซีจีไอ ดังนั้นเมื่อมีคำสั่งซีจีไอผ่านเข้ามา ส่วนที่จัดการเรื่องของซีจีไอซึ่งอยู่ภายในฟังก์ชัน http_handler จะคอยจัดส่งการทำงานให้กับฟังก์ชันที่ได้ลงทะเบียนไว้สำหรับจัดการซีจีไอ

- main เป็นฟังก์ชันหลักซึ่งเริ่มการทำงานด้วยการร้องขอเลขที่อยู่ไอพีจากเครื่องบริการดีเอชซีพี ซึ่งหากไม่มีเครื่องบริการดังกล่าว ระบบจะเลือกใช้เลขที่อยู่ไอพีโดยปริยาย (Default IP address) หลังจากนั้นระบบจะทำการกำหนดค่าเริ่มต้นต่างๆ ซึ่งรวมไปถึงการลงทะเบียนฟังก์ชันที่เปิดให้บริการผ่านเว็บด้วย สุดท้ายระบบจะสลับการทำงานระหว่างฟังก์ชัน http_handler และฟังก์ชัน websnmp_handler แสดงผังงานของฟังก์ชันได้ตามรูปที่ 4.18

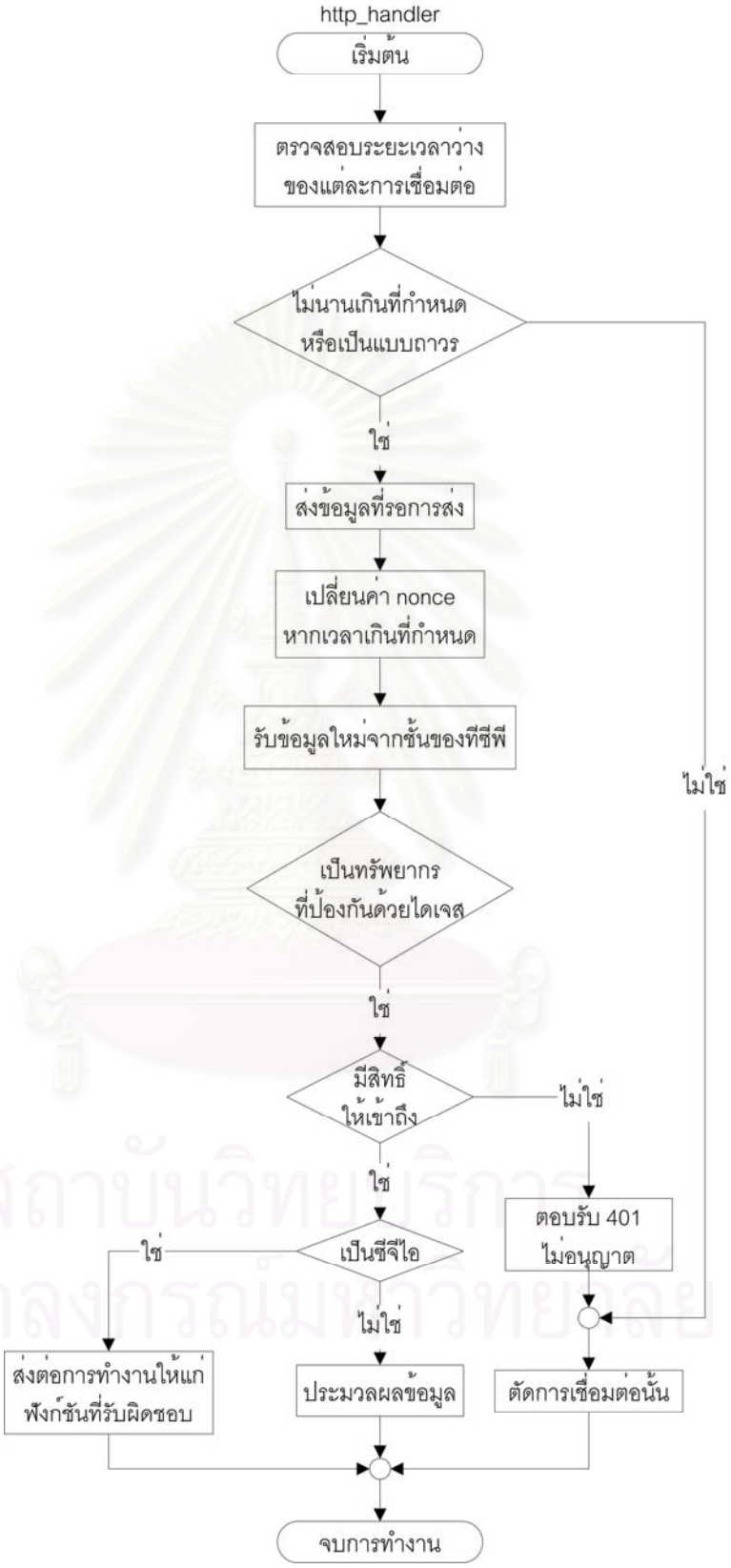


รูปที่ 4.18 ผังงานของฟังก์ชัน main

- http_handler เป็นฟังก์ชันซึ่งมีหน้าที่รับผิดชอบต่อการติดต่อมายังบริการเว็บโดยในทุกรอบการทำงาน ฟังก์ชันนี้จะทำหน้าที่ตรวจสอบขั้นตอนการทำงานให้กับทุกๆ การเชื่อมต่อ (Connection) หากการเชื่อมต่อใดๆ ที่อยู่ในขั้นตอนว่างเป็นเวลานาน และไม่ใช่การเชื่อมต่อแบบถาวรแล้วต้องตัดการเชื่อมต่อนั้นๆ นอกจากนั้นแล้วยังมีการเปลี่ยนค่า nonce ทุกๆ ช่วงเวลาเนื่องมาจาก

ความปลอดภัย หากมีการเปลี่ยนค่า nonce ทุกๆ ช่วงเวลา กุญแจลับที่ได้ในแต่ละครั้งจะมีค่าเปลี่ยนไป หลังจากนั้นหากมีการเข้าถึงทรัพยากรที่ถูกปกป้องด้วยไดเจสต้องตรวจสอบสิทธิ์ในการเข้าถึง หากค่าต่างๆ ที่ระบุมาไม่ถูกต้อง ระบบจะส่งค่าตอบรับเป็นค่า 401 เพื่อให้ทางเครื่องลูกข่ายรับรู้ถึงความผิดพลาดในการพิสูจน์ตน แต่หากมีสิทธิ์ในการเข้าถึง หรือทรัพยากรไม่ถูกปกป้องด้วยไดเจสแล้ว ต้องตรวจสอบว่าเป็นการร้องขอซีจีไอหรือไม่ หากใช่ต้องส่งการทำงานนั้นๆ ไปยังฟังก์ชันที่รับผิดชอบ ผังงานแสดงได้ดังรูปที่ 4.19

- websnmp_handler เป็นฟังก์ชันที่รับผิดชอบต่อการทำงานทางด้านการจัดการเอสเอ็นเอ็มพี โดยมีหน้าที่ต้องตรวจสอบสถานะการทำงานของอุปกรณ์เครือข่ายทุกๆ ช่วงเวลาตามที่ได้ระบุไว้ โดยตรวจสอบช่วงเลขที่อยู่ไอพีที่ผู้ใช้สามารถกำหนดเองได้ นอกจากนี้แล้วยังต้องแสดงสถานะการเชื่อมต่อของเครื่องลูกข่ายที่กำลังติดต่ออยู่กับระบบฝังตัวในขณะนั้น และเนื่องจากค่า nonce ได้ถูกนำมาสร้างเป็นกุญแจลับสำหรับการเข้ารหัสถอดรหัสด้วย ดังนั้นฟังก์ชันนี้ต้องคอยตรวจสอบค่า nonce ของแต่ละการเชื่อมต่อ หากมีการเปลี่ยนแปลงต้องสร้างกุญแจลับชุดใหม่ขึ้นมาแทนที่ชุดเดิม ฟังก์ชันนี้ยังรับหน้าที่ตรวจสอบการเกิดเหตุการณ์ที่ผู้ใช้ได้ตั้งการตรวจสอบไว้ และคอยตรวจสอบเอสเอ็นเอ็มพีแทรพด้วย (สำหรับรายละเอียดโครงสร้างข้อมูลเอสเอ็นเอ็มพีแทรพนั้นแสดงอยู่ในภาคผนวก ค) ผังงานของฟังก์ชันนี้แสดงได้ตามรูปที่ 4.20



รูปที่ 4.19 ผังงานของฟังก์ชัน http_handler

รูปที่ 4.20 ผังงานของฟังก์ชัน websnmp_handler

นอกจากเรื่องของซีจีไอแล้วยังมีจุดสังเกตอีกประการคือ เมื่อมีการติดต่อเข้ามาของผู้ใช้ ระบบจะกำหนดให้การเชื่อมต่อสำหรับคำสั่งต่างๆ ในครั้งนั้นเป็นแบบถาวร (persistent) ซึ่งตราบเท่าที่ยังไม่มีการสิ้นสุดการเชื่อมต่อจะไม่มีการตัดการเชื่อมต่อเนื่องจากไม่มีการรับส่งข้อมูลเป็นระยะเวลานาน โดยปกติแล้วหากเป็นการร้องขอข้อมูลเว็บเพจส่วนของระบบจะไม่กำหนดให้เป็นแบบถาวร แต่หากผู้ใช้เริ่มสั่งงานกับส่วนย่อยการจัดการโดยผ่านซีจีไอ ระบบจะเริ่มการกำหนดการเชื่อมต่อแบบถาวรให้แก่ผู้ใช้นั้น

การจัดการของส่วนย่อยนี้แบ่งได้ 2 วิธีซึ่งมีวิธีการทำงานคล้ายคลึงกัน แต่ส่วนสั่งงานไม่เหมือนกัน ดังนี้

- วิธีการจัดการด้วยมือ (Manual mode) วิธีนี้การสั่งงานเกิดจากผู้ใช้ผ่านทางเว็บบราวเซอร์ ดังนั้นคำสั่งจึงเกิดจากซีจีไอที่ส่งมาจากเว็บบราวเซอร์ และการตอบรับกับการจัดการนี้เกิดขึ้นทันทีหลังจากระบบประมวลผลคำสั่งเรียบร้อยแล้ว
- วิธีการจัดการแบบอัตโนมัติ (Automatic mode) ในวิธีนี้เริ่มต้นต้องมีการกำหนดเหตุการณ์อัตโนมัติที่ต้องการให้ระบบคอยตรวจสอบเป็นระยะ ซึ่งการกำหนดเหตุการณ์นี้กระทำผ่านคำสั่งซีจีไอเช่นเดียวกัน แต่การจัดการมิได้เกิด

จากคำสั่งซีจีไอโดยตรง แต่เกิดจากส่วนย่อยการจัดการเอง โดยระบบจะแปลงคำสั่งซีจีไอพร้อมทั้งตัวแปรเสริม (parameter) ต่างๆ เก็บไว้ในส่วนย่อยการจัดการ ซึ่งหากค่าที่ได้ระบุมาพร้อมกับตัวแปรเสริมต่างๆ เกิดขึ้นตามเงื่อนไขที่ได้ตั้งไว้ ส่วนย่อยนี้จะเรียกใช้ฟังก์ชันสำหรับเก็บข้อมูลการเกิดเหตุการณ์ไว้เพื่อการตอบรับกับฟังก์ชันตรวจสอบการเกิดเหตุการณ์

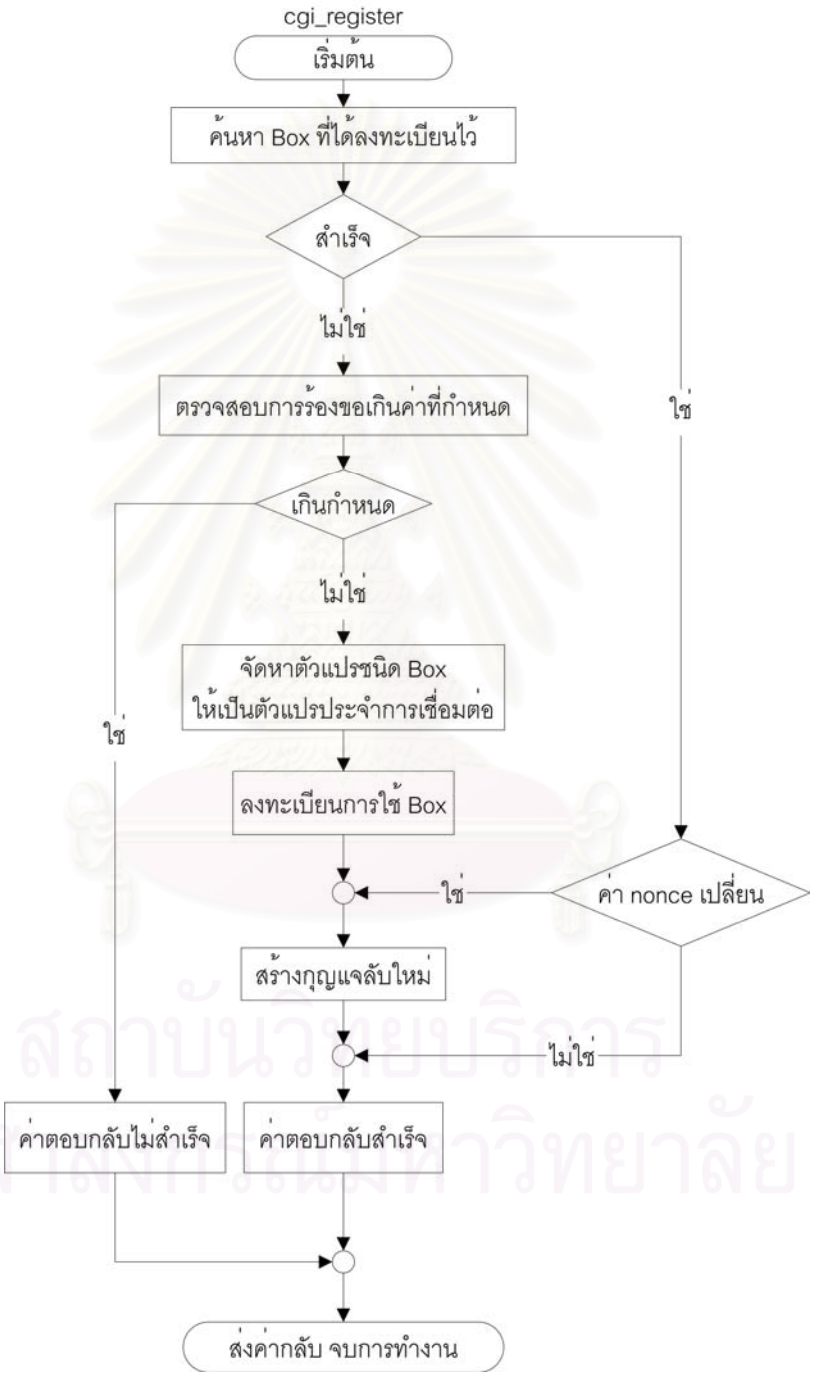
4.2.1.3 ส่วนย่อยการเชื่อมต่อเว็บ (Web connection section)

การทำงานของส่วนนี้อาศัยการทำงานของซีจีไอ ซึ่งผู้ใช้ที่ได้ส่งคำสั่งชนิดซีจีไอเข้ามาเป็นครั้งแรก ระบบจะทำการตรวจข้อจำกัดของการสร้างการเชื่อมต่อแบบถาวรซึ่งจำกัดไว้สำหรับการสร้างการเชื่อมต่อแบบถาวรได้พร้อมกัน 3 การเชื่อมต่อ ซึ่งหากมีการร้องขอการเชื่อมต่อที่เกินกว่ากำหนด ระบบจะไม่กำหนดการเชื่อมต่อแบบถาวรให้ มีฟังก์ชันต่างๆ ดังนี้

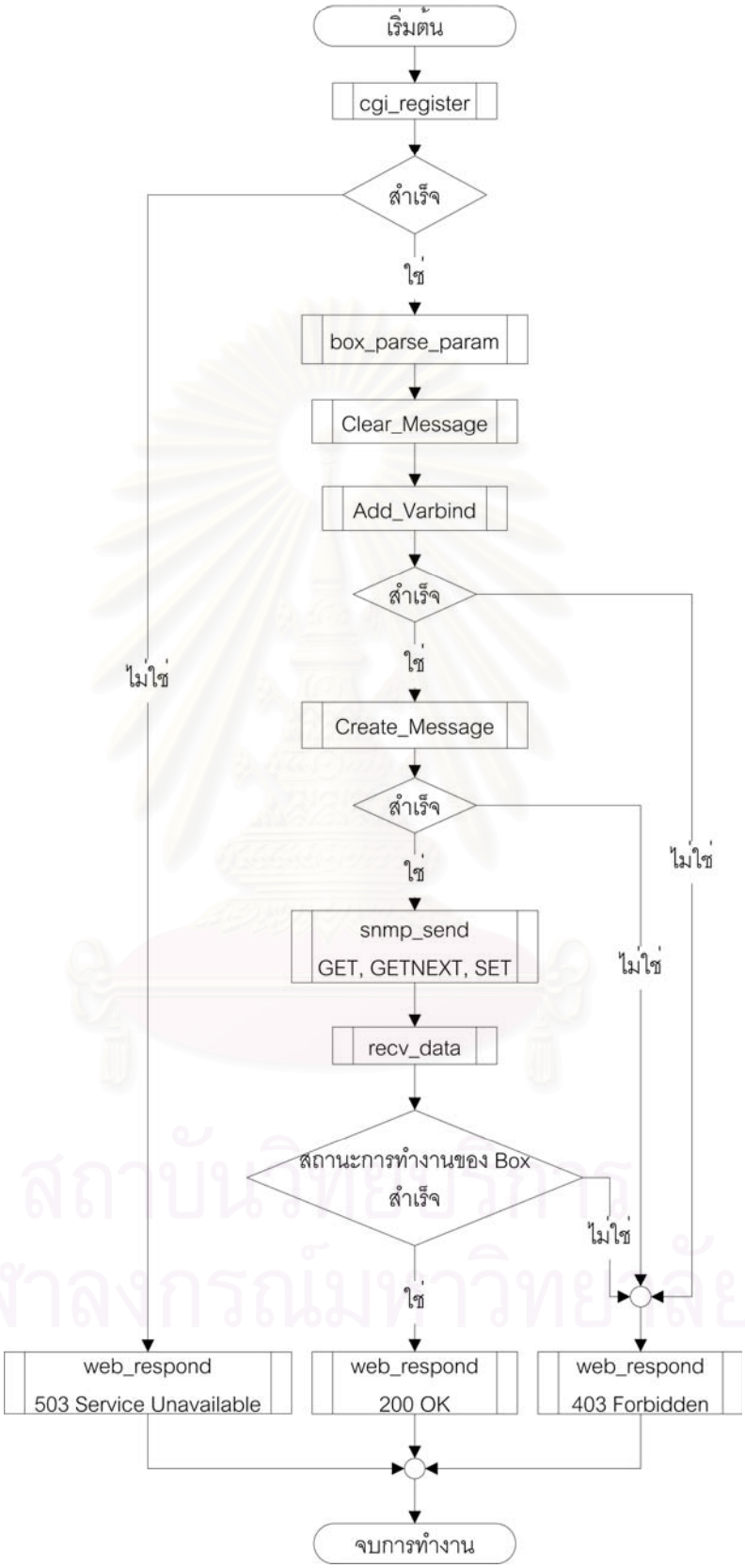
- cgi_register เป็นฟังก์ชันสำหรับการลงทะเบียนขอเริ่มใช้การเชื่อมต่อแบบถาวร หากจำนวนการเชื่อมต่อแบบถาวรยังไม่ถึงขอบเขตที่กำหนดไว้ ฟังก์ชันนี้จะจัดหาตัวแปรชนิด Box เพื่อใช้เป็นตัวแปรประจำสำหรับการเชื่อมต่อนั้นๆ ซึ่งหากการเชื่อมต่อในครั้งนั้นได้มีการลงทะเบียนไว้แล้ว ฟังก์ชันนี้จะไม่ลงทะเบียนซ้ำให้ แต่จะคืนค่าตัวแปรชนิด Box ประจำสำหรับการเชื่อมต่อนั้นๆ ผังงานของฟังก์ชันนี้แสดงได้ดังรูป 4.21

- cgi_snmp ทำหน้าที่รับคำสั่งเกี่ยวกับการติดต่อกับเอสเอ็นเอ็มพีเอเจนต์โดยตรง ซึ่งการเลือกให้ทำคำสั่งเอสเอ็นเอ็มพีใดนั้นต้องกระทำผ่านการส่งค่าตัวแปรเสริม action โดยมีคำสั่งที่เป็นไปได้ทั้งหมด 8 คำสั่งด้วยกัน สามารถดูค่าตัวแปรเสริมที่เป็นไปได้ทั้งหมดจากตาราง 4.3 ซึ่งคำสั่งต่างๆ นั้นบางคำสั่งมีลักษณะการทำงานคล้ายกัน จึงขอแสดงผังงานสำหรับการทำงานที่คล้ายกันรวมเป็นผังงานเดียวกัน เริ่มจากคำสั่งประเภทกำหนด หรือร้องขอค่าข้อมูลแบบพื้นฐาน ตัวอย่างคำสั่งประเภทนี้คือ คำสั่งซึ่งมีค่าตัวแปรเสริม action เป็น get getnext getgroup และ set โดยมีผังงานแสดงได้ดังรูปที่ 4.22 คำสั่งกลุ่มถัดไปเป็นประเภทร้องขอค่าข้อมูลทั้งหมดที่ขึ้นต้นด้วยโอไอดีที่ระบุ ซึ่งเป็นคำสั่งที่มีค่าตัวแปรเสริม action เป็น walk โดยมีผังงานดังรูปที่ 4.23 กลุ่มคำสั่งสำหรับการตรวจสอบสถานะของอุปกรณ์ต่างๆ มีค่าตัวแปรเสริม

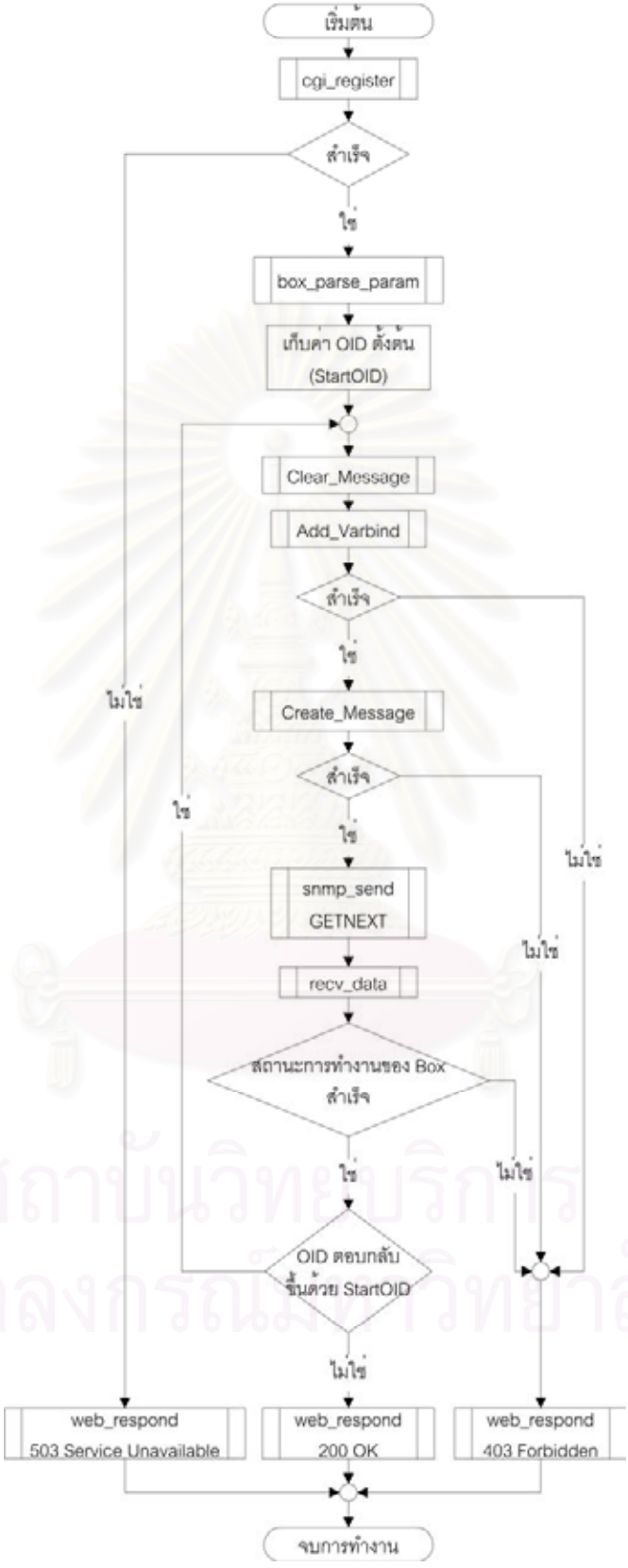
action เป็น ping โดยแสดงผังงานได้ตามรูปที่ 4.24 และกลุ่มคำสั่งสุดท้ายเกี่ยวกับเหตุการณ์สำหรับการจัดการแบบอัตโนมัติ ประกอบด้วยคำสั่งที่มีค่าตัวแปรเสริม action เป็น getevent และ setevent ซึ่งมีผังงานตามรูปที่ 4.25



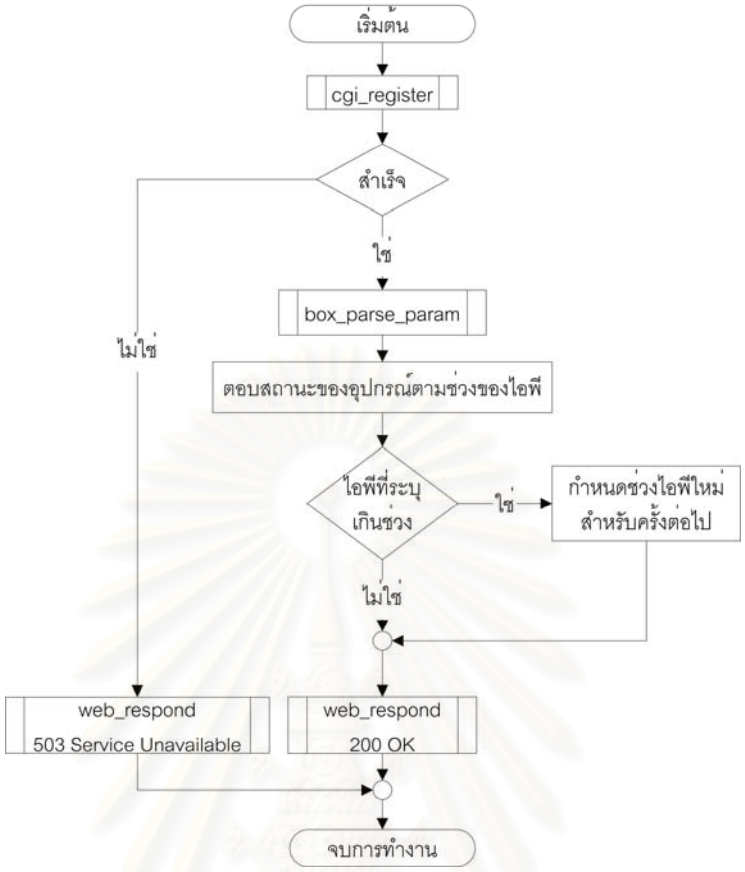
รูปที่ 4.21 ผังงานของฟังก์ชัน cgi_register



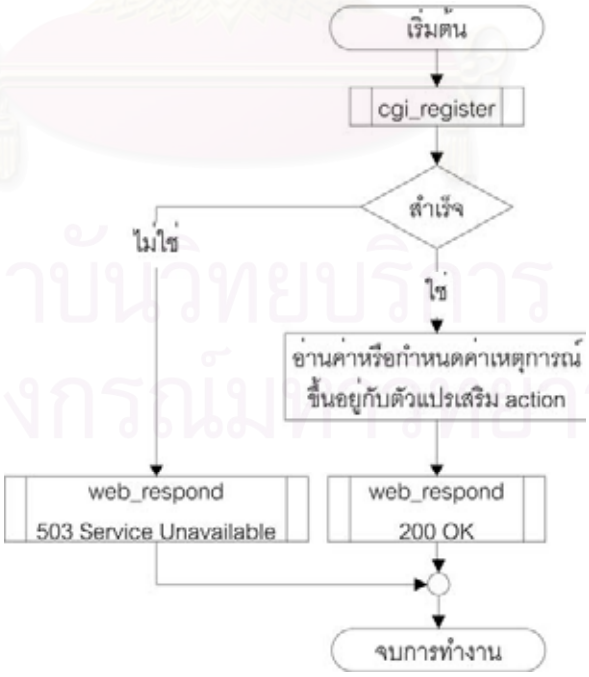
รูปที่ 4.22 ผังงานของคำสั่งประเภทกำหนด หรือร้องขอข้อมูลแบบพื้นฐาน



รูปที่ 4.23 ผังงานของคำสั่งประเภทประเภทร้องขอค่าข้อมูลทั้งหมดที่ขึ้นต้นด้วยโอไอดีที่ระบุ



รูปที่ 4.24 ผังงานของคำสั่งประเภทการตรวจสอบสถานะของอุปกรณ์ต่างๆ



รูปที่ 4.25 ผังงานของคำสั่งประเภทเหตุการณ์สำหรับการจัดการแบบอัตโนมัติ

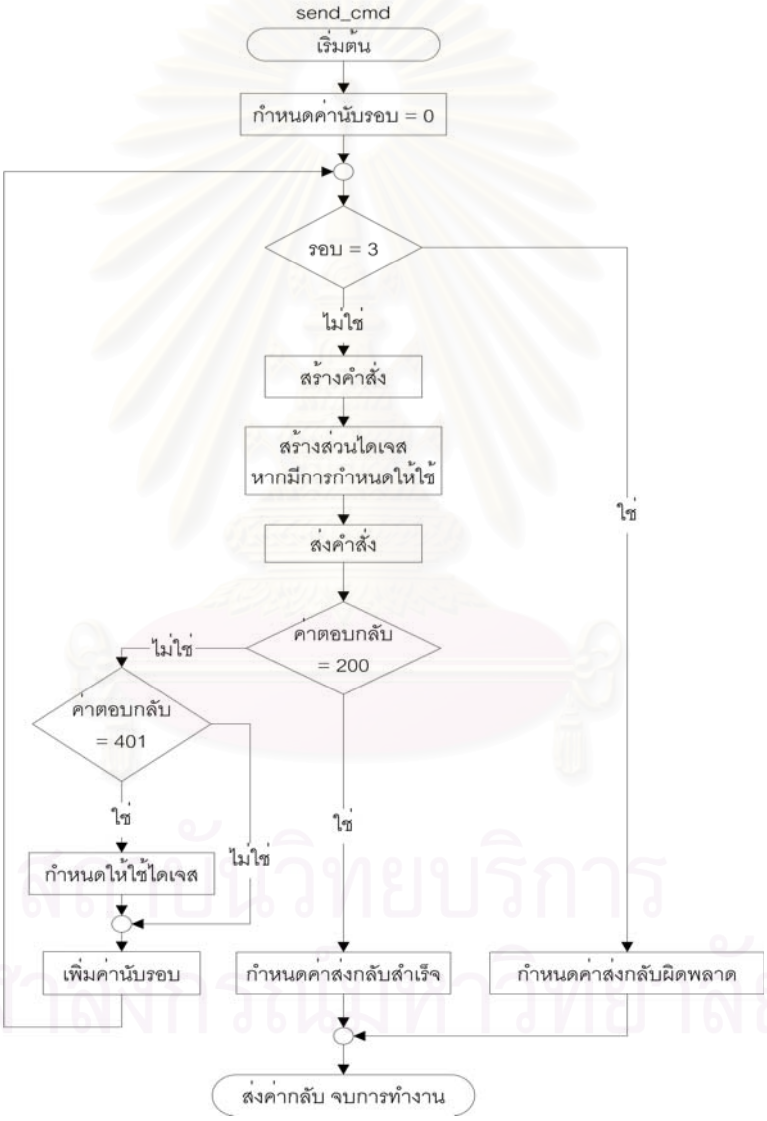
4.2.2 ส่วนเว็บ (web part)

ภายในส่วนเว็บนี้พัฒนาโดยใช้ภาษาจาวาในรูปของแอปเพล็ต (Applet) ซึ่งถึงแม้ว่าไดนามิกเว็บเพจ (Dynamic web page) สามารถรองรับการทำงานในส่วนนี้ได้ แต่หากส่วนติดต่อผู้ใช้มีการเปลี่ยนแปลงการทำงานจำเป็นต้องเปลี่ยนโปรแกรมภายในระบบฝังตัวโดยใส่โปรแกรมใหม่ผ่านทางส่วนต่อประสานการเขียนโปรแกรมเท่านั้น ในขณะที่หากพัฒนาในรูปแบบของแอปเพล็ตแล้วผู้พัฒนาสามารถเปลี่ยนแปลงได้ผ่านทางเว็บบราวเซอร์ด้วย และในเรื่องของความปลอดภัย ตามมาตรฐานแล้วรองรับเฉพาะในส่วนของการพิสูจน์ตนเท่านั้น ด้วยเหตุนี้จึงได้นำแอปเพล็ตมาใช้ร่วมด้วย เพื่อให้สามารถทำงานในด้านความปลอดภัย และรวมถึงการติดต่อกับผู้ใช้แทนเว็บเพจธรรมดา เนื่องจากมีประสิทธิภาพที่ดีกว่า ทั้งยังสามารถพัฒนาโปรแกรมให้ทำงานได้ซับซ้อนและหลากหลายกว่ามาตรฐานเอชทีทีพีธรรมดา หน้าที่หลักของจาวาแอปเพล็ตที่นำมาประยุกต์ใช้ คือต้องแปลความการเลือกตัวเลือกต่างๆ ของผู้ใช้จากหน้าจอการติดต่อ ให้กลายเป็นความต้องการของผู้ใช้เพื่อส่งข้อมูลไปสั่งการกับส่วนจัดการ เมื่อผู้พัฒนาทำการสร้างแอปเพล็ตขึ้นมาต้องอาศัยเครื่องคอมพิวเตอร์ในการแปลโปรแกรมให้อยู่ในรูปแบบคลาส (.class) เสียก่อนจึงนำมาใช้งานได้ และเพื่อความสะดวกยิ่งขึ้นในการนำไปเก็บไว้ในอุปกรณ์ฝังตัว จึงได้รวบรวมคลาสทั้งหมดที่ใช้ให้อยู่ในแฟ้มข้อมูลเพียงแฟ้มเดียวในรูปแบบจาร์ (.jar) และทำการคัดลอกไปยังอุปกรณ์ฝังตัว ประโยชน์ของการรวบรวมคลาสให้เป็นรูปแบบจาร์ นั้นนอกจากจะได้รับความสะดวกในการคัดลอกไปยังอุปกรณ์ฝังตัวแล้ว รูปแบบแฟ้มข้อมูลประเภทนี้ยังเก็บข้อมูลแบบบีบอัดทำให้ใช้ทรัพยากรของอุปกรณ์ฝังตัวได้อย่างคุ้มค่า และสำหรับผู้ใช้งานเว็บบราวเซอร์ที่สนับสนุนการทำงานของเอชทีทีพีรุ่น 1.1 ยังสามารถลดจำนวนข้อมูลที่เว็บบราวเซอร์ร้องขอแฟ้มข้อมูลจากอุปกรณ์ฝังตัวด้วยการใช้ความสามารถในการเรียกข้อมูลจากแคช (Cache) ถ้าหากข้อมูลไม่มีการเปลี่ยนแปลง ในกรณีที่มีการติดต่อมายังอุปกรณ์ฝังตัว ซอฟต์แวร์ภายในอุปกรณ์ฝังตัวจะส่งข้อมูลรูปแบบจาร์นี้ไปให้กับเว็บบราวเซอร์ที่ผู้ใช้ทำการติดต่ออยู่ ซึ่งเว็บบราวเซอร์จะรับผิดชอบในการหาคลาสจากแฟ้มข้อมูลจาร์ จากนั้นการทำงานของส่วนเว็บจึงเริ่มขึ้นที่ส่วนเว็บบราวเซอร์

4.2.2.1 ส่วนย่อยการเชื่อมต่อระบบฝังตัว (Embedded Connection Section)

ส่วนย่อยนี้สามารถติดต่อกับอุปกรณ์ฝังตัวผ่านเอชทีทีพี ดังนั้นต้องสร้างการติดต่อชนิดเอชทีทีพี และเนื่องจากคำสั่งต่างๆ ที่ระบบฝังตัวกำหนดให้เรียกใช้ได้นั้นเป็นคำสั่งซีจีไอ ดังนั้นภายในส่วนย่อยนี้ประกอบด้วยฟังก์ชันสำคัญสำหรับการสร้างชุดข้อมูลเป็นคำสั่งซีจีไอส่งไปยังระบบฝัง โดยประกอบด้วยฟังก์ชันดังนี้

- send_cmd เป็นฟังก์ชันสำหรับสร้างคำสั่ง โดยมีหน้าที่รับผิดชอบต่อความผิดพลาดจากการพิสูจน์ตนแบบไดเจสด้วย โดยเมื่อส่งคำสั่งใดๆ ไปให้กับระบบฝังตัวแล้ว ฟังก์ชันนี้ต้องรอรับข้อมูลพร้อมกับตรวจสอบค่าที่ตอบกลับมาซึ่งหากเป็นค่า 401 หมายความว่าการตรวจสอบไดเจสไม่สำเร็จ ฟังก์ชันนี้ต้องสร้างคำสั่งใหม่พร้อมกับใส่ข้อมูลไดเจสใหม่ไปพร้อมกับคำสั่ง โดยฟังก์ชันจะพยายาม 3 ครั้ง ผังงานของฟังก์ชันนี้แสดงได้ดังรูปที่ 4.26



รูปที่ 4.26 ผังงานของฟังก์ชัน send_cmd

ฟังก์ชันนี้ถูกเรียกใช้โดยฟังก์ชันอื่นๆ ซึ่งสามารถสร้างเป็นฟังก์ชันสำหรับส่งคำสั่งซีจีไอไปยังระบบฝังตัว หากผู้พัฒนาต้องการสร้างฟังก์ชันใหม่สามารถเรียกใช้งานฟังก์ชันนี้ นอก

จากนี้แล้วฟังก์ชันนี้ยังทำหน้าที่รับผิดชอบการเข้ารหัสหากผู้ใช้ต้องการติดต่อด้วยระบบรักษาความปลอดภัยซึ่งจำเป็นต้องสร้างกุญแจลับด้วยข้อมูลพื้นฐานเดียวกันกับที่ระบบฝังตัวใช้ [20]

4.2.2.2 ส่วนย่อยส่วนต่อประสานกับผู้ใช้ (User Interface Section)

ภายในส่วนนี้เป็นการติดต่อกับผู้ใช้โดยผ่านจาวาแอปเพล็ต ซึ่งเป็นส่วนต่อประสานกราฟฟิกกับผู้ใช้ โดยการเลือกควบคุมส่วนเอเจนต์ใดๆ นั้น ผู้ใช้สามารถเลือกจากสัญรูป (icon) ของเอเจนต์ที่ปรากฏภายในหน้าจอ นอกจากนั้นแล้วผู้ใช้ยังสามารถเลือกวิธีการจัดการด้วยมืออย่างสะดวกผ่านแผนภูมิต้นไม้เอ็มไอบี

4.2.3 การติดต่อระหว่างส่วนฝังตัว และส่วนเว็บ

การติดต่อระหว่างส่วนฝังตัว และส่วนเว็บพัฒนามาจากการใช้งานเอสเอ็นเอ็มพี-ยูอาร์แอล [9] โดยแบ่งส่วนการอธิบายไว้ 3 ส่วนด้วยกัน โดยภายใน 2 ส่วนแรกกล่าวถึงการติดต่อสื่อสารกันระหว่างส่วนเว็บ และส่วนฝังตัวโดยไม่มีระบบรักษาความปลอดภัย ซึ่งระบบรักษาความปลอดภัยจะได้กล่าวถึงในหัวข้อ 4.2.3.3

4.2.3.1 การส่งคำสั่งจากส่วนเว็บไปยังส่วนฝังตัว

การส่งคำสั่งจากส่วนเว็บไปยังส่วนฝังตัวนั้น อาศัยคำสั่งซีจีไอเพื่อส่งไปยังส่วนฝังตัว โดยสามารถแสดงรูปแบบของคำสั่งซีจีไอได้ดังรูปที่ 4.27

```
command?parameter1=value1&parameter2=value2&…
```

รูปที่ 4.27 รูปแบบคำสั่งซีจีไอ

คำสั่งซีจีไอที่ได้ออกแบบนั้นประกอบไปด้วย 2 ส่วน คือ ชื่อของคำสั่งซีจีไอ (command) และชุดของตัวแปรเสริม (parameter) ซึ่งเป็นส่วนข้อมูลที่อยู่หลังเครื่องหมายปรัศนี (?) คำสั่งบางคำสั่งนั้นสามารถดำเนินการได้ทันทีโดยไม่จำเป็นต้องมีชุดของตัวแปรเสริม ตัวอย่างของคำสั่งซีจีไอสามารถแสดงได้ดังรูปที่ 4.28

```
snmp?action=get&oid=1.3.6.1.2.1.1.5.0&community=public&
host=161.200.92.171&ver=1
```

รูปที่ 4.28 ตัวอย่างคำสั่งซีจีไอ

จากตัวอย่างแสดงให้เห็นคำสั่งซีจีไอ snmp โดยค่าของตัวแปรเสริม “action” เป็น “get” ซึ่งหมายถึงการร้องขอค่าข้อมูล คำสั่งซีจีไอทั้งหมดที่ระบบรองรับเป็นดังตารางที่ 4.3 โดยบางคำสั่งยังได้แบ่งเป็นคำสั่งย่อยด้วยการใช้ค่าของตัวแปรเสริม “action” เพิ่มเข้ามา สำหรับตัวแปรเสริมแต่ละชุดนั้น ประกอบไปด้วยชื่อของตัวแปรเสริม และค่าของตัวแปรเสริม ตัวแปรเสริมที่ใช้ได้นั้นขึ้นอยู่กับคำสั่ง ซึ่งได้แบ่งตัวแปรเสริมออกเป็น 5 ประเภทดังนี้

- ตัวแปรเสริมประเภททั่วไป สามารถใช้กับฟังก์ชันทุกประเภทยกเว้นฟังก์ชัน event และ protected ตัวแปรเสริมประเภทนี้ใช้สำหรับการกำหนดค่าต่างๆ ให้กับการจัดการด้วยมือเป็นส่วนใหญ่ และมีการกำหนดค่าทั่วๆ ไปสำหรับอุปกรณ์ฝังตัว สามารถแสดงตัวแปรเสริมประเภทนี้ได้ดังตาราง 4.4

- ตัวแปรเสริมประเภทการจัดการแฟ้มข้อมูล ใช้ได้เฉพาะกับคำสั่ง filemanager เท่านั้น โดยมีตัวแปรเสริมดังตาราง 4.5

- ตัวแปรเสริมประเภทการจัดการเรื่องเวลา ใช้ได้เฉพาะกับคำสั่ง RTC โดยมีเพียงตัวแปรเสริมเดียวเท่านั้นคือ time สำหรับกำหนดเวลาให้กับนาฬิกา โดยค่าที่ระบุมาต้องเป็นจำนวนวินาทีนับตั้งแต่วันที่ 1 มกราคม ค.ศ. 1980

- ตัวแปรเสริมประเภทการจัดการแบบอัตโนมัติ ใช้ได้เฉพาะกับคำสั่ง Event โดยมีตัวแปรเสริมดังตาราง 4.6

- ตัวแปรเสริมประเภทการจัดการข้อมูลเอสเอ็นเอ็มพีแทรพ ใช้ได้เฉพาะคำสั่ง trap โดยมีตัวแปรเสริมดังตาราง 4.7

ตารางที่ 4.3 รายการคำสั่งซีจีไอทั้งหมด

ชื่อคำสั่ง	ค่าตัวแปรเสริม action	ความหมายของคำสั่ง
snmp	get	ร้องขอข้อมูลจากเอสเอ็นเอ็มพีเอเจนต์
	getnext	ร้องขอข้อมูลหมายเลขโอไอดีถัดไปจากเอสเอ็นเอ็มพีเอเจนต์
	getgroup	ร้องขอข้อมูลเป็นชุดด้วยคำสั่งเอสเอ็นเอ็มพีชุดเดียว
	walk	ร้องขอข้อมูลด้วยการเริ่มต้นที่หมายเลขโอไอดีที่ระบุ จนกระทั่งถึงข้อมูลสุดท้ายที่ขึ้นต้นด้วยโอไอดีที่ต่างออกไป
	set	กำหนดค่าข้อมูลของเอสเอ็นเอ็มพีเอเจนต์
	ping	ตรวจสอบสถานะของอุปกรณ์เครือข่ายภายในช่วงของเลขที่อยู่ไอพีที่ระบุ
save2flash	-	บันทึกข้อมูลสำคัญลงในหน่วยความจำแฟลช
filemanager	add	เพิ่มแฟ้มข้อมูลลงภายในอุปกรณ์ฝังตัว
	remove	ลบแฟ้มข้อมูลออกจากอุปกรณ์ฝังตัว
	list	แสดงรายชื่อแฟ้มข้อมูลทั้งหมดที่อนุญาตให้มีการแก้ไขเปลี่ยนแปลงได้
RTC	set	กำหนดเวลาภายในนาฬิกาของอุปกรณ์ฝังตัว
	get	ร้องขอเวลาภายในนาฬิกาของอุปกรณ์ฝังตัว
event	set	กำหนดเงื่อนไขการจัดการแบบอัตโนมัติ
	get	ร้องขอเงื่อนไขการจัดการแบบอัตโนมัติภายในขณะนั้น
	getnew	ตรวจสอบเหตุการณ์ที่เกิดขึ้นใหม่นับจากครั้งก่อน
trap	setserver	กำหนดชื่อเครื่องบริการเอสเอ็มทีพีสำหรับส่งอีเมลแทรพ
	getserver	ร้องขอชื่อเครื่องบริการเอสเอ็มทีพีสำหรับส่งอีเมลแทรพ
	addemail	เพิ่มรายชื่ออีเมลสำหรับการแจ้งเตือนแทรพ
	rememail	ลบรายชื่ออีเมลสำหรับการแจ้งเตือนแทรพ
	getemail	ร้องขอรายชื่ออีเมลทั้งหมดที่ถูกกำหนดไว้
	getnew	ตรวจสอบการพบแทรพดักใหม่นับตั้งแต่ครั้งก่อน
protected	-	จัดการติดต่อโดยใช้การเข้ารหัสเออีเอส

ตารางที่ 4.4 รายชื่อตัวแปรเสริมประเภททั่วไป

ชื่อของตัวแปรเสริม	ความหมายของตัวแปรเสริม
Host	เลขที่อยู่ไอพีของเอเจนต์ซึ่งเป็นข้อมูลชนิดสายอักขระ (String)
Ver	รุ่นของเอสเอ็นเอ็มพีซึ่งในงานวิจัยนี้สนับสนุนเฉพาะรุ่นที่ 1 เท่านั้น
Community	คอมมิวนิตี้สตริง เพื่อขอเข้าใช้งานเอเจนต์นั้นๆ
Oid	หมายเลขโอไอดีของเอเจนต์
Oidgroup	ชุดของตัวแปรเสริม “oid” โดยแต่ละหมายเลขโอไอดีถูกแบ่งด้วยเครื่องหมายบวก (+)
Type	ชนิดข้อมูลของค่าอ็อบเจกต์เอสเอ็นเอ็มพี
Val	ค่าของอ็อบเจกต์เอสเอ็นเอ็มพี
Startip	เลขที่อยู่ไอพีเริ่มต้นสำหรับการตรวจสอบสถานะ (ping)
Stopip	เลขที่อยู่ไอพีสิ้นสุดสำหรับการตรวจสอบสถานะ

ตารางที่ 4.5 รายชื่อตัวแปรเสริมประเภทการจัดการแฟ้มข้อมูล

ชื่อของตัวแปรเสริม	ความหมายของตัวแปรเสริม
filename	ชื่อของแฟ้มข้อมูล
filesize	ขนาดของแฟ้มข้อมูล
filetarget	กำหนดตำแหน่งในการเก็บข้อมูลในกรณีของการเพิ่มแฟ้มข้อมูลใหม่ โดยค่าที่เป็นไปได้คือ ram-สำหรับกำหนดให้เก็บในหน่วยความจำแรมสถิต และ flash-สำหรับกำหนดให้เก็บในหน่วยความจำแฟลช

เนื่องจากในส่วนของการจัดการแบบอัตโนมัตินั้นประกอบไปด้วยเอเจนต์ที่ผู้ใช้ต้องการให้มีการตรวจสอบเหตุการณ์ และเอเจนต์ที่ผู้ใช้ต้องการเปลี่ยนแปลงค่าหากมีเหตุการณ์เกิดขึ้น ดังนั้นจึงกำหนดเอเจนต์ที่ก่อให้เกิดเหตุการณ์ด้วยคำว่า เอเจนต์ต้นทาง (Source agent) และกำหนดเอเจนต์ที่ผู้ใช้ต้องการเปลี่ยนแปลงค่าหากมีเหตุการณ์เกิดขึ้นด้วยคำว่า เอเจนต์ปลายทาง (Destination agent)

ตารางที่ 4.6 รายชื่อตัวแปรเสริมประเภทการจัดการแบบอัตโนมัติ

ชื่อของตัวแปรเสริม	ความหมายของตัวแปรเสริม
condnum	ลำดับที่ของเหตุการณ์ที่ต้องการตั้งค่าเริ่มที่ลำดับที่ 1
polltime	รอบเวลาสำหรับการตรวจสอบเหตุการณ์มีหน่วยเป็นวินาที ซึ่งหากมีค่าเป็น 0 หมายถึงให้ปิดการตรวจสอบเหตุการณ์ทุกเหตุการณ์
srcAgent	เลขที่อยู่ไอพีสำหรับเอเจนต์ต้นทาง
destAgent	เลขที่อยู่ไอพีสำหรับเอเจนต์ปลายทาง
srcComm	คอมมิวนิตี้สตริงสำหรับเอเจนต์ต้นทาง
destComm	คอมมิวนิตี้สตริงสำหรับเอเจนต์ปลายทาง
srcOID	หมายเลขโอไอดีสำหรับเอเจนต์ต้นทาง
destOID	หมายเลขโอไอดีสำหรับเอเจนต์ปลายทาง
Cond	เงื่อนไขสำหรับการตรวจสอบ โดยมีเงื่อนไขที่เป็นไปได้คือ >, <, =, >=, <= และ !=
srcVal	ค่าที่ผู้ใช้ใส่เข้ามาจากแอปเพล็ต สำหรับเปรียบเทียบกับค่าที่ร้องขอมาจากเอเจนต์ต้นทาง
desttype	ชนิดของข้อมูลสำหรับเอเจนต์ปลายทาง
destVal	ค่าของข้อมูลสำหรับเอเจนต์ปลายทาง
Enable	เมื่อกำหนดให้เป็น 0 หมายถึง ปิดการตรวจสอบเหตุการณ์ลำดับที่ตัวแปรเสริม “condnum” ได้กำหนดไว้ หากค่านอกเหนือจากนี้ให้เปิดการตรวจสอบเหตุการณ์

ตารางที่ 4.7 รายชื่อตัวแปรเสริมประเภทการจัดการข้อมูลเอสเอ็นเอ็มพีแทรพ

ชื่อของตัวแปรเสริม	ความหมายของตัวแปรเสริม
email	อีเมลแอดเดรสที่ต้องการเพิ่มในรายชื่ออีเมลสำหรับการแจ้งเตือน
server	ชื่อเครื่องบริการเอสเอ็มทีพี

4.2.3.2 การตอบรับคำสั่งกลับจากส่วนฝังตัว

เมื่อระบบฝังตัวได้รับคำสั่งซีจีไอพร้อมทั้งประมวลผลข้อมูลตามคำสั่งที่ได้รับมาแล้ว ระบบต้องทำการตอบรับคำสั่งกลับไปยังส่วนเว็บ ซึ่งการตอบรับกลับนี้ประกอบไปด้วยส่วนหัว (header) ตามด้วยส่วนเนื้อหา (content) ในการตอบรับคำสั่งกลับนั้นส่วนหัวจำเป็นต้องมีการตอบรับไปกับทุกๆ คำสั่งไม่ว่าคำสั่งนั้นๆ จะทำงานสำเร็จหรือไม่ก็ตาม หากต้องมีส่วนเนื้อหาแล้วต้องเพิ่มเขตข้อมูล “Content-Length” เข้าไปกับส่วนหัวด้วย เพื่อให้สามารถประมวลผลได้ว่าประกอบด้วยส่วนของเนื้อหาขนาดกี่ไบต์ ตัวอย่างของการตอบรับกลับแสดงได้ดังนี้

HTTP/1.0 200 OK
Date: Sun. 27 June 2004 3:37:14 GMT
Content-Length: 25
Content-Type: text/html

1.3.6.1.2.1.1.5.0
ong

จากตัวอย่างเป็นการตอบรับกลับโดยให้รหัสกลับคืนเท่ากับ 200 พร้อมทั้งมีข้อมูล 1 ชุด ซึ่งเป็นข้อมูลของหมายเลขโอไอดี 1.3.6.1.2.1.1.5.0 (ชื่อเครื่องของเอเจนต์) โดยมีค่าเป็น “ong” รหัสกลับคืนที่เป็นไปได้สำหรับการตอบรับกลับของระบบ อ้างอิงจากมาตรฐานของเอชทีทีพี โดยมีค่าที่เป็นไปได้ดังตารางที่ 4.8

ตารางที่ 4.8 แสดงรหัสกลับคืนของระบบฝังตัว

รหัสกลับคืน	ค่าตอบรับกลับพร้อมทั้งความหมาย
200	OK – การประมวลผลคำสั่งสำเร็จ
403	Forbidden – เกิดข้อผิดพลาดระหว่างการประมวลผลคำสั่ง
503	Service Unavailable – มีการร้องขอเปิดการเชื่อมต่อชนิดถาวรในขณะที่จำนวนการเชื่อมต่อชนิดถาวรเปิดให้บริการจนถึงข้อจำกัดแล้ว

4.2.3.3 การติดต่อสื่อสารด้วยระบบความปลอดภัย

การทำงานของระบบฝังตัวที่ได้ออกแบบนั้น สามารถติดต่อสื่อสารกันโดยใช้การติดต่อแบบมาตรฐานซีจีไอทั้งแบบปกติ คือไม่มีการเข้ารหัสข้อมูล และยังสามารถติดต่อสื่อสารกันด้วยระบบการเข้ารหัสข้อมูลตามมาตรฐานเออีเอสด้วย โดยปรับปรุงระบบรักษาความปลอดภัยที่ได้นำเสนอไว้ [21] การออกแบบระบบให้สามารถรองรับการติดต่อทั้ง 2 รูปแบบนั้นมีข้อกำหนดเบื้องต้นสำหรับการติดต่อสื่อสารด้วยระบบความปลอดภัยดังนี้

- การเข้าถึงคำสั่งซีจีไอใดๆ บนระบบฝังตัวต้องมีการพิสูจน์ตนด้วยเอชทีทีพีไดเจส ซึ่งถึงแม้ข้อมูลที่ผ่านระบบเครือข่ายอินเทอร์เน็ตจะไม่มีการเข้ารหัสก็ตาม แต่อย่างน้อยระบบยังสามารถป้องกันการเข้าถึงจากผู้ที่ไม่มีสิทธิ์

- การเข้ารหัสของระบบใช้การเข้ารหัสแบบเออีเอสซึ่งทำงานด้วยกุญแจลับที่มีขนาด 128 บิต โดยปรับปรุงการทำงานจากไลบรารี (Library) ที่ได้รับการพัฒนาจาก Colin Percival [22] ให้สามารถเข้ารหัสและถอดรหัสแบบซีบีซี (CBC: Cipher Block Chaining) พร้อมทั้งยังปรับให้เหมาะสมกับการใช้งานผ่านระบบฝังตัวโดยการเก็บตารางการเข้ารหัสในหน่วยความจำส่วนเพิ่ม กำหนดให้สัญลักษณ์ AES_CBC_128 แทนการเข้ารหัสด้วยวิธีดังกล่าว

- ในกรณีที่ต้องการเข้ารหัสข้อมูล ส่วนหัวของคำสั่งจำเป็นต้องมีเขตข้อมูลนอกเหนือจากการส่งคำสั่งเอชทีทีพีปกติอย่างน้อย 2 ค่า คือค่าของ Encryption ต้องเป็น AES_CBC_128 และค่าของ Content-Length ระบุขนาดของข้อมูลภายในส่วนเนื้อหา หากไม่มีการเข้ารหัสข้อมูล เขตข้อมูล Encryption ต้องระบุค่า Plain หรือถ้าไม่มีเขตข้อมูลดังกล่าวถือว่าไม่มีการเข้ารหัสข้อมูล

- การเข้ารหัสข้อมูลสามารถใช้ได้เฉพาะกับคำสั่ง protected เท่านั้น โดยข้อมูลที่เข้ารหัสต้องอยู่ภายในส่วนเนื้อหาของคำสั่ง

- การเรียกใช้บริการจากคำสั่งอื่นๆ นอกเหนือจาก protected นั้น หากต้องการเข้ารหัสด้วยแล้ว ต้องส่งการร้องขอนั้นๆ ให้อยู่ในรูปแบบเข้ารหัส โดยนำการร้องขอที่ถูกเข้ารหัสใส่ในส่วนเนื้อหาของคำสั่ง protected จากนั้นระบบฝังตัว

จึงรับผิดชอบในการถอดรหัสข้อมูล และส่งการทำงานให้กับฟังก์ชันที่รับผิดชอบ ซึ่งหลักการเข้ารหัสด้วยวิธีนี้จะกล่าวถึงในภายหลัง

ระบบฝังตัวกำหนดการพิสูจน์ตนแบบเอชทีทีพีไดเจสไว้เป็นการพิสูจน์ตนพื้นฐานสำหรับการติดต่อสื่อสารที่ต้องการการเข้ารหัส เนื่องจากระบบที่ได้ออกแบบต้องอาศัยข้อมูลบางส่วนจากการพิสูจน์ตนแบบเอชทีทีพีไดเจสมาเป็นส่วนสร้างกุญแจลับ แต่การพิสูจน์ตนแบบพื้นฐานที่มีอยู่ในเอชทีทีพี 1.0 ยังสามารถใช้งานได้ เพียงแต่ไม่สามารถเข้ารหัสข้อมูลตามระบบที่ได้ออกแบบไว้ หากผู้พัฒนาไม่ต้องการการทำงานส่วนการเข้ารหัสสามารถยกเลิกการใช้งานชุดคำสั่งสำหรับการเข้ารหัส ทำให้ประหยัดทรัพยากรของระบบฝังตัวในกรณีที่ไม่ต้องการเข้ารหัสข้อมูล

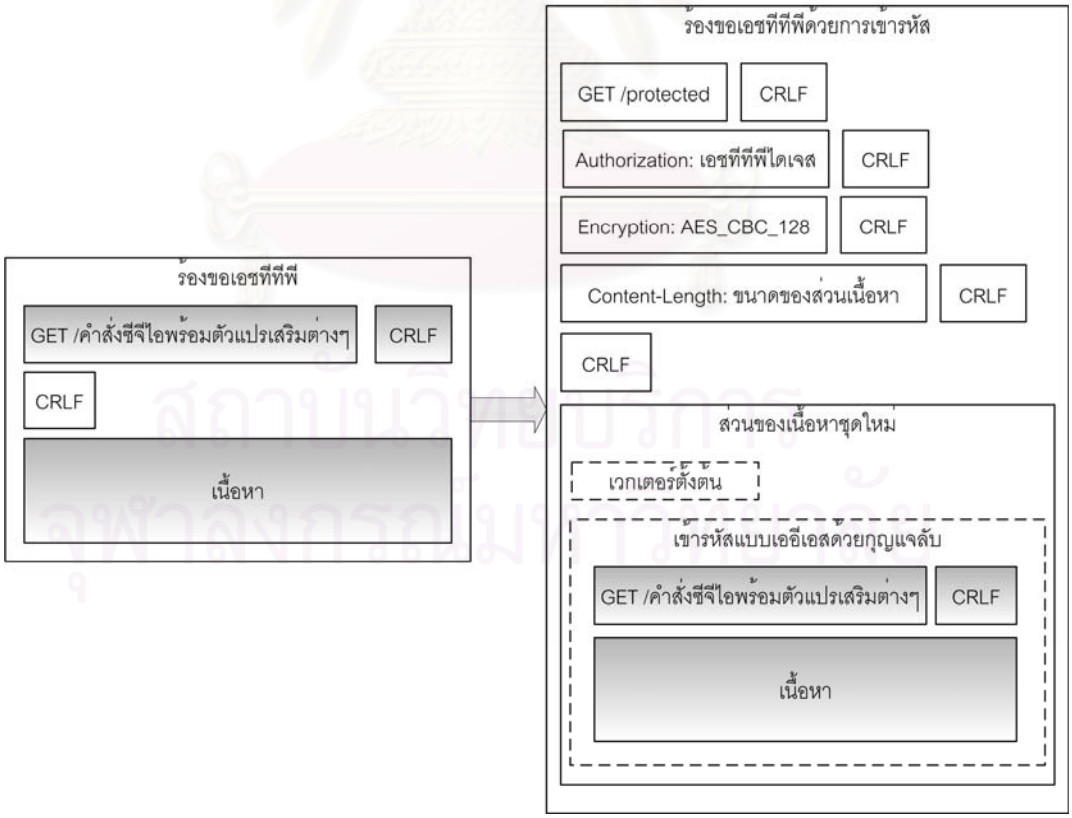
กุญแจลับเกิดจากข้อมูลพื้นฐานที่รู้กันแค่ 2 ฝ่าย คือระบบฝังตัว และผู้ดูแลระบบที่ต้องการขอเข้าใช้งาน ซึ่งการออกแบบระบบนี้ใช้ข้อมูลลับประกอบกับข้อมูลพื้นฐานเพื่อสร้างกุญแจลับโดยข้อมูลทั้งหมดที่ใช้สร้างกุญแจลับประกอบด้วย ชื่อผู้ใช้งาน รหัสผ่าน ชื่อเรียลม์ (Realm) และค่าน็อนซ์ (Nonce) สำหรับข้อมูล 4 ชุดนี้ มีเพียงรหัสผ่านเท่านั้นที่จะไม่ถูกเปิดเผยระหว่างการติดต่อสื่อสาร โดยกุญแจลับนั้นเป็นค่าตัวแทนข้อมูลเอ็มดี 5 ของข้อมูล 4 ชุดนี้ ซึ่งแสดงการสร้างกุญแจลับได้ดังสมการ 4.1

กุญแจลับ = MD5 (ชื่อผู้ใช้งาน + รหัสผ่าน + ชื่อเรียลม์ + น็อนซ์)……4.1

จากสมการข้างต้นเห็นได้ว่ากุญแจลับมีการเปลี่ยนได้ต่อเมื่อมีค่าใดค่าหนึ่งของชุดข้อมูล 4 ชุดเปลี่ยนไป แต่จากการออกแบบระบบกำหนดให้ค่าน็อนซ์เปลี่ยนไปทุกๆ ช่วงเวลาโดยเป็นค่าสุ่มขึ้นมา ดังนั้นการสร้างกุญแจลับจึงมีกุญแจลับที่เปลี่ยนไปทุกๆ ช่วงเวลา (Session key) ค่าน็อนซ์เกี่ยวข้องกับการพิสูจน์ตนแบบเอชทีทีพีไดเจส ดังนั้นหากค่าน็อนซ์เปลี่ยนแล้วการติดต่อสื่อสารใดที่ยังใช้ค่าน็อนซ์เดิมจะถูกเข้ารหัสด้วยกุญแจลับซึ่งหมดอายุแล้ว ส่งผลให้ข้อมูลที่ได้จากการเข้ารหัสไม่ถูกต้อง ระบบที่ติดต่อด้วยจำเป็นต้องส่งข้อมูลใหม่ซึ่งได้จากกุญแจลับชุดใหม่มาแทน แต่ทั้งนี้ระบบได้มีการป้องกันปัญหานี้ไว้ด้วยการพิสูจน์ตนเสียก่อน หากค่าน็อนซ์เปลี่ยนไปแล้วแต่ระบบที่ติดต่อเข้ามายังระบบฝังตัวยังใช้ค่าน็อนซ์เป็นค่าเดิม ย่อมส่งผลให้การพิสูจน์ตนผิดพลาดและไม่สนใจส่วนเนื้อหานั้นๆ ซึ่งจะเห็นได้ว่าระบบภายนอกจำเป็นต้องสร้างกุญแจลับชุดใหม่ออกมาในการติดต่อครั้งต่อไป นอกจากนี้แล้วกุญแจลับที่ได้จะไม่ถูกส่งออกไปยังเครือข่ายดังนั้นกุญแจลับจึงไม่ถูกเปิดเผย ฟังก์ชันสำหรับการสร้างตัวแทนข้อมูลดัดแปลงจากไลบรารีมาตรฐาน [23] วิธีการในการสร้างกุญแจลับนี้คล้ายคลึงกับข้อตกลงใช้กุญแจลับดิฟฟี่-เฮลแมน (Diffie-Hellman Key Agreement Protocol) ซึ่งมีการแลกเปลี่ยนข้อมูลซึ่งเปิดเผยได้ระหว่าง

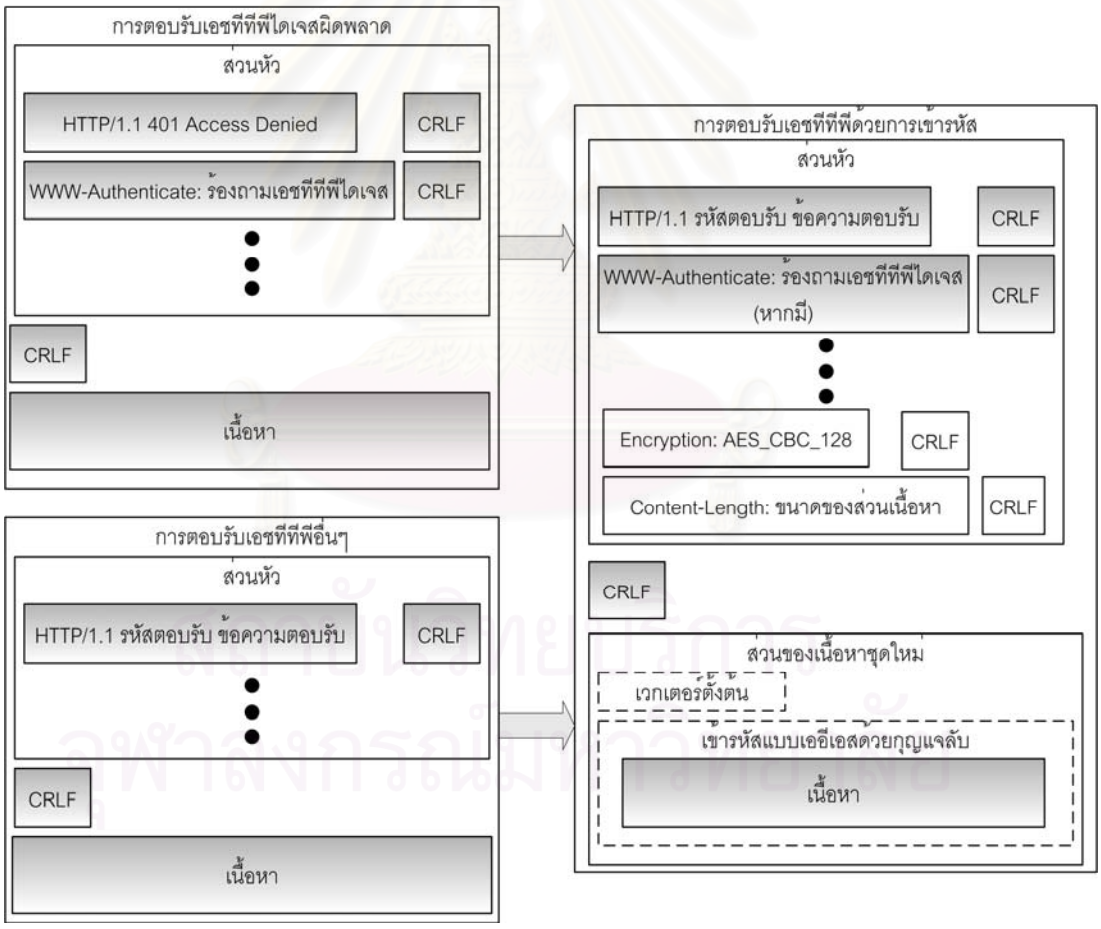
สองฝั่งการติดต่อเพื่อนำไปสร้างกุญแจลับชุดที่เหมือนกัน ดังนั้นสามารถนำมาใช้แทนการสร้างกุญแจลับในงานวิจัยนี้ได้แต่ต้องเพิ่มการคำนวณเลขยกกำลังซึ่งมีความซับซ้อนมากกว่าการสร้างตัวแทนข้อมูล ดังนั้นงานวิจัยนี้จึงได้เลือกใช้วิธีการสร้างกุญแจลับ

หลังจากการพิสูจน์ตนแล้ว การสื่อสารใดต้องการติดต่อด้วยการเข้ารหัสข้อมูลย่อมทำได้เพราะกุญแจลับได้ถูกสร้างขึ้นในขั้นตอนการพิสูจน์ตนแล้ว การติดต่อสื่อสารด้วยการเข้ารหัสนั้นอาศัยการติดต่อขั้นพื้นฐานภายในหัวข้อ 4.2.3.1 และ 4.2.3.2 นำมาเข้ารหัสเออีเอส และใส่ในส่วนเนื้อหาของคำสั่ง protected ซึ่งการติดต่อขั้นพื้นฐานนั้นมีส่วนสำคัญ 2 ส่วนด้วยกันคือ ส่วนคำสั่งซีจีไอ และส่วนเนื้อหา ดังนั้นหากต้องการร้องขอบริการซีจีไอด้วยการเข้ารหัสข้อมูล จำเป็นต้องรักษา 2 ส่วนสำคัญนี้ไว้ ซึ่งตามระบบที่ได้ออกแบบไว้นั้น คำสั่ง protected มีหน้าที่เป็นเสมือนส่วนครอบคำสั่งต่างๆ ให้อยู่ภายใต้การรักษาความปลอดภัยด้วยการเข้ารหัส โดยขั้นตอนการแปลงการร้องขอบริการแบบทั่วไป ให้อยู่ในรูปแบบของการเข้ารหัสนั้นแสดงได้ดังรูปที่ 4.29 ซึ่งเป็นขั้นตอนที่เกิดขึ้นหลังจากการพิสูจน์ตนแบบเอชทีทีพีไดเจสสำเร็จแล้ว ดังนั้นทั้งระบบฝั่งตัวและฝ่ายผู้ใช้งานจึงมีกุญแจลับที่ตรงกัน



รูปที่ 4.29 การแปลงคำสั่งซีจีไอให้อยู่ในรูปแบบเข้ารหัส

จากรูปแสดงให้เห็นถึงการแปลงการร้องขอในรูปแบบซีจีไอ ให้อยู่ในรูปแบบของการร้องขอที่มีการเข้ารหัสข้อมูล คำสั่งซีจีไอสำหรับการร้องขอแบบมีการเข้ารหัสนั้นต้องเป็นคำสั่ง protected เท่านั้น และส่วนหัวของการร้องขอแบบมีการเข้ารหัสยังต้องประกอบด้วยเขตข้อมูลอื่นๆ ซึ่งเป็นข้อมูลเกี่ยวกับการเข้ารหัสในครั้งนั้นๆ ส่วนของเนื้อหานั้นต้องระบุเวกเตอร์ตั้งต้นเพื่อใช้เป็นเวกเตอร์ตั้งต้นสำหรับการถอดรหัส เมื่อระบบฝั่งตัวได้รับการร้องขอซึ่งมีรูปแบบการเข้ารหัสแล้ว หากการตรวจสอบเอชทีทีพีไดเจสสำเร็จ ฟังก์ชัน protected ต้องถอดรหัสข้อมูลและส่งการทำงานให้กับฟังก์ชันที่รับผิดชอบทำงานต่อไป หลังจากการทำงานเสร็จสิ้นและต้องมีการตอบรับกลับนั้น หากการร้องขอเอชทีทีพีมีการระบุให้ใช้การเข้ารหัส การตอบรับกลับในครั้งนั้นต้องมีการเข้ารหัสด้วยเช่นกัน โดยการแปลงการตอบรับกลับที่ไม่มีการเข้ารหัส ให้เป็นการตอบรับกลับโดยใช้การเข้ารหัสนั้น แสดงได้ดังรูปที่ 4.30



รูปที่ 4.30 การแปลงการตอบรับให้อยู่ในรูปแบบการเข้ารหัส

จากรูปแสดงให้เห็นถึงการตอบรับซึ่งแบ่งไว้เป็น 2 ลักษณะใหญ่ๆ คือ การตอบรับเอชทีทีพีไดเจสผิดพลาด และการตอบรับเอชทีทีพีอื่นๆ การตอบรับในลักษณะแรกนั้นเกิดขึ้นใน

กรณีที่การร้องขอเอชทีทีพีไม่สามารถพิสูจน์ตนสำเร็จ จึงต้องส่งการร้องขอให้มีการพิสูจน์ตนด้วยข้อมูลท้าทายตามมาตรฐานเอชทีทีพี ส่วนการตอบรับเอชทีทีพีแบบอื่นๆ นั้นใช้ในกรณีที่การพิสูจน์ตนสำเร็จ ซึ่งหากการร้องขอมีการกำหนดให้ใช้การเข้ารหัสแล้ว การตอบรับต้องมีการเข้ารหัสด้วยเช่นเดียวกันดังรูปที่ 4.30 แต่หากเป็นการตอบรับกับการร้องขอที่มิได้มีการกำหนดให้ใช้การเข้ารหัส การตอบรับในครั้งนั้นๆ สามารถส่งข้อมูลไปได้ในทันทีโดยไม่ต้องมีการเข้ารหัส

บทที่ 5
ผลการทดสอบการทำงาน

5.1 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือสำหรับงานวิจัยนี้ประกอบไปด้วยโปรแกรม 4 โปรแกรมด้วยกัน คือ

5.1.1 โปรแกรมไดนามิกซี (Dynamic C)

เป็นโปรแกรมสำหรับการพัฒนาซอฟต์แวร์ภาษาซี และภาษาแอสเซมบลีสำหรับอุปกรณ์ฝังตัวอาร์ซีเอ็ม โดยลักษณะของการเขียนโปรแกรมนั้นคล้ายคลึงกับภาษาซีมาก แต่ยังมีข้อแตกต่างอยู่บ้างเล็กน้อย โดยเมื่อผู้พัฒนาเขียนซอฟต์แวร์สำหรับอุปกรณ์อาร์ซีเอ็มแล้ว ต้องผ่านการแปลโปรแกรม (compile) จากนั้นโปรแกรมไดนามิกซีจะรับผิดชอบสำหรับการบรรจุโปรแกรมลงบนอุปกรณ์อาร์ซีเอ็ม ซึ่งสามารถเลือกวิธีการในการบรรจุโปรแกรมได้ 2 แบบ คือ บรรจุโปรแกรมลงบนหน่วยความจำแฟลช (โปรแกรมยังคงอยู่เมื่อไม่มีไฟฟ้าจ่ายให้แก่อุปกรณ์) และการบรรจุโปรแกรมลงบนหน่วยความจำแรมสถิต แต่เนื่องจากหน่วยความจำแฟลชมีจำนวนครั้งของการบรรจุโปรแกรมที่จำกัด ดังนั้นวิธีการบรรจุโปรแกรมลงบนหน่วยความจำแรมสถิตจึงเป็นวิธีที่เหมาะสำหรับการตรวจหาข้อผิดพลาดในระหว่างขั้นตอนการพัฒนา

5.1.2 โปรแกรมเจบิวเดอร์ (Jbuilder)

ใช้สำหรับสร้างโปรแกรมจาวาแอปเพล็ตภายในส่วนเว็บ เพราะการสร้างส่วนต่อประสานกับผู้ใช้ทำได้โดยง่าย ซึ่งหลังจากการสร้างโปรแกรมเสร็จสมบูรณ์แล้ว ต้องนำแฟ้มข้อมูลคลาสทั้งหมด รวมถึงแฟ้มข้อมูลต่างๆ ที่จำเป็นต้องใช้ภายในส่วนเว็บมารวมกันให้เป็นแฟ้มข้อมูลประเภทจาร์

5.1.3 โปรแกรมสนิฟเฟอร์ (Sniffer)

เป็นเครื่องมือสำหรับการตรวจสอบข้อมูลที่รับส่งผ่านทางเครือข่ายอีเทอร์เนต เนื่องจากเมื่อข้อมูลถูกส่งผ่านเข้าไปในระบบเครือข่ายแล้ว หากไม่มีเครื่องมือสำหรับตรวจสอบข้อมูลอาจทำให้การแก้ไขข้อบกพร่องเป็นไปได้ยากยิ่งขึ้น ดังนั้นจึงจำเป็นต้องอาศัยการทำงานของโปรแกรมนี้ โดยการทำงานของโปรแกรมขึ้นอยู่กับข้อมูลแบบแพร่สัญญาณ (broadcast) ของอุปกรณ์ฮับ ซึ่งข้อมูลทุกชุดที่ถูกส่งมายังฮับจะถูกแพร่กระจายออกไปทุกๆ การเชื่อมต่อบนฮับ ดังนั้นต้องใช้โปรแกรมสนิฟเฟอร์บนเครื่องคอมพิวเตอร์ที่ต่ออยู่กับอุปกรณ์ฮับเดียวกันกับอุปกรณ์อาร์

ซีเอ็ม ซึ่งหากนำไปใช้กับเครื่องคอมพิวเตอร์ที่มิได้ต่ออยู่บนฮับเดียวกันกับอุปกรณ์อาร์ซีเอ็มแล้วจะไม่สามารถเรียกดูข้อมูลที่อุปกรณ์อาร์ซีเอ็มรับส่งกับเครือข่ายอีเทอร์เนตได้

5.1.4 โปรแกรม trapgen

เป็นโปรแกรมสำหรับจำลองข้อมูลเอสเอ็นเอ็มพีแทรพเพื่อส่งไปยังส่วนจัดการเอสเอ็นเอ็มพี โดยโปรแกรมสามารถกำหนดเลขที่อยู่ไอพีของเอสเอ็นเอ็มพีเอเจนต์ที่ต้องการสร้างพร้อมทั้งระบุชนิดของข้อมูลเอสเอ็นเอ็มพีแทรพได้

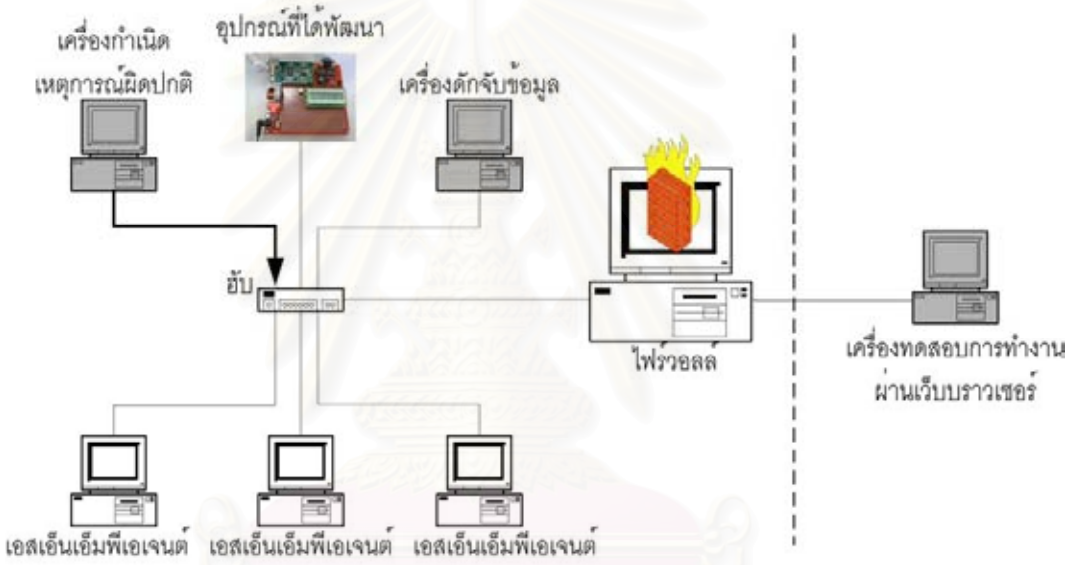
5.2 ขั้นตอนในการทดสอบ

5.2.1 ขั้นตอนการเตรียมอุปกรณ์

การนำอุปกรณ์ที่ได้พัฒนามาทดสอบนั้นต้องจัดเตรียมอุปกรณ์ต่างๆ ให้พร้อม โดยมีอุปกรณ์ที่สำคัญ ดังนี้

- เอสเอ็นเอ็มพีเอเจนต์อย่างน้อย 3 เครื่อง
- ไฟร์วอลล์ที่สามารถระบุพอร์ตทีซีพีไอพีเพื่อกำหนดการเข้าออกของข้อมูล โดยในการทดสอบนี้ได้นำเครื่องคอมพิวเตอร์ที่ลงโปรแกรมไฟร์วอลล์มาใช้
- เครื่องคอมพิวเตอร์สำหรับทดสอบการติดต่อกับระบบฝังตัวผ่านเว็บ ซึ่งประกอบไปด้วยเครื่องกำเนิดเหตุการณ์สำหรับสร้างเหตุการณ์ผิดปกติกับเครือข่ายทดสอบ และเครื่องทดสอบการทำงานผ่านเว็บบราวเซอร์
- อุปกรณ์ที่ได้พัฒนาขึ้น ตัวปรับแรงดันไฟฟ้า (Adapter) ที่สามารถแปลงแรงดันไฟฟ้าให้เป็นแรงดันไฟฟ้ากระแสตรงระดับ 9 ถึง 12 โวลต์ และสายเชื่อมต่อข้อมูลอนุกรมสำหรับการโปรแกรมลงบนอุปกรณ์
- เครื่องคอมพิวเตอร์ดักจับข้อมูล สำหรับตรวจสอบความถูกต้องของข้อมูลภายในเครือข่าย โดยต้องทำการลงโปรแกรมสนิฟเฟอร์ไว้
- ฮับสำหรับเชื่อมต่ออุปกรณ์ต่างๆ เข้าด้วยกัน

การต่ออุปกรณ์ต่างๆ เข้าด้วยกันแสดงได้ดังรูปที่ 5.1 โดยรูปแสดงให้เห็นถึงเอสเอ็นเอ็มพีเอเจนต์สามเครื่องด้วยกัน ซึ่งการต่อเครือข่ายนั้นเอสเอ็นเอ็มพีสามารถกระจายกันอยู่ตามเครือข่ายย่อยต่างๆ ภายในองค์กรได้โดยไม่จำเป็นต้องอยู่บนฮับเครื่องเดียวกัน เครื่องกำเนิดเหตุการณ์จะคอยสร้างสถานการณ์ที่ผิดปกติขึ้นกับเครือข่ายที่เอสเอ็นเอ็มพีเอเจนต์อยู่ อุปกรณ์ที่พัฒนาขึ้นมานั้นสามารถสื่อสารกับเอสเอ็นเอ็มพีเอเจนต์ได้เพื่อรับรู้ถึงปัญหาที่อาจเกิดขึ้น และผู้ดูแลระบบสามารถตรวจสอบเอสเอ็นเอ็มพีเอเจนต์ได้จากภายนอกเครือข่ายผ่านเครื่องทดสอบการทำงานถึงแม้ว่าไฟร์วอลล์จะปิดกั้นข้อมูลเอสเอ็นเอ็มพี โดยผู้ดูแลระบบติดต่อไปยังอุปกรณ์ที่ได้พัฒนาขึ้นผ่านทางเอชทีทีพี



รูปที่ 5.1 การต่ออุปกรณ์ต่างๆ สำหรับทำการทดลอง

5.2.2 ขั้นตอนการเตรียมส่วนโปรแกรม

จัดสร้างโปรแกรมซึ่งต้องมีความสามารถจัดการเครือข่ายผ่านเอสเอ็นเอ็มพีโดยแยกพัฒนาเป็น 2 ส่วน คือ โปรแกรมสำหรับระบบฝังตัวเพื่อใช้เป็นส่วนจัดการเครือข่ายเอสเอ็นเอ็มพีพร้อมทั้งมีความสามารถเป็นเครื่องบริการเว็บด้วย และโปรแกรมอีกส่วนหนึ่งทำงานผ่านเว็บบราวเซอร์เพื่อใช้เป็นส่วนต่อประสานผู้ใช้ โปรแกรมในส่วนแรกนั้นพัฒนาด้วยภาษาซี และแอสเซมบลี เมื่อจัดการแปลโปรแกรมให้อยู่ในลักษณะที่ระบบฝังตัวเข้าใจได้แล้วนำเก็บโปรแกรมไว้ในหน่วยความจำแฟลช โปรแกรมในส่วนที่ 2 นั้นสร้างจากภาษาจาวาและทำการแปลโปรแกรมให้อยู่ในรูปแฟ้มข้อมูลคลาส แต่เนื่องจากมีแฟ้มข้อมูลคลาสจำนวนมากจึงส่งผลให้ไม่สะดวกสำหรับการจัดเก็บลงในหน่วยความจำแฟลช ดังนั้นต้องรวบรวมแฟ้มข้อมูลคลาสทั้งหมดพร้อมทั้ง

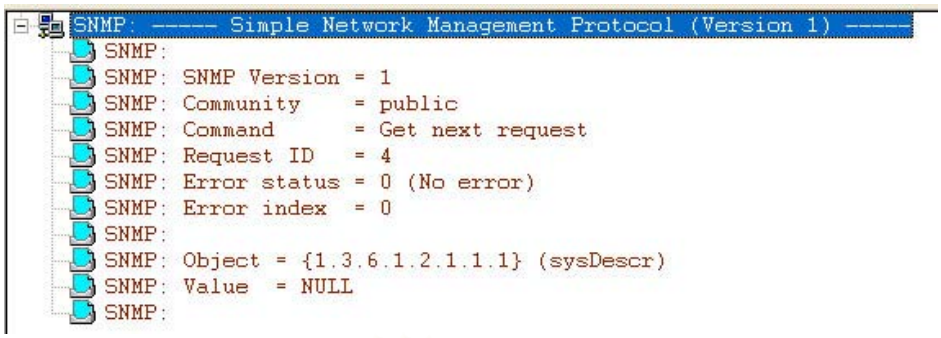
แฟ้มข้อมูลอื่นๆ ที่จำเป็นนำมาสร้างเป็นแฟ้มข้อมูลประเภทจาร์เพียงแฟ้มข้อมูลเดียว ด้วยการใช้เครื่องมือช่วยของจาวาโดยใช้คำสั่ง “jar –cvf [jar-file] files” ซึ่งต้องระบุชื่อแฟ้มข้อมูลจาร์ใหม่ที่ต้องการสร้าง พร้อมกับรายชื่อแฟ้มข้อมูลทั้งหมดที่ต้องการเก็บรวบรวมไว้ในแฟ้มข้อมูลจาร์ ยกตัวอย่างเช่น หากต้องการสร้างแฟ้มข้อมูลจาร์ชื่อ “AllClass.jar” สำหรับรวบรวมแฟ้มข้อมูลประเภทคลาสทุกแฟ้มข้อมูล สามารถสร้างแฟ้มข้อมูลจาร์ได้ด้วยคำสั่ง “c:\>jar –cvf AllClass.jar *.class” สำหรับแอปเพล็ตโดยทั่วไปนั้นผู้พัฒนาสามารถนำแฟ้มข้อมูลจาร์ที่ได้เก็บลงในหน่วยความจำแฟลชได้ทันที แต่หากแอปเพล็ตมีคำสั่งซึ่งไม่อนุญาตสำหรับแอปเพล็ตทั่วๆ ไปเช่น คำสั่งสำหรับการเปิดแฟ้มข้อมูล หรือคำสั่งสำหรับการเปิดการเชื่อมต่อไปยังเครื่องบริการเว็บอื่นนอกเหนือจากเครื่องบริการเว็บที่แอปเพล็ตติดต่ออยู่ ผู้พัฒนาต้องทำการรับรองแอปเพล็ตนั้นด้วยใบรับรอง (Certificate) ซึ่งประกอบด้วย 2 ขั้นตอนดังนี้

- สร้างใบรับรองโดยใช้คำสั่ง “c:\>keytool –selfcert –genkey –keystore <keystorename> -alias <aliasname> -keypass <keypass>” ซึ่งเป็นการสร้างใบรับรองส่วนบุคคล
- รับรองแฟ้มข้อมูลจาร์ด้วยคำสั่ง “c:\>jarsigner –keystore <url> -storepass <password> -keypass <password> jar-file โดยใช้ใบรับรองที่ได้จากคำสั่งก่อนหน้านี้สำหรับการรับรอง

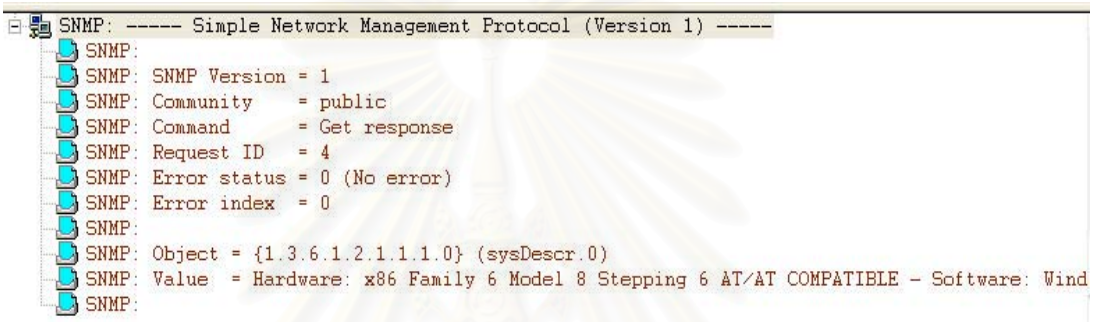
5.3 ผลการทดสอบ

5.3.1 ผลการทดสอบ การติดต่อกับเอสเอ็นเอ็มพีเอเจนต์

เมื่ออุปกรณ์ทุกอย่างถูกติดตั้งเสร็จสิ้นแล้วเปิดการทำงานของอุปกรณ์ฝังตัว ระบบจะทำการตรวจสอบอุปกรณ์ทั้งหมดภายในช่วงเลขที่อยู่ไอพีเพื่อตรวจหาเอสเอ็นเอ็มพีเอเจนต์ทั้งหมด ซึ่งภายในขั้นตอนนี้สามารถตรวจสอบการติดต่อกับเอสเอ็นเอ็มพีเอเจนต์ได้โดยการจับดูข้อมูลจากเครื่องดักจับข้อมูล ซึ่งจากผลการทดสอบนั้นอุปกรณ์ฝังตัวสามารถส่งข้อมูลเอสเอ็นเอ็มพีไปยังเอสเอ็นเอ็มพีเอเจนต์ได้ดังรูปที่ 5.2 เป็นการร้องขอค่าถัดไปของเอสเอ็นเอ็มพีเอเจนต์ หากเลขที่อยู่ไอพีใดเปิดการทำงานเอสเอ็นเอ็มพีเอเจนต์จะมีข้อมูลเอสเอ็นเอ็มพีส่งกลับมายังอุปกรณ์ฝังตัวดังรูปที่ 5.3



รูปที่ 5.2 ข้อมูลเอสเอ็นเอ็มพีที่ถูกส่งออกจากอุปกรณ์ฝังตัว



รูปที่ 5.3 ข้อมูลเอสเอ็นเอ็มพีที่ได้รับกลับมายังอุปกรณ์ฝังตัว

5.3.2 ผลการทดสอบ การใช้เว็บบราวเซอร์เป็นส่วนติดต่อกับอุปกรณ์ฝังตัว

ผู้ดูแลระบบสามารถติดต่อไปยังอุปกรณ์ฝังตัวผ่านทางเว็บบราวเซอร์ได้ทั้งจากภายใน และภายนอกไฟร์วอลล์ ผลของการเรียกใช้งานผ่านทางเว็บแสดงได้ดังรูปที่ 5.4 ซึ่งประกอบไปด้วยแถบการทำงานต่างๆ 5 แถบด้วยกันดังนี้

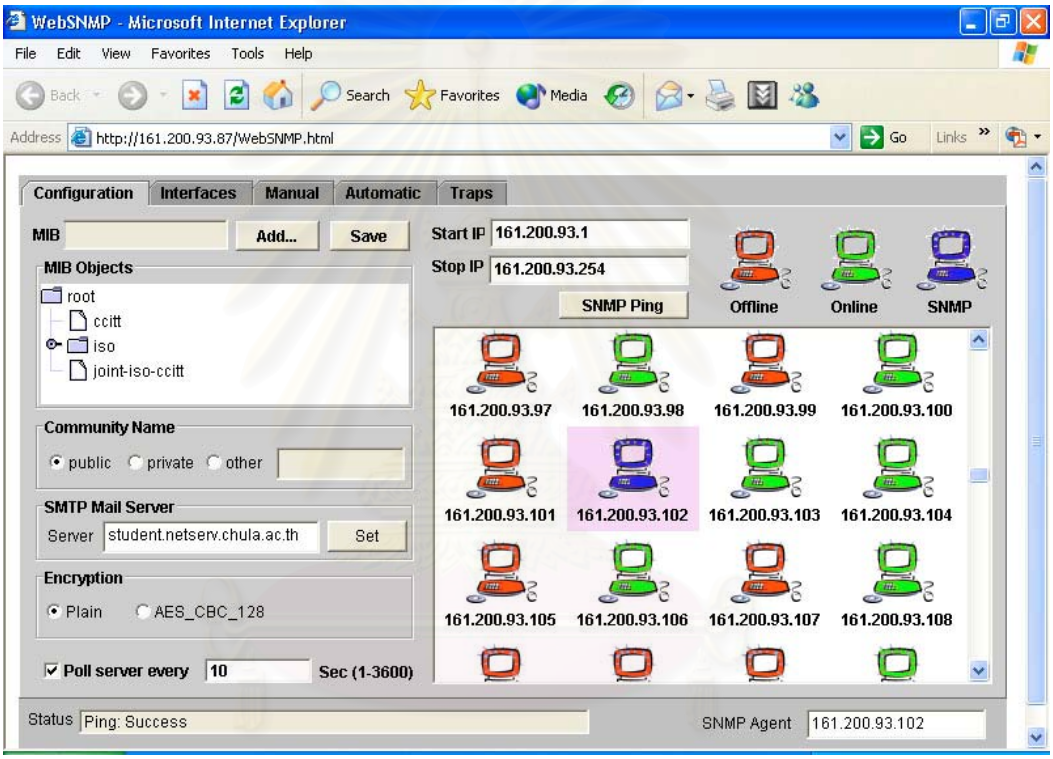
- โครงแบบ (Configuration) ใช้สำหรับการกำหนดค่าต่างๆ ให้กับอุปกรณ์ฝังตัว ประกอบด้วยเอ็มไอบีมาตรฐาน คอมมิวนิตี้สตริง เครื่องบริการเอสเอ็มทีพี ช่วงของเลขที่อยู่ไอพี

- ส่วนต่อประสานเครือข่าย (Interfaces) สำหรับตรวจสอบค่าต่างๆ ของส่วนต่อประสานแต่ละตัวของเอสเอ็นเอ็มพีเอเจนต์ ผู้ดูแลระบบสามารถตรวจสอบเอเจนต์ที่กำลังติดต่ออยู่ได้ว่า มีส่วนต่อประสานเครือข่ายอยู่ทั้งหมดเท่าใด

- การจัดการด้วยมือ (Manual) ผู้ดูแลระบบสามารถตรวจสอบค่าของเอสเอ็นเอ็มพีเอเจนต์ได้โดยตรง ซึ่งการเปลี่ยนเอสเอ็นเอ็มพีเอเจนต์ที่ต้องการติดต่อต้องกระทำผ่านส่วนโครงแบบเท่านั้น

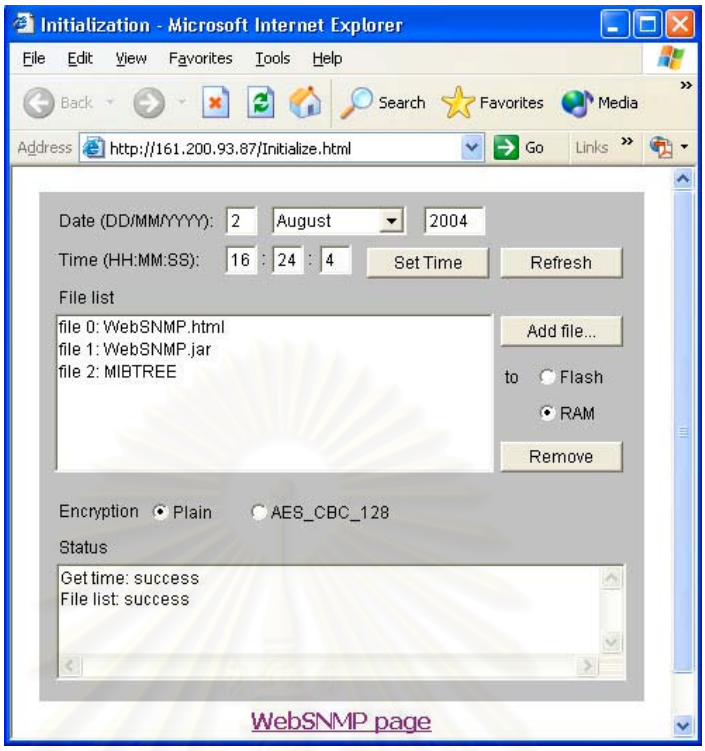
- การจัดการแบบอัตโนมัติ (Auto) ช่วยให้ผู้ดูแลกำหนดเหตุการณ์ที่คาดว่าอาจทำให้เครือข่ายทำงานไม่สมบูรณ์ พร้อมทั้งค่าข้อมูลที่ต้องการเปลี่ยนแปลงหากมีเหตุการณ์เกิดขึ้น

- ข้อมูลเอสเอ็นเอ็มพีแทรพ (Traps) แสดงผลข้อมูลแทรพซึ่งอาจบ่งบอกถึงการเกิดเหตุการณ์ผิดปกติกับเอสเอ็นเอ็มพีเอเจนต์ หรือเหตุการณ์อื่นๆ ที่ถูกส่งมาจากเอสเอ็นเอ็มพีเอเจนต์



รูปที่ 5.4 ส่วนติดต่อผู้ใช้ผ่านทางเว็บบราวเซอร์

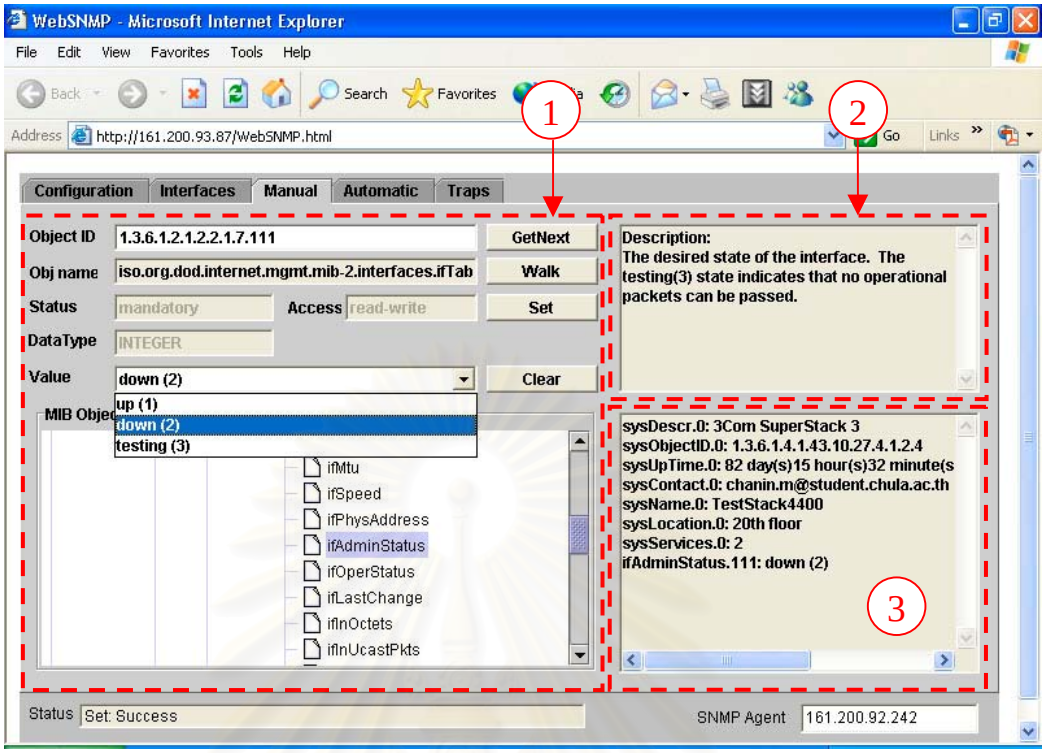
ในการใช้งานนั้นหากผู้พัฒนาต้องการเปลี่ยนส่วนของการติดต่อผ่านเว็บบราวเซอร์ หรือต้องการแก้ไขบางส่วนของแอปเพล็ต ผู้พัฒนาต้องทำการแปลโปรแกรมใหม่ทั้งหมดและนำส่วนติดต่อผ่านเว็บเก็บลงในหน่วยความจำแฟลชซึ่งไม่สะดวกต่อการใช้งานจริง ดังนั้นในส่วนของการติดต่อผ่านเว็บจึงมีโปรแกรมขนาดเล็กสำหรับการเปลี่ยนหน้าจอการติดต่อหลักไว้ด้วยดังแสดงได้ตามรูปที่ 5.5 ซึ่งผู้ดูแลระบบสามารถเลือกพื้นที่สำหรับการเก็บแฟ้มข้อมูลได้ทั้งในแฟลช หรือเก็บในหน่วยความจำแรม (กรณีต้องการทดสอบการทำงาน) นอกจากนี้แล้วภายในส่วนนี้ยังใช้สำหรับการกำหนดเวลาภายในระบบฝังตัวด้วย



รูปที่ 5.5 การเปลี่ยนส่วนติดต่อหลัก

5.3.3 ผลการทดสอบการร้องขอค่าข้อมูล และการกำหนดข้อมูล

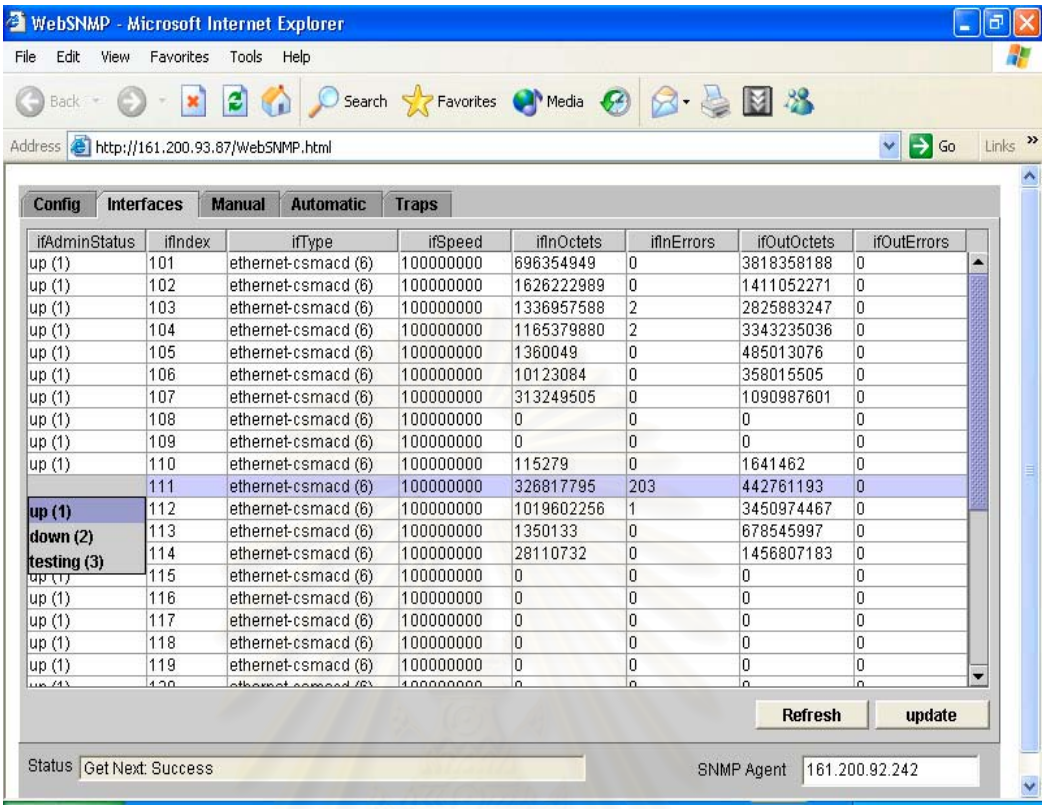
ผู้ดูแลสามารถตรวจสอบค่าของเอสเอ็นเอ็มพีเอเจนต์ได้โดยตรง แต่การเลือกเอสเอ็นเอ็มพีเอเจนต์ต้องเลือกจากส่วนโครงแบบ เช่นเดียวกับการกำหนดคอมมิวนิตี้สตริงต้องกำหนดในหน้าของส่วนโครงแบบ การทดสอบการร้องขอค่าข้อมูล และการกำหนดค่าข้อมูลสามารถแสดงได้ดังรูป 5.6 ซึ่งประกอบไปด้วย 3 ส่วนโดยส่วนที่ 1 สำหรับการกำหนดความต้องการของผู้ใช้ โดยผู้ใช้สามารถเลือกอ็อบเจกต์ด้วยแผนภาพต้นไม้ หรือพิมพ์หมายเลขโอไอดีโดยตรงได้เอง จากนั้นผู้ใช้เลือกคำสั่งที่ต้องการสั่งงานจากปุ่มต่างๆ ส่วนที่ 2 แสดงคำอธิบายรายละเอียดของอ็อบเจกต์นั้นๆ และส่วนที่ 3 แสดงค่าการตอบรับจากเอสเอ็นเอ็มพีเอเจนต์ โดยการทดลองนี้แสดงให้เห็นการร้องขอข้อมูลเป็นชุดจากกลุ่มข้อมูลระบบ (อ็อบเจกต์ที่ขึ้นต้นด้วย iso.org.dod.internet.mgmt.mib-2.system) และการกำหนดข้อมูลสั่งปิดการทำงานส่วนต่อประสานเครือข่ายหมายเลข 111 (iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus.111)



รูปที่ 5.6 แสดงแถบการทำงานสำหรับการร้องขอค่า หรือการกำหนดค่าเอสเอ็นเอ็มพีเอเจนต์

5.3.4 ผลการทดสอบการตรวจส่วนต่อประสานเครือข่าย

แสดงรายละเอียดของส่วนต่อประสานเครือข่ายของอุปกรณ์เอสเอ็นเอ็มพี แสดงได้ดังรูป 5.7 จำนวนของบรรทัดข้อมูลบ่งบอกถึงจำนวนของส่วนต่อประสานเครือข่าย โดยที่ข้อมูลแต่ละค่ามีความหมายที่แตกต่างกัน ดังแสดงไว้ในตาราง 5.1 ในการทดสอบนี้ได้ตรวจดูข้อมูลของส่วนต่อประสานทั้งหมดของอุปกรณ์สวิทช์ทดสอบ ซึ่งสามารถระบุค่าต่างๆ ได้อย่างถูกต้องเมื่อเทียบกับชุดข้อมูลเอสเอ็นเอ็มพีที่ใช้โปรแกรมสนิฟเฟอร์ในการตรวจสอบ นอกจากนี้แล้วระบบยังอนุญาตให้ผู้ใช้งานมีการเปิด หรือปิดการทำงานของส่วนต่อประสานเครือข่ายใดๆ ได้อย่างสะดวก ซึ่งเป็นตัวอย่างของการกำหนดค่าส่วนต่อประสานเครือข่ายของอุปกรณ์สวิทช์ โดยผู้ดูแลระบบควรกำหนดคอมมิวนิตี้สตริงภายในหน้าโครงแบบให้ถูกต้องเสียก่อน หากผู้ใช้งานไม่ทราบถึงคอมมิวนิตี้สตริงที่ถูกต้องของอุปกรณ์เอสเอ็นเอ็มพีนั้น ระบบจะไม่สามารถเปลี่ยนสถานะการทำงานของส่วนต่อประสานเครือข่ายนั้นได้



รูปที่ 5.7 แสดงส่วนต่อประสานเครือข่าย

ตารางที่ 5.1 แสดงข้อมูลต่างๆ ของส่วนต่อประสานเครือข่าย

ชื่อข้อมูล	ความหมายของข้อมูล
IfAdminStatus	สถานะการทำงาน ซึ่งผู้ใช้สามารถเปลี่ยนแปลงได้
IfIndex	หมายเลขของส่วนต่อประสาน ใช้สำหรับการอ้างอิงถึงส่วนต่อประสานนั้นๆ โดยระบุหมายเลขอ็อบเจกต์ตามด้วยเครื่องหมายมหัพภาค (.) และต่อท้ายด้วยหมายเลขส่วนต่อประสาน
IfType	ชนิดของส่วนต่อประสาน
IfSpeed	ความเร็วในการรับส่งข้อมูลของส่วนต่อประสาน
IfInOctets	จำนวนข้อมูลรับเข้า (หน่วยเป็นไบต์)
IfInErrors	จำนวนชุดข้อมูลรับเข้าที่ผิดพลาด (หน่วยเป็นเฟรม)
IfOutOctets	จำนวนข้อมูลส่งออก (หน่วยเป็นไบต์)
IfOutErrors	จำนวนชุดข้อมูลส่งออกที่ผิดพลาด (หน่วยเป็นเฟรม)

5.3.5 ผลการทดสอบ การจัดการเหตุการณ์แบบอัตโนมัติ

ส่วนจัดการแบบอัตโนมัตินี้ตรวจสอบเหตุการณ์จากเอสเอ็นเอ็มพีเอเจนต์ต้นทาง (Source) สังเกตได้จากส่วนที่ 1 ของรูป 5.8 และกำหนดค่าข้อมูลให้กับเอสเอ็นเอ็มพีเอเจนต์ปลายทาง (Destination) ซึ่งเป็นส่วนที่ 2 ของรูป 5.8 โดยข้อมูลต่างๆ มีคำอธิบายดังตาราง 5.2 การกำหนดอุปกรณ์ต้นทาง และอุปกรณ์ปลายทางนั้นไม่จำเป็นต้องเป็นอุปกรณ์เอสเอ็นเอ็มพีที่กำลังติดต่ออยู่ แต่เป็นอุปกรณ์เอสเอ็นเอ็มพีใดๆ ก็ได้ ซึ่งวิธีการของการตรวจสอบการเกิดเหตุการณ์คือ ระบบจะร้องขอค่าอ็อบเจกต์ของอุปกรณ์ต้นทางมาเป็นค่าตั้งต้นเพื่อนำมาเปรียบเทียบกับค่าใน srcVal ซึ่งเป็นค่าเปรียบเทียบ โดยใช้เงื่อนไขตาม Cond ที่กำหนด จากรูปแสดงให้เห็นตัวอย่างของเหตุการณ์ที่กำหนดพร้อมทั้งการจัดการไว้ 4 เหตุการณ์ดังนี้

เหตุการณ์ที่ 1: แสดงให้เห็นการตรวจสอบข้อมูลซึ่งเป็นข้อมูลตัวอักษร หากชื่อ (iso.org.dod.internet.mgmt.mib-2.system.sysName.0) ของสวิตช์ (161.200.92.242) เท่ากับ “TestStack4400” ไม่ต้องจัดการใดๆ แต่ต้องแจ้งเตือน

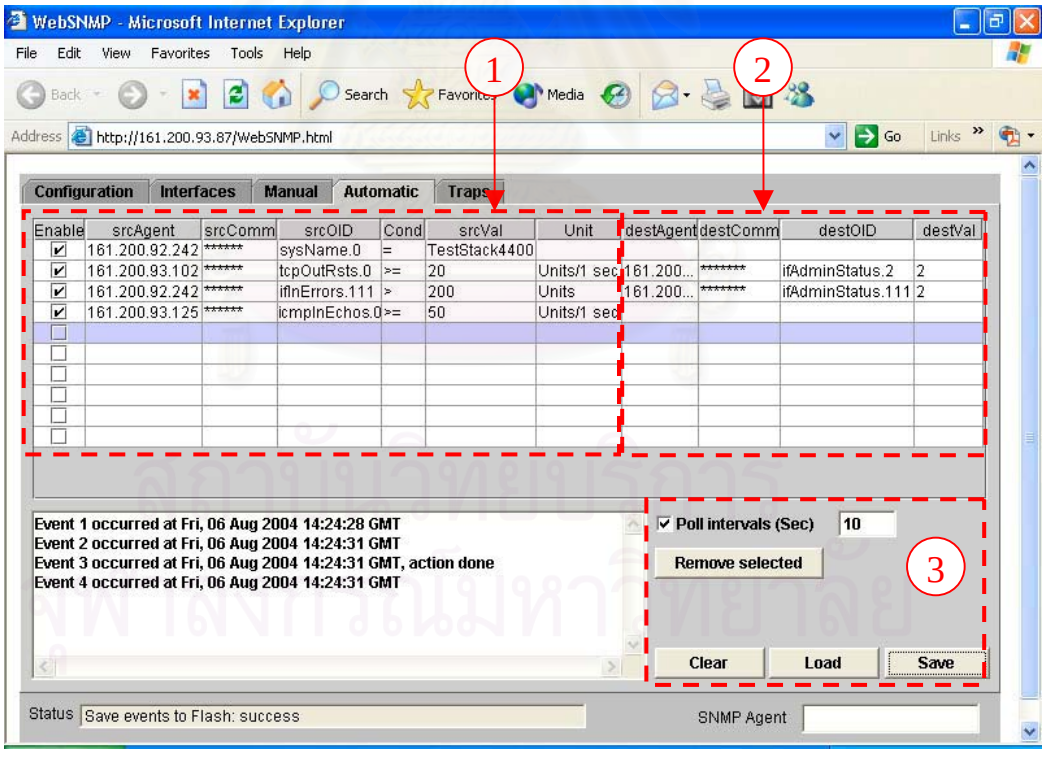
เหตุการณ์ที่ 2: แสดงให้เห็นการตรวจสอบเหตุการณ์ลักษณะการกราดตรวจการเปิดพอร์ต (Port scan) ซึ่งหลังจากมีการกราดตรวจนั้น เอสเอ็นเอ็มพีเอเจนต์จะส่งข้อมูลพร้อมแฟลก “RST” หากมีจำนวนข้อมูลส่งออกซึ่งมีแฟลกนี้มากผิดปกติแสดงว่ามีการกราดตรวจการเปิดพอร์ต โดยแสดงให้เห็นการตรวจจำนวนข้อมูลส่งออกที่มีแฟลก “RST” (iso.org.dod.internet.mgmt.mib-2.tcp.tcpOutRsts.0) มากกว่าหรือเท่ากับ 20 หน่วยต่อวินาที ให้จัดการปิด (ค่าเป็น 2) ส่วนต่อประสานหมายเลข 2 (iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus.2)

เหตุการณ์ที่ 3: แสดงให้เห็นการตรวจเหตุการณ์ของสวิตช์ทดสอบ (161.200.92.242) มีข้อมูลรับเข้าของส่วนต่อประสานหมายเลข 111 ผิดพลาด (iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInErrors.111) มากกว่า 200 หน่วย ให้จัดการปิดส่วนต่อประสานนั้น

เหตุการณ์ที่ 4: แสดงให้เห็นการตรวจเหตุการณ์ประเภทการโจมตีด้วยการตรวจสถานะ (ping flood) หากเกิดเหตุการณ์มีข้อมูลประเภทการร้องขอ

ไอซีเอ็มพีเอ็กโค (iso.org.dod.internet.mgmt.mib-2.icmp.icmpInEchos.0) ของเอสเอ็นเอ็มพีเอเจนต์ 161.200.93.125) มากกว่าหรือเท่ากับ 50 หน่วยต่อวินาที ไม่ต้องจัดการใดๆ เพียงแต่ต้องแจ้งเตือน

การตั้งเหตุการณ์ทั้งหมดจะมีผลให้ทำงานต่อเมื่อได้กำหนดให้ตรวจสอบเหตุการณ์นั้นๆ โดยการเลือกเปิดการตรวจสอบ (Enable) สำหรับเหตุการณ์นั้นๆ แต่ทั้งนี้แล้วยังถูกควบคุมด้วยการตรวจสอบหลักอีกชั้นหนึ่งซึ่งต้องเลือกเปิดการวนตรวจสอบ (Poll intervals) โดยจะมีผลกับทุกๆ เหตุการณ์ ซึ่งระบบฝังตัวจะวนตรวจสอบทุกๆ ช่วงเวลาที่กำหนด ผู้ดูแลระบบสามารถทำการบันทึกเหตุการณ์ลงในอุปกรณ์ฝังตัวโดยการกดปุ่มบันทึก (Save) ซึ่งปรากฏอยู่ในส่วนที่ 3 ของรูป 5.8 ในทางตรงข้ามผู้ใช้งานสามารถบรรจุเหตุการณ์ในส่วนจัดการอัตโนมัติได้จากปุ่มบรรจุ (Load) นอกจากนี้แล้วผู้ใช้ยังสามารถลบเหตุการณ์ใดๆ ออกไปจากรายการได้โดยง่ายด้วยการกดปุ่มนำเหตุการณ์ที่เลือกออก (Remove selected)



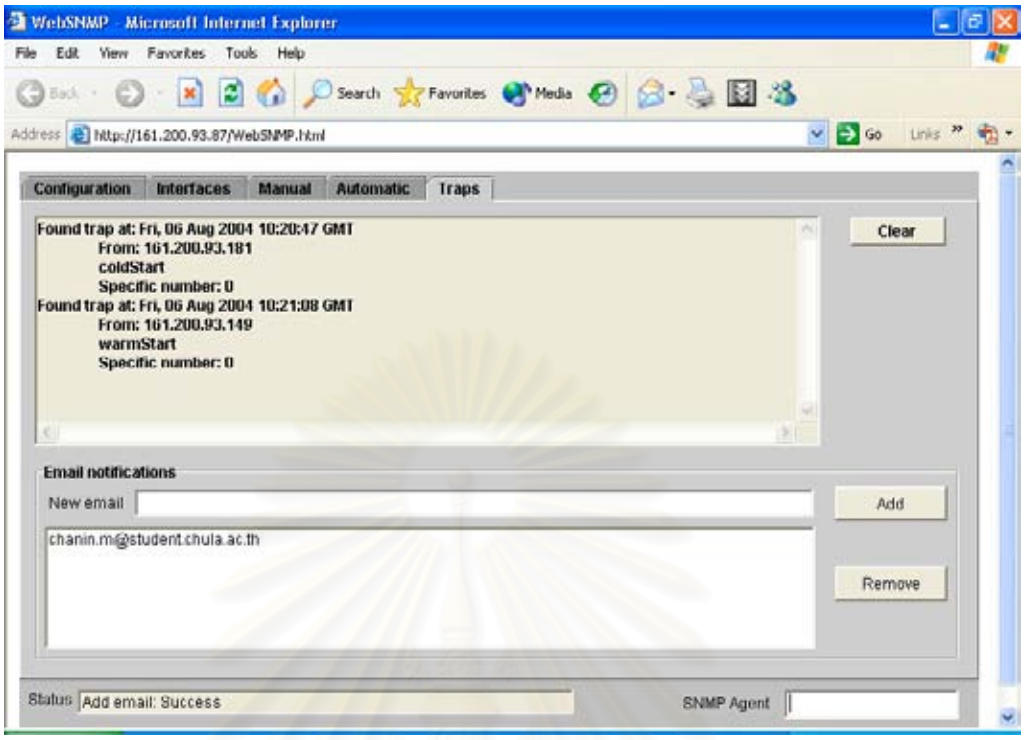
รูปที่ 5.8 แสดงส่วนจัดการแบบอัตโนมัติ

ตารางที่ 5.2 แสดงความหมายของข้อมูลต่างๆ ภายในส่วนจัดการแบบอัตโนมัติ

ชื่อข้อมูล	ความหมายของข้อมูล
Enable	สำหรับการเปิดและปิดการตรวจสอบเหตุการณ์นั้นๆ
srcAgent	เลขที่อยู่ไอพีของอุปกรณ์ต้นทาง
srcComm	คอมมิวนิตี้สตริงของอุปกรณ์ต้นทาง
srcOID	หมายเลขโอไอดีที่ต้องการตรวจสอบ
Cond	เงื่อนไขในการตรวจสอบซึ่งมีค่าที่เป็นไปได้คือ <, <=, =, >= และ >
srcVal	ค่าที่จะถูกนำไปเป็นค่าตรวจสอบเทียบกับค่าที่ได้จากอุปกรณ์ต้นทาง
destAgent	เลขที่อยู่ไอพีของอุปกรณ์ปลายทาง
DestComm	คอมมิวนิตี้สตริงของอุปกรณ์ปลายทาง
DestOID	หมายเลขโอไอดีที่ต้องการเปลี่ยนของอุปกรณ์ปลายทาง
DestVal	ค่าที่ต้องการเปลี่ยน

5.3.6 ผลการทดสอบข้อมูลเอสเอ็นเอ็มพีแทรพ

ส่วนแสดงผลข้อมูลเอสเอ็นเอ็มพีแทรพสามารถบ่งบอกได้ถึงการเกิดเหตุการณ์ซึ่งต้องแจ้งเตือนโดยได้กำหนดให้เอสเอ็นเอ็มพีเอเจนต์ทดสอบทุกๆ ตัวส่งข้อมูลเอสเอ็นเอ็มพีแทรพมายังอุปกรณ์ฝังตัว และนอกจากนั้นแล้วยังได้ใช้โปรแกรม “trapgen” สร้างข้อมูลเอสเอ็นเอ็มพีแทรพจำลองส่งมายังอุปกรณ์ฝังตัวด้วย สามารถแสดงการตรวจพบข้อมูลเอสเอ็นเอ็มพีแทรพได้ดังรูปที่ 5.9 โดยข้อมูลดังกล่าวจะถูกส่งต่อไปยังบัญชีเลขที่อยู่อีเมล (e-mail address) ทั้งหมดที่ได้ลงทะเบียนไว้กับระบบฝังตัว โดยส่งไปยังเครื่องบริการเอสเอ็มทีพี (SMTP server) ที่ได้ระบุไว้ภายในส่วนของโครงแบบ เพื่อให้เครื่องบริการเอสเอ็มทีพีจัดส่งในลำดับถัดไป



รูปที่ 5.9 แสดงส่วนแสดงผลข้อมูลเอสเอ็นเอ็มพีแทรพ

5.3.7 ผลการทดสอบการทำงานผ่านไฟร์วอลล์

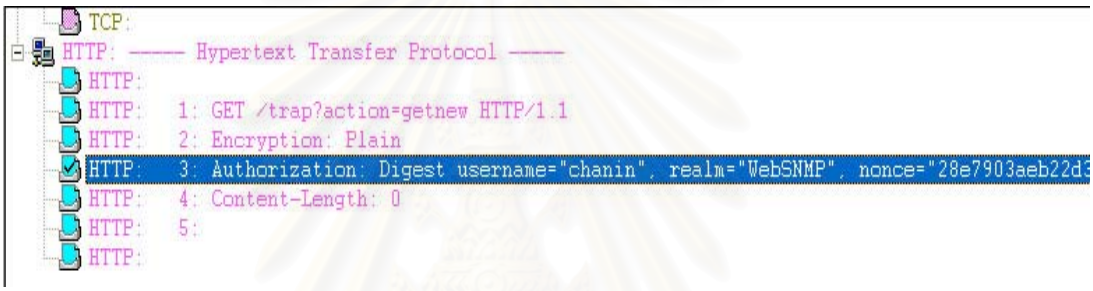
ไฟร์วอลล์ที่ใช้ในการทดลองได้ทำการปิดกั้นข้อมูลเอสเอ็นเอ็มพีมิให้ไหลเข้าหรือออกจากเครื่องทดสอบการทำงานผ่านเว็บบราวเซอร์ได้ ดังนั้นหากมีโปรแกรมใดภายในเครื่องทดสอบการทำงานผ่านเว็บบราวเซอร์ซึ่งติดต่อโดยใช้ข้อตกลงเอสเอ็นเอ็มพี โปรแกรมนั้นจะไม่สามารถค้นหาอุปกรณ์เอสเอ็นเอ็มพีเอเจนต์ได้เลย เสมือนกับเป็นเครื่องคอมพิวเตอร์จากภายนอกองค์กรที่ไม่สามารถติดต่อสื่อสารมายังเครือข่ายภายในองค์กรโดยใช้เอสเอ็นเอ็มพี จากการทดสอบข้อมูลที่ไหลจากเครื่องทดสอบการทำงานผ่านเว็บบราวเซอร์ผ่านไฟร์วอลล์มายังระบบฝังตัวในระดับชั้นโปรแกรมประยุกต์นั้นเป็นข้อมูลเอชทีทีพีทั้งสิ้นดังรูปที่ 5.10

	No.	Status	Source Address	Dest Address	Summary	Len (B
☐	1	M	[192.168.0.10]	[161.200.93.87]	HTTP: C Port=1300 GET /WebSNMP.html HTTP/1.1	501
☐	2		[161.200.93.87]	[192.168.0.10]	HTTP: R Port=1300 HTTP/1.1 Status=Not Modified	206
☐	3		[192.168.0.10]	[161.200.93.87]	HTTP: C Port=1301 GET /WebSNMP.jar HTTP/1.1	339
☐	4		[161.200.93.87]	[192.168.0.10]	HTTP: R Port=1301 HTTP/1.1 Status=Not Modified	208
☐	5		[192.168.0.10]	[161.200.93.87]	HTTP: C Port=1301 GET /Initialize.jar HTTP/1.1	342
☐	6		[161.200.93.87]	[192.168.0.10]	HTTP: R Port=1301 HTTP/1.1 Status=Not Modified	208
☐	7		[192.168.0.10]	[161.200.93.87]	HTTP: C Port=1302 GET /snmp?action=get&ver=1&c	148
☐	8		[161.200.93.87]	[192.168.0.10]	HTTP: R Port=1302 HTTP/1.1 Status=Unauthorized	287

รูปที่ 5.10 ข้อมูลเอชทีทีพีที่ไหลผ่านไฟร์วอลล์

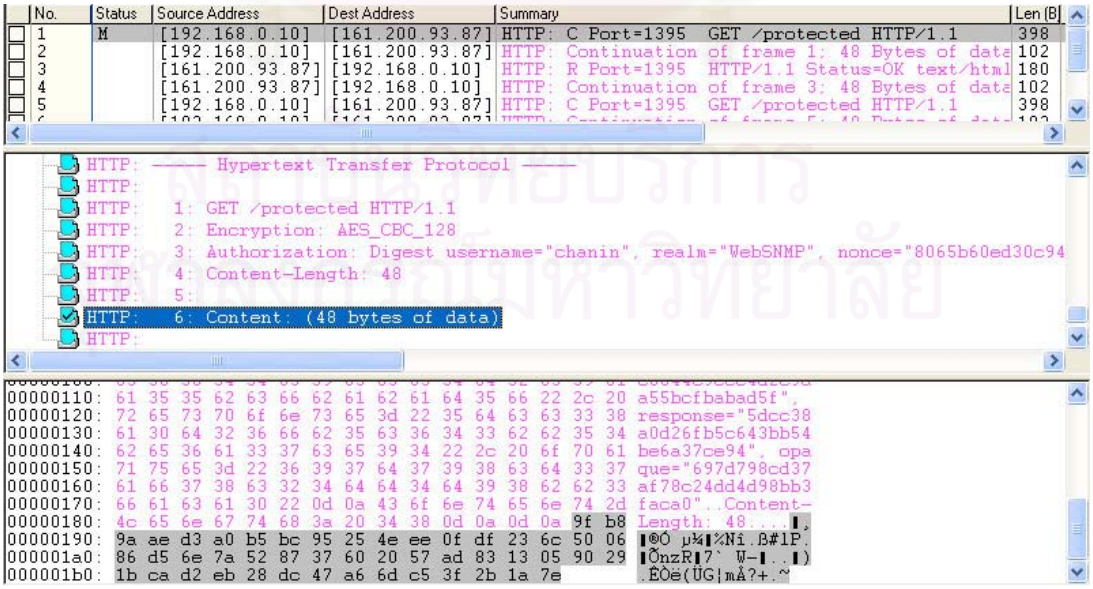
5.3.8 ผลการทดสอบ การทำงานผ่านระบบรักษาความปลอดภัย

จากผลการทดสอบการทำงานผ่านไฟร์วอลล์เห็นได้ว่าข้อมูลเอชทีทีพีที่ติดต่อกันนั้นเป็นข้อมูลประเภทที่มนุษย์สามารถอ่านได้ ซึ่งไม่ปลอดภัยในแง่การนำมาใช้งานจริง ดังนั้นระบบที่พัฒนาได้เพิ่มส่วนของระบบรักษาความปลอดภัยให้สามารถเข้ารหัส และถอดรหัสข้อมูลระหว่างการติดต่อได้โดยอาศัยการเข้ารหัสเออีเอสขนาด 128 บิต ผลการทดสอบนั้นในขั้นตอนของการพิสูจน์ตนระบบฝังตัวถามรหัสผ่านกับผู้ใช้ 3 ครั้ง หากภายใน 3 ครั้งไม่สามารถใส่รหัสผ่านที่ถูกต้องได้ระบบฝังตัวจะตัดการติดต่อในครั้งนั้นทันที ซึ่งระหว่างการพิสูจน์ตนจนกระทั่งการพิสูจน์ตนเสร็จสิ้นข้อมูลรหัสผ่านมิได้ถูกส่งไปในเครือข่ายเลยแสดงได้ดังรูปที่ 5.11



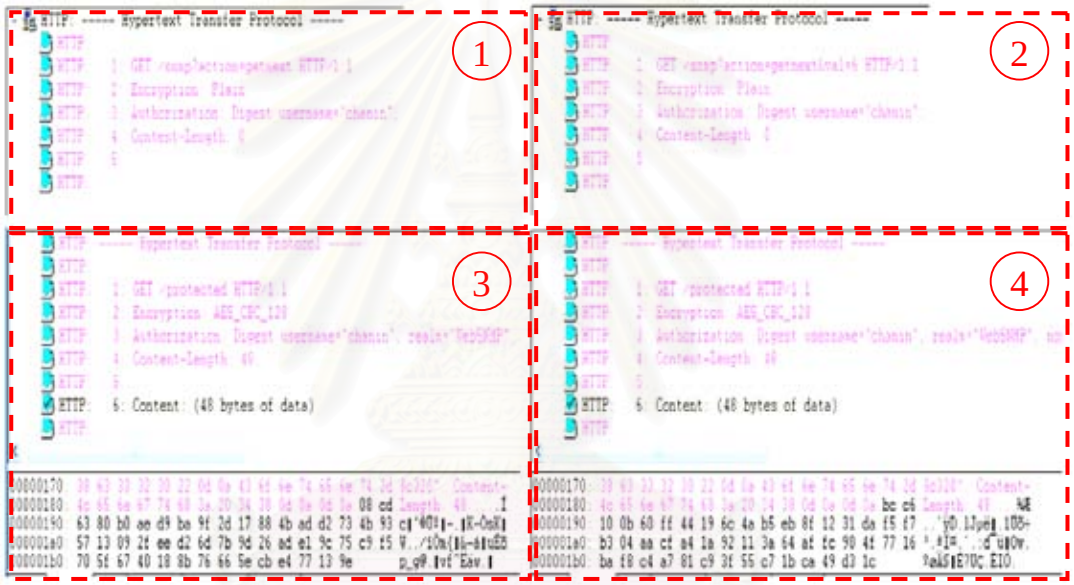
รูปที่ 5.11 ข้อมูลการติดต่อระหว่างขั้นตอนการพิสูจน์ตน

ภายหลังจากการพิสูจน์ตนเสร็จสิ้นข้อมูลต่างๆ ซึ่งเป็นการสื่อสารระหว่างระบบฝังตัว และเครื่องทดสอบการทำงานผ่านเว็บบราวเซอร์ถูกเข้ารหัสด้วยเออีเอสดังรูปที่ 5.12



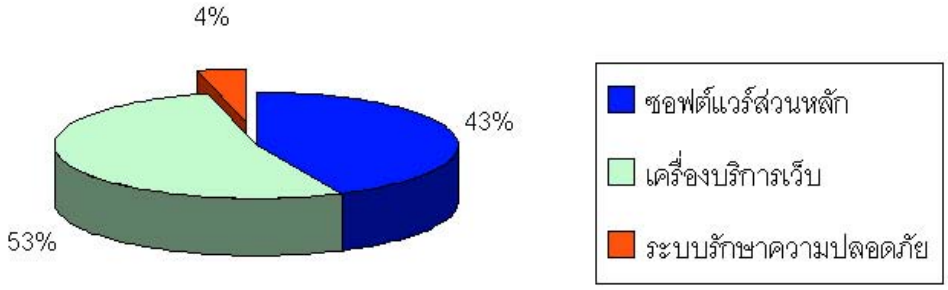
รูปที่ 5.12 ข้อมูลติดต่อระหว่างระบบฝังตัว และเครื่องทดสอบผ่านเว็บบราวเซอร์ที่ผ่านการเข้ารหัส

ในระหว่างการติดต่อกับระบบฝังตัว และเครื่องทดสอบผ่านเว็บบราวเซอร์ในทุกๆ ชุดข้อมูลที่มีการติดต่อ เวกเตอร์ตั้งต้นเป็นค่าข้อมูลสุ่มแสดงได้ดังรูปที่ 5.13 ดังนั้นถึงแม้ข้อมูลชุดเดียวกันถูกส่งติดต่อกัน แต่ข้อมูลที่ถูกเข้ารหัสแล้วจะไม่เหมือนกัน จากรูปในส่วนที่ 1 แสดงให้เห็นถึงการร้องขอข้อมูลจำนวนการเชื่อมต่อของทีซีพี ณ.ขณะนั้น (iso.org.dod.internet.mgmt.mib-2.tcp.tcpCurrEstab.0) โดยส่งเป็นข้อมูลที่มิได้เข้ารหัส 2 ครั้งเห็นได้ว่าข้อมูลที่มิได้ผ่านการเข้ารหัสนั้นมีข้อมูลต่างๆ ที่เหมือนกัน ในขณะเดียวกันหากกำหนดให้มีการเข้ารหัสแล้ว และร้องขอข้อมูลชุดเดียวกันอีก 2 ครั้ง ข้อมูลดังกล่าวจะไม่เหมือนกันเนื่องจากเวกเตอร์ตั้งต้นที่เปลี่ยนไป ดังแสดงในส่วนที่สองของรูป 5.13

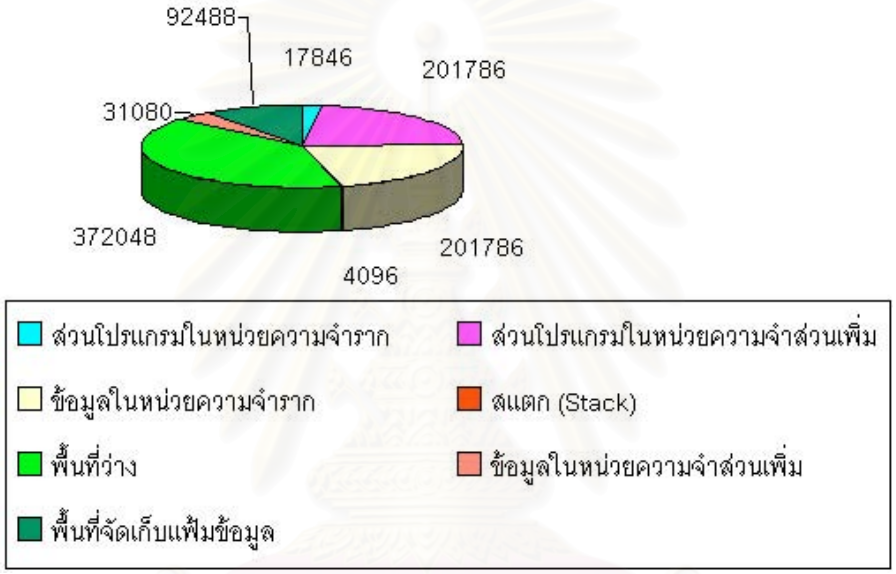


รูปที่ 5.13 เวกเตอร์ตั้งต้นในแต่ละชุดข้อมูลเป็นข้อมูลสุ่ม

โปรแกรมที่ได้พัฒนาสามารถแบ่งออกเป็นสัดส่วนตามหน้าที่การทำงานได้ดังรูปที่ 5.14 โดยประกอบไปด้วยเครื่องบริการเว็บซึ่งได้ดัดแปลงแก้ไขจากไลบรารีของไดนามิกซีซอฟต์แวร์ส่วนหลักซึ่งรวมการจัดการเอสเอ็นเอ็มพีผ่านเว็บทั้งหมด และระบบรักษาความปลอดภัย ส่วนโปรแกรมทั้งหมดถูกเก็บอยู่ในทรัพยากรของระบบฝังตัว แบ่งตามชนิดของทรัพยากรได้ดังรูปที่ 5.15 ซึ่งส่วนที่สำคัญมากที่สุดอยู่ที่สแตก (Stack) เพราะหากเก็บข้อมูลเกินขอบเขตจะทำให้โปรแกรมทำงานผิดพลาดได้ ต้องจัดการให้ข้อมูลขนาดใหญ่เก็บในพื้นที่ว่างแทนการเก็บลงสแตก



รูปที่ 5.14 แสดงอัตราส่วนของซอฟต์แวร์ของระบบ



รูปที่ 5.15 แสดงขนาดของโปรแกรมแบ่งตามชนิดทรัพยากรที่จัดเก็บ

ในการทดสอบการทำงานผ่านระบบรักษาความปลอดภัยนั้นได้ทดสอบเปรียบเทียบผลการทำงานระหว่างการทำงานที่อาศัยระบบรักษาความปลอดภัย (Encryption: AES_CBC_128) และการทำงานที่ไม่ใช้ระบบรักษาความปลอดภัย (Encryption: Plain) โดยแบ่งเป็น 5 แบบการทดสอบ

- รูปแบบที่ 1 ทดสอบด้วยชุดคำสั่งที่มีความซับซ้อนน้อยสำหรับโปรแกรมขนาดเล็ก โดยพัฒนาโปรแกรมขนาดเล็กที่สามารถรับคำสั่งซีจีไอได้ และตอบกลับในทันทีเพื่อตรวจสอบว่าระบบฝังตัวยังทำงานอยู่

- รูปแบบที่ 2 ทดสอบด้วยชุดคำสั่งสร้างแฟ้มข้อมูล (GET /filemanager?action=add) สำหรับโปรแกรมขนาดเล็ก

- รูปแบบที่ 3 ทดสอบด้วยชุดคำสั่งพื้นฐานของระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัว ประกอบไปด้วยคำสั่งการร้องขอและการกำหนดค่าเอสเอ็นเอ็มพี (GET /snmp ซึ่งมีค่าตัวแปรเสริม “action” เป็น “get” “getnext” “set” หรือ “ping”

- รูปแบบที่ 4 ทดสอบด้วยชุดคำสั่งที่ต้องการการประมวลผลที่ซับซ้อน หรือมีการร้องขอ และกำหนดค่าหลายครั้ง โดยใช้กับระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัว ประกอบด้วยการร้องขอข้อมูลเอสเอ็นเอ็มพีเป็นกลุ่ม (GET /snmp?action=getgroup) และการร้องขอข้อมูลเอสเอ็นเอ็มพีโดยระบุโอไอดีเริ่มต้น (GET /snmp?action=walk)

- รูปแบบที่ 5 ทดสอบด้วยชุดคำสั่งสร้างแฟ้มข้อมูลเช่นเดียวกับรูปแบบที่ 2 แต่ใช้กับระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัว

ตารางที่ 5.3 ผลการทดสอบเปรียบเทียบระหว่างการใช้ระบบรักษาความปลอดภัย กับระบบที่ไม่ใช้ระบบรักษาความปลอดภัย

รูปแบบที่	จำนวนข้อมูลในส่วนข้อมูลเอชทีทีพี (ไบต์)	ระยะเวลา (วินาที)	
		ไม่ใช้ระบบรักษาความปลอดภัย	ใช้ระบบรักษาความปลอดภัย
1	-	0.095	0.158
2	0	0.123	0.189
	512	0.237	0.536
	51200	9.977	22.916
3	-	0.206	1.203
4	-	1.096	4.340
5	0	0.347	0.959
	512	0.465	8.310
	51200	10.262	489.974

ผลการทดสอบแสดงได้ดังตารางที่ 5.3 ซึ่งการทดสอบด้วยรูปแบบที่ 2 และรูปแบบที่ 5 ซึ่งเป็นการสร้างแฟ้มข้อมูลโดยนำข้อมูลจากฝั่งเว็บมาเป็นข้อมูลภายในแฟ้มข้อมูล

สามารถทดสอบการทำงานได้เป็นอย่างดีเนื่องจากมีการส่งข้อมูลจำนวนมากภายในระยะเวลาติดต่อกันโดยการทดสอบการสร้างแฟ้มข้อมูลตามขนาดข้อมูลภายในแฟ้ม การทดสอบแต่ละรูปแบบวัดระยะเวลานับตั้งแต่เริ่มส่งข้อมูลร้องขอเอชทีทีพี จนกระทั่งมีการตอบรับข้อมูลจนครบทั้งหมด ซึ่งผลการทดสอบเป็นค่าเฉลี่ยการทดสอบ 3 ครั้ง

จากผลการทดสอบดังกล่าวแสดงให้เห็นว่าระบบฝังตัวที่ใช้ระบบรักษาความปลอดภัยนั้น ใช้เวลาในการทำงานมากกว่าระบบฝังตัวที่ไม่ใช้ระบบรักษาความปลอดภัยอย่างเห็นได้ชัด ซึ่งเมื่อพิจารณาเรื่องความปลอดภัยแล้วคุ้มค่าสำหรับการสื่อสารข้อมูลที่ปลอดภัย แต่หากผู้ดูแลระบบใช้งานอยู่ภายในองค์กรเอง และต้องการความรวดเร็วในการทำงาน ผู้ดูแลระบบสามารถปิดการทำงานของระบบรักษาความปลอดภัยได้เพื่อความรวดเร็วในการติดต่อกับระบบฝังตัว สำหรับโปรแกรมขนาดเล็ก (รูปแบบที่ 1 และ รูปแบบที่ 2) การใช้ระบบรักษาความปลอดภัยนั้นใช้เวลาในการทำงานมากกว่า ระบบที่ไม่ใช้ความปลอดภัยอยู่ประมาณ 1 เท่าซึ่งผลที่ได้ถือว่าอยู่ในระดับที่ยอมรับได้ แต่สำหรับชุดคำสั่งสร้างแฟ้มข้อมูลบนระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัว (รูปแบบที่ 5) ใช้เวลาสำหรับการทำงานมากซึ่งเกิดขึ้นเนื่องจากโปรแกรมขนาดเล็กนั้นใช้ทรัพยากรของระบบไปเพียงน้อยนิดทำให้เหลือทรัพยากรเพียงพอให้ระบบรักษาความปลอดภัยสามารถทำงานอยู่ภายในหน่วยความจำรากได้ ในขณะเดียวกันนั้นระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัวใช้ทรัพยากรไปมากเพราะมีส่วนของโปรแกรมที่ซับซ้อนกว่ามาก ทำให้หน่วยความจำรากมีเนื้อที่เหลือไม่เพียงพอในการเก็บส่วนโปรแกรมสำหรับการรักษาความปลอดภัย จึงต้องนำส่วนโปรแกรมสำหรับการรักษาความปลอดภัยนำไปเก็บในหน่วยความจำส่วนเพิ่มและนำบางส่วนออกมาเก็บในหน่วยความจำส่วนรากเพื่อประมวลผลระหว่างการทำงาน ซึ่งเป็นเหตุผลที่การทดสอบด้วยระบบรักษาความปลอดภัยจึงใช้เวลาค่อนข้างมาก

ผลการทดสอบทั้งหมดแสดงให้เห็นว่า การติดต่อสื่อสารกับระบบฝังตัวนั้นจำเป็นต้องใช้ระบบรักษาความปลอดภัยเพื่อปกป้องข้อมูลที่สำคัญของเอสเอ็นเอ็มพี ถึงแม้ว่าในกรณีของระบบจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บสำหรับอุปกรณ์ฝังตัวจะใช้เวลาในการทำงานที่มาก แต่โดยส่วนใหญ่แล้วการติดต่อกับระบบฝังตัวเพื่อจัดการเอสเอ็นเอ็มพีเอเจนต์นั้นมักเป็นคำสั่งที่มีขนาดไม่ยาวมากนัก และคำสั่งการสร้างแฟ้มข้อมูลมิได้เกิดขึ้นบ่อยๆ เกิดในกรณีที่ผู้ดูแลระบบต้องการแก้ไข หรือปรับปรุงแอปเพล็ตให้ตรงกับการใช้งานมากขึ้นเท่านั้น

บทที่ 6
สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้มุ่งเน้นไปที่การออกแบบระบบฝังตัวให้มีความสามารถในการจัดการเครือข่ายคอมพิวเตอร์ผ่านเว็บได้เพื่อให้ผู้ดูแลระบบเครือข่ายสามารถควบคุมดูแล และจัดการเครือข่ายได้จากทุกที่โดยผ่านระบบเครือข่ายอินเทอร์เน็ต ระบบฝังตัวสามารถนำมาติดตั้งภายในระบบเครือข่ายได้อย่างรวดเร็วและสะดวก อีกทั้งยังใช้พื้นที่น้อย และประหยัดพลังงานมากกว่าเครื่องคอมพิวเตอร์ ซึ่งงานวิจัยนี้ประกอบไปด้วยส่วนของฮาร์ดแวร์ และซอฟต์แวร์โดยทำงานร่วมกัน ส่วนของซอฟต์แวร์ถือเป็นหัวใจหลักของระบบเพื่อทำให้เกิดระบบการจัดการเครือข่าย โดยการออกแบบซอฟต์แวร์ให้ทำงานผ่านเว็บได้นั้น ทำให้การจัดการเป็นไปได้สะดวกยิ่งขึ้น เนื่องจากผู้ดูแลระบบไม่จำเป็นต้องนำโปรแกรมติดตัวไปด้วยสำหรับการจัดการเครือข่าย ผู้ดูแลระบบเพียงติดต่อกับระบบเครือข่ายอินเทอร์เน็ตเพื่อเรียกส่วนของซอฟต์แวร์ภายในระบบฝังตัวออกมาทำงาน หากผู้ดูแลระบบพบว่าซอฟต์แวร์ภายในนั้นไม่เหมาะสมสำหรับการทำงาน ผู้ดูแลระบบยังสามารถเปลี่ยนซอฟต์แวร์ภายในให้เหมาะสมโดยไม่จำเป็นต้องอยู่ในบริเวณที่ระบบฝังตัวติดตั้งอยู่ ตราบเท่าที่การเปลี่ยนแปลงนั้นไม่กระทบกับส่วนย่อยการเชื่อมต่อเว็บ หรือส่วนย่อยการเชื่อมต่อระบบฝังตัว

ระบบรองรับการจัดการทั้งแบบการจัดการด้วยผู้ดูแลระบบเอง และการจัดการแบบอัตโนมัติ ดังนั้นการจัดการจึงเป็นไปได้อย่างทันท่วงที ลดปัญหาที่อาจเกิดขึ้นกับระบบเครือข่ายลงได้ถึงแม้ว่าการจัดการแบบอัตโนมัตินี้ยังต้องอาศัยการกำหนดเหตุการณ์เริ่มต้นในการตรวจสอบให้เหมาะสมก็ตาม แต่การจัดการด้วยวิธีดังกล่าวช่วยให้ระบบฝังตัวในปัจจุบันส่วนใหญ่ซึ่งมีทรัพยากรน้อยสามารถนำการจัดการเครือข่ายผ่านเว็บนี้ไปสร้างบนระบบฝังตัวได้โดยใช้แบบจำลองการทำงานของซอฟต์แวร์เดียวกันกับงานวิจัยนี้

ในด้านของความปลอดภัยในการใช้งานนั้น ผู้ดูแลระบบเท่านั้นที่ทราบถึงรหัสผ่านในการเข้าใช้งานดังนั้นข้อมูลต่างๆ ได้ถูกเข้ารหัสด้วยเออีเอสซึ่งมีความปลอดภัยสูงเมื่อเทียบกับวิธีการเข้ารหัสแบบกุญแจสมมาตรหลายๆ วิธี ซึ่งถึงแม้ว่าข้อมูลต่างๆ ถูกดักจับไปได้ระหว่างเส้นทางการเชื่อมต่อ แต่ผู้ดักจับไปนั้นไม่สามารถถอดรหัสได้โดยตรง ต้องใช้วิธีการทดลองสุ่มกุญแจลับ ซึ่งโอกาสที่กุญแจลับที่ถูกสุ่มขึ้นมาได้ถูกนั้นมีโอกาสเป็นไปได้ 1 จากจำนวนครั้งที่มากที่สุดถึงประมาณ 3.4×10^{38} ครั้งด้วยกัน (ขึ้นอยู่กับขนาดของกุญแจลับ โดยในงานวิจัยนี้อาศัย

กุญแจลับซึ่งมีความยาวที่ 128 บิต) แสดงให้เห็นถึงความปลอดภัยของระบบฝังตัวซึ่งอยู่ในระดับที่สูงพอสมควร นอกจากนี้แล้วเนื่องจากระบบฝังตัวที่ได้ออกแบบมามิได้ทำงานขึ้นอยู่กับระบบปฏิบัติการใดๆ ทั้งสิ้น ดังนั้นช่องโหว่ที่ผู้บุกรุกมักนำมาโจมตีกับระบบปฏิบัติการที่เป็นที่นิยมจึงไม่สามารถนำมาใช้กับระบบฝังตัวที่ได้ออกแบบ

สำหรับผู้พัฒนานั้นควรระวังในเรื่องของการเขียนโปรแกรม และการเก็บข้อมูลลงบนอุปกรณ์ฝังตัวเนื่องจากทรัพยากรอันจำกัดของระบบฝังตัว ในงานวิจัยนี้ประสบกับปัญหาหลักของการพัฒนาโปรแกรมลงบนระบบฝังตัวเรื่องขนาดของหน่วยความจำไม่เพียงพอสำหรับการทำงานในบางขั้นตอน ซึ่งปัญหานี้แก้ได้ด้วยการออกแบบระบบให้มีความซับซ้อนให้น้อยลงเพื่อหลีกเลี่ยงการเรียกใช้ฟังก์ชันซ้อนกันเป็นจำนวนมาก พร้อมกับการนำข้อมูลที่มีขนาดใหญ่เก็บลงในหน่วยความจำส่วนเพิ่มดังได้กล่าวถึงในบทที่ 4

งานวิจัยนี้แสดงให้เห็นว่าระบบฝังตัวนั้นสามารถนำมาสร้างระบบจัดการเครือข่ายคอมพิวเตอร์ได้ และใช้แทนเครื่องคอมพิวเตอร์ที่ลงโปรแกรมการจัดการเครือข่ายได้ในระดับหนึ่งเท่านั้น เนื่องจากโปรแกรมคอมพิวเตอร์สำหรับการจัดการเครือข่ายนั้นผู้พัฒนาสามารถพัฒนาให้สลับซับซ้อนได้มากกว่าอุปกรณ์ฝังตัว อีกทั้งยังมีหน่วยความจำที่เพียงพอสำหรับการพัฒนาโปรแกรมประยุกต์ที่มีการเก็บข้อมูลขนาดใหญ่ แต่อย่างไรก็ตามระบบฝังตัวที่ได้พัฒนามีข้อดีในหลายประการเช่นกัน จึงเหมาะสำหรับหน่วยงานหรือองค์กรที่ไม่ใหญ่มากแต่ต้องการเครื่องมือสำหรับการจัดการเครือข่ายคอมพิวเตอร์ ผู้ดูแลระบบยังได้รับความสะดวกในการดูแลจัดการเครือข่ายคอมพิวเตอร์ที่ตนเองรับผิดชอบได้ตลอดเวลาที่ต้องการ นอกจากนี้แล้วหากผู้พัฒนาต้องการให้ระบบมีความสามารถในการวิเคราะห์ข้อมูลที่สูงขึ้น ผู้พัฒนาสามารถมองระบบฝังตัวในงานวิจัยนี้เป็นเสมือนอุปกรณ์พื้นฐานสำหรับติดต่อกับเอสเอ็นเอ็มพีเอเจนต์เพื่อเก็บข้อมูลส่งต่อไปยังโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลขั้นสูงขึ้น หากผู้พัฒนาตระหนักถึงการทำงานในลักษณะนี้แล้วสามารถสร้างโปรแกรมประยุกต์ผ่านเว็บ แอปเพล็ต หรือโปรแกรมประยุกต์ที่ทำงานบนเครื่องคอมพิวเตอร์ให้มีความสามารถที่หลากหลายโดยไม่ขึ้นความสามารถอันจำกัดของอุปกรณ์ฝังตัวอีกต่อไป เนื่องจากระบบฝังตัวจะรับผิดชอบเฉพาะคำสั่งพื้นฐานเอสเอ็นเอ็มพี และโปรแกรมที่พัฒนาขึ้นมาใหม่จะนำข้อมูลไปวิเคราะห์ด้วยวิธีที่ซับซ้อนขึ้นเพื่อให้ได้ข้อมูลที่แม่นยำถูกต้องตรงกับความต้องการมากที่สุด

6.2 ข้อเสนอแนะ

เนื่องจากการทำงานของระบบรองรับการทำงานร่วมกับข้อตกลงเอสเอ็นเอ็มพีเพียงรุ่นเดียวเท่านั้น แต่แนวโน้มของการพัฒนาอุปกรณ์เอสเอ็นเอ็มพีให้ทำงานในรุ่นที่สูงขึ้นไปกำลังสูงขึ้น ดังนั้นระบบควรรองรับการทำงานกับเอสเอ็นเอ็มพีรุ่นที่ 2 และ 3 แต่ผู้พัฒนาควรคำนึงถึงทรัพยากรอันจำกัดของอุปกรณ์ฝังตัวด้วย เพราะการทำงานกับรุ่นที่เพิ่มขึ้นมานี้ ย่อมต้องใช้ทรัพยากรของระบบที่ต้องสูงขึ้นไปด้วย นอกจากนี้แล้วการทำงานของระบบฝังตัวอาจต้องการประสิทธิภาพที่สูงขึ้น ซึ่งเป็นผลให้ผู้พัฒนาอาจต้องเปลี่ยนอุปกรณ์ฝังตัวเป็นรุ่นที่สูงขึ้นเพื่อให้ตรงกับความต้องการ

รายการอ้างอิง

[1] H.-G. Hegering, S. Abeck, and B. Neumair. Integrated Management of Networked Systems: Concepts, Architectures, and Their Operational Application. Morgan Kaufmann, September, 1999.

[2] Future Software Limited, Indea. FCAPS [Online], October 2003, Available from: http://www.futsoft.com/pdf/fcapswp.pdf

[3] J. Case, J. A simple network management protocol (SNMP). RFC 1157 May, 1990.

[4] P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 June, 1999.

[5] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321 April, 1992.

[6] A. Kahate. Cryptography and Network Security, Advance Encryption Standard (AES), 107-109. New Delhi 110 020 : Tata McGraw-Hill, 2003.

[7] C. Kaufman, R. Perlman, and M. Speciner. Network Security PRIVATE Communication in a PUBLIC World, Advance Encryption Standard (AES), 81-91. United States of America : Prentice Hall PTR, 2002.

[8] V. Frederik. Symmetric Key Cryptography [Online], March 3, 2002, Available from: http://www.madchat.org/crypto/Part_III.pdf

[9] B. T. Ching-Wun, and B. C. Ruay-Shiung. SNMP through WWW. International Journal of Network Management, Volume 8, Issue 2, March 1998: 104-119.

[10] J. B. Siu, and Z. S. Guo. Web-based Network Configuration Management System, Communication Technology Proceeding 2000, Volume 1: 487-491.

[11] H. Liu, X. Shao, L. Kong, and W. Ding. Integeration of Web into an Embedded SNMP-based Management Environment, High Speed Networks and Multimedia Communications 5th IEEE International Conference, July 2002: 320-324.

[12] H.-T. Ju, M.-J. Cho, and J. W. Hong. An efficient and lightweight embedded Web server for Web-based network element management, International Journal of Network Management, Volume 10, Issue 5, September-October 2000: 261-275.

[13] H. Liu, D. Bai, and W. Ding. The Integration of SNMP and Web in Embedded Devices, Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on , Volume: 2 , 29 Oct.-1 Nov. 2001: 83 – 87.

[14] C. Sanchez-Avila, and R. Sanchez-Reillol. The Rijndael block cipher (AES proposal) : a comparison with DES, Security Technology, 2001 IEEE 35th International Carnahan Conference on , 16-19 Oct. 2001: 229 - 234

[15] S. Sen, S.I. Hossain, K. Islam, D.R. Chowdhuri, and P.P. Chaudhuri, Cryptosystem designed for embedded system security, VLSI Design, 2003. Proceedings. 16th International Conference on , 4-8 Jan. 2003: 271 – 276

[16] M. Chanin, and S. Boonchai. Web-based Network Management using Embedded System, The 4th Information and Computer Engineering Postgraduate Workshop 2004, Jan. 2004: 63 - 68

[17] Rabbit Semiconductor. Rabbit Memory Management In a Nutshell [Online], Rabbit Semiconductor, Available from: http://www.rabbitsemiconductor.com/ documentation/docs/refs/TN202/ TN202.htm

[18] H. Sean. Total SNMP: Exploring the Simple Network Management Protocol, Second Edition, Prentice Hall, 1998

[19] A. Shyam. Arbitrary Length Very Large Integer / Big Number [Online], New Jersy Institute of Technology, 2003, Available from: http://web.njit.edu/~sa39/Articles/ BigInt/index.php

[20] S. Rags. Using AES with Java Technology [Online], Sun Microsystems, 2003, Available from: http://java.sun.com/developer/technicalArticles/Security/ AES/AES_v1.html

[21] M. Chanin, and S. Boonchai. Security Methods for Web-based Application on Embedded System, IEEE TENCON 2004, November 2004

[22] P. Colin. AES Encryption [Online], the University of Oxford, 2002, Available from: http://web.comlab.ox.ac.uk/oucl/work/colin.percival/source/lib/

[23] E. J. Philip. RSA Data Security, Inc. MD5 Message Digest Algorithm [Online], efgh.com, 2002, Available from: http://efgh.com/software/md5.htm

ภาคผนวก

ภาคผนวก ก
ไลบรารีทั้งหมดภายในระบบฝังตัว

การทำงานของระบบฝังตัวซึ่งเป็นทั้งส่วนติดต่อกับเอสเอ็นเอ็มพีเอเจนต์ และเครื่องบริการเว็บเพื่อรองรับการติดต่อจากแอปเพล็ตประกอบไปด้วยไลบรารีต่างๆ ดังนี้

1. WebSNMP.lib เป็นไลบรารีเพื่อเก็บฟังก์ชันหลักในการเรียกใช้ฟังก์ชันย่อยจากไลบรารีอื่น ซึ่งฟังก์ชันที่สามารถเรียกได้จากภายนอกไลบรารี รวมถึงฟังก์ชันสำคัญภายในไลบรารีมีดังนี้
    - 1.1 websnmp_init สำหรับการกำหนดค่าเริ่มต้นของระบบฝังตัว
    - 1.2 websnmp_handler เป็นฟังก์ชันซึ่งผู้พัฒนาต้องเรียกใช้เพื่อให้ระบบฝังตัวจัดการกับข้อมูลรับเข้าและส่งออก ทั้งยังจัดการเกี่ยวกับเรื่องการแสดงผลการทำงาน และการจัดการกับเหตุการณ์ต่างๆ ด้วย
    - 1.3 websnmp_ping ตรวจสอบสถานะของกลุ่มเอสเอ็นเอ็มพีเอเจนต์
    - 1.4 ping_init สำหรับการกำหนดค่าเริ่มต้นทุกครั้งที่ระบบฝังตัวเริ่มการตรวจสอบสถานะของเอสเอ็นเอ็มพีเอเจนต์ชุดใหม่
    - 1.5 ping_reset เปลี่ยนแปลงช่วงของเลขที่อยู่ไอพีสำหรับการตรวจสอบสถานะของเอสเอ็นเอ็มพีเอเจนต์
    - 1.6 PingTable_init กำหนดค่าเริ่มต้นสำหรับตารางสถานะเอสเอ็นเอ็มพีเอเจนต์
    - 1.7 PingTable_setstatus กำหนดสถานะของเอสเอ็นเอ็มพีเอเจนต์
    - 1.8 PingTable_getstatus อ่านค่าสถานะของเอสเอ็นเอ็มพีเอเจนต์
    - 1.9 websnmp_distributeMail กระจายอีเมลให้กับรายชื่อที่อยู่อีเมล
    - 1.10 email_load อ่านรายชื่อที่อยู่อีเมลจากหน่วยความจำส่วนเพิ่มเพื่อนำมาเก็บในหน่วยความจำราก
    - 1.11 email_store นำรายชื่อที่อยู่อีเมลจากหน่วยความจำรากไปเก็บในหน่วยความจำส่วนเพิ่ม
    - 1.12 email_find สำหรับหาตำแหน่งรายชื่อที่อยู่อีเมล
    - 1.13 email_empty หาตำแหน่งว่างสำหรับเก็บที่อยู่อีเมล
    - 1.14 email_clearall ลบรายชื่อที่อยู่อีเมลทั้งหมด
    - 1.15 email_add เพิ่มรายชื่อที่อยู่อีเมลใหม่

1.16 emai_remove ลบรายชื่อที่อยู่อีเมลตามที่ระบุ

1.17 recv_data รอรับข้อมูลตอบกลับจากเอสเอ็นเอ็มพีเอเจนต์โดยไม่หยุดรอการทำงาน

1.18 loadfromflash อ่านข้อมูลสำคัญจากหน่วยความจำแฟลชเข้าสู่หน่วยความจำส่วนเพิ่ม และหน่วยความจำราก

1.19 events_2_flashfile เก็บข้อมูลการตรวจสอบเหตุการณ์ลงสู่แฟ้มข้อมูลซึ่งเก็บอยู่ภายในหน่วยความจำแฟลช

1.20 flashfile_2_events นำข้อมูลจากแฟ้มข้อมูลภายในหน่วยความจำแฟลชมากำหนดให้กับข้อมูลการตรวจสอบเหตุการณ์

1.21 cgi_register ลงทะเบียนการใช้งานของผู้ดูแลระบบซึ่งติดต่อมา

1.22 cgi_parse_param อ่านค่าตัวแปรเสริมทั้งหมดที่แอปเพล็ตส่งมาด้วย และกำหนดค่าให้กับตัวแปรต่างๆ

1.23 cgi_param_val อ่านค่าตัวแปรเสริมตัวใดๆ

1.24 cgi_snmp ฟังก์ชันรองรับคำสั่งเอสเอ็นเอ็มพีจากเอชทีทีพี

1.25 cgi_save2flash เก็บข้อมูลสำคัญจากหน่วยความจำส่วนเพิ่ม และหน่วยความจำรากลงสู่หน่วยความจำแฟลช

1.26 cgi_filemanager สำหรับจัดการแฟ้มข้อมูลภายในระบบฝังตัว

1.27 cgi_RTC ฟังก์ชันจัดการเวลาที่ใช้ภายในระบบฝังตัว

1.28 cgi_event รองรับการจัดการตรวจสอบเหตุการณ์

1.29 cgi_trap รองรับการจัดการข้อมูลแทรพ

1.30 cgi_protected รองรับการติดต่อด้วยระบบรักษาความปลอดภัย

2. MyHTTP.lib เป็นไลบรารีเครื่องบริการเว็บซึ่งถูกดัดแปลงมากจาก ไลบรารีเครื่องบริการเว็บของบริษัทแรบบิทเซมิคอนดักเตอร์ โดยเพิ่มความสามารถให้รองรับเอชทีทีพีรุ่น 1.1 ซึ่งใช้ประโยชน์ในเรื่องของแคชทำให้เครื่องลูกข่ายเรียกใช้บริการได้รวดเร็วขึ้น นอกจากนี้แล้วยังเพิ่มการพิสูจน์ตนด้วยเอชทีทีพี โดยมีฟังก์ชันต่างๆ ที่เพิ่มขึ้นมาดังนี้

2.1 Bin2Hex สำหรับการแปลงข้อมูลให้อยู่ในรูปของสายอักขระซึ่งเป็นเลขฐาน 16 โดยใช้ในการสร้างสายอักขระส่งให้กับเครื่องลูกข่ายในขั้นตอนการพิสูจน์ตนด้วยเอชทีทีพีไดเจส

2.2 DigestCalcResponse สำหรับสร้างตัวแทนข้อมูลเพื่อใช้เปรียบเทียบกับตัวแทนข้อมูลที่ได้รับมาจากแอปเพล็ตในการพิสูจน์ตนด้วยเอชทีทีพีไดเจส

3. MyZServer.lib เป็นไลบรารีพื้นฐานสำหรับเครื่องบริการ โดยดัดแปลงมาจากไลบรารีของแรบบิทเซมิคอนดักเตอร์เพื่อให้สามารถรองรับกับไลบรารีเครื่องบริการเว็บที่ถูกดัดแปลง ซึ่งดัดแปลงเพียงเพิ่มการบันทึกเวลาในการแก้ไขแฟ้มข้อมูลครั้งล่าสุด

4. MyWeb.lib รวบรวมฟังก์ชันสำหรับการสร้างการตอบรับเอชทีทีพี โดยรองรับการเข้ารหัสข้อมูลขณะส่งข้อมูลกลับด้วยซึ่งมีฟังก์ชันเรียกใช้ได้ดังนี้

4.1 web_init สำหรับการกำหนดค่าเริ่มต้นการใช้ไลบรารี

4.2 web_addbuff สำหรับสร้างกลุ่มข้อมูลตอบรับกลับไปยังแอปเพล็ตโดยสร้างข้อมูลเฉพาะในส่วนเนื้อหาของเอชทีทีพี

4.3 web_addparam สร้างเขตข้อมูลใหม่ภายในส่วนหัวของเอชทีทีพี

4.4 web_respond_event สร้างชุดข้อมูลตอบกลับคำสั่งร้องขอเหตุการณ์ที่ระบบฝังตัวตรวจสอบอยู่ โดยรับผิดชอบเฉพาะส่วนเนื้อหาของเอชทีทีพี

4.5 web_write_header สร้างส่วนหัวของเอชทีทีพีตามรหัสการตอบรับของเอชทีทีพี

4.6 web_write_content สร้างส่วนข้อมูลเอชทีทีพีสำหรับตอบรับกับคำสั่งทั้งหมดยกเว้น การตอบรับคำสั่งร้องขอเหตุการณ์

4.7 web_respond สำหรับส่งข้อมูลเอชทีทีพีทั้งส่วนหัว และส่วนข้อมูลโดยข้อมูลมีขนาดความยาวที่ระบุได้

4.8 web_respond_chunk ทำงานเหมือนกับฟังก์ชัน web_respond แต่สามารถสร้างข้อมูลที่มีความยาวข้อมูลไม่แน่นอน (chunk) ตามรูปแบบของเอชทีทีพีได้ด้วย

5. MyLCD.lib เป็นไลบรารีสำหรับติดต่อกับส่วนแสดงผล โดยมีฟังก์ชันที่สามารถเรียกใช้จากภายนอกไลบรารีดังนี้

5.1 lcd_init สำหรับการกำหนดค่าเริ่มต้นให้กับส่วนแสดงผล

5.2 lcd_on สำหรับการกำหนดวิธีการแสดงผล

5.3 lcd_clear ลบข้อมูลทั้งหมดบนส่วนแสดงผล

5.4 lcd_home เลื่อนตำแหน่งการแสดงผลไปยังตำแหน่งเริ่มต้น

5.5 lcd_move เลื่อนตำแหน่งการแสดงผลไปยังตำแหน่งใดๆ

5.6 lcd_write แสดงข้อมูลตัวอักษรที่ตำแหน่งการแสดงผล

5.7 lcd_print แสดงข้อมูลสายอักขระที่ตำแหน่งที่กำหนด

6. MyBox.lib ไลบรารีสำหรับเก็บฟังก์ชัน และโครงสร้างข้อมูลพื้นฐานของข้อมูลชนิด Box โดยมีฟังก์ชันเตรียมให้เรียกใช้จากภายนอกดังนี้

6.1 box_load นำข้อมูลชนิด Box จากหน่วยความจำส่วนเพิ่มมาเก็บไว้ในหน่วยความจำส่วนรากเพื่อประมวลผล

6.2 box_store นำข้อมูลชนิด Box จากหน่วยความจำส่วนรากเก็บลงในหน่วยความจำส่วนเพิ่มเพื่อคืนเนื้อที่หน่วยความจำส่วนรากให้กับการประมวลผลข้อมูลอื่น

6.3 box_clear ลบล้างข้อมูลทั้งหมดภายในข้อมูลชนิด Box

6.4 box_find ค้นหาข้อมูลชนิด Box ที่ตรงกับที่ระบุไว้จากข้อมูลทั้งหมดภายในหน่วยความจำ เพื่อนำไปใช้กับการติดต่อในครั้งนั้น

6.5 box_verify ตรวจสอบการติดต่อเอชทีทีพีของข้อมูลชนิด Box นั้นว่ามีการตัดการเชื่อมต่อหรือไม่

6.6 Clear_Message ลบข้อมูลที่เก็บอยู่ในหน่วยความจำชนิด my_buff ภายในข้อมูลชนิด Box

6.7 box_setagent กำหนดเอสเอ็นเอ็มพีเอเจนต์ให้กับข้อมูลชนิด Box เพื่อการติดต่อผ่านเอสเอ็นเอ็มพี

6.8 box_parse_param กำหนดค่าต่างๆ ให้กับข้อมูลชนิด Box ตามmuj ตัวแปรเสริมที่ระบุมา

6.9 box_setstate ใช้ในกรณีที่ข้อมูลชนิด Box ต้องการสร้างกุญแจลับชุดใหม่ เนื่องมาจากเครื่องบริการเว็บมีการร้องขอการเปลี่ยนค่าน็อนซ์

6.10 box_seats จำนวนของข้อมูลชนิด Box ที่ถูกใช้ในเวลาปัจจุบัน

6.11 box_respfind ค้นหาตำแหน่งเริ่มต้นของข้อมูลตอบรับกลับจากเอสเอ็นเอ็มพีเอเจนต์ภายในข้อมูลชนิด Box

7. MySNMP.lib เป็นไลบรารีซึ่งจัดรวบรวมฟังก์ชันที่ใช้ติดต่อกับเอสเอ็นเอ็มพีเอเจนต์โดยตรง รวมไปถึงฟังก์ชันสำหรับสร้างชุดข้อมูลตามรูปแบบข้อตกลงเอสเอ็นเอ็มพี โดยมีฟังก์ชันให้เรียกใช้ได้ดังนี้

7.1 mysnmp_init กำหนดค่าเริ่มต้นของไลบรารี

7.2 mysnmp_chktrap ตรวจสอบว่ามีข้อมูลแทรพหรือไม่ หากมีข้อมูลแทรพจะแปลความหมายข้อมูล และเก็บข้อมูล

7.3 snmp_send ส่งข้อมูลเอสเอ็นเอ็มพีไปยังเอเจนต์ โดยพยายามส่ง 3 รอบ หากยังไม่สามารถส่งได้จะถือว่าการส่งครั้งนั้นผิดพลาด ฟังก์ชันนี้เป็นฟังก์ชันซึ่งไม่รอการตอบรับหากส่งไปแล้วสามารถเรียกใช้ฟังก์ชันอื่นต่อได้ทันที

7.4 snmp_waitdata ตรวจสอบการตอบรับข้อมูลเอสเอ็นเอ็มพีโดยไม่หยุดรอในกรณีที่ยังไม่มีข้อมูลมาถึง

7.5 snmp_chkmessage ตรวจสอบความถูกต้องของข้อมูลส่วนหัวเอสเอ็นเอ็มพี

7.6 snmp_readgetresp แปลความหมายข้อมูลภายในส่วนของข้อมูลเอสเอ็นเอ็มพี และเก็บข้อมูลดังกล่าวลงในส่วนเก็บข้อมูลของตัวแปรประเภท Box

7.7 oid2ber แปลงข้อมูลในรูปแบบสายอักขระของหมายเลขโอไอดีให้อยู่ในรูปแบบชุดข้อมูลบีอีอาร์

7.8 ber2oid เปลี่ยนชุดข้อมูลบีอีอาร์เป็นสายอักขระหมายเลขโอไอดี

7.9 Create_Data สร้างชุดข้อมูลบีอีอาร์ 1 ชุดโดยสามารถกำหนดวิธีการเก็บข้อมูลทั้งแบบต่อท้ายข้อมูลเดิม และแทรกข้อมูลก่อนหน้า

7.10 Create_Message สร้างส่วนหัวของข้อมูลเอสเอ็นเอ็มพี

7.11 Add_Varbind เพิ่มค่าอ็อบเจกต์สำหรับการร้องขอข้อมูลเอสเอ็นเอ็มพี และกำหนดค่าข้อมูลเอสเอ็นเอ็มพี

7.12 easy_readint แปลความหมายข้อมูลเอสเอ็นเอ็มพีซึ่งเป็นค่าตัวเลขให้เป็นข้อมูลสายอักขระของตัวเลขนั้น โดยรองรับตัวเลขที่มีขนาดยาวเกิน 4 ไบต์ ซึ่งผู้พัฒนาสามารถระบุความยาวสูงสุดของเนื้อที่เก็บสายอักขระได้

7.13 chk_or_read ฟังก์ชันสำหรับการตรวจสอบความถูกต้องของชุดข้อมูลบีอีอาร์ 1 ชุด หรือแปลความหมายชุดข้อมูลบีอีอาร์และเก็บลงหน่วยความจำ

7.14 isINT ตรวจสอบชนิดข้อมูลเอสเอ็นเอ็มพีว่าเป็นชนิดตัวเลขหรือไม่

8. MySecurity.lib รวบรวมฟังก์ชัน และโครงสร้างข้อมูลพื้นฐานสำหรับการเข้ารหัส ถอดรหัสเออีเอสโดยได้ดัดแปลงจากไลบรารีเออีเอสที่ได้รับการพัฒนามาแล้ว [22] รวมไปถึงรวบรวมฟังก์ชันสำหรับสร้างตัวแทนข้อมูลชนิดเอ็มดี 5 [23] และปรับการทำงานให้เหมาะกับระบบฝังตัว

9. MyEvent.lib รวบรวมฟังก์ชันสำหรับการตรวจสอบการเกิดเหตุการณ์ และการจัดการกับเหตุการณ์ต่างๆ ที่เกิดขึ้นโดยมีฟังก์ชันสำหรับเรียกใช้จากภายนอกดังนี้

9.1 EventRecord_init สำหรับกำหนดค่าเริ่มต้นของไลบรารี

9.2 EventRecord_load นำข้อมูลเหตุการณ์ที่ต้องตรวจสอบจากหน่วยความจำส่วนเพิ่มเก็บลงในหน่วยความจำราก

9.3 EventRecord_store เก็บเหตุการณ์ซึ่งอยู่ภายในหน่วยความจำรากไว้ในหน่วยความจำส่วนเพิ่ม

9.4 EventRecord_clear ลบเหตุการณ์ที่ต้องตรวจสอบตามลำดับเหตุการณ์ที่ได้ระบุมา

9.5 EventRecord_checkall ตรวจสอบเหตุการณ์ทั้งหมดหากเหตุการณ์ใดตรวจพบจะเก็บข้อมูลการพบเหตุการณ์ และจัดการตามที่ผู้ดูแลระบบได้ระบุไว้

10. HistoryData.lib เป็นไลบรารีเพื่อจัดการกับข้อมูลจำพวกที่ต้องเก็บเป็นชุด เช่น ข้อมูลการเกิดเหตุการณ์ หรือข้อมูลแทรพที่พบ โดยมีฟังก์ชันเรียกใช้ได้ดังนี้

10.1 HistoryData_init กำหนดค่าเริ่มต้นสำหรับไลบรารี

10.2 HistoryData_load นำชุดข้อมูลจากหน่วยความจำส่วนเพิ่มเก็บลงในหน่วยความจำราก

10.3 HistoryData_store นำชุดข้อมูลจากหน่วยความจำรากจัดเก็บลงในหน่วยความจำส่วนเพิ่ม

10.4 HistoryData_clear ลบชุดข้อมูลตามลำดับที่ระบุไว้

10.5 HistoryData_clearAll ลบชุดข้อมูลทั้งหมดภายในไลบรารี

10.6 HistoryData_add เพิ่มชุดข้อมูลใหม่ภายในไลบรารี

10.7 HistoryData_getnew ตรวจสอบหาชุดข้อมูลใหม่นับตั้งแต่ได้ตรวจสอบครั้งล่าสุด

11. MyFile.lib เป็นไลบรารีซึ่งเก็บรวบรวมฟังก์ชันสำหรับการจัดการแฟ้มข้อมูลภายในระบบฝังตัว โดยมีฟังก์ชันที่เรียกใช้ได้ดังนี้

11.1 myfile_firstinit สำหรับการลบแฟ้มข้อมูลทั้งหมด และจัดโครงสร้างการจัดเก็บแฟ้มข้อมูลภายในระบบฝังตัว โดยผู้พัฒนาจะเรียกใช้เพียงครั้งเดียวในกรณีที่ต้องการสร้างระบบเก็บแฟ้มข้อมูลใหม่ หรือต้องการลบแฟ้มข้อมูลทั้งหมด

11.2 myfile_init กำหนดค่าเริ่มต้นสำหรับการใช้งานแฟ้มข้อมูลต่างๆ

11.3 myfile_savemapfiles บันทึกชื่อแฟ้มข้อมูลทั้งหมด พร้อมกับหมายเลขแฟ้มข้อมูลลงในแฟ้มข้อมูลที่เก็บรายชื่อแฟ้มข้อมูล เพื่อการเรียกใช้แฟ้มข้อมูลด้วยชื่อแฟ้มข้อมูล แทนการเรียกใช้ด้วยหมายเลขแฟ้มข้อมูล

11.4 myfile_newfilenum ค้นหาหมายเลขแฟ้มข้อมูลที่ยังไม่ได้ถูกใช้

11.5 myfile_new สร้างแฟ้มข้อมูลใหม่โดยการระบุหมายเลขแฟ้มข้อมูลเพื่อใช้ในการอ้างอิงของระบบฝังตัว

11.6 myfile_newhttpfile สร้างแฟ้มข้อมูลใหม่โดยการระบุชื่อแฟ้มข้อมูลและทำการเก็บรายชื่อแฟ้มข้อมูลควบคู่กับหมายเลขแฟ้มข้อมูล

11.7 myfile_delhttpfile ลบแฟ้มข้อมูล พร้อมกับลบหมายเลขแฟ้มข้อมูลที่คู่กัน

11.8 myfile_read อ่านข้อมูลภายในแฟ้มข้อมูลเพื่อนำมาเก็บในหน่วยความจำราก

11.9 myfile_write เก็บข้อมูลลงภายในแฟ้มข้อมูล

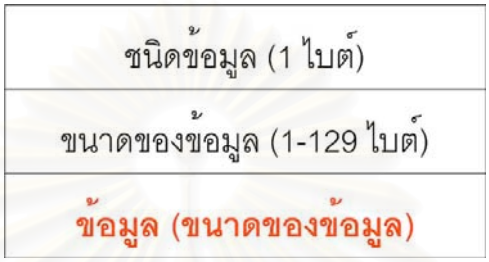
11.10 myfile_append เพิ่มข้อมูลต่อท้ายข้อมูลเดิมของแฟ้มข้อมูล

11.11 myfile_copy คัดลอกสำเนาแฟ้มข้อมูล

11.12 myfile_hasfile ตรวจสอบว่ามีแฟ้มข้อมูลที่ระบุอยู่หรือไม่

ภาคผนวก ข
โครงสร้างข้อมูลชนิดบีอีอาร์

โครงสร้างข้อมูลชนิดบีอีอาร์เป็นโครงสร้างข้อมูลหลักที่ใช้ในเอสเอ็นเอ็มพี ซึ่งข้อมูลที่ส่งตามข้อตกลงเอสเอ็นเอ็มพีนั้นจะประกอบไปด้วยข้อมูลย่อยๆ ซึ่งมีโครงสร้างบีอีอาร์เป็นหลักดังนั้นผู้พัฒนาจำเป็นต้องเข้าใจถึงโครงสร้างบีอีอาร์ โดยโครงสร้างบีอีอาร์นั้นแสดงได้ดังรูป



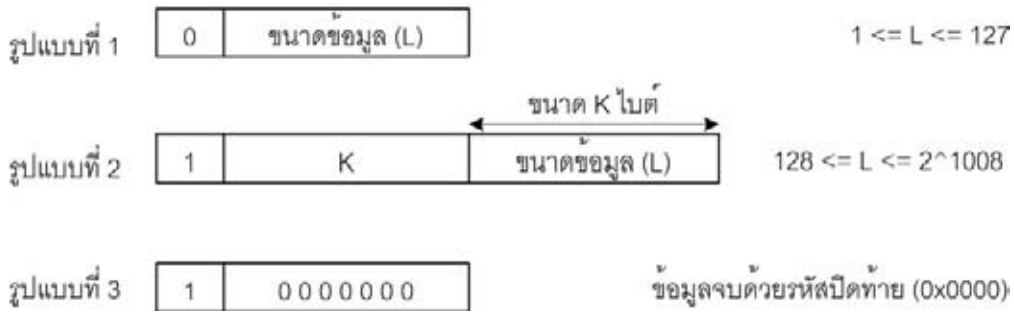
รูปแสดงโครงสร้างของข้อมูลชนิดบีอีอาร์

จากรูปชนิดของข้อมูลเป็นชนิดข้อมูลมาตรฐานซึ่งใช้ในข้อตกลงเอสเอ็นเอ็มพีโดยสามารถแสดงได้ดังตาราง

ตารางแสดงชนิดของข้อมูลที่ใช้ในงานวิจัย

ชนิดข้อมูล	รหัส
ตัวเลข (Integer)	0x02
สายอักขระ (Octet String)	0x04
ข้อมูลว่าง (NULL)	0x05
หมายเลขโอไอดี (Object ID)	0x06
กลุ่มของข้อมูล (SEQ)	0x30
เลขที่อยู่ไอพี (IP Address)	0x40
ตัวนับ (Counter)	0x41
เกจ (Gauge)	0x42
จำนวนช่วงเวลา (Timeticks)	0x43
Opaque	0x44

ขนาดของข้อมูลบ่งบอกถึงความยาวของข้อมูลที่จะตามมา โดยการเก็บขนาดของข้อมูลมีเนื้อที่ในการเก็บ 1-129 ไบต์ ซึ่งลักษณะของการเก็บขนาดข้อมูลมี 3 รูปแบบแสดงได้ดังรูป



รูปแสดงวิธีการเก็บขนาดข้อมูล

ข้อมูลซึ่งตามหลังขนาดของข้อมูลมีขนาดเท่ากับที่ระบุไว้ โดยการเก็บข้อมูลเก็บตามชนิดของข้อมูลซึ่งลักษณะของการเก็บข้อมูลนั้นแบ่งเป็นหลายกลุ่มดังนี้

- กลุ่มตัวเลข ประกอบไปด้วยชนิดข้อมูลตัวเลข ตัวนับ เกจ จำนวนช่วงเวลา และ Opaque
- กลุ่มสายอักขระ ประกอบด้วยชนิดข้อมูลสายอักขระเพียงชนิดเดียว
- กลุ่มข้อมูลว่าง บังคับให้เป็นข้อมูล 0x00 เท่านั้น
- กลุ่มหมายเลขโอไอดี
- กลุ่มเลขที่อยู่ไอพี
- กลุ่มข้อมูลชนิดพิเศษ ประกอบด้วยชนิดข้อมูล Opaque ซึ่งรวมกลุ่มข้อมูลชนิดอื่นด้วยการห่อหุ้มให้เป็นข้อมูลสายอักขระ และชนิดข้อมูลกลุ่มข้อมูลซึ่งรวมข้อมูลหลายๆ ชุดไว้ด้วยกัน

ภาคผนวก ค
โครงสร้างข้อมูลเอสเอ็นเอ็มพี

ข้อมูลเอสเอ็นเอ็มพีที่ติดต่อกันระหว่างเอสเอ็นเอ็มพีเอเจนต์ และระบบฝังตัวนั้นแบ่งออกเป็น 2 ลักษณะด้วยกัน คือ

1. โครงสร้างข้อมูลเอสเอ็นเอ็มพีสำหรับการร้องขอ หรือกำหนดค่าเอเจนต์ โดยการร้องขอและกำหนดค่านั้นมีลักษณะโครงสร้างข้อมูลพื้นฐานที่เหมือนกันต่างกันเพียงคำสั่งที่ระบุไปกับข้อมูลเอสเอ็นเอ็มพี (Command PDU) โดยโครงสร้างข้อมูลลักษณะนี้แสดงได้ดังรูป

รูปแสดงโครงสร้างข้อมูลการร้องขอและกำหนดค่าเอเจนต์

2. โครงสร้างข้อมูลเอสเอ็นเอ็มพีสำหรับการแจ้งข้อมูลแทรพให้กับระบบฝังตัว

รูปแสดงโครงสร้างข้อมูลแทรพ

ประวัติผู้เขียนวิทยานิพนธ์

นายชนินทร์ มหารักษ์ เกิดเมื่อวันที่ 3 มิถุนายน พ.ศ. 2522 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2543 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2544