ฟอลคอน: ตัววิเคราะห์แพคเก็ตเครือข่ายแบบปรับเปลี่ยนโปรโตคอลได้

FALCON: A Reconfigurable Protocol Analyzer

Technical Report : CPETR-42-001

สุรศักดิ์ สงวนพงษ์ และ อุรุพจน์ กัลยาสิริ
กลุ่มวิจัยเครือข่ายประยุกต์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ บางเขน กรุงเทพฯ 10900
โทร (02) 9428555(EXT:1433) โทรสาร 5796245 E-Mail:{nguan,g4265106}@ku.ac.th

บทคัดย่อ

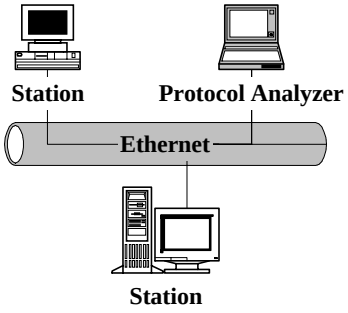
บทความนี้นำเสนอแนวคิดการออกแบบและพัฒนาตัววิเคราะห์โปรโตคอลสำหรับเครือข่ายคอมพิวเตอร์ที่มีคุณลักษณะพิเศษต่างจากระบบทั่วไป คือเปิดโอกาสให้ผู้ใช้งานสามารถเพิ่มหรือลดชนิดโปรโตคอลสำหรับการวิเคราะห์ได้โดยอิสระ ตัววิเคราะห์โปรโตคอลฟอลคอนพัฒนาขึ้นใช้งานบนระบบปฏิบัติการยูนิกซ์ผ่านเอ็กซ์วินโดวส์ โดยจับแพคเก็ตทั้งหมดในเครือข่ายนำมาวิเคราะห์ตามรูปแบบของโปรโตคอลที่ผู้ใช้กำหนด ดีมัลติเพล็กซ์โปรโตคอลทุกระดับชั้น พร้อมทั้งแสดงรายละเอียดของเฮดเดอร์และข้อมูลทั้งหมดในแพคเก็ต

Abstract

This article presents an approach to develop protocol analyzer with dynamic, run-time reconfiguration capability. Our approach relies on the protocol definition file, which describes the format and characteristics of protocol. With this technique, it is freely allowed to add, remove and update protocol entry without source code modification or recompilation. The configuration file can be modified by window configuration interface and manually modification is also allowed. FALCON, the first prototype of protocol analyzer, is implemented using this approach. FALCON captures packets from a live network in real time and provides packets details, various statistics, as well as tools for troubleshooting networks.

1. บทนำ

ระบบเครือข่ายคอมพิวเตอร์มีใช้อย่างแพร่หลายในองค์กรทั่วไป เมื่อผู้ดูแลระบบต้องการทราบสถิติของการใช้งานเครือข่าย หรือเครือข่ายเกิดปัญหาอย่างใดอย่างหนึ่งขึ้น จำเป็นต้องมีเครื่องมือสำหรับตรวจสอบและแก้ไข ตัววิเคราะห์โปรโตคอลทำหน้าที่ตรวจจับแพคเก็ตในเครือข่ายและนำมาวิเคราะห์ข้อมูลที่บรรจุอยู่ภายในแพคเก็ตนั้นเพื่อตรวจหาสาเหตุของปัญหาของเครือข่ายคอมพิวเตอร์ ตัววิเคราะห์โปรโตคอลอาจอยู่ในรูปของซอฟต์แวร์ที่ใช้งานได้กับคอมพิวเตอร์ทั่วไปหรือเป็นซอฟต์แวร์ที่ถูกออกแบบมาใช้กับคอมพิวเตอร์ที่มีฮาร์ดแวร์เฉพาะ เช่น สนิฟเฟอร์ [1] ซึ่งเป็นชื่อที่นิยมเรียกแทนตัววิเคราะห์โปรโตคอล ถึงแม้ว่าตัววิเคราะห์โปรโตคอลจะมีหน้าที่เฉพาะงานแต่ก็มีลักษณะสมบัติพื้นฐานและการจัดวางระบบเหมือนกับสถานีงานในเครือข่ายดังตัวอย่างในรูปที่ 1



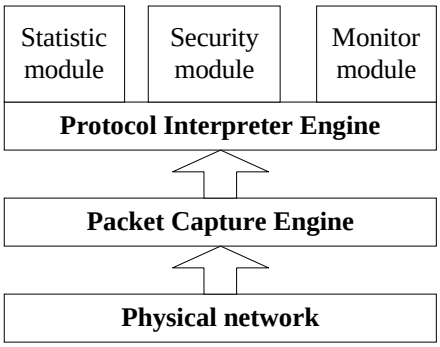
รูปที่ 1 ตัววิเคราะห์โปรโตคอลในเครือข่าย

ตัววิเคราะห์โปรโตคอลทั่วไปมักจะมีข้อจำกัดเหมือนกัน คือไม่สามารถวิเคราะห์โปรโตคอลได้ครอบคลุมทุกชนิด [2] เนื่องจากโปรโตคอลบางชนิดเป็นมาตรฐานที่นำมาใช้ใหม่ หรือบางชนิดเป็นโปรโตคอลส่วนตัวที่ผู้ใช้ออกแบบขึ้นเองเพื่อจุดประสงค์บางอย่าง โดยทั่วไปผู้พัฒนาตัววิเคราะห์โปรโตคอลจะแก้ไขข้อจำกัดนี้โดยการให้ผู้ใช้งานเขียนโปรแกรมเพิ่มเติมขึ้นเอง [3] เพื่อวิเคราะห์โปรโตคอลที่เพิ่มขึ้นโดยเฉพาะ ทำให้ยากต่อการใช้งานเนื่องจากผู้ใช้บางคนไม่มีความรู้ด้านเครือข่ายคอมพิวเตอร์และการเขียนโปรแกรมเพียงพอ งานวิจัยนี้มีแนวคิดที่จะแก้ไขข้อจำกัดโดยการแทนรูปแบบของโปรโตคอลในรูปของไฟล์กำหนดคุณสมบัติเพื่อให้ตัววิเคราะห์โปรโตคอลทราบรูปแบบและสามารถนำไปเปรียบเทียบกับโปรโตคอลที่ปรากฏอยู่ในแพคเก็ตจริง ทำให้วิเคราะห์ชนิดโปรโตคอล รายละเอียดของเฮดเดอร์ รวมถึงข้อมูลที่บรรจุทั้งหมดได้ ด้วยวิธีการนี้ฟอลคอนจะเป็นเครื่องมือที่มีประโยชน์สำหรับการวิเคราะห์แพคเก็ต ใช้งานง่ายและสะดวก เพราะสามารถเพิ่มหรือลดชนิดโปรโตคอลชนิดใดๆ ได้ตามต้องการ และไม่จำเป็นต้องเขียนโปรแกรมขึ้นเพิ่มเติม

เนื้อหาถัดไปจะกล่าวเป็นลำดับดังต่อไปนี้ หัวข้อที่ 2 เป็นการอธิบายรายละเอียดของโครงสร้างโดยทั่วไปของระบบ หัวข้อที่ 3 กล่าวถึงแนวคิดและรูปแบบโครงสร้างของไฟล์กำหนดคุณสมบัติโปรโตคอล หัวข้อที่ 4 แสดงผลการทดสอบการทำงานของระบบ และหัวข้อที่ 5 เป็นบทสรุปและแนวทางการพัฒนาต่อ

2. โครงสร้างของระบบ

ตัววิเคราะห์โปรโตคอลในระบบนี้ใช้แนวคิดการพัฒนาซอฟต์แวร์แบบโครงสร้าง เนื่องจากโปรแกรมทั้งหมดมีการทำงานเป็นลำดับทั้งสิ้น นอกจากแนวคิดนี้จะสะดวกต่อการพัฒนาแล้วยังง่ายต่อการแก้ไขข้อบกพร่องอีกด้วย ระบบนี้พัฒนาโดยใช้ภาษาซีบนระบบปฏิบัติการยูนิกซ์ ซึ่งเป็นระบบปฏิบัติการคอมพิวเตอร์ที่เหมาะกับการพัฒนาซอฟต์แวร์ที่เกี่ยวข้องกับระบบเครือข่าย นอกจากนั้นระบบปฏิบัติการยูนิกซ์ยังเป็นที่นิยมดังนั้นจึงมีซอฟต์แวร์จำนวนมากพัฒนาขึ้นภายใต้ระบบนี้ ทำให้ง่ายและสะดวกต่อการนำไลบรารีที่มีการพัฒนาแล้วมาประยุกต์ใช้งาน โครงสร้างโดยทั่วไปของระบบประกอบด้วยส่วนสำคัญ 3 ส่วนได้แก่ส่วนรับแพคเก็ตจากเครือข่าย ส่วนวิเคราะห์โปรโตคอล และส่วนโปรแกรมประยุกต์ ดังแสดงในรูปที่ 2

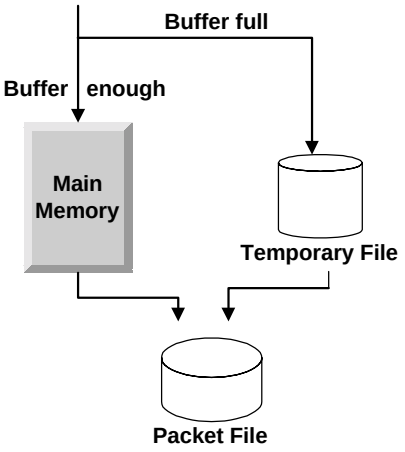


รูปที่ 2 โครงสร้างของระบบ

2.1 ส่วนรับแพคเก็ตจากเครือข่าย (Packet Capture Engine)

ในการส่งแพคเก็ตของระบบอีเธอร์เน็ต โฮสต์ต้นทางจะส่งแพคเก็ตกระจายไปในสายสัญญาณ ทุกโฮสต์ในเซกเมนต์จะตรวจแพคเก็ตที่ได้รับว่าส่งถึงตนเองหรือไม่ โดยการเปรียบเทียบฟิลด์แมคแอดเดรสปลายทางกับแมคแอดเดรสประจำตัว หากพบว่ามีค่าตรงกันก็จะรับแพคเก็ตนั้นไปดำเนินการ มิฉะนั้นจะปล่อยแพคเก็ตนั้นทิ้งไป ดังนั้นหลักการพื้นฐานของการสร้างส่วนรับแพคเก็ตจากเครือข่ายคือ การปรับการ์ดเครือข่ายให้อยู่ในภาวะรับทุกแพคเก็ตไปดำเนินการโดยไม่ตรวจสอบแมคแอดเดรสปลายทาง ภาวะนี้เรียกว่า โพรมิสคูอัส (Promiscuous) [4] และเป็นภาวะพิเศษที่สามารถปรับตั้งให้กับการ์ดเครือข่ายทั่วไป ยกเว้นการ์ดเครือข่ายบางผลิตภัณฑ์ที่ไม่อนุญาตให้มีภาวะการทำงานนี้เพื่อความปลอดภัยของระบบ ส่วนรับแพคเก็ตที่พัฒนาขึ้นมีการทำงานเป็นลำดับรูปแบบเดียวกันกับไลบรารี libpcap [5]

แพคเก็ตที่รับมาแต่ละครั้งจะถูกเก็บในโครงสร้างที่เหมาะสมโดยใช้หลักการของหน่วยความจำเสมือน (Virtual Memory) กล่าวคือมีแหล่งเก็บสองส่วนได้แก่ บัฟเฟอร์ซึ่งเป็นเนื้อที่ในหน่วยความจำหลักของระบบและดิสก์ แพคเก็ตจะถูกจัดเก็บในบัฟเฟอร์จนเต็มก่อน หลังจากนั้นจึงเก็บลงดิสก์ ดังแสดงในรูปที่ 3 เนื้อที่ที่ใช้ในการจัดเก็บมีขนาดไม่แน่นอนขึ้นอยู่กับจำนวนแพคเก็ต การออกแบบลักษณะนี้มีประโยชน์มากเพราะเป็นการใช้เนื้อที่เก็บให้คุ้มค่า และยังส่งผลถึงประสิทธิภาพการทำงานโดยรวมของโปรแกรมอีกด้วย



รูปที่ 3 การจัดเก็บแพคเก็ต

2.2 ส่วนวิเคราะห์โปรโตคอล (Protocol Interpreter Engine)

ส่วนวิเคราะห์โปรโตคอลมีหน้าที่สำคัญคือวิเคราะห์แพคเก็ตโดยเปรียบเทียบรูปแบบโปรโตคอลที่ปรากฏในแพคเก็ตนั้นกับโปรโตคอลที่อธิบายในไฟล์กำหนดคุณสมบัติที่ผู้ใช้กำหนดขึ้น โครงสร้างของไฟล์กำหนดคุณสมบัติโปรโตคอลที่ใช้อธิบายรูปแบบของโปรโตคอลจะอธิบายในหัวข้อที่ 3 หลังจากการวิเคราะห์โปรโตคอลเสร็จสิ้นแล้วจะมีการเก็บรายละเอียดต่างๆของผลที่ได้จากการวิเคราะห์แล้วส่งต่อให้ส่วนโปรแกรมประยุกต์เพื่อทำงานในรูปแบบต่างๆ ต่อไป การวิเคราะห์แพคเก็ตแบ่งเป็นสองประเภทคือ การวิเคราะห์ก่อนการจัดเก็บ และการวิเคราะห์หลังการจัดเก็บ การวิเคราะห์ก่อนการจัดเก็บ ได้แก่ การวิเคราะห์แมคแอดเดรส การวิเคราะห์โฮสต์ที่ไม่รู้จัก และการวิเคราะห์ชุดโปรโตคอล ส่วนการวิเคราะห์หลังการจัดเก็บ ประกอบด้วยการวิเคราะห์ชนิดโปรโตคอลทุกระดับชั้นที่ระบบรู้จัก การวิเคราะห์แมคแอดเดรส การวิเคราะห์โฮสต์ที่ไม่รู้จัก การดีมัลติเพล็กซ์แพคเก็ตทุกระดับชั้นและการแสดงข้อมูลในแพคเก็ตทั้งหมดด้วยเลขฐานสิบหก พร้อมกับอักขระแอสกีที่สามารถแสดงได้

2.3 ส่วนโปรแกรมประยุกต์

โปรแกรมประยุกต์เป็นส่วนของโปรแกรมที่ถูกพัฒนาขึ้นเพิ่มเติมเพื่อให้ระบบมีความสามารถหลากหลาย ตัววิเคราะห์โปรโตคอลฟอลคอนมีโปรแกรมประยุกต์ที่ได้พัฒนาขึ้นได้แก่ โปรแกรมแสดงสถิติที่ได้จากการจับแพคเก็ต [6] โปรแกรมตรวจสภาพโฮสต์และเครือข่าย [7] และ

โปรแกรมบริหารจัดการเครือข่ายโดยใช้โปรโตคอลเอสเอ็นเอ็มพี [8] เป็นต้น โปรแกรมแสดงสถิติการใช้งานเครือข่าย แสดงรายละเอียดต่างๆ ของโฮสต์ จำนวนแพคเก็ตบรอดคาส เปอร์เซ็นต์การใช้งานเครือข่าย มีการแสดงผลหลายรูปแบบ ได้แก่ กราฟแท่ง กราฟเส้น และตาราง เป็นต้น โปรแกรมตรวจสอบสภาวะเครือข่ายเป็นเครื่องมือพื้นฐานที่มีใช้กันในระบบปฏิบัติการยูนิกซ์ ซึ่งช่วยในการแสดงรายละเอียดสภาวะต่างๆ ได้แก่ โปรแกรม **ping** ใช้สำหรับตรวจสอบสภาวะของโฮสต์ โปรแกรม **netstat** ใช้แสดงคุณสมบัติทางเครือข่ายของโฮสต์ โปรแกรม **traceroute** ใช้แสดงเส้นทางการเชื่อมต่อจากโฮสต์ต้นทางไปยังโฮสต์ปลายทาง เครื่องมือพื้นฐานเหล่านี้ยังสามารถใช้ออพชันได้อีกหลายแบบ

โปรแกรมบริหารจัดการดูแลระบบเครือข่ายใช้งานเพื่อตรวจสถานะและข้อมูลของอุปกรณ์เครือข่ายที่มีเอสเอ็นเอ็มพีเอเจนต์ โปรแกรมนี้จะสร้างเอสเอ็นเอ็มพีเมสเสจเพื่อส่งไปถามข้อมูลที่เอสเอ็นเอ็นพีเอเจนต์โดยสามารถส่งชนิดของพีดียู (PDU: Protocol Data Unit) ทั้ง get request และ get next request และจะรอรับ get response ที่ตอบกลับมา

3. โครงสร้างไฟล์กำหนดคุณสมบัติโปรโตคอล

ตัวอย่างรูปแบบของไฟล์กำหนดคุณสมบัติของโปรโตคอลแสดงในรูปที่ 4 ประกอบด้วยชื่อโปรโตคอล (ProtocolName), รายละเอียดสั้นๆของโปรโตคอล (ProtocolDescription), ค่าระบุชนิดโปรโตคอล (ProtocolTypeValue) ในรูปตัวเลขฐานสิบ, ชื่อและขนาดของฟิลด์ (ProtocolField) โดยระบุขนาดเป็นจำนวนไบต์ เมื่อระบุจนครบทุกฟิลด์แล้วจะต้องมีข้อความจบฟิลด์ (End) และปิดท้ายด้วยรหัสจบโปรโตคอล (ProtocolEnd) ข้อความใดที่ปรากฏอยู่หลังเครื่องหมาย " # " จะถือว่าเป็นหมายเหตุและโปรแกรมจะละเว้นข้อความนั้นไปโดยไม่แปลความหมาย

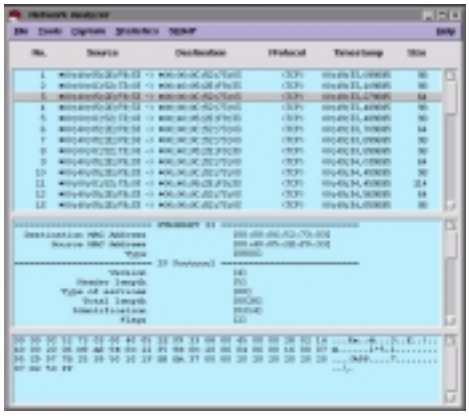
```
# Put the protocol name in UPPER CASE ONLY
# This is used for specify the dynamic protocol
ProtocolName=ARP
# [OPTION] Description of protocol
ProtocolDescription=Address Resolution Protocol
# The value of this protocol in decimal
# specified in lower-layer
ProtocolTypeValue=\
ETH2=2054
SNAP=2054
# Protocol field name and size of field
# in byte
ProtocolField=\
HardType=2
ProtTyep=2
HardSize=1
ProtSize=1
OP=2
SendAddr=6
SendIP=4
TargetAddr=6
TargetIP=4
# End of field
End=-1
# End of protocol
ProtocolEnd=ARP

ProtocolName=ICMP
ProtocolDescription=Message Error
ProtocolTypeValue=\
IP=1
ProtocolField=\
Type=1
Code=1
Checksum=2
End=-1
ProtocolEnd=ICMP

ProtocolName=MY_PROTOCOL
ProtocolDescription=My define protocol
ProtocolTypeValue=\
UDP=5353
TCP=5353
ProtocolField=\
AskId=2
AnswerId=2
Flags=2
NumberOfQuestions=1
NumberOfAnswers=1
NumberOfServices=2
ServiceRequest=2
ServiceReply=2
End=-1
ProtocolEnd=MY_PROTOCOL
```

รูปที่ 4 ไฟล์กำหนดคุณสมบัติโปรโตคอล

เมื่อผู้ใช้งานต้องการเพิ่มหรือลดชนิดโปรโตคอลสำหรับการวิเคราะห์ครั้งต่อไป สามารถแก้ไขเพิ่มเติมไฟล์กำหนดคุณสมบัติโปรโตคอลได้โดยตรง (ในระบบนี้ใช้ชื่อ protocol.conf) แต่หากไม่สะดวกหรือยุ่งยาก ระบบก็มีอินเทอร์เฟสบนเอ็กซ์วินโดวส์ให้ใช้เช่นกัน เมื่อกำหนดคุณสมบัติทั้งหมดเสร็จสิ้นแล้ว ระบบจะสร้างฐานข้อมูลโปรโตคอลโดยอัตโนมัติ ผู้ใช้สามารถเรียกใช้โปรแกรมได้ทันทีโดยไม่ต้องคอมไพล์โปรแกรมซ้ำอีกครั้ง

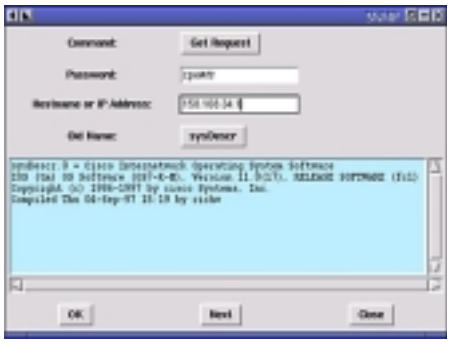
4. การทดสอบการทำงาน

ระบบนี้ได้ทดสอบทั้งภายใต้สถานการณ์จริง และสถานการณ์จำลองที่สร้างขึ้น โดยทดสอบกับเครือข่ายอีเธอร์เน็ต หน้าจอหลักของระบบแสดงดังรูปที่ 5 การทดสอบโดยการจับแพคเก็ตที่มีขนาดสูงสุดคือ 1,518 ไบต์ จำนวน 1,000,000 แพคเก็ต และแพคเก็ตขนาดต่ำสุดคือ 64 ไบต์ จำนวน 1,000,000 แพคเก็ต ภายใต้อัตราการส่งผ่านข้อมูลที่ 2, 4, 6, 8 และ 10 เมกะบิตต่อวินาที ผลที่ได้คือโปรแกรมสามารถจับแพคเก็ตได้ทั้งหมดโดยไม่มีแพคเก็ตสูญหาย



รูปที่ 5 หน้าจอหลัก

เมื่อทดสอบโปรแกรมตรวจสอบเครือข่ายโดยใช้โปรโตคอลเอสเอ็นเอ็มพีโดยการเลือก get request ผลที่ได้จะแสดงในรูปที่ 6 การใช้งานจะง่าย เพราะมีชนิดของคำสั่งกับอ็อบเจ็กต์ไอเดนติไฟเออร์ให้ผู้ใช้สามารถเลือกจากลิสต์ได้ สำหรับในรูปที่ 7 แสดงจำนวนแพคเก็ตเมื่อจำแนกตามชุดโปรโตคอลหลังจากการวิเคราะห์เสร็จสิ้น



รูปที่ 6 การตรวจสอบเครือข่ายโดยใช้ get request

รูปที่ 7 จำนวนแพคเก็ตจำแนกตามชุดโปรโตคอล

5. บทสรุปและแนวทางพัฒนาต่อ

ระบบปัจจุบันมีข้อจำกัดเรื่องระดับชั้นการนิยามโปรโตคอล คือสามารถกำหนดได้เฉพาะโปรโตคอลระดับชั้นบนสุดที่อยู่เหนือโปรโตคอลดังต่อไปนี้ (1) โปรโตคอลในระดับดาต้าลิงค์ทุกชุดโปรโตคอล (2) โปรโตคอลไอพี ทีซีพี และยูดีพี สำหรับชุดโปรโตคอลทีซีพี/ไอพี (3) โปรโตคอลไอพีเอ็กซ์ สำหรับชุดโปรโตคอลโนเวล

การระบุฟิลด์สำหรับโปรโตคอลที่เพิ่มนั้นทำได้เฉพาะฟิลด์ที่มีขนาดเป็นไบต์ เช่น ถ้าฟิลด์มีขนาด 3 บิตต้องแทนด้วยขนาด 1 ไบต์ ทำให้ยากต่อการวิเคราะห์ หรือหากในกรณีที่ฟิลด์ของโปรโตคอลชนิดใดที่มีขนาดไม่แน่นอนหรือมีได้หลายฟิลด์ต่อกันเป็นจำนวนไม่คงที่แล้ว โปรแกรมจะไม่สามารถระบุได้ว่าฟิลด์นั้นมีค่าเท่าใด แนวคิดการพัฒนาต่อไปในอนาคตของระบบนี้คือ การสร้างภาษาหรือรูปแบบการแทนโครงสร้างของโปรโตคอลทุกชนิดในทุกระดับชั้นซึ่งจะแก้ข้อจำกัดดังกล่าวข้างต้นได้ทั้งหมด

เอกสารอ้างอิง

[1] Network General Corporation. &copy1998 Network Associates, Inc. Makers of the one and only Sniffer Network Analyzer and Distributed Sniffer System

[2] นภัทร สระเอี่ยม, กอบชัย เดชหาญ, ชัยรินทร์ สูนย์ขันและเอกชัย พรหมมาส, “การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี”, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 20, หน้า 569-573

[3] Brecht Claerhout, “Sniffit Page.”, http://reptile.rug.ac.be/~coder/sniffit/sniffit.html,1998.

[4] Christopher Klaus, “Sniffer FAQ.”, http://www.iss.net/vd/packcapt.html, 1998.

[5] Steve McCanne, C. Leres and V. Jacobson, “libpcap.”, http://ee.lbl.gov, 1994.

[6] นันทิยา วิรัชอมรพันธุ์, “โปรแกรมแสดงผลการวิเคราะห์เครือข่าย”, โครงงานวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์, 2542.

[7] ศุภฤกษ์ อุดมเจริญสุข,“โปรแกรมตรวจสอบสภาวะเครือข่ายบนระบบเอ็กซ์วินโดวส์”, โครงงานวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์, 2542.

[8] วงศ์ชัย สุริยนนท์รินทร์, “คลังโปรแกรมบริหารและจัดการบนระบบเครือข่าย”, โครงงานวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์, 2542.