การประเมินสมรรถนะของโพรโทคอลค้นหาเส้นทางที่มีความมั่นคง บนเครือข่ายเฉพาะกิจเคลื่อนที่

PERFORMANCE EVALUATION OF SECURE ROUTING PROTOCOL ON MOBILE AD-HOC NETWORK

โดย

ปานแก้ว รัตนสิริภัทร

สิบมนัส เชิดเกียรติศักดิ์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2551

PERFORMANCE EVALUATION OF SECURE ROUTING PROTOCOL ON MOBILE AD-HOC NETWORK

PANKAEW RATTANASIRIPAT

SIBMANUS CHIRDKIATISAK

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2008

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2551

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประเมินสมรรถนะของโพรโทคอลค้นหาเส้นทางที่มีความมั่นคงบนเครือข่ายเฉพาะกิจเคลื่อนที่

Performance Evaluation of Secure Routing Protocol on Mobile Ad-hoc Network

ผู้จัดทำ

1. นางสาวปานแก้ว รัตนสิริภัทร รหัสประจำตัว 48070140
2. นางสาวสิบมนัส เชิดเกียรติศักดิ์ รหัสประจำตัว 48070170

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. จันทร์บูรณ์ สถิตวิริยวงศ์)

หัวข้อ	การประเมินสมรรถนะของโพรโทคอลค้นหาเส้นทางที่ความมั่นคงบนเครือข่ายเฉพาะกิจเคลื่อนที่
นักศึกษา	นางสาวปานแก้ว รัตนสิริภัทร นางสาวสิบมนัส เชิดเกียรติศักดิ์
รหัสนักศึกษา	48070140 48070170
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

ในโครงงานเครือข่ายเฉพาะกิจเคลื่อนที่ส่วนใหญ่ ได้ชี้ประเด็นสำคัญถึงการจัดหาบริการเรื่องเส้นทางโดยไม่คำนึงถึงการพิจารณาเรื่องความปลอดภัย เราจึงให้รายละเอียดความสำคัญของการต่อต้านภัยคุกคาม ที่จะเกิดขึ้นในเส้นทางของเครือข่ายเฉพาะกิจ โดยเฉพาะอย่างยิ่งในส่วนของโพรโตคอลเอโอดีวี ปริญญานิพนธ์เล่มนี้ได้วิเคราะห์โพรโทคอลที่มีความมั่นคงในเครือข่ายไร้สายเฉพาะกิจ ซึ่งโพรโทคอลเออาร์เอเอ็น ถูกจัดเป็นโพรโทคอลค้นหาเส้นทางที่มีความปลอดภัยแบบประเภทรีแอกทีฟ (ตามการร้องขอของเส้นทาง) นั่นหมายความว่าโพรโทคอลเออาร์เอเอ็น ไม่มีการส่งข้อมูลในการอัพเดทในเรื่องโทโพโลยีของเครือข่าย แต่จะมีการทำงานค้นหาเส้นทางไปยังปลายทาง โพรโทคอลเออาร์เอเอ็น อยู่บนพื้นฐานของการรับรอง (Certificate) และสามารถขจัดการโจมตีที่เข้ามาหลอกในเรื่องของเส้นทางได้สำเร็จโดยใช้การรับรองกับเฉพาะเส้นทางที่มีการเซ็นต์ (Sign) เท่านั้น

Project Title	Performance Evaluation of Secure Routing Protocol on Mobile Ad-hoc Network
Student	Miss Pankaew Rattanasiripat Miss Sibmanus Chirdkiatisak
Student ID.	48070140 48070170
Degree	Bachelor of Science
Programme	Information Technology
Academic Year	2008
Advisor	Asst. Prof. Dr. Chanboon Sathitwiriyawong

ABSTRACT

Most recent ad hoc network research has focused on providing routing services without considering security. We detail security threats against ad hoc routing protocols, specifically examining AODV. In this paper, we analyze one of the secure mobile ad hoc networks protocols, which is Authenticated routing for ad hoc networks (ARAN). Such protocol is classified as a secure reactive routing protocol, which is based on some type of query-reply dialog. That means ARAN does not attempt to continuously maintain the up-to-date topology of the network, but rather when there is a need, it invokes a function to find a route to the destination. Our protocol, ARAN, is based on certificates and successfully defeats all identified attacks.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จเรียบร้อยด้วยดี เพราะได้รับความกรุณาอย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร. โชติพัชร์ ภรณวลัย ผู้ช่วยศาสตราจารย์อัครินทร์ คุณกิตติ และอาจารย์ลภัส ประดิษฐ์ทัศนีย์ กรรมการสอบ ที่กรุณาให้แนวคิด คำปรึกษา และตรวจแก้ไขข้อบกพร่องต่างๆ ทำให้ปริญญานิพนธ์เล่มนี้มีความถูกต้องสมบูรณ์มากยิ่งขึ้น ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณคุณพ่อ และคุณแม่ ที่ได้ให้การสนับสนุนทั้งด้านกำลังใจและการเงินเสมอมา

ขอกราบขอบพระคุณอาจารย์คณะเทคโนโลยีสารสนเทศที่มอบความรู้ทั้งในห้องเรียนและนอกห้องเรียนให้กับผู้จัดทำมาตลอดระยะเวลาการศึกษา 4 ปี

ขอขอบคุณเพื่อนๆพี่ๆ นักศึกษาปริญญาโท และปริญญาเอก สาขาเทคโนโลยีสารสนเทศ ที่ให้ความช่วยเหลือคำแนะนำต่างๆ และแลกเปลี่ยนประสบการณ์

ท้ายสุดนี้ คุณค่าและประโยชน์ของปริญญานิพนธ์เล่มนี้ ขอมอบให้คุณพ่อ คุณแม่ และคณาจารย์ทุกท่านผู้ประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำ

ปานแก้ว รัตนสิริภัทร

สิบมนัส เชิดเกียรติศักดิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในสภาพแวดล้อมของเครือข่ายไร้สายเฉพาะกิจ จะมีข้อจำกัดหลายอย่างคือ ข้อจำกัดในเรื่องของช่วงความถี่คลื่นวิทยุที่ใช้ และโครงข่ายการติดต่อสื่อสารกับตัวกลางที่ใช้ในการเชื่อมต่อก็มีการเปลี่ยนแบบรูปแบบบ่อย แต่ในการใช้เครือข่ายชนิดนี้ประหยัดพลังงาน จึงทำให้สามารถเคลื่อนที่ได้สะดวกกว่าการเชื่อมต่อเครือข่ายแบบโครงสร้าง (Infrastructure)

แต่ยังมีข้อด้อยอีกคือ เครือข่ายยังไม่มีระบบความปลอดภัยเพียงพอ ทำให้มีผู้ที่ต้องการเข้ามาโจมตีหรือก่อกวนเครือข่าย ซึ่งมีทั้งการโจมตีแบบการทำลายการหาเส้นทาง (routing disruption) ผู้เข้าโจมตีมักจะพยายามทำให้ข้อมูลในแพ็คเกตค้นหาเส้นทางผิดปกติไป ทำให้ค้นหาเส้นทางในวิธีที่ผิดปกติ, การโจมตีแบบทำให้ทรัพยากรในเครือข่ายหมดไป (resource consumption) ผู้เข้าโจมตีมักจะแพร่กระจายแพ็คเกตที่ไม่ได้ใช้ประโยชน์เข้าไปในเครือข่ายโดยพยายามให้มีการใช้ทรัพยากรภายในเครือข่ายนั้นๆ ให้มากที่สุด, โหนดที่เห็นแก่ตัวหรือโหนดที่พฤติกรรมผิดปกติ (selfish node or misbehavior node) โหนดผู้ไม่ประสงค์ดีในระบบประพฤติตัวไปในทางที่เห็นแก่ตัวพยายามไม่ปฏิบัติตามกฎของการส่งต่อข้อมูล พยายามทำให้การให้บริการของเครือข่าย (Service Availability) ไม่สามารถทำงานต่อไปได้ หรือทำงานได้ประสิทธิภาพน้อยลง เช่น โหนดทิ้งแพ็คเกตในการส่งต่อข้อมูลของโพรโทคอลการหาเส้นทางอย่างไม่มีเหตุผล ทำให้โพรโทคอลในการหาเส้นทางไม่สามารถปรับปรุงเส้นทางให้เป็นไปตามการทำงานปกติได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงงานฉบับนี้มุ่งหวังเพื่อการศึกษาและทดสอบโพรโทคอลที่ใช้จัดหาเส้นทางแบบเอโอดีวีและนำไปเปรียบเทียบกับโพรโทคอลที่มีความมั่นคงปลอดภัยเออาร์เอเอ็น เพื่อดูประสิทธิภาพในการใช้งาน

1.3 สมมติฐานของการศึกษา

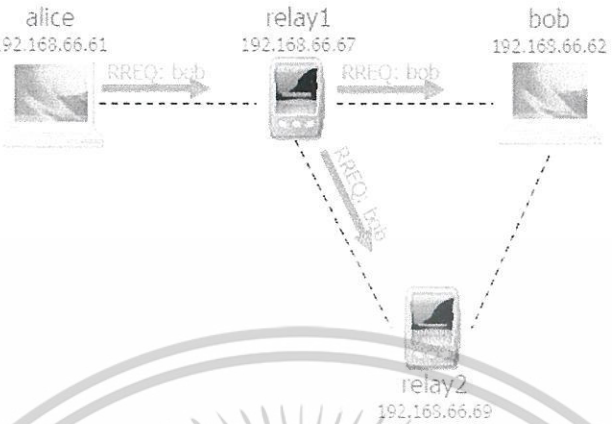
โหนดที่อยู่ในเครือข่ายไร้สายจะมีเลขที่ประจำโหนดที่เป็นเลขไม่ซ้ำกับโหนดอื่นๆในเครือข่าย แต่ละโหนดในเครือข่ายไร้สายมีหน่วยความจำการหาเส้นทาง (Routing cache) ซึ่งจะปรากฏในโพรโทคอลการหาเส้นทางที่ขึ้นอยู่กับความต้องการ (On Demand) ดังนั้นแต่ละโหนดมีเส้นทางสำรองในเครือข่าย โดยมีการกำหนดคำนิยามที่ใช้ในโพรโทคอลความปลอดภัยในงานวิจัยมีดังนี้

โหนดผู้ไม่ประสงค์ดี (Bad Node) ใช้เรียกสถานีไร้สายที่ไม่ประพฤติตัวตามกฎของการหาเส้นทาง สถานีไร้สายที่พยายามโจมตี หรือประสงค์ร้ายทำให้เครือข่ายไร้สายไม่สามารถทำงานได้ หรือจงใจทำให้ประสิทธิภาพโดยรวมของระบบลดลงอย่างไม่สมเหตุสมผล

รูปแบบพฤติกรรมที่ผิดปกติ (Misbehavior Model) เป็นพฤติกรรมที่โหนดผู้ไม่ประสงค์ดีในระบบประพฤติตัวไปในทางที่เห็นแก่ตัว ไม่พยายามปฏิบัติตามกฎของการส่งต่อข้อมูล พยายามทำให้การให้บริการของเครือข่าย (Service Availability) ไม่สามารถทำงานต่อไปได้ หรือทำงานได้ประสิทธิภาพน้อยลง

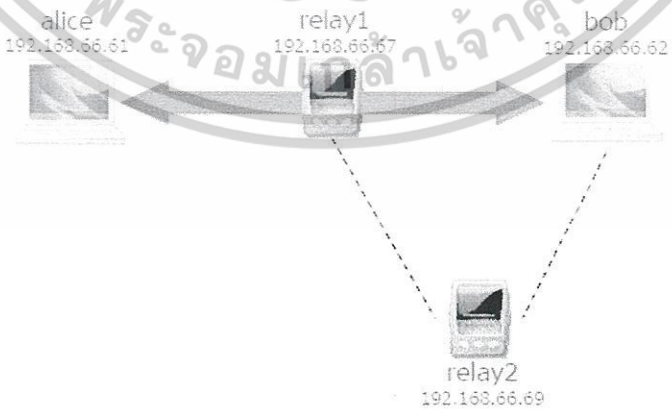
รูปแบบพฤติกรรมที่ก่อกวนเครือข่าย (Attacker Model) เป็นพฤติกรรมที่โหนดผู้ไม่ประสงค์ดีในระบบพยายามร่วมกันในการโจมตีระบบ และพยายามที่จะปลอมแปลงเพื่อถอดรหัสออกดูข้อมูล

ตัวอย่าง การโดนโจมตีบนการใช้โพรโทคอลเอโอดีวี

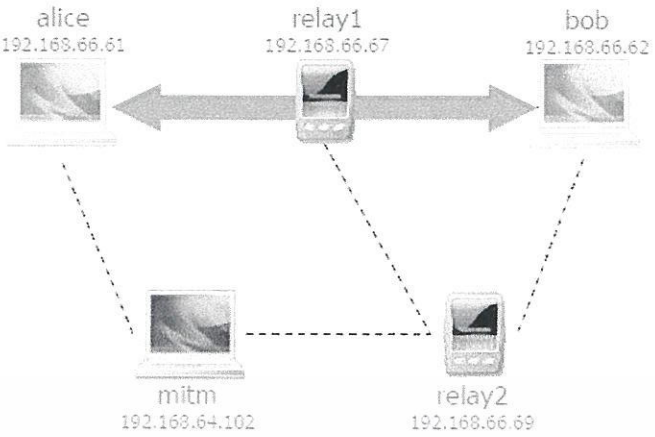


รูปที่ 1.1 alice ทำการส่งร้องขอเส้นทางไปหา bob

รูปที่ 1.2 bobตอบกลับ

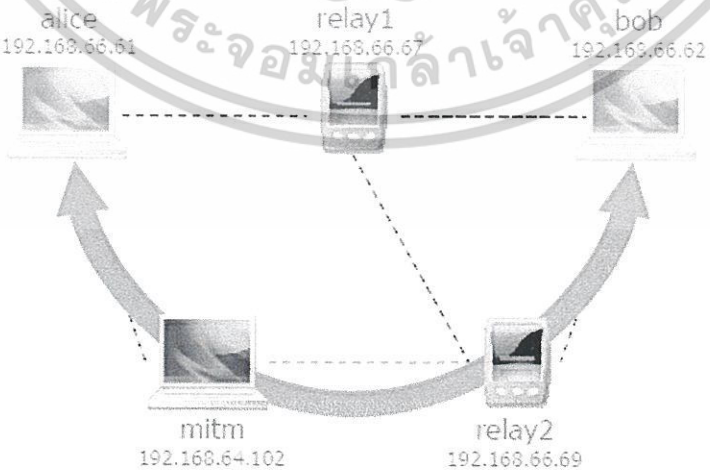


รูปที่ 1.3 เส้นทางถูกสร้างขึ้นระหว่าง alice กับ bob เส้นทางเป็นไปตาม Relay 1



รูปที่ 1.4 มีผู้โจมตีชื่อ mitm เข้ามา

รูปที่ 1.5 ผู้โจมตีส่ง RREP หลอกไปยัง alice และ bob



รูปที่ 1.6 เส้นทางการส่งเปลี่ยนไป โดยผ่านทางผู้โจมตี

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา

ในการสร้างความเชื่อมั่นของการค้นหาเส้นทางจะขึ้นอยู่กับความไว้วางใจระหว่างโหนด เนื่องจากโหนดแต่ละโหนดสามารถเคลื่อนที่ได้อย่างเป็นอิสระ ทำให้การกำหนดความไว้วางใจระหว่างกันทำได้ยาก เพราะโหนดที่เข้ามาร่วมในเครือข่ายไม่สามารถบ่งบอกได้ว่าเป็นโหนดที่มีเจตนามาทำการโจมตีหรือไม่ ในการจัดการกับความเสี่ยงที่จะเกิดขึ้น โหนดในเครือข่ายจะต้องมีการพิสูจน์โหนดที่กำลังจะติดต่อสื่อสารด้วยว่ามีความปลอดภัยมากน้อยเพียงใด ก่อนที่จะกำหนดเส้นทางในการส่งข้อมูลผ่านโหนดนั้น

1.5 ขอบเขตการศึกษา

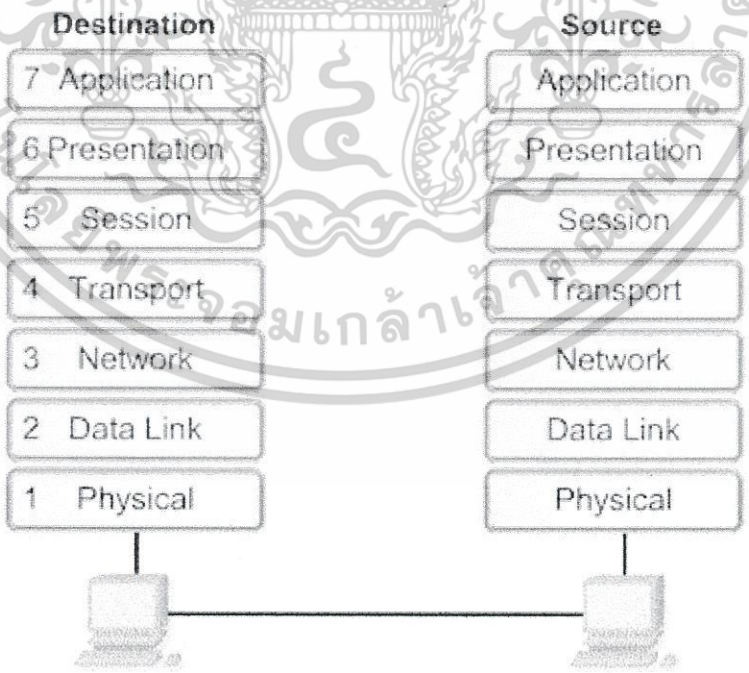
ในโครงงานนี้ได้ทำการศึกษาและทดลองการค้นหาเส้นทางเพื่อการใช้งานในเครือข่ายเฉพาะกิจเคลื่อนที่ทำให้การส่งข้อมูลระหว่างโหนดมีความปลอดภัยต่อการโจมตีของผู้ไม่ประสงค์ดี ทั้งยังสามารถระบุโหนดที่มีการติดต่อสื่อสารกันด้วยและแต่ละโหนดในเครือข่ายจะต้องมีคุณสมบัติที่เหมือนกัน ในการติดต่อสื่อสารนั้นไม่มีโหนดที่ใช้เป็นศูนย์กลางและมีการสื่อสารกันแบบสองทิศทาง แต่ละโหนดสามารถเคลื่อนที่ได้อย่างเป็นอิสระ

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในการศึกษา

2.1 แบบจำลองโอเอสไอ (OSI Model)

แบบจำลองโอเอสไอ เป็นแบบจำลองมาตรฐานในการสื่อสารซึ่งมีวัตถุประสงค์ ใช้สำหรับการสื่อสารระหว่างระบบ 2 ระบบ ระบบจะเปิดการติดต่อสื่อสารในเค้าโครงสำหรับออกแบบระบบเครือข่าย จะอนุญาตให้สื่อสารข้ามทุกรูปแบบของระบบคอมพิวเตอร์ในแต่ละชั้นแต่เกี่ยวเนื่องกันและเป็นรูปแบบมาตรฐานโอเอสไอ แบ่งการทำงานของระบบเครือข่ายออกเป็น 7 ชั้น ดังนี้

1. Physical Layer
2. Data link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer



รูปที่ 2.1 แสดงมาตรฐานการสื่อสารของแบบจำลองโอเอสไอ

แต่ละชั้นของแบบการสื่อสารข้อมูลเรียกว่าเลเยอร์ ประกอบด้วยเลเยอร์ย่อย ๆ ทั้งหมด 7 เลเยอร์ แต่ละชั้นทำหน้าที่รับส่งข้อมูลกับชั้นที่อยู่ติดกับตัวเองเท่านั้นจะไม่ติดต่อกระโดดข้ามไปยังชั้นอื่น ๆ เช่น เลเยอร์ 6 จะติดต่อกับเลเยอร์ 5 และ เลเยอร์ 7 เท่านั้นและการส่งข้อมูลจะทำไล่จากเลเยอร์ 7 ลงมาจนถึง เลเยอร์ 1 ซึ่งเป็นชั้นที่มีการเชื่อมต่อทางกายภาพ จากนั้นข้อมูลจะถูกส่งไปยังเครื่องผู้รับปลายทางโดยเริ่มจาก เลเยอร์1 ข้อมูลก็จะถูกถอดรหัส และส่งขึ้นไปตามเลเยอร์จนถึงเลเยอร์ 7 ก็จะประกอบกลับมาเป็นข้อมูล นำไปส่งให้โปรแกรมประยุกต์นำไปใช้แสดงผลต่อไป

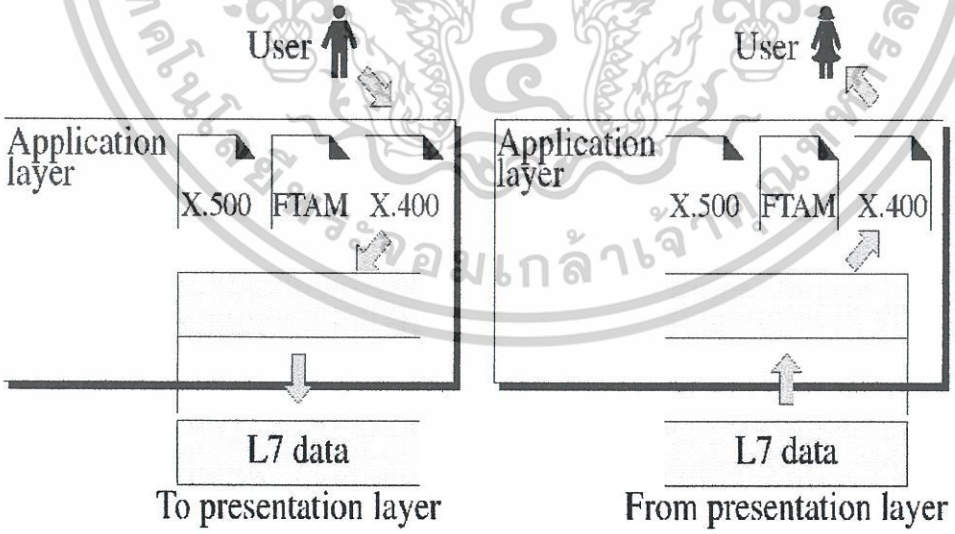
แบบจำลองโอเอสไอ ได้แบ่ง ตามลักษณะออกเป็น 2 กลุ่มใหญ่ ได้แก่

1. Application-oriented Layer เป็น 4 เลเยอร์ ด้านบนคือ เลเยอร์ ที่ 7,6,5,4 ทำหน้าที่เชื่อมต่อรับส่งข้อมูลระหว่างผู้ใช้กับโปรแกรมประยุกต์เพื่อให้รับส่งข้อมูลกับฮาร์ดแวร์ที่อยู่ชั้นล่างได้อย่างถูกต้อง ซึ่งจะเกี่ยวข้องกับซอฟแวร์เป็นหลัก

2. Network-dependent Layer เป็น 3 เลเยอร์ ด้านล่าง ทำหน้าที่เกี่ยวกับการรับส่งข้อมูลผ่านสายส่ง และควบคุมการรับส่งข้อมูลตรวจสอบข้อผิดพลาด รวมทั้งเลือกเส้นทางที่ใช้ในการรับส่ง ซึ่งจะเกี่ยวข้องกับฮาร์ดแวร์เป็นหลัก

หน้าที่ของแต่ละ เลเยอร์ของแบบจำลองโอเอสไอ

Layer 7: Application Layer



รูปที่ 2.2 แสดงหน้าที่การทำงานของ Application Layer

เป็นชั้นที่อยู่บนสุดของขบวนการรับส่งข้อมูล (ในแบบจำลองโอเอสไอเป็น เลเยอร์ 7 แต่ ทีซีพี/ไอพี (TCP/IP) ในเป็นเลเยอร์ 4 ทำหน้าที่ติดต่อกับผู้ใช้ โดยจะรับคำสั่งต่าง ๆ จากผู้ใช้ส่งให้คอมพิวเตอร์แปลความหมาย และทำงานตามคำสั่งที่ได้รับในระดับโปรแกรมประยุกต์ ซึ่งแบบจำลองโอเอสไอ Application Layer จะทำการแจกจ่ายร้องขอไปให้ Presentation Layer ด้วย โดยโพรโทคอลที่ใช้ใน แบบจำลองโอเอสไอนั้นจะแตกต่างจากโพรโทคอลที่ใช้ในทีซีพี/ไอพี เช่น

1. โปรแกรมประยุกต์ทางด้านรับ-ส่งไฟล์ ในแบบจำลองโอเอสไอใช้โพรโทคอล FTAM แต่ในแบบจำลอง ทีซีพี/ไอพี ใช้โพรโทคอล FTP เป็นต้น
2. การรับ-ส่งเมล์ในเครื่องลูกข่าย (clients) ในแบบจำลองโอเอสไอใช้โพรโทคอล X.400 แต่ในแบบจำลอง ทีซีพี/ไอพี ใช้โพรโทคอล SMTP หรือ POP3 หรือ IMAP เป็นต้น
3. เว็บบราวซ์เซอร์ (web browsers) ในแบบจำลองโอเอสไม่มีโพรโทคอลเป็นของตนเองให้ใช้ แต่ในแบบจำลอง ทีซีพี/ไอพี ใช้โพรโทคอล HTTP เป็นต้น

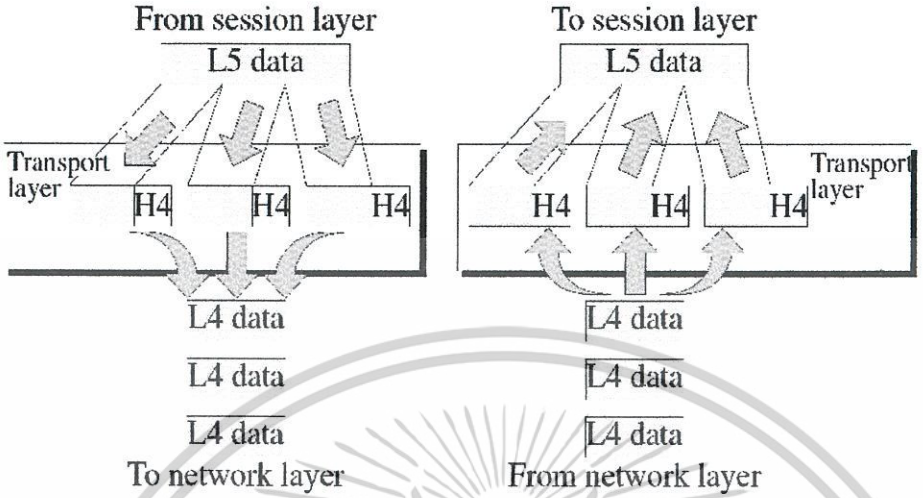
Layer 6: Presentation Layer

เป็นชั้นที่ทำหน้าที่ตกลงกับคอมพิวเตอร์อีกด้านหนึ่งในระดับชั้นเดียวกันว่า การรับส่งข้อมูลในระดับโปรแกรมประยุกต์จะมีขั้นตอนและข้อบังคับอย่างไร ข้อมูลที่รับส่งกันในเลเยอร์ที่ 6 จะอยู่ในรูปแบบของข้อมูลชั้นสูงมีกฎ (Syntax) บังคับแน่นอน เช่น ในการคัดลอกไฟล์จะมีขั้นตอนย่อยประกอบกัน คือสร้างไฟล์ที่กำหนดขึ้นมาเสียก่อน จากนั้นจึงเปิดไฟล์แล้วทำการรับข้อมูลจากปลายทางลงมาเก็บลงในไฟล์ที่สร้างขึ้นใหม่นี้ โดยเนื้อหาของข้อมูลที่ทำการรับส่งระหว่างกัน ก็คือคำสั่งของขั้นตอนย่อย ๆ ข้างต้น นอกจากนี้เลเยอร์ที่ 6 ยังทำหน้าที่แปลคำสั่งที่ได้รับจากเลเยอร์ที่ 7 ให้เป็นคำสั่งระดับปฏิบัติการส่งให้เลเยอร์ที่ 5 ต่อไป

Layer 5: Session Layer

ทำหน้าที่ควบคุม "จังหวะ" ในการรับส่งข้อมูลของคอมพิวเตอร์ทั้งสองด้าน ที่รับส่งแลกเปลี่ยนข้อมูลกันให้มีความสอดคล้องกัน (Synchronization) และกำหนดวิธีที่ใช้ในการรับส่งข้อมูล เช่น อาจจะเป็นในการสลับกันส่ง (Half Duplex) หรือการรับส่งข้อมูลพร้อมกันทั้งสองด้าน (Full Duplex) ข้อมูลที่รับส่งในเลเยอร์ที่ 5 จะอยู่ในรูปการสนทนา (Dialog) หรือประโยคสนทนาโต้ตอบกันระหว่างด้านรับและด้านส่งข้อมูล เช่น เมื่อได้รับข้อมูลส่วนแรกจากผู้ส่ง ก็จะตอบโต้กลับให้ผู้ส่งได้รู้ว่าได้รับข้อมูลส่วนแรกแล้ว พร้อมที่จะรับข้อมูลส่วนถัดไป ซึ่งคล้ายกับการสนนาโต้ตอบกันระหว่างผู้รับและผู้ส่งนั่นเอง

Layer 4: Transport Layer

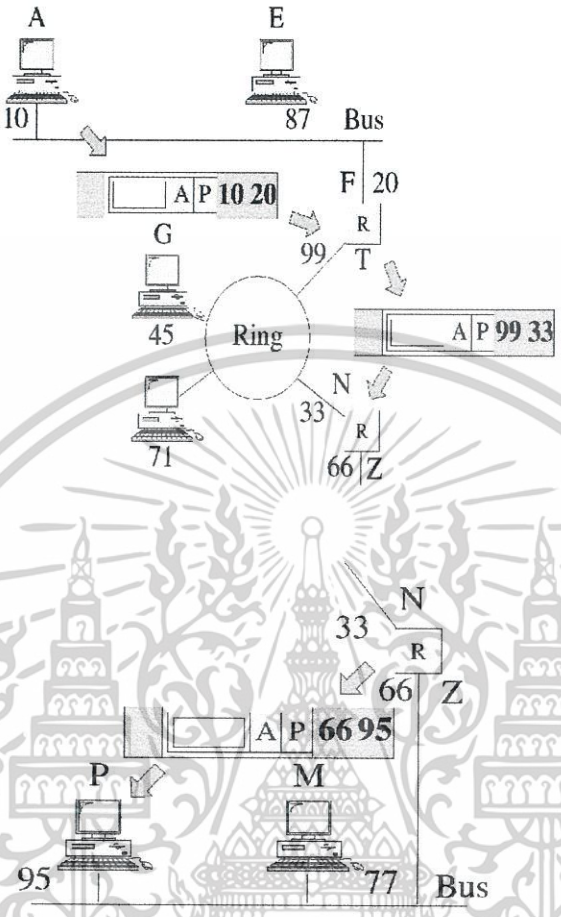


รูปที่ 2.3 แสดงหน้าที่การทำงานของ Transport Layer

ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลระดับสูงของเลเยอร์ที่ 5 มาเป็นข้อมูลที่รับส่งในระดับฮาร์ดแวร์ เช่น แปลงค่าหรือชื่อของเครื่องคอมพิวเตอร์ในเครือข่ายให้เป็นแอ็กเวิร์คแอดเดรสพร้อมทั้งเป็นชั้นที่ควบคุมการรับส่งข้อมูลจากปลายด้านส่งถึงปลายด้านรับข้อมูลให้ข้อมูลมีการไหลลื่นตลอดเส้นทางตามจังหวะที่ควบคุมจากเลเยอร์ที่ 5 โดยในเลเยอร์ที่ 4 นี้ จะเป็นรอยต่อระหว่างการรับส่งข้อมูลซอฟท์แวร์กับฮาร์ดแวร์การรับส่งข้อมูลของระดับสูงจะถูกแยกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลที่เลเยอร์ที่ 4 และจะไม่มีส่วนใดผูกติดกับฮาร์ดแวร์ที่ใช้รับส่งข้อมูลในระดับล่าง ดังนั้นฮาร์ดแวร์และซอฟท์แวร์ที่ใช้ควบคุมการรับส่งข้อมูลในระดับล่างลงไปจากเลเยอร์ที่ 4 จึงสามารถสับเปลี่ยน และใช้ข้ามไปมากับซอฟท์แวร์รับส่งข้อมูลในระดับที่อยู่ข้างบน (ตั้งแต่เลเยอร์ที่ 4 ขึ้นไปถึงเลเยอร์ที่ 7) ได้ง่าย

หน้าที่อีกประการหนึ่งของเลเยอร์ที่ 4 คือ การควบคุมคุณภาพการรับส่งข้อมูลให้มีมาตรฐานในระดับที่ตกลงกันทั้งสองฝ่าย และการตัดข้อมูลออกเป็นส่วนย่อย ๆ ให้เหมาะกับลักษณะการทำงานของฮาร์ดแวร์ที่ใช้ในเครือข่าย เช่น หากเลเยอร์ที่ 5 ต้องการส่งข้อมูลที่มีความยาวเกินกว่าที่ระบบเครือข่ายที่จะส่งให้เลเยอร์ที่ 4 ก็จะทำหน้าที่ตัดข้อมูลออกเป็นส่วนย่อย ๆ แล้วส่งไปให้ผู้รับ ข้อมูลที่ได้รับปลายทางก็จะถูกนำมาต่อกันที่เลเยอร์ที่ 4 ของด้านผู้รับ และส่งไปให้เลเยอร์ที่ 5 ต่อไป

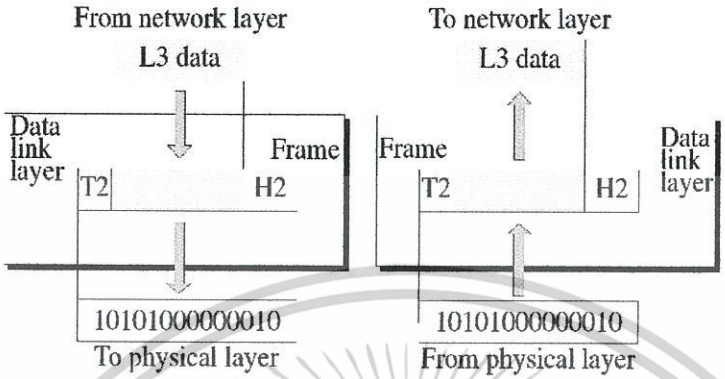
Layer 3: Network Layer



รูปที่ 2.4 แสดงหน้าที่การทำงานของ Network Layer

ทำหน้าที่เชื่อมต่อคอมพิวเตอร์ด้านรับ และด้านส่งเข้าหากันผ่านระบบเครือข่าย พร้อมทั้งเลือกหรือกำหนดเส้นทางที่จะใช้ในการรับส่งข้อมูลระหว่างกัน และส่งผ่านข้อมูลที่ได้รับไปยังอุปกรณ์ในเครือข่ายต่าง ๆ จนกระทั่งถึงปลายทาง ในเลเยอร์ที่ 3 ข้อมูลที่รับส่งกันจะอยู่ในรูปแบบของกลุ่มข้อมูลที่เรียกว่าแพ็กเกต หรือเฟรมมข้อมูล เลเยอร์ที่ 4, 5, 6 และ 7 มองเห็นเป็นคำสั่งและ การสนทนาต่าง ๆ นั้น จะถูกแปลงและผนึกรวมอยู่ในรูปของแพ็กเกต หรือเฟรมที่มีเพียงแอดเดรสของผู้รับ, ผู้ส่ง, ลำดับการรับส่ง และส่วนของข้อมูลเท่านั้น หน้าที่อีกประการหนึ่งคือ การเรียกติดตั้ง (Call Setup) หรือเรียกติดต่อคอมพิวเตอร์ปลายทางก่อนการรับส่งข้อมูล และการ Call Cleaning หรือการยกเลิกการติดต่อคอมพิวเตอร์เมื่อการรับส่งข้อมูลจบลงแล้ว ในกรณีที่มีการรับส่งข้อมูลนั้นต้องมีการติดต่อกันก่อน

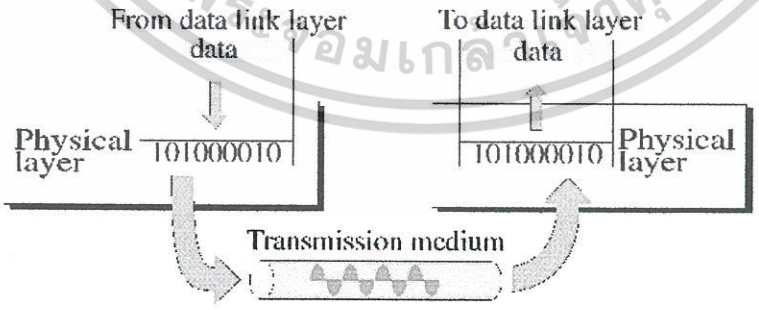
Layer 2: Data-link Layer



รูปที่ 2.5 แสดงหน้าที่การทำงานของ Data-link Layer

เป็นชั้นที่ทำหน้าที่เชื่อมต่อการรับส่งข้อมูลในระดับฮาร์ดแวร์ โดยเมื่อมีการสั่งให้รับข้อมูลจากในเลเยอร์ที่ 3 ลงมา เลเยอร์ที่ 2 จะทำหน้าที่แปลคำสั่งนั้นให้เป็นคำสั่งควบคุมฮาร์ดแวร์ที่ใช้รับส่งข้อมูล ทำการตรวจสอบข้อผิดพลาดในการรับส่งข้อมูลของระดับฮาร์ดแวร์ และทำการแก้ข้อผิดพลาดที่ได้ตรวจพบ ข้อมูลที่อยู่ในเลเยอร์ที่ 2 จะอยู่ในรูปของเฟรม เช่น ถ้าฮาร์ดแวร์ที่ใช้เป็น Ethernet LAN ข้อมูลจะมีรูปร่างของเฟรม ตามที่ระบุไว้ในมาตรฐานของอีเธอเน็ต หากว่าฮาร์ดแวร์ที่ใช้รับส่งข้อมูลเป็นชนิดอื่น รูปร่างของเฟรมก็จะเปลี่ยนไปตามมาตรฐานนั้น ๆ

Layer 1: Physical Layer



รูปที่ 2.6 แสดงหน้าที่การทำงานของ Physical Layer

เป็นชั้นล่างสุด และเป็นชั้นเดียวที่มีการเชื่อมต่อทางกายภาพระหว่างคอมพิวเตอร์สองระบบที่ทำการรับส่งข้อมูล ในเลเยอร์ที่ 1 นี้จะมีการกำหนดคุณสมบัติทางกายภาพของฮาร์ดแวร์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้งสองระบบ เช่น สายที่ใช้รับส่งข้อมูลจะเป็นแบบไหน ข้อต่อที่ใช้ในการรับส่งข้อมูลมีมาตรฐานอย่างไร ความเร็วในการรับส่งข้อมูลเท่าใด สัญญาณที่ใช้ในการรับส่งข้อมูลมีรูปร่างอย่างไร ข้อมูลในเลเยอร์ที่ 1 นี้จะมองเห็นเป็นการรับส่งข้อมูลทีละบิตเรียงต่อกันไป

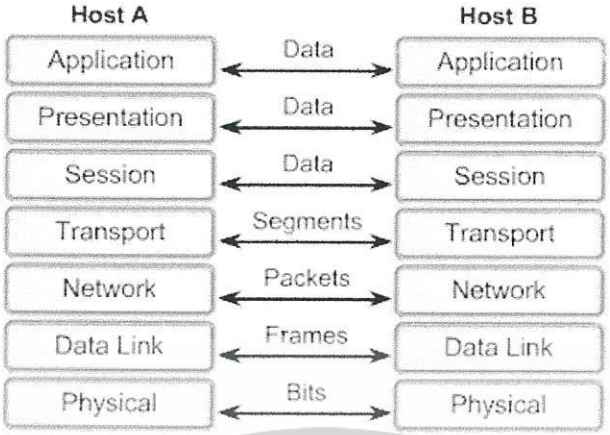
ตัวอย่างการส่งข้อมูลระหว่างคอมพิวเตอร์ 2 เครื่อง

เมื่อ คอมพิวเตอร์ A ต้องการส่งข้อมูลไปยัง คอมพิวเตอร์ B จะมีกระบวนการทำงานต่าง ๆ ตามลำดับดังนี้ ข้อมูลจากเลเยอร์ 7, 6, 5 จะถูกนำมาหั่นเป็นท่อน ๆ แล้วใส่ข้อมูลบางอย่างต่อเพิ่มเข้าไปในส่วนหัว เรียกว่า เฮดเดอร์ (Header) เพื่อใช้ในการบันทึกข้อมูลที่จำเป็นเช่น หมายเลขพอร์ตต้นทางและหมายเลขพอร์ตปลายทาง กลายมาเป็นก้อนข้อมูล (Segment) ในเลเยอร์ 4 ซึ่งเรียกว่า TCP Segment

จากนั้นข้อมูล เลเยอร์ 4 จะถูกส่งผ่านลงไปยังเลเยอร์ 3 และจะถูกใส่เฮดเดอร์อีกซึ่งเป็นการเพิ่มเฮดเดอร์เป็นชั้น ๆ เรียกว่า การห่อหุ้ม ซึ่งในส่วนนี้จะเหมือนกับการเอาเอกสารใส่ซองจดหมายแล้วจ่าหน้าซองระบุผู้ส่งและผู้รับ คือเป็นการบันทึกหมายเลขไอพี (IP address) ของโฮสต์ต้นทางและโฮสต์ปลายทางไว้ด้วย เมื่อทำการปกปิดเสร็จสิ้นจะได้ก้อนข้อมูลที่เรียกว่า แพ็คเก็ต (packet)

จากนั้นแพ็กเกตของข้อมูลจะถูกส่งผ่านไปยังระดับล่างอีก คือส่งไปให้เลเยอร์ 2 ในชั้นนี้ข้อมูลจะถูกใส่เฮดเดอร์ เพิ่มเข้าไปที่ส่วนหัวเพื่อเก็บหมายเลขอุปกรณ์ (MAC Address) ของต้นทางและปลายทาง และยังมีการใส่ข้อมูลต่อเพิ่มเข้าไปในส่วนหางด้วย ข้อมูลที่ต่อเพิ่มไปในส่วนหางนี้เรียกว่า Trailer จึงรวมกันกลายเป็นก้อนข้อมูลของ เลเยอร์ 2 ที่เรียกว่า Frame

จากนั้นเฟรมข้อมูลจะถูกแปลงให้กลายเป็นบิต (bit) ของข้อมูลเพื่อส่งไปตามสื่อ เช่น สาย UTP, Fiber ต่อไป การส่งสัญญาณทางไฟฟ้าไปตามสื่อต่าง ๆ นี้ เป็นการทำงานในระดับเลเยอร์ 1 เรียกว่า Physical Layer



รูปที่ 2.7 แสดงการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ในแต่ละเลเยอร์

2.2 การเข้ารหัสข้อมูลของเครือข่ายไร้สาย

มาตรฐานเครือข่ายไร้สาย

มาตรฐาน	ปีที่ประกาศ	ความถี่ที่ใช้	อัตราข้อมูลสูงสุด	ระยะทางโดยประมาณ
IEEE 802.11	1997	2.4 GHz	2 Mbps	N/A

ความรู้เบื้องต้นเกี่ยวกับมาตรฐานเครือข่ายไร้สาย

มาตรฐาน IEEE 802.11 ซึ่งได้รับการตีพิมพ์ครั้งแรกเมื่อปีพ.ศ. 2540 โดย IEEE (The Institute of Electronics and Electrical Engineers) และเป็นเทคโนโลยีสำหรับ WLAN ที่นิยมใช้กันอย่างแพร่หลายมากที่สุด คือข้อกำหนด (Specfication) สำหรับอุปกรณ์ WLAN ในส่วนของ Physical (PHY) Layer และ Media Access Control (MAC) Layer โดยในส่วนของ PHY Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้อุปกรณ์มีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1, 2, 5.5, 11 และ 54 Mbps โดยมีสื่อ 3 ประเภทให้เลือกใช้ได้แก่ คลื่นวิทยุที่ความถี่สาธารณะ 2.4 และ 5 GHz และอินฟราเรด (Infarred) (1 และ 2 Mbps เท่านั้น) สำหรับในส่วนของ MAC Layer มาตรฐาน IEEE 802.11 ได้กำหนดให้มีกลไกการทำงานที่เรียกว่า CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) ซึ่งมีความคล้ายคลึงกับหลักการ CSMA/CD (Collision Detection) ของมาตรฐาน IEEE 802.3 Ethernet ซึ่งเป็นที่นิยมใช้กันทั่วไปในเครือข่าย LAN แบบใช้สายนำสัญญาณ นอกจากนี้ในมาตรฐาน IEEE802.11 ยังกำหนดให้มีทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย IEEE 802.11 WLAN โดยกลไกการเข้ารหัสข้อมูล (Encryption) และการตรวจสอบผู้ใช้ (Authentication) ที่มีชื่อเรียกว่าดับเบิ้ลยูอีพี(Wired Equivalent Privacy) ด้วย

วิวัฒนาการของมาตรฐาน IEEE 802.11

มาตรฐาน IEEE 802.11 ได้รับการตีพิมพ์ครั้งแรกในปี พ.ศ. 2540 ซึ่งอุปกรณ์ตามมาตรฐานดังกล่าวจะมีความสามารถในการรับส่งข้อมูลด้วยความเร็ว 1 และ 2 Mbps ด้วยสื่อ อินฟราเรด (Infarred) หรือคลื่นวิทยุที่ความถี่ 2.4 GHz และมีกลไกดับเบิ้ลยูอีพีซึ่งเป็นทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย WLAN ได้ในระดับหนึ่ง เนื่องจากมาตรฐาน IEEE 802.11 เวอร์ชันแรกเริ่มมีประสิทธิภาพค่อนข้างต่ำและไม่มีการรองรับหลักการ Quality of Service (QoS) ซึ่งเป็นที่ต้องการของตลาด อีกทั้งกลไกรักษาความปลอดภัยที่ใช้ยังมีช่องโหว่อยู่มาก IEEE จึงได้จัดตั้งคณะทำงาน (Task Group) ขึ้นมาหลายชุดด้วยกันเพื่อทำการปรับปรุงเพิ่มเติมมาตรฐานให้มีศักยภาพสูงขึ้น โดยคณะทำงานกลุ่มที่มีผลงานที่น่าสนใจและเป็นที่รู้จักกันดีได้แก่ IEEE 802.11a, IEEE 802.11b, IEEE 802.11e, IEEE 802.11g, และ IEEE 802.11i

การรักษาความปลอดภัยตามมาตรฐาน IEEE 802.11 เริ่มมีผู้ค้นพบจุดอ่อนหรือช่องโหว่มากขึ้นเรื่อย ๆ ทั้งในระบบพิสูจน์ทราบตัวตน (Authentication), การรักษาความลับของข้อมูล (Data-privacy) และการรักษาความคงสภาพของข้อมูล (Integrity) ในระหว่างที่รับ-ส่งผ่านอากาศ

การป้องกันการเข้าถึงเครือข่ายโดยที่ไม่ได้รับอนุญาตนั้น จะใช้วิธีการพิสูจน์ทราบตัวตนในมาตรฐาน IEEE 802.11 มีข้อกำหนดเกี่ยวกับการพิสูจน์ทราบตัวตนของเครื่องฝั่งลูก อยู่ 2 วิธีคือ การพิสูจน์ทราบตัวตนแบบเปิด (Open System Authentication) และการพิสูจน์ทราบตัวตนแบบแชร์คีย์ (Shared Key Authentication)

เนื่องจากการส่งข้อมูลเครือข่ายไร้สาย เป็นการแพร่กระจายสัญญาณไปในอากาศ ทำให้ไม่ปลอดภัยหรืออาจมีปัญหาเกี่ยวกับการรักษาความลับของข้อมูล ตามมาตรฐานได้กำหนดให้การสื่อสารข้อมูลระหว่างเครื่องฝั่งลูกและแอ็กเซสพ็อยน์ (Access point) เข้ารหัสข้อมูลโดยใช้ดับเบิ้ลยูอีพี (Wired Equivalent Privacy) ถ้าเครื่องฝั่งลูกไม่มีดับเบิ้ลยูอีพีคีย์ก็จะไม่สามารถสื่อสารกับแอ็กเซสพ็อยน์ ได้ ตามมาตรฐาน IEEE 802.11 กำหนดให้ใช้คีย์ความยาว 40 และ104 บิต อย่างไรก็ตามมีการพิสูจน์แล้วว่าการใช้ดับเบิ้ลยูอีพีนั้นไม่ปลอดภัยอีกต่อไป ซึ่งจะกล่าวในหัวข้อถัดไป

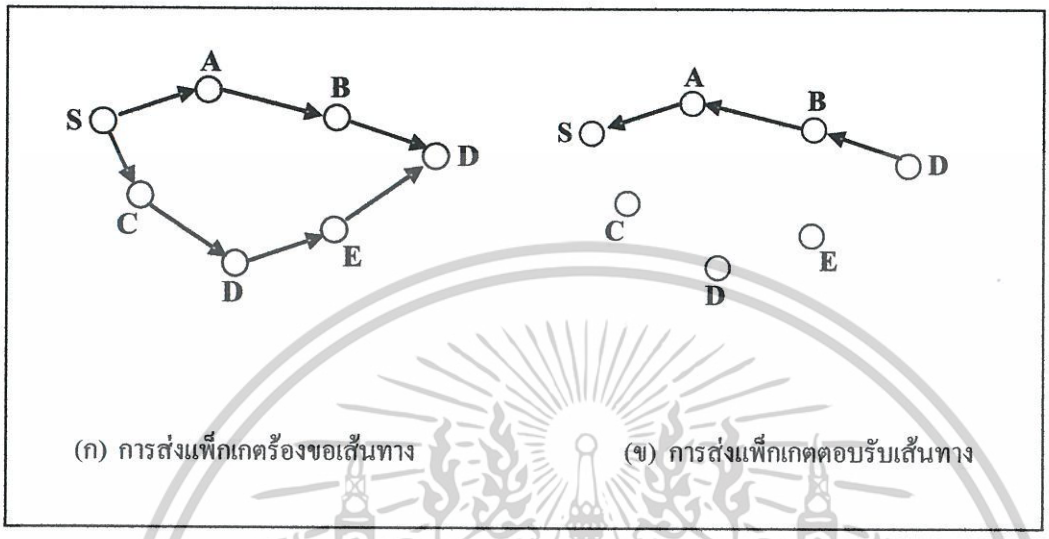
บทที่ 3
โพรโทคอลที่ใช้ในการศึกษา

3.1 โพรโทคอลในการค้นหาเส้นทาง

โพรโทคอลเอโอดีวี (AODV) เป็นโพรโทคอลการจัดเส้นทางในเน็ตเวิร์คไร้สายแบบเฉพาะกิจ ทำให้สถานีเชื่อมโยงสามารถติดต่อกันได้ โดยที่เส้นทางอาจมีหลายช่วงเชื่อมต่อ โพรโทคอลมีพื้นฐานมาจากโพรโทคอลเวกเตอร์บอกระยะ (Distance Vector) แต่เอโอดีวีจะมีการทำงานเป็นแบบรีแอคทีฟ คือขบวนการค้นหาเส้นทางเกิดขึ้นเมื่อมีการร้องขอใช้เส้นทางเท่านั้น และสถานีเชื่อมโยงไม่จำเป็นต้องทำการปรับปรุงข้อมูลเส้นทางไปยังสถานีเชื่อมโยงปลายทางที่ยังไม่ใช้งานในขณะนั้น และในขณะการสื่อสารดำเนินอยู่ โดยเส้นทางยังทำงานได้ เอโอดีวีก็จะไม่ทำงานใด ๆ เลย ข้อเด่นอย่างหนึ่งของโพรโทคอลเอโอดีวี คือการค้นหาเส้นและเลือกใช้เส้นทางของคู่สถานีเชื่อมโยงต้นทาง และปลายทางที่มีอยู่เพื่อให้การส่งข้อมูลนั้นเป็นไปอย่างถูกต้อง โพรโทคอลสถานะลิงค์และเวกเตอร์บอกระยะทำงานได้ในเน็ตเวิร์คไร้สายแบบเฉพาะกิจที่มีการเคลื่อนที่ของสถานีเชื่อมโยงน้อย ทำให้การเปลี่ยนแปลงของภูมิลักษณะของเน็ตเวิร์คไม่มากนัก แต่นอกจากปัญหาเกี่ยวกับการเปลี่ยนแปลงของภูมิลักษณะของเน็ตเวิร์คบ่อยแล้ว ในการทำงานของโพรโทคอลเหล่านี้ คือมีการส่งข้อความควบคุม (Control Messages) เป็นช่วง ๆ เพื่อใช้ในการกำหนดเส้นทางหรือปรับปรุงข้อมูลเส้นทาง

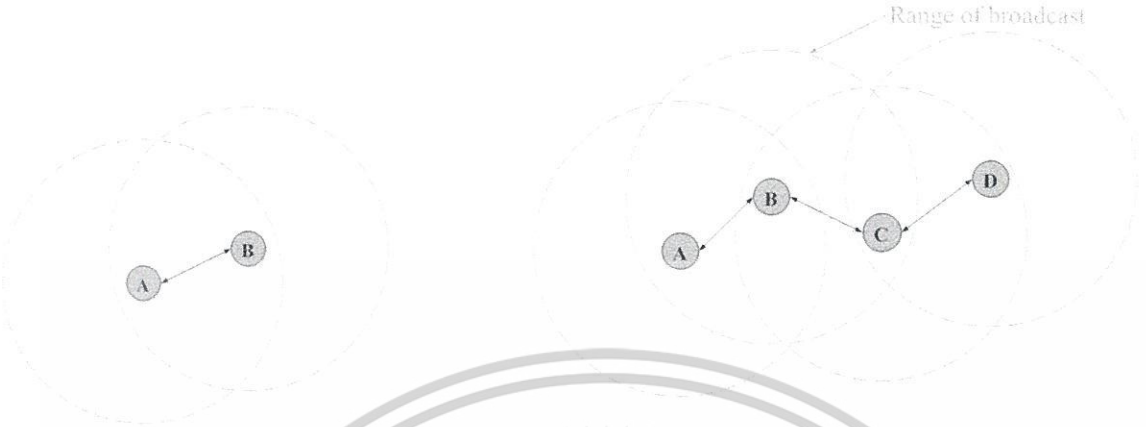
ในการค้นหาเส้นทางเมื่อโหนดต้นทางมีความต้องการในการส่งข้อมูล โหนดต้นทางก็จะกระจายแพ็กเกตร้องขอเส้นทางไปยังโหนดข้างเคียง เมื่อโหนดข้างเคียงได้รับก็จะสร้างเส้นทางย้อนกลับไปยังโหนดที่ส่งแพ็กเกตร้องขอเส้นทาง แล้วทำการส่งต่อแพ็กเกตร้องขอเส้นทางไปยังโหนดข้างเคียงอื่น ๆ โดยเพิ่มค่ามาตรวัดเส้นทางที่เป็นจำนวนฮ๊อบเข้าไปด้วย เมื่อปลายทางได้รับแพ็กเกตร้องขอเส้นทาง ก็จะทำการตรวจสอบ และทำการคัดเลือกแพ็กเกตร้องขอเส้นทางที่มีจำนวนฮ๊อบที่น้อยที่สุด แล้วสร้างแพ็กเกตตอบกลับเส้นทางส่งไปยังโหนดต้นทางตามเส้นทางที่แพ็กเกตร้องขอเส้นทางส่งมา แต่ถ้าโหนดระหว่างทางมีเส้นทางไปยังปลายทางอยู่แล้ว โหนดระหว่างทางก็จะสร้างข้อความตอบกลับเส้นทางและส่งไปยังโหนดที่ส่งแพ็กเกตร้องขอเส้นทางแทน ในส่วนของการบำรุงรักษาเส้นทาง เมื่อโหนดตรวจสอบว่ามีเส้นทางที่ไปยังโหนดใด ๆ เกิดการล้มเหลว ก็จะทำการสร้างแพ็กเกตเส้นทางผิดพลาดโดยเพิ่มเลขลำดับปลายทางขึ้นอีกหนึ่งค่า

แล้วส่งย้อนกลับไปยังโหนดต้นทาง เมื่อโหนดต้นทางได้รับแพ็กเกตเส้นทางผิดพลาดแล้วก็จะทำการค้นหาเส้นทางใหม่



รูปที่ 3.1 กระบวนการค้นหาเส้นทางของโพรโทคอลเอโอดีวี

เอโอดีวีเป็นโพรโทคอลที่เป็นแบบแผนของการค้นหาเส้นทางข้อมูล (Routing message) ระหว่างเครื่องคอมพิวเตอร์ที่เคลื่อนย้ายได้ (mobile computer) หรือโหนดในการส่งข้อมูลผ่านไปยังโหนดข้างเคียง (neighbor) เพื่อไปยังโหนดที่ต้นทาง ไม่สามารถติดต่อได้โดยตรง ในระหว่างทางที่ข้อมูลถูกส่งผ่านไป เอโอดีวีก็จะทำการค้นหาเส้นทางไป โดยจะมั่นใจได้ว่าจะไม่เกิดการวนลูป (loop) และพยายามหาเส้นทางที่สั้นที่สุดที่จะเป็นไปได้ อีกทั้งเอโอดีวียังสามารถควบคุมการเปลี่ยนแปลงของเส้นทาง (route) และสามารถสร้างเส้นทางใหม่หากเกิดข้อผิดพลาดได้อีกด้วย



(a) (b)

รูปที่ 3.2 แสดงขอบเขตของการสื่อสาร: (a) แสดงการสื่อสารแบบ one – hop;

(b) แสดงการสื่อสารแบบ multi - hop

จากรูปที่ 1 เป็นตัวอย่างการตั้งค่าให้กับโหนด A วงกลมที่เห็นจะแสดงขอบเขตการติดต่อสื่อสาร (Range of บรอดคาส) ของโหนด A และเพราะว่ามีการกำหนดขอบเขตการเชื่อมต่อทำให้แต่ละโหนดสามารถติดต่อกับโหนดที่อยู่ถัดจากตัวเองเท่านั้น ดังนั้นจากรูปที่ 1 (b) จะเห็นได้ว่าโหนด A สามารถติดต่อกับโหนด B ได้เท่านั้น เพราะโหนด C และ D อยู่นอกขอบเขตการติดต่อของ A จึงไม่สามารถติดต่อกันได้

สำหรับโหนดที่เราสามารถติดต่อได้โดยตรงนั้นเราจะเรียกว่าโหนดข้างเคียงโดยโหนดจะเก็บข้อมูลของโหนดข้างเคียงเมื่อได้รับ HELLO message ที่แต่ละโหนดจะทำการบรอดคาสออกมาตามช่วงเวลาที่กำหนดไว้

เมื่อมีโหนดใด ๆ ต้องการส่งข้อมูลไปยังโหนดอื่น ๆ ที่ไม่ใช่โหนข้างเคียงมันจะทำการบรอดคาสอาร์อาร์อีคิว (RREQ) Route Request message ซึ่งในอาร์อาร์อีคิวนี้จะประกอบไปด้วยคีย์บิตของข้อมูลหลายตัว เช่น ต้นทาง (source), ปลายทาง (destination), อายุ (lifespan) ของ message และหมายเลขลำดับ (sequence number) ที่เป็นไอดีเฉพาะ (Unique ID)

การทำงานของการจัดสรรเส้นทางแบบเอโอดีวี คือเมื่อมีโหนดต้นทางที่จะส่งข้อมูลไปยังโหนดปลายทาง โหนดต้นทางจะทำการส่งการร้องขอเส้นทาง ไปยังโหนดข้างเคียง และโหนดที่ได้รับก็จะทำการส่งต่อไปยังโหนดที่ใกล้เคียงต่อไปเรื่อย ๆ จนถึงโหนดที่ต้นทางต้องการจะติดต่อด้วย หรือโหนดปลายทางนั้นเอง และเมื่อโหนดปลายทางได้รับการร้องขอ ตัวแรกที่มาถึงที่โหนดปลายทาง โหนดปลายทางก็จะทำการส่งเส้นทางตอบกลับ (Route reply) กลับไปยังโหนดต้นทางที่ทำการส่งอาร์อาร์อีคิว มาให้โดยจะส่งกลับไปในเส้นทางที่อาร์อาร์อีคิวตัวแรกมาถึง เพราะถือว่าใช้เวลาน้อยที่สุดในการส่งอาร์อาร์อีคิว มาจากต้นทาง

3.2 โพรโทคอลในการรักษาความปลอดภัยให้กับเส้นทาง

3.2.1 โพรโทคอลเออาร์เอเอ็น (ARAN: Authenticated Routing for Ad hoc Networks) เป็นโพรโทคอลที่นำเสนอความมั่นคงในการพิสูจน์สำหรับการค้นหาเส้นทางแบบรีแอกทีฟโดยพิจารณาโพรโทคอลที่เป็นมาตรฐานของ IETF (Internet Engineering Task Force) ได้แก่ โพรโทคอลเอโอดีวี และโพรโทคอลดีเอสอาร์ ซึ่งทั้งสองโพรโทคอลนี้จะเป็นโพรโทคอลที่มีประสิทธิภาพทางด้านเครือข่ายดีกว่าโพรโทคอลอื่น ๆ

โพรโทคอลเออาร์เอเอ็น จะใช้ใบรับรองอิเล็กทรอนิกส์ที่มีการเข้ารหัส (Cryptographic Certificates) ที่ได้จากผู้ให้บริการออกใบรับรองที่น่าเชื่อถือ (Trusted Certificate Server) ในการรักษาความมั่นคงของการค้นหาเส้นทาง โดยใบรับรองอิเล็กทรอนิกส์จะประกอบด้วย ไอพีแอดเดรส กุญแจสาธารณะของโหนดนั้น รวมทั้งเวลาในการเริ่มใช้ใบรับรองและเวลาที่ใบรับรองหมดอายุ และใบรับรองที่ได้จะถูกลงลายมือชื่อโดยใช้กุญแจส่วนตัวของผู้ให้บริการออกใบรับรอง โหนดเมื่อได้รับใบรับรองก็จะใช้ใบรับรองนั้นส่งออกไปพร้อมข้อความร้องขอเส้นทาง

$$T \rightarrow A : \text{cert}_A = [IP_A, K_{A+}, t, e]K_{T-} \tag{3.1}$$

ผู้ให้บริการออกใบรับรองส่งใบรับรองอิเล็กทรอนิกส์ให้โหนดในเครือข่าย

เมื่อโหนดต้องการค้นหาเส้นทางไปยังปลายทาง ก็จะทำการส่งแพ็กเกตค้นหาเส้นทาง (RDP: Route Discovery Packet) ไอพีแอดเดรสของโหนดปลายทาง และเวลาปัจจุบันของโหนดต้นทาง แล้วทำการลงลายมือชื่อดิจิทัลโดยใช้กุญแจส่วนตัว กระจายออกไปพร้อมกับใบรับรองอิเล็กทรอนิกส์

$$A \rightarrow \text{broadcast} : [\text{RDP}, \text{IP}_X, N_A]K_{A-}, \text{cert}_A \tag{3.2}$$

แพ็กเกตร้องขอเส้นทางที่กระจายจากโหนดต้นทาง

โหนดที่รับแพ็กเกตจะสามารถตรวจสอบได้ โดยใช้กุญแจสาธารณะของผู้ให้บริการออกใบรับรองทำการตรวจสอบใบรับรองก่อน หลังจากที่ตรวจสอบใบรับรองแล้วก็จะได้กุญแจสาธารณะของโหนดที่ส่งแพ็กเกตมา จากนั้นจะทำการตรวจสอบแพ็กเกตค้นหาเส้นทางว่าต้องการส่งไปยังโหนดใด ถ้ายังไม่ใช่โหนดปลายทางก็จะทำการกระจายแพ็กเกตไปยังโหนดอื่น ๆ ต่อไป โดยเอาแพ็กเกตเดิมที่ได้มาจากโหนดก่อนหน้ามาลงลายมือชื่อดิจิทัล ด้วยกุญแจส่วนตัวของโหนดนั้น พร้อมทั้งส่งใบรับรองอิเล็กทรอนิกส์ของโหนดนั้นออกไปด้วย โหนดอื่น ๆ ก็จะทำลักษณะเดียวกัน แต่จะเอาใบรับรองของโหนดก่อนหน้าออกแล้วทำการเพิ่มใบรับรองของโหนดตัวเองเข้าไปแทน

$$B \rightarrow \text{broadcast} : [[\text{RDP}, \text{IP}_X, N_A]K_{A-}]K_{B-}, \text{cert}_A, \text{cert}_B$$

$$C \rightarrow \text{broadcast} : [[\text{RDP}, \text{IP}_X, N_A]K_{A-}]K_{C-}, \text{cert}_A, \text{cert}_C \tag{3.3}$$

แพ็กเกตร้องขอเส้นทางที่กระจายจากโหนดระหว่างทาง

เมื่อแพ็กเกตร้องขอเส้นทางไปถึงยังโหนดปลายทางแล้ว โหนดปลายทางก็จะทำการส่งแพ็กเกตตอบกลับ (Reply Packet) พร้อมกับไอพีแอดเดรสของโหนดต้นทาง ค่าของเวลาที่ส่งโดยต้นทาง ทำการลงลายมือชื่อดิจิทัลโดยใช้กุญแจส่วนตัวของโหนดปลายทาง พร้อมด้วยใบรับรองของโหนดปลายทาง โดยทำการส่งไปยังโหนดก่อนหน้าที่ส่งแพ็กเกตร้องขอเส้นทางมา

$$X \rightarrow D : [\mathrm{REP}, \mathrm{IP}_a, N_A]K_{X-}, \mathrm{cert}_x \tag{3.4}$$

แพ็กเกตตอบรับที่ส่งจากโหนดปลายทาง

โหนดที่ได้รับแพ็กเกตตอบกลับ ก็จะทำการตรวจสอบว่าเป็นโหนดต้นทางหรือไม่ ถ้าไม่ก็จะทำการส่งต่อไปยังโหนดที่ร้องขอเส้นทางก่อนหน้า โดยจะนำเอาแพ็กเกตตอบกลับจากโหนดปลายทางมาทำการลงลายมือชื่อดิจิทัลด้วยกุญแจส่วนตัวของตัวเอง พร้อมทั้งแนบใบรับรองอิเล็กทรอนิกส์ของโหนดตัวเองไปด้วย โหนดอื่น ๆ ก็จะทำลักษณะเดียวกัน แต่จะเอาใบรับรองอิเล็กทรอนิกส์ของโหนดก่อนหน้าออกแล้วทำการเพิ่มใบรับรองอิเล็กทรอนิกส์ของโหนดตัวเองเข้าไปแทน จนกระทั่งถึงโหนดต้นทาง

$$\begin{aligned} D \rightarrow C &: [[\mathrm{REP}, \mathrm{IP}_a, N_A]K_{X-}]K_{D-}, \mathrm{cert}_x, \mathrm{cert}_D \\ C \rightarrow B &: [[\mathrm{REP}, \mathrm{IP}_a, N_A]K_{X-}]K_{C-}, \mathrm{cert}_x, \mathrm{cert}_C \end{aligned} \tag{3.5}$$

แพ็กเกตตอบรับที่ส่งจากโหนดระหว่างทาง

ในการบำรุงรักษาเส้นทาง ถ้าไม่มีการส่งข้อมูลในเส้นทางจนกระทั่งหมดเวลาของเส้นทาง ก็จะทำการกระตุ้นเส้นทางอีกครั้ง ถ้าโหนดใดไม่สามารถส่งข้อมูลต่อไปยังโหนดอื่น ๆ ได้ โหนดนั้นก็จะทำการสร้างแพ็กเกตเส้นทางผิดพลาด (ERR: Route Error) โดยส่งไปพร้อมกับไอพีแอดเดรสของโหนดต้นทาง ไอพีแอดเดรสของโหนดปลายทาง เวลาในช่วงเวลานั้น ทำการลงลายมือชื่อดิจิทัลด้วยกุญแจส่วนตัวของโหนดนั้น และส่งออกไปกลับไปตามเส้นทางที่ไปยังต้นทาง พร้อมด้วยใบรับรองอิเล็กทรอนิกส์ของโหนด

$$C \rightarrow B : [ERR, \mathrm{IP}_A, \mathrm{IP}_X, N_c]K_{C-}, \mathrm{cert}_c \tag{3.6}$$

แพ็กเกตเส้นทางผิดพลาดที่ส่งจากโหนดระหว่างทาง

ในการยกเลิกใบรับรองอิเล็กทรอนิกส์ ผู้ให้บริการออกใบรับรองจะทำการกระจายแพ็กเกตที่ใช้ยกเลิกใบรับรอง (Revoke) ใบรับรองของโหนดที่ต้องการยกเลิก โดยทำการลงลายมือชื่อดิจิทัลด้วยกุญแจส่วนตัวของผู้ให้บริการออกใบรับรอง

$$T \rightarrow \text{broadcast} : [revoke, \text{cert}_r]K_{T-} \tag{3.7}$$

แพ็กเกตยกเลิกใบรับรองจากผู้ออกใบรับรอง

3.3 จุดอ่อนของการค้นหาเส้นทางบนเครือข่ายไร้สายเฉพาะกิจ

Denial-of-ServiceAttacks

การโจมตีที่มีจุดประสงค์เพื่อทำให้ไม่สามารถบริการได้หรือที่เรียกว่า Denial of Service หรือ DoS ถูกจัดให้มีคุณสมบัติเป็นความพยายามของผู้โจมตีหรือผู้บุกรุก ที่ต้องการทำให้เกิดภาวะของการไม่สามารถเข้าใช้บริการหรือทรัพยากรในระบบได้ ตัวอย่างเช่น

- กระทำการส่งแพ็กเกตจำนวนมากเข้าไปในเครือข่ายหรือ "Flooding" ทำให้ปริมาณทราฟฟิกในเครือข่ายเพิ่มสูงขึ้นในเวลาอันรวดเร็ว ทำให้การสื่อสารในเครือข่ายตามปกติช้าลงหรือใช้ไม่ได้
- กระทำการขัดขวางการเชื่อมต่อใด ๆ ในเครือข่ายทำให้เครื่องคอมพิวเตอร์หรืออุปกรณ์เครือข่ายไม่สามารถสื่อสารกันได้ ตัวอย่างเช่นการถอดสายเชื่อมต่อเครือข่ายของเครื่องเซิร์ฟเวอร์ออกจากอุปกรณ์สวิตซ์
- กระทำการใดก็ตามเพื่อขัดขวางมิให้ผู้ใช้ในระบบไม่สามารถเข้าใช้บริการในระบบ เช่น การปิดบริการเว็บเซิร์ฟเวอร์ลง
- กระทำการในการทำลายระบบหรือบริการในระบบ เช่นการลบชื่อและข้อมูลผู้ใช้ออกจากระบบ ทำให้ไม่สามารถเข้าสู่ระบบได้

การกระทำที่เกิดจากความประมาทของผู้ดูแลระบบก็อาจจะนำพาไปสู่การทำให้เกิด DoS ได้เช่นกัน หรืออาจจะเกิดจากสาเหตุทางธรรมชาติ ทำให้เครื่องเซิร์ฟเวอร์หรือระบบเครือข่ายไม่สามารถใช้งานได้ก็คือความหมายของ DoS ได้ทั้งสิ้น แต่ในทางปฏิบัติการโจมตีแบบ DoS มักจะเจาะจงไปที่ผู้บุกรุกหรือผู้ที่ไม่มีสิทธิในระบบมากกว่า เหตุการณ์ธรรมชาติหรือความผิดพลาดของผู้ดูแลระบบ

การโจมตีแบบ DoS มักเกิดสืบเนื่องมาจากการโจมตีประเภทอื่นร่วมด้วยเช่น การแพร่กระจายของไวรัสปริมาณมากในเครือข่ายทำให้เกิดปริมาณทราฟฟิกจำนวนมาก หรือการโจมตีข้อบกพร่องของซอฟต์แวร์ระบบเพื่อจุดประสงค์ในการเข้าถึงสิทธิที่สูงขึ้น การโจมตีในลักษณะดังกล่าวถ้าไม่สำเร็จมักจะทำให้ซอฟต์แวร์ระบบนั้นปิดตัวเองลงอัตโนมัติหรือไม่สามารถทำงานได้ต่อไป ซึ่งจัดอยู่ในข่ายว่าไม่สามารถให้บริการได้หรือเกิด DoS ได้เช่นเดียวกัน เป็นต้น

การเข้าใช้ทรัพยากรในระบบตามปกติก็อาจจะทำให้เกิด DoS ได้ เช่นการอนุญาตให้อัพโหลดไฟล์ผ่านทางบริการโอนย้ายไฟล์หรือ FTP ผู้บุกรุกสามารถจะใช้พื้นที่ที่อนุญาตนี้ทำการนำไฟล์สำคัญหรือข้อมูลทางการค้าจำนวนมากทำให้พื้นที่น้อยลงไปหรือหมดไปได้ รวมถึงอาจจะเป็นสาเหตุให้ทราฟฟิกเพิ่มขึ้นจากการโอนย้ายข้อมูลที่เกิดจากการกระทำของผู้บุกรุกด้วย

รูปแบบการโจมตี

1. การเข้าใช้หรือยึดครองทรัพยากรที่มีอย่างจำกัดหรือไม่สามารถสร้างขึ้นใหม่ได้โดยง่าย
   - การเชื่อมต่อเข้าระบบเครือข่าย การไม่ให้เครื่องคอมพิวเตอร์เป้าหมายหรืออุปกรณ์ เครือข่ายเป้าหมายไม่สามารถสื่อสารกับเครือข่ายได้
   - ใช้ทรัพยากรของเป้าหมายในการสร้างความเสียหายให้กับเป้าหมายเองได้
   - การใช้ทรัพยากรแบนด์วิธ
   - ผู้บุกรุกสามารถทำให้ทรัพยากรอื่นของระบบที่กำลังให้บริการลดลงได้
2. การเข้าทำลายหรือเปลี่ยนแปลงข้อมูลการทำงานของระบบ
3. การเข้าทำลายหรือเปลี่ยนแปลงอุปกรณ์เครือข่ายหรือชิ้นส่วนหนึ่งส่วนใดทางกายภาพ

บทที่ 4
เทคโนโลยีที่เกี่ยวข้องกับโพรโทคอลที่ใช้ในการศึกษา

4.1 ผู้ให้บริการรับรองหนังสือ Certificated Authority (CA)

โลกของอิเล็กทรอนิกส์ เราสามารถทำการตรวจสอบสถานะของบุคคลที่ประสงค์จะติดต่อได้จากใบรับรองอิเล็กทรอนิกส์ (Certificate) ซึ่งใช้สำหรับการยืนยันตัวบุคคล ใบรับรองฯ ดังกล่าวเป็นข้อมูลในรูปแบบอิเล็กทรอนิกส์ที่ออกให้โดยผู้ให้บริการออกใบรับรอง (Certification Authority - CA) ซึ่งต้องเป็นหน่วยงานหรือองค์กรที่เชื่อถือได้และมีความรู้ความเชี่ยวชาญในเทคโนโลยีความปลอดภัย โดยอาศัยเทคโนโลยีหลัก เรียกว่า เทคโนโลยีโครงสร้างพื้นฐานระบบกุญแจสาธารณะ (Public Key Infrastructure - PKI) ซึ่งมีระบบกุญแจคู่ อันประกอบด้วยกุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) เป็นพื้นฐานสำคัญ โดยที่ผู้ให้บริการออกใบรับรองจะทำการรับรองข้อมูลต่าง ๆ รวมทั้งกุญแจสาธารณะของบุคคลนั้น ซึ่งกุญแจสาธารณะจะใช้ในการเข้ารหัสข้อมูล (Encryption) และตรวจสอบลายมือชื่อดิจิทัล (Digital Signature) เพื่อเป็นการรักษาความลับ ตรวจสอบความถูกต้อง และที่มาของข้อมูลว่าบุคคลใดเป็นผู้ทำการส่งข้อมูลหรือลงลายมือชื่อดิจิทัล โดยที่ผู้ใช้งานควรที่จะมีความรู้ความเข้าใจเกี่ยวกับกระบวนการในการใช้งานใบรับรองอิเล็กทรอนิกส์ และข้อมูลที่ปรากฏอยู่ในใบรับรองอิเล็กทรอนิกส์ เพื่อก่อให้เกิดประสิทธิภาพในการใช้งาน รวมทั้งยังเป็นประโยชน์แก่ผู้ใช้งานอีกด้วย

ใบรับรองอิเล็กทรอนิกส์นั้นสามารถนำไปประยุกต์ใช้ได้ใน 2 ลักษณะดังนี้

1. การลงรายมือชื่อดิจิทัล (Digital Signature)

เป็นการรับรองว่าข้อมูลอิเล็กทรอนิกส์ที่ส่งมานั้นเป็นข้อมูลที่ส่งโดยผู้ส่งที่อ้างไว้จริง และใช้ลายมือชื่อดิจิทัลนี้ในการตรวจสอบข้อมูลว่ามีการปลอมแปลงในระหว่างขั้นตอนการส่งหรือไม่ เช่น การลงลายมือชื่อดิจิทัลกำกับจดหมายอิเล็กทรอนิกส์ ซึ่งผู้ส่งจะใช้กุญแจส่วนตัว (Private key) ของตนทำการลงลายมือชื่อ ดิจิทัลกำกับจดหมายอิเล็กทรอนิกส์ฉบับนั้น ซึ่งทำให้มั่นใจได้ว่าจดหมายอิเล็กทรอนิกส์เป็นของผู้ส่งที่อ้างไว้จริง โดยในการตรวจสอบนั้นผู้รับจะต้องใช้กุญแจสาธารณะ (Public key) ที่อยู่ในใบรับรองของผู้ส่งมาทำการตรวจสอบจดหมายอิเล็กทรอนิกส์ที่ส่งมาว่ามาจากผู้ส่งจริง และไม่มีการปลอมแปลงข้อมูลระหว่างขั้นตอนการส่ง

2. การเข้ารหัส (Encryption)

เป็นการแปรรูปข้อมูลธรรมดาให้อยู่ในรูปของข้อมูลที่ไม่สามารถอ่านเข้าใจได้ เพื่อป้องกันมิให้ผู้อื่นล่วงรู้ข้อมูล เช่น การเข้ารหัสจดหมายอิเล็กทรอนิกส์ โดยผู้ส่งจะใช้กุญแจสาธารณะ (Public key) ของผู้รับ (ซึ่งอยู่ในใบรับรองของผู้รับ) มาทำการเข้ารหัส ส่วนในการถอดรหัสผู้รับจะต้องใช้กุญแจส่วนตัว (Private Key) ของตนเองมาทำการถอดรหัส ในการใช้กุญแจส่วนตัว (Private Key) มาถอดรหัสนี้เป็นการมั่นใจได้ว่าผู้รับที่เป็นเจ้าของคู่กุญแจ (กุญแจส่วนตัวและกุญแจสาธารณะ) เท่านั้นที่สามารถอ่านข้อมูลได้

ใบรับรองอิเล็กทรอนิกส์มีประโยชน์อย่างไร

1. การรักษาความลับของข้อมูล (Data Confidentiality) ผู้ที่ไม่ได้รับอนุญาตจะไม่สามารถเปิดอ่านข้อมูลได้
2. การรักษาความลับของข้อมูล (Data Integrity) ข้อมูลจะถูกต้องไม่มีการเปลี่ยนแปลงก่อนถึงผู้รับ
3. การพิสูจน์สิทธิ์ (Authentication) การพิสูจน์ตัวตนว่าเป็นตัวจริง
4. การห้ามปฏิเสธความรับผิด (Non-repudiation) การป้องกันไม่ให้บุคคลผู้ส่งปฏิเสธว่าตนไม่ได้ส่ง

ใบรับรอง X.509

ใบรับรองดิจิทัลนี้ โดยภายในใบรับรองดิจิทัลจะบรรจุข้อมูลต่าง ๆ เช่น ชื่อของเจ้าของใบรับรอง วันที่ออก วันหมดอายุ ชื่อของหน่วยงานที่ออกกุญแจสาธารณะ (Public key) และข้อมูลอื่น ๆ และเพื่อให้การใช้งานใบรับรองไม่มีปัญหาในเรื่องของรูปแบบ หน่วยงาน ITU/U และ ISO จึงได้ร่วมกันออกมาตรฐานของรูปแบบใบรับรอง

X.509 Authentication Service

ระบบ X.509 เป็นระบบการพิสูจน์สิทธิ์ที่สำคัญมากในระบบเครือข่ายโดย X.509 เป็นอนุกรมย่อยของ X.500 ซึ่งกำหนดมาตรฐานโดย ITU-T โดยในขณะที่ X.500 เป็นตัวกำหนดโครงสร้างในลักษณะที่เป็นหมวดหมู่ (Directory) หรือ Hierarchy Tree นั้น X.509 จะทำหน้าที่ในการพิสูจน์สิทธิ์ให้กับส่วนต่าง ๆ ของหมวดหมู่นั้น สำหรับรูปแบบการใช้งานเมื่อเทียบกับ Kerberos แล้วการใช้งาน Kerberos จะเน้นไปที่การพิสูจน์สิทธิ์เพื่อเข้าใช้บริการ ซึ่งมักจะเป็นการ

พิสูจน์สิทธิ์ภายในองค์กรเดียวกัน แต่ X.509 จะเน้นไปที่การพิสูจน์ตัวบุคคล เพื่อยืนยันการติดต่อมากกว่า

การทำงานของ X.509 จะมีโครงสร้างการทำงานที่เป็นหมวดหมู่ โดยในที่นี้ หมวดหมู่จะทำหน้าที่เป็นที่เก็บข้อมูลที่ใช้ในการยืนยัน ซึ่งโดยทั่วไปจะอยู่ในรูปของใบรับรองอิเล็กทรอนิคส์ ซึ่งในใบรับรองอิเล็กทรอนิคส์จะบรรจุกุญแจสาธารณะของผู้ใช้ที่ทำสัญลักษณ์โดยกุญแจส่วนตัวขององค์กรที่จ่ายใบรับรองอิเล็กทรอนิคส์มาให้ สำหรับการทำงานของ X.509 นั้น จะมีขอบเขตการนำไปใช้งานที่กว้างขวางมาก เช่น ใช้ในการทำ Mail Security ใช้ในการทำ IP Security ใช้ในการทำ Web Security หรือหากจะกล่าวว่า เมื่อใดที่ต้องการการพิสูจน์หรือยืนยันบุคคล หรือยืนยันเครื่องคอมพิวเตอร์แล้ว ก็มักจะอยู่ในขอบข่ายการทำงานของ X.509 เสมอ

X.509 ได้ถูกนำเสนอเมื่อปี 1988 จากนั้นได้ผ่านการปรับปรุงเป็นลำดับขั้น ในประเด็นต่าง ๆ รวมทั้งเรื่องของความปลอดภัยด้วย จากนั้นก็ได้ออกมาเป็นข้อเสนอที่ปรับปรุงแล้วในปี 1993 และปรับปรุงอีกครั้งในปี 1995 โดยการทำงานของ X.509 จะใช้การเข้ารหัสแบบกุญแจสาธารณะ และใช้มาตรฐานลงชื่อดิจิตอล (Digital Signature) ในการทำสัญลักษณ์สำหรับอัลกอริทึมนั้น ไม่ได้ระบุแน่นอน โดยสามารถเลือกใช้ได้หลายตัว แต่ที่แนะนำ คือ RSA สำหรับลงชื่อดิจิตอลก็เช่นกัน ที่ไม่ได้กำหนดมาตรฐานของอัลกอริทึม Hash เอาไว้

Certificate

เนื่องจาก X.509 นั้นจะเปรียบเสมือนกับโครงสร้างที่ทำหน้าที่เก็บใบรับรองอิเล็กทรอนิคส์ ซึ่งทำหน้าที่เป็นใบรับรองกุญแจสาธารณะของแต่ละบุคคล หรือแต่ละเครื่องว่าเป็นกุญแจสาธารณะที่ทำหน้าที่เป็นตัวแทนของบุคคลนั้น หรือเครื่องนั้นจริง โดยใบรับรองอิเล็กทรอนิคส์จะสร้างขึ้นโดยผู้ให้บริการหนังสือรับรอง (Certificate Authority) หรือซีเอ (CA) ที่เชื่อถือ (Trust) ได้ จากนั้นก็จะนำมาเก็บในซีเอ ซึ่งอาจเป็นซีเอ ที่สร้างใบรับรองอิเล็กทรอนิคส์หรือไม่ก็ได้ ดังนั้นจุดเริ่มแรกที่เราจะต้องศึกษา คือ โครงสร้างของใบรับรองอิเล็กทรอนิคส์โดยแสดงรูปแบบทั่วไปของใบรับรองอิเล็กทรอนิคส์โดยมีรายละเอียดดังนี้

- Version แสดงหมายเลขเวอร์ชัน เพราะในแต่ละเวอร์ชันจะมีรูปแบบของข้อมูลที่ไม่เหมือนกันก็ได้ โดยปกติจะเป็นเวอร์ชัน 1 แต่หากในใบรับรองอิเล็กทรอนิคส์มีการใช้ Initiator Unique Identifier หรือ Subject Unique Identifier แล้วค่าเวอร์ชันจะต้องเป็น 2 และหากมีการใช้ต่อขยายใด ๆ ค่าของเวอร์ชันจะต้องเป็น 3

- Serial Number เป็นเลขจำนวนเต็ม โดยจะต้องไม่ซ้ำกันใน CA ที่จ่ายใบรับรองอิเล็กทรอนิคส์มา โดยเลขนี้จะเป็นเลขที่จะใช้อ้างถึงแต่ละใบรับรองอิเล็กทรอนิคส์ในแต่ละ CA ที่ได้สร้างขึ้นมา

- Signature Algorithm Identifier เป็นฟิลด์ที่ระบุอัลกอริทึมที่ใช้ในการลงชื่อใบรับรองอิเล็กทรอนิคส์พร้อมด้วยพารามิเตอร์ที่ใช้ แต่เนื่องจากค่านี้จะระบุอีกครั้งในฟิลด์ Signature ฟิลด์นี้จึงไม่มีการใช้งานมากนัก

- Issue Name เป็นชื่อของ CA ที่สร้างและ Sign Certificate ใบนี้

- Period of Validity เป็นตัวบอกว่าให้ใช้ใบรับรองอิเล็กทรอนิคส์นี้ตั้งแต่เมื่อไร ถึงเมื่อไร

- Subject Name เป็นชื่อของบุคคลที่ใบรับรองอิเล็กทรอนิคส์ใบนี้อ้างถึง หรือแทนบุคคลนั้น ซึ่งหมายถึงในใบรับรองอิเล็กทรอนิคส์นี้จะเก็บกุญแจสาธารณะที่มีบุคคลในฟิลด์นี้เป็นผู้เก็บ กุญแจส่วนตัวที่คู่กันอยู่

- Subject's Public Key Information เป็นฟิลด์ที่เก็บกุญแจสาธารณะและระบุถึงอัลกอริทึมที่ใช้ที่ใช้กับกุญแจนี้ และพารามิเตอร์อื่น ๆ

- Issuer Unique Identifier เป็นฟิลด์ Option ที่ใช้ในการระบุถึง CA ในกรณีที่ X.500 Name มีการนำไปใช้กับส่วนอื่น ๆ

- Subject Unique Identifier เป็นฟิลด์ Option ที่ใช้ในการระบุถึงบุคคล ในกรณีที่ X.500 Name มีการนำไปใช้กับส่วนอื่น

- Extension เป็นกลุ่มของฟิลด์ที่เพิ่มเติมข้อมูลอื่น ๆ เข้ามาด้วย

- Signature จะบรรจุ MD ของข้อมูลในทุกฟิลด์ ที่เข้ารหัสด้วย Private Key ของ CA เพื่อเป็นการยืนยันว่าใบรับรองอิเล็กทรอนิคส์นี้สร้างมาจาก CA จริง ๆ โดยจะมีข้อมูลที่ระบุวิธีการ Hash และวิธีการเข้ารหัสด้วย

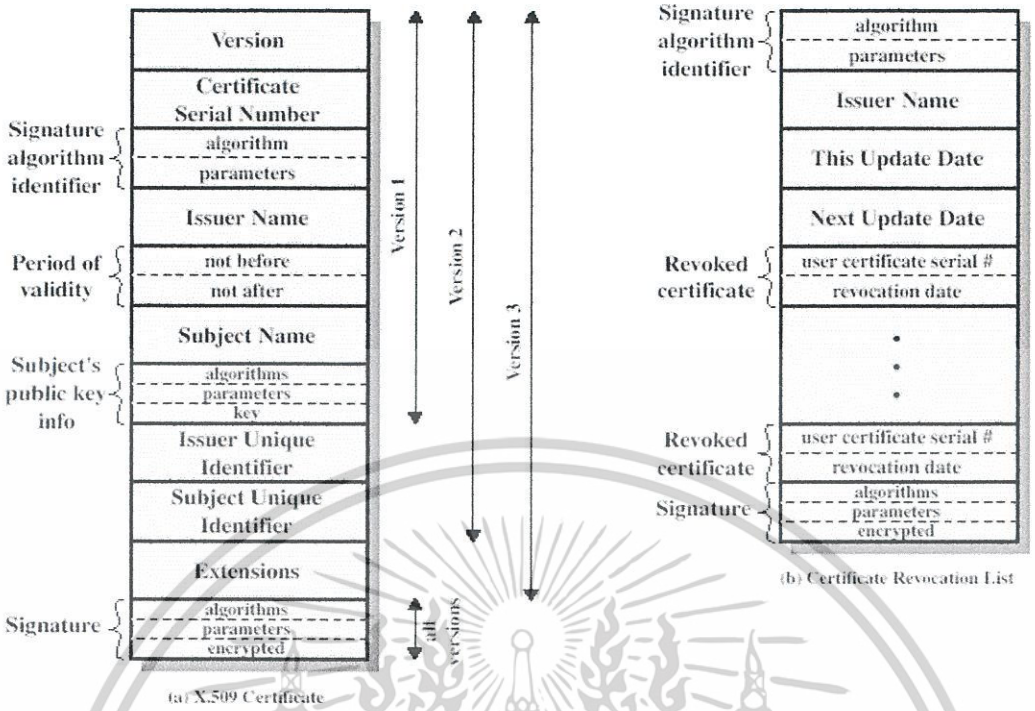


Figure 4.3 X.509 Formats

รูปที่ 4.1 แสดงรูปแบบของ X.509

4.2 กุญแจอาร์เอสเอ (RSA Key)

อาร์เอสเอเป็นการเข้ารหัส บนอินเตอร์เน็ตและระบบการรับรองที่ใช้อัลกอริทึมที่พัฒนาในปี 1977 โดยรอน ไรเวส (Ron Rivest) เอดิ ชาไมร์ (Adi Shamir) และลูนาร์ด อเดลแมน (Loonard Adleman) อัลกอริทึมอาร์เอสเอมีการใช้โดยทั่วไปในการเข้ารหัส และการรับรอง ซึ่งได้ร่วมเป็นส่วนหนึ่งของเว็บเบราเซอร์ จากเน็ตสแคปส์ (Netscaps) ไมโครซอฟต์ (Microsoft) รวมถึง โลตัสโน๊ตส์ (Lotus Notes) อินทูอิท ควิกเคน (Intuit Quicken) และผลิตภัณฑ์อื่น ๆ ระบบการเข้ารหัสเป็นของอาร์เอสเอ ซีเคียวริตี้ บริษัทต้องขออนุญาตการใช้เทคโนโลยีอัลกอริทึมและการขายชุดพัฒนาโปรแกรม เทคโนโลยีเป็นส่วนของมาตรฐานเว็บ อินเตอร์เน็ต และการคำนวณ

ระบบอาร์เอสเอ ทำงานอย่างไร

รายละเอียดทางคณิตศาสตร์ของอัลกอริทึมใช้ในการเก็บกุญแจส่วนตัว และกุญแจส่วนตัว อัลกอริทึมนี้ใช้ผลคูณของไพรม์นัมเบอร์ (prime number) ขนาดใหญ่ (ไพรม์นัมเบอร์หารลงตัวได้โดยตัวเลขและ1) และผ่านกระบวนการเพิ่มเติมที่มาจากกลุ่มของ 2 จำนวนที่เก็บกุญแจสาธารณะและอีกชุดเก็บ กุญแจส่วนตัว เมื่อมีการพัฒนากุญแจจำนวน ไพรม์นัมเบอร์ดังเดิม จะไม่มีความสำคัญและถูกลบทิ้ง ทั้งกุญแจสาธารณะและส่วนตัว ต้องการสำหรับการเข้ารหัส/การถอดรหัส แต่เฉพาะเจ้าของกุญแจส่วนตัวที่ต้องการทราบ การใช้ระบบอาร์เอสเอ, กุญแจส่วนตัวไม่ต้องมีการสั่งข้ามอินเตอร์เน็ตกุญแจส่วนตัวใช้ถอดรหัสข้อความที่ได้รับการเข้ารหัสด้วยกุญแจ

สาธารณะถ้ามีการส่งข้อความผู้ส่งสามารถค้นหากุญแจสาธารณะของผู้รับจากผู้บริหารกลางและเข้ารหัสข้อความไปให้ผู้รับด้วย กุญแจสาธารณะของผู้รับ ซึ่งผู้รับสามารถรับรองตัวเองกับผู้ส่ง โดยการใช้กุญแจส่วนตัวในการเข้ารหัสการรับรองดิจิตอล เมื่อผู้ส่งได้รับแล้ว ผู้ส่งสามารถใช้กุญแจสาธารณะของผู้รับเพื่อถอดรหัส

ตารางที่ 4.1 แสดงรายละเอียดการทำงานของระบบอาร์เอาเอ

To do this	Use whose	Kind of key
Send an encrypted message	Use the receiver's	Public key
Send an encrypted signature	Use the sender's	Private key
Decrypt an encrypted message	Use the receiver's	Private key
Decrypt an encrypted signature (and authenticate the sender)	Use the sender's	Public key

4.3 เครื่องมือในการวัดประสิทธิภาพบนเครือข่ายของระบบลินุกซ์

ไอเพิร์ฟ (Iperf) เป็นเครื่องมือในการทดสอบการใช้งานของเครือข่าย ซึ่งสามารถสร้างทีซีพีและยูดีพี และขนาดปริมาณงานที่ทำในช่วงเวลาหนึ่งของเครือข่าย เป็นเครื่องมือที่ทันสมัยสำหรับการวัดประสิทธิภาพเครือข่ายถูกเขียนใน C++

ไอเพิร์ฟยินยอมให้ผู้ใช้ติดตั้งค่าพารามิเตอร์ได้หลากหลาย ซึ่งสามารถถูกใช้ในการทดสอบเครือข่าย หรือเป็นทางเลือกให้เหมาะสมหรือสอดคล้องกันในเครือข่ายไอเพิร์ฟ มีการทำงานทางฝั่งของผู้รับบริการและผู้ให้บริการ และสามารถวัดขนาดของปริมาณงานที่ทำในช่วงเวลาหนึ่งระหว่าง 2 ฝั่ง โดยไปในทิศทางเดียวกันหรือทั้งสองทิศทาง มันคือซอฟแวร์ต้นรหัสเปิด (open source) และรันบนระบบที่เราใช้งานได้หลากหลาย เช่น ลินุกซ์ ยูนิกซ์ และวินโดว์ส

เมื่อถูกใช้การทดสอบปริมาณยูดีพี ไอเพิร์ฟยินยอมให้ผู้ใช้ระบุขนาดของดาต้าแกรมและผลของดาต้าแกรมทรูพุท และแพ็คเกตที่หายไป

เมื่อถูกใช้การทดสอบปริมาณทีซีพี ไอเพิร์ฟวัดปริมาณงาน และใช้ 1024*1024เมกกะไบต์ และ 1000*1000 เมกกะบิต

ผลที่ได้ของไอเพิร์ฟจะบรรจุรายงานเรื่องของการระบุเวลาของผลรวมของการส่งข้อมูลและการปริมาณงานที่ทำในช่วงเวลาหนึ่ง

ไอเพิร์ฟเป็นเครื่องมือที่ได้มาตรฐาน ซึ่งสามารถรันบนเครือข่ายและได้ผลลัพธ์ที่มีการวัดประสิทธิภาพที่ได้มาตรฐาน ดังนั้นจึงสามารถใช้เปรียบเทียบเครือข่ายใช้สายและไร้สายได้ และใช้เทคโนโลยีที่ตรงไปตรงมา และเป็นต้นรหัสเปิด เทคนิควิธีการวัดที่สามารถตรวจสอบได้โดยผู้ใช้

บทที่ 5
ขั้นตอนการทดลองและผลการทดลอง

ในการทดลองวัดสมรรถนะการทำงานของโพรโทคอลเออาร์เอเอ็น จะติดตั้งโพรโทคอลบนระบบปฏิบัติการอูบันตู เวอร์ชั่น 7.10 ซึ่งเป็นเวอร์ชั่นที่เก่าแล้ว จึงมีข้อผิดพลาดในการอัพเดทแพ็คเกจ (package) จากเครื่องแม่ข่ายของระบบปฏิบัติการนี้ จึงต้องเข้าไปทำการเปลี่ยนลิงค์ (link) ที่ใช้ระบุในการอัพเดทที่ไฟล์ (path) ของระบบปฏิบัติการคือที่

```
/etc/apt/source.list
```

ให้เปลี่ยนลิงค์ที่ชื่อนำหน้าด้วย **th.archrive** เป็น **old-releases** แทนจากนั้นซ้ำคำสั่ง

```
apt-get update
```

โพรโทคอลเออาร์เอเอ็น จะสามารถแบ่งออกเป็นสองส่วนหลักๆ ได้คือ

- ตัวโพรโทคอล
- ผู้ให้บริการหนังสือรับรอง (The ARAN Certificate Authority)

โดยแพ็คเกจ ที่ต้องลงเพื่อใช้ทำงานร่วมกับโพรโทคอลได้แก่

- OpenSSL
- Ad-Hoc Support Library เวอร์ชั่น 0.12
- Perl

5.1 ขั้นตอนการลงโพรโทคอล

ในขั้นตอนนี้จะเป็นการลงโพรโทคอลเพื่อทำการทดลองในขั้นต่อๆไป โดยรายละเอียดในการลงมีดังนี้

1. ลง Ad-Hoc Support Library หรือ LibASL

```
- แตกไฟล์โดยใช้คำสั่ง tar xzvf ASL-0.12
- ./configure
- sudo make install
```

2. สร้างอุโมงค์ (Tunnel) กรณีระบบปฏิบัติการไม่มีมาให้

- cd /dev/net
- mknod tun c 10 200 สร้างอุปกรณ์
- /sbin/depmod –a เพื่ออัพเดทโมดูล

3. ลง openssl

- apt-get install openssl

หรือ

- โหลดไฟล์มาแล้วมาแตกโดยคำสั่ง tar xzvf openssl-x.x.x.tar.gz
- ./configure
- sudo make install

4. ติดตั้งโพรโทคอล เออาร์เอเอ็น

- แตกไฟล์โดยใช้คำสั่ง tar xzvf arand-0.3.2
- mkdir build สร้างโฟลเดอร์ชื่อ build
- cd build
- ../configure
- sudo make install

5.2 ขั้นตอนการปรับแต่ง

ก่อนที่จะมีการใช้งานตัวโพรโทคอลนั้น ต้องมีการปรับเปลี่ยนค่าตัวแปรพารามิเตอร์ต่างๆ ให้เหมาะสมกับการทดลองโดยขั้นตอนการปรับแต่งมีดังต่อไปนี้

5.2.1 ปรับแต่งอุปกรณ์ให้อยู่เป็น 802.11 ระบบไร้สายเฉพาะกิจ โดยใช้คำสั่ง

- iwconfig eth1 mode ad-hoc สั่งให้อินเตอร์เฟสที่เป็นไร้สายเป็นไร้สายเฉพาะกิจ
- iwconfig eth1 essid 'test' ตั้งเครือข่ายไร้สายเฉพาะกิจขึ้นมาชื่อ 'test'
- iwconfig eth1 channel 11 ตั้งให้คลื่นสัญญาณอยู่ในช่อง 11
- ifconfig eth1 10.1.1.1 netmask 255.255.255.0 ตั้งค่าของหมายเลขไอพีที่ต้องการ และระบุเลขเครือข่ายเป็น /24
- ifconfig eth1 up สั่งให้เปิดพอร์ตไร้สาย

โดยในการทดลองใช้คอมพิวเตอร์สามเครื่องกำหนดให้หมายเลขไอพีคือ

- 10.1.1.1
- 10.1.1.2
- 10.1.1.3

5.2.2 สำหรับเครื่องที่เป็นผู้ให้บริการหนังสือรับรอง (Certification Authority)

- แตกไฟล์โดยใช้คำสั่ง tar xzvf เออาร์เอเอ็นca-0.1.tar.gz
- ./configure
- make install

เมื่อลงเสร็จ จะพบไฟล์ที่ชื่อ aranca.pl อยู่ในพาท aranca-0.1/bin/aranca.pl ให้ทำการคัดลอกไฟล์ไปเก็บไว้ที่โฟลเดอร์ใหม่ เพื่อที่จะได้ไม่สับสนในการสร้างหนังสือรับรอง

5.2.3 การสร้าง CA (Certification Authority)

- โดยใช้คำสั่ง aranca.pl newca

```
root@te-laptop: /home/te/Desktop/cert2
File  Edit  View  Terminal  Tabs  Help
root@te-laptop:/home/te# cd Desktop/cert2/
root@te-laptop:/home/te/Desktop/cert2# aranca.pl newca
Creating ARAN CA certificate...
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to './aran-ca/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
root@te-laptop:/home/te/Desktop/cert2#
```

รูปที่ 5.1 แสดงขั้นตอนการสร้างผู้ให้บริการใบรับรอง

โดยคีย์ที่ใส่คือคีย์ส่วนตัว (private key) จากนั้นทำการตรวจสอบภายใต้โฟลเดอร์ ซึ่งจะประกอบด้วยไฟล์หรือโฟลเดอร์ดังนี้

- Cacert.pem คือหนังสือรับรองที่ลงลายมือชื่อของผู้ให้บริการหนังสือรับรอง
- Certs

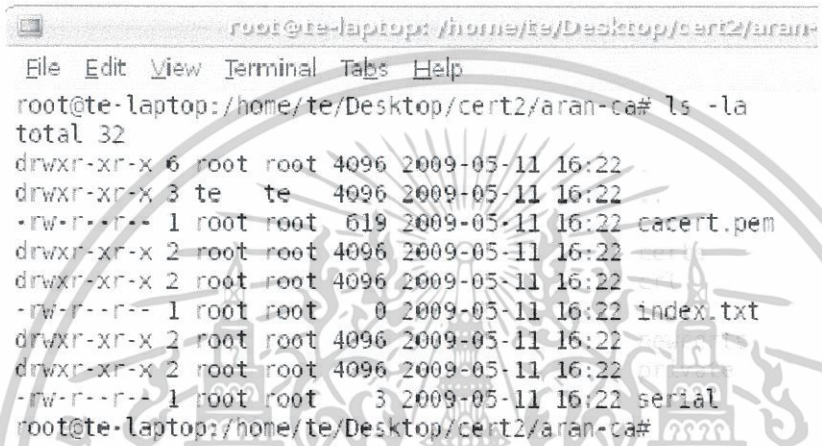
- Index.txt

  newcerts

- Private

- Serial

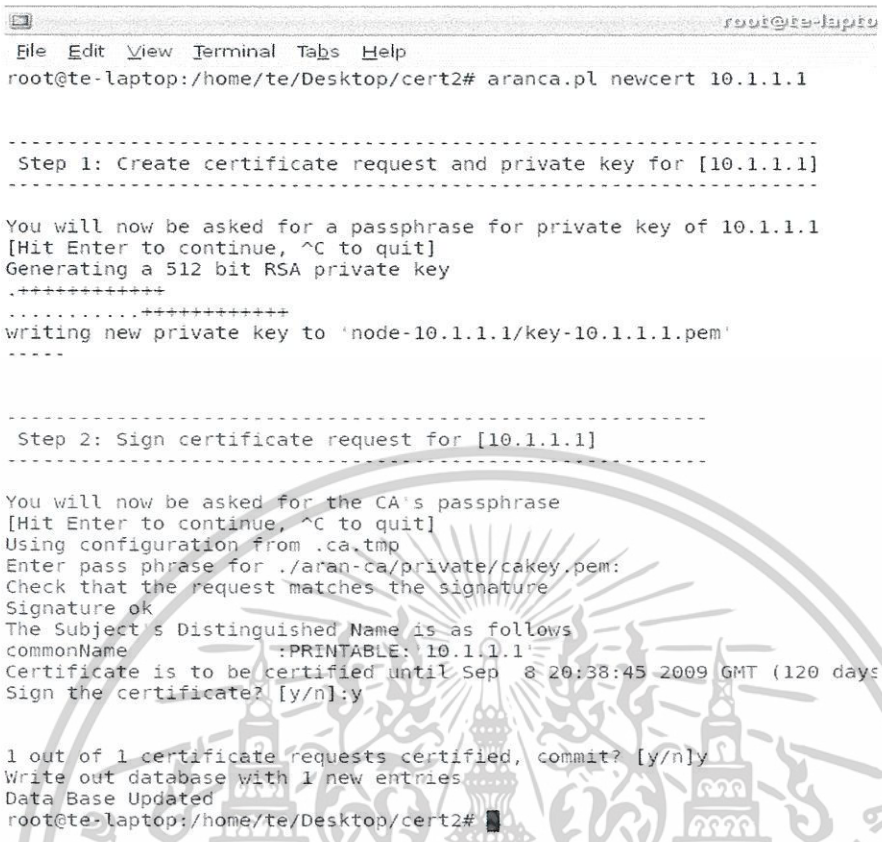
- cakey.pem คือคีย์ส่วนตัว RSAของผู้ให้บริการหนังสือรับรอง ซึ่งอยู่ภายใต้ folder private



รูปที่ 5.2 ไฟล์ที่อยู่ในโฟลเดอร์ เมื่อทำการสร้างผู้ให้บริการใบรับรอง

5.2.4 การสร้างใบรับรอง

- ใช้คำสั่ง aranca.pl newcert 10.1.1.1 สำหรับสร้างหนังสือรับรองให้แต่ละโหนด



รูปที่ 5.3 การสร้างใบรับรอง

โดยจะมีการถามถึง PEM passphrase ซึ่งใช้ในการลงลายชื่อจากผู้ให้บริการ แก่หนังสือรับรอง

- ทำเช่นนี้กับโหนด 10.1.1.2 และ 10.1.1.3

5.2.5 การติดตั้งหนังสือรับรอง

- คัดลอกไฟล์หรือใช้คำสั่ง scp 10.1.1.x.tar.gz ไปที่พาท /usr/local/etc/arand/
- แตกไฟล์โดยใช้คำสั่ง tar xzvf 10.1.1.x.tar.gz ออกมา

5.2.6 การเปลี่ยนแปลงค่าการปรับแต่งของไฟล์ arand.conf ให้เหมาะสม

- ใช้คำสั่ง vi /usr/local/etc/arand/arand.conf เพื่อแก้ไขค่าให้เหมาะสมกับเครื่องได้แก่

ca_cert

my_cert

my_private_key

โดยเปลี่ยนหมายเลขไอพีให้ตรงกับของเครื่อง

Interface แก้ไขให้ตรงกับอินเตอร์เฟสไร้สายของเครื่อง

5.3 คำสั่งในการรันโพรโทคอล

ใช้คำสั่ง /usr/local/bin/arand หรือ /usr/local/bin/arand-run-sh เพื่อเริ่มโพรโทคอลซึ่งผลจากการรันจะเป็นไปตามด้านล่าง

```
/usr/local/bin/arand
set_configurable_defaults(): entering
set_configurable_defaults(): leaving
process_cmd_line_args(): entering
process_cmd_line_args(): leaving
process_config_file(): entering
process_config_file(): read key/value pair interface eth1
process_cmd_line_args(): set interface to: eth1
process_config_file(): read key/value pair aran_port 1500
process_cmd_line_args(): set aran_port to: 1500
process_config_file(): read key/value pair CA_cert /usr/local/etc/
arand/trusted/cacert.pem
process_cmd_line_args(): CAcert: /usr/local/etc/ arand/trusted/cacert.pem
process_config_file(): read key/value pair my_cert /usr/local/etc/ arand/cert-
10.1.1.1.pem
process_cmd_line_args(): My cert: /usr/local/etc/ arand/cert-10.1.1.1.pem
interpret_config_entry(): set cert size to 339
process_config_file(): read key/value pair my_private_key /usr/local/etc/ arand/key-
10.1.1.1.pem
process_cmd_line_args(): My private key: /usr/local/etc/ arand/key-10.1.1.1.pem
interpret_config_entry(): set sig size to 64
process_config_file(): read key/value pair supress_hellos 0
process_cmd_line_args(): set supress_hellos to: 0
process_config_file(): read key/value pair hello_interval 5000
process_cmd_line_args(): set hello_interval to: 5000
process_config_file(): read key/value pair rdp_retries 3
process_cmd_line_args(): set rdp_retries to: 3
process_config_file(): read key/value pair rdp_wait_time 3000
process_cmd_line_args(): set rdp_wait_time to: 3000
```

```
process_config_file(): leaving
create_pid_file(): entering
create_pid_file(): myconfig->pid_file = /var/run/arand.pid
aranSocket_new(): sucessfully created aran socket
aranSocket_new(): bind successful for aransocket
statusSocket_new(): sucessfully created status socket
arans_et_periodic_timer(): entering
aran_set_periodic_timer(): tData type: 8
aran_set_periodic_timer(): tData data: 0
aran_set_periodic_timer(): PERIODIC_INTERVAL: 1000
aran_set_periodic_timer(): update routing table here
aran_set_periodic_timer(): leaving
aran_set_hello_timer(): entering
aran_set_hello_timer(): leaving
aran_set_status_timer(): entering
aran_set_status_timer(): leaving
routingTable_init(): entering
abroutingTable_init(): leaving
rdpForwardList_init(): entering
rdpForwardList_init(): leaving
rdpPendingList_init(): entering
rdpPendingList_init(): leaving
aran_registerHandlerFunction entering
aran_registerHandlerFunction leaving


In open_route_request
tun_init() : device tun is now up
open_route_request() : tun_init() done.
open_route_request() : rawsock_init() done.
open_route_request() : init_rtsocket() done; krt ready.
open_route_request() : packet_table_init() done.
open_route_request() : inserting route_check module in the kernel
```

```
If insmod returned error, check that  route_check is properly installed
 insmod ::insmod: can't read 'route_check': No such file or directory
open_route_request() : done.

aran_registerHandlerFunction entering
aran_registerHandlerFunction leaving
route_add() : dev is eth1
added route to kernel
added defferred route for 0.0.0.0
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         0.0.0.0         0.0.0.0         U     0      0        0 tun
aran_main_loop(): entering
aran_periodic_handler(): entering
rdpForwardList_updateEntries(): entering
rdpForwardList_updateEntries(): leaving
routingTable_refreshEntries(): entering
routingTable_refreshEntries(): leaving
****ARAND ROUTING TABLE****
****END ARAND ROUTING TABLE****

aran_periodic_handler(): leaving
aran_periodic_handler(): entering
rdpForwardList_updateEntries(): entering
rdpForwardList_updateEntries(): leaving
routingTable_refreshEntries(): entering
routingTable_refreshEntries(): leaving
****ARAND ROUTING TABLE****
****END ARAND ROUTING TABLE****
```

รูปที่ 5.4 แสดงผลเมื่อมีการรันโพรโทคอล

5.4 ขั้นตอนวัดสมรรถนะ

จะเป็นการทดสอบสมรรถนะในการ Authenticate Routing ของโพรโทคอล เออาร์เอเอ็น ว่ามีผลอย่างไรกับเครือข่าย

จากการศึกษานั้นเมื่อมีการรักษาความมั่นคงให้แก่เครือข่ายไร้สายเฉพาะกิจแล้ว สมรรถนะของระบบย่อมลดลง ดังนั้นจึงได้มีการศึกษาถึงสมรรถนะของโพรโทคอล เออาร์เอเอ็น ซึ่งเป็นโพรโทคอลที่พัฒนามาจากโพรโทคอลเอโอดีวีโดยมีการใช้หนังสือรับรองในการรักษาความมั่นคง การเข้ารหัสหนังสือรับรองจะกระทำที่ control msg. เช่น RDP REP โดย RDP นั้นก็คล้าย RREQ ของโพรโทคอลเอโอดีวีนั่นเอง เพียงแต่มีการลงลายเซ็นเข้ารหัสหนังสือรับรอง จึงเรียกว่า RDP ในโพรโทคอล เออาร์เอเอ็น

เราจึงศึกษาถึงปัจจัยถึงเวลาที่ใช้ไปเมื่อมีการใช้หนังสือรับรอง ดังนี้

<u>การทดลองที่ 1</u>

<u>วัตถุประสงค์</u> 1. เพื่อวัดค่าของ route overhead ของโพรโทคอล เออาร์เอเอ็น (Timing Phase)

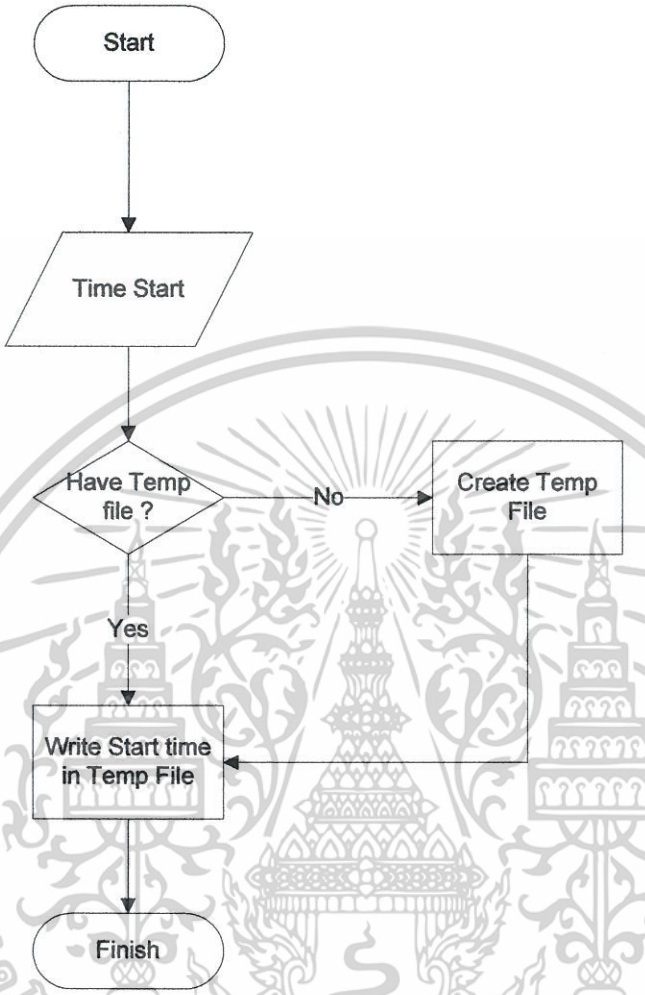
2. เพื่อศึกษาถึงค่าปัจจัยที่เปลี่ยนไปมีผลกระทบต่อการทำงานของโพรโทคอล เออาร์เอเอ็น (Key Size)

เนื่องด้วยการทำงานของโพรโทคอล เออาร์เอเอ็น นั้นจะมีการทำลายเซ็นในข้อมูลที่เป็นการแสดงเส้นทาง โดยในการทำลายเซ็นนั้นใช้หลักการของ certificate เข้ามาซึ่งในส่วนนี้สามารถเปลี่ยนขนาดของ กุญแจเข้ารหัส ได้ โดยในการทดลองนี้ ผู้ทดลองจะทำการวัดผลของขนาด กุญแจเข้ารหัส ที่แตกต่างกัน ดังนั้นผู้ทดลองได้ทำการเขียนโปรแกรม ขึ้นมาเพื่อจับเวลาเพื่อจับค่า เวลาที่ใช้ไป

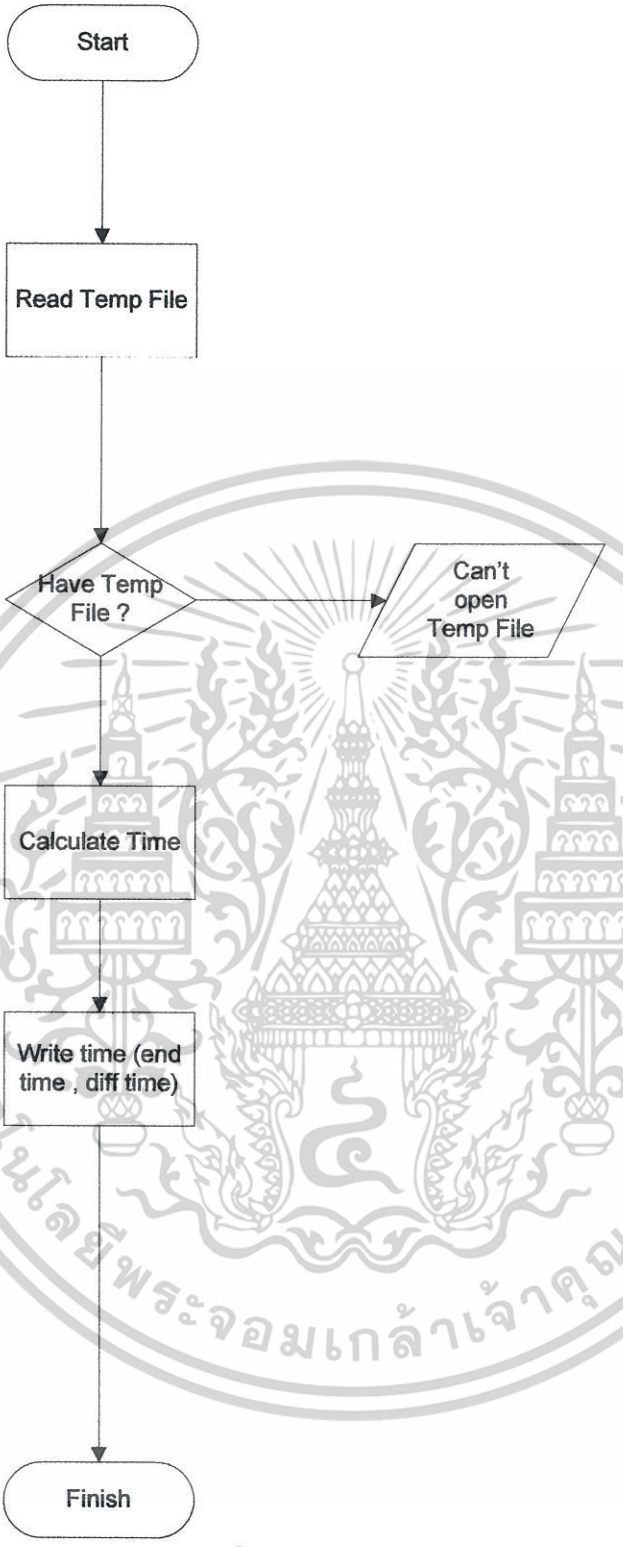
ทั้งนี้เวลาที่ใช้ไปในส่วนของโพรโทคอล เออาร์เอเอ็น นั้นจะมีอยู่สองช่วงคือ

1. ช่วงเวลาที่ใช้ไปกับการ Encrypt RDP Packet ที่จะส่งไปยังโหนดเพื่อนบ้าน
2. ช่วงเวลาที่ใช้ไปกับการตรวจสอบ RDP Packet ที่ได้รับมาจากเพื่อนบ้าน

โดยการทำงานของโปรแกรมนั้นมี Flow chart ตามภาพด้านล่างนี้



รูปที่ 5.5 แสดงกระบวนการทำงานในช่วงเริ่มต้นของการทำงานการจับเวลา



รูปที่ 5. 6 แสดงกระบวนการทำงานในช่วงเริ่มต้นของการทำงานของการหยุดเวลาและคำนวณค่าเก็บลงไฟล์

โดยการทดลองที่ 1 นี้ ผู้ทดลองได้ทำการจำลองดังภาพด้านล่าง

รูปที่ 5.7 แสดงแบบการจำลองการทดลองที่ 1

และมีการกำหนดค่าโดยแบ่งเป็นการทดลองย่อย ดังนี้

<u>การทดลองที่ 1.1</u> เพื่อหาค่าเราติ้งโอเวอร์เฮดเมื่อมีการกำหนดดังต่อไปนี้

- พารามิเตอร์ของระบบคือ กุญแจเข้ารหัสมีขนาด 768 บิต
- ระยะเวลาในการจับเวลาทั้งหมด คือ 5 นาที
- ภาระงานคือความถี่ในการส่งค้นหาเส้นทาง คือ 5 วินาที

ตารางที่ 5.1 ผลการทดลองเวลาที่ใช้ไปในการเข้ารหัส RDP Packet

1	start time = 25/05/09 11:25:00:59 , 11:25:00:64,[0h/0m/0s/5ms]
2	start time = 25/05/09 11:25:05:65 , 11:25:05:69,[0h/0m/0s/4ms]
3	start time = 25/05/09 11:25:10:99 , 11:25:10:103,[0h/0m/0s/4ms]
4	start time = 25/05/09 11:25:15:120 , 11:25:15:123,[0h/0m/0s/3ms]
5	start time = 25/05/09 11:25:20:125 , 11:25:20:127,[0h/0m/0s/2ms]
6	start time = 25/05/09 11:25:25:159 , 11:25:25:163,[0h/0m/0s/4ms]
7	start time = 25/05/09 11:25:30:168 , 11:25:30:172,[0h/0m/0s/4ms]
8	start time = 25/05/09 11:25:35:173 , 11:25:35:177,[0h/0m/0s/4ms]
9	start time = 25/05/09 11:25:40:198 , 11:25:40:202,[0h/0m/0s/4ms]
10	start time = 25/05/09 11:25:45:203 , 11:25:45:208,[0h/0m/0s/5ms]
11	start time = 25/05/09 11:25:50:209 , 11:25:50:213,[0h/0m/0s/4ms]
12	start time = 25/05/09 11:25:55:217 , 11:25:55:221,[0h/0m/0s/4ms]
13	start time = 25/05/09 11:26:00:224 , 11:26:00:228,[0h/0m/0s/4ms]
14	start time = 25/05/09 11:26:05:228 , 11:26:05:232,[0h/0m/0s/4ms]
15	start time = 25/05/09 11:26:10:235 , 11:26:10:239,[0h/0m/0s/4ms]
16	start time = 25/05/09 11:26:15:239 , 11:26:15:243,[0h/0m/0s/4ms]
17	start time = 25/05/09 11:26:20:244 , 11:26:20:248,[0h/0m/0s/4ms]
18	start time = 25/05/09 11:26:25:254 , 11:26:25:257,[0h/0m/0s/3ms]
19	start time = 25/05/09 11:26:30:257 , 11:26:30:261,[0h/0m/0s/4ms]
20	start time = 25/05/09 11:26:35:262 , 11:26:35:266,[0h/0m/0s/4ms]

```
21 start time = 25/05/09 11:26:40:268 , 11:26:40:272,[0h/0m/0s/4ms]
22 start time = 25/05/09 11:26:45:272 , 11:26:45:276,[0h/0m/0s/4ms]
23 start time = 25/05/09 11:26:50:276 , 11:26:50:280,[0h/0m/0s/4ms]
24 start time = 25/05/09 11:26:55:282 , 11:26:55:286,[0h/0m/0s/4ms]
25 start time = 25/05/09 11:27:00:286 , 11:27:00:290,[0h/0m/0s/4ms]
26 start time = 25/05/09 11:27:05:291 , 11:27:05:294,[0h/0m/0s/3ms]
27 start time = 25/05/09 11:27:10:296 , 11:27:10:299,[0h/0m/0s/3ms]
28 start time = 25/05/09 11:27:15:301 , 11:27:15:305,[0h/0m/0s/4ms]
29 start time = 25/05/09 11:27:20:306 , 11:27:20:309,[0h/0m/0s/3ms]
30 start time = 25/05/09 11:27:25:309 , 11:27:25:314,[0h/0m/0s/5ms]
31 start time = 25/05/09 11:27:30:314 , 11:27:30:318,[0h/0m/0s/4ms]
32 start time = 25/05/09 11:27:35:319 , 11:27:35:322,[0h/0m/0s/3ms]
33 start time = 25/05/09 11:27:40:323 , 11:27:40:327,[0h/0m/0s/4ms]
34 start time = 25/05/09 11:27:45:327 , 11:27:45:331,[0h/0m/0s/4ms]
35 start time = 25/05/09 11:27:50:332 , 11:27:50:334,[0h/0m/0s/2ms]
36 start time = 25/05/09 11:27:55:336 , 11:27:55:340,[0h/0m/0s/4ms]
37 start time = 25/05/09 11:28:00:340 , 11:28:00:344,[0h/0m/0s/4ms]
38 start time = 25/05/09 11:28:05:345 , 11:28:05:349,[0h/0m/0s/4ms]
39 start time = 25/05/09 11:28:10:349 , 11:28:10:353,[0h/0m/0s/4ms]
40 start time = 25/05/09 11:28:15:354 , 11:28:15:358,[0h/0m/0s/4ms]
41 start time = 25/05/09 11:28:20:358 , 11:28:20:362,[0h/0m/0s/4ms]
42 start time = 25/05/09 11:28:25:362 , 11:28:25:366,[0h/0m/0s/4ms]
43 start time = 25/05/09 11:28:30:367 , 11:28:30:371,[0h/0m/0s/4ms]
44 start time = 25/05/09 11:28:35:371 , 11:28:35:375,[0h/0m/0s/4ms]
45 start time = 25/05/09 11:28:40:380 , 11:28:40:384,[0h/0m/0s/4ms]
46 start time = 25/05/09 11:28:45:388 , 11:28:45:391,[0h/0m/0s/3ms]
47 start time = 25/05/09 11:28:50:392 , 11:28:50:396,[0h/0m/0s/4ms]
48 start time = 25/05/09 11:28:55:396 , 11:28:55:400,[0h/0m/0s/4ms]
49 start time = 25/05/09 11:29:00:401 , 11:29:00:405,[0h/0m/0s/4ms]
50 start time = 25/05/09 11:29:05:406 , 11:29:05:410,[0h/0m/0s/4ms]
51 start time = 25/05/09 11:29:10:410 , 11:29:10:414,[0h/0m/0s/4ms]
52 start time = 25/05/09 11:29:15:414 , 11:29:15:418,[0h/0m/0s/4ms]
53 start time = 25/05/09 11:29:20:418 , 11:29:20:422,[0h/0m/0s/4ms]
54 start time = 25/05/09 11:29:25:422 , 11:29:25:426,[0h/0m/0s/4ms]
55 start time = 25/05/09 11:29:30:426 , 11:29:30:430,[0h/0m/0s/4ms]
56 start time = 25/05/09 11:29:35:431 , 11:29:35:435,[0h/0m/0s/4ms]
57 start time = 25/05/09 11:29:40:435 , 11:29:40:439,[0h/0m/0s/4ms]
58 start time = 25/05/09 11:29:45:441 , 11:29:45:445,[0h/0m/0s/4ms]
59 start time = 25/05/09 11:29:50:445 , 11:29:50:449,[0h/0m/0s/4ms]
60 start time = 25/05/09 11:29:55:451 , 11:29:55:456,[0h/0m/0s/5ms]
```

ผลลัพธ์ที่ได้คือในเวลา 5 นาที มีการเข้ารหัส ทั้งหมด 61 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการเข้ารหัส ทั้ง 61 ครั้งเป็น 3.885 มิลลิวินาที

ตารางที่ 5.2 ผลการทดลองเวลาที่ใช้ในการตรวจสอบ RDP Packet

1	start time = 22/05/09 12:58:42:878 ,12:58:42:883,
2	[0h/0m/0s/5ms]
3	start time = 22/05/09 12:58:47:883 , 12:58:47:884, [0h/0m/0s/1ms]
4	start time = 22/05/09 12:58:52:888 , 12:58:52:894, [0h/0m/0s/6ms]
5	start time = 22/05/09 12:58:57:893 , 12:58:57:895, [0h/0m/0s/2ms]
6	start time = 22/05/09 12:59:02:897 , 12:59:02:899, [0h/0m/0s/2ms]
7	start time = 22/05/09 12:59:07:915 , 12:59:07:917, [0h/0m/0s/2ms]
8	start time = 22/05/09 12:59:12:908 , 12:59:12:909, [0h/0m/0s/1ms]
9	start time = 22/05/09 12:59:17:912 , 12:59:17:913, [0h/0m/0s/1ms]
10	start time = 22/05/09 12:59:22:919 , 12:59:22:921, [0h/0m/0s/2ms]
11	start time = 22/05/09 12:59:27:930 , 12:59:27:931, [0h/0m/0s/1ms]
12	start time = 22/05/09 12:59:32:925 , 12:59:32:926, [0h/0m/0s/1ms]
13	start time = 22/05/09 12:59:37:930 , 12:59:37:931, [0h/0m/0s/1ms]
14	start time = 22/05/09 12:59:42:935 , 12:59:42:936, [0h/0m/0s/1ms]
15	start time = 22/05/09 12:59:47:939 , 12:59:47:941, [0h/0m/0s/2ms]
16	start time = 22/05/09 12:59:52:971 , 12:59:52:972, [0h/0m/0s/1ms]
17	start time = 22/05/09 12:59:57:975 , 12:59:57:977, [0h/0m/0s/2ms]
18	start time = 22/05/09 13:00:02:980 , 13:00:02:982, [0h/0m/0s/2ms]
19	start time = 22/05/09 13:00:07:985 , 13:00:07:986, [0h/0m/0s/1ms]
20	start time = 22/05/09 13:00:12:990 , 13:00:12:992, [0h/0m/0s/2ms]
21	start time = 22/05/09 13:00:17:995 , 13:00:17:997, [0h/0m/0s/2ms]
22	start time = 22/05/09 13:00:23:0 , 13:00:23:1 , [0h/0m/0s/1ms]
23	start time = 22/05/09 13:00:28:58 , 13:00:28:59, [0h/0m/0s/1ms]
24	start time = 22/05/09 13:00:38:93 , 13:00:38:94, [0h/0m/0s/1ms]
25	start time = 22/05/09 13:00:43:19 , 13:00:43:20, [0h/0m/0s/1ms]
26	start time = 22/05/09 13:00:48:22 , 13:00:48:24, [0h/0m/0s/2ms]
27	start time = 22/05/09 13:00:53:44 , 13:00:53:46, [0h/0m/0s/2ms]
28	start time = 22/05/09 13:00:58:62 , 13:00:58:63, [0h/0m/0s/1ms]
29	start time = 22/05/09 13:01:03:80 , 13:01:03:81, [0h/0m/0s/1ms]
30	start time = 22/05/09 13:01:08:98 , 13:01:08:99, [0h/0m/0s/1ms]
31	start time = 22/05/09 13:01:13:46 , 13:01:13:48, [0h/0m/0s/2ms]
32	start time = 22/05/09 13:01:18:134 , 13:01:18:135, [0h/0m/0s/1ms]
33	start time = 22/05/09 13:01:23:152 , 13:01:23:153, [0h/0m/0s/1ms]
34	start time = 22/05/09 13:01:28:67 , 13:01:28:68, [0h/0m/0s/1ms]
35	start time = 22/05/09 13:01:33:85 , 13:01:33:87, [0h/0m/0s/2ms]
36	start time = 22/05/09 13:01:38:102 , 13:01:38:104, [0h/0m/0s/2ms]
37	start time = 22/05/09 13:01:43:121 , 13:01:43:122, [0h/0m/0s/1ms]
38	start time = 22/05/09 13:01:48:138 , 13:01:48:138, [0h/0m/0s/0ms]
39	start time = 22/05/09 13:01:53:155 , 13:01:53:156, [0h/0m/0s/1ms]
40	start time = 22/05/09 13:01:58:173 , 13:01:58:174, [0h/0m/0s/1ms]
41	start time = 22/05/09 13:02:03:192 , 13:02:03:193, [0h/0m/0s/1ms]
42	start time = 22/05/09 13:02:08:107 , 13:02:08:109, [0h/0m/0s/2ms]
43	start time = 22/05/09 13:02:13:103 , 13:02:13:106, [0h/0m/0s/3ms]
44	start time = 22/05/09 13:02:23:160 , 13:02:23:162, [0h/0m/0s/2ms]
45	start time = 22/05/09 13:02:28:178 , 13:02:28:179, [0h/0m/0s/1ms]
46	start time = 22/05/09 13:02:33:196 , 13:02:33:197, [0h/0m/0s/1ms]
47	start time = 22/05/09 13:02:38:214 , 13:02:38:215, [0h/0m/0s/1ms]
48	start time = 22/05/09 13:02:43:133 , 13:02:43:134, [0h/0m/0s/1ms]
49	start time = 22/05/09 13:02:53:165 , 13:02:53:166, [0h/0m/0s/1ms]
50	start time = 22/05/09 13:02:58:183 , 13:02:58:184, [0h/0m/0s/1ms]
51	start time = 22/05/09 13:03:03:201 , 13:03:03:202, [0h/0m/0s/1ms]
52	start time = 22/05/09 13:03:08:156 , 13:03:08:157, [0h/0m/0s/1ms]
53	start time = 22/05/09 13:03:13:236 , 13:03:13:237, [0h/0m/0s/1ms]
54	start time = 22/05/09 13:03:18:165 , 13:03:18:166, [0h/0m/0s/1ms]
55	start time = 22/05/09 13:03:23:174 , 13:03:23:175, [0h/0m/0s/1ms]
56	start time = 22/05/09 13:03:28:178 , 13:03:28:179, [0h/0m/0s/1ms]
57	start time = 22/05/09 13:03:38:183 , 13:03:38:189, [0h/0m/0s/6ms]
58	start time = 22/05/09 13:03:43:188 , 13:03:43:194, [0h/0m/0s/6ms]
59	start time = 22/05/09 13:03:48:193 , 13:03:48:201, [0h/0m/0s/8ms]

60	start time = 22/05/09 13:03:53:197 , 13:03:53:205, [0h/0m/0s/8ms] start time = 22/05/09 13:03:58:204 , 13:03:58:211, [0h/0m/0s/7ms]

ผลลัพธ์ที่ได้คือในเวลา 5 นาที มีการ Verify Certificate ทั้งหมด 60 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการ Verified ทั้ง 60 ครั้งเป็น 1.833 มิลลิวินาที

<u>การทดลองที่ 1.2</u> เพื่อหาค่าเราติ้งโอเวอร์เฮดเมื่อมีการกำหนดดังต่อไปนี้

- พารามิเตอร์ของระบบคือ กุญแจเข้ารหัสมีขนาด 1024 บิต
- ระยะเวลาในการจับเวลาทั้งหมด คือ 5 นาที
- ภาระงานคือความถี่ในการส่งค้นหาเส้นทาง คือ 5 วินาที

ตารางที่ 5.3 ผลการทดลองเวลาที่ใช้ไปในการเข้ารหัส RDP Packet

1	start time = 25/05/09 11:39:04:843 , 11:39:04:850,[0h/0m/0s/7ms]
2	start time = 25/05/09 11:39:09:851 , 11:39:09:858,[0h/0m/0s/7ms]
3	start time = 25/05/09 11:39:14:859 , 11:39:14:865,[0h/0m/0s/6ms]
4	start time = 25/05/09 11:39:19:866 , 11:39:19:872,[0h/0m/0s/6ms]
5	start time = 25/05/09 11:39:24:876 , 11:39:24:883,[0h/0m/0s/7ms]
6	start time = 25/05/09 11:39:29:888 , 11:39:29:895,[0h/0m/0s/7ms]
7	start time = 25/05/09 11:39:34:896 , 11:39:34:903,[0h/0m/0s/7ms]
8	start time = 25/05/09 11:39:39:903 , 11:39:39:910,[0h/0m/0s/7ms]
9	start time = 25/05/09 11:39:44:911 , 11:39:44:918,[0h/0m/0s/7ms]
10	start time = 25/05/09 11:39:49:919 , 11:39:49:923,[0h/0m/0s/4ms]
11	start time = 25/05/09 11:39:54:924 , 11:39:54:931,[0h/0m/0s/7ms]
12	start time = 25/05/09 11:39:59:932 , 11:39:59:939,[0h/0m/0s/7ms]
13	start time = 25/05/09 11:40:04:939 , 11:40:04:943,[0h/0m/0s/4ms]
14	start time = 25/05/09 11:40:09:943 , 11:40:09:950,[0h/0m/0s/7ms]
15	start time = 25/05/09 11:40:14:951 , 11:40:14:957,[0h/0m/0s/6ms]
16	start time = 25/05/09 11:40:19:958 , 11:40:19:962,[0h/0m/0s/4ms]
17	start time = 25/05/09 11:40:24:962 , 11:40:24:969,[0h/0m/0s/7ms]
18	start time = 25/05/09 11:40:29:970 , 11:40:29:976,[0h/0m/0s/6ms]
19	start time = 25/05/09 11:40:34:976 , 11:40:34:980,[0h/0m/0s/4ms]
20	start time = 25/05/09 11:40:39:981 , 11:40:39:987,[0h/0m/0s/6ms]
21	start time = 25/05/09 11:40:44:988 , 11:40:44:994,[0h/0m/0s/6ms]
22	start time = 25/05/09 11:40:49:995 , 11:40:50:1,[0h/0m/0s/6ms]
23	start time = 25/05/09 11:40:55:2 , 11:40:55:8,[0h/0m/0s/6ms]
24	start time = 25/05/09 11:41:00:9 , 11:41:00:15,[0h/0m/0s/6ms]
25	start time = 25/05/09 11:41:05:16 , 11:41:05:20,[0h/0m/0s/4ms]
26	start time = 25/05/09 11:41:10:22 , 11:41:10:28,[0h/0m/0s/6ms]
27	start time = 25/05/09 11:41:15:29 , 11:41:15:33,[0h/0m/0s/4ms]
28	start time = 25/05/09 11:41:20:36 , 11:41:20:44,[0h/0m/0s/8ms]
29	start time = 25/05/09 11:41:25:45 , 11:41:25:49,[0h/0m/0s/4ms]
30	start time = 25/05/09 11:41:30:49 , 11:41:30:56,[0h/0m/0s/7ms]
31	start time = 25/05/09 11:41:35:57 , 11:41:35:63,[0h/0m/0s/6ms]
32	start time = 25/05/09 11:41:40:64 , 11:41:40:69,[0h/0m/0s/5ms]
33	start time = 25/05/09 11:41:45:69 , 11:41:45:76,[0h/0m/0s/7ms]
34	start time = 25/05/09 11:41:50:77 , 11:41:50:81,[0h/0m/0s/4ms]
35	start time = 25/05/09 11:41:55:84 , 11:41:55:90,[0h/0m/0s/6ms]
36	start time = 25/05/09 11:42:00:90 , 11:42:00:94,[0h/0m/0s/4ms]
37	start time = 25/05/09 11:42:05:94 , 11:42:05:99,[0h/0m/0s/5ms]
38	start time = 25/05/09 11:42:10:99 , 11:42:10:103,[0h/0m/0s/4ms]

39	start time = 25/05/09 11:42:15:104 , 11:42:15:108,[0h/0m/0s/4ms]
40	start time = 25/05/09 11:42:20:109 , 11:42:20:113,[0h/0m/0s/4ms]
41	start time = 25/05/09 11:42:25:113 , 11:42:25:120,[0h/0m/0s/7ms]
42	start time = 25/05/09 11:42:30:121 , 11:42:30:124,[0h/0m/0s/3ms]
43	start time = 25/05/09 11:42:35:126 , 11:42:35:132,[0h/0m/0s/6ms]
44	start time = 25/05/09 11:42:40:139 , 11:42:40:145,[0h/0m/0s/6ms]
45	start time = 25/05/09 11:42:45:145 , 11:42:45:151,[0h/0m/0s/6ms]
46	start time = 25/05/09 11:42:50:152 , 11:42:50:156,[0h/0m/0s/4ms]
47	start time = 25/05/09 11:42:55:163 , 11:42:55:170,[0h/0m/0s/7ms]
48	start time = 25/05/09 11:43:00:171 , 11:43:00:177,[0h/0m/0s/6ms]
49	start time = 25/05/09 11:43:05:179 , 11:43:05:185,[0h/0m/0s/6ms]
50	start time = 25/05/09 11:43:10:192 , 11:43:10:198,[0h/0m/0s/6ms]
51	start time = 25/05/09 11:43:15:198 , 11:43:15:205,[0h/0m/0s/7ms]
52	start time = 25/05/09 11:43:20:206 , 11:43:20:212,[0h/0m/0s/6ms]
53	start time = 25/05/09 11:43:25:216 , 11:43:25:222,[0h/0m/0s/6ms]
54	start time = 25/05/09 11:43:30:223 , 11:43:30:227,[0h/0m/0s/4ms]
55	start time = 25/05/09 11:43:35:228 , 11:43:35:234,[0h/0m/0s/6ms]
56	start time = 25/05/09 11:43:40:235 , 11:43:40:241,[0h/0m/0s/6ms]
57	start time = 25/05/09 11:43:45:242 , 11:43:45:249,[0h/0m/0s/7ms]
58	start time = 25/05/09 11:43:50:276 , 11:43:50:284,[0h/0m/0s/8ms]
59	start time = 25/05/09 11:43:55:284 , 11:43:55:291,[0h/0m/0s/7ms]

ผลลัพธ์ที่ได้คือในเวลา 5 นาที มีการเข้ารหัส ทั้งหมด 59 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการ เข้ารหัส ทั้ง 59 ครั้งเป็น 5.81 มิลลิวินาที

ตารางที่ 5.4 ผลการทดลองเวลาที่ใช้ในการตรวจสอบ RDP Packet

1	start time = 22/05/09 12:06:01:481 , 12:06:01:482, [0h/0m/0s/1ms]
2	start time = 22/05/09 12:06:06:487 , 12:06:06:488, [0h/0m/0s/1ms]
3	start time = 22/05/09 12:06:11:494 , 12:06:11:496, [0h/0m/0s/2ms]
4	start time = 22/05/09 12:06:16:499 , 12:06:16:501, [0h/0m/0s/2ms]
5	start time = 22/05/09 12:06:21:511 , 12:06:21:514, [0h/0m/0s/3ms]
6	start time = 22/05/09 12:06:26:510 , 12:06:26:518, [0h/0m/0s/8ms]
7	start time = 22/05/09 12:06:31:515 , 12:06:31:517, [0h/0m/0s/2ms]
8	start time = 22/05/09 12:06:36:520 , 12:06:36:523, [0h/0m/0s/3ms]
9	start time = 22/05/09 12:06:41:525 , 12:06:41:527, [0h/0m/0s/2ms]
10	start time = 22/05/09 12:06:46:530 , 12:06:46:533, [0h/0m/0s/3ms]
11	start time = 22/05/09 12:06:51:536 , 12:06:51:539, [0h/0m/0s/3ms]
12	start time = 22/05/09 12:06:56:555 , 12:06:56:557, [0h/0m/0s/2ms]
13	start time = 22/05/09 12:07:01:561 , 12:07:01:565, [0h/0m/0s/4ms]
14	start time = 22/05/09 12:07:06:569 , 12:07:06:571, [0h/0m/0s/2ms]
15	start time = 22/05/09 12:07:11:574 , 12:07:11:576, [0h/0m/0s/2ms]
16	start time = 22/05/09 12:07:16:584 , 12:07:16:586, [0h/0m/0s/2ms]
17	start time = 22/05/09 12:07:21:590 , 12:07:21:593, [0h/0m/0s/3ms]
18	start time = 22/05/09 12:07:26:598 , 12:07:26:600, [0h/0m/0s/2ms]
19	start time = 22/05/09 12:07:31:598 , 12:07:31:607, [0h/0m/0s/9ms]
20	start time = 22/05/09 12:07:36:603 , 12:07:36:604, [0h/0m/0s/1ms]
21	start time = 22/05/09 12:07:41:608 , 12:07:41:611, [0h/0m/0s/3ms]
22	start time = 22/05/09 12:07:46:614 , 12:07:46:616, [0h/0m/0s/2ms]
23	start time = 22/05/09 12:07:51:619 , 12:07:51:623, [0h/0m/0s/4ms]
24	start time = 22/05/09 12:07:56:623 , 12:07:56:627, [0h/0m/0s/4ms]
25	start time = 22/05/09 12:08:01:629 , 12:08:01:631, [0h/0m/0s/2ms]
26	start time = 22/05/09 12:08:06:649 , 12:08:06:659, [0h/0m/0s/10ms]
27	start time = 22/05/09 12:08:11:654 , 12:08:11:656, [0h/0m/0s/2ms]
28	start time = 22/05/09 12:08:16:658 , 12:08:16:661, [0h/0m/0s/3ms]

29	start time = 22/05/09 12:08:21:664 , 12:08:21:666, [0h/0m/0s/2ms]
30	start time = 22/05/09 12:08:26:669 , 12:08:26:671, [0h/0m/0s/2ms]
31	start time = 22/05/09 12:08:31:674 , 12:08:31:676, [0h/0m/0s/2ms]
32	start time = 22/05/09 12:08:36:679 , 12:08:36:682, [0h/0m/0s/3ms]
33	start time = 22/05/09 12:08:41:684 , 12:08:41:686, [0h/0m/0s/2ms]
34	start time = 22/05/09 12:08:46:689 , 12:08:46:692, [0h/0m/0s/3ms]
35	start time = 22/05/09 12:08:51:695 , 12:08:51:698, [0h/0m/0s/3ms]
36	start time = 22/05/09 12:08:56:698 , 12:08:56:700, [0h/0m/0s/2ms]
37	start time = 22/05/09 12:09:01:703 , 12:09:01:705, [0h/0m/0s/2ms]
38	start time = 22/05/09 12:09:06:708 , 12:09:06:712, [0h/0m/0s/4ms]
39	start time = 22/05/09 12:09:11:719 , 12:09:11:721, [0h/0m/0s/2ms]
40	start time = 22/05/09 12:09:16:721 , 12:09:16:724, [0h/0m/0s/3ms]
41	start time = 22/05/09 12:09:21:741 , 12:09:21:744, [0h/0m/0s/3ms]
42	start time = 22/05/09 12:09:26:746 , 12:09:26:748, [0h/0m/0s/2ms]
43	start time = 22/05/09 12:09:31:752 , 12:09:31:754, [0h/0m/0s/2ms]
44	start time = 22/05/09 12:09:36:758 , 12:09:36:761, [0h/0m/0s/3ms]
45	start time = 22/05/09 12:09:41:764 , 12:09:41:767, [0h/0m/0s/3ms]
46	start time = 22/05/09 12:09:46:770 , 12:09:46:773, [0h/0m/0s/3ms]
47	start time = 22/05/09 12:09:51:776 , 12:09:51:779, [0h/0m/0s/3ms]
48	start time = 22/05/09 12:09:56:782 , 12:09:56:785, [0h/0m/0s/3ms]
49	start time = 22/05/09 12:10:01:792 , 12:10:01:794, [0h/0m/0s/2ms]
50	start time = 22/05/09 12:10:06:798 , 12:10:06:800, [0h/0m/0s/2ms]
51	start time = 22/05/09 12:10:11:804 , 12:10:11:807, [0h/0m/0s/3ms]
52	start time = 22/05/09 12:10:16:812 , 12:10:16:815, [0h/0m/0s/3ms]
53	start time = 22/05/09 12:10:21:818 , 12:10:21:820, [0h/0m/0s/2ms]
54	start time = 22/05/09 12:10:26:823 , 12:10:26:826, [0h/0m/0s/3ms]
55	start time = 22/05/09 12:10:31:828 , 12:10:31:830, [0h/0m/0s/2ms]
56	start time = 22/05/09 12:10:36:834 , 12:10:36:837, [0h/0m/0s/3ms]
57	start time = 22/05/09 12:10:41:840 , 12:10:41:842, [0h/0m/0s/2ms]
58	start time = 22/05/09 12:10:46:845 , 12:10:46:847, [0h/0m/0s/2ms]
59	start time = 22/05/09 12:10:51:850 , 12:10:51:853, [0h/0m/0s/3ms]
60	start time = 22/05/09 12:10:56:856 , 12:10:56:858, [0h/0m/0s/2ms]

ผลลัพธ์คือในเวลา 5 นาที มีการ Verify Certificate ทั้งหมด 60 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการ Verified ทั้ง 60 ครั้งเป็น 3.033 มิลลิวินาที

<u>การทดลองที่ 1.3</u> เพื่อหาค่าเราติ้งโอเวอร์เฮดเมื่อมีการกำหนดดังต่อไปนี้

- พารามิเตอร์ของระบบคือ กุญแจเข้ารหัสมีขนาด 768 บิต
- ระยะเวลาในการจับเวลาทั้งหมด คือ 5 นาที
- ภาระงานคือความถี่ในการส่งค้นหาเส้นทาง คือ 10 วินาที

ตารางที่ 5.5 ผลการทดลองเวลาที่ใช้ไปในการเข้ารหัส RDP Packet

1	start time = 25/05/09 13:30:08:81 , 13:30:08:85,[0h/0m/0s/4ms]
2	start time = 25/05/09 13:30:18:86 , 13:30:18:90,[0h/0m/0s/4ms]
3	start time = 25/05/09 13:30:28:91 , 13:30:28:96,[0h/0m/0s/5ms]
4	start time = 25/05/09 13:30:38:97 , 13:30:38:100,[0h/0m/0s/3ms]
5	start time = 25/05/09 13:30:48:101 , 13:30:48:106,[0h/0m/0s/5ms]
6	start time = 25/05/09 13:30:58:109 , 13:30:58:113,[0h/0m/0s/4ms]
7	start time = 25/05/09 13:31:08:116 , 13:31:08:120,[0h/0m/0s/4ms]
8	start time = 25/05/09 13:31:18:121 , 13:31:18:124,[0h/0m/0s/3ms]
9	start time = 25/05/09 13:31:28:126 , 13:31:28:130,[0h/0m/0s/4ms]

10	start time = 25/05/09 13:31:38:131 , 13:31:38:135,[0h/0m/0s/4ms]
11	start time = 25/05/09 13:31:48:135 , 13:31:48:139,[0h/0m/0s/4ms]
12	start time = 25/05/09 13:31:58:139 , 13:31:58:144,[0h/0m/0s/5ms]
13	start time = 25/05/09 13:32:08:145 , 13:32:08:148,[0h/0m/0s/3ms]
14	start time = 25/05/09 13:32:18:149 , 13:32:18:153,[0h/0m/0s/4ms]
15	start time = 25/05/09 13:32:28:154 , 13:32:28:158,[0h/0m/0s/4ms]
16	start time = 25/05/09 13:32:38:158 , 13:32:38:162,[0h/0m/0s/4ms]
17	start time = 25/05/09 13:32:48:163 , 13:32:48:167,[0h/0m/0s/4ms]
18	start time = 25/05/09 13:32:58:169 , 13:32:58:172,[0h/0m/0s/3ms]
19	start time = 25/05/09 13:33:08:173 , 13:33:08:178,[0h/0m/0s/5ms]
20	start time = 25/05/09 13:33:18:178 , 13:33:18:182,[0h/0m/0s/4ms]
21	start time = 25/05/09 13:33:28:184 , 13:33:28:188,[0h/0m/0s/4ms]
22	start time = 25/05/09 13:33:38:189 , 13:33:38:193,[0h/0m/0s/4ms]
23	start time = 25/05/09 13:33:48:193 , 13:33:48:197,[0h/0m/0s/4ms]
24	start time = 25/05/09 13:33:58:198 , 13:33:58:202,[0h/0m/0s/4ms]
25	start time = 25/05/09 13:34:08:203 , 13:34:08:207,[0h/0m/0s/4ms]
26	start time = 25/05/09 13:34:18:208 , 13:34:18:213,[0h/0m/0s/5ms]
27	start time = 25/05/09 13:34:28:213 , 13:34:28:217,[0h/0m/0s/4ms]
28	start time = 25/05/09 13:34:38:217 , 13:34:38:220,[0h/0m/0s/3ms]
29	start time = 25/05/09 13:34:48:220 , 13:34:48:224,[0h/0m/0s/4ms]
30	start time = 25/05/09 13:34:58:224 , 13:34:58:228,[0h/0m/0s/4ms]

ผลลัพธ์ที่ได้คือในเวลา 5 นาที มีการเข้ารหัส ทั้งหมด 30 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการ เข้ารหัส ทั้ง 30 ครั้งเป็น 4 มิลลิวินาที

ตารางที่ 5.6 ผลการทดลองเวลาที่ใช้ในการตรวจสอบ RDP Packet

1	start time = 22/05/09 12:38:06:469 , 12:38:06:470, [0h/0m/0s/1ms]
2	start time = 22/05/09 12:38:16:473 , 12:38:16:475, [0h/0m/0s/2ms]
3	start time = 22/05/09 12:38:26:478 , 12:38:26:480, [0h/0m/0s/2ms]
4	start time = 22/05/09 12:38:36:483 , 12:38:36:485, [0h/0m/0s/2ms]
5	start time = 22/05/09 12:38:46:488 , 12:38:46:489, [0h/0m/0s/1ms]
6	start time = 22/05/09 12:38:56:492 , 12:38:56:492, [0h/0m/0s/0ms]
7	start time = 22/05/09 12:39:06:495 , 12:39:06:497, [0h/0m/0s/2ms]
8	start time = 22/05/09 12:39:16:500 , 12:39:16:501, [0h/0m/0s/1ms]
9	start time = 22/05/09 12:39:26:505 , 12:39:26:506, [0h/0m/0s/1ms]
10	start time = 22/05/09 12:39:36:510 , 12:39:36:511, [0h/0m/0s/1ms]
11	start time = 22/05/09 12:39:46:602 , 12:39:46:604, [0h/0m/0s/2ms]
12	start time = 22/05/09 12:39:56:518 , 12:39:56:520, [0h/0m/0s/2ms]
13	start time = 22/05/09 12:40:06:523 , 12:40:06:524, [0h/0m/0s/1ms]
14	start time = 22/05/09 12:40:16:607 , 12:40:16:608, [0h/0m/0s/1ms]
15	start time = 22/05/09 12:40:26:533 , 12:40:26:534, [0h/0m/0s/1ms]
16	start time = 22/05/09 12:40:36:537 , 12:40:36:539, [0h/0m/0s/2ms]
17	start time = 22/05/09 12:40:46:542 , 12:40:46:544, [0h/0m/0s/2ms]
18	start time = 22/05/09 12:40:56:547 , 12:40:56:549, [0h/0m/0s/2ms]
19	start time = 22/05/09 12:41:06:553 , 12:41:06:554, [0h/0m/0s/1ms]
20	start time = 22/05/09 12:41:16:557 , 12:41:16:559, [0h/0m/0s/2ms]
21	start time = 22/05/09 12:41:26:562 , 12:41:26:564, [0h/0m/0s/2ms]
22	start time = 22/05/09 12:41:36:567 , 12:41:36:569, [0h/0m/0s/2ms]
23	start time = 22/05/09 12:41:46:572 , 12:41:46:574, [0h/0m/0s/2ms]
24	start time = 22/05/09 12:41:56:576 , 12:41:56:578, [0h/0m/0s/2ms]
25	start time = 22/05/09 12:42:06:582 , 12:42:06:583, [0h/0m/0s/1ms]
26	start time = 22/05/09 12:42:16:586 , 12:42:16:587, [0h/0m/0s/1ms]
27	start time = 22/05/09 12:42:26:591 , 12:42:26:593, [0h/0m/0s/2ms]
28	start time = 22/05/09 12:42:36:596 , 12:42:36:598, [0h/0m/0s/2ms]
29	start time = 22/05/09 12:42:46:601 , 12:42:46:603, [0h/0m/0s/2ms]
30	start time = 22/05/09 12:42:56:606 , 12:42:56:607, [0h/0m/0s/1ms]

ผลลัพธ์คือในเวลา 5 นาที มีการ Verify Certificate ทั้งหมด 30 ครั้ง โดยประมาณ ได้เวลาเฉลี่ยในการ Verified ทั้ง 30 ครั้งเป็น 1.533 มิลลิวินาที

สรุปการทดลองที่ 1

ก) จากผลการทดลองที่ 1.1 และ 1.2 พบว่า เมื่อเพิ่มขนาด กุญแจเข้ารหัส จาก 768 เป็น 1024 bits

- เวลาที่ใช้ไปกับการเข้ารหัส กุญแจเข้ารหัสในขนาด ที่แตกต่างกันคือ 768 และ 1024 bits นั้นได้ผลดังนี้

Key Size (bits)	Time (ms)
768	3.885
1024	5.81

- เวลาที่ใช้ไปกับการตรวจสอบในขนาด กุญแจเข้ารหัส ที่แตกต่างกันคือ 768 และ 1024 bits นั้นได้ผลดังนี้

Key Size (bits)	Time (ms)
768	1.833
1024	3.003

ข) จากผลการทดลองที่ 1.1 และ 1.3 พบว่า เมื่อเปลี่ยนการตั้งค่าความถี่ในการค้นหาเส้นทาง จากเดิม ทุกๆ 5 วินาที เป็น 10 วินาที พบว่า

- ความถี่ในการเข้ารหัสนั้นลดลงจาก 60 ครั้งเหลือ 30 ครั้งซึ่งมีความสอดคล้องกันเพราะเมื่อความถี่ในการค้นหาเส้นทางน้อยลง แพ็คเกตที่ต้องทำการเข้ารหัส ก็ลดลงด้วย ส่วนในเรื่องของเวลาที่ใช้ไปนั้นยังคงเดิม หรือใกล้เคียงกัน

Interval (Times)	Time (ms)
5	3.917
10	4

- มีความถี่ในการตรวจสอบ ลดลง 2 เท่าคือจาก 60 ครั้งเหลือ 30 ครั้ง ซึ่งมีความสอดคล้องกันเพราะเมื่อความถี่ในการค้นหาเส้นทางน้อยลง แพ็คเกตที่ต้อง Verify ก็ลดลงด้วย ส่วนในเรื่องของเวลาที่ใช้ไปนั้นยังคงเดิม หรือใกล้เคียงกัน

Interval (Times)	Time (ms)
5	1.833
10	1.533

จากการทดลองที่ 1

1. ทำให้เห็นว่าในขนาด กุญแจเข้ารหัส แตกต่างกันนั้นมีผลกับเวลาในการทำการเรียนรู้เส้นทางเพราะต้องมีการเสียเวลาไปในสองส่วนคือ ส่วนของการทำเข้ารหัสและส่วนของการตรวจสอบ
2. เรื่องของความถี่ในการส่ง RDP Packet นั้นไม่มีผลต่อเวลาที่ใช้ไปในแต่ละแพ็กเกต

ดังนั้น การทดลองที่ 1 จึงเป็นปัจจัยไปสู่การทดสอบวัดสมรรถนะในการรับส่ง ซึ่งจะเป็นการทดลองที่ 2 คือวัดทรูพุต ของ 3 กรณี คือ

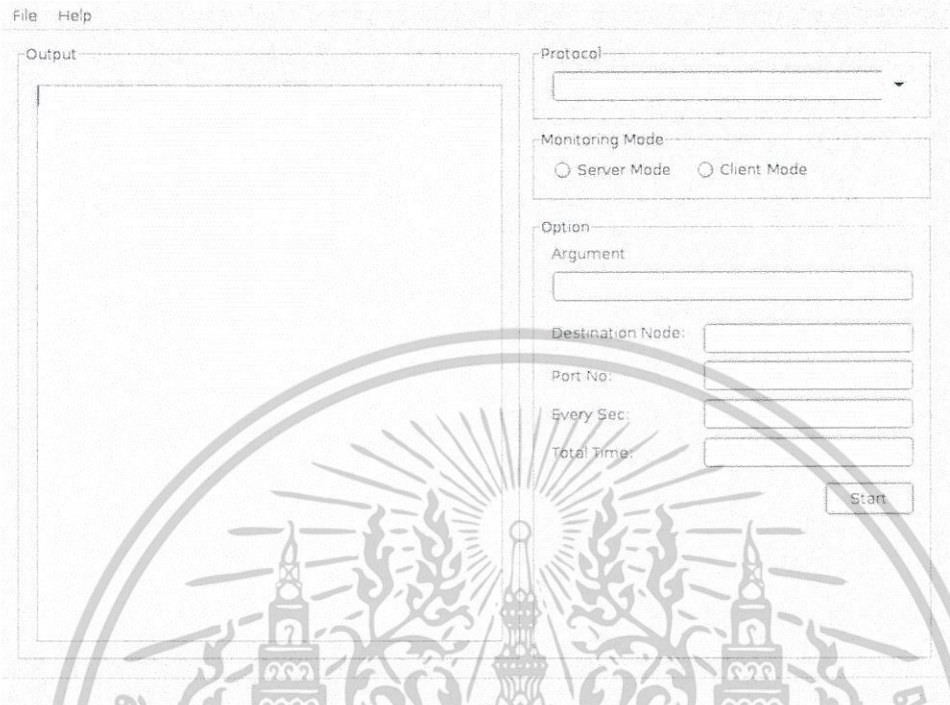
- เมื่อมี กุญแจเข้ารหัส ต่างกัน
- เมื่อมีความถี่ในการส่งค้นหาเส้นทางต่างกัน (ดูว่ามีผลต่อการ congestion ของเครือข่ายแค่ไหน)
- เมื่อมีการ Authenticate และไม่มี (เอโอดีวี , เออาร์เอเอ็น)

การทดลองที่ 2

<u>วัตถุประสงค์</u>

1. เพื่อศึกษาถึงประสิทธิภาพการใช้งานในกรณี กุญแจเข้ารหัส ขนาดต่างกัน
2. เพื่อศึกผลของความถี่ในการส่งค้นหาเส้นทางมีผลต่อประสิทธิภาพการทำงานหรือไม่

การวัดสมรรถนะในการทดลองนี้ ทำการวัดโดยสร้างแพ็คเกต เพื่อทำการทดสอบสมรรถนะ เพื่อความสะดวกในการใช้งาน จึงทำการเขียนโปรแกรมขึ้นมาดังรูปที่ 5.8



รูปที่ 5.8 หน้าจอโปรแกรมของระบบ

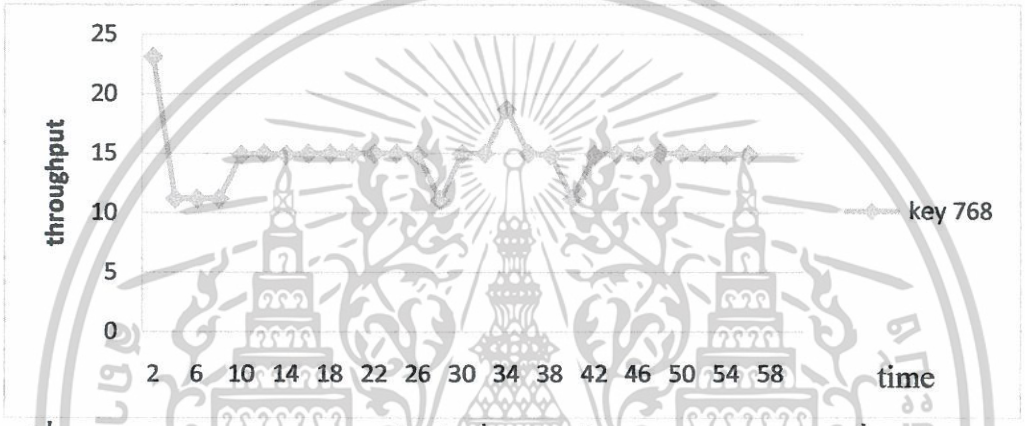
ในการทดลองนั้นมีการจำลองตามรูปที่ 5.9

รูปที่ 5.9 แสดงแบบการจำลองการทดลองที่ 2

โดยมีการทดลองย่อย ดังนี้

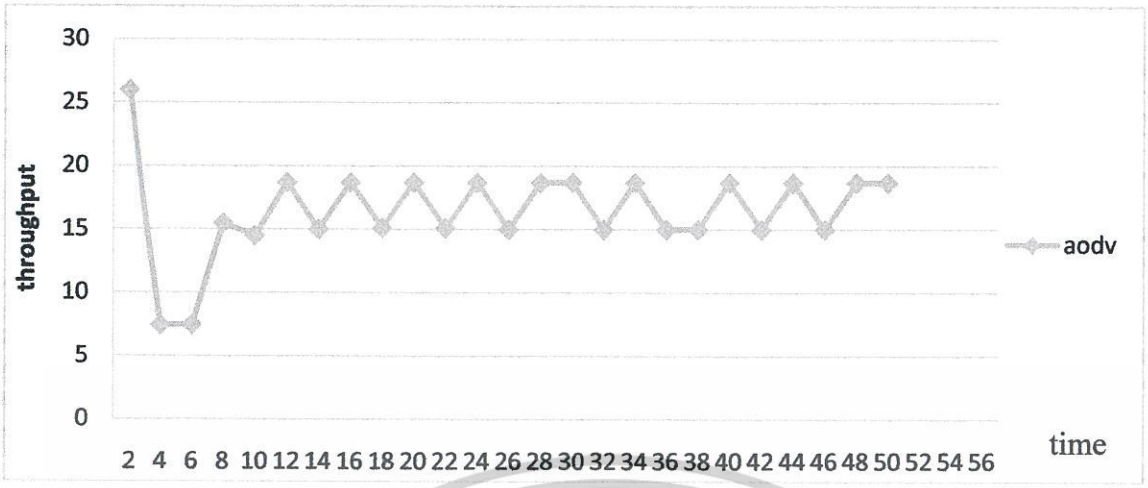
<u>การทดลองที่ 2.1</u> ทดสอบโดยกำหนดปัจจัย คือ

- โหนดห่างกัน 5 เมตร ไม่มีสิ่งกีดขวาง
- กำหนดการส่งแพ็คเกตคือ มีการส่งไปทดสอบทุกๆ 2 วินาที
- ขนาดของ กุญแจเข้ารหัส ที่ 768 และ 1024 bits
- ส่งแพ็คเกตทั้งหมดรวม 100 Mbytes



รูปที่ 5.10 ผลการทดสอบของเออาร์เอเอ็นที่ กุญแจเข้ารหัส = 786 บิต มีค่าเฉลี่ยของทรูพุต = 14.9

รูปที่ 5.11 ผลการทดสอบของเออาร์เอเอ็นที่ กุญแจเข้ารหัส = 1024 บิตมีค่าเฉลี่ยของทรูพุต = 13.9



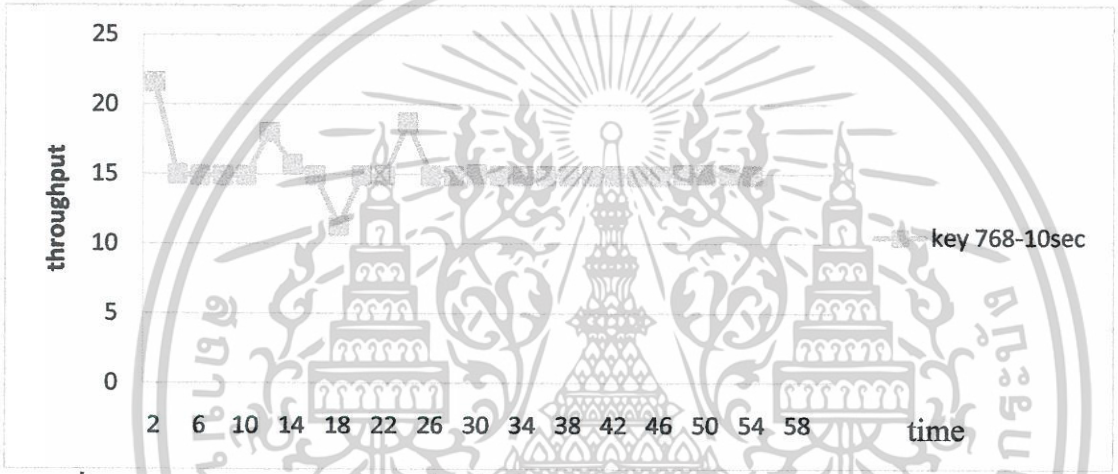
รูปที่ 5.12 ผลการทดสอบของ AODV มีค่าเฉลี่ยของทรูพุต = 16.5

จากการทดลองที่ 2.1 จะพบว่าค่าทรูพุตของโพรโทคอล ดีที่สุด เนื่องจากแพ็กเกตที่เกี่ยวกับการค้นหาเส้นทางไม่ได้มีการเข้ารหัสไว้ ดังนั้นจึงไม่ต้องสูญเสียเวลาในการเข้ารหัส และตรวจสอบ ดังเช่นโพรโทคอลเออาร์เอเอ็น และในส่วนของโพรโทคอลเออาร์เอเอ็นนั้น เมื่อมีกุญแจเข้ารหัสที่ใหญ่ขึ้น ทำให้ค่าของทรูพุตลดลงไปอีกเช่นกัน
ซึ่งจะเห็นได้จากกราฟของผลทดสอบทั้งสามอันเมื่อนำมาวาดร่วมกัน

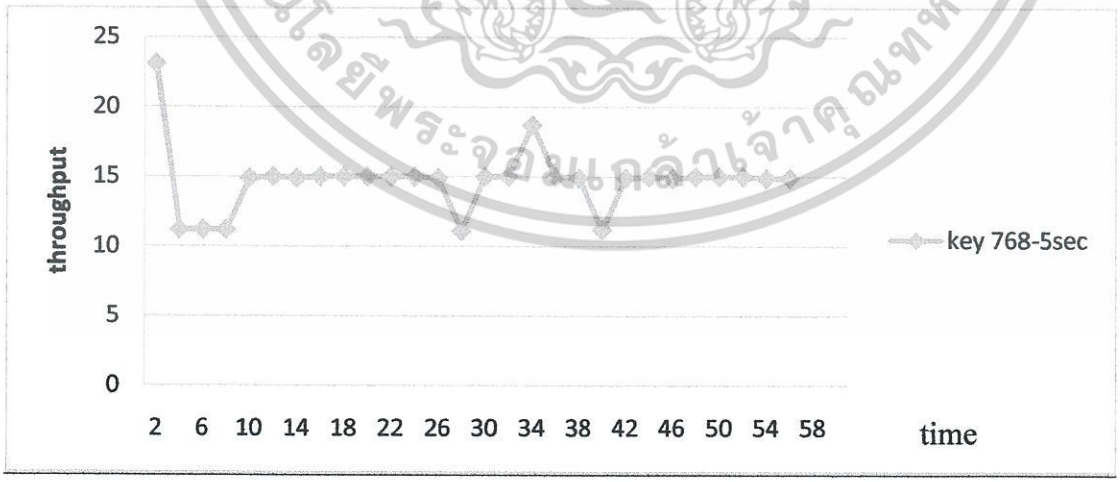
รูปที่ 5.13 ผลการทดสอบเปรียบเทียบ ระหว่างเออาร์เอเอ็นที่ กุญแจเข้ารหัส = 768 และ 1024 กับ AODV

<u>การทดลองที่ 2.2</u> ทดสอบโดยกำหนดปัจจัย คือ

- โนดห่างกัน 5 เมตร ไม่มีสิ่งกีดขวาง
- กำหนดการส่งแพ็คเกตคือ มีการส่งไปทดสอบทุกๆ 1
- ขนาดของ กุญแจเข้ารหัส ที่ 768 บิต
- กำหนดความถี่ในการค้นหาเส้นทางทุกๆ 5 และ10
- ส่งแพ็คเกตทั้งหมดรวม 100 Mbytes

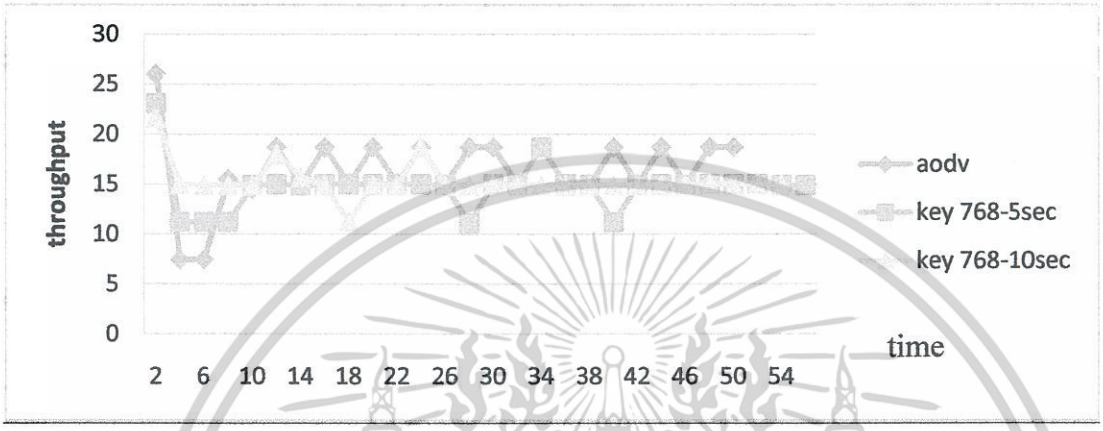


รูปที่ 5.14 ผลการทดสอบของเออาร์เอเอ็นที่ความถี่ในการค้นหาเส้นทาง = 10 วินาที มีค่าเฉลี่ยของทรูพุต = 15.3



รูปที่ 5.15 ผลการทดสอบของเออาร์เอเอ็นที่ความถี่ในการค้นหาเส้นทาง = 5 วินาที มีค่าเฉลี่ยของทรูพุต = 14.9

จากการทดลองที่ 2.2 จะพบว่าเมื่อมีการกำหนดให้ความถี่ในการค้นหาเส้นทาง ถี่น้อยลง ส่งผลให้ค่าทรูพุตที่ได้เพิ่มมากขึ้น เนื่องจากการลดความถี่ในการค้นหาเส้นทาง ทำให้ค่าเสียเวลาของการรักษาความปลอดภัยในการรักษาเส้นทางลดน้อยลงตามไปด้วย
ซึ่งจะเห็นได้จากกราฟของผลทดสอบทั้งสามอันเมื่อนำมาวาดร่วมกัน



รูปที่ 5.16 ผลการทดสอบเปรียบเทียบ ระหว่างเออาร์เอเอ็นที่ความถี่ในการค้นหาเส้นทาง 10, 5 วินาที กับ AODV

จากการทดลอง สามารถสรุปได้ว่า ทั้งปัจจัยในเรื่องของขนาดกุญแจเข้ารหัส และความถี่ในการค้นหาเส้นทาง มีผลกระทบต่อค่าทรูพุต โดยในเรื่องของขนาดกุญแจ เมื่อดีย์มีขนาดเพิ่มขึ้น 256 บิตทำให้ทรูพุตตกลงมากกว่าปัจจัยในเรื่องของความถี่ในการค้นหาเส้นทาง ดังนั้นเมื่อต้องใช้ขนาดกุญแจเข้ารหัสขนาดใหญ่จึงไม่ควรกำหนดให้ความถี่ในการค้นหาเส้นทางบ่อย แต่ถ้าหากใช้กุญแจเข้ารหัสขนาดเล็กคือไม่เกิน 768 บิตก็สามารถกำหนดให้มีความถี่ในการค้นหาเส้นทางบ่อยขึ้นได้

บทที่ 6

บทสรุป

6.1 สรุปผลการทดลอง

จากการศึกษาการค้นหาเส้นทางบนเครือข่ายไร้สายเฉพาะกิจ จะเห็นได้ว่าจุดสำคัญคือความรวดเร็วในการค้นหาเส้นทาง โดยที่ให้ได้เส้นทางนั้นสั้นที่สุดที่จะเป็นไปได้ แต่หากไม่มีความปลอดภัยแล้วเครือข่ายอาจโดนผู้ไม่ประสงค์ดีเข้าโจมตีเพื่อเข้าถึงข้อมูล เปลี่ยนแปลงเส้นทาง หรือทำให้ทรัพยากรของเครือข่ายลดลงได้ ในการทดลองนี้ได้มีการนำโพรโตคอลเออาร์เอเอ็น ซึ่งเป็นโพรโตคอลที่มีความปลอดภัยในการค้นหาเส้นทางบนเครือข่ายไร้สายเฉพาะกิจนี้ โดยโพรโตคอลเออาร์เอเอ็นนั้น มีการพัฒนารูปแบบการค้นหาเส้นทางมาจากโพรโตคอลเอโอดีวี และมีการเพิ่มประสิทธิภาพของความปลอดภัยโดยให้เครื่องที่เป็นหลักเป็นผู้ให้บริการออกใบรับรองออกมา และเครื่องที่เหลือนั้นนำหนังสือรับรอง ไปติดตั้งที่เครื่องของตัวเอง ซึ่งเครื่องหลักและเครื่องที่เหลือต้องมีการไว้วางใจซึ่งกันและกัน (trusted) หากไม่มีการติดตั้งหนังสือรับรอง เครื่องอื่นๆ โดยทั่วไปจะเข้ามาร่วม (join) เครือข่ายไร้สาย ปิงหาได้ แต่ก็เหมือนไม่มีตัวตนในมุมมองของโพรโตคอลเออาร์เอเอ็นเพราะไม่มีหนังสือรับรอง โพรโตคอลเออาร์เอเอ็นจึงไม่ระบุลงในตารางเส้นทาง ส่วนโนดที่มีหนังสือรับรองนั้นโพรโตคอลจะทำการตรวจสอบว่าหนังสือรับรองนั้นถูกต้องหรือไม่ หากถูกต้องก็จะถูกระบุไอพีลงในตารางเส้นทาง หากไม่ถูกต้องก็จะขึ้นฟ้องถึงความผิดพลาด นอกจากนี้โพรโตคอลยังมีการนำโมดูลของอุโมงค์ เข้ามาในการทำเป็นอินเตอร์เฟสเสมือนของอินเตอร์เฟสที่เราใช้ส่งจริง โดยการทำงานของมันคือ เมื่อมีการเริ่มใช้โพรโตคอลอุโมงค์จะถูกเปิดออก และเริ่มค้นหาเส้นทางข้อมูลจะถูกห่อหุ้มด้วยอุโมงค์นี้ก่อน เพื่อลงลายมือชื่อและหนังสือรับรองก่อนจะถูกส่งไปยังอินเตอร์เฟสจริงอีกครั้ง แต่กระบวนการเหล่านี้จะเกิดขึ้นในเฉพาะแพ็กเก็ตที่ค้นหาเส้นทางเท่านั้น ดังนั้นเมื่อมีการพิสูจน์ตัวตนเรียบร้อยแล้ว ทรูพุตของเครือข่ายจะลดลง ทั้งนี้ก็ขึ้นอยู่กับว่าจะขนาดของกุญแจเข้ารหัส ยิ่งใหญ่ก็ยิ่งส่งผลให้ทรูพุตลดลง นอกจากนี้หากมีการเปลี่ยนแปลงเส้นทาง หรือให้มีความถี่ในการส่งแพ็กเกตเพื่อค้นหาเส้นทางมาก ก็จะส่งผลให้ทรูพุตลดลงเช่นกัน

โพรโตคอลเออาร์เอเอ็นนั้นถูกพัฒนาในเรื่องของการค้นหาเส้นทางมาได้ไม่สมบูรณ์เท่าใดนัก เนื่องจากเมื่อมีการเดินทางมากกว่า 1 ฮ็อพจะเกิดการสูญเสียของแพ็กเกตเป็นจำนวนมาก และยังเกิดความผิดพลาดขึ้นในบางครั้ง สำหรับโพรโตคอลเออาร์เอเอ็น ยังต้องมีการพัฒนาในเรื่องของ

การค้นหาเส้นทางต่อไป เนื่องจากโพรโตคอลเออาร์เอเอ็น ณ ปัจจุบันพัฒนามาจากโพรโตคอลเอโอดีวีเวอร์ชั่น 0.5 แต่ ณ ปัจจุบันโพรโตคอลเอโอดีวีพัฒนาไปถึงเวอร์ชั่น 0.9.5 แล้ว ดังนั้นหากนำไปพัฒนาต่อควรปรับปรุงในเรื่องอัลกอริธึมในการค้นหาเส้นทาง และเลือกใช้คีย์ที่มีขนาดไม่เกิน 1024 bits

บรรณานุกรม

C. Barnes, T. Bautts, D. Lloyd, E. Ouellet, J. Posluns, D. M. Zendzian, N. O'Farrell. Syngress. 2002. **Hack Proofing Your Wireless Network**

C. Peikari and S. Fogie. SAMS. 2002.**Wireless Maximum Security**

K. C. Fisher. 2002. **Security Practicum**. [Online] . Available: http://www.arstechnica.com/paedia/w/wireless-security-howto/home-802.11b-1.html

K. Sarinnapakorn . 2001.**IEEE 802.11b High Rate Wireless Local Area Network**. [Online] . Available: http://alpha.fdu.edu/~kanoksri/IEEE80211b.html

M. Maxim and D. Pollino. RSA Press. 2002.**Wireless Security**

R. D. Vines. Wiley Publishing. 2002 .**Wireless Security Essential**

Sanzgiri K, Dahill B, Levine B. N, Shields C. and Royer E. M. **A Secure Routing Protocol for Ad Hoc Networks** [Online]. Available: http://signl.cs.umass.edu/arand/

S. Griffin. 2001. Security and the 802.11b Wireless LAN. [Online]. Available: http://www.sans.org/rr/wireless/80211b.php

Trinity Security Services. 2002. **Wireless LANs**. [Online]. Available: http://www.itsecurity.com/papers/trinity6.htm

T. Macaulay. 2002. **Hardening IEEE 802.11 Wireless Networks**. [Online]. Available: http://www.ewa-canada.com/Papers/Hardening_802.11.pdf

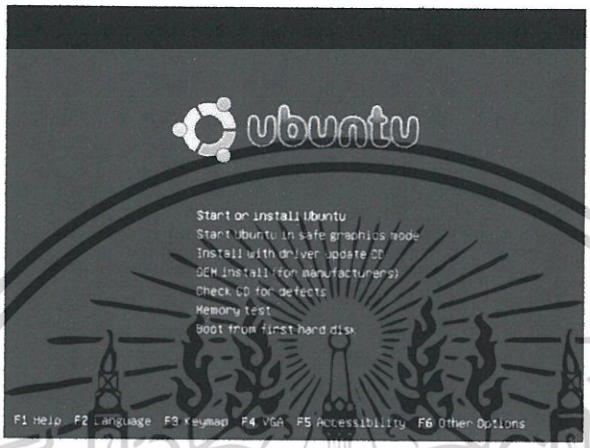
T. A. Dismukes . 2002 .**Wireless Security Black Paper**. [Online]. Available: http://www.arstechnica.com/paedia/w/wireless/security-1.html

ภาคผนวก ก

คู่มือการติดตั้งระบบ

ขั้นตอนการติดตั้ง UBUNTU 7.10

1. ใส่แผ่น CD/DVD และตรวจสอบว่า bios ถูกตั้งค่าให้เริ่มการทำงานที่ CD/DVD ก่อน



รูปที่ ก.1 ขั้นตอนการติดตั้งที่ 1

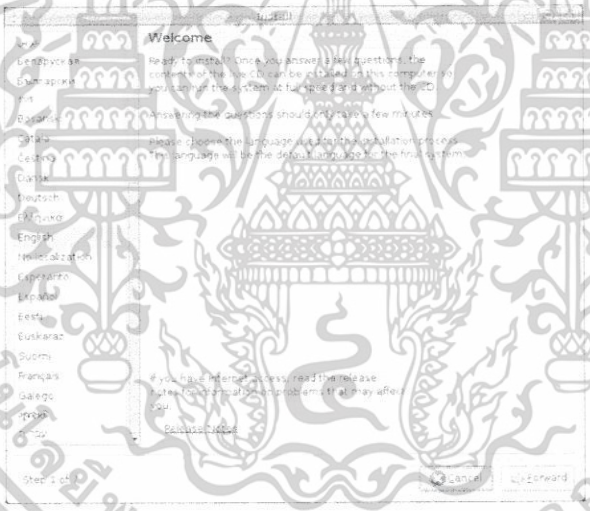
2. เริ่มทำการติดตั้ง

รูปที่ ก.2 ขั้นตอนการติดตั้งที่ 2

3. เข้าสู่หน้า desktop จากนั้นดับเบิ้ลคลิกที่ Install

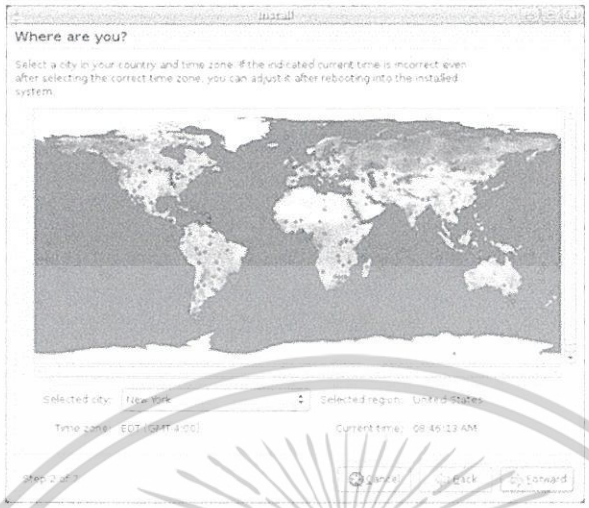
รูปที่ ก.3 ขั้นตอนการติดตั้งที่ 3

4. ทำการเลือกภาษา จากนั้นคลิก Forward



รูปที่ ก.4 ขั้นตอนการติดตั้งที่ 4

5. เลือกโซนเวลา ของประเทศที่เราใช้งาน



รูปที่ ก.5 ขั้นตอนการติดตั้งที่ 5

6. เลือกแบบที่เหมือนกับคีบอร์ด และทดลองพิมพ์ในช่องด้านล่าง จากนั้นคลิกที่ Forward

รูปที่ ก.6 ขั้นตอนการติดตั้งที่ 6

7. เลือกแบ่ง Partition ตามความต้องการ จากนั้นคลิกที่ Forward

รูปที่ ก.7 ขั้นตอนการติดตั้งที่ 7

8. ถ้าทำการเลือกที่ Manual จะปรากฏหน้าต่างขึ้น ในที่นี้สามารถสร้าง แก้ไข หรือลบ Partition ได้

รูปที่ ก.8 ขั้นตอนการติดตั้งที่ 8

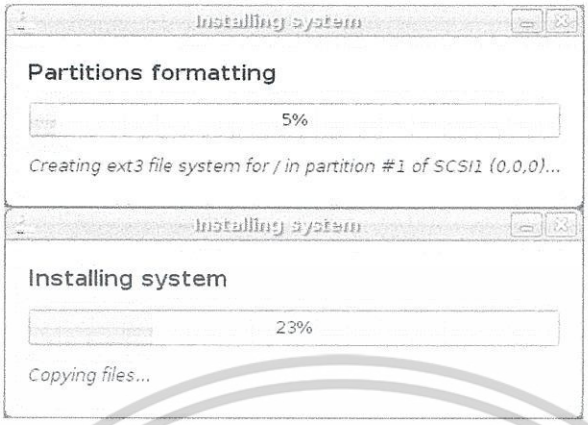
9. กรอกข้อมูล คือ ชื่อ, ชื่อที่ทำการล๊อกอิน, พาสเวิร์ด ให้ครบถ้วน จากนั้นคลิกที่ Forward

รูปที่ ก.9 ขั้นตอนการติดตั้งที่ 9

10. คลิก Install เพื่อเริ่มการติดตั้ง

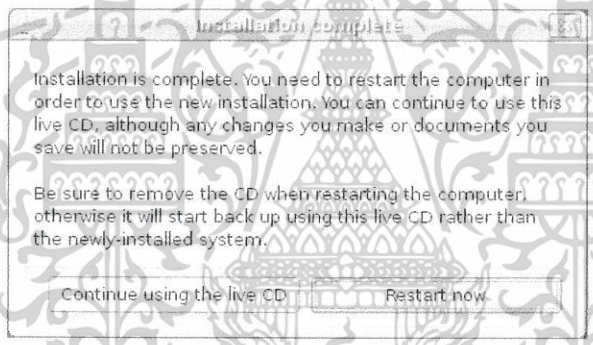
รูปที่ ก.10 ขั้นตอนการติดตั้งที่ 10

11. ระบบกำลังเริ่มทำการติดตั้ง



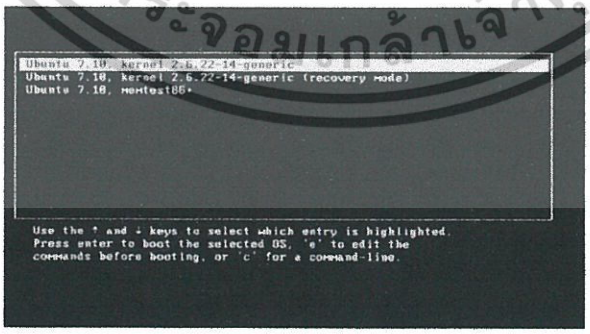
รูปที่ ก.11 ขั้นตอนการติดตั้งที่ 11

12. คลิกที่ Restart now



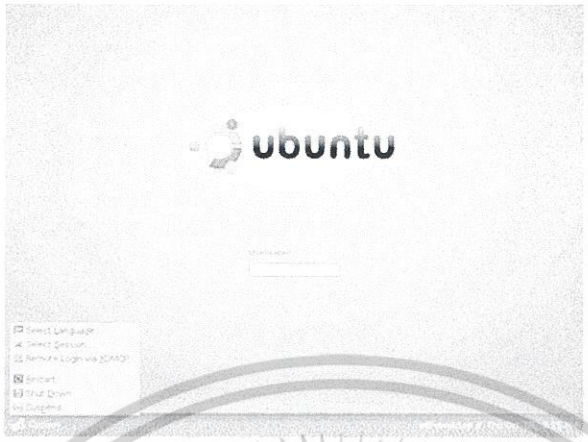
รูปที่ ก.12 ขั้นตอนการติดตั้งที่ 12

13. เลือก kernel ของ Ubuntu ที่ได้ทำการติดตั้งไว้



รูปที่ ก.13 ขั้นตอนการติดตั้งที่ 13

14. เข้าสู่หน้า Log in โดยระบบจะให้กรอก Username และ Password



รูปที่ ก.14 ขั้นตอนการติดตั้งที่ 14

15. เข้าสู่หน้าต่าง เพื่อเริ่มการใช้งาน

รูปที่ ก.15 ขั้นตอนการติดตั้งที่ 15

ภาคผนวก ข

รหัสคำสั่งที่ใช้ในการหาค่าเราติ้งโอเวอร์เฮด

รหัสคำสั่งตัวอย่างของโมดูลที่เพิ่มลงไปในโพรโทคอลส่วนของการจับเวลา เนื่องจากส่วนที่ใช้จริงได้ใส่รวมกับรหัสคำสั่งของโพรโทคอลไปแล้ว

ข.1 ส่วนของการเริ่มจับเวลา

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

#define SIZE 256

int main (void)
{
        time_t curtime;
        struct tm *loctime;
        struct timeval tv;
        struct timezone tz;
        long millisecValStart;
        long millisecValStop;

        long timeValStart;
        long timeValStop;

        long deltaMill;
        long deltaSec;

        long repreHOUR;
        long repreMIN;
        long repreSEC;
```

```
    FILE  *fp;
    char wordsStart[SIZE];
    char wordsEnd[SIZE];
    char wordsDelta[SIZE];
    char wordsMillStart[SIZE];
    char wordsMillStop[SIZE];


    // BEGIN BLOCK STOPWATCH START ***********************************//

/**/
    /**/ gettimeofday(&tv, &tz);
    /**/ curtime = time (NULL);
    /**/ loctime = localtime (&curtime);
/**/
    /**/ millisecValStart = tv.tv_usec/1000;
    /**/ timeValStart = tv.tv_sec;
    /**/
    /**/ strftime (wordsStart, SIZE, "%d/%m/%y %H:%M:%S:", loctime);
    /**/ fputs (wordsStart, stdout);
    /**/ printf("%d\n",millisecValStart);
/**/
    // END BLOCK STOPWATCH START *************************************//


    // WRITE TO TEMP ************************************************//
    /**/
    /**/      long tempValue[2];
    /**/      tempValue[0] = timeValStart;
    /**/      tempValue[1] = millisecValStart;
    /**/      int i;
    /**/
    /**/      if((fp=fopen("temp", "wb"))==NULL) {
```

```
/**/          printf("Cannot open file.\n");
/**/      }
/**/
/**/      if(fwrite(tempValue, sizeof(long), 2, fp) != 2)
/**/        printf("File read error.");
/**/                fclose(fp);
/**/
// WRITE TO TEMP ************************************************//


// Write A File
**********************************************************************************
*****************//
/**/ sprintf(wordsMillStart, "%d", millisecValStart);
                        //
/**/
                        //
/**/      if((fp = fopen("protocol.log","a+")) == NULL)
                        //
/**/      {
                        //
/**/                fprintf(stderr, "Can't open \"temp\" file. \n");
                    //
/**/                exit(1);
                    //
/**/      }
                            //
/**/                fprintf(fp, "%s%s", wordsStart,wordsMillStart);    //
/**/                rewind(fp);                         /* go back to beginning of file */
                            //
/**/                fclose(fp);
                            //
```

```
        /**/
                                //
        // Write A File
**********************************************************************************
******************//


        return 0;
}
```

ข.2 ส่วนของการหยุดจับเวลาและคำนวณ

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

#define SIZE 256
int main(void)
{

        time_t curtime;
        struct tm *loctime;
        struct timeval tv;
        struct timezone tz;
        long millisecValStart;
        long millisecValStop;

        long timeValStart;
        long timeValStop;

        long deltaMill;
        long deltaSec;

        long repreHOUR;
        long repreMIN;
        long repreSEC;

        FILE  *fp;
        char wordsStart[SIZE];
        char wordsEnd[SIZE];
        char wordsDelta[SIZE];
```

```
char wordsMillStart[SIZE];
char wordsMillStop[SIZE];

long tempValue[2];

// READ TEMP FILE ************************************************//
/**/      if((fp=fopen("temp", "rb"))==NULL) {
/**/       printf("Cannot open file.\n");
/**/      }
/**/
/**/       if(fread(tempValue, sizeof(long), 2, fp) != 2) {
/**/              if(feof(fp))
/**/                     printf(".");
/**/              else
/**/                     printf("File read error.");
/**/      }
/**/              fclose(fp);
/**/
/**/
/**/              timeValStart =  tempValue[0];
/**/              millisecValStart =  tempValue[1];
/**/
// READ TEMP FILE ************************************************//

// BEGIN BLOCK STOPWATCH STOP ***********************************//

/**/                                    //
    /**/ gettimeofday(&tv, &tz);                                    //
    /**/ curtime = time (NULL);                                     //
    /**/ loctime = localtime (&curtime);                            //
    /**/                                                            //
    /**/ millisecValStop = tv.tv_usec/1000;                         //
```

```
    /**/ timeValStop = tv.tv_sec;                    //
    /**/                                             //
    /**/ strftime (wordsEnd, SIZE, "%d/%m/%y %H:%M:%S:", loctime);  //
    /**/ fputs (wordsEnd, stdout);                   //
    /**/ printf("%d",millisecValStop);               //
/**/                                                 //
    // END BLOCK STOPWATCH STOP ***********************************//

    // Delta Time Calculation Block ***********************************//
    /**/                                             //
    /**/ if(millisecValStop < millisecValStart){      //
/**/        deltaMill = 1000 - millisecValStart;                    //
    /**/    deltaMill = deltaMill + millisecValStop;    //
    /**/    }else{                                      //
    /**/    deltaMill = millisecValStop - millisecValStart;}    //
    /**/ deltaSec = timeValStop - timeValStart;         //
    /**/ repreHOUR = deltaSec / 3600;                   //
    /**/ repreMIN = (deltaSec % 3600) / 60;             //
    /**/ repreSEC = (deltaSec % 3600) % 60;             //
    /**/ printf(" [%dh/%dm/%ds/", repreHOUR, repreMIN, repreSEC); //
    /**/ printf("%dms]\n", deltaMill);                  //
    /**/                                               //
    // Delta Time Calculation Block ***********************************//

    // Write A File
*****************************************************************************
*****************//
    /**/ sprintf(wordsDelta, "[%dh/%dm/%ds/%dms]\n", repreHOUR, repreMIN, repreSEC,
deltaMill);                  //
    /**/ sprintf(wordsMillStop, "%d", millisecValStop);
                             //
```

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวปานแก้ว รัตนสิริภัทร
วัน เดือน ปีเกิด	14 มิถุนายน 2530
ที่อยู่	198/38 ม.12 ม.มัณฑนาบางนา ถ.บางนา-ตราด ต.บางพลีใหญ่
	อ. บางพลี จ.สมุทรปราการ 10540
โทร	08-4072-4452
ประวัติการศึกษา	
มัธยมศึกษาตอนต้น	โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า
มัธยมศึกษาตอนปลาย	โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า
อุดมศึกษา	คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ-นามสกุล	นางสาวสิบมนัส เชิดเกียรติศักดิ์
วัน เดือน ปีเกิด	5 กันยายน 2530
ที่อยู่	37 ม.เมืองทอง2/1 ซ.บางปะกง3 ถ.พัฒนาการ แขวง/เขตประเวศ
	กรุงเทพฯ 10250
โทร	08-6521-0432
ประวัติการศึกษา	
มัธยมศึกษาตอนต้น	โรงเรียนเตรียมอุดมศึกษาพัฒนาการ
มัธยมศึกษาตอนปลาย	โรงเรียนเตรียมอุดมศึกษาพัฒนาการ
อุดมศึกษา	คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง