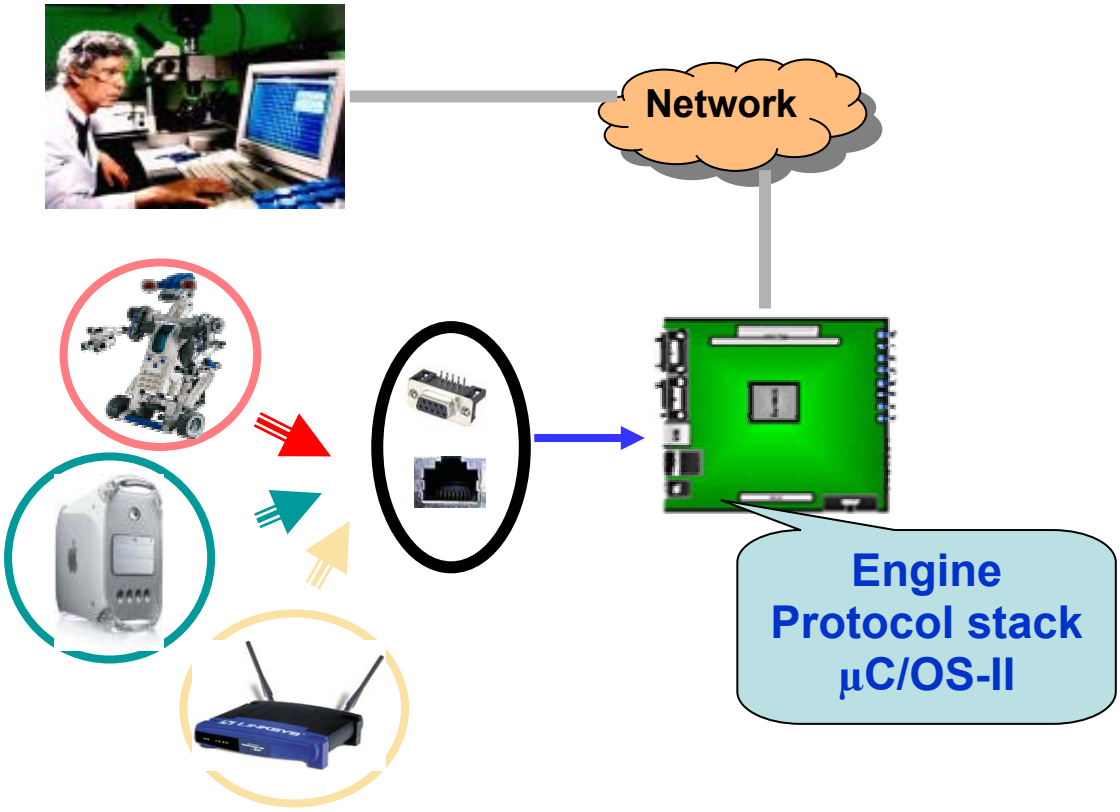
บทที่ 4

การออกแบบและพัฒนา

4.1 ขอบเขตของโครงงาน



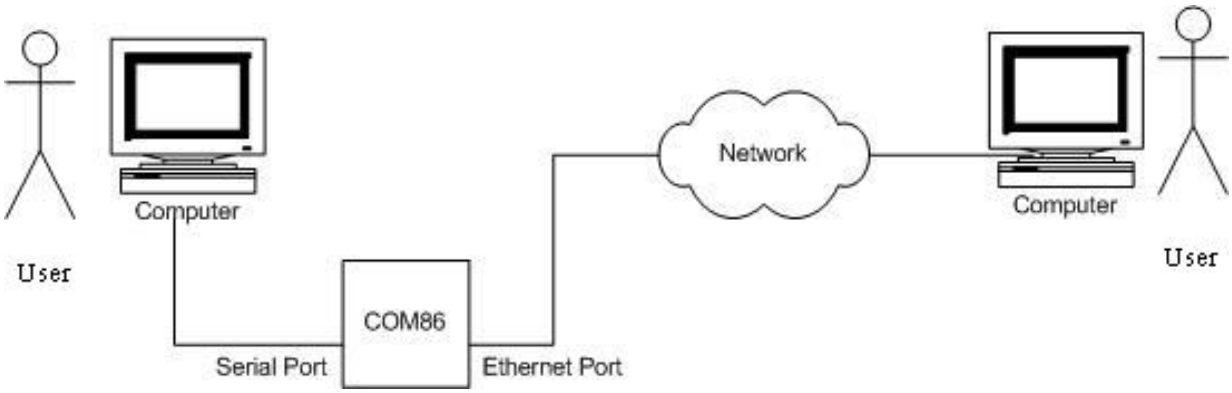
รูปที่ 4-1 ขอบเขตของโครงงาน

นอกจากการใช้ lwIP (Light weight TCP/IP protocol stack) ในการพัฒนาแล้วการทำงานยังอยู่ภายใต้ความสามารถของชุดพัฒนาคอม 86 ซึ่งสำหรับการทำงานของระบบนั้นมีส่วนที่สามารถติดต่อได้หลายส่วน และได้แบ่งออกเป็น

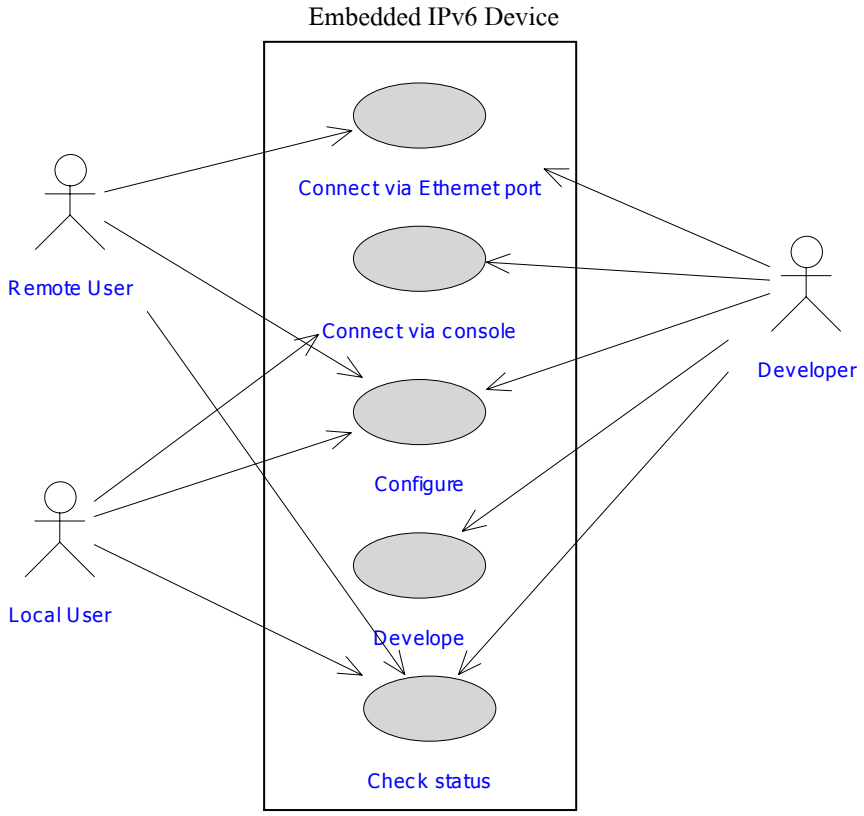
1. ส่วนคอนโซลพอร์ตอนุกรม ซึ่งผู้ใช้สามารถติดต่อกับอุปกรณ์จากเครื่องคอมพิวเตอร์ผ่านโปรแกรมไฮเปอร์เทอร์มินัลได้
2. ส่วนอีเทอร์เน็ทพอร์ต โดยผู้ใช้สามารถติดต่อกับอุปกรณ์ผ่านทางระบบเครือข่ายได้
3. ส่วนพอร์ตอื่นๆ ของชุดพัฒนาคอม 86 ผู้ใช้สามารถที่จะเพิ่มเติมส่วนต่างๆ ของระบบเข้าไปได้เอง

4.2 โครงสร้างและองค์ประกอบของระบบ

โครงสร้างและองค์ประกอบของระบบโดยรวมที่ใช้งานรวมทั้งลักษณะการเชื่อมต่อแสดงดังรูปที่ 4-2 และรูปที่ 4-3



รูปที่ 4-2 ภาพรวมของระบบและการเชื่อมต่อ

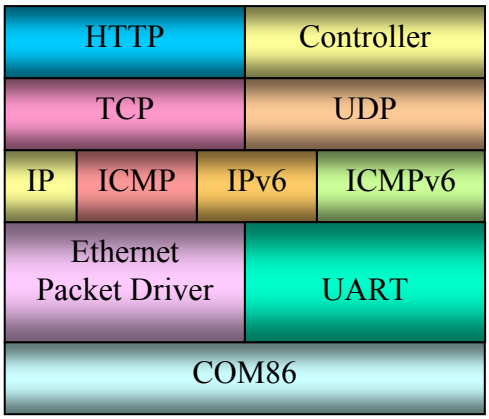


รูปที่ 4-3 แสดงการออกแบบลักษณะการใช้งานอุปกรณ์

ลักษณะการใช้งานอุปกรณ์นี้นั้นสามารถใช้งานได้ทั้งจากระยะไกลและในระยะใกล้ คือการเชื่อมต่อเพื่อใช้งานในระยะไกลนั้นสามารถเชื่อมต่อได้ผ่านระบบเครือข่ายเข้ามาทางอีเทอร์เน็ทพอร์ต ด้วย

สายเคเบิลอาร์เจ 45 (RJ-45 cable) ส่วนการเชื่อมต่อเพื่อใช้งานในระยะใกล้นั้นสามารถกระทำได้โดยผ่านทางคอนโซลพอร์ตอนุกรมโดยการใช้โปรแกรมไฮเปอร์เทอร์มินัลด้วยสายซีเรียลอาร์เอส-232 โดยผู้ใช้สามารถเข้าไปกำหนดค่าคุณสมบัติต่างๆอุปกรณ์ได้จากทั้ง 2 ทางคือทางเวบเพจหรือทางไฮเปอร์เทอมินัลก็ได้ และนักพัฒนาสามารถใช้อุปกรณ์นี้เป็นอุปกรณ์ต้นแบบในการพัฒนาเพื่อทำงานอื่นต่อไปได้

4.2.1 โครงสร้างของสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

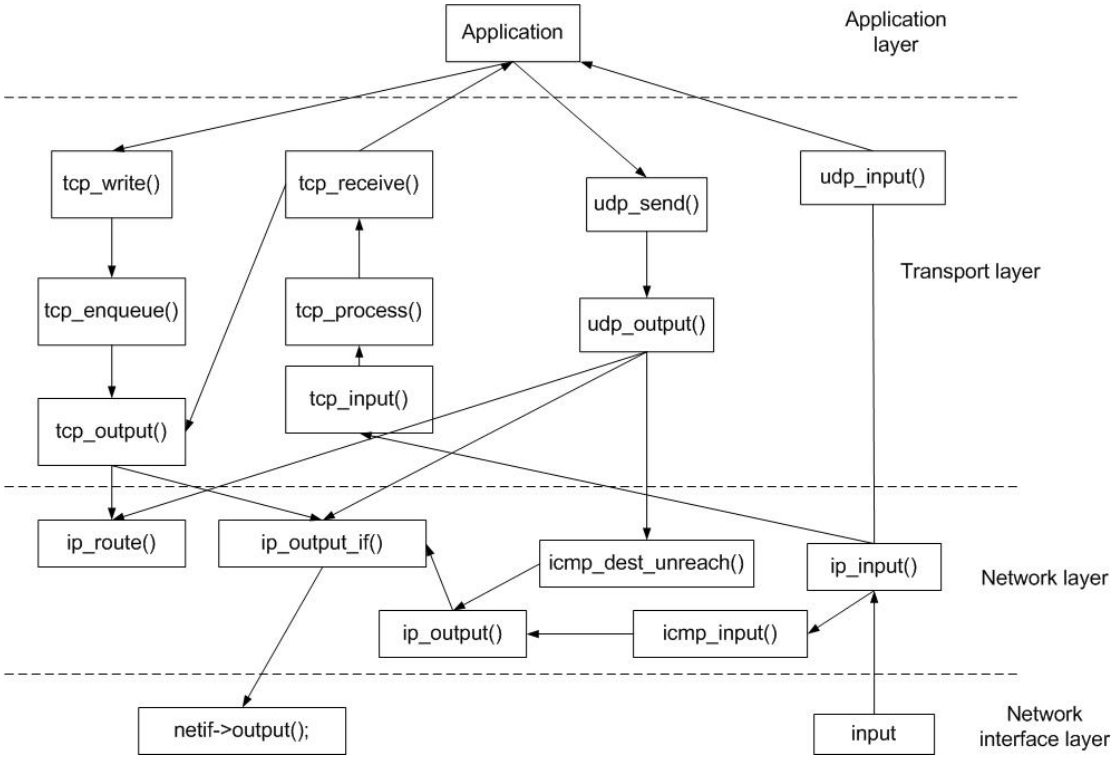


รูปที่ 4-4 แสดงโครงสร้างสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

จากรูปที่ 4-4 โครงสร้างสถาปัตยกรรมของระบบสามารถเทียบกับแบบอ้างอิงโอเอสไอได้ดังนี้

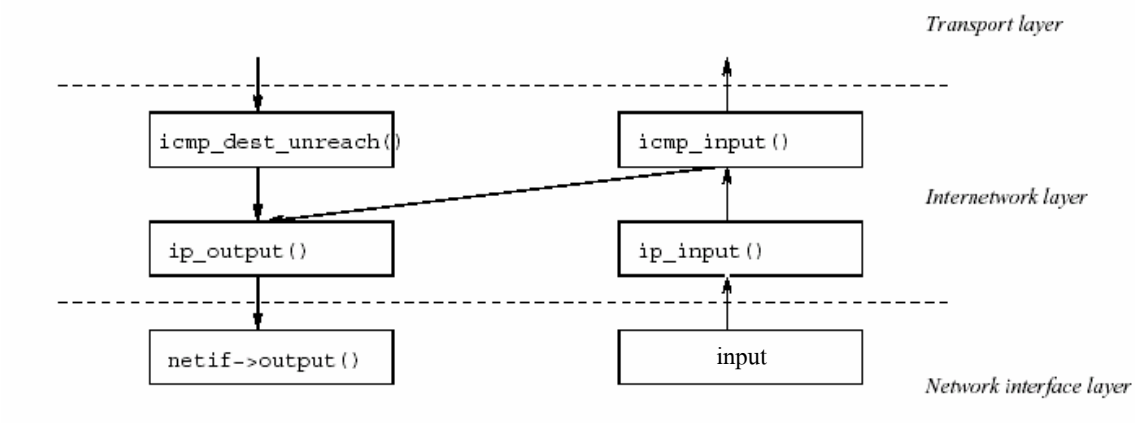
ชั้นฟิสิคัล	ชุดพัฒนาคอม 86 รวมทั้งสายเชื่อมต่อแบบต่างๆ
ชั้นเดทาลิงค์	อีเทอร์เน็ทแพ็กเก็ตไดร์เวอร์และยูอาร์ต
ชั้นเน็ตเวิร์ค	ไอพี, ไอซีเอ็มพี, ไอพีรุ่น 6, ไอซีเอ็มพีรุ่น 6
ชั้นทรานสพอร์ต	ทีซีพี, ยูดีพี
ชั้นแอพพลิเคชัน	เอชทีทีพีและส่วนที่ทำการควบคุมการทำงานของระบบ

จากรูปที่ 4-5 เป็นการแสดงโครงสร้างการทำงานของระบบระหว่างเลเยอร์เมื่อมีแพ็กเก็ตเข้ามาทางอีเทอร์เน็ทพอร์ต และการส่งข้อมูลข้ามเลเยอร์กัน



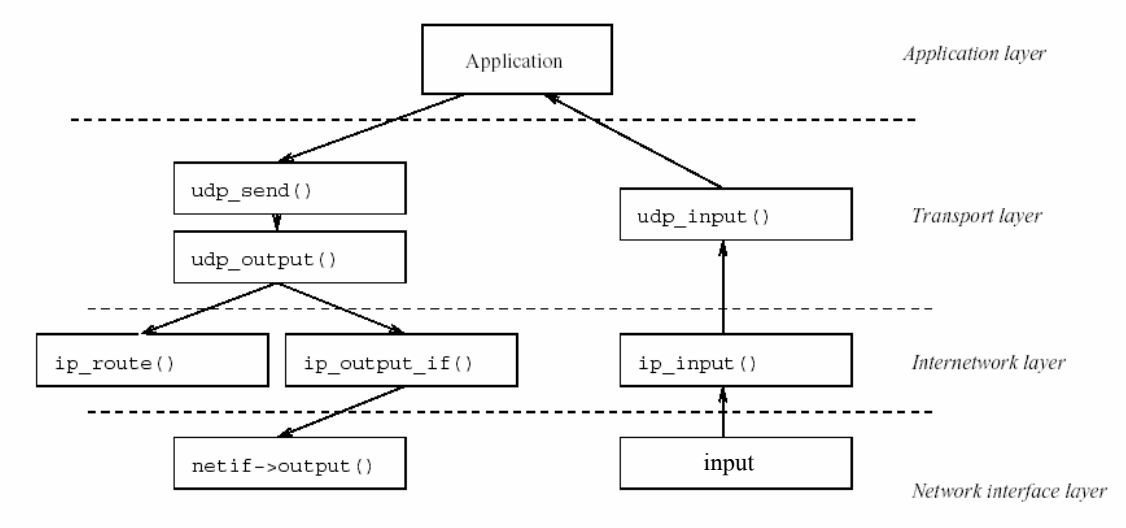
รูปที่ 4-5 แสดงโครงสร้างการทำงานระหว่างเลเยอร์

ซึ่งสามารถแยกรายละเอียดได้ดังนี้



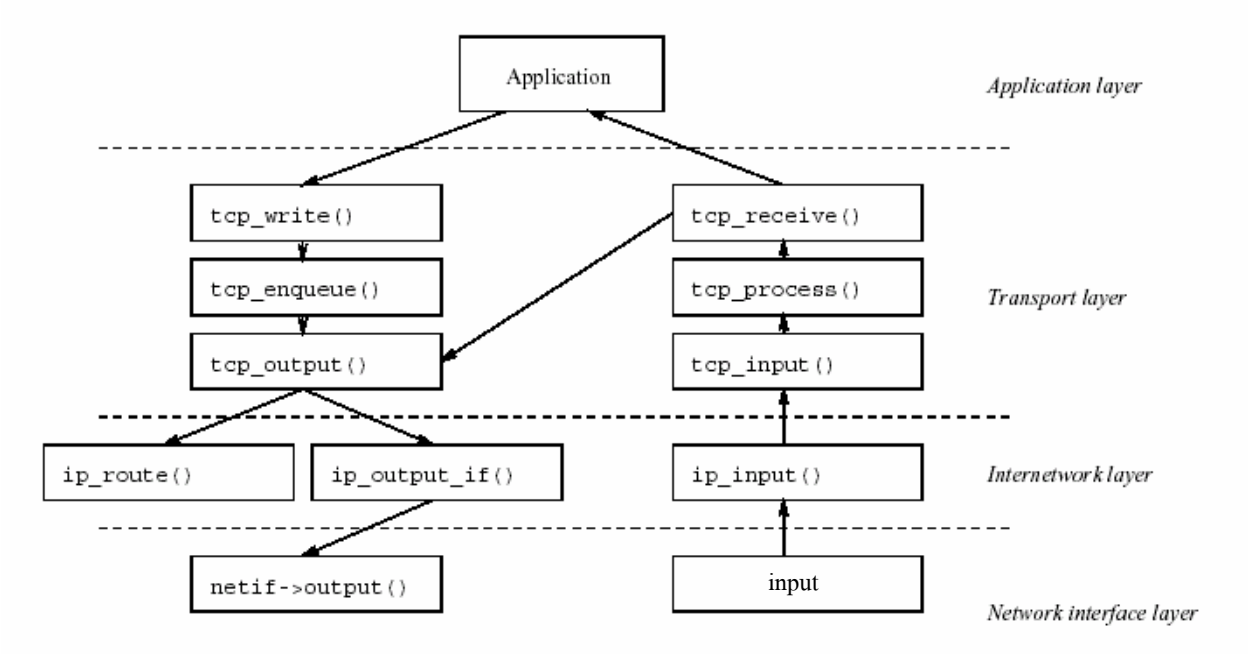
รูปที่ 4-6 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของไอซีเอ็มพี

เมื่อมีไอซีเอ็มพีแพ็กเก็ตเข้ามาที่ ip_input() ก็จะถูกส่งเข้าไปที่ icmp_input() ซึ่งเป็นส่วนที่ใช้ติดต่อกันระหว่างไอซีเอ็มพีและไอพี เมื่อมีการกระทำการกับไอซีเอ็มพีแพ็กเก็ตแล้วก็จะรู้ว่าต้องทำอะไรต่อไปหรือส่งให้ใคร เช่น โปรโตคอลยูดีพีจะมีการใช้ icmp_dest_unreach() ในการส่งเดสทิเนชันอันรีชเอเบิลแมสเซสแล้วก็จะส่งออกทาง ip_output() ต่อไป เป็นต้น



รูปที่ 4-7 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของยูดีพี

ในการส่งข้อมูลโปรแกรมประยุกต์ก็จะเรียกใช้ udp_send() ซึ่งจะไปเรียก udp_output() อีกทีเพื่อคำนวณเช็คซัมและเติมเฮดเดอร์เข้าไป อาจจะมีเรียก ip_route() ขึ้นมาเพื่อเลือกเน็ทเวิร์คอินเทอร์เฟสที่เหมาะสมแล้วจึงเรียก ip_output_if() เพื่อที่จะทำการส่งต่อไป



รูปที่ 4-8 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของทีซีพี

เมื่อโปรแกรมประยุกต์มีความต้องการที่จะส่งข้อมูลแบบทีซีพีก็จะมีการเรียกใช้งาน tcp_write() ซึ่งจะไปเรียก tcp_enqueue() ขึ้นมาทำงาน โดยจะแบ่งข้อมูลออกเป็นทีซีพีเซกเมนต์ที่มีขนาดที่เหมาะสมแล้วเอาไปเข้าคิวสำหรับการติดต่อสื่อสาร tcp_output() ก็จะตรวจสอบความเป็นไปได้ที่จะส่งข้อมูล เช่น

ขนาดวินด์โดว์ของฝั่งผู้รับ แล้วถึงจะส่งข้อมูลโดยเรียกใช้ ip_route() และ ip_output_if() ขึ้นมาทำงานต่อไป

เมื่อมีข้อมูลเข้ามา ip_input() ก็จะตรวจสอบเฮดเดอร์ของไอพีก่อนแล้วถึงจะส่งผ่านไปที่ tcp_input() ซึ่งจะตรวจสอบเกี่ยวกับเช็คซัมและออพชัน รวมทั้งตรวจสอบก่อนว่าเป็นของคอนเนคชันไหนก่อนที่จะส่งผ่านไปยัง tcp_process() ซึ่งทำงานตามทีซีพีสเตทแมชชีนและสเตททรานสิชันที่จำเป็นใดๆ แล้ว tcp_receive() ก็จะถูกเรียกเพื่อที่จะส่งต่อข้อมูลไปให้กับโปรแกรมประยุกต์ที่ถูกต้อง จากนั้นก็จะต้องมีการส่ง ACK กลับซึ่งจะมีการเรียกใช้งาน tcp_output() ขึ้นมาทำงานนี้

4.2.2 องค์ประกอบของระบบ

1. ระบบปฏิบัติการ

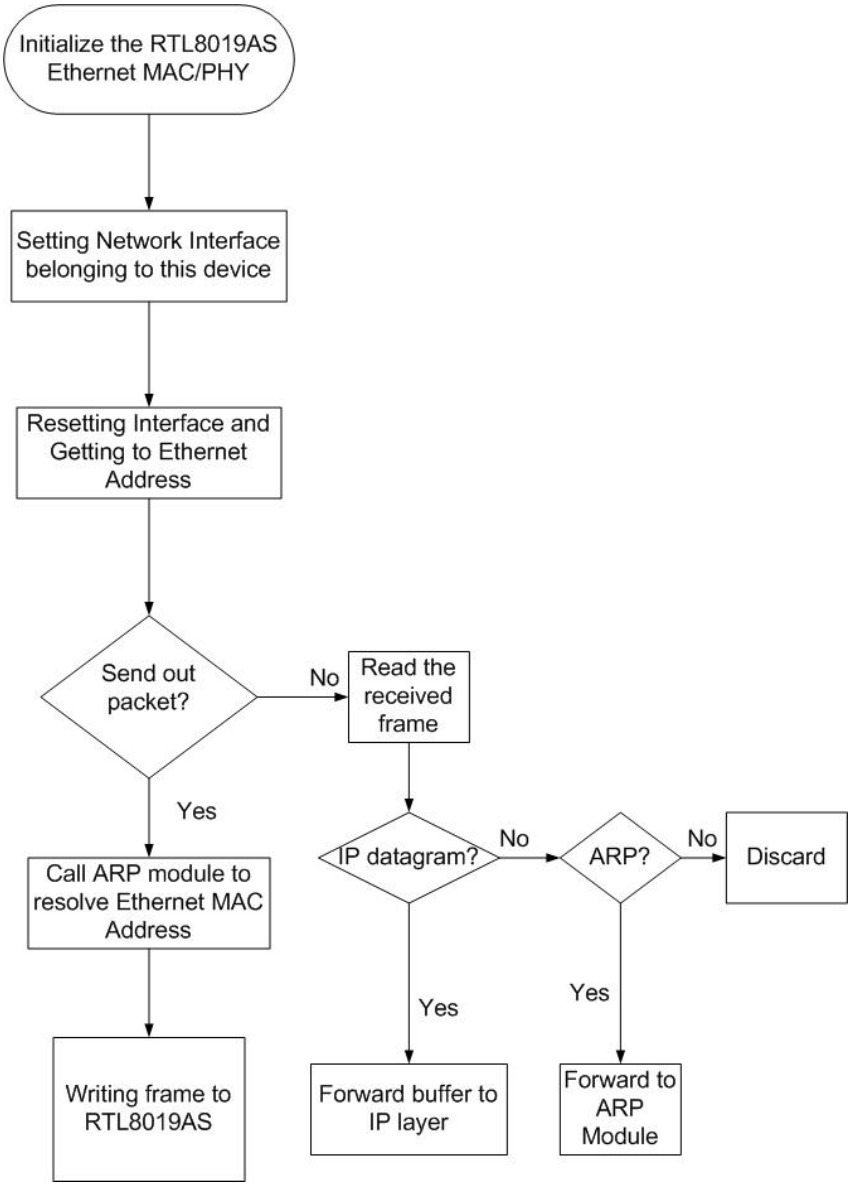
เนื่องจากการทำงานของระบบต้องสามารถทำงานแบบมัลติทาสก์ได้ เพื่อที่จะสามารถทำงานหลายๆ อย่างพร้อมกันได้และสามารถตอบสนองต่อการทำงานได้อย่างรวดเร็ว ระบบจึงจำเป็นที่จะต้องใช้ระบบปฏิบัติการที่สนับสนุนการทำงานแบบมัลติทาสก์ด้วย ซึ่งผู้พัฒนาได้เลือกใช้ uC/OS-II ซึ่งได้กล่าวถึงแล้วในบทที่ 3

2. โปรโตคอลแสต็ก

ในการสื่อสารในระบบเครือข่ายนั้นจำเป็นต้องอาศัยการทำงานของโปรโตคอลแสต็ก ซึ่งโปรโตคอลแสต็กนั้นต้องเป็นโปรโตคอลแสต็กที่เหมาะสมกับการทำงานบนระบบฝังตัวด้วย เพราะระบบฝังตัวนั้นมีหน่วยความจำน้อย ซึ่งผู้พัฒนาได้เลือกใช้ไลท์เวทโปรโตคอลแสต็ก ซึ่งได้กล่าวถึงแล้วในบทที่ 3

3. ดีไวซ์ไดรเวอร์

การทำงานร่วมกับอุปกรณ์ต่างๆนั้นต้องมีการทำงานของไดรเวอร์เข้ามาเกี่ยวข้องด้วย ซึ่งชุดพัฒนาคอม 86 นั้นใช้ RTL8019AS ในการทำงานเกี่ยวการรับส่งข้อมูลผ่านทางอีเทอร์เน็ตพอร์ต ดังนั้นการที่จะให้อุปกรณ์ทำงานกับโปรโตคอลแสต็กได้อย่างถูกต้องต้องมีการแก้ไขไดรเวอร์หรือเขียนไดรเวอร์ขึ้นมาใหม่ เพื่อให้ทำงานร่วมกันได้ โดยสามารถแสดงลักษณะการทำงานของไดรเวอร์ได้ดังนี้



รูปที่ 4-9 แสดงการทำงานของดีไวซ์ไดรเวอร์

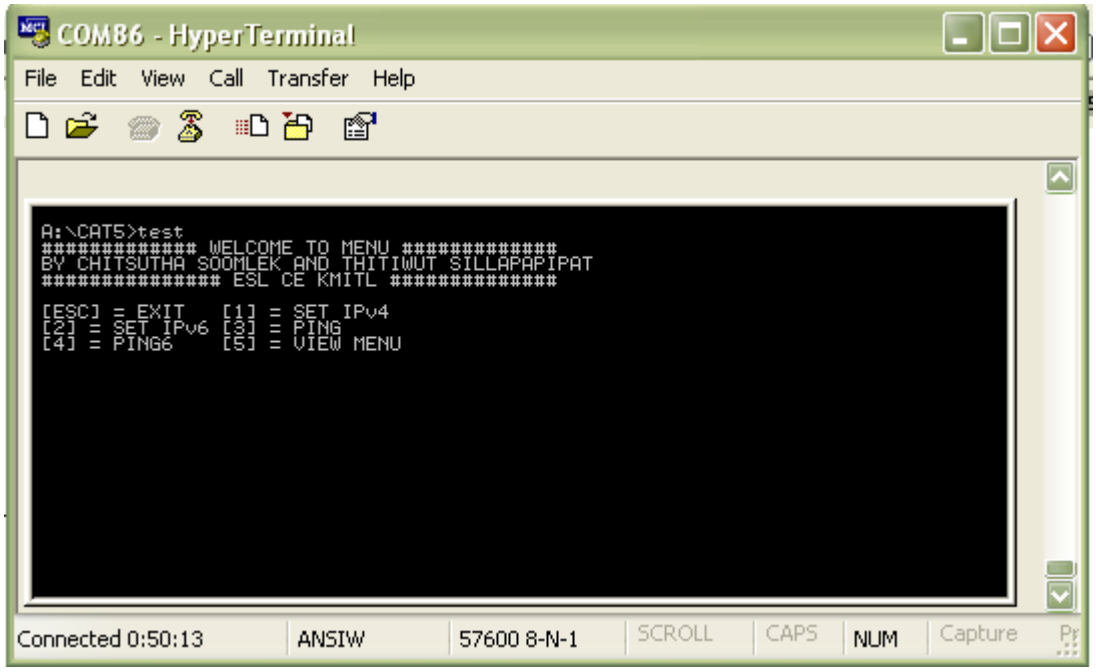
4. เอ็นจิน

การทำงานของเอ็นจินที่ควบคุมระบบมีส่วนประกอบหลักๆคือ ควบคุมการทำงานและการสื่อสารผ่านทางคอนโซลพอร์ตอนุกรม อีเทอร์เน็ทพอร์ต การทำงานกับระบบปฏิบัติการและโปรโตคอลแสต็ก ให้ทั้งระบบสามารถทำร่วมกันได้อย่างถูกต้องเหมาะสม

5. ส่วนที่ติดต่อกับผู้ใช้ (User Interface)

ส่วนที่ติดต่อกับผู้ใช้นั้นแบ่งออกเป็น 2 ส่วนด้วยกัน เนื่องจากผู้ใช้สามารถติดต่อกับระบบได้ผ่านทั้งทางคอนโซลพอร์ตอนุกรมและอีเทอร์เน็ทพอร์ต จึงมีการสร้างส่วนที่ติดต่อกับผู้ใช้ 2 แบบ ดังนี้

(1) ส่วนที่ติดต่อกับผู้ใช้ทางคอนโซลพอร์ตอนุกรม



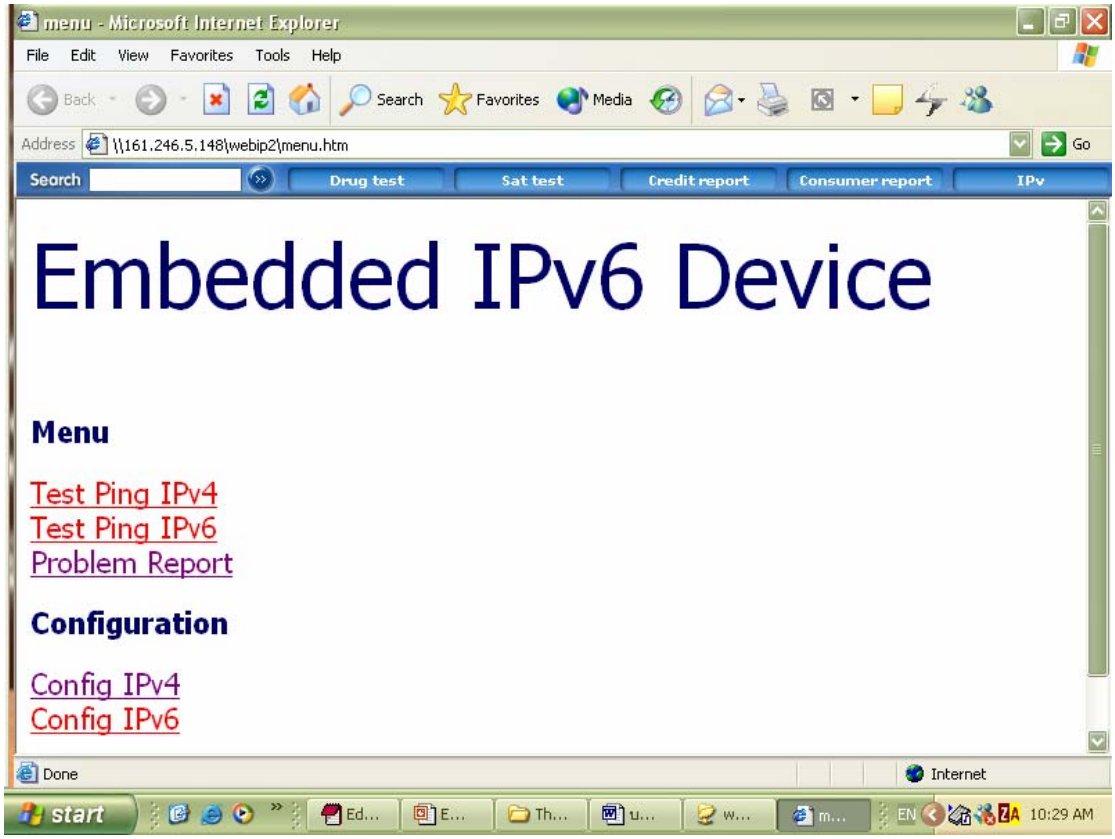
รูปที่ 4-10 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม

โดยผู้ใช้ที่ทำการติดต่อกับอุปกรณ์ผ่านทางคอนโซลพอร์ตอนุกรมนั้นจะต้องติดต่อผ่านเครื่องคอมพิวเตอร์ส่วนบุคคลที่เชื่อมต่อกับอุปกรณ์นี้อยู่ผ่านโปรแกรมเทอร์มินัล ซึ่งส่วนที่ติดต่อกับผู้ใช้นั้นแสดงดังภาพที่ 4-10 จะเห็นได้ว่าผู้ใช้สามารถเรียกใช้บริการจากเมนูที่ผู้พัฒนาได้กำหนดไว้ให้แล้ว

(2) ส่วนที่ติดต่อกับผู้ใช้ทางอีเทอร์เน็ท

ผู้ใช้ที่ทำการติดต่อกับอุปกรณ์ผ่านทางอีเทอร์เน็ทนั้นสามารถติดต่อสื่อสารจะที่ไหนก็ได้ในระบบเครือข่าย ดังภาพที่ 4-11 ผู้ใช้สามารถติดต่อขอใช้บริการโดยใช้โปรแกรมบราวเซอร์ทั่วไปเรียกส่วนที่ติดต่อกับผู้ใช้ขึ้นมาทำงาน

เนื่องจากข้อจำกัดในเรื่องของหน่วยความจำของตัวชุดพัฒนาคอม 86 และความเร็วในการประมวลผลจึงต้องทำส่วนที่ติดต่อกับผู้ใช้ออกมาในรูปแบบของเท็กซ์เบส (Text base) ดังภาพ



รูปที่ 4-11 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางอีเทอร์เน็ท

4.3 การพัฒนาไอพีรุ่น 6 บนระบบฝังตัว

เนื่องจากระบบฝังตัวที่ไม่ใช่พีซีนั้นมีข้อจำกัดในหลายๆด้านด้วยกัน ไม่ว่าจะเป็นเรื่องของหน่วยความจำ และการประมวลผล ซึ่งมีความแตกต่างจากคอมพิวเตอร์ส่วนบุคคลอยู่พอสมควร การที่จะพัฒนาไอพีรุ่น 6 ทั้งหมดตามที่มาตรฐานกำหนดนั้นเป็นเรื่องยากมาก ซึ่งผู้วิจัยได้เล็งเห็นว่าการพัฒนาอุปกรณ์ฝังตัวที่เป็นอุปกรณ์ปลายทาง (Terminal device) ซึ่งมีความสามารถเฉพาะงานนั้นๆและเป็นโฮสต์ไม่ใช่เราเตอร์ให้สามารถที่จะทำงานบนเครือข่ายไอพีรุ่น 6 นั้น ควรมีการพัฒนาสิ่งต่างๆ ดังนี้

4.3.1 การปรับค่าต่างๆของฮาร์ดแวร์

การปรับค่าต่างๆของฮาร์ดแวร์ (Hardware configuration) ควรมีการคำนึงถึงสิ่งเหล่านี้

- อุปกรณ์ดังกล่าวอาจมีหนึ่งเน็ตเวิร์คอินเทอร์เฟสหรือหลายเน็ตเวิร์คอินเทอร์เฟส แต่ถ้ามีหลายเน็ตเวิร์คอินเทอร์เฟสจะต้องมีการคำนึงถึงสิ่งอื่นๆ เพิ่มเติม ซึ่งจะกล่าวต่อไป
- ในที่นี้ถือว่าอีเทอร์เน็ท (Ethernet) และโปรโตคอลพีพีพี (Point-to-Point Protocol) นั้นเป็นส่วนพื้นฐานที่ต้องมีอยู่แล้ว
- ไม่จำเป็นที่จะต้องมีการบู้ตผ่านระบบเครือข่าย เพราะถือว่าอุปกรณ์เหล่านี้ทำงานแบบแสตนอะโลน

ค่าต่างๆที่ควรจะกำหนดไว้ตั้งแต่อุปกรณ์เริ่มทำงานจนหยุดทำงานนั้น ควรจะเป็นค่าที่ต้องใช้ตลอดการทำงานได้แก่ ค่า hop limit ในเฮดเดอร์ แอดเดรสของดีเอ็นเอสเซิร์ฟเวอร์ เป็นต้น

4.3.2 Internet Protocol, Version 6 (IPv6) Specification [rfc2460]

ตัวอุปกรณ์นั้นสามารถที่จะรับแพ็กเก็ตที่มีเฮดเดอร์ส่วนขยายแบบใดๆก็ได้ โดยการพัฒนานั้นควรที่จะคำนึงถึงสิ่งเหล่านี้

- ถ้าตัวอุปกรณ์นั้นไม่มีความจำเป็นที่จะต้องใช้ฟังก์ชันของเฮดเดอร์ส่วนขยาย (extension header function) เราก็สามารถที่ตัดการพัฒนาส่วนที่ใช้ในการตรวจสอบเฮดเดอร์ส่วนขยายได้
- ถ้ารับแพ็กเก็ตที่มีเฮดเดอร์ส่วนขยายเข้ามาควรมีการตรวจสอบก่อนเสมอว่าเป็นเฮดเดอร์ส่วนขยายที่เรารองรับหรือไม่ ถ้าไม่ก็สามารถทิ้งแพ็กเก็ตนั้นไปได้

โดยการจัดการกับเฮดเดอร์ส่วนขยายแบบต่างๆนั้น ควรมีการกระทำดังนี้

4.3.2.1 Hop-by-Hop Options Header

- ตัวอุปกรณ์ต้องสามารถที่จะส่ง Hop-by-Hop Options header สำหรับ ในกรณีที่ต้องการสนับสนุน Multicast Listener Discovery (MLD) เท่านั้น
- ตัวอุปกรณ์นั้นควรเก็บเลขออพชันไว้ด้วย และจะมีการกระทำเก็บแพ็กเก็ตที่มีเลขออพชันที่ตัวเองสนับสนุนเท่านั้น

4.3.2.2 Routing Header

- อุปกรณ์จะไม่มีการสร้าง Routing header จากตัวเอง
- ในกรณีที่ได้รับแพ็กเก็ตที่มีค่า RH type เป็น 0 และค่าใน segment_left ไม่เป็น 0 แล้วให้ฟอร์เวอร์ดไปที่อินเทอร์มิเดียดโหนดหรือตอบกลับด้วยไอซีเอ็มพีพารามิเตอร์พรอเบลิ้มแทนก็ได้

4.3.2.3 Fragment Header

- ถ้าผู้พัฒนาอุปกรณ์มีการกำหนดไว้และสามารถรับประกันได้ว่าไม่ให้ส่งแพ็กเก็ตที่ขนาดเกินเอ็มทียูที่น้อยที่สุดแล้ว ไม่จำเป็นที่จะต้องมีการทำ fragment header sending function
- ตัวอุปกรณ์ต้องสนับสนุนการ reassemble ให้กับแพ็กเก็ตที่ถูกเฟรกเมนต์มาและทำได้คราวละ 1 แพ็กเก็ตเท่านั้น โดยที่จะต้องมีบัฟเฟอร์ในการเก็บได้อย่างน้อย 2 ส่วนโดยขนาดน้อยกว่าหรือเท่ากับลิ้งค์เอ็มทียู

4.3.2.4 Destination Options Header

ในการส่งนั้นอุปกรณ์ต้องไม่มีการสร้าง destination options header จากตัวเอง สำหรับการรับแพ็กเก็ตที่เข้ามานั้นเราต้องสนับสนุนออพชันนั้นด้วยถึงจะมีการนำแพ็กเก็ตนั้นไปใช้ต่อไป

4.3.2.5 ไม่มี Next Header

ไม่มีการกระทำใดๆกับเฮดเดอร์ชนิดนี้ทั้งนั้น

4.3.3 Path MTU Discovery for IP version 6

เราต้องสามารถรับประกันได้ว่าจะไม่มีการส่งแพ็กเก็ตที่มีขนาดใหญ่เกินค่าเอ็มทียูที่น้อยที่สุดของไอพีรุ่น 6 จึงจะสามารถเลือกที่จะไม่ทำฟังก์ชันนี้ได้

4.3.4 Neighbor Discovery for IP Version 6 (IPv6) [rfc2461]

สืบเนื่องจากที่กล่าวไว้ในข้างต้นเกี่ยวกับตัวอุปกรณ์ ดังนั้นฟังก์ชันที่เกี่ยวกับเราเตอร์เหล่านี้ไม่จำเป็นที่จะต้องทำขึ้นมา

1. ส่ง Router advertisement messages
2. รับ Router solicitation messages
3. ส่ง Redirect messages

สำหรับการรับ Redirect message นั้นเป็นฟังก์ชันของโฮสต์อยู่แล้ว แต่จะอิมพลีเมนต์หรือไม่ก็ได้ เช่นเดียวกันกับการทำ destination cache ที่จะทำหรือไม่ทำก็ได้ ซึ่งถ้าต้องการที่จะสร้าง destination cache แล้วเรามีความจำเป็นที่จะต้องคำนึงถึงการทำงานของมันร่วมกันกับเนเอบร์อันรีชเอเบิลดีเทคชัน (Neighbor Unreachable Detection : NUD) ด้วย

ถ้าตัวอุปกรณ์นั้นมีหลายเน็ตเวิร์คอินเทอร์เฟสก็ต้องใช้ทรัพยากรมากขึ้นเพราะเราต้องคำนึงสิ่เหล่านี้เพิ่มเติม

1. ชนิดของข้อมูลในการเราท์ เช่น destination cache และ routing table
2. จำนวนข้อมูลที่เก็บอยู่ในแคชก็มากขึ้นไปด้วย เช่น Neighbor Discovery cache, Default Router list, Prefix list เป็นต้น

4.3.5 IPv6 Stateless Address Autoconfiguration [rfc2462]

ในการทำ Duplicated Address Detection (DAD) ทุกฟังก์ชันนั้นต้องอาศัยฟังก์ชันเนเบอร์ดิสเคิร์ฟเวอรี่ ดังนั้นการที่จะพัฒนาส่วนนี้ขึ้นมานั้นไม่มีผลกระทบต่อทรัพยากรของระบบมาก อาจจะพัฒนาขึ้นมาหรือไม่ก็ได้

4.3.6 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [rfc2463]

ดังที่กล่าวมาแล้วเกี่ยวกับตัวอุปกรณ์ เราจึงไม่มีความจำเป็นที่จะต้องทำไอซีเอ็มพีแมสเซสที่ใช้โดยกับเราเตอร์ แต่ในส่วนของ ICMP Echo Request/Reply นั้นเป็นพื้นฐานต่ำสุดที่พึงกระทำนอกจากนี้เรายังสามารถเพิ่มฟังก์ชันการทำงานของ ICMPv6 เข้าไปได้ตามความจำเป็น ดังตารางนี้

Type	Code	Necessary
Destination unreachable message	0 : No route to destination	No
	1: Administratively Prohibited	No
	3 : Address unreachable	No
	4 : Port unreachable	Yes
Packet Too Big Message	0 : Packet Too Big	No
Time Exceeded Message	0: Hop limit exceeded	No
	1: Fragment reassembly time exceeded	No
Parameter Problem Message	0: Erroneous header field	Yes
	1: Unrecognized next heade	Yes
	2: Unrecognized IPv6 option	Yes
Echo Request Message	0: Echo request	Yes
Echo Reply Message	0: Echo reply	Yes

ตารางที่ 4-1 แสดงเกี่ยวกับไอซีแมสแซสรุ่น 6 สำหรับระบบฝังตัว

4.3.7 IP Version 6 Addressing Architecture [rfc2373]

โดยปกติแล้วตัวอุปกรณ์นั้นจะต้องมีแอดเดรสเหล่านี้

1. Loopback (::1)
2. Node-local all nodes multicast (ff01::1)
3. Link-local all nodes multicast (ff02::1)

แต่ถ้าต้องการให้มีการทำ address autoconfiguration ตัวอุปกร์ควรจะมีสนับสนุนแอดเดรสต่อไปนี้ด้วยเช่นกัน

1. Link-local unicast (fe80/64 host ID)
2. Solicited-node multicast ที่เกี่ยวข้องกับ address autoconfiguration
3. Unicast (prefix/64 host ID) ที่เกี่ยวข้องกับพรีฟิกส์ออพชันของ router advertise messages
4. Solicited-node multicast ที่เกี่ยวข้องกับ address autoconfiguration

ในกรณีที่ต้องการให้มีการทำ manual configuration ด้วยควรมีการสนับสนุนแอดเดรสดังต่อไปนี้

1. Arbitrary unicast
2. Solicited-node multicast ที่เกี่ยวข้องกับ manual configuration

4.3.8 การพิจารณาเกี่ยวกับขอบเขต

ถึงแม้ว่าตัวอุปกรณ์จะมีแค่ 1 ฟิสิคัลอินเทอร์เฟส เราก็สามารถที่จะกำหนดให้มีได้หลายลอจิคัลอินเทอร์เฟสก็ได้ เช่น ลูปแบ็คอินเทอร์เฟส (Loopback interface) เป็นต้น ได้เช่นกัน

4.3.9 การเลือกแอดเดรสหลักสำหรับไอพีรุ่น 6

ตัวอุปกรณ์จะต้องสนับสนุนการเลือกแอดเดรสหลักสำหรับไอพีรุ่น 6 โดยควรพิจารณา

1. ถ้าตัวอุปกรณ์ไม่มีการสนับสนุน ICMP redirect message เราสามารถตัดส่วนของกระบวนการตรวจสอบออกไปได้
2. ในการเลือกแอดเดรสต้นทางหลักควรพิจารณาถึงว่าเรามีการสนับสนุนโมบิลิตี้หรือไม่ มีกี่อินเทอร์เฟส มีการทำ policy table หรือไม่ รวมทั้งสิ่งอื่นที่สนับสนุนและไม่สนับสนุนด้วยเช่นกัน
3. ในการเลือกแอดเดรสปลายทางหลักนั้นก็ต้องมีการคำนึงถึงเรื่องการสนับสนุนโมบิลิตี้ policy table รวมทั้งสิ่งอื่นที่สนับสนุนและไม่สนับสนุนด้วยเช่นกัน

4.3.10 Multicast Listener Discovery (MLD) for IPv6 [rfc2710]

เมื่อตัวอุปกรณ์นี้มีความจำเป็นที่จะต้องเข้าไปร่วมกับกลุ่มมัลติคาสต์ฟังก์ชันนี้เป็นสิ่งที่พึงกระทำ

4.3.11 IPsec สำหรับระบบฝังตัว

ไอพีรุ่น 6 ใช้โครงสร้างของ IPsec ในการเป็นพื้นฐานทางด้านความปลอดภัย ซึ่งมีด้วยกันหลายอย่างได้กัน แต่สำหรับระบบฝังตัวนั้นไม่มีความจำเป็นที่ต้องนำทั้งหมดมาใช้ โดยสามารถพิจารณาได้ดังนี้

4.3.11.1 IPsec specification สำหรับระบบฝังตัว

ตัวอุปกรณ์นั้นต้องมีการทำ IPsec ในโหมดทรานสปอร์ต โดยอาจจะทำใหมดเทอร์นเนลด้วยก็ได้ แต่ไม่จำเป็นสำหรับอุปกรณ์ที่เป็นโฮสต์ที่ไม่ได้มีการสื่อสารผ่านเซคเคียวริตี้เกตเวย์ และพารามิเตอร์บางอย่างของ Security Assocication อาจจะไม่ต้องทำก็ได้ เช่น automated key ซึ่งส่วนนี้ขึ้นอยู่กับความต้องการของนักพัฒนา

4.3.11.2 Security Policy Database

อย่างน้อยที่สุดต้องมีการทำในส่วนที่เกี่ยวข้องกับไอพีแอดเดรสขึ้นมา สำหรับส่วนอื่นๆนั้นจะทำหรือไม่ก็ได้

4.3.11.3 Security Protocol

นักพัฒนาจะต้องมีการพัฒนา IP Encapsulating Security Payload (ESP) [rfc2406] และ IP Authentication Header (AH) ขึ้นมาโดยการใช้นั้นจะใช้แค่อันใดอันหนึ่งหรือทั้งคู่ก็ได้

4.3.11.4 การทรานสฟอร์มและอัลกอลิทึม

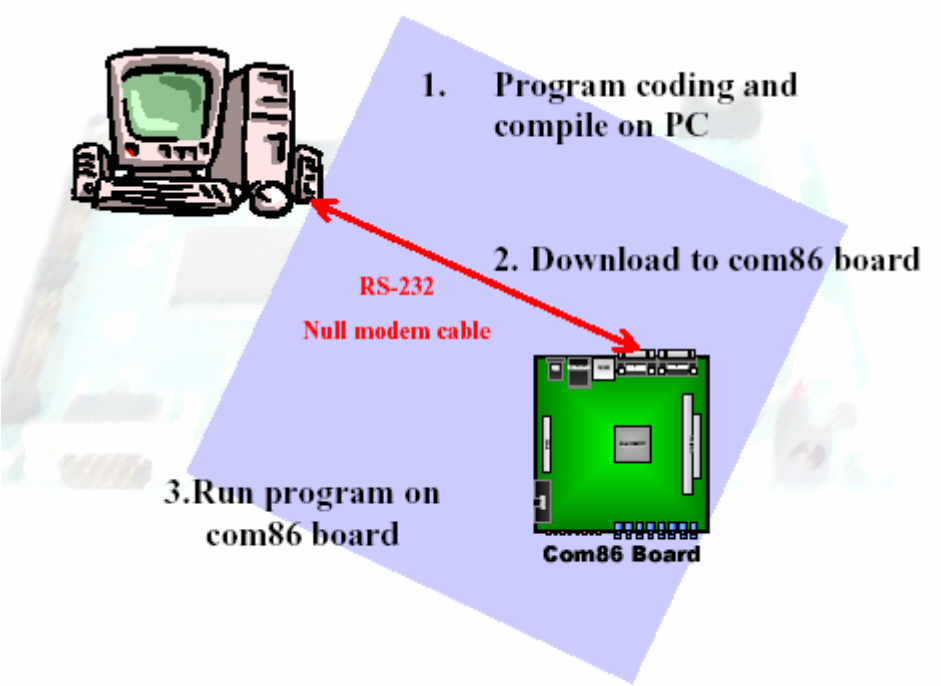
ในกรณีของการพัฒนาเอ็นคริปท์ อัลกอริทึม AES-CBC นั้นเป็นสิ่งที่ต้องกระทำ แต่สำหรับอัลกอริทึมอื่นๆ เช่น DES-CBC และ 3DES-CBC นั้นจะทำเฉพาะ backward compatibility หรือไม่ทำก็ได้ สำหรับอัลกอริทึม NULL Encryption อาจจะทำไว้ใช้ในการดีบักก็ได้ ส่วนการทำออเทนทิเคชันนั้นมีอัลกอลิทึมที่แนะนำว่าน่าจะทำ คือ HMAC-SHA-1-96 [rfc2404] และ HMAC-MD5-1-96 [rfc2403]

4.3.11.5 SA และการจัดการเกี่ยวกับคีย์

สำหรับ manual key management นั้นเป็นสิ่งที่ต้องพัฒนาขึ้นมา โดยมีจุดประสงค์ที่จะสุ่มค่าของอินเทอร์เฟสไอดีขึ้นมา ในส่วนของ automated key management นั้นอาจจะทำ IKE ก็ได้ ตามแต่ต้องการ

4.4 การพัฒนาโปรแกรมบนชุดพัฒนาคอม86

การพัฒนาโปรแกรมบนชุดพัฒนาคอม86 สามารถพัฒนาได้จากหลายภาษาด้วยกันอาทิเช่น ภาษาซี (C) หรือซีพลัสพลัส (C++) ภาษาปาสคาล (Pascal) ภาษาแอสเซมบลี (Assembly) เป็นต้น โดยการพัฒนาโปรแกรมสามารถทำได้บนเครื่องคอมพิวเตอร์บุคคล (Personal computer) แล้วทำการส่งไฟล์ที่ผ่านการคอมไพล์และสร้างเป็น ไฟล์ที่สามารถรันได้บนระบบปฏิบัติการดอส (executable file) ไปยังบอร์ดชุดพัฒนา แล้วทำการรัน รูปที่4-12



รูปที่ 4-12 ขั้นตอนพื้นฐานในการพัฒนาโปรแกรมทั่วไปบนชุดพัฒนาคอม 86

การที่จะส่งไฟล์ไปไว้บนบอร์ดชุดพัฒนาได้นั้น จะส่งผ่านทางโปรแกรมเทอร์มินัลทั่วไป โดยในชุดพัฒนาคอม86 ได้มีโปรแกรมยูทิลิตี้ตัวหนึ่งในการรองรับการทำงานในส่วนนี้เป็นชื่อโปรแกรมว่า transfer.exe เป็นโปรแกรมที่ช่วยในการรับส่งไฟล์ต่างๆ ผ่านทางพอร์ตอนุกรม ซึ่งจะใช้โปรโตคอล เอ็กซ์-โมเด็ม (X-modem) ในการรับส่งไฟล์ที่ความเร็ว 57600 bps ซึ่งรูปแบบการใช้งานโปรแกรมแสดงดังรูปที่ 4-10

```
TRANSFER [/S | /R] [/H][/?] filename
    where   /S   To send file
            /R   To receive file
                 /?   Show this help
                 /H   Show this help
the default option is /R
```

ตัวอย่างการเรียกใช้งาน

```
A:\>Transfer program1.exe
```

รูปที่ 4-13 รูปแบบการใช้โปรแกรมในการรับส่งไฟล์กับชุดพัฒนาคอม 86

4.4.1 การใช้งานพอร์ตอนุกรมของชุดพัฒนาคอม86

สำหรับพอร์ตอนุกรมบนชุดพัฒนาคอม86 ในที่นี้หมายถึงพอร์ตคอม2(COM2) ซึ่งโดยมากจะใช้ไปควบคุมหรือติดต่อกับอุปกรณ์อื่นๆ ต่อไป โดยคุณสมบัติทั่วไปของพอร์ตอนุกรมนี้ได้แก่

- สามารถรับส่งข้อมูลได้ที่อัตราเร็วสูงสุด 115200 bps
- การส่งข้อมูลเป็นแบบสองทิศทางในเวลาเดียวกัน(Full-duplex)
- ใช้บิตที่เป็นสต็อปบิต 1 หรือ 2 บิตก็ได้

ในการใช้งานพอร์ตอนุกรมนี้ผู้เขียนโปรแกรมจำเป็นต้องทราบรายละเอียดของรีจิสเตอร์ตัวต่างๆ ของพอร์ตว่าต้องมีการส่งค่าใดไปให้บ้าง นอกจากนี้ยังจะต้องทราบอัตราเร็วของการรับส่งข้อมูลผ่านทางพอร์ตอนุกรมของอุปกรณ์ที่ชุดพัฒนาจะไปต่อเชื่อมด้วย สำหรับรายละเอียดของรีจิสเตอร์ต่างๆที่เกี่ยวข้องกับการรับส่งข้อมูลผ่านซีเรียลพอร์ต แสดงดังรูป 4-11

SPCON0	Serial port control 0
SPCON1	Serial port control 1
SPSTAT	Serial port status
SPIMSK	Serial port Interrupt Mask
SPTXD	Serial port Transmit data
SPRXD	Serial port Receive data
SPRXDP	Serial port Receive data peak
SPBDV	Serial port Baud Rate Divisor

รูปที่ 4-14 รีจิสเตอร์ที่เกี่ยวข้องในการเขียนโปรแกรมเพื่อติดต่อผ่านทางพอร์ตอนุกรม

4.4.2 การเขียนโปรแกรมเพื่อให้ชุดพัฒนาคอม86 สามารถเชื่อมต่อเข้าสู่เครือข่าย

การที่จะให้ชุดพัฒนาคอม86 สามารถเชื่อมต่อเข้าสู่ระบบเครือข่ายท้องถิ่น (Local Area Network) นั้น ในขั้นแรกจะต้องทำการต่อสายยูทีพีเข้ากับชุดพัฒนาคอม86 จากนั้นต่อสายซัพพลายเข้ากับชุดพัฒนา และทำการโหลดโปรแกรมแพ็กเกตไดร์เวอร์ โดยพิมพ์คำว่า “pnppd 0x60” ที่คอมมานไลน์ เมื่อสิ้นสุดขั้นตอนนี้ ชุดพัฒนาคอม86 จะสามารถเชื่อมต่อ เข้าสู่เครือข่ายได้ โดยสามารถทดสอบว่าชุดพัฒนาคอม86 เชื่อมต่อเข้าสู่เครือข่ายได้ โดย โปรแกรม ping ที่อยู่ในไดเร็กทอรี a:/demo โดยการใช้โปรแกรม ping มีรูปแบบของคำสั่งดังนี้ “ping [หมายเลขไอพีที่ต้องการทดสอบ]”

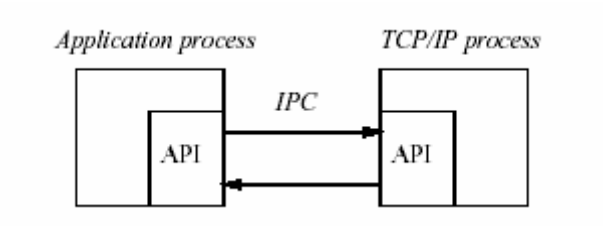
4.5 การพัฒนาโปรแกรมบนไลท์เวททีซีพี/ไอพีแสต็ก

การที่จะติดต่อสื่อสารกับตัวโปรโตคอลแสต็กนั้นมีด้วยกัน 2 วิธี คือ

1. การเรียกฟังก์ชันในการทำงานโดยตรง ซึ่งวิธีนี้จะทำให้โปรแกรมที่พัฒนาขึ้นมีความซับซ้อนและเข้าใจได้ยาก

2. ใช้เอลดับเบิลยูไอพีเอพีไอ (lwIP API) ซึ่งมีการทำงานคล้ายคลึงกันกับบีเอสดีซ็อกเก็ตเอพีไอ (BSD socket API) โดยตัวเอพีไอนั้นแบ่งออกเป็น 2 ส่วนด้วยกันตามลักษณะของโพรเซสโมเดลดังที่ได้กล่าวมาก่อนหน้านี้ โดยส่วนหนึ่งจะเป็นของทีซีพี/ไอพีโพรเซสอีกส่วนหนึ่งจะเป็นของแอพพลิเคชันโพรเซส

ซึ่งการสื่อสารจะใช้วิธีอินเทอร์โพรเซสคอมมิวนิเคชัน (interprocess communication: IPC) ซึ่งทำงานโดยอีมูเลชันเลเยอร์ของระบบปฏิบัติการของไลท์เวททีซีพีไอพีแสต็กทำให้ไม่ต้องอาศัยการทำงานของระบบปฏิบัติการ โดยอินเทอร์โพรเซสคอมมิวนิเคชันนั้นได้ใช้กระบวนการของแชร์เมมโมรี่ (shared memory) แมสเซสพาสซิ่ง (message passing) และเซมาฟอร์ (semaphores)



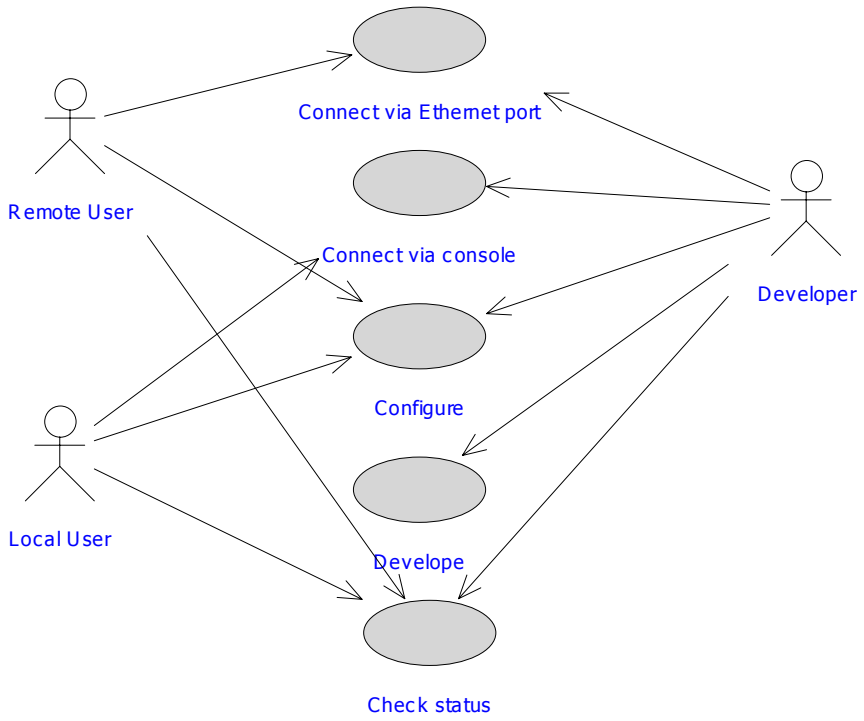
รูปที่ 4-15 การสื่อสารกันผ่านเอพีไอ

โดยใช้แชร์เมมโมรี่ในการส่งข้อมูลในบัฟเฟอร์ข้ามกันระหว่างแอพพลิเคชันโพรเซสและทีซีพี/ไอพีโพรเซส ซึ่งมีการใช้เซมาฟอร์เพื่อทำเมมโมรี่โพรเทคชัน สำหรับฟังก์ชันที่ทำงานเกี่ยวกับเน็ทเวิร์คคอนเนคชันนั้นจะสื่อสารกันด้วยการใช้กระบวนการของแมสเซสพาสซิ่งในการส่งเมสแซสเกี่ยวกับโอเปอร์เรชันต่างๆและอาร์กิวเมนต์ของโอเปอร์เรชันนั้นๆ ให้แก่กันและส่งผลลัพธ์กลับมาด้วยวิธีการเดียวกัน

4.6 การออกแบบระบบ

4.6.1 Use case diagram

ไดอะแกรมนี้แสดงฟังก์ชันการทำงานหลักๆ ที่สำคัญของระบบ และมีรายละเอียดแสดงไว้ดังนี้



รูปที่ 4-16 Use case diagram ของระบบ

Connect via Ethernet port

Use case	Connect via Ethernet port
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้ต้องสื่อสารผ่านโปรโตคอลเอชทีทีพีด้วยการใช้บราวเซอร์

Connect via console

Use case	Connect via console
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้ต้องสื่อสารผ่านพอร์ตอนุกรมด้วยโปรแกรมเทอร์มินัล

Configure

Use case	Configure
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้งานจะตั้งค่าคุณสมบัติให้กับระบบได้ก็ต่อเมื่อมีการเชื่อมต่อเข้ามาในระบบแล้วเท่านั้น โดยการปรับแต่งค่าคุณสมบัตินั้นจะแยกออกเป็น 2 ส่วนคือ - ผ่านทางคอนโลพอร์ตอนุกรม - ผ่านทางอีเทอร์เน็ต

Develope

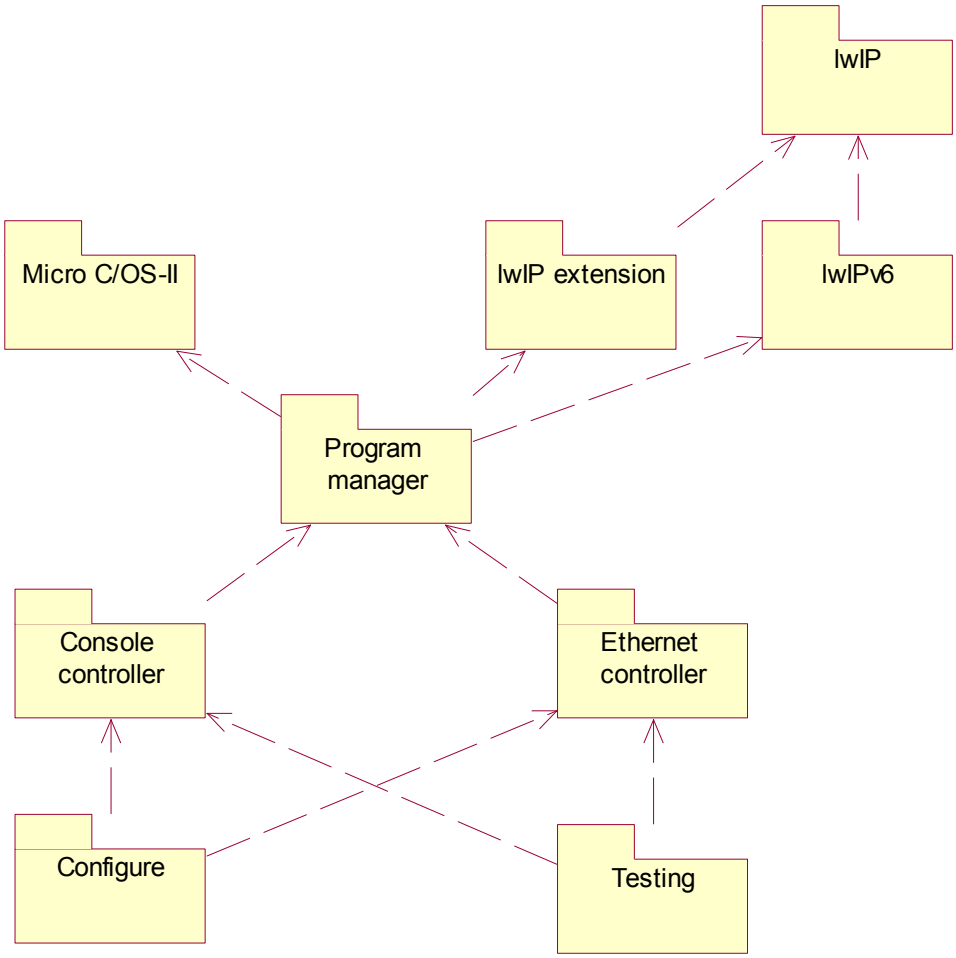
Use case	develop
ผู้ใช้	Developer
เงื่อนไข	การนำไปพัฒนาต่อนั้นขึ้นกับลักษณะงานที่นำไปใช้และข้อจำกัดของระบบ

Check status

Use case	Check status
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้สามารถตรวจสอบสถานะของการทำงานได้โดยใช้เครื่องมือที่มีให้หรือแก้ไขค่าในไฟล์ lwipopts.h เพื่อให้แสดงสถานะอื่นๆเพิ่มเติมได้

ตารางที่ 4-2 อธิบาย Use case diagram

4.6.2 Component diagram



รูปที่ 4-17 Component diagram ของระบบ

แผนภาพด้านบนนั้นแสดงถึงองค์ประกอบต่างๆของซอฟแวร์ของระบบ ซึ่งประกอบด้วย

1. Micro C/OS-II เป็นส่วนของเรียลไทม์โอเอส จัดการเกี่ยวกับทาสก์ต่างๆ ที่สร้างขึ้น รวมทั้งการสื่อสารระหว่างทาสก์โดยวิธีการต่างๆ เช่น semaphore, message boxes เป็นต้น

2. lwIP เป็นส่วนของโปรโตคอลแสต็ก ซึ่งทำหน้าที่ให้บริการเกี่ยวกับการสื่อสารผ่านเครือข่ายและการทำงานที่เกี่ยวข้องกับระบบเครือข่าย ตัวอย่างโปรโตคอลที่สนับสนุนได้แก่ ไอพี ทีซีพี ยูดีพี เอชทีทีพี เป็นต้น

3. Console controller จัดการเกี่ยวกับการติดต่อสื่อสารกับผู้ใช้กับระบบผ่านทางคอนโซลพอร์ตอนุกรม รวมทั้งการรับอินพุตเข้ามาและส่งเอาท์พุตออกไปแสดงผลด้วย

4. Ethernet controller จัดการติดต่อสื่อสารกับผู้ใช้กับระบบผ่านทางอีเทอร์เน็ท โดยการใช้โปรโตคอลเอชทีทีพี รวมทั้งการรับอินพุตเข้ามาและส่งเอาท์พุตออกไปแสดง

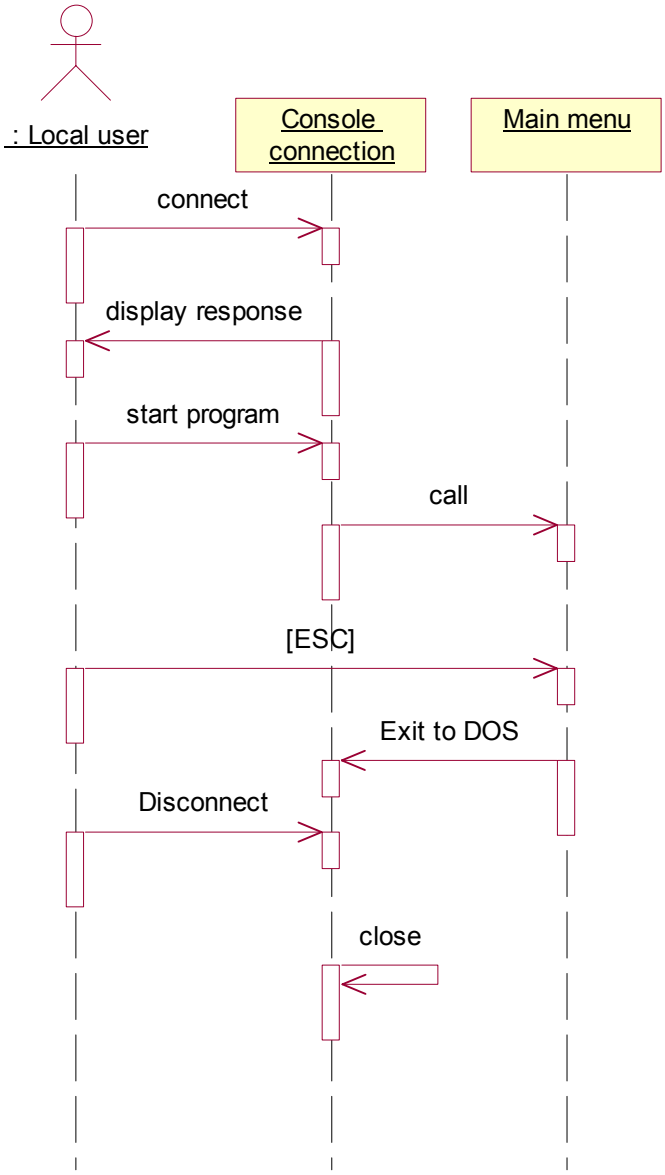
5. Configure ทำหน้าที่ในการกำหนดค่าคุณสมบัติต่างๆ ให้กับระบบ โดยรับค่าต่างๆ จากผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรมหรืออีเทอร์เน็ท เช่น การกำหนดไอพีแอดเดรสให้กับเน็ทเวิร์คอินเทอร์เฟส

6. Testing เป็นส่วนที่ใช้ในการทดสอบการทำงานว่าระบบสามารถทำงานได้ถูกต้องตามต้องการหรือไม่ เช่น ทดสอบว่าระบบสามารถทำงานในเครือข่ายได้ โดยการใช้โปรแกรมประยุกต์ ping มาที่ไอพีแอดเดรสที่กำหนดไว้ให้กับอุปกรณ์ แล้วดูว่าได้รับการตอบกลับหรือไม่ เป็นต้น

4.6.3 Sequence diagram

เป็นไดอะแกรมที่จำลองการทำงานของระบบ เพื่อแสดงให้เห็นถึงการทำงานที่เป็นลำดับขั้นตอนและการทำงานร่วมกันของอ็อบเจ็คต่างๆ ในระบบ

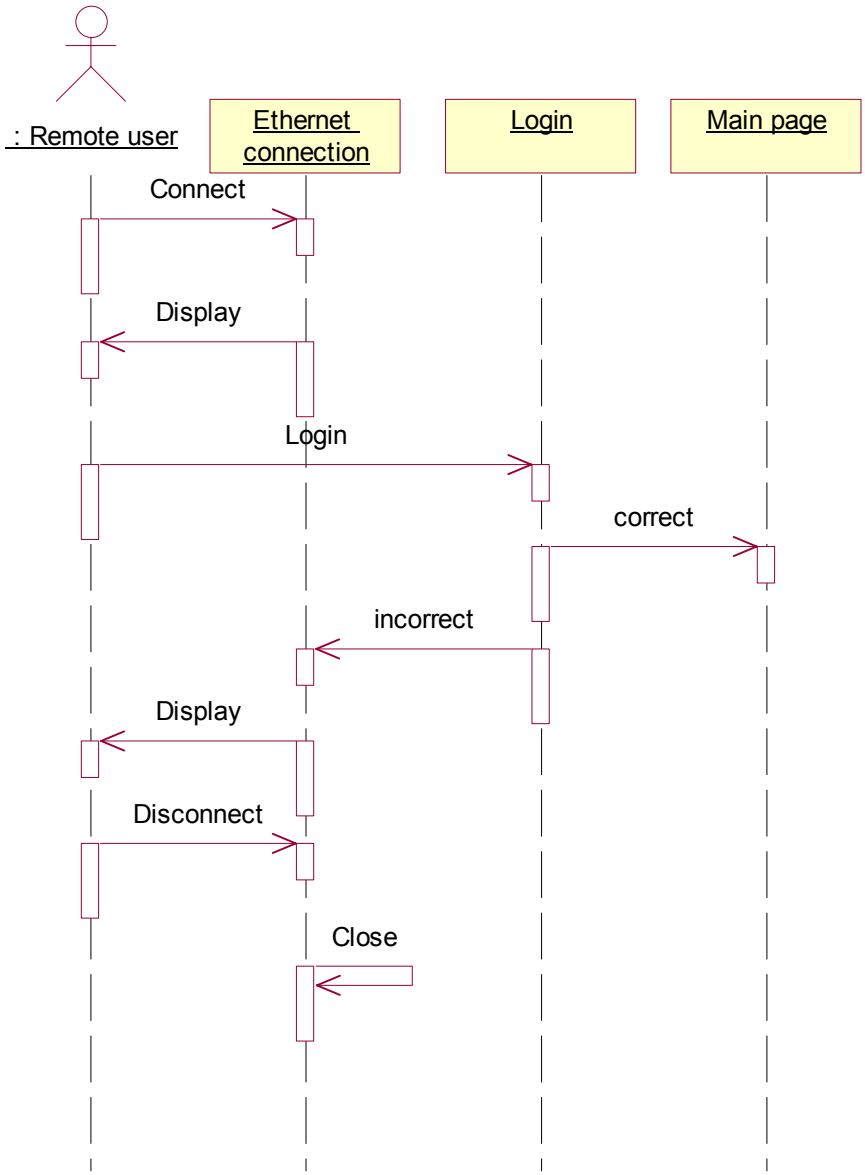
4.6.3.1 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม



รูปที่ 4-18 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม

จากรูปที่ 4-18 เมื่อผู้ใช้ติดต่อเข้ามาในระบบผ่านทางโปรแกรมเทอร์มินัลแล้ว ผู้ใช้สามารถเลือกการใช้บริการจากเมนูหลักก่อน แล้วทำตามขั้นตอนที่ระบบได้วางไว้ โดยถ้าต้องการออกจากระบบสามารถกดปุ่ม ESC ได้

4.6.3.2 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท

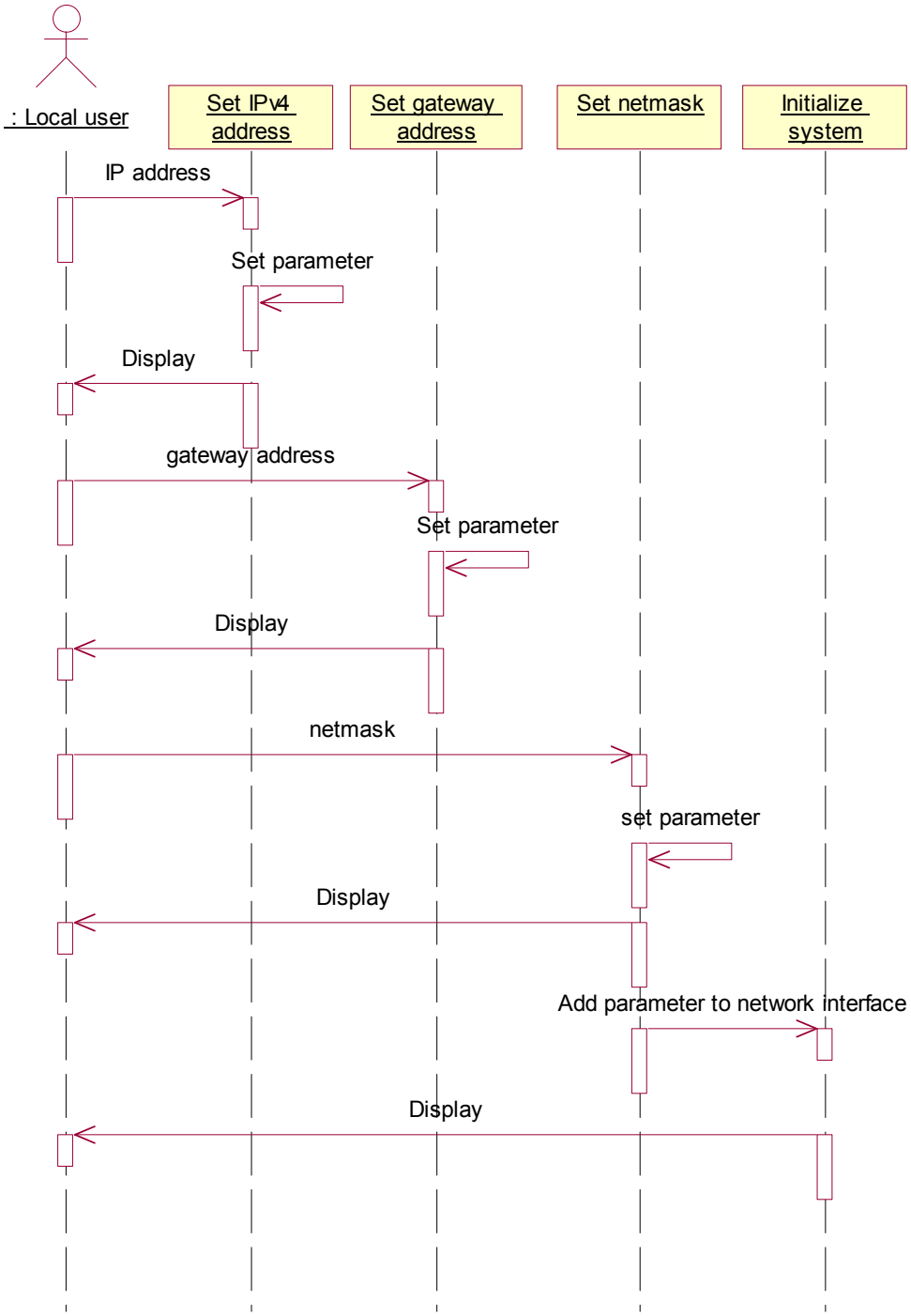


รูปที่ 4-19 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท

เมื่อผู้ใช้ต้องการติดต่อเข้ามาในระบบจะต้องมีการ log in จากภายนอกเข้ามาโดยการใช้โปรแกรมบราวเซอร์ขึ้นมาทำงาน เมื่อเข้ามาในระบบแล้วก็จะมีเมนูหลักให้เลือกใช้งาน เมื่อเลือกใช้งานส่วนไหนก็จะต้องทำงานตามขั้นตอนของส่วนนั้นต่อไป

4.6.3.3 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4

จากภาพที่ 4-20 แสดงขั้นตอนทำงานในการกำหนดค่าคุณสมบัติเกี่ยวกับไอพีรุ่น 4 ให้กับระบบ ผู้ใช้ต้องกระทำการป้อนค่า ไอพีแอดเดรส ตามด้วยค่าของเกตเวย์ และเน็ตมาสก์ตามลำดับ หลังจากที่ป้อนค่าทุกอย่างครบแล้ว ก็จะนำค่าเหล่านี้ไปกำหนดให้กับเน็ตเวิร์คอินเทอร์เฟส และอินนิเชียลค่าระบบขึ้นมาใหม่เพื่อแก้ไขค่าและทำงาน

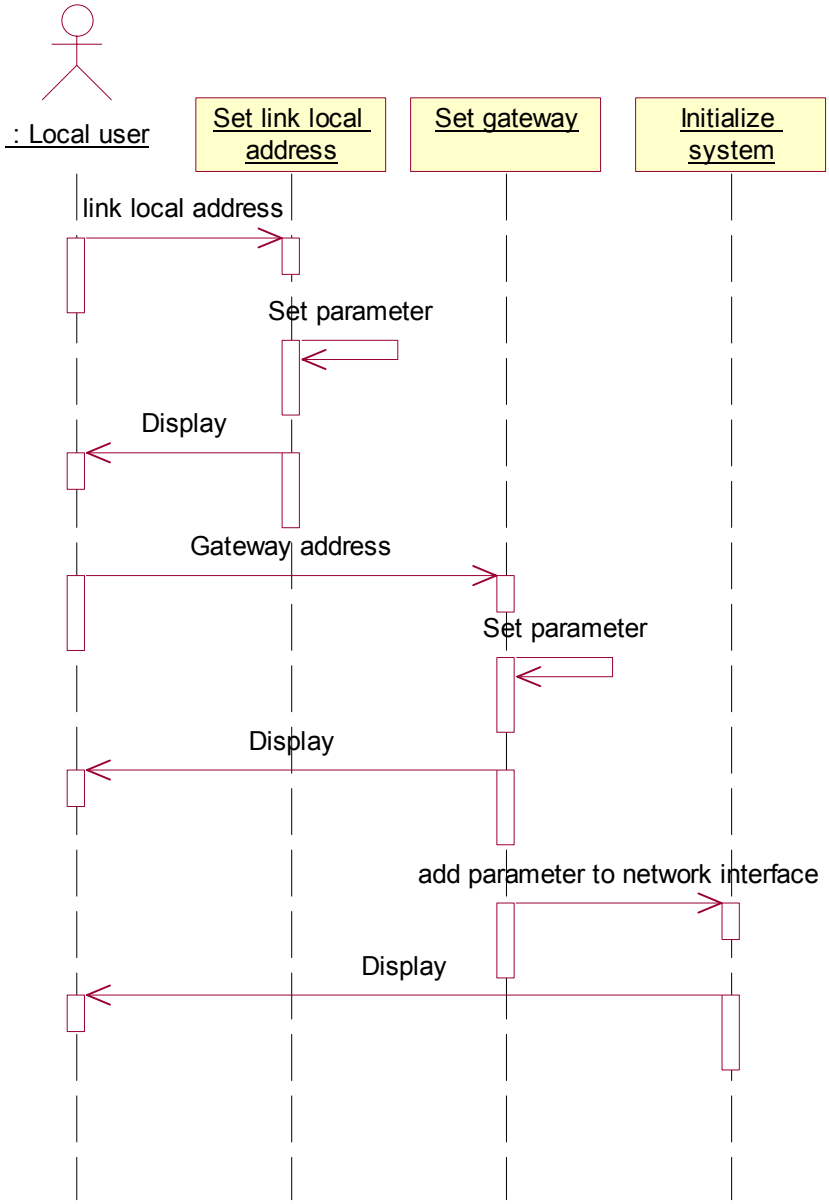


รูปที่ 4-20 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4

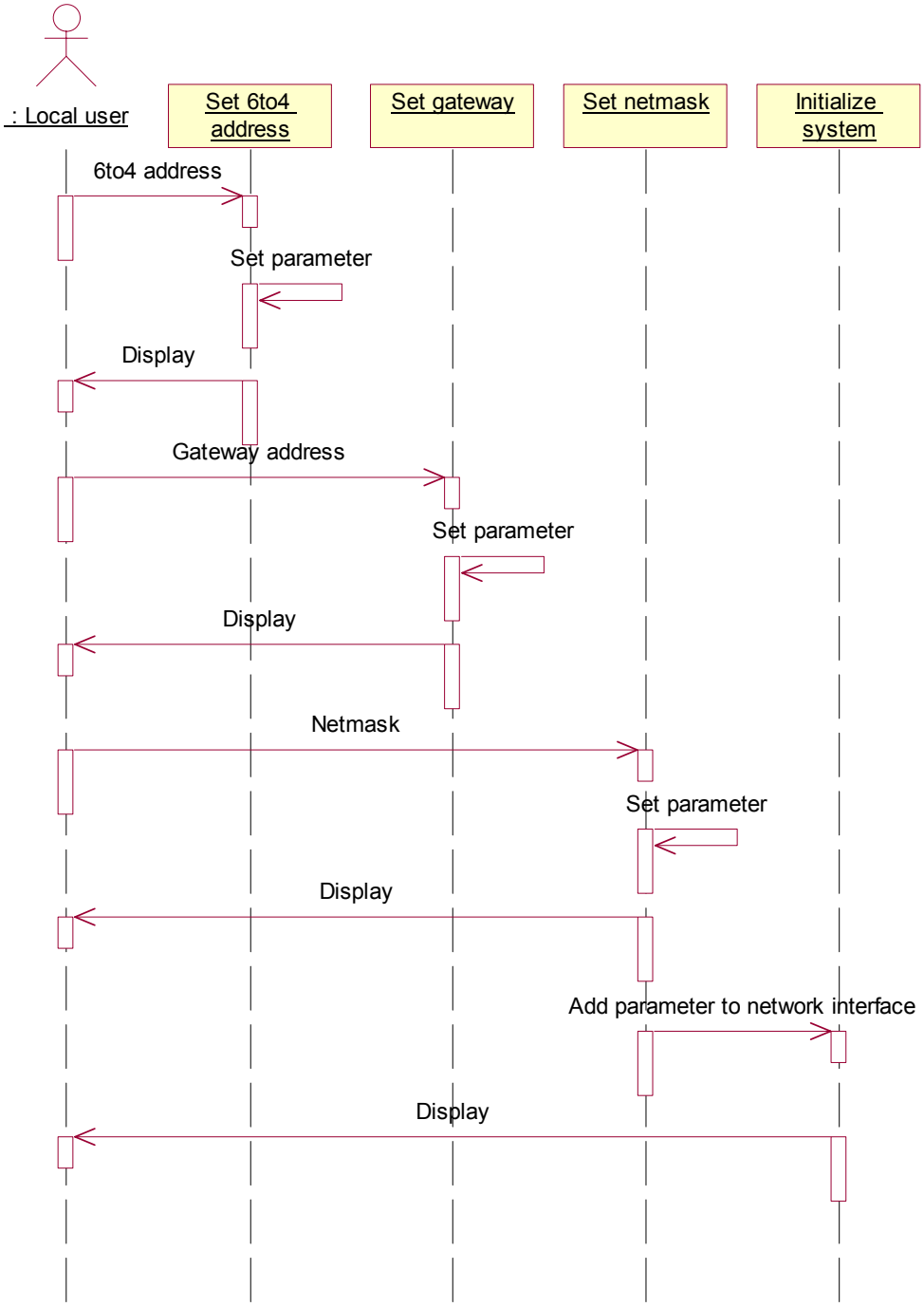
4.6.3.4 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 6

การตั้งค่าคุณสมบัติเกี่ยวกับไอพีรุ่น 6 นั้นแบ่งออกเป็นดังนี้

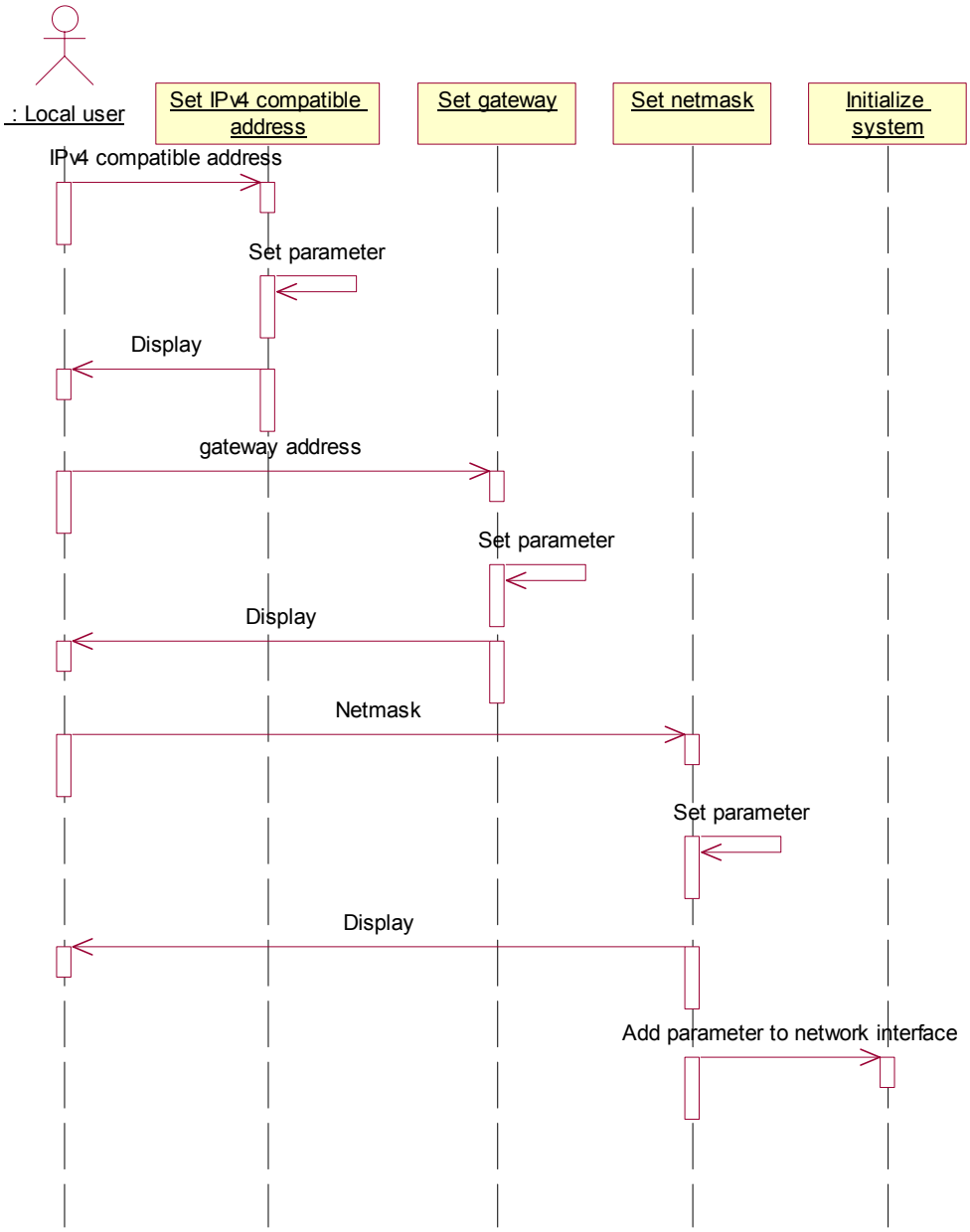
1. การตั้งค่าคุณสมบัติเกี่ยวกับลิงค์โลคอลแอดเดรส
2. การตั้งค่าคุณสมบัติเกี่ยวกับคอมแพททิเบิลแอดเดรส
   - แบบ 6to4
   - แบบ IPv4 compatible



รูปที่ 4-21 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบลิงค์โลคอลแอดเดรส



รูปที่ 4-22 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ 6to4



รูปที่ 4-23 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ IPv4 compatible