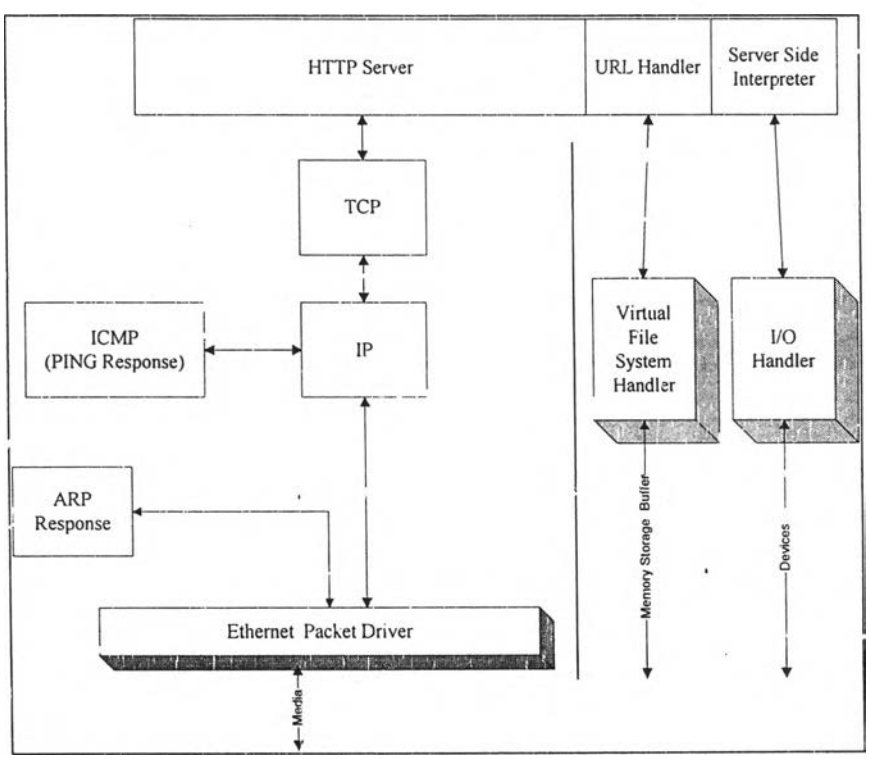
บทที่ 4
การพัฒนาซอฟต์แวร์เชื่อมต่อระบบเครือข่าย

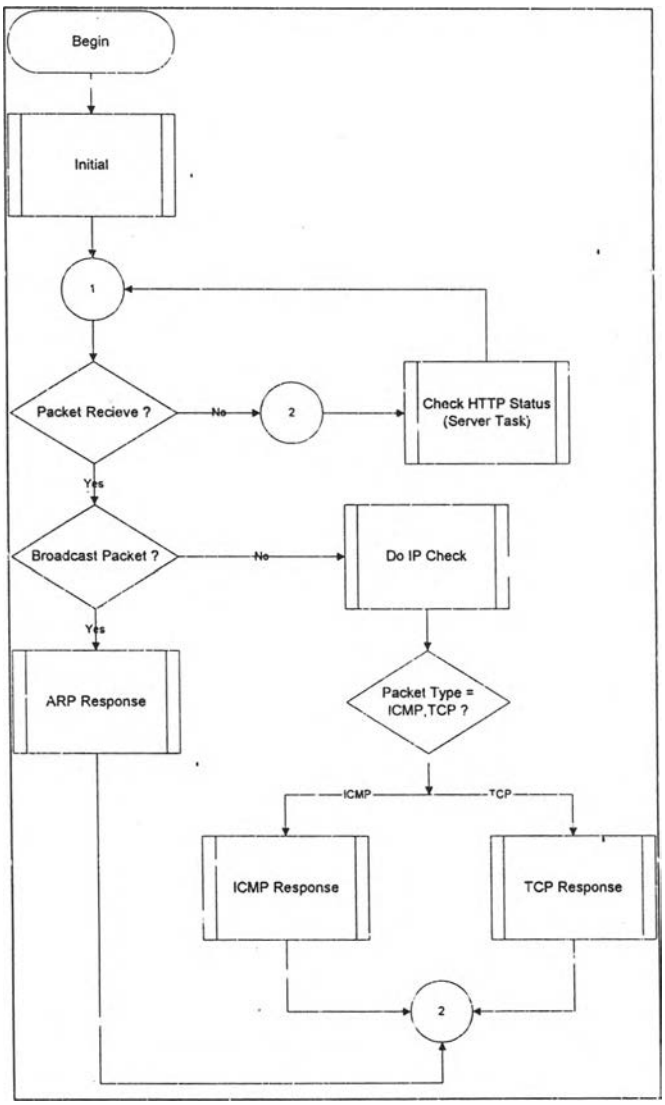
ซอฟต์แวร์เพื่อใช้ในการควบคุมและเชื่อมต่อระบบเครือข่ายนั้น บนระบบงานทั่วไปมักถูกจัดการโดยซอฟต์แวร์ระบบชนิดพิเศษ หรือระบบปฏิบัติการ (Operating System) ทั้งนี้เนื่องจากในงานวิจัยดังกล่าว มิได้ใช้ระบบปฏิบัติการบนระบบควบคุมแบบฝังตัวเข้าช่วยแต่อย่างใด การควบคุมงานทุกอย่างจึงถูกจัดการโดยซอฟต์แวร์ที่พัฒนาขึ้นโดยเฉพาะ ซึ่งประกอบไปด้วยซอฟต์แวร์ติดต่ออุปกรณ์การควบคุมการเชื่อมต่อระบบเครือข่าย (NIC Driver) และซอฟต์แวร์เพื่อช่วยในการจัดการการเชื่อมต่อระบบเครือข่ายตามมาตรฐาน TCP/IP เฉพาะส่วนที่จำเป็นสำหรับการสร้างระบบเว็บเซิร์ฟเวอร์แบบฝังตัว ซึ่งโครงสร้างของซอฟต์แวร์ส่วนที่เกี่ยวข้องทั้งหมดสามารถอธิบายได้ดังภาพที่ 14



ภาพที่ 14 ส่วนประกอบทางซอฟต์แวร์ของระบบ

การทำงานทางซอฟต์แวร์ของระบบแบ่งออกเป็นระดับชั้น ตามการทำงานได้โดยในระดับล่างจะเป็นซอฟต์แวร์ติดต่ออุปกรณ์ควบคุมการเชื่อมต่อระบบเครือข่ายหรือ Driver ซึ่งจะทำหน้าที่ควบคุมการถ่ายโอนข้อมูล ระหว่างหน่วยความจำหลักและหน่วยความจำชั่วคราวของอุปกรณ์ควบคุมการเชื่อมต่อระบบเครือข่าย เมื่อข้อมูลหรือ Packet ผ่านเข้าสู่ระบบ ซอฟต์แวร์ Driver ดังกล่าว จะทำการตรวจสอบว่าข้อมูลใน Packet นั้นเป็นข้อมูลประเภท Broadcast หรือไม่

หากข้อมูลที่ได้รับเป็นข้อมูลประเภท Broadcast ระบบจะส่งต่อให้กับโปรแกรม ARP Response เพื่อตอบรับต่อไป ส่วนกรณีที่ข้อมูลที่ได้รับมิได้เป็น Packet ประเภท Broadcast ระบบจะส่งต่อให้กับโปรแกรม IP เพื่อตรวจสอบและจำแนกข้อมูลซึ่งหากข้อมูลที่ได้รับถูกจำแนกเป็น PING โปรแกรม ICMP จะถูกเรียกเพื่อสร้าง Ping Response และ กรณีที่ข้อมูลที่ได้รับถูกจำแนกเป็น TCP โปรแกรม TCP จะตรวจสอบและแยกข้อมูลเพื่อส่งต่อให้กับซอฟต์แวร์ระบบเว็บเซิร์ฟเวอร์ต่อไป ในทำนองเดียวกันหากเป็นการส่งข้อมูล ข้อมูลจะถูกส่งผ่านขั้นตอนต่างๆ จากบนลงล่างเพื่อประมวล ตรวจสอบความถูกต้องและส่งข้อมูลออกสู่ระบบเครือข่ายในที่สุด ซึ่งจากลักษณะการทำงานดังกล่าว ทำให้การพัฒนาซอฟต์แวร์ ถูกแบ่งเป็นแต่ละส่วน และการทำงานระหว่างโปรแกรมย่อยต่างๆ ถูกควบคุมหรือเรียกใช้งานจากโปรแกรมหลัก ซึ่งรายละเอียดของซอฟต์แวร์ส่วนโปรแกรมหลักสามารถแสดงได้ดังภาพที่ 15



ภาพที่ 15 การทำงานของโปรแกรมหลัก

เมื่อเริ่มต้นการทำงาน โปรแกรมหลักจะทำการเตรียมข้อมูล (Initialize) และตั้งค่าสถานะต่างๆ ของระบบ ซึ่งในที่นี้รวมความถึงการตั้งค่า Stack และการตั้งค่าของหน่วยควบคุมการเชื่อมต่อระบบเครือข่ายด้วย จากนั้นโปรแกรมหลักจะทำการตรวจสอบว่าระบบได้รับ Packet จากหน่วยควบคุมการเชื่อมต่อเครือข่ายหรือไม่ หากไม่ได้รับก็จะทำการประมวลผลงานในระดับ Application อันได้แก่ซอฟต์แวร์เว็บเซิร์ฟเวอร์ ทั้งนี้หากต้องการให้ระบบทำงานอื่นๆ สามารถแทรกการทำงานเพิ่มเติมได้ที่ตำแหน่งนี้ ในกรณีที่มี Packet รออยู่ในหน่วยความจำ ซอฟต์แวร์จะประมวลผล Packet ที่รอบอยู่นั้น ตามแนวการทำงานดังอธิบายข้างต้น ซึ่งซอฟต์แวร์หลักจะประมวลผลเช่นนี้เรื่อยไป

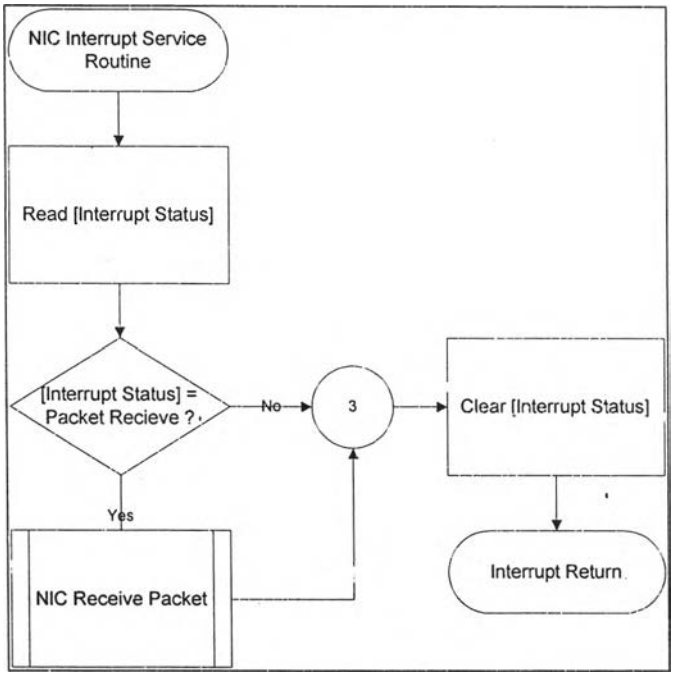
4.1. ซอฟต์แวร์ติดต่อหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย

การเชื่อมต่อระบบเครือข่ายตามมาตรฐานอีเทอร์เน็ตนั้น ในที่นี้ได้อาศัยคุณสมบัติของหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย ดังนั้นหน้าที่ของซอฟต์แวร์คือการรับและส่งข้อมูลระหว่างหน่วยความจำชั่วคราวของหน่วยควบคุมดังกล่าว และหน่วยความจำของหน่วยประมวลผลหลักเพื่อให้หน่วยประมวลผลหลักทำการประมวลผลต่างๆ

ตามมาตรฐานการทำงานของหน่วยควบคุมการเชื่อมต่อระบบเครือข่ายนั้น การโอนถ่ายข้อมูลเข้าและออกจะต้องการทำผ่านระบบการเข้าถึงหน่วยความจำโดยตรง (Direct Memory Access) เท่านั้น การทำงานของซอฟต์แวร์ในส่วนนี้จึงเป็นเพียงการสื่อสารเพื่อโอนย้ายข้อมูล ที่ได้ทำการถอดระหัสจากหน่วยประมวลการเชื่อมต่อระบบเครือข่ายแล้ว และส่งข้อมูลที่ต้องการออกสู่ระบบเครือข่ายให้กับหน่วยควบคุมผลการเชื่อมต่อระบบเครือข่ายเท่านั้น โดยอาศัยคุณสมบัติของวงจรเข้าถึงหน่วยความจำโดยตรงดังแสดงในบทที่ 3 หน้าที่ของซอฟต์แวร์ในส่วนนี้รวมถึงการจัดการสัญญาณขัดจังหวะ (Interrupt) ของหน่วยควบคุมผลพิเศษทางระบบเครือข่ายด้วย จากลักษณะการทำงานดังกล่าว ซอฟต์แวร์เพื่อช่วยประมวลผลในระดับนี้ จึงถูกจำแนกออกเป็นโปรแกรมย่อย 3 ส่วน ซึ่งได้แก่ โปรแกรมย่อยเพื่อให้บริการสัญญาณขัดจังหวะ (Interrupt Service Routine) โปรแกรมย่อยเพื่อรับข้อมูลจากระบบเครือข่าย และ โปรแกรมย่อยเพื่อการส่งข้อมูลเข้าสู่ระบบเครือข่าย ซึ่งการทำงานของโปรแกรมย่อยแต่ละส่วนนั้น สามารถแสดงได้ดังภาพที่ 16, 17 และ 18 ตามลำดับ

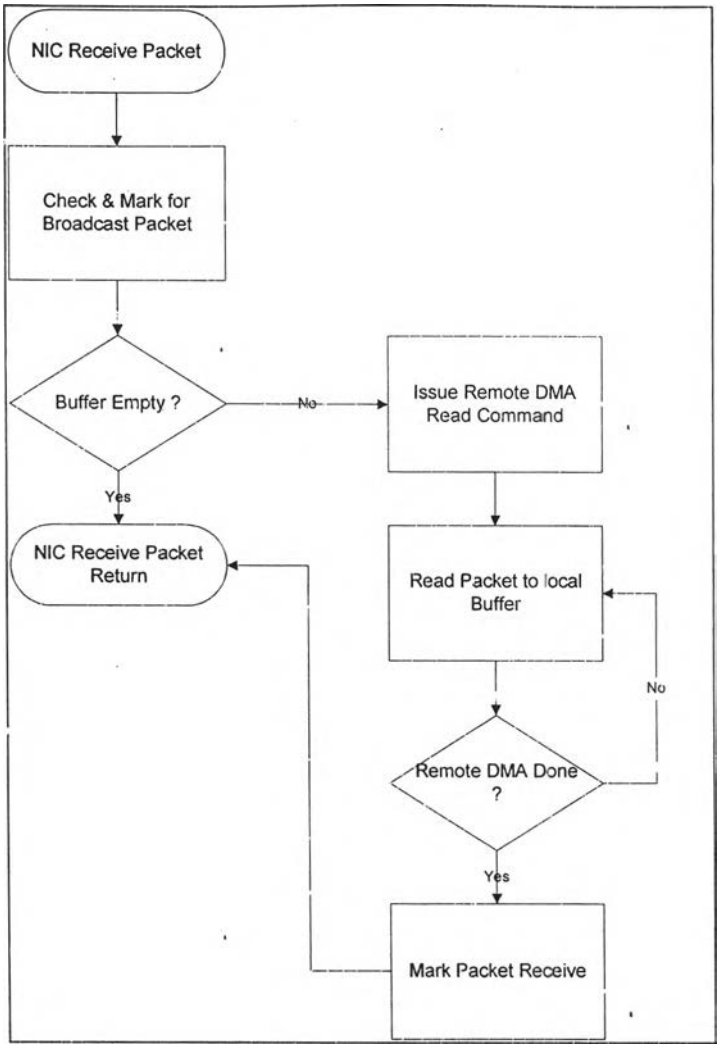
เมื่อได้รับสัญญาณขัดจังหวะ (Interrupt) จากหน่วยประมวลผลการเชื่อมต่อระบบเครือข่าย ระบบจะทำการเรียกใช้งานโปรแกรมย่อยเพื่อให้บริการสัญญาณขัดจังหวะ (Interrupt Service Routine) ซึ่งในบริการดังกล่าวจะเริ่มต้นด้วยการ จำแนกประเภทของสัญญาณขัดจังหวะ ซึ่งหากเป็นสัญญาณขัดจังหลวะของการรับข้อมูล โปรแกรมจะตรวจสอบว่ามี

Packet ค้างอยู่ในหน่วยความจำชั่วคราวของหน่วยควบคุมการเชื่อมต่อระบบเครือข่ายหรือไม่ หากยังมีข้อมูลตกค้างอยู่จึงย้ายข้อมูลผ่านระบบเข้าหน่วยความจำโดยตรง มายังหน่วยความจำหลักของระบบโดยอาศัยการทำงานของโปรแกรมย่อยเพื่อรับข้อมูล จากนั้นจึงลบค่าสถานะของสัญญาณขัดจังหวะ เพื่อรอรับสัญญาณขัดจังหวะครั้งต่อไป (ดูการทำงานประกอบในภาพที่ 16)



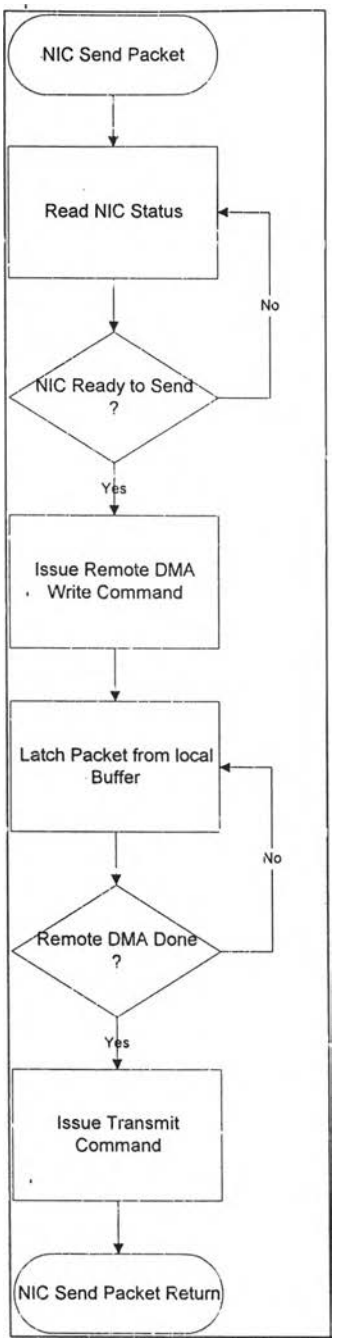
ภาพที่ 16 โปรแกรมย่อยเพื่อให้บริการสัญญาณขัดจังหวะ

การทำงานในโปรแกรมย่อยเพื่อรับข้อมูลจากระบบเครือข่ายนั้น สามารถอธิบายได้ตามภาพที่ 17 ดังนี้ เริ่มต้นด้วยการเก็บสถานะของข้อมูลที่ได้รับว่าเป็นข้อมูลประเภท Broadcast หรือไม่ ทั้งนี้เพื่อการใช้งานในโปรแกรมหลัก จากนั้นโปรแกรมจะตรวจสอบว่า มีข้อมูลอยู่ภายในหน่วยความจำชั่วคราวของหน่วยควบคุมการเชื่อมต่อระบบเครือข่ายหรือไม่ หากยังคงมีอยู่โปรแกรมจะร้องขอหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย ให้มีการย้ายข้อมูลเข้าสู่หน่วยประมวลผลหลัก โดยการส่งคำสั่ง Remote DMA Read และรอรับข้อมูลจนเสร็จสิ้น เมื่อเสร็จสมบูรณ์ จึงทำการแจ้งให้ซอฟต์แวร์ระดับที่สูงกว่าหรือโปรแกรมหลักทราบโดยตั้งค่าตัวแปร Packet Receive



ภาพที่ 17 โปรแกรมย่อยเพื่อรับข้อมูลจากระบบเครือข่าย

ในกรณีของการส่งข้อมูลนั้น เมื่อมีการเรียกขอจากซอฟต์แวร์ระดับที่สูงกว่าให้ส่งข้อมูล โปรแกรมย่อยเพื่อส่งข้อมูลเข้าสู่ระบบเครือข่ายจะตรวจสอบความพร้อมในการส่งข้อมูลของหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย และรอจนกว่าจะพร้อมส่งข้อมูล จากนั้นเมื่อหน่วยควบคุมพร้อมที่จะส่งข้อมูล โปรแกรมจะย้ายข้อมูลที่ต้องการจะส่งจากหน่วยความจำหลัก ไปยังหน่วยความจำชั่วคราวภายของหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย โดยการส่งคำสั่ง Remote DMA Write ให้กับหน่วยควบคุมการเชื่อมต่อระบบเครือข่าย และส่งค่าจนเสร็จสมบูรณ์ จากนั้นจึงสั่งให้หน่วยควบคุมการเชื่อมต่อระบบเครือข่ายส่งข้อมูลให้ ซึ่งหน่วยควบคุมการเชื่อมต่อระบบเครือข่ายจะสร้าง Frame Check sum และ ส่งข้อมูลเข้าสู่ระบบเครือข่ายดังอธิบายแล้วในบทที่3 เมื่อเสร็จทุกขั้นตอนดังกล่าว ถือเป็นการเสร็จสิ้นการทำงานส่งข้อมูล

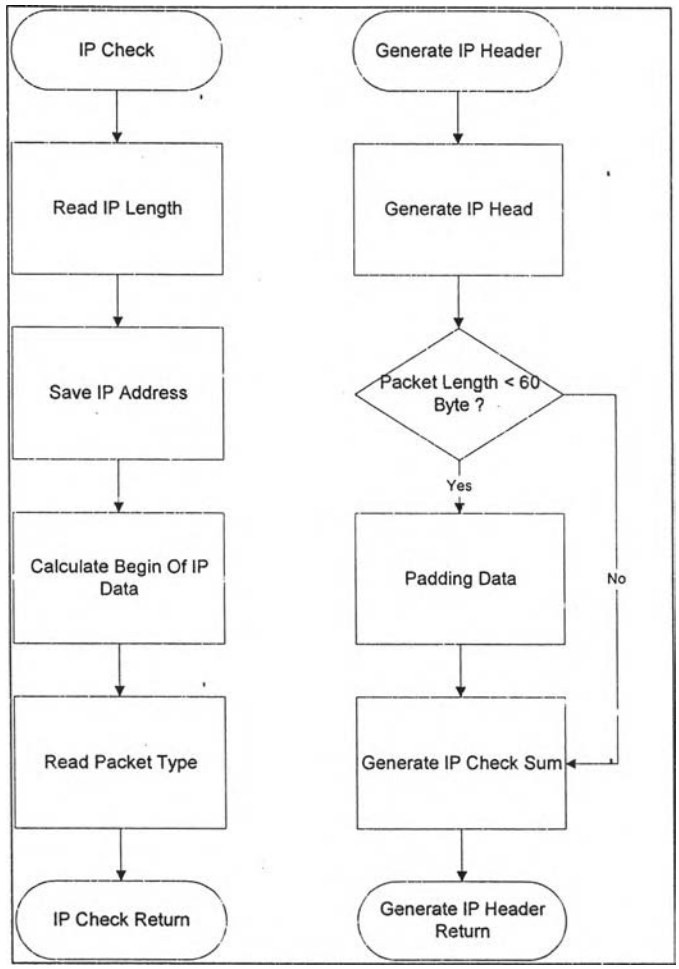


ภาพที่ 18 โปรแกรมย่อยเพื่อการส่งข้อมูลเข้าสู่ระบบเครือข่าย

4.2. IP

หน้าที่การทำงานของระบบซอฟต์แวร์ตามมาตรฐาน IP นั้น ประกอบด้วยการสร้าง IP Packet เพื่อระบุต้นทางและปลายทาง การวัดความยาวของข้อมูลที่ส่ง และการตรวจสอบความถูกต้องของหัว IP โดยสามารถแบ่งการทำงานตามทิศทางของการไหลของข้อมูลได้เป็นการรับและการส่งข้อมูล ซึ่งโปรแกรมย่อยที่เกี่ยวกับการทำงานดังกล่าว สามารถอธิบายได้ดังภาพที่ 19 ดังนี้ ด้านซ้ายแสดงการทำงานของโปรแกรมย่อยเพื่อตรวจสอบความถูกต้องของข้อมูลที่ได้

รับ และ ด้านขวาแสดงการทำงานของโปรแกรมย่อยเพื่อการสร้าง IP Header สำหรับใช้ในการส่งข้อมูล



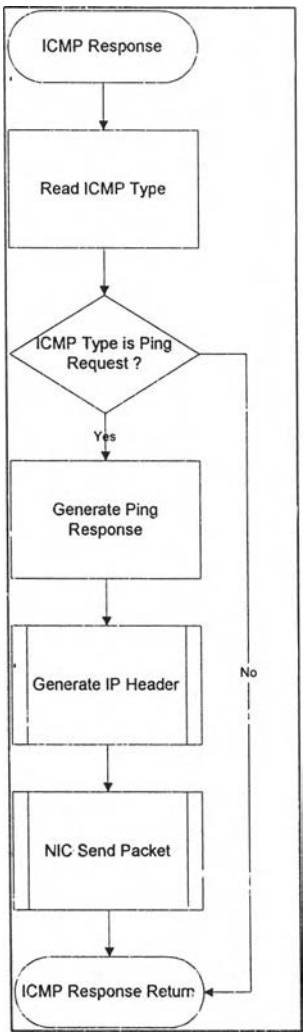
ภาพที่ 19 โปรแกรมย่อยในการประมวลผล IP

กรณีของการรับข้อมูล เมื่อระบบได้รับข้อมูลจากหน่วยประมวลผลระบบเครือข่าย ระบบจะคำนวณหาความยาวและตำแหน่งเริ่มต้นของข้อมูล และจำแนกประเภท Packet ว่าเป็น TCP หรือ ICMP เพื่อให้โปรแกรมหลักประมวลผลต่อไป

กรณีการส่งข้อมูล เมื่อ ICMP หรือ TCP ต้องการส่งข้อมูล IP จะทำการสร้างหัว IP เพื่อระบุต้นทางและปลายทางของข้อมูล ทั้งนี้หากข้อมูลที่ต้องการส่งมีขนาดสั้นเกินไป โปรแกรมจะทำ Padding หรือ เติมข้อมูลให้มีความยาวเพียงพอ จากนั้นจะทำการคำนวณหาความยาวทั้งหมดของข้อมูลที่ต้องการจะส่ง คำนวณ และบันทึกค่า IP Check sum ในส่วนหัวของ IP บน Packet ที่จะส่งออก

4.3. ICMP

ICMP นั้นโดยทั่วไปจะใช้ตรวจสอบสถานะของระบบเครือข่าย IP แต่ที่นี้จะทำหน้าที่เพียง Ping Response เพื่อตอบรับข้อมูลประเภท Ping ของ ICMP เท่านั้นการตอบรับข้อมูลประเภท ICMP เป็นเพียงการสลับผู้รับและผู้ส่งในส่วนหัวของ Packet เท่านั้น ดังนั้นการทำงานของซอฟต์แวร์ในส่วนนี้จึงไม่มีความซับซ้อนใดๆ เป็นเพียงการสลับที่ข้อมูลตามปกติ และสามารถอธิบายได้ดังภาพที่ 20

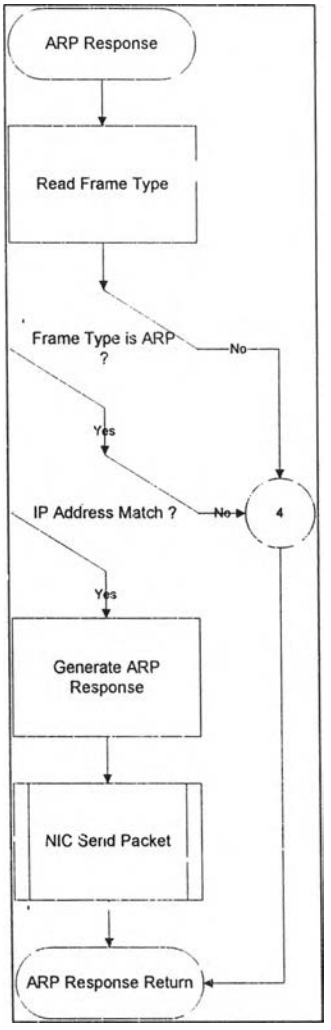


ภาพที่ 20 โปรแกรมย่อมเพื่อประมวลผล ICMP

การทำงานของโปรแกรมย่อยเพื่อสร้าง Ping Response ของระบบเว็บเซิร์ฟเวอร์แบบฝังตัวในที่นี้ เริ่มจากการตรวจสอบว่าข้อมูลที่ได้รับนั้น เป็นข้อมูลประเภท Ping Request หรือไม่ หากเป็นระบบจึงจะสร้าง Ping Response และ ร้องขอให้หน่วยควบคุมการเชื่อมต่อระบบเครือข่ายทำการส่งข้อมูลเข้าสู่ระบบเครือข่าย

4.4. ARP

ในที่นี้หมายถึงการตอบรับข้อมูลประเภท ARP อันเป็นการถามหาหมายเลข MAC Address ของหมายเลข IP ที่กำหนด โดยโปรแกรมในส่วนนี้จะตรวจสอบว่าข้อมูลที่ได้รับนั้นมีหมายเลข IP ตรงกับหมายเลข IP ของระบบหรือไม่ หากตรงกันระบบจะตอบหมายเลข IP ดังกล่าวด้วย MAC Address ของตน หากไม่ตรงกัน ระบบจะทำการละทิ้งข้อมูลชุดนั้นไป ซึ่งการทำงานดังกล่าวสามารถอธิบายเป็นแผนภาพและลำดับการทำงานได้ภาพที่ 21

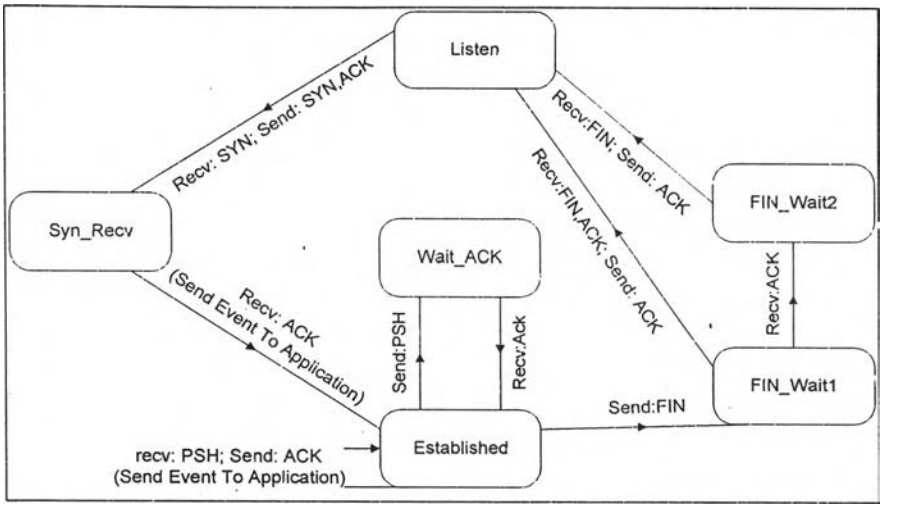


ภาพที่ 21. โปรแกรมย่อยเพื่อประมวลผล ARP

4.5. TCP

ระบบ TCP นั้นมีการทำงานที่ค่อนข้างซับซ้อนทั้งนี้เนื่องจากต้องมีการตรวจสอบสถานะและความถูกต้องของข้อมูลที่ทำการสื่อสารตลอดเวลา ในที่นี้การทำงานของ TCP จะเป็น

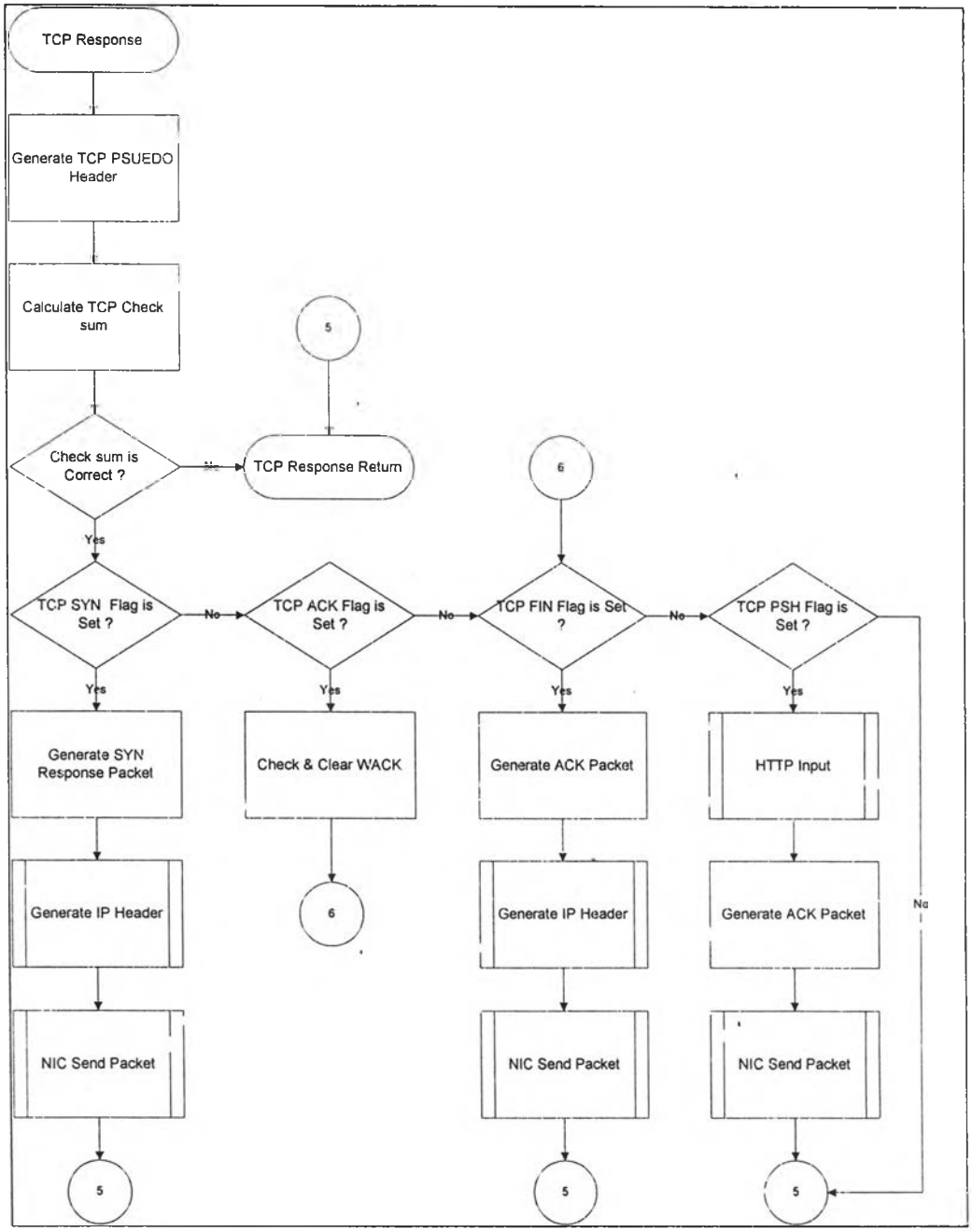
การทำงานในลักษณะของเซิร์ฟเวอร์ซึ่งจะรอฟังสัญญาณการร้องขอจากผู้ใช้งาน ลักษณะการทำงานดังกล่าวสามารถอธิบายโดยย่อได้ดังภาพที่ 22



ภาพที่ 22 สถานะการทำงานของ TCP บนระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

จากภาพที่ 22 จะพบว่าเมื่อมีการรับข้อมูลทุกครั้ง โปรแกรมจะตรวจสอบความถูกต้องของข้อมูลที่ได้รับและตอบรับด้วยการส่ง ACK เสมอ ซึ่งการส่ง ACK อาจส่งเพียง ACK เดี่ยวๆ หรือ ส่ง ACK ไปพร้อมกับการส่งข้อมูล (SYN, FIN หรือ PSH) ก็ได้ เมื่อเริ่มต้นระบบ TCP จะอยู่ในสถานะ Listen หากข้อมูลที่ได้รับเป็น SYN โปรแกรมจะตอบรับด้วย SYN และ ACK การได้รับข้อมูล จากนั้นรอจนกว่าจะได้รับข้อมูล ACK เพื่อยืนยันการ SYN ของเซิร์ฟเวอร์เมื่อเสร็จสิ้นตามขั้นตอนดังกล่าวจึงจะถือว่าการเปิด Session สมบูรณ์ ซึ่งที่สถานะนี้ระบบพร้อมสำหรับการรับและส่งข้อมูล โดยกรณีที่ได้รับข้อมูลโปรแกรมจะเก็บข้อมูลลงสู่หน่วยความจำและส่ง ACK เพื่อยืนยันการได้รับข้อมูล ส่วนกรณีการส่งนั้น ระบบจะส่งข้อมูลและรอคอยจนกว่าจะได้รับ ACK จึงจะทำงานต่อไป เมื่อเสร็จสิ้นการสื่อสาร โปรแกรมจะส่ง FIN และรอคอยจนกว่าจะได้รับสัญญาณ ACK และ FIN ตอบรับ เป็นการเสร็จสิ้นการสื่อสารข้อมูล นอกจากนี้เพื่อป้องกันความผิดพลาดที่เกิดขึ้นจากการสื่อสาร จึงมีการกำหนดเวลาในการสื่อสารแต่ละครั้ง ไม่เกิน 15 วินาทีโดยเริ่มจับเวลาตั้งแต่มีการเริ่ม Established หากระบบไม่มีการเปลี่ยนสถานะเป็น Listen ภายใน 15 วินาที โปรแกรมจะยกเลิกการสื่อสารในครั้งนั้นทั้งหมด ทั้งนี้อาศัยการทำงานของ Timer Interrupt ช่วยในการประมวลผลดังกล่าว (รายละเอียดเพิ่มเติมค้นคว้าได้จากคู่มือ MCS-51 ทั่วไป) การทำงานของซอฟต์แวร์ส่วนที่เกี่ยวข้องกับการทำงานดังกล่าวข้างต้นสามารถอธิบายได้ดังภาพที่ 23

เมื่อเริ่มต้นการประมวลผลเพื่อการตอบรับข้อมูลประเภท TCP ระบบจะตรวจสอบความถูกต้องของข้อมูล TCP โดยการสร้าง PSUEDO HEADER และคำนวณ Check sum ของข้อมูลที่ได้รับ จากนั้นจึงประมวลผลตาม Flag ของข้อมูลที่ได้รับ ได้แก่ SYN, ACK, FIN และ PSH



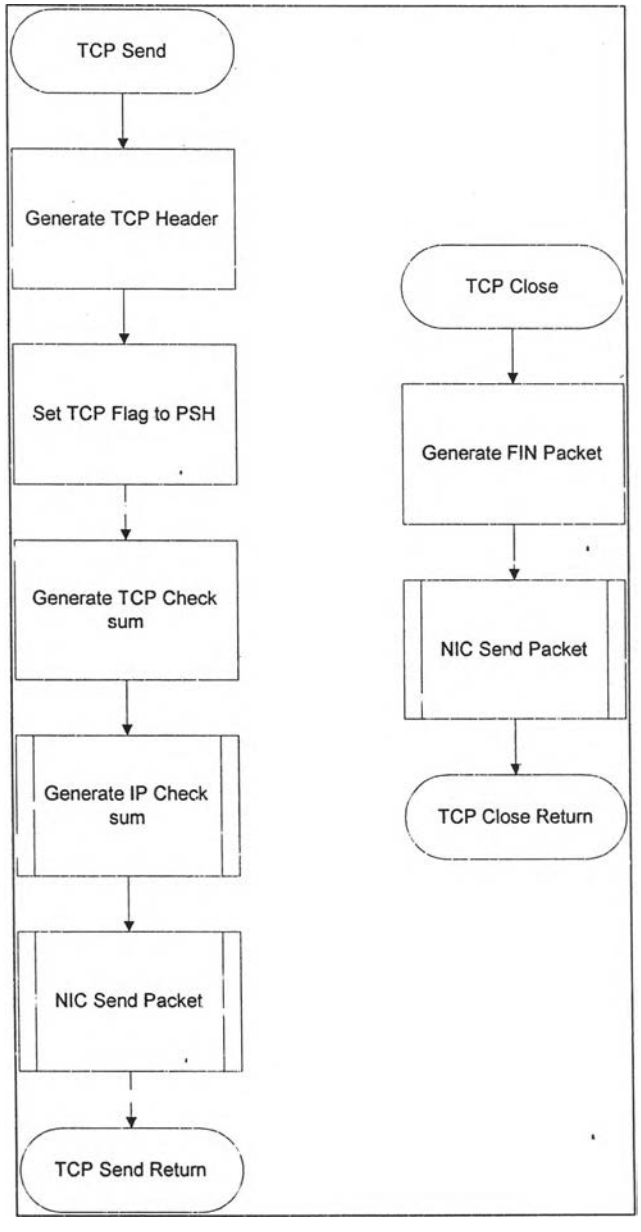
ภาพที่ 23 โปรแกรมย่อยเพื่อประมวลผล TCP

เมื่อซอฟต์แวร์ในระดับที่สูงกว่า ต้องการส่งข้อมูลบน TCP จะร้องขอให้ TCP ส่ง ซึ่งในขั้นตอนนี้ โปรแกรม TCP Send จะทำการสร้าง TCP Header โดยกำหนด Flag เป็น PSH สำหรับการส่งข้อมูล สร้าง TCP Check sum จากนั้นจึงร้องขอให้ IP สร้าง Check sum เมื่อเสร็จ

I19423433

สิ้นการทำงานดังกล่าว จึงขอให้ซอฟต์แวร์ Driver ทำการส่งข้อมูลออกสู่ระบบเครือข่าย (ดังแสดงในภาพที่ 24 ซ้าย)

หลักจากเสร็จสิ้นการทำงานทั้งหมด ซอฟต์แวร์เซิร์ฟเวอร์จะต้องร้องขอให้ TCP ปิดการสื่อสาร โดยการเรียกโปรแกรมย่อย TCP Close ซึ่งการทำงานของโปรแกรมดังกล่าวประกอบด้วยการสร้าง TCP FIN Packet และส่งออกสู่ระบบเครือข่าย ดังแสดงในทางขวาของภาพที่ 24



ภาพที่ 24 โปรแกรมย่อยเพื่อส่งข้อมูลบน TCP และปิดการสื่อสารบน TCP

อย่างไรก็ตาม การสร้างซอฟต์แวร์ทั้งหมด เป็นเพียงต้นแบบ มิได้มีการสร้างหรือพัฒนา TCP อย่างสมบูรณ์แบบ หากแต่เป็นเพียงการพัฒนาเพียงซอฟต์แวร์ส่วนที่จำเป็นสำหรับการใช้งานเพื่อเป็นเซิร์ฟเวอร์แบบฝังตัวออกมาเท่านั้น ซอฟต์แวร์การสื่อสาร TCP/IP ในที่นี้จึงไม่สามารถทำงานซับซ้อน เช่น IP fragment หรือ TCP Sliding Window ได้