บทที่ 3

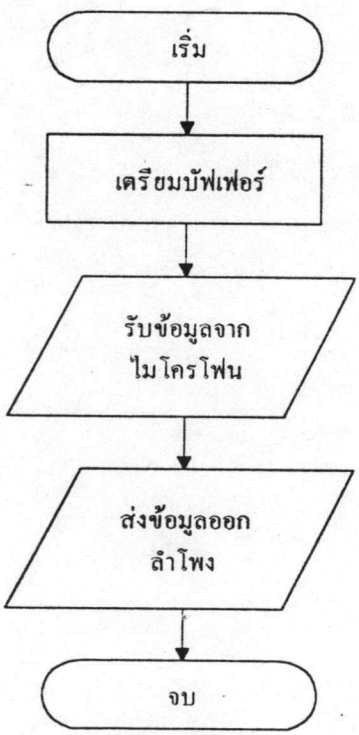
การพัฒนาระบบ

ในการพัฒนาโปรแกรมนี้มีเรื่องที่เกี่ยวข้องกันอยู่ 2 ส่วนใหญ่คือ เรื่องของการรับส่งเสียง และเรื่องของการส่งข้อมูลในเครือข่าย นอกจากนี้ ยังมีเรื่องของการพัฒนาส่วนควบคุมต่างๆ เข้ามาเกี่ยวข้องด้วย ดังนั้น จึงได้แบ่งการพัฒนาระบบ ออกเป็นส่วนๆดังต่อไปนี้

- การพัฒนาโปรแกรมในส่วนของการรับและส่งเสียง
- การพัฒนาโปรแกรมในส่วนของการรับและส่งข้อมูลในเครือข่าย
- นำโปรแกรมส่วนของการรับส่งเสียงและส่วนของการรับส่งข้อมูลในเครือข่ายมารวมกัน
- เพิ่มเติมส่วนของการควบคุมต่างๆ

3.1การพัฒนาโปรแกรมในส่วนของการรับและส่งเสียง

ขั้นตอนนี้เริ่มตั้งแต่ ทดลองศึกษาการทำงานของ ไดร์เวอร์ของการ์ดเสียง แล้วจึงทดลองรับเสียงจากไมโครโฟนเข้ามาเก็บในบัฟเฟอร์ จากนั้นก็ทดลองส่งข้อมูล จากบัฟเฟอร์ของคอมพิวเตอร์ออกไปยังลำโพง การทดลองตามจุดประสงค์ดังกล่าวนี้ ต้องดำเนินไปด้วยกันจึงจะทดสอบได้ เช่น ถ้าเรารับเสียงเข้ามาแล้ว ไม่ส่งออกลำโพง แล้วลองฟังดู เราก็จะไม่ทราบว่าข้อมูลที่รับมาถูกต้องหรือไม่ ในขั้นนี้ทดลองโดยใช้เครื่องเดียวเท่านั้น จึงไม่เกี่ยวข้องกับเครือข่าย โดยมีขั้นตอนการทำงานดังนี้



รูป 3.1 การรับและส่งเสียงโดยใช้เครื่องเดียว

การรับข้อมูล ใช้ฟังก์ชั่นหมายเลข 7 (Start Voice Input) และการส่งข้อมูล ออกลำโพง ใช้ฟังก์ชั่นหมายเลข 6 (Start Voice Output) สำหรับการทดลองในขั้นนี้ผ่านไปด้วยดี ไม่พบปัญหาใดๆ นอกจากนี้ยังได้ทดลองรับข้อมูล โดยอ่านจากแฟ้มเสียง ที่มีนามสกุลเป็น VOC แทนที่จะรับข้อมูลจากไมโครโฟน พบว่า ถ้าอ่านข้อมูล มาจากแฟ้มเสียง จุดเริ่มต้นในเนื้อหาที่ได้จะไม่ใช้ข้อมูลจริง แต่จะเป็นเฮดเดอร์ของไฟล์ ซึ่งมีข้อความว่า "Creative Voice File" และตามด้วยรหัส บางอย่างอีก 8 ไบต์ ซึ่งเมื่อเราเปิดแฟ้มเสียงมา เราสามารถตรวจสอบข้อมูลตรงจุดนี้ได้ว่า เป็นแฟ้มเสียงแบบ VOC ของการ์ดซาวนด์บลาสเตอร์หรือไม่ เนื่องจากช่วงแรกเป็นข้อมูลที่ใช้เพื่อตรวจสอบ จึงไม่ใช่เสียง ถ้าเราจะอ่านข้อมูลเพื่อส่งเสียงออกลำโพง จึงต้องเลื่อนตัวชี้ข้อมูล ไปให้พ้นส่วนเฮดเดอร์ (27 ไบต์) ถ้าเราเริ่มอ่านข้อมูล ผิดที่ เสียงที่ออกมาจะฟังไม่รู้เรื่อง

สิ่งที่ต้องพิจารณาอีกอย่างก็คือ คุณภาพของเสียง เช่น ถ้าเราใช้อัตราการชักตัวอย่างน้อยๆ เราก็จะได้เสียงที่ไม่มีคุณภาพเท่าที่ควร เนื่องจากการ์ดเสียงที่ใช้ในการทำวิทยานิพนธ์นี้ ใช้เทคนิคการแปลงสัญญาณแบบ Pulse Code Modulation ดังนั้น อัตราการชักตัวอย่างจึงมีผลอย่างมากต่อคุณภาพเสียง เราสามารถเซ็ตค่า อัตราการชักตัวอย่าง ได้ตั้งแต่ 4 กิโลเฮิรตซ์ ถึง 45 กิโลเฮิรตซ์ ในการทดลองพบว่า อัตรา 4000 เฮิรตซ์ ก็ถือว่าใช้ได้ เพราะเราสนใจแค่เสียงพูดของคน ไม่ได้ต้องการคุณภาพสูง เหมือนเสียงร้องเพลงหรือเสียงดนตรี เพราะฉะนั้น 4 กิโลเฮิรตซ์ ก็ถือว่าคุณภาพเสียงอยู่ในระดับที่ยอมรับได้ การที่ต้องพยายามใช้ค่าการชักตัวอย่างน้อยๆ ก็เพราะต้องการให้มีข้อมูลน้อยที่สุด เพราะเมื่อต่อกับระบบเครือข่ายแล้ว จะต้องส่งเสียงลงไปในเครือข่าย ซึ่งถ้าใช้อัตราการชักตัวอย่างสูงๆ จำนวนข้อมูลก็จะมากขึ้น เราอาจมีปัญหา เช่น ระบบเครือข่ายรับข้อมูลไม่ทัน ทำให้มีข้อมูลตกหล่นสูญหายได้ หรืออาจทำงานได้ แต่มีผลทำให้ระบบโดยรวมมีประสิทธิภาพต่ำลง เช่น ทำให้ผู้ที่ใช้เครือข่าย ที่ใช้งานโปรแกรมอื่นๆอยู่ ทำงานได้ช้าลง เพราะเราต้องส่งข้อมูลจำนวนมาก สำหรับรายละเอียด เกี่ยวกับการใช้ฟังก์ชั่นในการรับเสียงเข้าหรือส่งเสียงออก จะอยู่ในภาคผนวก ข

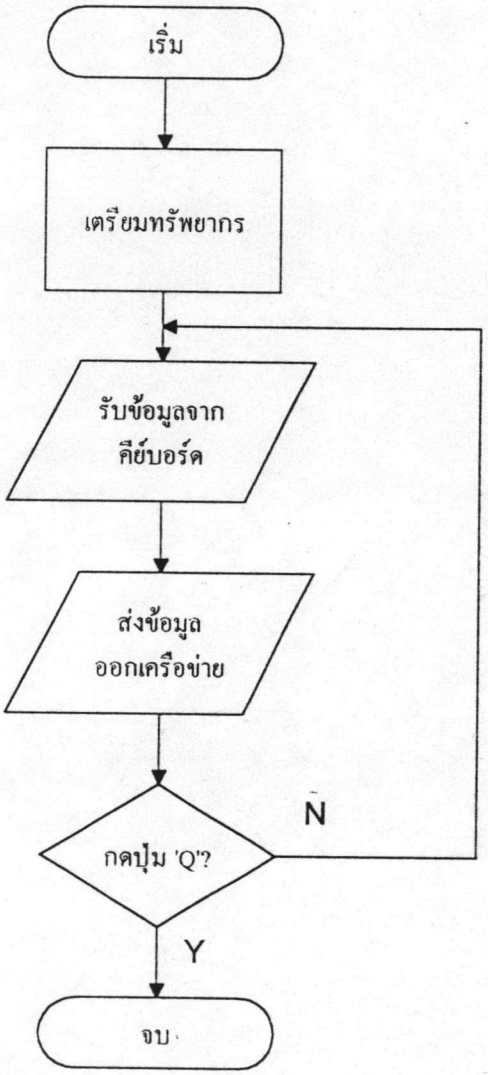
3.2การพัฒนาโปรแกรมในส่วนของการรับและส่งข้อมูลในเครือข่าย

เครือข่ายที่จะใช้ในการวิจัยนี้ เป็นระบบเครือข่ายเฉพาะที่แบบอีเทอเน็ต (Ethernet) แต่เนื่องจากในสถานที่ทดลอง มีเครือข่ายทั้งแบบอีเทอเน็ต และแบบโทเคนริง จึงได้ทำการทดลองและสังเกตผล กับเครือข่ายทั้งสองแบบเพื่อเปรียบเทียบกัน การทดลองในส่วนนี้ เริ่มต้นจากสิ่งที่ง่ายที่สุดก่อน คือส่งตัวอักษร จากเครื่องหนึ่งกระจาย (Broadcast) ไปยังเครื่องอื่นๆ โดยใช้ฟังก์ชั่น IPXSendPacket

i 17339169

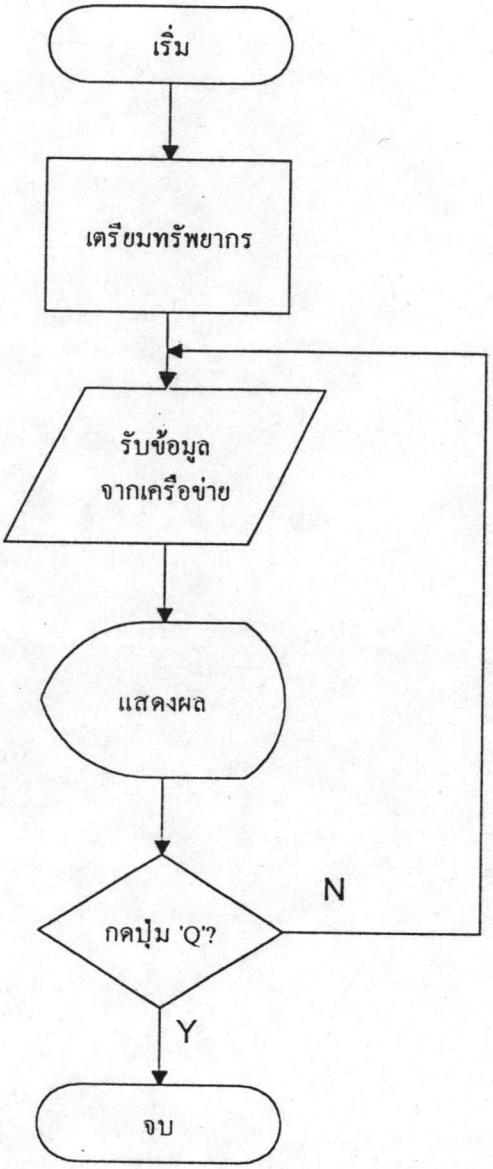
การที่ต้องใช้การส่งแบบกระจาย เป็นเพราะว่าลักษณะงานของการประชุม เราต้องการส่งเสียงจากผู้พูด ไปออกที่กลุ่มของผู้ฟังซึ่งมีหลายเครื่อง และเนื่องจาก IPX ไม่มีฟังก์ชั่นสำหรับส่งกระจายเฉพาะกลุ่ม ดังนั้นจึงต้องส่งข้อมูลกระจายลงเครือข่าย ให้ทุกๆเครื่องที่ต่ออยู่ ในเครือข่ายสามารถรับได้ แต่จะต้องเขียนโปรแกรมควบคุมไว้ ให้เครื่องที่มีสิทธิเท่านั้น ที่สามารถรับข้อมูลไปประมวลผลต่อได้ คำว่าเครื่องที่มีสิทธิในที่นี้หมายถึงเครื่องที่มีรหัสผ่าน สำหรับการประชุมถูกต้อง

การส่งตัวอักษรมีขั้นตอนดังต่อไปนี้



รูป 3.2 การส่งตัวอักษรเข้าเครือข่าย

การรับตัวอักษรมีขั้นตอนดังต่อไปนี้



รูป 3.3 การรับตัวอักษรจากเครือข่าย

สำหรับการเตรียมทรัพยากร ในที่นี้หมายถึงการเตรียมบัฟเฟอร์ โครงสร้าง ตลอดจนตัวแปรต่างๆ ที่จำเป็น ให้พร้อมสำหรับการรับและส่งข้อมูล เช่นต้องจองเนื้อที่ สำหรับโครงสร้าง ECB (Event Control Block) และโครงสร้าง IPX Header แล้วกำหนดค่าต่างๆ ให้เหมาะสม สำหรับการส่งและการรับ เป็นต้นว่า พื้นที่ซึ่งใช้เก็บข้อมูล (สำหรับการส่งและการรับ) อยู่ที่ไหน บัฟเฟอร์ยาวเท่าไร

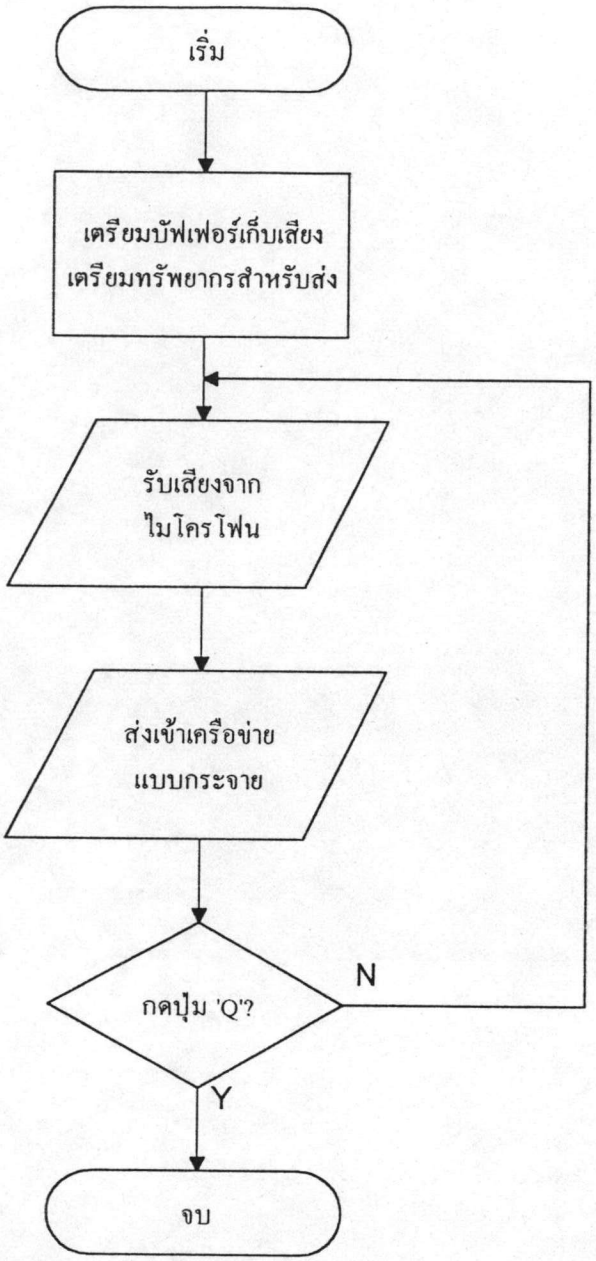
นอกจากการเตรียมตัวแปรต่างๆ แล้ว ยังต้องเตรียมโปรแกรมรองรับบางส่วนเช่น เตรียม ESR (Event Service Routine) ไว้เพื่อบอกเครื่องว่า ถ้ามีแพ็คเก็ตเข้ามาจะให้ทำอย่างไรต่อไป รายละเอียดของการเตรียมการต่างๆ เพื่อการส่งและรับข้อมูลโดยใช้ฟังก์ชั่น ของ IPX นี้ ได้กล่าวไว้แล้ว ในบทการเขียนโปรแกรมโดยใช้ฟังก์ชั่นของ IPX

เมื่อทดลองรับส่งข้อมูลทีละ 1 ตัวอักษรได้ผล ต่อไปก็ขยายผลโดยทำการทดลองส่งตัวอักษรเป็นชุด (Character Array) ซึ่งก็ได้ผลดีเช่นกัน

3.3นำโปรแกรมส่วนการรับส่งเสียงและส่วนของการรับส่งข้อมูลในเครือข่ายมารวมกัน

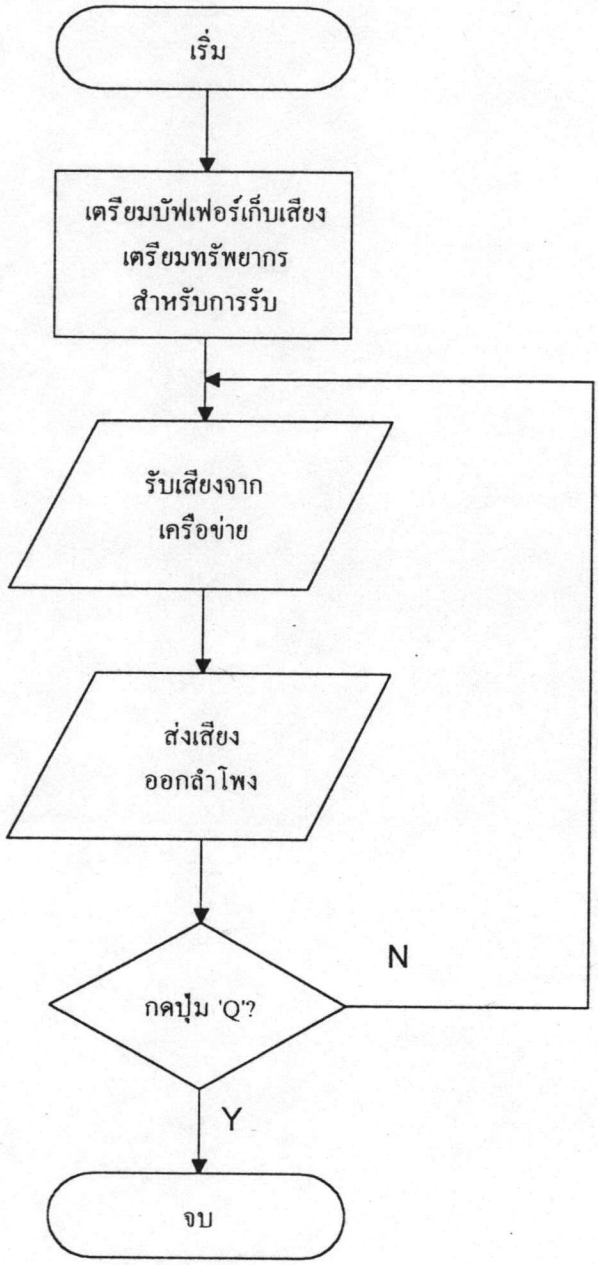
ในขั้นตอนที่หนึ่ง เราสามารถอัดเสียงเข้าสู่คอมพิวเตอร์ได้ และ สามารถเล่นเสียง กลับได้โดยใช้เครื่องเดียว ในขั้นตอนที่สอง เราสามารถส่งข้อมูล ที่เป็นชุดของตัวอักษร จากเครื่องหนึ่งกระจายออกไปยังเครื่องอื่นๆได้ โดยผ่านเครือข่ายเฉพาะที่ สำหรับขั้นตอนที่สามนี้ พยายามจะรวม 1 และ 2 เข้าด้วยกัน เพื่อให้สามารถส่งเสียงจากเครื่องส่ง (เครื่องต้นทาง) ไปยังกลุ่มเครื่องรับ (เครื่องปลายทาง) ได้

โปรแกรมสำหรับส่ง มีขั้นตอนดังนี้



รูป 3.4 การส่งเสียงเข้าเครือข่าย

โปรแกรมสำหรับรับมีขั้นตอนดังนี้



รูป 3.5 การรับเสียงจากเครือข่าย

ในการทดลอง ได้ใช้กับเครือข่ายเฉพาะที่แบบโทเคนริงก่อน ซึ่งได้ผลดี คือ เครื่องรับสามารถรับเสียงได้ และมีคุณภาพเสียง ที่รับฟังได้ชัดเจน ถึงแม้จะใช้อัตราการชักตัวอย่าง แค่ 4 กิโลเฮิรตซ์ จากภาพของขั้นตอนการส่ง จะเห็นได้ว่าเป็นการทำงานแบบวน สิ่งที่ต้องพิจารณาตรงจุดนี้คือ การที่รับเสียงจากไมโครโฟนเข้าบัฟเฟอร์นั้น จะใช้เวลานานเท่าไร เช่น รับเสียงแค่ 1 วินาที แล้วส่งออกเครือข่าย แล้ววนกลับไปรับเสียงใหม่ หรือจะรับเสียงครั้งละ 30

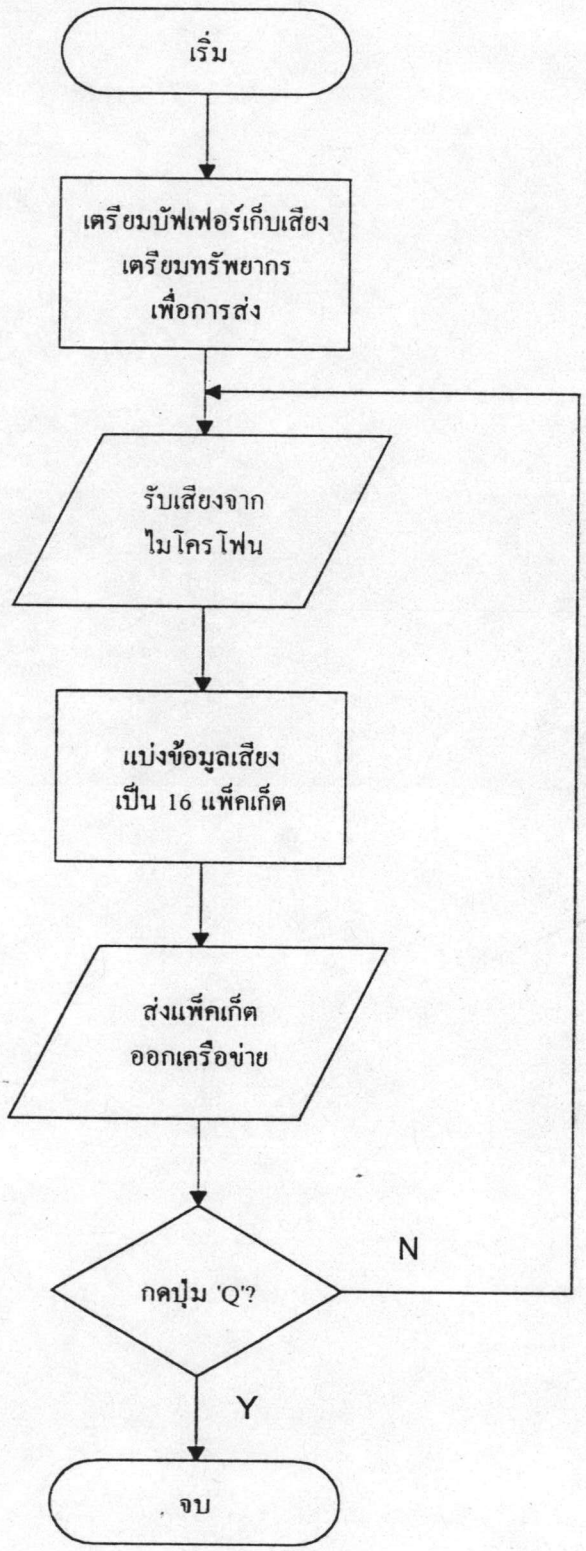
หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วินาทีแล้วส่งลงเครือข่าย ผลกระทบจุดนี้ก็คือ ถ้าใช้เวลาสั้นเกินไป เสียงจะมีการขาดช่วงบ่อยคือถ้าใช้ 1 วินาที เสียงก็จะมีการสะดุดทุกๆวินาที ทำให้คุณภาพเสียงไม่ดี แต่ถ้าใช้เวลานานๆเช่น 30 วินาที เสียงก็จะมีการสะดุดทุก 30 วินาที ซึ่งทำให้เสียงโดยรวมถือว่าดี ข้อเสียของการใช้เวลานานก็คือ เกิดการล่าช้าของเสียง คือสิ่งที่ผู้ฟังกำลังฟังอยู่ เป็นสิ่งที่ผู้พูดได้พูดไปแล้วเมื่อครึ่งนาทีก่อน ดังนั้น ถ้าหากเป็นการประชุมจริง และผู้ฟังต้องการจะขอแย้งในสิ่งที่ได้ฟัง ผู้พูดอาจจะเลยไปถึงประเด็นอื่น หรือเปลี่ยนเรื่องไปแล้ว เป็นต้น จากการทดลองใช้เวลาประมาณ 5 วินาที น่าจะเป็นค่าที่พอเหมาะ เพราะเสียงขาดช่วงไม่บ่อย ในขณะที่ ความล่าช้า (Delay) ก็น้อย จนกระทั่งเกือบเป็นเวลาจริง (Realtime)

เมื่อการทดลองในการส่งเสียง กระจายไปออกเครื่องอื่นๆได้ผล จึงได้ทดลอง กับเครือข่ายเฉพาะที่แบบอีเทอเน็ตบ้าง แต่ปรากฏว่าโปรแกรมรับส่งข้อมูลไม่ทำงาน โดยที่ไม่ทราบสาเหตุ จึงต้องย้อนกลับไปสู่ขั้นที่ 2 คือ ทดลองส่งข้อมูลทีละตัวอักษร และ ส่งชุดของตัวอักษร ปรากฏว่าสามารถส่งและรับได้ ต่อมาได้ทดลองส่งเสียง โดยใช้บัฟเฟอร์หลายๆขนาด ผลปรากฏว่า ถ้าใช้บัฟเฟอร์ ขนาดเล็กๆ เช่น 1000 ไบต์ จะสามารถใช้งานได้ แต่ถ้าใช้บัฟเฟอร์ขนาดใหญ่ เช่น 5000 ไบต์ จะส่งและรับไม่ได้ ซึ่งแสดงว่าขนาดของแพ็คเก็ต มีผลต่ออีเทอเน็ต เมื่อทดลองเปลี่ยนหลายๆ ค่าก็พบว่าค่าโดยประมาณที่ใช้ได้คือ 1500 ไบต์ และเมื่อค้นคว้าต่อไป ก็พบว่าขนาดของฟิลด์ข้อมูล (Data Field) ในเฟรมของอีเทอเน็ต จะต้องยาวไม่เกิน 1500 ไบต์ (MILLER, 1992) ซึ่งสอดคล้องกับการทดลอง ทั้งนี้อาจเป็นเพราะหลักการเข้าถึงตัวกลาง แบบ Carrier Sense Multiple Access/Collission Detect (CSMA/CD) อาจมีการชนกันของแพ็คเก็ตได้ ดังนั้นจึงต้องพยายาม ให้ขนาดของแพ็กเก็ต อยู่ในระดับที่เหมาะสม เพื่อที่จะมีการชนกัน ให้น้อยที่สุด

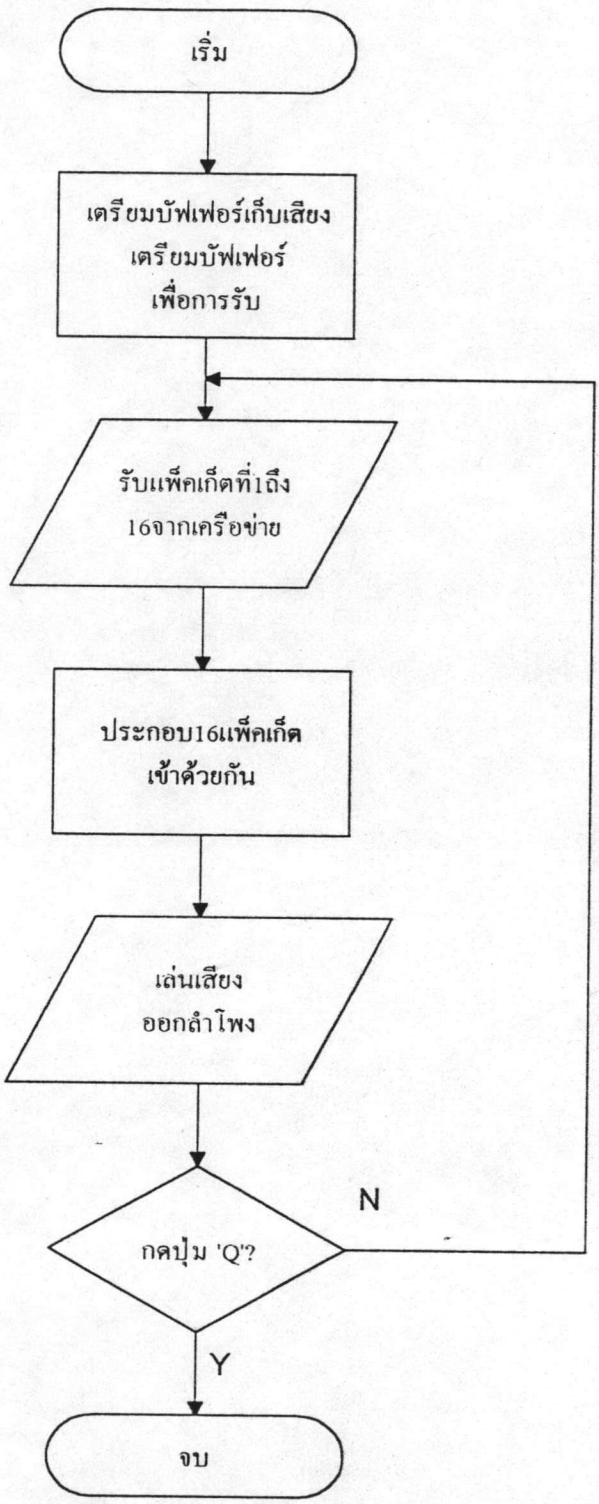
ปัญหาประการต่อมาก็คือ ถ้าเราใช้แพ็คเก็ตสั้นๆ (ในที่นี้ใช้ 1450 ไบต์) เสียงก็จะขาดช่วงบ่อยมาก ทำให้ผลที่ออกมาฟังแล้วไม่ต่อเนื่องและน่ารำคาญ ดังนั้น จึงต้องหาวิธี ทำให้ส่งข้อมูลได้ยาวขึ้น เทคนิคที่ใช้ก็คือ การใช้บัฟเฟอร์ขนาดใหญ่ รับข้อมูลเข้ามาครั้งละมากๆ แล้วแบ่งย่อยออกเป็นหลายๆแพ็คเก็ต (ในที่นี้ใช้ 16 แพ็คเก็ต) แล้วทยอยส่งออกไป ทางฝ่ายรับก็ต้องรับแพ็คเก็ตทั้งหมดแล้วนำมาประกอบกันเข้า เมื่อได้ครบแล้วจึงเล่นเสียงเพื่อส่งออกทางลำโพง

ขั้นตอนของการส่งมีดังนี้



รูป 3.6 การส่งเสียงเข้าเครือข่ายแบบใช้แพ็คเก็ตย่อย

ขั้นตอนของการรับมีดังนี้



รูป 3.7 การรับเสียงจากเครือข่ายแบบใช้แพ็คเก็ตย่อย

สิ่งที่ต้องพิจารณาก็คือ ข้อมูลต้องได้รับการประกอบกันอย่างถูกต้อง คือ แพ็คเก็ตที่หนึ่ง ตามด้วย สอง สาม...ไปจนถึง 16 สมมุติว่ามีการพลาดเกิดขึ้น คือ ผู้รับไม่สามารถรับแพ็คเก็ตที่หนึ่งได้

แต่ได้รับแพ็คเก็ตที่สอง แล้วคิดว่าแพ็คเก็ตที่สองเป็นแพ็คเก็ตที่หนึ่ง และสะสมข้อมูลเรื่อยไปจนถึงแพ็คเก็ตสุดท้าย ซึ่งที่จริงแล้วเป็นแพ็คเก็ตที่หนึ่งของข้อมูลชุดถัดไป ถ้ามีการผิดพลาดเช่นนี้ ข้อมูลเสียงที่ถูกเล่นออกมาจะฟังไม่รู้เรื่อง เพราะว่าตอนต้นของข้อมูลเสียงที่เราบันทึกได้ จะมีสัญลักษณ์บางอย่างเพื่อบอกว่า ข้อมูลที่ตามมาต่อไปนี้ มีลักษณะอย่างไร โมโน หรือสเตอริโอ อัตราการชักตัวอย่างเท่าไร เป็นต้น เพื่อว่าทางผู้รับจะได้เล่นเสียง ออกทางเอาต์พุตได้อย่างถูกต้อง ถ้าการรับผิดพลาดและตอนต้นของข้อมูลหายไป จะทำให้ขาดสิ่งที่บ่งชี้ถึงคุณลักษณะ ของเสียงที่ตามมา และยิ่งกว่านั้นจะทำให้ตีความผิดพลาดเพราะคิดว่า ข้อมูลในแพ็คเก็ตที่สอง จะเป็นตัวบ่งบอกเกี่ยวกับลักษณะของเสียงที่ตามมา ดังนั้นเมื่อเล่นกลับออกมาจึงฟังไม่รู้เรื่อง ถ้าเราต้องการแก้ปัญหานี้ โดยใช้การโต้ตอบกันไปมา เช่น ถ้าเครื่องที่รับ ไม่สามารถรับแพ็คเก็ตที่หนึ่งได้ (ในกรณีนี้แต่ละแพ็คเก็ตต้องมีหมายเลขบอกลำดับด้วย) ก็จะส่งข้อความไปบอกเครื่องส่งว่า กรุณาส่งแพ็คเก็ตที่หนึ่งมาใหม่ และเครื่องส่งก็จะส่งแพ็คเก็ตที่หนึ่งให้ใหม่ วิธีนี้นี้จะใช้ไม่ได้ เพราะจะทำให้เกิดการล่าช้า ของแพ็คเก็ตต่อๆมา และทำให้แพ็คเก็ตต่อๆมา สูญหายมากขึ้นเรื่อยๆ เพราะเครื่องส่งจะต้องกลับไปส่งแพ็คเก็ตเก่าๆ จึงไม่มีเวลามาบริการแพ็คเก็ตใหม่ๆ อีกเหตุผลหนึ่งที่ไม่น่าจะใช้การแก้ปัญหาแบบนี้ได้ก็คือ การส่งเป็นการส่งแบบกระจาย ดังนั้นถ้าผู้รับคนหนึ่งรับไม่ได้และร้องขอแพ็คเก็ตเก่า ผู้รับรายอื่นๆที่ไม่ได้ร้องขอก็จะได้รับแพ็คเก็ตเก่าด้วย โดยที่ตีความว่าเป็นแพ็คเก็ตใหม่ จากการทดลองพบว่า ถ้าเราไม่ส่งแพ็คเก็ตเร็วเกินไป ก็จะไม่มีปัญหา ดังนั้นในการส่งแพ็คเก็ต จำนวน 16 แพ็คเก็ตติดต่อกัน ต้องทำการถ่วงเวลาระหว่าง แต่ละแพ็คเก็ต โดยเวลาที่ใช้ได้สำหรับการถ่วง ต้องมีค่า 1 มิลลิวินาทีหรือมากกว่านี้ ซึ่งถ้าไม่ถ่วงเวลาเลย จะทำให้มีปัญหารับไม่ทันดังได้กล่าวไปแล้ว แต่ถ้าถ่วงเวลามากไป จะทำให้เครื่องต้องใช้เวลาตอนนั้นมาก และกลับไปรับเสียงครั้งต่อไปได้ช้า ทำให้การขาดหายของเสียง มีมากกว่าเดิม เช่น ถ้าใช้การถ่วงเวลา 30 มิลลิวินาที ระหว่างการส่ง 16 แพ็คเก็ต รวมเป็นเวลาทั้งหมด เท่ากับ 30 x 15 = 450 มิลลิวินาที ต่อการส่งหมดบัฟเฟอร์หนึ่งครั้ง แต่ถ้าใช้การถ่วงเวลา เท่ากับ 10 มิลลิวินาที การถ่วงเวลาทั้งหมดจะเป็น 10 x 15 = 150 มิลลิวินาที ต่อการส่งหมดบัฟเฟอร์หนึ่งครั้ง

3.4 เพิ่มเติมส่วนของการควบคุมต่างๆ

ในขั้นตอนที่สาม สามารถทำการส่งเสียงจากเครื่องส่งลงเครือข่ายให้เครื่องรับ ทำการรับได้ แต่โปรแกรมสำหรับเครื่องส่งจะส่งได้เท่านั้น และโปรแกรมสำหรับเครื่องรับ จะรับได้เท่านั้น ในขั้นตอนนี้ จะรวมโปรแกรมส่งและรับเข้าไว้ด้วยกัน เพราะในการประชุม มีทั้งการพูดและการฟัง ดังนั้นจึงต้องมีความสามารถในการพูดและฟังอยู่ในโปรแกรมเดียวกัน ในวิทยาพนธ์นี้จะมีโปรแกรมอยู่ 2 โปรแกรมคือ Master สำหรับประธานการประชุม และ Slave สำหรับสมาชิกที่เข้าร่วมประชุม ทั้ง Master และ Slave จะมีความสามารถ ทั้งส่งและรับ หรือเรียกอีก

อย่างว่าพูดและฟัง แต่จะทำได้อย่างใดอย่างหนึ่งเท่านั้น ในเวลาขณะใดขณะหนึ่ง ถ้าอยู่ในภาวะการทำงานแบบพูด ก็พูดได้อย่างเดียว จะพูดและฟังพร้อมๆกันไม่ได้ ในขั้นตอนนี้นอกจากจะทำการรวมโปรแกรม ส่งและรับเข้าด้วยกันแล้ว ยังต้องใส่ฟังก์ชั่น ในการควบคุมต่างๆ เพื่อให้ Master และ Slave ติดต่อกันได้ เช่น Master สามารถสั่งให้ Slave เปลี่ยนภาวะการทำงานเป็นพูดหรือฟังได้ เป็นต้น

ข้อมูลที่ส่งไปมา ระหว่าง Master และ Slave มี 2 แบบ คือ ข้อมูลเสียง และข่าวสารควบคุม (Control Message) การที่เราจะแยกแยะว่าแพ็คเก็ตใดเป็นข่าวสารควบคุม หรือเสียง เราต้องมีวิธีที่จะบอกโปรแกรม ในวิทยานิพนธ์นี้ ใช้การส่งต่างซ็อกเก็ตกัน คือ ส่งและรับเสียงใช้ซ็อกเก็ตกลุ่มหนึ่ง ส่วนการส่งและรับข่าวสารควบคุม ใช้ซ็อกเก็ตอีกกลุ่มหนึ่ง

ประเภทข้อมูล	ซ็อกเก็ตส่ง	ซ็อกเก็ตรับ
ข่าวสารควบคุม	8585	8989
ข้อมูลเสียง	8181	8383

ตาราง 3.1 หมายเลขซ็อกเก็ตที่ใช้

ที่จริงแล้ว การส่งข่าวสารควบคุมนี้ ใช้เทคนิคแบบพื้นฐานทั่วไป คือ ส่งชุดของตัวอักษร ซึ่งประกอบด้วย ฟิลด์ต่างๆ ซึ่งขนาดของฟิลด์ ก็จะต้องเป็นที่รู้กัน ระหว่าง Master และ Slave เพื่อที่จะแยกฟิลด์ได้ถูก เช่น คำขอของสมาชิกที่ ชื่อสมศักดิ์ ในการเข้าร่วมประชุมก็คือ

IN	สมศักดิ์	ชื่อการประชุม	รหัสผ่าน

และคำตอบของประธาน ก็เป็นไปได้ 2 แบบ ขึ้นอยู่กับว่า ชื่อของการประชุม และรหัสผ่านที่สมาชิกส่งมานั้น ถูกต้องหรือไม่

OK	สมศักดิ์

NO	สมศักดิ์

สำหรับข่าวสารควบคุมที่ Master และ Slave ใช้คุยกันนั้นยังแบ่งได้เป็นอีก 2 กรณีคือ

1. Slave ร้องขอ
2. Master ออกคำสั่ง

กรณีที่ Slave ร้องขอ

ปฏิบัติการของ Slave	ปฏิบัติการของ Master
ขอเข้าประชุม	ให้เข้าประชุมได้ ไม่ให้เข้าประชุม
ขอพูด	รับรู้ว่าสมาชิกต้องการพูด
ขอแจ้งว่าเปลี่ยนภาวะจากพูดเป็นฟัง	รับรู้ว่าสมาชิกเปลี่ยนภาวะจากพูดเป็นฟัง
ขอออกจากการประชุม	รับรู้ว่าสมาชิกออกจากการประชุม

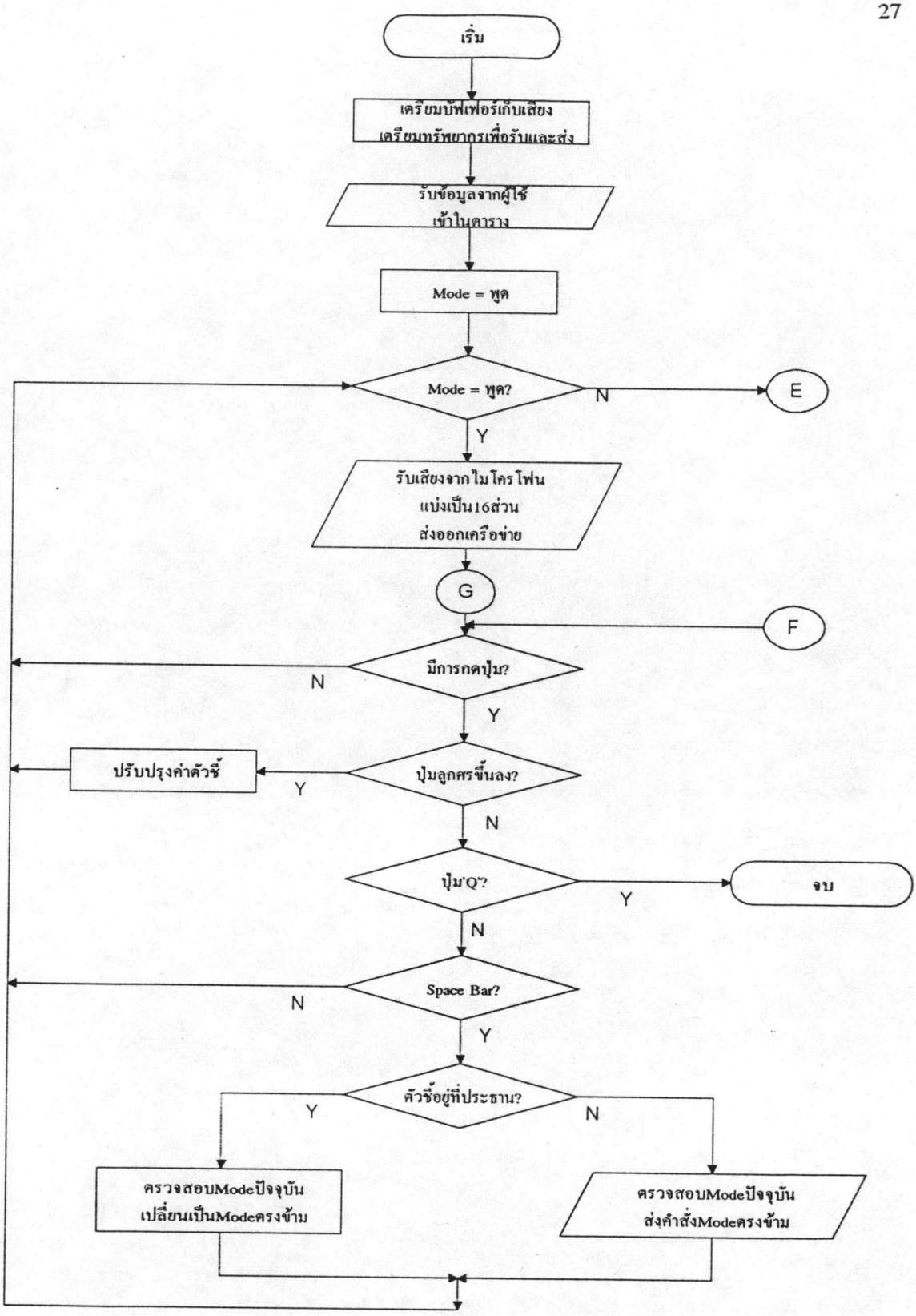
ตาราง 3.2 ข่าวสารควบคุมที่ Slave ร้องขอ

กรณีที่ Master ออกคำสั่ง

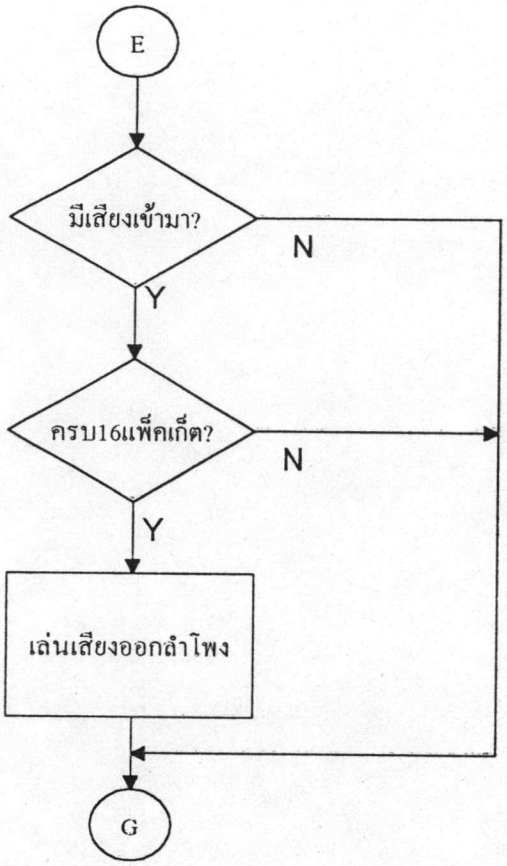
ปฏิบัติการของ Master	ปฏิบัติการของ Slave
ให้เปลี่ยนภาวะจากพูดเป็นฟัง	เปลี่ยนภาวะจากพูดเป็นฟัง
ให้เปลี่ยนภาวะจากฟังเป็นพูด	เปลี่ยนภาวะจากฟังเป็นพูด

ตาราง 3.3 ข่าวสารควบคุมที่ Master ออกคำสั่ง

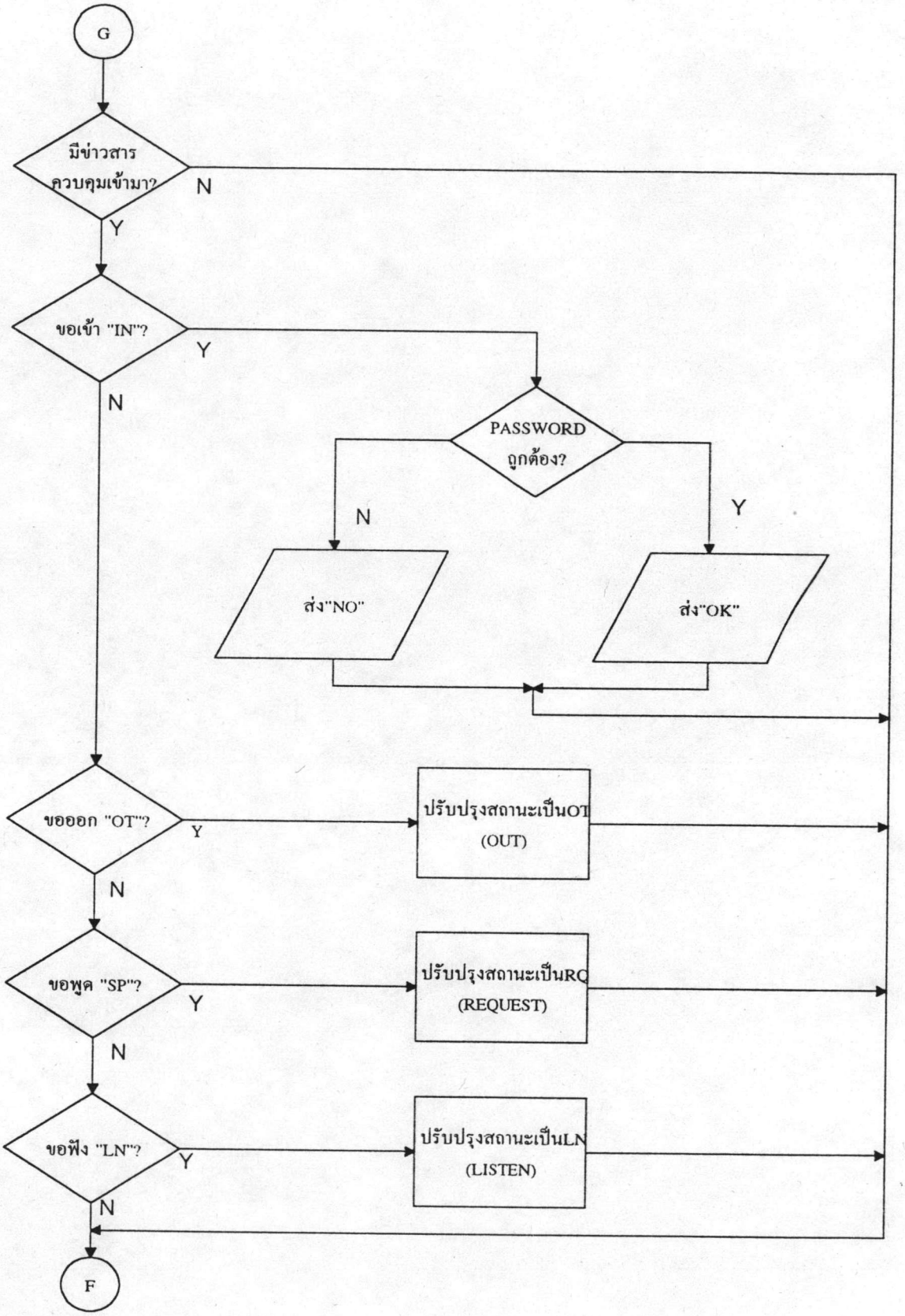
ผังงานแสดงขั้นตอนการทำงานของโปรแกรม Master มีดังต่อไปนี้



รูป 3.8 การทำงานของ Master

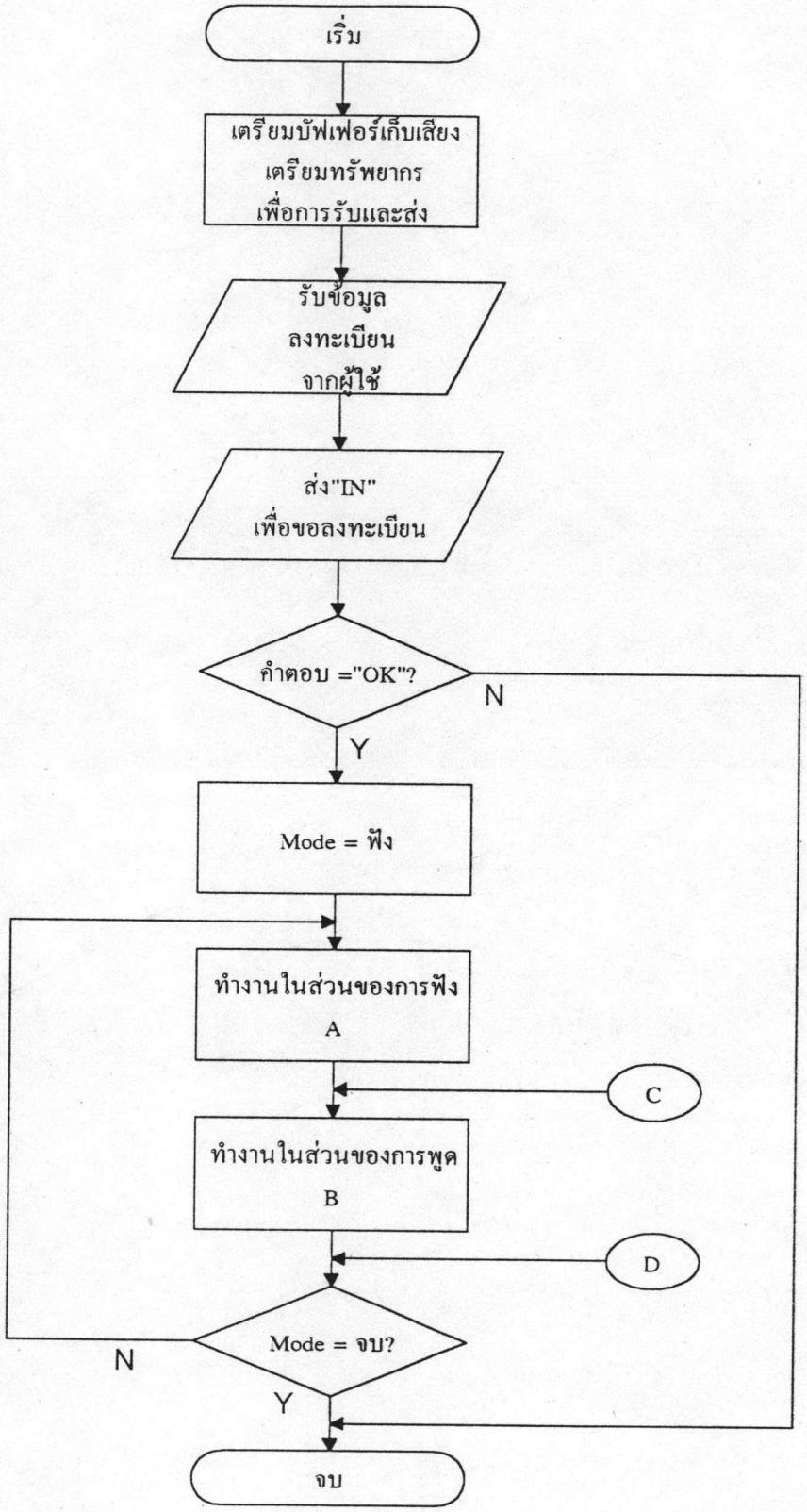


รูป 3.8 (ต่อ)

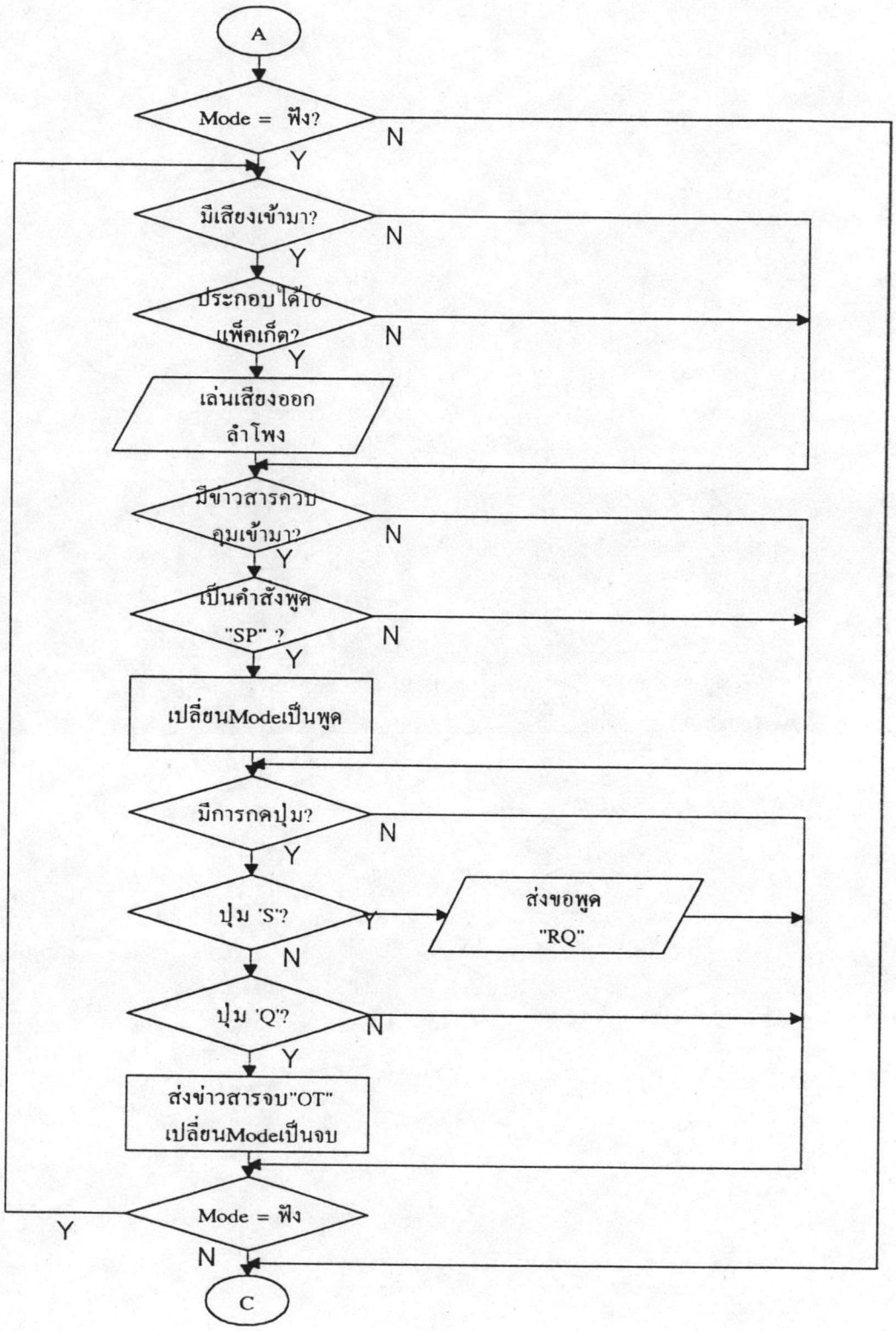

G
มีข่าวสาร
ควบคุมเข้ามา?
N
Y
ขอเข้า "IN"?
Y
N
PASSWORD
ถูกต้อง?
N
Y
ส่ง"NO"
ส่ง"OK"
ขอออก "OT"?
Y
ปรับปรุงสถานะเป็นOT
(OUT)
N
ขอพูด "SP"?
Y
ปรับปรุงสถานะเป็นRQ
(REQUEST)
N
ขอฟัง "LN"?
Y
ปรับปรุงสถานะเป็นLN
(LISTEN)
N
F

รูป 3.8 (ต่อ)

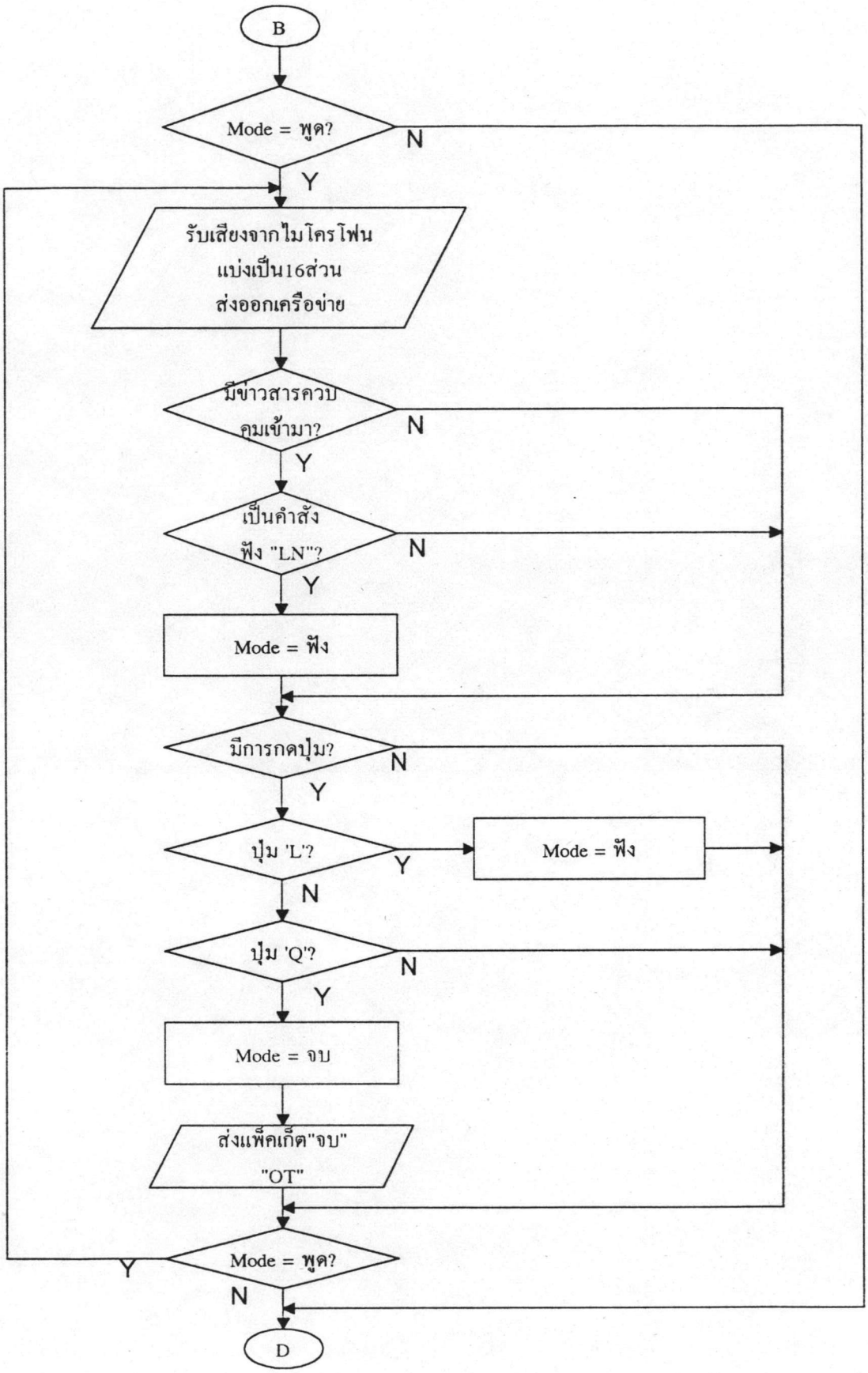
ผังงานแสดงขั้นตอนการทำงานของโปรแกรม Slave มีดังต่อไปนี้



รูป 3.9 การทำงานของ Slave


A
Mode = ฟัง?
N
Y
มีเสียงเข้ามา?
N
Y
ประกอบได้เป็น
แพ็คเก็ต?
N
Y
เล่นเสียงออก
ลำโพง
มีข่าวสารควบ
คุมเข้ามา?
N
Y
เป็นคำสั่งพูด
"SP" ?
N
Y
เปลี่ยนModeเป็นพูด
มีการกดปุ่ม?
N
Y
ปุ่ม 'S'?
Y
ส่งขอพูด
"RQ"
N
ปุ่ม 'Q'?
N
Y
ส่งข่าวสารจบ"OT"
เปลี่ยนModeเป็นจบ
Mode = ฟัง
Y
N
C

รูป 3.9 (ต่อ)


B
Mode = พูด?
N
Y
รับเสียงจากไมโครโฟน
แบ่งเป็น16ส่วน
ส่งออกเครือข่าย
มีข่าวสารควบ
คุมเข้ามา?
N
Y
เป็นคำสั่ง
ฟัง "LN"?
N
Y
Mode = ฟัง
มีการกดปุ่ม?
N
Y
ปุ่ม 'L'?
Y
Mode = ฟัง
N
ปุ่ม 'Q'?
N
Y
Mode = จบ
ส่งแพ็คเก็ต"จบ"
"OT"
Mode = พูด?
Y
N
D

รูป 3.9 (ต่อ)

ปัญหาที่พบในการส่งข่าวสารควบคุมไปมาก็คือ เรามักจะได้คำตอบที่ไม่ได้คาดหวังไว้ เช่น ถ้า Slave ขอเข้าร่วมประชุม (IN) สิ่งที่คาดหวังก็คือ คำคอบควรจะเป็นตกลง (OK) หรือไม่ตกลง (NO) แต่สิ่งที่ได้รับก็คือ IN นั้นเอง ทั้งนี้เป็นเพราะในการส่งข่าวสารควบคุม เราใช้หมายเลขซ็อกเก็ตชุดเดียวกันทั้ง Master และ Slave ดังนั้น เมื่อเราส่งอะไรออกไป เราจึงได้รับด้วยเสมอ เปรียบเสมือนกับเราตะโกนออกไปว่า "ขอเข้าประชุม" แล้วคาดหวังคำตอบว่า "ตกลง" แต่สิ่งที่กลับมาเข้าหูเราก่อน ก็คือคำว่า "ขอเข้าประชุม" นั่นเอง การแก้ปัญหาในกรณีนี้คือว่าถ้า Slave ส่งคำถามอะไรบางอย่างออกไป แล้วคาดหวังคำตอบกลับมา เมื่อมีคำตอบกลับมาให้ตัดคำตอบแรก ทิ้งเสมอ แล้วสิ่งที่รับมา พิจารณาก็คือ ข่าวสารต่อๆ ไป

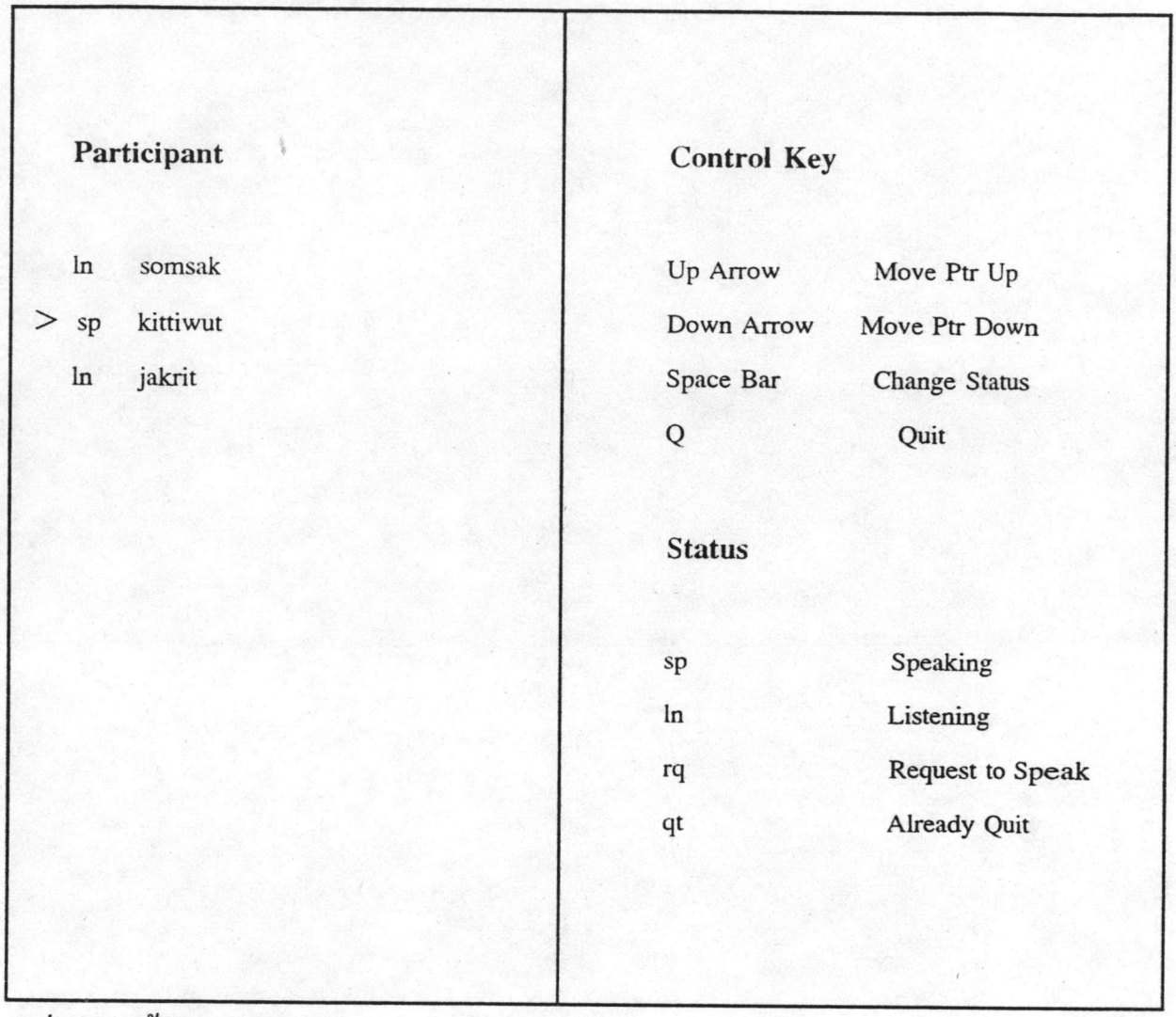
เนื่องจากโปรแกรม Master เป็นศูนย์กลางควบคุมสำหรับการประชุม ซึ่งประธานจะใช้โปรแกรมนี้เพื่อทำงานต่างๆ ที่จำเป็น เช่น ดูว่ามีใครขอพูดบ้าง สั่งให้พูด สั่งให้หยุดพูด เป็นต้น ในการนี้ตัวโปรแกรม Master จะต้องมีตารางซึ่งเก็บข้อมูลเกี่ยวกับสมาชิก ที่เข้าประชุมอยู่ ซึ่งในตารางนี้จะประกอบด้วย รายชื่อของสมาชิกและสถานะของสมาชิกแต่ละคน สถานะของสมาชิกเป็นไปได้ 4 แบบดังต่อไปนี้

Speak	อยู่ในภาวะพูด
Listen	อยู่ในภาวะฟัง
Request	อยู่ในภาวะฟัง และต้องการขอพูด
Out	ออกจากการประชุมไปแล้ว

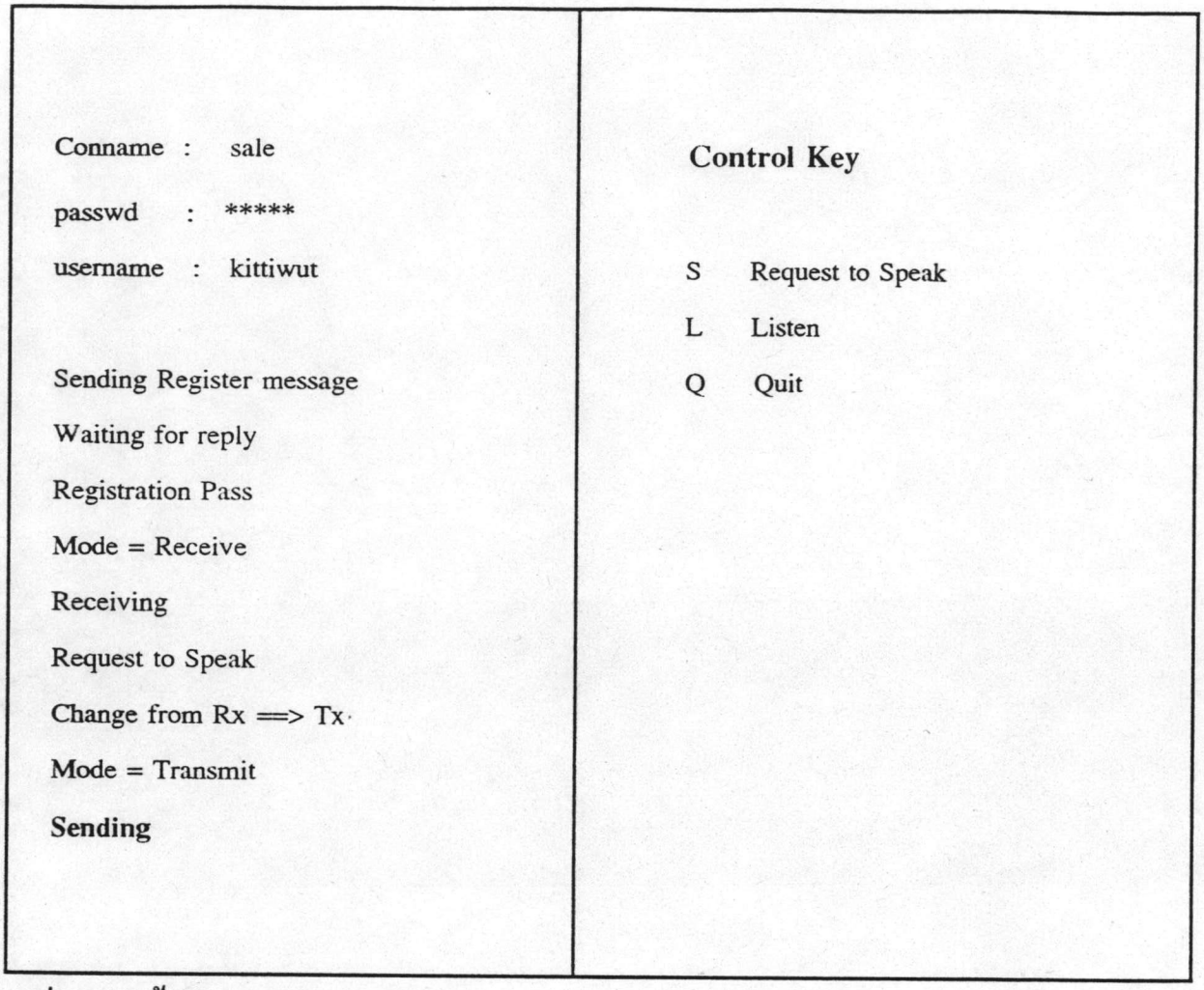
โปรแกรม Master จะปรับปรุงตารางของตนเอง เมื่อประธาน กดปุ่มสั่งอะไรบางอย่าง ซึ่งส่งผลต่อการเปลี่ยนแปลงสถานะของสมาชิก หรือเมื่อสมาชิก ทำอะไรบางอย่างที่มีผล ต่อการเปลี่ยนสถานะ เช่น เมื่อสมาชิกพูดจบแล้วและขอจบ ก็จะส่งข่าวสารควบคุมมาบอกประธานว่าขอเปลี่ยนภาวะจากพูดเป็นฟัง ประธานก็จะเปลี่ยนสถานะของสมาชิกในตารางของตนเองจากภาวะพูด เป็นภาวะฟัง เป็นต้น

สำหรับหน้าจอของโปรแกรม Master และ Slave ได้แสดงไว้ในรูป 3.10 และ 3.11 ตามลำดับ ในหน้าจอของ Master ครึ่งซ้ายของจอ จะเป็นรายชื่อ ของคนที่ได้ลงทะเบียนเข้ามาแล้ว เรียงตามลำดับก่อนหลัง โดยที่ประธานจะเป็นชื่อแรกเสมอ สิ่งที่สำคัญอีกอย่าง คือสถานะของผู้เข้าร่วม ส่วนหน้าจอซีกขวาจะเป็นคำอธิบายเกี่ยวกับสถานะและปุ่มต่างๆ

ในหน้าจอของ Slave ครึ่งขวาของจอจะเป็นคำบรรยายคล้ายกับจอของ Master ส่วนครึ่งซ้ายจะแสดงขั้นตอนการทำงานต่างๆเรียงลำดับไปเรื่อยๆ ตั้งแต่ข้างบนลงมา เริ่มตั้งแต่การลงทะเบียน แล้วเข้าสู่ภาวะฟัง หลังจากนั้นก็จะแสดงข้อความต่างๆแล้วแต่สถานการณ์ไปเรื่อยๆ เหตุการณ์ล่าสุด จะอยู่บรรทัดล่างสุด และตัวอักษรจะกระพริบ



รูป 3.10 หน้าจอของ Master



รูป 3.11 หน้าจอของ Slave