โปรแกรมมัลติมีเดียช่วยในการเรียนการสอนระยะใกล้ของซอฟต์แวร์ภายใต้วินโดวส์

ศักดิ์ชัย เอี่ยมกิจสัมฤทธิ์* และ พิพัฒน์ หิรัณย์วณิชชากร**

1. บทนำ

ในปัจจุบันมีการนำเอาไมโครคอมพิวเตอร์มาใช้งานกันอย่างกว้างขวาง จึงมีซอฟต์แวร์ที่ถูกพัฒนามาใช้กับไมโครคอมพิวเตอร์มากมาย การฝึกอบรมบุคลากรให้มีความรู้ ความชำนาญในการใช้ซอฟต์แวร์เหล่านี้มีความจำเป็น การฝึกอบรมปัจจุบันอาจมีการใช้อุปกรณ์ เช่น Datashow และ LCD projector ซึ่งฉายภาพหน้าจอของผู้สอนไปบนจอภาพขนาดใหญ่ แต่อุปกรณ์นี้มีข้อจำกัดคือ ใช้สอนกับผู้รับการฝึกอบรมจำนวนไม่มาก และผู้สอนไม่ทราบผลของการปฏิบัติของผู้รับการฝึกอบรมได้ จึงมีการผลิตซอฟต์แวร์ช่วยในการสอนบน LAN เช่น DIDACT ซึ่งผู้รับการฝึกอบรมสามารถมองเห็นภาพหน้าจอของผู้สอนปรากฏบนหน้าจอของตัวเอง และผู้สอนสามารถตรวจดูภาพบนหน้าจอของผู้รับการฝึกอบรมได้ แต่โปรแกรมนี้ไม่สามารถทำให้มีการโต้ตอบระหว่างผู้สอนและผู้รับการฝึกอบรม ผู้วิจัยจึงได้พัฒนาโปรแกรมช่วยในการเรียนการสอนระยะใกล้[1] ซึ่งแบ่งหน้าจอออกเป็น 2 ส่วน ทำให้นักเรียนสามารถเห็นหน้าจอของครูและหน้าจอของตัวเองได้พร้อม ๆ กัน จึงทำให้นักเรียนสามารถปฏิบัติตามครูได้ และยังมีการพูดคุยโต้ตอบระหว่างครูและนักเรียนได้ แต่อย่างไรก็ตาม โปรแกรมดังกล่าวนี้ได้ถูกพัฒนาเพื่อใช้สอนโปรแกรมประยุกต์ต่าง ๆ ที่ทำงานบนระบบปฏิบัติการดอส เช่น LOTUS, WordStar, Dbase, และราชวิถีเวิร์ด ซึ่งส่วนใหญ่เป็นโปรแกรมที่มีการแสดงผลในรูปของตัวอักษร นอกจากนั้น ระบบการแสดงตัวอักษรไทยบนระบบปฏิบัติการดอสนั้น มักจะเกี่ยวข้องกับหน่วยความจำ Video RAM ซึ่งทำให้การทำงานของโปรแกรมช่วยในการเรียนการสอนนั้นผิดเพี้ยนไป

ปัจจุบันนี้ ระบบวินโดวส์ได้ถูกใช้งานอย่างแพร่หลาย โปรแกรมประยุกต์ที่พัฒนาส่วนใหญ่ก็จะทำงานภายใต้สภาพแวดล้อมของวินโดวส์และมีการแสดงผลในรูปแบบของกราฟิก ดังนั้นผู้วิจัยจึงปรับปรุงโปรแกรมช่วยในการเรียนการสอนให้สามารถฝึกอบรมโปรแกรมประยุกต์บนวินโดวส์ได้คือ โปรแกรมช่วยในการเรียนการสอนระยะใกล้ภายใต้วินโดวส์[2] (CAINET) ซึ่งปกติแล้วระบบวินโดวส์นั้นผู้ใช้สามารถเรียก windows มาใช้งานได้หลายช่อง จึงสอดคล้องกับแนวคิดของผู้วิจัยที่ต้องการให้ผู้เรียนสามารถ

* นักศึกษาปริญญาโท คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์

** รองศาสตราจารย์ คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์

เรียก windows มาใช้งานได้หลายช่อง จึงสอดคล้องกับแนวคิดของผู้วิจัยที่ต้องการให้ผู้เรียนสามารถเห็นภาพหน้าจอที่ครูสอน แสดงบนจอภาพของตนเอง เพื่อตนเองจะได้ปฏิบัติตามบน window ของตนเองได้ นอกจากนั้นเนื่องจากระบบวินโดวส์มีการจัดการจอภาพซึ่งอาจถูกใช้โดยโปรแกรมประยุกต์หลายตัวพร้อม ๆ กัน โดยการไม่ยอมให้โปรแกรมประยุกต์ดัดแปลงการใช้งานจอภาพโดยตรง จะต้องมีการขอใช้โดยผ่านระบบวินโดวส์นั้นทำให้วิธีการแสดงผลของโปรแกรมประยุกต์ต่าง ๆ บนวินโดวส์อยู่ในลักษณะเดียวกัน และทำให้โปรแกรมช่วยในการเรียนการสอนดังกล่าวนี้สามารถใช้กับโปรแกรมประยุกต์ทั่ว ๆ ไปบนระบบวินโดวส์ได้

แต่อย่างไรก็ตามโปรแกรมช่วยในการเรียนการสอ นบนวินโดวส์นี้ ใช้หลักการของการส่งข้อมูลภาพหน้าจอและข้อมูลตัวอักษรไปเก็บไว้ที่ file server เพื่อให้ฝั่งตรงข้ามดึงไปแสดงบนหน้าจอของตนเอง จึงทำให้เกิดปัญหาคอขวดขึ้นได้ นอกจากนี้โปรแกรมนี้ยังไม่สามารถให้มีการโต้ตอบด้วยเสียงระหว่างครูและนักเรียนได้

ในบทความนี้ ผู้วิจัยได้เสนอแนวคิดแก้ไขปัญหาคอขวดของโปรแกรมช่วยในการเรียนการสอนบนวินโดวส์โดยแทนที่จะใช้ file server เป็นตัวกลางในการส่งข้อมูลภาพหน้าจอระหว่างครูและนักเรียน งานวิจัยนี้ใช้การแพร่กระจายข้อมูลผ่านเครือข่ายจากครูไปยังนักเรียนทุกคน และให้นักเรียนและครูสามารถส่งข้อมูลโต้ตอบกันได้โดยตรง นอกจากนั้นเพื่อให้การเรียนการสอนมีประสิทธิภาพมากขึ้น จึงได้มีการพัฒนาโปรแกรมนี้ให้สามารถส่งข้อมูลภาพและเสียงไปพร้อม ๆ กันได้

2. หลักการทำงานของระบบ

รูปที่ 1 แสดงการทำงานของโปรแกรม CAINET ซึ่งใช้ file server เป็นตัวกลางในการส่งข้อมูลระหว่างครูและนักเรียนจึงทำให้เกิดปัญหาคอขวด จากรูปจะเห็นว่าเมื่อมีนักเรียนจำนวนมากต้องการจะ access แฟ้มข้อมูลหน้าจอของครูพร้อมกันก็จะมีได้เพียง 1 คนเท่านั้น ณ ขณะใด ๆ ที่จะสามารถดึงข้อมูลหน้าจอของครูไปได้ เช่นตอนนี้ Student 3 เท่านั้นที่สามารถจะได้รับภาพหน้าจอของครูไปปรากฏที่หน้าจอของตนเอง สำหรับ Student 1 และ Student 2 นั้นจะต้องรอจนกว่า Student 3 อ่านข้อมูลภาพหน้าจอของครูเสร็จก่อนจึงจะมีโอกาสนำข้อมูลภาพหน้าจอของครูไปได้ ดังนั้นภาพหนึ่งภาพของครูนั้นถ้ามีนักเรียนในระบบ 3 คน ก็จะต้องมีการอ่านแฟ้มข้อมูลหน้าจอจาก file server ถึง 3 ครั้ง (คนละ 1 ครั้ง จำนวน 3 คน) เพื่อให้นักเรียนได้รับภาพเดียวกัน และเมื่อจำนวนนักเรียนมากขึ้น การไปดึงแฟ้มข้อมูลจาก file server ก็มากขึ้น จึงทำให้เกิดปัญหาคอขวดและการทำงานของโปรแกรมจะช้าลงมาก

ปัจจุบันนี้ TCP/IP เป็นโปรโตคอลที่นิยมอย่างแพร่หลายในการควบคุมการส่งข้อมูลของเครือข่ายอินเทอร์เน็ต โปรโตคอลนี้ในระดับชั้นทรานสปอร์ตแบ่งเป็น TCP (Transmission Control

Protocol) และ UDP (User Datagram Protocol) สำหรับโปรโตคอล TCP ต้องมีการสร้างการติดต่อระหว่างผู้ใช้ปลายทางจึงเหมาะสำหรับการส่งข้อมูลระหว่าง 2 จุด

ส่วน UDP เป็นโปรโตคอลที่ไม่ต้องสร้างการติดต่อระหว่างผู้ใช้จึงสามารถส่งข้อมูลได้อย่างรวดเร็วและเหมาะสำหรับการส่งข้อมูลแบบแพร่กระจาย ดังนั้นผู้วิจัยจึงคิดว่า ถ้าสามารถส่งข้อมูลภาพหน้าจอของครูออกไปเพียง 1 ครั้ง และสามารถทำให้นักเรียนทุกคนที่อยู่ในเครือข่ายเดียวกันสามารถได้รับพร้อม ๆ กัน จะทำให้ประสิทธิภาพในการรับ-ส่งข้อมูลดีขึ้น ดังแสดงในรูปที่ 2 จะเห็นได้ว่าข้อมูลของครูเมื่อถูกแพร่กระจายออกไปนั้น นักเรียนทุกคนที่เชื่อมต่ออยู่ในเครือข่ายเดียวกันกับครูจะสามารถได้รับข้อมูลนั้นพร้อม ๆ กัน ทำให้ประสิทธิภาพในการส่งข้อมูลจากครูไปยังนักเรียนดีขึ้น ส่วนในกรณีที่มีการโต้ตอบโดยตรงระหว่างครูและนักเรียนนั้น ในงานวิจัยใช้โปรโตคอล TCP ในการติดต่อส่งข้อมูลเนื่องจากเป็นการติดต่อในลักษณะ 1 ต่อ 1 กับครูจึงไม่จำเป็นต้องใช้ UDP

ต่อไปจะอธิบายถึงหลักการทำงานของระบบโดยละเอียด ซึ่งแบ่งการทำงานออกเป็น 5 ส่วนใหญ่ ๆ คือ

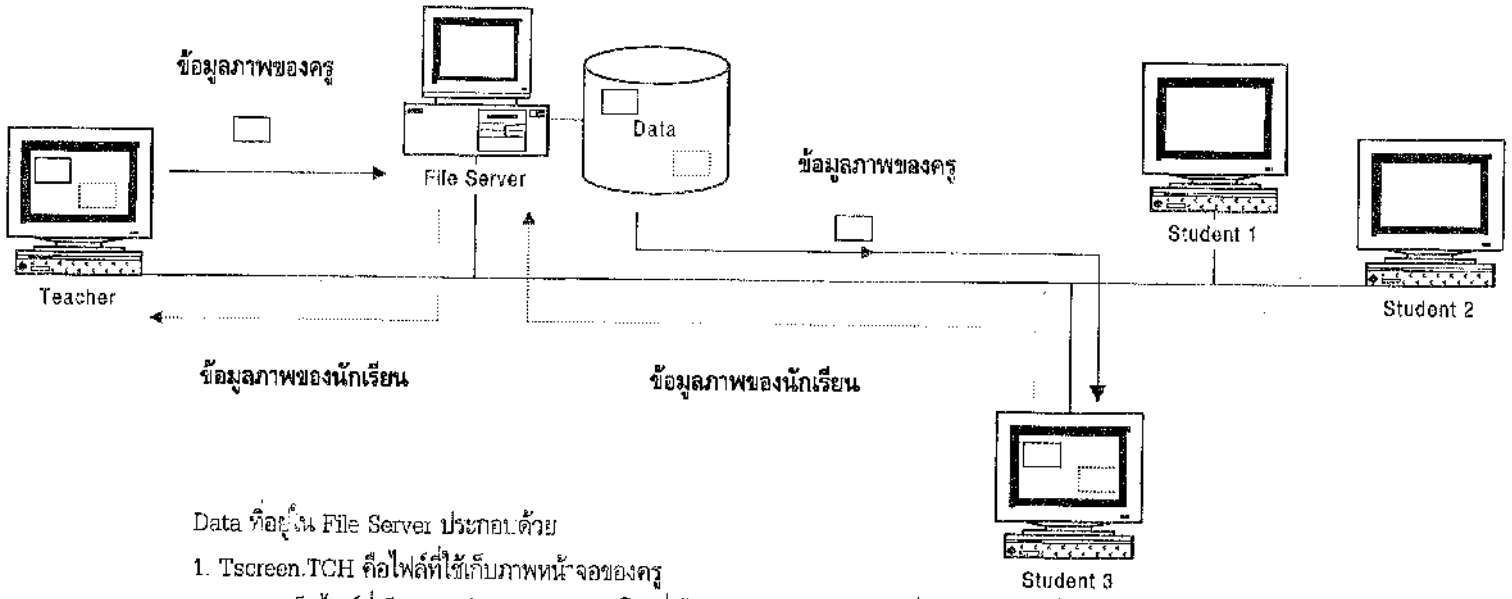
ก. การนำภาพหน้าจอของครูไปแสดงบนหน้าจอนักเรียน

ข. การนำภาพหน้าจอของนักเรียนไปแสดงบนหน้าจอครู

ค. การจัดการเกี่ยวกับเสียง

ง. โปรโตคอลในการควบคุมการรับ-ส่งข้อมูลในระดับงานประยุกต์

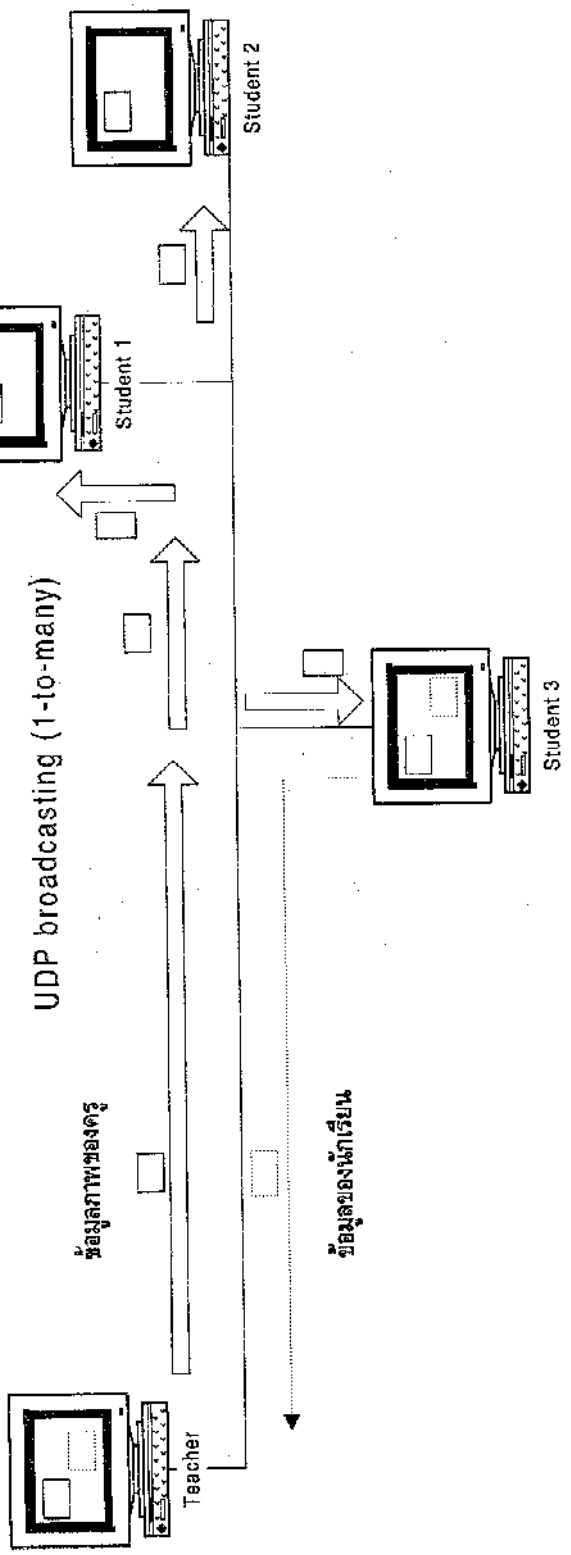
จ. การโต้ตอบกันระหว่างครูและนักเรียน



Data ที่อยู่ใน File Server ประกอบด้วย

1. Tscreen.TCH คือไฟล์ที่ใช้เก็บภาพหน้าจอของครู
2. *.STD เป็นไฟล์ที่เก็บภาพหน้าจอของนักเรียนโดยชื่อไฟล์เป็นรหัสของนักเรียนที่ได้แจ้งไว้ตอนเริ่มรันโปรแกรม
3. Teacher.MES เป็นไฟล์ที่เก็บรายละเอียดเกี่ยวกับการ chat ระหว่างครูกับนักเรียน โดยจะเก็บรหัสของนักเรียนที่ต้องการจะ chat และ message ที่ครู chat ด้วย
4. *.MES เป็นไฟล์ที่เก็บ message ที่นักเรียนต้องการจะ chat กับครูโดยชื่อไฟล์จะเป็นชื่อเดียวกันกับรหัสของนักเรียน
5. Chat.QUE จะเก็บรายละเอียดว่ามีนักเรียนคนใดต้องการที่จะ chat กับครูบ้าง

รูปที่ 1 แสดงการติดต่อกันและไฟล์ต่าง ๆ ที่ใช้ในโปรแกรม CAINET ซึ่งเก็บไว้ที่ File Server



รูปที่ 2 แสดงการส่งข้อมูลของครูไปยังนักเรียนด้วยการ broadcast ของโปรโตคอลUDPและจากนักเรียนไปหาครูด้วย TCP

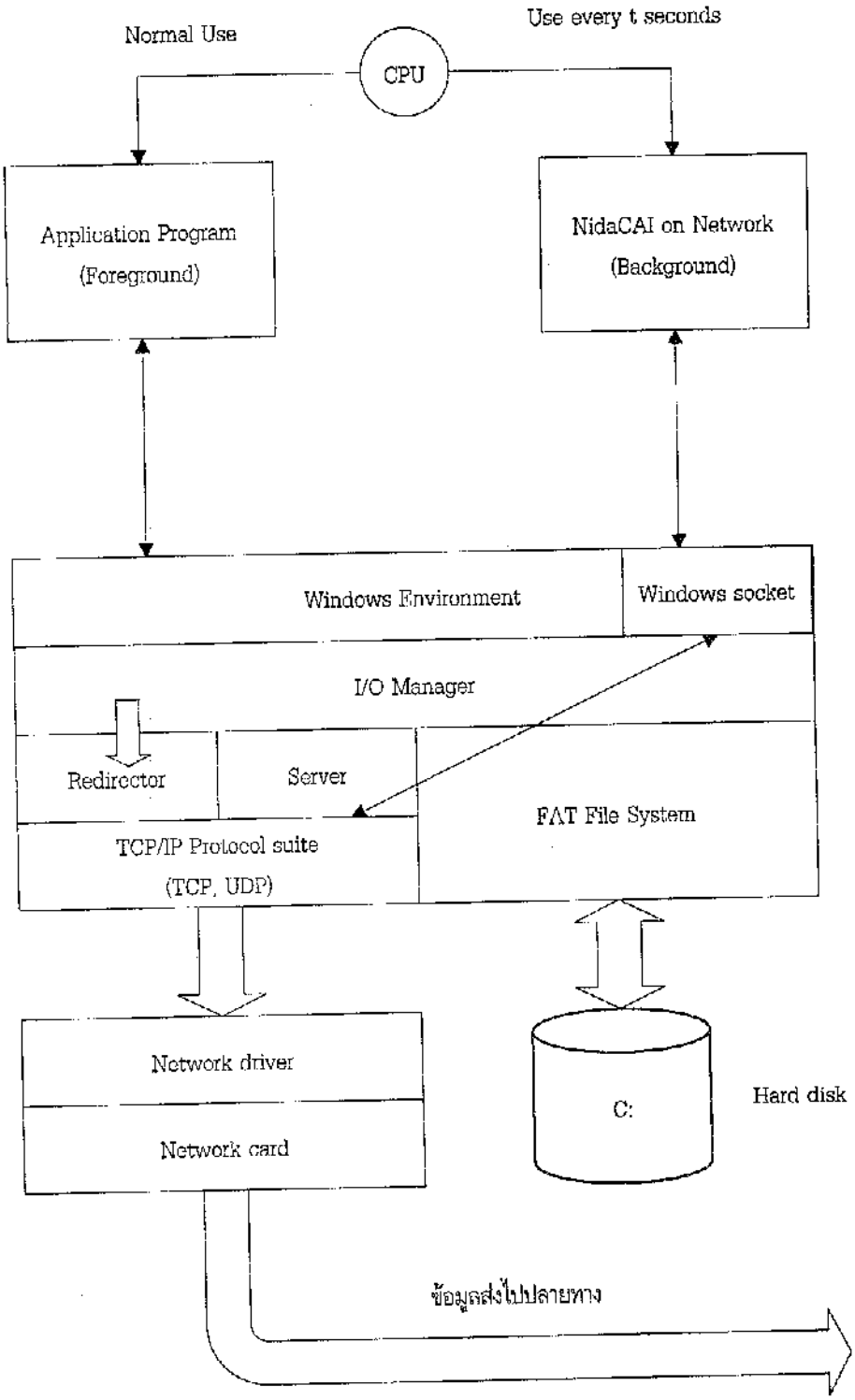
2.1 การนำภาพหน้าจอของครูไปแสดงบนหน้าจอนักเรียน

เมื่อครูทำการสอนโปรแกรมประยุกต์ ระบบจะคัดลอกเอาภาพหน้าจอในส่วนที่ครูสอนไปแสดงบนส่วนของหน้าจอของนักเรียน เพื่อนักเรียนจะสามารถปฏิบัติตามครูได้ ซึ่งการทำงานในขั้นตอนนี้สามารถแสดงได้ในรูปที่ 3 จะเห็นว่าเมื่อมีการเรียกโปรแกรมประยุกต์เช่น Excel มาทำงาน โปรแกรมประยุกต์จะใช้ CPU ในการทำงาน และหน้าจอจะแสดงผลของโปรแกรม ทำให้มีการโต้ตอบระหว่างผู้ใช้กับโปรแกรมประยุกต์ กล่าวคือทำงานในลักษณะของ foreground นอกจากนั้นอาจจะมีการส่ง message ไปยังระบบวินโดวส์เพื่อแจ้งความประสงค์ในการใช้ทรัพยากรส่วนกลางร่วมกับโปรแกรมอื่น ๆ เช่นการขอใช้จอภาพ หน่วยความจำ เป็นต้น แต่เมื่อมีการขอใช้บริการเกี่ยวกับแฟ้มข้อมูล ระบบวินโดวส์จะไปเรียกใช้ module I/O manager เมื่อ I/O manager ได้รับการขอใช้บริการก็จะแยกว่า ถ้าเป็นการขอใช้แฟ้มข้อมูลของเครื่อง workstation เอง ก็จะส่งการขอรับบริการไปยัง FAT file system แต่หากเป็นการขอใช้บริการทรัพยากรของเครือข่ายแล้ว I/O manager จะส่งคำขอผ่าน redirector ซึ่งอาจใช้โปรโตคอล TCP/IP เป็นตัวจัดการการส่งข้อมูลผ่านไปยัง Network driver เพื่อส่งข้อมูลไปยัง server (server process ที่ให้บริการด้านแฟ้มข้อมูลในเครือข่าย)

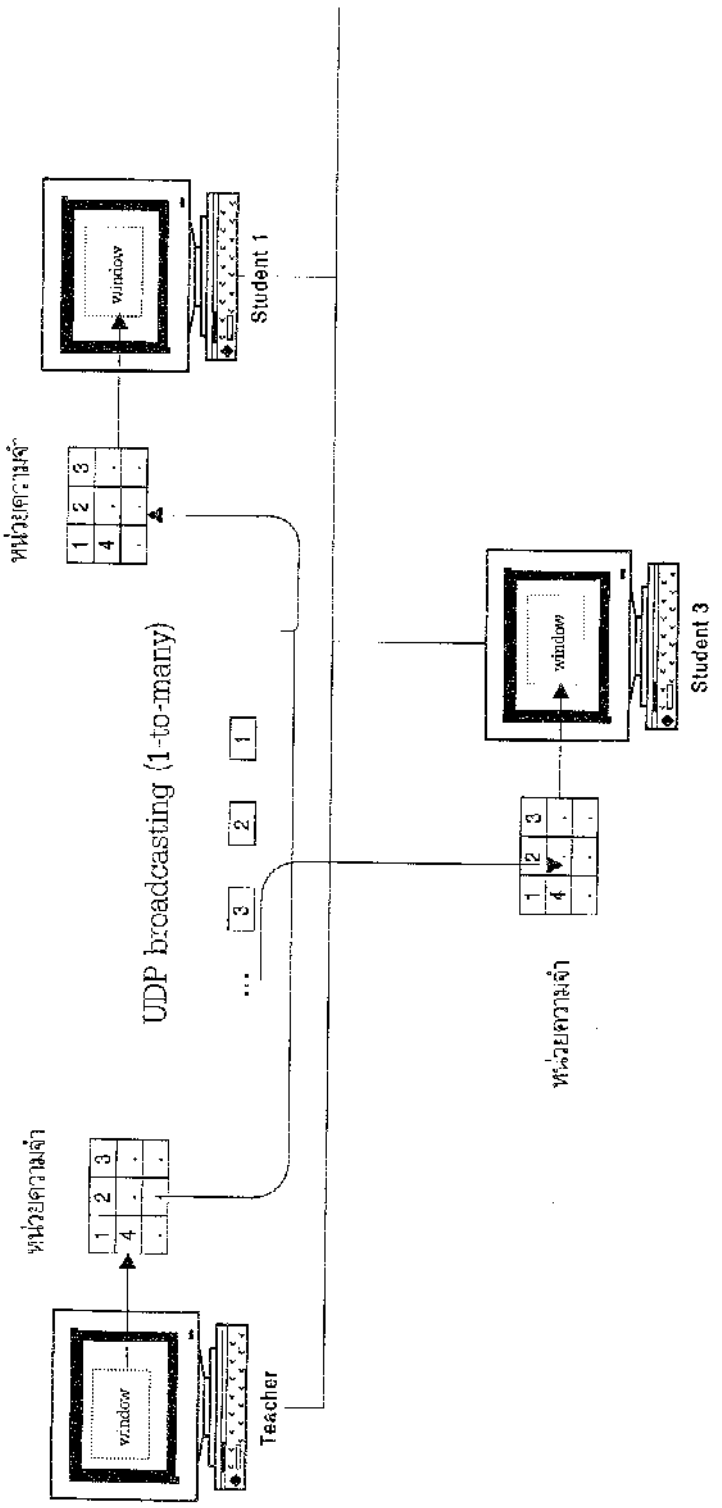
สำหรับโปรแกรมช่วยในการเรียนการสอนในส่วนของเครื่องผู้สอนจะทำงานในลักษณะของ background กล่าวคือ ในขณะที่ครูกำลังสอนโปรแกรมประยุกต์ซึ่งทำงาน foreground อยู่นั้น ทุก ๆ เวลา t หน่วย โปรแกรมจะใช้ CPU time ในการคัดลอกหน้าจอของครู ส่งไปเก็บไว้ในหน่วยความจำ หลังจากนั้นก็จะแบ่งข้อมูลหน้าจอออกเป็น block ย่อย ๆ ทยอยส่งออกไปในเครือข่ายด้วย TCP/IP โปรแกรมจะส่งคำขอผ่าน Windows socket ซึ่งจะไปเรียกใช้ TCP/IP เป็นตัวส่งข้อมูลผ่านเครือข่ายไปยังเครื่องปลายทางดังแสดงในรูปที่ 3 และ 4 เพื่อที่โปรแกรมในส่วนของนักเรียนจะได้นำไปรวบรวมแล้วจึงนำไปแสดงบนจอของนักเรียนต่อไป

ในขณะที่โปรแกรมบนเครื่องผู้สอนทำการคัดลอกหน้าจอและส่งไปยังผู้เรียนนั้นจะไม่มีการแสดงผลปรากฏบนหน้าจอ ดังนั้นผู้ใช้จึงดูเสมือนว่า CPU time ยังถูกใช้งานโดยโปรแกรมประยุกต์ (กล่าวคือทำงานเป็น foreground) เมื่อส่งข้อมูลหน้าจอเสร็จแล้วก็จะปล่อยให้โปรแกรมประยุกต์ใช้ CPU time อีก

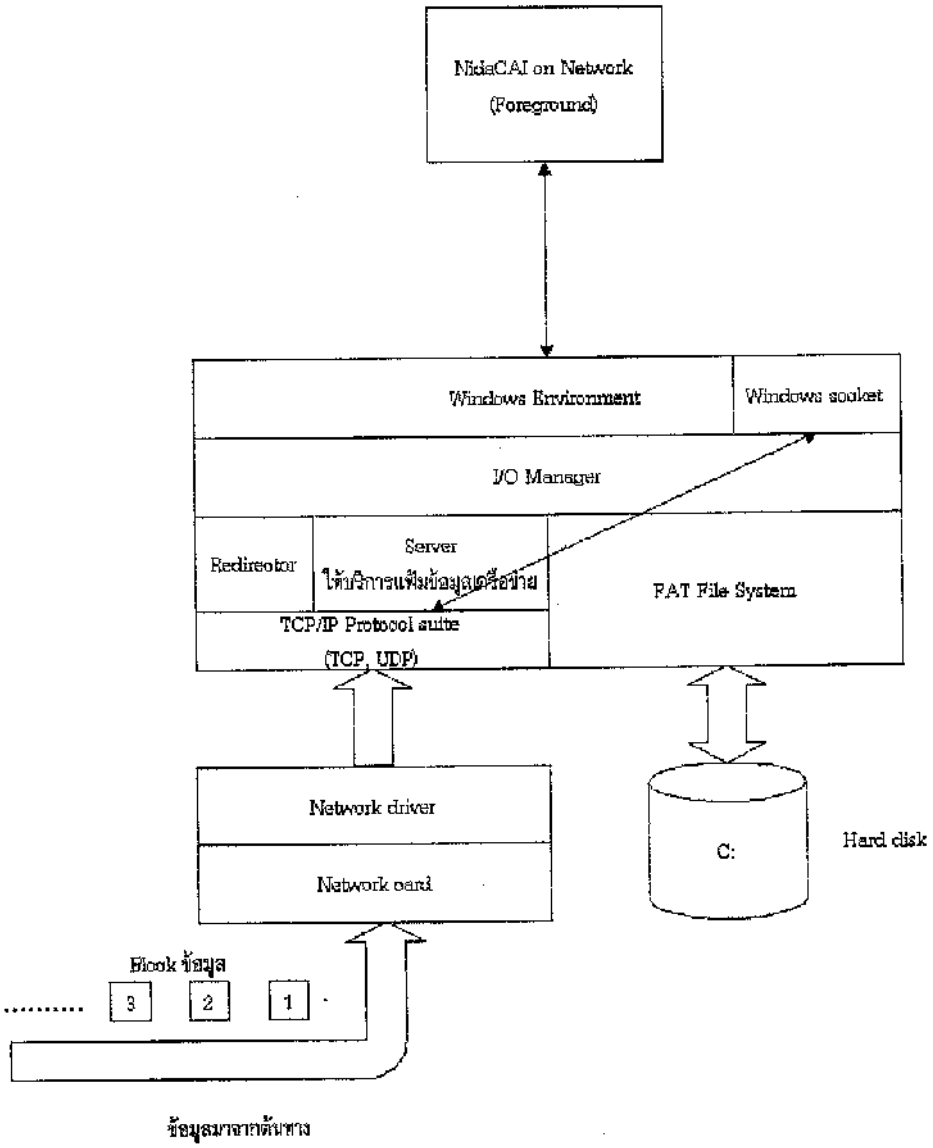
ในส่วนของโปรแกรมบนเครื่องของนักเรียนแต่ละคนจะทำงานดังแสดงในรูปที่ 5 โดยเมื่อ block ข้อมูลภาพหน้าจอของครูมาถึงเครื่องของนักเรียนก็จะผ่านขึ้นมาจาก Network card แล้วก็จะถูกรับโดย TCP/IP แล้วโปรแกรมของนักเรียนจะนำ block ข้อมูลจาก TCP/IP (ผ่าน Windows socket) ที่ได้นี้ไปเก็บรวบรวมไว้ที่หน่วยความจำ จนกระทั่งครบ 1 ภาพ แล้วจึงนำไปแสดงบน window ของจอภาพของนักเรียน



รูปที่ 3 แสดงการทำงานและการส่งข้อมูลผ่าน TCP/IP ของโปรแกรม



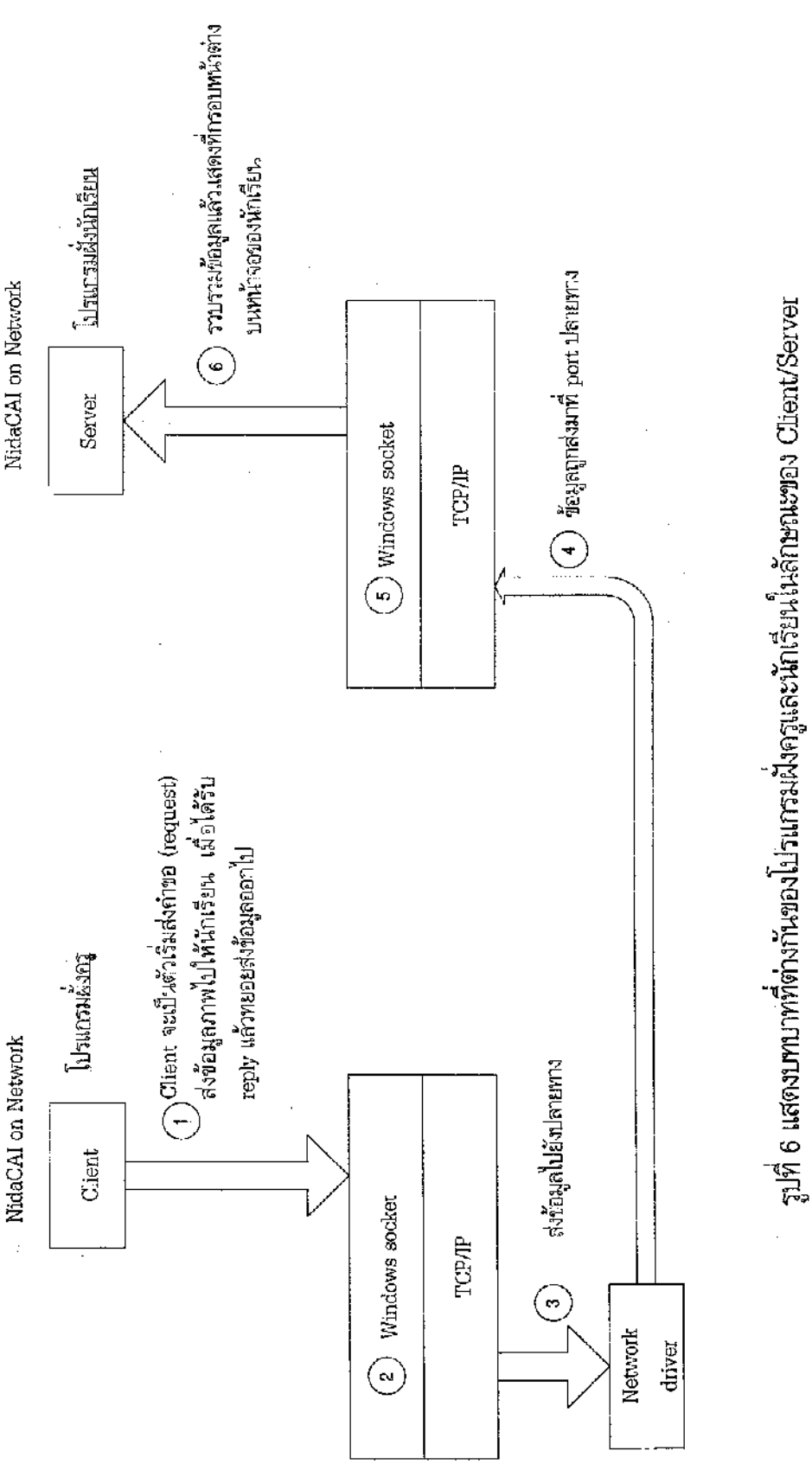
รูปที่ 4 แสดงการแพร่กระจายข้อมูลจากครูไปยังนักเรียน



รูปที่ 6 แสดงการรับ block ข้อมูลผ่าน TCP/IP

ในการใช้โปรโตคอล TCP/IP ในการส่งข้อมูลผ่านเครือข่ายนั้น โปรแกรมประยุกต์จะเขียนอยู่ในรูปแบบของ client/server โดยที่ client จะเป็นโปรแกรม (process) ที่เริ่มติดต่อและส่ง request ไปยัง server แล้วรอรับ reply ตอบกลับมา ส่วนโปรแกรม server ซึ่งรอรับการติดต่อจาก client เมื่อได้รับ request ก็จะทำงานตามคำขอนั้น แล้วส่ง reply กลับไปยัง client

สำหรับในกรณีนี้ผู้สอนจะถือว่าเป็น client ที่ขอส่งหน้าจอไปให้กับนักเรียนซึ่งทำงานเป็น server ในการรับภาพหน้าจอของผู้สอนมาแสดงบนกรอบหน้าต่างที่แสดงภาพหน้าจอของครูบนหน้าจอของเครื่องนักเรียน รูปที่ 6 จะแสดงให้เห็นถึงบทบาทที่ต่างกันของโปรแกรมฝั่งครูและนักเรียนในลักษณะของ client/server



รูปที่ 6 แสดงบทบาทที่ต่างกันของโปรแกรมฝั่งครูและนักเรียนในลักษณะของ Client/Server

2.1.1 การคัดลอกหน้าจอของโปรแกรมประยุกต์บนวินโดวส์

การคัดลอกหน้าจอของผู้สอนเพื่อส่งไปยังฝั่งตรงข้าม นับว่าเป็นสิ่งสำคัญอย่างหนึ่งของโปรแกรมช่วยในการเรียนการสอนนี้ ซึ่งการคัดลอกหน้าจอนี้มีขั้นตอนต่าง ๆ ดังต่อไปนี้

ก. การเตรียมที่เก็บสำหรับข้อมูลภาพที่จะคัดลอก

ขั้นแรกก่อนที่จะเรียกใช้งานโปรแกรมประยุกต์ที่จะถูกจัดเก็บหน้าจอ จะต้องจัดเตรียมที่เก็บข้อมูลภาพนี้ในหน่วยความจำซึ่งทำได้โดยจัดสร้าง memory device context (MemDC) ซึ่งมีโครงสร้างข้อมูลเป็น bitmap

ข. หาหมายเลขประจำตัวของโปรแกรม

ขั้นตอนต่อไปโปรแกรมจะต้องทราบว่า window ของโปรแกรมประยุกต์ที่ต้องการคัดลอกหน้าจอภาพนั้น อยู่ที่ไหนบนหน้าจอ ในการนี้จะต้องหาหมายเลขประจำตัวของโปรแกรมประยุกต์นั้นก่อน ซึ่งในงานวิจัยนี้ทำได้โดยในการเรียกใช้โปรแกรมประยุกต์มาทำงานนั้น จะผ่านโปรแกรมมัลติมีเดียช่วยในการเรียนการสอนระยะใกล้ของซอฟต์แวร์ภายใต้วินโดวส์ ดังนั้นจึงทำให้โปรแกรมฯ สามารถเก็บหมายเลขของโปรแกรมประยุกต์ไว้ได้

ค. คัดลอกหน้าจอ

เมื่อได้หมายเลขประจำตัวของโปรแกรมประยุกต์แล้ว ก็สามารถคัดลอกหน้าจอของโปรแกรมนั้น โดยเรียกใช้ฟังก์ชันในการคัดลอกข้อมูลที่ระบบวินโดวส์ได้จัดเตรียมเอาไว้เป็น API เช่น ฟังก์ชัน StretchBlt หรือ BitBlit เพื่อนำเอาภาพที่ต้องการไปเก็บยัง MemDC ซึ่งจะถูกนำไปแบ่งเป็น block ข้อมูลย่อยที่จะทยอยส่งไปที่ฝั่งตรงข้ามต่อไป

การคัดลอกหน้าจอนี้ สามารถแบ่งขั้นตอนย่อยออกได้ดังนี้คือ

ค.1 การหาตำแหน่ง window ของโปรแกรมประยุกต์ เมื่อเทียบกับตำแหน่งของ Windows Desktop [Left, Top, Right, Bottom] ดังแสดงในรูป 7(ก)

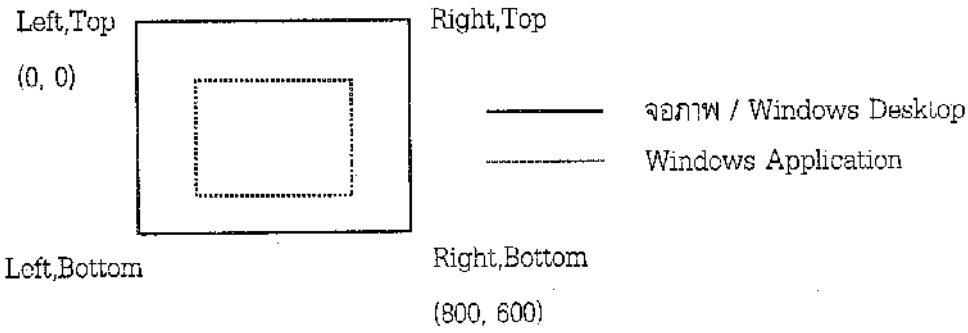
ค.2 เนื่องจากบางครั้ง window ของโปรแกรมประยุกต์ อาจจะเกินขอบเขตของ Windows Desktop ดังแสดงในรูป 7(ข) และ 7(ค) แต่ในส่วนของครูก็จะเห็นเฉพาะส่วนของ window ที่อยู่ใน Desktop ดังนั้นหากคัดลอก window ทั้งหมดของครูแล้ว ส่วนที่เกินขอบเขตของ Desktop จะได้ภาพที่ไม่ต้องการ (ลักษณะคล้ายขยะ) ดังนั้น จึงต้องปรับตำแหน่ง window ของโปรแกรมประยุกต์ให้อยู่ภายใต้ขอบเขตของ Desktop เพื่อจะได้คัดลอกหน้าจอเฉพาะส่วนที่อยู่ใน Desktop เท่านั้น

ค.3 คำนวณหาความสูงและความกว้างจริงของ window ของโปรแกรมประยุกต์โดย

ความกว้าง = ค่าตำแหน่ง ณ จุด Right - ค่าตำแหน่ง ณ จุด Left

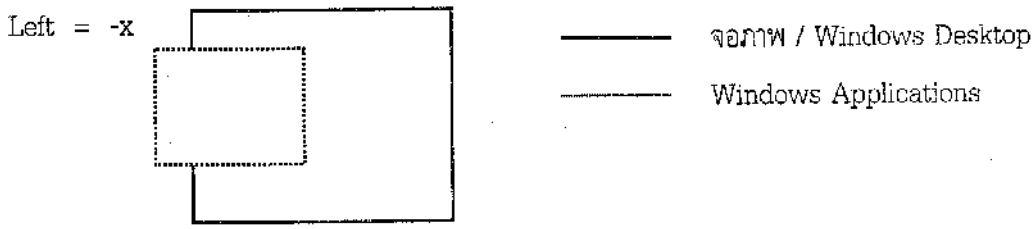
ความสูง = ค่าตำแหน่ง ณ จุด Bottom - ค่าตำแหน่ง ณ จุด Top

ค่าความกว้าง ความสูง ที่คำนวณได้นี้จะนำไปใช้เป็น Header ของ block ข้อมูลภาพที่จะถูกส่งออกไปในเครือข่ายซึ่งจะกล่าวถึงต่อไป



รูปที่ 7(ก) แสดงตำแหน่งกรอบหน้าต่างของโปรแกรมประยุกต์

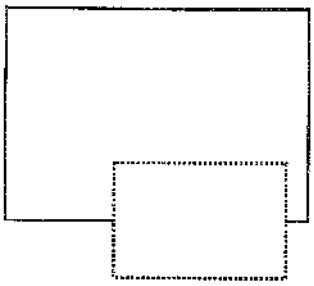
สำหรับ Left ของกรอบ window ของเราเท่ากับ -x ของ Desktop (Left = -x)



เพราะฉะนั้นต้องปรับค่า Left ของกรอบ window ที่จะส่งไปเท่ากับ x ของกรอบ window เดิม

รูปที่ 7(ข) แสดงการปรับตำแหน่งกรอบหน้าต่างของโปรแกรมประยุกต์

สำหรับ Bottom ของกรอบ window ของเรามีค่ามากกว่า Bottom ของ Desktop



Bottom > 600

ดังนั้นจึงปรับค่า Bottom ของกรอบ window ที่จะส่งไปเท่ากับ 600 ซึ่งเป็น Bottom ของ Desktop

รูปที่ 7(ค) แสดงการปรับตำแหน่งกรอบหน้าต่างของโปรแกรมประยุกต์

2.1.2 รูปแบบของ block ข้อมูลภาพที่ส่งผ่านเครือข่าย LAN

เมื่อโปรแกรมทำการคัดลอกหน้าจอของโปรแกรมประยุกต์บนวินโดวส์เข้ามาไว้ในหน่วยความจำแล้วก็จะทำการส่งข้อมูลออกไปในเครือข่าย LAN แต่เนื่องจากว่าเป็นการส่งด้วย UDP ดังนั้นจึงมีข้อจำกัดในเรื่องของขนาดของข้อมูลที่จะถูกส่งออกไปในเครือข่าย เพราะว่า message ที่ถูกส่งออกไปในระดับ Transport Layer จะต้องมีขนาดน้อยกว่า 65,535 ไบต์ (เป็นข้อจำกัดที่มาจากขนาดของ IP datagram ที่จะห่อหุ้มเอา UDP datagram ที่จะส่งออกไป) ในขณะที่รูปภาพหน้าจอที่ได้มีขนาดอย่างน้อย 153,600 ไบต์ ((640x480x4) ÷ 8) ซึ่งคำนวณจากการแสดงผลที่ความละเอียด 640x480 และมี 16 สี (4 bits)

ดังนั้นจะเห็นว่ามีความจำเป็นที่จะต้องแบ่งข้อมูลภาพที่อยู่ในหน่วยความจำออกเป็น block ย่อย แล้วจึงทำการส่งออกไปในเครือข่าย สำหรับรูปแบบของ block ข้อมูลภาพที่จะถูกส่งผ่านเครือข่าย LAN จะมีขนาด 8,000 ไบต์ ซึ่งประกอบไปด้วย 2 ส่วน แสดงได้ดังรูปที่ 8

(1) ส่วนที่เป็น Header มีขนาด 13 ไบต์ โดยแบ่งเป็นส่วนย่อย 5 ส่วนคือ

(1.1) P เป็นส่วนเริ่มต้นของ block มีค่าเป็น ASCII 'P' เสมอเพื่อบอกว่าเป็น block ของข้อมูลภาพ มีขนาด 1 ไบต์

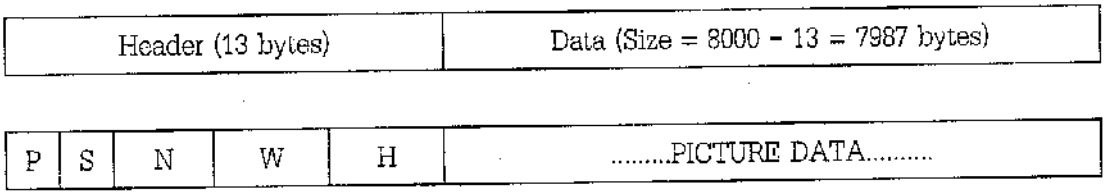
(1.2) S แสดงหมายเลข synchronization ที่จะใช้ร่วมกับข้อมูลเสียง โดยจะมีค่าเป็น ASCII ของ '0' - '9' ซึ่งมีขนาด 1 ไบต์

(1.3) N แสดงหมายเลขของ block ข้อมูลว่าเป็น block ข้อมูลลำดับที่เท่าไร มีค่าตั้งแต่ '000' - '999' มีขนาด 3 ไบต์

(1.4) W ใช้เพื่อบอกให้ฝั่งรับทราบว่าภาพที่ได้รับมีความกว้างกี่ pixels มีขนาด 4 ไบต์

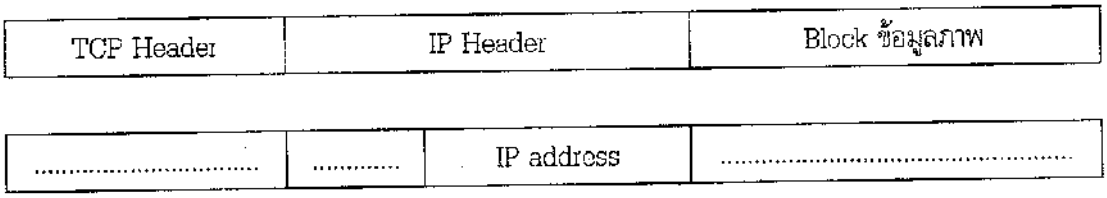
(1.5) H ใช้เพื่อบอกให้ฝั่งรับทราบว่าภาพที่ได้รับมีความสูงกี่ pixels มีขนาด 4 ไบต์

(2) ส่วนที่เป็น Data จะเป็น array of bytes ของข้อมูลภาพที่จะถูกแบ่งออกเป็นส่วนย่อยเพื่อที่จะทำการส่งออกไป



รูปที่ 8(ก) แสดงองค์ประกอบของ block ข้อมูลภาพซึ่งมีความยาวทั้งหมด 8,000 ไบต์

block ข้อมูลนี้จะถูกใส่ไว้ใน packet ของ TCP/IP ดังแสดงในรูปที่ 8(ข)



รูปที่ 8(ข) แสดง packet ของ TCP/IP ที่ห้อหุ้ม block ข้อมูลเอาไว้

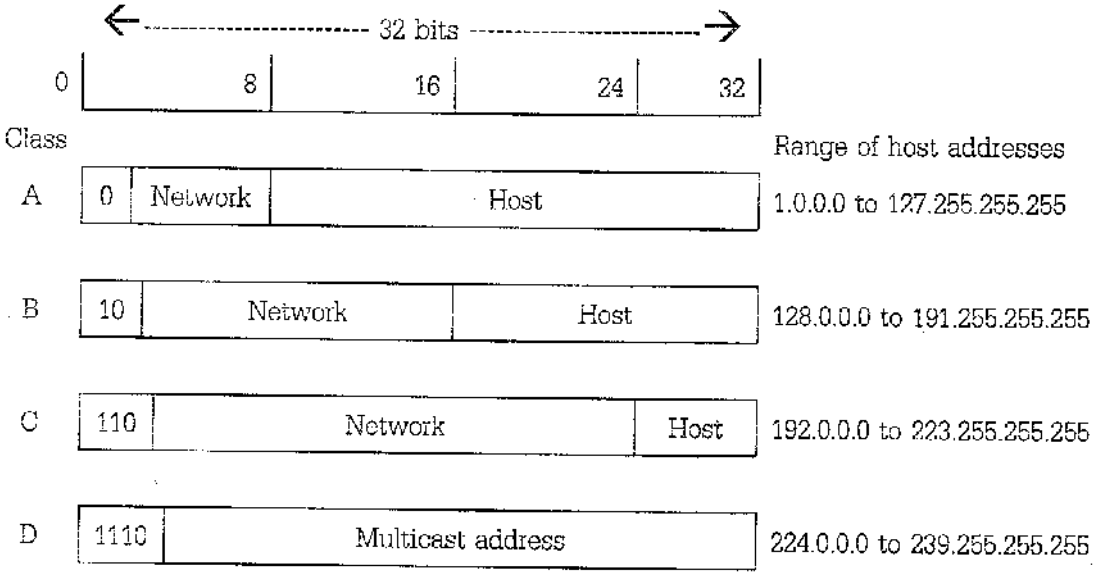
2.1.3 การส่งข้อมูลแบบ broadcast ข้ามเครือข่าย LAN

ในการเรียนการสอนบางครั้งผู้สอนกับผู้เรียนอาจจะอยู่กันคนละเครือข่าย LAN ซึ่งโดยปกติแล้วถ้าเป็นการใช้ TCP/IP ผู้ส่งข้อมูลเพียงทราบว่าผู้รับมี IP address อะไร ก็ส่งข้อมูลไปตาม address นั้น แต่ถ้าเป็นการส่งแบบ UDP/IP ในลักษณะ broadcast แล้วจะต้องมีเทคนิคในการหา broadcast address ซึ่งจะได้กล่าวต่อไป

ก่อนอื่นขออธิบายถึงเรื่องของ IP address และ Subnet mask พอสังเขปดังนี้

IP address เป็น address ในระดับชั้นเครือข่ายของ Internet Protocol ซึ่งไม่ขึ้นอยู่กับฮาร์ดแวร์ หรือรูปแบบของเครือข่าย (network configuration) หรือเครือข่าย token ring, Ethernet ซึ่งอยู่ในระดับชั้นต่ำกว่า ดังนั้นจึงทำให้สามารถส่งข้อมูลข้ามเครือข่ายต่าง ๆ ทั้ง LAN และ WAN ที่ใช้โปรโตคอล IP ได้ ปกติ IP address จะประกอบด้วย 4 ไบต์ ซึ่งบ่งบอกทั้ง network และ local host ซึ่ง IP address 4 ไบต์นี้จะถูกเขียนแยกแต่ละไบต์ด้วยจุดทศนิยม และแต่ละไบต์จะมีค่าซึ่งบ่งบอกด้วยเลขฐานสิบ เช่น IP address 129.47.6.17 เป็นต้น

สำหรับการกำหนด network address และ host address ยังแบ่งออกได้เป็นชนิดต่าง ๆ คือ class A, class B, class C และ multicast ซึ่งไม่ว่าจะเป็น class ใดก็ตาม ทุก ๆ node ใน network หนึ่งจะต้องมีส่วนของ network address เหมือนกัน นอกจากนั้นจะต้องมีส่วนของ host address แตกต่างกันไปสำหรับแต่ละ node ดังแสดงได้ดังรูปที่ 9



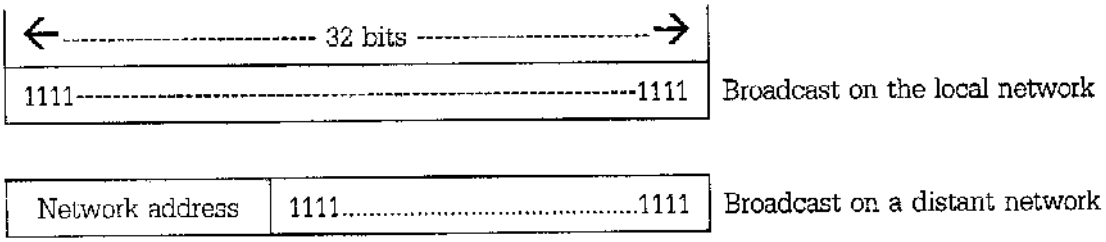
รูปที่ 9 แสดงรูปแบบของ IP address

นอกจากนั้น IP address ต่อไปนี้จะถูกใช้สำหรับวัตถุประสงค์เฉพาะอย่าง

- Network address สำหรับ IP address ที่มีส่วนของ host address มีค่าเป็น 0 จะถูกใช้เป็น network address เช่น 202.44.73.0 เป็น network address ของ class C network ดังนั้น IP ไม่ยอมให้ node ได้มี address เป็น 0
- Broadcast address

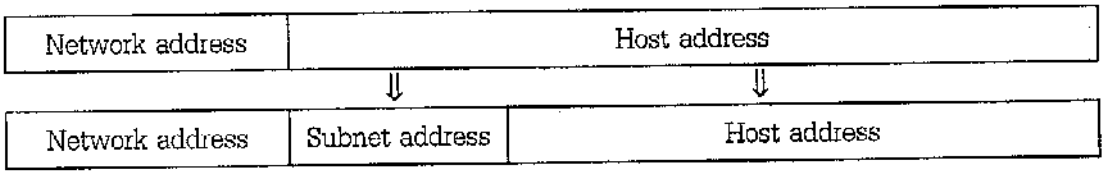
(1) สำหรับ IP address ที่มีค่าเป็น 1 ทั้งหมด จะถูกใช้สำหรับ broadcast packet ไปยังทุก node ของ local network ดังนั้นปกติจะไม่มีการกำหนดให้ node ใดมี IP address เป็น 1 ทั้งหมด

(2) สำหรับ IP address ซึ่งส่วนของ host address มีค่าเป็น 1 ทั้งหมดจะถูกใช้สำหรับ broadcast packet ไปยังทุก node ใน network ที่ค่า address ถูกกำหนดไว้ในส่วนของ network address



รูปที่ 10 แสดง broadcast address

สำหรับเครือข่ายที่ใช้โปรโตคอล IP ในการส่งข้อมูลผ่านเครือข่ายนั้น เพื่อเพิ่มประสิทธิภาพของการส่งข้อมูลในเครือข่ายและเพิ่ม security ภายในเครือข่าย อาจมีความต้องการในการแบ่งเครือข่ายที่มี IP address หนึ่งออกเป็นเครือข่ายย่อยหลายวง ซึ่งมีหลักการในการกำหนดแอดเดรสของเครือข่ายย่อยดังแสดงในรูปที่ 11 กล่าวคือ จะทำการแบ่งส่วนของ Host address ออกเป็นส่วนของ Subnet address และส่วนของ Node address ที่อยู่ใน Subnet นั้น เช่นใช้ 3 บิต สำหรับ Subnet address ของเครือข่าย 202.44.73.0 (หรือ 11001010.00101100.01001001.00000000) ซึ่งเป็น class C แล้วเครือข่ายย่อยจะมีได้ 6 เครือข่าย (001 ถึง 110) ซึ่งมีแอดเดรส 202.44.73.32 ถึง 202.44.73.192 (หรือ 11001010.00101100.01001001.00100000 ถึง 11001010.00101100.01001001.11000000) และมีโหนดภายในเครือข่ายย่อยนี้ได้ 30 โหนด ดังนั้นการแพร่ข้อมูลลงไปในเครือข่ายย่อย 202.44.73.32 ต้องกำหนดแอดเดรสเป็น 202.44.73.63 นั่นคือ 11001010.00101100.01001001.00111111



รูปที่ 11 แสดงการแบ่งส่วนของ Host address ออกเป็น Subnet address และ Host address

ในทางปฏิบัติแล้วจะไม่ทราบแอดเดรสของเครือข่ายย่อยจากระบบปฏิบัติการ แต่จะทราบ IP address ของโหนดภายในเครือข่าย และ Subnet mask ซึ่งมีค่าเป็น 1 ทั้งหมดในส่วนของ network address เช่น Subnet mask ของเครือข่ายย่อย 202.44.73.32 จะมีค่า 255.255.255.224 นั่นคือ 11111111.11111111.11111111.11100000 ซึ่งสามารถหาแอดเดรสของเครือข่ายย่อย โดยเอา Subnet mask ไป AND กับ IP address ของโหนดในเครือข่ายย่อยนั้น

ยกตัวอย่างเช่น ถ้าโหนดใดในเครือข่ายย่อยนี้มี IP address เท่ากับ 202.44.73.34 (หรือ 11001010.00101100.01001001.00100010) จะหาแอดเดรสของ Subnet ที่โหนดนี้ได้จาก

202.44.73.34		11001010.00101100.01001001.00100010
AND	หรือ	AND
255.255.255.224		11111111.11111111.11111111.11100000
202.44.73.32		11001010.00101100.01001001.00100000

เพราะฉะนั้นจะได้แอดเดรสของเครือข่ายย่อยคือ 202.44.73.32

2.2 การนำภาพหน้าจอของนักเรียนไปแสดงบนหน้าจอครู

ในกรณีที่นักเรียนเริ่มเข้ามาใช้งานโปรแกรมนั้น ขั้นแรกจะต้องมีการแจ้งให้กับครูทราบก่อนว่ามี host เพิ่มเข้ามาในระบบการเรียนการสอน โดยการ establish connection ในแบบของ TCP/IP ไปหาครู เพื่อส่งข้อมูลบอกชื่อนักเรียน, IP address ของเครื่องที่นักเรียนใช้ รวมทั้ง Subnet mask ไปให้ครูทราบ (ค่า IP address และ Subnet mask นักเรียนไม่ต้องใส่ เพราะโปรแกรมจะอ่านค่านี้จาก registry ของระบบวินโดวส์เอง ส่วนชื่อนักเรียนจะต้องพิมพ์เข้าไปในหน้าจอ login) เพราะฉะนั้นกรณีนี้จะถือว่านักเรียนเป็น client ที่ initiate request ไปขอลงทะเบียนในระบบ โดยเครื่องครูจะเป็น server ที่คอยรับข้อมูลไปบันทึกในระบบว่ามีนักเรียนคนใดในระบบบ้าง

เนื่องจากว่าในการเรียนการสอนนั้นจะมีนักเรียนจำนวนหลายคน ดังนั้นเป็นไปได้ที่ว่าครูที่เป็น server จะได้รับ request เข้ามาพร้อม ๆ กันมากกว่า 1 request เพราะฉะนั้น server ที่ฝั่งครูจะต้องทำงานในลักษณะ concurrent โดย server ฝั่งครูจะสร้าง socket ขึ้นมาใหม่ 1 อัน ต่อ 1 request จากนักเรียน เพื่อใช้เป็นช่องทางในการรับข้อมูลการลงทะเบียนจากนักเรียน เมื่อได้รับข้อมูลแล้วก็จะทำการ reply กลับไปให้นักเรียนทราบว่าได้ทำการบันทึกข้อมูลแล้ว หลังจากนั้นเมื่อ client ที่ฝั่งนักเรียนได้รับ

reply ก็จะขอยกเลิกการติดต่อไป ซึ่งครูก็จะ close socket ที่เปิดขึ้นสำหรับนักเรียนคนนั้นไป ซึ่งลักษณะการทำงานแบบนี้ เรียกว่า Single-Process, concurrent server(TCP) ซึ่งแสดงดังรูปที่ 12

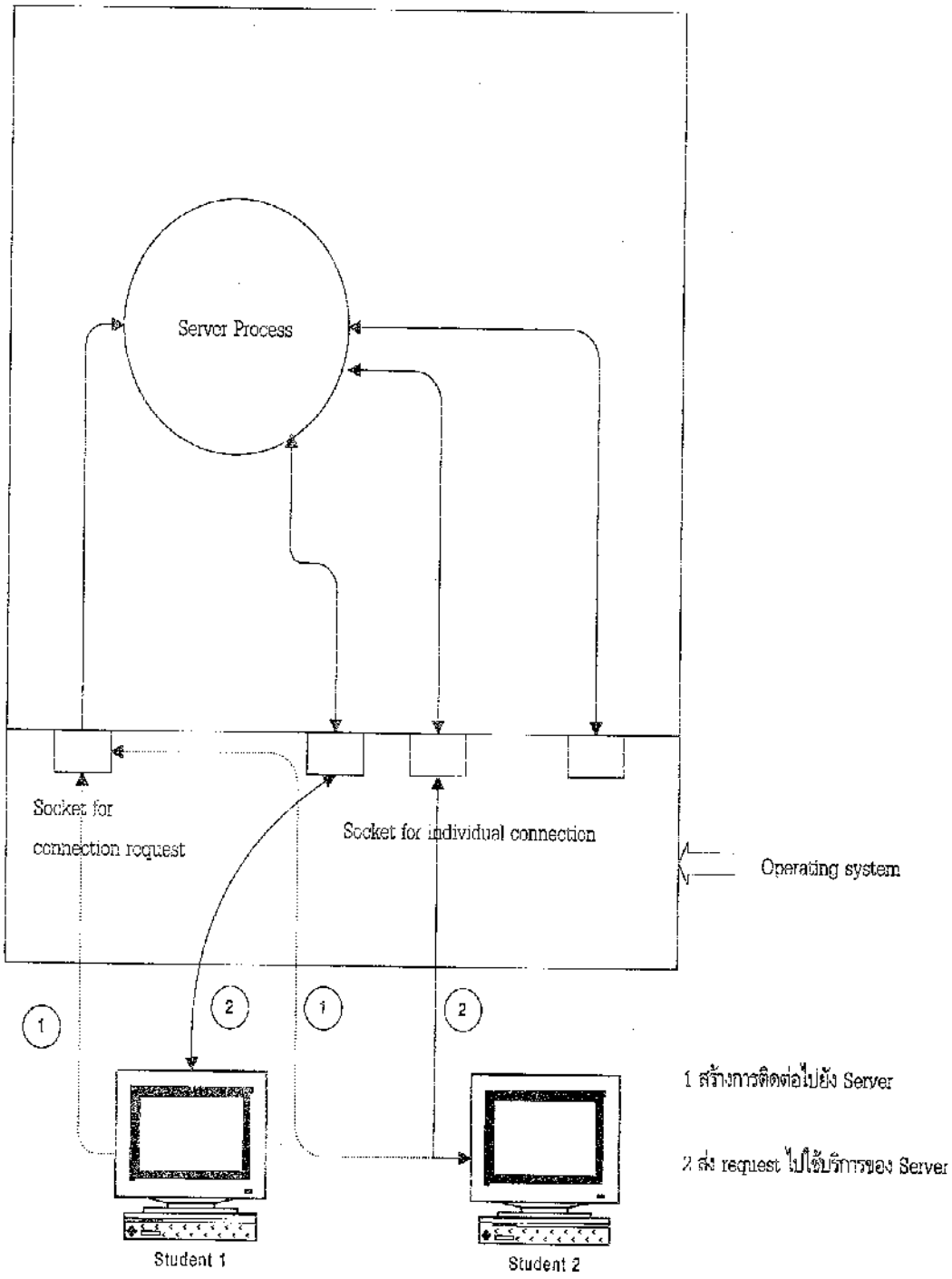
ในบางครั้งครูเองอาจต้องการดูการปฏิบัติตามของนักเรียนว่าถูกต้องหรือไม่ ในการนี้โปรแกรมในส่วนของครูจะทำการส่ง request ไปที่เครื่องของนักเรียนเพื่อขอข้อมูลภาพหน้าจอโปรแกรมในส่วนของนักเรียนก็จะได้รับ request ว่าครูต้องการดูหน้าจอของตน ก็จะทำงานเช่นเดียวกับรูปที่ 3 คือทุก ๆ p หน่วยเวลาจะคอยคัดลอกหน้าจอของโปรแกรมประยุกต์ของนักเรียนส่งออกไปเป็น block ย่อย ๆ ส่งให้กับครู ซึ่งโปรแกรมในส่วนของครูจะทำงานเช่นเดียวกับรูปที่ 5 คือจะอ่าน block ข้อมูลจนกระทั่งครบ 1 ภาพแล้วจึงนำไปแสดงบนหน้าจอของครู

สำหรับการทำงานในส่วนนี้ถือว่าครูเป็น client เนื่องจากเป็นผู้เริ่มส่ง request ไปขอให้นักเรียนส่งภาพหน้าจอมาให้ ส่วนโปรแกรมฝั่งนักเรียนจะทำหน้าที่เป็น server ในการให้บริการส่งข้อมูลหน้าจอของโปรแกรมประยุกต์ออกไป และเนื่องจากเป็นการติดต่อแบบ 1 ต่อ 1 จึงใช้ TCP/IP แทนการ broadcast แบบ UDP

2.3 การจัดการเกี่ยวกับเสียง

เพื่อให้การเรียนการสอนมีประสิทธิภาพ โปรแกรมช่วยในการเรียนการสอนควรจะเป็นระบบมัลติมีเดีย เพื่อจะอธิบายเนื้อหาที่สอนได้อย่างชัดเจนและรวดเร็ว อีกทั้งยังทำให้ครูและนักเรียนสามารถโต้ตอบกันได้ โดยเฉพาะในกรณีที่ครูและนักเรียนอาจอยู่กันคนละสถานที่ซึ่งอาจจะเป็น LAN ต่างวงกัน

ปกติแล้วเสียงเป็นสัญญาณอนาลอก เมื่อต้องการประมวลผลเสียงด้วยคอมพิวเตอร์ จะต้องมีการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล เมื่อประมวลผลแล้วแปลงสัญญาณดิจิตอลให้กลับเป็นสัญญาณเสียงอีกครั้งให้แก่ผู้ฟัง ซึ่ง การ์ดเสียง (Soundcard) โดยทั่วไปจะมีความสามารถในการแปลงสัญญาณเหล่านี้ แต่ในการประมวลผลสัญญาณเสียงนั้น มีปัญหาอย่างหนึ่งคือข้อมูลเสียงที่ได้จากการแปลงสัญญาณอนาลอกนี้มีขนาดโตมาก ซึ่งหากส่งข้อมูลเสียงทั้งหมดระหว่างผู้ใช้ของเครือข่าย จะใช้เวลานานทำให้การโต้ตอบไม่เป็นลักษณะของ real-time นอกจากนั้นยังสิ้นเปลืองแบนด์วิชด์ของระบบส่งข้อมูลด้วย ดังนั้นผู้วิจัยจึงต้องพยายามหาวิธีการบีบอัดข้อมูลเสียงที่ได้ให้มีขนาดเล็กลง ซึ่งผู้วิจัยได้เลือกวิธีการบีบอัดข้อมูลเสียงตามมาตรฐาน MPEG (Motion Picture Experts Group)



รูปที่ 12 แสดงลักษณะการทำงานแบบ Single-Process, concurrent server(TCP)

2.3.1 การบีบอัดเสียงให้เล็กลงด้วยมาตรฐาน MPEG

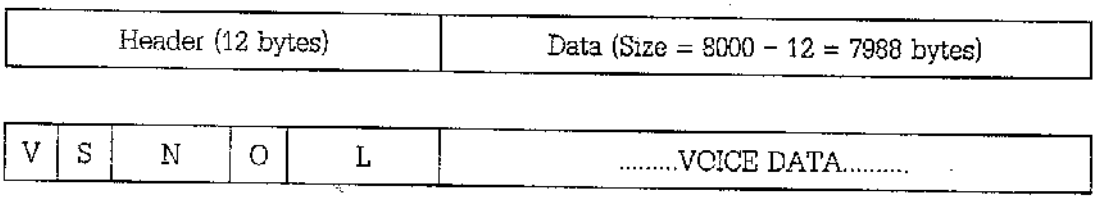
มาตรฐาน MPEG ถือว่าเป็นอัลกอริทึมที่สำคัญในเรื่องของการบีบอัด audio และ video ที่เป็นมาตรฐานสากลตั้งแต่ปี ค.ศ. 1993 ซึ่งมาตรฐานนี้ถูกจัดแบ่งออกเป็น 3 layers ซึ่งแต่ละ Layer ก็จะเพิ่มเทคนิคต่าง ๆ ในการทำการบีบอัดข้อมูลภาพและเสียงให้มากขึ้น เช่น Layer 1 จะได้คุณภาพการบีบอัดภาพและเสียงลงไป 4 เท่า, Layer 2 จะได้คุณภาพการบีบอัดระหว่าง 6-8 เท่า และที่ Layer 3 จะได้ประมาณ 10-12 เท่า

MPEG-1 audio compression จะทำโดยการ sampling waveform ด้วยความถี่ 32 kHz, 44.1 kHz, หรือ 48 kHz ซึ่งสามารถจะบีบอัดได้ทั้งแบบ disjoint stereo (ทำการบีบอัดในแต่ละ channel แยกกันไป) และแบบ joint stereo ซึ่งในการบีบอัดเสียงนั้นถูกทำโดยกระบวนการ fast Fourier transformation กับสัญญาณเสียงเพื่อเปลี่ยนรูปจาก time domain ไปเป็น frequency domain ซึ่งทำให้ spectrum ที่ได้ถูกแบ่งออกไปเป็น 32 frequency bands ซึ่งแต่ละ band ที่ได้จะถูกนำไปทำการเข้ารหัสเพื่อบีบอัดข้อมูลลงไปแยกต่างหากจากกัน ผลลัพธ์ที่ได้จาก MPEG-1 คือ audio stream ที่ส่งออกไปมีอัตราส่งตั้งแต่ 32 KBPS ถึง 448 KBPS ซึ่งขึ้นกับซอฟต์แวร์และฮาร์ดแวร์ที่ใช้

2.3.2 การส่งข้อมูลเสียงจากครูไปยังนักเรียน

นอกจากการบีบอัดข้อมูลให้เล็กลงแล้ว เพื่อให้การส่งข้อมูลเสียงเป็นไปอย่างรวดเร็วและมีความต่อเนื่อง อีกทั้งสอดคล้องกับการส่งข้อมูลภาพหน้าจอ ในงานวิจัยนี้จึงได้ทำการบันทึกเสียงแล้วทยอยส่งออกไปในเครือข่ายเป็นช่วง ๆ ซึ่งมีระยะเวลาสั้น ๆ แล้วส่งไปให้นักเรียนรับฟังไปด้วยในลักษณะเป็น Block stream ของเสียงในแบบ real-time กล่าวคือถ้าครูใช้เวลาบรรยายประกอบภาพหน้าจอของโปรแกรมประยุกต์ประมาณ 1 นาที ซึ่งอาจจะได้ข้อมูลเสียงถึง 1 MB แทนที่จะต้องรอส่งข้อมูลประมาณ 1 MB ไปให้ฟังทีเดียวก็ซึ่งทำให้นักเรียนต้องรอไปถึง 1 นาทีก่อนจึงจะได้รับฟังข้อความที่ครูบรรยายเสร็จไปแล้ว ดังนั้นถ้าเราทำการทยอยบันทึกข้อมูลทุกระยะเวลาสั้น ๆ สมมติว่าทุกระยะเวลา 3 วินาที เราจะได้ข้อมูลประมาณ 66 กิโลไบต์ และถ้าเรานำข้อมูลนี้มาบีบอัดด้วยมาตรฐาน MPEG ลงไปได้ประมาณ 6 เท่า ก็จะเหลือข้อมูลที่จะถูกส่งออกไปประมาณ 11 กิโลไบต์ ซึ่งมีขนาดไม่มาก แล้วค่อยทยอยส่งข้อมูลจำนวนนี้ออกไปเป็น block ข้อมูลเสียงไปให้นักเรียน ก็จะทำให้นักเรียนได้รับฟังคำบรรยายที่ต่อเนื่องกันมากกว่าที่จะต้องรอเป็นเวลานาน

สำหรับรูปแบบของ Block ข้อมูลเสียงที่จะถูกส่งออกไปในเครือข่ายจะมีขนาด block เท่า ๆ กันคือ 8,000 ไบต์ ซึ่งประกอบไปด้วย 2 ส่วนหลัก ๆ ดังรูปที่ 13



รูปที่ 13 แสดงองค์ประกอบของ block ข้อมูลเสียงซึ่งมีความยาวทั้งหมด 8,000 ไบต์

สำหรับส่วนแรกคือ Header นั้นจะมีขนาด 12 ไบต์ โดยแบ่งเป็นส่วนย่อย 5 ส่วนคือ

(1) V เป็นส่วนเริ่มต้นของ block ข้อมูล มีค่าเป็น ASCII ของ 'V' เสมอ เพื่อบอกว่าเป็น block ของข้อมูลเสียง

(2) S แสดงหมายเลข synchronization ที่จะใช้ร่วมกับข้อมูลภาพโดยจะมีค่าเป็น ASCII ของ '0' – '9' มีขนาด 1 ไบต์

(3) N แสดงหมายเลขของ block ข้อมูลว่าเป็น block ลำดับที่เท่าไร มีขนาด 3 ไบต์

(4) O แสดงลำดับของไฟล์เสียงว่าเป็นไฟล์เสียงลำดับที่เท่าใด มีค่าเป็น ASCII ของ '0' – '9' มีขนาด 1 ไบต์

(5) L ใช้เพื่อบอกให้ฝั่งรับทราบว่าข้อมูลไฟล์เสียงที่จะส่งมาทั้งหมดสำหรับลำดับของ O มีความยาวเท่าใด มีขนาด 6 ไบต์

การที่ต้องใช้ฟิลด์ O ในการแสดงลำดับของไฟล์เสียงเนื่องจากเราทำการทยอยบันทึกข้อมูลทุกระยะเวลาสั้น ๆ เช่น 3 วินาที หลังจากบีบอัดข้อมูลแล้วจะได้ไฟล์ข้อมูลเสียงประมาณ 11 กิโลไบต์ แล้วทยอยแบ่งออกเป็น block ขนาด 8,000 ไบต์ส่งออกไป ซึ่งถ้าครูใช้เวลาบรรยายทั้งหมด 1 นาที ก็จะได้ไฟล์เสียงมีจำนวนถึง 20 ไฟล์ที่จะต้องถูกส่งออกไป เพราะฉะนั้นจึงต้องทำการบันทึกลำดับของไฟล์เสียงด้วยว่าไฟล์ใดเป็นเสียงที่ครูบรรยายไปก่อนหน้า เพื่อไม่ให้ผู้เรียนสับสนตอนรับฟัง และสำหรับฟิลด์ L ใช้เพื่อบอกให้ฝั่งรับทราบว่าไฟล์เสียงที่จะส่งออกมาเป็น block ข้อมูลทุกระยะ 3 วินาทีนั้นมีขนาดทั้งหมดเท่าใดจะได้จัดเตรียม buffer ที่มีขนาดเท่ากันไว้คอยจัดเก็บ

สำหรับส่วนหลังของ Header คือ Voice data จะเป็น array of bytes ของไฟล์ข้อมูลเสียงที่ถูกแบ่งออกเป็นส่วนย่อยเพื่อที่จะส่งออกไปในเครือข่าย

2.3.3 การส่งเสียงไป synchronize กับภาพหน้าจอของครู

เนื่องจากในการสอนนั้นเมื่อครูทำการอธิบายประกอบภาพของโปรแกรมประยุกต์ที่แสดงอยู่บนหน้าจอของครู เสียงที่ครูพูดออกไปนั้นเมื่อไปถึงนักเรียนจำเป็นที่จะต้องให้การรับฟังนั้นตรงกันกับรูปที่ครูอธิบายอยู่ เพราะฉะนั้นจึงต้องมีกระบวนการ synchronization ข้อมูลภาพและเสียงให้สอดคล้องกัน โดยทุกครั้งที่โปรแกรมของครูทำการคัดลอกภาพหน้าจอของโปรแกรมประยุกต์เพื่อทำการส่งออกไปเป็น block ของข้อมูลภาพนั้น จะเห็นว่าจากรูปที่ 8 ในส่วนของ Header นั้นโปรแกรมจะมีการใส่ค่าที่ S (synchronization number) ด้วยว่าเป็นเท่าใด ซึ่งเมื่อโปรแกรมใส่ค่าให้ S เป็นเท่าใดแล้ว (ปกติจะใช้การใส่ค่าวนซ้ำระหว่าง '0' – '9' คือเริ่มใส่ที่ '0' พอได้รูปถัดไปจะใส่ '1' เช่นนี้ไปจนถึงรูปที่ '9' แล้วจะวนกลับไปที่ '0' ใหม่) ก็จะจดจำเอาไว้

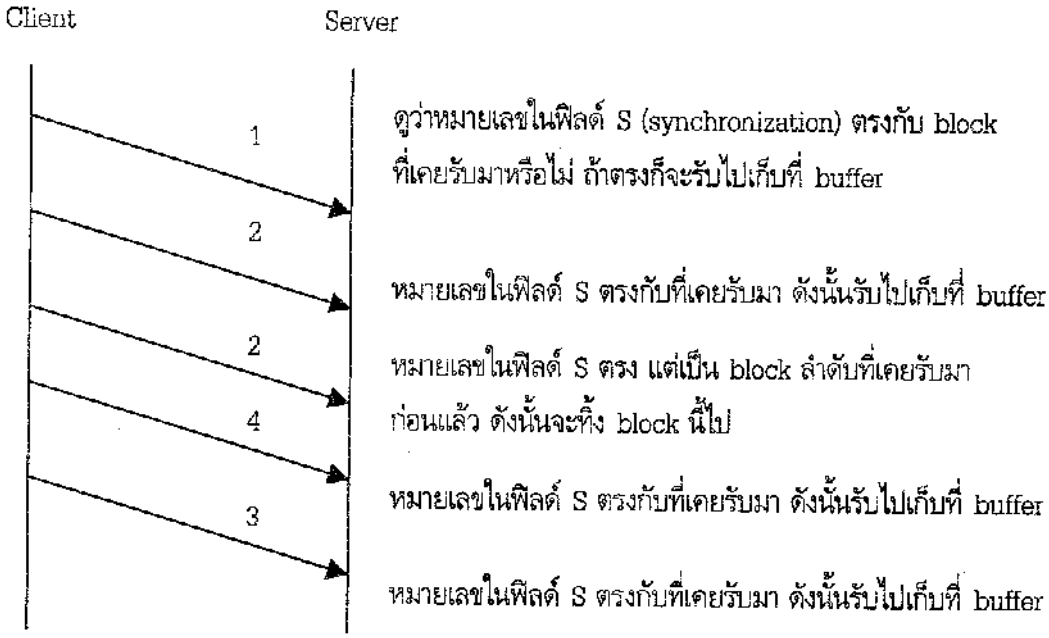
เมื่อครูทำการอธิบายประกอบภาพโปรแกรมก็จะหยุดการจับหน้าจอภาพชั่วคราวจนกว่าครูจะหยุดการพูด (ด้วยการกดปุ่มบอกให้โปรแกรมทราบ) ซึ่งในระหว่างที่ครูทำการพูดอยู่นั้นข้อมูลไฟล์เสียงที่ได้ออกมา ในช่วงระยะเวลาสั้น ๆ ก็จะถูกแบ่งออกเป็น block ของข้อมูลเสียงและจากรูปที่ 13 ข้างต้นจะเห็นว่าในส่วนของ Header ก็จะมีองค์ประกอบย่อยเป็น S เช่นเดียวกับรูปที่ 8 ซึ่งโปรแกรมก็จะนำค่าหมายเลขที่จดจำไว้ก่อนหน้าที่มาใส่ให้กับ block ข้อมูลเสียง เพราะฉะนั้นเมื่อ block ข้อมูลเสียงไปถึงนักเรียน โปรแกรมฝั่งนักเรียนจะดูว่าค่าหมายเลข S ใน block ที่ได้รับมานั้นมีค่าตรงกับหมายเลขที่ได้รับตอนที่ block ข้อมูลภาพหน้าจอของครูถูกส่งมาหรือไม่ ถ้าตรงกันก็จะทำการรวบรวม block ข้อมูลไฟล์เสียงขึ้นมาเป็นไฟล์แล้วทำการเล่นกลับให้นักเรียนฟัง

ดังนั้นด้วยวิธีการเช่นนี้ก็จะทำให้ภาพและเสียงของครูตรงกัน และสำหรับค่าของ O ในส่วนของ Header ในรูปที่ 13 นั้นจะใช้บอกลำดับของไฟล์เสียงด้วยว่าไฟล์เสียงใดต้องเล่นกลับก่อนเพื่อไม่ให้เกิดความสับสนในการฟัง ซึ่งค่าหมายเลขนั้นมีค่าตั้งแต่ '0'-'9' และจะใช้วนกลับเมื่อถึง '9'

2.4 โปรโตคอลในการควบคุมการรับ-ส่งข้อมูลในระดับงานประยุกต์

เนื่องจากงานวิจัยนี้ใช้โปรโตคอล UDP ในการแพร่กระจายข้อมูลมัลติมีเดียจากครูไปยังนักเรียน ซึ่ง UDP เป็นโปรโตคอลแบบไม่ต้องสร้างการติดต่อ จึงทำให้การส่งข้อมูลเป็นไปได้อย่างรวดเร็ว แต่โปรโตคอลนี้ไม่มีการควบคุมการส่งข้อมูล ดังนั้นจึงอาจเกิดความผิดพลาดในการส่งข้อมูลได้ เช่นข้อมูลสูญหาย เสียหาย ข้อมูลซ้ำ หรือข้อมูลไม่เรียงลำดับกัน แต่อย่างไรก็ตามในการส่งข้อมูลผ่านเครือข่าย LAN นั้น LAN มีคุณสมบัติสำคัญคืออัตราความผิดพลาดของข้อมูลต่ำมาก ดังนั้นปัญหาของข้อมูลสูญหายหรือเสียหายแทบไม่มี (หรือหากเกิดข้อมูลเสียหายหรือสูญหายก็ตาม ในกรณีที่ต้องการส่งข้อมูลภาพเคลื่อนไหวหรือข้อมูลเสียงนั้นผู้รับสามารถที่จะทนทานต่อความผิดพลาดเหล่านี้ เช่น ในการพูด

โทรศัพท์ที่อาจจะมีคลื่นรบกวนบางขณะ แต่คนเราก็สามารถจะเข้าใจความหมายโดยรวมได้) แต่ในกรณีที่ส่งข้อมูลข้ามเครือข่ายอาจทำให้เกิดข้อมูลซ้ำ หรือข้อมูลไม่เรียงลำดับกัน ซึ่งทำให้การแสดงภาพและเสียงผิดปกติไปมาก ดังนั้นในการส่งข้อมูลด้วย UDP นี้จึงมีการควบคุมการส่งข้อมูล โดยมีการใช้เลขลำดับควบคุม block ข้อมูล และใช้โปรโตคอลดังแสดงในรูปที่ 14 นอกจากนั้นยังมีการใช้ timer ในการตรวจเช็คว่าภาพที่ได้มาครบหรือไม่หากเกิด timeout ก่อนที่ภาพจะมาครบแสดงว่า client หยุดส่งข้อมูลแล้วจึงนำภาพเท่าที่รับได้ไปแสดงถึงแม้ว่าอาจจะได้รับภาพไม่ครบก็ตาม อีกทั้งถ้าได้รับ block ข้อมูลของภาพใหม่ก็จะนำภาพที่อยู่ใน buffer ไปแสดงแล้วจะเริ่มการรับภาพใหม่เช่นเดียวกับรูปที่ 14 อีก ในกรณีของการส่งข้อมูลเสียงก็จะมีหลักการคล้ายกัน



รูปที่ 14 แสดงโปรโตคอลในการควบคุมการรับ-ส่งข้อมูลในระดับงานประยุกต์

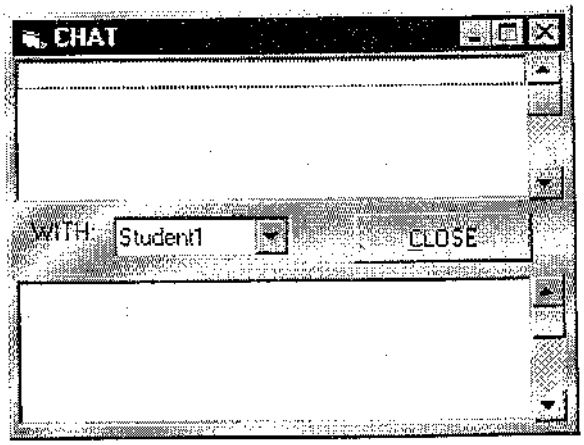
2.5 การโต้ตอบกันระหว่างครูและนักเรียน

เมื่อนักเรียนสงสัยในการสอนของครู นักเรียนก็สามารถที่จะถามครูได้ ซึ่งโปรแกรมมัลติมีเดียฯ นี้จะช่วยให้มีการโต้ตอบกันระหว่างครูและนักเรียนใน 2 ลักษณะดังนี้

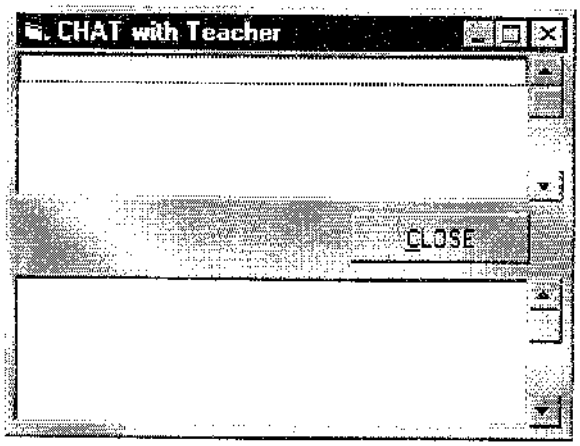
2.5.1 การโต้ตอบกันด้วยข้อความ

สำหรับการโต้ตอบกันด้วยข้อความระหว่างครูและนักเรียน จะทำการเปิด window สำหรับการโต้ตอบ ดังแสดงในรูปที่ 15(ก) และ 15(ข) เมื่อนักเรียนป้อนข้อมูลที่ถามเสร็จแล้ว โปรแกรมในส่วนของนักเรียนจะส่งข้อความผ่านเครือข่ายด้วย TCP ไปยังครู ซึ่งในส่วนของโปรแกรมของครูนั้นก็จะทำงานเป็น server รอรับการติดต่อจากนักเรียนและเมื่อมี packet ข้อมูลเข้ามาก็จะเปิด window ของครูขึ้นเพื่อแสดงข้อความที่ได้รับมาไปที่ chat window ของครู เมื่อครูป้อนข้อมูลตอบโต้กับนักเรียนเสร็จ โปรแกรมจะส่งข้อความนี้ไปที่เครื่องของนักเรียน ซึ่งเมื่อโปรแกรมฝั่งนักเรียนได้รับก็จะนำไปแสดงที่ chat window ของตนเอง

ในกรณีที่นักเรียนหลายคนติดต่อเข้ามาพร้อมกัน เช่น มีการติดต่อเข้ามา 3 คน พร้อมกันก็จะเกิด chat window ขึ้นมา 3 window ที่เครื่องครู ส่วนครูก็จะเลือกว่าจะโต้ตอบกับใครก่อนก็ได้ คนที่ยังไม่ได้รับการโต้ตอบก็จะรอจนกว่าครูจะเลือกโต้ตอบด้วย



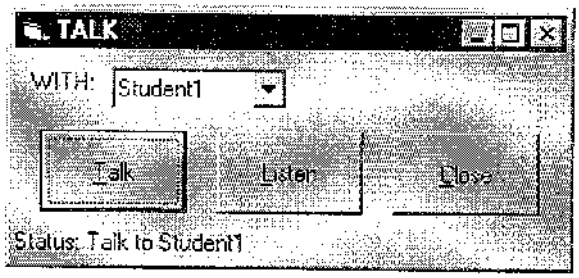
รูปที่ 15(ก) แสดง chat window ของครูสำหรับการโต้ตอบข้อความกับนักเรียน



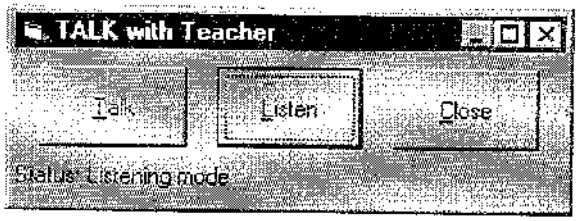
รูปที่ 15(ข) แสดง chat window ของนักเรียนสำหรับการโต้ตอบข้อความกับครู

2.5.2 การโต้ตอบกันด้วยเสียง

สำหรับการโต้ตอบกันด้วยเสียงนั้น จะทำการเปิด window สำหรับควบคุมการโต้ตอบดังแสดงในรูปที่ 16(ก) และ 16(ข)



16(ก)



16(ข)

รูปที่ 16 แสดง window ควบคุมการพูดโต้ตอบระหว่างครูกับนักเรียน

เมื่อนักเรียนต้องการสอบถามด้วยการพูดจะต้องเปิดโปรแกรมส่วนควบคุมการพูดขอ งตน ขึ้นมาก่อน ดังรูปที่ 16(ข) และจะต้องทำการกด Request เพื่อบอกให้ครูทราบว่าตนเองต้องการสอบถาม สำหรับโปรแกรมที่ฝั่งครูเมื่อได้รับ request ก็จะแสดง List ของนักเรียนที่ต้องการพูดโต้ตอบกับครู ถ้าครู เลือกโต้ตอบกับนักเรียนคนใดก็จะเปิด window ควบคุมดังรูป 16(ก) เมื่อครูจะพูดต้องกด Talk เสร็จแล้วจึงจะพูดไปหานักเรียนได้ ส่วนโปรแกรมฝั่งนักเรียนก็จะอยู่ในโหมด Listen เมื่อครูพูดจบก็จะ กด Listen จะทำให้โปรแกรมฝั่งนักเรียนอยู่ที่โหมด Talk นักเรียนจึงจะพูดได้ ซึ่งการพูดโต้ตอบกัน ระหว่างครูกับนักเรียนจะเป็นแบบ half duplex นั่นเอง (ฝั่งหนึ่งพูดอีกฝั่งจะรับฟัง)

3. การออกแบบระบบงานด้วยแนวคิดเชิงวัตถุ

ปกติแล้วในระบบงานที่ออกแบบด้วยแนวคิดเชิงวัตถุ (OOD – Object-Oriented Design) นั้น object ในระบบจะมีการเปลี่ยนสถานะ (state) จากการทำงานหนึ่งไปสู่การทำงานอีกแบบหนึ่ง ต่อเมื่อ มี event เข้ามา ซึ่งในระบบวินโดวส์ก็จะมีการทำงานตาม event (หรือ message) ที่ได้รับมาเช่นกัน ทำให้ การออกแบบระบบงานด้วย OOD บนวินโดวส์ทำได้ง่ายกว่าแบบโครงสร้าง (Structured Design)

ข้อดีอีกอย่างของ OOD คือเรื่องของการนำกลับมาใช้ใหม่ (Reuse) ที่นำเอาวัตถุที่สร้างขึ้นมา แล้วมาใช้อีกในงานอื่น ทำให้นักพัฒนาไม่จำเป็นต้องเริ่มจากไม่มีอะไรเลย ซึ่งจะขอยกตัวอย่างดังรูปที่ 17 จะเห็นว่ามี class อยู่ 3 แบบ คือ Human, Teacher และ Student โดย Human จะมีพฤติกรรมเช่น CaptureScreen, Talk เมื่อพิจารณาจากรูปจะเห็นว่า Teacher กับ Student จะเป็น Subclass ที่ สืบทอดคุณสมบัติและพฤติกรรมของ Human (ซึ่งถือว่าเป็น Superclass) แต่จะเห็นได้ว่า ทั้ง Teacher และ Student ต่างก็มีพฤติกรรมที่แตกต่างกัน เป็นต้นว่าการ Talk สำหรับนักเรียนแล้วจะเป็นการพูด โต้ตอบกับครูเพียงคนเดียว แต่สำหรับครูแล้วการ Talk นี้จะต้องเลือกด้วยว่าจะพูดกับใครก่อนแล้วจึงจะ สามารถพูดได้ ดังนั้นผู้เขียนก็เพียงแต่เขียนเฉพาะส่วนที่แตกต่างไปก็พอ แต่สำหรับพฤติกรรม CaptureScreen นั้น ทั้ง Teacher และ Student จะทำงานเหมือนกัน เพราะฉะนั้นไม่จำเป็นต้องไปเขียน โปรแกรมในส่วนของการ CaptureScreen อีก

สำหรับขั้นตอนโดยทั่วไปในการออกแบบระบบงานด้วยแนวคิดเชิงวัตถุนั้นจะประกอบไปด้วย 3 ขั้นตอนที่สำคัญคือ

3.1 Analysis Modeling

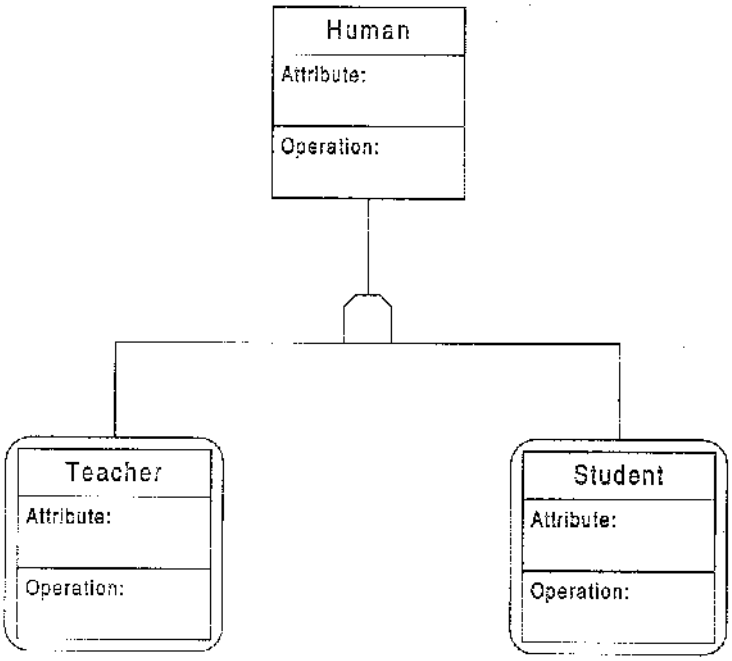
3.2 Design Modeling

3.3 Implementation Modeling

3.1 Analysis Modeling

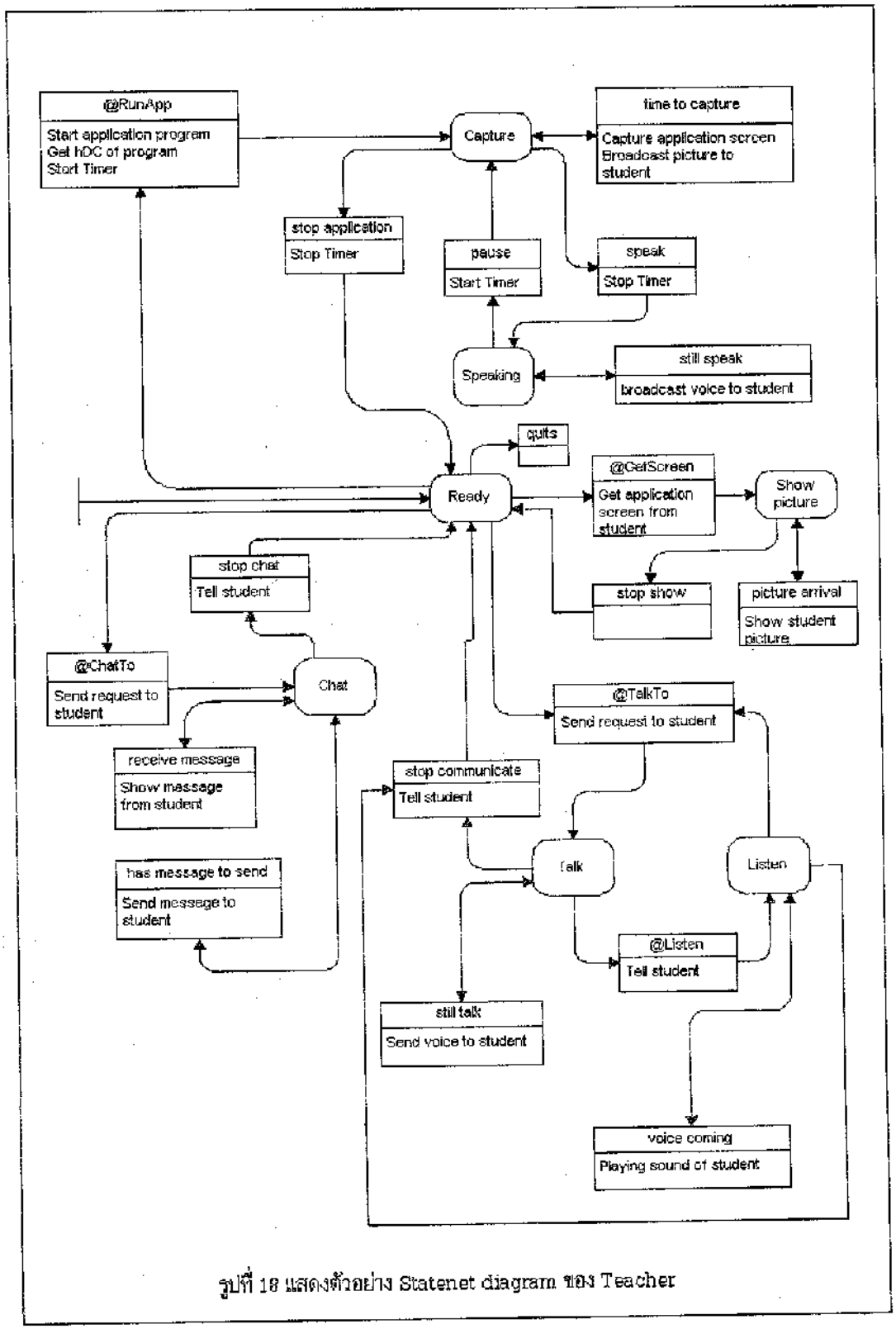
ในขั้นตอนนี้จะวิเคราะห์ระบบด้วยการจำลองระบบขึ้นมา ซึ่งใน Structured analysis ก็คือ การเขียน DFD - Data Flow Diagram แต่สำหรับ OO (Object-Oriented) แล้วจะไม่ใช้การเขียน DFD ซึ่งการจำลองโมเดลจะแบ่งเป็น 2 ส่วนคือ

3.1.1 Structure analysis (การสร้างตัวแบบโครงสร้าง) ซึ่งจะดูโครงสร้างของระบบทั้งหมดว่าต้องประกอบด้วย object อะไรบ้าง แต่ละ object มีความสัมพันธ์กันแบบไหน ตัวอย่างเช่นในระบบงานนี้จะได้ object ชื่อ Teacher, Student, Application, Voice, Timer เป็นต้น และความสัมพันธ์ของ Teacher และ Student จะเป็นดังรูปที่ 17



รูปที่ 17 แสดงความสัมพันธ์แบบ Superclass กับ Subclass

3.1.2 Behavior analysis (การสร้างตัวแบบพฤติกรรม) จะวิเคราะห์ว่าแต่ละ object มีพฤติกรรมเป็นอย่างไร เริ่มต้นตั้งแต่ object เข้ามาในระบบได้อย่างไรแล้วเปลี่ยนสถานะไปอย่างไรบ้าง จนกระทั่งออกไปจากระบบ สำหรับตัวอย่างของการสร้างตัวแบบพฤติกรรมแสดงได้ดังรูปที่ 18



รูปที่ 18 แสดงตัวอย่าง Statenet diagram ของ Teacher

ใน Analysis Modeling นี้สิ่งที่จะไม่มองเลยคือข้อจำกัดของฮาร์ดแวร์, ระบบปฏิบัติการ, ฐานข้อมูล, ภาษาที่ใช้ในการโปรแกรม (Programming Language) จะมองเฉพาะระบบที่เรากำลังจะทำ ไม่มองสภาวะแวดล้อมของระบบ

3.2 Design Modeling

ในขั้นตอนนี้จะทำตัวแบบของระบบงาน ซึ่งจะทำให้ได้

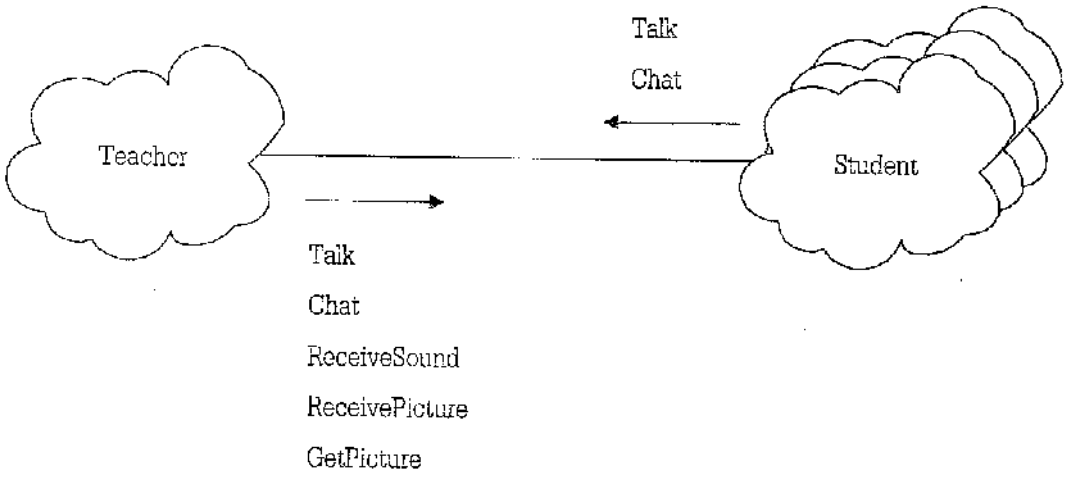
3.2.1 Class specification จะบอกว่าแต่ละ class มี description อย่างไร, attribute, method, operation เป็นอย่างไร

3.2.2 Specification ของแต่ละ operation ต้องได้ algorithm ของแต่ละ method รวมไปถึง interface ในการส่ง message และ parameters นอกจากนี้ต้องออกแบบ User interface ด้วย

3.2.3 Quality ของซอฟต์แวร์ คือดูว่าผู้ใช้มี requirement อย่างไร ต้องออกแบบให้ได้ตามต้องการ

นอกจากนี้ต้อง design ไปถึงการ backup, recovery, error checking, จะใช้ระบบปฏิบัติการตัวไหน มีข้อจำกัดของอย่างไร, ใช้ภาษาอะไร (Programming Language) มีข้อจำกัดหรือไม่, จะใช้ DBMS หรือ file system ในการเก็บข้อมูล และระบบของเราจะเป็นแบบ multitasking หรือ sequential หรือ real-time

Design Modeling นี้จะมองสภาวะแวดล้อมของระบบด้วย และต้อง design ให้ระบบสามารถไปทำงานอยู่ในสภาวะแวดล้อมนั้นได้



รูปที่ 19 แสดงตัวอย่างบางส่วนของการส่ง message ระหว่าง Teacher และ Student

3.3 Implementation Modeling

ในขั้นตอนนี้จะดูว่าจะ implement object ได้อย่างไร, ฮาร์ดแวร์เป็นอย่างไรบ้าง และแต่ละ object จะไปอยู่ที่ไหนบ้าง ตัวอย่างเช่น เครื่องของครูก็จะต้องมี object Teacher ส่วนเครื่องของนักเรียนจะต้องมี object Student และทั้งเครื่องครูและนักเรียนจะมี object Mouse, Voice, Application เป็นต้น ส่วนเครือข่ายจะเป็นแบบ LAN ใช้โปรโตคอล TCP/IP ฮาร์ดแวร์ที่ใช้ก็จะมีเครื่องไมโครคอมพิวเตอร์ที่มีชุดมัลติมีเดีย

บทสรุป

บทความนี้เสนอผลงานวิจัยเพื่อพัฒนาโปรแกรมมัลติมีเดียช่วยในการเรียนการสอนระยะใกล้ของซอฟต์แวร์ภายใต้วินโดวส์ โปรแกรมนี้ช่วยทำให้การเรียนการสอนระหว่างครูและนักเรียนเป็นไปได้อย่างมีประสิทธิภาพ โดยจะมีการส่งภาพการสอนของครูที่ปรากฏบนกรอบหน้าต่างของจอภาพของครูให้แก่นักเรียน ซึ่งภาพนั้นจะปรากฏบนกรอบหน้าต่างในส่วนของครูบนจอภาพของนักเรียน ทำให้นักเรียนสามารถเปิดกรอบหน้าต่างส่วนของตนเอง เพื่อจะปฏิบัติตามครูได้อย่างถูกต้อง นอกจากนั้นเพื่อให้การสอนเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น จึงมีการส่งเสียงคำอธิบายของครูไปด้วย ในการส่งข้อมูลมัลติมีเดียจากครูไปสู่นักเรียนหลายคนพร้อมกันนั้น ได้อาศัยโปรโตคอล UDP ในการแพร่กระจายข้อมูลจากผู้ส่งถึงผู้รับหลายคนพร้อม ๆ กัน ซึ่ง UDP เป็นโปรโตคอลที่ไม่ต้องสร้างการติดต่อระหว่างผู้ใช้จึงสามารถส่งข้อมูลได้อย่างรวดเร็วและเหมาะสำหรับการส่งข้อมูลแบบแพร่กระจาย แต่อย่างไรก็ตามโปรโตคอลนี้ไม่มีการควบคุมความผิดพลาดของการส่งข้อมูล ในงานวิจัยจึงต้องเพิ่มกฎเกณฑ์ในระดับชั้นงานประยุกต์ เพื่อควบคุมความผิดพลาดที่อาจเกิดขึ้นในเครือข่ายระยะใกล้

นอกจากนั้นโปรแกรมนี้ยังทำให้นักเรียนสามารถโต้ตอบกับครูได้ทั้งข้อความและเสียง อีกทั้งครูสามารถดึงภาพบนจอของนักเรียนมาตรวจดูได้ ซึ่งในการส่งข้อมูลลักษณะนี้งานวิจัยนี้ได้ใช้โปรโตคอล TCP ซึ่งเหมาะสำหรับการติดต่อระหว่าง 2 จุด ในการควบคุมการส่งข้อมูล

ในการส่งข้อมูลมัลติมีเดียในลักษณะของ real-time นั้น มีปัญหาที่ต้องการความรวดเร็วและความต่อเนื่องของการส่งข้อมูล เพื่อแก้ปัญหานี้ ในงานวิจัยนี้ได้มีการบีบอัดขนาดของข้อมูลเสียงด้วยมาตรฐาน MPEG อีกทั้งยังใช้หลักการของการแบ่งข้อมูลเสียงเป็นส่วนสั้น ๆ เพื่อสามารถส่งข้อมูลในลักษณะของ data stream

ในการออกแบบระบบงานนั้น ได้ใช้แนวคิดเชิงวัตถุซึ่งจะสอดคล้องกับการพัฒนาโปรแกรมบนระบบวินโดวส์ที่จะทำงานตาม event ที่เกิดขึ้นในระบบ นอกจากนั้นยังสามารถได้ใช้การนำกลับมาใช้ใหม่ (Reuse) ที่นำเอาวัตถุที่สร้างขึ้นมาแล้วมาใช้อีกในงานอื่น ทำให้นักพัฒนาไม่จำเป็นต้องเริ่มจากไม่มีอะไรเลย

บรรณานุกรม

1. เยาวดี เต็มธนาภัทร์, สุชาติ วุฒิสิวะชาติกุล และพิพัฒน์ หิรัณย์วณิชชากร, "โปรแกรมช่วยในการเรียนการสอนระยะไกล้", การประชุมทางวิชาการ สถิติประยุกต์ ครั้งที่ 8, 2533.
2. พิพัฒน์ หิรัณย์วณิชชากร, ระวีวรรณ เอื้อพันธ์วิริยะกุล, ศักดิ์ชัย เอี่ยมกิจสัมฤทธิ์ และคณะ, "โปรแกรมช่วยในการเรียนการสอนระยะไกล้ภายใต้วินโดวส์", คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์, 2539.
3. Andrew S. Tanenbaum, Computer Networks, 3rd edition, Prentice-Hall Inc., 1996.
4. Edward Yourdon, Object-Oriented Systems Design, An Integrated Approach, Prentice-Hall Inc., 1994.
5. Embley, David W., Object-Oriented Systems Analysis A Model-Driven Approach, Yourdon Press Computing Series, 1992.
6. Booch, Grady, Object Oriented Design with applications, The Benjamin/Cummings Publishing Company, Inc., 1991.
7. Douglas E. Comer, Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture, 2nd edition, Prentice-Hall Inc., 1991.
8. Douglas E. Comer, and David L. Stevens, Internetworking with TCP/IP, Volume II: Design, Implementation, and Internals, Prentice-Hall Inc., 1991.
9. Douglas E. Comer, and David L. Stevens, Internetworking with TCP/IP, Volume III: Client-Server Programming and Applications, BSD Socket Version, Prentice-Hall Inc., 1993.
10. Jeffrey Richter, Advanced Windows, 3rd edition, Microsoft Press, 1997.