โปรโตคอลของโครงข่ายบริการร่วมระบบดิจิตอล

ISDN PROTOCOL

โดย

นายประพันธ์ วัจฉลพงษ์

นายพิชิต แก้วมาคูณ

นางสาวรัชฎาวรินณ สุดแสง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032597

ปริญญานิพนธ์ปีการศึกษา 2535

เรื่อง โปรโตคอลของโครงข่ายบริการร่วมระบบดิจิตอล (ISDN PROTOCOL)

ผู้จัดทำ

1.นายประพันธ์ วัจฉลพงษ์ 321170

2.นายพิชิต แก้วมาคูณ 321213

3.นางสาวรัชฎาวรรณ สุดแสง 321257

(รศ.ถวิล พึ่งมา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032597

Title : ISDN Protocol

Name : PRAPAN WATCHALAPONG

PICHIT KEAWMAKOON

RUCHADAWAN SUDSEANG

Department : Telecommunication Engineering

Faculty : Engineering

Advisor : ASSOC.PROF. TAWIL PUANGMA

Abstract

This thesis presents software programming and development for com--puter communication in Integrated Service Digital Network (ISDN), which is applied with a protocol analyzer for analytical process. Therefore, a pro--tocol analyzer is simulated to operate as ISDN network, to analyze the data transferred between terminal equipments, which in this case, are microcompu--ters. In the analytical process, suppose that the data have not error for allowing us to know the pattern, for example type and sequence of the process. The pattern, type and sequence of the transferring data can be easily under--stood. Authough, the cause of error might be occured during the communica--tion, but it can be easily detected.

This software is executed with M-card64 (Terminal Adapter) and can communicate between computers in both circuit and packet switched types. It can also use for telephone calling and sending data in circuit switched or packet switched types.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเรื่อง โปรโตคอลของโครงข่ายบริการร่วมระบบดิจิตอล

ISDN Protocol

ชื่อ นายประพันธ์ วัจฉลพงษ์

นายพิชิต แก้วมาคูณ

นางสาวรัชฎาวรรณ สุดแสง

ภาควิชา วิศวกรรมโทรคมนาคม

คณะ วิศวกรรมศาสตร์

อาจารย์ที่ปรึกษา รศ.ถวิล พึ่งมา

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการพัฒนาโปรแกรมที่ใช้ในการติดต่อสื่อสาร ระหว่างเครื่องคอมพิวเตอร์ในโครงข่ายบริการร่วมระบบดิจิตอล (ISDN) ซึ่งใช้เครื่องโปรโตคอลอนาไลเซอร์มาช่วยในการวิเคราะห์โดยได้ทำการเลียนแบบเครื่องโปรโตคอลอนาไลเซอร์ให้ทำหน้าที่เป็นโครงข่าย ISDN ซึ่งจะทำให้สามารถวิเคราะห์ข้อมูลที่มีการส่งระหว่างอุปกรณ์สื่อสารปลายทาง ซึ่งในที่นี้ใช้ไมโครคอมพิวเตอร์ ซึ่งการวิเคราะห์ข้อมูลของโปรโตคอลอนาไลเซอร์นั้น จะถือว่าข้อมูลของผู้ใช้ขณะส่งและรับไม่มีการผิดพลาดเกิดขึ้น จึงเป็นส่วนที่ช่วยให้ทราบถึงรูปแบบ ชนิด และลำดับขั้นตอนในการติดต่อของข้อมูลที่ใช้ในการสื่อสาร ซึ่งจะทำให้เข้าใจถึงรูปแบบและชนิดของข้อมูลที่ส่งตลอดจนถึงการพิจารณาหาสาเหตุของความผิดพลาดต่าง ๆ ที่อาจจะเกิดขึ้นในขณะทำการติดต่อสื่อสารได้

โปรแกรมนี้สามารถทำการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ได้ทั้งในแบบวงจรสวิตช์ (Circuit Switched) และแพคเก็ตสวิตช์ (Packet Switched) โดยมี M_card64 ทำหน้าที่เป็นเทอร์มินัลอแดปเตอร์ (Terminal Adapter)ให้กับเครื่องไมโครคอมพิวเตอร์ โดยสามารถทำการติดต่อโทรศัพท์และส่งข้อมูลในแบบเซอร์กิตสวิตช์ นอกจากนี้ยังสามารถส่งข้อมูลในแบบแพคเก็ตสวิตช์ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎี	
	2.1 วิวัฒนาการ ISDN	3
	2.2 การอินเตอร์เฟสกับผู้ใช้	5
	2.3 การให้บริการกับ ISDN	7
	2.4 โครงสร้างของระบบการส่งสัญญาณ	8
	2.5 การเชื่อมต่อในระบบ ISDN	10
	2.6 การเชื่อมต่อระหว่างโครงข่าย ISDN กับผู้ใช้บริการ	17
บทที่ 3	การวิเคราะห์โปรโตคอล	
	3.1 การมอนิเตอร์	34
	3.2 การซิมมูเลท	35
	3.3 การซิมมูเลทโดยเครื่องโปรโตคอลอนาไลเซอร์	36
	3.4 การวิเคราะห์ความหมายในโหมดของ History	41
บทที่ 4	หลักการออกแบบซอฟต์แวร์	59
บทที่ 5	การทดลองและผลการทดลอง	
	การเรียกและรับโทรศัพท์	74
	การส่งข้อมูลแบบเซอร์กิตสวิตซ์	77
	การส่งข้อมูลแบบแพคเก็ตสวิตซ์	76
บทที่ 6	สรุป	82
บทแทรก		83
หนังสืออ้างอิง		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในสังคมปัจจุบันการแลกเปลี่ยนข่าวสารข้อมูลที่รวดเร็วเป็นสิ่งจำเป็นอย่างยิ่ง การส่งสารข้อมูลระหว่างเครื่องคอมพิวเตอร์ กระทำได้เฉพาะในพื้นที่ ที่โครงข่ายสื่อสารข้อมูลเข้าไปถึงเท่านั้น ซึ่งก็เป็นที่แน่นอนว่า ยังไม่ครอบคลุมพื้นที่ได้ทั่วถึงเท่าโครงข่ายโทรศัพท์ ดังนั้นในบางพื้นที่ยังคงต้องใช้โครงข่ายโทรศัพท์ในการส่งข้อมูล ทำให้ความเร็วในการส่งข้อมูลช้าลงอย่างมาก จากข้อเสียในจุดนี้จึงได้มีการพัฒนาโครงข่ายบริการร่วมระบบดิจิตอล ISDN ย่อมาจาก Integrated Service Digital Network เป็นโครงข่ายที่ให้บริการโทรคมนาคมทุกชนิด ซึ่งในปัจจุบันแยกให้บริการในแต่ละโครงข่าย มารวมไว้ในโครงข่ายเดียวกันและสามารถให้บริการอุปกรณ์สื่อสารปลายทาง (Terminal Equipment) ทุกชนิดโดยต่อเข้ากับคู่สายโทรศัพท์เพียงคู่สายเดียว สำหรับการสื่อสารจะใช้ระบบดิจิตอลทั้งหมด (End to End Digital Network)

บริการโทรคมนาคมที่ ISDN สามารถให้บริการได้นั้นจะประกอบด้วย บริการที่มีการให้บริการอยู่ในปัจจุบันแล้ว เช่นบริการโทรศัพท์ บริการแพ็คเก็ตสวิตชิ่ง (Packet Switching) บริการเทเล็กซ์ นอกจากนี้ ISDN ยังสามารถให้บริการโทรคมนาคมรูปแบบใหม่ เช่น บริการเทเลเท็กซ์ บริการวีดีโอเท็กซ์ บริการวีดีโอโฟน และการบริการวีดีโอคอนเฟอเรนซ์ เป็นต้น

ในปริญญานิพนธ์ฉบับนี้ ประกอบด้วย 6 บท โดยบทที่ 1 จะเป็นบทนำซึ่ง จะกล่าวเกริ่นนำโครงข่าย ISDN

บทที่ 2 จะเป็นส่วนของทฤษฎี โดยจะกล่าวถึงโปรโตคอลของ ISDN โดยจะกล่าวถึงระดับชั้นฟิสิคัล (Physical layer) โปรโตคอล LAP-D และระดับชั้นเน็ตเวิร์ก (Network layer)

บทที่ 3 กล่าวถึงการวิเคราะห์โปรโตคอลโดยใช้โปรโตคอลอนาไลเซอร์

บทที่ 4 จะกล่าวถึงหลักการออกแบบซอฟต์แวร์ ที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ในโครงข่าย ISDN

บทที่ 5 จะเป็นส่วนของผลการทดลองที่ได้จากการใช้ซอฟต์แวร์ที่เขียนขึ้นทำการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์โดยผ่านโครงข่าย ISDN ซึ่งได้จากการชิมมูเลทเครื่องโปรโตคอลอนาไลเซอร์

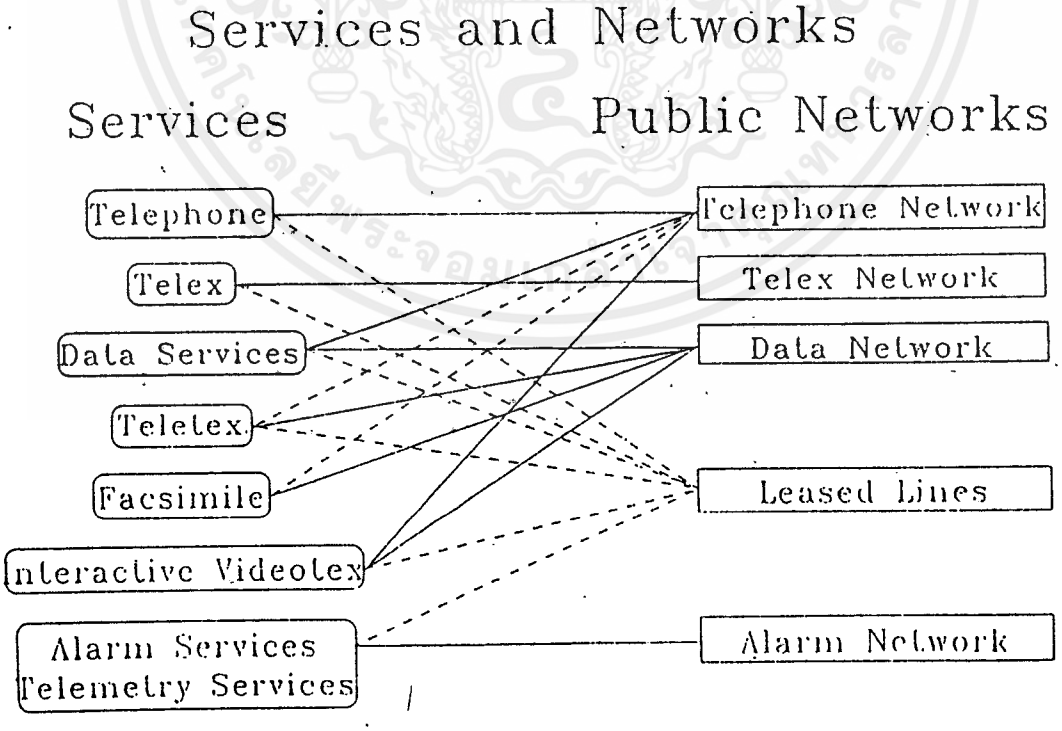
บทที่ 6 จะเป็นบทสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

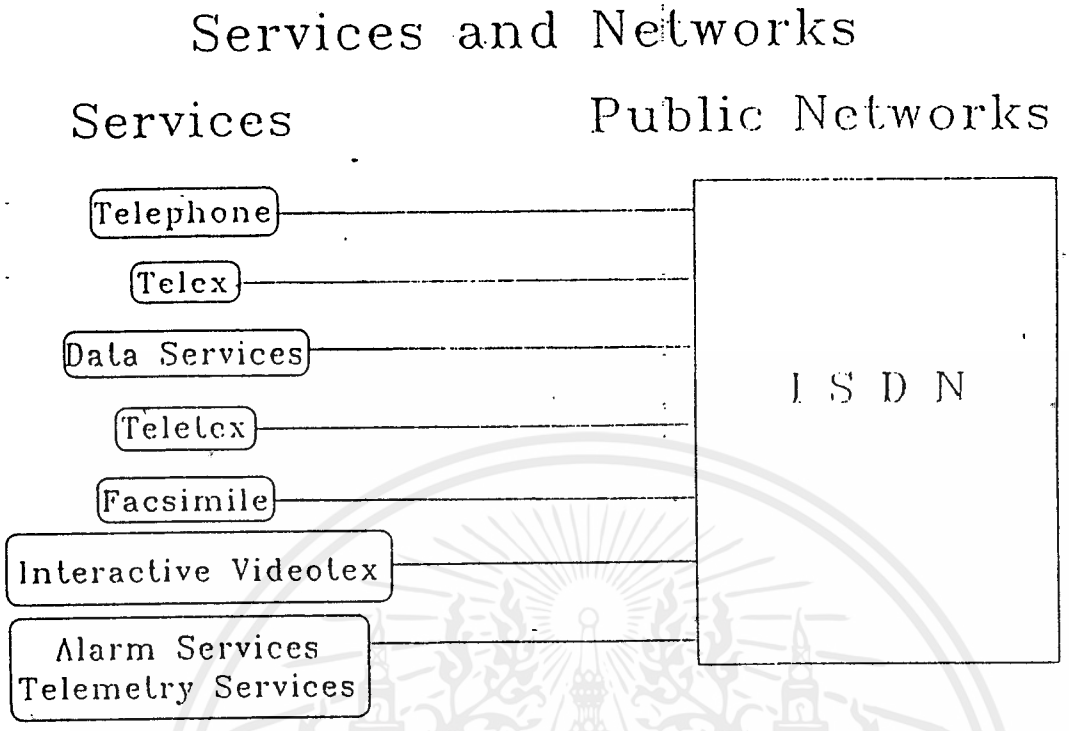
ทฤษฎี

ในอดีตระบบอุปกรณ์ชุมสายและอุปกรณ์สื่อสัญญาณใช้เทคโนโลยีอนาลอกการให้บริการของ โทรศัพท์ โทรเลข โทรสาร ใช้ระบบโครงข่ายที่แยกจากกัน ดังรูปที่ 2.1 ต่อมาได้มีการพัฒนาโครงข่ายโทรศัพท์ให้เข้าสู่เทคโนโลยีทางด้านดิจิตอล เรียกว่า IDN (Integrated Digital Network) ให้บริการทางโทรศัพท์อย่างเดียว สำหรับการส่งข้อมูลที่ไม่ใช่สัญญาณเสียงก็มีการพัฒนาเช่นกันโดยพัฒนาออกเป็นสองแนวทาง คือ Circuit Switching และ Packet Switching จะเห็นได้ว่า การให้บริการทางด้านการสื่อสาร ไม่ว่าแบบใดก็ตามมีระบบโครงข่ายที่แยกจากกันก็ต้องพัฒนามาจากเทคโนโลยีที่แตกต่างกัน ในเรื่องการลงทุนสำหรับจัดการการบริการก็ต้องแยกกันในแต่ละโครงข่ายซึ่งจะใช้ค่าใช้จ่ายมากกว่าโครงข่ายที่สามารถให้บริการได้ทุกประเภทกับอุปกรณ์สื่อสารปลายทางทุกประเภทดังนั้น โครงข่ายที่สามารถให้บริการได้ทุกประเภทนี้จึงเริ่มต้นพัฒนาจากโครงข่าย IDN เนื่องจากเป็นโครงข่ายโทรศัพท์ครอบคลุมพื้นที่มากที่สุด พัฒนาเป็นโครงข่าย ISDN (Integrated Service Digital Network) ดังรูปที่ 2.2 ซึ่งเป็นโครงข่ายที่ใช้ระบบดิจิตอลทั้งหมดสามารถส่งสัญญาณด้วยความเร็วสูง ซึ่งในบทนี้เราจะกล่าวถึงส่วนประกอบต่างๆและการเชื่อมต่อที่จำเป็นในระบบ ISDN



รูปที่ 2.1 แสดงการให้บริการของแต่ละโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการให้บริการของระบบ ISDN

2.1 วิวัฒนาการของ ISDN

ISDN เป็นการวิวัฒนาการจากโครงข่ายร่วมแบบดิจิตอล (IDN) การวิวัฒนาการของโครงข่าย IDN เป็นผลมาจากความต้องการที่จะจัดให้มีการสื่อสารทางเสียงที่ประหยัด อย่างไรก็ตาม ก็ได้มีการรวม การบริการที่มีความเหมาะสมเข้าด้วยกัน เพื่อตอบสนองความต้องการในการบริการ ข้อมูลทางดิจิตอล ต่างๆ ซึ่งตามความหมายของ "I" ในคำว่า ISDN จะเป็นการรวมการบริการต่างๆ ทั้งทางเสียงและข้อมูล ซึ่งพิจารณา จาก CCITT สามารถมองเห็นแนวทางการวิวัฒนาการของ ISDN ดังนี้

1.ISDN ที่วิวัฒนาการจากระบบโทรศัพท์ ความมุ่งหมายของการพัฒนา ISDN จากโครงข่ายโทรศัพท์ที่มีอยู่เดิม สามารถแบ่งได้เป็นสองส่วนคือ ส่วนแรก เทคโนโลยีของ IDN จะนำมาใช้พัฒนาโครงข่ายโทรศัพท์ที่มีอยู่เดิมในรูปของพื้นฐานที่จะใช้ในการจัดเป็นโครงข่าย ISDN ส่วนที่สอง เป็นส่วนที่ให้การบริการพิเศษ เช่นการสนทนาร่วม 3 บุคคล

2.การเปลี่ยนแปลงไปสู่ ISDN จะเปลี่ยนแปลงอย่างช้าๆ ซึ่งการสับเปลี่ยนการทำงานที่ซับซ้อนกว่าจาก พื้นฐานทางเทคนิคหนึ่งไปยังอีกเทคนิคที่ใหม่กว่า จึงต้องมีการทำงานร่วมกันระหว่างเทคโนโลยีเดิมกับเทคโนโลยีใหม่ ทำให้มีความจำเป็นที่ต้องมีการแปลงโปรโตคอลในการติดต่อระหว่างเทคโนโลยีเก่า และเทคโนโลยีใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การใช้เครงข่ายเดิมที่มีอยู่ จะเป็นส่วนที่ละเอียดอ่อนกว่าในขั้นที่ 2 มาก ตัวอย่างเช่น ISDN ที่จัด การบริการแพคเกตสวิตช์ ดังนั้นจำเป็นต้องมีการ อินเตอร์เฟส การบริการด้วย X.25 โดยการนำ แพคเก็ตสวิตชิ่งความเร็วสูง และการควบคุมแบบเวอร์ชวล (virtual call) เพื่อที่จะสามารถใช้อิน เตอร์เฟสกับการบริการใหม่ในอนาคต

4.การจัดการระหว่างผู้ใช้กับเครงข่ายชั่วคราว จะเป็นการพิจารณาส่วนของลูปผู้ใช้ที่เป็นแบบดิ จิตอล ที่มีจำนวนไม่มากนัก ซึ่งอาจจะทำให้การนำเอาบริการทางดิจิตอลมาใช้ต้องล่าช้าออกไป

5.การเชื่อมต่อกับจุดอื่นที่ใช้ความเร็วนอกเหนือจาก 64 kbps เนื่องจากสัญญาณเสียงในรูปแบบ ดิจิตอลอาจทำให้ความเร็วมีค่าสูงหรือต่ำกว่า 64 kbps ดังนั้น ISDN จะต้องมีความสามารถรับส่งสัญ ญาณความเร็ว นอกเหนือจาก 64kbps

หลักการสำหรับ ISDN ได้ถูกนิยามโดย CCITT ซึ่งสามารถสรุปได้ดังนี้

1.รับรองการใช้งานด้านเสียงและไม่ใช่เสียง โดยการกำหนดมาตราฐาน ซึ่งหลักการนี้หมายถึง ทั้งวัตถุประสงค์ และความหมายของการทำให้ ISDN สำเร็จ ISDN สามารถให้บริการต่างๆได้ หลายชนิด ทั้งที่เกี่ยวกับเสียง (โทรศัพท์) หรือการสื่อสารที่ไม่ใช่เสียง (การสื่อสารข้อมูลดิจิตอล) การบริการเหล่านี้จะถูกกำหนดให้ตรงกับมาตราฐาน (กำหนดโดย CCITT) ที่ระบุถึงการอินเตอร์เฟส การส่งข้อมูล

2.รับรองการใช้งานทั้งแบบสวิตช์และแบบไม่สวิตช์ ISDN จะสามารถรองรับการใช้งานในเซอร์ กิตสวิตชิ่ง และแพคเก็ตสวิตชิ่งได้ ซึ่งเป็นกรณีที่มีการสวิตช์ทั้งสองแบบ ส่วนในกรณีที่ไม่สวิตช์ก็จะเป็น การใช้งาน ของสายเช่า(lease line) ที่เป็นการต่อตรงจากจุดหนึ่งไปยังอีกจุดหนึ่งตลอด

3.ความเชื่อถือในการเชื่อมต่อที่ 64 kbps การใช้อัตราเร็วนี้เพราะว่า เป็นอัตรามาตรฐาน ของสัญญาณที่อยู่ในรูปแบบดิจิตอล และโดยเหตุผลนี้จึงได้นำมาใช้ในการวิวัฒนาการ ISDN ด้วย แม้ว่าที่ อัตราขนาดนี้จะสามารถใช้งานได้ แต่ก็อาจเป็นไปได้ว่า ในอนาคตอาจมีการใช้อัตราเร็วที่สูงกว่านี้ เพื่อ ให้ ISDN มีความคล่องตัวสูงขึ้น

4.ความสามารถในการรับรู้ในเครงข่าย การให้บริการใน ISDN ได้มุ่งหวังที่จะสามารถจัดให้มี การบริการใหม่ๆ ที่มากกว่าการเซ็ตอัปการเรียกของเซอร์กิตสวิตชิ่งที่ทำได้ นอกจากนี้การจัดการ เครงข่ายและความสามารถในการจัดบำรุงรักษา จะต้องมีมากกว่าเดิม

5.สถาปัตยกรรมของโปรโตคอล โปรโตคอลที่ถูกพัฒนาเพื่อให้ผู้ใช้สามารถ เข้าสู่ ISDN ได้จะมี การใช้โปรโตคอลในระดับต่างๆ

-เป็นมาตรฐานที่พัฒนาเรียบร้อยแล้ว สำหรับการใช้งานที่มีความสัมพันธ์กับ OSI โมเดลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจถูกนำมาใช้ใน ISDN ตัวอย่างเช่น การใช้ X.25 ในระดับชั้นที่ 3 สำหรับการใช้บริการแพคเก็ตสวิตชิ่ง ใน ISDN

-มาตรฐานใหม่ที่เกี่ยวกับ ISDN สามารถอาศัยมาตรฐานที่มีอยู่แล้วได้ เพื่อเป็นการลดค่าใช้จ่ายในการดำเนินการใหม่ ตัวอย่างเช่น LAP-D จะอาศัยมาตรฐานจาก LAP-B

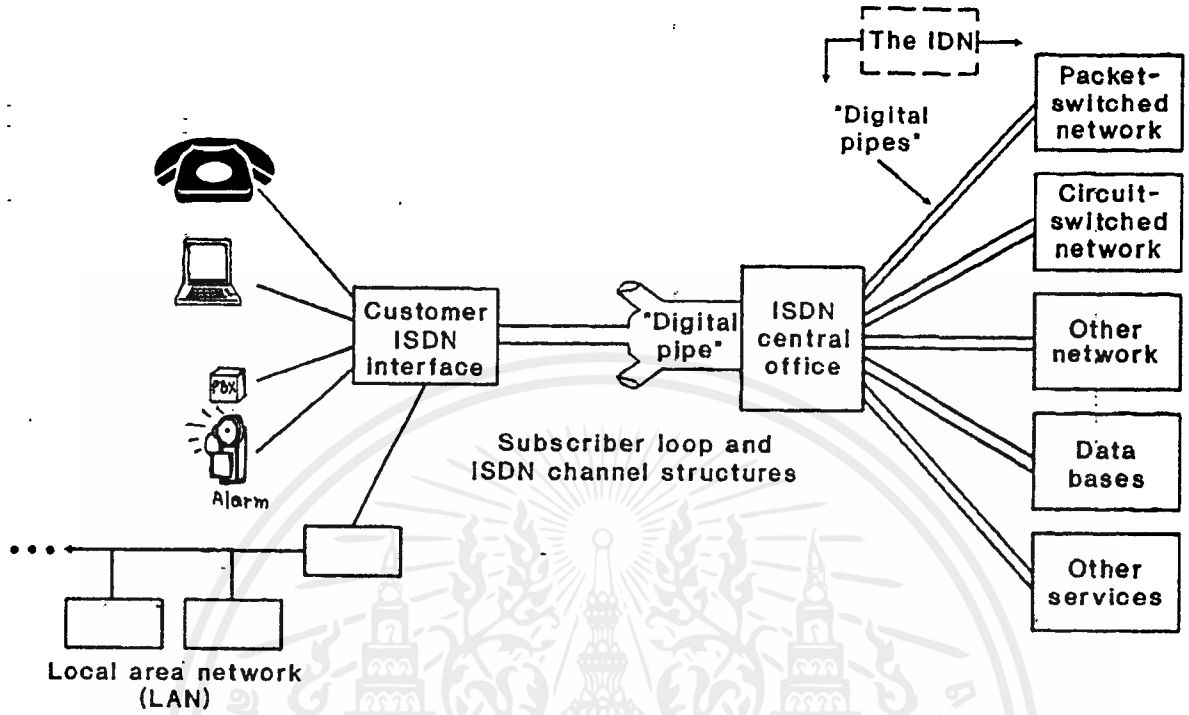
-มาตรฐานสามารถพัฒนาและดำเนินการได้อย่างอิสระ สำหรับระดับชั้นต่างๆ และสำหรับฟังก์ชั่นต่างๆภายในแต่ละชั้น จะยอมให้มีการดำเนินการอย่างค่อยเป็นค่อยไป ในการบริการ ISDN ที่เหมาะสมสำหรับผู้จัดเตรียมที่ที่ถูกกำหนด หรือลูกค้าที่ถูกกำหนดเป็นพื้นฐาน

6.คุณลักษณะหลายๆ แบบ สามารถจัดรูปแบบ ทางพิสิคัลได้มากกว่าหนึ่งแบบ สำหรับดำเนินการ ISDN ซึ่งจะยอมให้มีการใช้งานตามนโยบาย ของแต่ละประเทศแตกต่างกันได้ ตามสถานะของเทคโนโลยี่และความจำเป็นในอุปกรณ์เดิมที่มีอยู่แล้วเป็นพื้นฐาน

2.2 การอินเตอร์เฟสกับผู้ใช้

การพิจารณาแนวความคิด ของ ISDN จากมุมมองจากผู้ใช้สามารถแสดงได้ดังรูปที่ 2.3 ผู้ใช้ที่เข้าสู่ ISDN นั้นหมายถึง การอินเตอร์เฟสสัญญาณจะส่งผ่านท่อดิจิตอล (digital pipe) ที่มีการส่งสัญญาณที่บิทเรทค่าหนึ่ง ค่าดิจิตอลนี้จะมีขนาดแตกต่างกัน ตัวอย่างเช่น ผู้ใช้ที่อยู่ตามบ้าน(ไม่ได้ทำธุรกิจ) อาจจะกำหนดความจุให้เหมาะสมเพียงพอในการใช้งานทางโทรศัพท์ (videdotex) แต่ในสำนักงานที่มีการต่อกับ ISDN โดยผ่าน PBX แบบดิจิตอล และกำหนดให้ท่อดิจิตอลมีความจุได้สูงกว่า

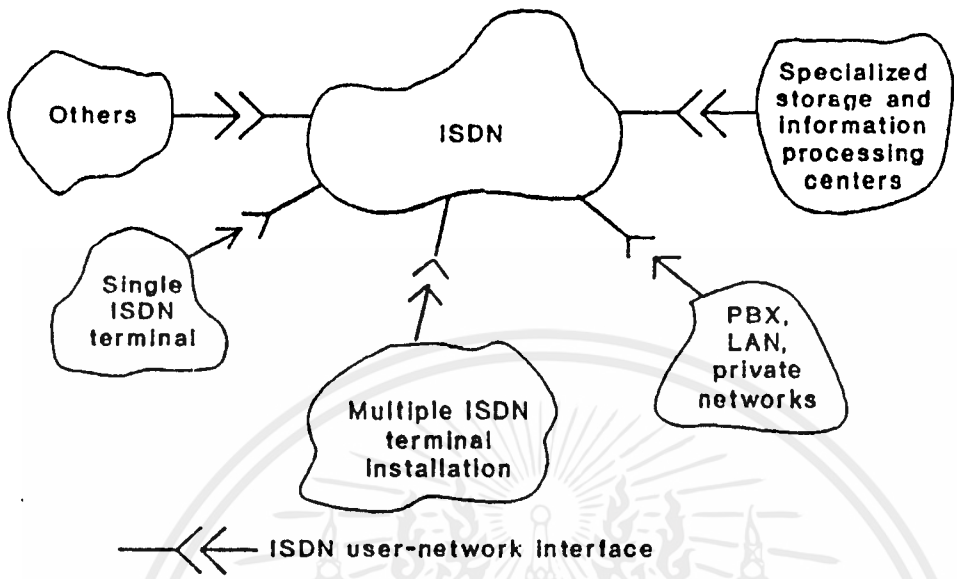
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ลักษณะการเชื่อมต่อของแนวความคิด ISDN

ความต้องการต่ำสุดจะเป็นเทอร์มินอลเดี่ยวเช่นโทรศัพท์ตามบ้านเป็นต้น หรือหลายๆเทอร์มินอลที่มีการให้บริการร่วมกัน เช่น โทรศัพท์ คอมพิวเตอร์ ระบบเตือนภัย เป็นต้น ส่วนการทำงานนั้นจะต่ออุปกรณ์เข้ากับ PBX หรือ LAN แล้วจึงต่อเข้าโครงข่าย ISDN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างการอินเตอร์เฟสระหว่างผู้ใช้กับโครงข่ายใน ISDN

ถ้าให้หลายๆจุดที่เวลาหนึ่ง ท่อดิจิตอลที่ต่ออยู่กับผู้ใช้มีความจุตายตัว แต่ทราฟฟิคบนแต่ละท่ออาจจะมีตัวแปรที่รวมกันขึ้น เพื่อเป็นการกำหนดค่าความจุในท่อนั้น ดังนั้นผู้ใช้อาจเข้าสู่การบริการยืนๆ ISDN จะมีการกำหนดสัญญาณควบคุมที่ค่อนข้างซับซ้อน เพื่อสั่งให้แยกข้อมูลที่มีการมัลติเพล็กอยู่ออกมาได้อย่างไร และจัดเตรียมบริการตามที่กำหนดไว้ในสัญญาณควบคุม ลักษณะที่สำคัญของการอินเตอร์เฟส คือ ในบางเวลาที่ผู้ใช้ใช้งานน้อยกว่าความจุในท่อนั้น และการคิดเงินจะคิดเต็มความจุของท่อนั้น ดังนั้นจึงต้องพยายามใช้ประโยชน์ให้คุ้มค่ามากที่สุด โดยใช้หลักการใช้ร่วมกัน (concentrator) ของส่วนมัลติเพล็กเซอร์และการจัดแบ่งการทำงานในคู่สาย

2.3 การให้บริการ ISDN

ในโครงข่าย ISDN เราจะสามารถจัดการบริการได้หลายรูปแบบ ที่เป็นการรองรับการใช้งานทางด้านเสียงและข้อมูล ซึ่งการบริการที่กำหนดโดย CCITT จะแบ่งออกเป็น 3 ส่วน ที่รับผิดชอบการให้บริการต่างๆ คือ Bearer Service, Teleservice และ Supplementary Service ในการบริการของส่วน Bearer Service จะเป็นส่วนของตัวกลางในการรับส่งข่าวสาร (เสียง ข้อมูล ภาพ ฯลฯ) ระหว่างผู้ใช้ในลักษณะ Real time และจะไม่มีการเปลี่ยนแปลงข้อมูลข่าวสาร การให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริการนี้จะตรงกับ 3 ระดับล่างของ OSI โมเดล ส่วน Teleservice จะรวมฟังก์ชันในการส่งข้อมูลกับฟังก์ชันในการประมวลผลข่าวสารเข้าไว้ด้วยกัน โดยจะมีการใช้ฟังก์ชันในระดับที่สูงขึ้นด้วย ซึ่งก็คือระดับชั้นที่ 4-7 ของ OSI โมเดล ตัวอย่างของ Teleservice ได้แก่ โทรศัพท์ เทเลเทกซ์ วิดีโอเทกซ์ และการส่งข่าวสาร ทั้ง Bearer Service และ Teleservice อาจถูกยกระดับการให้บริการให้ดีขึ้นโดย Supplementary Service ซึ่งเป็นการให้บริการแบบหนึ่งที่มีการใช้บริการร่วมกับการบริการใดบริการหนึ่งหรือมากกว่าหนึ่ง ของบริการแบบ Bearer Service และ Teleservice

2.4 โครงสร้างของระบบการส่งสัญญาณ

ช่องสัญญาณ ISDN

ในการส่งสัญญาณดิจิตอลระหว่างชุมสายโทรศัพท์ดิจิตอลกับผู้ใช้ปลายทาง ISDN จะต้องมีการใช้ช่องสัญญาณในการสื่อสารจำนวนมากเอาไว้ ซึ่งความจุของเส้นทางในการส่งสัญญาณที่ใช้ได้อาจจะมีการเปลี่ยนแปลงไปในผู้ใช้แต่ละราย โครงสร้างในการส่งสัญญาณในเส้นทางต่างๆจะถูกสร้างเป็นช่องสัญญาณที่มีขนาดความจุแตกต่างกันไป ซึ่งสามารถแบ่งออกเป็นช่องสัญญาณต่างๆได้ดังนี้

- ช่องสัญญาณ B มีความจุ 64 kbps
- ช่องสัญญาณ D มีความจุ 16 หรือ 64 kbps
- ช่องสัญญาณ H มีความจุ 384, 1536 หรือ 1920 kbps

1)ช่องสัญญาณ B คือ ช่องสัญญาณที่ผู้ใช้สามารถใช้ในการส่งสัญญาณเสียงที่ถูกทำให้เป็นดิจิตอลโดยใช้วิธีการของ PCM หรือข้อมูลทางดิจิตอลที่มีอัตราต่างๆกัน รวมทั้งสัญญาณเสียงที่มีการเข้ารหัสที่น้อยกว่า 64 kbps ในกรณีที่มีการส่งช่องสัญญาณรวมกันหลายๆช่องสัญญาณ ช่องสัญญาณ B ทั้งหมดจะต้องถูกกำหนดให้ปลายทางเป็นจุดเดียวกัน การเซ็ตอัพบนช่องสัญญาณ B สามารถที่จะใช้การเชื่อมต่อได้หลายประเภท

เซอร์กิตสวิตชิ่ง เปรียบเสมือนสวิตช์แบบดิจิตอลในการให้บริการ ผู้ใช้สามารถที่จะแทนการเรียกและการเชื่อมต่อกับผู้ใช้ที่อยู่ในโครงข่ายอื่น แต่การเซ็ตอัพการเรียกจะไม่กระทำบนช่องสัญญาณ B แต่จะถูกกระทำในช่องสัญญาณร่วม (หรือช่องสัญญาณ D)

แพคเก็ตสวิทชิ่ง ผู้ใช้จะถูกเชื่อมโยงเข้ากับโหนดของแพคเก็ตสวิทชิ่ง และข้อมูลจะถูกแลกเปลี่ยนกับผู้ใช้อื่นๆผ่านทาง X.25

แบบกึ่งถาวร (semipermanent) จะเป็นการเชื่อมต่อกับผู้ใช้อื่นๆ โดยจะมีการกำหนดเส้นทางของสัญญาณไว้ก่อนล่วงหน้าแล้ว และไม่มีการกำหนดโปรโตคอลของการสร้างการเรียก ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทียบได้กับการใช้คู่สายเช่า (leased line)

การกำหนดมาตราฐานอัตราช่องสัญญาณของผู้ใช้เป็น 64 kbps ซึ่งเป็นอัตราที่สัญญาณเสียงที่อยู่ในรูปของดิจิตอลมีประสิทธิภาพดีที่สุด ยังมีการพัฒนาเทคโนโลยีให้มีอัตราเหลือเพียง 32 kbps โดยคุณภาพของเสียงเหมือนกับที่อัตรา 64 kbps เพื่อนำมาใช้ในการส่งสัญญาณเสียงในช่องสัญญาณ B

2) ช่องสัญญาณ D เป็นช่องสัญญาณที่ให้บริการเพื่อจุดประสงค์ 2 ประการ คือ ประการแรกจะใช้ในการนำพาข้อมูลของสัญญาณควบคุมในลักษณะของช่องสัญญาณร่วม เพื่อใช้ในการควบคุมการทำงานของการเรียกในโหมดเซอร์กิตสวิทช์บนช่องสัญญาณ B ที่ต่อรวมกันอยู่ไปยังผู้ใช้ ประการที่ 2 ช่องสัญญาณ D จะสามารถใช้เพื่อทำการส่งข้อมูลข่าวสารในโหมดของแพคเก็ตสวิตช์ได้ ที่เวลาเมื่อไม่มีการส่งข้อมูลควบคุม ซึ่งดังในตารางที่ 2.1 ได้แสดงการเปรียบเทียบทราฟฟิกของข้อมูลในช่องสัญญาณ B และ D

3) ช่องสัญญาณ H ใช้สำหรับข้อมูลของผู้ใช้ที่มีบิทเรทสูงซึ่งผู้ใช้อาจจะใช้ช่องสถานีเป็นทรั้งค์ความเร็วสูงหรือใช้ในการแบ่งช่องสัญญาณในแบบ TDM ทางด้านผู้ใช้เอง เพื่อรองรับอุปกรณ์ได้หลายตัว ตัวอย่างของการใช้งานจะมี โทรสารความเร็วสูง สัญญาณภาพ ข้อมูลความเร็วสูง และสายธารของข่าวสารที่มัลติเพล็กซ์จากข้อมูลที่มีความเร็วต่ำกว่ารวมกัน

ตารางที่2.1 ฟังก์ชันการใช้งานของช่องสัญญาณ B และ D

ช่องสัญญาณ B (64 kbps)	ช่องสัญญาณ D (16 kbps)
1) สัญญาณเสียงดิจิตอล - 64kbps - ที่บิทเรทต่ำ (32 kbps) 2) ข้อมูลความเร็วสูง - เซอร์กิตสวิตช์ - แพคเก็ตสวิตช์ 3) อื่นๆ - โทรสาร - สัญญาณภาพแบบสแกนช้า	1) สัญญาณควบคุม - พื้นฐาน - ระดับสูง 2) ข้อมูลความเร็วต่ำ - วิดีโอเทกซ์ - เทอร์มินอล 3) Telemetry - การบริการแจ้งภัย - การจัดการเกี่ยวกับการใช้พลังงานเช่น น้ำ ไฟฟ้า เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
032597

จากช่องสัญญาณต่างๆเหล่านี้จะถูกจัดเป็นกลุ่มโครงสร้างของระบบการส่งที่จะให้บริการแก่ผู้ใช้ ซึ่งโครงสร้างที่ได้มีการกำหนดให้กับผู้ใช้ในปัจจุบันนี้มีอยู่ 2 รูปแบบคือ

1) BRI (Basic Rate Interface)

2) PRI (Primary Rate Interface)

Basic Rate Interface (BRI) โครงสร้างของการเชื่อมต่อแบบนี้ประกอบด้วย ช่องสัญญาณ "B" 2 ช่องสัญญาณ และช่องสัญญาณ "D" 1 ช่องสัญญาณ หรือเรียกว่า การอินเตอร์เฟสแบบ 2B+D ซึ่งมีอัตราเร็วในการส่งสัญญาณ 144 Kbits/s

Primary Rate Interface (PRI) โครงสร้างของการเชื่อมต่อแบบนี้ ประกอบด้วยช่องสัญญาณ "B" หลาย ๆ ช่องสัญญาณ ช่องสัญญาณ "D" ที่มีความเร็วในการรับส่งข้อมูล 64 Kbits/s อีกหนึ่งช่องสัญญาณ ในปัจจุบันโครงสร้างแบบนี้แบ่งออกเป็น 2 กลุ่ม คือระบบที่รับส่งข้อมูลด้วยอัตราเร็ว 1.544 Mbits/s จะมีโครงสร้างการเชื่อมต่อแบบ 23B+D ใช้กันมากในกลุ่มประเทศอเมริกาเหนือ และระบบการรับส่งข้อมูลด้วยความเร็ว 2.048 Mbits/s ใช้กันมากในกลุ่มประเทศยุโรป ซึ่งประเทศไทยก็มีการใช้การเชื่อมต่อแบบนี้

2.5 การเชื่อมต่อใน ISDN

ใน ISDN ได้มีการจัดให้บริการ การเชื่อมต่อจากต้นทางไปยังปลายทาง 3 ชนิดคือ

1.การเรียกในแบบเซอร์กิตสวิตช์ ผ่านช่องสัญญาณ B

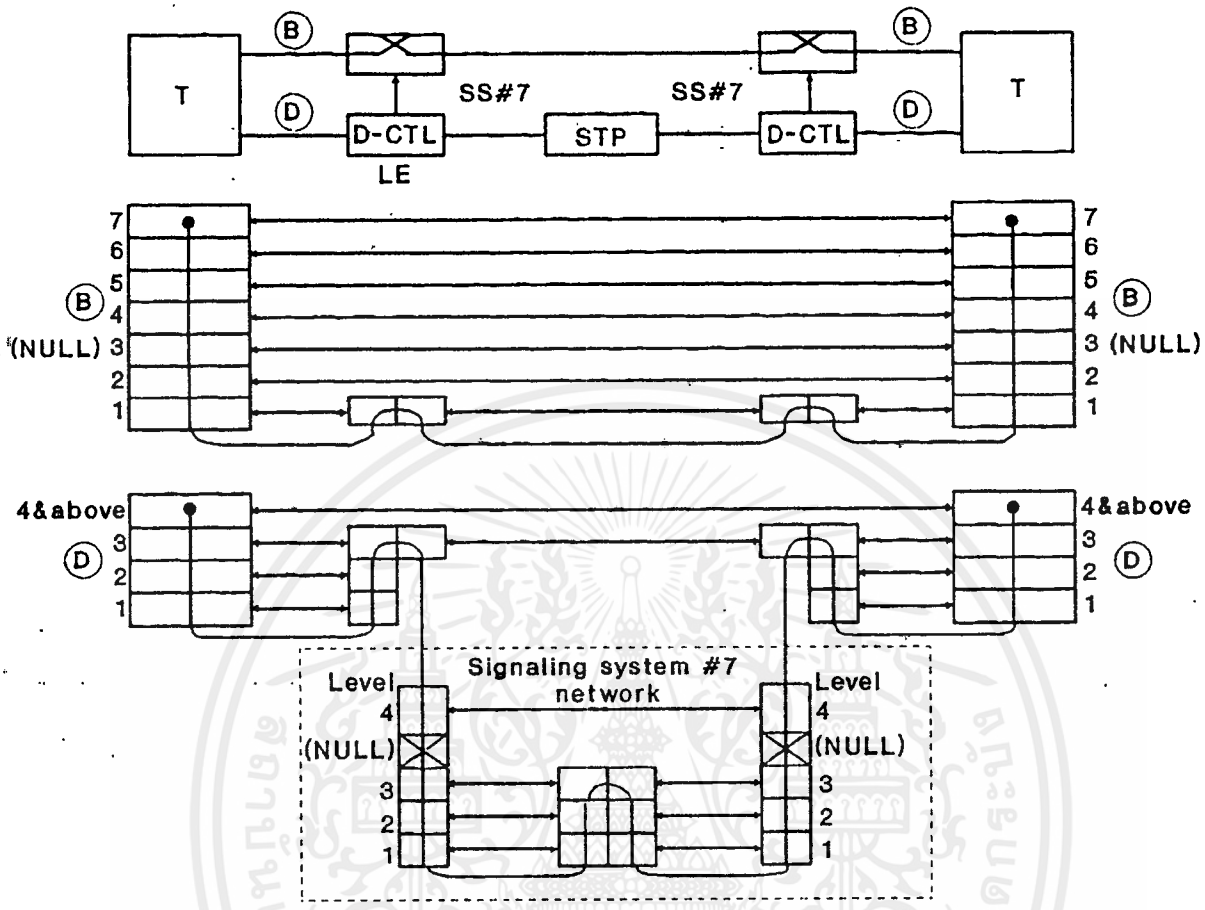
2.การเรียกในแบบแพคเก็ตสวิตช์ ผ่านช่องสัญญาณ B

3.การเรียกในแบบแพคเก็ตสวิตช์ ผ่านช่องสัญญาณ D

เซอร์กิตสวิตชิ่ง

ลักษณะของโครงข่ายและโปรโตคอลสำหรับเซอร์กิตสวิตชิ่งในการใช้บนช่องสัญญาณB และ D จะถูกใช้ในการสับเปลี่ยนข้อมูลของผู้ใช้ การติดต่อสื่อสารของผู้ใช้อาจจะมีการใช้โปรโตคอลหลาย ๆ โปรโตคอล ส่วนช่องสัญญาณ D จะถูกใช้ในการสับเปลี่ยนข่าวสารที่ใช้ในการควบคุมระหว่างผู้ใช้กับโครงข่าย เพื่อสร้างการเรียกและยกเลิกการเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

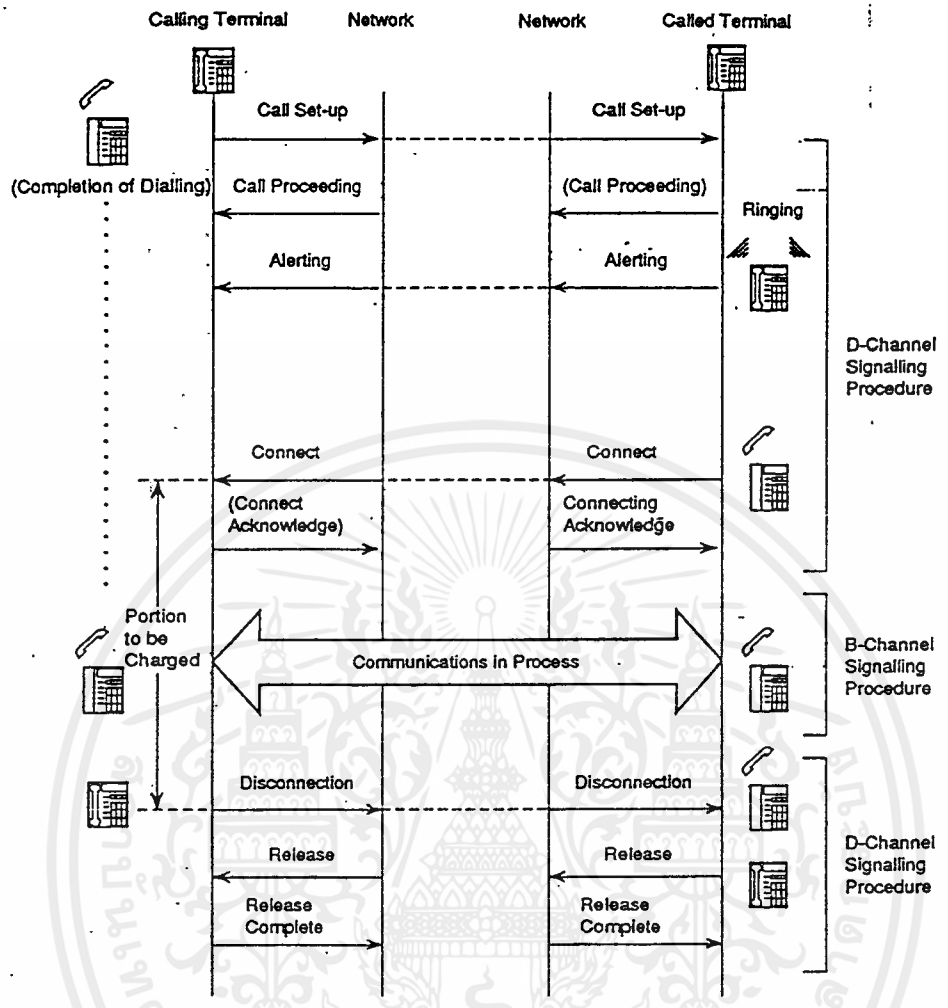


รูปที่ 2.5 แสดงลักษณะโปรโตคอลของเซอร์กิตสวิตชิ่ง

จากรูปที่ 2.5 เป็นการแสดงสถาปัตยกรรมของโปรโตคอลที่มีการใช้ในเซอร์กิตสวิตชิ่ง โดยจะพิจารณาร่วมกับค่าต่างๆ ดังตารางที่ 2.2 ซึ่งเป็นส่วนที่ใช้ในการบอกรายละเอียดที่สำคัญให้ กับรูป 2.5 - 2.8 ช่องสัญญาณ B จะถูกใช้บริการโดย NT1 หรือ NT2 โดยใช้ฟังก์ชันของระดับชั้นที่ 1 เพียงฟังก์ชันเดียว ทางด้านผู้ใช้ปลายทางจะมีการใช้โปรโตคอลทั้งหลายผ่านทางระดับชั้นที่ 3 ทั้งหมด ส่วนบนช่องสัญญาณ D จะถูกใช้สำหรับโปรโตคอลทั้ง 3 ระดับชั้น ส่วนกระบวนการในการสร้างวงจรผู้ใช้ผ่านโครงข่าย ISDN เพื่อทำการเซ็ตอัพการเชื่อมต่อไปยังผู้ใช้ การกระทำของสวิตช์จะใช้การควบคุมโดย SS7

การใช้บริการในแบบนี้จะมีลำดับขั้นตอนของการติดต่อระหว่างอุปกรณ์สื่อสารปลายทางดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการเชื่อมต่อการส่งข้อมูลแบบเซอร์กิตสวิตช์

แพคเก็ตสวิตช์

ในโครงข่าย ISDN จะอนุญาติให้ผู้ใช้สามารถเข้าสู่การบริการแบบแพคเก็ตสวิตช์ สำหรับทำการส่งข้อมูลภายใต้พื้นฐานของการให้บริการของแพคเก็ตสวิตช์ซึ่ง ซึ่งในรูปที่ 2.7 และ 2.8 เป็นการแสดงความเป็นไปได้ 2 แบบในการใช้งานแพคเก็ตสวิตช์

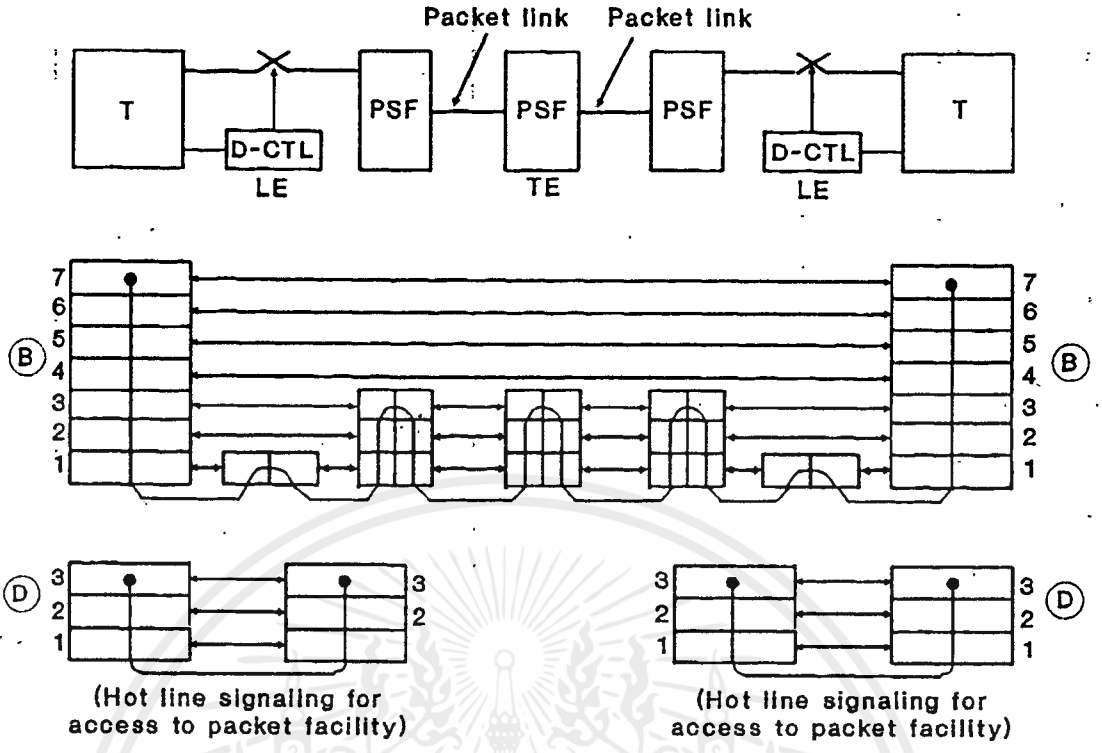
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงส่วนสำคัญที่ใช้ในรูป 2.5 - 2.8

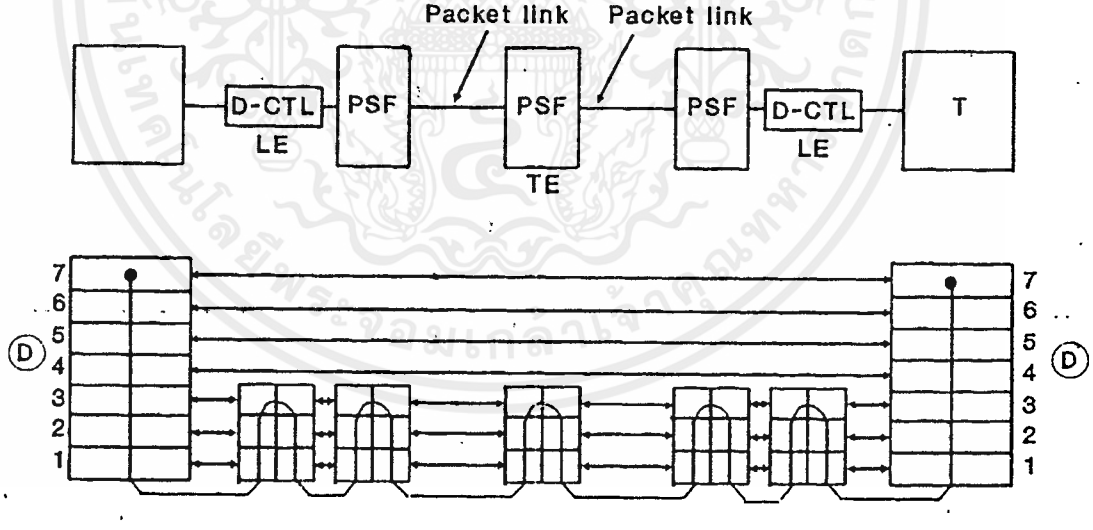
B	= An ISDN B channel
D	= An ISDN D channel
T	= Terminal
D-CTL	= D-channel controller
SS7	= CCITT signalling system 7
STP	= Signalling transfer point
(Null)	= Channel not present
7,6,5,4,3,2,1	= Layers in ISO basic reference model
LEVEL	= Levelin SS7
LE	= Local exchange
TE	= Transit exchange
PSF	= Packet-switching factility
Horizontal line	= Peer-to-peer protocol
Vertical line	= Layer-to-layer data flow

จากในรูปที่ 2.7 จะเป็นการส่งข้อมูลแบบแพคเก็ตสวิตช์ความเร็วสูง ที่ 64 kbps ผ่านทางช่องสัญญาณ B โดยที่มีการจัดการเชื่อมต่อระหว่างผู้ใช้ไปยังโหมดของแพคเก็ตสวิตช์ซิ่ง การอินเตอร์เฟสกับผู้ใช้สาธารณะจะใช้รูปแบบในระดับฟิสิกคอลเพียงอย่างเดียว ในการติดต่อระหว่างผู้ใช้กับโหมดแพคเก็ตสวิตช์ซิ่ง การส่งข้อมูลระหว่างผู้ใช้จะเป็นการติดต่อแบบ X.25 ในระดับชั้นที่ 2 และ ที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ลักษณะของโครงข่ายและโปรโตคอลที่ใช้ในแพคเก็ตสวิตช์ชิ่งบนช่องสัญญาณ B



Note: There is another possibility: that LE is transparent to layer 3.

รูปที่ 2.8 ลักษณะของโครงข่ายและโปรโตคอลที่ใช้ในแพคเก็ตสวิตช์ชิ่งบนช่องสัญญาณ D

การทำงานในระดับชั้นที่ 2 และ ที่ 3 จะแบ่งตัวแปรอยู่ 2 ชุด โดยชุดหนึ่งจะเป็นชุดที่ทำหน้าที่ในการเชื่อมโยงจากผู้ใช้ไปยังโหนดแพคเก็ตสวิตช์ชิ่ง และตัวแปรอีกชุดหนึ่งจะเป็นการให้บริการข้อมูลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพคเก็ตสวิตช์ชิ่ง การเชื่อมต่อระหว่างผู้ใช้ (ทางช่องสัญญาณ B) กับโหนดแพคเก็ตสวิตช์ชิ่ง จะเป็นการเชื่อมโยงการสื่อสารที่อาจใช้แบบเซอร์กิตสวิตช์หรือแบบกึ่งถาวร(semipermanent) ในกรณีแรกจะเป็นการร้องขอแบบ X.25 เพื่อทำการเซ็ตอัพวงจรเชื่อมต่อแบบ Virtual ไปยังผู้ใช้อื่น ๆ ในกรณีหลังจะเป็นส่วนที่เกี่ยวข้องกับการใช้ช่องสัญญาณ D ซึ่งการทำงานจะเป็นขั้นตอนดังนี้

- มีการร้องขอจากผู้ใช้ ทางช่องสัญญาณ D ในโปรโตคอลของการควบคุมการเรียก (I.451) โดยจะมีการเชื่อมต่อในแบบเซอร์กิตสวิตช์ บนช่องสัญญาณ B จากผู้ใช้ไปยังโหนดของแพคเก็ตสวิตช์ชิ่ง

- การเชื่อมต่อจะถูกเซ็ตอัพโดยใช้ SS7 ในโหนดของแพคเก็ตสวิตช์ชิ่ง และทางผู้ใช้ ISDN จะมีการใช้ช่องสัญญาณ D

- ผู้ใช้จะทำการเซ็ตอัพวงจรผู้ใช้แบบ Virtual ไปยังผู้ใช้อื่นผ่านทางกระบวนการสร้างการเรียกของ X.25 บนช่องสัญญาณ B

- ผู้ใช้ปลายทางของวงจรเชื่อมต่อจะใช้กระบวนการของ X.25 บนช่องสัญญาณ B

- หลังจากที่มีการเรียกในแบบ Virtual หนึ่งครั้งหรือมากกว่านั้น แล้วปรากฏว่าผู้ใช้ไม่สามารถสร้างวงจรในการเชื่อมต่อได้ สัญญาณที่มีการส่งบนช่องสัญญาณ D จะยกเลิกการเชื่อมต่อที่ไปยังโหนดแพคเก็ตสวิตช์

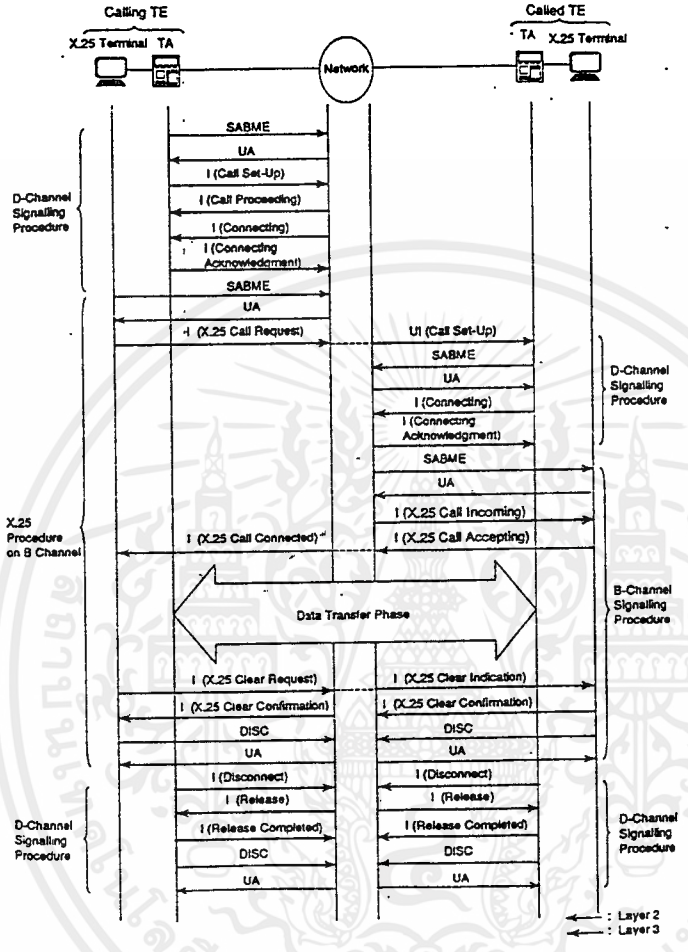
- ยกเลิกการเชื่อมต่อโดย SS7

การให้บริการของแพคเก็ตสวิตช์ชิ่งสามารถที่จะใช้บนช่องสัญญาณ D โดยการเข้าสู่ช่องสัญญาณ D นั้นโครงข่าย ISDN สามารถที่จะทำการจัดความสามารถของโครงข่ายให้สามารถใช้ได้กับแพคเก็ตสวิตช์ชิ่ง ช่องสัญญาณ D จะเป็นการจัดการเชื่อมต่อแบบกึ่งถาวรไปยังโหนดของแพคเก็ตสวิตช์ ชิ่งผู้ใช้จะใช้ X.25 โปรโตคอลระดับ 3 ซึ่งไม่อยู่ในกรณีของการเรียกแบบ Virtual บนช่องสัญญาณ B แต่ช่องสัญญาณ D จะถูกใช้สำหรับส่งสัญญาณในการควบคุม บางความหมายจะใช้ในการแบ่งแยกระหว่างทราฟฟิคของแพคเก็ต X.25 และทราฟฟิคในการควบคุมของ ISDN การใช้ช่องสัญญาณ D ในการส่งข้อมูลแบบแพคเก็ตสวิตช์ จะถูกนำไปแทรกในการใช้งานในช่องสัญญาณ D หลังจากที่มีใช้ในการควบคุมการเชื่อมต่อให้กับช่องสัญญาณ B แล้ว ช่องสัญญาณ D จะว่างไม่ถูกใช้ ดังนั้นเราจึงใช้ช่วงนี้ในการส่งแพคเก็ตได้

ตามที่ได้กล่าวมาแล้วข้างต้นในแบบแพคเก็ตสวิตช์นั้นจะมีได้ทั้งในช่องสัญญาณ B และช่องสัญญาณ

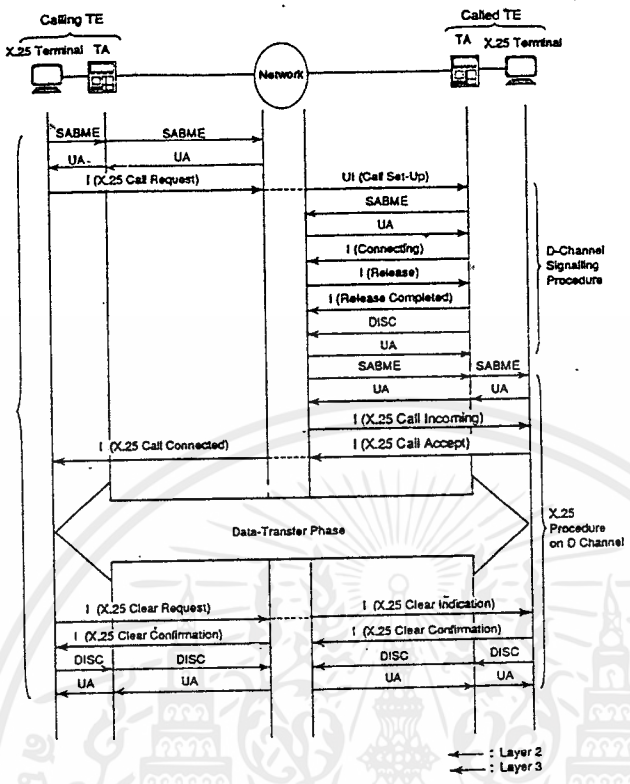
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D ดังนั้นลำดับขั้นตอนในการติดต่อระหว่างอุปกรณ์สื่อสารปลายทางจึงแบ่งเป็น 2 แบบ ดังแสดงในรูปที่ 2.9 และ รูปที่ 2.10 ตามลำดับ



รูปที่ 2.9 แสดง sequence ของการส่ง Packet ใน ช่องสัญญาณ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

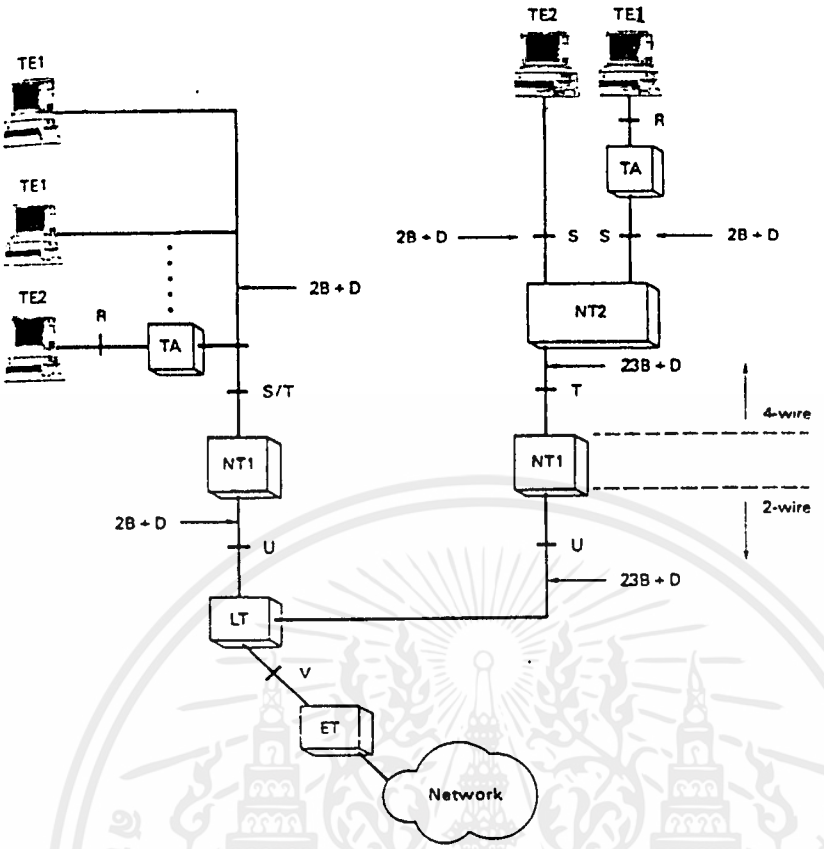


รูปที่ 2.10 แสดง sequence ของการส่ง packet ใน ช่องสัญญาณ D

2.6 การเชื่อมต่อระหว่างโครงข่าย ISDN กับผู้ใช้บริการ

การเชื่อมต่อระหว่างโครงข่าย ISDN กับผู้ใช้บริการประกอบด้วยอุปกรณ์ที่ทำหน้าที่ต่าง ๆ กันดังนั้นเพื่อเป็นการง่ายในการเข้าถึงแต่ละอุปกรณ์ จึงได้มีการกำหนดจุดเชื่อมต่อมาตรฐาน (Reference Point)ไว้ 5 จุดซึ่งก็คือ จุดเชื่อมต่อมาตรฐาน R,S,T,U และ V ดังแสดงในรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงอุปกรณ์และจุดเชื่อมต่อต่าง ๆ

จากรูปที่ 2.11 สามารถอธิบาย อุปกรณ์ และจุดเชื่อมต่อต่างๆได้ดังนี้

2.6.1 อุปกรณ์ในระบบ ISDN

1.Terminal Equipment (TE1) เป็นอุปกรณ์ปลายทางที่ถูกออกแบบ ให้ใช้ได้เฉพาะกับโครงข่าย ISDNโดยตรงโดยมีข้อกำหนดการอินเตอร์เฟสเป็นไปตาม CCITT "I series" ตัวอย่าง เช่นโทรศัพท์ระบบดิจิตอล โทรสารกลุ่มที่ 4 เป็นต้น

2.Terminal Equipment(TE2) เป็นอุปกรณ์ปลายทางที่ไม่ได้ออกแบบมาเพื่อใช้กับโครงข่าย ISDN โดยตรงซึ่งก็คืออุปกรณ์ที่มีใช้อยู่ในปัจจุบัน อุปกรณ์ดังกล่าวทำงานแตกต่างกันในแง่การส่งข้อมูล โปรโตคอลและรูปแบบการส่ง เช่น V.24 terminal , X.25 packet switched terminal และการที่จะนำอุปกรณ์การส่งมาต่อเข้ากับโครงข่าย ISDN ต้องผ่านอุปกรณ์ TA (Terminal Adapter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.Terminal Adapter (TA) เป็นอุปกรณ์ที่ใช้ในการอินเตอร์เฟสระหว่าง TE2 เข้ากับโครงข่าย ISDN หน้าที่ของTA มีดังนี้

- เปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลของ TE2 ที่จุดอ้างอิง R ให้ได้เอาท์พุตที่จุดอ้างอิง S เป็นสัญญาณดิจิตอลที่มีความเร็ว 64 Kbits/s ตัวอย่างเช่น ถ้า TE2 เป็นอุปกรณ์อนาลอกที่จุดอ้างอิง R จะเป็นสัญญาณอนาลอกจึงต้องใช้อุปกรณ์ TA ในการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอลที่มีความเร็ว64Kbits/sที่จุดอ้างอิง S เพื่อที่จะสามารถต่อเข้ากับโครงข่าย ISDN ได้

- เปลี่ยนแปลงโปรโตคอลที่ใช้กับอุปกรณ์ TE2 เช่น X.25 , V.24 ให้ไปเป็นโปรโตคอลที่ใช้กับช่องสัญญาณ "D" ที่เรียกว่า LAP-D

4.Network Termination 1(NT1) เป็นอุปกรณ์ที่เชื่อมต่อกับปลายสายของผู้ใช้(subscriber line) โดยมีฟังก์ชันการทำงานอยู่ในระดับชั้น 1 ของ OSI ซึ่งเป็นการเชื่อมต่อทางไฟฟ้า หน่วยงานที่ให้บริการโครงข่าย ISDN จะติดตั้งอุปกรณ์ NT นี้ให้กับผู้ใช้ตามบ้านผู้เช่า

5.Network Termination 2 (NT2) เป็นอุปกรณ์ที่มีฟังก์ชันการทำงานอยู่ในระดับชั้น 2 และ 3 ของ OSI ซึ่งมีหน้าที่ switching และรวบรวมข้อมูลของผู้ใช้ หรืออาจจะกล่าวได้อีกอย่างว่าทำหน้าที่ในการเปลี่ยนจาก Basic Rate Interface ไปเป็น Primary Rate Interface ตัวอย่างอุปกรณ์ NT2 ได้แก่ PABX

6.Line Termination(LT) เป็นอุปกรณ์ที่ทำหน้าที่ต่อสายผู้ใช้เข้ากับชุมสาย ISDN อุปกรณ์นี้จะติดตั้งอยู่ที่ชุมสาย ISDN

7.Exchange Terminal (ET) เป็นอุปกรณ์ที่ติดตั้งอยู่ในชุมสาย ISDN ทำหน้าที่เริ่มต้นการติดต่อชุมสาย ISDN ด้วยกัน

2.6.2 จุดอ้างอิงในระบบ ISDN

1.จุดอ้างอิง R เป็นจุดที่ต่อระหว่าง TE2 กับ TA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.จุดอ้างอิง S เป็นจุดที่ต่อระหว่าง TE1 กับ NT หรือ TA กับ NT

คุณสมบัติที่สำคัญของจุดอ้างอิง S คือ

- ต้องมีโปรโตคอลที่ใช้กับ Basic Rate Interface

- เป็นระบบสายส่งแบบ 4-wire รับส่งข่าวสารด้วยอัตราเร็ว 192 Kbits/s และ ข่าวสารอยู่ในรูปแบบรหัส AMI Code

การต่ออุปกรณ์ TE ที่จุดอ้างอิง S CCITTได้กำหนดชนิดของการเชื่อมต่ออุปกรณ์ TE ที่จุดอ้างอิง S ออกเป็น 2 ลักษณะคือ 1.แบบจุดต่อจุด (Point to Point) คือสายสัญญาณที่จุดอ้างอิง S ต่อเข้ากับ อุปกรณ์ TE หนึ่งเครื่อง 2.แบบจุดต่อหลายจุด (Point to Multipoint) คือ สายสัญญาณที่จุดอ้างอิง Sต่อเข้ากับอุปกรณ์ TE ได้หลายเครื่อง

3.จุดอ้างอิง T คือจุดต่อระหว่าง NT1 กับ NT2 ในกรณีที่ไม่ใช้ NT2 จุดอ้างอิง T ก็คือจุดอ้างอิง S แต่เมื่อมีการใช้อุปกรณ์ NT2 จุดอ้างอิง T ก็เป็นจุดอินเตอร์เฟสแบบ PRI (Primary Rate Interface) ซึ่งมีความเร็ว 2.048 Mbits/s

4.จุดอ้างอิง U คือจุดที่ต่อระหว่าง NT1 กับ LT ซึ่งมีคุณสมบัติที่สำคัญ คือ

- เป็นสัญญาณแบบ 2-wire ที่ใช้รหัส Differential Bi-Phase Code ในการรับส่งข่าวสาร

- มีการอินเตอร์เฟสแบบ 2B+D (BRI) หรือ 30B+D (PRI) ซึ่งก็แล้วแต่ความ จุของข้อมูลของข้อมูลข่าวสารของผู้ใช้

เนื่องจากว่าจุดอ้างอิง U ยังไม่มีมาตรฐานที่ทั่วโลกใช้เป็นอันหนึ่งอันเดียวกัน ดังนั้นหน่วยงานการสื่อสารของแต่ละประเทศก็จะใช้มาตรฐานจุดอ้างอิง U ที่แตกต่างกันออกไป

5.จุดอ้างอิง V เป็นจุดที่ต่อระหว่าง LT กับ ET จะมีอัตราเร็วในการส่งข้อมูล 2.048 Mbits/s

2.6.3 ระดับชั้นฟิสิกคอล (physical layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับชั้นฟิสิกคอลของ โครงข่าย ISDN จะเป็นการแสดงถึงการส่งสัญญาณของผู้ใช้ที่ จุดอ้างอิงต่างๆ คือจุด s และจุด t ซึ่งฟังก์ชั่นของระดับชั้นฟิสิกคอลจะประกอบด้วยฟังก์ชั่นต่างๆ(โดยอ้างอิงถึง OSI โมเดลในระดับชั้นที่ 1)

-ทำการเข้ารหัสข้อมูลดิจิตอลเพื่อทำการส่งข้อมูล ผ่านจุดที่เชื่อมต่อกับโครงข่าย

-การส่งข้อมูลในช่องสัญญาณ B จะเป็นแบบฟูลดูเพล็ก (Full-duplex)

-การส่งข้อมูลในช่องสัญญาณ D จะเป็นแบบฟูลดูเพล็ก (Full-duplex)

-มัลติเพล็กช่องสัญญาณ(ช่องสัญญาณ B และ D)ให้เข้ารูปแบบ โครงสร้างในการ อินเตอร์ เฟส แบบ BRI(2B+D) หรือ PRI (23B+D)

-ทำงานหรือหยุดทำงานในการเชื่อมต่อของวงจรทางฟิสิกคอล

-จ่ายพลังงานจาก NT ไปยัง TE

-มีการบ่งชี้ถึงเทอร์มินอลต่างๆ

-แบ่งแยกความผิดพลาดที่เกิดขึ้นกับเทอร์มินอล

-การเข้าสู่ส่วนต่างๆที่อยู่ในช่องสัญญาณ D

ในฟังก์ชั่นท้ายสุด จะมีการใช้งานงานแบบหลายจุด (multipoint) ในการอินเตอร์เฟสแบบ BRI ซึ่งโดยปกติแล้วการเชื่อมต่อทางฟิสิกคอล จะมีฟังก์ชั่นการใช้ที่แตกต่างกัน ระหว่างการอินเตอร์เฟส แบบ BRI และแบบ PRI ดังนั้นเราจึงมาพิจารณาถึงการอินเตอร์เฟสทั้งสองแบบ

<u>การอินเตอร์เฟสระหว่างผู้ใช้กับโครงข่ายแบบ BRI</u>

ข้อกำหนดที่จะใช้ในการเชื่อมต่อของระดับชั้นที่ 1 ในการทำงานแบบ BRI นี้จะอยู่ใน Rec I.430 การอินเตอร์เฟสแบบ BRI นี้จะมีโครงสร้างของสัญญาณเป็น 2B+D ที่ 192 kbps ซึ่งมีส่วนที่สำคัญ อยู่ 3 ประการในการเชื่อมต่อคือ

-การเข้ารหัสเพื่อใช้ในการส่ง

-การจัดข้อมูลให้เป็นรูปแบบเฟรมเพื่อทำการมัลติเพล็ก

-ลักษณะของมัลติดรอป(multidrop)

<u>การอินเตอร์เฟสระหว่างผู้ใช้กับโครงข่ายแบบ PRI</u>

การเชื่อมต่อแบบ PRI จะมีลักษณะคล้ายคลึงกับการเชื่อมต่อแบบ BRI จะมีความแตกต่างกันในเรื่องการนำช่องสัญญาณต่างๆ มาทำการมัลติเพล็ก และทำการส่งไปบนตัวกลางเพียงคู่สายเดียว จะมีการเชื่อมต่อในแบบ จุดหนึ่งไปยังอีกจุดหนึ่ง(point-to-point) และการเชื่อมต่อจะอยู่ในจุดอ้าง T

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเป็นส่วนของ PBX แบบดิจิตอล หรือ อุปกรณ์อื่นๆที่สามารถใช้งาน กับระบบโครงข่าย ISDN ได้ การอินเตอร์เฟสแบบ PRI จะมีการใช้อัตราของข้อมูล 2 อัตรา คือที่ 1.544 Mbps กับ 2.048 Mbps

2.6.4 โปรโตคอล LAP-D (ISDN DATA LINK LAYER)

การสื่อสารทั้งหมดในช่องสัญญาณ D จะเป็นการใช้โปรโตคอลในการเชื่อมโยงที่มีชื่อเรียกว่าLAP-D (Link Access Protocol D channel) จะเป็นโปรโตคอลในการกำหนดการเชื่อมโยงข้อมูลระหว่างผู้ใช้กับโครงข่าย (TE กับ NT) ที่จุดอ้างอิง S และ T บนช่องสัญญาณ D ข้อกำหนดทั่วไปของโปรโตคอล LAP-D จะอยู่ใน CCITT Rec.Q.920 (I.440) และกระบวนการทำงานจะอยู่ใน Rec. Q.921 (I.441)

2.6.4.1 การให้บริการในโปรโตคอล LAP-D

จุดประสงค์ของ LAP-D ก็คือ ใช้ในการส่งข้อมูลของผู้ใช้ในระดับชั้นที่ 3 ซึ่งเป็นส่วนของผู้ใช้ผ่านในโครงข่าย ISDN บนช่องสัญญาณ D การบริการของ LAP-D สามารถจะรองรับส่วนต่างๆได้ดังนี้

- เทอร์มินอลได้หลายเทอร์มินอลที่มีการติดตั้งทางด้านผู้ใช้บริการ
- สามารถใช้กับการใช้งานในระดับชั้นที่ 3 ได้หลายประเภท (เช่น X.25 ในระดับชั้น 3 เป็นต้น)

มาตราฐานของ LAP-D ได้มีการจัดไว้ 2 รูปแบบทางด้านการบริการ LAP-D ของผู้ใช้คือ การส่งข้อมูลแบบไม่ต้องการการตอบรับ (Unacknowledged information transfer service) และการส่งข้อมูลแบบต้องการการตอบรับ (Acknowledged information transfer service)

1)การส่งข้อมูลแบบไม่ต้องการการตอบรับ การให้บริการแบบนี้ จะไม่สามารถรับประกันได้ว่าข้อมูลที่จะส่งจากผู้ใช้ จะสามารถส่งผ่านไปถึงผู้ใช้บริการอีกคนหนึ่งได้อย่างแน่นอน รูปแบบของการใช้แบบนี้ จะไม่มีส่วนในการควบคุมการส่งข้อมูล และการตรวจสอบความผิดพลาดของข้อมูล การให้บริการนี้จะสามารถรองรับได้ทั้งแบบที่มีผู้ใช้เพียงคนเดียว (point-to-point) และแบบผู้ใช้หลายคน(point-to-multipoint)การให้บริการจะสามารถส่งผ่านข้อมูลได้อย่างรวดเร็ว และสะดวกในการจัดการต่างๆ เช่น เกี่ยวกับข้อความในการเตือนภัย หรือข้อความที่ต้องการส่งไปยังผู้ใช้บริการทุกๆคน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2)การส่งข้อมูลแบบต้องการการตอบรับ เป็นการให้บริการที่มีการทำงานคล้ายคลึงกับการให้บริการของ LAP-B และ HDLC การบริการนี้จะเป็นการเชื่อมต่อที่สร้างขึ้นระหว่างผู้ใช้ LAP-D ทั้งสอง ซึ่งจะมีการทำงานอยู่ 3 ช่วง คือ ช่วงของการสร้างการเชื่อมต่อ ช่วงของการถ่ายโอนข้อมูล และช่วงยกเลิกการเชื่อมต่อ ในช่วงแรกที่ทำการสร้างการเชื่อมต่อ ผู้ใช้บริการ 2 คน จะทำการแลกเปลี่ยนสัญญาณที่มีการโต้ตอบกัน ผู้ใช้คนหนึ่งจะทำการส่งสัญญาณร้องขอการเชื่อมต่อไปยังผู้ใช้บริการอีกคนหนึ่ง ถ้าหากผู้ใช้นี้ว่างอยู่พร้อมที่จะทำการเชื่อมต่อ สัญญาณร้องขอจะได้รับการตอบรับกลับไปยังผู้เรียก ซึ่งก็หมายความว่า การบริการ LAP-D ในผู้ใช้แต่ละด้านของการต่อเชื่อมจะแทรกอยู่ในเฟรมที่ส่ง เพื่อใช้ในการควบคุมการส่งข้อมูลและควบคุมความผิดพลาดของข้อมูล ส่วนในช่วงของการถ่านโอนข้อมูล LAP-D จะรับประกันว่าเฟรมทั้งหมดได้ทำการส่งออก เมื่อได้รับการตอบรับจากทางด้านผู้รับ ซึ่งจะมีการตอบรับหลังจากได้รับข้อมูลแล้ว ส่วนช่วงสุดท้ายเป็นช่วงของการยกเลิกการเชื่อมต่อ เมื่อผู้ใช้ส่งข้อมูลหมดเรียบร้อยแล้ว ผู้ใช้คนใดคนหนึ่งจะมีการร้องขอการยกเลิกการเชื่อมต่อ ซึ่งจะทำให้โครงข่ายรับทราบและยกเลิกการเชื่อมต่อระหว่างผู้ใช้ทั้งสอง

2.6.4.2 คุณสมบัติพื้นฐานของโปรโตคอล LAP-D

โปรโตคอล LAP-D เป็นรูปแบบของโปรโตคอลที่มีหลังจากโปรโตคอล LAP-B ที่ใช้ใน X.25 และ HDLC ทั้งข้อมูลของผู้ใช้บริการและข้อมูลควบคุม รวมทั้งพารามิเตอร์ที่มีการส่งเฟรมด้วย เนื่องจาก LAP-D จะมีรูปแบบของการให้บริการอยู่ 2 รูปแบบดังได้กล่าวมาแล้ว ดังนั้นรูปแบบของกระบวนการทำงานจะมี 2 รูปแบบดังต่อไปนี้คือ

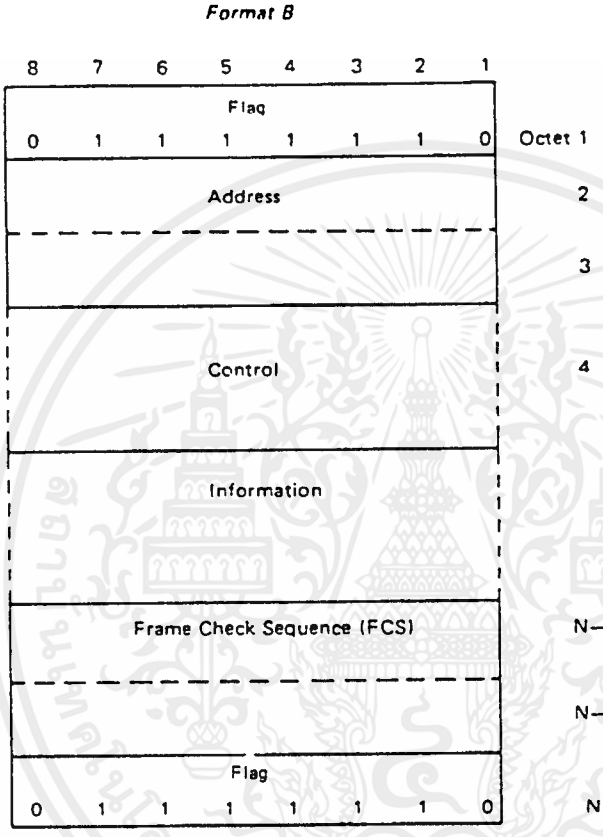
1)การทำงานแบบไม่มีการตอบรับ (Unacknowledged Operation) ข้อมูลของระดับชั้นที่ 3 ซึ่งเป็นข้อมูลของส่วนผู้ใช้ จะถูกส่งไปในรูปแบบเฟรมที่ไม่มีการตอบรับ ส่วนที่ใช้ในการตรวจสอบความผิดพลาด เมื่อตรวจพบความผิดพลาดเกิดขึ้น จะทำการตัดข้อมูลชุดนั้นทิ้ง แต่จะไม่ตัดในส่วนของข้อมูลที่ไม่มีความผิดพลาดเกิดขึ้น

2)การทำงานแบบมีการตอบรับ(Acknowledged Operation) ข้อมูลของระดับชั้นที่ 3 ซึ่งเป็นข้อมูลของผู้ใช้ ในรูปของเฟรม ซึ่งจะประกอบด้วยหมายเลขของลำดับในแต่ละเฟรม และเฟรมในการตอบรับ ส่วนการควบคุมความผิดพลาดและส่วนควบคุมการส่งข้อมูลจะมีโปรโตคอลรวมอยู่ด้วย ซึ่งรูปแบบของการบริการต่อไปนี้จะอ้างถึงรูปแบบมาตรฐานของการทำงาน

แบบมัลติเฟรม การทำงานทั้ง 2 แบบนี้จะกระทำบนช่องสัญญาณ D สำหรับการปฏิบัติงานแบบมีการตอบรับได้ สามารถใช้งานได้กับการเชื่อมต่อแบบของ LAP-D หลายๆส่วนในขณะเดียวกันด้วย ซึ่งจะคล้ายคลึงกับการทำงานของ X.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4.3 โครงสร้างของเฟรม LAP-D ข้อมูลทั้งหมดของผู้ใช้บริการและข้อความที่ใช้ในโปรโตคอลจะถูกส่งในรูปแบบของเฟรมดังในรูป 2.12 ซึ่งแสดงลักษณะโครงสร้างของเฟรมของ LAP-D ซึ่งจะมีส่วนประกอบต่างๆได้ดังนี้



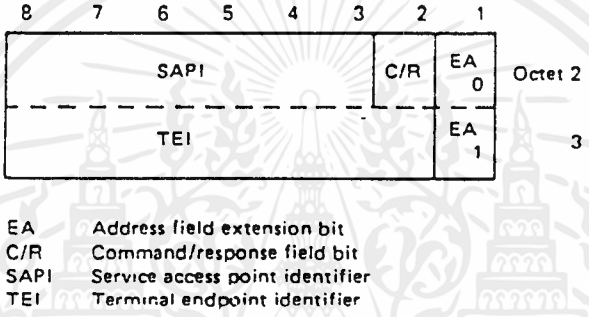
รูปที่ 2.12 รูปแสดงลักษณะของเฟรม LAP-D

ฟิลด์ของแฟล็ก (flag field) ฟิลด์นี้จะเป็นส่วนที่กำหนดขอบเขตของเฟรมที่จุดเริ่มต้นและท้ายสุดของเฟรม โดยมีการกำหนดรูปแบบที่แน่นอนคือ 01111110 แฟล็กหนึ่งอาจถูกใช้เป็นแฟล็กในการปิดท้ายเฟรมหนึ่ง และจะเป็นแฟล็กเริ่มต้นของเฟรมต่อไป ทางด้านผู้ใช้และโครงข่ายจะมีการรับข้อมูลและพิจารณาค่าของแฟล็กเพื่อใช้ในการซิงโครไนซ์

LAP-D จะมีการใช้บิท stuffing สำหรับป้องกันการส่งข้อมูลผิดพลาด โดยทางด้านส่งจะมีการตรวจสอบลำดับของบิทที่ส่ง ถ้ามีบิท 1 ติดต่อกัน 5 บิท ก็จะมีการกำหนดให้บิทต่อไปเป็น 0 ซึ่งก็ถือว่าเป็นชุดของแฟล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ฟิลด์ของแอดเดรส</u> (adress field) LAP-Dจะมีแนวทางในการมัลติเพล็กซ์ 2 ระดับ คือ ในระดับแรกจะทำการมัลติเพล็กที่ด้านของผู้ใช้ ซึ่งเป็นอุปกรณ์ของผู้ใช้หลายๆเครื่อง ที่มีการแบ่งภาระในการอินเตอร์เฟสทางฟิสิคอลที่เหมือนกันที่เชื่อมโยงไปยังโครงข่าย การมัลติเพล็กระดับที่สอง จะเป็นการใช้กับทราฟฟิกหลายๆชนิด เช่น ข้อมูลแพคเก็ตสวิทช์ และสัญญาณควบคุม เป็นต้น เมื่อนำเอาทั้งสองส่วนมาใช้งาน LAP-D เฟรมในฟิลด์ของแอดเดรสจึงแบ่งออกเป็นสองส่วน คือ TEI (Terminal Endpoint Identifier) และ SAPI (Service Access Point Identifier) ดังแสดงในรูป 2.13



<u>รูปที่ 2.13 แสดงรูปแบบของ Adress field</u>

ในทางปฏิบัติแล้ว TEI ที่อุปกรณ์เครื่องใช้แต่ละเครื่องจะไม่เหมือนกัน เพื่อเป็นการแยกแยะถึงอุปกรณ์ของผู้ใช้ที่ต่อร่วมอยู่ในคู่สายทางกายภาพเส้นเดียวกัน หมายเลข TEI สามารถที่จะกำหนดค่าได้โดยอัตโนมัติเมื่ออุปกรณ์ของผู้ใช้เริ่มทำการเชื่อมต่อในครั้งแรก หรืออาจจะกำหนดหมายเลข TEI โดยผู้ใช้ก็ได้ แต่ในกรณีหลังจะต้องระมัดระวังในเรื่องของอุปกรณ์ที่มีการเชื่อมต่ออยู่ในคู่สายเดียวกันไม่ให้หมายเลข TEI เหมือนกัน ในตารางที่ 2.3 ได้แสดงการกำหนดหมายเลข TEI ซึ่งจะเห็นว่าที่ค่า TEI เป็น 0-63 เป็นการกำหนดค่า TEI จากอุปกรณ์ของผู้ใช้ไว้แล้ว แต่ถ้าเป็นหมายเลข 64-126 จะเป็นหมายเลขที่ถูกกำหนดโดยอัตโนมัติจากโครงข่าย

ส่วนค่าของ SAPI นั้นจะเป็นตัวบ่งชี้ถึงการใช้บริการ LAP-D ของผู้ใช้ในระดับชั้น 3 ดังนั้น SAPI จะมีความสอดคล้องกับโปรโตคอลในระดับชั้น 3 ของอุปกรณ์ทางผู้ใช้ ซึ่งมีการกำหนดไว้ 4 ค่า ดังแสดงในตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEI Value	User type
0-63	Non-automatic TEI assignment user equipment
64-126	Automatic TEI assignment user equipment

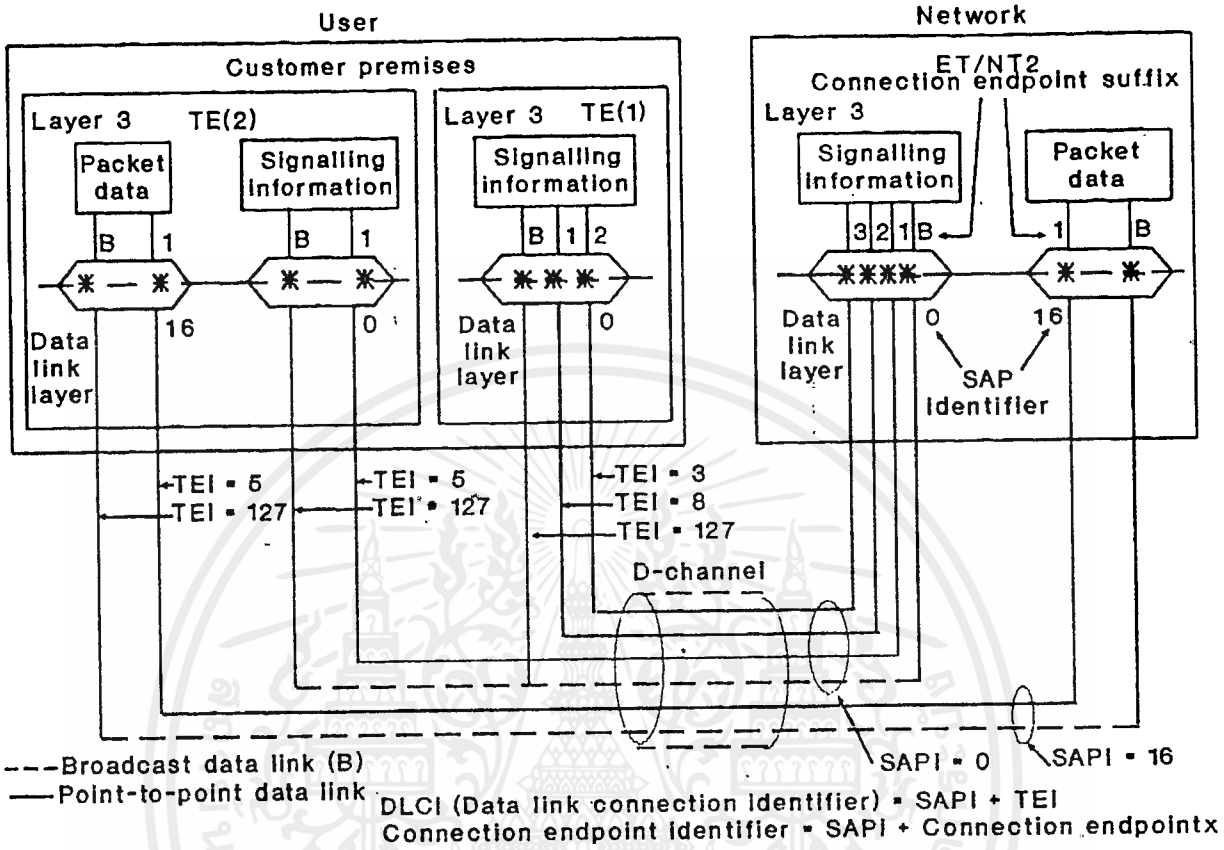
SAPI value	Related entity
0	Call control procedures
16	Packet communication procedures
32-47	Reserved for national use
63	Management procedures (see Note 1)
All others	Reserved for future standardization

ตารางที่ 2.3 การกำหนดค่า SAPI และ TEI

ค่าของ SAPI ที่เป็น 0 จะแสดงว่าเป็นกระบวนการในการควบคุมการเรียก (call control) ส่วนค่า SAPI ที่ 16 จะสำรองไว้ใช้ในการสื่อสารแบบแพคเก็ตบนช่องสัญญาณ D โดยใช้โปรโตคอล X.25 และค่า SAPI ที่ 63 เป็นการใช้สำหรับการส่งหรือรับข้อมูลในระดับชั้น 2 การกำหนดค่าอีกค่าหนึ่งก็คือ SAPI เท่ากับ 1 นั้น เป็นการใช้ในการสื่อสารแบบแพคเก็ต โดยใช้ตามข้อกำหนด I.451 ซึ่งเป็นสัญญาณระหว่างผู้ใช้กับผู้ใช้

ค่าของ SAPI จะมีความสัมพันธ์กับ TEI ในการเชื่อมต่ออุปกรณ์ผู้ใช้หลายๆอุปกรณ์ที่ต่ออยู่บนคู่สายเดียวกัน ในรูปที่ 2.14 เป็นตัวอย่างแสดงการใช้ค่า SAPI และ TEI ในการทำงานระหว่างอุปกรณ์ผู้ใช้แต่ละอุปกรณ์ ซึ่งจะเห็นว่าการสื่อสารทั้งหมดจะกระทำผ่านทางช่องสัญญาณ D เท่านั้น รูปแบบของฟิลด์แอดเดรสจะมีลักษณะดังในรูปที่ 2.13 ซึ่งจะมีส่วนของ SAPI จำนวน 6 บิทตั้งแต่บิทที่ 3-8 และหมายเลขของ TEI จะใช้ 7 บิทตั้งแต่บิทที่ 10-16 นอกจากนั้นในส่วนของฟิลด์แอดเดรสยังมีส่วนของ C/R (Command / Response) จะเป็นส่วนในการบอกถึงข้อความของ LAP-D ทั้งหมดนี้เป็นคำสั่งหรือเป็นการส่งการตอบรับกลับมา ซึ่งจะอยู่บิทที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงตัวอย่างของความสัมพันธ์ระหว่าง SAPI และ TEI

ฟิลด์ควบคุม (control field) ใน LAP-D จะแบ่งชนิดของเฟรมเป็น 3 ชนิด ดังแสดงในรูปที่ 2.15 ซึ่งในแต่ละชนิดจะมีรูปแบบของฟิลด์ควบคุมแตกต่างกัน ซึ่งสามารถจะแบ่งได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control field bits (modulo 8)	8	7	6	5	4	3	2	1	
I format	N(R)			P	N(S)			0	Octet 4
S format	N(R)			P/F	S	S	0	1	Octet 4
U format	M	M	M	P/F	M	M	1	1	Octet 4

Control field bits (modulo 128)	8	7	6	5	4	3	2	1	
I format	N(S)							0	Octet 4
	N(R)							P	5
S format	X	X	X	X	S	S	0	1	Octet 4
	N(R)							P/F	5
U format	M	M	M	P/F	M	M	1	1	Octet 4

N(S) Transmitter send sequence number
N(R) Transmitter receive sequence number
S Supervisory function bit
M Modifier function bit
P/F Poll bit when issued as a command, final bit when issued as a response
X Reserved and set to 0

รูปที่ 2.15แสดงรูปแบบของ control field

- เฟรมในการถ่ายโอนข้อมูล(Information Transfer frame - I frame) เป็นเฟรมที่ใช้ในการส่งข้อมูลของผู้ใช้ และจะมีส่วนของข้อมูลในการควบคุมการส่งข้อมูลและข้อมูลควบคุมความผิดพลาดของข้อมูล

- เฟรมในการตรวจตรา (Supervisory frame - S frame) เป็นการดูแลการส่งข้อมูลและควบคุมความผิดพลาดของข้อมูล

- เฟรม Unnumbered (U frame) เป็นส่วนในการควบคุมการเชื่อมโยง และใช้ในการสนับสนุนการทำงานในแบบไม่ต้องการการตอบรับ

ฟิลด์ข้อมูล (Information field) จะแสดงอยู่เฉพาะ I-frame และ U-frame เป็นฟิลด์ที่บรรจุด่วยบิทข้อมูลต่างๆหลายๆออกเตท ความยาวของฟิลด์ข้อมูลจะขึ้นอยู่กับจำนวนของข้อมูล แต่จะมีค่าไม่เกิน 260 ออกเตท ซึ่งข้อมูลต่างๆเหล่านี้จะเป็นข้อมูลในการควบคุมและข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลของแพคเก็ต

<u>ฟิลด์ตรวจสอบเฟรม</u> (FCS - Frame Check Sequency) จะเป็นฟิลด์สำหรับตรวจสอบความผิดพลาดต่างๆ โดยใช้วิธีการตรวจสอบรหัสแบบ CRC (Cyclic Redundancy Check)

<u>2.6.5 การอินเตอร์เฟสระหว่างผู้ใช้กับโครงข่ายในระดับชั้น 3</u> (Layer 3 User-network Interface)

สำหรับสัญญาณที่ใช้ควบคุมการเรียกในช่องสัญญาณ D บนการอินเตอร์เฟสระหว่างผู้ใช้กับโครงข่าย ในระดับชั้น 3 จะอยู่ในข้อกำหนด I.450 และ I.451 ซึ่งเป็นรายละเอียดของกระบวนการสำหรับการสร้างการเชื่อมต่อบนช่องสัญญาณ B โดยมีการแบ่งการใช้งานบนช่องสัญญาณ D เพื่อทำการควบคุมการเชื่อมต่อบนช่องสัญญาณ B นั้น

<u>2.6.5.1 ชนิดของเทอร์มินอล</u>

เทอร์มินอลต่างๆของผู้ใช้ในโครงข่าย ISDN โดยพื้นฐานสามารถแบ่งออกได้เป็น 2 ชนิด คือ

- เทอร์มินอลที่ทำตามหน้าที่ (Functional Terminal)
- เทอร์มินอลที่ต้องมีการกระตุ้น (Stimulus Terminal)

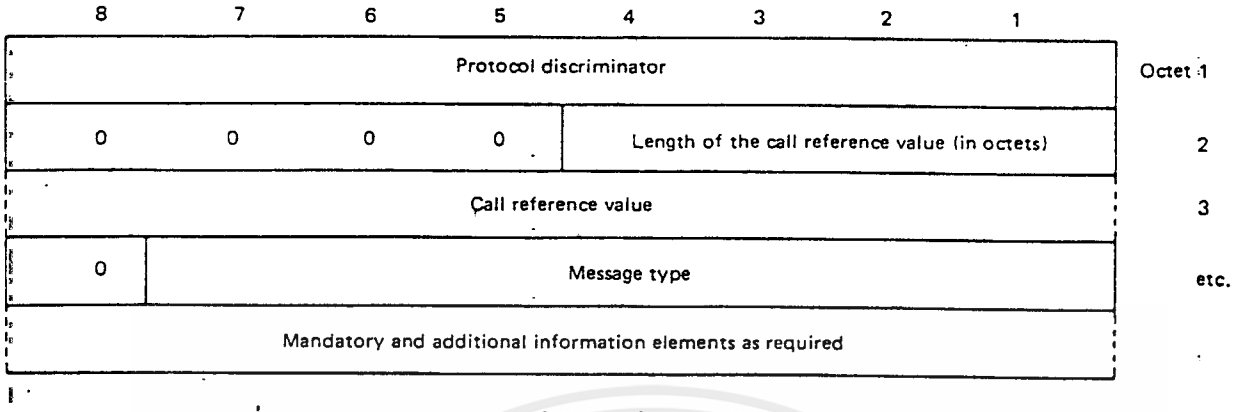
<u>1) เทอร์มินอลที่ทำตามหน้าที่</u> จะประกอบไปด้วยอุปกรณ์ที่สามารถคิดและสามารถทำงานได้ตามมาตรฐานข้อกำหนด I.451 ได้ทั้งหมด และพารามิเตอร์ที่ใช้สำหรับควบคุมการเรียกสัญญาณข้อมูลทั้งหมดที่มีการส่ง ข้อความในการควบคุมแต่ละข้อความจะมีหน้าที่แตกต่างกัน

<u>2) เทอร์มินอลที่ต้องมีการกระตุ้น</u> เป็นอุปกรณ์ที่ความสามารถพื้นฐานในการใช้สัญญาณควบคุมได้ ตัวอย่างได้แก่ โทรศัพท์แบบดิจิตอล เป็นต้น ข้อความจากโครงข่ายจะส่งไปยังเทอร์มินอลเหล่านี้ เพื่อทำการสร้างผลในการทำงานโดยตรงไปยังเทอร์มินอลผู้ใช้ (เช่น การยกหูฟัง การกดคีย์โทรศัพท์ เป็นต้น)

<u>2.6.5.2 ข้อความที่ใช้ในระดับชั้น 3</u>

ในกระบวนการสร้างการเชื่อมต่อ การควบคุม และการยกเลิกการเชื่อมต่อ จะเกิดขึ้นจากผลของข้อความในการควบคุม ที่มีการแลกเปลี่ยนระหว่างผู้ใช้กับโครงข่ายบนช่องสัญญาณ D รูปแบบของข้อความจะมีพื้นฐานในการใช้เหมือนกันทั้งหมด ซึ่งมีการกำหนดไว้ใน Rec.Q.931 (I.451) สำหรับโครงสร้างของข้อความใน Q.931 แสดงในรูป 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 รูปแสดงโครงสร้างของข้อความใน Q.931

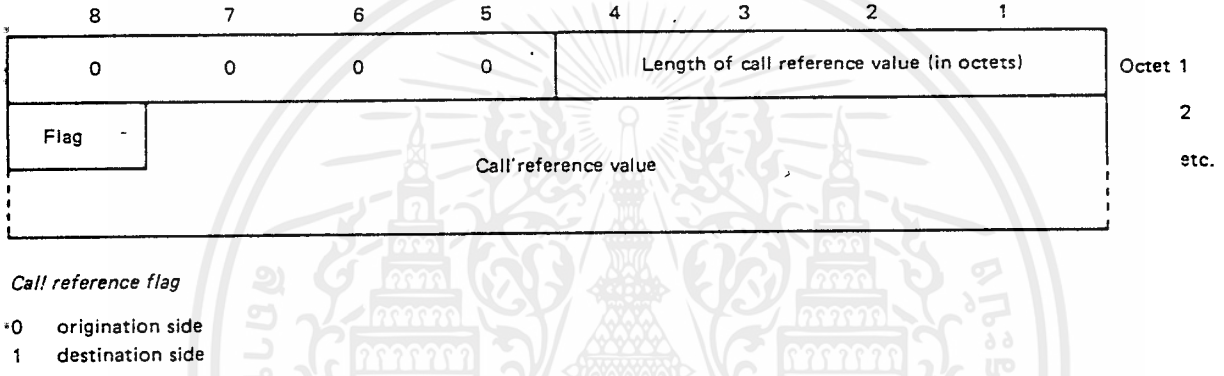
1) โปรโตคอลดิสคริมิเนเตอร์ (PD- Protocol Discriminator) จะใช้ในการแบ่งแยกข้อความควบคุม สำหรับการควบคุมการเรียกระหว่างผู้ใช้กับโครงข่าย ออกจากข้อความอื่นๆ ซึ่งจะปรากฏอยู่ส่วนแรกของทุกๆข้อความ โดยจะมีการเข้ารหัสและแสดงความหมายดังตารางที่ 2.4

8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	reserved.
0	0	0	0	0	0	0	1	reserved for other network layer or layer 3 protocols.
through								
0	0	0	0	0	1	1	1	
0	0	0	0	1	0	0	0	I.451 (Q.931) Recommendation user-network call control messages.
0	0	0	0	1	0	0	1	reserved for other messages within this Recommendation.
through								
0	0	0	0	1	1	1	1	
0	0	0	1	0	0	0	0	reserved for other network layer or layer 3 protocols, including Recommendation X.25.
through								
0	0	1	1	1	1	1	1	
0	1	0	0	0	0	0	0	national use.
through								
0	1	0	0	1	1	1	1	
0	1	0	1	0	0	0	0	reserved for other network layer or layer 3 protocols, including Recommendation X.25.
through								
1	1	1	1	1	1	1	0	
1	1	1	1	1	1	1	1	reserved for extension.

ตารางที่ 2.4 ตารางแสดงค่าของโปรโตคอลดิสคริมิเนเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ค่าอ้างอิงในการเรียก (Call Reference) เป็นค่าที่ใช้ในการแสดงเพื่อการเรียกบนช่องสัญญาณ B ที่มีการอ้างอิงถึงในข้อความ เช่น การใช้หมายเลขในเวอร์ช่วลเซอร์กิตของ X.25 ค่าอ้างอิงในการเรียกจะประกอบด้วยฟิลด์ย่อย 3 ฟิลด์ด้วยกัน ฟิลด์ย่อยส่วนแรกจะเป็นการบอกความยาวของฟิลด์ค่าอ้างอิงในการเรียก ฟิลด์ย่อยอีกส่วนจะเป็นส่วนรายละเอียดของค่าอ้างอิงในการเรียก ส่วนฟิลด์ย่อยสุดท้ายจะเป็นค่าแฟล็ก ซึ่งใช้แสดงจุดปลายของการเชื่อมต่อของผู้ใช้ โดยค่าแฟล็กเป็น 1 แสดงว่าเป็นผู้ถูกเรียก ส่วนค่าแฟล็กเป็น 0 แสดงว่าเป็นผู้เรียก



รูปที่ 2.17 แสดงส่วนของ call reference

3) ชนิดของข้อความ (Message type) จะเป็นการใช้ข้อความที่อ้างอิงใน Rec. Q.931 (I.451) เพื่อทำการส่งระหว่างผู้ใช้กับโครงข่าย ส่วนรายละเอียดของข้อความต่างๆ จะใช้ในการควบคุมการทำงานของผู้ใช้และทางโครงข่าย ซึ่งรายละเอียดของข้อความจะขึ้นอยู่กับชนิดของข้อความ ดังแสดงในตาราง 2.5

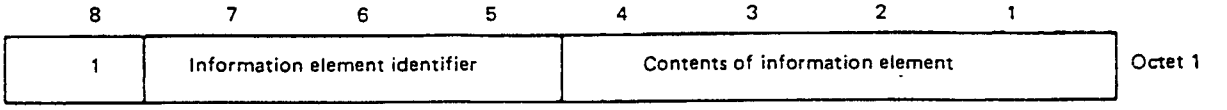
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8	7	6	5	4	3	2	1	
0	0	0	-	-	-	-	-	*Call establishment messages:*
			0	0	0	0	1	– ALERTing.
			0	0	0	1	0	– CALL PROCeeding.
			0	0	1	1	1	– CONNect.
			0	1	1	1	1	– CONNect ACKnowledge.
			0	0	1	0	1	– SETUP.
			0	1	1	0	1	– SETUP ACKnowledge.
0	0	1	-	-	-	-	-	*Call information phase messages:*
			0	0	1	1	0	– RESume.
			0	1	1	1	0	– RESume ACKnowledge.
			0	0	0	1	0	– RESume REJect.
			0	0	1	0	1	– SUSPend.
			0	1	1	0	1	– SUSPend ACKnowledge.
			0	0	0	0	1	– SUSpend REJect.
			0	0	0	0	0	– USER INFOrmation.
0	1	0	-	-	-	-	-	*Call disestablishment messages:*
			0	0	0	0	0	– DETach.
			0	1	0	0	0	– DETach ACKnowledge.
			0	0	1	0	1	– DISConnect.
			0	1	1	0	1	– RELease.
			1	1	0	1	0	– RELease COMplete.
0	1	1	-	-	-	-	-	*Miscellaneous messages:*
			0	0	0	0	0	– CANCel.
			0	1	0	0	0	– CANCel ACKnowledge.
			1	0	0	0	0	– CANCel REJect.
			1	1	0	0	1	– CONgestion CONtrol.
			0	0	0	1	0	– FACility.
			0	1	0	1	0	– FACility ACKnowledge.
			1	0	0	1	0	– FACility REJect.
			1	1	0	1	1	– INFOrmation.
			0	0	1	0	0	– REGister.
			0	1	1	0	0	– REGister ACKnowledge.
			1	0	1	0	0	– REGister REJect.
			1	1	1	0	1	– STATUS.

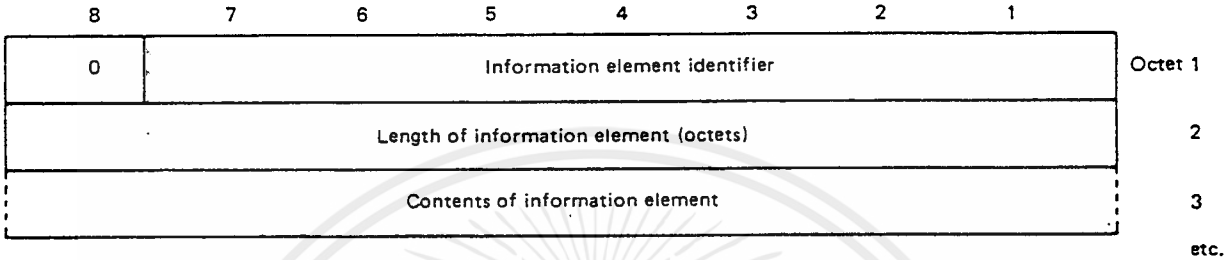
ตารางที่ 2.5 ตารางแสดงชนิดของข้อความ

4)ส่วนของข้อมูล (Information element) ภายในข้อความแต่ละข้อความจะมีรายละเอียดของข้อความอยู่ ซึ่งอาจจะมีเพียงออกเตทเดียวหรือหลายออกเตท โดยรูปแบบของส่วนของข้อมูลจะแสดงในรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



a) Single octet information element format



b) Variable length information element format

รูปที่ 2.18 รูปแสดงส่วนของข้อมูล

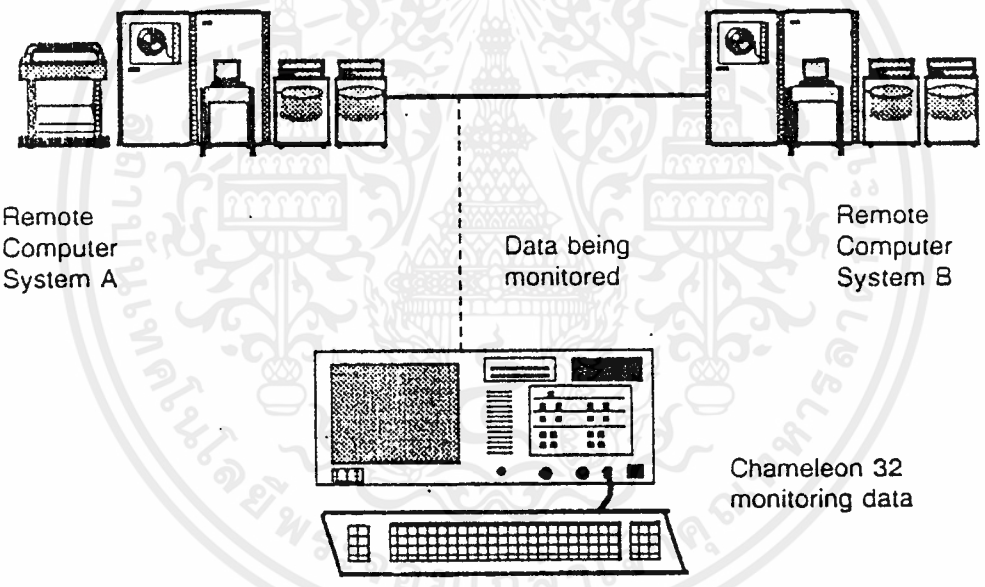
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์โปรโตคอล

เครื่องโปรโตคอลอนาไลเซอร์ สามารถใช้กับระบบ ISDN ที่ใช้การอินเตอร์เฟสแบบ Basic Rate Interface ,BRI ที่มีการส่งข้อมูล แบบ 2B+D หรือการอินเตอร์เฟสแบบ Primary Rate Interface,PRI ที่มีการส่งข้อมูล 23B+D หรือ 30B+D เครื่องโปรโตคอลอนาไลเซอร์สามารถช่วยให้เข้าใจลำดับการทำงานของการติดต่อสื่อสารได้ดียิ่งขึ้น และยังสามารถใช้ในการพัฒนาสร้างอุปกรณ์บริการปลายทางหรือโครงข่ายให้มีประสิทธิภาพสูงขึ้น โดยสร้างโปรแกรมที่ใช้ในการจำลองโครงข่าย ISDN โดยใช้ภาษา C หรือ Basic

3.1 การมอนนิเตอร์(monitor)



รูปที่ 3.1 การมอนนิเตอร์ข้อมูลที่มีการส่งระหว่าง ระบบ A กับ B

การใช้เครื่องโปรโตคอลอนาไลเซอร์ในการมอนนิเตอร์เป็นการนำเครื่องโปรโตคอลอนาไลเซอร์ไปเชื่อมต่อระหว่างการสื่อสารที่ต้องการมอนนิเตอร์ การเข้าไปแทรกการติดต่อสื่อสารจะไม่มีผลกระทบต่อการติดต่อสื่อสาร ภายในเครื่องโปรโตคอลอนาไลเซอร์มีหน่วยความจำบัฟเฟอร์ขนาด 1เมกกะไบท์ เพื่อรับข้อมูลที่เข้ามาเพื่อทำการวิเคราะห์ข้อมูลที่ได้

-สามารถวิเคราะห์โปรโตคอลชนิดต่างๆเช่น X.25, SNA, X.75, Q.921, Q.931, BSC, ASYNC, QLLC, PSH, DPNSS, V.120, DMI MODE2, DDCMP และ SS#7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-สามารถพิจารณาถึงสาเหตุที่เกิดขึ้นจากการทำงานหรือการซิมมูเลท โดยการพิจารณาจากการประยุกต์ใช้ Realtime หรือ History

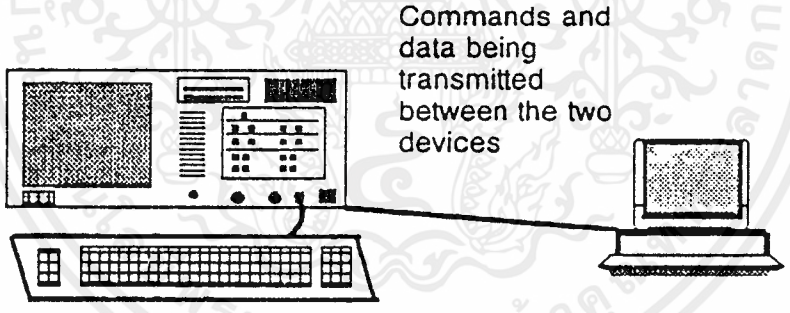
-สามารถแสดงผลและพิมพ์สถานะ ของ SNA, BSC, ISDN, SS#7 และการอินเตอร์เฟส PRI

-สามารถนำข้อมูลจากดิสก์ที่เก็บไว้มาทำการวิเคราะห์ได้ ซึ่งสามารถเก็บได้ขนาด 30 เมกกะไบท์(ฮาดดิสก์ขนาด 40 เมกกะไบท์ใช้เก็บโปรแกรมของระบบ 10 เมกกะไบท์)

-สามารถเลือกพิจารณาการเกิดทราฟฟิกโดยการใช้ทริกเกอร์ซึ่งสามารถใช้ได้กับโปรโตคอล HDLC, X.25, Q921, Q931, SS#7, DASS2

-สามารถแสดงเหตุการณ์(event)ที่เกิดขึ้นในทราฟฟิก ซึ่งจะแสดงเหตการณ์เป็นตัวอักษรตัวเดียว ต่อ หนึ่งเหตุการณ์

3.2 การซิมมูลเลท(simulation)



รูปที่ 3.2 การซิมมูเลทของโปรโตคอลอนาไลเซอร์

การซิมมูเลทจะใช้การจำลองการทำงานของอุปกรณ์ปลายทาง ISDN หรือโครงข่าย ISDN โดยใช้การพัฒนาจากการสร้างโปรแกรม เพื่อใช้การจำลองการทำงานที่เครื่องโปรโตคอลอนาไลเซอร์สามารถจำลองการทำงานของโปรโตคอลต่างๆได้ คือ โปรโตคอล X.25, SNA, ASYNC, BSC, ISDN, Q931, ISDN Q931, SS#7

การใช้เครื่องโปรโตคอลอนาไลเซอร์ในการจำลองการทำงานของโครงข่าย หรืออุปกรณ์บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทาง สามารถใช้ในการทดลองการทำงานของอุปกรณ์นั้นได้ เพื่อตรวจสอบการทำงานของอุปกรณ์ต่างๆ ซึ่งจะมีประโยชน์อย่างมากในการพัฒนาการทำงานของโครงข่าย หรืออุปกรณ์ปลายทางบางอย่าง เช่น

-คอมพิวเตอร์ เมนเฟรม

-โมเด็ม

-เทอร์มินอล หรือ ตัวควบคุมเทอร์มินอล

-โครงข่าย ISDN

-อุปกรณ์ที่ใช้ในการสื่อสารอื่น

การสร้างการชิมมูเลทสามารถสร้างได้โดยการพัฒนาซอฟต์แวร์เพื่อที่จะสร้างฟังก์ชันการทำงานต่างๆ โดยการเขียนลำดับการทำงานของอุปกรณ์นั้น และเมื่อเรานำเครื่องโปรโตคอลอนาไลเซอร์มาวิเคราะห์โปรโตคอลของระบบต่างๆ จะทำให้เราสามารถทราบรายละเอียดของโปรโตคอล รายละเอียดของปัญหา ซึ่งจะทำให้เราสามารถเข้าใจระบบนั้นได้มากขึ้น หรือเมื่อมีปัญหาเกิดขึ้นก็สามารถจะทำการวิเคราะห์เพื่อแก้ปัญหานั้นได้

3.3 การชิมมูเลทโดยเครื่องโปรโตคอลอนาไลเซอร์

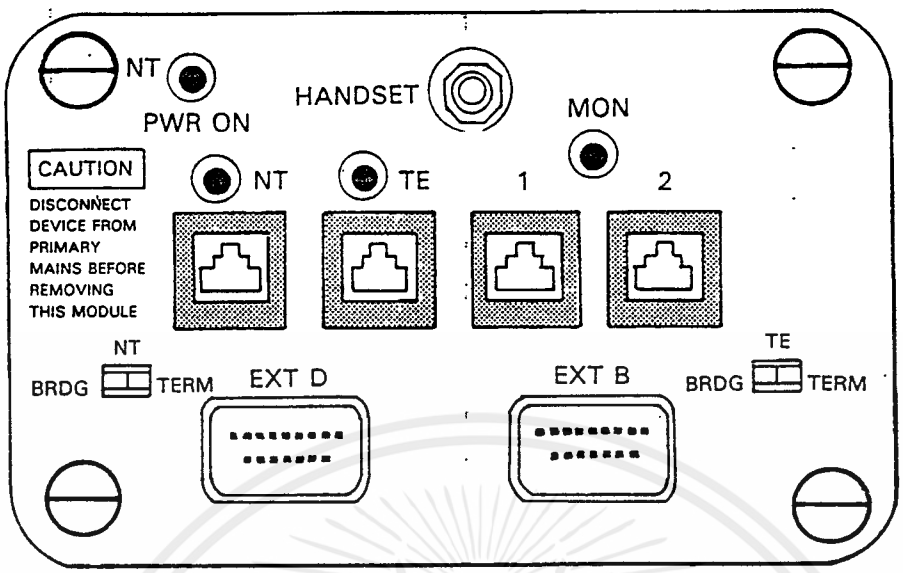
3.3.1 การชิมมูเลทโครงข่ายโดยใช้ INS-64

INS-64 เป็นชื่อของซอฟท์แวร์ที่ใช้ในการชิมมูเลทโครงข่าย INS (Information Network System) ซึ่งเป็นโครงข่าย ISDN ที่ใช้ในประเทศญี่ปุ่น ซึ่งเป็นการชิมมูเลทที่ใช้ในการอินเตอร์เฟสแบบ BRI (2B+D) การกำหนดการใช้งานของเครื่องโปรโตคอลอนาไลเซอร์ได้ดังนี้

การต่อพอร์ตสำหรับใช้งาน

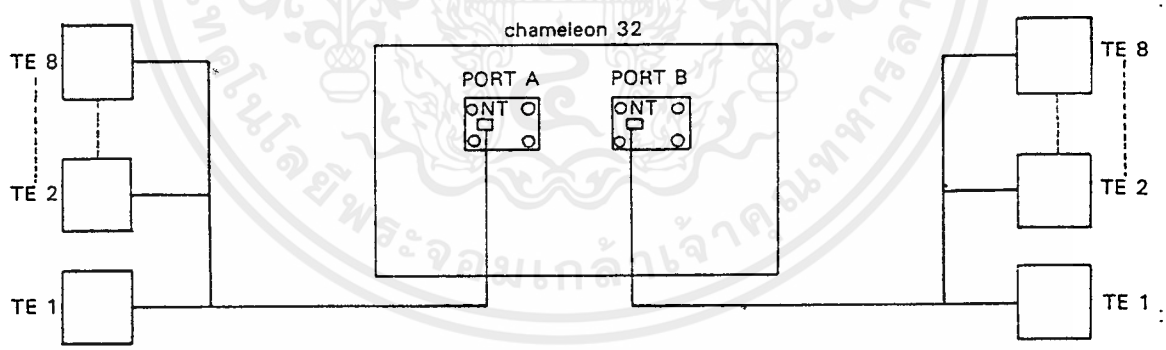
ที่ด้านหลังของเครื่องโปรโตคอลอนาไลเซอร์ จะมีพอร์ตสำหรับต่อใช้งานอยู่ 2 พอร์ตคือ พอร์ต A และ พอร์ต B ซึ่งเป็นชนิด BRI โดยทั้งพอร์ต A และพอร์ต B จะมีลักษณะเหมือนกันดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงจุดต่อใช้งานของพอร์ต A และ B

การใช้เครื่องโปรโตคอลอนาไลเซอร์เพื่อซิมมูเลทการทำงานของโครงข่าย INS จะต้องนำอุปกรณ์สื่อสารปลายทาง (Terminal Equipment) มาต่อที่ พอร์ต A และ พอร์ต B ของจุด NT เท่านั้น ดังแผนผังในรูปที่ 3.4



รูปที่ 3.4 แสดงการใช้งานของพอร์ตในการซิมมูเลทโดยโครงข่าย INS

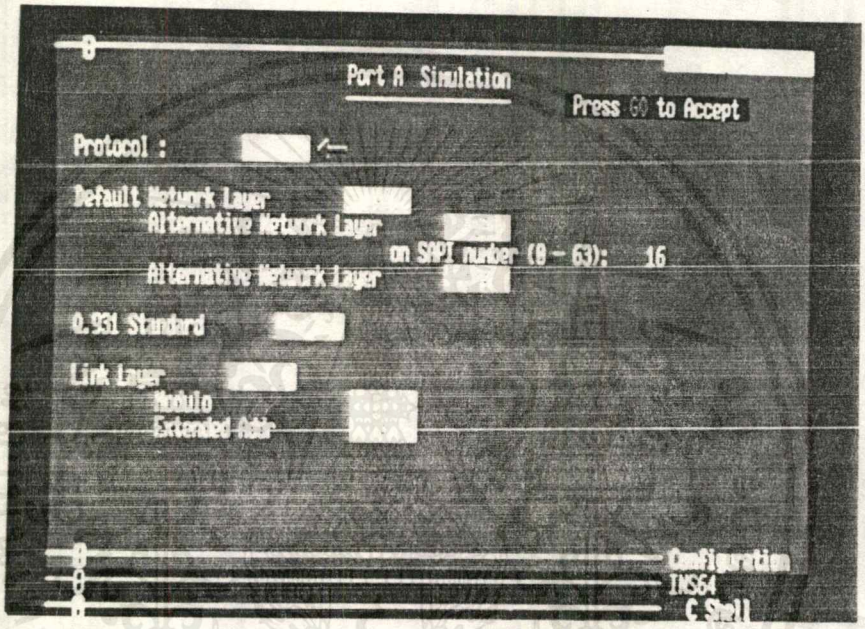
ขั้นตอนการซิมมูเลท

หลังจากเปิดสวิทช์เพาเวอร์ เครื่องโปรโตคอลอนาไลเซอร์จะทำการบูทระบบซอฟต์แวร์ และเมื่อทำเสร็จเรียบร้อยแล้ว จะปรากฏที่หน้าจอเป็นไฟล์ต่างๆของการเช็คเครื่องโปรโตคอลอนาไลเซอร์ ถ้ามีไฟล์ที่สร้างไว้สำหรับการซิมมูลเลทโครงข่าย INS แล้ว ก็สามารถโหลดไฟล์นั้นได้เลย แต่ถ้าไม่มีก็สามารถไปทำการกำหนดการซิมมูเลทได้จากเมนูของการกำหนดลักษณะการใช้งานของพอร์ต A และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต B ให้เป็นการซิมมูเลทการอินเตอร์เฟสทางฟิสิคัลเป็นแบบ Basic rate โปรโตคอลให้เป็นแบบ ISDN และให้ C Developement System เซ็ตเป็น ON

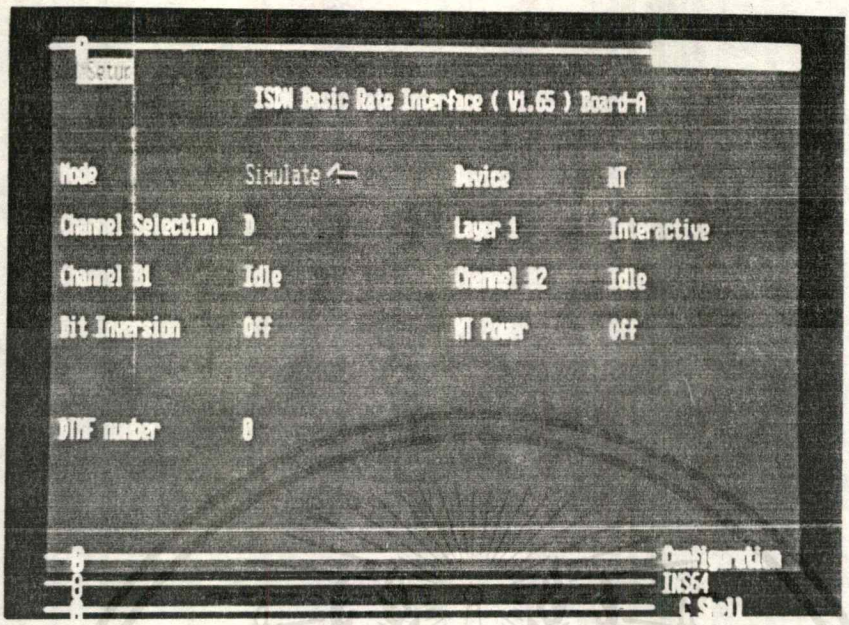
การกำหนดชนิดของโปรโตคอลให้กด F6 จะเข้าสู่จอภาพของการซิมมูเลท พอร์ต A กำหนดรายละเอียดต่างๆดังรูปที่ 3.5 จากนั้นกดคีย์ GO จะเข้าสู่จอภาพของการซิมมูเลท พอร์ต B ให้กำหนดรายละเอียดต่างๆเหมือนกับ พอร์ต A จากนั้นกดคีย์ GO อีกที จะเข้าสู่จอภาพของเมนูหลักของการเซ็ตการใช้งานอีกครั้ง



รูปที่ 3.5 แสดงการกำหนดรายละเอียดโปรโตคอลของพอร์ต A และ พอร์ต B

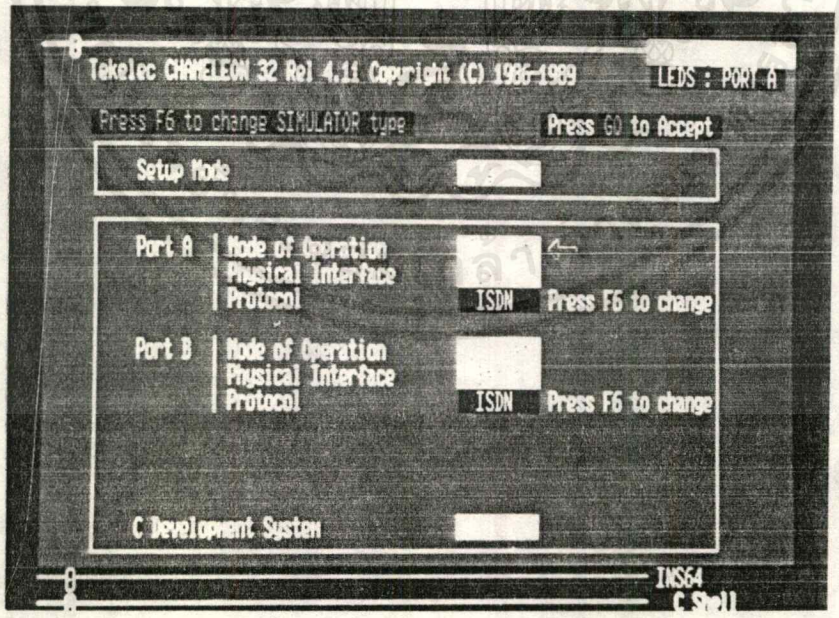
เมื่อกำหนดชนิดของโปรโตคอลเสร็จแล้ว จึงเข้าไปทำการกำหนดรายละเอียดในส่วนอินเตอร์เฟสทางฟิสิคัล โดยกดฟังก์ชันคีย์ F7 จะปรากฏผลที่หน้าจอดังในรูปที่ 3.6 ซึ่งเป็นการแสดงการเซ็ตของ ISDN Basic Rate Interface ของพอร์ต A และเมื่อกำหนดรายละเอียดต่างๆเสร็จเรียบร้อยจึงกดคีย์ GO จะปรากฏจอภาพของ ISDN Basic Rate Interface ของพอร์ต A ส่วนพอร์ต B จะกำหนดรายละเอียดต่างๆเหมือนกับพอร์ต A แล้วกด GO จะกลับเข้าสู่จอภาพของ Setup Mode Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการกำหนดรายละเอียดของ Basic Rate Interface ของพอร์ต A และ พอร์ต B

เมื่อกำหนดลักษณะการใช้งานครบทั้งหมดแล้ว จะเข้าสู่จอภาพ Setup Mode Menu ซึ่งจะมีลักษณะดังรูป 3.7 เมื่อทำการเซ็ตทุกอย่างพร้อมสำหรับใช้ในการซิมมูเลทโครงข่าย INS จึงกด GO ก็จะเข้าสู่ Menu ของการประยุกต์ใช้งาน

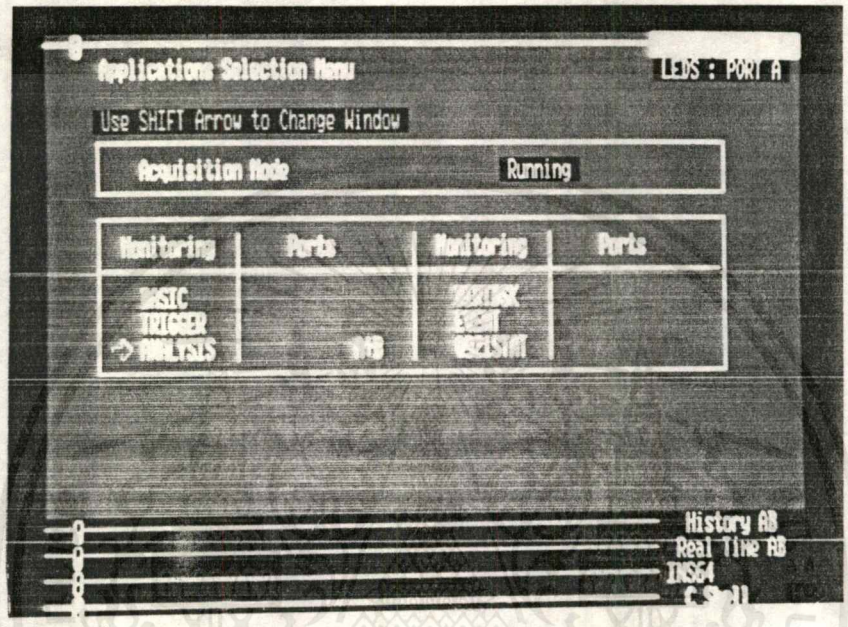


รูปที่ 3.7 แสดง Setup Mode Menu เมื่อกำหนดลักษณะการใช้งานแล้ว

เมื่อเข้าสู่เมนูของการประยุกต์ใช้งาน เลือกโหมดของการ Analysis เพื่อใช้ในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต A หรือ พอร์ต B หรือวิเคราะห์พอร์ต A และพอร์ต B พร้อมกัน โดยการกด F1, F2 หรือ F3 ตามลำดับจากนั้นกด GO จะปรากฏหน้าจอ History และ Realtime ตามพอร์ตที่โหลดไว้ ตัวอย่างเช่นถ้าต้องการจะแสดงการซิมมูเลทของทั้งสองพอร์ตก็กด F3 โหลด AB จากนั้นกด GO จะพบ banner ของ History AB และ Realtime AB ที่ด้านล่างของจอภาพดังรูปที่ 3.8



รูปที่ 3.8 รูปแสดงการเลือกการประยุกต์ใช้งาน

<u>การเข้าสู่โปรแกรมการซิมมูเลท INS 64</u>

การเข้าสู่โปรแกรมการซิมมูเลทโครงข่าย INS-64 จาก C-shell นั้นจะเข้าสู่ Directory ของ A:\USR หลังจากนั้นจึงเข้าสู่ Directory ของ A:\USR\INS64 แล้วเรียกไฟล์ INS64 หลังจากเรียกโปรแกรมการซิมมูเลทโครงข่าย INS64 จะมีหน้าจอดังแสดงในรูป 3.9 ซึ่งจะมีการเลือกจากฟังก์ชันคีย์ 4 ฟังก์ชัน

F1 จะเป็นการซิมมูเลทโครงข่าย INS-64

F2 เป็นการกำหนดลักษณะการใช้งานพอร์ต A

F3 เป็นการกำหนดลักษณะการใช้งานพอร์ต B

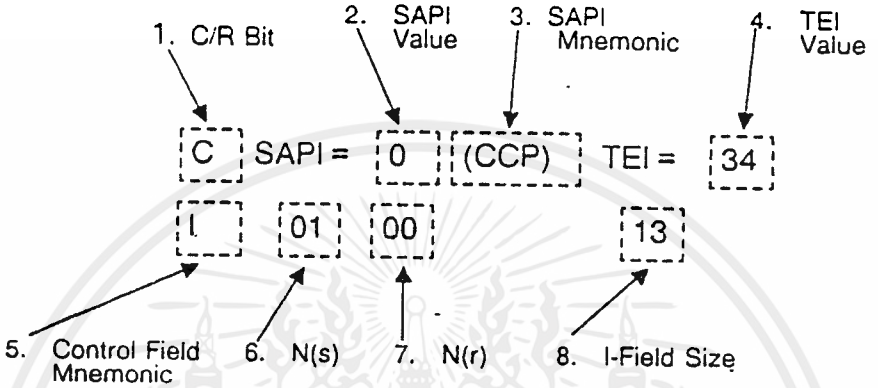
F4 เป็นการแสดงการส่ง-รับข้อมูลระหว่างพอร์ต Aและพอร์ต B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การวิเคราะห์ความหมายในโหมดของ History

การแสดงความหมายของข้อความใน Layer 2

ซึ่งมีส่วนประกอบของข้อความดังนี้



รูปที่ 3.11 การแสดงข้อความใน layer 2

1. C/R Bit

แสดงความหมายของบิตว่าเป็นบิตคำสั่งหรือบิตตอบรับ ได้แก่

C หมายถึง บิตคำสั่ง

R หมายถึง บิตตอบรับ

2.SAPI Value

เป็นค่าของ Service Access Point Identifier คือการแบ่งชนิดของข้อความในระดับชั้น 2 ที่จะส่งต่อไปยังระดับชั้น 3 หรือ ระดับชั้นที่สูงกว่าได้แก่

0 หมายถึง Call-Control Procedures ซึ่งถูกใช้ใน Circuit Switching และ B-Channel Packet Switching

16 หมายถึง Packet-communications procedures ซึ่งใช้ใน D-Channel packet Switching

63 หมายถึง การ Management Procedures ในระดับชั้น 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.SAPI Mnemonic

แสดงความหมายของ Service Access Point Identifier เป็นตัวอักษร ได้แก่

CCP หมายถึง Call Control Procedures

PCP หมายถึง Packet Communications Procedures

MP หมายถึง Management Procedures

4.TEI (Terminal Endpoint Identifier)

แสดงการแบ่ง Multiple terminal ที่มีลักษณะการติดต่อแบบ Basic Rate Interface เหมือนกันซึ่งเราสามารถที่จะกำหนดค่าของ TEI ได้สูงสุดถึง 128 ค่าโดยมีการกำหนดความหมายดังนี้

0-63 ใช้สำหรับเทอร์มินอลที่มีการเซ็ต TEI ปกติ ซึ่งกำหนดโดยเทอร์มินอล

64-126 ใช้สำหรับเทอร์มินอลที่มีการเซ็ต TEI อัตโนมัติ กำหนดโดยเนตเวิร์ค

127 ใช้สำหรับการต่อแบบดาต้าลิงค์แบบ broadcast

5.Control Field Mnemonic

เป็นตัวอักษรที่แสดงชนิดของเฟรม Control field ซึ่งได้แก่

I หมายถึง เฟรม Information Transfer เป็นบิตคำสั่งที่ใช้สำหรับการส่งข้อความต่อไปยังระดับชั้น 3

RR Recieve Ready เป็นบิตคำสั่งหรือบิตตอบรับก็ได้จะอยู่ในเฟรม Supervisory (S) ใช้สำหรับการตอบรับการ Recieving Readiness และ Acknowledgement ของ Information frame

RNR Recieve Not Ready แสดงเป็นบิตคำสั่งหรือบิตตอบรับก็ได้จะอยู่ในเฟรม Supervisory ใช้สำหรับแสดง Information frame ที่ไม่สามารถรับได้

REJ Reject แสดงเป็นบิตคำสั่งหรือบิตตอบรับจะอยู่ในเฟรม Supervisory ใช้แสดงการร้องขอให้ส่ง Information frame มาใหม่

SABME Set Asynchronous Balanced Mode Extended เป็นบิตคำสั่งที่อยู่ใน Unnumbered (U) frame ใช้สำหรับเซ็ตการติดต่อระหว่างผู้ใช้และเนตเวิร์คในการกำหนดการทำงานของ Multiframe acknowledged

DM Disconnected Mode เป็นบิตตอบรับอยู่ใน U เฟรม ใช้สำหรับบอกว่าการส่งข้อความในโหมดการทำงานของ Multiframe Acknowledged ไม่สามารถทำได้

UI Unnumbered Information เป็นบิตคำสั่งอยู่ใน U เฟรม ใช้สำหรับการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังระดับชั้น 3 และข้อความของ Management ในโหมด Unacknowledged

DISC Disconnected เป็นบิตคำสั่งอยู่ใน U เฟรมใช้สำหรับปิดการทำงานของโหมด multi-frame acknowledged

UA Unnumbered Acknowledgement เป็นบิตตอบรับอยู่ใน U เฟรมใช้สำหรับแจ้งว่าได้รับรองการ SABME และ DISC

FRMR Frame Reject เป็นบิตตอบรับอยู่ใน U เฟรมใช้สำหรับแจ้งสภาพของข้อผิดพลาดที่ไม่สามารถชดเชยได้ด้วยการส่งข้อความใหม่

6. N(s)

คือ Send Sequence number หมายถึงจำนวน Sequence ของเฟรมที่ใช้ส่งข้อความ โดยแสดงเป็นจำนวนเลขฐาน 16

7.N(r)

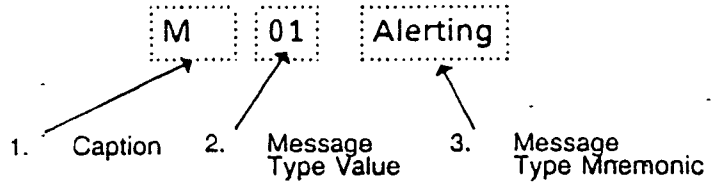
คือ Receive Sequence number หมายถึง จำนวน Sequence number ของ Information frame ตัวต่อไปที่จะได้รับ โดยแสดงเป็นจำนวนเลขฐาน 16

การแสดงความหมายของข้อความใน layer 3

เครื่องโปรโตคอลอนาไลเซอร์สามารถแสดงข้อความของ layer 3 ได้ โดยโครงสร้างของข้อความใน layer 3 จะมีส่วนของ protocal discriminator , call reference , message type และ information element ซึ่งได้กล่าวมาแล้ว โดยในส่วนของ protocal discriminator และ call reference แสดงให้เห็นอย่างชัดเจนอยู่แล้วในข้อมูลที่ได้วิเคราะห์ของเครื่องโปรโตคอลอนาไลเซอร์ ในที่นี้จึงจะขออธิบายในส่วนของการกำหนดชนิดข้อความ (message type) และในส่วนของ Information element

จากข้อมูลที่ใช้ในการกำหนดชนิดของข้อความจากเครื่องโปรโตคอลอนาไลเซอร์ จะมีโครงสร้างดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



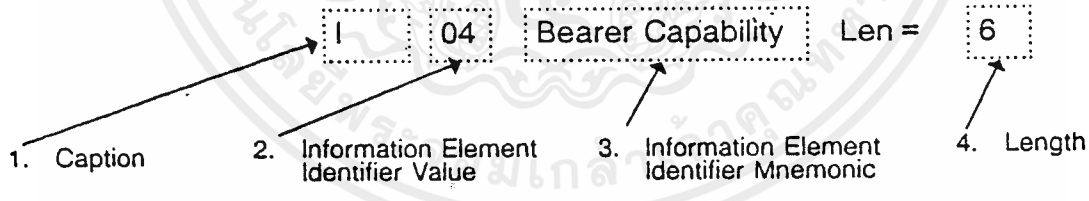
รูปที่ 3.12 แสดงส่วนของชนิดข้อความ

Caption จะเป็นการแสดงหัวของชนิดของข้อความ จะแสดงด้วย M ซึ่งก็คือชนิดของข้อความ

Message type จะเป็นส่วนแสดงด้วยค่าของตัวเลขฐานสิบหก โดยการบอกว่าเป็นข้อความใด

Message type mnemonic จะเป็นส่วนที่ใช้บอกถึงชนิดของข้อความ

รายละเอียดต่างๆของข้อมูลข่าวสาร(information) ภายในข้อความแต่ละข้อความจะมีรายละเอียดของข้อความแต่ละข้อความอยู่ ซึ่งมี octet เดียวหรือหลาย octet ลักษณะของการบอกรายละเอียดของข้อความจะแสดงดังในรูปที่ 3.13



รูปที่ 3.13 แสดงรายละเอียดต่างๆของข้อมูลข่าวสาร

caption จะเป็นการแสดงหัวข้อของข้อความใด โดยถ้าเป็น

I แสดงเป็นการเริ่มต้นของรายละเอียดของข้อมูล

I+ แสดงว่ารายละเอียดของข้อมูลที่ได้รับไม่น่าเชื่อถือ

information element identifier value แสดงด้วยค่าเลขฐานสิบหกซึ่งสามารถที่จะเทียบได้จากตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8	7	6	5	4	3	2	1	
1	:	:	:	-	-	-	-	*Single octet information elements:*
	0	0	0	-	-	-	-	reserved
	0	0	1	-	-	-	-	shift
	0	1	0	0	0	0	0	more data
	0	1	1	-	-	-	-	congestion level
0	:	:	:	:	:	:	:	*Variable length information element:*
	0	0	0	0	1	0	0	bearer capability
	0	0	0	1	0	0	0	cause
	0	0	0	1	1	0	0	connected address
	0	0	1	0	0	0	0	call identity
	0	0	1	0	1	0	0	call state
	0	0	1	1	0	0	0	channel identification
	0	0	1	1	1	0	0	CCITT-standardized facilities
	0	1	0	0	0	0	0	network-specific facilities
	0	1	0	0	1	0	0	terminal capabilities
	0	1	0	1	0	0	0	display
	0	1	0	1	1	0	0	keypad
	0	1	1	0	0	0	0	keypad echo
	0	1	1	0	1	0	0	signal
	0	1	1	0	1	1	0	switchhook
	1	1	0	1	1	0	0	origination address
	1	1	1	0	0	0	0	destination address
	1	1	1	0	1	0	0	redirecting address
	1	1	1	1	0	0	0	transit network selection
	1	1	1	1	1	0	0	low layer compatibility
	1	1	1	1	1	0	1	high layer compatibility
	1	1	1	1	1	1	0	user-user information
	1	1	1	1	1	1	1	reserved (escape)

All other values are reserved.

ตารางที่ 3.1 แสดงค่าของ information element identifier

information element mnemonic เป็นส่วนที่แสดงให้ทราบถึงรายละเอียดของข้อมูลว่าเป็นข้อมูลอะไร

length เป็นการแสดงถึงความยาวของข้อมูลว่ามีกี่ octet

1. พารามิเตอร์ต่างๆในข้อมูลของ Bearer Capabability ในพารามิเตอร์ของส่วน Bear Capability จะใช้ในการแสดงถึงการจัดการทำงานต่างๆ เช่น มาตรฐานในการเข้ารหัส โหมดในการถ่ายโอนข้อมูล อัตราการส่งข้อมูล เป็นต้น ลักษณะของข้อมูลใน Bearer Capability จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียดต่างๆดังในรูปที่ 3.14

8	7	6	5	4	3	2	1	
0	0	0	0	0	1	0	0	Octet 1
			Bearer capability Information element identifier					
Length of the bearer service identification								2
1	Coding standard		Information transfer capability					3
0/1 Ext.	Transfer mode		Information transfer rate					4
0/1 Ext.	Structure			Configuration		Establishment		4a[a]
0/1 Ext.	Symmetry		Information transfer rate (destination → origination)					4b[b]
0/1 Ext.	Multiplier or layer identification		Bearer capability multiplier/protocol identification					5[c]

a) This octet may be omitted, unless octet 4b is present.
b) This octet may be omitted. If present, octet 4a shall also be present.
c) This octet may be omitted. This octet may be repeated; e.g., to identify several protocols at one or more layers.

รูปที่ 3.14 แสดงรายละเอียดต่างๆของ Bearer Capability

ใน octet ที่ 1 จะเป็นการชี้บอกถึงข้อมูลว่าเป็นข้อมูลของ Bearer Capability และใน Octetที่ 2 จะเป็นการบอกถึงความยาวของข้อมูลของ Bearer Capacity ว่ามีทั้งหมดกี่ Octet ใน บิทที่ 8 ของ Octet ตั้งแต่ Octet ที่ 4 ไปทั้งหมด จะเป็นบิทที่ใช้บอกถึงมีการขยายเพิ่มข้อมูลหรือไม่

<u>บิทข้อมูลเพิ่มเติม(extention bit)</u> ใน Octetที่ 4 เป็นต้นไป

0 จะมีข้อมูล Octet ต่อไปอีก

1 เป็นข้อมูล Octet สุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานการเข้ารหัส(coding standard) ใน octet ที่ 3 แสดงดังตารางที่ 3.2

7	6	รายละเอียด
0	0	ตามมาตรฐานของ CCITT
0	1	ไว้สำหรับมาตรฐายของประเทศอื่นๆ
1	0	มาตรฐานของประเทศนั้น
1	1	มาตรฐานตามรายระเอียดของโครงข่ายที่ใช้อยู่ และ โครงข่ายที่เชื่อมโยงอยู่

ตารางที่ 3.2 แสดงมาตรฐานการเข้ารหัส

ลักษณะของการถ่ายโอนข้อมูล(information transfer capabity) ใน Octet ที่ 3 แสดงดังตาราง ที่ 3.3

บิทที่ 5	4	3	2	1	รายละเอียด
0	1	0	0	0	ไม่กำหนดว่าจะเป็นเฉพาะข้อมูลดิจิตอล
0	1	0	0	1	กำหนดเฉพาะข้อมูลดิจิตอล
1	0	0	0	0	เป็นสัญญาณออดิโอ 3.1kHz
1	0	0	0	1	เป็นสัญญาณออดิโอ 7 kHz
1	0	0	1	0	เป็นสัญญาณออดิโอ 15 kHz
1	1	0	0	0	เป็นสัญญาณภาพ

ตารางที่ 3.3 แสดงลักษณะของการโอนถ่ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดในการถ่ายโอนข้อมูล(Transfer Mode) อยู่ใน Octet ที่ 4 แสดงได้ดังตารางที่ 3.4

บิทที่ 7 6	รายละเอียด
0 0	เป็นโหมดของ circuit-switched
1 0	เป็นโหมดของ packet-switched

ตารางที่ 3.4 โหมดในการโอนถ่ายข้อมูล

อัตราในการถ่ายโอนข้อมูล(Information Transfer Rate) อยู่ใน Octet ที่ 4 และ 4b แสดงได้ดังตารางที่ 3.5

บิตที่ 5 4 3 2 1	โหมดของ circuit-switch	โหมดของ Packet-Switched
0 0 0 0 0	ตามขนาดของสัญญาณ	(หมายเหตุ)
1 0 0 0 0	64kbit/s	-
1 0 0 1 1	384	-
1 0 1 0 1	1536	-
1 0 1 1 1	1920	-

ตารางที่ 3.5 แสดงอัตราในการโอนถ่ายข้อมูล

หมายเหตุ จะขึ้นอยู่กับอัตราการส่งข้อมูลของ Packet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของข้อมูล(structure) อยู่ใน Octet ที่ 4a ดังแสดงได้ดังตารางที่ 3.6

บิทที่ 7 6 5	รายละเอียด
0 0 0	จะเป็นไปตามหมายเหตุ
0 0 1	ข้อมูลที่ 8 kHz ตามมาตรฐาน I.130
1 0 0	รวมเป็นหน่วยข้อมูลหนึ่งที่ใช้บริการ
1 1 1	ไม่มีโครงสร้างที่แน่นอน

ตารางที่ 3.6 แสดงโครงสร้างของข้อมูล

หมายเหตุ โครงสร้างของส่วนนี้จะขึ้นอยู่กับโหมดของการถ่ายโอนข้อมูล และลักษณะของข้อมูลที่ใช้เพื่อที่จะกำหนดโครงสร้างได้ดังตารางที่ 3.7

โหมดในการถ่ายโอนข้อมูล	ลักษณะของข้อมูล	โครงสร้าง
circuit	เสียงพูด	8 kHz
circuit	สัญญาณที่ไม่เป็นดิจิตอล	8 kHz
circuit	สัญญาณที่เป็นดิจิตอล	8 kHz
circuit	ออดิโอ	8 kHz
circuit	สัญญาณภาพ	8 kHz
packet	สัญญาณที่ไม่เป็นดิจิตอล	จะรวมเป็นหน่วยข้อมูลหนึ่ง

ตารางที่ 3.7 แสดงลักษณะโหมดการโอนถ่ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ลักษณะในการสื่อสาร(Configulation)</u> อยู่ใน Octet ที่ 4a

บิทที่		รายละเอียด
4	3	
0	0	จะใช้แบบ point-to-point
1	0	จะใช้แบบ multipoint

ตารางที่ 3.8 ลักษณะในการสื่อสาร

หมายเหตุ ถ้าไม่มี Octet ที่ 4a จะเป็นการใช้แบบ point to point

<u>สถานที่ (Establishment)</u> อยู่ใน Octet ที่ 4a จะอยู่ที่บิท ที่ 2 และ 1 ซึ่งจะถูกกำหนดให้เป็น 0 ซึ่งหมายถึงการกำหนดตามความต้องการ

<u>การสมมาตร(Symmetry)</u> อยู่ใน Octet ที่ 4b

บิทที่		รายละเอียด
7	6	
0	0	สมมาตรทั้งสองทิศทาง
0	1	ไม่สมมาตรทั้งสองทิศทาง
1	0	เป็นทิศทางเดียว (จากต้นทางไปยังปลายทาง)
1	1	เป็นทิศทางเดียว (จากปลายทางไปยังต้นทาง)

ตารางที่ 3.9 แสดงลักษณะการสมมาตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงถึงระดับชั้นต่าง ๆ ในโปรโตคอล

บิทข้อมูล							ข้อกำหนดต่างๆ
7	6	5	4	3	2	1	
0	0	-	-	-	-	-	เป็นการกำหนดอัตราการส่งข้อมูลซิงโครนัสซึ่งจะมีตามที่บิท 5-1
		0	0	0	0	1	0.6 kbits/s, Rec.X.1 และ I.461
		0	0	0	1	0	1.2 kbits/s, Rec.X.1 และ I.461
		0	0	0	1	1	2.4 kbits/s, Rec.X.1 และ I.461
		0	0	1	0	0	3.6 kbits/s, Rec.V.6 และ I.463
		0	0	1	0	1	4.8 kbits/s, Rec.X.1 และ I.461
		0	0	1	1	0	7.2 kbits/s, Rec.V.6 และ I.463
		0	0	1	1	1	8 kbits/s, Rec.I.460
		0	1	0	0	0	9.6 kbits/s, Rec.X.1 และ I.461
		0	1	0	0	1	14.4 kbits/s, Rec.V.6 และ I.463
		0	1	0	1	0	16 kbits/s, Rec.I.460
		0	1	0	1	1	19.2 kbits/s, Rec.I.463
		0	1	1	0	0	32 kbits/s, Rec.I.460
		0	1	1	1	0	48 kbits/s, Rec.X.1 และ I.461
		0	1	1	1	1	56 kbits/s, Rec.I.463
0	1	-	-	-	-	-	เป็นการกำหนดข้อมูลของผู้ใช้ใน Layer 1
		0	0	0	0	0	ตาม Rec.I.412
		0	0	0	0	1	มีการเปลี่ยนแปลงอัตราการส่งข้อมูลซิงโครนัส
		0	0	0	1	0	ตาม Rec.G.711 ใช้การเข้ารหัสสัญญาณเสียงพูดด้วย μ-law
		0	0	0	1	1	ตาม Rec.G.711 ใช้การเข้ารหัสสัญญาณเสียงพูดด้วย A-law
		0	0	1	0	0	ตาม Rec.G.721 ใช้การเข้ารหัสสัญญาณเสียงพูดแบบ ADPCM 32 kbits/s และตามมาตรฐาน I.460

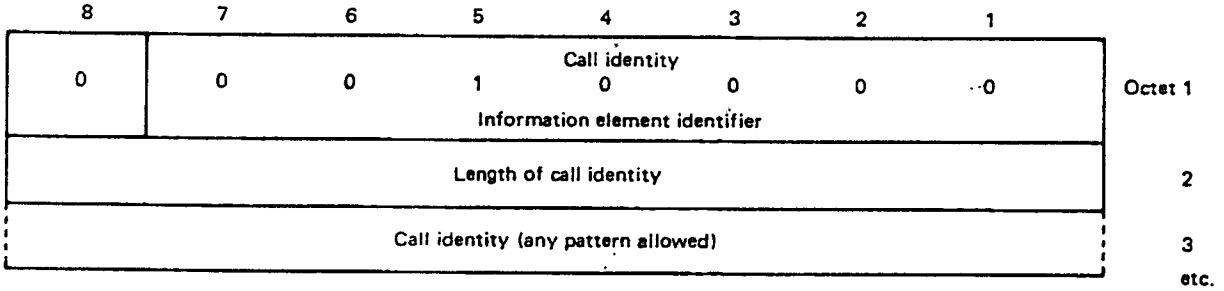
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทข้อมูล						ข้อกำหนดต่างๆ
7	6	5	4	3	2	1	
1	0	-	-	-	-	-	เป็นการกำหนดข้อมูลผู้ใช้ใน Layer 2
		0	0	0	0	0	ไม่กำหนด
		0	0	0	0	1	ตาม Rec.Q.921(I.441)
		0	0	1	0	0	ตาม Rec.Q.710
		0	0	1	1	0	ตาม Rec.X.25, link level
1	1	-	-	-	-	-	เป็นการกำหนดข้อมูลของผู้ใช้ใน Layer 3
		0	0	0	0	0	ไม่กำหนด
		0	0	0	1	0	ตาม Rec.Q.931(I.451)
		0	0	1	1	0	ตาม Rec.X.25, link level

ตารางที่ 3.10 แสดงถึงชั้นต่างๆในโปรโตคอล

พารามิเตอร์ต่าง ๆ ในข้อมูล Call Identity

พารามิเตอร์นี้จะเป็นส่วนที่ใช้ในการบอกถึงหมายเลขที่ใช้ในการเรียกและหมายเลขที่ถูกเรียก ลักษณะรูปแบบข้อมูล call identity จะมีลักษณะดังรูปที่ 3.15



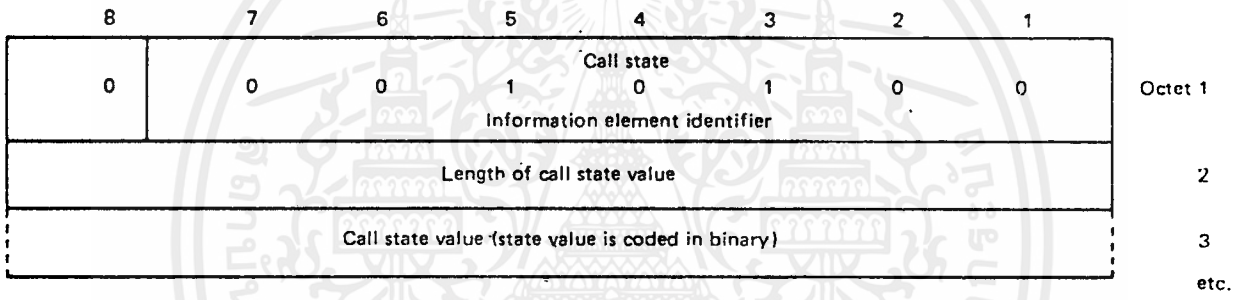
รูปที่ 3.15 แสดงการบอกหมายเลขในการเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน octet ที่ 1 จะเป็นข้อมูลที่ใช้ในการบอกถึง ข้อมูลต่อไปนี้จะเป็นพารามิเตอร์ที่ใช้ในการบอกหมายเลขของการเรียกใน octet ที่ 2 จะเป็นการบอกจำนวนของ octet ที่ใช้ octet ที่ 3 จะเป็นการบอกถึงหมายเลขของการเรียก ซึ่งจะใช้แทนรหัสแอสกี้ ซึ่งแต่ละ octet จะเป็นการบอกหมายเลขเพียงดิจิตเดียว

<u>การบอกถึงสถานะของการเรียก (call state)</u>

ในการอธิบายถึงรายละเอียดของข้อมูลเพื่อใช้ในการอธิบายถึงสภาวะของการเรียก ซึ่งจะมีลักษณะดังในรูปที่ 3.16



รูปที่ 3.16 แสดงรายละเอียดของสภาวะในการเรียก

ใน octet ที่ 1 จะเป็นข้อมูลที่ใช้ในการบอกถึง ข้อมูลต่อไปนี้จะเป็นพารามิเตอร์ของสภาวะในการเรียก

ใน octet ที่ 2 จะเป็นการบอกถึงจำนวนของ octet ที่ใช้

ใน octet ที่ 3 ขึ้นไปจะเป็นการบอกถึงสภาวะของการเรียก ซึ่งเราสามารถจะนำมาเขียนเป็นตารางได้ดังตารางที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

State value (octet 3)	User state	Network state
0	Null	Null
1	Call init	Dial tone sending
2	Overlap-sending	Overlap-sending
3	Outgoing call proceeding	Outgoing call proceeding
4	Call delivered	Call delivered
5	Negotiate	Negotiate
6	–	Call present
7	Call received	Call received
8	Connect request	Connect request
9	Incoming call proceeding	Incoming call proceeding
10	Active	Active
11	Disconnect request	Disconnect request
12	Disconnect indication	Disconnect indication
13	Detach request	Detach request
14	Detach	Detach
15	Suspend request	Suspend request
16	Local suspend	Local suspend
17	Resume request	Resume request
18	–	Tone active
19	Release request	Release request
20	Remote facility request	Remote facility request
21	Local facility request	Local facility request

ตารางที่ 3.11 แสดงการบอกถึงสภาวะการเรียก

<u>การบอกสาเหตุที่เกิดขึ้น (Cause)</u>

จุดประสงค์ของส่วนนี้ จะใช้ในการอธิบายถึงเหตุผลในการสร้างข้อมูลในข้อความต่าง ๆ หรือใช้เพื่อบอกสาเหตุของกระบวนการที่เกิดความผิดพลาดขึ้น และช่วยในการบอกขอบเขตของจุดที่เสียได้ รูปแบบของการบอกสาเหตุนี้จะเป็นลักษณะดังในรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8	7	6	5	4	3	2	1	
Cause 0	0	0	0	1	0	0	0	Octet 1
Information element identifier								
Length of cause information element								2
0/1 Ext.	Coding standard		0	0	Location			3
1 Ext.	Recommendation							3a*
0/1 Ext.	(Class)		Cause		Value (value in the class)			4
								4a*
Diagnostic(s) (if any)								5

* This octet may be omitted.

รูปที่ 3.17 ส่วนประกอบของส่วนที่ใช้ในการบอกถึงข้อมูลต่อไปและสาเหตุที่เกิดขึ้น

ใน octet ที่ 1 จะเป็นส่วนที่ใช้ในการบอกถึงข้อมูลต่อไป และสาเหตุที่เกิดขึ้นของความผิดพลาดหรือขอบเขตของจุดเสีย

ใน octet ที่ 2 จะเป็นส่วนที่ใช้ในการบอกถึงความยาวของข้อความนี้

ใน octet ที่ 3 จะเป็นส่วนที่ใช้บอกถึง มาตรฐานของการเข้ารหัส ดังแสดงในตารางที่ 3.12 และบอกถึงขอบเขตของจุดที่เสีย ดังแสดงในตารางที่ 3.13

7	6	รายละเอียด
0	0	ตามมาตรฐานของ CCITT
0	1	สำรองไว้สำหรับมาตรฐานของประเทศอื่นๆ
1	0	มาตรฐานของประเทศนั้น
1	1	มาตรฐานตามรายละเอียดของโครงข่ายที่ใช้อยู่ และ โครงข่ายที่เชื่อมโยงอยู่

ตารางที่ 3.12 แสดงมาตรฐานของการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทที่ 3 2 1	รายละเอียด
0 0 0	ในส่วนของผู้ใช้ (User)
0 0 1	ในส่วนของ (Local Private Network)
0 1 0	ในส่วนของโครงข่ายท้องถิ่น (Local Network)
0 1 1	ในส่วนของโครงข่ายส่งผ่าน (Transit Network)
1 0 0	ในส่วนของ RLN (Remote Local Network)
1 0 1	ในส่วนของ RPN (Remote Primitive Network)

ตารางที่ 3.13 แสดงถึงขอบเขตของจุดเสีย

ในส่วนของ octet ที่ 3 เป็นส่วนของ Recommendation ซึ่งจะแสดงดังตารางที่ 3.14

บิทที่ 7 6 5 4 3 2 1	รายละเอียด
0 0 0 0 0 0 0	Rec.Q.930/I.450
0 0 0 0 0 0 1	สำรองไว้สำหรับ Rec.I.211
0 0 0 0 0 1 0	สำรองไว้สำหรับ Rec.I.212
0 0 0 0 0 1 1	Rec.X.21
0 0 0 0 1 0 0	Rec.X.25

ตารางที่ 3.14 เป็นส่วนของ Recommendation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน octet ที่ 4 เป็นต้นไปจะเป็นส่วนของข้อมูลที่ใช้บอกถึงสาเหตุที่เกิดขึ้น ซึ่งสามารถเขียนได้ดังตารางที่ 3.15

Cause value		Cause number	Cause	Diagnostics
Class	Value			
765	4321	–		
101	0001	81	Invalid call reference value	Call reference value
101	0010	82	Identified channel does not exist	Channel identity
101	0011	83	Call identity does not exist	Call identity
101	0100	84	Call identity, in use	Call identity
101	0101	85	Invalid digit value for number	Address information element
101	0110	86	Non-existent closed user group	CUG member
101	0111	87	Destination address not member of CUG	Destination address, CUG member
101	1000	88	Incompatible destination	Destination address, incompatible parameter
101	1001	89	Non-existent abbreviated address entry	Copy of address element
101	1010	90	Destination address missing, and direct call not subscribed	–
101	1011	91	Transit network does not exist	Transit network identity
101	1100	92	Invalid facility parameter	Network identity facility
101	1101	93	Mandatory information element is missing	Information element identifier
110	0001	97	Message type non-existent or not implemented	Message type
110	0010	98	Message not compatible with call state	Message type
110	0011	99	Information element non-existent or not implemented	Information element
110	0100	100	Invalid information element contents	Information element

All other values are reserved.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cause value		Cause number	Cause	Diagnostics
Class	Value			
765	4321	–		
001	0001	17	User busy	Destination address
001	0010	18	No user responding	Destination address
001	0011	19	This call waiting at destination	Destination address
001	0100	20	Circuit operational	Circuit identity
001	0101	21	Call rejected	User supplied diagnostic
001	0110	22	Number changed	Destination address, new destination address
001	0111	23	Reverse charging rejected	Destination address
001	1000	24	Call suspended	Suspending address
001	1001	25	Call resumed	Resuming address
010	0001	33	Circuit out of order	Circuit identity
010	0010	34	No channel available	–
010	0011	35	Destination not obtainable	Destination address
010	0100	36	Out of order	Destination address
010	0101	37	Degraded service (e.g., excessive error rate)	–
010	0110	38	Network out of order	Transit network identity
010	0111	39	Transit delay range cannot be achieved	Minimum available transit delay
010	1000	40	Throughput range cannot be achieved	Maximum available throughput
010	1001	41	Network failure	Destination address
010	1010	42	Network congestion	Network identity
010	1011	43	User information discarded	Copy of first 32 octets of user-user information element
011	0001	49	Overlap sending not allowed	–
011	0010	50	Requested facility not subscribed	Network identity, facility
011	0011	51	Reverse charging not allowed	–
011	0100	52	Outgoing calls barred	–
011	0101	53	Outgoing calls barred within CUG	CUG identity
011	0110	54	Incoming calls barred	Destination address; optional user specified information
011	0111	55	Incoming calls barred within CUG	CUG identity, destination address
011	1000	56	Call waiting not subscribed	Destination address
100	0001	65	Bearer service not implemented	Service type
100	0010	66	Channel type not implemented	Channel type
100	0011	67	Transit network selection not implemented	–
100	0100	68	Message not implemented	Message type
100	0101	69	Requested facility not implemented	Network identity, facility
100	0110	70	Only restricted digital information bearer capability is available	–

ตารางที่ 3.15 เป็นส่วนของข้อมูลที่บอกถึงสาเหตุที่เกิดขึ้น

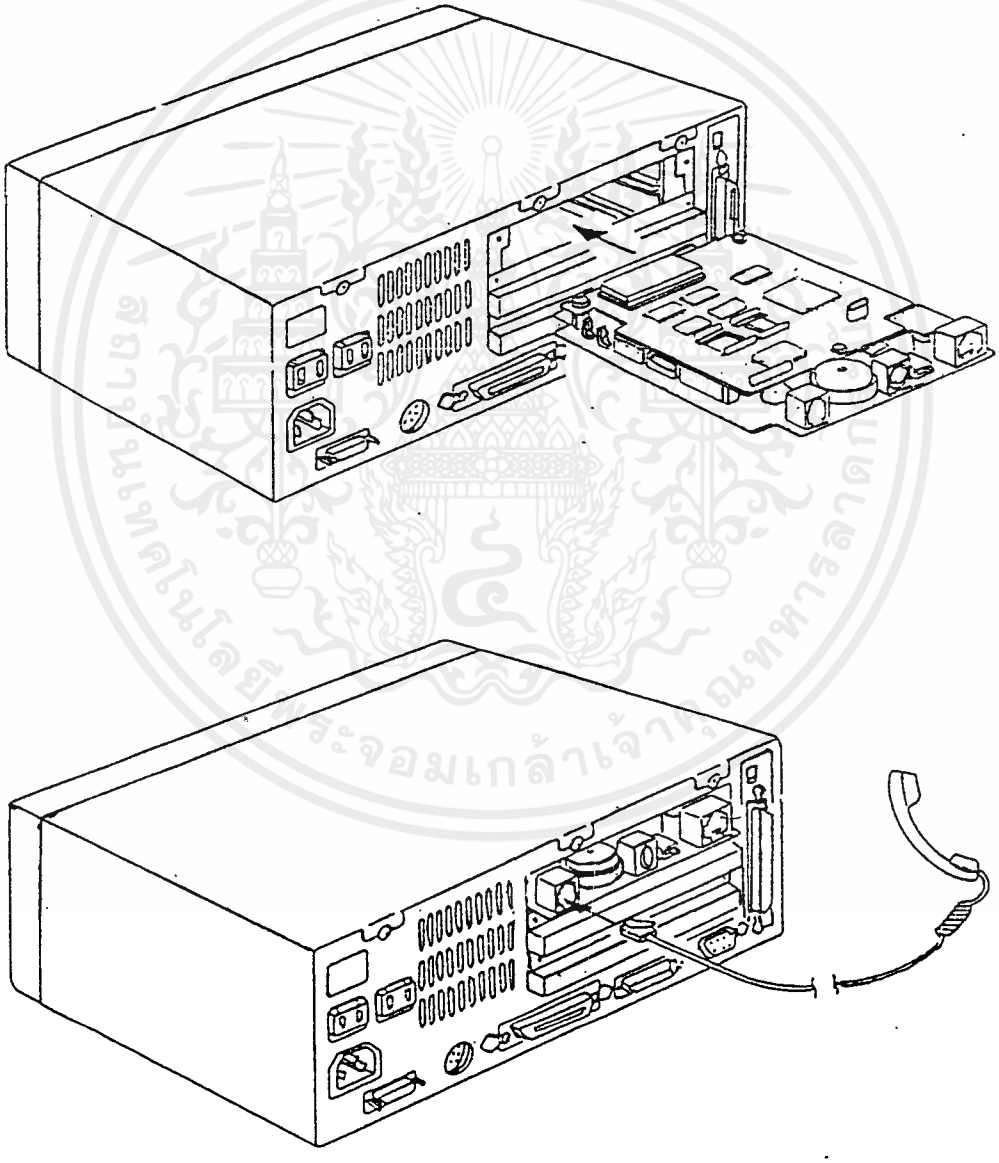
สำหรับการวิเคราะห์ข้อมูลทั้งหมดนี้เป็นเพียงบางส่วนเท่านั้น ซึ่งจะมีการเพิ่มเติมแก้ไขเปลี่ยนแปลงตามมาตรฐานของ CCITT ที่มีการปรับปรุงอยู่ใน Recommendation I Series

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

หลักการออกแบบซอฟต์แวร์

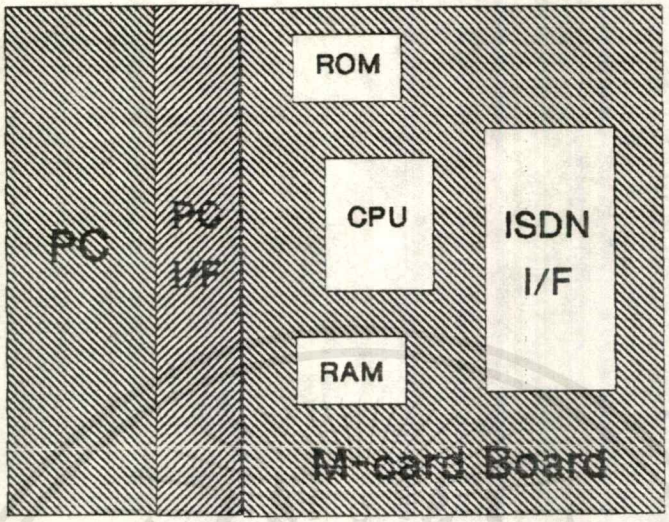
ในการเขียนซอฟต์แวร์ที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ ในโครงข่ายบริการร่วมระบบดิจิตอล (ISDN) จะมีส่วนที่เกี่ยวข้องก็คือ M-card64 ทำหน้าที่เป็น เทอร์มินอลอะแดปเตอร์ (Terminal Adapter,TA) ให้กับไมโครคอมพิวเตอร์ ลักษณะของ M-card 64 ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 แสดง M-card 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M-card 64 นี้ผลิตโดยบริษัท เอ็นทีที-ไอที (NTT-IT)ซึ่งมี บล็อกไดอะแกรม ดังแสดงต่อไปนี้



PC Personal Computer

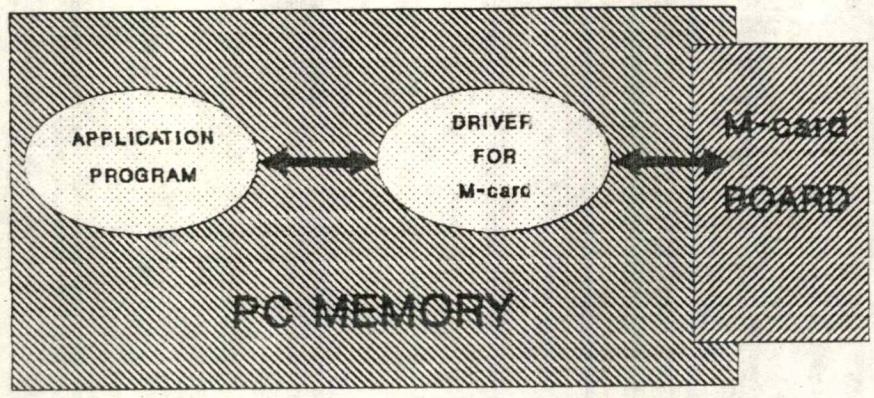
PC I/F PC Interface

ISDN I/F ISDN Interface

รูปที่ 4.2 แสดงบล็อคไดอะแกรมของ M-card 64

จากบล็อคไดอะแกรมในส่วนของ PC I/F จะมีทั้งส่วนที่เป็นฮาร์ดแวร์(hard ware) และซอฟต์แวร์ (software) ประกอบกัน โดยในส่วนของซอฟต์แวร์ จะมีลักษณะเป็น ไดรเวอร์(driver) เพื่อบอกให้ไมโครคอมพิวเตอร์ทราบว่า มี M-cardเชื่อมต่ออยู่ และการเรียกใช้งาน ไดรเวอร์นี้ เช่นเดียวกับการใช้งาน เมาท์ หรือ เครื่องพิมพ์ ดังนั้นก่อนที่จะใช้เครื่องคอมพิวเตอร์ในการติดต่อสื่อสารผ่านระบบ ISDN ต้องทำการโหลด ไดรเวอร์ของ M-card ก่อนทุกครั้ง ลักษณะซอฟต์แวร์ที่เขียนขึ้นเพื่อใช้ในการติดต่อสื่อสารจะเป็นลักษณะของแอปพลิเคชั่นโปรแกรม ซึ่งจะทำหน้าที่ควบคุม การทำงานของ M-card ผ่านทางไดรเวอร์ดังแสดงในรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

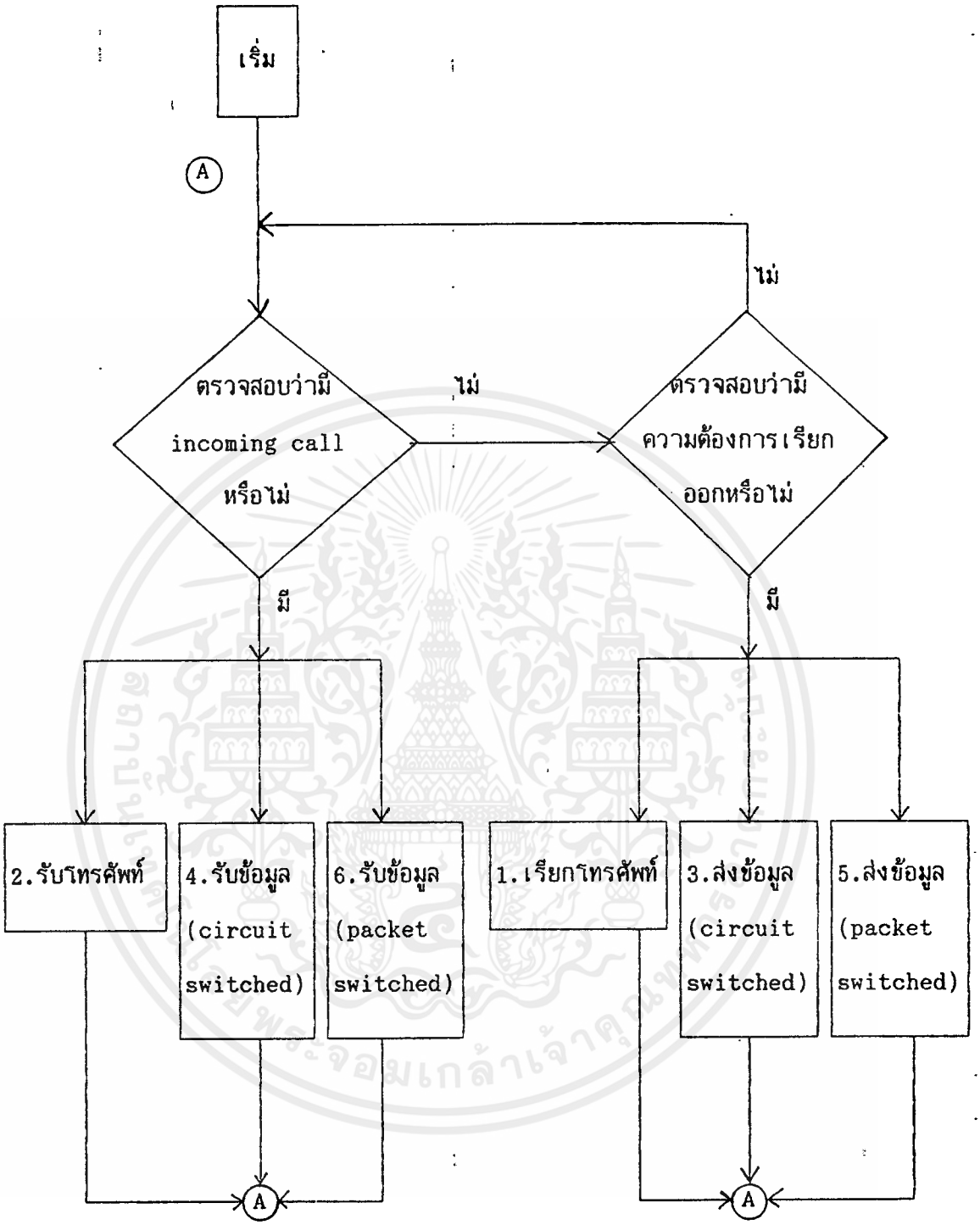


รูปที่ 4.3 แสดงความสัมพันธ์ระหว่างแอปพลิเคชั่น ซอฟต์แวร์ กับ M-card ผ่าน ไดรเวอร์

ในส่วนของ แอปพลิเคชั่นซอฟต์แวร์จะมีอัลกอริทึม ดังต่อไปนี้โดยจะแยกอธิบายเป็นส่วนๆ คือในส่วนการติดต่อโทรศัพท์,การส่ง ข้อมูลในช่องสัญญาณ บี (Circuit Switched Mode) และการส่งข้อมูลในช่องสัญญาณ ดี (Packet Switched Mode)

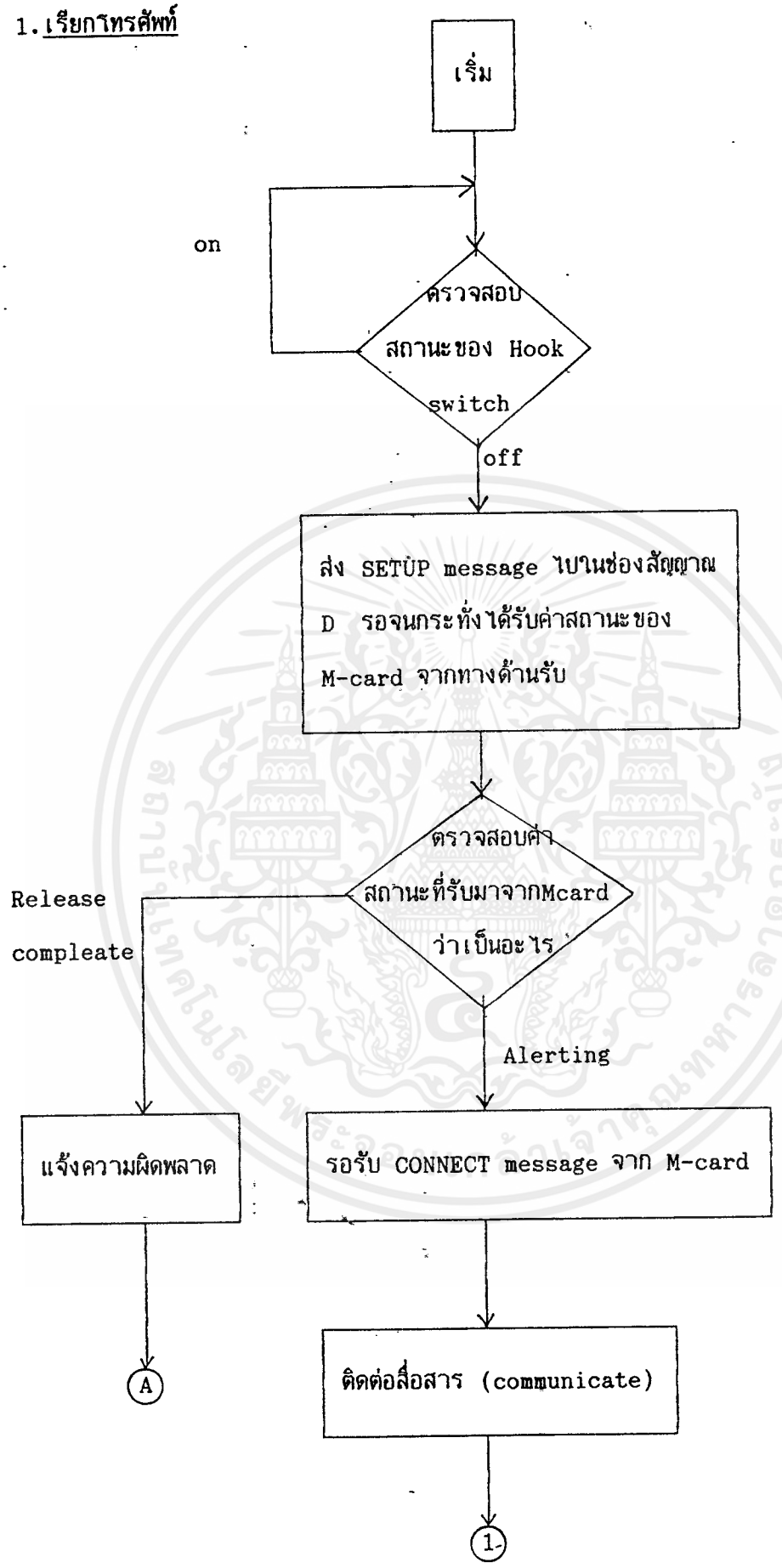
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมหลัก

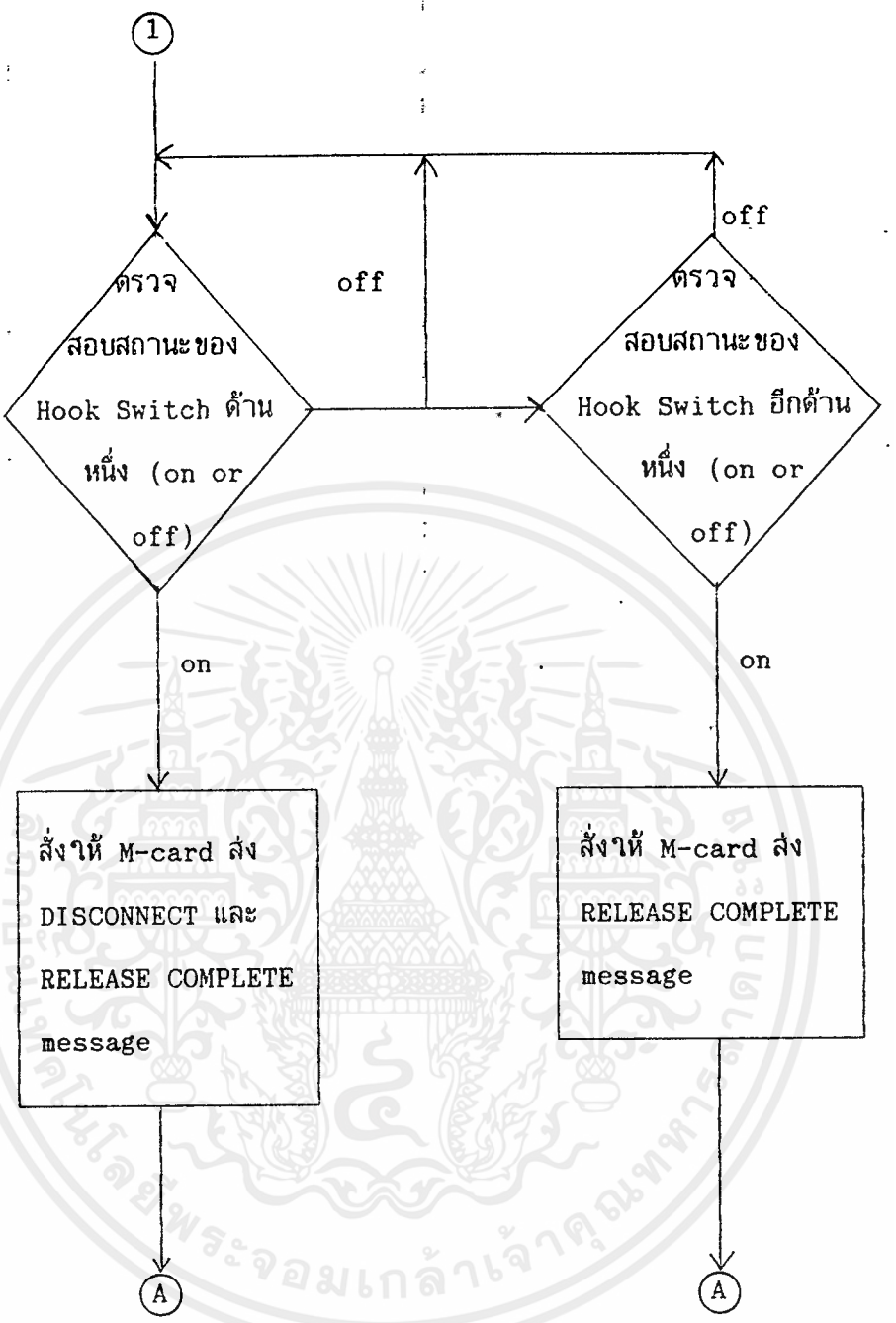


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เรียกโทรศัพท์

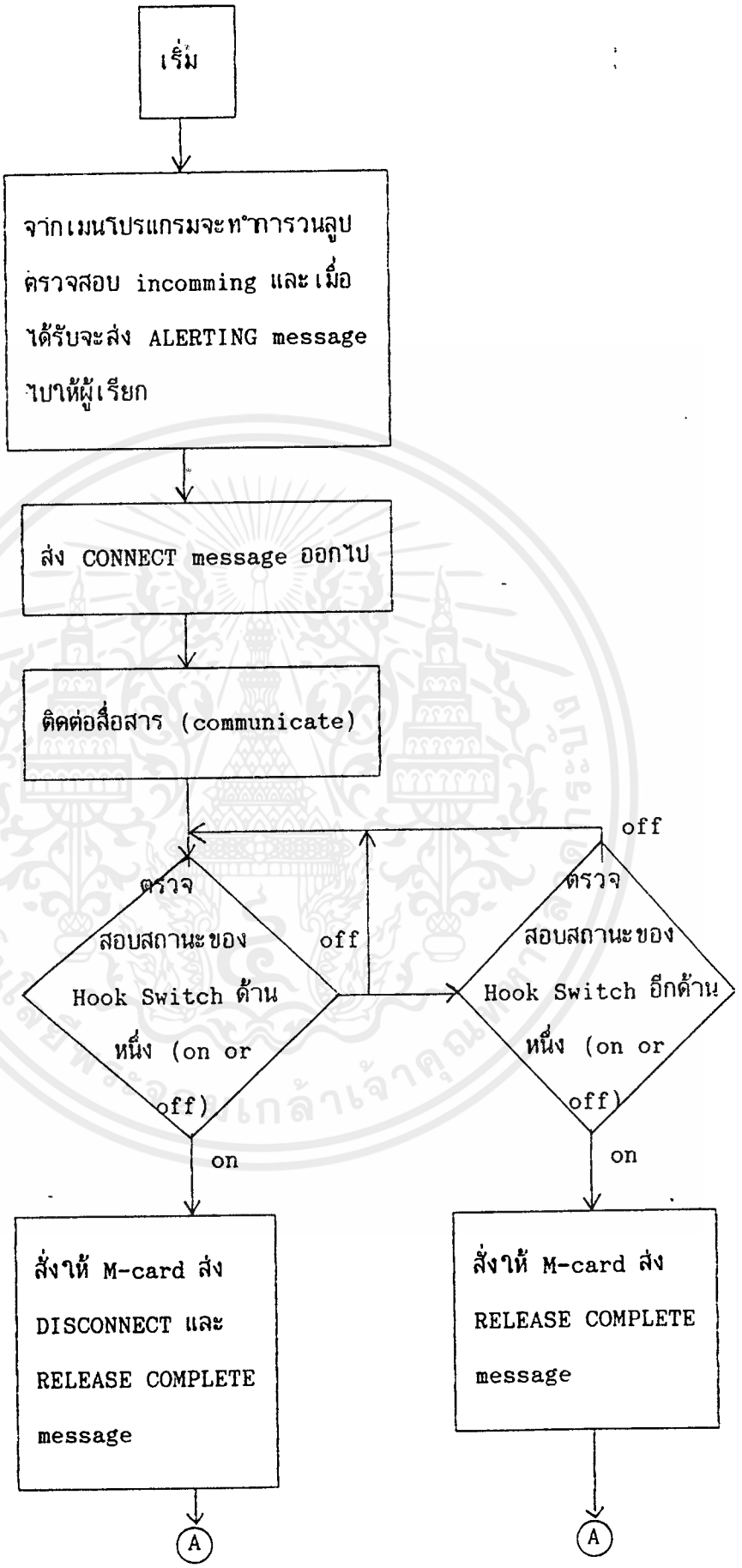


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



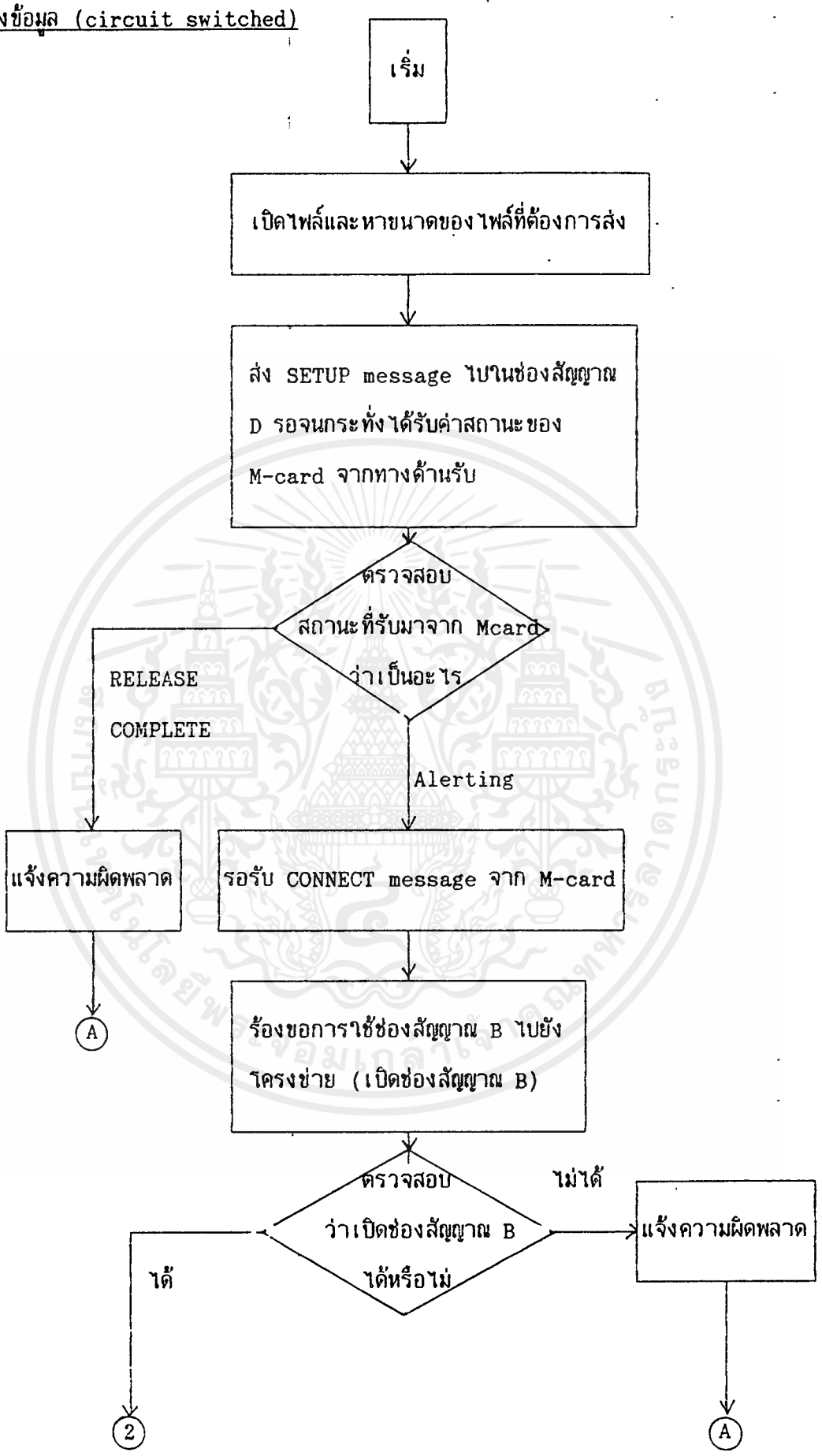
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.รับโทรศัพท์

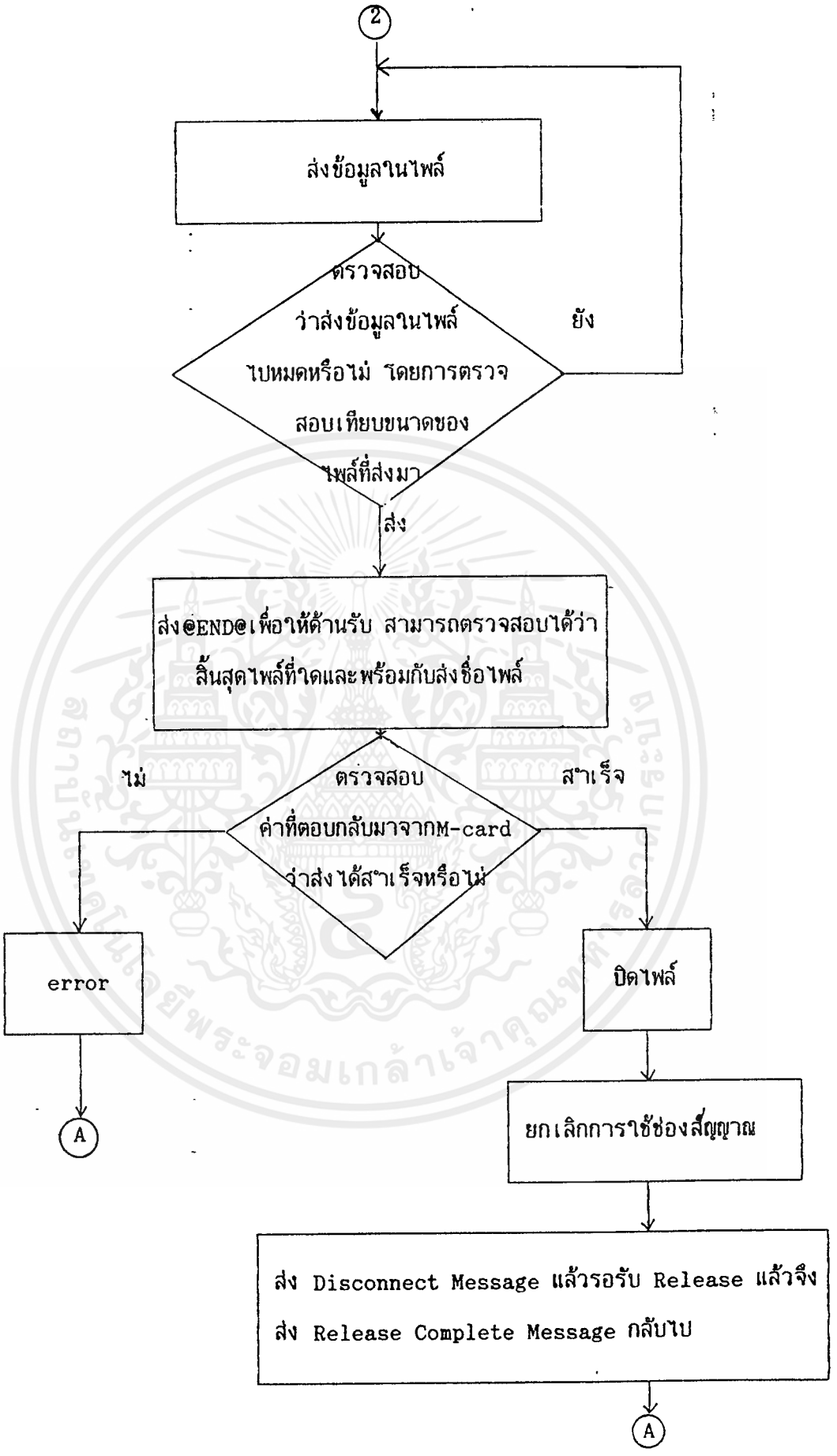


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ส่งข้อมูล (circuit switched)

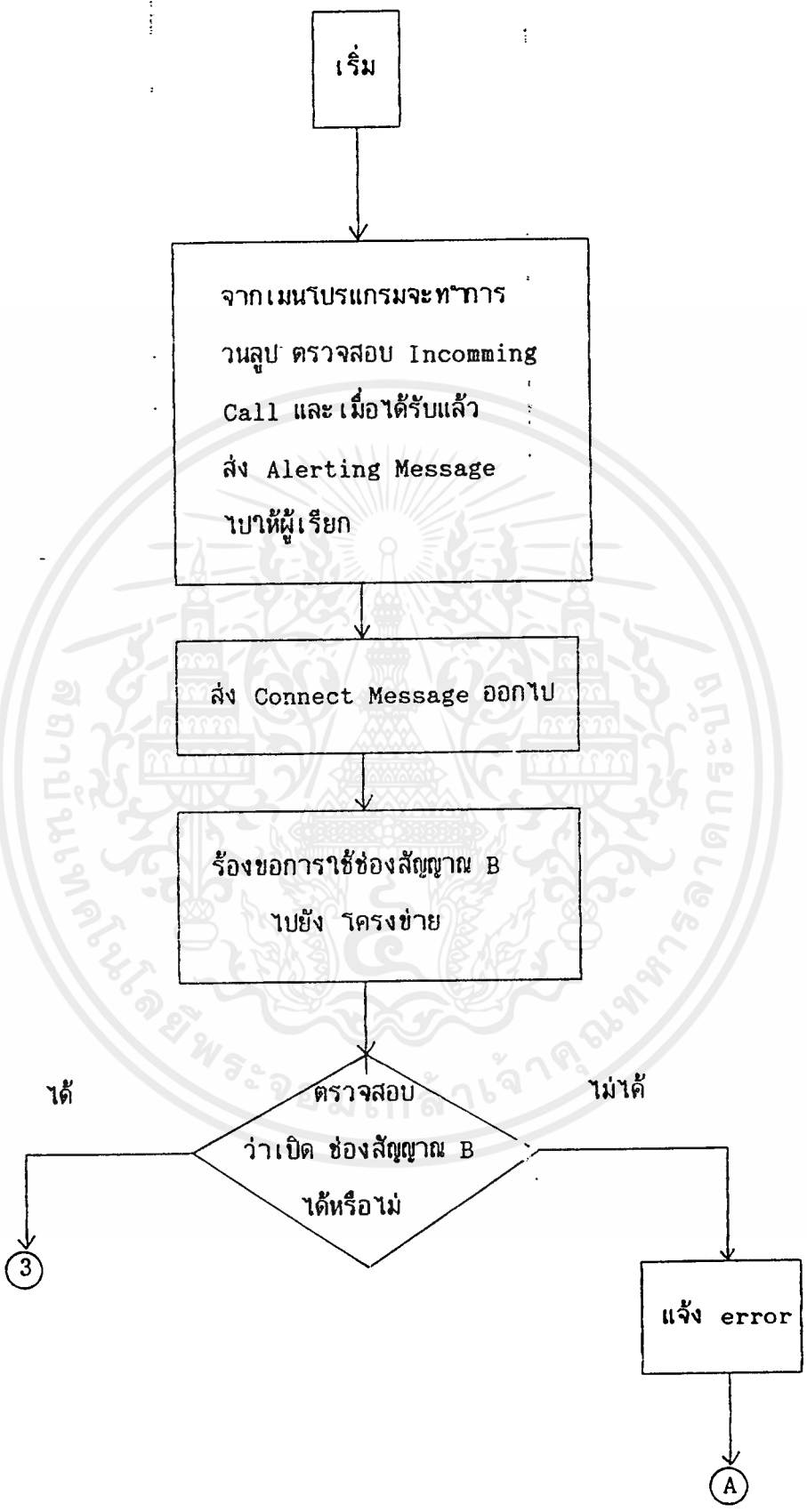


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

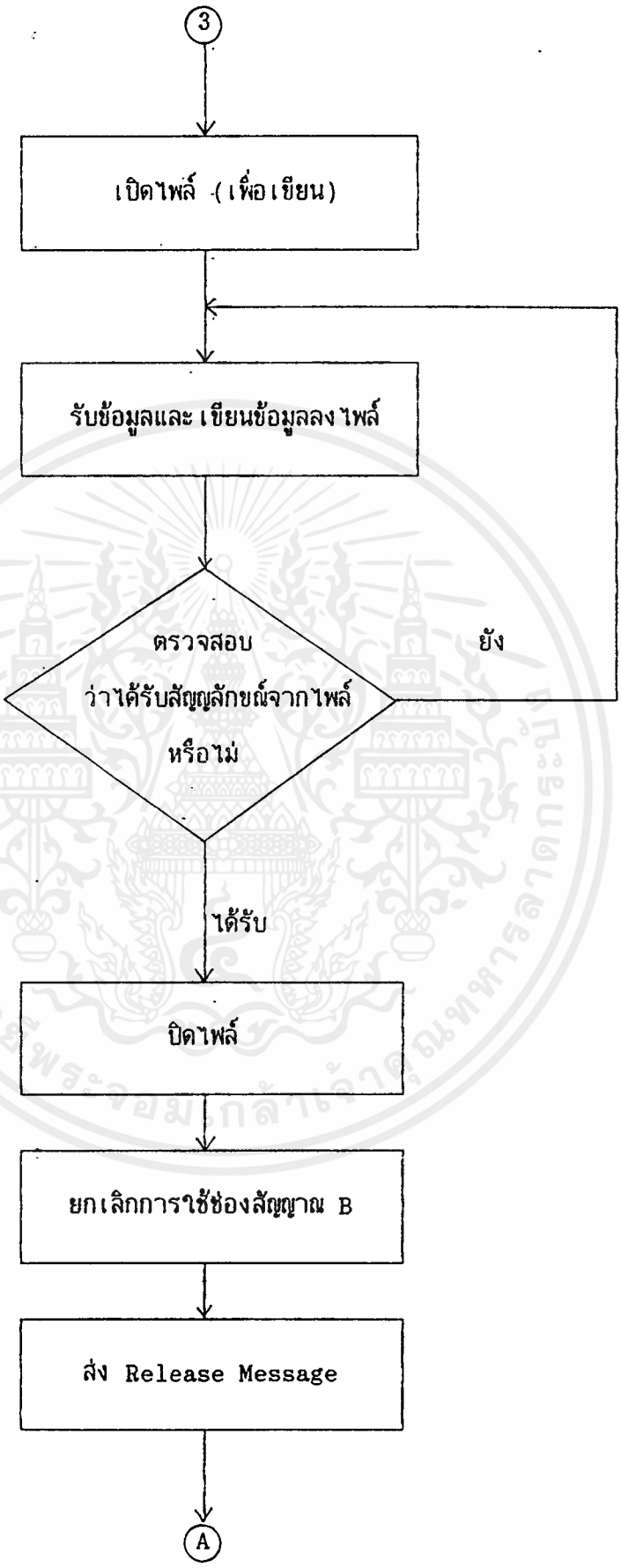


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.<u>รับข้อมูล</u> (Circuit Switched)

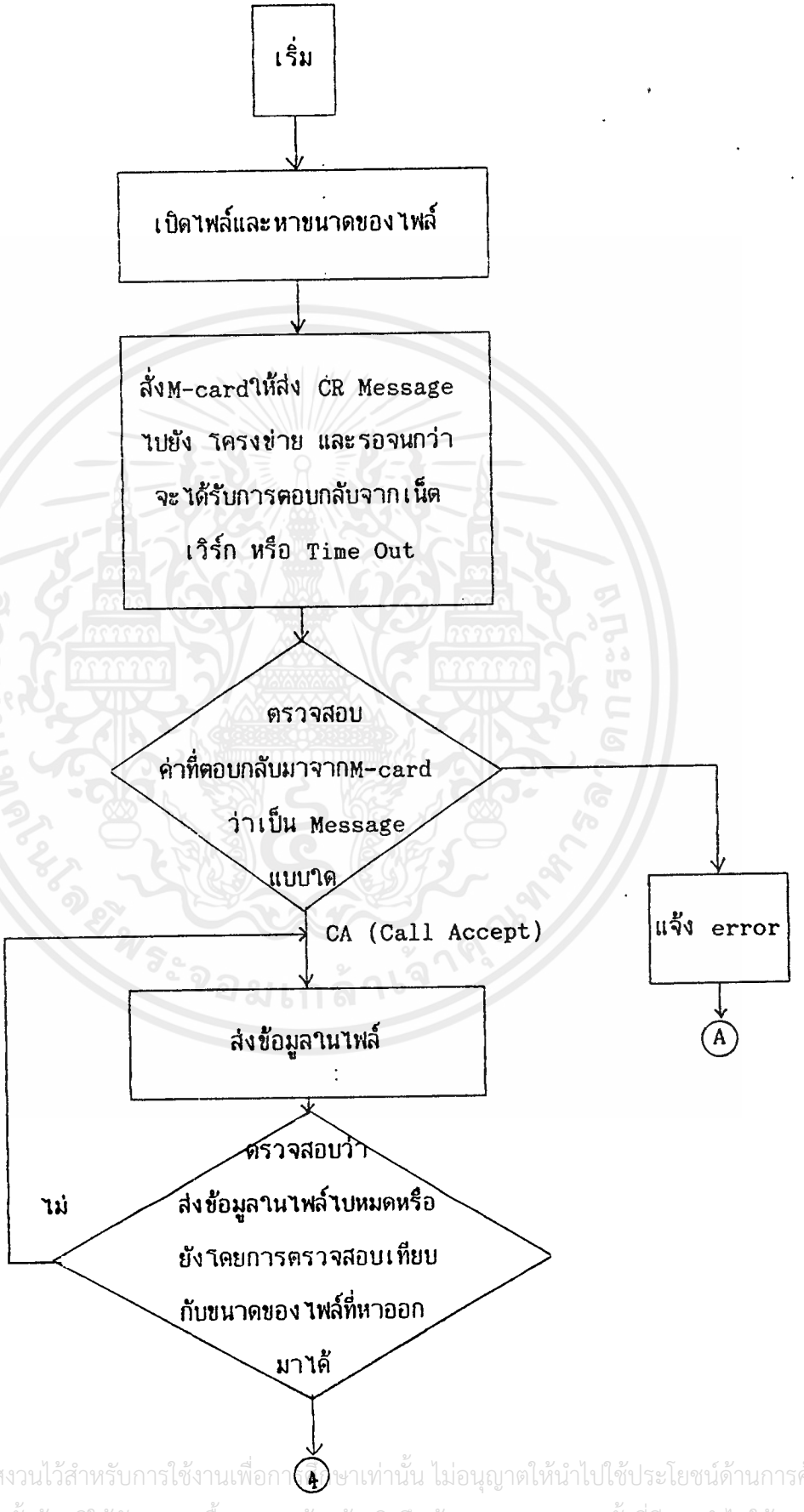


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

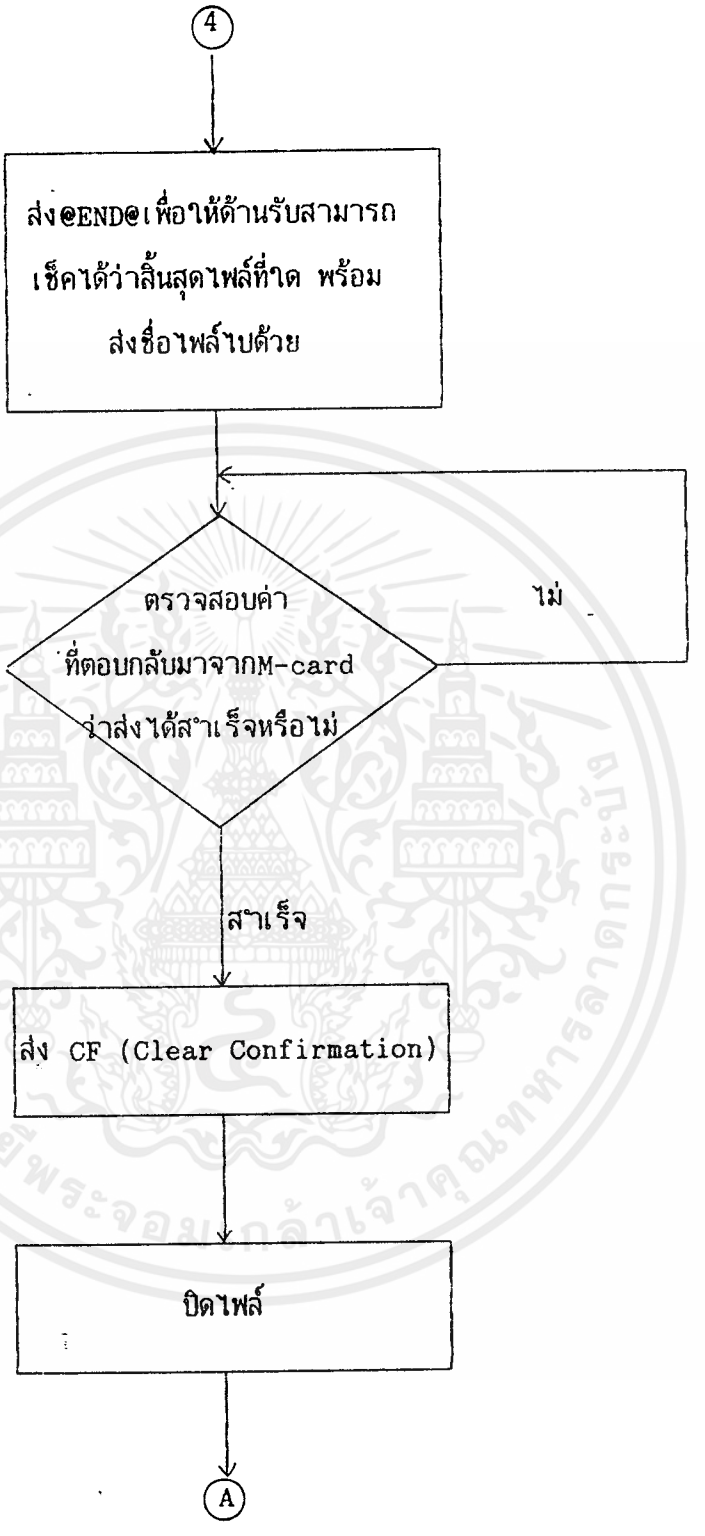


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.ส่งข้อมูล (Packet Switched)

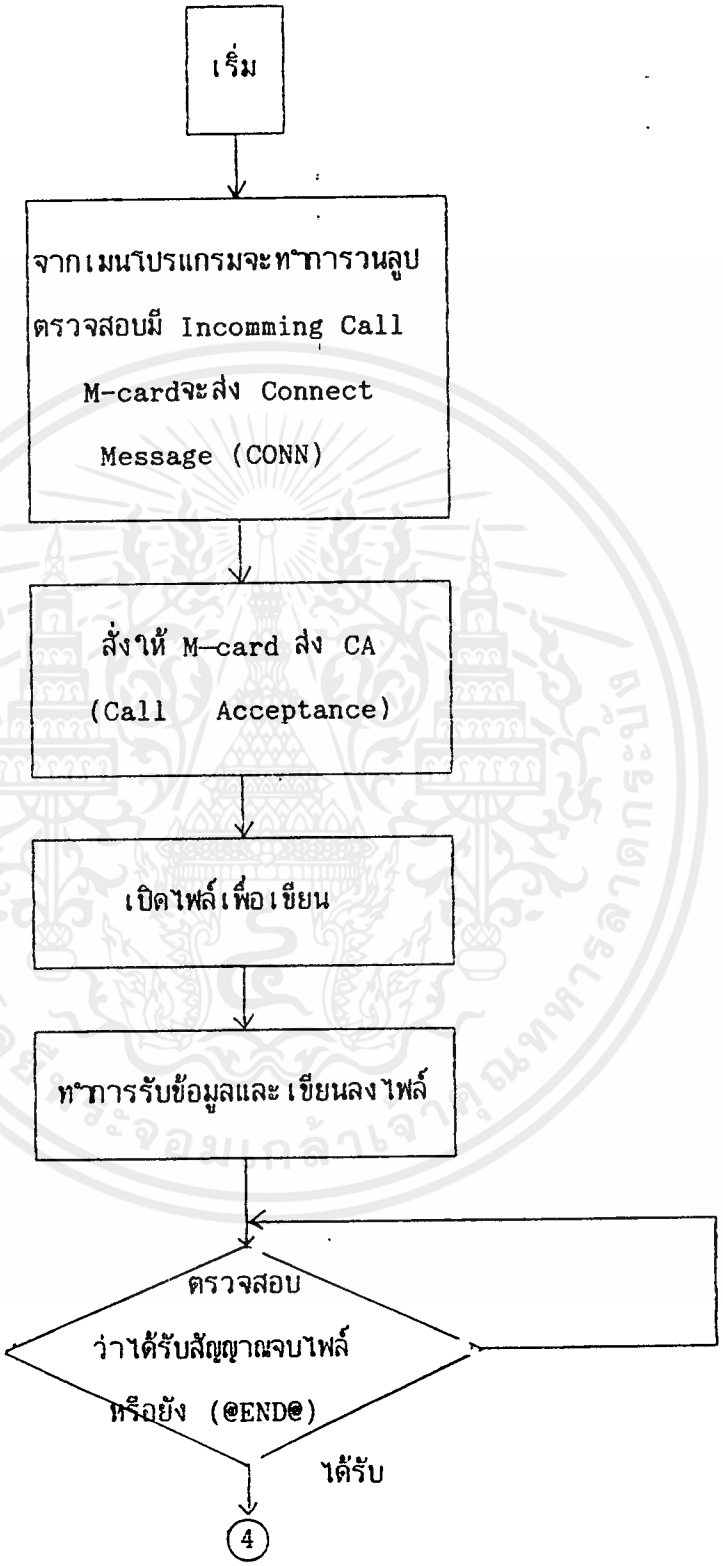


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

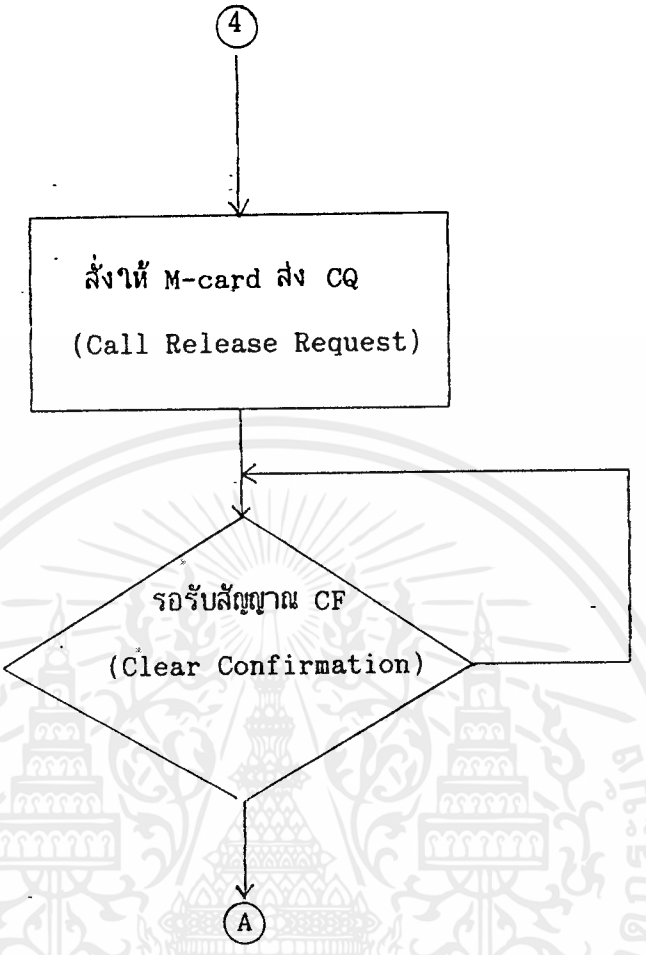


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.รับข้อมูล(Packet Switched)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

ในส่วนของการทดลองได้ทำการใช้ซอฟต์แวร์สื่อสารที่เขียนขึ้น ทำการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ผ่านโครงข่าย ISDN ซึ่งได้จากการซิมมูเลทโดยเครื่องโปรโตคอลอนาไลเซอร์ และเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองนี้คือ NEC 9801 เครื่องโปรโตคอลอนาไลเซอร์ คือ CHAMELEON 32 และเทอร์มินัลอแดปเตอร์ที่ใช้คือ M-card64 ซึ่ง M-card64 นี้จะมีอยู่ 2 พอร์ต คือใช้ต่อแฮนด์เซต (handset) ของโทรศัพท์ และอีกพอร์ตหนึ่งต่อเข้ากับโครงข่าย ISDN

ในการใช้เครื่องโปรโตคอลอนาไลเซอร์เพื่อซิมมูเลทการทำงานของโครงข่าย ISDN ที่พอร์ต A และพอร์ต B จะนำคอมพิวเตอร์มาต่อที่จุด NT โดยที่เลขหมายของพอร์ต A คือ 3333333 และเลขหมายของพอร์ต B คือ 7654321 ในการทดลองได้ทำการทดลองดังต่อไปนี้คือ

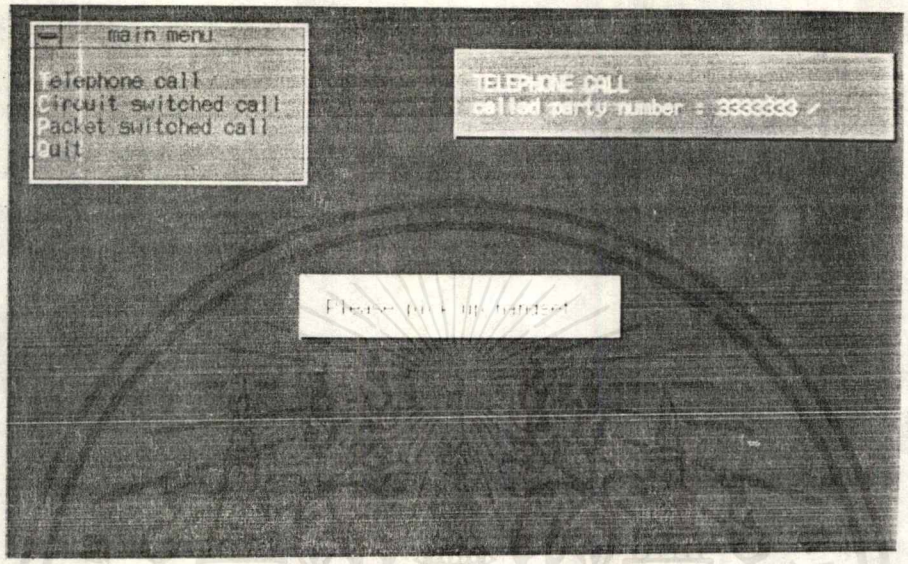
1. ทำการทดลองติดต่อโทรศัพท์
2. ส่งข้อมูลในแบบเซอร์กิตสวิตช์
3. ส่งข้อมูลในแบบแพคเก็ตสวิตช์

ซึ่งผลการทดลองแสดงได้ดังต่อไปนี้

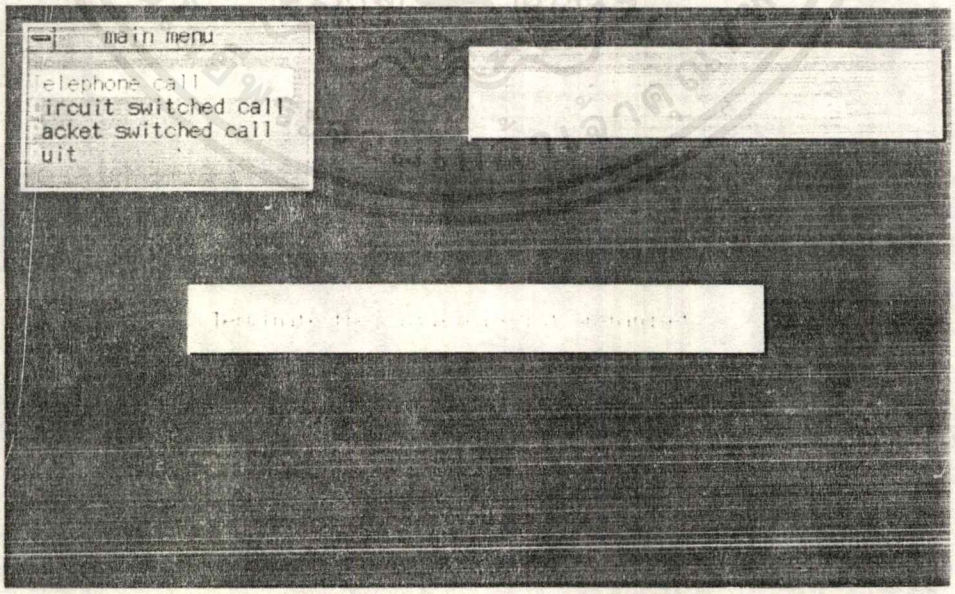
การเรียกและรับโทรศัพท์

เมื่อนำซอฟต์แวร์ที่เขียนขึ้นมาทดลองทำการติดต่อโทรศัพท์ โดยผ่านเครื่องโปรโตคอลอนาไลเซอร์เป็นโครงข่าย ISDN โดยให้เครื่องโทรศัพท์ (แฮนด์เซ็ทที่ต่ออยู่กับ M-card64) ที่ต่ออยู่ที่พอร์ต B ซึ่งมีเลขหมาย 7654321 เรียกมายังเครื่องโทรศัพท์เลขหมาย 3333333 ที่ต่ออยู่กับพอร์ต A ซึ่งการแสดงผลของคอมพิวเตอร์จะแสดงที่จอคอมพิวเตอร์ ซึ่งจะเป็นส่วนที่ติดต่อกับผู้ใช้ เช่น บอกให้ผู้ใช้ใส่เลขหมายปลายทาง ยกหูโทรศัพท์ บอกให้วางหู แจ้งเวลาที่ใช้ในการติดต่อ หรือแม้แต่แจ้งความผิดพลาดที่เกิดขึ้น (ในกรณีที่เกิดความผิดพลาดขึ้น) และนอกจากนั้นการแสดงผลบนจอของเครื่องโปรโตคอลอนาไลเซอร์ ก็จะแสดงลำดับขั้นในการติดต่อระหว่างเครื่องโทรศัพท์ 2 เครื่องนี้ สามารถแสดงตัวอย่างได้ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

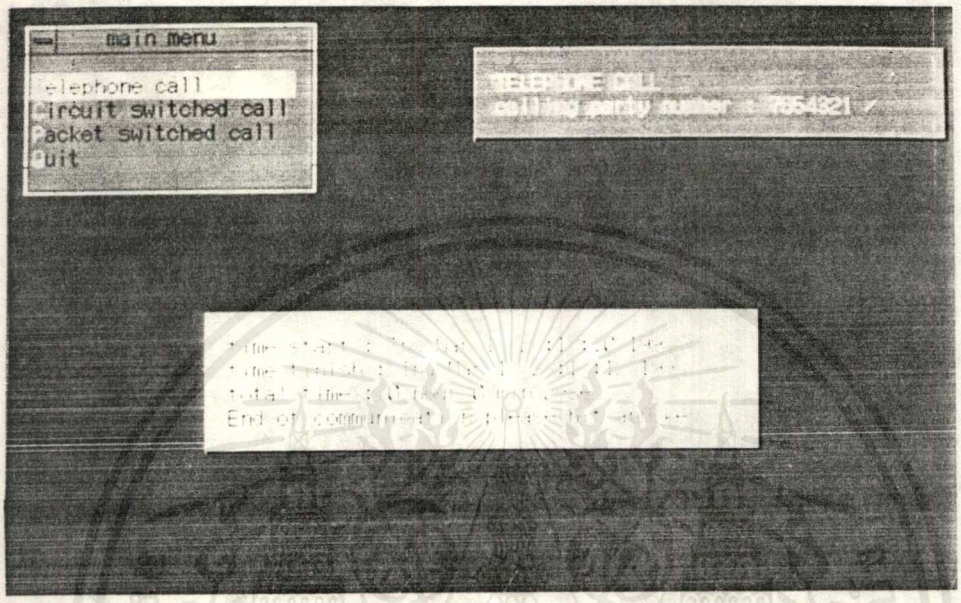


รูปที่ 5.1 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ยกหูโทรศัพท์

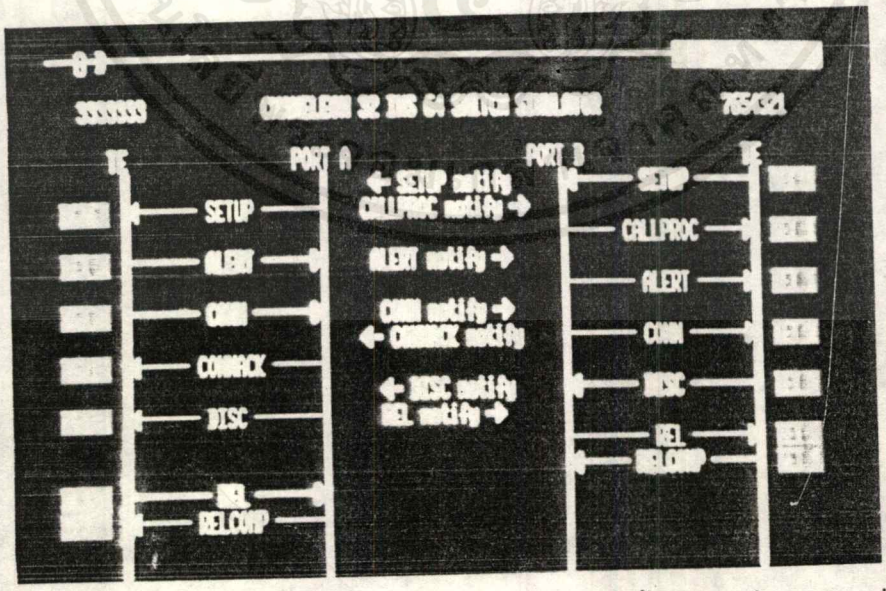


รูปที่ 5.2 การแสดงผลของคอมพิวเตอร์เพื่อบอกผู้ใช้ให้วางหูโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 การแสดงผลของคอมพิวเตอร์เพื่อแจ้งเวลาทั้งหมดที่ใช้ในการติดต่อของผู้ใช้

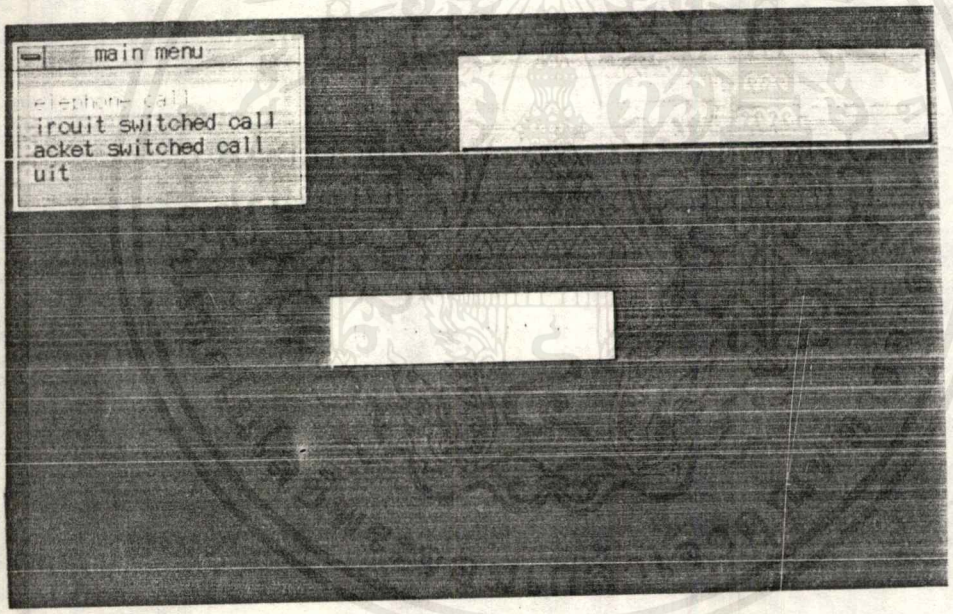


รูปที่ 5.4 การแสดงลำดับขั้นในการติดต่อระหว่างโทรศัพท์ทั้งสองเครื่องโดยเครื่องโปรโตคอลอนาไลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

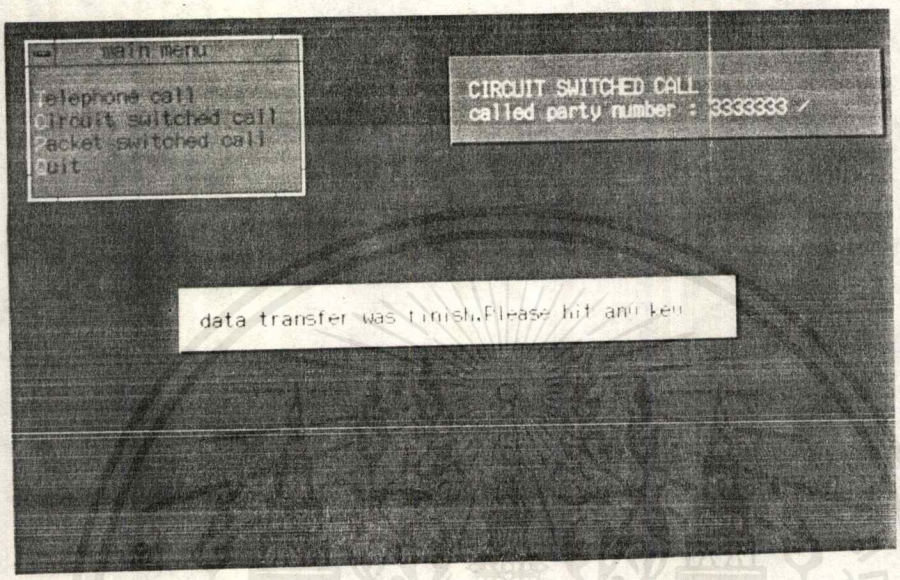
การส่งข้อมูลแบบเซอร์กิตสวิตช์

เมื่อนำซอฟต์แวร์ที่เขียนขึ้นมาทดลองทำการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ โดยผ่านโครงข่าย ISDN ซึ่งได้จากการชิมมูเลทเครื่องโปรโตคอลอนาไลเซอร์ โดยคอมพิวเตอร์ที่ต่อกับพอร์ต B (เลขหมาย 7654321) เรียกไปยังเครื่องคอมพิวเตอร์ที่ต่ออยู่กับพอร์ต A (เลขหมาย 3333333) เพื่อทำการส่งไฟล์ข้อมูล ในการแสดงผลของคอมพิวเตอร์ จะแสดงที่จอคอมพิวเตอร์ซึ่งจะเป็นส่วนที่ติดต่อกับผู้ใช้ คือจะบอกให้ผู้ใช้ใส่เลขหมายปลายทาง ใส่ชื่อไฟล์ แจ้งสถานะว่าส่งหรือรับข้อมูลเรียบร้อยแล้ว หรืออาจจะแจ้งความผิดพลาดที่เกิดขึ้นได้ ซึ่งแสดงตัวอย่างได้ดังรูปต่อไปนี้

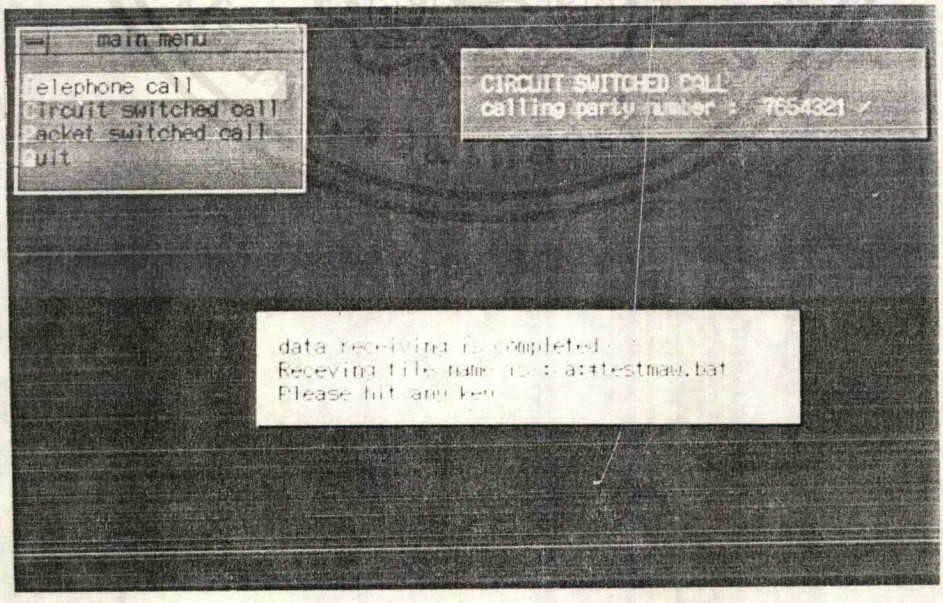


รูปที่ 5.5 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ทราบว่าขณะนี้ทำการรับข้อมูล (ทางด้านรับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

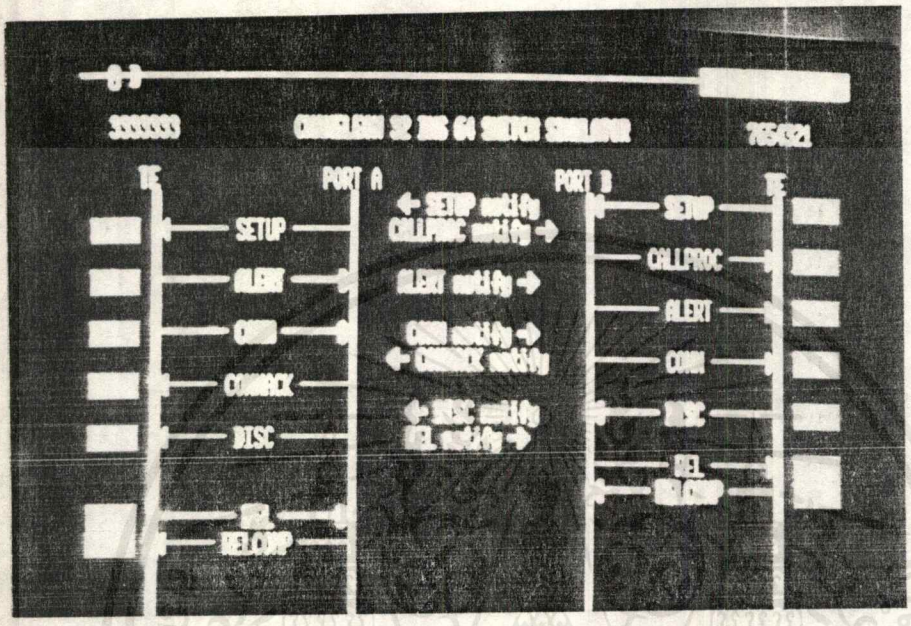


รูปที่ 5.6 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ทราบว่าทำการส่งข้อมูลเรียบร้อยแล้ว (ด้านส่ง)



รูปที่ 5.7 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ทราบว่าทำการรับข้อมูลเสร็จแล้ว (ด้านรับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

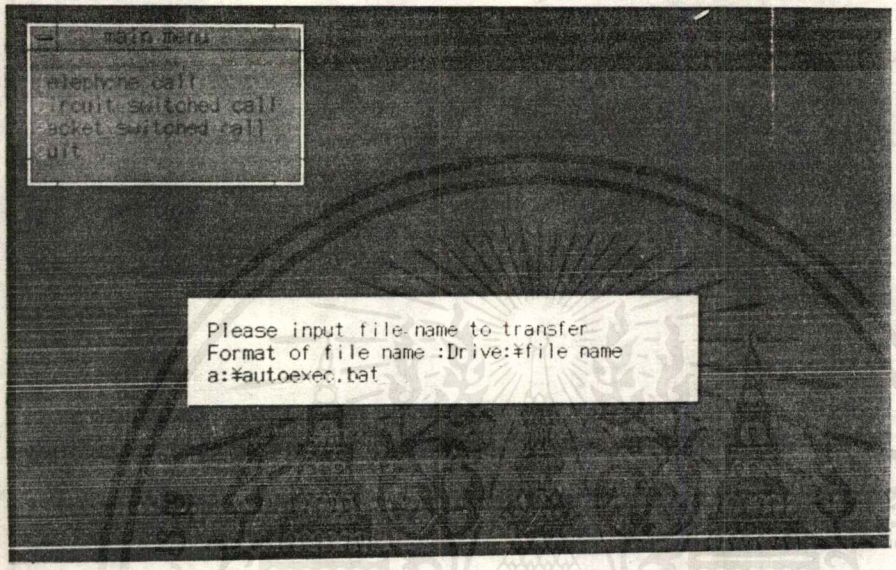


รูปที่ 5.8 การแสดงลำดับขั้นตอนในการติดต่อระหว่างเครื่องคอมพิวเตอร์ในแบบเซอร์กิตสวิตช์

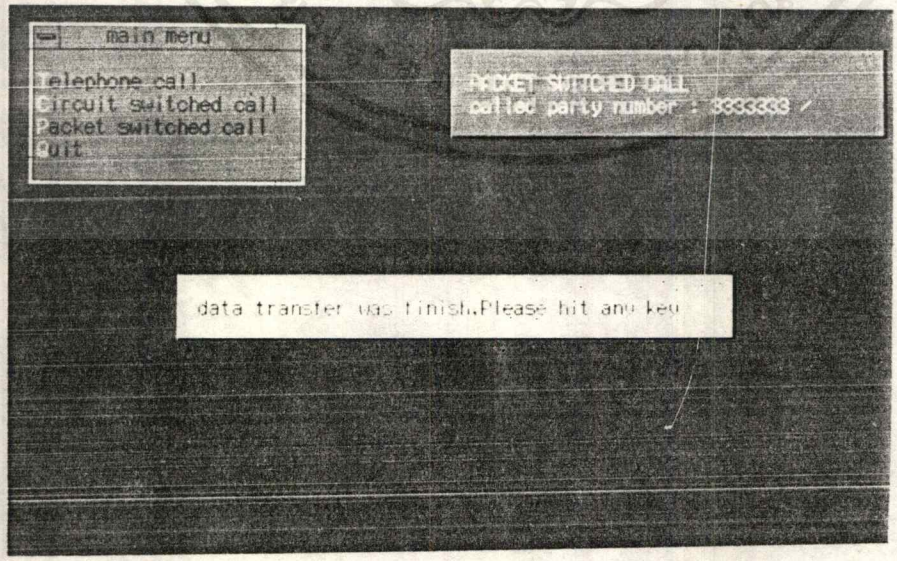
<u>การส่งข้อมูลแบบแพคเก็ตสวิตช์</u>

ในการทดลองทำการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ผ่านโครงข่าย ISDN ซึ่งได้จากการซิมมูเลทเครื่องโปรเซสคอลอนาไลเซอร์ โดยให้คอมพิวเตอร์ที่ต่อกับพอร์ต B (เลขหมาย 7654321) เรียกไปยังเครื่องคอมพิวเตอร์ที่ต่ออยู่กับพอร์ต A (เลขหมาย 3333333) เพื่อทำการส่งไฟล์ ในการแสดงผลของคอมพิวเตอร์จะแสดงที่จอของคอมพิวเตอร์ ซึ่งเป็นส่วนที่ติดต่อกับผู้ใช้ โดยจะบอกให้ผู้ใช้ใส่เลขหมายปลายทาง ใส่ชื่อไฟล์ แจ้งว่าส่งหรือรับข้อมูลเสร็จเรียบร้อยแล้ว หรือในบางครั้งจะแจ้งความผิดพลาดที่เกิดขึ้น (ในกรณีที่เกิดความผิดพลาดเกิดขึ้น) ซึ่งแสดงตัวอย่างได้ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

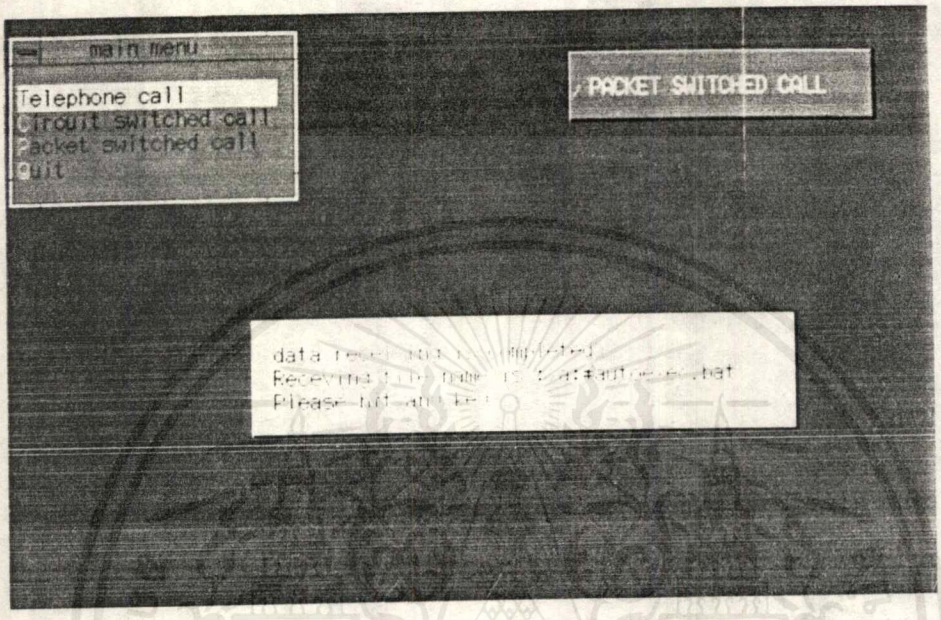


รูปที่ 5.9 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ใส่ชื่อไฟล์ที่ต้องการส่ง

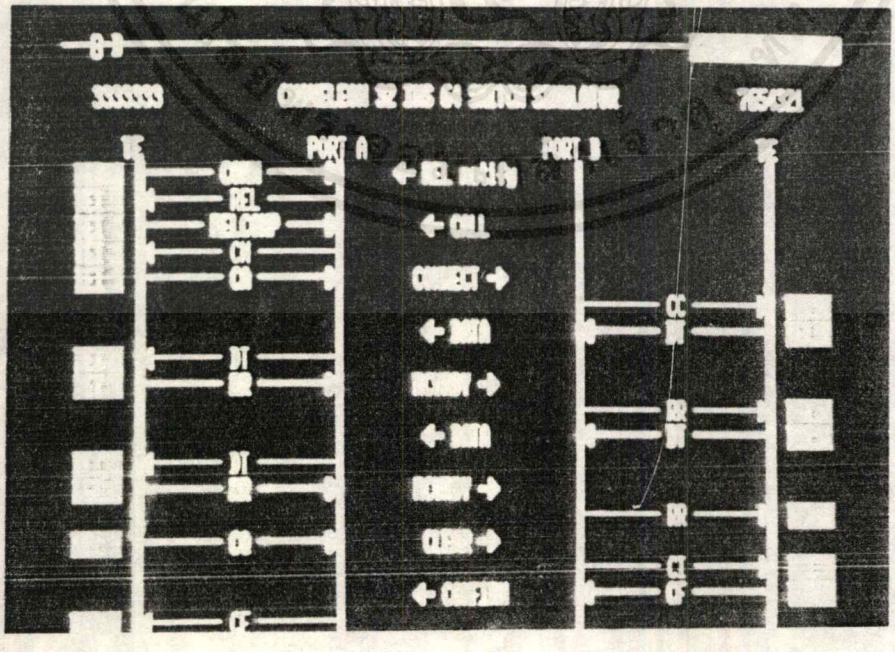


รูปที่ 5.10 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ทราบว่าการส่งข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 การแสดงผลของคอมพิวเตอร์เพื่อบอกให้ผู้ใช้ทราบว่าทำการรับข้อมูลเสร็จแล้ว (ด้านรับ)



รูปที่ 5.12 การแสดงลำดับขั้นในการติดต่อระหว่างคอมพิวเตอร์ในแบบแพคเก็ตสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุป

สำหรับปริญญานิพนธ์ฉบับนี้เป็นการเขียนและพัฒนาซอฟต์แวร์ที่ใช้ใน การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ผ่านโครงข่าย ISDN ซึ่งใช้โปรโตคอลอนาไลเซอร์มาช่วยในการวิเคราะห์ โดยได้ทำการชิมมูเลทเครื่องโปรโตคอลอนาไลเซอร์ให้ทำหน้าที่เป็นโครงข่าย ISDN ซึ่งจะทำให้สามารถวิเคราะห์ข้อมูลที่มีการส่งระหว่างเครื่องคอมพิวเตอร์ได้ จากการทดลองจะพบว่าสามารถทำการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ผ่านโครงข่าย ISDN ได้สำเร็จ ซึ่งในการติดต่อนั้นจะสามารถติดต่อได้ทั้งในแบบเซอร์กิตสวิตช์หรือแพคกิตสวิตช์แล้วแต่ความต้องการของทางผู้ใช้ ดังแสดงในผลการทดลองในบทที่ 5

โดยสรุปแล้วในการเขียนและพัฒนาซอฟต์แวร์ โดยใช้โปรโตคอลอนาไลเซอร์มาช่วยในการวิเคราะห์จะทำให้สามารถแก้ไขปัญหาได้ง่ายขึ้น สามารถตรวจได้ว่าเกิดการผิดพลาดขึ้นที่ใด ทำให้ประหยัดเวลาในการหาข้อผิดพลาด นอกจากนี้แล้วยังสามารถทำให้เข้าใจการทำงานของระบบได้ดียิ่งขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทแทรก

ในบทแทรกนี้จะเป็นส่วนของซอร์สไฟล์ (sourcefile) ที่เขียนขึ้นเพื่อใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์ โดยจะมีส่วนของเมนูหลัก ส่วนของ การติดต่อโทรศัพท์, การส่งข้อมูลในแบบเซอร์กิตสวิตซ์ และการส่งข้อมูลในแบบแพคเก็ตสวิตซ์และนอกจากนี้ยังมีส่วนของการอธิบายวิธีการใช้ฟังก์ชันต่าง ๆ ของ M-card64 พร้อมทั้งบอกถึงชนิดของตัวแปรที่กำหนดขึ้นมาใหม่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include    <stdio.h>
#include    <memory.h>
#include    <conio.h>
#include    <string.h>
#include    <direct.h>
#include    <stdlib.h>
#include    <graph.h>
#include    "isdnlib.h"
#include    "isdndef.h"

void opn_insfile( void );
void initial( void );
void dialing( void );
void _windows(int,int,int,int);
void _dspon(int,int,int,int);
void show(int ,char*);
void showerr(int,char*);
int  sub1( int* );
int  sub2( int* );
int  sub3( int* );
int  sub4( int* );
int  sub5( int* );
int  sub6( int* );
int  sub11( int* );

/*****************************************************************************/
/*       Windows interface functons.                                         */
/*****************************************************************************/

void _windows(int x1,int y1,int x2,int y2)
  {

   _rectangle(_GFILLINTERIOR,x1,y1,x2,y2);
   _setcolor(6);
   _rectangle(_GBORDER,x1,y1,x2,y2);
   _rectangle(_GBORDER,x1+1,y1+1,x2-1,y2-1);
   _setcolor(1);
   _rectangle(_GBORDER,x1-1,y1-1,x2+1,y2+1);
   _rectangle(_GBORDER,x1+2,y1+2,x2-2,y2-2);
   _moveto(x1-1,y1+18); _lineto(x2+1,y1+18);
   _moveto(x1-1,y2-18); _lineto(x1+1,y2-18);
   _moveto(x2-1,y2-18); _lineto(x2+1,y2-18);
   _moveto(x1+20,y1-1); _lineto(x1+20,y1+18);
   _moveto(x1+20,y2-1); _lineto(x1+20,y2+1);
   _moveto(x2-20,y1-1); _lineto(x2-20,y1+1);
   _moveto(x2-20,y2-1); _lineto(x2-20,y2+1);
   _rectangle(_GBORDER,x1+5,y1+8,x1+16,y1+10);
   _setcolor(0);
   _moveto(x1+7,y1+11); _lineto(x1+17,y1+11);
   _moveto(x1+17,y1+9); _lineto(x1+17,y1+11);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/*     diplay on                                                         */
/*************************************************************************/
void _dspon(int x1,int y1,int x2,int y2)
{
  _rectangle(_GFILLINTERIOR,x1,y1,x2,y2);
  _setcolor(10);
  _rectangle(_GBORDER,x1,y1,x2,y2);
  _setcolor(15);
  _moveto(x1+1,y1+1); _lineto(x2-2,y1+1);
  _moveto(x1+1,y1+1); _lineto(x1+1,y2-1);
  _setcolor(0);
  _rectangle(_GFILLINTERIOR,x1+2,y2-2,x2-1,y2-1);
  _rectangle(_GFILLINTERIOR,x2-1,y1+1,x2-2,y2-1);
}

/*************************************************************************/
/*       set position display                                            */
/*************************************************************************/
void show(int leng,char *strg)
{
_setcolor(7);
_dspon(296-leng*4,192,344+leng*4,240);
_settextposition(14,40-leng/2);
_settextcolor(0);
_outtext(strg);
}

/*************************************************************************/
/*       display error                                                   */
/*************************************************************************/
void showerr(int leng, char *strg)
{
_setcolor(4);
_dspon(228-leng*4,336,248+leng*4,384);
_settextposition(23,30-leng/2);
_settextcolor(7);
_outtext(strg);
getch();
_setcolor(1);
_rectangle(_GFILLINTERIOR,228-leng*4,332,252+leng*4,388);
_settextposition(23,30-leng/2);
_outtext("                                                              ");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/*       main                                                            */
/*************************************************************************/
void main()
{
int  process;
int  result;
int  y,key;
initial();
opn_insfile();
result = 0;
callid[0] = -1;
callid[1] = -1;
callid[2] = -1;

_setvideomode(_98RESS16COLOR);
_settextcursor( 0x2000  );
_setcolor(1);
_windows(1,1,639,399);
_settextcolor(7);
_settextposition(8,35) ;_outtext("WELCOME TO                                    ");
_settextposition(10,27); _outtext(" ISDN COMMUNICATION SOFTWARE                 ");
_settextposition(12,18);_outtext("DEPARTMENT OF TELECOMMUNICATION ENGINEERING ");
_settextposition(14,37);_outtext("KMITL                                         ");
_settextposition(17,27);_outtext("press any key to continue                     ");
getch();
_settextposition(8,35); _outtext("                                              ");
_settextposition(10,27); _outtext("                                             ");
_settextposition(12,18);_outtext("                                              ");
_settextposition(14,37);_outtext("                                              ");
_settextposition(17,27);_outtext("                                              ");
/****more detials about program ****/

y=4;
do{
_setcolor(3);
_windows(18,16,208,128);
_setcolor(3);_settextcolor(0);
_settextposition(2,4);_outtext("       main menu       ");
_settextcolor(4);
_settextposition(4,4);_outtext("T");
_settextcolor(0);
_settextposition(4,5);_outtext("elephone call");
_settextcolor(4);
_settextposition(5,4);_outtext("C");
_settextcolor(0);
_settextposition(5,5);_outtext("ircuit switched call");
_settextcolor(4);
_settextposition(6,4);_outtext("P");
_settextcolor(0);
_settextposition(6,5);_outtext("acket switched call");
_settextcolor(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_settextposition(7,4);_outtext("Q");
_settextcolor(0);
_settextposition(7,5);_outtext("uit");
/*check Mcard and wait for Hit Key */
do{
_setcolor(6);
_rectangle(_GFILLINTERIOR,24,16*(y-1),194,16*y);
do{
  process = -1;
    if ( callid[ 0 ] < 0 ) {
#if __MCARD
      callid[ 0 ] = tacptck( &xret_info ); /* check inconing speech call */
#endif
      if (callid[ 0 ] >= 0) {
        process = 1;
        }
      }
else {
#if __MCARD
    status=ssense(callid[ 0 ],NULL,NULL,NULL);
#else
    status = 0xa00;
#endif
    status=status & 0x3f00;
    if(status==0) {
        process = 4;
      }
}
    if ( callid[ 1 ] < 0 && callid[ 0 ] < 0) {
#if __MCARD
      callid[ 1 ] = dacptck( &xret_info );   /* check incoming data call */
#endif
      if (callid[ 1 ] >= 0) {
        process = 2;
        }
      }
    if ( callid[ 2 ] < 0 && callid[ 0 ] <0 && callid[ 1 ] <0) {
#if __MCARD
      callid[ 2 ] = pacptck( &pret_info ); /* check incoming packet call */
#endif
       if (callid[ 2 ] >= 0) {
        process = 3;
        }
      }
  switch( process ) {
    case 1:
      result = sub2( &callid[ 0 ] );
     break;
    case 2:
      result = sub4( &callid[ 1 ] );
      break;
    case 3:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      result = sub6( &callid[ 2 ] );
      break;
    case 4:
      result = sub11( &callid[ 0 ]);
      break;
    }

}while( !(kbhit()) );
key=getch();
_setcolor(3);
_rectangle(_GFILLINTERIOR,24,16*(y-1),194,16*y);
   switch(key){
  case 0x0b:if(y>4)y--; else y=7;  break;
  case 0x0a:if(y<7)y++; else y=4;  break;
  case 't' :{y=4; key=0x0d;}break;
  case 'c' :{y=5; key=0x0d;}break;
  case 'p' :{y=6; key=0x0d;}break;
  case 'q' :{y=7; key=0x0d;}break;  }

}while(key!=0x0d);
switch(y){
   case 4:
     result=sub1(&callid[ 0 ]);break;
   case 5:
     result=sub3(&callid[ 1 ]);break;
   case 6:
     result=sub5(&callid[ 2 ]);break;  }

}while(y!=7);
#if __MCARD
status=scclose();
#endif
_clearscreen( _GCLEARSCREEN );
_setvideomode(_98TEXT80);
_settextcursor(0x0f0f);
 }

/*****************************************************************************/
/*       initialize                                                          */
/*****************************************************************************/
void initial()
{
/*set up data*/
setup_para.pcadd    = scadd;
setup_para.pcsubadd= NULL;
setup_para.pgadd    = NULL;
setup_para.pgsubadd= NULL;
setup_para.pkeypad = NULL;
setup_para.puserinf= NULL;

/*return data*/
xret_info.pcadd     = pcadd;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xret_info.pcsubadd = pcsubadd;
xret_info.pgadd    = pgadd;
xret_info.pgsubadd = pgsubadd;
xret_info.puserinf = &puserinf;
xret_info.pcause   = &pcause;
xret_info.pcharge  = &pcharge;
xret_info.pchannel = &pchannel;

/*set up data for packet switching*/
memset(&cr_para, 0, sizeof(XPCRPARA) );
cr_para.pcadd    = scadd;                /* called address          */
cr_para.pcsubadd = scsubadd;             /* called subaddress       */
cr_para.ppktsize = ppktsize;             /* packet size             */
cr_para.pwndsize = pwndsize;             /* window size pointer     */
cr_para.pthrough = &pthrough;            /* throughput class        */
ppktsize[0]      = 128;
ppktsize[1]      = 128;
pwndsize[0]      = 4;
pwndsize[1]      = 4;
pthrough         = 0xaa;

memset(&pret_info, 0, sizeof(XPCRPARA) );
pret_info.pcadd    = pcadd;
pret_info.pcsubadd = pcsubadd;
pret_info.pgadd    = pgadd;
pret_info.pgsubadd = pgsubadd;
pret_info.puserinf = &puserinf;

memset(&cq_para, 0, sizeof(XPCQPARA) );
cq_para.pdiag = &pdiag;
cq_para.cause = 0;
cq_para.puserinf = &puserinf;
cq_para.puserinf = NULL;
pdiag = 0;

memset(&ca_para, 0, sizeof(XPCAPARA) );
ca_para.ppktsize = ppktsize;
ca_para.pwndsize = pwndsize;
ca_para.pthrough = &pthrough;
}

/*************************************************************************/
/*       open insfile                                                    */
/*************************************************************************/
void opn_insfile()
{
getcwd (path, 20);
chdir("a:¥¥isdn¥¥dev");
fp=fopen("a:inssetup.dat", "rb");
chdir(path);
if (fp == NULL) {
  showerr(42,"inssetup.dat can't open file,Hit any key");
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  getch();
  _setcolor(1);
  _rectangle(_GFILLINTERIOR,28,384,420,388);
  _settextposition(23,7);
  _outtext(prnbuf);
  exit(0);
  }
for (i=0; i<1024; i++) {
  if (feof(fp) != (int) NULL) break;       /* eof ? */
    k=(uchar)fgetc(fp);
    ins_file[i]=(uchar)k;
  }
fclose(fp);
#if __MCARD
status=scopen(ins_file);                   /* start communication */
if ( status != 0 ) {
  showerr(41,"can't execute scopen function,Hit any key");
  getch();
  _setcolor(1);
  _rectangle(_GFILLINTERIOR,32,348,416,388);
  _settextposition(23,7);
  _outtext(prnbuf);
  exit(0);
  }
#endif
}

/*************************************************************************/
/*       input dial number                                               */
/*************************************************************************/
void dialing()
{
int  i, n, nmax, eoi, s;
char k, dnum[ 26 ];
_setcolor(7);
_settextcolor(0);
_dspon(120,192,520,256);
sprintf(prnbuf,"Please input dial number and push RETURN key");
_settextposition(14,18);
_outtext(prnbuf);

 n         = 0;
 nmax      = 0;
 eoi       = 0;
 dnum[ 0 ] = '¥0';
 _settextcursor ( 0x0f0f );                /* cursor on ( 25line, '_') */
 for ( ; ; ) {
   _settextposition( 15,18  + n );
   k = (char) getch();
   switch ( k ) {
     case K_DEL:
       if ( n > 0 ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nmax   = nmax - 1;
        n      = n - 1;
        for ( i = n; i < nmax; i++ ) {
          dnum[ i ] = dnum[ i + 1 ];
          }
        _settextposition( 15, 18 + nmax );
        printf( " " );
        dnum[ nmax ] = '¥0';
        _settextposition( 15,18 + n );
        _outtext( &dnum[ n ] );
        }
      break;
    case K_LEFT:
      if ( n > 0 ) {
        _settextposition( 15,18 + n );
        n = n - 1;
        }
      break;
    case K_RIGHT:
      if ( n < nmax ) {
        _settextposition( 15, 18 + n );
        n = n + 1;
        }
      break;
    case '0':
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
    case '/':
      if ( n < 25 ) {
        dnum[ n ] = k;
        if ( n == nmax ) {
          nmax++;
          dnum[ nmax ] = '¥0';
          }
        _settextposition( 15, 18 + n );
        _outtext( &dnum[ n ] );
        n++;
        break;
        }
    case K_CR:
      dnum[ n ] = '¥0';
      eoi = 1;
      break;
    }
  if ( eoi == 1 ) break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
_settextcursor ( 0x2000 );                    /* cursor off */
_settextposition( 15,18 );
if (( nmax > 0 ) && ( dnum[ 0 ] != '/' )) {
  n = 0;
  s = 0;
  for ( i = 0; i < nmax; i++ ) {
    if ( dnum[ i ] == '/' ) {
      n = 0;
      s = 1;
      }
    else {
      if ( s == 0 ) {
        scadd[ n ]     = dnum[ i ];
        scadd[ n + 1 ] = '¥0';
        n++;
        }
      else {
        if ( n > 4 ) break;
        scsubadd[ n ]     = dnum[ i ];
        scsubadd[ n + 1 ] = '¥0';
        n++;
        }
      }
    }
  }
_setcolor(1);
_rectangle( _GFILLINTERIOR,116,188,524,260);
_settextposition(14,18);
_outtext("                                                    ");
_settextposition(15,18);
_outtext("                                                    ");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include    <stdio.h>
#include    <memory.h>
#include    <conio.h>
#include    <string.h>
#include    <direct.h>
#include    <stdlib.h>
#include    <graph.h>
#include    <time.h>
#include    "isdnlib.h"
#include    "isdndef.h"

time_t  start,finish;
void wela ( time_t,time_t);
void dialing( void );
void opn_insfile(void);
void show(int,char*);
void showerr(int,char*);
void _dspon(int,int,int,int);

/*************************************************************************/
/*       calling ( circuit switching / speech)                           */
/*************************************************************************/
int sub1( int* cidad )
{
    int err_flg;
    start = 0;
    err_flg = 0;
    if (( callid[ 1 ] < 0 ) && ( callid[ 2 ] < 0 )) {
        dialing();
       }
    _setcolor(5);
    _settextcolor(7);
    _dspon(312,32,632,96);
    sprintf(prnbuf,"TELEPHONE CALL");
    _settextposition(4,42);
    _outtext(prnbuf);
    sprintf(prnbuf,"called party number : %s / %s",scadd,scsubadd);
    _settextposition(5,42);
    _outtext(prnbuf);
    #if __MCARD
       status = thook();                          /*  sense hook switch   */
    #endif
    if (status == 0) {
      show(23,"Please pick up handset");
       #if __MCARD
         while (status == 0) {                    /*  on hook or off hook */
             status=thook();                      /*  sense hook switch   */
           }
       #endif
      }
     _setcolor(1);
     _rectangle(_GFILLINTERIOR,200,188,440,244);
```

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_settextposition(14,28);
_outtext("                                    ");
#if __MCARD
timeout_count=60;                              /* time out count 60 sec */
     /* request a call */
  *cidad = tdial(&setup_para, &xret_info, timeout_count);
#endif
  if (*cidad < 0){
     switch ( *cidad ){
     case (int)0xfb01:
         showerr(54,"line was cut off because time out,please hit any key");
         _setcolor(1);
         _rectangle(_GFILLINTERIOR,308,28,636,100);
         _settextposition(4,42);
         _outtext("                                                    ");
         _settextposition(5,42);
         _outtext("                                                    ");
         err_flg = *cidad;
         *cidad = -1;
         return -1;
         break;
      default:
         showerr(56,"error was detected so line was cut off,Please hit any key");
         _setcolor(1);
         _rectangle(_GFILLINTERIOR,308,28,636,100);
         _settextposition(4,42);
         _outtext("                                                    ");
         _settextposition(5,42);
         _outtext("                                                    ");
         err_flg = *cidad;
         *cidad = -1;
         return -1;
         break;
      }
     #if __MCARD
     status=thook();
     #endif
        if(status!=0) {
            showerr(37,"Please put on handset and hit any key");
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,308,28,636,100);
            _settextposition(4,42);
            _outtext("                                                    ");
            _settextposition(5,42);
            _outtext("                                                    ");
             #if __MCARD
              while((thook() !=0) || (ssense(*cidad,NULL,NULL,NULL) !=0)) {
                 }
             #endif
          }
        else {
            showerr(37,"error was detected,Please hit any key");
```

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,308,28,636,100);
            _settextposition(4,42);
            _outtext("                                          ");
            _settextposition(5,42);
            _outtext("                                          ");
          }
       }
    else {
       time(&start);
       show(42,"Terminate the call,Please put on handset");
      }
    if(err_flg==0) return 0;
    else {
       *cidad = -1;
       return -1;
       }
    }

/*************************************************************************/
/*       called ( circuit switching / speech)                            */
/*************************************************************************/
int sub2( int* cidad )
{
   int err_flg;
   start = 0;
   err_flg = 0;

    _setcolor(5);
    _settextcolor(7);
    _dspon(312,32,632,96);
    sprintf(prnbuf,"TELEPHONE CALL");
    _settextposition(4,42);
    _outtext(prnbuf);
    sprintf(prnbuf,"calling party number : %s / %s",pgadd,pgsubadd);
    _settextposition(5,42);
    _outtext(prnbuf);
   #if __MCARD
      status = thook();                    /*  sense hook switch  */
   #endif
      if (status == 0) {
        show(23,"Please pick up handset");
      #if __MCARD
         while (status == 0) {
         status=ssense(*cidad, NULL, NULL, NULL);
         status=status & 0x3f00;
             if (status == 0) break;
         status = thook();                 /*  sense hook switch  */
           }
      #endif
      status=ssense(*cidad,NULL,NULL,NULL);
      status=status & 0x3f00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if (status != 0) {
      time(&start);
      _setcolor(1);
      _rectangle(_GFILLINTERIOR,200,188,440,244);
      _settextposition(14,28);
       outtext("                                                    ");
      show(42,"Terminate the call,Please put on handset");
      }
       else {
          #if __MCARD
            status=thook();
          #endif
            if ( status != 0 ) {
               #if __MCARD
                 while(( thook()!= 0 )||(ssense(*cidad,NULL,NULL,NULL)!= 0 )){
                   }
          #endif
         }
       err_flg = 1;
      }
    }
    if ( err_flg == 0 ) return  0;
    else
        return -1;
}

/*****************************************************************************/
/*       disconnect ( circuit switching / speech )                           */
/*****************************************************************************/
int subll( int* cidad )
{
#if __MCARD
    status=thook();
#endif
    if ( status != 0 ) {
      time(&finish);
      status = scclose();
      status = scopen(ins_file);
      _setcolor(1);
      _rectangle(_GFILLINTERIOR,124,188,516,244);
      _settextposition(14,19);
      _outtext("                                                          ");
      wela(start,finish);
       }
     else {
       time(&finish);
       _setcolor(1);
       _rectangle(_GFILLINTERIOR,124,188,516,244);
       _settextposition(14,19);
       _outtext("                                                          ");
       wela(start,finish);
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#if __MCARD
    while (( thook() != 0 ) || ( ssense(*cidad,NULL,NULL,NULL) != 0 )) {
      }
#endif
    _setcolor(1);
    _rectangle(_GFILLINTERIOR,308,28,636,100);
    _settextposition(4,42);
    _outtext("                                        ");
    _settextposition(5,42);
    _outtext("                                        ");
    *cidad = -1;
    return  0;
  }

/*************************************************************************/
/*       show total time of communication                                */
/*************************************************************************/
 void wela (time_t start,time_t finish)
 {
      double dif_time ;
      int hour,min,sec;
      long rehour;

      if (start==0){
         dif_time = 0.0 ;
        }
      else{
        dif_time = difftime(finish,start);
        }
      hour = (int)dif_time / 3600;
      rehour = (long)dif_time % 3600;
      min = (int)rehour / 60 ;
      sec = (int)rehour % 60 ;
      _setcolor(7);
      _settextcolor(0);
      _dspon(136,208,504,304);
      sprintf(prnbuf,"time start : %s",ctime(&start));
      _settextposition(15,20);
      _outtext(prnbuf);
      sprintf(prnbuf,"time finish : %s",ctime(&finish));
      _settextposition(16,20);
      _outtext(prnbuf);
      sprintf(prnbuf,"total time : %d hour %d min %d sec",hour,min,sec);
      _settextposition(17,20);
      _outtext(prnbuf);
      sprintf(prnbuf,"End of communication please hit any key");
      _settextposition(18,20);
      _outtext(prnbuf);
      getch();
      _setcolor(1);
      _rectangle(_GFILLINTERIOR,132,204,508,308);
      _settextposition(15,20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        _outtext("                                                    ");
        _settextposition(16,20);
        _outtext("                                                    ");
        _settextposition(17,20);
        _outtext("                                                    ");
        _settextposition(18,20);
         outtext("                                                    ");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include    <stdio.h>
#include    <io.h>
#include    <memory.h>
#include    <conio.h>
#include    <string.h>
#include    <direct.h>
#include    <stdlib.h>
#include    <graph.h>
#include    "isdnlib.h"
#include    "isdndef.h"

void     dialing( void );
void     fninp( void );
void     opn_insfile(void);
void     showerr(int,char*);
void     show(int,char*);
void     _dspon(int,int,int,int);

/*************************************************************************/
/*       calling ( circuit switching / data )  and sending file          */
/*************************************************************************/
int sub3( int* cidad )
{
    int err_flg;
    err_flg = 0;
    fninp();
    if((  callid[ 0 ] < 0 ) && ( callid[ 2 ] < 0 )) {
      dialing();
      }
    _setcolor(5);
    _settextcolor(7);
    _dspon(312,32,632,96);
    sprintf(prnbuf,"CIRCUIT SWITCHED CALL");
    _settextposition(4,42);
    _outtext(prnbuf);
    sprintf(prnbuf,"called party number : %s / %s",scadd,scsubadd);
    _settextposition(5,42);
    _outtext(prnbuf);
    #if __MCARD
    timeout_count=60;                          /* timeout_count = 60 sec */
    /*   request originating a circuit switching data call to Mcard   */
    *cidad=ddial(&setup_para,&xret_info,timeout_count);
    #endif
    if (*cidad < 0) {
      switch (*cidad) {
        case (int)0xfb01:
             showerr(54,"line was cut off because tine out,please hit any key");
             _setcolor(1);
             _rectangle(_GFILLINTERIOR,308,28,636,100);
             _settextposition(4,42);
             _outtext("                                          ");
             _settextposition(5,42);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            _outtext("                                                    ");
            err_flg = *cidad;
            return -1;
            break;
       default:
            showerr(56,"error was detected so line was cut off,Please hit any key");
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,308,28,636,100);
            _settextposition(4,42);
            _outtext("                                                    ");
            _settextposition(5,42);
            _outtext("                                                    ");
            err_flg = *cidad;
            return -1;
            break;
       }
     }
   else {
     #if __MCARD
     timeout_count=30;
     /*  request opening Bch data path to Mcard    */
     status=dopenr(*cidad, NULL);
     #endif
     if (status < 0) {
        option=0x2;
        #if __MCARD
         /* request disconecting a circuit switching data call to Mcard */
        status=dclosew(*cidad, option, NULL, timeout_count);
        status=ddisc(*cidad, NULL, NULL);
        #endif
        showerr(52,"disconnected because B ch openning error,Hit any key");
        _setcolor(1);
        _rectangle(_GFILLINTERIOR,308,28,636,100);
        _settextposition(4,42);
        _outtext("                                                    ");
        _settextposition(5,42);
        _outtext("                                                    ");
        err_flg = -1;
        return -1;
       }
     else {
          e=0;
          s=0;
          count=0;
          countr=0;
          show(18,"Now transfer file");
          #if __MCARD
          while (e == 0) {
                memset(c_buff, (char)NULL, 2048);
                for (length=0; length<2048; length++) {
                    if(count==len){
                          e=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
              }
            else {
                k=(uchar)fgetc(fp);
                c_buff[length]=k;
                countr=count+length;
                    if(countr==len) {
                          e=1;
                          break;
                        }
                }
          }
    /*   request sending data to Mcard   */
    status=dsendim(*cidad, length, &c_buff[0], &xret_info);
    sr=++s;
    count=2048*sr;
    status=1;
    while (status == 1) {
       status=dsendcf(*cidad, &xret_info);
     }
    if (status < 0) {
       e=-1;
      }
    }
if (e == 1) {
    memset(&c_buff[0], (char)NULL, 2048);
    strcpy(&c_buff[0], "@END@");
    strcpy(&c_buff[5], &file_name[0]);
    length=strlen(&c_buff[0])+1;
    status=dsend(*cidad, length, &c_buff[0], NULL, 30 );
    if (status < 0) e=-1;
      }
#endif
fclose(fp);
if (e == -1) {
  showerr(54,"during data receiving error was occured, Hit any key");
   _setcolor(1);
   _rectangle(_GFILLINTERIOR,308,28,636,100);
   _settextposition(4,42);
   _outtext("                                          ");
   _settextposition(5,42);
   _outtext("                                          ");
   return -1;
 }
else {
   _setcolor(1);
   _rectangle(_GFILLINTERIOR,220,188,420,244);
   _settextposition(14,31);
   _outtext("                                                  ");
   show(44,"data transfer was finish,Please hit any key");
   getch();
   _setcolor(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            _rectangle(_GFILLINTERIOR,116,188,524,244);
            _settextposition(14,18);
            _outtext("                                                    ");
           }
          option=0x1;
          #if __MCARD
          /*  request closing Bch data path to Mcard  */
          status = dclosew(*cidad, option, NULL, 30);
          /* request disconecting a circuit switching data call to Mcard  */
          status=ddisc(*cidad, NULL, &xret_info);
         #endif
        }
      }
    *cidad = -1;
    _setcolor(1);
    _rectangle(_GFILLINTERIOR,308,28,636,100);
    _settextposition(4,42);
    _outtext("                                          ");
    _settextposition(5,42);
    _outtext("                                          ");
    status=scclose();
    opn_insfile();
    if ( err_flg == 0 ) return  0;
    else
         return -1;
        *cidad = -1;
   }

/*************************************************************************/
/*       called ( circuit switching / data ) and receiving file          */
/*************************************************************************/
int sub4( int* cidad )
{
    int err_flg;
    err_flg = 0;

    _setcolor(5);
    _settextcolor(7);
    _dspon(312,32,632,96);
    sprintf(prnbuf,"CIRCUIT SWITCHED CALL");
    _settextposition(4,42);
    _outtext(prnbuf);
    sprintf(prnbuf,"calling party number :");
    _settextposition(5,42);
    _outtext(prnbuf);
    sprintf(prnbuf,"%s / %s", pgadd,pgsubadd);
    prnbuf[ 24 ] = '¥0';
    _settextposition(5,66);
    _outtext(prnbuf);
    status = dpermit( *cidad,&puserinf,&xret_info );
    if ( status >= 0 ) {
        status=dopenw(*cidad, NULL, 30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
if (status < 0) {
  option=0x2;
  status=dclosew(*cidad, option, NULL, timeout_count);
  /*  request disconecting a circuit switching data call to Mcard  */
  status=ddisc(*cidad, NULL, NULL);
  showerr(52,"disconnected because B ch openning error,Hit any key");
  _setcolor(1);
  _rectangle(_GFILLINTERIOR,308,28,636,100);
  _settextposition(4,42);
  _outtext("                                    ");
  _settextposition(5,42);
  _outtext("                                    ");
  err_flg = -1;
  return -1;
  }
else {
    if((fp=fopen("%instmp%","wb"))==(int)NULL) {
       option=0x2;
       status=dclosew(*cidad, option, NULL, timeout_count);
      /*  request disconecting a circuit switching data call to Mcard  */
       status=ddisc(*cidad, NULL, NULL);
       showerr(56,"error was detected so line was cut off,Please hit any key");
       _setcolor(1);
       _rectangle(_GFILLINTERIOR,308,28,636,100);
       _settextposition(4,42);
       _outtext("                                    ");
       _settextposition(5,42);
       _outtext("                                    ");
       err_flg = -1;
       return -1;
     }
    else {
       show(18,"Now receiving file");
       for( ; ; ) {
          status=1;
          while (status == 1) {
             status=drecim(*cidad, &length, &c_buff[0], NULL);
             if (status < 0) {
                option=0x2;
                status=dclosew(*cidad, option, NULL, timeout_count);
       /*  request disconecting a circuit switching data call to Mcard  */
                status=ddisc(*cidad, NULL, NULL);
                showerr(52,"during data receiving error was occured,Hit any key");
                _setcolor(1);
                _rectangle(_GFILLINTERIOR,308,28,636,100);
                _settextposition(4,42);
                _outtext("                                    ");
                _settextposition(5,42);
                _outtext("                                    ");
                err_flg = -1;
                return -1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
          }
        if (memcmp(&c_buff[0], "@END@", 5) == 0) {
           fclose(fp);
           memcpy(&c_buff[1000], &c_buff[0], 512);
           p=strchr(&c_buff[1005], '.');
             if (p != (char)NULL) {
                 memset(p, (char)NULL, 1);
                }
           strcat(&c_buff[1005], ".bak");
           remove(&c_buff[1005]);
           rename(&c_buff[5],   &c_buff[1005]);
           rename("%instmp%", &c_buff[5]);
           break;
          }
        for (i=0; i<length; i++) {
           (uchar)fputc(c_buff[i], fp);
         }
      }
    option=0x1;
    status=dcloser(*cidad, option, NULL);
    status=1;
    while (status != 0) {
      /*+++ sense the status of Mcard +++*/
      status=ssense(*cidad,NULL,NULL,NULL);
      status=status & 0x3f00;
      }
    _setcolor(1);
    _rectangle(_GFILLINTERIOR,220,188,524,244);
    _settextposition(14,29);                                          ");
    _outtext("
    _setcolor(7);
    _settextcolor(0);
    _dspon(176,208,544,288);
    sprintf(prnbuf,"data receiving is completed");
    _settextposition(15,25);
    _outtext(prnbuf);
    sprintf(prnbuf,"Receving file name is : %s",&c_buff[5]);
    _settextposition(16,25);
    _outtext(prnbuf);
    sprintf(prnbuf,"Please hit any key");
    _settextposition(17,25);
    _outtext(prnbuf);
    getch();
    _setcolor(1);
    _rectangle(_GFILLINTERIOR,172,204,548,292);
    _settextposition(15,25);                                          ");
    _outtext("
    _settextposition(16,25);                                          ");
    _outtext("
    _settextposition(17,25);                                          ");
    _outtext("
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }
    }
   *cidad = -1;
    _setcolor(1);
    _rectangle(_GFILLINTERIOR,308,28,636,100);
    _settextposition(4,42);
    _outtext("                                          ");
    _settextposition(5,42);
    _outtext("                                          ");
    status=scclose();
    opn_insfile();
    if ( err_flg == 0 ) return  0;
    else                return -1;
}

/*************************************************************************/
/*       input file name                                                 */
/*************************************************************************/
void fninp()
{
   _setcolor(7);
   _settextcolor(0);
   _dspon(128,208,496,288);
   sprintf(prnbuf,"Please input file name to transfer");
   _settextposition(15,19);
   _outtext(prnbuf);
   sprintf(prnbuf,"Format of file name :Drive:¥¥file name");
   _settextposition(16,19);
   _outtext(prnbuf);
   _settextposition(17,19);
   _settextcolor(0);
   _settextcursor(0x0f0f); /*cursor on*/
   scanf("%s",file_name);
   _settextcursor(0x2000); /*cursor off*/
   for( ; ; ) {
       if((fp=fopen(file_name,"rb"))!=(int)NULL){
          handle=fileno(fp);
          len=filelength(handle);
           break;
         }
      _setcolor(1);
      _rectangle(_GFILLINTERIOR,124,204,500,292);
      _settextposition(15,19);
      _outtext("                                                ");
      _settextposition(16,19);
      _outtext("                                                ");
      _settextposition(17,19);
      _outtext("                                                ");
      _setcolor(7);
      _settextcolor(0);
      _dspon(128,208,496,288);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    sprintf(prnbuf,"File name not found,input file name again");
    _settextposition(15,19);
    _outtext(prnbuf);
    sprintf(prnbuf,"Format of file name : Drive:¥¥file name ");
    _settextposition(16,19);
    _outtext(prnbuf);
    _settextposition(17,19);
    _settextcolor(0);
    _settextcursor(0x0f0f); /*cursor on*/
    scanf("%s",file_name);
    _settextcursor(0x2000); /*cursor off*/
  }
 _setcolor(1);
 _rectangle(_GFILLINTERIOR,124,204,500,292);
 _settextposition(15,19);
 _outtext("                                                    ");
 _settextposition(16,19);
 _outtext("                                                    ");
 _settextposition(17,19);
 _outtext("                                                    ");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include    <stdio.h>
#include    <io.h>
#include    <memory.h>
#include    <conio.h>
#include    <string.h>
#include    <direct.h>
#include    <graph.h>
#include    <stdlib.h>
#include    "isdnlib.h"
#include    "isdndef.h"

void    dialing( void );
void    fninp( void );
void    opn_insfile(void);
void    showerr(int,char*);
void    show(int,char*);
void    _dspon(int,int,int,int);

/*************************************************************************/
/*       calling ( packet switching )                                    */
/*************************************************************************/
int sub5( int* cidad )
{
        int err_flg;
        err_flg = 0;
        fninp();

        if (( callid[ 0 ] < 0 ) && ( callid[ 1 ] < 0 )) {
              dialing();
            }
           _setcolor(5);
           _settextcolor(7);
           _dspon(392,32,600,80);
           sprintf(prnbuf,"PACKET SWITCHED CALL");
           _settextposition(4,52);
           _outtext(prnbuf);
        /*  request originating a packet switching data call to Mcard  */
        timeout_count = 60;
        *cidad = pdial( &cr_para,&pret_info,timeout_count );
        if( *cidad < 0 ) {
             switch(*cidad) {
                 case (int)0xfb01:
                 showerr(54,"line was cut off because time out,Please hit any key");
                 pdisc(*cidad,&cq_para,&pret_info);
                 _setcolor(1);
                 _rectangle(_GFILLINTERIOR,388,28,604,84);
                 _settextposition(4,52);
                 _outtext("                          ");
                 return -1;
                 break;
              default:
                 showerr(56,"error was detected so line was cut off,Please hit any key");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pdisc(*cidad,&cq_para,&pret_info);
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,388,28,604,84);
            _settextposition(4,52);
            _outtext("                          ");
            return -1;
            break;
          }
        }
      else {
          option = 0x0;
          e=0;
          s=0;
          count=0;
          countr=0;
          show(18,"Now transfer file");
            while (e== 0) {
                 memset(p_buff,(char)NULL,128);
                     for (length=0;length<128;length++)  {
                             if(count ==len){
                                 e=1;
                                 break;
                              }
                             else {
                                 k=(uchar)fgetc(fp);
                                 p_buff[length]=k;
                                 countr=count+length;
                                   if(countr==len) {
                                     e=1;
                                     break;
                                    }
                              }
                      }
               status=1;
               while(status==1){
                 status = psend(*cidad,length,&p_buff[0],option,&pret_info);
                          }
               sr=++sr;
               count=128*sr;
               }
              if( e==1 )  {
                option=0x1;
                memset(&p_buff[0],(char)NULL,256);
                strcpy(&p_buff[0],"@END@");
                strcpy(&p_buff[5],&file_name[0]);
                length = strlen(&p_buff[0])+1;
                option=0x1;
                status=1;
                while(status==1) {
                  status = psend(*cidad,length,&p_buff[0],option,&pret_info);
                }
                status=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            while(status!=0) {
                status = ssense(*cidad,NULL,NULL,NULL);
                status = status & 0x00ff;
              }
            if(status<0)   e=-1;
           }
          _setcolor(1);
          _rectangle(_GFILLINTERIOR,220,188,420,244);
          _settextposition(14,31);
          _outtext("                                                    ");
          fclose(fp);
          if(e==-1) {
            showerr(52,"during data transfer error was occured, Hit any key");
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,388,28,604,84);
            _settextposition(4,52);
            _outtext("                              ");
            return -1;
              }
            show(44,"data transfer was finish,Please hit any key");
            getch();
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,116,188,524,244);
            _settextposition(14,18);
            _outtext("                                                  ");

          }
         _setcolor(1);
         _rectangle(_GFILLINTERIOR,388,28,604,84);
         _settextposition(4,52);
         _outtext("                              ");
         *cidad = -1;
         status=scclose();
         opn_insfile();
         if(err_flg==0)
           return 0;
         else
           return -1;
      }

/*************************************************************************/
/*       called ( packet switching )                                     */
/*************************************************************************/
int sub6( int* cidad )
{
        int err_flg;
        err_flg = 0;

        _setcolor(5);
        _settextcolor(7);
        _dspon(392,32,600,80);
        sprintf(prnbuf,"PACKET SWITCHED CALL");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_settextposition(4,52);
_outtext(prnbuf);
*cidad=paccept(&ca_para,&pret_info,timeout_count);
if((fp=fopen("%instmp%","wb"))==(int)NULL)  {
   showerr(56,"error was detected so line was cut off,Please hit any key");
   pdisc(*cidad,&cq_para,&pret_info);
   _setcolor(1);
   _rectangle(_GFILLINTERIOR,388,28,604,84);
   _settextposition(4,52);
   _outtext("                          ");
   return -1;
   }
else  {
show(18,"Now receiving file");
status=0;
while(status==0) {
      kind[0]=0x0;
      status=1;
      while(status==1){
      status = precim(*cidad,&kind[0],&length,&p_buff[0],&pret_info);
                  }
      if ( status < 0) {
         pdisc(*cidad,NULL,NULL);
         showerr(52,"during data receiving error was occured, Hit any key");
         _setcolor(1);
         _rectangle(_GFILLINTERIOR,388,28,604,84);
         _settextposition(4,52);
         _outtext("                          ");
         return -1;
          }
        if (memcmp(&p_buff[0],"@END@",5) == 0) {
          fclose(fp);
          memcpy(&p_buff[100],&p_buff[0],251);
          p = strchr(&p_buff[105],'.');
          if (p!=(char) NULL) {
               memset(p,(char)NULL,1);
              }
          strcat(&p_buff[105],".bak");
          remove(&p_buff[105]);
          rename(&p_buff[5],&p_buff[105]);
          rename("%instmp%",&p_buff[5]);
          break;
           }
        for (i = 0;i <length;i++) {
           (uchar)fputc(p_buff[i],fp);
           }
       }
       cq_para.puserinf = NULL;
       cq_para.pdiag  = &pdiag;
       puserinf.length = 2;
       puserinf.protid = 0;
       puserinf.userinfo[0] = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pret_info.pcadd = pcadd;
            pret_info.pcsubadd = pcsubadd;
            pret_info.pgadd = pgadd;
            pret_info.pgsubadd = pgsubadd;
            pret_info.puserinf = &puserinf;
            status= pdisc(*cidad,&cq_para,&pret_info);
            status=1;
            while (status!=0) {
                  status = ssense(*cidad,NULL,NULL,NULL);
                  status = status & 0xff;
               }
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,220,188,524,244);
            _settextposition(14,29);
            _outtext("                                                    ");
            _setcolor(7);
            _settextcolor(0);
            _dspon(176,208,544,288);
            sprintf(prnbuf,"data receiving is completed");
            _settextposition(15,25);
            _outtext(prnbuf);
            sprintf(prnbuf,"Receving file name is : %s",&p_buff[5]);
            _settextposition(16,25);
            _outtext(prnbuf);
            sprintf(prnbuf,"Please hit any key");
            _settextposition(17,25);
            _outtext(prnbuf);
            getch();
            _setcolor(1);
            _rectangle(_GFILLINTERIOR,172,204,548,292);
            _settextposition(15,25);
            _outtext("                                                    ");
            _settextposition(16,25);
            _outtext("                                                    ");
            _settextposition(17,25);
            _outtext("                                                    ");
          }
         _setcolor(1);
         _rectangle(_GFILLINTERIOR,388,28,604,84);
         _settextposition(4,52);
         _outtext("                          ");
        *cidad = -1;
        status=scclose();
        opn_insfile();
        if(err_flg==0)
           return 0;
        else
           return -1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define END_OF_FILE 0x1a

/************************/
/* Key assign for PC98 */
/************************/
#define    K_ESC          0x1b        /* ESC          */
#define    K_F1           0x53        /* F・1         */
#define    K_F2           0x54        /* F・2         */
#define    K_F3           0x55        /* F・3         */
#define    K_F4           0x56        /* F・4         */
#define    K_F5           0x57        /* F・5         */
#define    K_F6           0x45        /* F・6         */
#define    K_F7           0x4a        /* F・7         */
#define    K_F8           0x50        /* F・8         */
#define    K_F9           0x51        /* F・9         */
#define    K_F10          0x5a        /* F・10        */
#define    K_CR           0x0d        /* Return Key   */
#define    K_INS          0x50        /* INS Key      */
#define    K_DEL          0x44        /* DEL Key      */
#define    K_UP           0x0b        /* ↑ Key        */
#define    K_DOWN         0x0a        /* ↓ Key        */
#define    K_LEFT         0x08        /* ← Key        */
#define    K_RIGHT        0x0c        /* → Key        */
#define    K_SPC          0x20        /* SPACE Key    */

typedef    unsigned char uchar;
/*****************************************************************************/
/* fuctions for M-card                                                       */
/*****************************************************************************/

/*****************************************************************************/
/* callid = tdial( setup_para, xret_info, time_count )                       */
/*  [function]                                                               */
/*   To request originating a speech call to M-card, and                     */
/*     be waiting for the result returned from M-card.                       */
/*   This function is not completed until Mcard receives a CONNECT message   */
/*     or the time-out count is completed.                                   */
/*   When Mcard receives this request, it sends a SETUP message to the       */
/*     network.                                                              */
/*  [input]                                                                  */
/*   setup_para    : pointers of sending information elements to be included */
/*                   in SETUP message                          -> M-card     */
/*   xret_info     : pointers of return data                   <- M-card     */
/*   timeout_count : timeout time ( seconds )                               */
/*  [output]                                                                 */
/*    callid : 0 - 7 : call reference number                                 */
/*             > 0   : originating call is completed normally and call is    */
/*                     connected                                             */
/*           : < 0   : error                                                 */
/*****************************************************************************/
int  tdial( XXSETUP*, XXRINFO*, int );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************************/
/* callid = taccept(user_info, xret_info, timeout_count )                */
/*  [function]                                                           */
/*   To be waiting for an incoming speech call.                          */
/*   This function is not completed until Mcard receives an incoming speech */
/*     call or the time-out count is completed.                          */
/*   If Mcard receives a incoming speech call before time-count is completed,*/
/*     it sends a CONNECT message to the network.                        */
/*  [input]                                                              */
/*    user_info     : pointers of user-user information element   <- M-card */
/*    xret_info     : pointers of return data                     <- M-card */
/*    timeout_count : timeout time ( seconds )                           */
/*  [output]                                                             */
/*    callid : 0 - 7 : call reference number                             */
/*           : >=0   : call is connected                                 */
/*           : < 0   : error or no incoming call                         */
/*************************************************************************/
int  taccept( XUINFO*, XXRINFO*, int );


/*************************************************************************/
/* callid = tacptck( xret_info )                                         */
/*  [function]                                                           */
/*   To check an incoming speech call.                                   */
/*   This function only checks an incoming speech call and doesn't wait. */
/*  [input]                                                              */
/*    xret_info     : pointers of return data                     <- M-card */
/*  [output]                                                             */
/*    callid : 0 - 7 : call reference number                             */
/*           : >=0   : Mcard receives a incommnig call                   */
/*           : < 0   : error or no inconing call                         */
/*************************************************************************/
int  tacptck( XXRINFO* );


/*************************************************************************/
/* status = trefuse ( callid, user_info )                                */
/*  [function]                                                           */
/*   To request refusing an incoming speech call to Mcard.               */
/*   When Mcard receives this request before the call is connected, it sends */
/*     a DISCONNECT message to the network.                              */
/*  [input]                                                              */
/*    callid    : call reference number                        -> M-card */
/*    user_info : pointers of received user-user information           */
/*                                                  element   <- M-card */
/*  [output]                                                             */
/*    status : >= 0 : call refusal is completed normally                 */
/*             < 0  : error                                              */
/*************************************************************************/
int  trefuse( int, XUINFO* );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = tpermit ( callid, user_info, xret_info )                         */
/*  [function]                                                               */
/*   To request acceptance of an comming speech call to Mcard.               */
/*   This function is available after "tacptck" function is executed.        */
/*   When Mcard receives this request, it sends a CONNECT message to the     */
/*     network.                                                              */
/*  [input]                                                                  */
/*   callid    : call reference number                        -> M-card      */
/*   user_info : pointers of sending user-user information                   */
/*                                                  elenent   -> M-card      */
/*   xret_info : pointers of return data                      <- M-card      */
/*  [output]                                                                 */
/*   status : >= 0 : the call is connected normally                          */
/*            <    : error ( the call is not connected )                     */
/*****************************************************************************/
int  tpermit( int, XUINFO*, XXRINFO* );


/*****************************************************************************/
/* status = tdisc ( callid, user_info, xret_info )                           */
/*  [function]                                                               */
/*   To request disconnecting a speech call to Mcard.                        */
/*   When Mcard receives this request, it sends a DISCONNECT messageto the   */
/*     network.                                                              */
/*  [input]                                                                  */
/*   callid    : call reference number                        -> M-card      */
/*   user_info : pointers of received user-user information                  */
/*                                                  element   <- M-card      */
/*   xret_info : pointers of return data                      <- M-card      */
/*  [output]                                                                 */
/*   status : >= 0 : the call is disconnected normally                       */
/*            <    : error                                                   */
/*****************************************************************************/
int  tdisc( int, XUINFO*, XXRINFO* );


/*****************************************************************************/
/* status = thook( )                                                         */
/*  [function]                                                               */
/*   To check the state of hook switch.                                      */
/*  [output]                                                                 */
/*   status : 0 : Hook on                                                    */
/*            1 : Hook off                                                   */
/*****************************************************************************/
int  thook( void );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = tpb( number )                                                    */
/*  [function]                                                               */
/*   To request sending multi-frequency tone to Mcard.                       */
/*  [input]                                                                  */
/*   number : '0','1','2','3','4','5','6','7','8','9'.'#','*'               */
/*  [output]                                                                 */
/*   status : 0 : completed normally                                         */
/*          > 0 : error                                                      */
/*****************************************************************************/
int  tpb( char );

/*****************************************************************************/
/* callid = ddial( setup_para, xret_info, timeout_count )                    */
/*  [function]                                                               */
/*   To request originating a data call and be waiting for the result        */
/*     returned from Mcard.                                                  */
/*   This function is not completed until Mcard receives a CONNECT message  */
/*     or the time-out count is completed.                                   */
/*   When Mcard receives this request, it sends a SETUP message to the       */
/*     network.                                                              */
/*  [input]                                                                  */
/*   setup_para    : pointers of sending information elements to be included */
/*                   in SETUP message                          -> M-card     */
/*   xret_info     : pointers of return data                   -> M-card     */
/*   timeout_count : timeout time ( seconds )                               */
/*  [output]                                                                 */
/*    callid : 0 - 7 : call reference number                                 */
/*           : >=0   : originating call is completed normally and call is    */
/*                     connected                                            */
/*           : < 0   : error                                                */
/*****************************************************************************/
int  ddial( XXSETUP*, XXRINFO*, int );

/*****************************************************************************/
/* callid = daccept( user_info, xret_info, timeout_count )                   */
/*  [function]                                                               */
/*   To be waiting for an incoming data call.                                */
/*   This finction is not completed until Mcard receive an incoming call    */
/*     or the time count is counted.                                         */
/*   If Mcard receives a incoming data call before time-count is completed,  */
/*     it sends a CONNECT message to the network.                            */
/*  [input]                                                                  */
/*    user_info     : pointers of user-user information element   <- M-card  */
/*    xret_info     : pointers of return data                     <- M-card  */
/*    timeout_count : timeout time ( seconds )                               */
/*  [output]                                                                 */
/*    callid : 0 - 7 : call reference number                                 */
/*           : >=0   : call is connected                                     */
/*           : < 0   : error or no incoming call                             */
/*****************************************************************************/
int  daccept( XUINFO*, XXRINFO*, int );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* callid = dacptck( xret_info )                                             */
/*  [function]                                                               */
/*   To check an incoming data call.                                         */
/*   This function only checks an incoming data call and doesn't wait.       */
/*  [input]                                                                  */
/*   xret_info     : pointers of return data                    <- M-card    */
/*  [output]                                                                 */
/*    callid : 0 - 7 : call reference number                                 */
/*           : >=0   : Mcard receives a inconmnig call                       */
/*           : < 0   : error or no incoming call                             */
/*****************************************************************************/
int  dacptck( XXRINFO* );


/*****************************************************************************/
/* status = drefuse( callid, user_info )                                     */
/*  [function]                                                               */
/*   To request refusing an incoming data call to Mcard.                     */
/*   When Mcard receives this request before the call is connected, it sends */
/*     a DISCONNECT message to the network.                                  */
/*  [input]                                                                  */
/*    callid    : call reference number                        -> M-card    */
/*    user_info : pointers of received user-user information               */
/*                                                    element   <- M-card    */
/*  [output]                                                                 */
/*    status : >= 0 : call refusal is completed normally                     */
/*             < 0  : error                                                  */
/*****************************************************************************/
int  drefuse( int, XUINFO* );


/*****************************************************************************/
/* status = dpermit( callid, user_info, xret_info )                          */
/*  [function]                                                               */
/*   To request accepting an incoming data call to Mcard.                    */
/*   This function is available after "dacptck" function is executed.        */
/*   When Mcard receives this request, it sends a CONNECT message to         */
/*     the network.                                                          */
/*  [input]                                                                  */
/*   callid    : call reference number                        -> M-card    */
/*   user_info : pointers of sending user-user information                 */
/*                                                    element   -> M-card    */
/*   xret_info : pointers of return data                      <- M-card    */
/*  [output]                                                                 */
/*   status : >= 0 : the call is connected normally                         */
/*            <    : error ( the call is not connected )                    */
/*****************************************************************************/
int  dpermit( int, XUINFO*, XXRINFO* );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = ddisc ( callid, user_info, xret_info )                           */
/*  [function]                                                               */
/*   To request disconnecting a call data to Mcard.                          */
/*   When Mcard receives this request, it sends DISCONNECT message to        */
/*     the network.                                                          */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   user_info : pointers of sending user-user information                   */
/*                                                  element    -> M-card    */
/*   xret_info : pointers of return data                        <- M-card    */
/*  [output]                                                                 */
/*   status : >= 0 : the call is disconnected normally                       */
/*            <    : error                                                   */
/*****************************************************************************/
int  ddisc( int, XUINFO*, XXRINFO* );


/*****************************************************************************/
/*  status = dopenr( callid, xret_info )                                     */
/*  [function]                                                               */
/*   To request opening a B_channel for data transmissin to Mcard.           */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   xret_info : pointers of return data                        <- M-card    */
/*  [output]                                                                 */
/*   status : >= 0 : completed normally                                      */
/*            <    : error                                                   */
/*****************************************************************************/
int  dopenr( int, XXRINFO* );


/*****************************************************************************/
/* status = dopenw( callid, xret_info, timeout_count )                       */
/*  [function]                                                               */
/*   To request opening a B_channel for data transmissin to Mcard and        */
/*     be waiting for completion of openning.                                */
/*   To request opening a B_channel for data transmissin to Mcard and        */
/*     be waiting for completion of openning.                                */
/*   This function is not completed until openning Bch is completed          */
/*     or the time-out count is completed.                                   */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   xret_info : pointers of return data                        <- M-card    */
/*   timeout_count : timeout time ( seconds )                                */
/*  [output]                                                                 */
/*   status : >= 0 : completed normally                                      */
/*            <    : error                                                   */
/*****************************************************************************/
int  dopenw( int, XXRINFO*, int );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = dcloser( callid, option, xret_info )                             */
/*  [function]                                                               */
/*   To request closing a B channel for data transmission to Mcard.          */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   option    : 0 : link close                                 -> M-card    */
/*             : 1 : line close                                              */
/*   xret_info : pointers of return data                        <- M-card    */
/*  [output]                                                                 */
/*   status :   0 : link close is completed                                  */
/*              1 : line close is completed                                  */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  dcloser( int, char, XXRINFO* );


/*****************************************************************************/
/* status = dclosew( callid, option, xret_info, timeout_count )              */
/*  [function]                                                               */
/*   To request closing a B channel for sending data to Mcard.               */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   option    : 0 : link close                                 -> M-card    */
/*             : 1 : line close                                              */
/*   xret_info : pointers of return data                        <- M-card    */
/*   timeout_count : timeout time ( seconds )                                */
/*  [output]                                                                 */
/*   status :   0 : link close is completed                                  */
/*              1 : line close is completed                                  */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  dclosew( int, char, XXRINFO*, int );


/*****************************************************************************/
/* status = dsend( callid, length, c_buffer, xret_info, timeout_count )      */
/*  [function]                                                               */
/*   To send the data and be waiting until data sending is completed or      */
/*     time_out count is completed.                                          */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   length    : data length                                    -> M-card    */
/*   c_buffer  : pointer of sending data                        -> M-card    */
/*   xret_info     : pointers of return data                    <- M-card    */
/*   timeout_count : timeout time ( seconds )                                */
/*  [output]                                                                 */
/*   status : >= 0 : completed                                               */
/*            < 0  : error                                                   */
/*****************************************************************************/
int  dsend( int, int, uchar*, XXRINFO*, int );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = dsendin( callid, length, c_buffer, xret_info )                   */
/*  [function]                                                               */
/*   To send the data.                                                       */
/*   This function doesn't wait for completion of data sending.              */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   length    : data length                                    -> M-card    */
/*   c_buffer  : pointer of sending data                        -> M-card    */
/*   xret_info     : pointers of return data                    <- M-card    */
/*  [output]                                                                 */
/*   status : = 0 : data send request is completed                           */
/*   status : = 1 : data send request is not completed because of busy       */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  dsendim( int, int, uchar*, XXRINFO* );


/*****************************************************************************/
/* status = dsendcf( callid, xret_info )                                     */
/*  [function]                                                               */
/*   To check whether data sending is completed or not.                      */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   xret_info     : pointers of return data                    <- M-card    */
/*  [output]                                                                 */
/*   status :   0 : completed                                                */
/*              1 : waiting                                                  */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  dsendcf( int, XXRINFO* );


/*****************************************************************************/
/* status = drec( callid, &length, c_buffer, xret_info, timeout_count )      */
/*  [function]                                                               */
/*   To check whether Mcard receives data or not.                            */
/*   This function is waiting until any data is received or time_out count   */
/*     is completed.                                                         */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   &length   : pointer of received data length                <- M-card    */
/*               length <= 1024                                              */
/*   c_buffer  : pointer of received data                       <- M-card    */
/*   xret_info     : pointers of return data                    <- M-card    */
/*   timeout_count : timeout time ( seconds )                               */
/*  [output]                                                                 */
/*    status   : >= 0 : receive data                                         */
/*               <  0 : error                                                */
/*****************************************************************************/
int  drec( int, int*, char*, XXRINFO*, int );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* drecim( callid, &length, c_buffer, xret_info )                            */
/*  [function]                                                               */
/*   To check whether Mcard receives data or not.                            */
/*   This function doesn't waite data receiving.                             */
/*  [input]                                                                  */
/*   callid    : call reference number                          -> M-card    */
/*   &length   : pointer of received data length               <- M-card    */
/*               length <= 1024                                              */
/*   c_buffer  : pointer of received data                       <- M-card    */
/*   xret_info     : pointers of return data                    <- M-card    */
/*  [output]                                                                 */
/*    status   :  0 : receive data                                           */
/*                1 : no received data                                       */
/*              < 0 : error                                                  */
/*****************************************************************************/
int  drecim( int, int*, char*, XXRINFO* );

/*****************************************************************************/
/* callid = pdial( cr_para, pret_info, timeout_count )                       */
/*  [function]                                                               */
/*   To request originating a packet call to M-card,  and                    */
/*      be waiting for the result returned from M-card.                      */
/*   This function is not completed until Dch packet path is establised      */
/*     or the time-out count is completed.                                   */
/*   When Mcard receives this request, it sends a CR packet to the network.  */
/*  [input]                                                                  */
/*   cr_para       : pointers of the parameters of a CR packet to be send    */
/*                                                          -> M-card        */
/*                                                          <- M-card        */
/*   pret_info     : pointers of return data                                 */
/*   timeout_count : timeout time ( seconds )                                */
/*  [output]                                                                 */
/*    callid : 0 - 7 : logical channel number of Dch packet                  */
/*           : >=0   : Dch packet path is connected                          */
/*           :  < 0  : error                                                 */
/*****************************************************************************/
int  pdial( XPCRPARA*, XPRINFO*, int );

/*****************************************************************************/
/* callid = paccept( ca_para, pret_info, timeout_count )                     */
/*  [function]                                                               */
/*   To be waiting for an incoming packet call.                              */
/*   This function is not completed until Mcard receives an incoming packet  */
/*     call or the time count is counted.                                    */
/*  [input]                                                                  */
/*   ca_para       : pointers of the parameters of a CA packet to be send    */
/*                                                          -> M-card        */
/*                                                          <- M-card        */
/*   pret_info     : pointers of return data                                 */
/*   timeout_count : timeout time ( seconds )                                */
/*  [output]                                                                 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*    callid : 0 - 7 : logical channel number of Dch packet                */
/*           : >=0   : Dch packet path is connected                        */
/*           : < 0   : error                                               */
/*********************************************************************/
int  paccept( XPCAPARA*, XPRINFO*, int );

/*********************************************************************/
/* pacptck( pret_info )                                                  */
/*  [function]                                                           */
/*   To check an incoming packet call                                    */
/*   This function only check an incoming call and doesn't wait.         */
/*  [input]                                                              */
/*   pret_info     : pointers of return data                  <- M-card  */
/*  [output]                                                             */
/*    callid : 0 - 7 : logical channel number of Dch packet              */
/*           :  >=0  : CN packet is received                             */
/*           :  < 0  : error                                             */
/*********************************************************************/
int  pacptck( XPRINFO* );

/*********************************************************************/
/* status = prefuse( callid, cq_para, pret_info )                        */
/*  [function]                                                           */
/*   To request refusing an incoming packet call to Mcard.               */
/*   When Mcard receives this request before "ppermit" function is not   */
/*     executed, it sends a disconnect message to the network            */
/*  [input]                                                              */
/*   callid    : logical channel number                      -> M-card  */
/*   cq_para   : pointers of the parameters of a CQ packet to be send   */
/*                                                           -> M-card  */
/*   pret_info : pointers of return data                  <- M-card     */
/*  [output]                                                             */
/*   status :  0 : Dch packet path is disconected normally               */
/*          < 0 : error                                                  */
/*********************************************************************/
int  prefuse( int, XPCQPARA*, XPRINFO* );

/*********************************************************************/
/* status = ppermit( callid, ca_para )                                   */
/*  [function]                                                           */
/*   To request according an comming call to Mcard.                      */
/*   This function is available after "tacptck" function executed.       */
/*   When Mcard receives this request, Mcard sends CONNECT message to the */
/*     network.                                                          */
/*  [input]                                                              */
/*   callid    : logical channel number                      -> M-card  */
/*   ca_para   : pointers of return data                     -> M-card  */
/*  [output]                                                             */
/*   status :  0 : Dch packet path is connected normally                 */
/*          < 0 : error                                                  */
/*********************************************************************/
int  ppermit( int, XPCAPARA* );
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* pdisc( callid, cq_para, pret_info )                                       */
/*  [function]                                                               */
/*   To request disconnecting a call to Mcard.                               */
/*   When Mcard receives this request, it sends a CQ packet the network.     */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card     */
/*   cq_para   : pointers of sending information elements                    */
/*                                  included in CQ packet <- M-card          */
/*   pret_info : pointers of return data                      <- M-card     */
/*  [output]                                                                 */
/*   status :  0 : Dch packet path is disconected normally                   */
/*           < 0 : error                                                     */
/*****************************************************************************/
int  pdisc( int, XPCQPARA*, XPRINFO* );


/*****************************************************************************/
/* status = psbuffck( callid, pret_info )                                    */
/*  [function]                                                               */
/*   To check the packet sending buffer.                                     */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card     */
/*   pret_info : pointers of return data                       <- M-card     */
/*  [output]                                                                 */
/*   status : > 0 : sending buffer is not full                               */
/*              0 : all the sending buffer is busy                           */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  psbuffck( int, XPRINFO* );


/*****************************************************************************/
/* status = psend( callid, length, p_buffer, option, pret_info )             */
/*  [function]                                                               */
/*   To send the data through the Dch packet channel and don't wait for      */
/*     completion of data sending.                                           */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card     */
/*   length    : data length                                   -> M-card     */
/*   p_buffer  : pointer of sending buffer                     -> M-card     */
/*   option    : 00H : there is more data to be sent           -> M-card     */
/*               01H : no more data                                          */
/*   pret_info : pointers of return data                       <- M-card     */
/*  [output]                                                                 */
/*   status :   0 : completed normally                                       */
/*              1 : sending buffer is busy                                   */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  psend( int, int, char*, char, XPRINFO* );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = prec( callid, &kind, &length, p_buffer, pret_info, timeout_count)*/
/*  [function]                                                               */
/*   To be waiting for reception of a DT packet or a IT packet.              */
/*   This function is waiting until Mcard receives DT packet or IT packet    */
/*     or time_out count is completed.                                       */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card    */
/*   &kind     : pointer of received packet type               <- M-card    */
/*        kind = 00H : DT packet                                            */
/*               01H : IT packet                                            */
/*   &length   : pointer of received packet length            <- M-card    */
/*   p_buffer  : pointer of receiving buffer                   <- M-card    */
/*   pret_info : pointer of return data                        <- M-card    */
/*   timeout_count : timeout time ( seconds )                               */
/*  [output]                                                                 */
/*   status :   0 : completed normally                                       */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  prec( int, char*, int*, char*, XPRINFO*, int );

/*****************************************************************************/
/* status = precim( callid, &kind, &length, p_buffer, pret_info )            */
/*  [function]                                                               */
/*   To check the received D channel packet data.                            */
/*   This function dosen't waite for receiving DT packet and IT packet.      */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card    */
/*   &kind     : pointer of received packet type               <- M-card    */
/*        kind = 00H : DT packet                                            */
/*               01H : IT packet                                            */
/*   &length   : pointer of received packet length            <- M-card    */
/*   p_buffer  : pointer of receiving buffer                   <- M-card    */
/*   pret_info : pointer of return data                        <- M-card    */
/*  [output]                                                                 */
/*   status :   0 : receive data                                             */
/*              1 : no received data                                         */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  precim( int, char*, int*, char*, XPRINFO* );

/*****************************************************************************/
/* status = psendint( callid, length, p_buffer, pret_info, timeout_count )   */
/*  [function]                                                               */
/*   To send a IT packet.                                                    */
/*   This function is waiting until IT packet sending is completed or        */
/*     time_out count is completed.                                          */
/*  [input]                                                                  */
/*   callid    : logical channel number                        -> M-card    */
/*   length    : sending packet length ( *lengrh < 32 )        -> M-card    */
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*   p_buffer  : pointer of sending packet data                    -> M-card  */
/*   xret_info : pointer of received information elements          <- M-card  */
/*   timeout_count : timeout time ( seconds )                                 */
/*  [output]                                                                  */
/*   status :   0 : completed normally                                        */
/*            < 0 : error                                                     */
/*****************************************************************************/
int  psendint( int, int, char*, XPRINFO*, int );


/*****************************************************************************/
/* status = scopen( ins_file )                                               */
/*  [function]                                                               */
/*   To read the system data from the system and send it to Mcard.           */
/*   After this function is executed, PC can access to Mcard.                */
/*   At the start of communication, this function must be executed.          */
/*  [input]                                                                  */
/*   ins_file : pointer of the system file name                              */
/*  [output]                                                                 */
/*   status : >= 0 : completed normally                                      */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  scopen( char* );


/*****************************************************************************/
/* status = scclose( )                                                       */
/*  [function]                                                               */
/*   To deactivate Mcard.                                                    */
/*   If this function is executed, PC can't access to Mcard,  until "scopen" */
/*     function is executed again.                                           */
/*  [output]                                                                 */
/*   status : >= 0 : completed normally                                      */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  scclose( void );


/*****************************************************************************/
/* status = sreset( )                                                        */
/*  [function]                                                               */
/*   To reset Mcard                                                          */
/*   If this function is executed, PC can't access to Mcard,  until "scopen" */
/*     function is executed again.                                           */
/*  [output]                                                                 */
/*   status : >= 0 : completed normally                                      */
/*            < 0 : error                                                    */
/*****************************************************************************/
int  sreset( void );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/* status = ssense( callid, class, xret_info, pret_info )                    */
/*  [function]                                                               */
/*   To check the call status of Mcard and the the status of data            */
/*     transmission of Mcard                                                 */
/*  [input]                                                                  */
/*   callid    : call reference number or logical channel number  -> M-card  */
/*   class     : pointer of class                                 <- M-card  */
/*      *class = 00H : undesingnated                                         */
/*               01H : telephone ( speech or 3.1kHz audio )                  */
/*               02H : unrestricted digital                                  */
/*               03H : Dch packet                                            */
/*   xret_info : pointer of received information elements       <- M-card  */
/*               of circuit switching                                        */
/*   pret_info : pointer of received information elements       <- M-card  */
/*               of packet  switching                                        */
/*  [output]                                                                 */
/*   status :                                                                */
/*     status & 0x40 = 0 : Mcard is connected to line                        */
/*                     1 : Mcard is not connected to line                    */
/*     status & 0x3f : call status  ( circuit switching )                    */
/*                   = 0 : NULL                                              */
/*                   = 1 : outgoing cal                                      */
/*                   = 3 : outgoing call accept                              */
/*                   = 4 : call delivered                                    */
/*                   = 7 : alerting                                          */
/*                   =10 : active                                            */
/*                   =11 : disconnect request                                */
/*                   =19 : release request                                   */
/*                   : Rno ( packet switching )                              */
/*     status & 0xff : NULL ( circuit switching, speech )                    */
/*                     LAPB status number ( circuit switching, data )        */
/*                     Pno ( packet switching )                              */
/*****************************************************************************/
int  ssense( int, char*, XXRINFO*, XPRINFO* );


char    ins_file[1024];      /* character string for scopen        */
uchar   c_buff[2048];        /* data buffer for circuit switching */
uchar   p_buff[256];         /* data buffer for packet switching  */
char    prnbuf[ 80 ];        /* print out buffer                   */
char    smsbuf[ 80 ];        /* sending message buffer             */
char    rmsbuf[ 80 ];        /* receiving message buffer           */
char    prnmes[ 80 ];        /* message buffer to be printed out   */

int     callid[ 4 ];         /* call reference number              */
                             /* callid[ 0 ] : speech               */
                             /* callid[ 1 ] : C.S. data            */
                             /* callid[ 2 ] : packet               */
                             /* callid[ 3 ] :                      */
int     length;              /* data length                        */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*      return data for circuit switching and packet switching      */
char    pcadd[33];              /* called address                   */
char    pcsubadd[21];           /* called subaddress                */
char    pgadd[33];              /* calling address                  */
char    pgsubadd[21];           /* calling subaddress               */
XUINFO  puserinf;               /* user_user information            */

/*      return data for circut switching                            */
XXRINFO xret_info;
char    option;                 /* option of Bch function           */
XCAUSE  pcause;
XCHARGE pcharge;
char    pchannel;

/*      return data for packet switching                            */
XPRINFO pret_info;
char    kind[1];                /* type                             */

/*      SETUP data for circuit switching and packet switching       */
char    scadd[33];              /* called-party's address           */
char    scsubadd[21];           /* called party's subaddress        */
char    sgadd[33];              /* own address                      */
char    sgsubadd[21];           /* own aubaddress                   */

/*      SETUP data for circuit switching                            */
XXSETUP setup_para;
XKEYPAD setkeypad;
XUINFO  setuinf;

/*      SETUP  data for packet swiching                             */
XPCRPARA cr_para;               /* CR                               */
XPCAPARA ca_para;               /* CA                               */
XPCQPARA cq_para;               /* CQ                               */
int      ppktsize[2];           /* packet size                      */
int      pwndsize[2];           /* window size                      */
char     pthrough;              /* through put class                */
char     pdiag;

int    status;
int    timeout_count;
int    i, e;
char   file_name[41];
char   *p;
char   *path;
char   wk[256];

FILE   *fp, *fopen();
long int  len, count,countr;
int    handle,s,sr;
uchar k;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define __MCARD    1

/*.$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ */
/*                                                                          */
/*        M-card64  Libr. header                                            */
/*                                                                          */
/* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ */

typedef    struct  XKEYPAD
           {
               int  length;              /* data length                    */
               char kpadinfo[32];        /* key pad information element    */
           } XKEYPAD;


typedef    struct  XUINFO
           {
               int  length;              /* dat length                     */
               char protid;
               char userinfo[129];       /* user to user information       */
           } XUINFO;


typedef    struct  XCAUSE
           {
               int  length;              /* data length                    */
               char cause[2];            /* location of cause              */
           } XCAUSE;


typedef    struct  XCHARGE
           {
               int  length;              /* data length                    */
               long intcharg;            /* charging                       */
               char fracharg;            /* charging                       */
           } XCHARGE;


/*****************************************/
/* call originating parameter            */
/*****************************************/

typedef    struct  XXSETUP
           {
               char     far *pcadd;      /* code number destination address      */
               char     far *pcsubadd;   /* code number destination sub address  */
               char     far *pgadd;      /* originating address                  */
               char     far *pgsubadd;   /* originating sub address              */
               XKEYPAD  far *pkeypad;    /* key pad information element          */
               XUINFO   far *puserinf;   /* user to user information             */
           } XXSETUP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************************************/
/* return status of code control for ciruit switching    */
/*********************************************************/

typedef     struct XXRINFO
            {
                char     far *pcadd;     /* code number destination address      */
                char     far *pcsubadd;  /* code number destination sub address  */
                char     far *pgadd;     /* originating address                  */
                char     far *pgsubadd;  /* originating sub address              */
                XUINFO   far *puserinf;  /* user to user information             */
                XCAUSE   far *pcause;    /* cause indication                     */
                XCHARGE  far *pcharge;   /* charging                             */
                char far far *pchannel;  /* B channel number used                */
            } XXRINFO;


/*********************************************************/
/* return status of code control for packet switching    */
/*********************************************************/

typedef     struct XPRINFO
            {
                char     kind;           /* type of call
                                            */
                char     far *pcadd;     /* pointer of called party number
                                            (address)    */
                char     far *pcsubadd;  /* pointer of called party number
                                            (sub address) */
                char     far *pgadd;     /* pointer of calling party number
                                            (address)    */
                char     far *pgsubadd;  /* pointer of calling party number
                                            (sub address) */
                XUINFO   far *puserinf;  /* user to user information
                                            */
                int      pktsize[2];     /* packet size
                                            */
                int      wndsize[2];     /* window size
                                            */
                char     through;        /* throughput class
                                            */
                char     dccause;        /* cause of disconnect
                                            */
                char     rscause;        /* cause of restart
                                            */
                char     diag;           /* diagnostic code
                                            */
                char     kakin;          /* charging
                                            */
            } XPRINFO;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
typedef     struct XPCRPARA
            {
                char      far *pcadd;
                char      far *pcsubadd;
                char      far *pgadd;
                char      far *pgsubadd;
                XUINFO    far *puserinf;
                int       far *ppktsize;
                int       far *pwndsize;
                char      far *pthrough;
                char      rsvcharge;
            } XPCRPARA;

typedef     struct XPCAPARA
            {
                XUINFO    far *puserinf;
                int       far *ppktsize;
                int       far *pwndsize;
                char      far *pthrough;
            } XPCAPARA;

typedef     struct XPCQPARA
            {
                char      cause;
                char      far *pdiag;
                XUINFO    far *puserinf;
            } XPCQPARA;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] William Stallings, "ISDN An Introduction" : MACMILLAN Publishing Company , 1989.

[2] Gray C. Kessler, "ISDN Concepts, Facilites and Services" : McGraw-Hill , 1990.

[3] Uyless Black, "OSI a Model for Computer Communication Standards": Prentice-Hall , 1991.

[4] CCITT, "Integrated Services Digital Network (ISDN)" : Reccommendations of the Series I, VIIth Plenary Assembly, Malaga-Torremolinos , 8-19 October 1984.

[5] CCITT, "Data Communications Networks Interface" : Reccommendations X.20-X.32, VIIth Plenary Assembly Malaga-Torremolinos , 8-19 October 1984.

[6] CCITT, "Digital Subcriber Signalling System No.1 (DSS1), Network Layer, User- Network Management" : Reccommendations Q930-Q940, IXth Plenary Assembly Melbourne , 14-25 November 1989 .

[7] ประสิทธิ์ ทีฆพุดิ , "โครงข่ายบริการสื่อสารร่วมระบบดิจิตอล" : กรุงเทพฯ : วิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์ , สิงหาคม 2535.

[8] มนูญ สุขเกษม และ ถวิล พึ่งมา "เทคโนโลยีการวิเคราะห์โปรโตคอลของโครงข่ายบริการร่วมระบบดิจิตอล" กรุงเทพฯ: สำนักพิมพ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2536.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะสำเร็จไปไม่ได้ ถ้าขาดความช่วยเหลือและการให้คำปรึกษาจาก อาจารย์กวิล พึ่งมา อาจารย์มนูญ สุขเกษม อาจารย์อิทธิชัย อรุณศรีแสงไชย และ ผู้เชี่ยวชาญ ญี่ปุ่น Mr.KANAME HIRAGURI รวมไปถึงพี่ๆปริญญาโทที่ห้อง T201 และเพื่อนๆทุกคนที่คอยช่วย เหลือและให้กำลังใจมาตลอด จนทำให้ปริญญานิพนธ์นี้สามารถสำเร็จขึ้นมาได้ ทางคณะผู้จัดทำมี ความรู้สึกซาบซึ้งใจเป็นอันมาก จึงขอขอบพระคุณทุกท่านที่ได้กล่าวมาข้างต้นไว้ ณ ที่นี้ และขอ อำนาจคุณพระศรีรัตนตรัย ดลบันดาลให้ทุกท่านประสพแต่ความสุขความเจริญตลอดไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้