หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

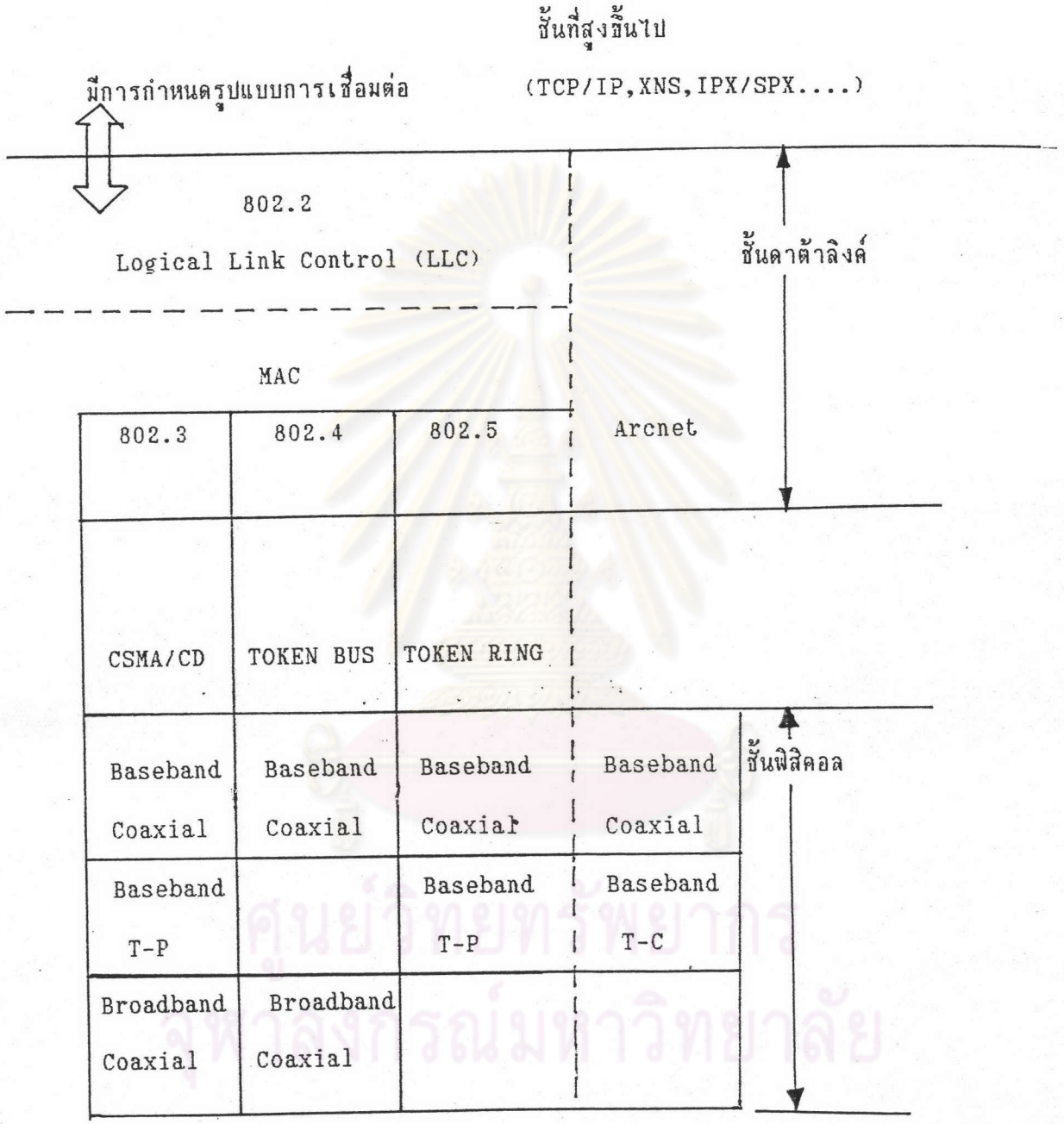
การติดต่อกับแลนอินเตอร์เฟสการ์ดแบบโทเกนริงกับโปรแกรมในระดับต่างๆ

5.1 โครงสร้างของเน็ตเวิร์คแบบเป็นลำดับชั้น (Jordan 1990)

ตามรูปที่ 5.1 แสดงมาตรฐานของการติดต่อกันของระบบสื่อสารข้อมูล OSI 7 ชั้นของ ISO เทียบกับผลิตภัณฑ์ที่มีจริงๆในท้องตลาด การแบ่งลำดับชั้นนี้ชั้นบนจะเรียกใช้บริการชั้นภายใต้ลงมา โดยที่ OSI กำหนดเพียงฟังก์ชันต่างๆเท่านั้น ซึ่งผลิตภัณฑ์ที่มีอยู่จะพยามเทียบเคียงให้เข้ากันได้กับมาตรฐาน OSI นี้ แต่ก็มีข้อกำหนดที่แตกต่างหรือเพิ่มเติมไปบ้าง ส่วนรูปที่ 5.2 นั้นเปรียบเทียบผลิตภัณฑ์ของเครือข่ายท้องถิ่นเทียบกับมาตรฐาน

IEEE 802		
แอปลิเคชั่น (Application)		เน็ตเวิร์คโอเอส แลนดูทิลิตี้, ไฟล์ทรานสเฟอร์, เทอร์มินอลอิมูเลชั่น
พรีเซ็นเตชั่น (Presentation)		โอเอสของโฮสต์ (DOS, OS/2, UNIX..)
เซสชั่น (Session)		เน็ตเวิร์คโอเอส TCP/IP NETBIOS IBM PC LAN Suppost Program
ทรานสปอร์ต (Transport)		
เน็ตเวิร์ค (Network)		
ดาต้าลิงค์ (Datalink)		ระบบแลนระดับล่าง (ETHERNET, TOKENRING, ARCNET....)
ฟิสิคอล (Physical)		

รูปที่ 5.1 เปรียบเทียบโพรโตคอลแบบต่างๆกับมาตรฐาน OSI 7 ชั้น(Jordan 1990)



รูปที่ 5.2 มาตรฐาน IEEE และผลิตภัณฑ์ที่มีอยู่

(Jordan 1990)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.2 โครงสร้างของ IBM PC LAN และการติดต่อกับโปรแกรมระดับสูง (Martin 1989)

ในการติดต่อกับแลนอินเตอร์เฟสการ์ดแบบโทเกนริงนั้นเราจำเป็นต้องทราบโครงสร้างผลิตภัณฑ์เกี่ยวกับแลนของบริษัทไอบีเอ็ม ซึ่งเป็นผู้เริ่มพัฒนาผลิตภัณฑ์จนกลายเป็นมาตรฐานในการอุตสาหกรรมไปหลายอย่าง

5.2.1 ผลิตภัณฑ์ของแลนของ IBM PC

ผลิตภัณฑ์แลนของ IBM มีความสัมพันธ์กับ OSI และ IEEE ดังรูป

<table>
<tr><td colspan="3">ผลิตภัณฑ์ IBM PC LAN</td><td colspan="2">OSI โมเดล</td></tr>
<tr><td colspan="3" rowspan="2">IBM PC LAN Program</td><td>Application</td><td rowspan="4"></td></tr>
<tr><td>Presentation</td></tr>
<tr><td colspan="3" rowspan="3">IBM PC LAN Support Program
(NETBIOS)</td><td>Session</td></tr>
<tr><td>Transport</td><td rowspan="2">IEEE 802
โมเดล</td></tr>
<tr><td>Network</td></tr>
<tr><td colspan="3" rowspan="2">LLC (DLC)</td><td rowspan="2">Datalink</td><td>LLC</td></tr>
<tr><td>MAC</td></tr>
<tr><td>IBM
TOKEN
RING</td><td>IBM-PC
NETWORK
Broadband</td><td>IBM-PC
NETWORK
Baseband</td><td>Physical</td><td>Physical</td></tr>
</table>

รูปที่ 5.3 ผลิตภัณฑ์ IBM LAN เทียบกับมาตรฐาน OSI และ IEEE (Martin 1989)

ในระดับ MAC นั้นมีให้เลือก 3 แบบ

5.2.1.1 IBM Token-Ring Network

ตามมาตรฐานของ IEEE802.5 โทเกนริงเน็ตเวิร์ค

โดยใช้การเดินสายแบบเบสแบนด์

5.2.1.2 PC Network-Broadband

ตามมาตรฐานของ IEEE802.3 CSMA/CD เน็ตเวิร์ค รุ่นนี้จะใช้การส่งสัญญาณแบบบอร์ดแบนด์

5.2.1.3 PC Network-Baseband

ตามมาตรฐานของ IEEE802.3 CSMA/CD เน็ตเวิร์ค แต่ใช้ระบบการส่งแบบเบสแบนด์ ผลิตภัณฑ์เหล่านี้จะประกอบด้วยชั้นฟิสิคอล และ ชั้นย่อย MAC ซึ่งเป็นส่วนหนึ่งของชั้น ดาต้าลิงค์ ซึ่งถูกบรรจุอยู่ในฮาร์ดแวร์ในรูปแบบของเน็ตเวิร์คอินเตอร์เฟสการ์ดที่จะเสียบลงไปในสลอตของเครื่อง PC ของ IBM หลายๆรุ่น

IBM Local Area Network Support Program เป็น ผลิตภัณฑ์ซอฟต์แวร์ของ IBM ซึ่งจะจัดเสนอการอินเตอร์เฟสในระดับ LLC กับ ฮาร์ดแวร์ทั้งสามตัวของ IBM การอินเตอร์เฟสตรงตามมาตรฐาน IEEE802.2 LLC ซึ่งเดิมทีเดียวนั้น โปรแกรมนี้เรียกว่า Adapter Support Interface นอกจากนี้ โปรแกรมนี้ยังมีการให้ติดต่อกับโปรแกรมในระดับสูงขึ้นไปได้ โดยผ่านทาง NETBIOS ซึ่งฟังก์ชันในการเรียกผ่านนี้เทียบได้กับชั้น เน็ตเวิร์ค, ทรานสปอร์ต และ เซสชัน โดย NETBIOSนี้จะจัดเสนอการติดต่อได้ สองแบบ คือการส่งข้อมูลแบบเชื่อถือได้ (Reliable data service) และ ส่งแบบดาต้าแกรม (Datagram)โดยที่การส่งแบบเชื่อถือได้นั้นจะต้องมีการสร้างเซสชัน (Session) ระหว่างผู้รับและผู้ส่ง แต่ถ้าแบบดาต้าแกรม นั้นไม่มีการสร้างเซสชันระหว่างผู้รับและผู้ส่ง

ตัวผลิตภัณฑ์อื่นก็มีโปรแกรม IBM PC LAN Program ซึ่งจะช่วยให้ตัวโปรแกรมประยุกต์อินเตอร์เฟสกับแลนได้ โปรแกรมนี้เทียบได้กับระบบปฏิบัติการของเน็ตเวิร์ค ซึ่งจะจัดเสนอการใช้งานทางด้านการส่งข้อมูล การทำพริ้นและไฟล์เซิร์ฟเวอร์สำหรับเครื่องพิมพ์และการแชร์ไฟล์ การเรียกใช้งานสามารถเรียกทางโปรแกรมประยุกต์ที่เขียนขึ้นมา หรือเรียกใช้โดยตรงจากผู้ใช้ที่หน้าจอ ผ่านทางระบบเมนูที่มีอยู่แล้วก็ได้ ความสามารถเช่นนี้ จึงทำให้โปรแกรมประยุกต์ต่างๆสามารถ ใช้ประโยชน์จากสภาพ

แวดล้อมในแลนได้

5.3 เฟรมต่างๆของ IBM โทเกนริง (Nance 1990)

5.3.1 MAC เฟรม

เฟรมในการส่งแบบโทเกนริง ประกอบด้วย โทเกน และ เฟรม ดังนี้

5.3.1.1 โทเกน

จะวิ่งวนอยู่ในวงแหวนจนกว่าจะมีสถานีใด ต้องการส่งเฟรมข้อมูลออกมาก็จะจับโทเกนแล้วเอาข้อมูลใส่เข้าไป

5.3.1.2 เฟรมข้อมูล (Data Frame)

ในเฟรมจะบรรจุข้อมูลที่ต้องการส่งให้สถานีอื่น และ บางครั้งก็จะบรรจุข้อมูลที่ใช้ในการควบคุมการส่งของโทเกนริงอินเตอร์เฟสการ์ดเอง

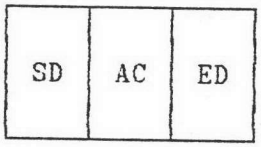
5.3.1.3 ซีเควนซ์การยกเลิก (AbortSequence)

ซึ่งไม่ถือว่าเป็นเฟรมใช้ในการส่ง แต่เพื่อบอกให้อินเตอร์รัปต์หรือยกเลิกการส่งนั้นเสีย

Starting Delimeter	Access Control	Frame Control	Destination Address	Source Address	Routing Information
1 ไบต์	1 ไบต์	1 ไบต์	6 ไบต์	6 ไบต์	0-r ไบต์

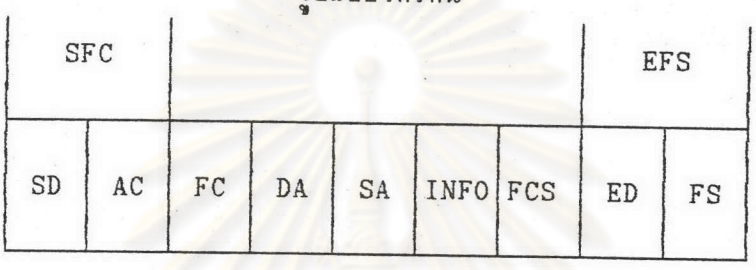
Information	Frame Check Sequence	Ending Delimeter	Frame Status
0-n ไบต์	4 ไบต์	1 ไบต์	1 ไบต์

รูปที่ 5.4 โทเกนริงเฟรมของ IBM (Martin 1989)



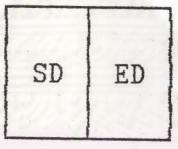
1 1 1

รูปแบบโทเกน



1 1 1 2-6 2-6 0-n 4 1 1

รูปแบบเฟรมข้อมูล



1 1

รูปแบบ Abort Sequence

รูปที่ 5.5 โทเกนและเฟรมของโทเกนริง (Nance 1990)

ฟิลด์ต่างๆของเฟรมข้อมูลของโทเกนริงประกอบด้วย

Starting Delimiter : ฟิลด์นี้จะเป็นตัวบอกตำแหน่งของ เฟรม หรือโทเกน ในเฟรมนี้จะไม่บรรจุเฉพาะ 0 และ 1 เท่านั้น แต่จะมีการส่งสัญญาณไฟฟ้าระดับพิเศษขึ้นมาเพื่อจะแยกแยะได้ชัดเจนว่านี่เป็น ตัวเริ่มต้นเฟรมเท่านั้น

AccessControl : จะมีทั้งในเฟรมและโทเกนและมีรูปแบบเป็น

PPPTMRRR

โดยที่ 3 บิตแรก(PPP) ใช้สำหรับกำหนดลำดับความสำคัญ ของเฟรมหรือโทเกน บิตต่อมา(T)จะแยกแยะว่าเป็นเฟรมหรือเป็นโทเกน ส่วน monitor(M) บิตจะใช้กับแอคทีฟมอนิเตอร์เพื่อตรวจจับการหมุนวนโทเกนไปในริง ว่าเป็นไปโดยต่อเนื่องหรือเปล่า reservation บิต (RRR) ใช้สำหรับการจองการเรียกใช้โทเกนโดยจะสัมพันธ์กับ priority บิต

Frame Control : ใช้บอกจุดประสงค์การใช้เฟรมนั้นว่าเป็น MAC หรือ LLC เฟรม สำหรับ MAC เฟรม ในฟิลด์นี้จะมีตัวกำหนดเกี่ยวการทำบัฟเฟอร์ด้วย รูปแบบของบิตในฟิลด์นี้คือ field type และ MAC control ID

FF CCCCCC

ถ้า FF มีค่าเป็น 00 จะเป็น MAC เฟรม ถ้า 01 จะเป็น LLC เฟรม โดยที่ MAC เฟรมนั้น ตัว MAC control ID จะเป็นตัวระบุประเภทของ MAC เฟรม คือ

C C C C C C	Type
0 0 0 0 1 1	Claim Token
0 0 0 0 0 0	Duplicate Address Test
0 0 0 1 0 1	Active Monitor Present
0 0 0 1 1 0	Standby Monitor Present
0 0 0 0 1 0	Beacon
0 0 0 1 0 0	Purge

Address Fields : จะมีขนาด 6 ไบต์(48 บิตแอดเดรส) ตัว destination address จะบอกถึงตัวแอดเดรสของเครื่องที่จะทำการรับข้อมูล ส่วน source address จะเป็นตัวบอกถึงสถานีที่ทำการส่งข้อมูล และบิตแรกของไบต์แรกจะเป็นตัวบอกว่าในเฟรมนั้นมีข้อมูลเกี่ยวกับการทำเส้นทาง (Routing) หรือไม่

Routing Information : ฟิลด์นี้ขนาดความยาวจะไม่คงที่ และจะใช้ก็ต่อเมื่อมีการส่งเฟรมข้ามบริดจ์ไปยังริงอื่น ในการต่อกันแบบมีริงหลายวงต่อกัน ตัว

คำสั่ง	ฟังก์ชั่น
Transmit.Dir.Frame	ส่งเฟรมข้อมูล
Receive	รับข้อมูลจากสถานีที่กำหนด
Recieve.Cancel	หยุดการรับข้อมูลจากสถานีที่กำหนด
Dir.Cancel.Timer.Group	ยกเลิกกลุ่มคำสั่งไทเมอร์
Dir.Close.Adapter	ปิดอะแดปเตอร์และยกเลิกการติดต่อ
Dir.Define.MIF.Environment	กำหนดสภาพแวดล้อมต่างๆที่ต้องการสำหรับโปรแกรม NETBIOS จำลอง
Dir.Interrupt	บังคับให้อะแดปเตอร์ทำการอินเตอร์รัปต์
Dir.Initialize	ทำการอินิเชียลไลซ์ตารางอินเตอร์เฟส, บัฟเฟอร์, และพื้นที่ต่างๆ
Dir.Modify.Open.Parms	แก้ไขพารามิเตอร์ต่างๆที่เกี่ยวอะแดปเตอร์
Dir.Open.Adapter	ทำให้อะแดปเตอร์พร้อมที่จะทำการติดต่อ
Dir.Read.Log	อ่านข้อมูลที่บันทึกไว้แล้วรีเซ็ต
Restore.Open.Parms	เรียกคืนพารามิเตอร์ที่เกี่ยวกับอะแดปเตอร์
Dir.Functional.Address	เซ็ตแอดเดรสสำหรับอะแดปเตอร์ที่จะรับข้อมูล
Dir.Set.Group.Address	เซ็ตแอดเดรสกลุ่มสำหรับอะแดปเตอร์ที่จะรับข้อมูล
Dir.Set.User.Appendage	เปลี่ยนแอดเดรสของรูทีนของผู้ใช้ที่จะให้ไปทำเมื่อเสร็จการทำคำสั่ง
Dir.Status	อ่านข้อมูลสถานภาพทั่วไป
Dir.Timer.Cancel	ยกเลิกไทเมอร์
Dir.Timer.Set	เซ็ตไทเมอร์
PDT.Trace.On	จัดให้มีการติดตามอินเตอร์รัปต์สำหรับการส่งผ่านข้อมูลทั้งหมดของอะแดปเตอร์
PDT.Trace.Off	ยกเลิกการติดตาม

ตารางที่ 5.1 คำสั่งที่ใช้ในการทำ Direct Interface (ระดับ MAC) ของ IBM โทเกนริง ซึ่งเรียกว่า Register Direct (Martin 1989)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บอกการทำเส้นทางนี้ไม่มีในข้อกำหนดมาตรฐานของ IEEE 802.5

Information : ฟิลด์นี้จะบรรจุข้อมูล ที่ใช้ในการส่งข้ามเน็ตเวิร์ค ซึ่งอาจเป็น MAC เฟรมประเภทใดประเภทหนึ่งตามที่กล่าวมาแล้ว หรือเป็นข้อมูลของโพรโตคอลที่อยู่สูงขึ้นไป เช่น IPX หรือ NETBIOS ขนาดความยาวของฟิลด์จะไม่คงที่ขึ้นอยู่กับขนาดข้อมูล

Frame Check Sequence : ในฟิลด์นี้จะบรรจุค่า CRC ซึ่งเป็นการตรวจสอบความถูกต้องของข้อมูลที่ส่งจากตัวส่งมาถึงตัวรับดูว่าถูกต้องหรือไม่

Ending Delimiter : เป็นตัวบอกว่าสิ้นสุดเฟรม ซึ่งข้อมูลที่บรรจุอยู่จะคล้ายกับตัว starting delimiter ที่มีตัวสัญญาณไฟฟ้าพิเศษเพื่อแยกแยะว่านี่เป็นตัวปิดท้าย และในเฟรมนี้มี intermediate frame bit เพื่อใช้ในการส่งแบบเฟรมหลายเฟรม โดยใช้โทเกนเดียว และนอกจากนี้ยังมีบิต error-detectd ซึ่งถ้าสถานีรับตรวจสอบว่ามีข้อผิดพลาดเกิดขึ้น(โดยตรวจสอบจาก CRC) จะทำการเซตบิตนี้ สถานีอื่นที่ได้รับก็จะส่งผ่านต่อไปเรื่อยๆจนถึงสถานีส่ง และสถานีส่งก็จะทำการส่งโทเกนหรือ เฟรม นั้นออกไปใหม่

Frame Status : ฟิลด์นี้ใช้กับสถานีรับ ในการที่จะส่งข้อมูลสถานภาพกลับมายังสถานีส่ง โดยจะมีรูปแบบบิตเป็น

ACxxACxx

โดยที่ A คือ address-recognized บิต จะถูกเซตเมื่อสถานีรับตรวจสอบพบว่าแอดเดรสที่ได้รับนั้นเป็นแอดเดรสของตัวเอง

C คือ copied บิต จะเป็นตัวบอกว่าเฟรมนั้นได้ถูกก๊อปปี้ลงในบัฟเฟอร์รับของสถานีนั้น และถ้า A และ C ถูกเซตส่งกลับไปยังสถานีส่ง สถานีส่งก็จะทราบว่าเฟรมที่ตัวส่งไปนั้นได้รับและมีการก๊อปปี้ไปแล้ว และการที่ต้องมีบิตนี้สองชุดก็เพื่อป้องกันการผิดพลาด

5.3.2 การอินเตอร์เฟสกับระดับ MAC (Martin 1989)

โทเกนริงเน็ตเวิร์ค ของ IBM ได้กำหนดการบริการระดับ MAC

โดยยึดตามมาตรฐาน IEEE802.5 ซึ่งมีดังนี้

SEND_AC_DATA : ซึ่งจะทำให้เกิดการส่งข้อมูลหนึ่งหน่วยไปยัง MAC เพื่อส่งไปยังเน็ตเวิร์ค คำสั่งนี้เท่ากับ MA_DATA.requestในมาตรฐาน IEEE 802.5

RECIEVE_AC_DATA : ทำให้หน่วยข้อมูลที่ได้รับจากเน็ตเวิร์คส่งผ่านจากชั้น MAC มายัง LLC คำสั่งนี้เท่ากับ MA_DATA.indication ในมาตรฐาน IEEE 802.5

CONFIRM_AC_DATA : จากเป็นผลตอบสนองจากคำสั่ง SEND_AC _DATA ไม่ว่าการส่งครั้งนั้นมีความสมบูรณ์หรือไม่ก็ตาม คำสั่งนี้เท่ากับ MA_DATA. confirmation ในมาตรฐาน IEEE 802.5

การที่จะให้โปรแกรมประยุกต์ ติดต่อโดยตรงกับ MAC นั้น สามารถทำได้โดยการ เรียกโดยตรงเรียกว่าการทำการอินเตอร์เฟสโดยตรง (Direct interface)โดยใช้คำสั่งดังแสดงใน ตารางที่ 5.1 ในการเรียกแบบนี้นั้นจะเรียกผ่านทาง CCB(Command Control Block) ซึ่งใน CCB จะประกอบด้วยข้อมูลที่จำเป็นในการรันคำสั่งนั้นๆ CCBจะประกอบไปด้วยรหัสคำสั่งซึ่งจะเป็นตัวกำหนด ฟังก์ชันที่จะให้ทำงาน นอกจากนี้ก็เป็นการกำหนดพื้นที่ทำงาน และ พารามิเตอร์ ที่ต้องใช้ในการทำฟังก์ชันนั้น และจะมีส่วนที่เป็น พื้นที่รับรหัสผลการทำงาน (Completion code area)ซึ่งจะใช้ในการตอบกลับว่าการทำงานตามคำสั่งนั้นสำเร็จหรือไม่และนอกจากนี้เรายังอาจให้CCBกำหนดค่าแอดเดรสของโปรแกรมของผู้ใช้ให้ทำงานถ้าการทำคำสั่งนั้นเสร็จเรียบร้อยแล้ว

การทำคำสั่งเหล่านี้สรุปได้ว่าเป็นการปฏิบัติการ 5 อย่างดังนี้

ก. เปิดและปิดอะแดปเตอร์

ข. เซตและแก้ไขค่าในอะแดปเตอร์ และเอ็กซีคิวต์ค่าสภาพแวดล้อมต่างๆ

ค. หาค่าผิดพลาดต่างๆ

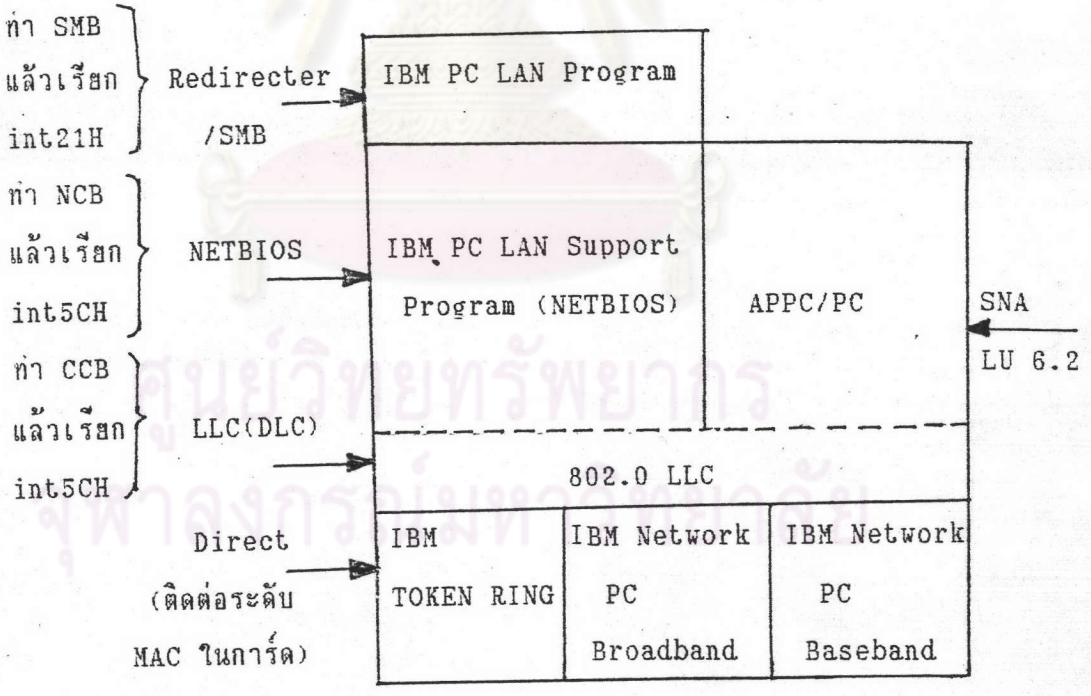
ง. เซตแอดเดรส

จ. ส่งเฟรม

โดยปรกติแล้วในระดับชั้นย่อย MAC นี้จะเป็นเฟิร์มแวร์(Firmware) อยู่ในอินเตอร์เฟสการ์ดอยู่แล้ว เพราะฉะนั้นจึงเป็นการโปรแกรมระดับต่ำสุด ซึ่งปรกติมันจะถูกเรียกใช้จาก LLC ชั้น MAC นี้จะไปกำเนิดสัญญาณในระดับฟิสิคอล อีกที

5.4 การอินเตอร์เฟสในระดับ LLC (Martin 1989; Roetzeim 1991)

IBM LAN Support Program จะเสนอการอินเตอร์เฟสในระดับ LLC ไว้ให้ด้วย แต่จะใช้คำว่า DLC(Data Link Control) แทน โดยที่การอินเตอร์เฟสนี้จะยึดตามมาตรฐาน IEEE 802.2 ชั้น LLC นี้จะถูกเรียกใช้โดยโปรแกรมระดับสูงขึ้นไป เช่น NETBIOSเป็นต้นรูปที่ 5.6 จะแสดงการอินเตอร์เฟส IBM PC LAN โดยซอฟต์แวร์ในระดับต่างๆ



รูปที่ 5.6 การอินเตอร์เฟสกับ IBM LAN ในระดับต่างๆ (Martin 1989)

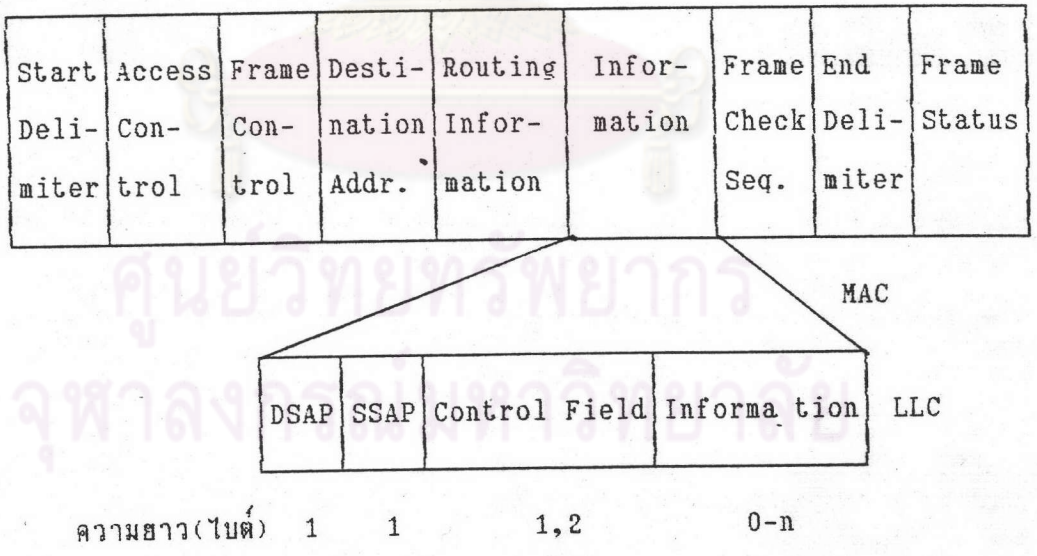
Logical Link Control มีการบริการอยู่สองแบบคือ

แบบที่ 1 : Connectionless Service หรือ Datagram Service

ในการรับส่งระหว่างสถานีนั้นจะไม่มีการรับรองว่าข้อมูลจะได้รับหรือไม่

แบบที่ 2 : Connection-oriented Service หรือ Virtual Circuit

จะต้องมีการสร้างเซสชั่นกันก่อนระหว่างสถานี แล้วจึงค่อยมีการรับส่งกัน ซึ่งทั้งนี้จะทำให้การส่งข้อมูลมีความเชื่อถือได้มากขึ้น เพราะจะมีการตอบรับ ถ้าข้อมูลได้รับแล้ว การส่งข้อมูลผ่านจากชั้นย่อย LLCไปยังชั้นย่อย MAC ตัว LLC เฟรมจะกลายเป็นฟิลด์ข้อมูลใน MAC เฟรม



รูปที่ 5.7 หน่วยข้อมูลของ MAC ที่เป็น LLC (Martin 1989)

LLC header จะประกอบด้วย DSAP (Destination Service Access Point)และ SSAP (Source Service Access Point) ซึ่งตัว SAP นี้จะเป็นจุดที่จะเรียกติดต่อจากระดับเน็ตเวิร์คมาหรือไปยังชั้นย่อย LLC อีกทีหนึ่ง ตัวอุปกรณ์จะใช้ SAP เป็นจุดส่งหน่วยข้อมูล ไปยังอุปกรณ์ตัวอื่น อุปกรณ์หนึ่งตัวสามารถใช้ SAP หลายๆจุดในการประมวลผลในการรับหน่วยข้อมูล จากหลายๆแหล่ง ตัว SAP ที่เกี่ยวกับการส่ง หน่วยข้อมูล จะเรียก 'SSAP'แต่ถ้าเกี่ยวกับการรับ หน่วยข้อมูล จะเรียก 'DSAP'

การส่งผ่านค่าต่างๆเช่น SAP จากชั้น LLC ไปยัง MAC จะส่งผ่านเป็นพารามิเตอร์สำหรับรูปแบบของฟิลด์ควบคุม ขึ้นอยู่กับ ประเภทของการบริการและตัวคำสั่งที่ หน่วยข้อมูลบรรจุอยู่

คำสั่งในการเรียกใช้บริการในระดับ LLC นี้ตามมาตรฐาน IEEE 802.2 ได้กำหนดรูปแบบการบริการพื้นฐาน(Service primitive)ไว้ 3 รูปแบบคือ

Request : จะเป็นการเรียกใช้การบริการอย่างใดอย่างหนึ่งให้ทำงาน

Indication : ใช้ในการบอกให้รู้ว่าเกิดอะไรขึ้นแล้วในการเรียกใช้บริการนั้น

Confirm : เป็นการบอกย้ำให้รู้การเรียกใช้บริการนั้น

ในมาตรฐาน IEEE 802 (ANSI/IEEE Std. 802.2 1984) ได้กำหนดชุดคำสั่งในการเรียกใช้บริการ LLC ไว้เป็นโครงร่างโดยที่ไม่ได้ระบุรูปแบบและภาษาที่ใช้เป็นการตายตัวซึ่งมีดังต่อไปนี้

Connectionless Service

L_DATA.request

L_DATA.indication

Connection-Oriented Service

L_CONNECT.request

L_CONNECT.indication

L_CONNECT.confirm

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

L_DATA_CONNECT.request

L_DATA_CONNECT.indication

L_DATA_CONNECT.confirm

L_DISCONNECT.request

L_DISCONNECT.indication

L_DISCONNECT.confirm

L_RESET.request

L_RESET.indication

L_RESET.confirm

L_CONNECTION_FLOWCONTROL.request

L_CONNECTION_FLOWCONTROL.indication

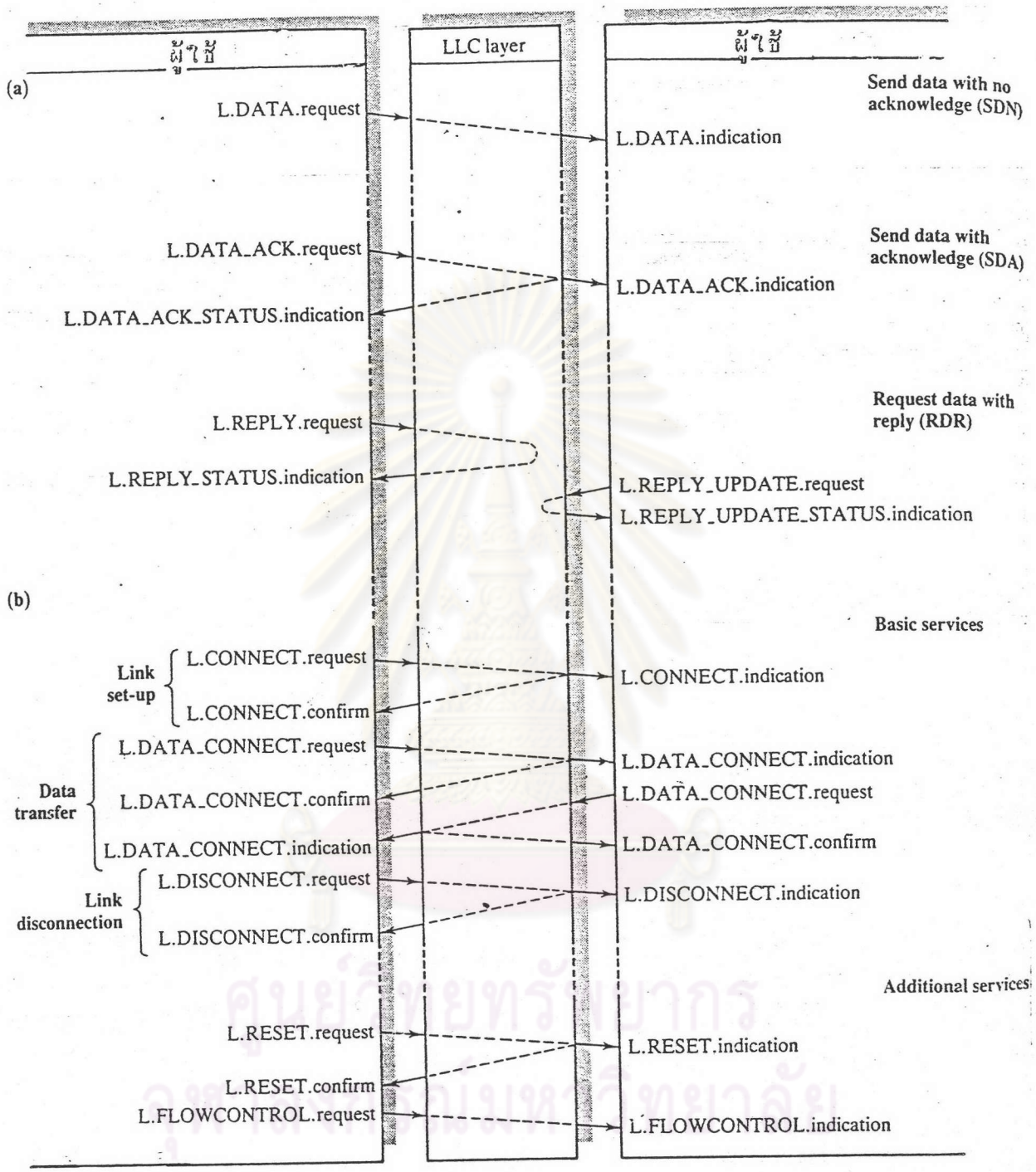
ตัวอย่างการเรียกใช้งานก็เช่น

```
L_DATA.request(
                local_address,
                remote_address,
                l_sdu,
                service_class
              )
```

ตัวแปรที่ส่งผ่านมีความหมายดังนี้

local_address : เป็นค่า SSAP และ sending station

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

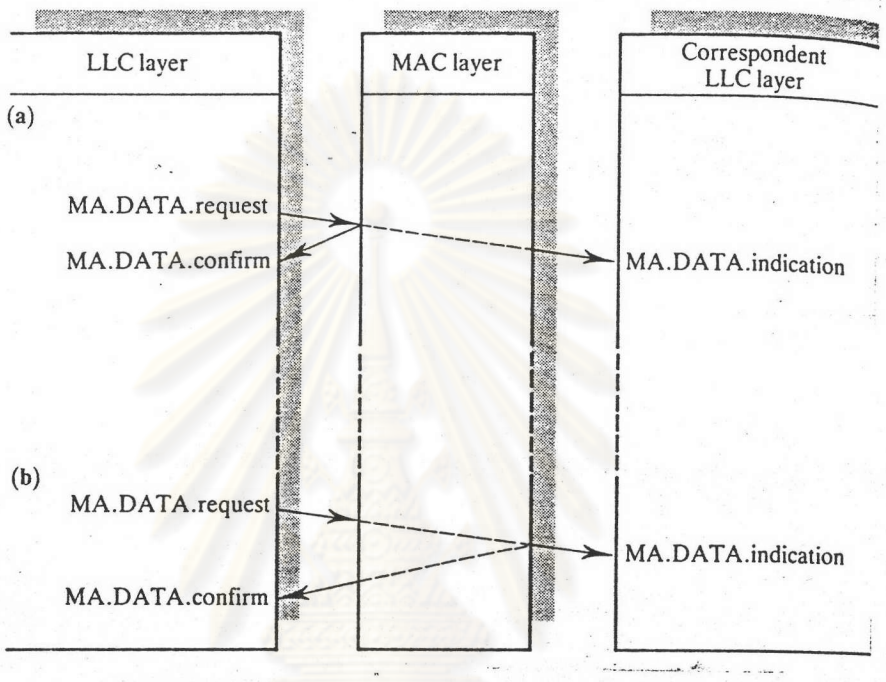


รูปที่ 5.8 แสดงโครงร่างต้นแบบการให้บริการในระดับ LLC แก่โปรแกรมระดับสูงขึ้นไป กำหนดโดย IEEE โดยที่

a) เป็นแบบ connectionless

b) เป็นแบบ connection oriented

(Halsall 1989)



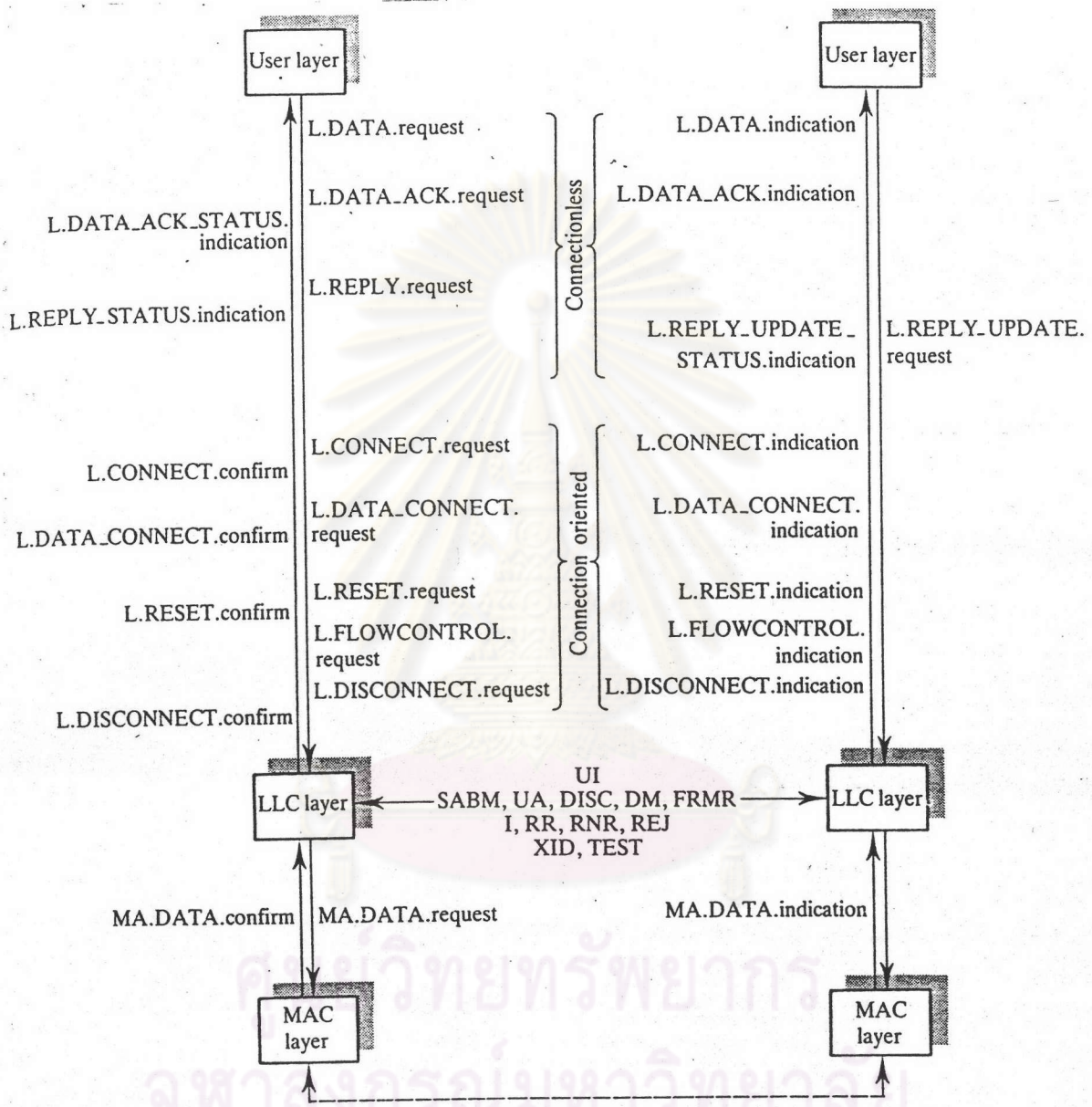
รูปที่ 5.9 แสดงโครงร่างต้นแบบการให้บริการในระดับ MAC ให้แก่ระดับ LLC กำหนดโดย IEEE โดยที่

a) แบบ CSMA/CD

b) แบบ โทเกนริง/บัส

(Halsall 1989)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.10 แสดงการเรียกใช้บริการในแต่ระดับชั้น ซึ่งกำหนดโดย IEEE

(Halsall 1989)

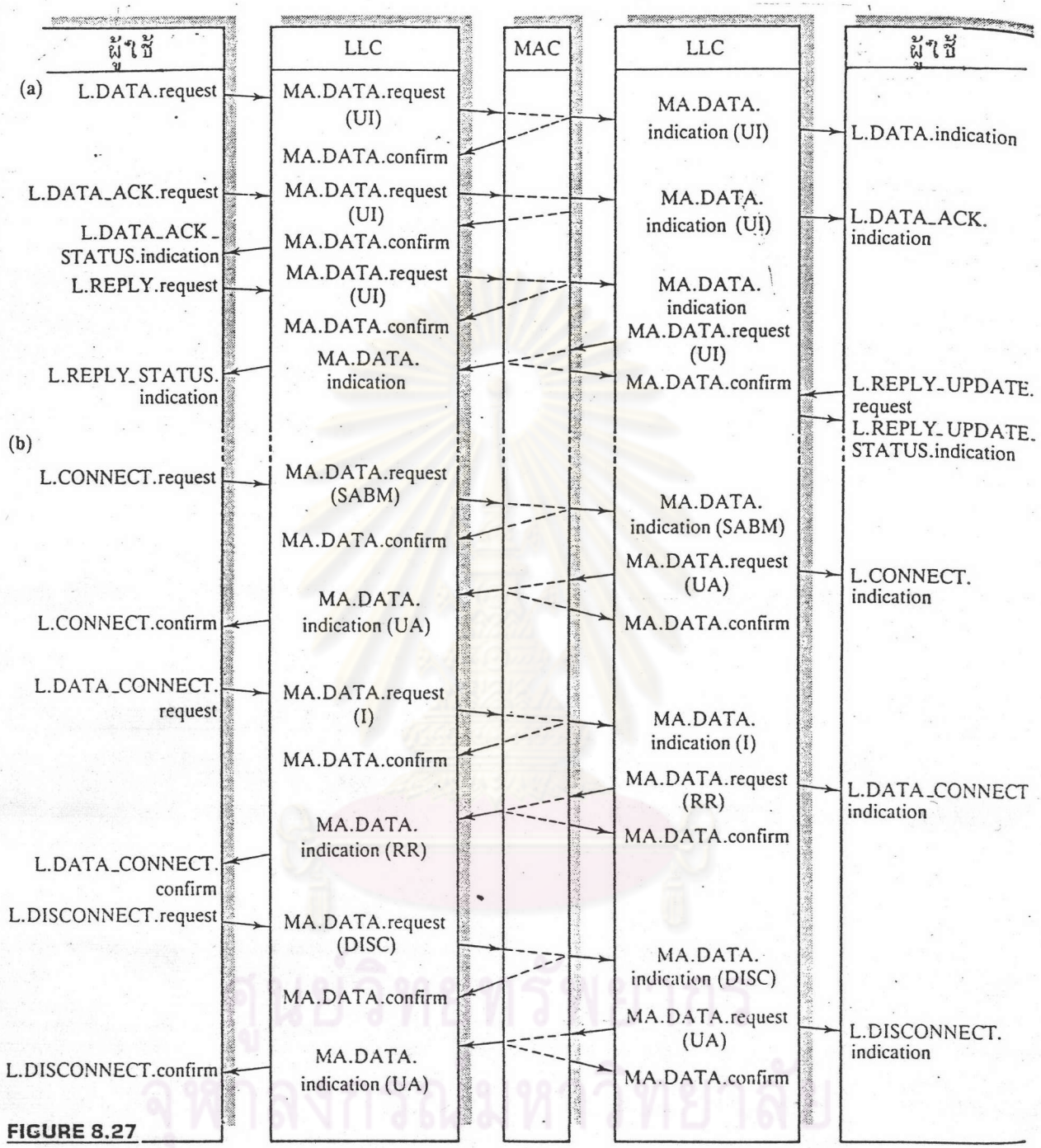


FIGURE 8.27

รูปที่ 5.11 แสดงการโครงร่างต้นแบบการโต้ตอบระหว่างโปรแกรมประยุกต์สองจุดโดยเรียกผ่าน ชั้น MAC และ LLC ซึ่งกำหนดโดย IEEE

a) แบบ connectionless

b) แบบ connection oriented

(Halsall 1989)

คำสั่ง	ฟังก์ชั่น
Dir.Initialize	อินิเชียลไลซ์ค่าเริ่มต้นต่างๆ
Dir.Open.Adapter	เตรียมอะแดปเตอร์ในการติดต่อ
Dir.Close.Adapter	ยกเลิกการติดต่อของอะแดปเตอร์
Transmit.Dir.Frame	ส่งข้อมูลโดยใช้การไดเร็กอินเตอร์เฟส
Buffer.Free	ส่งคืนบัฟเฟอร์กลับบัฟเฟอร์พูล
Buffer.Get	เรียกใช้บัฟเฟอร์
DLC.Modify	แก้ไขค่าที่เกี่ยวกับ SAP
DLC.Reset	ทำให้ค่า SAP เป็นค่าเดิม
DLC.Statistics	อ่านและรีเซ็ตข้อมูลที่เก็บไว้
Transmit.XID.Cmd	ส่งคำสั่ง XID
Transmit.XID.Resp.Final	ส่งคำสั่ง XID ตอบโดยที่ไฟนอลบิตเป็น on
Transmit.XID.Resp.Not.Final	ส่งคำสั่ง XID ตอบโดยที่ไฟนอลบิตเป็น off
Transmit.TEST.Cmd	ส่งคำสั่ง TEST

ตารางที่ 5.2 คำสั่ง LLC (DLC) ทั่วไป ของ IBM โทเกนริง
(Martin 1989)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คำสั่ง	ฟังก์ชั่น
DLC.Open.SAP	ทำให้ SAP ทำงานและสำรองจำนวนลิงค์สเตชั่นให้ด้วย
DLC.Close.SAP	ทำให้ SAP เลิกทำงาน
DLC.Open.Station	สำรองสิ่งต่างๆสำหรับลิงค์สเตชั่น
DLC.Connect.Station	ส่ง คำสั่ง/ค่าตอบสนอง ที่ต้องการในการสร้างการติดต่อระหว่างสองสถานี
DLC.Close.Station	ยกเลิกลิงค์สเตชั่น
Receive	รับข้อมูลจากลิงค์สเตชั่น
Receive.Cancel	หยุดรับข้อมูลจากสถานีที่กำหนด
Receive.Modify	รับข้อมูลและเก็บลงในบัฟเฟอร์
Transmit.UI.Frame	ส่งเฟรมแบบ unnumbered information
Transmit.I.Frame	ส่งเฟรมแบบ information

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5.3 คำสั่ง LLC (DLC) เกี่ยวกับการส่ง ของ IBM โทเกนริง

(Martin 1989)

address

remote_address : เป็นค่า DSAP และdestination station address ซึ่งอาจเป็นแบบเดี่ยวหรือเป็น group address ก็ได้

l_sdu : เป็นตัว data unit ที่ต้องการส่ง

service_class : กำหนด ลำดับความสำคัญที่ต้องการในการส่ง

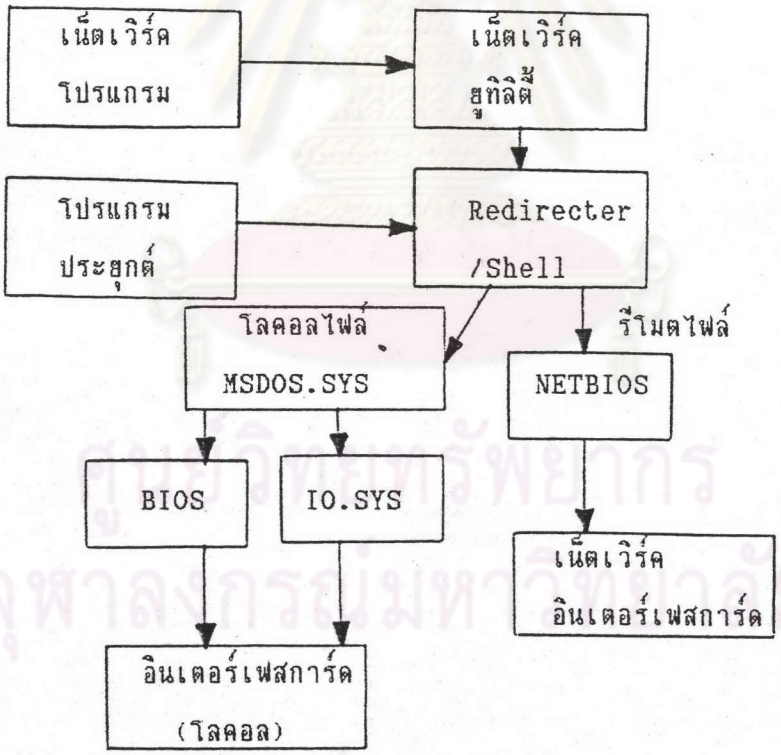
การกำหนดรูปแบบการใช้งานต่างๆนี้ ในทางปฏิบัติแล้วรูปแบบการเรียกและตัวแปรต่างๆที่ใช้นั้นจะขึ้นอยู่กับตัวผลิตภัณฑ์เกี่ยวกับแลน ทั้งฮาร์ดแวร์และซอฟต์แวร์ที่บริษัทต่างๆผลิตขึ้นมาว่าจะให้มีการเรียกใช้อย่างไร ในรูป 5.8 - 5.11 ได้แสดงการติดต่อกันในระดับเดียวกัน ตามมาตรฐาน IEEE ซึ่งจะเห็นว่า ระดับ LLC ติดต่อกับ ระดับ LLC ด้วยกัน ก็จะติดต่อกันโดยมองไม่เห็นระดับ MAC (รูปที่ 5.8) ซึ่งในระดับ LLC นี้ ความจริงก็ไปเรียกใช้บริการระดับ MAC อีกทีหนึ่ง (รูปที่ 5.9) ส่วนรูปที่ 5.10 และ 5.11 จะแสดงให้เห็นการติดต่อระหว่างระดับชั้นและในระดับเดียวกัน ทั้งหมดซึ่งจะเห็นความสัมพันธ์ของการเรียกใช้บริการแต่ละคำสั่งได้ชัดเจน

ในผลิตภัณฑ์ IBM PC LAN Support program นั้นการอินเตอร์เฟสกับ LLC นั้นเราจะเรียกผ่าน CCB (Command Control Block) ตัวโปรแกรมที่จะเรียกจะต้องสร้าง CCB ขึ้นมาก่อนซึ่งมันจะบรรจุข้อมูลและตัวแปรต่างๆที่เหมาะสมในการใช้งานแล้วก็เรียก อินเตอร์รัปต์ 5CH เพื่อเป็นการส่งไปยังชั้นย่อย LLC ข้อมูลต่างๆที่ส่งผ่านโดย CCB นั้นจะเป็นข้อมูลที่จำเป็นในการเอ็กซีคิวต์เช่นแอดเดรสของพื้นที่ทำงานและพื้นที่ข้อมูล บริษัทไอบีเอ็มได้จัดให้การอินเตอร์เฟสระดับ LLC (เรียกว่า DLC) โดยมีโปรแกรมไดร์เวอร์ให้ทั้งแบบเป็นเฟิร์มแวร์และแบบเป็นโปรแกรมแยกต่างหาก คำสั่งในการติดต่อระดับ LLC (DLC) ของ IBM แสดงดังตารางที่ 5.3 และ 5.3

5.5 NETBIOS และการเรียกใช้งาน (Glass 1989; Nance 1990)

5.5.1 ความหมายของ NETBIOS

NETBIOS ย่อมาจาก Network Basic Input/Output System เป็นตัวโพรโตคอลที่ช่วยให้โปรแกรมประยุกต์ใช้งานติดต่อกันบนเน็ตเวิร์คได้ ถ้ามองในอีกแง่หนึ่งก็คือมันเป็นตัว API(Application Program Interface) คือเป็นตัวอินเตอร์เฟสสำหรับโปรแกรมประยุกต์ ให้เรียกใช้งานเพื่อติดต่อกันบนเน็ตเวิร์คได้ NETBIOS จะช่วยให้โปรแกรมประยุกต์(รวมทั้ง DOS) ติดต่อกับเฟิร์มแวร์ และซอฟต์แวร์บนแลนซึ่งบริการการทำงานในระดับ ทรานสปอร์ต ได้ การให้บริการของ NETBIOS นี้เทียบได้กับ การที่ IO.SYS และ BIOS ให้บริการอุปกรณ์เฉพาะที่ (Local) ให้กับ MS-DOS นั่นเอง ส่วนการที่จะรู้ว่าจะเรียกใช้ NETBIOS หรือ BIOS ตัวเชลล์หรือรีไดเร็กเตอร์จะเป็นผู้สลับทิศทางให้ ดังแสดงในรูป 5.12

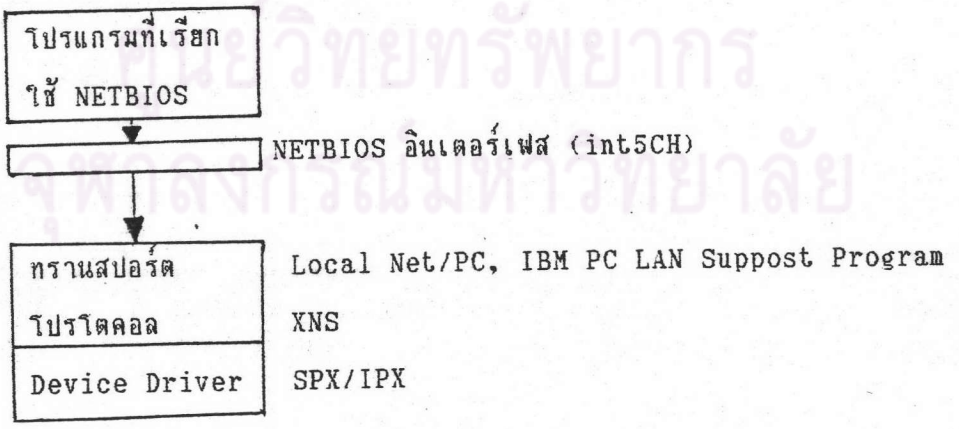


รูปที่ 5.12 โครงสร้างซอฟต์แวร์ระบบของเวิร์คสเตชัน (Jordan 1990)

ถ้ามองเทียบกับมาตรฐาน OSI แล้ว NETBIOS จะเป็นตัวอินเตอร์เฟสระหว่างชั้น 6,7 ติดต่อกับชั้น 5 โดยที่โพรโตคอลในระดับ 1-5 นั้นเดิมทีเดียวนั้นพัฒนาในรูปของ firmware บน IBM PC เน็ตเวิร์คอินเตอร์เฟสการ์ดซึ่งพัฒนาโดยบริษัท Systek ภายใต้ลิขสิทธิ์ของ IBM ต่อจากนั้น ก็ได้มีการพัฒนา NETBIOS บน Token-Ring ,IBM PC broadband และ baseband ด้วย การ์ดโทเกนริง นั้นมีเพียงชั้น 1 และ 2 เท่านั้นส่วนเลเยอร์ที่สูงขึ้นไปนั้นก็ขึ้นกับว่าซอฟต์แวร์บนโฮสต์จะมีการจำลอง (Emulation) ตัว NETBIOS ในชั้นไหนบ้าง ตัวอย่างเช่น IBM PC LAN Support Program จะมีเฉพาะชั้นที่ 5 เท่านั้น

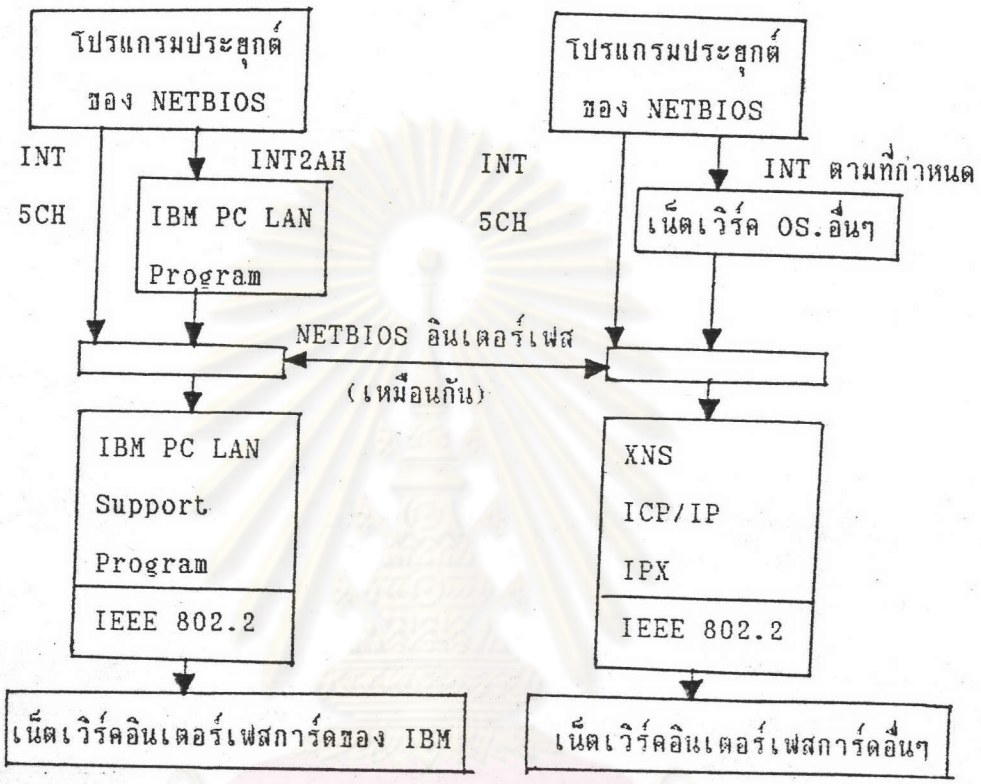
NETBIOS ได้ถือว่าเป็นมาตรฐานตามความนิยม (De facto standard) ของการอินเตอร์เฟสแลนกับโปรแกรมประยุกต์ เพราะมีผู้ใช้กันมาก ตัวโปรแกรมประยุกต์นั้นจะเรียกใช้และรับค่าจาก NETBIOS อินเตอร์เฟสโดยที่จะมองไม่เห็น (Transparent)การทำงานภายใต้นั้นเลย ปัจจุบันนี้เน็ตเวิร์คหลายๆตัวได้ทำ NETBIOS จำลอง โดยที่ส่วนใหญ่จะทำการจำลอง ในชั้น 5 ส่วนชั้นทรานสปอร์ต และเน็ตเวิร์ค นั้นจะแตกต่างกันไป เช่น XNS, TCP/PC และ SPX/IPX ของบริษัทโนเวล ดังรูป 5.14

การเรียกใช้งาน NETBIOS นั้นจะทำโดยเรียกคำสั่งใน IBM PC LAN Support Program โดยผ่านอินเตอร์รัปต์เบอร์ 2A ส่วนประแกรมประยุกต์อื่นๆ สามารถเรียก NETBIOS จำลองโดยเรียกผ่าน อินเตอร์รัปต์ 5CH ดังแสดงในรูป 5.13



รูปที่ 5.13 การอินเตอร์เฟสชั้นแอปลิเคชันและทรานสปอร์ตกับ NETBIOS

(Jordan 1990)

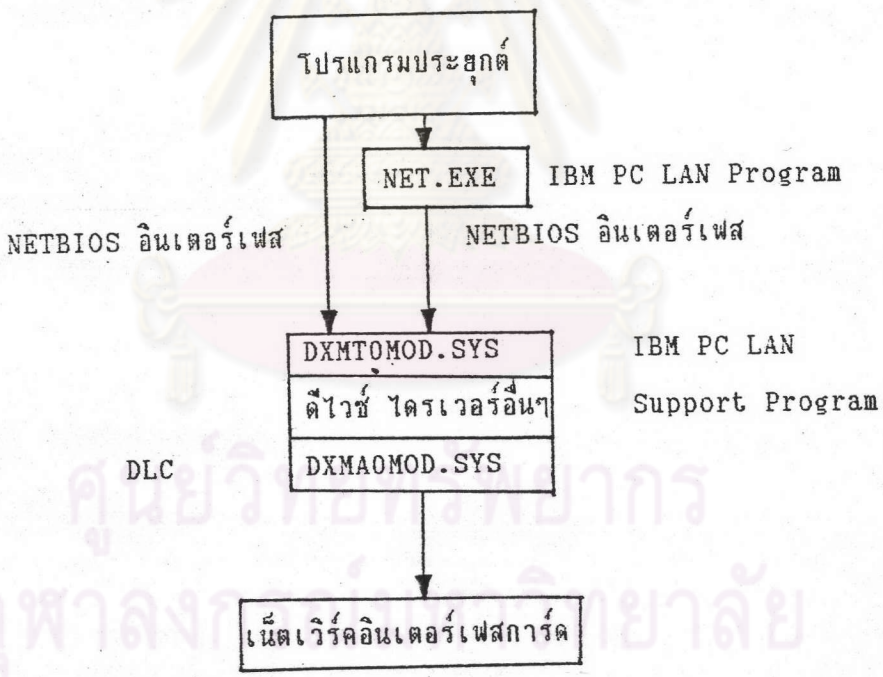


รูปที่ 5.14 การเปรียบเทียบระหว่างเน็ตเวิร์คของ IBM และที่ไม่ใช่ IBM
(Jordan 1990)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.5.1.1 NETBEUI และ DXMT0MOD.SYS

NETBEUI.COMเป็นโปรแกรมที่ถูกพัฒนาโดย IBM ใช้บนโทเกนเน็ตเวิร์ค ซึ่งจะมีการอินเตอร์เฟสพื้นฐานเหมือน NETBIOS ทุกประการ โปรแกรมนี้เป็นโปรแกรมประเภทค้างในหน่วยความจำ(Memory-resident) ต่อมาได้มีการพัฒนา IBM PC LAN Support Program เข้ามาแทนที่ NETBEUI ในรูปของโปรแกรมย่อยควบคุมอุปกรณ์ (Device driver) คือ โปรแกรม DXMT0MOD.SYS โดยมันจะติดต่อโดยตรงกับชิปเซต บนโทเกนริงอินเตอร์เฟสการ์ด และไฟล์ DXMT0MOD.SYS นี้ก็คือ NETBIOS นั่นเอง ซึ่งโครงการเรียกใช้งานนั้นแสดงดังรูป 5.15



รูปที่ 5.15 แสดงสถาปัตยกรรมของ IBM PC LAN Support Program

(Jordan 1990)

การเรียกติดต่อ NETBIOS นั้นเรียกโดยผ่าน MCB (Message Control Block) หรือเรียกว่า NCB (Network Control Block) โดยที่ NCB จะบรรจุข้อมูลที่จำเป็นต่างๆในการทำงานของคำสั่งนั้น แล้วก็เรียกอินเตอร์รัปต์เบอร์ 5CH การใช้ NETBIOS อินเตอร์เฟสนั้นเรียกใช้เหมือนกันกับ LLC อินเตอร์เฟสแต่จะแตกต่างกันที่ข้อมูลในไบต์แรกของบล็อกควบคุม

คำสั่งของ NETBIOS นั้นจะแบ่งได้เป็น 2 แบบ ใหญ่ๆ คือคอย (Wait) และไม่คอย (No-wait) เมื่อเรียกใช้แบบ 'คอย'NETBIOS จะส่งการควบคุมกลับยังโปรแกรมหลักก็ต่อเมื่อมันทำงานเสร็จเรียบร้อยเท่านั้น แต่ถ้าเป็นการใช้แบบ 'ไม่คอย' ก็จะมีทางเลือกอยู่สองแบบคือ การหยั่งเสียง (Polling)โดยจะทำการวนลูปรอจนโปรแกรมเสร็จ หรือ มีการกำหนดแอดเดรสของรูทีนที่เราจะให้ไปทำหลังจาก NETBIOS ทำงานเสร็จแล้วซึ่งรูทีนนี้เรียกว่ารูทีน POST

5.5.2 การให้บริการของ NETBIOS ประกอบฟังก์ชัน 4 ประเภทดังนี้

Status and control : เป็นคำสั่งที่ควบคุม และแสดงสถานภาพการใช้งานในการทำเซสชัน อยู่และรวมทั้งการอินเตอร์เฟสกับ NETBIOS ทั้งหมดด้วย

Name service : ให้บริการในการตั้งชื่อสถานีใดๆ ในเน็ตเวิร์คเพื่อเป็นการแยกแยะแต่ละสถานี และใช้ในการอ้างถึง

Session service : ให้บริการในการสร้างเซสชัน ในระหว่างคู่ติดต่อ 1 คู่ ซึ่งการทำเซสชัน นี้จะทำให้ข้อมูลที่ส่งนั้นมีความเชื่อถือได้

Datagram service : ให้บริการการส่งและรับข้อมูลในแบบดาต้าแกรม คือ ตัวส่งและตัวรับไม่ต้องมีการสร้างเซสชัน ขึ้นมาก่อน แต่ทว่าข้อมูลที่ส่งไปนั้นจะไม่รับประกันว่าได้รับสมบูรณ์ถูกต้องหรือไม่

จุดอ่อนของ NETBIOS คือไม่มีคุณสมบัติในการทำการสื่อสารระยะไกล และการทำระบบชื่อ(Naming) ไม่มีการบริหารที่ส่วนกลางจึงทำให้ยุ่งยากในการควบคุม

รายละเอียดการให้บริการของ NETBIOS ในแต่ละประเภท

5.5.2.1 NETBIOS NAME

NETBIOS นั้นใช้วิธีอ้างอิงถึงกันโดยใช้ชื่อ (Name) เป็นหลักในแต่ละเวิร์คสเตชันและไฟล์เซิร์ฟเวอร์ได้ถูกแยกแยะโดยชื่อที่มีขนาด 16 ไบต์หนึ่งชื่อหรือมากกว่า ตารางที่ใช้ในการเก็บชื่อเหล่านี้ถูกเก็บอยู่ใน NETBIOS และในตารางเก็บชื่อเหล่านี้จะมีชื่อถาวร (Permanent node name) อยู่เสมอ โดยที่ ชื่อนี้จะนำมาจาก 6 ไบต์ของเน็ตเวิร์คแอดเดรสจากเน็ตเวิร์คอินเตอร์เฟสการ์ด และใส่เลข 0 นำหน้าเป็นจำนวน10ไบต์(ก็พอดี16ไบต์) ซึ่งตัวชื่อถาวร นี้จะไม่ซ้ำใครเลยในเน็ตเวิร์ค ในโปรแกรมของเราสามารถที่จะตรวจสอบ เพิ่มชื่อ ลบชื่อได้ตามต้องการ ยกเว้น ชื่อถาวร เท่านั้น นอกจากนี้ชื่อพิเศษคือ ชื่อกลุ่ม (Group name) สามารถเพิ่มเข้าไปได้ในตาราง ซึ่งตัวชื่อกลุ่มนี้จะสามารถซ้ำกันได้ คือหลายๆเวิร์คสเตชันบนเน็ตเวิร์คสามารถมีชื่อกลุ่ม เหมือนกันได้ในเวลาเดียวกัน ตารางของชื่อเหล่านี้จะถูกสร้างขึ้นเมื่อ NETBIOS ถูกโหลดและแต่ละสถานีจะมีชื่อที่ตั้งไว้แล้ว (Default) ไว้ 16 ชื่อ และชื่อเหล่านี้จะใช้ในการอ้างอิงถึงเป็นแอดเดรสต้นทางหรือปลายทาง สำหรับการรับส่งเมื่อมีการสร้างเซสชันขึ้นมา และ NETBIOS จะให้ตัวเลขแก่ชื่อแต่ละชื่อเรียกว่าหมายเลขชื่อ (Name number) ซึ่งจะใช้ตอนส่งดาต้าแกรม

และเพราะว่าแต่ละชื่อจะยาวขนาด16 ไบต์เสมอเพราะฉะนั้นถ้าเป็นชื่อสั้นๆก็ต้องใส่ช่องว่างปิดท้ายด้วยก่อนที่จะมีการเพิ่มชื่อ และห้ามใส่อักษรตัวแรกของชื่อเป็น '*' หรือ ไบนารี 0 '0x00' และที่ควรรู้ก็คือตัวอักษรตัวแรกของชื่อที่เป็นตัว 'I','B' และ 'M' นั้นไม่สามารถตั้งได้เพราะเป็น ชื่อที่สำรองไว้

คำสั่งที่เกี่ยวกับชื่อ นั้นมี add name, add group name, delete name และถ้าเราใช้คำสั่ง reset ของ NETBIOS มันจะทำการลบชื่อทั้งหมดนอกจากชื่อถาวรออกจากตารางชื่อ เพราะฉะนั้นจึงไม่ควรใช้คำสั่ง reset ในขณะกำลังรัน IBM PC LAN Program อยู่

ในการที่จะรับประกันว่า ชื่อที่เพิ่มเข้าไปในตารางชื่อนั้นจะไม่ซ้ำใคร NETBIOS ก็จะทำการวิ่งหาใน ตารางชื่อโลคอล (Local name table)

จนกระทั่งทั้งหมดในเน็ตเวิร์คเพื่อตรวจสอบว่ามีการใช้ชื่อนี้อยู่ก่อนแล้วหรือไม่ ในการ search บนเน็ตเวิร์คนั้น NETBIOS จะส่งเฟรม name claim กระจาย(broadcast) ลงไปในเน็ตเวิร์คและถ้าได้รับตอบเป็นเฟรม name claim response ก็แสดงว่ามีการใช้ชื่อนั้นก่อนแล้ว

5.5.2.2 NETBIOS ดาต้าแกรม

การส่งข้อมูลแบบดาต้าแกรมนั้น เป็นแบบที่ไม่มีการรับประกันในการส่งข้อมูล NETBIOS สามารถทำให้มีการส่งแบบนี้ได้ทั้ง การกำหนดเฉพาะสถานี หรือ การกำหนดเป็นกลุ่มหรืออาจจะส่งไปยังทุกๆสถานีบนเน็ตเวิร์คเลยก็ได้สำหรับข้อมูลที่การส่งแบบนี้สามารถส่งได้ จะมีความยาวไม่เกิน 512 ไบต์ สำหรับคำสั่งเกี่ยวกับการส่งแบบดาต้าแกรม ก็มี send datagram, send broadcast datagram, recieve datagram และ receive broadcast datagram

คำสั่ง	ฟังก์ชั่น
Msg.Add.Name	เพิ่มชื่อไม่ซ้ำใครลงในตารางชื่อของสถานีนั้น
Msg.Add.Group.Name	เพิ่มชื่อกลุ่มลงในตารางชื่อของสถานีนั้น
Msg.Delete.Name	ลบชื่อจากตารางชื่อที่สถานีนั้น
Msg.Find.Name	หาตำแหน่งของชื่อที่กำหนดซึ่งจะได้ค่ากลับมาเป็นแอดเดรส และข้อมูลเส้นทาง

ตารางที่ 5.4 คำสั่งเกี่ยวกับ Name Service ของ NETBIOS
(Martin 1989)

คำสั่ง	ฟังก์ชั่น
Msg.Send.Datagram	ส่งข้อมูลโดยใช้ดาต้าแกรม
Msg.Send.Broadcast.Datagram	ส่งข้อมูลโดยใช้ดาต้าแกรมไปยังทุกสถานี
Msg.Receive.Datagram	รับข้อมูลดาต้าแกรมที่ส่งมาให้สถานีนี้
Msg.Receive.Broadcast.Datagram	รับข้อมูลส่งแบบกระจายมาให้

ตารางที่ 5.5 คำสั่งเกี่ยวกับ Datagram Service ของ NETBIOS (Martin 1989)

5.5.2.3 NETBIOS เซสชัน

เราสามารถสร้างเซสชัน ในระหว่างชื่อ 2 ชื่อ ในเน็ตเวิร์คได้ และเรายังสามารถสร้างเซสชั่นหลายๆเซสชัน ในระหว่าง 2 ชื่อก็ได้ นอกจากนี้เรายังสามารถสร้างแม้กระทั่งเซสชัน ในระหว่าง 2 ชื่อบนสถานีเดียวกันก็ได้ ในขณะที่เซสชันกำลังแอคทีฟอยู่นั้นจะรับประกันได้ว่าจะมีการรับส่งข้อมูลไปยังสถานีที่ถูกต้อง และในลำดับที่ถูกต้องด้วย

ในการสร้างเซสชัน นั้นสถานีหนึ่งจะต้องเรียกคำสั่ง listen (ไม่ว่าจะเป็นแบบ คอย หรือ ไม่คอย) ซึ่งคำสั่งนี้จะเป็นการให้รอรับการเรียก(call) จากสถานีใดสถานีหนึ่ง แล้วก็จะมีเวิร์คสเตชันอีกตัวหนึ่งเรียกคำสั่ง call ซึ่งหมายถึงว่า 'ชื่อ' ตัวนั้นเริ่มทำการเรียก และเมื่อมีคำสั่งในแต่สถานีสมบูรณ์แล้ว NETBIOS จะทำการส่งค่า LSN (Local Session Number) ไปให้แต่ละโปรแกรม

ตารางที่ 5.6 การโต้ตอบในระดับ NETBIOS ของสองเวิร์คสเตชั่น (Nance 1990)

ก. แบบ ดาต้าแกรม

เวิร์คสเตชั่น A	เวิร์คสเตชั่น B
1. Add Name A	1. Add Name B
2. Send message to B	2. Receive a message
3. Recieve a message	3. Send message to A
4. Delete Name A	4. Delete Name B

ข. แบบ เซสชั่น

เวิร์คสเตชั่น A	เวิร์คสเตชั่น B
1. Add Name A	1. Add Name B
2. Listen (Wait for call)	2. Call workstation A
3. Send a message to B	3. Recieve a message
4. Receive a message	4. Send a message to A
5. Hang up on B	5. Hang up on A
6. Delete Name A	6. Delete Name B

ประยุกต์แต่ละตัว (LSN ที่ส่งไปให้เวิร์คสเตชันแต่ละตัวนั้นไม่จำเป็นจะต้องเหมือนกัน) ดังนั้น แต่ละเวิร์คสเตชันก็สามารถนำ LSN มาใช้ในการอ้างในการทำ open ,active session ได้ การส่งข้อมูลแบบเซสชัน นี้สามารถส่งข้อมูลได้ยาวถึง 65,535 ไบต์ ซึ่งคำสั่งที่ต้องใช้คือ send, chain send (ซึ่งใช้ในการส่งข้อมูลแบบเป็นชุดๆต่อเนื่องกัน) และ receive ในตอนสุดท้ายของการสนทนากันนั้นเวิร์คสเตชันทั้งสองตัวต้องสั่งคำสั่ง hang up ทั้งสองด้านเพื่อเป็นการปิดเซสชัน

คำสั่ง	ฟังก์ชัน
Msg.Call	เปิดเซสชันโดยชื่ออื่น
Msg.Listen	ทำให้เซสชันเปิดโดยชื่อที่กำหนดในคำสั่ง
Msg.Hang.Up	ปิดเซสชันโดยชื่ออื่น
Msg.Send	ส่งข้อมูลไปยังคู่สนทนาในเซสชัน
Msg.Chain.Send	ส่งข้อมูลไปยังคู่สนทนาในเซสชันโดยใช้สองบัฟเฟอร์ซึ่งต่อกันเป็นแบบลูกโซ่
Msg.Recieve	รับข้อมูลจากคู่สนทนาในเซสชัน
Msg.Receive.Any	รับข้อมูลจากคู่สนทนาในเซสชันใดๆ
Msg.Session.Status	รับค่าสถานภาพของเซสชันจากเซสชันทั้งหมดหรือจากชื่อที่กำหนด

ตารางที่ 5.7 คำสั่ง Session Service ของ NETBIOS (Martin 1989)

5.5.2.4 การบริการทั่วไปของ NETBIOS

ในการให้บริการอย่างใดอย่างหนึ่งของ NETBIOS โดยเฉพาะนั้น NETBIOS ก็ได้มีคำสั่งช่วยในการควบคุมและสนับสนุนการทำงานในจุดประสงค์ต่างๆ ตามที่แสดงในตาราง ที่ 5.8

คำสั่ง	ฟังก์ชั่น
Msg.Reset	รีเซ็ต NETBIOS อินเตอร์เฟส
Msg.Cancel	ยกเลิกคำสั่ง
Msg.Status	รับค่าสถานภาพของ NETBIOS อินเตอร์เฟส
Msg.Unlink	มีไว้สำหรับให้เข้ากันได้กับ NETBIOS รุ่นแรกๆ
Msg.Trace	ทำการติดตามการทำงานของคำสั่งต่างๆที่ส่งไปยัง NETBIOS อินเตอร์เฟส และ NETBIOS recieve และ transmit

ตารางที่ 5.8 คำสั่งทั่วไปของ NETBIOS (Martin 1989)

5.5.3 โครงสร้างและการใช้งาน NCB

ในการเรียกคำสั่ง NETBIOS นั้นตัวโปรแกรมประยุกต์จะต้องมีการสร้าง NCB แล้วทำการเรียกอินเตอร์รัปต์ 5CH ในรูปที่ 5.16 จะแสดงถึงโครงสร้างของ NCB โดยแต่ละฟิลด์มีรายละเอียดดังนี้

เราต้องเซต ฟิลด์คำสั่งขนาด 1 ไบต์ เพื่อจะบอกว่าเราจะเรียกคำสั่งไหนมาใช้งาน และ ถ้าชุดบิตสูงถูกเซต คำสั่งนั้นจะเป็นแบบ 'ไม่คอย' ฟิลด์ return code มีขนาด 1 ไบต์จะเก็บค่า immediate error code ซึ่งจะเซตโดย NETBIOS เมื่อเริ่มเอ็กซีคิวต์คำสั่ง

หลังจากคำสั่ง listen หรือ call เริ่มทำงานแล้ว ฟิลด์ LSN ขนาด 1 ไบต์จะบรรจุค่า LSN ที่กำหนดให้กับเซสชัน นั้นและเมื่อเราใช้คำสั่ง send หรือ receive เราต้องใส่ค่า LSN สำหรับเซสชันนั้นลงในฟิลด์นี้

NETBIOS จะส่งกลับค่าในฟิลด์ name number ซึ่งมีขนาด 1 ไบต์ หลังจากสั่งคำสั่ง add name หรือ add group ซึ่งเราใช้ตัวเลขนี้แทนชื่อถ้าเราใช้คำสั่งที่เกี่ยวกับดาต้าแกรม หรือคำสั่ง receive any

ฟิลด์ data buffer address จะบรรจุค่าตัวชี้แบบไกล (Far pointer)ขนาด 4 ไบต์(segment:offset)ชี้ไปยังบัฟเฟอร์ข้อมูล ที่ใช้กับคำสั่ง send หรือ receive

ฟิลด์ data buffer length มีขนาด 2 ไบต์ เพื่อเก็บความยาวของบัฟเฟอร์ข้อมูล

เราต้องเซตค่า call name ฟิลด์ซึ่งมีขนาด 16 ไบต์เพื่อที่จะบอกถึงชื่อของเวิร์คสเตชันอื่นที่เราต้องการที่จะติดต่อด้วย

เราต้องเซตค่า local name ฟิลด์ซึ่งมีขนาด 16 ไบต์เพื่อจะระบุว่าเราจะใช้ชื่อไหนในตารางชื่อโลคอล(หรือ หรือโหนดถาวร) ที่จะให้โปรแกรมประยุกต์รับรู้เพื่อใช้ในการอ้างถึง

เมื่อเราสั่งคำสั่ง call หรือ listen เราต้องเซตฟิลด์ receive timeout ที่มีขนาด 1 ไบต์เป็นค่าที่จะบอกว่าเราจะตั้งค่า การดอย ไว้กี่ช่วง ครึ่งวินาที ในการรอคำสั่ง receive ทำจนสมบูรณ์ ถ้าตั้งค่าไว้เป็น 0 แสดงว่าไม่มีการตั้งค่า time-out

เมื่อเราสั่งคำสั่ง call หรือ listen เราต้องเซตฟิลด์ send timeout ที่มีขนาด 1 ไบต์เป็นค่าที่จะบอกว่าเราจะตั้งค่า การดอย ไว้กี่ช่วงครึ่งวินาที ในการรอคำสั่ง send ทำจนสมบูรณ์ ถ้าตั้งค่าไว้เป็น 0 แสดงว่าไม่มีการตั้งค่า time-out ฟิลด์ POST routine address มีขนาด 4 ไบต์จะเป็นตัวชี้ระยะไกล ที่ชี้ไปยังแอดเดรสของรูทีนที่จะไปทำหลังจาก NETBIOS ทำงานเสร็จแล้ว(คำสั่งนี้จะมีความหมายก็ต่อเมื่อใช้คำสั่งแบบ 'ไม่ดอย') ถ้าเราเซตฟิลด์นี้เป็น 0 (null) เราควรจะทำการหยั่งเสียง final return code เพื่อที่จะตรวจสอบว่าเมื่อไรที่ทำคำสั่งเสร็จและเสร็จสมบูรณ์หรือไม่

เราเซตค่า adapter number ขนาด 1 ไบต์เพื่อจะบอกว่าเราจะใช้เน็ตเวิร์คอินเตอร์เฟสการ์ดตัวไหน(0 สำหรับ primary 1 สำหรับ alternate ทั้งนี้เพราะว่าในแต่ละเวิร์คสเตชันสามารถใส่การ์ดโทเกนริงได้ถึง 2 การ์ด)

ฟิลด์ final return code ขนาด 1 ไบต์จะบรรจุค่า 0xFF ในขณะที่คำสั่งกำลังทำงานอยู่ หลังจากคำสั่งทำงานเสร็จแล้วฟิลด์นี้จะถูกเซ็ตค่าเพื่อแสดงค่าว่าคำสั่งนั้นทำสำเร็จหรือไม่

ที่สำรอง ขนาด 14 ไบต์จะไม่ใช้โดยโปรแกรมประยุกต์แต่ตัว NETBIOS จะเก็บไว้ใช้โดยเฉพาะ

	ความยาว (ไบต์)
Command ID	1
Immediate return code	1
Local Session number	1
Network name number	1
Address of data/message	4
Length of data/message	2
Remote computer name	16
Our computer name	16
Recieve time-out	1
Send time-out	1
Address or POST routine	4
Adapter number	1
Final return code	1
Reserved area	14

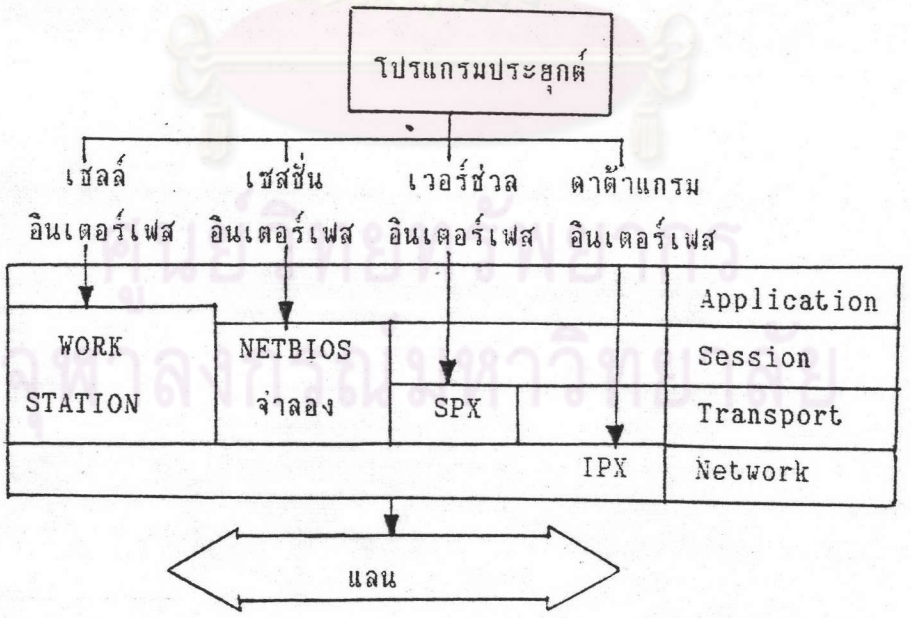
รูปที่ 5.16 รูปแบบของ NCB (Nance 1990)

5.4.5 NETBIOS เฟรม

NETBIOS เฟรมจะถูกสร้างและควบคุมโดย NETBIOS โดยเฉพาะ เราไม่ต้องจัดการโดยตรงกับมันและในการสร้างเฟรมนี้นั้น NETBIOS จะใช้ค่าที่ได้จาก NCB ที่เรากำหนดไว้รวมทั้งค่าต่างๆที่ได้จาก เซสชัน และ ชื่อ ที่กำลังใช้งานอยู่ มาประกอบเป็นเฟรมต่างๆอีกทีหนึ่ง เฟรมต่างๆนี้จะใช้สำหรับ NETBIOS โดยเฉพาะ ตัวอย่างเช่น name query frame, add name query frame, name recognized frame, session initialize frame เป็นต้น

5.6 โครงสร้างของ Novell NetWare และการติดต่อกับโปรแกรมระดับสูง

5.6.1 การใช้งาน SPX/IPX ของ เน็ตแวร์(NetWare)



รูปที่ 5.17 การอินเตอร์เฟสในระดับต่างๆของเน็ตแวร์ (Burdge 1989)

เวิร์คสเตชันที่รันภายใต้ดอสติดต่อกับเน็ตแวร์ ไฟล์เซิร์ฟเวอร์โดยผ่านทางโปรแกรมที่ฝังตัวในหน่วยความจำสองโปรแกรม ตัวแรกก็คือ IPX.COM ซึ่งจะส่งและรับแพกเก็ต IPX(Inter NetworkPacket Exchange) นี้เทียบได้ว่าอยู่ใน ชั้นเน็ตเวิร์คของ OSI ส่วนโปรแกรมตัวที่สองคือ NET3.COM ซึ่งเป็นเชลล์ที่จะตรวจสอบดูว่าโปรแกรมประยุกต์ที่เรียกนั้นเรียกใช้DOS หรือเน็ตเวิร์ค ในกรณีที่เป็นการเรียกจากเน็ตเวิร์ค ตัวเชลล์จะสร้างแพกเก็ตซึ่งจะบรรจุสิ่งที่ต้องการและส่งไปยัง IPX เพื่อส่งไปในเน็ตเวิร์ค โปรแกรม NET3 นี้จัดอยู่ในชั้นแอปลิเคชัน และ พรีเซ้นเตชัน ใน OSI

IPX เป็น การบริการของดาต้าแกรม ของเน็ตแวร์ โดยพัฒนามาจากมาตรฐาน Xerox Network Standard(XNS) ของบริษัท Xerox IPX จะทำงานได้แบบดาต้าแกรม การบริการนี้เทียบได้กับชั้นเน็ตเวิร์ค ของ OSI โมเดลโดยที่โพรโตคอลนี้จะให้บริการเกี่ยวกับ การให้แอดเดรส,การให้เส้นทาง และ การสวิตชิง ในการที่จะส่งแพกเก็ตข้อมูลไปยังปลายทาง การส่งข้อมูลแบบดาต้าแกรมนั้นจะทำให้การส่งทำได้เร็วกว่าแบบ เซสชันแต่จะไม่มีการรับประกันข้อมูลที่ส่ง แต่โนเวลคุยว่าแพกเก็ตจะส่งถูกต้องถึง 95 เปอร์เซ็นต์

ในเน็ตแวร์ จะมี IPX อยู่ในตัวแล้วและในบางรุ่นอาจจะมี SPX ให้มาด้วย

5.6.2 IPX แอดเดรสปลายทาง

แอดเดรสปลายทาง ใน IPX จะเป็น internetwork address ซึ่งประกอบด้วย ส่วนต่างๆ 3 ส่วนคือ network number, node address และ socket

network number : จะเป็นตัวแยกแต่ละส่วน ของเน็ตเวิร์คที่มีหลายเซิร์ฟเวอร์ ผู้ที่กำหนดเลขนี้ก็คือ ผู้ควบคุมระบบ ตั้งแต่ตอนทำขบวนการเริ่มต้นเน็ตเวิร์คครั้งแรก

node address : จะเป็นตัวที่ไม่ซ้ำกัน เพื่อจะเป็นตัวแยกแยะแต่ละเน็ตเวิร์คอินเตอร์เฟสการ์ด

socket : จะเป็นตัวแทนของโปรแกรมประยุกต์ บนปลายทางจะรันบนเวิร์คสเตชันปลายทาง โปรแกรมประยุกต์จะเปิด และปิดซ้อกเก็ต(Socket) เช่นเดียวกับการเปิด และปิดไฟล์ แอดเดรสปลายทาง จะบรรจุ แอดเดรสกลุ่ม แต่ SPX นั้น ฟิลด์แอดเดรสปลายทาง จะเก็บค่าแอดเดรส ของเวิร์คสเตชันใดเวิร์คสเตชันหนึ่งโดยเฉพาะ ถ้าเราต้องการที่จะทำการติดต่อกับเวิร์คสเตชันใดโดยเฉพาะ เราสามารถใช้บริการของ IPX ในการที่จะหาแอดเดรสปลายทางของเวิร์คสเตชันนั้น เราจะส่ง IPX แพกเก็ตโดยกำหนด แอดเดรสปลายทาง(network, node, socket) แต่ในการรับนั้นจะทำโดยการกำหนดซ้อกเก็ต

5.6.3 การให้บริการในการโปรแกรมของ IPX

เราสามารถใช้บริการ IPX เพื่อทำการสื่อสารระหว่าง PC โดยที่ IPX มีการให้บริการต่างๆ โดยที่การให้บริการนั้นจะมีโครงสร้างข้อมูล 3 แบบใหญ่ๆดังนี้

IPX header : ประกอบด้วย IPX แพกเก็ตขนาด 30 ไบต์

data record หรือ message : คือข้อมูล ที่เราต้องการจะส่งหรือรับนั่นเอง

Event Control Block (ECB) : จะไม่ถูกส่งจริงๆ แต่จะบรรจุข้อมูลของ IPX operation ที่เราเซตในครั้งแรก

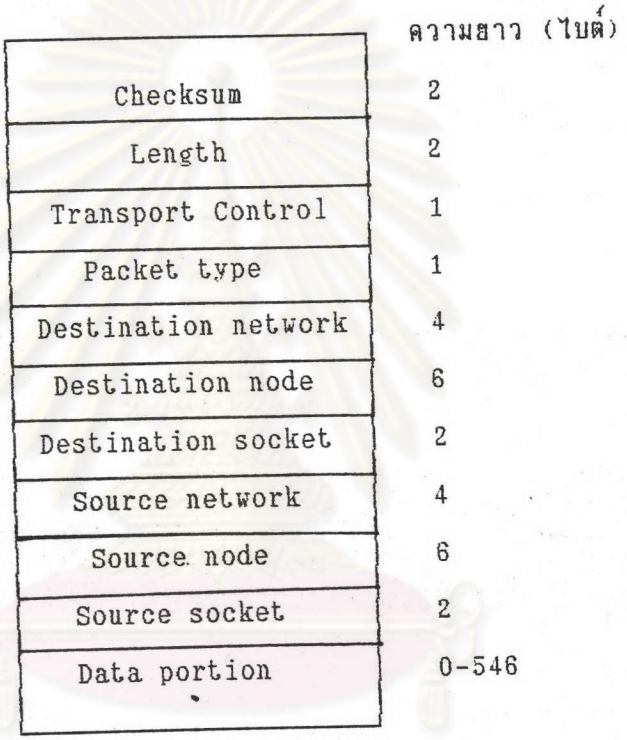
5.6.4 ฟังก์ชันในการทำขบวนการเริ่มต้นและขบวนการสิ้นสุด

เมื่อ PC สองตัวบนเน็ตเวิร์คส่งข้อมูลไปมาโดยใช้ IPX ตัวโปรแกรมประยุกต์บนทั้งสองเวิร์คสเตชันจะต้องทำการเปิดซ้อกเก็ต ก่อนโดยเรียกฟังก์ชัน IPXOpenSocket socket number ซึ่งเปิดที่บนเวิร์คสเตชัน A จะต้องรู้ไปถึงโปรแกรมประยุกต์ที่วิ่งบนเวิร์คสเตชัน B และในทางกลับกันด้วย ในเวิร์คสเตชันสทั้งสองตัวต้องรู้แอดเดรสปลายทางของแต่ละตัว

5.6.5 รูปแบบ IPX แพ็กเก็ต

ในรูปที่ 5.18 แสดงถึงรูปแบบของ IPX แพกเก็ตโดยที่แพกเก็ตนี้เป็นรายการข้อมูล ที่จะใส่ลงไปในอีเทอร์เน็ตหรือโทเกนริงเฟรม ในแต่ละฟิลด์ของ IPX

แพกเก็ตประกอบด้วย



รูปที่ 5.18 รูปแบบของ IPX แพกเก็ต (Nance 1990)

checksum : มีขนาด 2 ไบต์ แต่เราไม่ได้ใช้เพราะว่าโพรโตคอลระดับต่ำได้ทำการตรวจสอบความผิดพลาด ไว้หมดแล้วแต่ที่ยังคงมีฟิลด์นี้อยู่ก็เพราะต้องการทำ ให้ตรงต้นแบบของ XNS (IPX จะเซตเป็น 0xFFFF)

length : ขนาดของแพกเก็ตทั้งหมดรวมทั้งส่วนหัวของ IPX และส่วน ข้อมูล ขนาดของแพกเก็ตเล็กที่สุดคือ 30 ไบต์ (เฉพาะส่วนหัวของ IPX อย่าง

เดียว) และขนาดใหญ่ที่สุดคือ 576 ไบต์ (ส่วนหัว IPX 30 ไบต์ รวมกับข้อมูล 546 ไบต์) IPX จะคำนวณตัวเลขนี้เมื่อเราส่งข้อมูลที่จะส่งให้ IPX โดยที่เราไม่ต้องไปเซตค่า

Transport Control : IPX แพกเก็ตสามารถส่งผ่านข้ามได้ถึงบริดจ์ได้ 16 ตัว และIPX จะเซตค่าในฟิลด์นี้เพิ่มขึ้นทีละ 1 เมื่อส่งผ่านบริดจ์ครั้งหนึ่ง และเมื่อถึง 16 แล้วมันก็จะไม่สนใจแพกเก็ตนั้น

Packet Type : จะแสดงถึงประเภทของแพกเก็ตนั้นถ้าเป็น IPX แพกเก็ตจะเซตค่าเป็น 4 แต่ถ้าเป็น SPX จะเซตเป็น 5

Destination Network : มีขนาด 4ไบต์จะระบุถึงเวิร์คสเตชันปลายทางที่จะส่งข้อมูลไปให้

Destination Node : จะบอกถึง แอดเดรสทางกายภาพของสถานีปลายทาง

Destination socket :จะบรรจุ หมายเลขซ็อกเก็ต โดยที่ซ็อกเก็ต ต้องถูกเปิดโดย โปรแกรมประยุกต์ที่สถานีปลายทางเสียก่อน

Source Network : เช่นเดียวกับ destination network

Source Node : เช่นเดียวกับ destination node

Source Socket : เช่นเดียวกับ destination socket

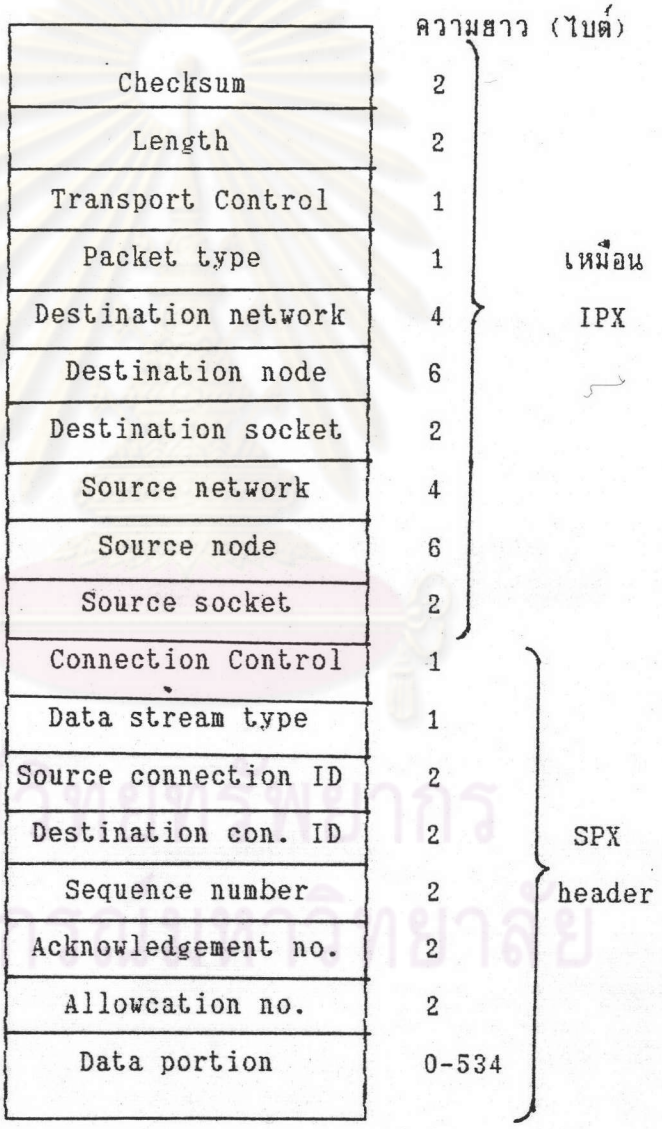
DataPortion : เป็นส่วนข้อมูลที่โปรแกรมประยุกต์ ต้องการจะส่งซึ่งจะมีความยาวได้ถึง 546 ไบต์ ถ้าข้อมูลมีความยาวเป็น 0 อาจใช้ในความหมายพิเศษเช่น เป็นการ ตอบรับ แพกเก็ตที่ได้รับก่อนหน้านั้น

5.6.6 SPX การสื่อสารแบบเซสชัน

SPX ย่อมาจาก Sequence Packet Exchange เป็นการสื่อสารแบบเชื่อมต่อ (Connection oriented) โพรโตคอล คือก่อนการส่งและรับจะต้องมีการสร้างเซสชัน ขึ้นมาก่อน เมื่อสร้างขึ้นมาแล้ว แมสเซสก็สามารถส่งไปมาได้ทั้งสองด้าน โดยจะมีการรับรองว่าข้อมูลจะได้รับถูกต้อง IPX นั้นให้บริการในระดับชั้นทรานสปอร์ต แต่ SPX จะให้บริการสูงไปอีกชั้นคือชั้น เน็ตเวิร์ค แล้วก็ยังมีลักษณะบาง

ประการของชั้นเซสชัน ด้วย

5.6.7 รูปแบบของ SPX แพกเก็ต



รูปที่ 5.19 รูปแบบของ SPX แพกเก็ต (Nance 1990)

SPX แพกเก็ตนั้นประกอบด้วย IPX packet header ขนาด 30 ไบท์ตามด้วยส่วนหัวของ SPX รวมทั้งหมดเป็น 52 ไบต์ โดยที่ 30 ไบต์ ของส่วนหัว IPX มีความหมายเหมือนกันในส่วน IPX ของ SPX เพียงแต่ว่าในฟิลด์ packet type จะเซตค่าเป็น 5 แทน ในฟิลด์ส่วนที่เป็น SPX นั้นประกอบด้วยส่วนต่างๆดังนี้

connection control : มีรูปแบบบิตดังนี้

S A ! E XXXX

โดยที่ S คือ system packet บิต

A คือ acknowledgement required บิต

! คือ attention บิต

E คือ end-of-message บิต

XXXX คือไม่ได้กำหนดค่าไว้

datastream type : เราสามารถกำหนดค่า ตั้งแต่ 0-253 (0xFD)ในฟิลด์ ให้แก่แต่ละแพกเก็ต โดยที่ SPX จะไม่เซตค่าในช่วงนี้ และมันจะใช้ค่าตัวเลข 254,255 ดังนี้ เมื่อเวิร์คสเตชัน A บอก SPX ให้ยกเลิกการติดต่อ ตัว SPX ก็จะส่งแพกเก็ตสุดท้ายไปให้เวิร์คสเตชัน B โดยในฟิลด์นี้จะมีค่าเป็น 254 เวิร์คสเตชัน B ก็จะรู้ว่าเวิร์คสเตชัน A เลิกการติดต่อ และสำหรับแพกเก็ตที่เรียกใช้กันภายใน SPX มันจะเซตค่า ในฟิลด์นี้เป็น 255

source connection ID : SPX จะเซตค่าฟิลด์นี้ซึ่งมีขนาด 2 ไบต์ เพื่อ แสดงถึงค่า connection ID ของ SPX เซสชันนั้นที่เวิร์คสเตชันต้นทาง

destination connection ID : เช่นเดียวกับฟิลด์ที่แล้ว แต่เป็นของเวิร์คสเตชันปลายทาง และถ้ามีการทำ การติดต่อพร้อมๆกันหลายชุด ในซ้อกเก็ตเดียว SPX จะใช้ฟิลด์นี้เพื่อจะติดตาม message ว่าฟิลด์ ไหนใช้กับ connection ID ไหน

sequence number : จะถูกเซตโดย SPX เช่นกัน โดยที่ SPX จะใช้ฟิลด์นี้ในการแยกแยะและทิ้งแพกเก็ตที่ซ้ำซ้อนกัน

acknowledge number : ฟิลด์นี้จะถูกเซตโดย SPX จะติดตาม sequence number ของแพกเก็ตต่อไปที่ SPX คาดว่าจะได้รับใน connection ID นั้น

allocation number : ฟิลด์นี้จะถูกเซตโดย SPX และ SPX จะใช้ฟิลด์นี้เพื่อนับจำนวนของแพกเก็ตที่ถูกส่งออกมา แต่ไม่ได้ถูกรับโดยเวิร์คสเตชันอื่น

data portion : ขนาดมากสุดของ SPX แพกเก็ตคือ 534 ไบต์ซึ่งจะมีขนาดน้อยกว่าของ IPX เล็กน้อย

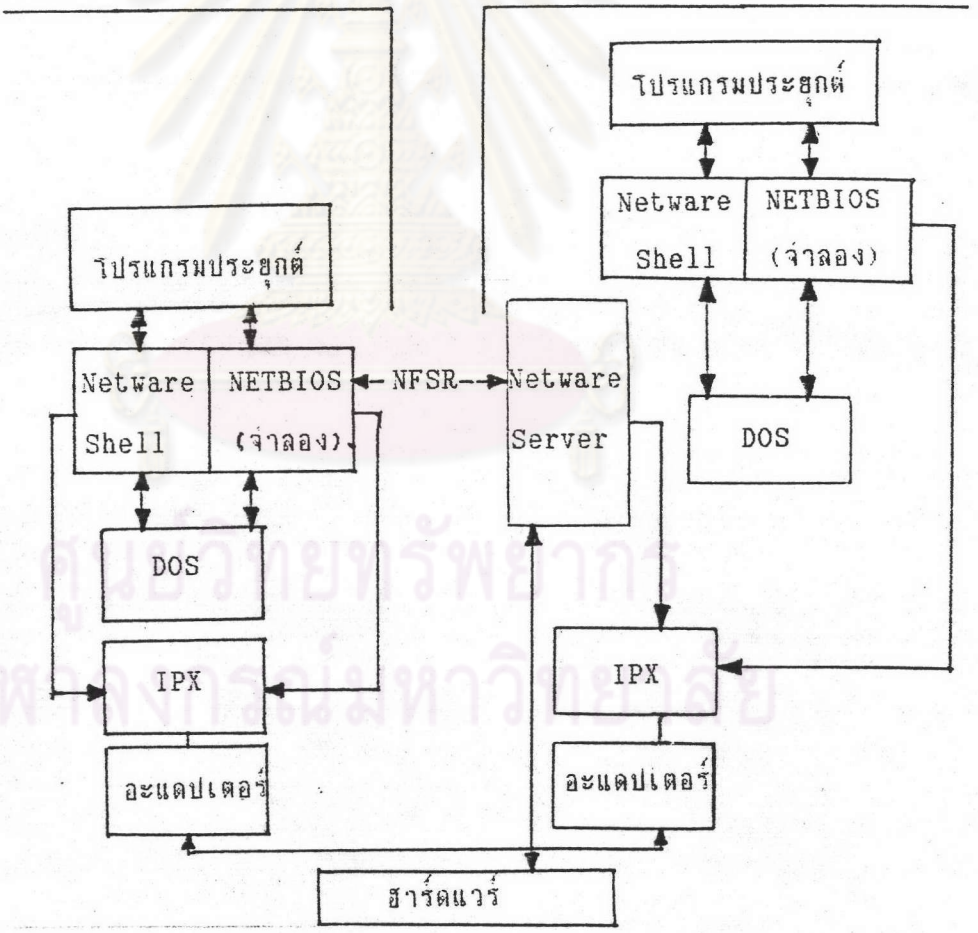
5.6.8 โครงสร้างของแอดวานซ์เน็ตแวร์

จากรูปที่ 5.21 เชลล์ (Shell) จะเป็นตัวทำฟังก์ชันพื้นฐานที่จำลองการทำงานของ รีไดเร็กเตอร์ (Redirector) และ SMB โพรโตคอล แต่จะมีข้อแตกต่างกันระหว่างเชลล์ของแอดวานซ์เน็ตแวร์ กับ รีไดเร็กเตอร์ ของ IBM โดยที่ตัวรีไดเร็กเตอร์ นั้นการอินเตอร์รัปท์จะผ่าน DOS ก่อน และ DOS จะผ่านอินเตอร์รัปต์เบอร์ 21H มาให้ รีไดเร็กเตอร์ ซึ่งมันจะจัด SMB แล้วส่งผ่านโดย NETBIOS เพื่อส่งผ่านเข้าเน็ตเวิร์คอีกทีหนึ่ง แต่การใช้แอดวานซ์เน็ตแวร์นั้นอินเตอร์รัปต์ทั้งหมดจะถูกดักโดย เชลล์และถ้าตัวไม่เกี่ยวกับเน็ตเวิร์คมันก็จะส่งไปให้ดอส ทำงานแต่ถ้าเกี่ยวกับเน็ต-เวิร์คมันก็จะสร้าง NFSP (Network File Service Protocol) แล้วส่งไปยัง IPX โพรโตคอลสำหรับส่งไปในเน็ตเวิร์ค การอินเตอร์รัปต์ที่ส่งผ่านโดยตรงไปยัง NETBIOS จะถูก ปฏิบัติ โดย ฟังก์ชันของ NETBIOS จำลองและจะส่งผ่านไปในเน็ตเวิร์คโดย IPX เช่นเดียวกัน

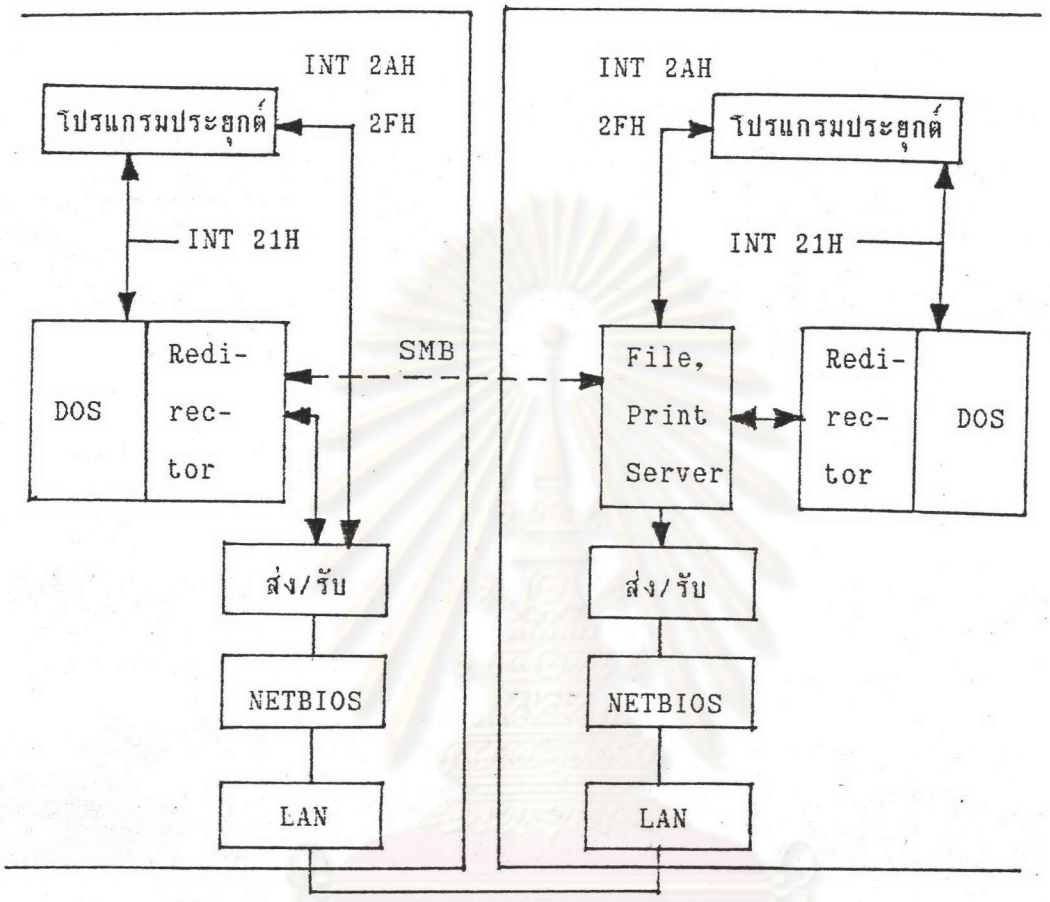
5.7 IBM Redirector/SMB อินเตอร์เฟส (Martin 1989)

IBM PC LAN Program นั้น เป็นโปรแกรมในระดับชั้นแอปลิเคชัน ของผลิตภัณฑ์ IBM แลนซึ่งในระดับนี้จะให้บริการที่เป็นประโยชน์โดยตรงแก่ โปรแกรมประยุกต์ที่ใช้งานกับแลน การบริการนี้รวมทั้ง การทำ แชร์พริ้นเตอร์, แชร์ไฟล์ ,การส่งผ่านไฟล์ ,อิเลคทรอนิคส์เมล์ ซึ่งรูปที่ 5.21 จะแสดงโครงสร้างการติดต่อระหว่างไฟล์เซิร์ฟเวอร์และเวิร์คสเตชัน

ผู้ใช้สามารถอินเตอร์เฟสกับฟังก์ชันเหล่านี้ โดยการเรียกใช้จากเมนูโดยตรง หรือสั่งจากจุดป้อนคำสั่งของ DOS ส่วนตัวโปรแกรมประยุกต์สามารถเรียกฟังก์ชันเหล่านี้ โดยสร้าง SMB (Server Message Block) ซึ่งเป็นโครงสร้างข้อมูล แล้วเรียกอินเตอร์รัปต์ 21H ตัว รีไดเร็กเตอร์ ฟังก์ชันจะตรวจสอบว่าการเรียกอินเตอร์รัปต์นั้นเป็นการเรียกเซิร์ฟเวอร์หรือเป็นการเรียกตัวโลคอล ถ้าเป็นการเรียกใช้เซิร์ฟเวอร์มันก็จะสร้าง SMB ส่งผ่านเน็ตเวิร์คไป โดยที่มันจะไปเรียกใช้บริการ NETBIOS อีกทีหนึ่ง ผลของการร้องขอนี้จะส่งกลับไปยังสถานีต้นทางโดยผ่าน SMB เช่นเดียวกัน 2A และ2F จะเป็นอินเตอร์รัปต์โดยตรงที่มีฟังก์ชันเพื่อทำการควบคุมเน็ตเวิร์ค



รูปที่ 5.20 โครงสร้างของเน็ตแวร์ (Martin 1989)



รูปที่ 5.21 การทำงานของเน็ตเวิร์คระหว่างเซิร์ฟเวอร์และเวิร์คสเตชั่น (Martin 1989)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.8 เฟรมต่างๆที่ใช้ในการติดต่อของโพรโตคอลระดับเดียวกัน (Peer-to-peer)

เฟรมที่ใช้ติดต่อกันในโทเกนริง แบ่งเป็น MAC เฟรมและ non-MAC เฟรม โดยจะบอกในฟิลด์ FC ถ้าเป็น MAC เฟรมแล้วในส่วนฟิลด์ข้อมูลจะบรรจุคำสั่งต่างๆในระดับ MAC ของโทเกนริง ซึ่งจะเป็นการจัดการเกี่ยวกับเรื่องต่างๆบนริงนั่นเอง ส่วน non-MAC เฟรมนั้นในส่วนฟิลด์ข้อมูลจะบรรจุ LLC เฟรมพร้อมกับข้อมูลที่ส่งผ่านมาจากโพรโตคอลระดับสูงขึ้นไป

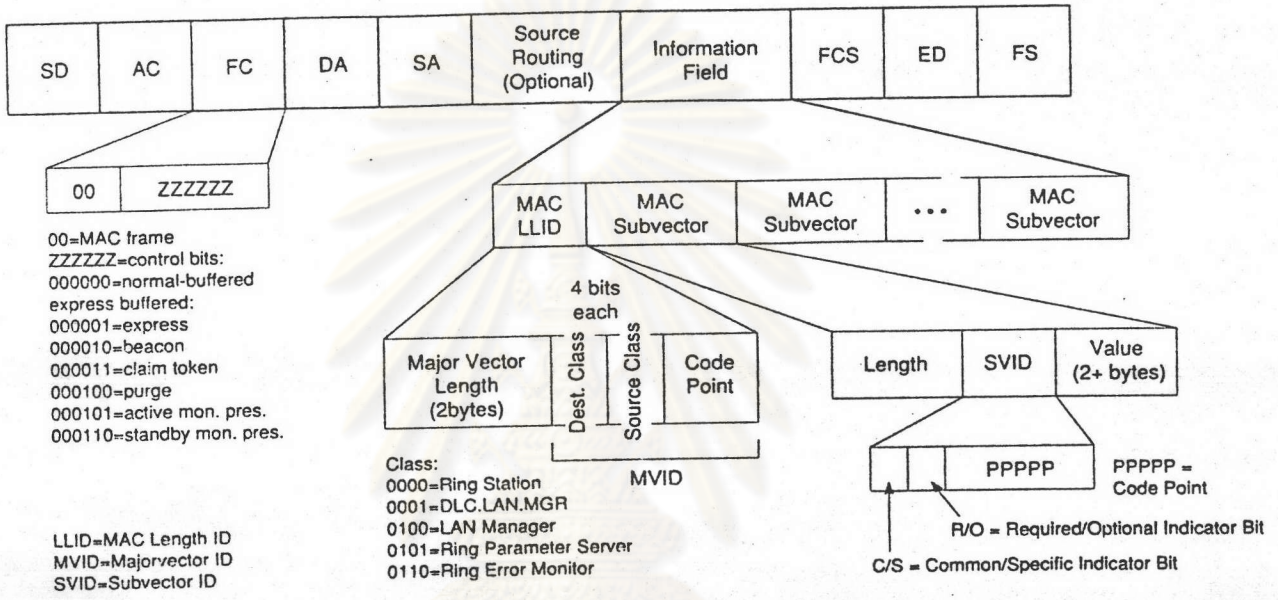
5.8.1 ระดับ MAC ของโทเกนริง (Miller 1990; Halsall 1989; TMS380 Adapter Chipset User's Guide 1986)

ปรกติจะเป็น โทเกนริงเฟรม โดยที่ในฟิลด์ FC บอกว่าเป็น MAC เฟรม ใน MAC เฟรมจะประกอบไปด้วยฟิลด์ต่างๆดังรูป Length ฟิลด์จะบอกความยาวของ Major Vector MAC เฟรมทั้งหมด ,Class ฟิลด์จะบอก class ของสถานีต้นทางและปลายทาง (โดยที่ประเภทของ class คือ Ring station, LLC Manager, Network manager, Ring Parameter Server, Ring Error Monitor) , ฟิลด์คำสั่งของ MAC ซึ่งเรียกว่า Code point ซึ่งจะเป็นตัวบอกว่า จะให้ตัวรับทำอะไรบ้าง ทั้งหมดนี้เรียก Major Vector MAC (ซึ่งประกอบด้วย Major Vector length และ Major Vector ID)

ส่วนที่ต่อท้าย MAC Major Vector จะเป็น MAC Subvector ซึ่งอาจมีหลายๆ Subvector ต่อกันไปเรื่อยๆได้ ส่วนประกอบในแต่ละฟิลด์ใน MAC Subvector คือ Length ฟิลด์ ซึ่งจะบอกความยาวของ Subvector นั้นๆ, Subvector ID ฟิลด์ จะบอกถึงประเภทของข้อมูลในส่วนที่ต่อมา, ฟิลด์ข้อมูล จะเป็นข้อมูลที่ใช้โดย Subvector นั้นๆ

MAC เฟรมนั้นจะแบ่งได้ 25 ชนิดตาม คำสั่งที่อยู่ใน MVID โดยจะแบ่งเป็นหมวดหมู่ได้ 4 ประเภท คือ

1. Medium Control
2. Station initialization



MVID Code Point:

00H=Response
02H=Beacon
03H=Claim Token
04H=Ring Purge
05H=Active Mon. Pres.
06H=Standby Mon. Pres.
07H=Duplicate Addr. Test
08H=Lobe Test
09H=Transmit Forward
0BH=Remove Ring Station
0CH=Change Parameters
0DH=Initialize Ring Stn.
0EH=Request Ring Stn. Addresses
0FH=Request Ring Stn. State
10H=Request Parameters
20H=Request Initialization
22H=Report Ring Stn. Addresses
23H=Report Ring Stn. State
24H=Report Ring Stn. Attachments
25H=Report New Active Monitor
26H=Report NAUN change
27H=Report Neighbor Notif. Incomplete
28H=Report Active Monitor Error
29H=Report Soft Error
2AH=Report Transmit Forward

SVID Code Point:

01H=Beacon Type
02H=NAUN
03H=Local Ring Number
04H=Assign Physical Location
05H=Soft Error Report Timer Value
06H=Enabled Function Classes
07H=Allowed Access Priority
09H=Correlator (for response)
0AH=Addr. of Last Neigh. Notification
0BH=Physical Location
20H=Response Code
21H=Reserved
22H=Product Instance ID
23H=Ring Station Microcode Level
26H=Wrap Data
27H=Frame Forward
28H=Station Identifier
29H=Ring Station Status Vector
2AH=Transmit Status Code
2BH=Group Address
2CH=Functional Address
2DH=Isolating Error Counts
2EH=Non-Isolating Error Counts
30H=Error Code

รูปที่ 5.22 แสดงรายละเอียดต่างๆของ MAC เฟรม (Miller 1990)

3. Error monitoring

4. Network Management

MAC เฟรมต่างๆเหล่านี้จะถูกส่งผ่านไปมาในระหว่างสถานีต่างๆ จากสถานีซึ่งมี Class เป็น RS (Ring station) , AM (Active Monitor), RPS (Ring Parameter Server), REM (Ring Error Monitor) และ CRS (Configuration Report Server) ซึ่งการอ้างอิงถึงสถานีที่มี Class ต่างๆนี้จะอ้างอิงแบบ ฟังก์ชัน ซึ่งเรียกว่า ฟังก์ชันแอดเดรส(Functional Address) แทนการอ้างถึงแอดเดรสจริงๆ(Physical Address)ปรกติการตั้งค่าให้สถานีไหนมีฟังก์ชันแอดเดรสแบบไหนทำตอนสั่ง OPEN_ADAPTER ครั้งแรก

ค่า MAC Subvector จะใช้ประกอบกับ Major Vector เพื่อบอกรายละเอียดต่างเพิ่มเติมให้ Major vector เปรียบเสมือนเป็นพารามิเตอร์นั่นเอง ซึ่งในส่วน Subvector Code Point จะบอกชนิด ของ Subvector นั้นส่วนตัวข้อมูลที่จะใช้จริงๆจะอยู่ในส่วนข้อมูลของ Subvector นั่นเอง เช่น คำสั่ง Major Vector เป็น 'Initialize Ring' ตัว Subvector ที่จะตามมา ก็จะมีชนิดที่ Code Point เป็น 'Local Ring Number' อยู่ด้วย โดยที่ส่วนข้อมูลของ Subvector ก็จะเป็น หมายเลขของริงของอะแดปเตอร์ที่จะส่ง นั่นเอง

การส่งและรับเฟรม และโทเกน ในริง โดยมีระบบการจัดลำดับความสำคัญนั้นได้กล่าวถึงไว้ในบทที่ 3 แล้ว ส่วนต่อไปนี้จึงขอกล่าวเพิ่มเติมเกี่ยวกับการบริหารริงโดยใช้ MAC เฟรมชนิดต่างๆ ก่อนที่การส่งและรับเฟรมข้อมูลในริงจะเกิดขึ้นได้ ริงจะต้องมีการการเริ่มตั้งต้นก่อน ดังนั้นถ้ามีสถานีใหม่ต้องการเข้ามาเพิ่มในริงที่กำลังทำงานอยู่ก็ต้องมีขบวนการเริ่มต้นเพื่อให้แน่ใจว่าจะไม่ทำให้การทำงานของริงเสียไป

5.8.1.1 ขบวนการเริ่มต้น

เมื่อมีสถานีต้องการมีส่วนในริง ขั้นแรกต้องมีการตรวจสอบก่อนว่า มีสถานีอื่นที่ใช้แอดเดรสเดียวกันหรือไม่ ขั้นต่อมาจะบอกกับสถานีถัดไปที่อยู่ติดกัน (Down stream) ว่ามีสถานีใหม่แทรกเข้ามาเพิ่ม ในการตรวจสอบการซ้ำ

ของแอดเดรสนั้น จะทำโดยสถานีนั้นจะส่ง DAT (Duplicate Address Test) เฟรมออกมา โดยเซตบิต A ใน FS ฟิลด์เป็น 0 และตั้งค่า แอดเดรสปลายทางเป็นแอดเดรสตัวเอง แล้วส่งเฟรมออกไป ซึ่งสถานีอื่นๆในเมื่อได้รับเฟรมก็จะดูว่า DA เป็นแอดเดรสของตัวเองหรือเปล่าถ้าใช่ก็จะเซต บิต A เป็นหนึ่งเพื่อแสดงว่ามีการรับเฟรมนั้นแล้ว และเมื่อเฟรมนั้นวนกลับมายังสถานีเดิม ถ้าพบว่าค่า A เปลี่ยนไปแสดงว่าเกิดมีแอดเดรสซ้ำกับสถานีตัวเองขึ้นแล้ว มันก็จะทำการแจ้งตัวบริหารเน็ตเวิร์คทราบและถอดตัวเองออกจากริง แต่ถ้าค่า A ยังคงเป็น 0 สถานีก็จะดำเนินการขบวนการเริ่มต้นต่อไปโดยส่งเฟรม SMP(Standby Monitor Present) ออกมา โดยบิต A และ C ถูกเซตเป็น 0 สถานีถัดไปที่ได้รับเฟรมนี้เป็นสถานีแรกก็จะทำการนำค่า SA ของเฟรมนั้นเก็บไว้ที่สถานีตัวเอง ใช้เป็นค่า UNA(Upstream neighbours Address) คือแอดเดรสสถานีต้นทางที่อยู่ติดกันนั่นเอง ทั้งนี้จะเก็บไว้เพื่อ เวลามีปัญหาเกิดขึ้นจะได้บอกสถานีข้างเคียงได้ ถึงจุดนี้ก็ถือว่าขบวนการเริ่มต้นเรียบร้อยแล้ว

5.8.1.2 สแตนด์บายมอนิเตอร์(Standby Monitor)

หลังจากทำขบวนการเริ่มต้นเสร็จเรียบร้อยแล้ว สถานีก็สามารถส่งและรับเฟรมและโทเกนตามปรกติได้ ในขณะนั้นสถานีจะถือว่าอยู่ในสถานะสแตนด์บายมอนิเตอร์ด้วยโดยอัตโนมัติ (ทุกๆสถานีในริงจะเป็นสแตนด์บายมอนิเตอร์ยกเว้นจะมีสถานีหนึ่งเป็นแอคทีฟมอนิเตอร์) ซึ่งจะมีหน้าที่คอยตรวจสอบการปรากฏของเฟรม AMP(Active Monitor Present) ซึ่งจะถูกส่งออกมาเป็นระยะๆจาก แอคทีฟมอนิเตอร์(จะกล่าวในหัวข้อต่อไป) ถ้าไม่ได้รับเฟรมนี้ภายในกำหนดเวลาที่ตั้งไว้(จากไทเมอร์ 2 ตัว) มันก็จะเข้าสู่สถานะ เคลมโทเกน(Claim Token state) ซึ่งมันจะทำการส่งเฟรม CT(Claim Token) ออกไปซึ่งจะมีค่า UNA อยู่ด้วย และมันจะตรวจสอบ CT เฟรมที่ได้รับว่ามีค่า SA เท่ากับ แอดเดรสของตัวเองหรือไม่และ มีค่า UNA เท่ากับที่มันเก็บไว้หรือไม่ ถ้าเท่าก็แสดงว่าเฟรม CT วนไปรอบริงแล้ววิ่งกลับมาโดยปรกติ สถานีนั้นก็จะเปลี่ยนตัวเองเป็น แอคทีฟมอนิเตอร์ ตัวใหม่ทันที แต่ถ้าได้รับเฟรม CT ที่มีค่า SA ใหญ่กว่าของตัวเองแสดงว่ามีสถานีอื่นแย่งการขอเป็นแอคทีฟมอนิ

เตอร์ไปก่อนแล้ว สถานีนี้ก็จะเลิกสถานะการแย่งชิงการเป็นแอคทีฟมอนิเตอร์ของสถานีในริงแล้วกลับเข้าสู่สถานะสแตนด์บายมอนิเตอร์ตามเดิม

5.8.1.3 แอคทีฟมอนิเตอร์

ถ้าสถานีแย่งชิงการเป็นแอคทีฟมอนิเตอร์สำเร็จ (ทั้งในตอนเริ่มต้นริงครั้งแรก หรือ ต้นสร้างแอฟทีฟมอนิเตอร์ใหม่) ประการแรกที่ต้องทำก็คือทำการแทรก บัฟเฟอร์เวลาแฝง(Latency buffer) เข้ามาในริงและทำการเปิดสัญญาณนาฬิกาของตัวเอง แล้วส่ง PRG(Purge)เฟรมออกมาเพื่อความแน่ใจว่ามีโทเคนหรือเฟรมค้างอยู่ในริง ก่อนที่จะทำการส่งโทเคนออกไปใหม่ และถ้ามันได้รับ PRG กลับมาปรกติก็แสดงว่าทำการล้างทิ้งเฟรมที่ค้างอยู่ในริงสำเร็จ หลังจากนั้นมันก็เข้าสู่ขบวนการรับรู้สถานีเพื่อนบ้าน โดยส่ง AMP เฟรมออกไป สถานีที่อยู่ถัดจากแอฟทีฟมอนิเตอร์เมื่อได้รับ AMP เฟรมโดยมีบิต A เป็น 0 จะทำการอ่านค่า UNA ที่ส่งมาด้วยแล้วทำการปรับปรุงค่า UNA ที่อยู่ในสถานีตนเองซึ่งจะกลายเป็นค่าแอดเดรสของสถานีต้นทางที่อยู่ติดกันที่ทันสมัยที่สุด แล้วทำการเปลี่ยนบิต Aและ C เป็น 1 เพื่อบอกว่าได้รับเฟรม AMP แล้ว AMP เฟรมจะส่งผ่านไปยังสถานีอื่นๆในริงซึ่งเมื่อเห็นจะไม่การอ่านค่าใดๆแต่จะทำการรีเซตค่า AMP เฟรมไทเมอร์เท่านั้น

หลังจากสถานีที่อยู่ถัดจากแอฟทีฟมอนิเตอร์ ได้รับเฟรม AMP แล้วตัวเองก็จะทำ'ขบวนการรับรู้สถานีเพื่อนบ้าน'ต่อไป โดยส่ง SMP เฟรมออกมา สถานีต่อไปที่อยู่ถัดจากสถานีนี้เมื่อได้รับเฟรมนี้ก็จะอ่านค่า UNA ที่ส่งมาด้วยแล้วปรับปรุงค่า UNA ที่อยู่ในสถานีตัวเอง แล้วส่งเฟรมนั้นต่อไปโดยเซต A และ C บิตเป็น 1 ทุกๆสถานีบนริงจะทำอย่างนี้ต่อๆกันไปจนครบทุกสถานี ซึ่งก็จะทำให้ค่า UNA ของแต่สถานีเป็นค่าที่ทันสมัยที่สุด การส่ง AMP เฟรมของแอฟทีฟมอนิเตอร์นั้นจะส่งออกมาเป็นระยะๆตามเวลาที่ตั้งไว้ทำให้ทุกสถานีรับรู้ว่าริงยังมีการทำงานปรกติอยู่ แต่ถ้าไม่ได้รับเฟรมนี้ตามเวลากำหนดมันก็จะส่ง CT เฟรมออกมาแย่งการเป็น แอฟทีฟมอนิเตอร์ตัวใหม่อีกที แต่ถ้ายังคงมีความผิดพลาดเกิดขึ้น ก็จะเข้าสู่ขบวนการตรวจสอบความผิดพลาดที่เรียกว่า เบคอน(Beaconing)

5.8.1.4 ขบวนการเบคอน

ถ้ามีความผิดปรกติที่ร้ายแรงเกิดขึ้นในริง เช่นสายขาด แต่ละสถานีในริงก็จะส่งข้อมูลเพื่อบอกให้หยุดระบบไว้ก่อนจนกว่าจะแก้ไขได้เรียบร้อย ซึ่งเรียกขบวนการนี้ว่า ขบวนการเบคอน ซึ่งสิ่งที่ประกอบขบวนการคือ

1. สถานีที่รายงานการเกิดข้อผิดพลาดขึ้น เรียกว่าสถานีเบคอน (Beaconing station)

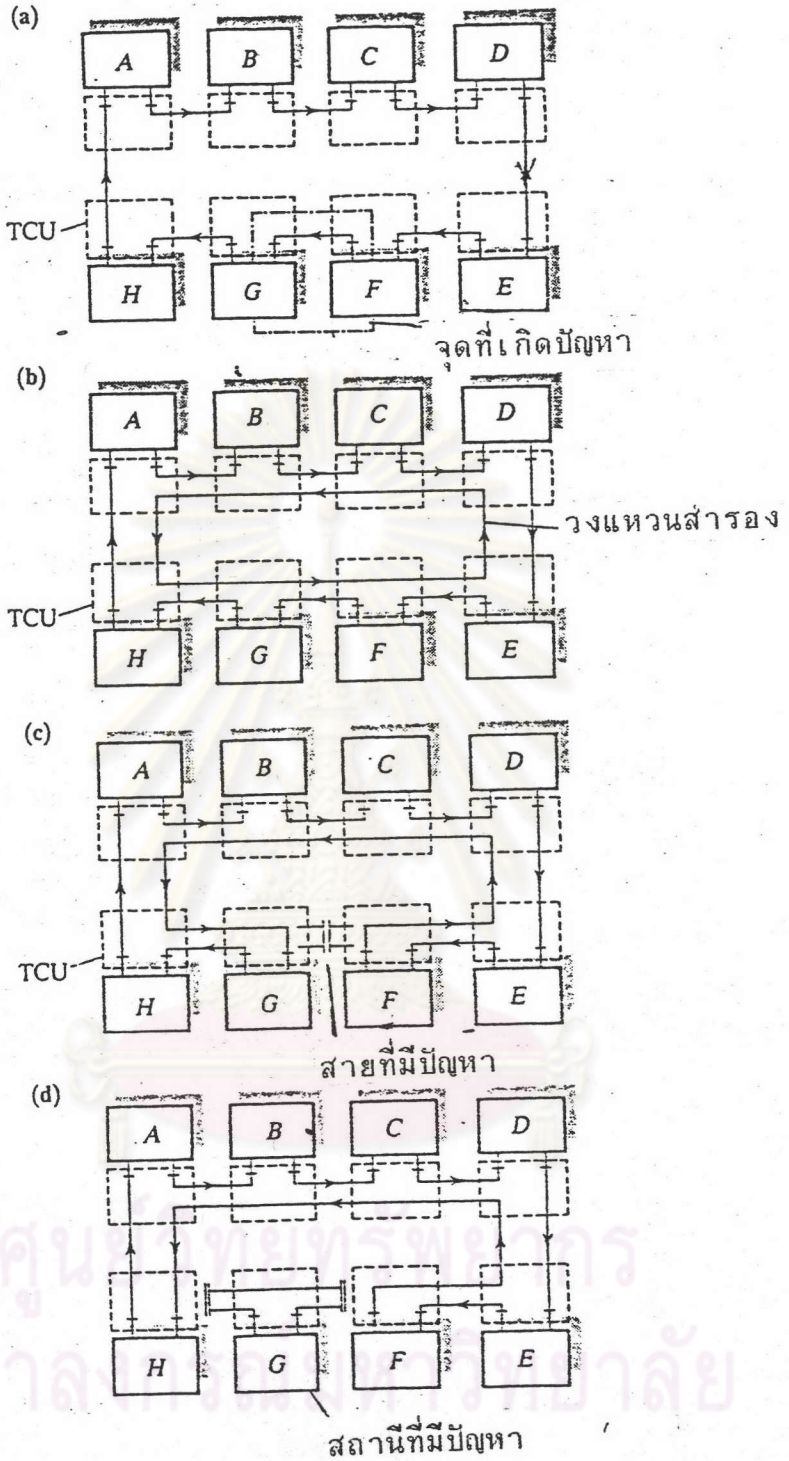
2. สถานี DTE ที่อยู่ก่อนหน้าสถานีเบคอน

3. ตัวกลายสายระหว่างสองสถานีนี้

ตามตัวอย่างในรูปจะเห็นว่าถ้าเกิดมีสายขาด ระหว่างสถานี F และ G ในที่นี้ G จะเป็นสถานีเบคอน และ F เป็นสถานีที่อยู่ก่อนหน้าสถานีเบคอน โดยปรกติแล้วสถานะเบคอนจะเกิดขึ้นในกรณีที่ AMP ไทเมอร์นั้นหมดเวลาลงโดยไม่ได้รับเฟรม AMP ซึ่งในสถานะนี้สถานี(ซึ่งจะกลายเป็นสถานีเบคอน)ก็จะส่ง BCN (Beacon) เฟรมออกมาอย่างต่อเนื่องจนกระทั่งเฟรมนี้ได้รับหรือหมดเวลาที่ตั้งไว้ ซึ่งในกรณีหลังนี้ตัวบริหารเน็ตเวิร์คจะรับทราบแล้วทำการหยุดขบวนการส่งทั้งหมด แต่ถ้าเฟรม BCN ได้รับกลับคืนโดยที่ค่า SA เท่ากับของตัวมันเองก็แสดงว่าข้อผิดพลาดได้ถูกแก้ไขแล้วสถานีก็จะเข้าสู่การ ส่ง CT เฟรมต่อไป แต่ถ้าถ้ามันได้รับ BCN เฟรมที่มีค่า SA แตกต่างจากของสถานีตัวเองมันก็จะเข้าสู่สถานะสแตนด์บายมอนิเตอร์แทน

ในกรณีที่มันไม่ได้รับ BCN กลับคืนตัวบริหารเน็ตเวิร์คก็จะรู้ว่ามีข้อผิดพลาดเกิดขึ้นก่อนสถานีเบคอนนั้น ตัวบริหารเน็ตเวิร์คจะทำให้มีการหยุดการส่งผ่านเฟรมในริงชั่วคราว ต้องจะทำการแก้ไขริงส่วนที่มีปัญหาก่อนโดยทำการใช้ริงที่สำรองไว้มาต่อแทน ทำให้สถานีที่มีปัญหา (ในที่นี้คือสถานี G) ถูกตัดออกจากริงโดยขบวนการทางแมคคานิค แล้วเริ่มขบวนการส่งเฟรมตามปรกติต่อไปใหม่

เราจะเห็นได้ว่าขบวนการระดับ MAC เฟรมของโทเกนริงนั้นยุ่งยากซับซ้อนทีเดียว แต่โดยปรกติแล้วขบวนการเหล่านี้ผู้ใช้โทเกนริงจะมองไม่เห็นเพราะว่าในเน็ตเวิร์คอินเตอร์เฟสการ์ดของโทเกนริงมีส่วนการจัดการระดับ MAC อยู่ในการ์ดแล้ว



รูปที่ 5.23 ขบวนการเบคอนเพื่อแก้ไขส่วนที่มีปัญหาในริง (Halsall 1989)

5.8.2 ระดับ LLC (Martin 1989; Miller 1990)

การติดต่อกันในระดับ LLC นั้น แบ่งเป็น 2 ชนิด คือ ชนิดที่ 1 (Type 1) ซึ่งจะเป็นแบบไม่ได้สร้างการเชื่อมต่อ และ ชนิดที่ 2 (Type 2)แบบสร้างการเชื่อมต่อ และเฟรมที่ใช้ในการติดต่อกันนั้นยึดตาม โพรโตคอลแบบ HDLC(High-level Datalink Control) โดยที่จะแบ่งเฟรมเป็น 3 ประเภทคือ U (Unnumber) เฟรม,I(Information) เฟรม และ S(Supervisory) เฟรม การแบ่งประเภทเฟรมต่างๆนี้จะดูที่ ฟิลด์ควบคุม ของ LLC เฟรม ซึ่งเฟรมแต่ละประเภทจะมีค่าที่แตกต่างกัน ตามรูปแบบที่กำหนดไว้

โดยที่การติดต่อกันแบบชนิดที่ 1 จะใช้เฟรมประเภท U เฟรม บางแบบ(มีหลายแบบ) คือ UI(Unnumber Information), XID(Exchange ID) และ TEST

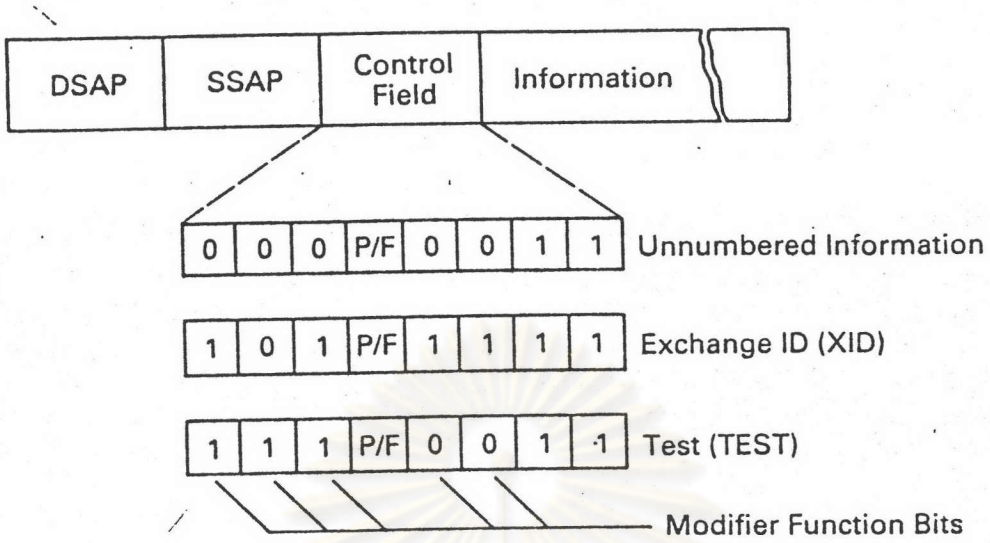
ส่วนการติดต่อกันแบบชนิดที่ 2 นั้นจะใช้เฟรมในการติดต่อ ทั้งประเภท U เฟรม(บางแบบ) ,I เฟรม และ S เฟรม

5.8.2.1 การติดต่อชนิดที่ 1 (ไม่สร้างการเชื่อมต่อ)

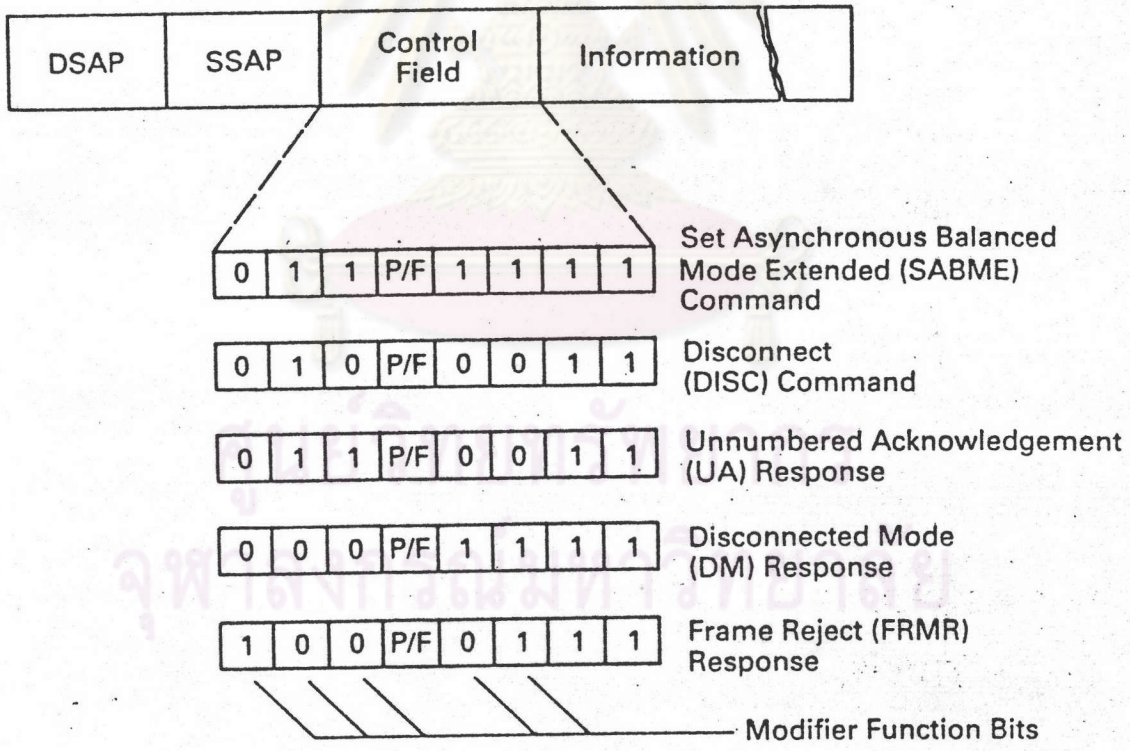
การติดต่อส่งผ่านข้อมูลในแบบนี้จะส่งให้กัน ระหว่าง SAP โดยไม่มีการตอบรับ ซึ่งจะใช้เฟรมชนิด UI ของ U เฟรม การส่งชนิดนี้ข้อมูลอาจหายได้เพราะไม่มีการรับประกัน ส่วน U เฟรมชนิด XID และ TEST จะมีไว้สำหรับทำหน้าพิเศษ คือ XID เฟรมจะใช้ในการแลกเปลี่ยนข้อมูลระหว่างสองสถานีรวมทั้งชนิดของการให้บริการที่สถานีนั้นมีอยู่ ส่วน TEST เฟรมจะใช้ในการทดสอบหาเส้นทางในการส่งผ่านข้อมูลในระหว่างสองสถานี

5.8.2.2 การติดต่อชนิดที่ 2 (สร้างการเชื่อมต่อ)

การติดต่อกันแบบนี้ จะมีตัวเลขกำกับลำดับของเฟรมที่ส่งผ่านกันไปมาระหว่างสองสถานี ซึ่งจะใช้ I เฟรมในการส่งข้อมูล โดยที่ในส่วนฟิลด์ควบคุมจะมี ตัวเลขลำดับการส่งของเฟรมทั้งฝ่ายส่งและฝ่ายรับ คือ N(R) และ N(S) และที่แต่ละสถานีจะมีตัวนับการส่งและการรับ ซึ่งสร้างขึ้นตอนเปิดเซสชัน โดยจะมีไว้

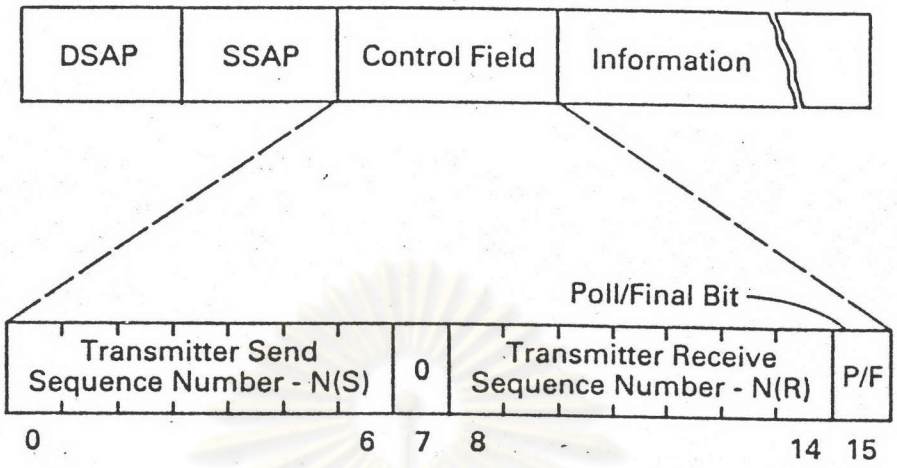


รูปที่ 5.24 แสดงรายละเอียดฟิลด์ต่างๆ ของ U เฟรม ที่ใช้กับการติดต่อชนิดที่ 1 (Martin 1989)

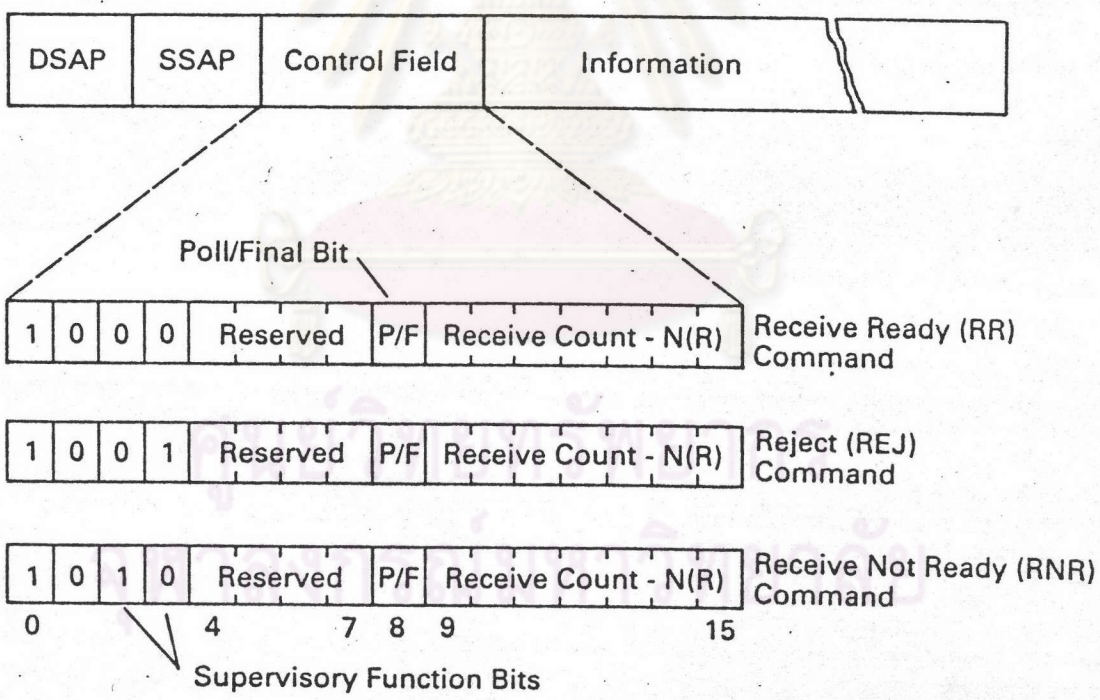


รูปที่ 5.25 แสดงรายละเอียดฟิลด์ต่างๆ ของ U เฟรม ที่ใช้กับการติดต่อชนิดที่ 2

(Martin 1989)



รูปที่ 5.26 แสดงรายละเอียดฟิลด์ต่างๆ ของ I เฟรม (Martin 1989)



รูปที่ 5.27 แสดงรายละเอียดฟิลด์ต่างๆ ของ S เฟรม (Martin 1989)

ตรวจสอบตัวเลขกับค่า N(S) และ N(R) ว่าตรงกันหรือไม่ และนอกจากนี้จะมีการส่งเฟรมการตอบรับ(Acknowledgment) ในช่วงเวลาหนึ่งตั้งไว้หรือตามจำนวนเฟรมที่ส่งผ่านที่ตั้งไว้ ซึ่งเรียกว่าค่าการตั้ง 'หน้าต่าง'(Windowing) (ซึ่งปรกติตั้งไว้ที่ 127 ตามขนาดของ N(R) และ N(S) แต่สามารถตั้งค่าใหม่โดยคำสั่ง XID) โดยในการตอบรับนั้นก็จะใช้ I เฟรมเช่นกันและในฟิลด์ข้อมูลก็จะส่งข้อมูลกลับมาให้ด้วย ทำให้เป็นทั้งเฟรมการตอบรับและเฟรมข้อมูลด้วยในคราวเดียวกัน แต่ถ้าหากว่าขณะไม่มีข้อมูลจะส่งก็จะใช้เฟรมประเภท S ในการตอบรับแทน ในการส่งเฟรมการตอบรับนั้นก่อนอื่นมันจะดูค่า P/F (Poll/Final) บิตก่อนถ้าเป็น 1 แสดงว่ามันต้องการการตอบรับในทันทีทันใดโดยไม่ต้องรอให้ถึงค่าเวลาหรือค่าจำนวนเฟรมที่ตั้งไว้ ในกรณีนี้มันจะตอบรับโดยการส่งเฟรมประเภท S เฟรม เป็นการตอบรับ ซึ่งจะมี 3 แบบคือ RR(Receive Ready) ถ้าการส่งไม่มีปัญหา, RNR(Receive Not Ready) ถ้าการส่งมีปัญหา และ REJ(Reject) ถ้าต้องการให้มีการส่งใหม่

ในการสร้างการติดต่อแบบชนิดที่ 2 นี้ในตอนสร้างเซสชันและปิดเซสชันนั้นต้องมีการใช้ U เฟรมชนิดต่างๆ คือ SABME(Set Asynchronous Balance Mode Extended) ใช้ในการร้องขอการสร้างเซสชัน และสถานีรับจะตอบด้วย UA(Unnumbered Acknowledgment) ส่วน เฟรม DM(Disconnected Mode)จะการปฏิเสธการสร้างเซสชัน และ เฟรม DISC(Disconnect) จะใช้ส่งเพื่อเลิกการติดต่อ สำหรับเฟรม FRMR(Frame Reject) จะใช้ส่งเมื่อสถานีรับตรวจสอบว่ามีข้อผิดพลาดเกิดขึ้นและแก้ไม่ได้โดยการส่งใหม่

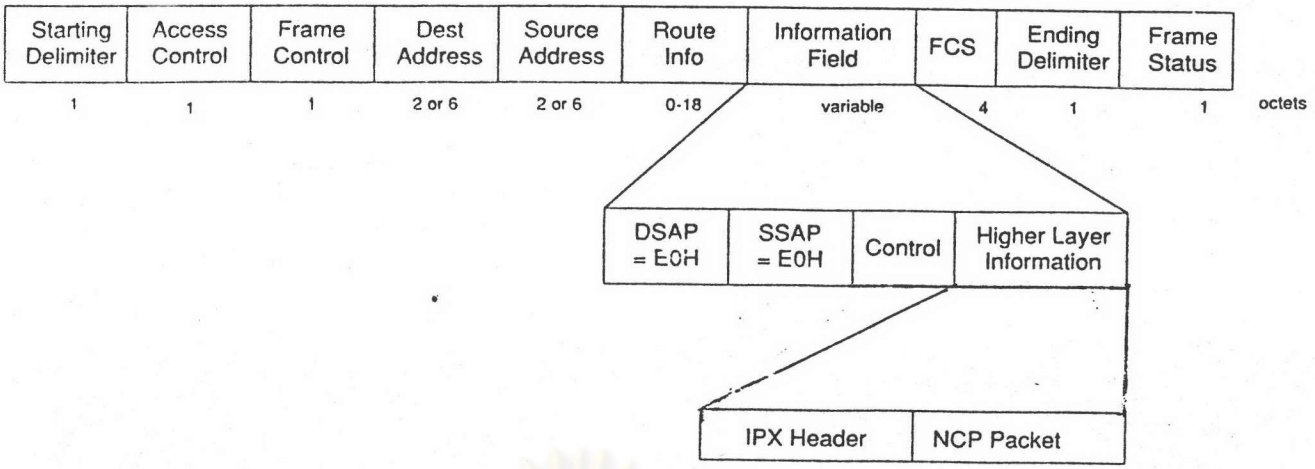
ในการติดต่อในระดับ LLC นี้จะต้องมีการสร้าง SAP ขึ้นมาก่อนดังที่กล่าวรายละเอียดในหัวข้อ 5.4 SAP จะเปรียบเสมือนเป็น พอร์ตในการติดต่อกับโพรโตคอลระดับสูงขึ้นไป เพราะฉะนั้น SAP จึงทำให้โพรโตคอลระดับสูงขึ้นไปหลายๆ ชนิดสามารถติดต่อกับโพรโตคอลระดับต่ำร่วมกันได้ ซึ่งใน LLC ของ โทเกนริงจะแบ่งประเภทของ SAP ที่โพรโตคอลระดับสูงขึ้นไปใช้ในการติดต่อโดยกำหนด SAP แอดเดรสดังนี้

ตัวอย่าง SAP	การใช้งาน
00	NULL
02	individual LLC sublayer management
03	Group LLC sublayer management
04	SNA Path Control (individual)
05	SNA Path Control (Group)
06	ARPANET's Internet Protocol
80	3Com XNS
AA	TCP/IP SNAP Protocol
E0	Novell NetWare
F0	IBM NETBIOS
F4	LAN Management (individual)
F5	LAN Management (group)
F8	IBM Remote Program Load
FE	ISO Network Layer
FF	Global

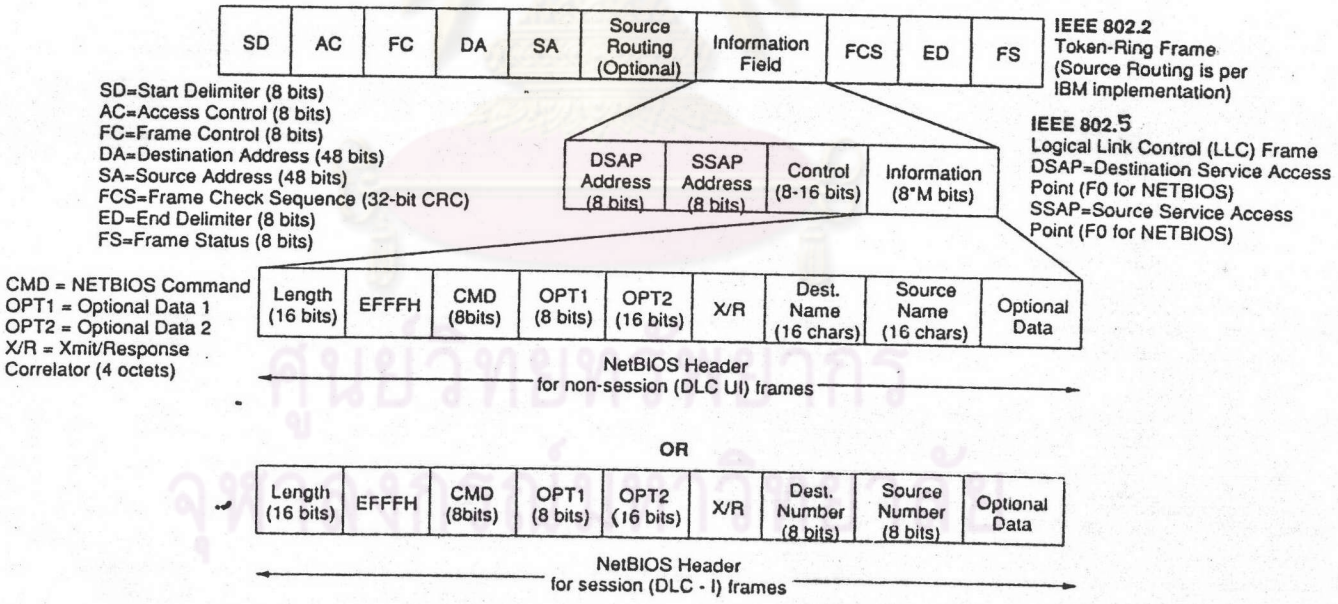
5.8.3 ระดับ NETBIOS และ IPX/SPX (Martin 1989; Nance 1990)

NETBIOS เฟรมนั้นจะบรรจุอยู่ในส่วนข้อมูลของ LLC เฟรม ซึ่งจะประกอบด้วย NETBIOS Header และส่วนข้อมูลซึ่งอาจจะมีหรือไม่มีก็ได้ เฟรมของ NETBIOS นี้แบ่งเป็น 2 ประเภท คือ ประเภท U และ I ซึ่งก็คือส่วน ฟิลด์ควบคุม เฟรมในส่วนหัวของ LLC จะเป็นตัวบอกว่า เป็น U หรือ I ตามที่กล่าวในหัวข้อที่แล้วนั่นเอง และในฟิลด์ SSAP กับ DSAP ในส่วนหัวของ LLC จะมีค่าเป็น 0xF0 ถ้าในส่วนข้อมูลของมันเป็น NETBIOS เฟรม

ในส่วนหัวของ NETBIOS ประเภท UI เฟรม จะประกอบไปด้วย



รูปที่ 5.28 แสดงให้เห็นว่า IPX จะเป็นส่วนข้อมูลของ LLC เฟรม (Miller 1990)



รูปที่ 5.29 แสดงรายละเอียดของ NETBIOS เฟรม (Miller 1990)

ฟิลด์บอกความยาวของส่วนหัว , ฟิลด์ค่าคงที่ EFFFFh, ฟิลด์คำสั่งของ NETBIOS, ฟิลด์ข้อมูลที่เลือกได้ 1 และ 2, ฟิลด์ Transmit/Response correlator มีขนาด 4 ไบต์ และสองฟิลด์สุดท้ายคือ ฟิลด์ชื่อต้นทางและ ชื่อปลายทาง(NETBIOS อ้างอิงกันด้วยระบบชื่อ)

สำหรับในส่วนหัวของ NETBIOS ประเภท I เฟรม นั้นส่วนประกอบจะเหมือนกับ UI เฟรมเกือบทุกอย่าง ผิดกันที่ ฟิลด์ชื่อต้นทางและชื่อปลายทางนั้นจะเป็น หมายเลขเซสชันของต้นทางและปลายทางแทน เพราะว่าการติดต่อในแบบนี้จะมีการสร้างเซสชันขึ้นก่อน เพราะฉะนั้นการอ้างอิงก็อ้างอิงถึงหมายเลขเซสชันแทน

ฟิลด์คำสั่งของ NETBIOS เฟรมนั้นจะมีคำสั่งทั้งหมด 22 คำสั่งซึ่งแต่ละคำสั่งจะมีรหัสประจำคำสั่ง ขนาด 2 ไบต์ รหัส 00H-13H จะเป็น U เฟรม แต่ถ้ารหัส 14H-1FH จะเป็น I เฟรม

ในส่วนฟิลด์ข้อมูลของ NETBIOS เฟรมจะเป็นข้อมูลของโพรโตคอลในระดับสูงขึ้นไปเช่น SMB ของ IBM PC LAN เป็นต้น

สำหรับ เฟรมของ IPX/SPX นั้น ตัวเฟรมจะบรรจุอยู่ในส่วนข้อมูลของ LLC เฟรมเช่นเดียวกับ NETBIOS โดยที่รายละเอียดของฟิลด์ต่างๆในเฟรมนั้นได้กล่าวถึงในหัวข้อก่อนหน้านี้มาแล้ว ถ้าส่วนข้อมูลของ LLC เป็น SPX/IPX เฟรมแล้วในฟิลด์ SSAP และ DSAP จะเป็นค่า E0H สำหรับส่วนข้อมูลของ IPX/SPX เฟรมนั้นจะเป็นข้อมูลของโพรโตคอลระดับสูงขึ้นไปเช่น NCP (Netware Core Protocol) ซึ่งเป็นของเน็ตแวร์เป็นต้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย