

最 終 研 究 報 告 書

고속지역 정보망 개발 및 구성 연구

A Study on the Development of
High-Speed
Metropolitan Area Network

주관연구기관 : 한국전자통신연구소

과 학 기 술 처

제 출 문

과학기술처 장관 귀하

본 보고서를 "고속 지역 정보망 개발 및 구성 연구"의 1차년도 최종 연구보고서로 제출합니다.

1990. 9. 11

주관연구기관명 : 한국전자통신연구소
총괄연구책임자 : 정 선 종(책임연구원)
연구 책임 자 : 이 남 회(컴퓨터통신연구실장)
선 임 연 구 원 : 홍 승 호
연 구 원 : 김 명 섭
연 구 원 : 김 진 철
위탁 연구 기관 : 디지콤연구소(김희동 소장)
한양대학교 (박용진 교수)

요 약 문

1. 과제명 : 고속 지역 정보망 개발 및 구성 연구
2. 주관연구기관 : 한국전자통신연구소
3. 연구 책임자 : 이 남 회 (컴퓨터통신연구실장)
4. 연구개발의 내용 및 범위

가. MAN 프로토콜 연구

- DQDB MAN 프로토콜 구조 분석
- DQDB MAN 프로토콜 블럭 설계
- DQDB MAN 프로토콜 성능 평가

나. 고속 동기접속 장치 개발

- Token Ring 지역망과 X.25망과의 고속 동기 접속
- LLC Firmware 및 X.25, Conversion S/W 내장
- DTE/DCE 접속 연구

다. 멀티미디어 MHS 설계

- Multimedia 메시지 구조 연구
- Multimedia 메시지 프로토콜 연구 설계
- Multimedia 에디터 연구 및 설계

5. 주요 연구 개발 및 연구 결과

- 가. DQDB MAN 프로토콜 블록 설계
- 나. DQDB MAN 성능 평가(Simulation)
- 다. 고속 동기접속 장치 개발
- 라. 멀티미디어 MHS 설계

6. 기대 효과

- 고속 컴퓨터통신망 구성 기술 축적
- 광섬유의 고속 전산망 활용 기술 배양
- 고속 지역 정보망 개발기반 확보

SUMMARY

As the demand of high-speed, multi-media services is increased in many areas, MAN(Metropolitan Area Network) has emerged and reached the experimental stage. MAN is a relatively new network system which provides not only data services but also voice and video services over a high-speed network medium. MAN covers relatively large areas up to 50 Km in diameter. The IEEE 802.6 Working Group on Metropolitan Area Networks has developed the DQDB(Distributed Queue Dual Network) protocol, a generic name for a high-speed, shared-medium-access protocol consisting of pairs of buses funneling data in opposite direction. Although it is still in draft form, it is expected to be quickly embraced by MAN providers and equipment vendors once the group votes to approve it.

During the first year of this project, research is oriented to the basic study of implementing DQDB MAC(Medium Access Control) protocol. DQDB MAC is mainly divided into three blocks of MAC convergence function block, QA function block and Common Function block. MAC convergence function block is responsible for adapting the segment-payload based service provided by QA function block to the standard MAC service required by the LLC sublayer. QA function block provides an asynchronous data transfer service for 48 octet segment payloads. Common Function block transfers slots and management information octets between the QA function block and local Physical Layer entity. Common Function block also supports head of bus function, the configuration control function and MID page allocation function.

As the fundamental research for the implementation of DQDB MAC protocol, this report provides the functional block diagrams of MAC convergence function block, QA function block and Common Function block, and also defines the relationships among these blocks. The standardization procedure of DQDB protocol is delayed mainly because of the performance characteristics such as unfairness in utilizing the medium and malfunction of priority mechanism. These performance characteristics are examined in this study by using the simulation model. This study also provides some suggestions which may alleviate the problems of unfairness and priority malfunction.

목 차

제 1 장	서 론	1
제 2 장	DQDB MAN 프로토콜	7
2.1	DQDB MAN 프로토콜 구조	9
2.2	DQDB MAN 표준화 동향	16
제 3 장	DQDB MAN 블럭 설계	23
3.1	MAC Convergence Function	25
3.2	QA Function	42
3.3	Common Function	57
제 4 장	DQDB MAN 성능 분석	73
4.1	DQDB의 성능 영향 요소	75
4.2	DQDB의 공평성 개선	82
제 5 장	결 론	87
[부 록]	A. 고속 동기 접속 기능 개발	93
	B. 멀티미디어 MHS 설계	124
[참고문헌]		153

제 1 장 서 론

제 1 장 서론

통신 기술의 급속한 발전에 따른 정보화 사회의 확산과 함께 각 분야에서 정보 교환에 대한 요구가 증대됨에 따라 LAN (Local Area Network) 및 WAN (Wide Area Network)과 같은 기존의 통신망으로는 사용자의 다양한 욕구를 충족시킬 수 없게되었다. LAN은 정보를 전달할 수 있는 거리가 제한되어 있고 적용 분야가 세분화되어 있는 반면, WAN은 수십 Km 이상의 장거리 데이터 전송에 적합한 구조를 가지고 있다. 따라서 직경 50 Km 정도의 도시 지역을 수용할 수 있는 MAN (Metropolitan Area Network)의 필요성이 대두되었다. MAN은 LAN보다 빠른 속도로 데이터 전송할 수 있도록 고속화되어야 하며 음성 및 화상 서비스 까지 제공될 수 있도록 광대역화 되어야 할 필요가 있다. 또한 MAN은 앞으로 출현할 B-ISDN (Broadband ISDN)과도 접속이 용이하여야 한다.

MAN은 멀티미디어(multimedia)를 처리하는 워크스테이션과 고속의 대량정보를 출력하는 슈퍼컴퓨터들이 망에 직접 연결될 수 있으며 현재 곳곳에 설치되어 운용되고 있는 LAN (IEEE 802.3, 4, 5 및 FDDI) 들을 용이하게 상호 접속해 주고, 위성지상망, 텔리포트 (Teleport) 망, CATV 망 등으로서 사용될 수 있다. 현재 미국의 AT&T와 BOC, 프랑스의 Alcatel, 독일의 Siemens, 호주의 QPSX 등 세계 굴지의 통신회사 들이 MAN의 개발에 경쟁적으로 참여하고 있으며, 호주의 멜버른, 프랑스의 파리 및 미국의 여러 지역에서 시범망의 운용이 이루어지고 있다.

현재 개발된 또는 개발중인 HSLAN(High-Speed LAN) 및 MAN 프로토콜로는 DQDB[1], METROCORE[2], BWN (Backbone Wideband Network)[3], CFR(Cambridge Fast Ring)[4], CBN(Cambridge Backbone Network)[5], LION(Local Integrated Optical Network)[6] 등이 있다. BWN은 134Mbps의 전송 속도를 가지고 FDDI(Fiber Distributed Data Interface)[7]와 같이 토큰링 방식으로 동작되며, 우선 순위 기능은 제공하지 않는다. 100Mbps로 동작되는 CFR은 슬롯 링 방식으로 작동되며 각 노드는 마스터 스테이션에서 생성되는 슬롯을 통하여 정보를 전달한다. CBN은 CFR과 같이 슬롯 링 방식으로 동작되나 전송 속도를 1000Mbps까지 증가시켰다. LION은 이중/중첩(double/folded) 버스 구조를 가지며 각 노드는

마스터 스테이션에서의 제어 신호에 따라 정보를 전달한다. LION은 현재 140Mbps로 동작되나 2단계로 565Mbps를 고려하고 있다. METROCORE와 DQDB는 이중 버스의 구조를 가지며 슬롯을 통하여 정보를 전달한다. METROCORE는 순번제(round robin) 접속 방식을 사용하며 마스터 스테이션에서 라운드를 통제한다. METROCORE는 150Mbps의 전송 속도를 제공한다. DQDB는 각 노드가 슬롯의 사용권을 신청하는 분산큐 방식으로 동작된다. DQDB는 전송 시스템에 제한을 두고 있지 않으나 SONET/SDH[8]가 사용될 경우 155Mbps의 전송 속도를 가진다.

MAN의 MAC(Medium Access Control)의 일반화가 예상되는 프로토콜로는 FDDI 및 FDDI-II[9], 미국의 Bellcore에서 개발 중인 METROCORE와 IEEE 802.6 DQDB MAN 등이 있다. 토큰 링의 구조를 갖는 FDDI는 MAN이라기 보다는 데이터의 고속 전송을 위한 HSLAN (High Speed LAN)의 특성을 가지고 있으며, 음성 및 화상과 같은 등시성 (Isochronous) 데이터의 전송을 위한 HRC (Hybrid Ring Control) 기능이 추가되어 현재 표준안 작성 단계에 있는 FDDI-II는 그 프레임 구조상 B-ISDN 등과의 호환성이 용이하지 않다는 단점이 있다. Bellcore에서 Fasnet 프로토콜을 기본으로 하여 현재 개발 중인 METROCORE는 8개의 MAC을 동시에 병렬 처리 함으로써 최고 1.7 Gbps까지의 전송속도를 가진다고 보고되고 있으며 화상 서비스까지도 제공하고 있으나 아직은 실험망의 단계에 머무르고 있는 상태이다. 따라서 현재 IEEE 802.6 위원회에서 표준화를 추진하고 있는 DQDB (Distributed Queue Dual Bus) 프로토콜이 일반적으로 MAN 망으로 가장 적합하다고 인식되고 있으며 이는 다음과 같은 특성을 가지고 있다.

- (1) 비교적 양이 적고 bursty하나 실시간 요구 특성이 약한 정보들을 전달하기 위한 비연결형 서비스 (connectionless service, 예: 일반 데이터), 양이 많으면서 실시간 요구 특성이 약한 정보들을 전달하기 위한 연결형 서비스 (connection-oriented service, 예: 화일 정보)와 고정된 대역폭 및 실시간적 전달을 요구하는 등시성 서비스 (isochronous service, 예: 음성 및 화상)를 모두 제공한다.
- (2) 광섬유를 전송 매체로 사용하는 것을 전제로 최대 전송속도는 제한되지 않고 있으며,

현재 기본표준안에는 전송 시스템으로 ANSI DS3 (44.736 Mbps)만이 정의되어 있으나 차후에 CCITT G.703(34.368 Mbps 및 139.264 Mbps), CCITT G.707-709(155.520 Mbps)등을 규정할 계획에 있다.

- (3) CCITT에서 장래에 공중 광대역 서비스를 제공할 수 있도록 규정 권고 단계에 있는 광대역 종합 정보 통신망 (B-ISDN)의 ATM (Asynchronous Transfer Mode) 과 같은 셀(cell) 구조를 가지고 있어 상호 연동이 용이하도록 규정되고 있다. 이는 MAN이 사설망은 물론 공중망으로도 사용될 수 있음을 나타낸다.

본 과제에서는 1986.5 - 1989.7 까지 수행되었던 '지역정보통신망 구성연구' 과제에서 얻어진 기술을 바탕으로 하여 세계적으로 그 요구가 확산되어가고 있는 MAN의 구현을 위한 기초연구에 착수하게 되었다. MAN을 구현하기 위하여서는 기본적으로 MAC 기능을 담당하는 프로토콜의 구현이 선행되어야 한다. 본 보고서에서는 IEEE 802.6 DQDB MAN의 구조와 표준화 동향에 대하여 고찰하였고, MAC 프로토콜의 구현을 위하여 블럭 설계도를 작성하였으며, 또한 시뮬레이션 모델을 통하여 DQDB MAN의 성능을 분석하였다. 부록으로는 본 과제에서 위탁 과제로 수행 중인 다중 매체 MHS 기능 개발 및 RIU 고속 접속 기능 개발 연구 결과를 실었다.

제 2 장 DQDB MAN 프로토콜

제 2 장 DQDB MAN 프로토콜

2.1 DQDB MAN 프로토콜의 구조

IEEE 802.6 위원회에서는 호주의 Telecom Australia에서 제안한 QPSX (Queued Packet and Synchronous Exchange) MAN망을 DQDB (Distributed Queue Dual Bus)라 이름하여 MAN의 표준화 작업을 추진하고 있다. DQDB 망은 두개의 버스 구조를 갖는 고속의 데이터 경로를 통하여 비연결형 데이터, 연결형 데이터, 동시성 데이터 등의 다양한 데이터들을 수용한다. DQDB 망에서 이중 버스의 구조는 그림 2.1에 나타난 것과 같이 두개의 단방향 버스에 여러개의 노드들이 접속된 형태로 구성된다.

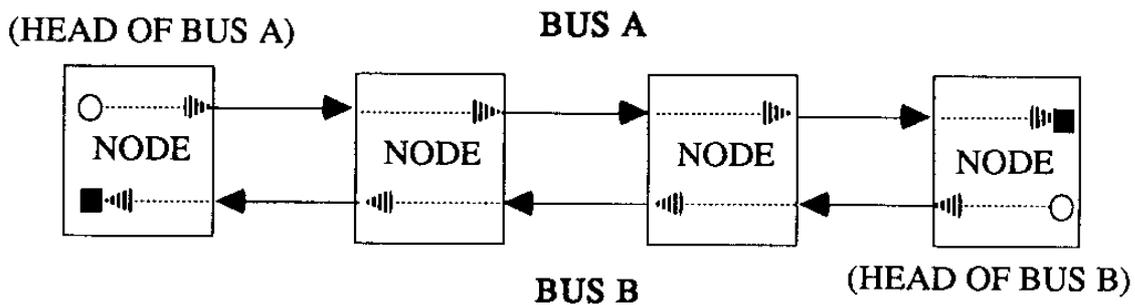
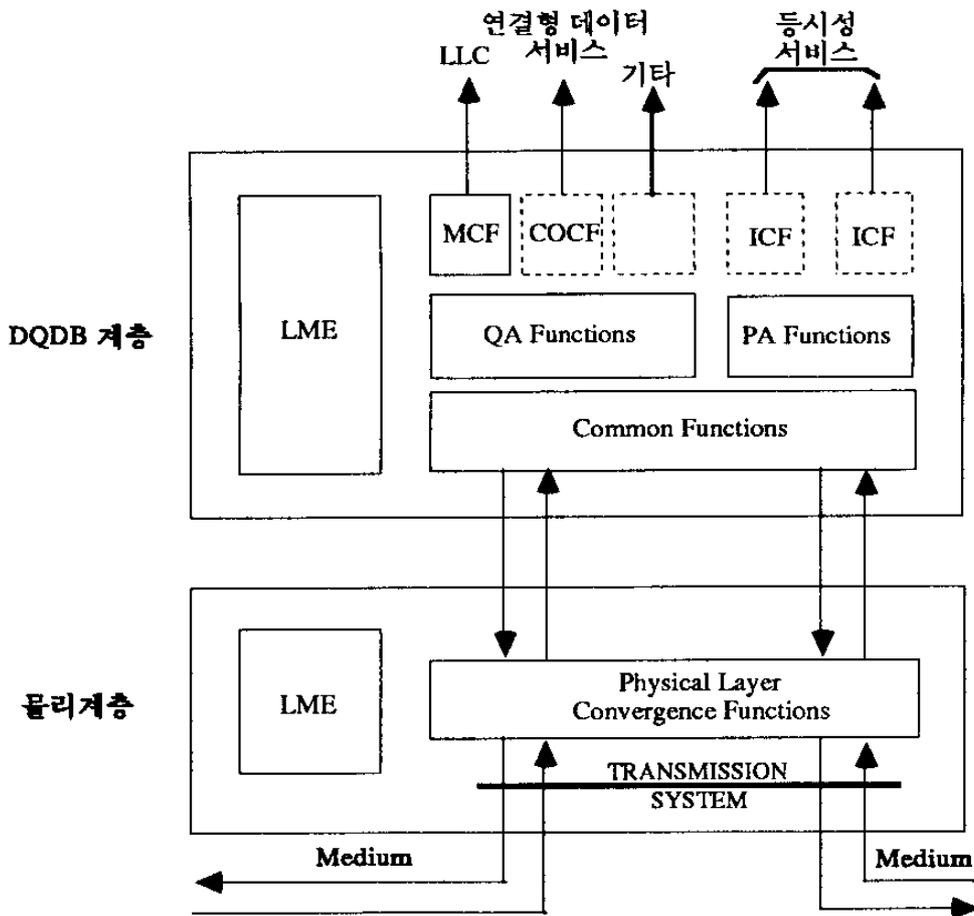


그림 2.1 DQDB의 이중 버스 구조

버스 A와 버스 B로 표시된 두개의 버스는 서로 반대 방향으로 데이터를 전송하며 각각의 노드는 데이터의 전송 방향에 따라 두 버스 중의 하나를 이용하여 원하는 곳에 데이터를 전송할 수 있다. DQDB 망에서는 서로 독립적으로 운용되는 두개의 버스가 동시에 작동함으로써 단일 버스에 비하여 망의 용량을 두배로 증가시킬 수 있다. 버스의 양단에 위치한 노드는 헤드라고 불리우며 각 버스의 헤드에서는 일정한 길이의 슬롯을 생성한다. 생성된 슬롯은 버스를 따라 전달되며 각 노드는 빈 슬롯에 준비된 데이터를

삽입하여 전송한다. 모든 슬롯들은 각 버스의 끝단에서 폐기된다. 망의 정상적인 작동을 위하여서는 망 내에 슬롯 타이밍을 관리하는 하나의 노드(일반적으로 헤드노드)가 필요하며 분산된 큐들로 구성된 모든 노드들은 이 슬롯 타이밍을 기준으로 하여 데이터를 전송함으로써 모든 노드들이 같은 비율로 데이터들을 주고 받게 된다.

IEEE 802.6 DQDB MAN은 크게 MAC 계층 (또는 DQDB 계층) 과 물리 계층으로 구분되며 그림 2.2에 각 노드의 기능에 대한 구조가 나타나 있다.



* 주 : LLC = Logical Link Control
COCF = Connection Orient Convergence Function
ICF = Isochronous Convergence Function
LME = Layer Management Entity

그림 2.2 DQDB 노드의 기능 구조

DQDB 망은 그림 2.2에 나타난 것과 같이 LLC (Logical Link Control) 계층으로부터 전달되는 비연결형 데이터 서비스와 연결형 데이터 서비스 및 등시성 데이터 서비스 등을 제공하도록 하고 있다. DQDB 망의 DQDB 계층은 두가지 접속 방식을 제공한다. 하나는 연결형 및 비연결형 데이터 등과 같은 비등시성 서비스를 위한 QA (Queued Arbitrated) 접속이고 다른 하나는 음성과 화상 등과 같은 등시성 데이터를 전송하기 위한 PA (Pre-Arbitrated) 접속이며 이들은 각각 QA 및 PA 슬롯을 사용하여 준비된 데이터를 전달한다. 헤드에서는 default로 QA슬롯을 생성하나 어떤 노드 에서든지 등시성 데이터 전송의 요구가 발생하면 PA 슬롯을 생성한다.

비연결형의 데이터인 경우 송신 노드에서는 LLC로부터 들어오는 MSDU (MAC Service Data Unit)를 QA 슬롯에 삽입할 수 있는 일정한 길이로 분리하여 QA 기능 블럭으로 보내고 수신 노드에서는 분리되어 들어온 데이터를 재조립하여 LLC로 보낸다. MSDU를분리하고 재조립하는 작업은 MCF (MAC Convergence Function)에서 수행되며 이러한 과정이 그림 2.3에 나타나 있다.

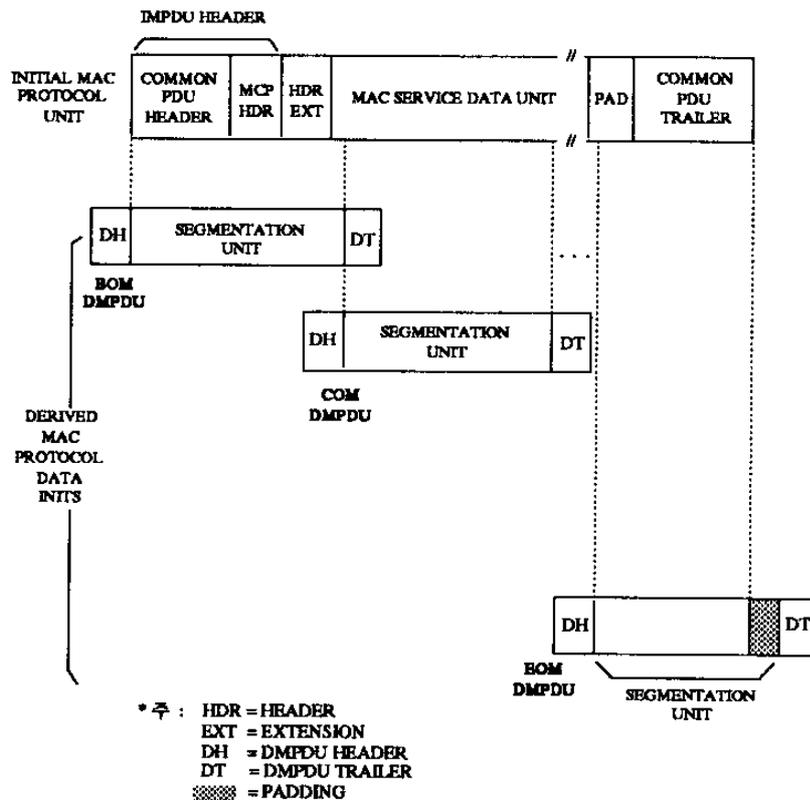


그림 2.3 MSDU 의 분리 과정

그림 2.3에 나타난 바와 같이 MSDU에 데이터의 길이, 송신 주소, 수신 주소, 우선 순위 등급 등을 수록한 헤더 (Common PDU Header)와 트레일러 (Common PDU Trailer) 등을 첨부하여 IMPDU (Initial MAC Protocol Data Unit)를 만든다. 이 IMPDU는 여러개의 SU(Segmentation Unit)로 나뉘어지고 이 SU에 SU 형태, MID (Message Identifier), CRC 등을 수록한 헤더와 트레일러를 첨부하여 헤더에서 생성되는 슬롯에 삽입할 수 있는 길이의 DMPDU (Derived MAC Protocol Data Unit)를 만든다. 하나의 IMPDU에서 파생된 DMPDU들은 MID로 확인되며 이를 통하여 수신 노드에서는 DMPDU를 IMPDU로 재조립한다. MCF에서 48옥텟으로 생성된 비등시성 데이터의 세그먼트들은 QAF(QA Function)를 통하여 전송된다.

그림 2.2에 나타난 COCF(Connection-Oriented Convergence Function)는 연결형 데이터의 전송 기능을 담당하며 데이터 경로의 설정, 유지 및 단절에 관한 절차는 상위 계층과 밀접한 관계를 가지고 수행되어야 할 사항이나 이에 대한 정의는 아직 완성되지 않은 상태이다. COCF 기능도 MCF 기능과 마찬가지로 데이터의 분리 및 재조립 과정을 거쳐서 QAF(QA Function)을 통하여 데이터를 주고 받는다.

등시성 데이터는 PAF(PA Function)를 통하여 전송된다. 그러나 헤더에서 PA 슬롯이 정확한 주기로 생성되지 않기 때문에 등시성 데이터의 전송을 위하여서는 버퍼링 기능이 필요하며 이러한 기능이 ICF (Isochronous Convergence Function)에서 수행된다. 등시성 서비스의 제공을 위하여서는 연결형 데이터의 경우와 마찬가지로 경로의 설정, 유지 및 단절에 관한 절차가 필요하나 이에 대한 정의는 아직 완성되지 않은 상태이다.

CF(Common Function)는 PA 및 QA 슬롯을 물리계층으로 전송하는 기능과 함께 망 관리를 위한 정보를 제공하는 기능을 수행한다. DQDB계층에서의 망관리 기능에는 각 버스에 헤더의 역할을 수행하는 노드를 제공하는 기능, 노드 또는 링크에 결함이 생겼을 경우 이중 버스 구조를 유지시켜주는 기능, 각 노드에 MID를 할당하는 기능 및 노드들 간에 타이밍을 동기화(synchronization)시키는 기능 등을 수행한다.

QAF에서는 비등시성 데이터의 전송을 위한 QA 접속 방식을 제공하고 PAF에서는 등시성 데이터의 전송을 위한 PA 접속 방식을 제공하며 이들에 대한 설명이 다음에 나타나 있다.

(1) QA 접속 방식

버스 A에 대한 QA 접속 방식의 기본 동작이 그림 2.4에 나타나 있다. 버스 B는 버스 A에서와 같은 동작을 역방향에서 독립적으로 수행한다.

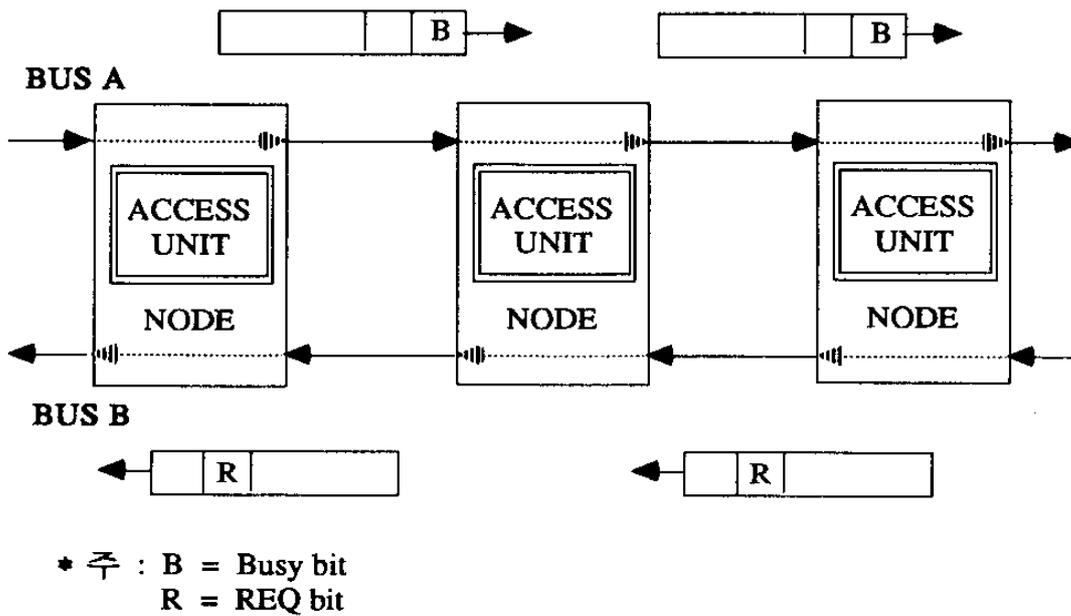


그림 2.4 QA 접속 방식의 기본 동작

QA 접속 방식은 QA 슬롯을 통한 데이터 전달을 제어하며 슬롯 내의 두개의 제어 필드를 기본으로 하여 동작된다. 하나는 슬롯이 데이터 전송에 이미 사용되었는가를

나타내는 BUSY 비트 필드이고 다른 하나는 보내야 할 데이터의 세그먼트가 큐에 대기하고 있음을 역방향의 버스를 통하여 다른 노드들에 알려주는 REQ (request) 비트 필드이다. 각 노드의 AU(Access Unit)에 QA 세그먼트가 도착하면 세그먼트가 전송 되어야 할 방향의 반대 방향으로 흐르는 슬롯의 REQ 필드 값을 1로 세트시킴으로써 자기보다 왼쪽(헤드에서 가까운 쪽)에 위치한 노드들에게 QA 세그먼트가 큐에 대기하고 있음을 알린다. 각각의 노드의 AU에는 RQ (request) 계수기가 있어 역방향으로 1로 세트된 REQ 비트가 통과할 때마다 계수기의 값을 증가시키고 정방향으로 빈 슬롯이 통과할 때마다 값을 감소시킴으로써 항상 자기보다 아래쪽 (헤드에서 먼 쪽)에 위치한 모든 노드들에 대기하고 있는 QA 세그먼트들의 전체 갯수를 유지한다.

AU에 전송해야 할 세그먼트가 도착하면 RQ 계수기의 값을 CD (countdown) 계수기로 전환하고 RQ 계수기의 값은 다시 0으로 전환하여 앞서 설명한 과정을 반복한다. CD 계수기의 값은 정방향으로 빈 슬롯이 통과될 때마다 감소되며 CD 계수기의 값을 0으로 하는 빈 슬롯을 이용하여 대기하고 있던 QA 세그먼트를 전송한다. 이 방식은 망 내에 먼저 도착한 세그먼트가 먼저 전송되도록 함으로써 헤드에서 가까운 노드가 빈 슬롯을 독점하는 효과를 줄이기 위한 것이지만 슬롯 통과 시간 (slot propagation delay) 내에 같이 도달하는 QA 세그먼트들에 대해서는 헤드에서 가까운 쪽의 노드가 전송의 우선권을 갖게 된다. 이로 인하여 망의 트래픽이 증가함에 따라 노드들이 망의 용량을 공평하게 사용하지 못하는 경우가 발생할 수 있으며, 노드 간의 거리는 DQDB 망의 공평성(fairness)에 큰 변수로 작용한다.

높은 트래픽에서도 망의 공평성을 유지하기 위하여 DQDB 망에서는 헤드에서 가까운 쪽의 노드가 때때로 빈 슬롯의 사용권을 헤드에서 먼 쪽의 노드에 넘겨줌으로써 분산된 큐들이 망의 용량을 효과적으로 공유하도록 하는 기능을 제공하고 있다. 이 기능은 BWB_MOD (bandwidth balancing modulus)라는 시스템 파라미터와 각 노드 내에 있는 BWB 계수기으로써 작동된다. 각 노드에서는 QA 세그먼트가 전송될 때마다 BWB 계수기의 값을 하나씩 증가시킨다. BWB 계수기 값이 (BWB_MOD-1)에 달하면 QA 세그먼트가 대기하고 있지 않은 상태에서는 RQ 계수기의 값을 하나 증가시키고, QA 세그먼트가

대기하고 있는 경우에는 CD 계수기를 하나 증가시킴으로써 다음에 오는 빈슬롯의 사용권을 헤드에서 먼쪽의 노드에게 넘겨준다. 이때 BWB 계수기의 값은 다시 0으로 전환된다. 주어진 트래픽의 상태에 따라 BWB_MOD의 값을 어떻게 설정하여야 할 것인가에 관한 문제는 DQDB 망을 효과적으로 운용하기 위해서 앞으로 연구되어야 할 중요한 문제 중의 하나이다.

분산 큐 프로토콜은 전송될 QA 세그먼트의 중요도에 따라 세 단계의 우선 순위 기능을 제공한다. 각 노드 내의 AU는 세개의 분리된 우선 순위 큐들을 가지고 있으며 각각의 QA 세그먼트들은 우선 순위에 따라 해당되는 큐에서 대기한다. 우선 순위 기능은 역방향으로 흐르는 QA 슬롯 내의 세개의 우선 순위 REQ 비트 필드들에 의해서 작동된다. 각각의 우선 순위 큐들은 독립된 RQ 및 CD 계수기들을 가지고 있으며 QA 세그먼트가 대기하고 있지 않을 경우에는 역방향에서 자기와 같거나 자기보다 높은 우선 순위의 REQ 비트가 통과할 때마다, 또는 같은 노드에서 자기보다 높은 우선 순위의 세그먼트가 도착할 때마다 RQ 계수기의 값을 증가시키고 정방향으로 빈 슬롯이 통과할 때마다 RQ 계수기의 값을 감소시킨다. 우선 순위 큐에 QA 세그먼트가 대기하고 있을 경우에는 RQ 계수기는 역방향에서 자기와 같은 우선 순위의 REQ 비트가 통과할 때마다 RQ 계수기의 값을 증가시킨다. 그러나 CD 계수기는 역 방향에서 자기와 같거나 자기보다 높은 우선 순위의 REQ 비트가 통과할 때마다, 또는 같은 노드에서 자기보다 높은 우선 순위의 세그먼트가 도착할 때마다 그 값을 증가시키고 정방향에서 빈 슬롯이 통과할 때마다 그 값을 감소시키며 CD 계수기의 값을 0으로 하는 빈 슬롯을 이용하여 대기하고 있던 QA 세그먼트를 전송한다. DQDB 망에서의 우선 순위 기능은 높은 우선 순위의 데이터가 낮은 우선 순위 데이터에 영향을 받지않고 우선적으로 전송되도록 의도하고 있으나 주어진 트래픽 조건 하에서 요구되는 성능을 만족시키기 위한 우선 순위 부여 문제는 우선 순위 기능 자체에 대한 성능 평가와 함께 앞으로 연구되어야 할 사항 중의 하나이다.

(2) PA 접속 방식

PA 슬롯은 등시성 데이터 옥텟을 전송하는데 사용되며 헤드 노드에서 생성된다. PA

슬롯은 48개의 옥텟 길이로 구성되며 각각의 옥텟을 서로 다른 AU들이 사용함으로써 하나의 PA 슬롯을 여러 노드들이 공유한다. 버스의 헤드는 PA 슬롯을 생성하기 이전에 등시성 데이터를 송/수신하는 AU들에 VCI(Virtual Channel Identifier) 값을 준다. 각각의 AU는 어떤 VCI 값을 갖는 PA 슬롯의 몇번째 옥텟부터 몇번째 옥텟 까지를 사용하여 송/수신할 수 있는가를 미리 알고 있어야 하며 이는 DQDB 계층 관리 기능을 통하여 주어진다. 그러나 이러한 절차에 대한 규정은 아직 완성되지 않은 상태이다.

그림 2.2에 나타난 것과 같이 DQDB 망의 물리계층은 전송시스템, PLCF (Physical Layer Convergence Function) 및 LME (Layer Management Entity) 등으로 구성된다. DQDB 망은 특정한 전송시스템에 제한을 두고 있지 않고 있다. PLCF는 DQDB 계층이 특정한 전송시스템에 구애받지 않고 독립적으로 기능을 수행하도록 하기 위하여 제공되며 DQDB 계층에서 제공되는 PA 슬롯, QA 슬롯과 망 관리 정보 들을 주어진 전송 시스템을 통하여 전송될 수 있도록 그 포맷을 변형시키는 기능을 수행한다. 따라서 각각의 전송시스템은 그에 맞는 PLCF를 필요로 하며 현재 기본 표준안에는 ANSI DS3 (44.736 Mbps)에 대한 PLCF 만이 정의되어 있다. 물리 계층의 LME는 노드 및 링크의 고장탐지, 고장난 노드의 고립화 등의 기능을 수행한다.

2.2 DQDB MAN 표준화 동향

IEEE 802.6 위원회에서 작성 중인 DQDB MAN 표준안은 (1990년 8월 6일) 현재 Draft Version 14 까지 나와있으며 지금까지 많은 진척을 보여왔지만 앞으로도 해결해야 할 몇가지 문제점들을 가지고 있다. 이 절에서는 DQDB MAN의 표준화 진척 상황과 현재 IEEE P802.6에서 논의되고 있는 쟁점들에 대하여 언급하기로 한다.

DQDB MAN의 성능은 그동안 많은 사람들에 의하여 검토되어왔으며 특히 공평성 (Fairness)과 우선 순위 부여 (Priority Scheme)에 대하여 문제점들이 제기되어 왔다. DQDB MAN의 초기 version에는 구조적으로 헤드에서 가까운 쪽에 위치한 노드가 빈 슬롯을 우선적으로 사용 함으로써 망의 용량이 각 노드에 공평하게 분배되지 못하는 특성이

있었다. 이러한 단점을 보완하기 위한 방안으로 각 노드에 RQ 계수기 (Request Counter)와 CD 계수기 (Countdown Counter)를 추가함으로써 먼저 도착한 데이터에 전송의 우선권을 주어 공평성에 대한 문제를 크게 완화하였다. 그러나 슬롯의 통과 지연 (propagation delay)에 의한 영향으로 노드간의 거리가 멀 경우 (즉, 노드와 노드 사이에 많은 슬롯들이 통과될 경우) 트래픽이 증가할수록 헤드에서 가까운 쪽에 위치한 노드들이 빈 슬롯을 차지하는 비율이 증가하여 공평성에 대한 문제를 완전히 해결하지는 못하였다. 이를 위하여 P802.6 위원회에서는 BWB (Bandwidth Balancing) 계수기를 추가함으로써 헤드에서 가까운 쪽의 노드들은 일정한 수의 빈 슬롯을 사용한 후 다음에 도착하는 빈 슬롯은 그대로 통과 시켜 헤드에서 먼 쪽의 노드가 사용할 수 있도록 하였다. 그러나 주어진 트래픽 조건 하에서 몇개까지의 빈 슬롯을 사용할 것인가에 대한 기술적인 문제의 해결책은 아직 제시되지 않고 있는 상태이며 이 문제는 앞으로 많은 연구가 수행되어야 할 사항이다. 현재까지 완료된 DQDB MAN의 표준안은 다음과 같은 트래픽 조건하에서는 공평성에 대하여 만족할 만한 성능을 나타내는 것으로 알려지고 있다.

1. 노드에 부여되는 트래픽이 높지 않은 경우
2. 각 노드에 부여되는 트래픽이 서로 비슷한 경우
3. 노드간의 거리가 멀지 않은 경우
4. 모든 노드가 같은 우선 순위를 가지고 작동될 경우

1항과 2항은 IEEE 802.6 DQDB 표준안이 MAN 망으로 타당한가에 관한 문제 라기보다는 망 사용자가 어떻게 망을 운용할 것인가에 대한 기술적인 문제라고도 볼 수 있을 것이다. 그러나 3항에 대한 문제는 망의 규모 및 데이터의 전송 속도 등에 따라 직접적으로 영향을 받는 요소이므로 비중있게 고려되어야 할 사항이다. 즉, 53 옥텟의 길이를 가지는 슬롯이 44.736Mbps의 전송 속도에서는 2Km, 155.520Mbps에서는 546m, 622.080Mbps에서는 137 m마다 하나씩 위치하게 되어 데이터 전송 속도와 망의 규모에 상관관계가 존재하며 따라서 DQDB MAN 망이 현실적으로 최고 155 Mbps의 전송 속도로 제한될 수 밖에 없을 것이라는 예측도 나오고 있다[10]. 여러개의 우선 순위가 동시에 작동될 경우 노드간의 거리에 대한 문제는 더욱 심각해지며, 이때에는 노드간의 거리 뿐만이 아니라

높은 우선순위 노드와 낮은 우선순위 노드들간에 상대적인 위치까지도 망의 성능에 영향을 미치게 된다. 시뮬레이션 모델을 통하여 보고된 결과에 의하면 트래픽이 높을 때 우선순위 노드들이 헤드로부터 H-H-L-L-L-L-H-H (H:높은 우선순위, L:낮은 우선순위)의 순서로 배치되면 낮은 우선순위의 노드들과 헤드에서 먼쪽에 위치한 높은 우선순위 노드들간에 망이용도의 차이가 거의 없어져 우선순위 기능이 효과적으로 작동되지 않는 경우가 발생한다. 망의 공평성과 우선순위 기능 문제는 트래픽 상태, 노드간의 거리, 전송속도, 노드의 위치 등 여러가지 복합적인 요소를 가지고 있다. 이 문제를 해결하기 위하여 현재 네단계의 우선 순위를 세단계로 줄이고 슬롯 헤더의 ACF(Access Control Field)에 reserved 비트를 두개로 확장함으로써 우선 순위 기능이 효과적으로 작동하는데 쓰이도록 하는 등의 새로운 방식들이 제안되고 있으나 복합적인 문제를 일시에 해결할 수 있는 방안을 제시하기는 어려울 것으로 보이며 국부적인 문제들을 해결해 나가면서 어느 선에서 타협점을 찾는 방향으로 귀결될 것으로 보인다

IEEE 802.6 MAN의 기본 표준안은 비연결형 데이터 서비스를 위한 MAC 계층과 DS3 물리 계층으로 구성된다. 현재까지 P802.6 위원회의 주요 활동은 기본 표준안의 작성에 치중되어 왔으나 충분한 MAN 서비스를 제공하기 위하여서는 다음과 같은 사항들에 관한 표준안의 작성도 병행되어야 하며 앞으로 추진될 계획에 있다[10].

1. Multiport bridge interface: 완전한 MAN 망은 이중 버스의 구조를 갖는 여러개의 DQDB subnetwork들을 서로 연결함으로써 구축될 수 있다. DQDB subnetwork들의 연결은 Multiport bridge를 통하여 이루어진다. IEEE 802.6A (Multiport bridge Subworking Group)에서는 이를 위하여 다음과 같은 사항들에 대한 표준안을 추진하고 있다.

- o Architecture of the Multiport Bridge
- o Bridging of service types and interfaces
- o Routing and multicast
- o Address mapping and conversion
- o Flow and congestion control

o Data collection for charging

2. DS3 premises extension: 현재 802.6 기본 표준안은 미국의 전화 회사들에서 44.736Mbps로 디지털 음성을 전송하는데 주로 이용되는 전기적 인터페이스의 DS3를 물리 계층의 전송시스템으로 채택하고 있다. 그러나 앞으로의 광통신 요구에 부응하기 위하여서는 광통신 인터페이스를 위한 표준안이 작성되어야 할 필요가 있으며 이를 위하여 광섬유의 형태, 커넥터, 전송 파워등에 대한 표준안의 작성을 IEEE 802.6B 위원회에서 추진하고 있다.
3. DS1 (T1) physical layer: 원래의 MAN 망은 고속 통신망을 제공하는 것을 원칙으로 하고 있다. 그러나 MAN이 사설망으로 사용될 경우 customer와 central office간에 전용선이 설치되어야 하며 이때에는 고가의 광통신 장비를 설치하는 것보다는 저가의 구리선을 사용하는 것이 경제적이다. 따라서 현재 미국 내에 널리 보급되어 있는 1.544 Mbps의 DS1 (T1)을 DQDB MAN의 전송 시스템으로 사용할 수 있도록 표준안을 제정할 필요를 느끼게 되었으며 이 작업을 IEEE 802.6C 위원회에서 추진하고 있다. 그러나 이 안은 T1이 널리 보급되어 있는 북미나 일본 등지에 국한되는 것이고 DQDB MAN이 ISO 국제 표준안으로 제안될 경우 이 안은 문제를 일으킬 소지가 있다.
4. SONET physical layer: 데이터의 고속 전송을 위하여서는 DQDB 물리 계층의 전송 시스템으로 155Mbps의 SONET 물리 계층을 채택하였으며 이의 표준안 작성을 IEEE 802.6D 위원회에서 하고있다. SONET[8]은 국제 표준안이므로 DQDB MAN이 ISO 국제 표준안으로 제안될 경우에 용이하게 통과될 것이다.
5. Eraser Node: DQDB의 MAC(Medium Access Control)계층에서는 throughput을 증가시키기 위한 방법으로 eraser node를 두고 있다. 즉, eraser node에서는 이미 수신이 완료된 데이터를 싣고 있는 슬롯의 데이터 세그먼트 부분을 지워서 빈 슬롯화함으로써 다른 노드에서 이 슬롯을 재사용하도록 한다. DQDB MAN의 기본 표준안에는 이러한 기능을 수행하도록 하기 위하여 슬롯의 헤더의 ACF에 PSR (previous segment received)

비트 영역을 제공하여 각 (수신)노드에서는 바로 전 슬롯의 데이터 세그먼트의 수신이 완료되었는가를 다음에 도달하는 슬롯의 헤더에 표시하도록 하고 있다. Eraser node에서는 도착하는 슬롯을 잡고 있다가 다음에 도달하는 슬롯의 헤더에 PSR비트가 세트되지 않았으면 대기하고 있던 슬롯을 그대로 통과시키고, PSR 비트가 세트되었으면 대기하고 있던 슬롯을 빈 슬롯화한다. 이러한 과정을 수행함에 있어서 eraser node는 한 슬롯 길이만큼의 시간 지연을 초래하며 따라서 모든 노드가 eraser node의 기능을 갖는 것은 바람직하지 않다. Eraser node 기능은 망의 용량을 크게 증가시킬 수 있으며 이의 표준화 작업이 802.6E 위원회에서 수행되고 있다.

6. Conformance (PICS Proforma): 현재 ISO에서는 새로운 표준안이 제안될 때 PICS (Protocol Implementation Conformance Statement) Proforma를 요구하고 있다. 따라서 IEEE 802.6이 ISO표준안으로 제안될 경우 이에 대비하여 PICS Proforma를 작성하여야 할 필요가 있으며 이에 대한 작업을 802.6F에서 수행하고 있다.
7. Layer management: 802.6G 위원회에서는 DQDB MAN의 계층 관리 (layer management)에 대한 표준안을 작성하고 있다. 이 작업은 IEEE 802.1 위원회와 협력하에 이루어지고 있으며 802.1B (LAN/MAN Management)와 802.1F (Guideline for the development of layer management standards)를 바탕으로 작성되고 있다.
8. Isochronous services over the MAN: 802.6위원회에서는 처음부터 MAN이 데이터뿐만 아니라 음성 및 화상과 같은 등시성 데이터의 서비스를 제공할 수 있도록하는데 역점을 두었다. 현재 기본 표준안에는 데이터 전송을 위한 비연결형 서비스에 대하여서는 잘 정의되어 있으나 다른 서비스들에 대하여서는 응용시스템들의 시장성 등을 감안하여 그 표준안 작업이 연기되었다. 현 기본 표준안은 등시성 데이터의 전송을 위한 세그먼트의 구조 등에 대하여서는 정의되어 있으나 등시성 데이터의 전송을 위한 경로의 설정, 유지 및 단절에 대한 표준안은 아직 작성되지 않은 상태이다. 따라서 등시성 데이터 서비스에 대한 표준안 작업을 802.6H에서 수행하고 있으며 앞에서 언급한 사항 이외에 time jitter의 허용치 등 물리 계층에 관한 표준안도 제정할 계획에

있다.

9. Remote bridging via MAN: MAN은 앞으로 몇년 동안은 여러 곳에 분산된 LAN들을 서로 연동하는데 주로 이용될 것으로 예측되고 있다. 즉, 기존의 LAN연결 방식인 point-to-point line들이 MAN으로 대체될 것이다. 이를 위하여서는 (헤더와 CRC를 포함하는) LAN frame은 encapsulation되어 MAN을 통하여 전송되어야 할 것이다. 그러나 사용자들은 여러 vendor들을 통하여 MAN장비를 구입할 것이며 따라서 encapsulation 방식에 대한 표준안이 제정되어야 할 필요가 있다. 802.6I에서는 802.x (x=3,4,5) LAN들이 MAN을 통하여 용이하게 연동될 수 있도록 하기 위하여 encapsulation에 대한 표준안을 제정할 계획에 있다.

10. Connection-oriented data services: MAN은 연결형 데이터의 서비스를 제공할 수 있도록 계획되어 왔다. 연결형 데이터의 서비스에 대한 작업은 현재 CCITT에서 주로 수행하고 있으며 802.6J에서는 CCITT 안을 수용하여 DQDB MAN이 CCITT에서 제정 중인 B-ISDN (Broadband ISDN)에 용이하게 접속될 수 있도록 하기 위한 작업을 하고 있다.

IEEE 802.6 DQDB MAN은 이밖에 기본 표준안에서 (1) 10-bit CRC를 32-bit CRC로 대체하여 에러탐지 기능을 높여야 하며, (2) DMPDU에 sequence number를 주어 IMPDU로 재조립 시에 에러 탐지 기능을 보강해야 한다는 등의 새로운 제안이 들어오고 있는 상태이어서 기본 표준안의 완성이 원래 계획했던 일정보다도 상당히 지연되고 있는 상태이다.

기본 표준안의 작성이 지연되는 가장 큰 이유 중의 하나는 바로 DQDB MAN이 구조적으로 공평성 (fairness)와 우선 순위 기능(priority scheme)을 보장하지 못한다는 데 있다. 현재의 추세는 MAN이 높은 트래픽 상태로 장시간 지속될 확율이 적을 것으로 판단되어 이러한 문제들이 망의 성능에 크게 영향을 주지 않을 것이라는 쪽으로 기울고 있으나 다른 한편에서는 엔지니어의 입장에서 이미 노출된 문제를 해결한 후에 표준안을 제정해야 한다는 주장도 만만치 않게 나오고 있다. 그러나 DQDB MAN은 데이터뿐만 아니라 음성 및 동화상 서비스까지도 제공할 수 있고 또한 공중망으로 사용될 경우

B-ISDN과 같은 WAN과의 접속이 용이하다는 등의 장점이 있어 DQDB 표준안 작업은 계속 추진될 것이다. DQDB MAN의 표준안이 완성되면 AT&T, Alcatel, Siemens, QPSX 등에서 바로 이를 구현하여 상품화할 것으로 발표되고 있다.

제 3 장 DQDB MAN 블럭 설계

제 3 장 DQDB계층 블럭설계

LLC(Logical Link Control)에 MAC으로서 서비스를 제공하기 위한 DQDB MAC은 MCF(MAC Convergence Function), QA(Queued Arbitrated), Common Function 등 3개의 블럭에 의하여 수행한다. 메시지를 송신할 때 MCF 기능은 LLC로부터 받은 메시지를 주소, 우선순위, 프로토콜 구분에 따라 메시지당 IMPDU(Initial MAC PDU)를 생성하고, 생성된 IMPDU를 세그먼트 단위로 조각을 내어 QA 기능블럭에서 전송할 수 있도록 한다. QA 기능블럭은 MCF 블럭으로부터 받은 DMPDU(Derived MAC PDU) 를 전송할 수 있도록 QA 세그먼트로 만든 다음 분산 큐 알고리즘에 의하여 전송하게 된다. Common Function 블럭은 QA 기능블럭에서 QA 세그먼트를 전송할 수 있도록 슬롯의 ACF를 QA 기능블럭에 전달하여 QA 기능블럭이 하는 일을 보조한다. 한편 메시지를 수신할 때는 Common Function에 의해 QA 기능블럭으로 보내고 QA 세그먼트를 분산 큐 알고리즘에 의하여 수신한 다음 MCF에게 보낸다. MCF는 QA 기능블럭으로부터 받은 QA 세그먼트 정보필드를 DMPDU로 간주하여 이로부터 세그먼트단위를 추출하여 조립한 다음 IMPDU를 만들게 된다. IMPDU는 확인과정을 거쳐 MSDU를 추출하여 LLC에 보낸다. 이상과 같은 기능을 수행하는 DQDB MAC 계층을 프로토콜의 기능 측면에서 블럭 레벨로 기술하였다.

3.1 MCF 기능

MAC 수렴 기능은 상위계층(주로 LLC 계층)으로부터 전송할 PDU를 받아 미디어를 통해 전송하기 위한 사전 작업으로써 IMPDU를 생성한다. IMPDU는 IMPDU 분할 기능을 거쳐 메시지를 전송 알고리즘에 의하여 전송할 수 있도록 알맞는 크기로 분할된다. 분할된 메시지의 조각들을 세그먼트 단위(Segment Unit)이라 부르며, 전송될 수 있는 가장 큰 메시지의 크기가 되는데 그 크기는 44 옥텟이다.

세그먼트 단위는 다른 세그먼트 단위와 구분하기 위하여 세그먼트의 종류(예: SSM, BOM, COM, EOM)가 붙여지고 메시지 레벨에서 구분할 수 있는 MID(Message Identifier)가 붙여진다. 또한 세그먼트내에 메시지가 얼마나 들어있는지를 알려주는 Length 필드와 CRC를 붙여 세그먼트의 에러상태를 알 수 있도록 한다. 이러한 작업에 의해서 만들어진

PDU를 DMPDU라고 부른다. 따라서 MCF 송신기능은 LLC 계층으로부터 메시지를 받아서 IMPDU를 생성한 다음 분할과정을 거친 후 DMPDU를 생성하기까지의 작업이며 상위계층으로부터 받은 메시지를 전송 알고리즘에 의하여 전송될 수 있는 형태로 조각을 내어준비하는 단계를수행하는기능이라 할수 있다. 한편 MCF블럭은 QA 기능블럭으로부터 DMPDU를 수신하고 수신 DMPDU로부터 세그먼트단위를 추출하여 IMPDU 조립과정에 들어간다. 거기서 조립된 각각의 IMPDU는 IMPDU 확인 기능을 거치고 확인된 IMPDU로부터 MSDU를 추출하게 된다. 이러한 MSDU(MAC Service Data Unit)를 가지고 MCF 기능블럭은 LLC에 비연결형 데이터가 도달하였음을 알려주는 프리미티브인 MA_UNITDATA indication을 발생시킨다. MCF에서 수행하는 기능을 그림 3.1에 나타내었다.

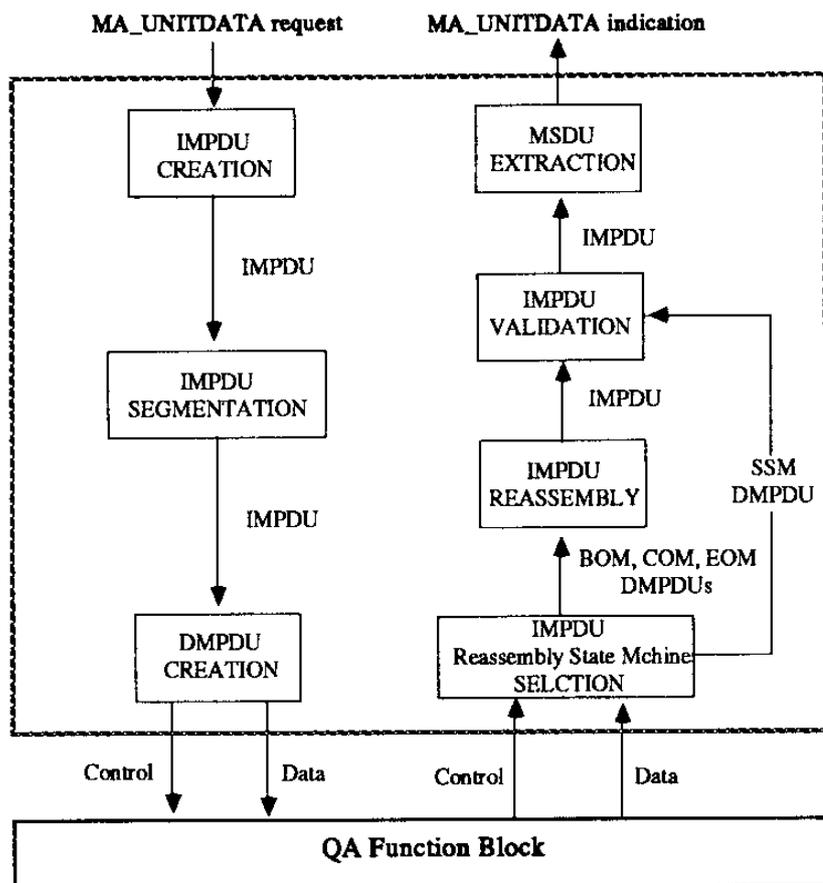


그림 3.1 MCF 블럭의 기능

3.1.1 MCF 송신 기능

MCF의 송신기능은 그림 3.2에 나타낸 바와 같이 IMPDU 생성블럭(IMPDU CREATION BLOCK), IMPDU 분할블럭(IMPDU SEGMENTATION BLOCK), DMPDU 생성 블럭(DMPDU CREATION BLOCK)등으로 구성된다.

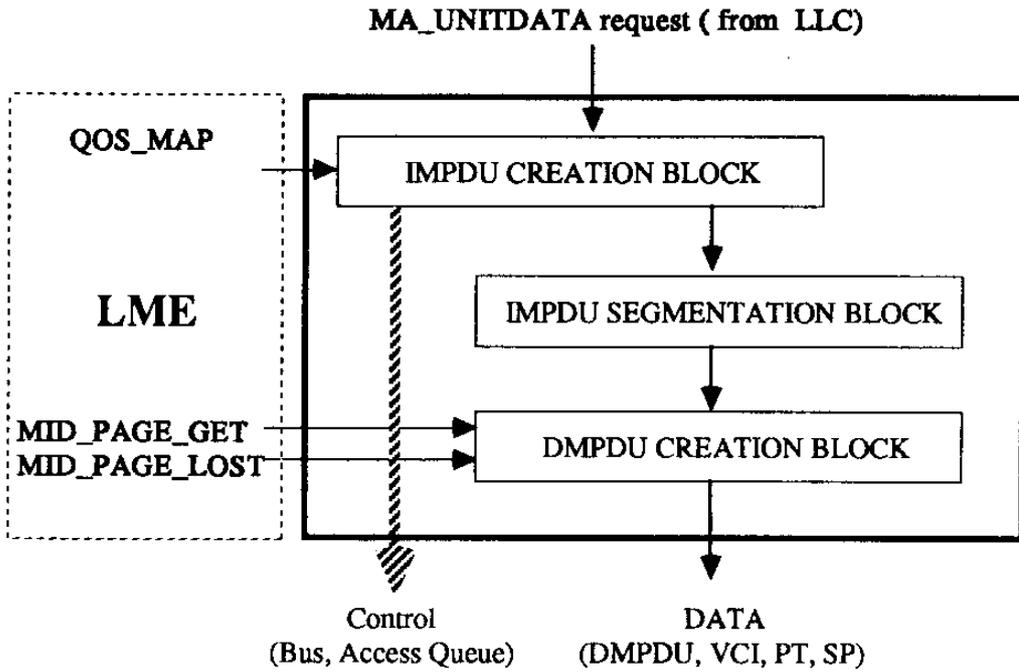


그림 3.2 MCF의 송신 기능

3.1.1.1 IMPDU 생성 블럭

LLC에서 다음과 같은 프리미티브를 통하여 데이터를 MAC으로 보내게 된다.

- MA_UNITDATA request (
 - source_address,
 - destination_address,
 - priority,
 - data,
 - service_class

)
 여기서

- source_address, destination_address : MAC address
- priority : 메시지의 전송 우선순위
- data : LLC PDU
- service_class : 서비스 등급

IMPDU 생성 블록은 LLC로부터 MA_UNITDATA request를 받으면, MSDU(MAC Service Data Unit) 인 MA_UNITDATA request내에 포함된 파라미터로부터 IMPDU를 생성한다. IMPDU 생성 블록의 기능은 Common PDU 헤더 생성MCP(MAC Convergence Protocol) 헤더 생성, 헤더확장 필드 생성, INFO 필드 생성, PAD 필드 생성, Common 트레일러 생성 등 6 부분으로 설명할 수 있다. Common PDU 헤더 생성 부분은 Reserved, BEtag, BAsize 등 필드들을 코드한다. MCP 헤더 생성 부분은 목적지 및 출발지 주소, PI(Protocol Identifier), QOS_DELAY, QOS_LOSS, HEL, BRIDGING 등 필드들을 코드한다. 헤더확장 필드 생성 부분은 IMPDU의 HE_selection_info 값의 일치에따라 HE_value 값으로 코드한다. INFO 필드 생성은 MSDU에 따라 코드한다. PAD 필드 생성은 정보 필드의 길이와 PAD 필드의 합이 4 옥텟의 정수가 되도록 코드한다. Common 트레일러 생성부분은 Reserved, BEtag, Length 필드 등을 코드한다. 이와 같이 하여 생성된 IMPDU의 포맷을 그림 3.3에 나타내었다.

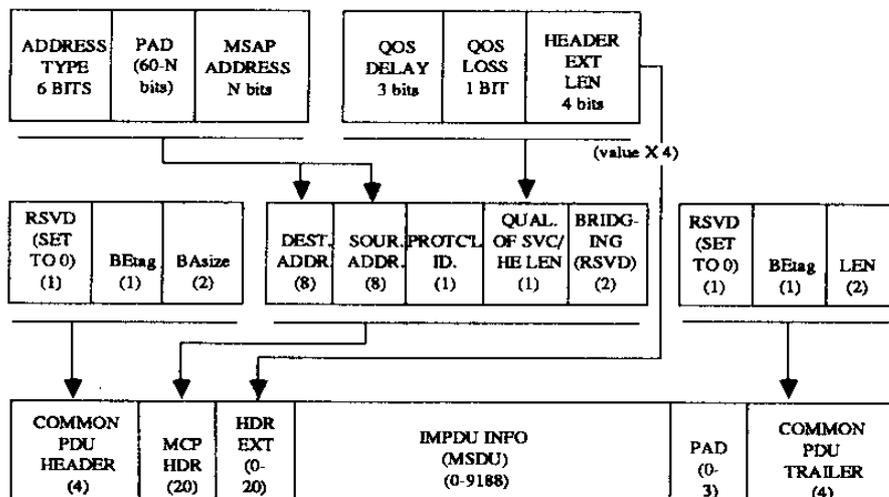


그림 3.3 IMPDU의 포맷

위 그림 3.3에 나타낸 IMPDU를 생성하기 위한 절차의 흐름도를 나타내면 아래와 같다.

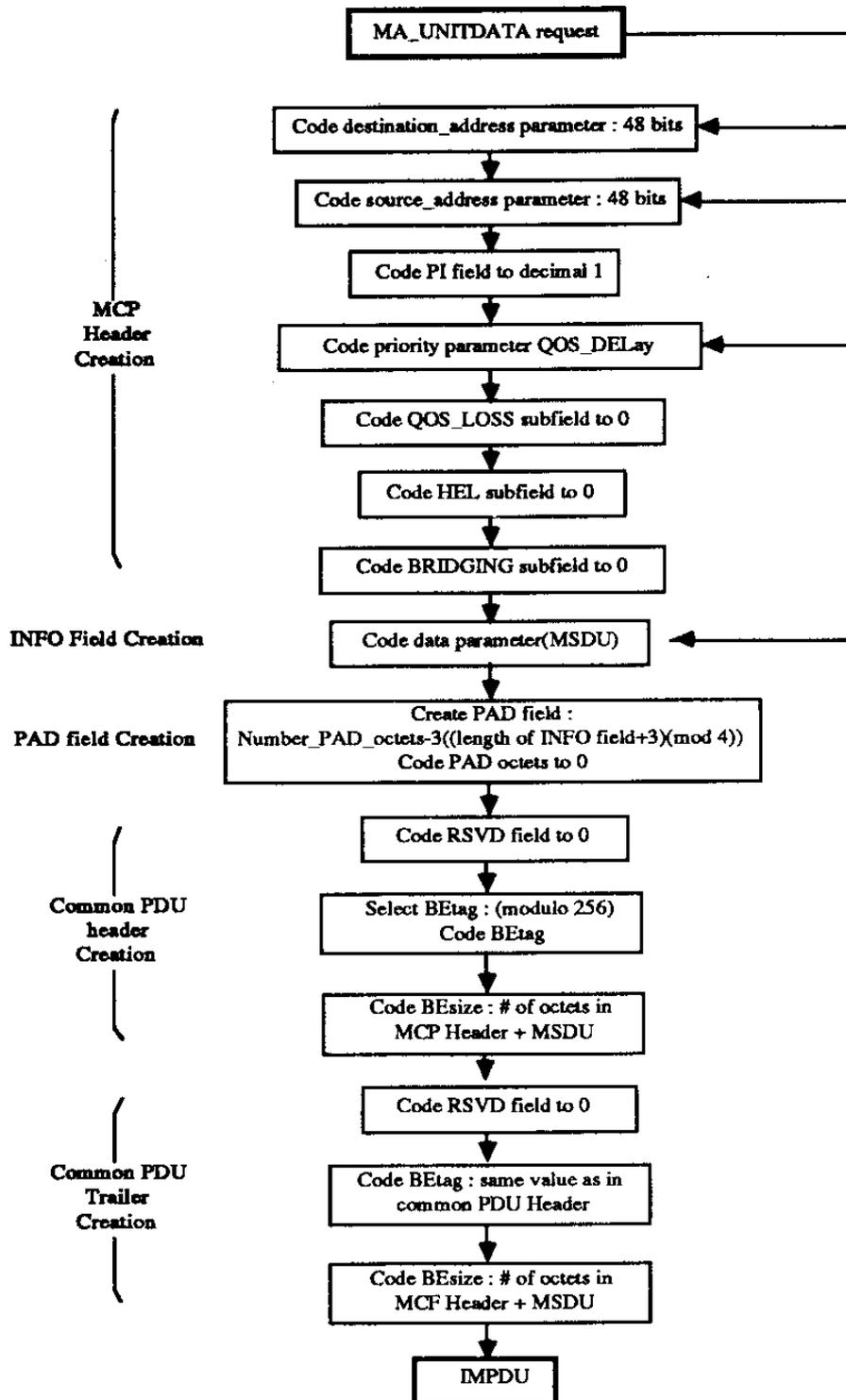


그림 3.4 IMPDU를 생성하는 과정

3.1.1.2 IMPDU 분할 블럭

MCF 블럭은 이미 앞에서 언급한 바와 같이 하나의 IMPDU를 하나 혹은 여러개의 세그먼트로 분할한다. 각 세그먼트의 단위는 44 옥텟 길이를 갖게되며, IMPDU의 마지막 부분을 제외하면 44 옥텟의 IMPDU를 포함하게 된다. 세그먼트 단위의 수와 형태는 IMPDU의 길이에 따라 결정된다. 세그먼트를 하는 과정에서 IMPDU의 길이가 44 옥텟의 길이보다 적거나 같게되면 단지 하나의 세그먼트 단위가 생성되며 이를 SSM(Single Segment Message) 세그먼트 단위라고 한다. 하나의 SSM은 모든 IMPDU를 포함하게 된다. 만일 하나의 IMPDU가 44 옥텟보다 크게되면, 여러개의 세그먼트 단위로 분할되는 과정을 거치게 되는데, 첫번째 세그먼트 단위에게는 BOM(Beginning Of Message) 세그먼트 단위라고 부르고 마지막 세그먼트는 EOM(End Of Message) 세그먼트라고 한다. 이때 하나의 IMPDU가 2개 이상의 길이를 갖게되면 중간의 세그먼트는 COM(Continuation Of Message)라고 한다. 이와 같은 관계를 그림 3.5에 나타내었다. 분할된 세그먼트에 SSM, BOM, COM, EOM 등의 이름을 붙이는 것은 세그먼트가 상대 노드에 전달되었을 때, 조립하기 용이 하도록하는 것이며 전달되는 도중에 세그먼트의 순서가 바뀌는 경우는 없다.

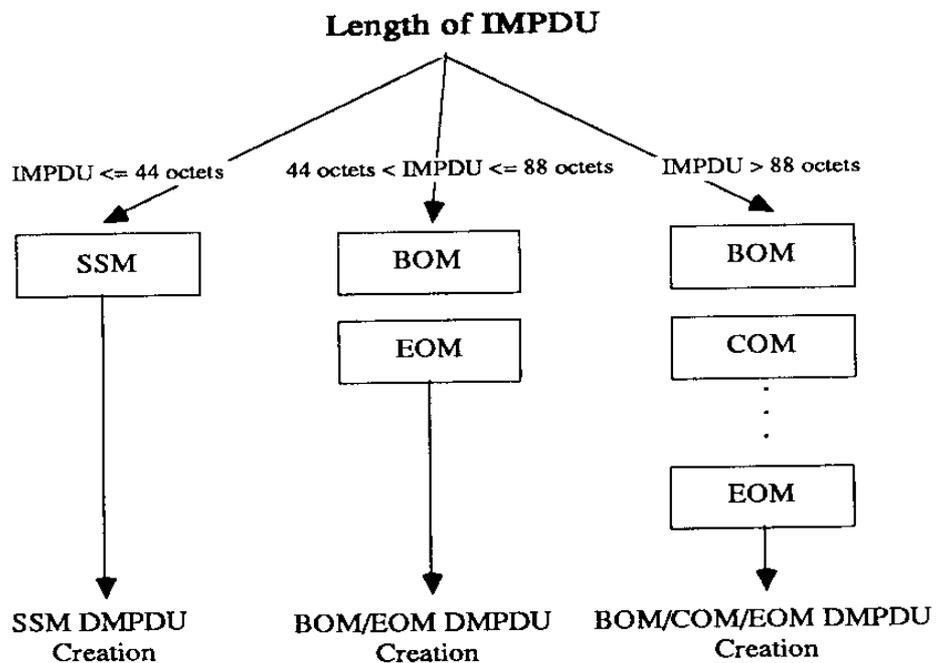


그림 3.5 IMPDU 분할 블럭 기능

3.1.1.3 DMPDU 생성 블록

하나의 IMPDU로부터 생성된 세그먼트 단위는 DMPDU 생성 블록을 통하여 DMPDU로 만들어지게 되는데, 앞에서 언급한 바와 같이 세그먼트 단위에 그 세그먼트가 어떤 것인지(SSM, BOM, COM, EOM 중 하나)를 DMPDU의 필드를 통하여 기록하게 된다. 이와 같이 세그먼트의 형태를 필드에 기록하는 것은 수신 노드에서 DMPDU를 조립하여 IMPDU를 생성할 수 있도록 하는 것이다. 그러나 DMPDU의 종류에 대한 정보만 가지고 수신노드에서 IMPDU를 생성할 수 없다. 즉 여러 노드에서 동시에 특정노드에 메시지를 전달할 때, 다른 종류의 IMPDU로부터 파생한 DMPDU들도 같이 수신하게 된다. 이러한 경우를 대비하여 각노드는 MID 페이지 이상을 갖고 있으며 하나의 IMPDU에서 파생한 DMPDU들은 같은 MID값을 갖도록 한다. MID의 할당은 망관리 기능에 의하여 이루어지며, LME(Layer Management Entity)를 통하여 DMPDU에 삽입하게 된다. 이에 대한 관계가 그림 3.2에 나타나 있으며 MID 할당기능은 Common 블록에서 자세히 다루어질 것이다. IMPDU로부터 분할과정을 거쳤을 때 하나의 세그먼트로 되는 SSM DMPDU에는 MID 값을 모두 0으로 코드한다. 이는 SSM DMPDU로부터 MID 정보가 필요없이 IMPDU를 생성할 수 있기 때문이다. DMPDU는 세그먼트 단위에 세그먼트 형태, MID, 정보필드길이, CRC 등을 붙이므로 해서 생성된다. 이는 MAC의 기능을 DMPDU 레벨에서 점검할 수 있도록 한 것이다. MAC의 주 기능은 QA 기능블록에서 수행하게 되는데, DMPDU 레벨에서 CRC로 세그먼트를 체크하는 의도는 IMPDU 레벨에서 체크하는 오버헤드를 줄이고, 실제로 MAC의 기능을 세그먼트 단위의 메시지를 전송하는 수단이라 할 수 있으므로 DMPDU 레벨에서 정보의 상태를 체크하는 것이 바람직하다. DMPDU를 생성하는 과정을 그림 3.6에 나타내었다.

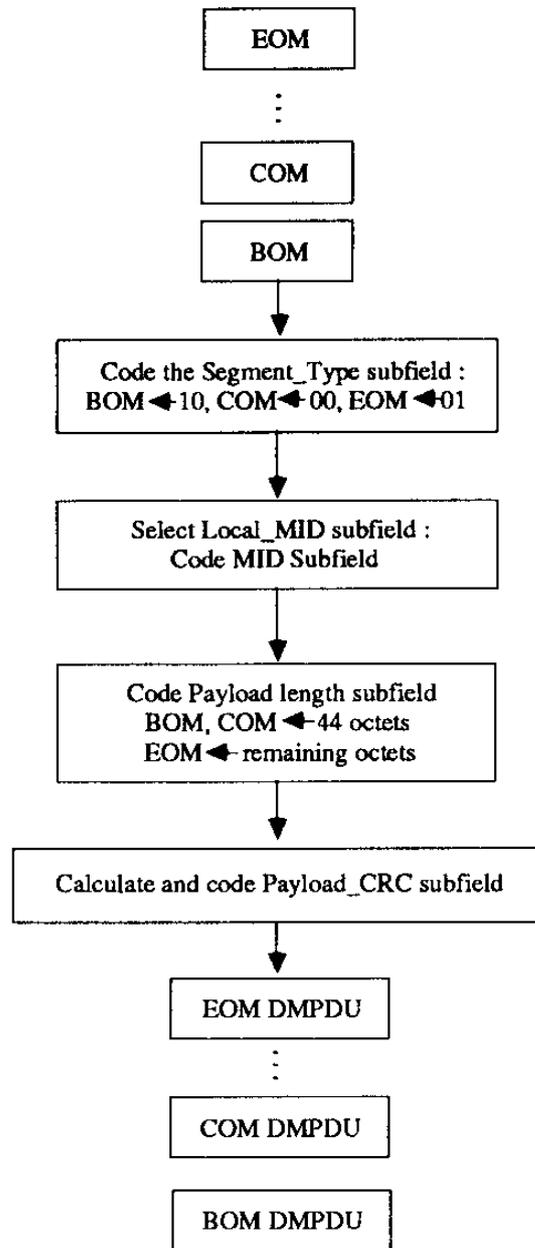


그림 3.6 DMPDU를 생성하기 위한 흐름도

3.1.1.4 MCF와 QA간의 상호 동작

MCF의 기능은 MAC의 기능을 보조하기 위한 준비 단계의 작업을 수행하는 블록으로 볼 수 있다. 즉 LLC로부터 받은 MSDU를 IMPDU로 생성한 다음 MAC의 알고리즘을 수행하는 QA 기능블럭에서 세그먼트의 레벨로 처리할 수 있도록 세그먼트로 분할하고

수신 노드에서 상태를 체크하거나 IMPDU를 재생할 수 있도록 돕기 위하여 DMPDU를 생성한다. MCF에서 생성된 DMPDU는 제어신호와 함께 QA 기능블럭으로 전달되며 QA 기능블럭에서는 일련의 과정을 거쳐 DMPDU를 전송한다. DMPDU를 전송하기 위한 MCF 블럭과 QA 기능블럭간의 신호를 그림 3.7에 나타내었다.

제어신호는 TX_BUS_SIGNAL과 ACCESS_QUEUE_SIGNAL이 사용되게 되는데, 전자는 MCF 블럭이 DMPDU를 어느 버스로 전달할 것(버스 A, 버스 B 혹은 양쪽)인가를 나타내고 후자는 어떤 우선순위레벨로 전달할 것인가를 나타낸다. TX_BUS_SIGNAL의 값은 IMPDU를 생성할 때 목적지 주소에 의하여 그 값을 정할 수 있으며, ACCESS_QUEUE_SIGNAL은 정보의 긴급성에 따라 우선순위를 정할 수 있다. LLC로부터 MA_UNITDATA request의 프리미티브를 통하여 priority 파라미터의 값이 있는 경우 그 값에 의하여 ACCESS_QUEUE_SIGNAL의 값을 정하게 된다. 우선순위값은 그림 3.2에서와 같이 IMPDU의 QOS_MAP_J의 값으로 되며 이 값에 의하여 ACCESS_QUEUE_SIGNAL 값이 정해진다. 일반적으로 LLC로부터 오는 비연결형 데이터는 가장 낮은 우선순위를 갖게 되며, 망관리에 필요한 정보는 가장 높은 우선순위를 가지고 전송하게 된다. 중간 레벨의 우선순위 사용에 대해서는 이렇다할 용도를 제시하고 있지는 않지만 운용중인 망의 상태에 따라서 특정 노드에 트래픽이 집중되는 경우 다른 노드보다 우선적으로 빈 슬롯을 많이 받을 수 있도록 높은 우선순위를 할당할 수 있을 것이다.

VCI_DATA는 IMPDU로부터 생성된 모든 DMPDU에 대해 동일한 값을 갖게 되며 DMPDU를 전송하기 위해 사용되는 QA 세그먼트의 헤더에 위치하게 된다. VCI_DATA는 현재 비연결형 데이터인 경우 20비트를 모두 1로 세트하는 것만 규정되어 있다. DQDB 망에 멀티포트 브리지에 연결될 경우 사용되도록 규정된 PT_DATA 및 SP_DATA 값도 현재는 디폴트로 00으로 세트하는 것만 규정되어 있다. PT_DATA와 SP_DATA는 하나의 IMPDU로부터 생성된 모든 DMPDU에 대해 동일한 값을 가지게 되며 이 값들은 DMPDU를 전송할 때 QA 세그먼트의 헤더에 위치하게 된다.

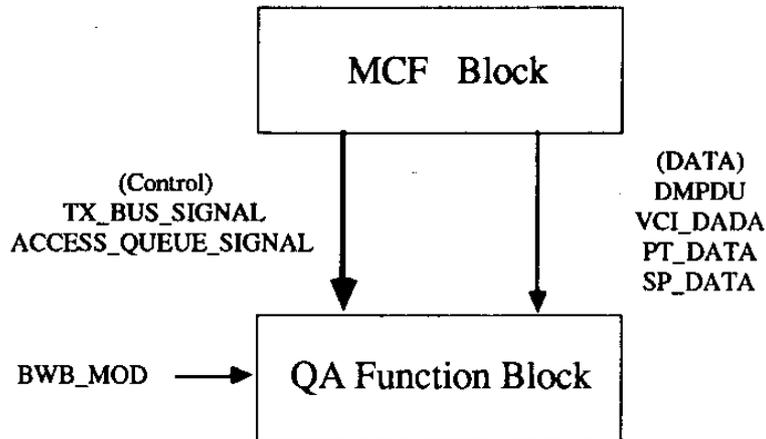


그림 3.7 MCF 블럭과 QA 기능블럭간 송신을 위한 상호동작

3.1.2 MCF 수신 기능

MCF 수신 기능은 QA 기능블럭으로부터 수신된 DMPDU를 재조립하여 IMPDU를 형성한 다음 IMPDU의 온전함을 확인한 다음 MSDU(MAC Service Data Unit)을 추출하는 일련의 과정이다. 여러개의 DMPDU를 조립하여 하나의 IMPDU를 만들 때에는 IMPDU 조립과정이 필요하게 되는데 이때 RSM(Reassembly State Mashine)이 동작하게 된다. 다른 IMPDU로부터 파생한 DMPDU들이 한 노드에 도달할 경우 MCF 블럭은 동시에 여러 RSM을 동작시킨다. RSM은 모든 VCI의 모든 MID에 대하여 하나의 RSM을 갖는다. 그러므로 각 RSM은 VCI/MID 쌍에 대해 유일하게 관련된다. 그러나 각 노드에 대해 동시에 지원할 수 있는 재조립과정의 수는 구현시 선택할 문제이다. SSM DMPDU를 수신하는 경우는 RSM이 필요없는데 이는 그 DMPDU내에 IMPDU가 내장되어 있기때문이다. 본 절에서는 MCF 블럭이 QA 기능블럭으로부터 DMPDU를 수신할 때 수행되는 동작에 대하여 기술한다. 그림 3.8은 MCF 블럭이 QA 기능블럭으로부터 DMPDU를 수신했을 때 그 DMPDU의 온전함을 확인하는 기능과 DMPDU 종류에 따른 조립과정을 나타내고 동작을 기능별로 구분한 것을 나타내고 있다.

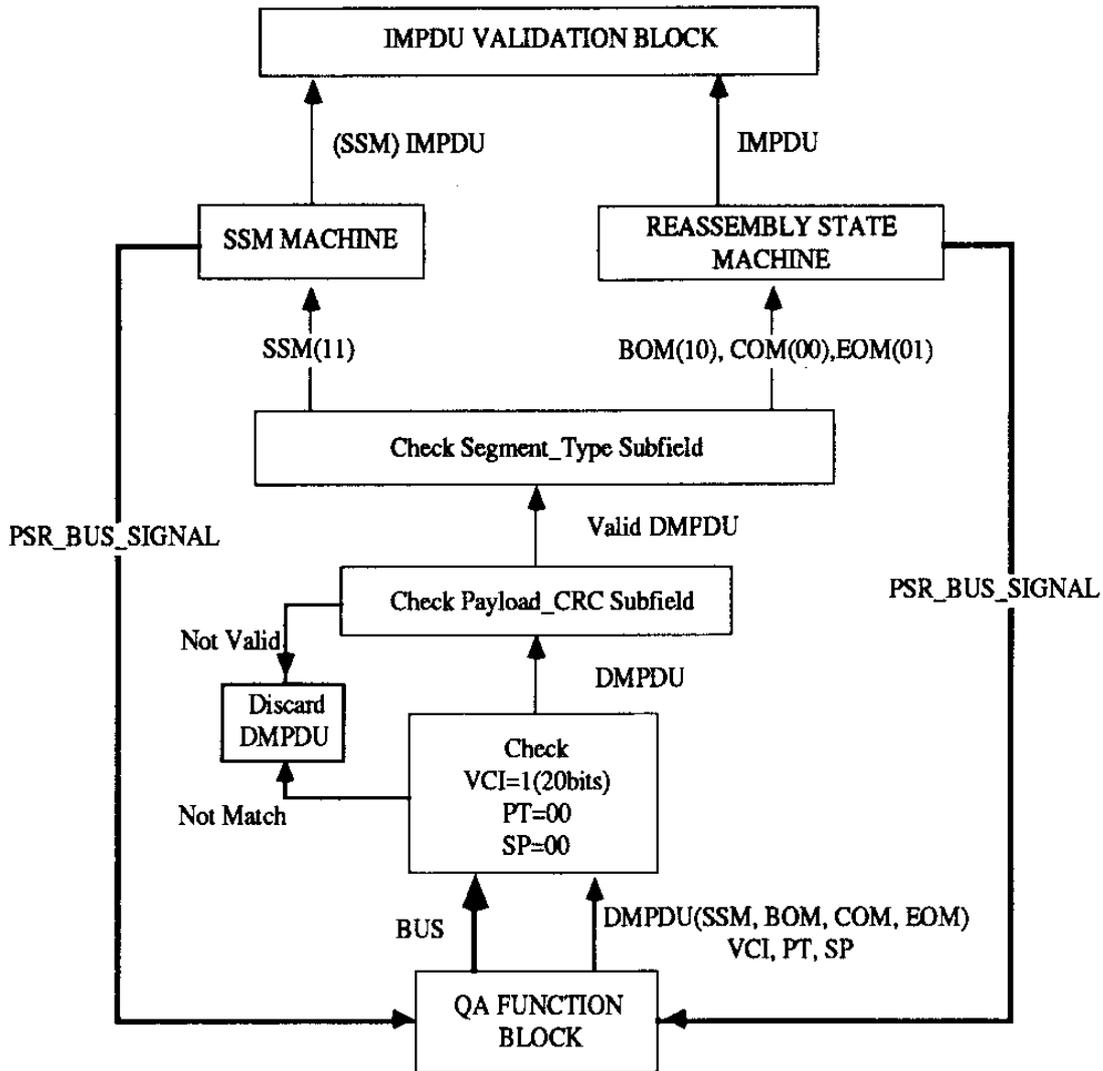


그림 3.8 MCF 수신기능

3.2.1.1 RSM 선택

MCF블럭이 수신하도록 프로그램된 VCI를 갖는 DMPDU가 QA기능블럭으로부터 수신되면 DMPDU 트레일러에 있는 Payload_CRC 필드를 사용하여 DMPDU의 정확성을 확인한 후 만일 DMPDU에서 에러가 검출되지 않는 경우에는 DMPDU의 헤더 필드의 Segment_Type의 값에 따라 다음 동작중 하나를 따른다.

(1) Segment_Type = BOM, COM, EOM

만일 DMPDU의 Segment_Type이 BOM,COM,EOM 중에 하나이면 MCF 블럭은 DMPDU의 헤더에 포함된 VCI_DATA 값과 MID에 따른 MID/VCI 쌍에 관련된 RSM(Reassembly State Machine)으로 DMPDU를 보낸다. RSM 기능에 대한 블럭도가 그림 3.9에 나타나 있다.

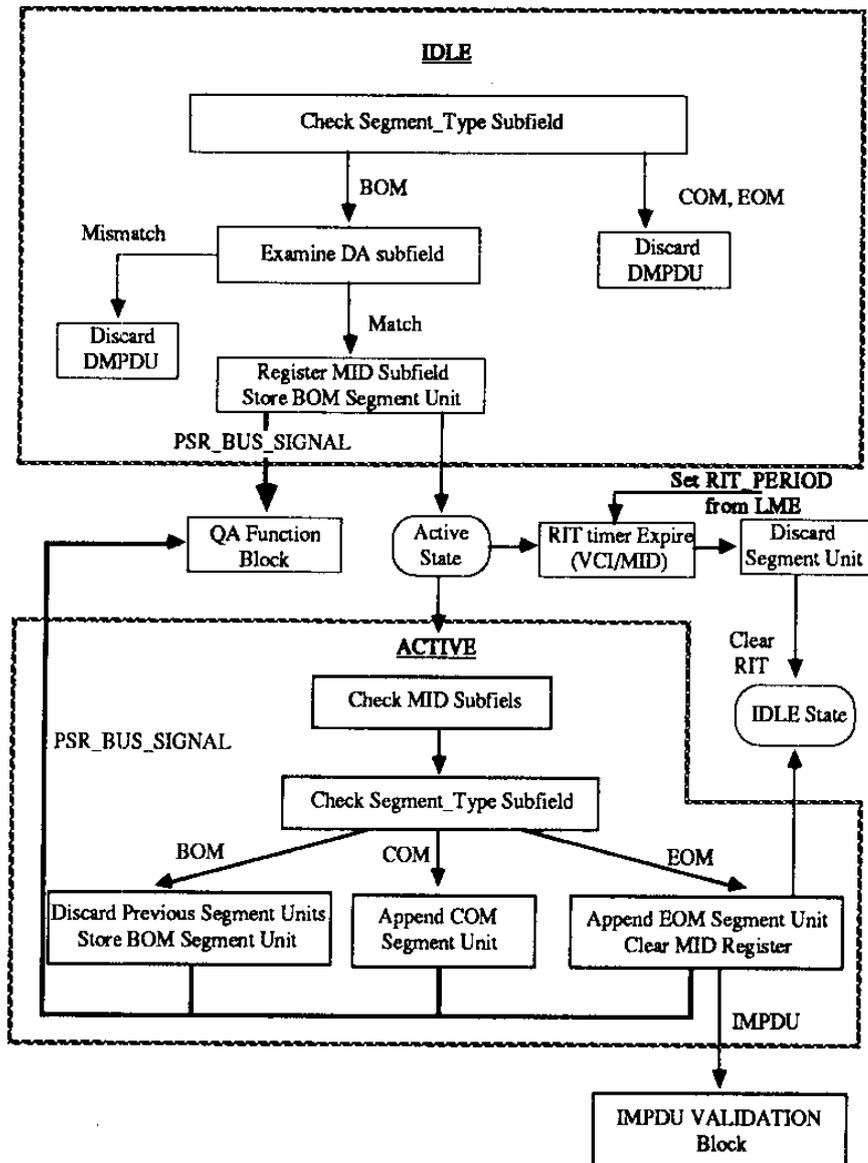


그림 3.9 RSM 기능

RSM 기능은 IDLE 상태와 ACTIVE 상태로 나뉘어진다. 여기서 IDLE 상태는 재조립을 위한 DMPDU를 수신하고 있지 않는 상태이고 ACTIVE 상태에서는 재조립과정을 수행하고 있는 상태를 말한다. IDLE 상태에서는 BOM만을 수신하며, 수신된 BOM의 DA 필드의 주소가 해당 노드와 일치하면 ACTIVE 상태로 전환한다. 이와 동시에 PSR_BUS_SIGNAL 제어신호를 QA 기능블럭으로 보낸다.

IMPDU가 2개의 세그먼트 이상으로 분할되는 경우 MCF 블럭은 BOM DMPDU에 있던 VCI_DATA, MID와 동일한 값을 갖는 일련의 COM DMPDU들을 수신해야 한다. 조립과정은 그러한 각 COM DMPDU로부터 세그먼트단위를 추출하여 이전에 수신된 BOM 세그먼트단위와 COM 세그먼트단위(들)에 첨부한다. 그리고 MCF 블럭은 위와 동일한 방법으로 EOM DMPDU를 수신해야 한다. 그리고 그것에서 추출한 EOM 세그먼트단위를 이전에 수신한 BOM 세그먼트단위와 COM 세그먼트단위에 덧붙인다. 이와 같이 하여 조립과정을 마친다. 조립된 IMPDU는 IMPDU 확인과정에 들어간다. 재조립은 RIT(Reassembly IMPDU Timer) 기간내에 완료되어야 하며 이 시간이 경과될 때까지 재조립과정이 완료되지 않았을 경우는 재조립과정을 중단하고 IDLE 상태로 되돌아 간다. 만일 그 DMPDU가 오직 이 노드에 전달되는 것이면 (즉, IG_Flag=INDIVIDUAL), MCF 블럭에 수신될 때 갖고 있던 RX_BUS_SIGNAL의 값과 동일한 값으로 PSR_x_SIGNAL의 값을 정한다. 아래와 같은 조건에 의하여 PSR_x_SIGNAL을 세트하여 QA 기능 블럭으로 보낸다.

- (i) 만일 DMPDU가 BOM이고 MCP 헤더의 DA(Destination Address)가 그 노드가 수신하도록 프로그램된 개개 MSAP 주소와 일치하면 PSR_x_Signal을 세트한다.
- (ii) 만일 DMPDU가 COM,EOM DMPDU이고 MID/VCI_DATA 쌍이 개개 MSAP 주소에 대해 동작중인 조립과정과 관련하여 MCP 헤더의 MID 값과 일치하면 PSR_x_SIGNAL을 세트한다.

조건 (i), (ii)가 성립되지 않으면 MCF 블럭은 PSR_x_SIGNAL을 세트하지 않는다.

(2) Segment_Type = SSM

만일 DMPDU의 Segment_Type이 SSM이면 그 DMPDU는 하나의 IMPDU를 포함하고 으므로 MCF 블록은 조립과정으로 보낼 필요가 없다. SSM IMPDU의 조립과정이 그림 3.10에 나타나 있다.

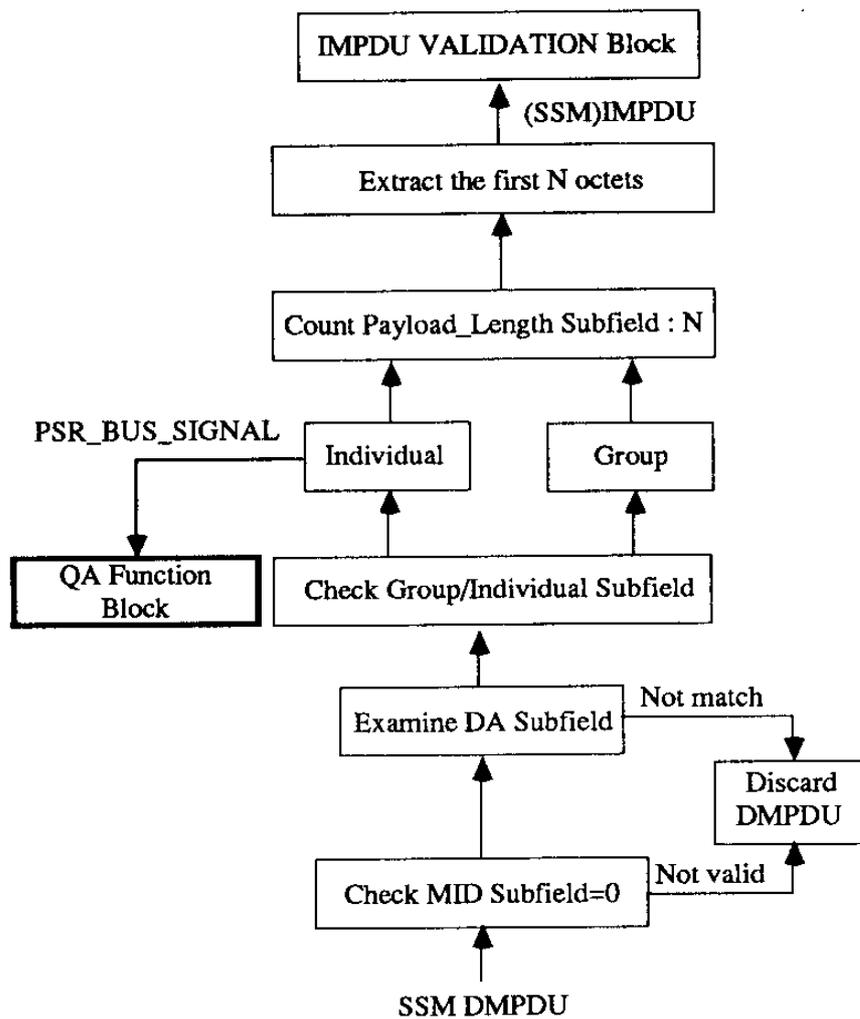


그림 3.10 SSM IMPDU 조립과정

MCF 블록은 DMPDU 헤더에 있는 MID 값을 조사한다. 만일 DA 필드가 개개 주소이면 MCF 블록은 DMPDU가 수신할 때 세트된 RX_BUS_SIGNAL의 값과 동일한 버스 x에 PSR_x_SIGNAL을 세트한다.

만일 수신한 DMPDU가 분명하지 않으면 그 DMPDU는 버린다. 아래 조건 A, B, C가 성립하면 버스 x에 대해 DMPDU가 수신될 때 RX_BUS_SIGNAL의 값과 동일한 값으로 PSR_x_SIGNAL을 정한다. 만일 아래 조건 A, B, C 중 어떤 조건도 성립하지 않는 경우 MCF 블록은 PSR_x_SIGANL을 띄우지 않는다.

조건 A

DMPDU의 Segment_Type이 BOM이나 SSM이고 MCP 헤더의 DA가 이 노드가 수신하도록 프로그램된 개개 MSAP 주소와 일치한다.

조건 B

DMPDU의 Segment_Type이 SSM이고 MID 값이 14비트가 모두 0으로 세트된 보유된 SSM과 일치하지 않는다.

조건 C

DMPDU의 Segment_Type이 COM이나 EOM이고 이 노드는 DMPDU 트레일러에 있는 Payload_CRC를 거쳐 DMPDU 헤더의 단일 비트에러수정을 지원하지 않는다. 그리고 그 MID 값이 MID/VCL_DATA 쌍으로 개개 MSAP에 대해 동작중인 조립과정과 관련된 MID와 일치한다.

- IMPDU 확인

SSM DMPDU로부터 추출한 IMPDU와 완전히 조립과정을 거쳐 만들어진 IMPDU는 아래와 같이 3가지 동작에 의해서 확인하게 된다.

1. Common PDU 트레일러의 Length 필드에 있는 값은 그 IMPDU의 옥텟 수와 비교한다. 수신된 IMPDU의 수에서 8을 뺀 값은 Length 필드의 값에서 Length 필드에 3을 더한 수를 포함한 범위에 있어야 한다. 만일 일치하지 않는 경우 MCF 블록은 그 IMPDU를 버린다.
2. Common PDU 헤더의 Bntag의 값을 Common PDU 트레일러에 있는 Bntag 값과

비교한다. 만일 일치하지 않으면 그 IMPDU는 버린다.

3. MCP 헤더에 있는 Header Extension Length 필드에 있는 값을 검사한다. 지정된 범위인 0에서 5의 값을 벗어날 때 MCF 블록은 그 IMPDU를 버린다.

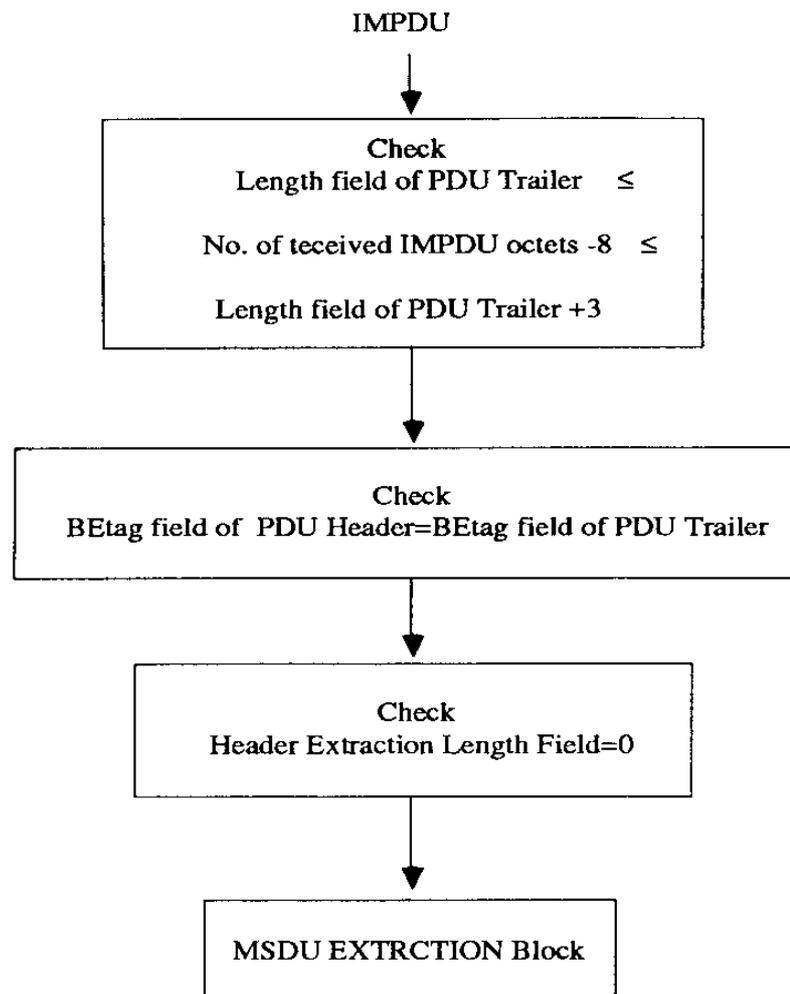


그림 3.11 IMPDU 확인

- MSDU 추출

MSDU 추출기능은 애러없는 IMPDU만을 취하는 것이다. 그리고 MA_UNITDATA indication내에 전달될 파라미터를 생성하기 위하여 다음과 같은 기능을 수행한다.

- 1) MCP 헤더의 DA 필드에 있는 값을 destination_address 파라미터를 생성하기 위해 사용한다.
- 2) MCP 헤더의 SA 필드에 있는 값을 source_address 파라미터를 생성하기 위해 사용한다.
- 3) MCP 헤더의 QOS_DELAY 필드에 있는 값을 priority 파라미터를 생성하기 위해 사용한다.
- 4) INFO 필드에 포함된 MAC Service Data Unit를 추출하여 data 파라미터를 생성한다.

MSDU 추출기능은 MA_UNITDATA indication을 발생시킨다. MSDU 추출 기능은 IMPDU 확인기능으로부터 IMPDU를 수신한 순서대로 MA_UNITDATA indication 프리미티브를 발생시킨다.

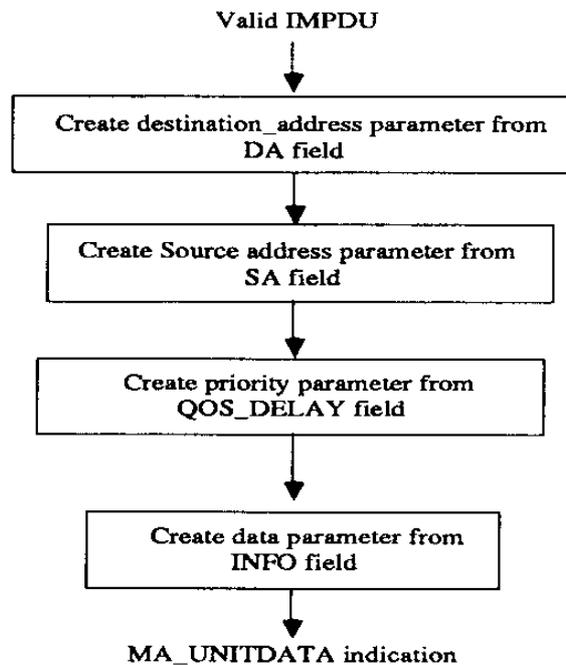


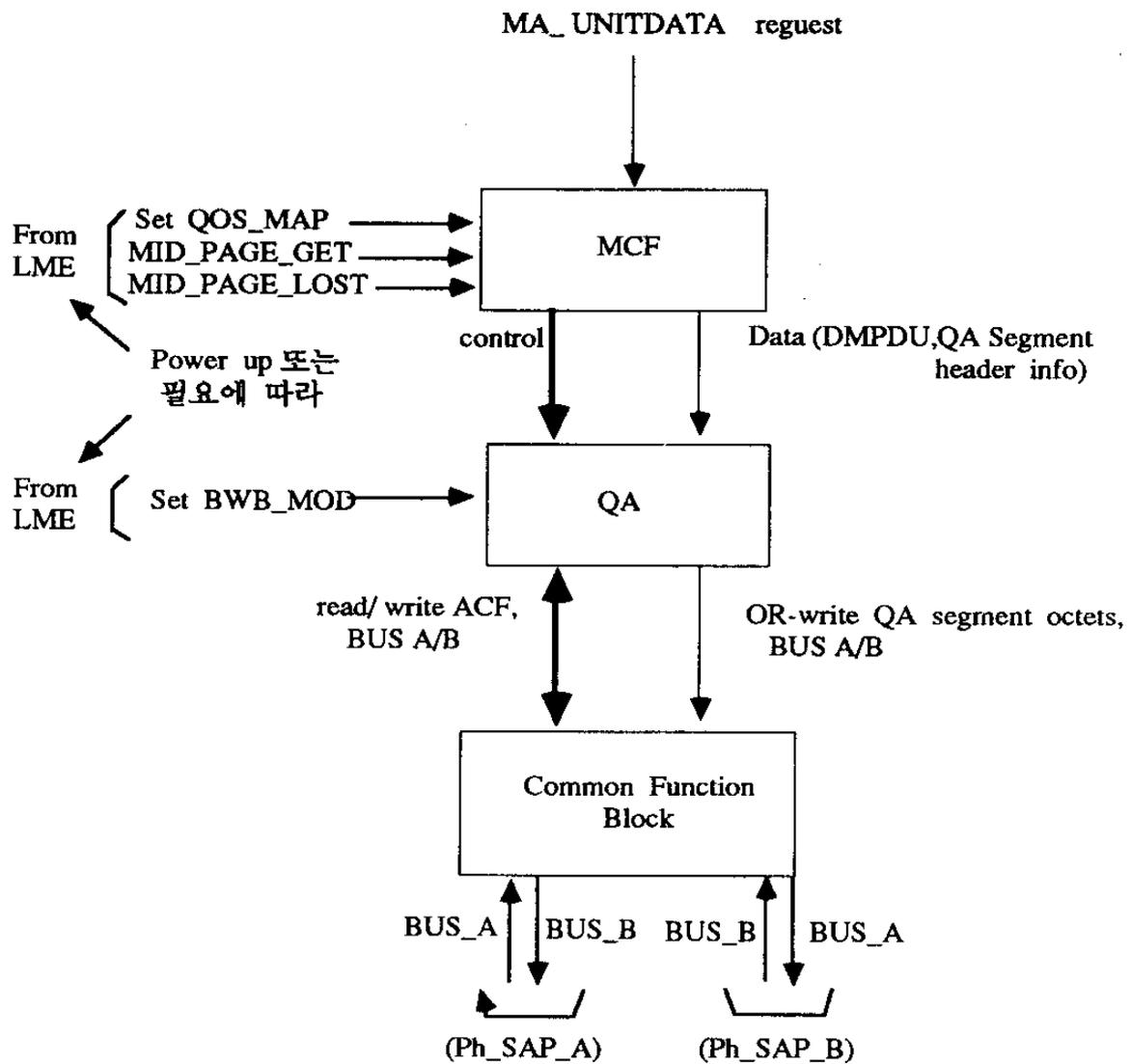
그림 3.12 MSDU 추출기능

3.2 QA 기능 블록

QA 기능 블록은 MCF에 의하여 생성된 DMPDU를 QA 세그먼트로 만든 다음 QA 세그먼트 단위로 전송하도록 제어한다. 생성된 QA 세그먼트는 분산 큐 알고리즘에 의하여 전달된다. 본 절에서는 QA 세그먼트의 생성과정, 전송과정, 수신과정, QA 세그먼트 추출과정 등의 각 블록의 기능에 대하여 기술한다.

3.2.1 QA 세그먼트 송신기능

QA 송신기능은 MCF로부터 QA 세그먼트 정보(DMPDU)를 받을 때 시작된다. QA 세그먼트 정보는 QA 세그먼트 헤더를 생성하는데 필요한 데이터와 QA 세그먼트 정보를 전송하는데 필요한 제어 정보와 함께 QA 블록에 전달된다. 그림 3.13에 MSDU를 송신하기 위한 기능블록을 나타내었다. 또한 그림 3.14에는 QA 송신 블록의 내부구조를 보였다. QA 송신 기능 블록은 (1) QA 세그먼트를 생성하기 위한 QA 세그먼트생성기능(QA Segment Creation Block), (2) QA 세그먼트를 전송하기 위한 대기장소로서 FIFO 큐(FIFO Queue Block), (3) 분산 큐 알고리즘에 의하여 QA 세그먼트를 하나씩 전송하기 위한 장소인 분산큐블록(Distributed Queue Block) 등 세가지 기능을 수행한다. 이러한 기능들은 비연결형 데이터를 전송하는 DQDB MAC의 기능 중 핵심적인 기능이라 할 수 있다. 다른 LAN 프로토콜에서 메시지 단위로 전송을 위한 MAC 기능과 달리 DQDB MAN에서는 세그먼트 단위로 스케줄링 알고리즘을 수행한다. 즉 LLC로부터 받은 MSDU는 QA 기능블록에 도달하기 전에 분할과정을 거친후 QA 기능블록에 전달되며 거기서 세그먼트 단위로 전송을 위해 스케줄링하게 된다.



* LME : Layer Management Entity
 Ph_Sap_A/B : Physical Layer Service Access Point for Bus A/B

그림 3.13 MCF 및 Common Fuction 블럭과의 상호동작

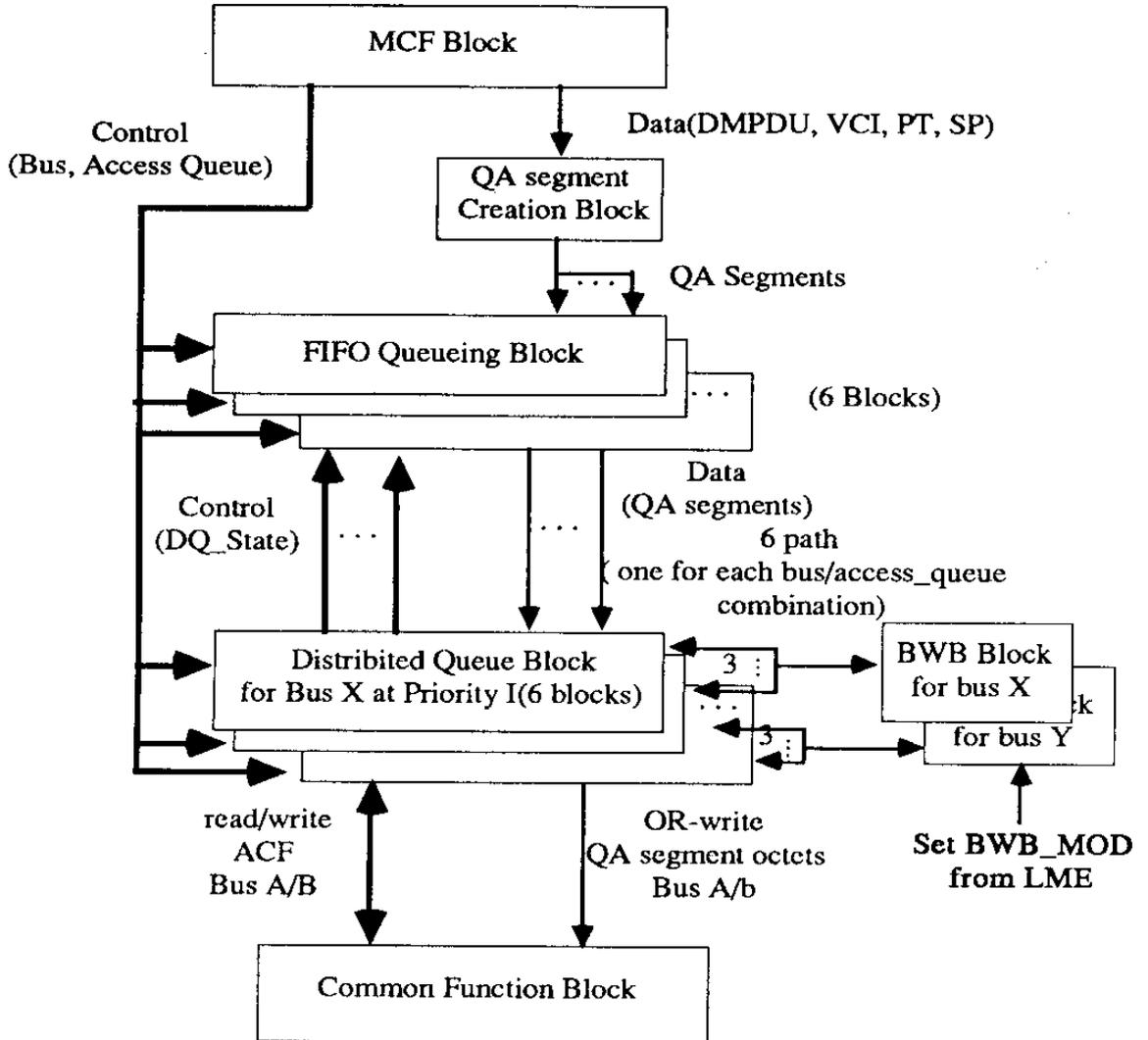


그림 3.14 QA 기능블럭의 송신부 내부구조

3.2.1.1 QA 세그먼트 생성 블럭

QA 세그먼트는 MCF 기능블럭으로부터 QA 세그먼트 정보와 QA 세그먼트를 생성하는데 필요한 정보를 받음으로써 생성된다. QA 세그먼트를 생성하는 과정은 그림 3.15에 나타내었다. QA 세그먼트는 하나의 QA 슬롯에 위치하는 PDU이다. 각 QA 세그먼트는 4옥텟의 헤더와 48옥텟의 정보를 갖고있다. QA 세그먼트헤더는 VCI, PT(Payload_Type), SP(Segment Priority), HCS 등 4개의 필드로 구성되어 있다. 20비트의 VCI는 세그먼트가 전달될 가상채널을 구분하기 위한 수단으로서 사용된다. 본 과제에서는 20비트의 VCI 값을 모두 비연결형 데이터의 디폴트 값인 1로 세트하도록 구현한다. 디폴트 비연결형 VCI를 사용하는 MCF에 대해 QA 세그먼트 정보를 보내기 위한 모든 QA 세그먼트내에 PT 값은 00으로 세트하여야 한다. 다른 VCI를 갖는 QA 세그먼트내에 PT의 값은 연구중이다. SP의 2비트의 필드는 추후 멀티포트 브리지에 사용하도록 유보되어 있고 현재는 00으로 세트하도록 한다. 8비트의 HCS는 QA 세그먼트 헤더에 있는 에러를 검출 및 수정하는 기능 제공한다. HCS(Header Check Sequence) 필드는 QA 세그먼트 헤더필드에 의해 계산된 8비트 CRC를 포함하고 있다.

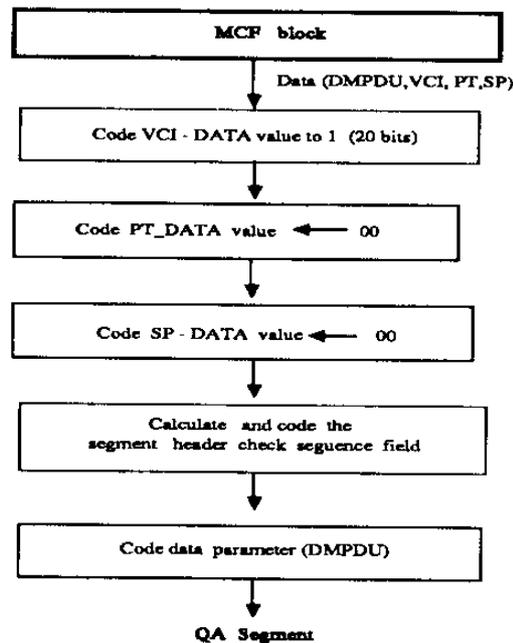


그림 3.15 QA 세그먼트 생성과정

3.2.1.2 QA 세그먼트의 FIFO 큐에 전달

각 노드당 분산 큐에는 최대 6개의 QA 세그먼트가 대기할 수 있는데 그림 3.14에서 보는 바와 같이 각 QA 세그먼트는 QA 기능 블록에 전달되는 TX_BUS_SIGNAL과 ACCESS_QUEUE_SIGNAL의 제어신호에 의하여 전달되게 된다. QA 기능블럭은 어떤 priority 레벨에 어떤 버스로도 전송할 수 있도록 여러개의 QA 세그먼트의 정보를 받을 수 있다. 이러한 동작은 버스 및 priority 레벨에 대하여 QA 세그먼트를 대기시킬 수 있는 국부 FIFO 큐를 제어함으로써 이루어 진다. QA 세그먼트 정보가 QA 기능블럭에 전달될 때 TX_BUS_SIGNAL과 ACCESS_QUEUE_SIGANL에 의해서 해당 FIFO 큐에 위치하게 되는데 이때 TX_BUS_SIGNAL이 BOTH로 되면 양쪽 버스 큐 모두에 전달된다. FIFO 큐의 맨 앞에 위치하는 QA 세그먼트는 DQSM(Distributed Queue State Machine)의 DQ state 제어신호에 따라 DQSM이 idle 상태에 있을 때 분산 큐로 전달된다.

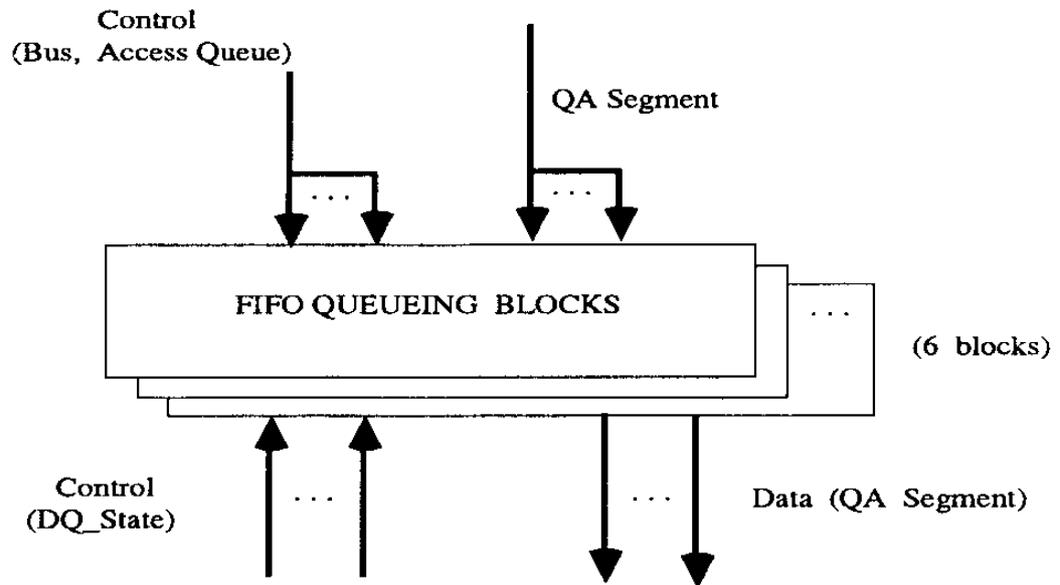


그림 3.16 FIFO 대기블럭

3.2.1.3 분산 큐

분산 큐의 기능은 QA 세그먼트를 빈 슬롯에 write함으로써 MAC 기능을 수행한다. 분산 큐는 RQ 계수기, CD 계수기, REQ 계수기를 동작시킨다. 또한 각 버스에 대하여 BWB 계수기를 동작시킨다. 버스 X(X=A or B)에 대한 RQ 계수기와 CD 계수기는 버스 X상의 모든 ACF내 모든 BUSY 비트와 SL_TYPE 비트를 읽고 반대 방향의 버스인 버스 Y(Y=B or A) 상의 모든 슬롯의 ACF내의 REQUEST 필드를 읽는 동작을 한다. 버스 x와 접근 큐 우선순위 레벨 I(I=0,1,2)의 조합에 대하여 REQ_I_CNTR_X와 CD_I_CNTR_X로 표시되는 계수기를 제어한다. 우선순위 I 레벨로 버스 X에 접근하기 위한 요청은 반대 방향의 버스인 버스 Y로 지나가는 각 슬롯의 ACF 내에 0의 값을 갖는 REQ_I 비트를 1로 세트함으로써 다른 노드에게 자신이 요청했다는 사실을 알린다. 이러한 동작이 DQSM_X_I 머신에서 수행된다. 각 DQSM에 대하여 반대방향의 버스 Y상에 우선순위 레벨 I로 요청을 보내는 것을 제어하기 위해 대응되는 REQ_X_I 머신이 하나 존재한다. 또한 각 버스 X에 대하여 대역폭 균형 계수기를 제어하기 위한 하나의 BWBM(Bandwidth Balancing Machine)이 있다. 그림 3.17에 DQSM을 구성하는 분산 큐의 기능을 나타내었다.

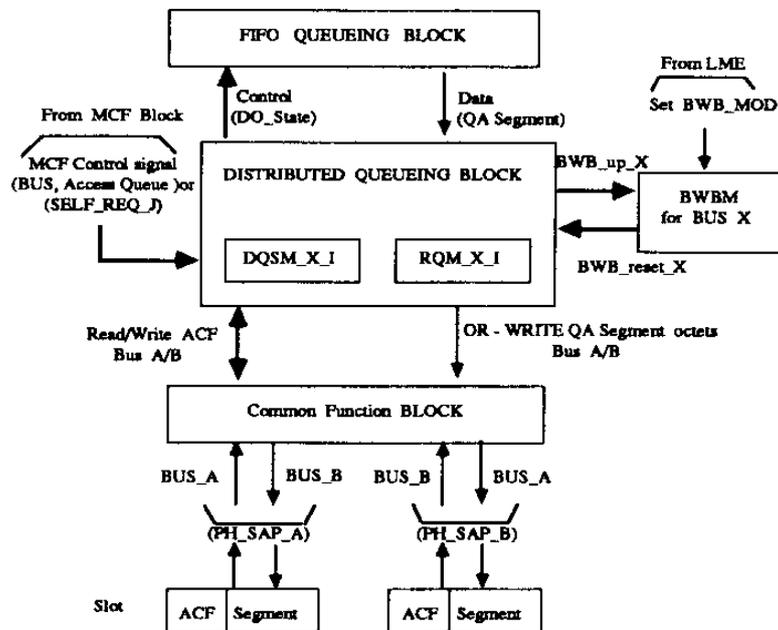


그림 3.17 분산 큐의 구조

그림 3.18에서는 버스 X에서 우선순위 I의 DQSM(DQSM_X_I)의 동작을 나타내고 있다. DQSM 동작은 IDLE QA 상태(세그먼트가 대기하고 있지 않은 상태)와 COUNTDOWN 상태(세그먼트가 대기하고 있는 상태)로 나누어지게 된다. IDLE 상태에서는 (1) MCF 블럭으로부터 SELF_REQ_J 신호가 도착하고 우선순위레벨 J가 해당 DQSM의 우선순위 I보다 크거나 같을 때 (2) 버스 Y를 통과하는 슬롯의 REQ_J가 I보다 크거나 같을 때 (3) BWBM(Bandwidth Balancing Machine)으로부터 BWB_reset_X 시그널이 도착할 때마다 RQ_I 계수기의 값을 하나씩 증가시킨다. RQ_I 계수기의 값은 2^6-1 까지 증가될 수 있다. 또한 RQ_I의 계수기 값은 버스 X를 통하여 빈 QA 슬롯이 통과할 때마다 하나씩 감소한다.

FIFO 큐로부터 QA 세그먼트가 도착하면 DQSM을 IDLE 상태에서 COUNTDOWN상태로 전환하며, 이때 RQ_I 계수기 값을 CD_I 계수기 값으로 전환하고 RQ_I의 값을 0으로 세트한 다음 DQ 신호를 이용하여 FIFO 큐에 COUNTDOWN 상태로 되었음을 알린다. 동시에 RQM(Request Queue Machine)에 SELF_REQ_I 신호를 보내어 RQM_X_I 머신이 동작하도록 한다.

DQSM이 COUNTDOWN상태에 있을 경우는 (1) MCF 블럭으로부터 SELF_REQ_J 시그널이 도착하고 우선순위 레벨 J가 해당 DQSM의 우선순위 I보다 크거나 같을 때 (2) 버스 Y를 통과하는 슬롯의 REQ_J가 1로 세트되어 있고 J가 I보다 큰 경우에 (3) BWBM으로부터 BWB_reset_X 시그널이 도착할 때마다 CD_I 계수기 값을 하나씩 증가시킨다. 그러나 버스 Y를 통과하는 슬롯의 REQ_J가 1로 세트되어 있고 J가 I와 같은 경우에는 RQ_I 계수기 값을 하나 증가시킨다. CD_I 계수기 값은 버스 X를 통하여 빈 QA 슬롯이 통과할 때마다 하나씩 감소한다. 이때 CD_I 계수기 값이 0으로 되면 빈 슬롯을 BUSY 슬롯으로 전환한 후 대기하고 있던 QA 세그먼트를 전송하고 동시에 BWBM에 BWB_up_X 신호를 보내 BWB 계수기가 증가되도록 하며 DQSM은 COUNTDOWN 상태에서 IDLE 상태로 전환된다. 이때 DQSM은 DQ 상태 신호를 이용하여 FIFO 큐에 IDLE 상태임을 알린다.

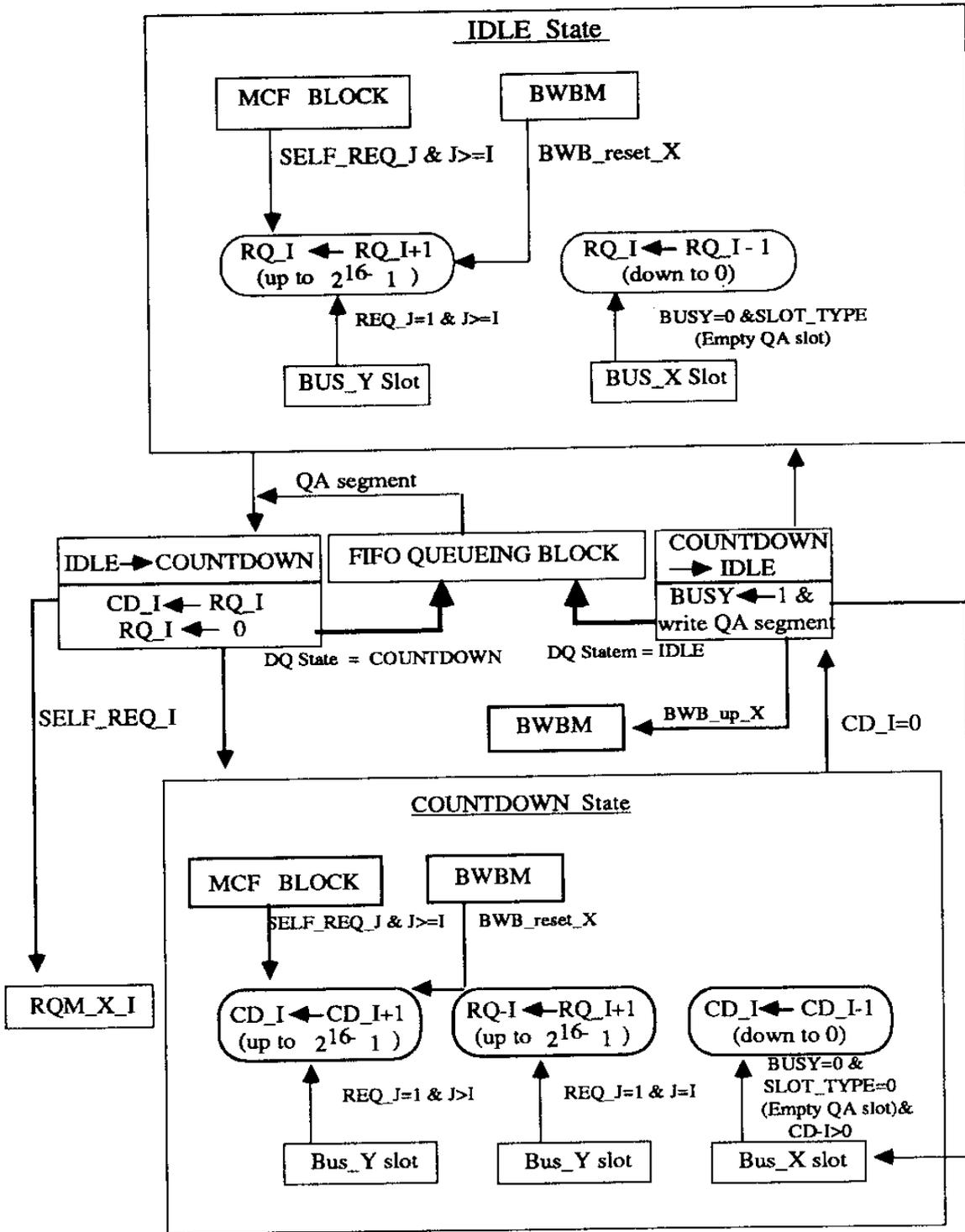


그림 3.18 분산 큐의 동작

우선순위 레벨이 I인 DQSM은 IDLE 상태에서 COUNTDOWN상태로 전환하면서 반대방향 버스를 통과하는 슬롯의 ACF에 REQ_I를 세트하도록 한다. REQ_I 비트를 세트하는 동작은 실제 RQM(REQ Queue Machine)에 의해 제어된다. RQM은 REQ_I_Q_Y 계수기를 동작시킨다. REQ_I_Q_Y 계수기는 버스 Y에 보내야 할 우선순위 I 레벨의 REQ_I의 수를 유지한다. 그 계수기는 DQSM으로부터 SELF_REQ_I 신호가 도착할 때마다 증가 하며 노드가 버스 Y에 지나가는 슬롯의 ACF 필드에 REQ_I 비트가 0인 값을 읽고 그 비트를 1로 세트함으로써 계수기의 수를 감소시킨다. 그림3.18과 3.19에서 보는 바와 같이 RQM 비트를 세트하는 동작과 QA 세그먼트를 보내는 동작은 서로 독립적으로 수행된다.

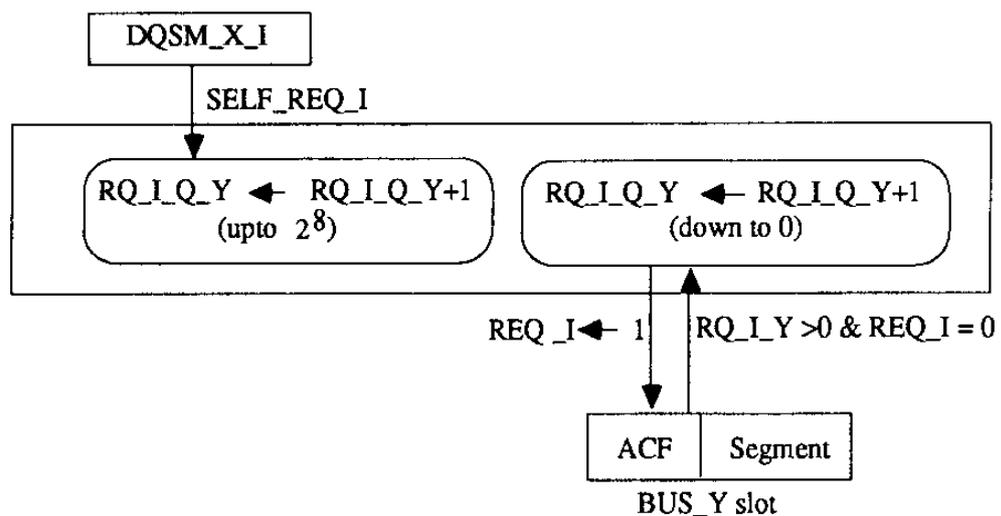


그림 3.19 REQ 큐 머신의 동작

BWBM은 버스 X에 대한 대역폭 균형 계수기인 BWB_CNTR_X를 동작시킨다. BWBM의 동작이 그림 3.20에 나타나 있다. 이러한 동작에는 대역폭 균형 계수인 BWB_MOD가 관계된다. 만일 BWB_MOD가 0인 값을 가지게 되면 대역폭 균형

메카니즘은 동작하지 않는다. 버스 X에 대한 3개의 DQSM중에 어떤 것도 버스 X에 대한 BWBM에 BWB_up_X 신호를 보낼 수 있는데 이때 BWB_CNTR_X의 값이 BWB_MOD-1인 값이 되지 않을 경우 그 카운터는 1 만큼 증가하게 된다. 버스 X에 대한 BWBM이 BWB_up_X 신호를 받을 때 BWB_CNTR_X가 BWB_MOD-1이 되는 경우는 그 BWBM은 BWB_CNTR_X를 0으로 리셋하고 BWB_reset_X 신호를 버스 X위에 있는 세개의 DQSM에게 각각 보내게 된다. 이 신호는 DQSM에게 버스 X위에 지나가는 하나의 빈 슬롯을 사용하지 말고 포기하도록 알리는 것이다.

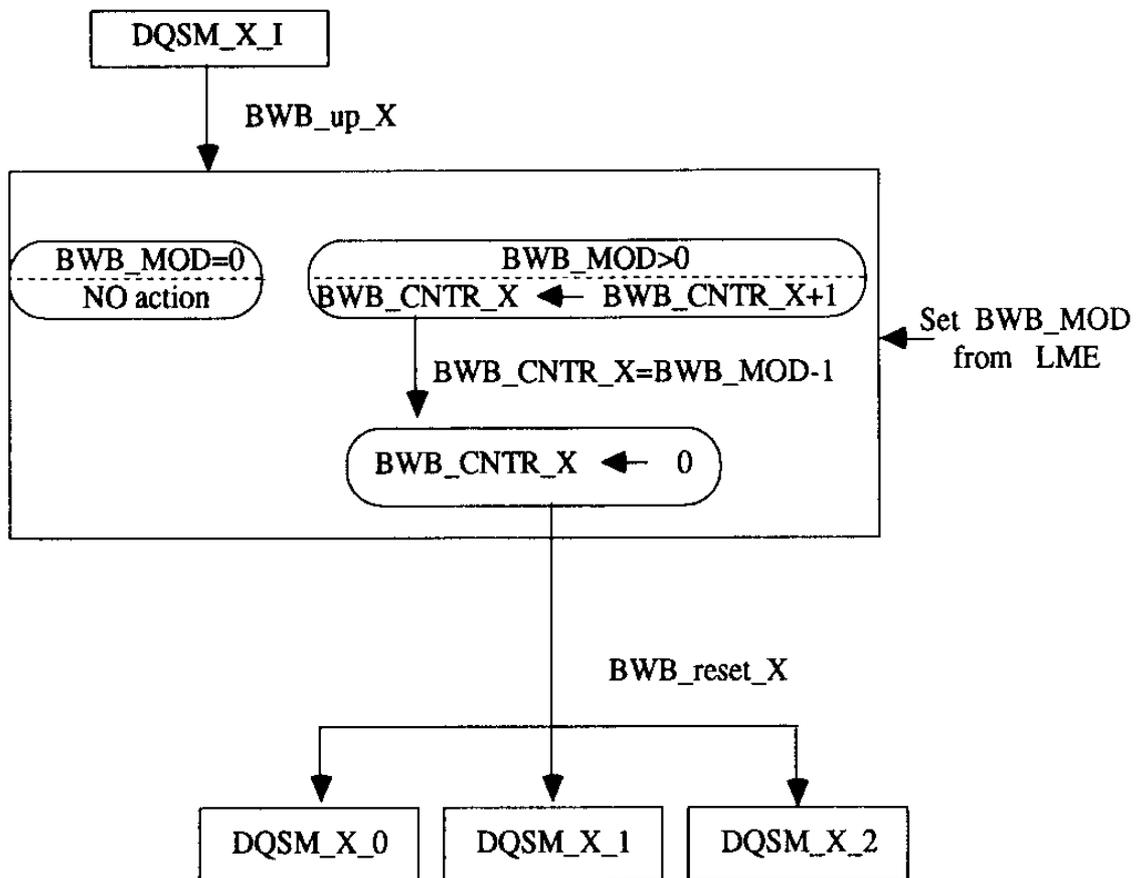


그림 3.20 BWB 머신의 동작

3.2.2 QA 수신 기능

그림 3.21에 QA 수신기능과 MCF 및 Common 블록간의 상호관계가 나타나 있다. QA 수신 기능은 BUSY 비트가 1로 세트되고 SL_TYPE 비트가 1로 세트된 ACF를 가진 슬롯(즉 busy인 QA 슬롯)을 수신했을 때 수신 기능을 수행한다. 수신 기능은 busy인 QA 슬롯이 Common Function에 의해 전달되면 QA 헤더를 검사한 후 QA 세그먼트 정보필드 추출한다. 어떤 버스에 대해서 수신된 QA 슬롯에 대한 MCF나 QA 기능의 작동은 다음 슬롯이 그 버스에 이르기 전에 완료되어야 하며, 그 이유는 MCF 블록이 다음 슬롯에 PSR 비트를 세트하여야 할지를 결정해야 하기 때문이다(MCF 블록 참조).

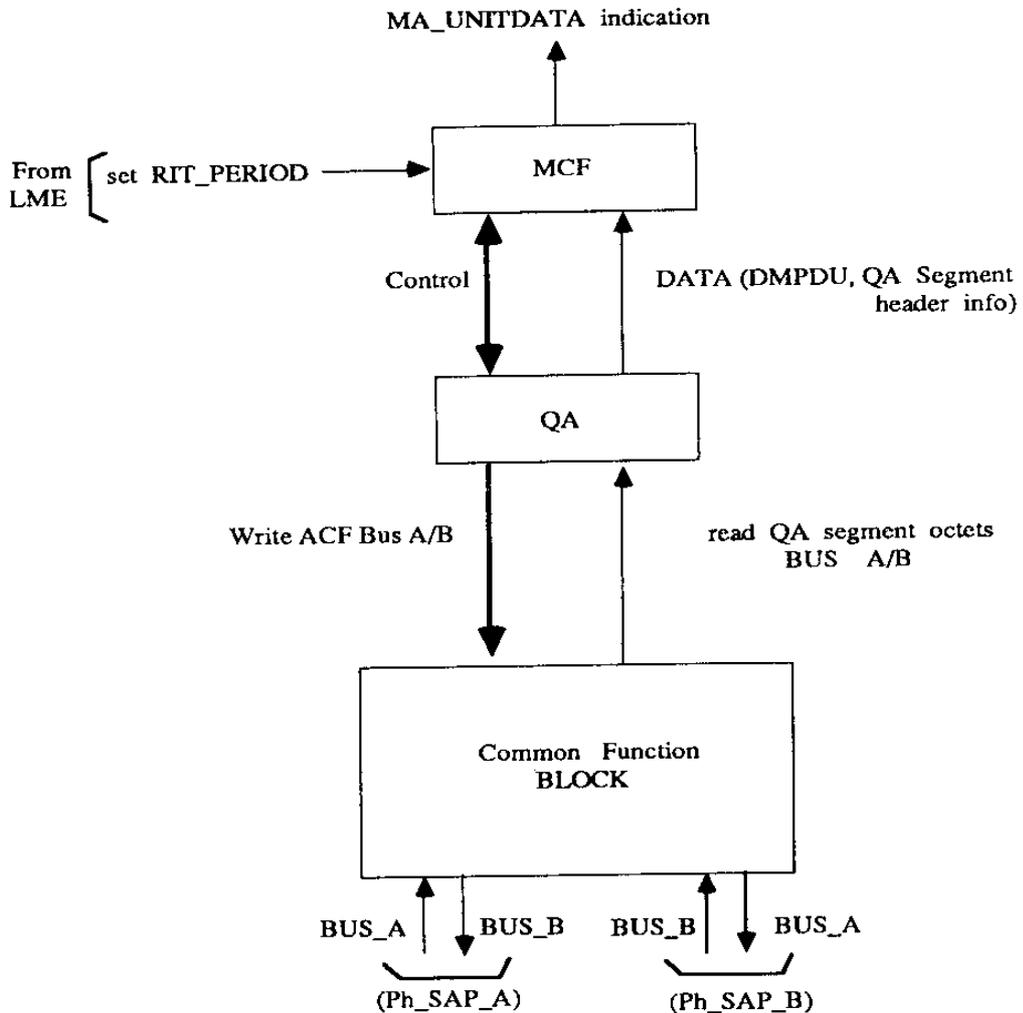


그림 3.21 QA 수신시 MCF 및 Comon Function 블록과의 상호동작

QA 수신 기능은 (1) BUSY, QA슬롯의 처리블럭(Busy, QA Slot Processing Block), (2) QA 세그먼트 검사블럭(QA Segment Header Validation Block), (3) 수렴기능 선택블럭(Convergence Function Selection Block) 등으로 나뉘어지며, 아래 그림 3.22 각 블럭간의 제어신호와 데이터 신호를 보였다.

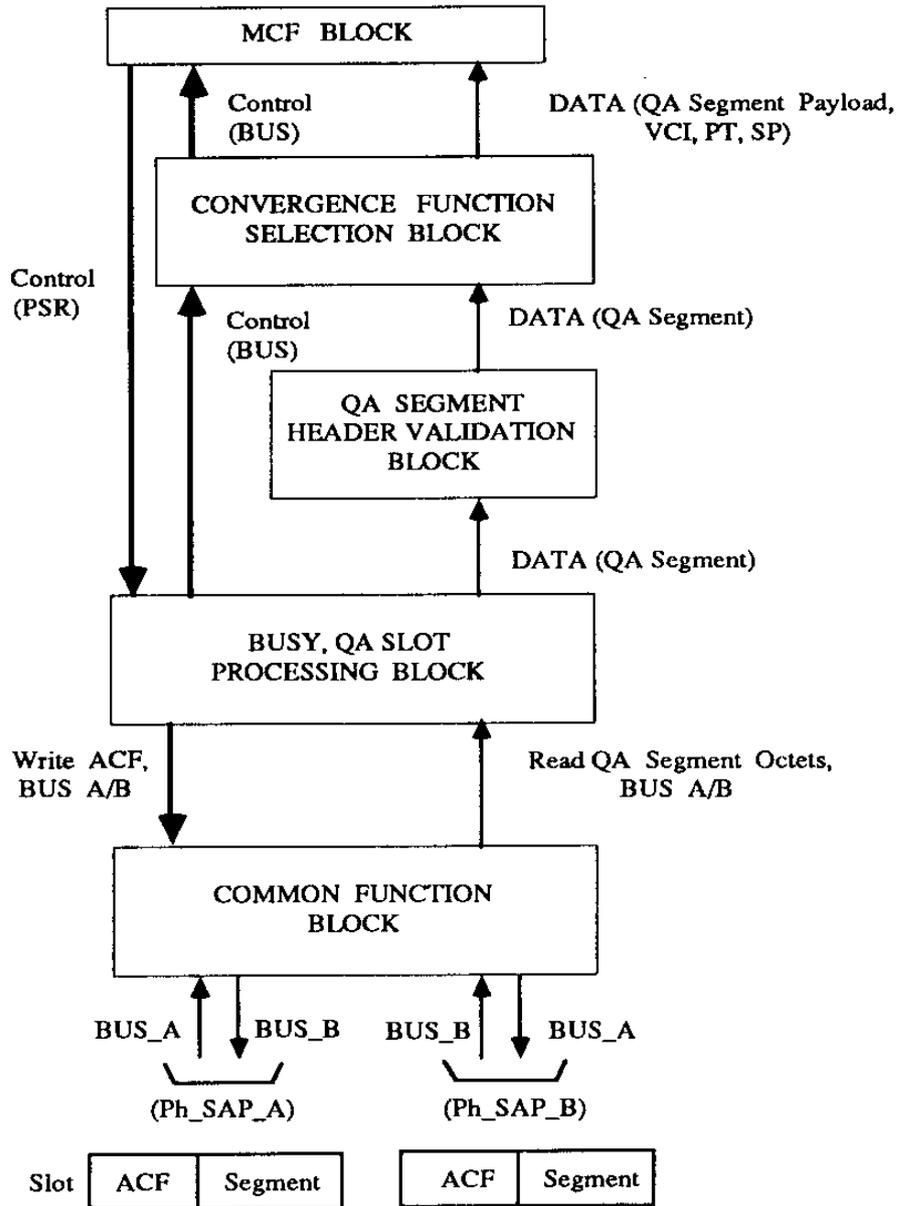


그림 3.22 QA 블럭의 수신기능

그림 3.23에 BUSY, QA 슬롯처리 블록이 나타나 있다. Common Function 블록은 BUSY 비트가 1로 세트되고 SL_TYPE비트가 0으로 된 ACF를 수신하면 그 슬롯의 QA 세그먼트의 옥텟을 복사하여 QA 기능 블록에 전달한다. 또한 MCF 블록에서 PSR 비트를 세트하라는 제어신호(PSR_x_SIGNAL)가 오면 다음에 도착하는 QA 슬롯내의 PSR비트를 1로 세트한다.

ACF of a slot

BUSY (1)	SL_TYPE (1)	RSVD (1)	PSR (1)	REQUEST (4)	Segment (52 Octets)
-------------	----------------	-------------	------------	----------------	------------------------

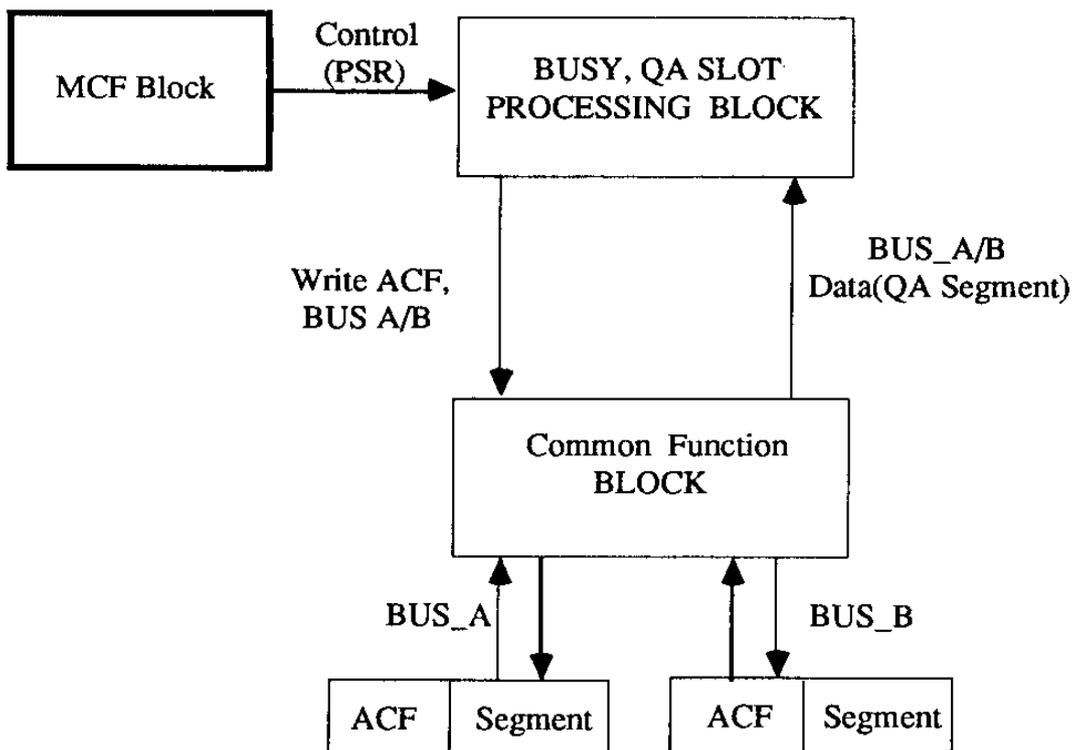


그림 3.23 Busy, QA 슬롯처리블럭

QA 세그먼트 검사블럭에서는 QA 세그먼트를 받으면 HCS를 사용하여 QA 세그먼트의 헤더의 정확성을 검사한다. HCS에서 에러가 검출되면 QA 기능 블럭은 그 세그먼트는 버리고, HCS에서 에러가 검출되면 그 세그먼트는 수렴기능선택으로 전달된다. 이 과정이 그림 3.24에 나타나 있다.

- QA Segment Header Fields

VCI (20)	PT (2)	SP (2)	HCS (8)	QA segment Payload (48 octets)
-------------	-----------	-----------	------------	-----------------------------------

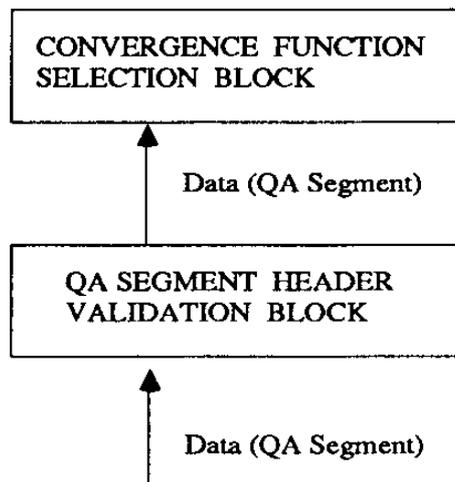
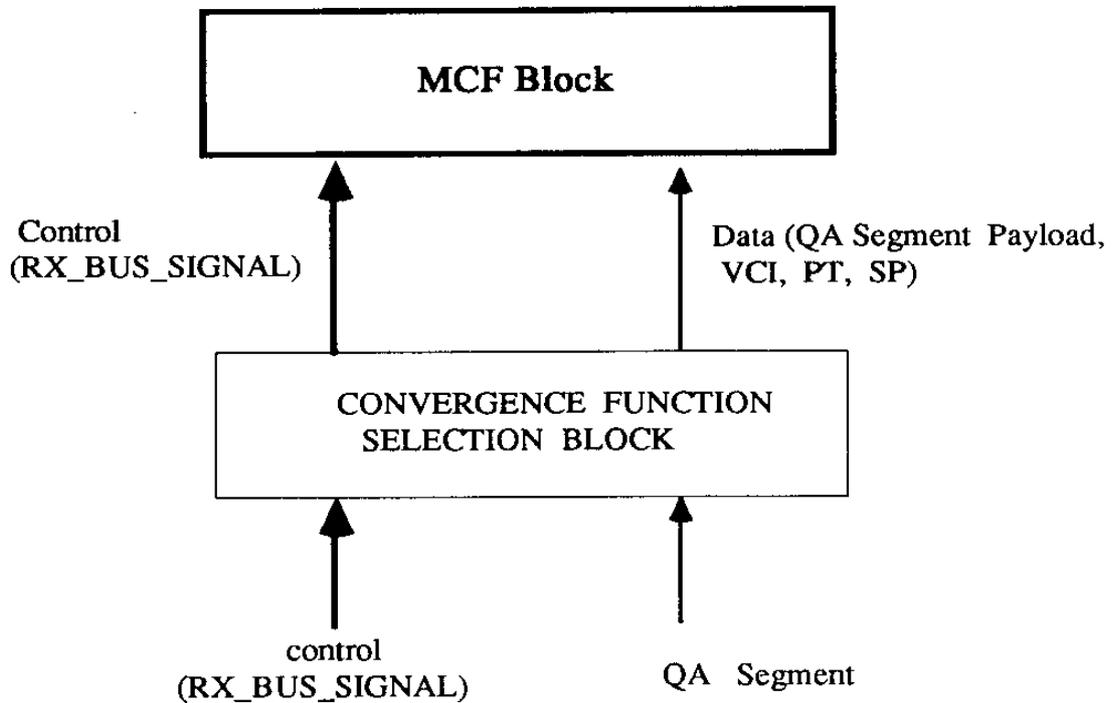


그림 3.24 세그먼트 검사블럭의 기능

QA 세그먼트 헤더 확인 기능이 완료되면 수렴기능선택블럭에서는 QA 세그먼트 헤더로부터 VCI 값을 추출한다. 그 VCI 값이 이 노드의 수렴기능과 관련이 있는지를 검사한다. 만일 그 VCI가 이 노드에 관련이 없으면 수신된 세그먼트는 버린다. 만일 그 VCI가 이 노드에 있는 수렴기능과 관련이 있으면 그 QA 세그먼트가 수신된 버스를 따라 적당한 수렴기능블럭에 전달된다.



- RX_BUS_SIGNAL Control : BUS A or BUS B
- VCI_DATA
- PT_DATA
- SP_DATA
- QA Segment Payload Data

그림 3.25 수신시 수렴기능선택블럭과 MCF 블럭사이의 동작

그림 3.25에서 보는바와 같이 QA 기능블럭에서 MCF블럭으로 데이터 세그먼트를전달할 때 어느 버스에서 데이터가 수신되었는가를 알리기 위한 RX_BUS_SIGNAL 제어신호가 필요하다. MCF수렴기능 선택블럭에서는 QA 세그먼트 정보이외에 VCI, PT, SP 등을 MCF블럭으로 보낸다. VCI_DATA는 QA 세그먼트의 헤더에 수신된 VCI 값을 나타낸다. PT_DATA는 QA 세그먼트의 헤더에 수신된 Payload_Type의 값을 포함하며, SP_DATA는 QA 세그먼트의 헤더내에 수신된 Segment_Priority의 값을 갖는다.

3.3 Common Function(CF)

DQDB 프로토콜의 Common Function block에서는 옥텟 단위의 데이터를 취급하며, 이들은 크게 LLC 등의 상위 계층에서 전달되는 정보를 전송하기 위한 슬롯 옥텟과 DQDB MAC계층 및 물리 계층의 관리 및 제어를 위한 관리 정보 옥텟의 두가지로 나눌 수 있다. DQDB 프로토콜의 Common Function block에서는 이러한 슬롯 옥텟과 관리 정보 옥텟을 전달하여 주는 기능을 비롯하여, 헤드 노드에서 슬롯을 생성하는 기능, 노드 또는 링크에 단절 등의 결함이 생겼을 때 망의 기능을 유지시켜 주는 망구성 제어기능 및 같은 IMPDU(Initial MAC Protocol Data Unit)에서 파생된 DMPDU (Derived MAC Protocol Data UNit)들이 수신 노드에서 재조립될 수 있도록 해주는 MID 페이지 할당기능 등을 수행한다.

본 과제에서는 데이터 전송만을 위한 QA 기능만을 구현하는 것으로 하며, 따라서 Common Function도 QA기능만을 지원하도록 구현될 것이다. Common Function block에서 데이터의 경로를 선정하기 위한 기능 블록도가 그림 3.26에 나타나 있다.

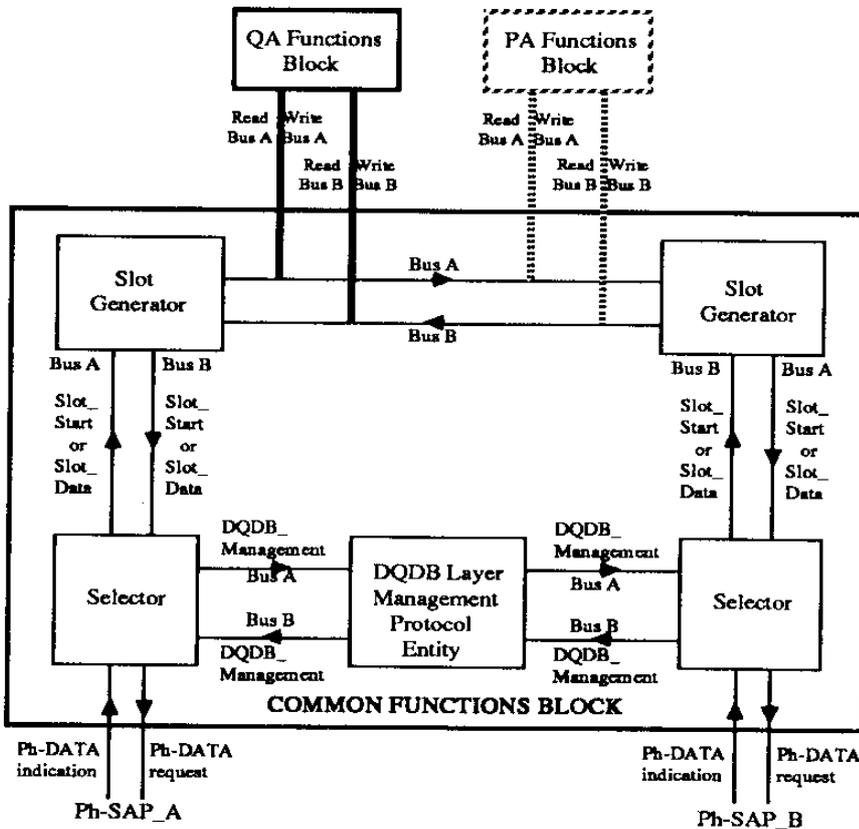
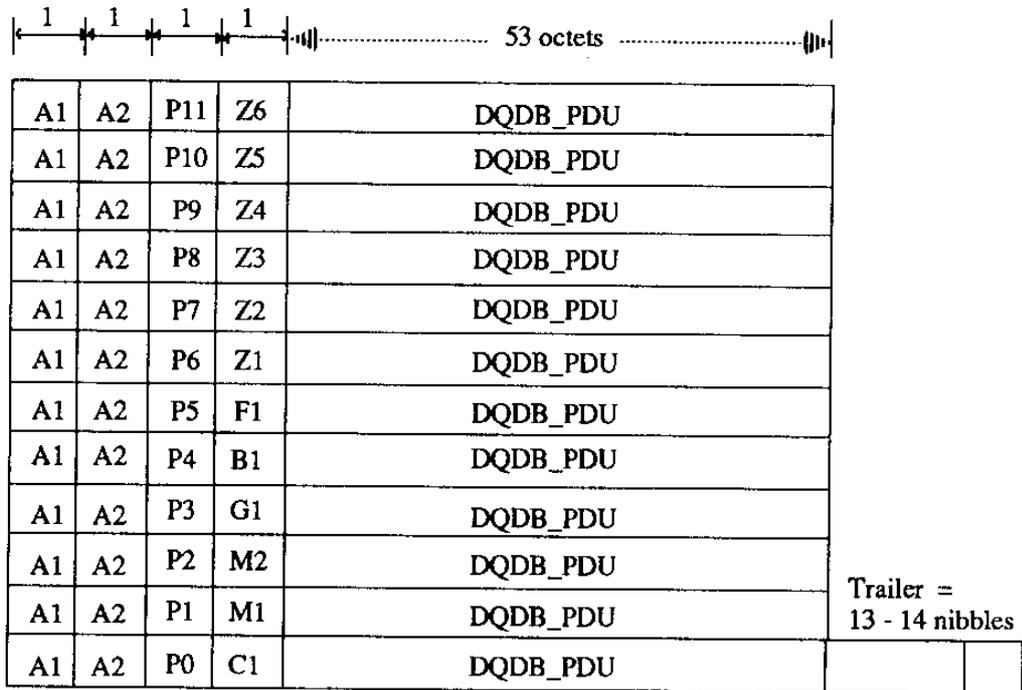


그림 3.26 Common Function Block의 데이터 경로 기능 블록도

Slot Generator에서는 버스 A와 버스 B에 대한 슬롯 옥텟을 생성하며, DQDB LMPE (Layer Management Protocol Entity)에서는 관리 정보 옥텟을 생성한다. 그림 3.26에 나타난 바와 같이 슬롯 옥텟과 관리 정보 옥텟은 모두 Selector를 통하여 물리계층과 접속된다. 슬롯 옥텟과 관리 정보 옥텟을 전송하는 주기는 물리 계층의 전송시스템에 따라 달라지나 본 과제에서는 현재 IEEE802.6 Draft Proposal에 유일하게 규정되어 있는 DS3를 바탕으로 구현하는 것을 원칙으로 하며, 이 경우에 물리 계층의 PLCP(Physical Layer Convergence Procedure)에서는 그림 3.27에 나타난 것과 같은 프레임 형태를 125 usec를 주기로 하여 전송한다. 즉, 125 usec동안 53개의 옥텟으로 구성된 12개의 QA슬롯이 전송되며, 각각의 QA슬롯 앞에는 2개의 프레이밍 옥텟을 포함한 4개의 관리 정보 옥텟이 삽입된다. 따라서 CF block의 Selector에서는 슬롯 옥텟과 관리 정보 옥텟이 그림 3.27에 주어진 프레임 구조의 정해진 위치에 제대로 삽입되도록 주기적으로 경로를 스위칭해야 한다.



PLCP 경로 오버헤드 옥텟 :

Z6 - Z1 = Growth Octets

F1 = PLCP Path User Channel

B1 = BIP-8

G1 = PLCP Path Status

M2 - M1 = Physical Layer Control Information

C1 = Cycle/Stuff Counter

125 usec

그림 3.27 DS3 PLCP 프레임 포맷 구조

헤드 노드에서는 빈 QA슬롯을 생성한다. 이때 QA슬롯의 ACF(Access Control Field)를 포함한 모든 QA세그먼트는 0으로 세트되며 (QA슬롯 포맷 참조), 따라서 헤드 노드의 slot generator에서 생성되는 모든 SLOT_START 또는 SLOT_DATA 옥텟은 모두 0으로 세트되어 생성된다. 각 노드에서는 4.2절에서 설명된 QA 기능 블록을 통하여 생성된 QA슬롯을 슬롯옥텟으로 분리하여 생성한다. 망구성 제어기능과 MID 할당기능은 복잡한 구조를 가지며 이들에 대한 기능 블록도에 대한 설명이 각각 3.3.1 및 3.3.2에 나타나 있다.

3.3.1 망구성 제어기능

DQDB 프로토콜은 노드 또는 링크에 결함이 발생할 경우에도 자동적으로 헤드 노드의 위치를 바꾸어 줌으로써 망의 기능이 유지되도록 하고 있다 (IEEE 802.6 draft proposal Appendix H 참조). 이때 노드는 SG_D(Default Slot Generator) 기능을 지원하는 노드와 그렇지 않은 노드로 구분되며, 이들에 대한 구조가 그림 3.28에 나타나 있다.

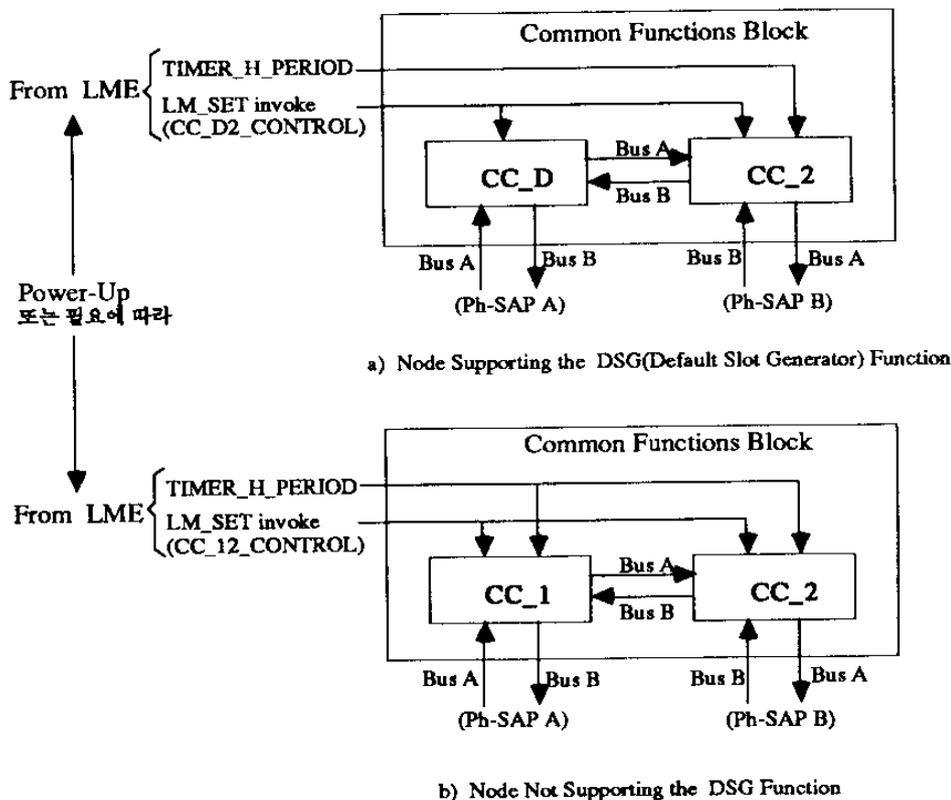


그림 3.28 망구성 제어 기능 구조

Slot Generator는 SG_D(Default Slot Generator), SG_1(Slot Generator Type 1), SG_2(Slot Generator Type 2) 등의 세가지 타입으로 구분되고, 이들은 각각 CC_D, CC_1, CC_2의 망구성 제어기능을 수행한다. SG_D를 가지고 있는 노드는 CC_D와 CC_2 기능 블록을 가지며, SG_D를 가지고 있지 않은 노드는 CC_1과 CC_2 기능 블록을 가지고 있다. 망구성 제어기능 블록의 작동을 위하여서는 망의 초기화 때 또는 망관리자(network administrator)가 망의 재구성이 필요하다고 생각될 때 DQDB LMI(Layer Management Interface)를 통하여 TIMER_H_PERIOD 값을 지정해 주고 LM_invoke(CC_FLAG, flag_type, flag_value) 시그널을 보내야 한다. Flag_type은 SG_D를 가지고 있는가에 따라 CC_D2_CONTROL인가 CC_12_CONTROL인가로 구분되며, flag_value는 NORMAL인가 DISABLE인가를 나타낸다. TIMER_H_PERIOD는 CC_1 및 CC_2 기능 블록에서 망을 재구성하도록 요구되는 시간의 한계를 나타내며 이는 default값을 20초로 한다.

망구성 제어 기능에서 각 노드는 바로 인접 노드들에서 물리 계층 (Ph_SAP_A 또는 Ph_SAP_B)를 통하여 들어오는 G1 관리 정보 옥텟(그림 3.27 참조)의 값을 바탕으로 하여 현재의 망의 상태를 판단하며, M1옥텟내의 비트들을 적절히 세트함으로써 헤드 노드의 위치를 바꾸어 망을 재구성한다. G1 및 M1 옥텟의 구조가 그림 3.29에 나타나 있다.

G1 Octet

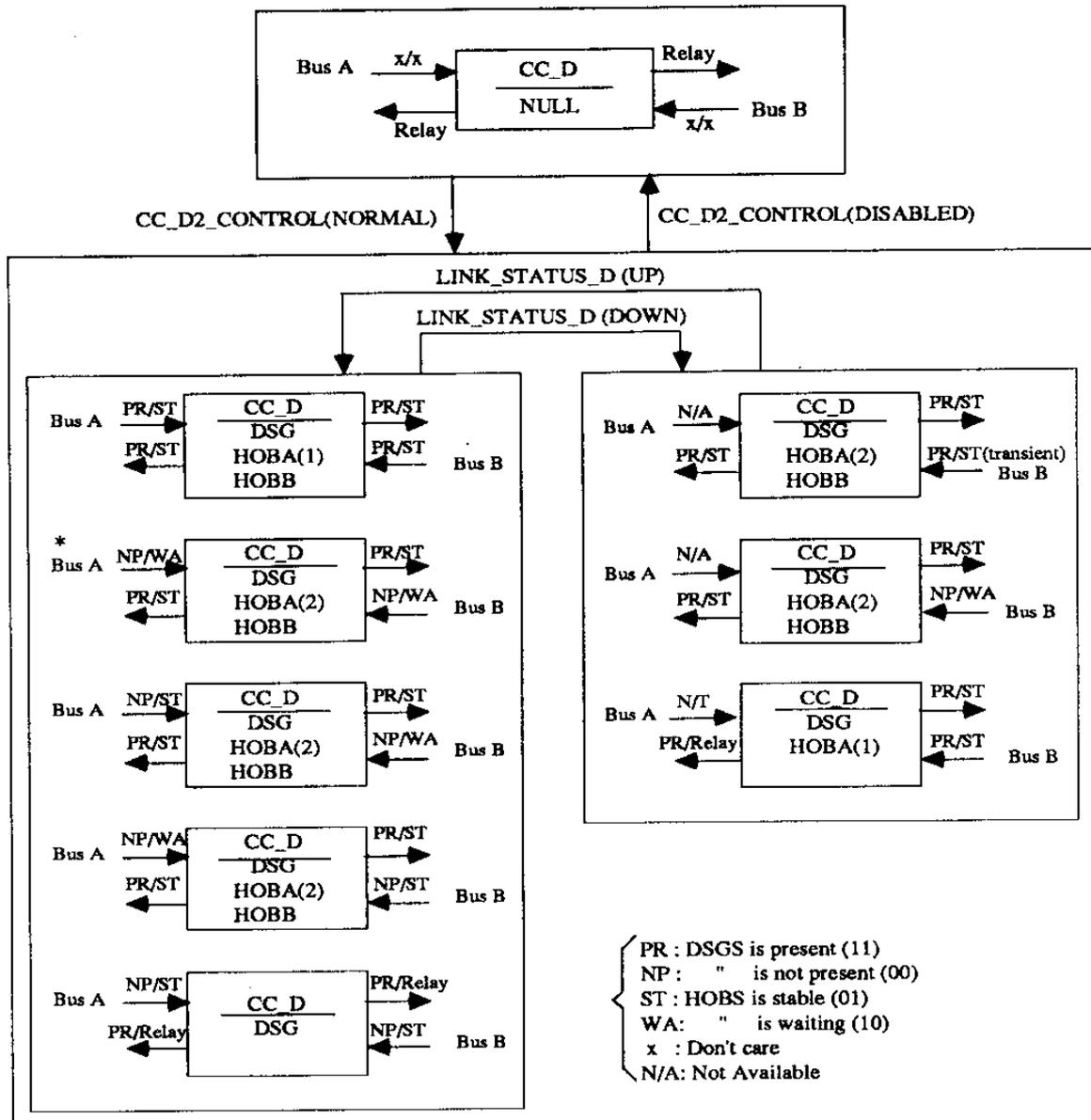
Far End Block Error (4bits)	Yellow Signal (1bit)	Link Status Signal (4bits)
--------------------------------	-------------------------	-------------------------------

M1 Octet

Type	Bus Identification Field(BIF)	Default Slot Generator Subfield(DSGS)	Head Of Bus Subfield(HOBS)	External Timing Source Subfield (ETSS)
0	01 Bus A 10 Bus B	00 Not-Present 11 Present	01 Stable 10 Waiting	0 Not-Present 1 Present

그림 3.29 G1 및 M1 옥텟의 구조

CC_D의 기능 블록도에서 M1 옥텟의 값을 세트하는 방법이 그림 3.30에 나타나 있다.

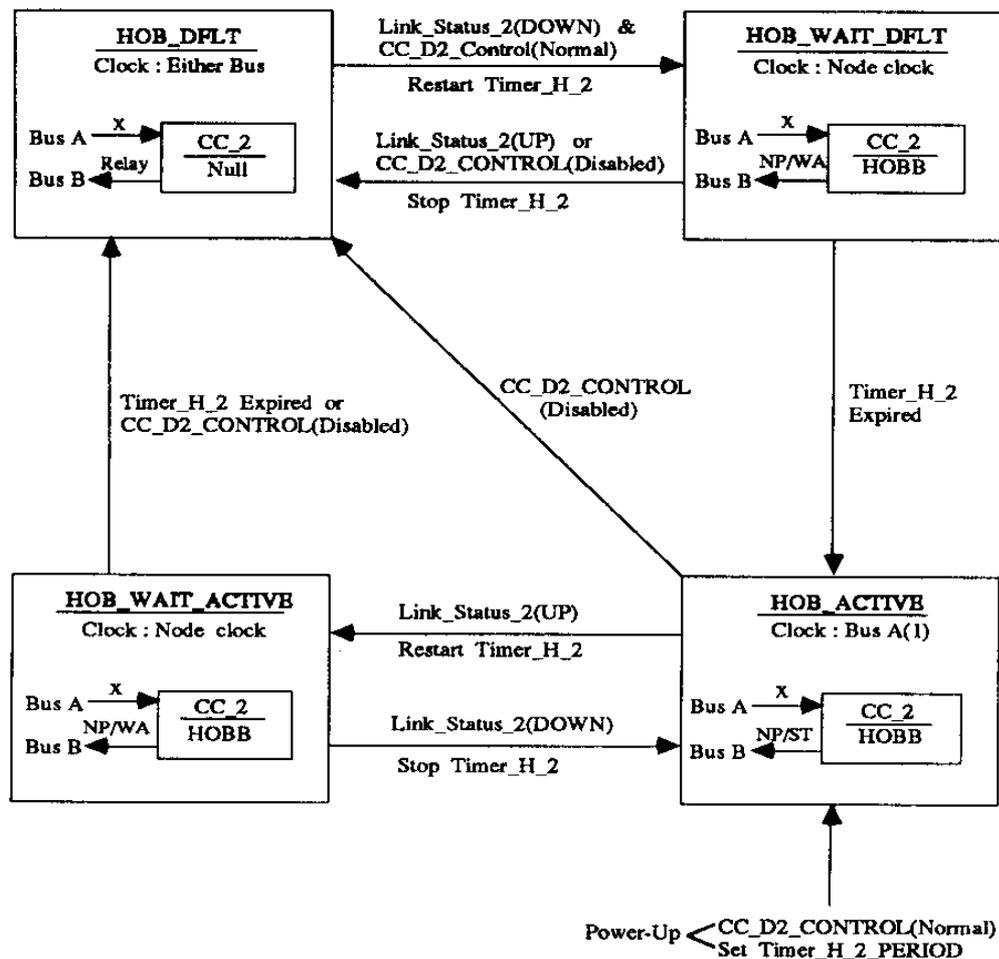


- HOBA(1) : Page Counter State Machine for Head Of Bus A is enabled
- HOBA(2) : " " " " " " disabled
- External Timing Source is disabled (ETSS is set to 0)

그림 3.30 CC_D의 기능 블럭도

그림 3.30에 나타난 바와 같이 CC_D에서는 CC_D2_CONTROL이 DISABLE인 경우에는 버스 A나 버스 B를 통하여 들어오는 M1옥텟을 변환하지 않고 그대로 보낸다, CC_D2_CONTROL이 NORMAL인 경우에는 G1 옥텟의 LSS값에 따라 LINK_STATUS_D의 상태가 UP 또는 DOWN인 가를 판단하여 M1값을 변환한다. 예를 들어 그림에서 *로 표시된 경우와 같이, LINK_STATUS_D 값이 UP일 때, 인접 노드들에서 버스 A와 버스

B를 통하여 M1의 DSGS(Default Slot Generator Dubfield) 필드값이 NP(00)로, HOBS(Head Of Bus Subfield) 필드값이 WA(10)로 세트되어 들어오면, 이 노드는 DSGS 필드를 PR(11)로 또 HOBS 필드를 ST(01)로 변환하여 다음 노드로 보낸다. 이때 이 노드는 그림의 box내에 표시된 것과 같이 DSG기능과 A 및 B 버스의 헤드 기능을 수행한다. 또한 A 버스의 헤드에서 PCSM(Page Counter State Machine)은 작동되지 않도록 한다 (다음의 MID 페이지 할당 기능 참조).



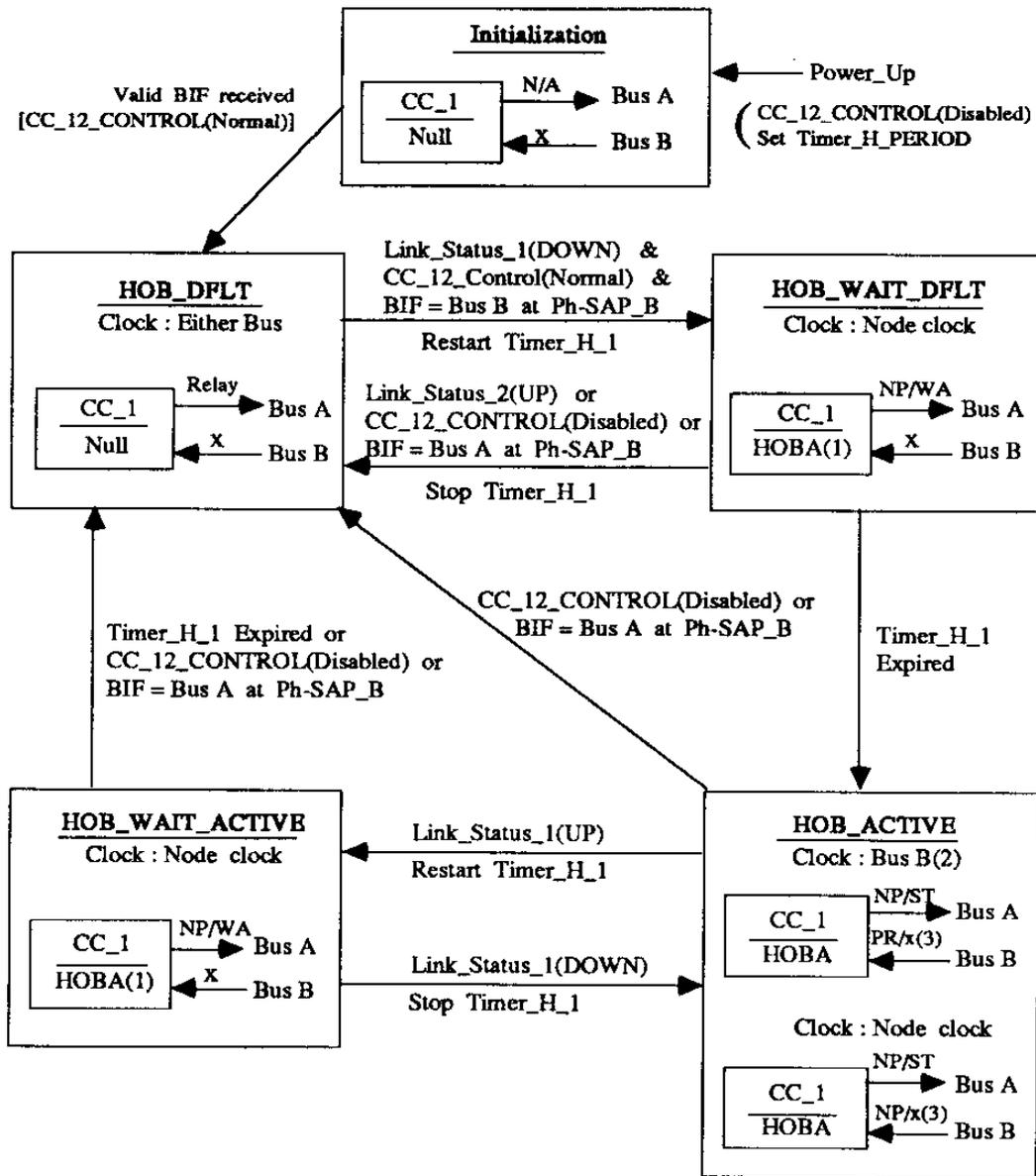
Note
(1) If Link_Status_D is DOWN Clock : Node clock.

그림 3.31 CC_D 기능을 가지고 있는 노드에서 CC_2 기능 블럭도

CC_D 기능 블럭을 가지고 있는 노드에서 CC_2 기능 블럭의 동작 상태도가 그림 3.31에 나타나 있으며, LINK_STATUS의 상태, CC_D2_CONTROL 시그널 및 TIMER_H_PERIOD의 상태에 따라 각 노드가 M1옥텟의 DSGS 및 HOBS 필드 값을 어떻게

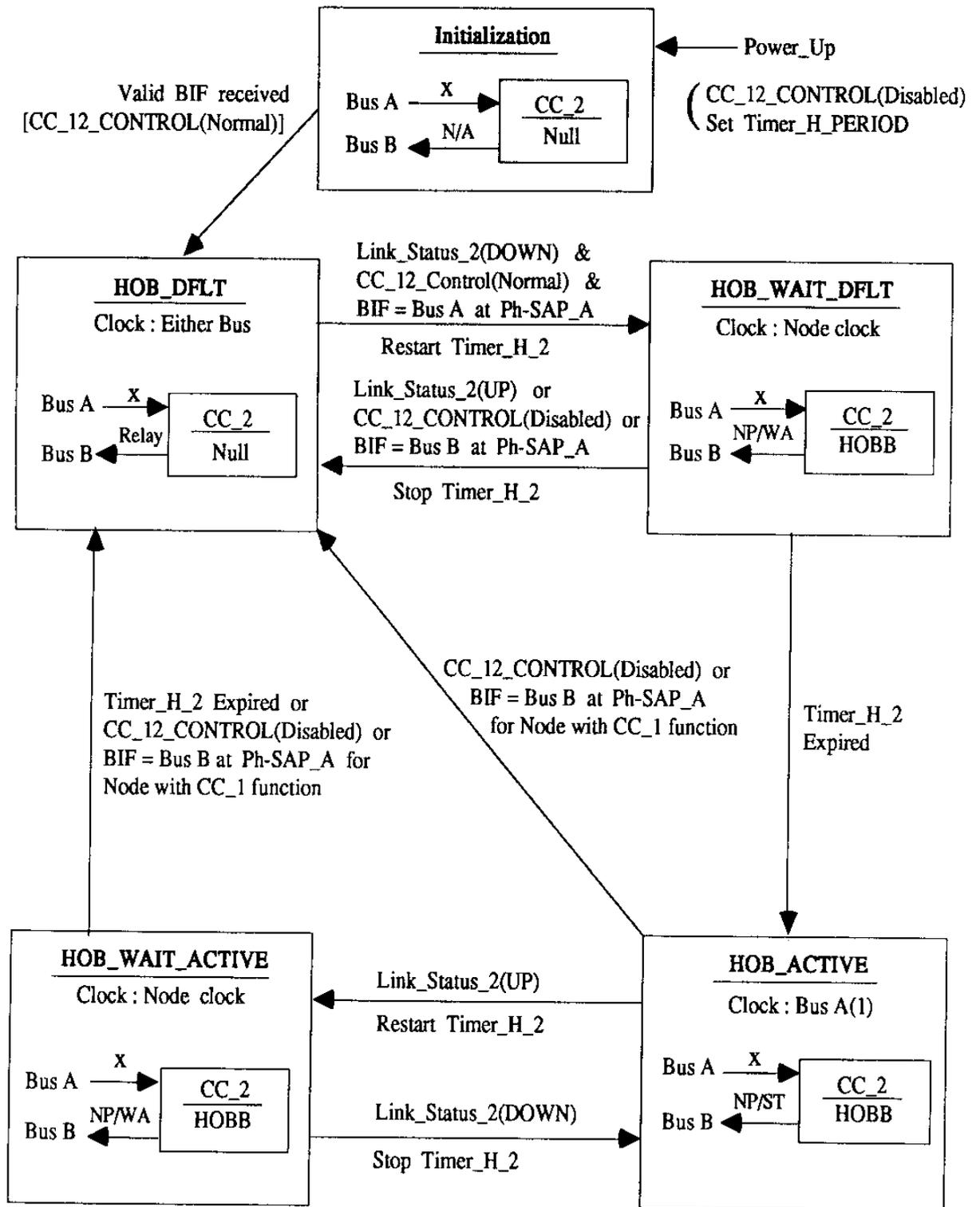
세트하여야 할 것인가, 또한 CC_2블럭은 어떠한 기능을 수행하여야 할 것인가에 대한 것들이 나타나 있다.

그림 3.32와 그림 3.33에는 CC_1 기능 블럭과 CC_1기능 블럭을 가지고 있는 노드에서 CC_2 기능 블럭이 어떻게 작동되어야 할 것인가에 대한 상태도가 각각 나타나 있다.



- . HOBA(1) : Page Counter State Machine for Head Of Bus A is Disabled
- . Clock : Bus B(2) : If Link_Status_2 is DOWN — Clock : Node clock
- . X(3) : If HOBS = ST : Page Counter State Machine for Head Of Bus A is Enabled
- . " = WA: " = Disabled.

그림 3.32 CC_1 기능 블럭도



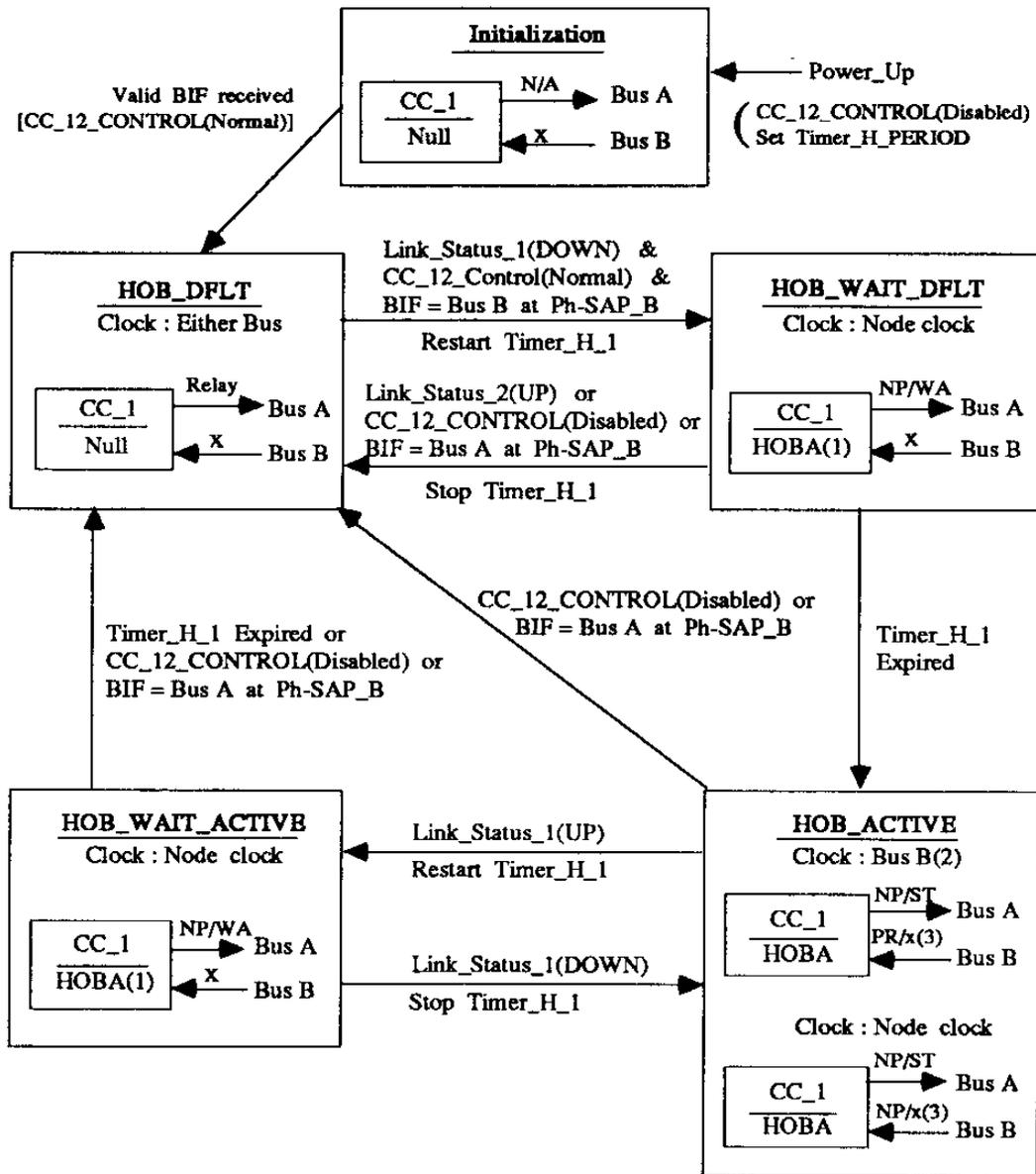
Note

(1) If Link_Status_D is DOWN → Clock : Node clock

그림 3.33 CC_1 기능을 가지고 있는 노드에서 CC_2 기능 블럭도

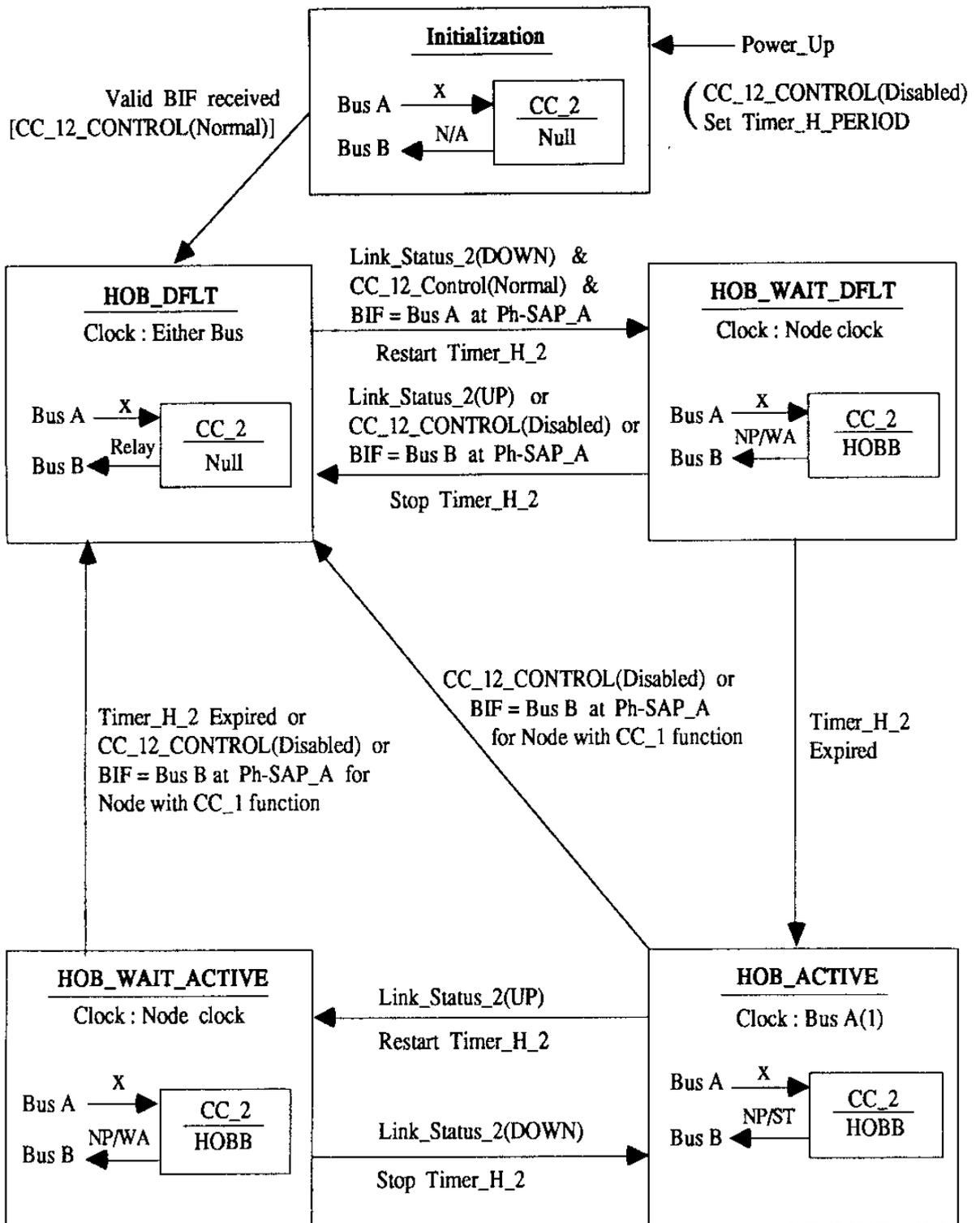
세트하여야 할 것인가, 또한 CC_2블럭은 어떠한 기능을 수행하여야 할 것인가에 대한 것들이 나타나 있다.

그림 3.32와 그림 3.33에는 CC_1 기능 블럭과 CC_1기능 블럭을 가지고 있는 노드에서 CC_2 기능 블럭이 어떻게 작동되어야 할 것인가에 대한 상태도가 각각 나타나 있다.



- . HOBA(1) : Page Counter State Machine for Head Of Bus A is Disabled
- . Clock : Bus B(2) : If Link_Status_2 is DOWN — Clock : Node clock
- . X(3) : If HOBS = ST : Page Counter State Machine for Head Of Bus A is Enabled
- . " = WA: " = Disabled.

그림 3.32 CC_1 기능 블럭도



Note

(1) If Link_Status_D is DOWN → Clock : Node clock

그림 3.33 CC_1 기능을 가지고 있는 노드에서 CC_2 기능 블럭도

3.3.2. MID 페이지 할당 기능

DQDB의 MAC 계층에서는 LLC에서 전달된 MSDU(MAC Service Data Unit)로 IMPDU를 만들고 이를 QA 슬롯에 삽입될 수 있도록 DMPDU로 분리하여 전달한다. 그러나 BOM (Beginning of Message) DMPDU만이 수신 노드의 주소를 가지고 있고 그 이후에 오는 COM(Continuation of Message) 또는 EOM(End of Message) DMPDU들에는 수신 노드의 주소를 기록하지 않는다. 따라서 수신 노드에서는 자기한테 오는 DMPDU들인가를 확인하기 위한 과정이 필요하며, 이를 위하여 송신 노드에서 각 DMPDU의 MID(Message Identifier) 필드에 노드 고유의 MID값을 기록하여 보내고, 수신 노드에서는 이를 통하여 같은 IMPDU에서 파생된 DMPDU들을 추출한다. 따라서 각 노드에 고유의 MID페이지 값을 할당하여야 하며, 이 작업이 Common Function block에서 수행된다.

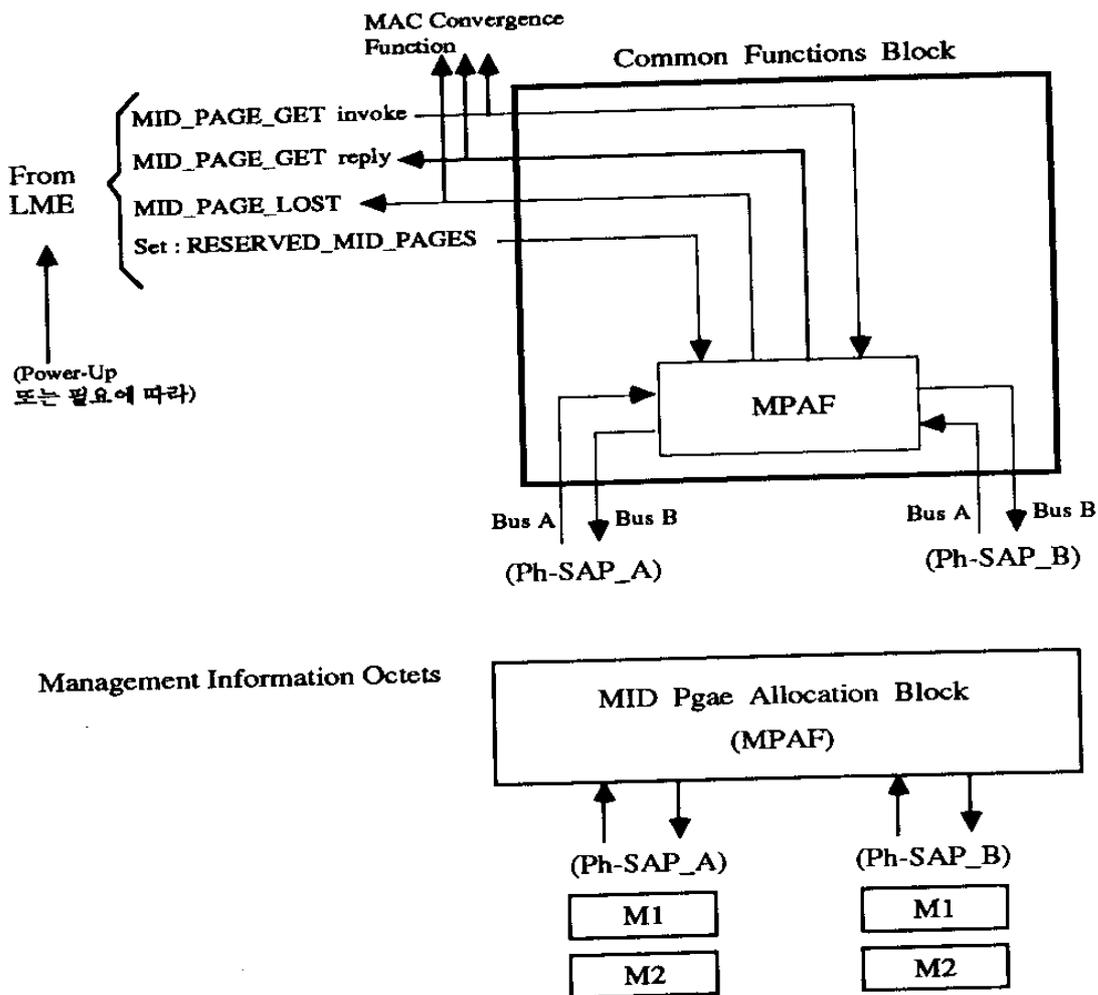


그림 3.34 MID 할당 기능 구조

그림 3.34에서 보는 바와 같이 MID 페이지 할당 기능의 작동을 위하여서는 LMI(Layer Management Interface)를 통하여 다음과 같은 프리미티브들을 송수신 하여야 한다.

- o LM_ACTION invoke (MID_PAGE_GET)
- o LM_ACTION reply (MID_PAGE_GET, status, [mid_page_id], [reason])
- o LM_EVENT notify (MID_PAGE_LOST, mid_page_id, reason)
- o LM_SET invoke (SYSTEM_PARAMETER, parameter_type, parameter_value)

여기서 LM_SET invoke의 parameter_type에는 RESERVED_MID_PAGES가 있으며 default값은 0이다. 각 프리미티브들의 역할은 IEEE802.6 Draft Proposal에 나와 있다.

MID 페이지 할당 기능의 실제 동작은 인접 노드에서 물리 계층을 통하여 들어오는 M1 및 M2 정보 관리 옥텟내의 값들을 적절히 변환함으로써 이루어진다. M1 옥텟의 구조는 그림 3.29에 나와 있으며, M2 옥텟의 구조는 그림 3.35에 주어졌다.

Type	Reserved	Page Reservation (PR)	Page Counter Modulus (PCM)		Page Counter Control (PCC)
1	1	00 : Not-Reserved 11 : Reserved	00 : 0 10 : 2	01 : 1 11 : 3	01 : Reset- 10 : Increase

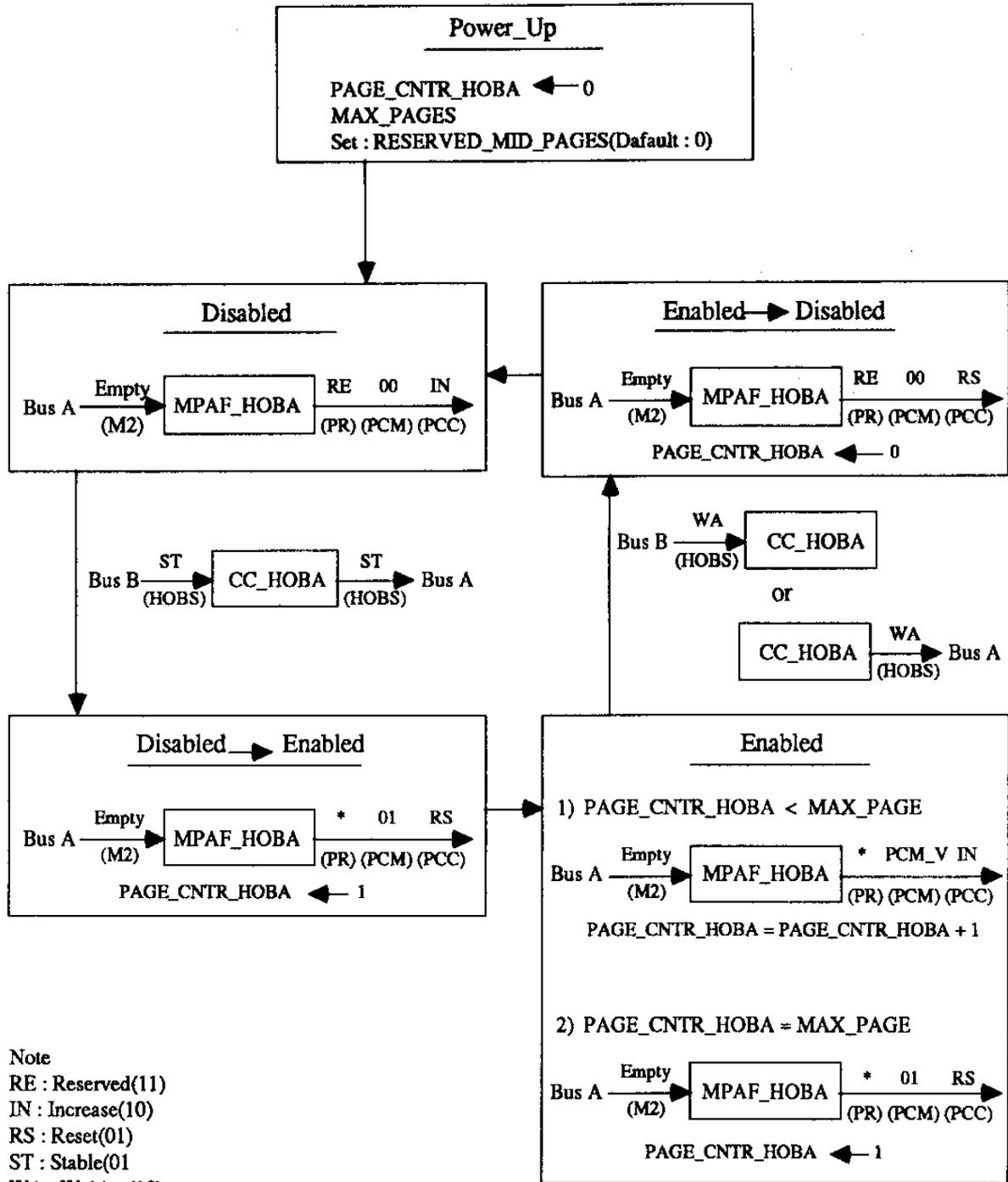
10 : Not-Reserved on Bus A
01 : Reserved on Bus B

00 Increase
11

그림 3.35 M2 옥텟의 구조

MID 할당 기능에서 각 노드는 서로 다른 MID값을 가져야 하며, 버스 A의 헤드에서 일련의 번호를 제공하고 각 노드에서는 이미 다른 노드에서 사용되지 않은 번호를 자기 노드의 MID값으로 채택하는 것으로 이루어진다. 그림 3.36에 MID 페이지 할당을 위한 버스 A의 헤드에서의 기능 블록도가 나타나 있다.

- Counter : PAGE_CNTR_HOBA
- System Parameter : MAX_PAGE, RESERVED_MID_PAGES



Note
 RE : Reserved(11)
 IN : Increase(10)
 RS : Reset(01)
 ST : Stable(01)
 WA : Waiting(10)

PAGE_CNTR_HOBA ≤ RESERVED_MID_PAGES : RE(11)
 PAGE_CNTR_HOBA > RESERVED_MID_PAGES : NR : Not-Reserved(00)
 PCM_V = (Value of PAGE_CNTR_HOBA) (mod 4)

그림 3.36 버스 A의 헤드에서의 MID 페이지 할당 기능 블록도

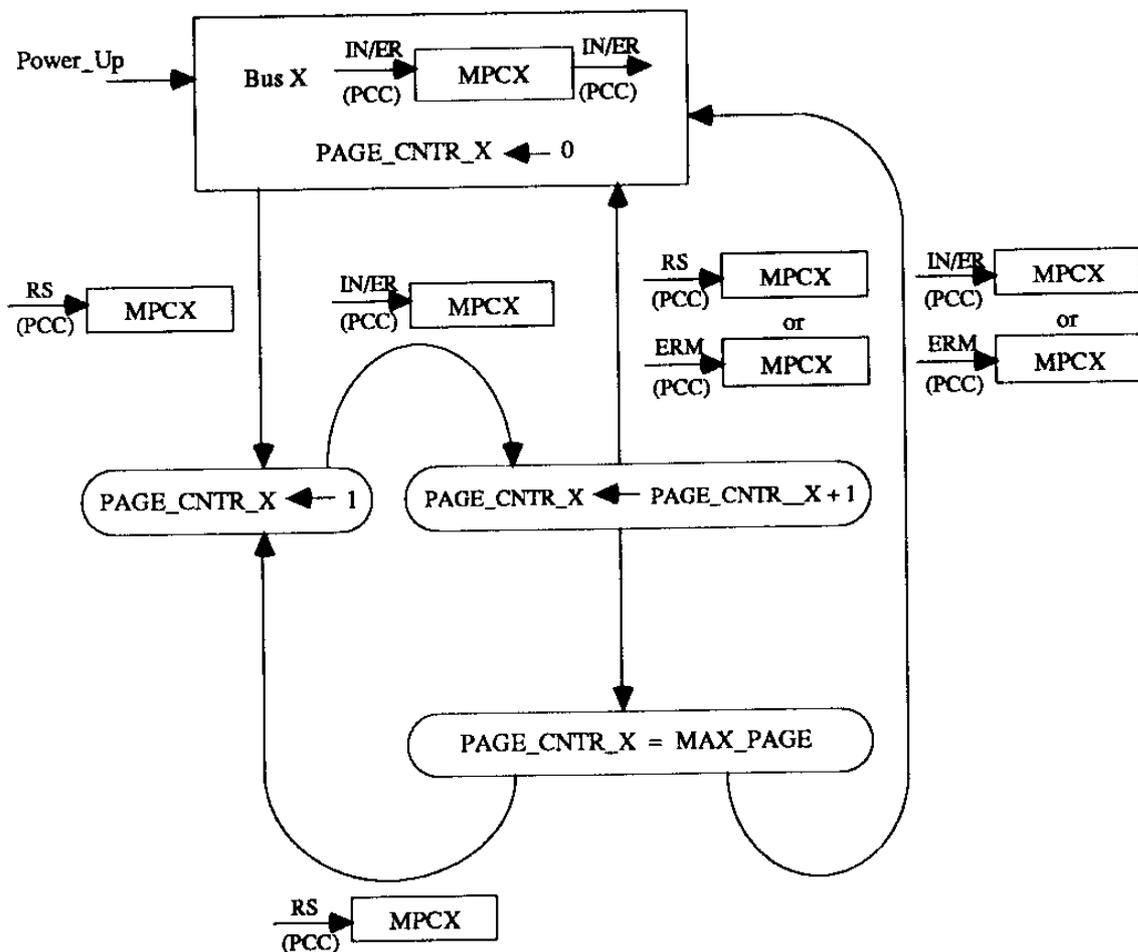
버스 A의 헤드에는 PAGE_CNTR_HOBA라는 계수기와 시스템 파라미터인 MAX_PAGE와 RESERVED_MID_PAGES가 있다. 망의 초기화 때 PAGE_CNTR_HOBA 계수기 값은 0으로 세트한다. 망의 초기화가 끝난 후 버스 A의 헤드에서의 상태는 ENABLE, DISABLE 및 이들의 중간 천이 상태로 나누어지며, 각 상태에서 M2 정보 관리 옥텟을 그림에 나타난 대로 변환한다. 즉, DISABLE 상태에 있을 때 버스 B의 M1옥텟의 HOBS 필드 값이 ST(01)로 세트되어 들어오면, 버스 A로 전달되는 M1옥텟의 HOBS 필드값을 ST(01)로 세트하여 보냄으로써 DISABLE 상태에서 ENABLE 상태로 천이하게 되며, 이때 PAGE_CNTR_HOBA 계수기 값을 1로 하여 버스 A의 헤드는 ENABLE 상태로 들어가게 된다. ENABLE 상태에서 버스 A의 헤드는 M2옥텟이 전송될 때마다 PAGE_CNTR_HOBA 계수기의 값을 하나씩 증가시키며, PAGE_CNTR_HOBA 계수기 값이 시스템 파라미터인 MAX_PAGE 값에 도달할 때까지 M2 옥텟의 PCC 필드 값을 IN(10)으로 세트하여 내보내고, MAX_PAGE값에 일치하면 RS(01)로 세트하여 내보낸다. 이때 PAGE_CNTR_HOBA 계수기 값은 다시 1로 세트된다. 만일 망의 어떤 부분에 결함이 생겨 버스 B의 M1 옥텟의 HOBS 필드 값이 WA(10)으로 세트되어 들어오면, 버스 A로 전달되는 M1 옥텟의 HOBS 필드값을 WA(10)으로 세트하여 보내고, PAGE_CNTR_HOBA 계수기 값을 0으로 하여 DISABLE 상태로 들어간다.

버스 A 헤드 노드를 제외한 각 노드에는 MAX_PAGE 시스템 파라미터를 비롯하여, KPSM(Keep Page State Machine)을 작동하기 위한 PAGE_CNTR_A 계수기와 GPSM(Get Page State Machine)을 작동하기 위한 PAGE_CNTR_B 계수기가 있으며, 이 두 계수기의 값은 항상 헤드 노드의 PAGE_CNTR_HOBA 계수기 값과 일치하여야 한다. 각 노드에서 PAGE_CNTR_X (X=A 또는 B) 계수기는 버스 A의 헤드 노드로부터 전달되는 M2 옥텟의 PCC 필드값에 따라 조정된다. 버스 A의 헤드 노드로부터 전달되기 시작한 M2 옥텟이 버스 B의 헤드에 도달하면 버스 B의 헤드에서는 이를 반복하여 다시 버스 A의 헤드로 보낸다. PAGE_CNTR_X 계수기는 각 노드의 MPCX(MID Page Counter Machine for Bus X)에서 동작되며, 그림 3.37에 MPCX의 구조도가 나타나 있다.

그림 3.37에서 보는 바와 같이 MPCX에서 PAGE_CNTR_X 값이 0으로 되어 있을때, 상위 인접 노드에서 전달되는 M2 옥텟의 PCC 필드값이 RS(01)로 세트되어 있으면 PAGE_CNTR_X 값을 1로 전환한다. 이후에 오는 옥텟의 PCC 값이 IN 또는 ER (10, 00, 11)로 세트되어 들어올 때마다 PAGE_CNTR_X의 값을 하나씩 증가한다. 이때 만일 PCC 필드값이 RS로 세트되거나 PCM 필드값으로부터 예러가 탐지될 경우에는 PAGE_

CNTR_X 값을 0으로 전환한 후 PCC 필드값이 RS로 세트된 M2 옥텟이 도달할 때까지 대기상태로 들어간다. PAGE_CNTR_X 값이 MAX_PAGE에 도달하면 다음에 오는 M2 옥텟의 PCC 필드값이 RS로 세트되어 있으면 PAGE_CNTR_X 값을 다시 1로 세트하여 위의 과정을 반복하고, PCC 필드값이 IN 또는 ER로 세트되어 있거나 또는 PCM 필드에서 에러가 검출되면 PAGE_CNTR_X 값을 0으로 전환한 후 PCC필드 값이 RS로 세트된 M2 옥텟이 도달할 때까지 대기상태로 들어간다.

- Counter : PAGE_CNTR_X (X = A or B)
- System Parameter : MAX_PAGE

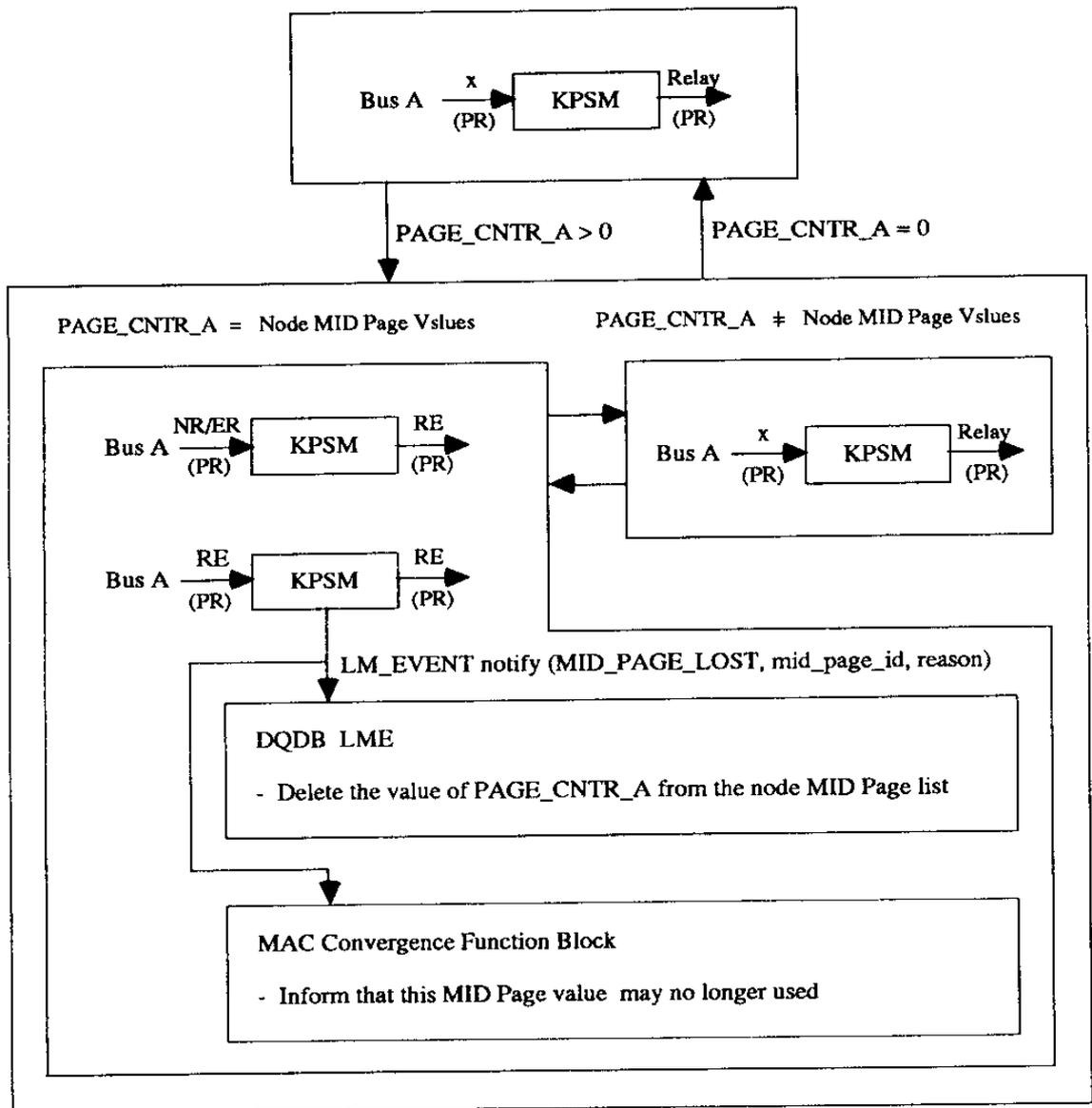


Note :
 IN/ER : Increment or Error (10, 11, 00)
 RS : Reset(01)
 ERM : PCM(x) <> [(PAGE_CNTR_X) (mod 4)]

그림 3.37 MPCX의 기능 구조

각 노드가 현재 가지고 있는 MID 페이지 값을 그대로 보유할 것인가는 KPSM에서 결정되며, 이는 버스 A의 PAGE_CNTR_A 계수기를 통하여 작동된다. 그림 3.38에 KPSM의 기능 구조도가 나타나 있다.

- Counter : PAGE_CNTR_A



Note
 NR/RE : Not-Reserved or Error(00, 10, 01)
 RE : Reserved(11)

그림 3.38 KPSM의 기능 구조

PAGE_CNTR_A 계수기가 0일 경우에는 KPSM은 작동되지 않으며, 이 때는 버스 A를 통하여 들어오는 M2 옥텟의 PR 필드를 변환하지 않고 그대로 다음 노드로 전달한다 (MPCX 대기 상태). MPCX가 정상으로 작동함에 따라 PAGE_CNTR_A 계수기 값은 M2 옥텟이 도착할 때마다 하나씩 증가된다.

PAGE_CNTR_A 계수기 값이 현재 노드가 보유하고 있는 MID 페이지 값과 다를 경우에는 M2 옥텟을 그대로 다음 노드로 보낸다. PAGE_CNTR_A 계수기 값이 현재 노드가 보유하고 있는 MID 페이지 값과 같을 경우에, 도착한 M2 옥텟의 PR 필드가 NR 또는 ER (00, 10, 01)로 세트되어 있으면 그 값을 RE(11)로 전환하여 보냄으로써 현재 PAGE_CNTR_A 계수기에 해당하는 번호가 이미 MID 페이지 값으로 할당되었음을 다른 노드에 알린다. 그러나 도착한 M2 옥텟의 PR 필드가 RE로 세트되어 있으면 해당되는 MID 페이지 값이 이미 다른 노드에서 사용되고 있음을 나타내는 것이므로 이때는 즉시 LM_EVENT notify 프리미티브를 LMI를 통하여 보냄으로써 망 관리자에게 자신의 MID 값이 중복되었음을 알리고, 또한 같은 프리미티브를 MAC Convergence Function 블록에 보냄으로써 현재의 MID값을 더이상 사용하지 못하도록 한다.

각 노드가 새로운 MID 페이지 값을 할당 받을 것인가에 대한 기능은 GPSM에서 수행되며, 이는 버스 B의 PAGE_CNTR_B 계수기를 통하여 작동된다. 그림 3.39에 GPSM의 기능 구조도가 나타나 있다.

GPSM은 망 관리자로부터 LMI를 통하여 LM_ACTION invoke(MID_PAGE_GET) 프리미티브를 받음으로써 작동된다. 이때 PAGE_CNTR_B 계수기 값이 0이면 버스 B를 통하여 들어오는 M2옥텟을 변환하지 않고 그대로 다음 노드로 전달한다. PAGE_CNTR_B 값이 0보다 클 때 버스 B로 들어오는 M2옥텟의 PR 필드값이 RE 또는 ER (11, 10, 01)로 세트되었으면 M2 옥텟을 그대로 통과시키고 PR 필드값이 NR(00)로 세트되었으면 이때의 PAGE_CNTR_B 값을 그 노드의 MID 페이지 값으로 취하며, PR 필드값을 RE(11)로 전환하여 M2옥텟을 다음 노드로 전달한다. 이 노드는 또한 망 관리자에게 LM_ACTION reply(MID_PAGE_GET, SUCCESSFUL, mid_page_id) 프리미티브를 통해 MID값을 성공적으로 할당받았음을 알리고, 같은 프리미티브를 MAC Convergence Function 블록에 보냄으로써 이 값을 MID페이지 값으로 사용하도록 한다.

- Counter : PAGE_CNTR_B

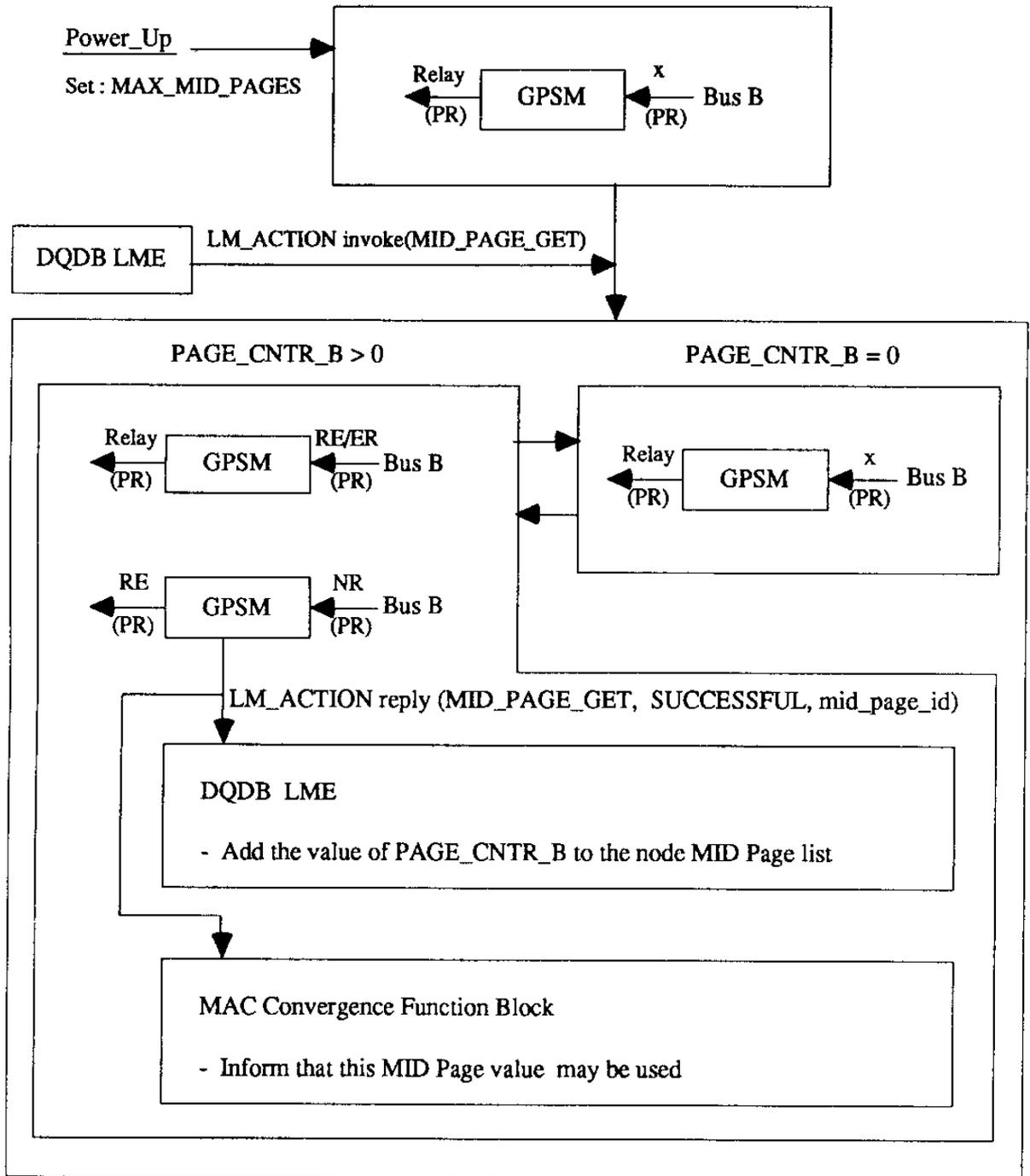


그림 3.39 GPSM의 기능 구조

제 4 장 DQDB MAN 성능 분석

제 4 장 DQDB MAN의 성능 분석

성능 분석은 DQDB MAN이 실제로 구현되었을 경우 어느 정도의 서비스를 제공할 수 있을 것인 가를 미리 예측할 수 있도록 하며, 따라서 망 구현시 고려하여야 할 중요항 사항 중의 하나이다. 특히 현재 IEEE802.6 DQDB MAN의 표준안 작성에 최대의 걸림돌이 되고 있는 공평성과 우선순위에 관한 문제가 실제 망에서 어느 정도 심각하게 일어나는 가를 검토하여 보았다. 또한 망의 어떠한 요소들이 망의 공평성과 우선순위에 영향을 주는 가를 검토하였으며, 또한 이러한 문제점들을 완화시킬 수 있는 방안들을 모색하여 보았다.

DQDB MAN의 성능 분석은 시뮬레이션 모델을 사용하였으며, 시뮬레이션 모델은 discrete event simulation을 위하여 개발된 SIMAN[11]을 이용하여 만들어졌다.

4.1 DQDB 성능 영향요소

DQDB는 구조적으로 헤드에서 가까운 쪽의 노드가 빈 슬롯을 우선적으로 사용하게 됨으로써 더 많은 대역폭을 차지하여 노드간에 공평성이 유지되지 못하는 특성이 있다. DQDB 성능의 공평성 및 우선순위 문제는 처음 다음과 같은 세가지 시나리오에 의하여 제기되었다[12]. 각노드의 트래픽부하는 100%이고 노드사이는 100슬롯거리로 가정하였다. 그림 4.1과 같은 경우에 upstream에 위치한 노드가 낮은 우선순위의 메시지를 전송하고 downstream 노드가 높은 우선순위의 메시지를 전송하는 경우 시뮬레이션 결과 두 노드가 모두 동일한 대역폭인 47%를 사용하는 결과로 나타내고 있다.



그림 4.1 낮은 우선순위 노드가 헤더에 가까이 위치한 경우

그림 4.2는 두 노드의 우선 순위가 높은 경우 시뮬레이션 결과 upstream 노드와 downstream 노드가 각각 대역폭을 89%, 9%로 나누어 사용됨을 나타내고 있다.

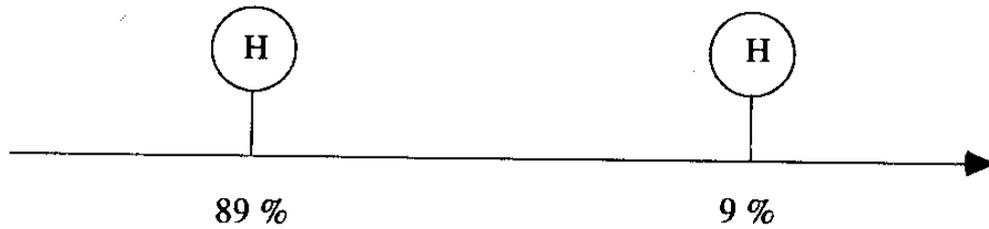


그림 4.2 높은 우선순위 노드간에 거리가 상당히 떨어져 있는 경우

그림 4.3은 높은 우선순위를 갖는 노드 사이에 낮은 우선순위의 노드가 위치할 경우 시뮬레이션 결과 upstream 노드가 80%, 중간노드가 9%, downstream 노드가 9%로 대역폭을 사용하는 것으로 나타났다.

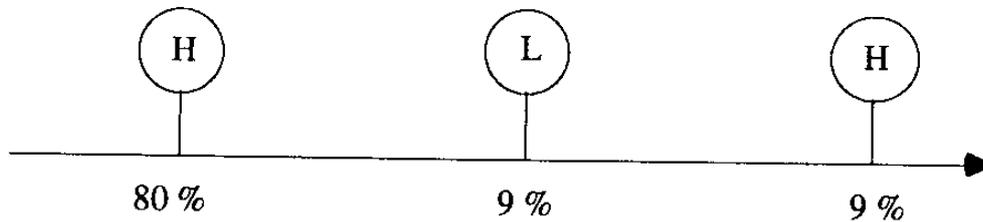


그림 4.3 높은 우선순위 노드사이에 낮은 우선순위 노드가 위치한 경우

그림 4.1은 낮은 우선순위를 갖는 노드가 높은 우선순위를 갖는 노드와 동일하게 대역폭을 사용하는 된 결과이고, 그림 4.2은 동일하게 높은 우선순위를 갖는 노드간에도 upstream에 위치한 노드가 대역폭을 거의 사용하는 결과이다. 그림 4.3의 경우는 높은 우선 순위를 갖는 노드 사이에 낮은 우선순위를 갖는 노드가 위치할 경우 중간의 노드와 downstream 노드가 사용할 수 있는 대역폭은 일부에 지나지 않는 결과를 보이고 있다. 이러한 경우가 실제 망에서 발생하는가의 문제가 IEEE 802.6 회의에서 제기되었다. 실제 망에서

트래픽 부하가 100%이상으로 되는 경우는 매우 드물지만 트래픽이 높아짐에 따라 이에 가까운 현상이 나타나게 되며 망 운용시 전송되어야 할 높은 우선순위의 메시지가 낮은 우선순위를 갖는 메시지 보다 늦게 전달되는 경우도 발생한다. 이 문제는 현재 IEEE 802.6 DQDB MAN의 표준안 작업에 최대 걸림돌이 되고 있다. 이 절에서는 DQDB의 공정성 및 우선순위 성능을 개선하기 위한 기초작업으로 이들이 망의 어떠한 요소에 의하여 영향을 받는가에 대하여 알아보았다. 각 요소들에 의한 성능변화를 관측하기 위해 메시지의 길이는 평균 5 슬롯이며 길이는 기하분포를 갖도록하였다. 지연특성은 하나의 메시지를 보내는데 몇개의 슬롯시간만큼 지연되는가를 슬롯지연으로 나타내었다.

4.1.1 트래픽 부하에 의한 성능 변화

앞서 언급한 바와 같이 공정성과 우선순위는 망의 트래픽 부하에 크게 영향을 받는다. 트래픽 부하가 망의 성능에 영향을 시뮬레이션 모델을 통하여 알아보았으며 이에 대한 결과가 그림 4.9에 나와 있다. 이 그림에서 노드번호는 헤드에서 멀수록 증가하는 것으로 하였다. 각 노드에서 메시지 분포는 동일한 것으로 하였으며 메시지의 도착율을 증가시킴으로써 트래픽 부하가 증가하도록 하였다. 그림에서 보는 바와 같이 낮은 트래픽에서는 공정성 문제가 고려될 필요가 없으나 트래픽이 증가할수록 노드간의 불균형이 심화되며, 비록 트래픽 부하가 100%이하가 되더라도 망 운용시 트래픽 부하는 일정한 상태에 있지 않고 수시로 달라지게 되므로 높은 트래픽 부하에서 공정성문제는 심각하게 고려되어야 할 것으로 판단된다.

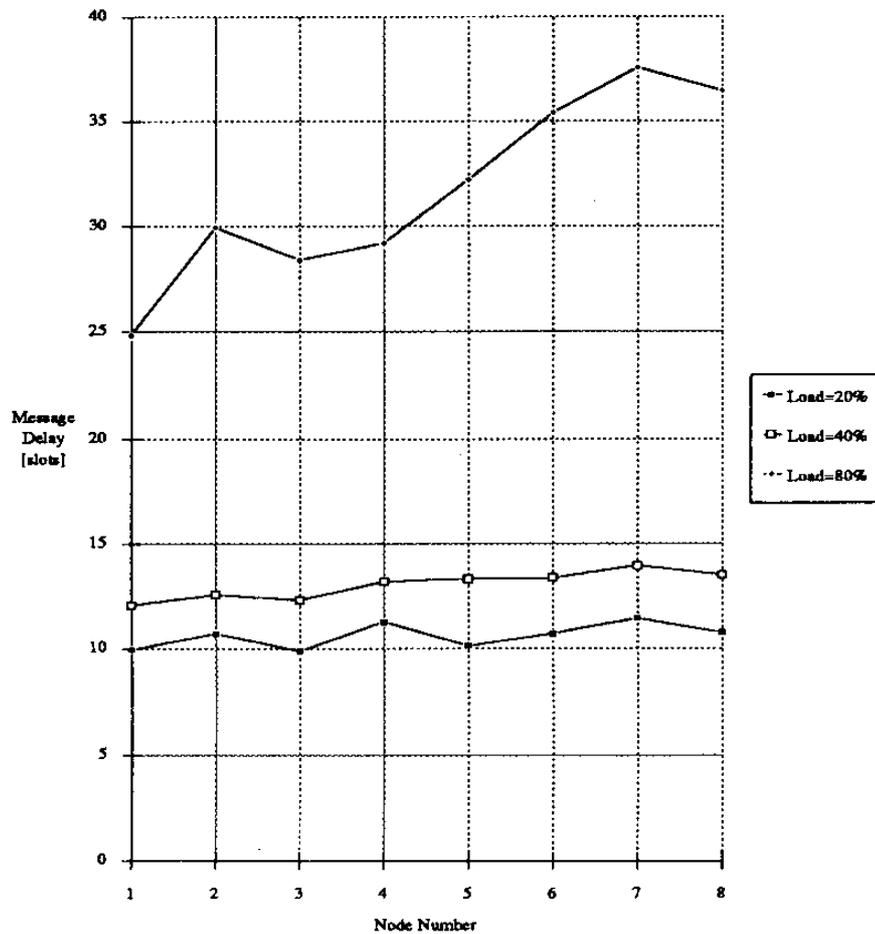


그림 4.4 트래픽 부하에 의한 성능 변화

4.1.2 노드간 거리에 의한 성능 변화

그림 4.5는 DQDB망에서 노드간의 거리가 메시지 지연 특성에 주는 영향을 시뮬레이션의 결과를 통하여 알아보았다. 여기서 각 노드는 같은 우선순위를 가지며 동일한 메시지의 분포를 갖는 것으로 하였다. 그림에서 보는 바와 같이 불균형은 노드간의 거리가 증가될수록 심화된다. 현재 DQDB 망에서 upstream 노드는 downstream 노드가 빈슬롯을 사용하기

위해 역방향으로 REQ 비트를 세트한 사실을 감지하면 곧 자신의 RQ 계수기나 CD 계수기를 증가시켜 순방향으로 전달되는 빈 슬롯을 사용하지 않고 통과시킴으로써 빈 슬롯을 양보한다. 그러나 노드간 거리가 증가되면 downstream 노드로부터 REQ 비트를 세트한 사실이 전달되어 오는 시간이 증가되고, 이 사이에 자신이 먼저 REQ 비트를 세트하여 헤드로부터 전달되어 오는 빈 슬롯을 사용하기 때문에 노드간의 거리가 증가할수록 망의 불균형이 심화된다. 또한 downstream의 높은 우선순위의 노드가 REQ 비트를 세트한 사실도 upstream의 낮은 우선순위 노드에 전달되는데 시간이 걸림으로 해서 낮은 우선순위 노드가 높은 우선순위 노드보다 빈 슬롯을 우선적으로 사용할 수 있다. 따라서 노드간의 거리는 DQDB 망을 설계하는데 중요한 요소로 고려하여야 한다.

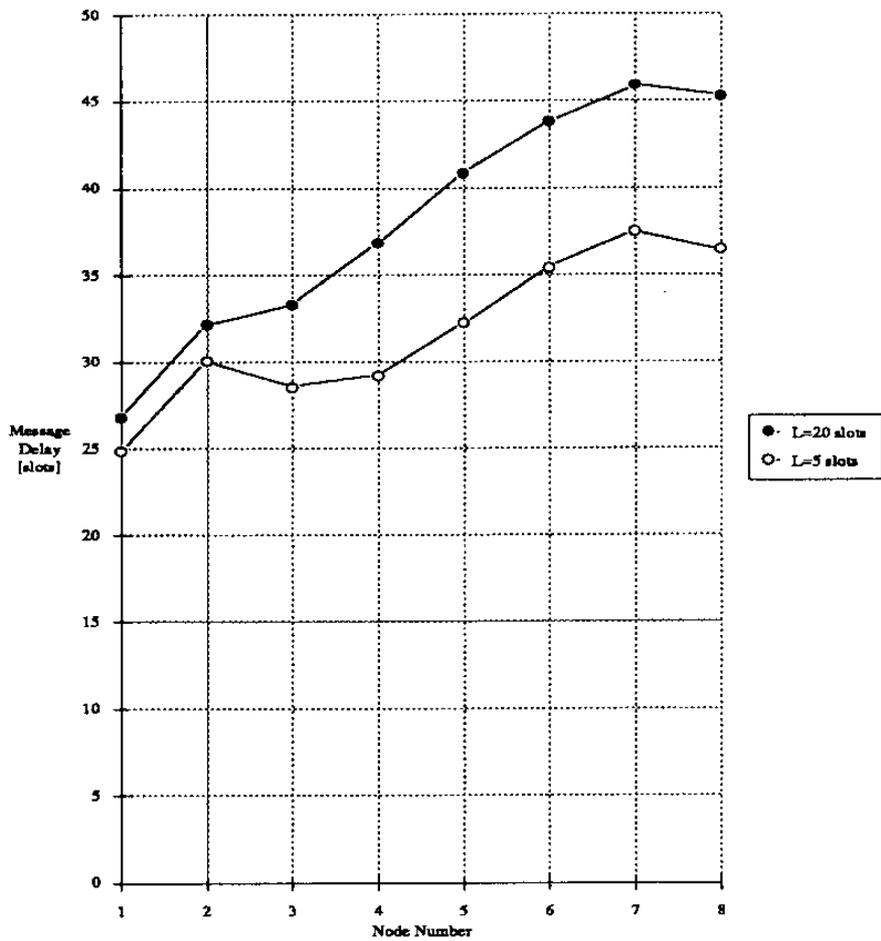


그림 4.5 노드간 거리에 의한 성능 변화

4.1.3 전송 속도에 의한 성능 변화

DQDB 프로토콜의 전송시스템으로써 현재 1.5Mbps, 45Mbps, 150Mbps 등이 검토되고 있다. DQDB에서는 슬롯단위로 REQ 비트를 세트하며 속도가 높아짐에 따라 노드 사이에 많은 빈 슬롯이 존재하게 된다. 따라서 downstream 노드가 REQ 비트를 세트한 슬롯이 앞서 지나간 많은 슬롯이 upstream 노드를 통과한 후까지 기다렸다가 도달하는 결과가 되며, 이러한 현상은 속도가 높아짐에 따라 노드간 유효 슬롯거리(단위 거리당 몇개의 슬롯이 포함되는가 하는 상대적 개념)가 증대하는 효과가 나타나서 그림 4.5에서와 같이 노드간의 불균형이 심화된다.

4.1.4 우선순위 노드의 위치에 의한 성능 변화

그림 4.6에서는 낮은 우선순위 노드(L)와 높은 우선순위 노드(H)간의 상대적인 위치가 망의 성능에 어떠한 영향을 미치는가에 대하여 알아보았다. 그림에서 보면 낮은 우선순위 노드가 높은 우선순위 노드보다 upstream 방향에 위치하면 높은 우선순위의 정보는 그만큼 지연된다. 최악의 경우에는 낮은 우선순위 노드와 높은 우선순위 노드간의 메시지 지연의 차이가 거의 없게 된다. 우선순위 노드의 위치에 따라 성능이 변한다는 사실은 망의 설계 및 관리를 매우 어렵게 만드는 요소이며 현재 이를 해결하기 위한 많은 방안들이 제시되고 있다.

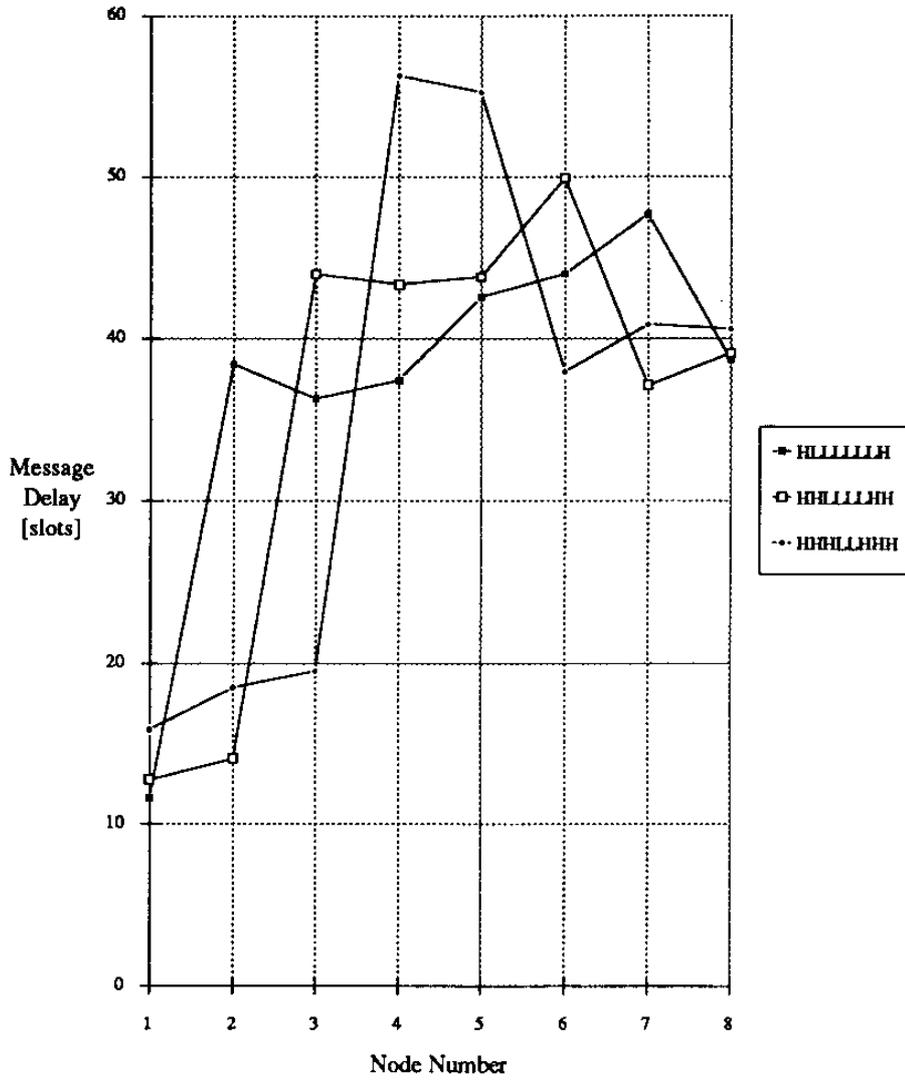


그림 4.6 우선순위 노드의 위치에 의한 성능 변화

다음 절에서는 우선순위 노드의 개념을 적용할 경우 망의 공평성을 개선할 수 있는 몇가지 방안을 제시하였다. 제시된 방안의 타당성에 대해서는 현재 검증중에 있다.

4.2 DQDB의 공평성 개선

DQDB의 특성은 이미 앞에서 언급한 바와 같이 노드간의 거리가 멀수록 각 노드마다 세그먼트를 전송지연특성이 달라진다. 특히 높은 우선순위를 갖는 노드임에도 불구하고 낮은 우선순위를 갖는 노드보다 전송능력이 떨어지게 된다. 이러한 현상은 트래픽이 높은 경우에 더욱 심하게 나타나게 됨으로써 문제의 심각성을 더하고 있다. 여기서는 이러한 문제점을 해결하기 위한 몇가지 방안을 제시하고 있으며 제안된 방안들은 현재 시뮬레이션에 의해 검토중에 있다.

4.2.1 대역폭균형계수기에 의한 제어

대역폭균형계수기의 동작은 세그먼트를 많이 송신한 노드에게 송신한 만큼의 일부 대역폭을 downstream 노드에게 양보함으로써 높은 헤드에서 먼 위치에 있는 노드의 성능을 개선하는 것이다. 즉, upstream 노드들로 하여금 자신이 빈 슬롯을 사용한 수가 대역폭균형계수의 값에 이르면 RQ 계수기나 CD 계수기를 증가시켜 하나의 빈 슬롯을 downstream 노드가 사용할 수 있도록 전달한다. DQDB는 각 노드에 하나의 대역폭균형계수기가 있어 서로 다른 우선순위 큐들이 하나의 대역폭균형계수기에 의하여 제어된다. 따라서 높은 우선순위의 세그먼트를 전송하기 위한 정확한 제어를 할 수 없게 된 것이다. 즉 어느 한 우선순위의 세그먼트가 대역폭균형계수만큼 전송되더라도 모든 우선순위 큐의 RQ 계수기나 CD 계수기가 하나씩 증가되기 때문에 우선순위에 따라 선별적으로 제어하는 기능이 없어 높은 우선순위를 갖는 세그먼트의 전송지연특성을 예측하기 어렵게 만든다. 따라서 이러한 문제를 해결하기 위해서는 한 노드내에서 대역폭균형계수기의 동작은 분리시킬 필요가 있다. 즉 서로 다른 우선 순위 큐들이 각각의 서로 다른 대역폭균형계수기에 의하여 제어되어 낮은 우선순위 큐가 높은 우선순위의 큐보다 많은 대역폭을 양보하도록 하는 것이다. 이와같은 동작은 DQDB의 분산 큐 메카니즘은 그대로 유지하면서 수행할 수 있다. 이러한 방안에 의하여 높은 우선순위의 세그먼트를 전송하는 형식은 매우 간단하게 구현할 수 있다. 즉 각 노드에는 우선순위별로 3개의 대역폭균형계수기만을 가지면 된다. 그러나 하나노드에 3개 우선순위에 대하여 각각 대역폭균형계수를 두게 됨으로써

망의 규모가 커질 경우 이러한 계수기에 값을 어떻게 할당하는가 하는 문제가 발생된다. 이것은 FDDI에서 우선순위 기능을 사용할 때 각 우선순위 큐의 TTRT(Target Token Ration Time)를 어떻게 결정할 것인가와 유사한 문제이며 망관리자는 모든 노드의 각 우선 큐에 어떤 대역폭균형계수를 가져야 망이 최대의 성능으로 동작하는지를 알고 있어야 하고 이는 연구과제로 남는다.

4.2.2 WPC(Weighted Priority Control)

Downstream 노드가 메시지를 빨리 전송하기 위해서는 자기 노드보다 upstream 노드가 빈 슬롯을 많이 양보해야 한다. 특히 우선순위가 높은 세그먼트를 전송할 경우 자기보다 upstream 노드중 낮은 우선순위 노드는 될 수 있으면 빨리 빈 슬롯을 통과시키도록 하여야 한다. 결국 우선순위가 높은 세그먼트를 전송하기 위해서는 낮은 우선순위를 갖는 노드의 세그먼트를 지연시키는 희생을 감수하여야 한다. 현재의 DQDB 방식은 높은 우선순위가 낮은 우선순위에 대하여 1 만큼 비중이 높다고 볼 수 있다. 즉 낮은 순위의 노드는 downstream 노드로부터 높은 우선순위 REQ 비트가 세트되어 있음을 감지하면 1 만큼 RQ 계수기나 CD 계수기를 증가시키는 방식으로 볼 수 있다. WPC 방안은 높은 우선순위 세그먼트를 전송하기 위해 REQ 비트에 1 이상의 가중치를 두어 자신보다 우선순위가 낮은 upstream 노드에 보다 많은 빈 슬롯을 통과하도록 요구하는 것이다. 이 방식을 구현하기 위해서는 노드당 하나의 우선순위균형계수기를 둔다. 이 계수기는 대역폭균형계수기의 동작과 매우 비슷하게 동작한다. 우선순위 균형계수기의 값이 0이면 이 모드로 동작하지 않는다. 이 계수기의 값은 0보다 클 때에만 동작한다. 이 방식은 DQDB 기본모드로 동작하면서 높은 우선순위의 요청이 있을때마다 우선순위 균형계수기의 값을 증가시키고, 계수기의 값이 기준치에 도달하면 RQ 계수기나 CD 계수기값의 증가에 가중치를 더하여 낮은 우선순위 세그먼트의 전송이 지연되도록 유도하는 것이다.

4.2.3 HPC(Head Priority Control)

각 노드가 높은 우선순위에 대해 요구하는 방식과 달리 HPC 방안은 헤드 노드에 이

기능을 두는 것이다. 어떤 노드가 높은 우선순위 세그먼트를 전송하기 위해 REQ 비트를 세트하면 그 사실은 헤드에 알려지게 된다. 헤드에서는 REQ 비트가 세트된 슬롯이 도착할 때마다 순방향으로 발생시키는 빈슬롯의 Reserved 필드에 REQ 비트의 값을 세트하는 것이다. 이 빈 슬롯이 순방향으로 진행함에 따라 각 노드는 Reserved 필드를 검사하여 만일 높은 우선순위에 해당하는 값으로 세트되어 있는 경우 상황에 따라 낮은 우선순위의 RQ 계수기나 CD 계수기를 증가시키는 방법이다. 이 방법은 upstream의 낮은 우선순위 노드가 빈 슬롯 사용권을 보다 많이 포기하도록 하는 것이다. 이러한 동작을 지원하기 위해서는 순방향으로 지나가는 Reserved 필드에 대해 동작하는 HPCM(Head Priority Control Modulus)와 HPC(Head Priority Counter)를 둔다. HPM이 0인 경우 이 방식은 동작하지 않는다. 낮은 우선순위 노드는 Reserved 필드에 높은 우선순위 값이 세트된 슬롯이 지나갈 때마다 HPC를 증시킨다. 만일 HPC 값이 HPCM에 이르면 HPC는 0으로 리셋되고 RQ 계수기나 CD 계수기를 증가시킨다.

4.2.4 BPC(Bidirectional Priority Control)

방안 4.2.3의 HPC에서는 높은 우선순위 세그먼트를 전송하고자하는 노드가 역방향으로 지나가는 슬롯에 REQ 비트를 세트함으로써 알려지게 되는데 만일 그 노드가 헤드에서 상당히 떨어져 있는 경우 헤더가 이 사실을 감지하는데 오랜 시간이 경과한 뒤에 알게 된다. 따라서 트래픽이 높아져 망 전체가 과부하 상태에 돌입하였는데도 헤더에서는 이를 감지하는데 시간이 걸림으로써 순방향으로 보내는 빈슬롯의 reserved 필드에 마크하는 동작이 늦어지고 결국 헤더에서 먼 위치에 있는 높은 우선순위의 노드가 세그먼트를 전송하는데 시간이 걸림으로써 전송지연특성이 저하될 수 있는 것이다. 이러한 동작을 개선하기 위해서는 높은 우선순위의 노드는 자신이 세그먼트를 전달할 때마다 양방향으로 낮은 우선순위 노드들에게 빈슬롯을 통과하도록 유도하는 것이다. 즉 역방향에 대한 동작은 WPC 방식으로 제어하고 순방향으로는 순방향으로 지나가는 슬롯의 reserved 필드에 높은 우선순위의 값을 적어 놓음으로써 downstream 노드가 이 슬롯을 받을 때마다 BPC(Bidirectional Priority Control) 계수기의 값을 증가시킨다. BPC 계수기의 값이 BPCM(Bidirectional Priority Control Modulus)에 이를 때마다 RQ 계수기나 CD 계수기를

하나씩 증가시켜 빈 슬롯을 하나 포기하도록 하는 것이다. 이 방식에서는 망 구성시 높은 우선순위 노드들이 노드의 위치에 대한 영향을 다른 방식보다 적게 받으며 우선순위가 낮은 노드들에 대한 지연을 고르게 할 수 있다는 장점이 있다. 그러나 양방향으로 낮은 우선순위의 노드에게 빈 슬롯을 양보하도록 함으로써 낮은 우선순위 노드에게는 다른 방식보다 더 오래 지연될 것으로 예상된다.

DQDB 망에서 공정성 문제를 해결하는 방법으로 4가지 방법을 제시하였다. 이러한 방법은 모두 높은 우선순위 노드가 세그먼트를 전송하거나 요청이 있음을 알게되면 낮은 우선순위에게 빈 슬롯을 통과하도록 유도하는 방식이다. 즉 낮은 우선순위가 사용할 대역폭을 높은 우선순위 노드가 사용할 수 있도록 하여 높은 우선순위 세그먼트를 빨리 전송할 수 있도록 하는 방식이다.

제 5 장 결 론

제 5 장 결 론

본 과제에서는 고속 지역 정보망의 구축을 위한 1차년도 과제로써 현재 IEEE 802.6에서 표준화 작업이 진행 중이며 앞으로 MAN 프로토콜로 각광을 받을 것으로 예상되는 DQDB MAN 프로토콜의 MAC 계층의 구현을 위한 기초 작업을 수행하였다. MAC 계층은 (1) LLC로부터 전달되는 데이터를 슬롯에 삽입될 수 있도록 분리하는 작업과, 분리되어 수신되는 데이터 세그먼트들을 재조립하여 LLC 계층으로 전달하는 기능을 수행하는 MAC Convergence Function, (2) 데이터 세그먼트를 슬롯에 삽입하여 미디어에 접속시키는 기능을 수행하는 QA Function, (3) 노드 또는 링크에 결함이 발생할 경우 망을 재구성하고, 데이터 세그먼트의 재조립하는데 필요한 MID 페이지를 할당하는 기능을 수행하는 Common Function 등으로 나누어질 수 있다. 본 과제에서는 DQDB MAC 계층구현을 위한 기초 작업으로 각 블록의 세부 기능 블록도를 제시하였다. 또한 DQDB 망의 구현 시 이의 서비스 수행 능력을 예측하기 위하여 DQDB 망의 성능을 시뮬레이션 모델을 통하여 분석하였으며, 이를 통하여 망의 공정성 및 우선순위 문제를 완화할 수 있는 방안을 모색하였다.

부 록

Appendix A. 고속 동기 접속 장치 개발

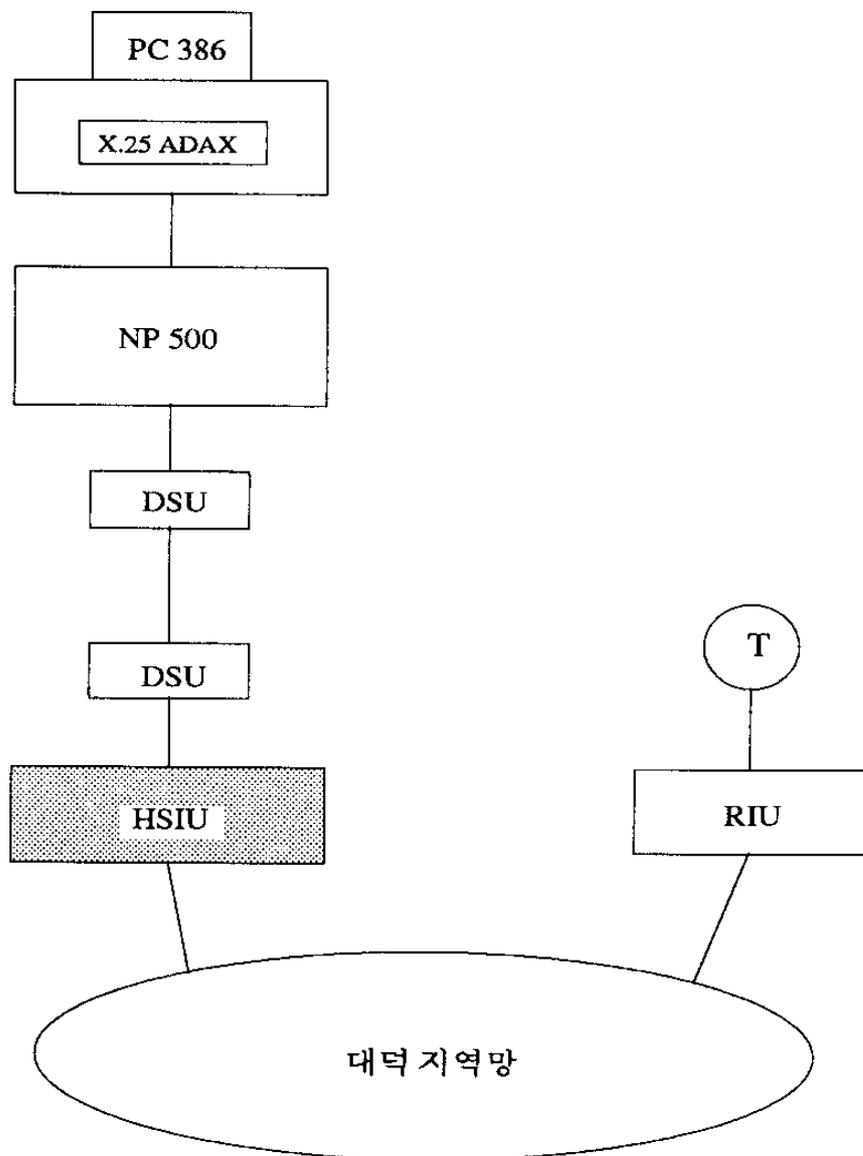
A.1 개요

대덕 연구 단지에 시범적으로 구축된 대덕 지역망은 지역내의 자원 공유를 위해 ETRI와 과기대, 표준연구소 사이에 설치되어 시범 운용중이며, 원거리의 다른 자원을 access하기 위해 DacomNet과 접속하는 Gateway가 구현되어 있다. 여기서 원거리의 traffic이 많은 특정 기관 집단 사이에서는 DacomNet을 통하는 것보다는 상대적으로 고속인 Trunk를 두어 정보 교환을 원활히 하는 것이 바람직하다. 지역망내에는 프로토콜 처리 능력을 가지는 여러 가지 Packet DTE(PDTE)들이 존재하며, Data Link 및 Network 계층 일부의 표준안으로 쓰이는 X.25 프로토콜 처리 능력을 가지는 X.25 host(혹은 X.25 PDTE)들이 점차 보급되고 있는 실정이다. 따라서 본 위탁 과제에서는 대덕 지역망과 SERI간에 고속의 동기 접속을 담당하는 High-Speed Interface Unit(HSIU)를 개발하는 것을 목적으로 하며, 본 Appendix A에서는 이에 대해 기술한다.

고속 동기 접속 장치인 HSIU는 Token Ring 타입의 대덕 지역망과 접속할 수 있도록 하는 RIM(Ring Interface Module), X.25 protocol과 LLC간의 프로토콜 conversion을 담당하는 MCPM(Main Control Processor Module), X.25와의 고속 동기 접속을 담당하는 HIOM(High-Speed Input Output Module) 및 LDB(Line Drive Board) 등으로 구성된다. 고속 처리를 위해 지역망의 MAC과 LLC 프로토콜은 전문 프로세서에서 처리하며 이때 LLC 기능은 LLC firmware를 이용하였다. 이들 각 module은 전체적으로 IEEE 796 Multibus에 의해 연결된다.

대덕 지역망은 가입자 node를 Optical Fiber에 접속시켜 주는 FIM(Fiber Interface Module), Ring 형태의 LAN과 user 및 host를 접속시켜 주는 RIU(Ring Interface Unit), DacomNet을 통한 원거리의 자원 접속을 지원하는 Gateway 등으로 구성되어 있다. 여기서 대덕 지역망과 SERI간에 고속으로 정보 교환을 하기 위해 고속 동기 접속 장치가 필요하게 된다. 이에 따라 본 연구에서는 대덕 지역망과 X.25 PDTE간을 고속 동기 접속시켜주는 HSIU를 개발하였다. 연구의 목적상 X.25 PDTE와 HSIU를 연결하여 접속 테스트를 실시해야 하나 현재는 접속 가능한 X.25 PDTE가 없으므로 이를 대신하여 X.25 프로토콜로 고속 동기 접속을 할 수 있는 NP(Node Processor) 500으로 테스트를 수행하였다. 이 경우의 접속 구성도는 그림A.1과 같이 된다. 즉, NP 500과

대덕 지역망의 HSIU는 56kbps의 X.25 프로토콜로 동기 접속된다. 테스트를 위하여 Adax X.25 카드가 내장된 PC-386을 NP에 접속하고 HSIU를 거쳐 지역망에 물려있는 RIU의 한 port를 call하거나 그 반대의 방향으로 connection을 성립시킨다. NP 500과 HSIU가 원거리에 위치하는 경우에는 Data Service Unit(DSU)를 사용하여 거리 제한을 극복할 수 있다. DSU와 HSIU는 V.35 규격으로 접속시킬 수 있고 이때 clock은 DSU가 제공한다.



그림A.1 NP 500과 HSIU의 접속도

A.2 고속 동기 접속 장치의 Hardware

HSIU의 H/W는 그림A.2와 같이 크게 Ring과의 접속을 담당하는 RIM과 프로토콜 conversion을 담당하는 MCPM, I/O 접속 S/W를 처리하는 HIOM 및 고속 접속용 LDB module로 구성된다. 각 module은 common system bus인 IEEE 796 Multibus에 의해 연결된다.

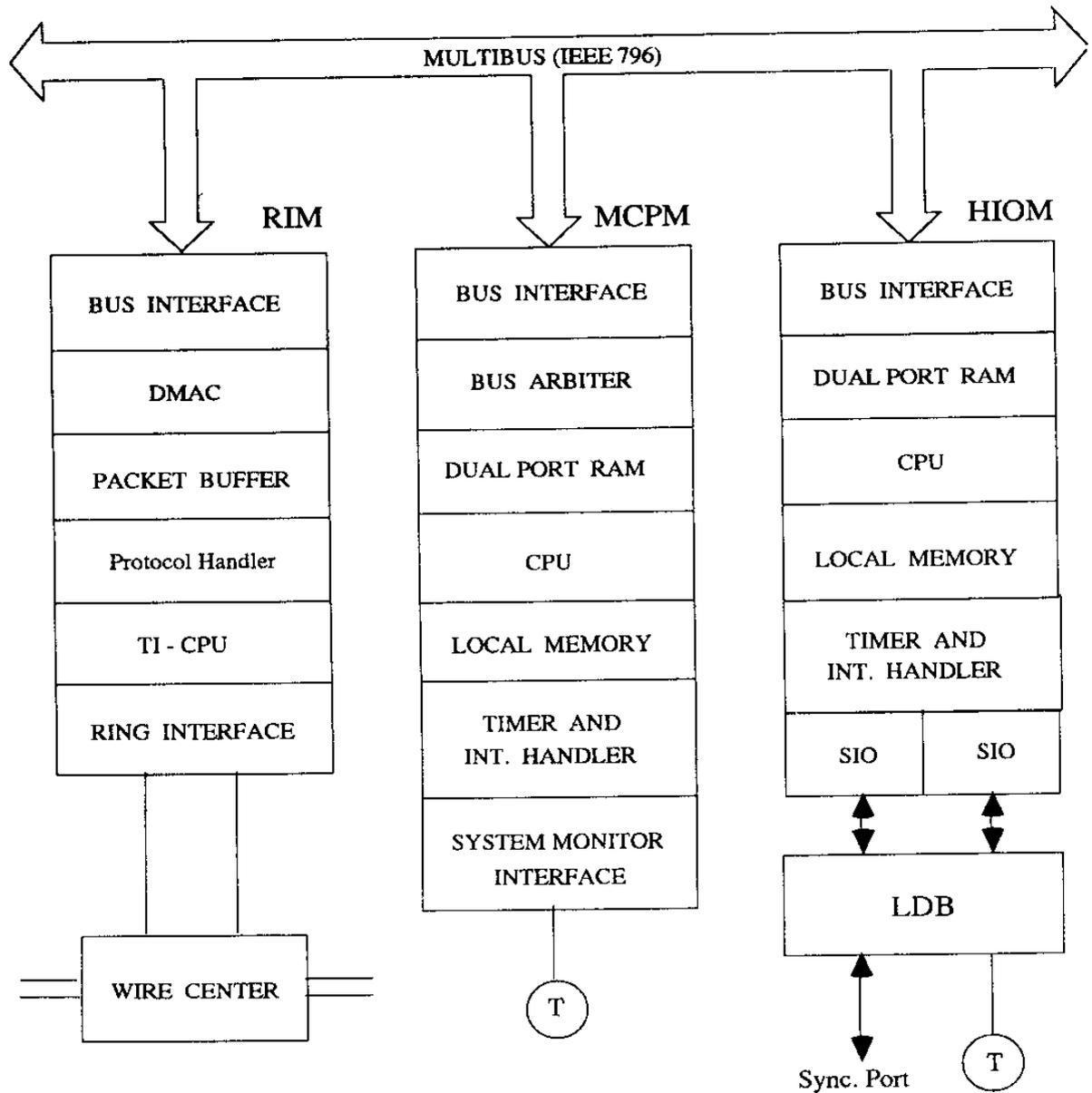
기본적으로 고속 동기 접속 장치는 X.25 packet 망과 Token ring 타입의 지역망간에 protocol conversion 기능을 수행한다. 이에 따라 RIM은 TMS-380 chipset에 의해 IEEE 802.5 MAC protocol인 Token Ring protocol을 구현하며, MCPM과 HIOM은 16bit MC68000을 CPU로 사용하며, 고속 처리를 위해 각각 Clock은 10MHz를 사용한다.

MCPM은 고속 접속 장치 시스템의 중추 master로서 System bus의 arbitration을 담당하며, X.25 PDTE에서 HIOM으로 입력된 SERI NP의 X.25 packet과 지역망으로부터 RIM에 입력된 LLC frame을 protocol conversion을 위해 필요한 processing을 담당한다.

HIOM은 X.25 PDTE와의 동기 접속을 담당하는 module로서 X.25 동기 port로부터 들어오는 data를 처리한 후 Common Memory에 쌓아놓고 이를 MCPM에게 알린다. 또한 지역망으로부터 X.25 PDTE로 데이터가 전송되는 경우는 MCPM에 의해 Common Memory에 쓰면 HIOM은 X.25 data packet을 받아서 동기 port로 내보내는 기능을 담당한다.

LDB는 고속 접속 line drive를 위해 준비된 것으로 접속될 동기 port의 전기적인 특성에 따라 선택적으로 사용할 수 있도록 되어 있다. 저속의 경우 RS-232C에 준하는 신호를 처리하는 것과 고속의 경우 56kbps로 V.35에 준하는 신호를 처리하는 module로 대별된다.

고속 동기 접속 장치의 각 module은 해당 protocol 기능을 수행하여 처리된 데이터를 module간의 통신으로 교환한다. MCPM과 HIOM간의 통신은 MCPM이 master가 되어 HIOM에 있는 dual port Common Memory와 IMC register를 이용하여 수행된다. 데이터가 HIOM에서 MCPM으로 이동할 때에는 우선 HIOM이 IMC register가 clear되어 있는가를 확인하고 clear인 경우 데이터를 CM에 쌓는다. 그리고 MCPM에 interrupt request하면 MCPM이 Multibus를 거쳐 CM에 있는 데이터를 fetch해 가고, HIOM에 있는 IMCregister를 clear한다. 즉, IMC register를 semaphore flag로 두어 mutual exclusive를 보장한다.



그림A.2 HSIU의 Hardware 계통도

MCPM에서 RIM으로 데이터를 이동시키려면 MCPM이 System Command Block(SCB)에 Tx 명령과 Transmit List의 시작 번지를 쓰고 SIF에 interrupt를 건다. 그러면 SIF는 DMA로 Transmit List를 읽어가고 순차적으로 파악된 전송 데이터를 DMA해 간다. RIM은 읽어들이는 데이터를 적당히 처리한 후 status를 알리기 위해 MCPM에 있는 System Status Block(SSB)에 처리 상태를 DMA로 전하고 MCPM에 interrupt를 건다. 그러면 MCPM은 SSB를 검사하여 명령이 처리된 상태를 파악하게 된다. RIM에서 MCPM으로 데이터를 이동할 때도 비슷한 절차를 거친다.

A.2 고속 동기 접속 장치의 Hardware

HSIU의 H/W는 그림A.2와 같이 크게 Ring과의 접속을 담당하는 RIM과 프로토콜 conversion을 담당하는 MCPM, I/O 접속 S/W를 처리하는 HIOM 및 고속 접속용 LDB module로 구성된다. 각 module은 common system bus인 IEEE 796 Multibus에 의해 연결된다.

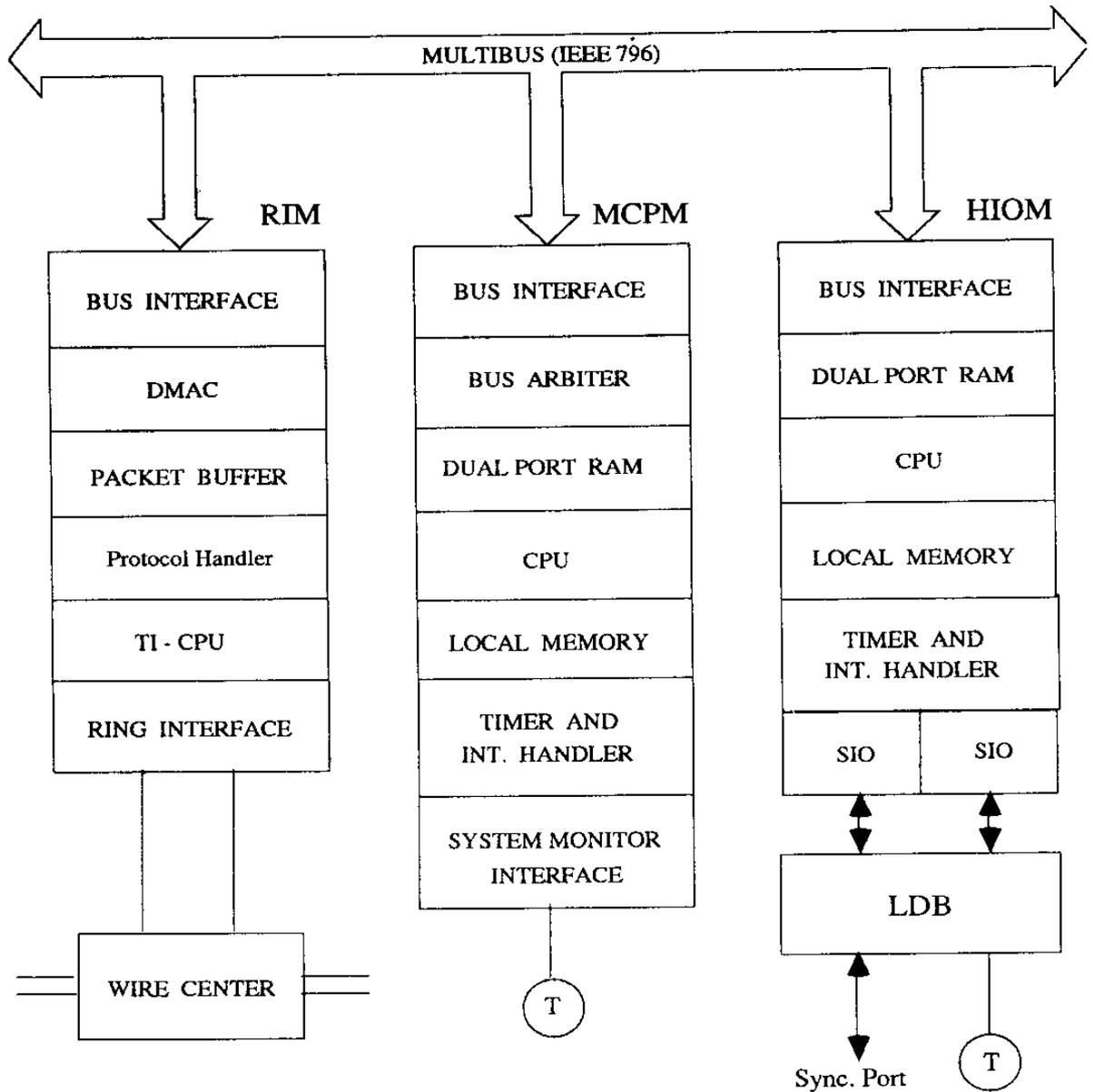
기본적으로 고속 동기 접속 장치는 X.25 packet 망과 Token ring 타입의 지역망간에 protocol conversion 기능을 수행한다. 이에 따라 RIM은 TMS-380 chipset에 의해 IEEE 802.5 MAC protocol인 Token Ring protocol을 구현하며, MCPM과 HIOM은 16bit MC68000을 CPU로 사용하며, 고속 처리를 위해 각각 Clock은 10MHz를 사용한다.

MCPM은 고속 접속 장치 시스템의 중추 master로서 System bus의 arbitration을 담당하며, X.25 PDTE에서 HIOM으로 입력된 SERI NP의 X.25 packet과 지역망으로부터 RIM에 입력된 LLC frame을 protocol conversion을 위해 필요한 processing을 담당한다.

HIOM은 X.25 PDTE와의 동기 접속을 담당하는 module로서 X.25 동기 port로부터 들어오는 data를 처리한 후 Common Memory에 쌓아놓고 이를 MCPM에게 알린다. 또한 지역망으로부터 X.25 PDTE로 데이터가 전송되는 경우는 MCPM에 의해 Common Memory에 쓰면 HIOM은 X.25 data packet을 받아서 동기 port로 내보내는 기능을 담당한다.

LDB는 고속 접속 line drive를 위해 준비된 것으로 접속될 동기 port의 전기적인 특성에 따라 선택적으로 사용할 수 있도록 되어 있다. 저속의 경우 RS-232C에 준하는 신호를 처리하는 것과 고속의 경우 56kbps로 V.35에 준하는 신호를 처리하는 module로 대별된다.

고속 동기 접속 장치의 각 module은 해당 protocol 기능을 수행하여 처리된 데이터를 module간의 통신으로 교환한다. MCPM과 HIOM간의 통신은 MCPM이 master가 되어 HIOM에 있는 dual port Common Memory와 IMC register를 이용하여 수행된다. 데이터가 HIOM에서 MCPM으로 이동할 때에는 우선 HIOM이 IMC register가 clear되어 있는가를 확인하고 clear인 경우 데이터를 CM에 쌓는다. 그리고 MCPM에 interrupt request하면 MCPM이 Multibus를 거쳐 CM에 있는 데이터를 fetch해 가고, HIOM에 있는 IMCregister를 clear한다. 즉, IMC register를 semaphore flag로 두어 mutual exclusive를 보장한다.



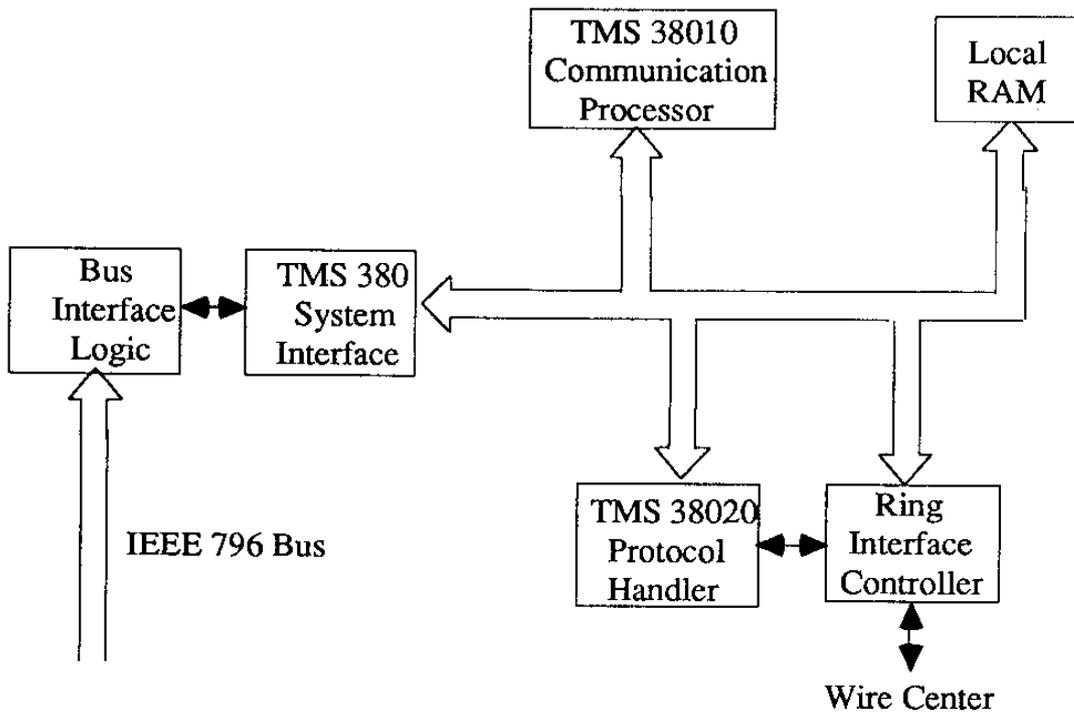
그림A.2 HSIU의 Hardware 계통도

MCPM에서 RIM으로 데이터를 이동시키려면 MCPM이 System Command Block(SCB)에 Tx 명령과 Transmit List의 시작 번지를 쓰고 SIF에 interrupt를 건다. 그러면 SIF는 DMA로 Transmit List를 읽어가고 순차적으로 파악된 전송 데이터를 DMA해 간다. RIM은 읽어들이는 데이터를 적당히 처리한 후 status를 알리기 위해 MCPM에 있는 System Status Block(SSB)에 처리 상태를 DMA로 전하고 MCPM에 interrupt를 건다. 그러면 MCPM은 SSB를 검사하여 명령이 처리된 상태를 파악하게 된다. RIM에서 MCPM으로 데이터를 이동할 때도 비슷한 절차를 거친다.

구현된 장치는 고속 접속을 목적으로 하므로 각 module간의 System Bus Arbitration logic이 신속하고 신뢰성이 높아야 한다. 이를 위해 각 module의 bus arbitration logic을 견고한 TTL logic을 이용하여 안정화되도록 하였다. MCPM이나 RIM 중 어느 하나가 먼저 system bus를 사용하고 있을 때에는 다른 Module의 bus request는 기다리게 되며, 두 Module이 동시에 system bus를 요구한 경우에는 deadlock 방지를 위해 RIM에 우선권을 준다.

A.2.1 RIM

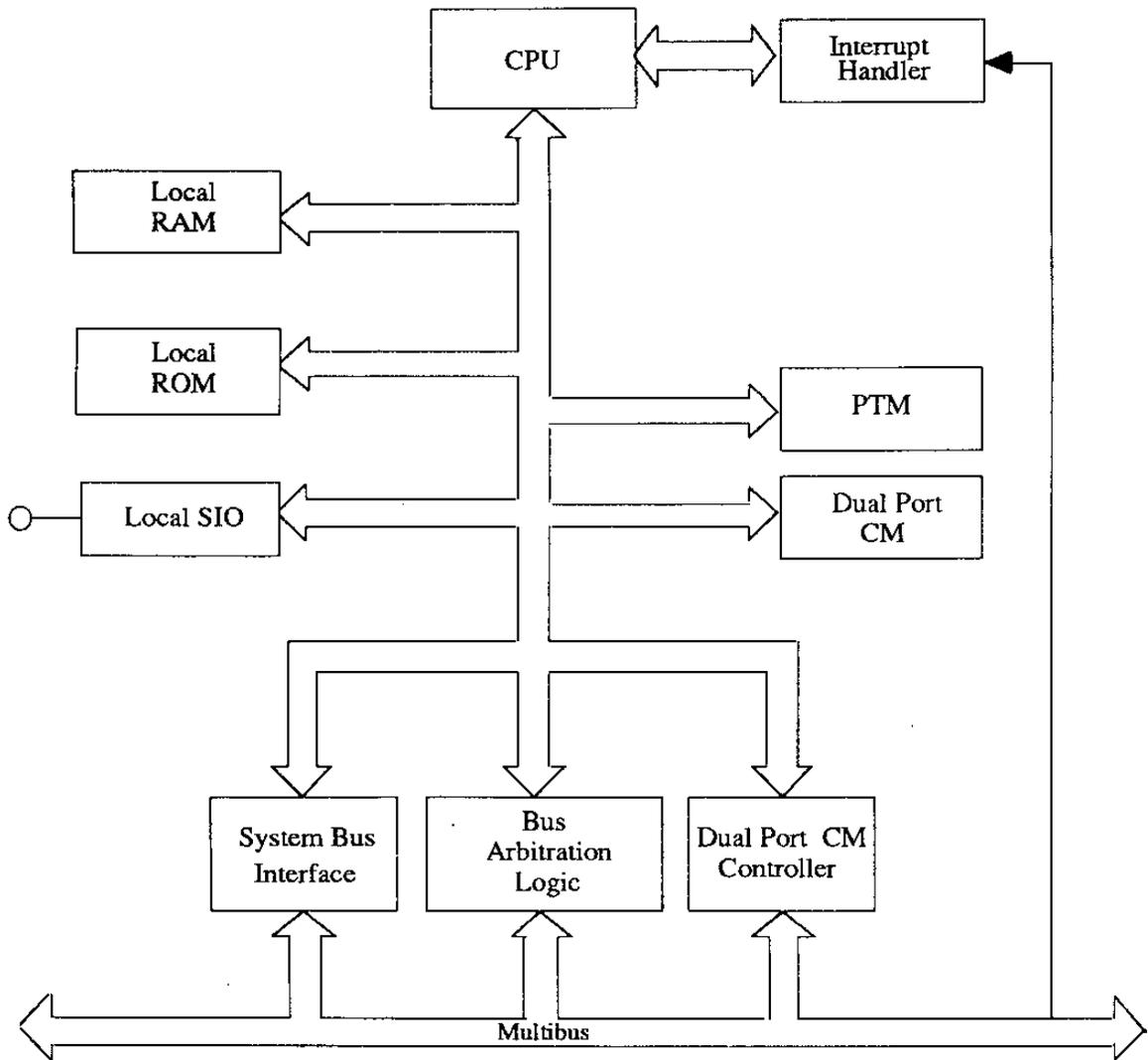
RIM은 TMS380 chipset와 local memory, system bus 접속 및 system bus arbitration을 위한 logic과 Ring drive analog assembly part로 구성되며, 그 구조는 그림A.3과 같다. MCPM과의 DMA 및 DIO는 TMS38030 System Interface(SIF)를 통해 이루어지며, TMS 38020은 Token ring protocol 기능을 담당하고 TMS38010은 RIM 내부의 CPU로 동작한다. TMS38051과 TMS38052는 Ring으로 신호를 송수신하기 위한 Transceiver를 이룬다. 사용되는 신호는 Differential Manchester coding 방식을 따르므로 DC offset이 없고 쉽게 동기를 맞출 수 있는 잇점이 있다.



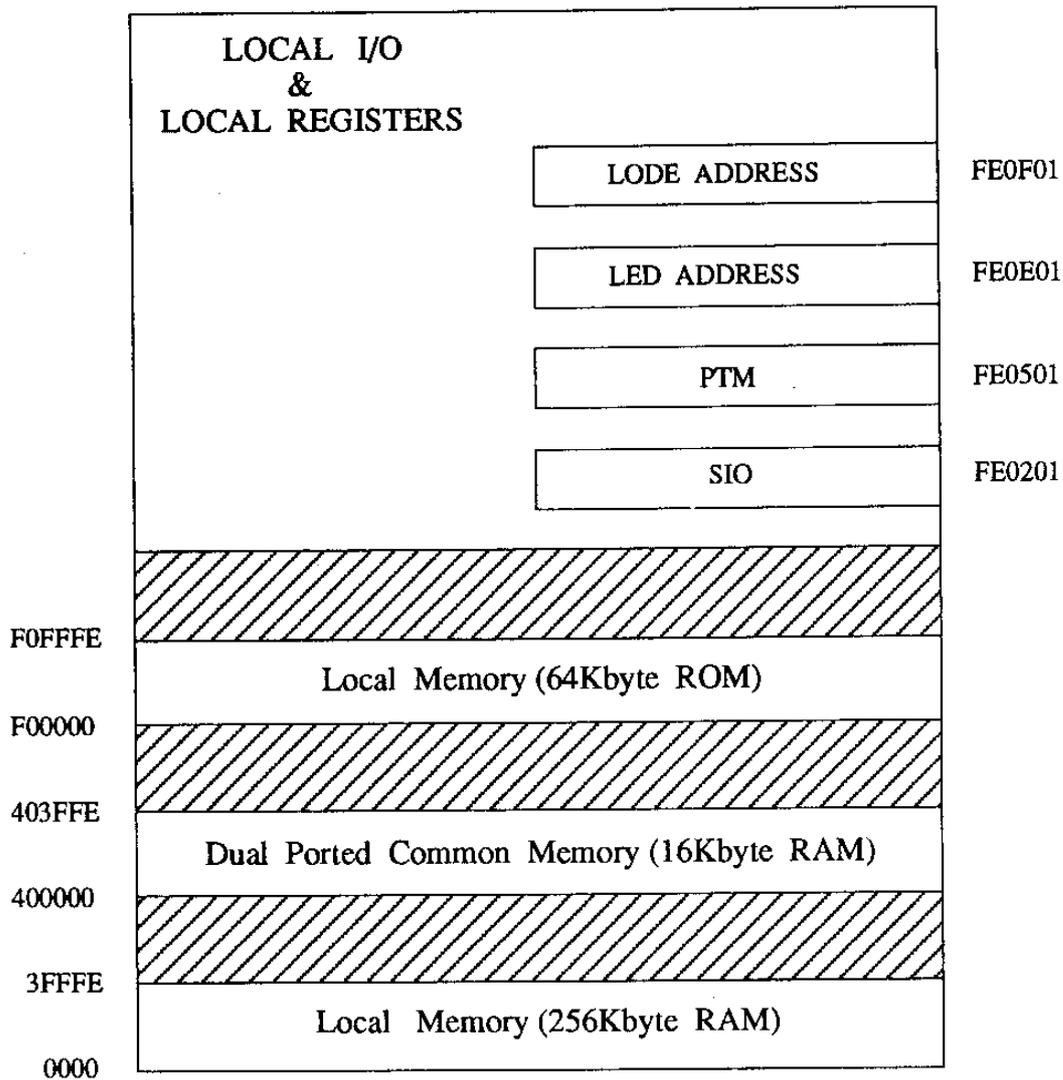
그림A.3 RIM의 구조

A.2.2 MCPM

MCPM은 기본적으로 CPU board 이외에 다른 module과의 통신을 위한 System Bus interface 및 arbitration logic, PTM과 SIO로 구성된다. MCPM은 HIOM의 dual port CM을 통하여 HIOM과 데이터를 주고 받는데 이때 MCPM이 master가 된다. RIM과는 DIO 및 DMA 방식으로 데이터를 교환하며, RIM의 SIF가 DMAC로 주된 기능을 수행한다. MCPM의 구성은 그림A.4와 같으며, Memory Map은 그림 A.5와 같다.



그림A.4 MCPM의 구성

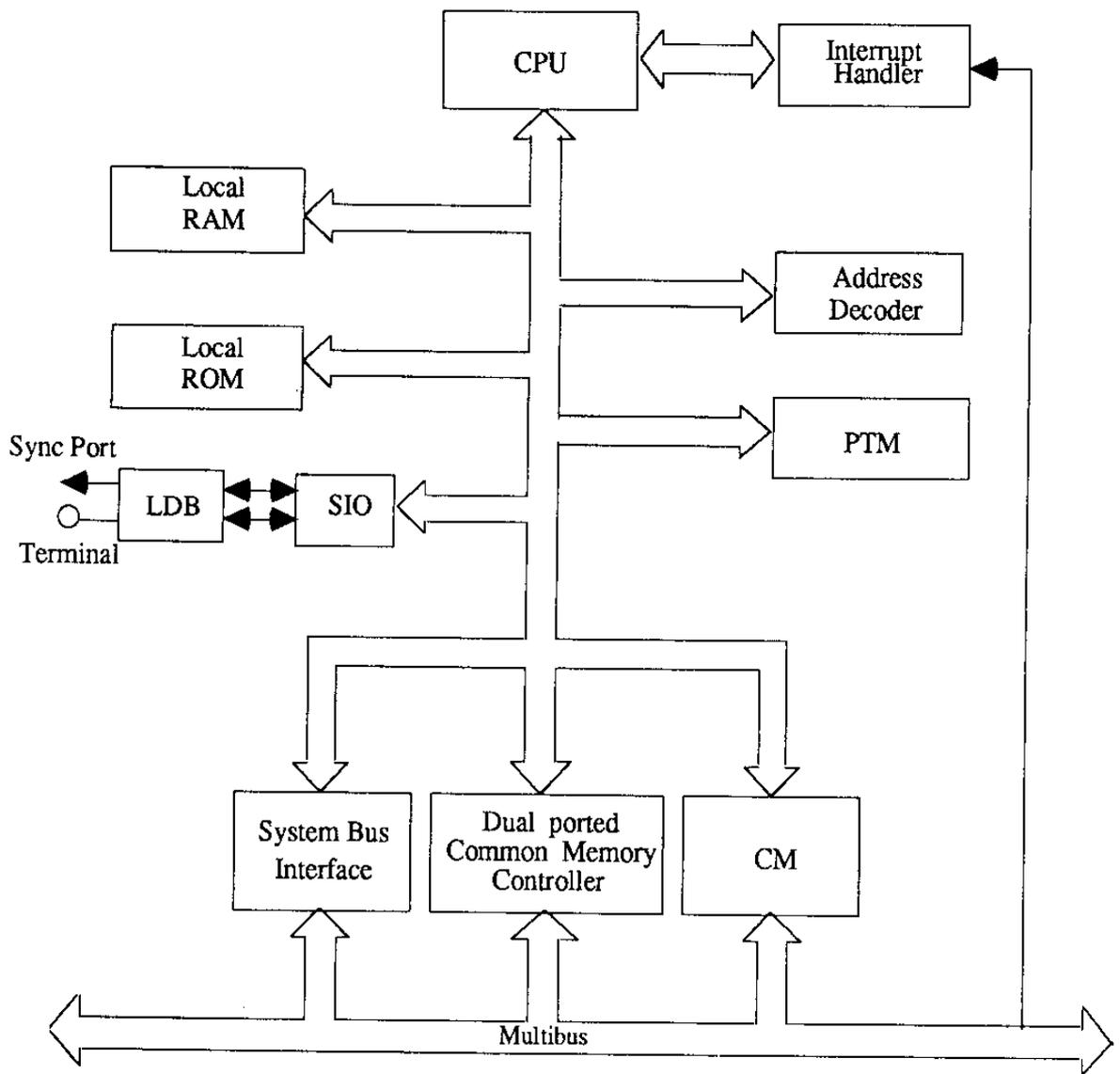


그림A.5 MCPM의 Memory Map

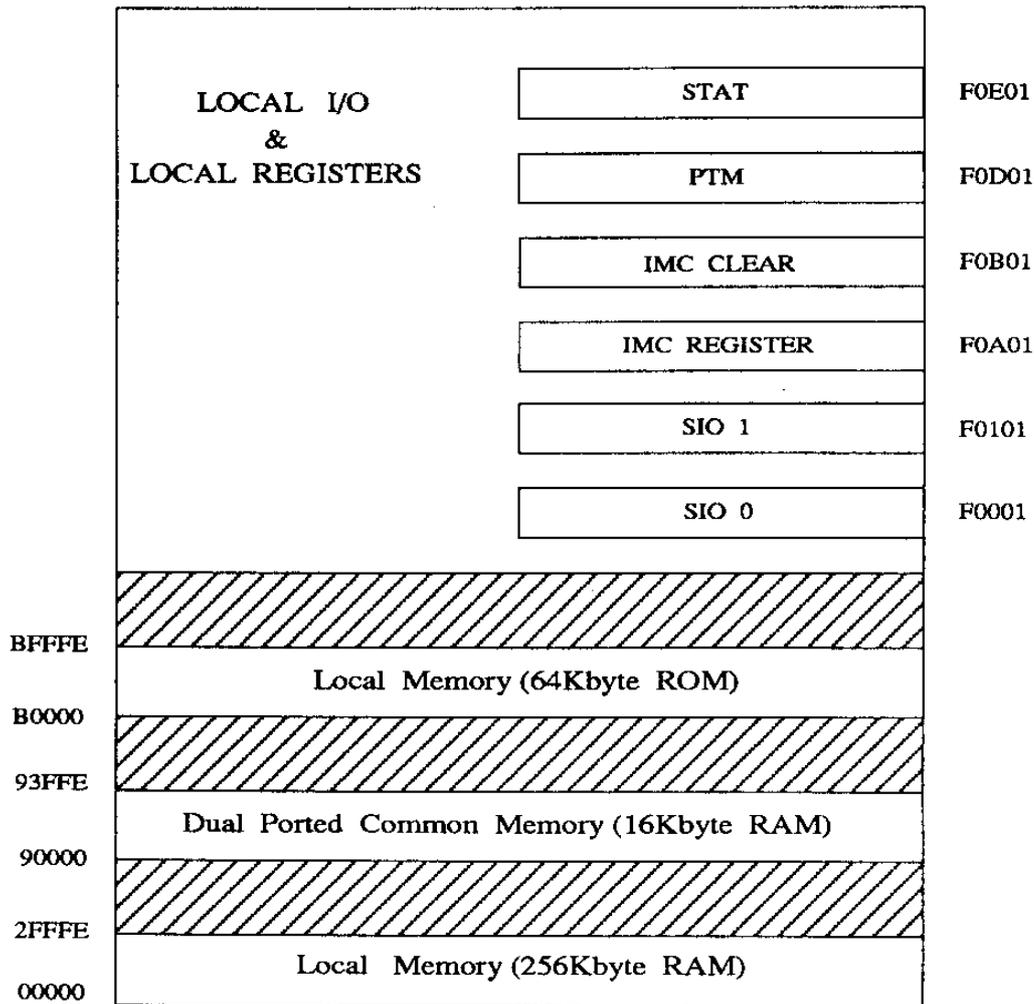
A.2.3 HIOM과 LDB

HIOM은 X.25망과의 접속을 위한 module로서 그 구성은 그림A.6의 기능별 구조와 같이 기본적으로 CPU와 주변 회로, Local Memory, PTM 및 MCPM과의 통신을 위한 dual port Common Memory로 구성된다. HIOM은 접속되는 망의 전기적 특성에 따라 알맞은 Line Drive Board(LDB)를 선택적으로 사용할 수 있게 되어 있다. 그림 A.7은 HIOM의 Memory Map이다. LDB는 그림 A.6과 같이 line drive chip과 DTE/DCE mode

전환 jumper로 구성되며, 저속의 RS-232C 담당 module과 고속의 V.35 접속 담당용이 있다. RS-232C용 LDB는 CCITT의 V.28 전기적 특성을 가지며, connector와 pin의 배치는 25 pin의 ISO 2110을 따른다. V.35용 LDB는 CCITT의 V.35 전기적 특성으로 connector와 pin의 배치는 34 pin의 ISO 2593을 따르게 된다. DTE/DCE간을 V.35로 접속할 때 Tx, Rx 데이터 및 clock에 관계되는 신호는 Balanced mode로, 그 외의 circuit 신호는 Unbalanced mode로 동작한다.



그림A.6 HIOM과 LDB의 계통도



그림A.7 HIOM의 Memory Map

A.2.4 V.35 접속

일반적으로 컴퓨터망에서 DTE와 DCE를 접속시키기 위해서는 Communication Interface(CI)가 필요하다. CI는 유럽의 CCITT나 미국쪽의 EIA에서 표준안을 제정해 놓고 있으며, 대표적인 것들은 표 A.1과 같다. 가장 흔한 접속 방안인 RS-232C는 V.24의 interchange 회로 규정과 V.28의 전기적 신호 특성, ISO 2110의 25핀 Connector와 핀 규정을 따른다. RS-232C는 DTE와 DCE간에 unbalanced mode로 15m 이내에서 최대 20kbps의 전송 속도까지를 규정하고 있다. 따라서 장거리에서 고속의 접속을 요하는 경우에는 신뢰성을 위해 다른 접속 규정을 선택해야 하며, 대표적으로 생각할 수 있는 접속 방법은 EIA RS-449A와 V.35이다.

EIA RS-449A는 ISO 4902의 37핀 및 9핀의 DTE/DCE connector와 핀 규정을 사용하며, 전기적 규정으로 unbalanced 모드인 경우는 RS-423A로 balanced 모드를 위해서는 RS-422A를 두고 있다. 이는 synchronous나 asynchronous로 동작한다.

V.35는 48kbps로 동작하는 아날로그 wide-band 모뎀과의 synchronous 접속을 규정한다. V.35 접속은 unbalanced 모드의 V.24/V.28 신호와 balanced 모드의 V.35 신호를 혼합하여 사용하며, 최대 케이블 길이에 대한 제한은 RS-232C와 같은 15m이다. Balanced 모드인 경우 원 신호와 반전된 두 신호를 같이 전송하며 전압 레벨은 각각 +0.5Volt이다. DTE/DCE 접속 connector는 MRAC 34핀 connector를 사용한다. Balanced 모드를 사용하는 신호는 전송 및 수신 데이터, 송수신 clock 관계 signal이며, 이외는 unbalanced 모드 신호를 사용한다.

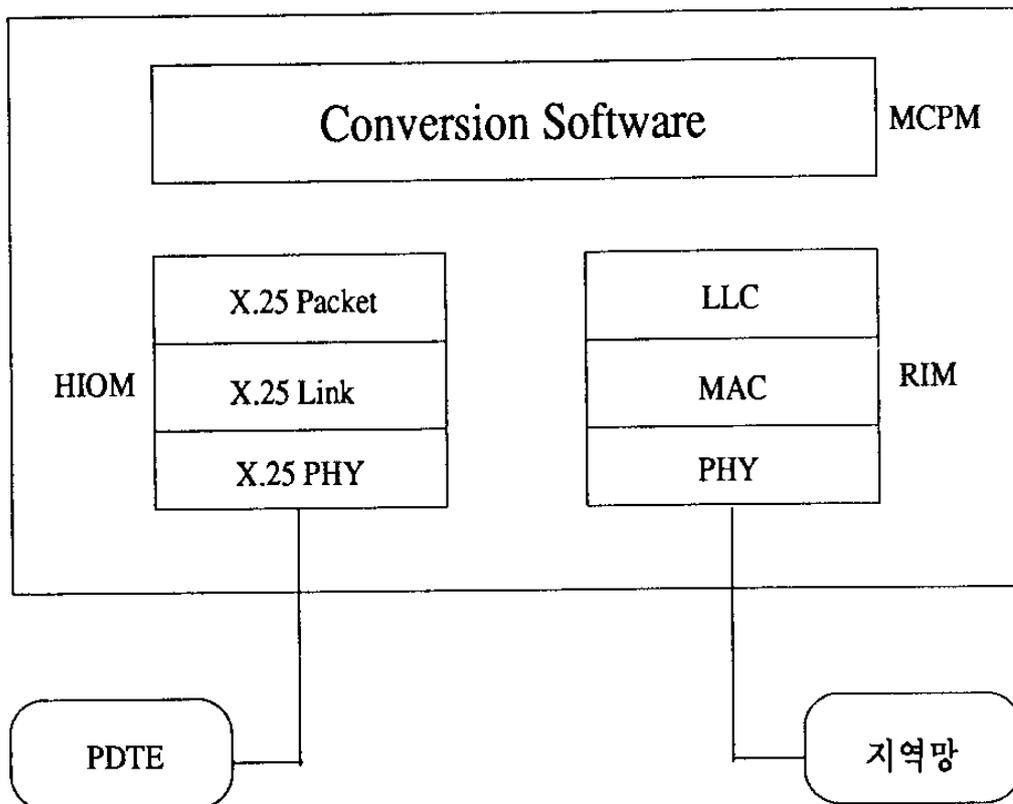
Interface Specification		Sync./ Asynch.	Balanced/ Unbalanced	Max. bit/s
European CCITT	American EIA			
V.24/28	= RS-232-C	A	U	20,000
V.25	= RS-366-A	A	U	20,000
V.25 bis		A + S	U	20,000
	RS-449-A	A + S	U/B	2,000,000
X.26(V.10)	= RS-423-A	A + S	U	100,000
X.27(V.11)	= RS-422-A	A + S	B	10,000,000
X.21 (X.26/27)		S	U/B	48,000
X.21bis(V.28)		S	U	9,600
X.21bis(X.26)		S	U	9,600
X.21bis(X.26/27))		S	U/B	48,000
X.21bis(V.35)		S	U+B	48,000
V.35		S	U+B	168,000
Current Loop(20/60mA)		A	B	20,000
Telegraph(80-0-80)		A	U	110
Telex(80-0-80 or SCVF)		A	U	50
(SCVF maximum)		A	U	300

테이블A.1 국제적인 CI 표준안

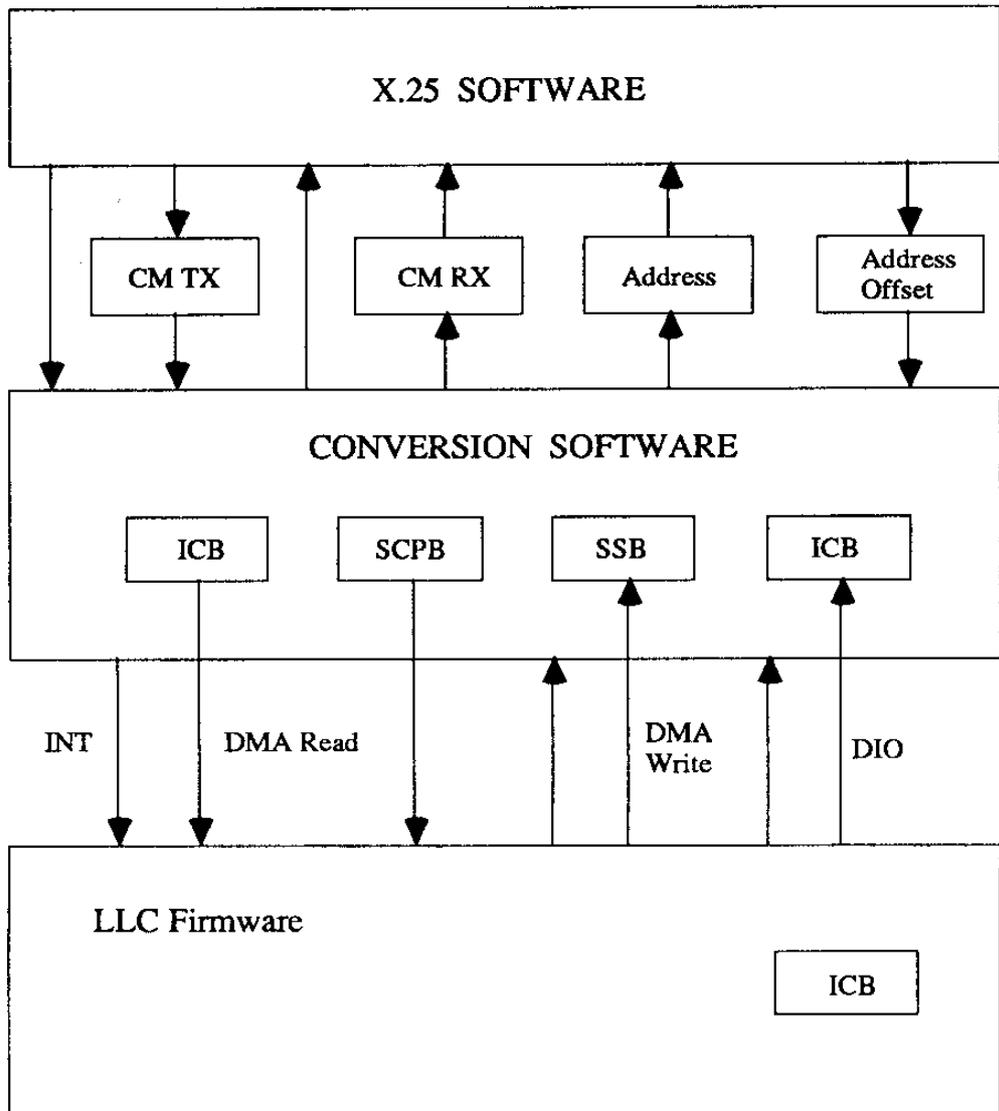
A.3 고속 동기 접속 장치의 Software

A.3.1 전체 Software 구조

고속 동기 접속 장치는 X.25 프로토콜을 사용하는 PDTE와 IEEE 802.2 LLC 프로토콜을 사용하는 RIU 사이에서 통신이 가능하도록 해 준다. 이때 PDTE와 RIU가 사용하는 프로토콜이 서로 다르므로 고속 동기 접속 장치는 프로토콜 conversion 기능이 요구된다. 이를 위하여 HIOM module에 X.25 S/W를 구현하고 MCPM module에 Conversion S/W, RIM module에 LLC Firmware를 그림 A.8과 같이 구현하였다. 고속 동기 접속 장치의 각 module 사이의 interface는 그림 A.9와 같다.



그림A.8 고속 동기 접속 장치의 Software 구조



그림A.9 고속 동기 접속 장치 Module간의 Interface

A.3.2 LLC S/W

RIM 모듈의 LLC는 ROM에서 제공하는 firmware로 구현하였고 Conversion S/W에서는 LLC firmware에서 제공하는 System Command를 이용하여 X.25 S/W와 토큰링 지역망으로의 데이터 전송 및 수신을 효율적으로 수행한다. 현재 IEEE 802 표준안에서 규정하는 서비스는 Type 1 (connectionless LLC), Type 2 (connection-oriented LLC) 및 Type 3 (connectionless acknowledged LLC)가 있는데 RIM의 LLC firmware는 Type 1과 Type 2 서비스를 제공한다.

A.3.2.2 Conversion S/W와의 통신용 Register

LLC firmware는 Conversion S/W와의 통신을 위해서 4개의 register를 이용한다.

1. System Interface(SIF) Interrupt Register

SIF interrupt register는 Conversion S/W가 LLC firmware에 INT를 요구하거나 또는 LLC firmware가 Conversion S/W에게 INT를 요구할 때 사용한다.

2. SIF Address Register

SIF address register는 RIM의 RAM 영역에 데이터를 read하거나 write하려는 address를 가리킨다.

3. Data Register

RIM의 RAM에 데이터를 read하거나 write하는 경우 Data register를 이용한다.

4. Data Register with Autoincrement

RIM의 RAM에 데이터를 read하거나 write한 후 SIF address register 내용이 자동적으로 증가된다.

A.3.2.3 Interrupt Codes

RIM의 LLC firmware가 Conversion S/W에 interrupt를 요구할 때에는 SIF interrupt register에 interrupt code를 write한다. 그러면 Conversion S/W는 SIF interrupt register를 읽고 그에 대응하는 INT 서비스를 수행한다. Interrupt code는 다음과 같다.

1. Ring Status INT

Ring에서 신호가 loss되거나 에러가 규정 이상 발생했을 때 등 ring의 status가 변하는 경우 사용된다.

2. Receive Pending INT

LLC firmware가 지역망으로부터 데이터 프레임을 받으면 Conversion S/W에게 Receive pending INT를 요구한다.

3. Receive status INT

LLC firmware가 Conversion S/W에게 Receive pending INT를 요구하면 Conversion S/W는 데이터 프레임을 받을 버퍼를 할당하고 SIF interrupt register를 통하여 Receive valid INT를 보낸다. 이때 LLC firmware는 데이터 프레임을 MCPM에 준비된 버퍼에 DMA write한 후 Receive status INT를 요구한다.

4. Transmit status INT

Conversion S/W가 Transmit command를 LLC firmware로 보내면 LLC는 이 command에 대한 수행을 한 후 그 결과를 SSB에 쓰고 다시 Transmit command를 수행할 수 있음을 Conversion S/W에게 Transmit status INT로 알린다.

5. LLC status INT

Logical Link의 status가 변하게 되면(e.g., SABME, DISC 프레임을 송신 혹은 수신하면) 그 결과를 ICB(Interface Control Block)에 기록하고 Conversion S/W에게 LLC status INT로 알려준다.

6. Command status INT

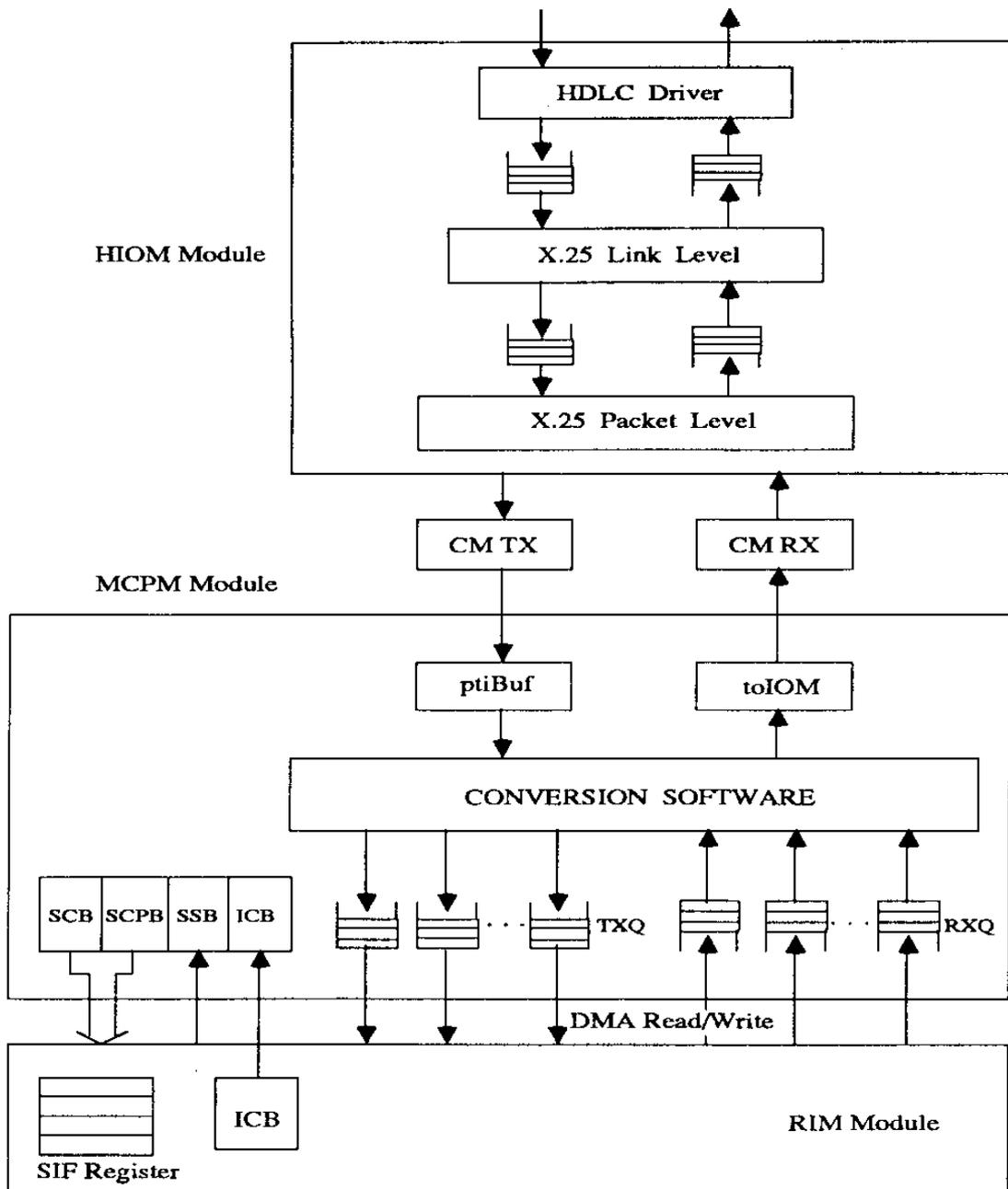
Conversion S/W가 내린 시스템 command를 LLC firmware가 수행한 후 그 결과를 SSB에 write하고 Command status INT로 시스템 command를 수행했음을 Conversion S/W에게 알린다.

7. Adapter check INT

RIM의 H/W나 S/W에 회복할 수 없는 에러가 발생하면 이것을 Adapter check INT로 Conversion S/W에게 알린다.

A.3.3 Conversion Software

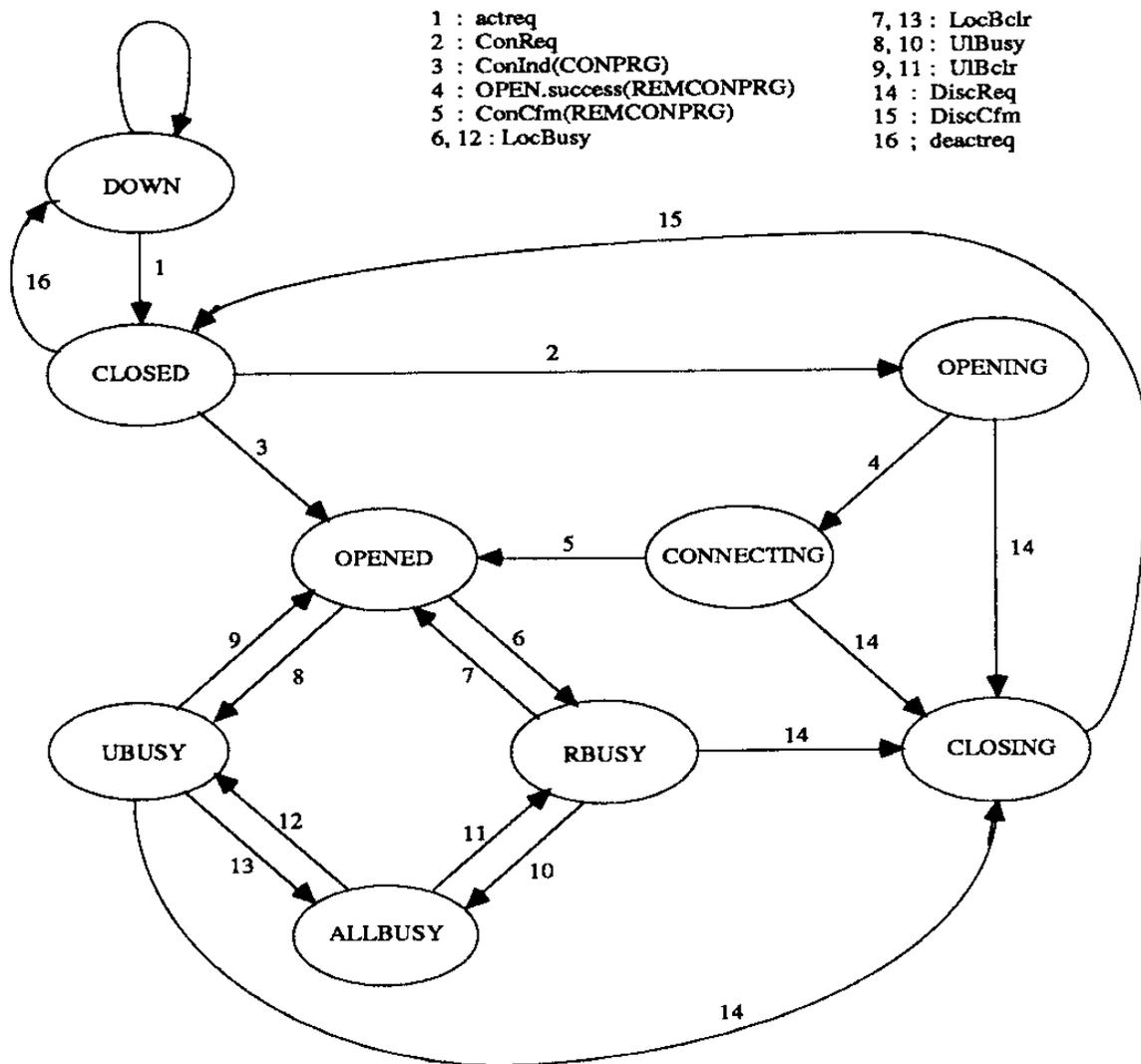
고속 동기 접속 장치에서는 X.25 protocol과 LLC protocol 사이의 conversion 기능이 요구된다. 이를 위하여 HIOM module에 X.25 S/W를 구현하고 MCPM module에는 Conversion S/W와 LLC S/W를 구현하였다. 고속 동기 접속 장치의 main 부분인 Conversion S/W의 queue 구조는 그림 A.11과 같다.



그림A.11 Conversion S/W의 Queue 구조

HIOM의 X.25 S/W가 trunk로부터 Call Request packet을 받으면 call processing을 한 후 Call Request 프리미티브를 Common Memory(CM) 400000(MCPM에서 본 주소) 번지에 쌓은 후 MCPM에 interrupt를 request한다.

Conversion S/W는 Call Request 프리미티브에서 원거리 LAN의 RIU 주소와 destination port(DSAP)를 구하여 Connect Request 프레임(SABME)을 보낸다. 그후 RIM의 LLC firmware가 Connect Confirm 프레임(UA)을 받아서 Conversion S/W에게 알리면 Conversion S/W가 Call Accepted 프리미티브를 X.25에 알려 call이 성립된다. 일단 call이 성립되면 데이터 전송 상태가 되고 X.25에서 수신된 데이터 프레임들은 pibuf 버퍼에 쌓고 packet-to-frame 변환을 한 후 SAP별로 마련된 TXQ queue에 쌓는다.



그림A.12 SAP State Transition Diagram

그리고 원거리 RIU에서 보내진 데이터 프레임들은 SAP별로 마련된 RXQ queue에 쌓고 frame-to-packet 변환을 한 후 toIOM 버퍼에 쌓은 후 CM 40200(MCPM에서 본 주소) 번지를 통하여 X.25 S/W에 전달된다. 즉, Conversion S/W는 TXQ queue와 RXQ queue를 SAP별로 polling하여 적절한 서비스를 수행하도록 수현하였다.

Conversion S/W가 SAP을 polling할 때 각 SAP의 상태는 그림 A.12와 같이 정의하여 지정된 SAP으로 오는 action code에 따라 상태가 천이된다. SAP의 초기 상태는 DOWN state이고 actreq code에 의해서 CLOSED 상태가 된다. CLOSED 상태는 X.25에서 Call Request 프리미티브를 받으면 OPENING 상태로 천이하고 call request에 대한 적절한 서비스를 수행한 후 데이터 전송 상태인 OPENED 상태로 천이한다. 이와 반대로 원거리 RIU로부터 Connect Request 프레임(SABME)을 받을 경우 RIM의 LLC가 적절한 서비스를 수행한 후 그 결과를 알리면 CLOSED 상태에서 OPENED 상태로 천이한다. 일단 OPENED 상태가 되면 Conversion S/W는 각 SAP별로 UBUSY와 RBUSY 상태를 두어 X.25와 LLC 사이에서 flow control을 수행한다. 데이터 전송이 끝나고 X.25에서 Clear Request 프리미티브나 혹은 remote RIU에서 Disconnect Request 프레임(DISC)을 받으면 LLC firmware에게 알리고 call 해제 서비스를 수행한 후 Conversion S/W에게 그 결과를 알려 CLOSING 상태를 거쳐 CLOSED 상태가 된다.

A.3.3.1 Conversion S/W와 LLC firmware와의 통신

MCPM 모듈의 Conversion S/W와 RIM 모듈의 LLC firmware 사이의 통신은 firmware에서 제공하는 System Command를 사용한다. 이들 firmware의 System Command들은 다음과 같다.

- 1) Open
- 2) Transmit
- 3) Receive
- 4) Close
- 5) Open.sap
- 6) Close.sap
- 7) Open.station
- 8) Close.station
- 9) Connect.station

Conversion S/W가 위의 command를 사용하기 위해서는 여러 개의 테이블이 필요하며, 이 테이블들의 역할은 다음과 같다.

1. SCB(System Command Block)와 SSB(System Status Block)

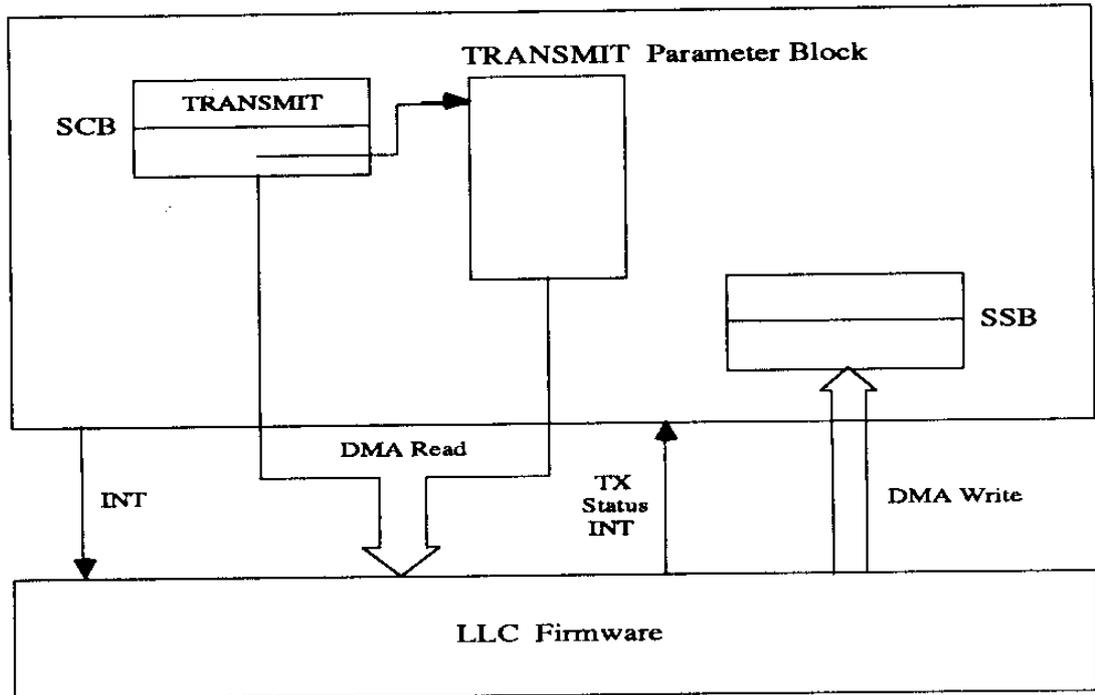
MCPM 모듈에 SCB와 SSB를 두고 Conversion S/W에서 LLC firmware command와 그와 관련되는 System Command 파라미터를 SCB와 SCPB(System Command Parameter Block)에 쓴 후 SIF interrupt register를 통하여 firmware에 interrupt를 요구한다. 그러면 RIM의 LLC는 MCPM의 SCB와 SCPB의 firmware command와 파라미터를 DMA로 읽고 적절한 동작을 취한 후 그 결과를 SSB에 DMA로 write한 후 아래 내용의 INT 중 하나를 SIF interrupt register를 통하여 MCPM에 요구한다. RIM에서 MCPM에 요구하는 INT는 다음과 같다.

- o Ring Status INT
- o Receive Status INT
- o Transmit Status INT
- o Command Status INT
- o Receive Pending INT
- o LLC Status INT

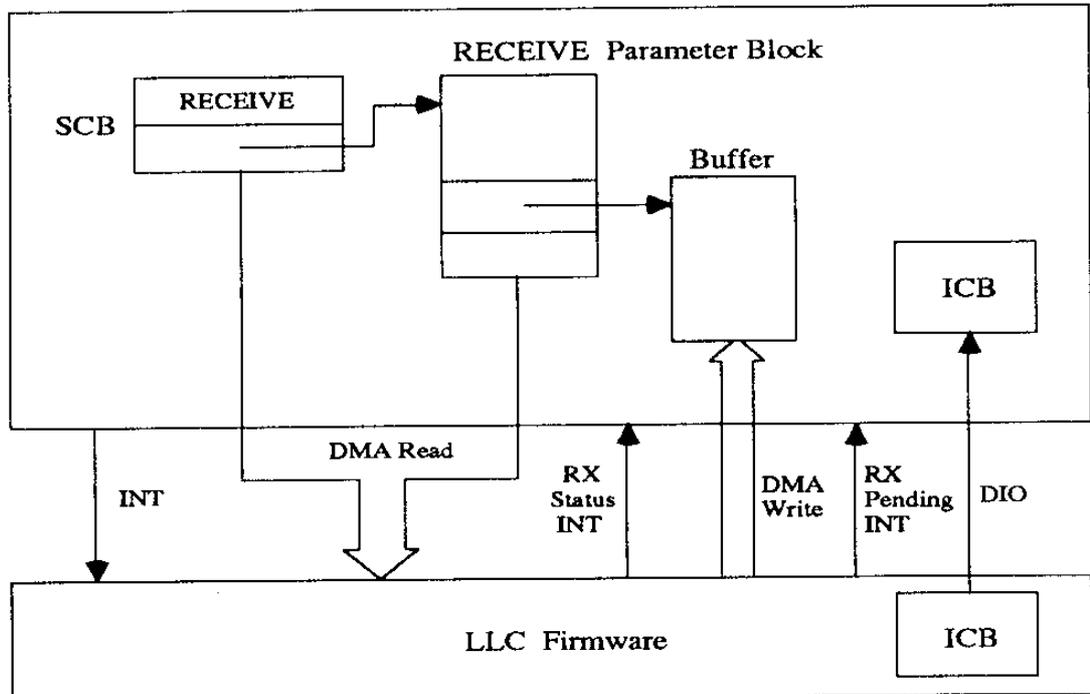
2. SCPB(System Command Parameter Block)

Conversion S/W에서는 LLC firmware의 각 command에 대하여 하나씩의 파라미터 블록을 준비해야 한다. 이들중 receive command와 transmit command의 파라미터 블록을 SCB, SSB, 그리고 Interface Control Block(ICB) 상호 관계로 나타내면 그림 A.13, A.14와 같다.

Transmit command 파라미터 블록은 전송할 데이터의 위치와 transmit command status, 프레임 크기, DSAP header 길이를 포함한다. Receive command 파라미터 블록은 프레임을 수신할 때 필요한 버퍼 주소, 최대 수신할 수 있는 프레임 크기, receive command status를 포함한다.



그림A.13 TRANSMIT command와 Parameter



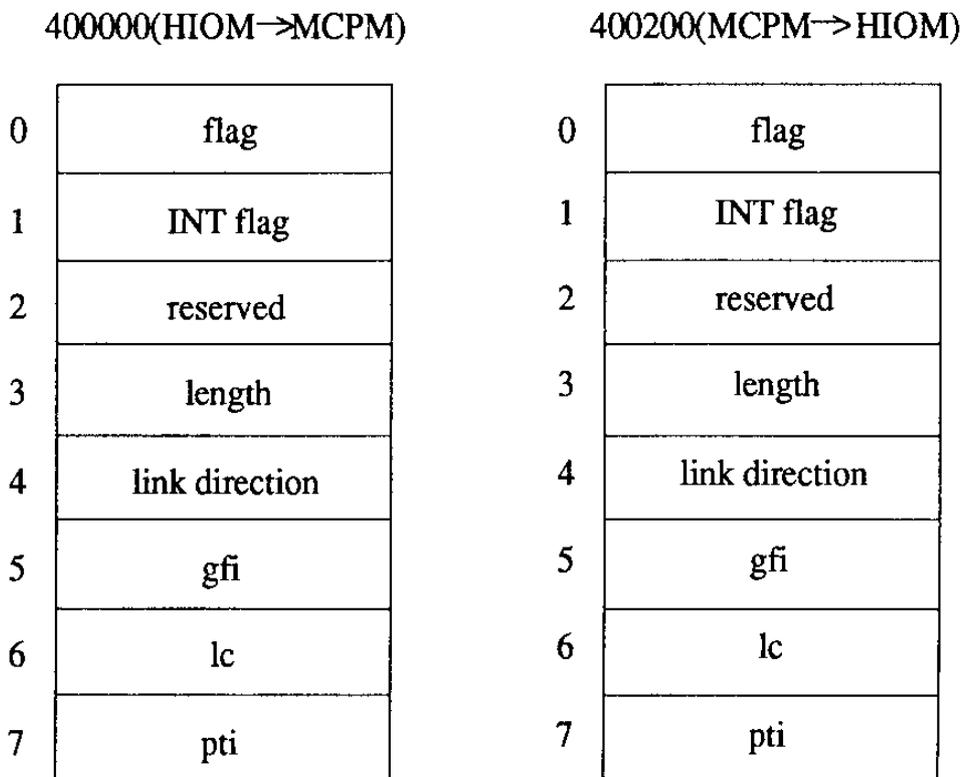
그림A.14 RECEIVE Command와 Parameter Block

3. Interface Control Block(ICB)

RIM의 LLC는 네트워크에서 Connect Request 프레임(SABME)이나 데이터 프레임, RR, RNR, DISC, FRMR 프레임 등을 받으면 그 프레임과 네트워크 링크에 대한 정보 및 상태를 RIM에 있는 memory에 보관한 후 MCPM에 LLC Status INT 혹은 Receive Pending INT를 요구한다. 그러면 MCPM의 Conversion S/W는 RIM memory의 내용을 DIO로 ICB에 읽어들이고 후 적절한 서비스를 수행한다.

A.3.3.2 Conversion S/W와 X.25와의 통신

HIOM 모듈의 X.25와 MCPM의 Conversion S/W와의 통신은 CM을 통하여 이루어지며, 이들 사이의 프리미티브는 X.25 S/W 부분에서 상세 기술하기로 한다. 그리고 CM의 포맷은 그림 A.15와 같다.



그림A.15 Common Memory Format

A.3.3.3 Conversion S/W의 동작 절차

Conversion S/W가 HIOM의 X.25나 RIM의 LLC에서 Call Request 프리미티브나 Connect Request 프레임 받은 경우에 수행하는 절차는 다음과 같다.

1. X.25로부터 Call Request 프리미티브를 받은 경우 :

MCPM의 Conversion S/W가 X.25로부터 Call Request 프리미티브를 받으면 idle한 SAP을 하나 할당받아서 OPEN.station과 CONNECT.station command를 RIM의 LLC에게 보내준다. 이때 LLC는 remote RIU의 LLC와 call을 성립시킨 후 그 결과를 MCPM의 SSB에 쓰고 MCPM에 Command Status INT를 요구한다. 그러면 Conversion S/W는 X.25에 Call Connected 프리미티브를 보내어 end-to-end call이 성립되어 데이터 전송 상태가 된다.

2. Remote RIU로부터 Connect Request 프레임(SABME)을 받은 경우 :

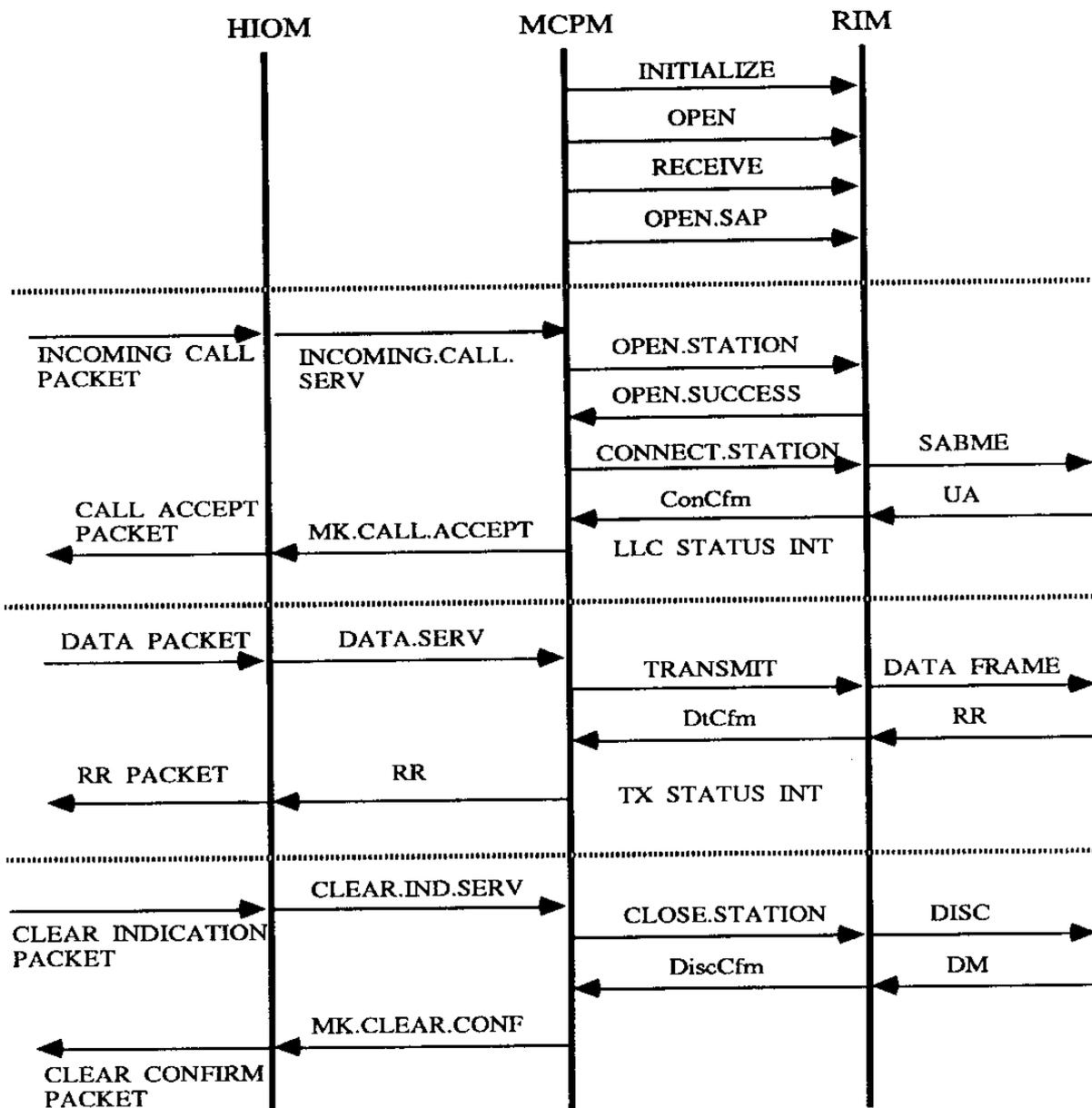
RIU의 LLC는 remote RIU 주소 및 필요한 정보를 ICB에 쓴 후 MCPM에 LLC Status INT를 요구한다. 그러면 Conversion S/W는 ICB에서 필요한 정보를 취한 후 CONNECT.station command를 LLC에 내려 Conversion S/W와 remote RIU의 LLC와 call을 성립시킨다. 일단 LLC 사이에서 call이 성립되고 데이터 전송 상태가 되면 Conversion S/W는 최초로 받은 데이터 프레임으로부터 remote X.25의 called address를 얻은 후 Call Request 프리미티브를 통하여 X.25에게 call을 요구하고 Call Confirm 프리미티브를 받으면 end-to-end call이 성립되어 데이터 전송 상태가 된다.

3. 데이터 전송 상태

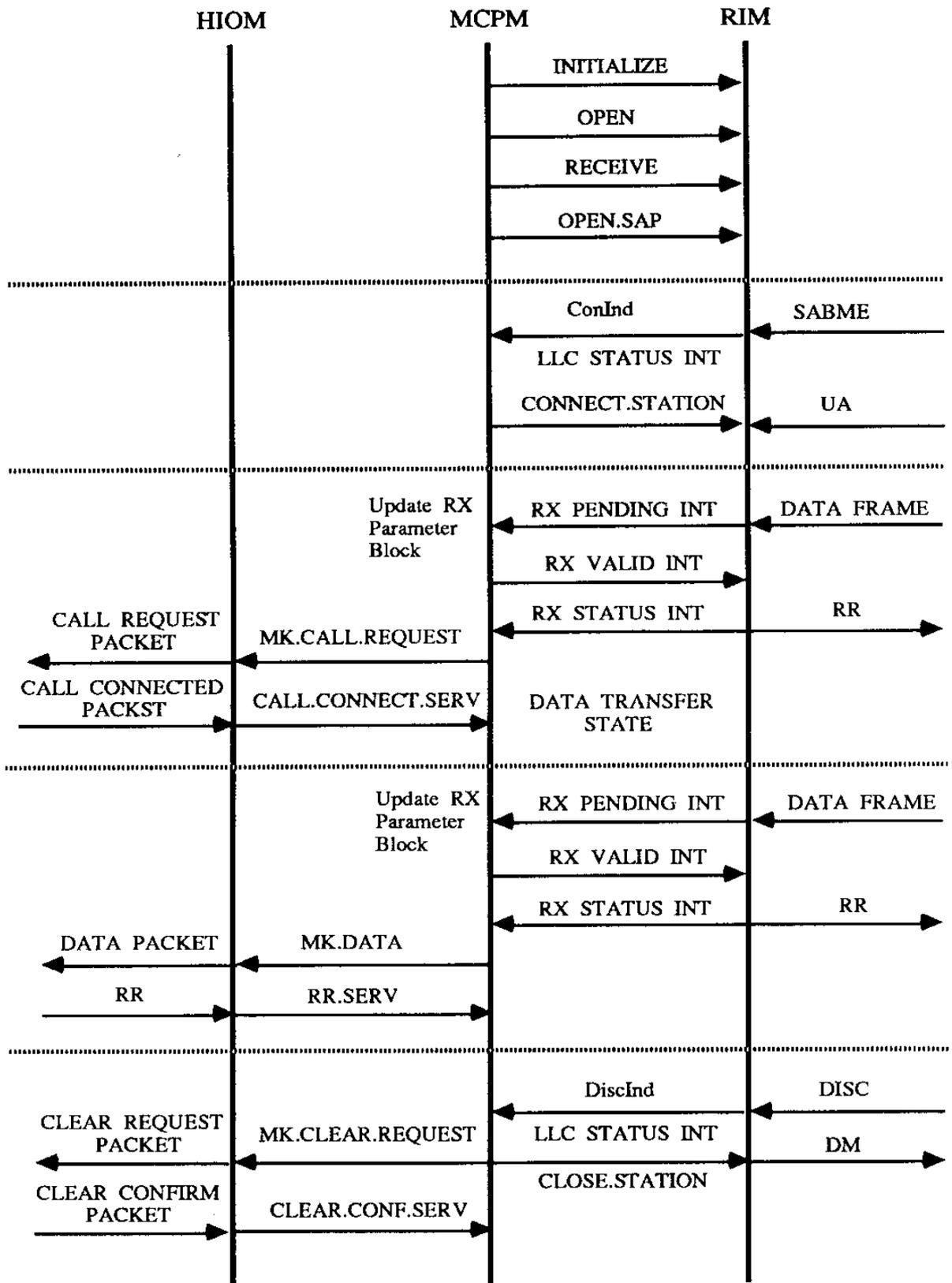
일단 데이터 전송 상태가 되면 Conversion S/W에서는 flow control을 해야 한다. Conversion S/W와 LLC와의 flow control에 관련되는 System Command는 Type 1 LLC의 XIDc와 XIDr 프레임을 이용한다. 즉, X.25로 보낼 packet 양의 상한 threshold 수준보다 많은 경우에 XIDc를 remote RIU에 보내어 프레임의 송신을 중지시키고, packet의 양이 하한 threshold 수준보다 적으면 다시 XIDc를 보내어 프레임의 재송신을 계속한다. 이와 반대로 Conversion S/W에서 remote LLC로 보내는 프레임의 경우에는 LocBusy와 LocBclr code를 사용하여 flow control을 수행한다.

데이터 전송이 끝난 후 call을 해제하는 경우는 X.25에 Clear Request 프리미티브를 받는 경우, Conversion S/W는 CLOSE.station command를 LLC에 내리고 LLC가 이에 대한 응답을 하면 Clear Confirm 프리미티브를 X.25에 보내어 call을 해제한다.

Remote RIU에서 Disconnect Request 프레임(DISC)을 받는 경우는 LLC가 DISC 프레임을 받고 LLC Status INT를 요구하면 Conversion S/W는 CLOSE.station command를 내리고 X.25에 Clear Request 프리미티브를 보낸다. 그리고 X.25에서 Clear Confirm 프리미티브를 받으면 call이 해제된다. 이들 전체의 절차를 그림 A.16에 나타내었다.



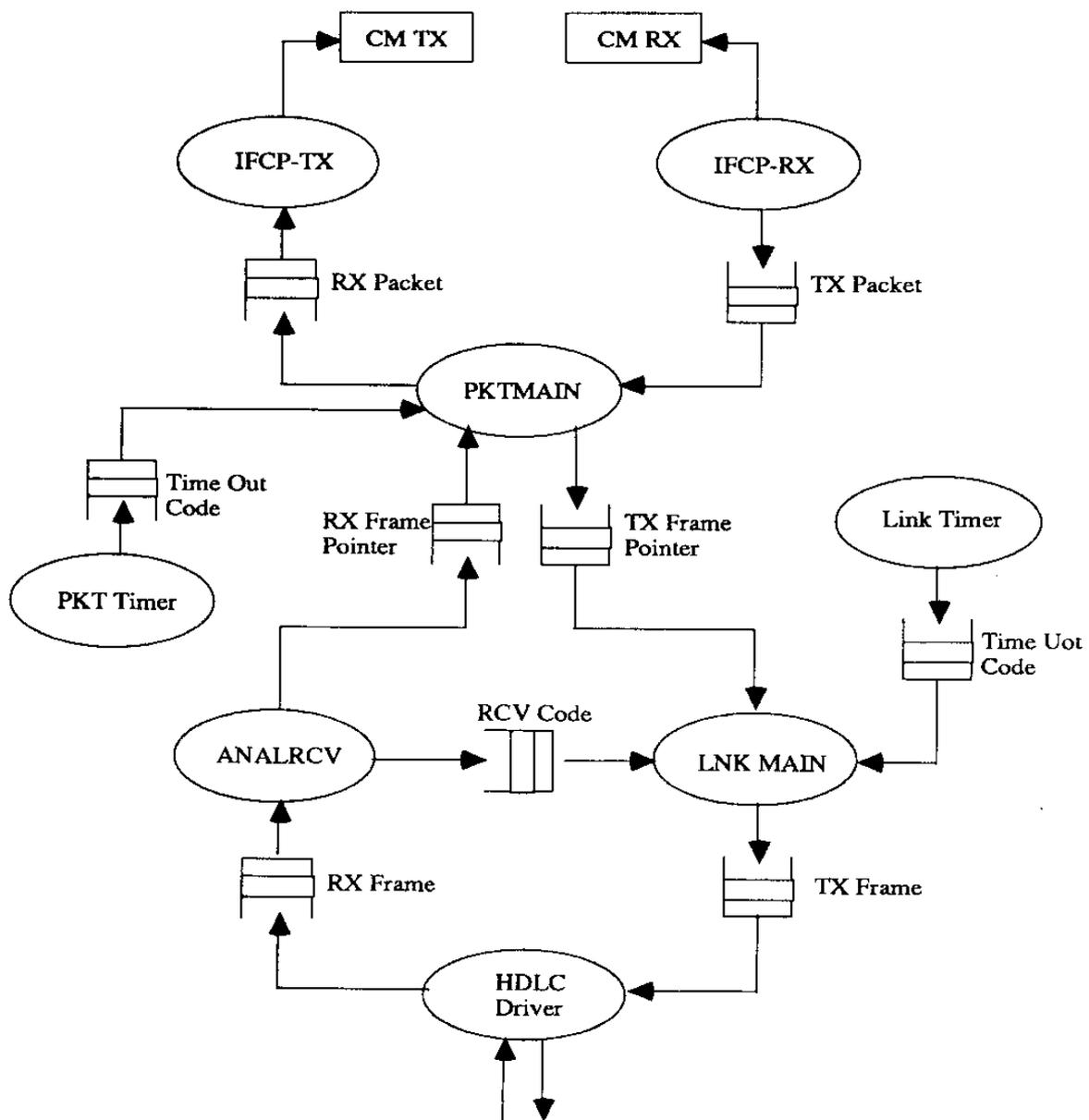
그림A.16 (a) HIOM에서 Call Request Packet을 받을 경우 DATA FLOW



그림A.16 (b) RIM에서 SABME를 받을 경우 Data Flow

A.3.4 X.25 Software

CCITT X.25는 Physical level, Link level, Packet level로 나뉘어져 있다. 여기서는 Physical level이 SIO를 사용하여 구현되었으며 관련된 S/W로는 초기화 routine과 HDLC 구동 routine이 있다. 초기화 routine은 H/W와 S/W가 사용하는 입출력 관련 system parameter를 초기화하며, HDLC 구동 routine은 Link level의 frame 전달을 위해 RHDQ와 THDQ를 사용한다. X.25 S/W의 전체 구조는 그림A.17과 같다.



그림A.17 X.25 Software 구조

Link level은 Physical level로부터 온 frame을 해석하는 FRMANAL routine과 protocol에 따른 처리와 응답을 담당하는 FRMMAIN routine으로 구성된다. FRMANAL routine은 packet level의 monitor routine에 의해 RHDQ에 frame이 수신되는 경우 구동되어 frame을 검진한다.

Packet level은 기본 monitor 역할을 수행하는 PKTMAIN routine과 필요시 구동되는 해석 routine 및 여러 Service routine으로 구성된다.

A.3.4.1 Packet level의 S/W 구조

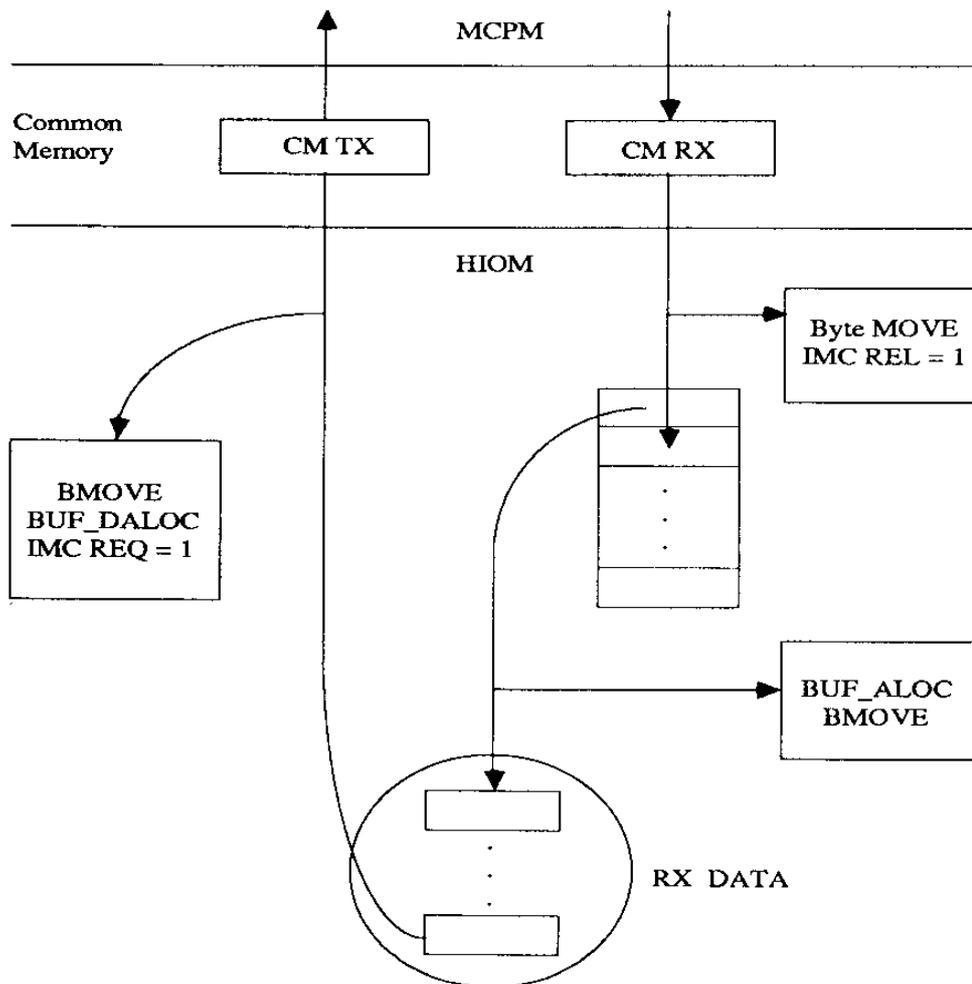
Packet level의 기본 Monitor routine은 시스템내의 여러 queue와 테이블들을 polling하여 Event의 발생 유무를 항상 감시하면서 필요시 해당되는 subroutine을 동작시키는 무한 loop 구조를 가진다. 여기서 Monitor는 Link level에서 Packet level로 packet을 전달하는데 쓰이는 RLKQ를 이 packet level 프로토콜에 따라서 오류 검진을 행한다. 즉 packet의 길이와 general format identifier, modular format, logical channel number 등을 체크하여 프로토콜에 따라 버릴 packet은 버리고 개설된 기존의 통화로를 해제할 필요가 있을 만큼 회복 불가능한 경우는 통화로를 해제시킨다. 오류 검진을 통과한 packet은 packet type identifier를 읽어 해당 서비스 routine을 구동시킨다. 그리고 RHDQ를 체크하여 physical level에서 link level로 전달된 프레임이 있는 경우에는 link level의 해석 routine인 FRMANAL을 구동시킨다. 또 상위 계층(Conversion S/W)으로부터 packet level로 전달되는 서비스 프리미티브를 처리하기 위해 IMCRX queue를 체크하여 처리해준다. 여기서는 이 서비스 프리미티브의 format을 프로그램의 간소화를 위해 X.25 packet level의 format과 동일하게 사용하였다.

Packet level의 service routine에는 PKTINIT, DTSERV, DSERV, PSERV, RSERV 등이 있다. PKTINIT은 warm start와 cold start로 나누어지는 packet level 초기화 routine이며, DTSERV는 packet level의 data 전송을 관리하는 routine으로 internal queue를 이용해 busy 상태의 설정 및 해제 등의 Flow Control과 이미 성립되어 있는 통화로를 통해 data 전송을 수행한다. DSERV는 이미 성립되어 있는 가상 통화로상에서 필요한 Reset, Interrupt 절차를 관리하며, PSERV는 통화로의 개설 및 해제를, RSERV는 packet level의 Restart 절차를 수행한다.

A.3.4.2 X.25와 Conversion S/W의 접속

X.25와 Conversion S/W는 서로 다른 module로 구성되어 독자적으로 동작되므로 Common Memory를 이용한 H/W interrupt 방식으로 정보를 교환한다. 우선 X.25에서 Conversion S/W로 packet을 넘겨주는 경우를 생각하면, X.25의 SENDPKT routine은 현재 interrupt를 걸 수 있는가를 check한 후 IMCTXB buffer가 비어 있으면 packet을 이 buffer로 옮긴 후 Conversion S/W로 interrupt를 건다.

Conversion S/W에서 X.25로 packet을 보내는 경우에는 interrupt를 받으면 interrupt service routine에서 receive queue로 IMCRXB buffer에 있는 packet을 옮긴 후 IMCRXB buffer가 비어 있는 것을 알린 후 interrupt를 해제시킨다. 그림A.18에 위 과정을 도시하였다.



그림A.18 X.25와 Conversion S/W의 Packet 교환 과정

한편 X.25와 Conversion S/W 사이의 정보 교환을 위해서는 둘 사이에서 약속된 서비스 프리미티브가 필요하다. 이 서비스 primitive는 정보 교환에 필요한 일정한 format을 갖추어야 하며, 여기서는 기존의 X.25 S/W 구조를 고려하여 이 서비스 프리미티브 형태를 X.25 packet format으로 정하였다. 서비스 프리미티브를 교환하기 위한 Common Memory의 사용에는 여러 가지 flag가 필요하며, 그림A.19에 이들 flag와 IMC buffer의 사용 format을 나타내었다.

Flag	내 용
IMCREQ	Interrupt Request Flag (X.25 → LLC)
IMCREL	Interrupt Request Flag (LLC → X.25)

IMC RX Buffer

90200	90201	90202	90203
IMCRSTA	IMCRIFLG	RESERVED	PACKET

IMC TX Buffer

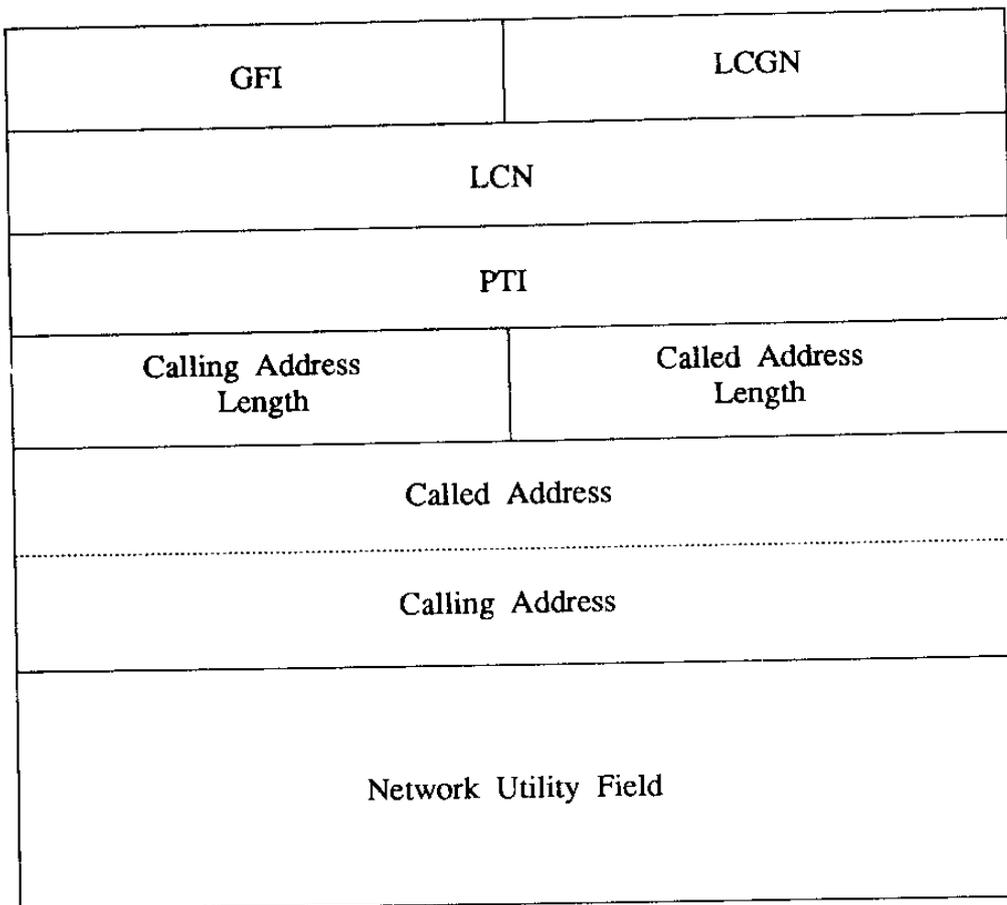
90000	90001	90002	90003
IMCTSTA	IMCTIFLG	RESERVED	PACKET

Flag	내 용
IMCRSTA	RX Buffer status (1 : empty, 2 : full)
IMCRIFLG	RX Interrupt flag (SET = 1)
IMCTSTA	TX Buffer status (1 : empty, 2 : full)
IMCTIFLG	TX Interrupt flag (SET = 1)

그림A.19 IMC Buffer Format 및 Flag

A.3.4.3 Addressing 기능

서로 다른 두 network을 연결할 때에는 각 network간의 addressing scheme이 다를 수 있으므로 고속 동기 접속 장치에서 이를 해결해 주어야 한다. X.25 protocol에 의하여 DTE가 접속되는 PDTE에서 addressing 정보가 들어가는 Call Request packet의 형식은 그림A.20과 같다. 즉 address field에 나타난 것과 같이 calling address와 called address의 길이만큼의 called address와 calling address를 사용한다. 한편 IEEE 802.2 LLC frame에서는 DSAP과 SSAP field에 의하여 addressing을 행한다.



그림A.20 X.25 규정에 따른 Call Packet 형식

만일 대덕 지역망의 RIU에 접속된 사용자가 고속 동기 접속 장치를 통하여 PDTE에 접속된 어떤 DTE와 통신을 하고자 할 때, 사용자는 call setup시 고속 동기 접속 장치로 상대방의 address를 보내주어야 한다. 그러나 LLC의 SABME frame에는 상대방의

address를 보낼 만한 여지가 없으므로 이것을 SABME frame 다음에 I frame을 보내어 해결하였다. 즉, SABME frame에 의하여 고속 동기 접속 장치의 특정한 SAP(1-8)과 연결을 한 다음 첫번째 I frame에 상대방의 address를 보내고 고속 동기 접속 장치에서는 이 address를 사용하여 PDTE로 X.25 call packet을 전송하게 된다. 반대로 PDTE에 접속된 DTE에서 고속 동기 접속 장치를 통하여 대덕 지역망의 가입자에게 call request를 시도할 경우는 PDTE의 addressing scheme에 의하여 고속 동기 접속 장치가 접속된 trunk의 address 뒤에 sub-address를 첨가하여 이 sub-address field에 대덕 지역망의 addressing scheme을 첨가해 준다.

A.3.4.4 Flow Control 기능

접속된 device들간의 속도의 불일치를 해결하기 위하여 일반적으로 network에서는 flow control을 하게 된다. 대덕 지역망에서는 LLC의 RR, RNR frame 등에 의하여 flow control을 하고, X.25 프로토콜로 접속되는 PDTE에서는 X.25 packet level의 RR, RNR packet 등에 의하여 flow control을 해결한다. 즉, 대덕 지역망과 PDTE는 각각 독자적인 flow control scheme을 가지고 있으며, 고속 동기 접속 장치를 통하여 두 network를 연결하고자 할 때에는 end user들 사이의 원활한 통신을 위하여 각 network내에서의 flow control 뿐만 아니라 접속점이 되는 고속 동기 접속 장치에서의 flow control이 있어야 한다.

고속 동기 접속 장치에서의 flow control은 X.25 S/W와 LLC S/W 사이의 원활한 data 전송을 하기 위한 것으로 양방향으로의 TX, RX buffer가 넘치지 않도록 하기 위해 Conversion S/W에서는 Upper layer busy(UIBusy), Local busy(LocBusy) flag와 txflag 및 rxflag를 각 SAP별로 운영한다.

먼저 PDTE로부터 지역망으로의 data 흐름을 control하는 과정을 보면, 어느 순간 지역망에서 처리할 수 없는 많은 양의 packet이 X.25 trunk로부터 들어오면 고속 동기 접속 장치의 Conversion S/W는 지역망의 end user에 접속된 LLC로부터 RNR frame을 받게 된다. 이때 Conversion S/W에서는 LocBusy flag를 set하고 지역망 user로의 데이터 전송을 중지한다. 이때 PDTE로부터의 데이터 전송은 계속되므로 Conversion S/W내의 buffer는 점점 커지게 되며, 어느 threshold가 되면 txflag를 set하고 X.25 S/W 쪽으로 RNR packet을 보내어 input 흐름을 막는다. 이 상태로 일정 시간이 경과하면 지역망의

end user측 LLC로부터 쌓여진 frame의 처리가 완료되며 RR frame이 들어오게 되어 Coversin S/W에서는 buffer로부터 frame을 전송하게 되고 어느 threshold 이하로 buffer size가 줄게 되면 txflag를 reset하고 X.25 S/W로 RR packet을 보내어 input 흐름을 허용한다. 이와 마찬가지로 지역망에서 PDTE로의 데이터 흐름에 대한 flow control도 같은 scheme으로 운영된다.

A.3.4.5 X.25 Diagnostics 기능

X.25 규격에는 여러 가지 option과 parameter가 있으며 주변 상황에 따라 이들을 맞추어주어 원활한 통신을 이루기 위해 가상적인 port인 diagnostic port가 필요하다. 이 port는 실제 port는 아니지만 call을 성립시켜 이용자가 필요한 정보를 교환할 수 있다. Diagnostic port는 password로 보호되며 다음과 같은 목적에 이용된다. 첫째, X.25 trunk(Link level 및 Packet level)에 대한 각종 parameter 값을 수정하여 trunk를 구성 또는 재구성하기 위하여 이용된다. 둘째, X.25 trunk에 대한 각종 정보와 통계치를 얻기 위하여 쓰인다.

고속 동기 접속 장치에는 한자리의 지정된 Diagnostic port address가 있으며 이는 power on시 DIP switch에 의해 결정된다. Diagnostic port를 access한 후에 Diag: prompt가 나오면 해당 명령어로 필요한 기능을 수행할 수 있다. Diagnostic command에는 모두 15가지가 있으며 'HELP' 명령어를 통하여 참조할 수 있다.

X.25 trunk나 LAN에서 고속 동기 접속 장치의 diagnostic port를 call하게 되면, 일반적인 call request packet으로 간주하여 Pserva.c file의 CALserv() routine을 call한다. 이 routine에서는 RoutFct routine으로 diagnostic port에 대한 call인지를 check한다. Diagnostic port에 대한 call이 확인되면 dmain.c file의 dcall()을 부른다. 이 routine에서는 Dact를 1로 set하고 call confirm을 보낸다. 그 후부터는 packet level main polling routine에서 Dact를 check하여 dmain0(), dmain1() routine을 call하여 diagnostic command를 처리한다.

고속 동기 접속 장치의 address는 최대 11 digit이며, 끝의 두 자리는 LLC의 SAP address를 나타내고 나머지 자리는 고속 동기 접속 장치의 address를 나타내어 전체적으로 고속 동기 접속 장치의 port address를 형성한다.

Appendix B. PC 환경에서의 멀티미디어 MHS 연구

오늘날 컴퓨터를 이용하여 지리적으로 먼 거리의 사용자와 정보를 주고 받을 수 있는 기술중의 하나로 MHS(Message Handling System)가 널리 이용되고 있다. 그러나 현존하는 MHS들은 대부분 전달할 수 있는 정보가 문자(Text) 형식으로 제한되어 있다. 하지만 일상 생활에서는 문자 뿐만 아니라 말이나 그림, 사진 등으로 의사를 전달하기도 한다. 컴퓨터를 사용한 정보 전달에 있어서도 가능한 이에 가깝게 문자, 화상, 그래픽 및 음성과 같은 다양한 매체(Media)로 표현된 메시지를 작성하여 전송할 수 있도록 요구되고 있다. 이와 같이 사용자가 컴퓨터를 이용하여 멀티미디어 정보를 표현하고 전달할 수 있는 Store-and-Forward 방식의 전자 우편 시스템이 Multimedia MHS이다.

본 연구에서는 멀티미디어 MHS 구현시 고려해야 할 제반 요소를 알아보고, 이를 토대로 PC 환경에서의 문자, 그래픽스, 화상을 처리할 수 있는 멀티미디어 시스템 구현을 모색한다.

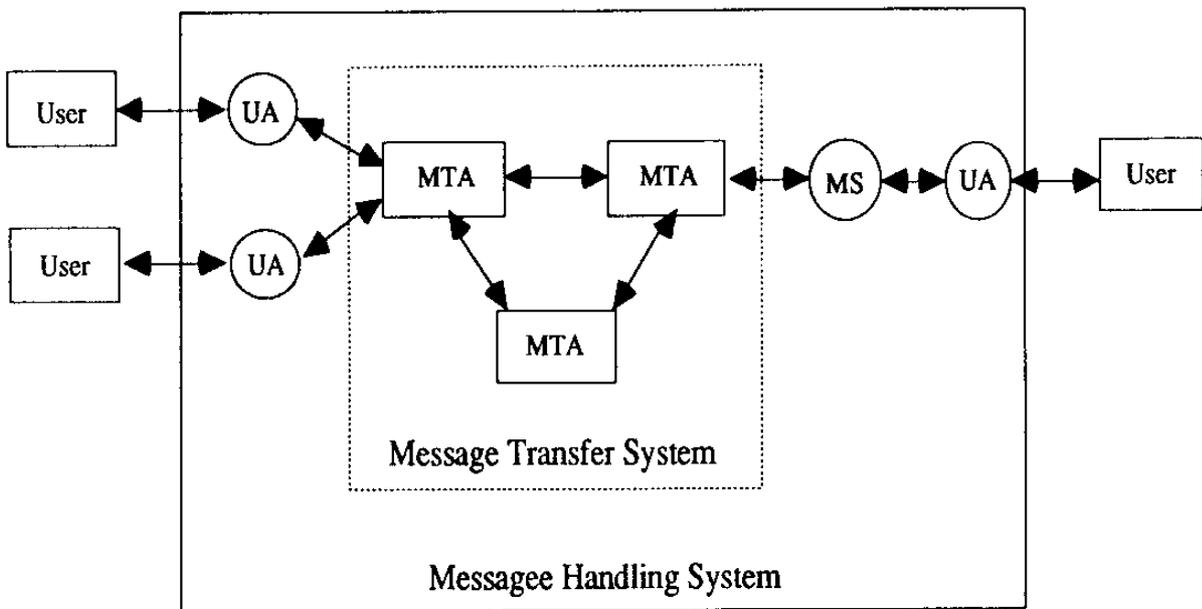
B.1 MHS의 개요

MHS는 Store-and-Forward 방식으로 메시지를 교환해주는 전자 우편 시스템으로 컴퓨터망을 통하여 송신자가 수신자에게 메시지를 전달하는 방법이다. 현재 여러 방식의 MHS가 개발되어 사용되고 있으며 또한 새로운 MHS가 개발되고 있으나, MHS 서비스도 전화나 우편 서비스의 경우처럼 다른 시스템이나 가입자와도 교신이 가능하여야 하므로 전세계를 하나의 MHS 서비스권으로 형성하기 위해 MHS에 대한 표준 규격을 제정하게 되었다.

CCITT의 X.400 프로토콜로 권고되고 있는 MHS 표준안은 1984년에 제안되어 1988년 수정, 보완을 거쳐 새롭게 발표되었다. MHS는 OSI의 Reference Model 이론에 맞추어 설계되었으며, 표현 계층의 서비스와 응용 서비스 요소가 제공하는 서비스를 사용한다. Message Handling 서비스에는 IPM(Interpersonal Messaging) 서비스와 MT(Message Transfer) 서비스가 있는데, IPM 서비스는 현존하는 CCITT의 텔리텍스와 텔레마틱 서비스를 포함하는 개인간의 통신을 정의하며 MT 서비스는 응용 프로그램에 독립적인 메시지의 전송을 담당한다.

B.1.1 X.400 MHS의 기능적 모델

X.400에서 정의하는 MHS의 모델은 그림B.1과 같다. 여기서 사용자는 사람 혹은 응용 프로그램이 될 수 있다. 송신자(Originator)는 UA(User Agent)의 도움을 받아 메시지를 작성하게 되는데, UA는 MTS(Message Transfer System) 혹은 MS(Message Store)와 연결되어 상호 동작하는 응용 프로세스이다. 송신자가 작성한 메시지는 UA 혹은 MS를 통하여 MTS로 제출(submit)되며, 제출된 메시지는 MT 서비스에 의해 수신측 MTA로 전달된 후 수신UA 혹은 수신MS로 배달(deliver)된다. UA는 MTS로부터 직접 메시지의 배달을 받아 들이거나, 배달된 메시지를 일단 MS에 저장한 후 복구(retrieve)할 수 있다. MTS는 그림B.1과 같이 여러 개의 MTA로 구성된다.



그림B.1 MHS의 기능적 모델

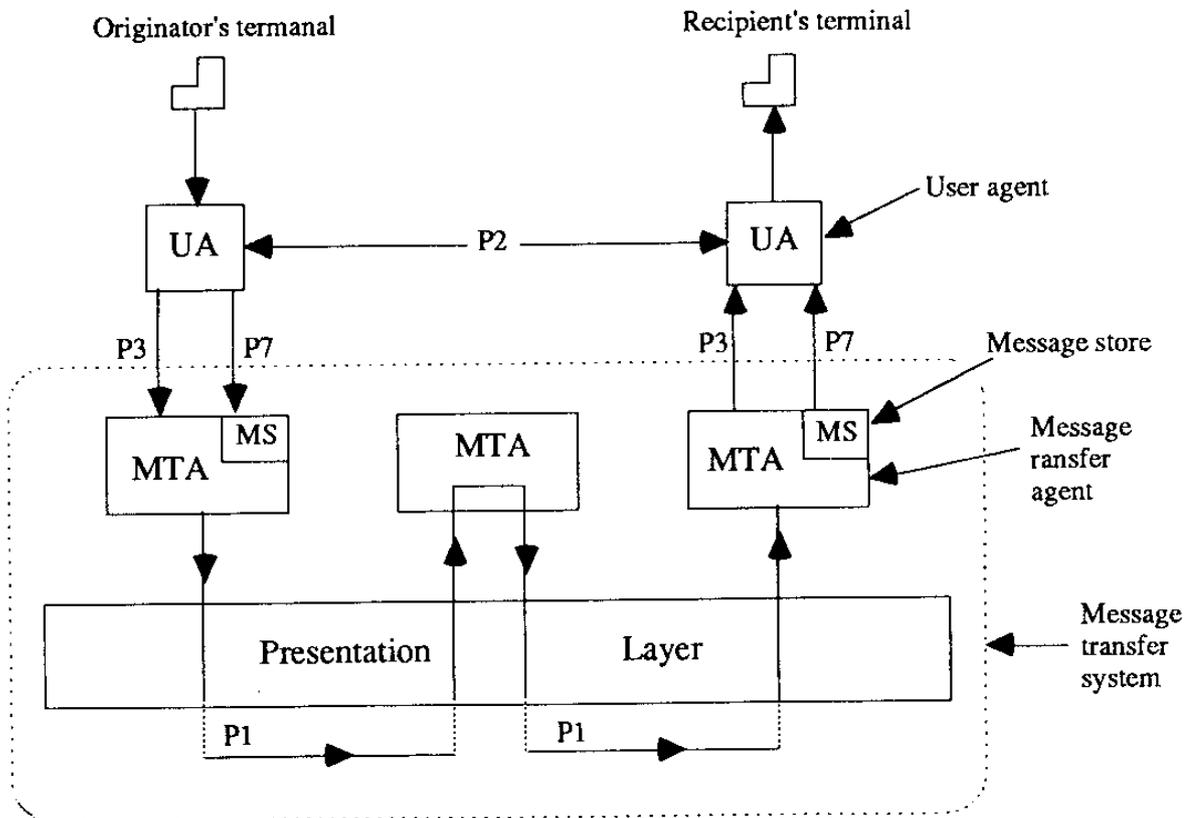
B.1.2 X.400 MHS 메시지 전송 서비스

MTS는 응용 프로그램에 독립적으로, 메시지를 Store-and-Forward 방식으로 목적지로 전달하는 기능을 제공한다. 또한 UA와 메시지를 교환하기 위하여 제출과 배달의 상호 작용을 제공하는데, 이때 메시지의 내용에 MTA가 요구하는 Envelope(봉투)가 추가되며 배달은 Envelope가 요구한 대로 행해진다. MTA가 제공하는 메시지 전송 서비스는 다음과 같다.

- 제출 : 송신UA 혹은 MS가 MTA로 메시지의 내용과 제출 봉투를 전송하는 수단
- 배달 : MTA가 수신UA 혹은 MS로 메시지의 내용과 배달 봉투를 전송하는 수단
- 전송 : 송신측 MTA로부터 수신측 MTA로 메시지를 전송할 때 MTA 사이에서 일어나는 메시지 전송 수단으로 각 MTA는 다른 MTA로 전송 동작이 일어날 때 메시지의 내용에 전송 봉투를 추가하게 되는데, 이 봉투에서 요구한 대로 전송을 수행하게 된다.
- 통지 : 전송한 메시지가 수신자에게 제대로 배달되었는지, 혹은 배달되지 않았는지를 알려주는 서비스로 배달 통지와 비배달 통지로 이루어져 있다.

B.1.3 X.400 MHS의 프로토콜

X.400에 정의되어 있는 프로토콜에는 다음의 4가지가 있으며, 이 프로토콜을 포함하는 MHS의 일반적인 모델은 그림B.2와 같다.



그림B.2 MHS의 일반적 모델

- MTS Transfer Protocol(P1) : MTA간의 메시지 전달에 필요한 상호 작용을 정의한다.
- Interpersonal Message Protocol(P2) : IPM UA간에 전송하는 메시지 내용(content)의 구문(Syntax)과 의미(Semantics)를 정의한다.
- MTS Access Protocol(P3) : 원격 UA가 MTS의 서비스를 제공받을 수 있도록 원격 UA와 MTS간에 정의된 프로토콜이다.
- MS Access Protocol(P7) : 원격 UA가 MS의 서비스를 제공받을 수 있도록 원격 UA와 MS간에 정의된 프로토콜이다.

B.1.4 EAN

EAN은 CCITT에서 권고한 X.400 프로토콜에 바탕을 두고 만들어진 분산 메시지 시스템으로서 Canada에서 개발되어 현재 유럽 등지에서 사용중으로 X.400 프로토콜에 따라 만들어진 MHS 중에서 가장 우수한 것으로 평가되고 있다. EAN은 다음과 같은 3가지의 구성 성분으로 나뉘어진다.

- User Service :
메시지를 작성하고 보내고 받으며 저장하는 기능.
- Message Transfer Service :
송신자로부터 메시지를 받아서 수신자까지의 경로를 찾아서 메시지를 수신자에게 전달해 주는 기능.
- Directory Service :
사용자의 이름을 사용자의 주소로 mapping시켜 주며, 또 사용자의 우편 주소, 전화 번호 및 직업 등을 알려준다.

EAN은 다양한 OS와 컴퓨터에서 수행되며, 주요 기능으로는 directory service, distribution list, auto filing, status reports, timed delivery 그리고 ucbmail과의 교환 기능 등이 있다.

B.2 멀티미디어 MHS의 현황

컴퓨터와 통신의 결합은 정보의 수집, 처리 및 전달 기술을 획기적으로 발전시켜 다양한 resource들의 효율적인 이용 및 부하 분산, 기능 분산 등을 위한 분산 처리 시스템으로 발전하고 있다. 지금까지의 컴퓨터 이용은 주로 문자 데이터로 된 문서 취급에 국한되어 왔으나 화상, 그래픽 또는 음성 등의 다양한 매체를 포함한다면 보다 구체적인 의미의 전달이 가능해진다. 따라서 앞으로의 컴퓨터 발전 방향은 다양한 매체 즉 문서, 화상, 컴퓨터 그래픽 및 음성으로 표현되는 멀티미디어 정보를 처리할 수 있는 멀티미디어 시스템이 중요한 이슈가 될 것이다. 이러한 다양한 메시지의 정보를 수집, 처리, 전달할 수 있는 MHS를 Multimedia MHS라고 한다. 멀티미디어 MHS는 여러 매체를 사용하여 메시지를 작성함으로써 정보를 충실히 교환할 수 있는 등 컴퓨터망의 가입자에게 유용한 facility를 제공해준다.

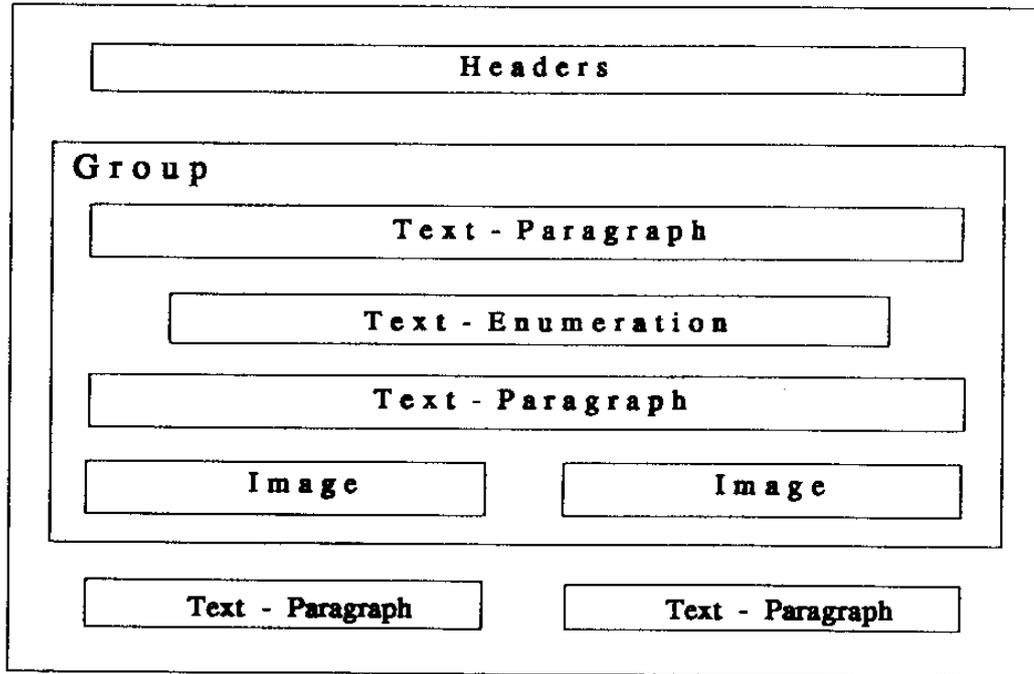
최근 주목을 받고 있는 CBMC(Computer-Based Multimedia Communication)는 컴퓨터 이용자간에 멀티미디어를 이용하여 표현된 정보를 처리 및 전달하는 것을 뜻하며, 이때의 interaction은 store-and-forward interaction과 real-time interaction이 있다. Store-and-forward 방식은 시간에 구애받지 않고 편리한 시간에 사용할 수 있어 지리적으로 분산된 사용자간에 정보를 전달하는데 매우 편리하다. 현재까지 개발된 멀티미디어 MHS는 아직 실험적인 단계를 벗어나지 못하였지만 조만간 완전한 시스템이 등장할 전망이다. 연구소나 학교 등에서 연구가 수행되고 있다.

B.2.1 Diamond 시스템

Diamond는 멀티미디어 문서를 작성, 편집, 보관, 전달 및 출력하기 위한 시스템으로 BBN(Bolt Beranet and Newman Inc.) Lab에서 개발하였다. Diamond 시스템으로 구성된 메시지는 그림B.3과 같으며 이 시스템의 특징은 다음과 같다.

- 서로 다른 매체의 데이터는 중첩시킬 수 없다.
- 시각적인 표현이 가능한 매체의 데이터는 display 화면상에 표현이 가능하다.
- 화상을 독립적인 매체로 취급하였다.

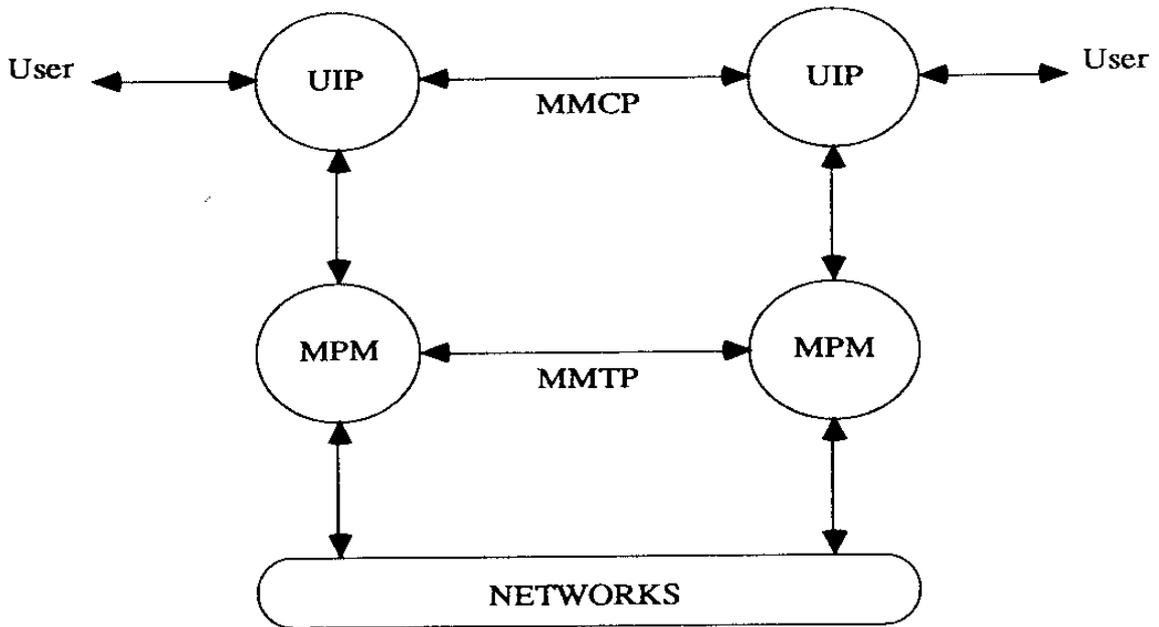
또한 다른 멀티미디어 시스템에 비해 특징적인 것은 절대 위치 및 상대 위치를 지정하여 데이터의 위치를 '배열'할 수 있는 장점을 가지며, 모든 매체의 데이터에 대해 칼라가 제공된다. 현재까지 Diamond에서 이용가능한 매체는 문서, 컴퓨터그래픽, 화상, 음성, Electronic Spread Sheet 등이다.



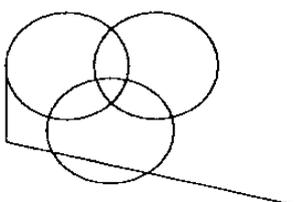
그림B.3 Diamond Message Content

B.2.2 ISI 시스템

이 시스템은 미국 DARPA(Defence Advanced Research Projects Agency)의 컴퓨터 네트워크 통신 프로젝트의 일부분으로 멀티미디어 통신을 위해 개발되었다. ISI 시스템은 DARPA Internet 프로토콜의 Application으로서 메시지의 routing, transmission, delivery를 담당하는 MPM(Message Processing Module)과 사용자가 메시지를 작성, 읽기, 송달해주는 UIP(User Interface Program)의 두 부분으로 그림B.4와 같이 구성된다. UIP들 간에는 문서에 대한 형식을 정의한 MMCP(Multimedia Mail Content Protocol) 프로토콜을 사용하는데 이 MMCP는 복잡한 데이터 object들의 작성과 machine-independent한 데이터 코딩을 제공한다. 그리고 MPM들 간에는 Internet을 통하여 멀티미디어 메시지들이 전송되도록 전송 절차와 형식을 정의한 MMTP(Multimedia Mail Transfer Protocol)을 사용한다. MPM들 사이에 전송되는 메시지의 구조는 ID(Unique Identifier), CMD (Command), DOC(Documentation)의 3 부분으로 구성된다. CMD 부분은 메시지를 routing하기 위해 MPM에 의해 사용되는 정보가 포함되어 있는데 이 정보는 UIP에 의해 제공된다. DOC 부분은 header와 body로 나뉘어지며, 문서의 body 부분은 4 가지의 매체가 표현될 수 있다.



그림B.4 MPM과 UIP의 관계

INFORMATION	BITMAP
HEADER 1 INFO TO: FROM: ***** HEADER 2 INFO TO: FROM: ***** HEADER 3 INFOR TO: FROM: *****	 <p>Some text In the bit map</p>
COMMANDS	TEXT
PLAY MSG OUTLINE MODE CREATE MODE CHECK FOR MAIL QUIT	THis is some text .

그림B.5 ISI Message Content

텍스트는 다양한 스타일로 formatting을 할 수 있고, 3 종류의 time-ordered control (sequential, simultaneous, independent)이 가능한데 이 시스템에서는 sequential presentation만이 실현되었다. 이 시스템은 PERQ PC상에서 4개의 윈도우로 구성되어 그림B.5와 같이 화면상에 디스플레이된다.

B.2.3 AGORA 시스템

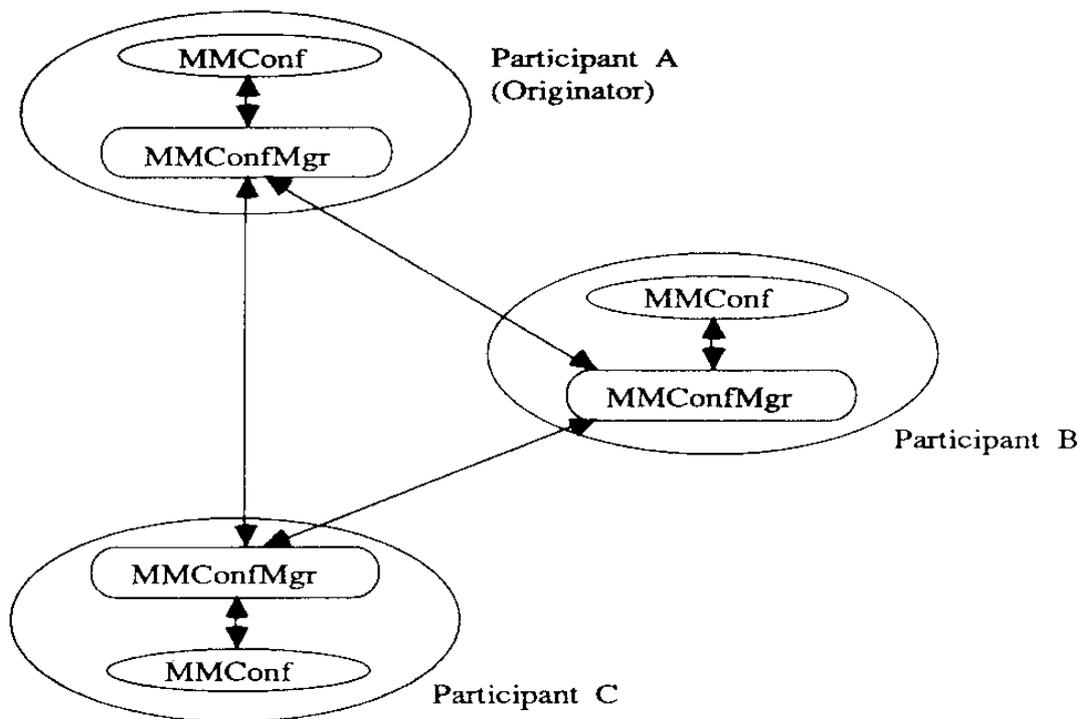
이 시스템은 프랑스에서 완전하고 종합적인 Electronic Office를 만들기 위한 Kayak이라는 프로젝트의 부분으로 설계되었고 AGORA의 기능적 모델은 IFIP (International Federation for Information Processing)의 모델에 따라 정의되었으며, Buroviseur 워크스테이션에서 구현되었다. Buroviseur에는 문서 관리를 위한 Virtail이란 윈도우 패키지, Plume이란 WYSIWYG 에디터가 있다. Virtail은 open window, display object, create menu, read input과 같은 프리미티브를 다루는 사용자에게 인터페이스를 제공해준다. Plume은 chapter, section, paragraph, 여러 폰트의 문자, 알파벳 등의 entity를 포함하는 구조화된 문서들을 다루게 해주며, 워크스테이션간의 상호 통신은 CSMA/CD 방식의 Danube LAN을 이용한다.

B.2.4 Columbia 대학교 시스템

Columbia 대학교의 Distributed Computation and Communication Group에서는 멀티미디어 관리 시스템을 개발하였는데, 이 시스템을 위하여 UNIX OS의 SUN 워크스테이션을 Ethernet에 연결하여 사용하였다. 이를 위하여 두 가지의 드라이버를 실현하였는데 하나는 흐름제어와 저속 입력 디바이스를 위하여, 다른 하나는 실시간 응용을 위한 것이다. 이들 드라이버는 사용자의 요구에 의해 동작되고 I/O 디바이스와 워크스테이션 메모리간에 블록 단위로 정보를 전송해준다. 후자의 드라이버를 위하여 UNIX 커널을 실시간 요구에 맞도록 수정하고, 이상적인 관리 시스템을 위하여 문서들의 상호 참조, 사용자 인터페이스, 데이터의 저장과 관리의 3가지를 중점으로 멀티미디어(텍스트, 그래픽, 음성) 에디터를 만들었다.

B.2.5 MMConf

BBN Lab에서는 MMConf라는 멀티미디어 회의 시제품을 개발하였다. MMConf는 지리적으로 떨어진 사람들이 동시에 컴퓨터를 사용하여 의사를 결정하거나 공동으로 작업할 때 실시간 대화형 방식으로 한 테이블에서 회의를 하는 것 같이 해준다. MMConf에서 화면은 참가자 각자에게 한 라인씩 할당되어 참가자에 대한 여러 정보를 제공해주는 회의상태 윈도우와, 모든 참가자들이 볼 수 있는 멀티미디어 회의윈도우로 나뉜다. 화면상에는 여러 가지 매체들이 표시될 수 있는데 MMConf에서 제공하는 매체는 문자, 그래픽, 화상, Electronic Spread Sheet 등이다. MMConf는 TCP/IP 프로토콜상에서 만들어진 일련의 분산 프로세스로 구현되었는데, 그림B.6은 MMConf 구성 요소들의 상호 작용을 보여준다. 여기서 MMConf 프로세스들은 회의 시스템에 대한 사용자 인터페이스를 수행하고, MMConfMgr 구성 요소들은 상호 작용을 보여준다.



그림B.6 MMConf component들의 관계

B.3 멀티미디어 표준화 동향

멀티미디어의 표준화 노력은 여러 국제 표준화 기구에서 시도되고 있는데, 특히 ISO에서는 ODA를 중심으로 멀티미디어 자료 표준화를 구체화하고 있다. 그 구체적인 내용으로는 멀티미디어의 제공, 멀티미디어 자료의 저장 및 관리, 멀티미디어 자료를 문서내에 포함한 통신, 멀티미디어의 동기화 등이다. 멀티미디어 문서 표준은 크게 각 매체에 대한 표준과 매체를 조합하여 구조화시키는 부분에 대한 표준으로 나눌 수 있다. 전자는 문자 표준안인 TROFF, LaTeX, SGML 및 ODA 등과 그래픽스 표준안인 GKS, CGM과 PHIGS, 그리고 화상 표준안인 G3, G4 FAX 코딩 방식 등이 있고, 후자는 X.400, PDL과 ODA 등이 있다.

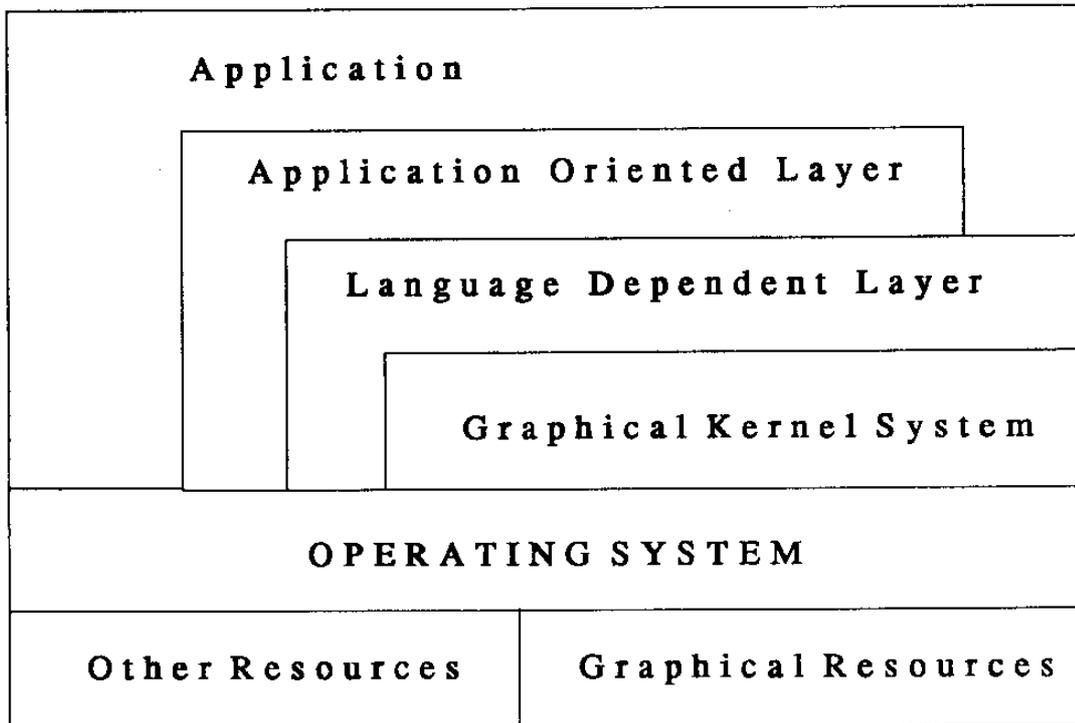
B.3.1 GKS(Graphical Kernel System)

현재 널리 사용되는 GKS는 컴퓨터 그래픽 프로그래밍에 대한 일련의 기능으로 구성된 기본적인 그래픽 시스템으로 ISO에 의해 국제적 표준으로 공인된 2D 그래픽 표준안으로, ANSI에서도 GKS에 관한 일련의 기능들을 서술하였다. 여기에는 그래픽 입력과 Picture segmentation에 대한 기본적 기능을 기술하고 있다. 또한 GKS에 워스테이션에 관한 기본 개념을 소개하여 단일 입력 장치와 단일 출력 장치, 다수의 워스테이션을 동시에 제공하였다.

GKS는 응용 프로그램과 그래픽 입력과 출력과의 기능적인 인터페이스를 제공하는 것으로 uniform 출력 프리미티브와 입력 프리미티브로 구성되어 있다. Picture manipulation과 change를 위하여 segment, dynamic attributes, transformation 개념을 소개하고 GKS 시스템이 넓은 범위의 응용에 쓰이므로 전체를 12 레벨로 나누어 각각의 응용에 따라 여러 가지 응용에 효율적으로 대처하도록 하였다. 그림B.6은 그래픽 시스템에서의 GKS 역할을 나타낸다.

B.3.2 CGM(Computer Graphics Metafile)

CGM은 그래픽 소프트웨어에 의하여 생성된 그래픽 정보의 저장과 재생에 대한 적절한 file format을 제공한다. File format은 다른 기능을 가지고 상이하게 고안된 장치나 다른 구조의 시스템간에도 호환성이 있도록 그래픽 정보를 표현하고 있는 요소(element)들의 집합으로 구성된다. 이러한 CGM을 표준화하고 있는 이유는 다음과 같다.



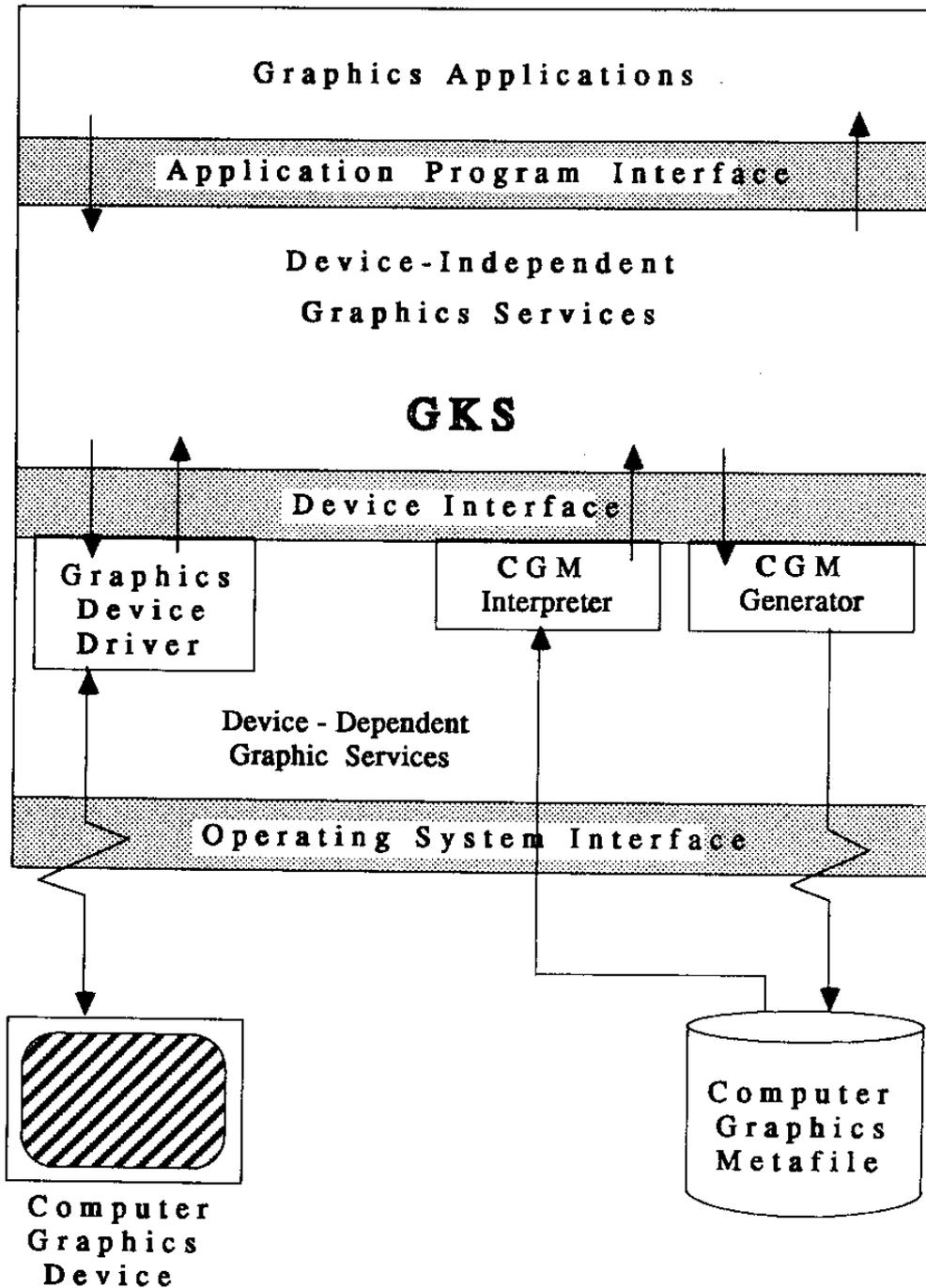
그림B.6 Layer Model of GKS

- 그래픽 정보가 그래픽 소프트웨어 시스템상에서 균일한 형식에 맞추어서 저장될 수 있도록 한다.
- 다른 그래픽 소프트웨어 시스템 사이에도 그래픽 정보를 전송할 수 있게 해준다.
- 다른 그래픽 장치에서도 그래픽 정보를 볼 수 있도록 해준다.

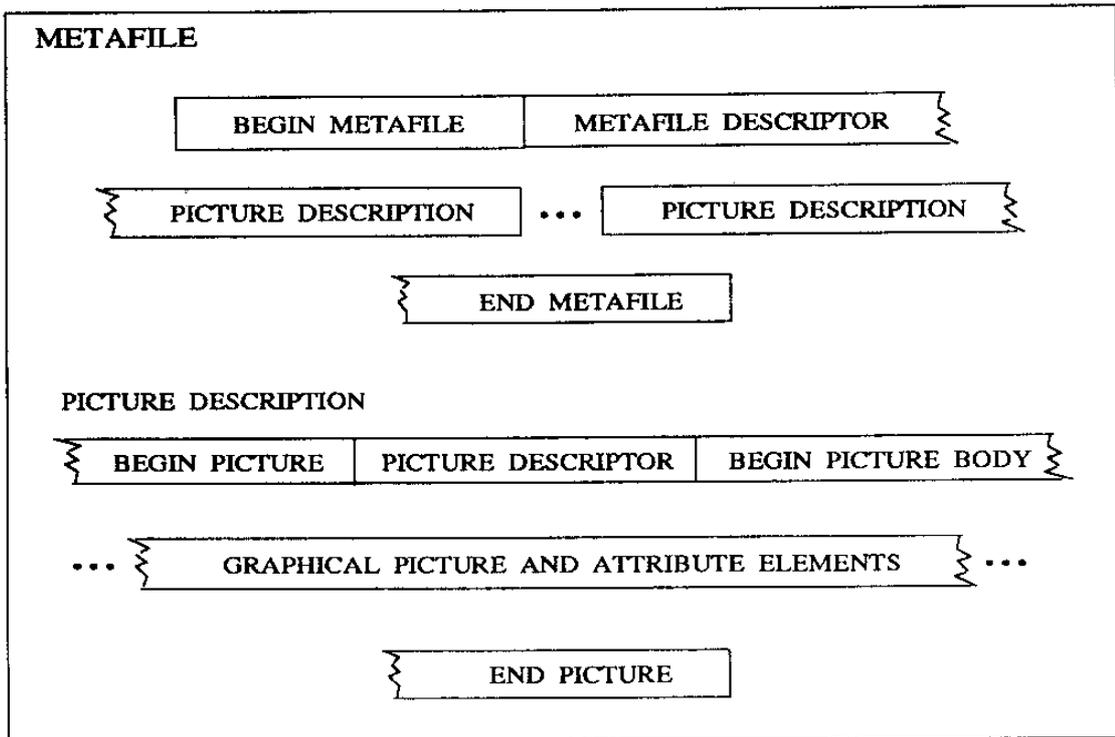
그래픽 정보에서 CGM file을 만들고 재생하는 방법은 그림B.7과 같이 CGM Interpreter와 Generator를 이용하여 그래픽 정보를 처리하고 있다. 여기서 Metafile은 다양한 그래픽 정보의 전송을 위하여 적절한 요소를 제공할 수 있어야 하며, 또한 Metafile의 입력과 출력 기능이 GKS에서 사용 가능하여야 한다. Metafile의 구조는 그림B.8과 같으며 CGM은 코딩 방법에 따라 Character encoding, Binary encoding, Clear-text encoding의 3가지로 구분할 수 있다.

Character encoding은 Metafile을 바이트 단위로 부호화하는 방법으로 Metafile의 크기가 작고 문자 지향이므로 메모리가 절약되고 컴퓨터 시스템간의 전송에 적합하다. Binary encoding은 각 요소를 식별자와 그에 따른 데이터 부분으로 부호화함으로써 Metafile을 만들고 해독하는 것이 쉽기 때문에 Metafile을 처리할 때 오버헤드가 적으며, 컴퓨터

시스템간의 교환에도 적합하다. Clear-text encoding은 Metafile의 내용이 사용자가 읽고 쉽게 이해할 수 있도록 문장 형식으로 되어 있어서 편집하기가 쉽다. 반면에 Metafile의 크기가 커지고 처리 속도가 떨어진다.



그림B.7 Graphic Model



그림B.8 Metafile의 구조

B.3.3 FAX Coding

CCITT는 T-시리즈 권고안에서 팩시밀리 장치 및 전송 방식에 관하여 상세한 표준을 마련하고 있다. 1980년 디지털 팩시밀리에 관한 G3 권고안이 채택된 이후 G3 기종의 팩시밀리가 본격 등장하였으며, 또한 디지털망에 접속되어 사용되는 고속 전송 방식의 G4 팩시밀리에 관한 권고안이 1984년 제정되어 향후 신세대 팩시밀리 보급의 기틀을 마련하였다.

G3 권고안은 원고의 화상 정보를 디지털 신호화하여 redundancy를 제거하여 데이터를 감축시키는 방식이 권고되고 있다. 전송 방식으로는 고속 전송이 가능한 PSK(Phase Shift Keying) 또는 QAM(Quadrature Amplitude Modulation) 변조 방식을 채택하였다. G3 기종은 동종 또는 이기종간의 원활한 통신을 위하여 다소 복잡한 프로토콜 기능을 처리하며 이에 자동 다이얼 및 자동 수신 기능을 포함하고 있다.

G4 기종에 관한 권고안은 고속 디지털망에서 사용되기 위한 고속 전송 기능(48kbps, 56kbps, 64kbps)과 고해상도(200, 300, 400dpi) 및 오류 정정 기능에 관한 권고를 포함하고 있다. 또한 혼합 모드 및 텔레마틱 서비스 기능 등에 따라 class 1, class 2, class

3의 세가지 기종으로 구분되며, class 1기종은 고휘상도, 고속 데이터 전송, MMR 방식의 데이터 감축을 주기능으로 한다. Class 2 기종은 텔리텍스 및 혼합 모드 수신 기능이 추가되며, class 3 기종은 텔리텍스 및 혼합 모드 송수신 기능을 포함한다.

B.3.3.1 화상 데이터 압축 기술

화상 정보는 CCD 카메라나 이미지 스캐너, 팩시밀리를 이용하여 얻을 수 있으며, 문자 정보에 비해 정보량이 방대하므로 화상 정보를 압축 및 확장하는 부호화가 필요하다.

화상의 부호화 방식은 비트열로 처리하는 비트맵 방식과 G3 및 G4 팩시밀리 부호화 방식이 있으며, 팩시밀리 부호화 방식은 Huffman 부호를 사용한다. Huffman 부호는 원고의 흑, 백의 runlength를 확률 분포에 따라 가변 runlength 부호로 구성한 것으로 화상 데이터의 압축 redundancy를 제거한다. 이와 같은 방식은 다시 1차원 부호화 방식과 2차원 부호화 방식으로 나뉘어지며, 1차원 부호화 방식은 MH(Modified Huffman) 방식이 사용된다. MH 방식은 Huffman 부호를 실현용 부호로 수정한 것으로서 0-63 runlength를 개개의 부호로 할당한 terminating 부호와 64의 배수로 1728까지의 runlength를 갖는 make-up 부호로 구성함으로써 64개 이상의 runlength를 갖는 경우 부호화는 terminating 부호와 make-up 부호의 조합형 형태로 표시되며 각 단위로 부호화된 마지막에 EOL(End Of Line) 부호가 삽입되어 전송된다. 2차원 부호화 방식은 MR (Modified Relative element address designate) 방식으로 권고되고 있으나 전송 오류가 발생한 경우 다음 라인의 복호화에 영향을 미치므로 전송로가 좋지 못한 경우 사용이 어렵다. 이 방식은 고휘상도에 매우 효과적이다.

B.3.4 PDL(Page Description Language)

ODA에 따라 작성되고 formatting된 문서를 스크린이나 종기와 같은 presentation 매체에 imaging하는 것은 출력 장치에 종속적이다. 이때 각종 문서의 출력을 장치에 종속적으로 구현한다면 문서의 작성, 처리 및 출력에 이르는 일련의 과정이 사무 환경에 미치는 영향이 매우 클 것이다. 이러한 관점에서 장치 독립적으로 구현이 가능하도록 한 것이 PDL을 장착한 LBP가 있다.

현재 LBP는 고휘상도 출력 장치로서 가장 많이 사용되고 있는데 그 이유는 ODA에서 문서를 정의하는 것처럼 장치 독립적으로 페이지를 기술할 수 있는 언어, 즉 PDL을

사용하여 다양한 문서의 출력을 제어할 수 있기 때문이다. 가장 널리 알려진 PDL로서는 Adobe 시스템의 PostScript와 Xerox Corporation의 Interpress 그리고 Imagen사의 DDL이 있는데, 대부분의 LBP가 PDL로서 PostScript를 채택하며 이는 사실상 산업계의 표준으로 되고 있다.

PostScript는 인터프리트식 언어로서 장치 독립적으로 작성되었으며, 문자, 도형, 흑백 화상, gray scale, 칼라로 구성된 페이지를 작성할 수 있다. PostScript 문자는 폰트명과 그래픽 오브젝트를 지정하여 발생시키며, PostScript 프로그램에서 정규화, 파라미터화된 형식으로 그래픽 오브젝트를 지정하고 적절한 변환을 거쳐 페이지상에 이미지를 발생시킨다. PostScript 언어는 7비트 ASCII 형태로 읽을 수 있도록 되어 있으며, 주요 특징으로는 스택 지향 언어, postfix 표현, 페이지 단위 처리, painting operation에 의한 출력 등을 들 수 있다.

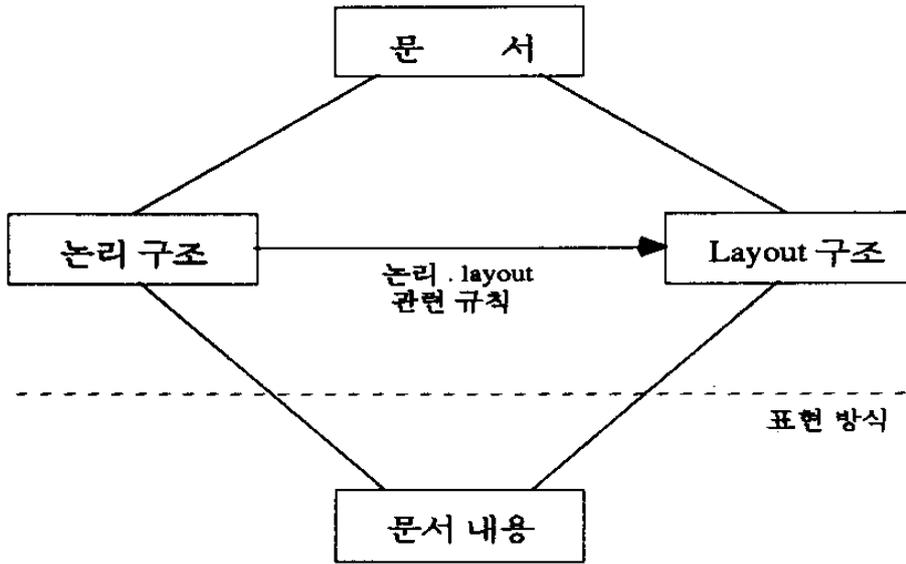
B.3.5 ODA/ODIF

문서 표현의 대표적인 모델인 ODA는 CCITT T73의 선언적 문서 배치 구조 개념과 Interscript 문서 개념을 이용하고 문서의 논리 구조 개념을 응용하여 ISO에 의해 표준으로 정의되었다. 그리하여 ODA는 이기종간의, 서로 다른 문서 제작 시스템간의 문서 교환 작업을 가능하게 하였으며, 문자, 영상 등이 통합된 진정한 의미에서의 멀티미디어 통신의 기초가 되었다.

B.3.5.1 문서 구조

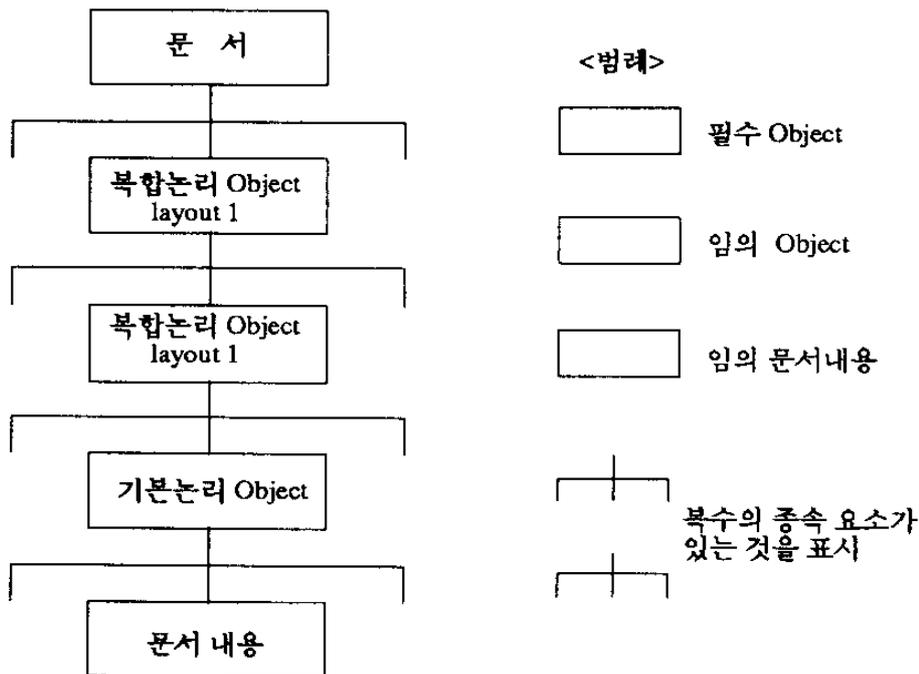
ODA에서는 취급 대상 문서를 사무실에서 작성하는 일반 문서로 한정시켜 문자, 그래픽, 화상, 그림과 같은 질이 다른 정보가 섞여 있는 문서 구조를 정의하고 있으며, ODA 확장 부분에는 디지털화된 음향 등도 포함된다. 이 ODA에서 정의한 문서를 교환하기 위해서 구조 및 data stream 형식으로 정의한 것을 ODIF(Office Document Interchange Format)이라 한다.

ODA 문서는 그림B.9와 같이 문서내용, 논리구조, 배치구조 및 관련 규칙으로 구성된다. 논리구조는 장, 절, 각주와 같은 논리 오브젝트와 각각의 논리 오브젝트간의 관계를 정의한 것이다. 배치구조는 문서내용을 물리적으로 배치하기 위한 구조이며, 문서내용은 문서 그 자체를 말한다.



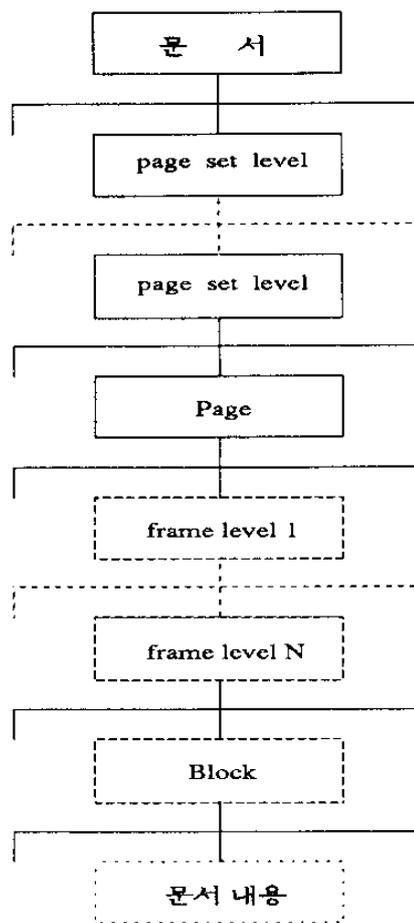
그림B.9 문서 구조

문서 논리구조의 구성 요소는 장, 절과 같은 복합 논리 오브젝트와 문장 및 그림과 같은 기본논리 오브젝트 및 그 문서 내용으로 구성된다. 복합 논리 오브젝트는 적용 분야에 따라 정해지며, 기본논리 오브젝트에 대응하는 문서내용에는 하나의 표현 형식(문자, 그래픽, 화상)밖에 포함되지 않는다. 문서 논리구조는 그림B.10과 같다.



그림B.10 논리 구조

문서 배치구조는 그림B.11과 같이 페이지 세트, 페이지, 프레임, 블록 및 그 내용으로 구성된다. 페이지 세트는 페이지의 집합을 식별하기 위한 것이고, 페이지는 문서의 내용부의 위치 및 표현 형식을 정하기 위한 직사각형 영역이다. 프레임은 페이지내의 배치 경계를 결정하는 것이고, 논리 및 배치 관련 규칙을 제어하는데 기본이 되며, 문서내용을 블록내에 편집 및 할당할 때 블록의 범위를 제한한다. 블록은 문서내용부를 포함하는 기본 단위이며, 블록내에는 한 종류의 표현 형식밖에는 허용되지 않는다.



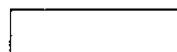
<범례>



필수 Object



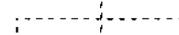
임의 Object



임의 문서내용



복수의 종속 요소가 있는 것을 표시



그림B.11 배치 구조

B.3.5.2 문서 내용구조

현재 ODA에서 정의하고 있는 문서내용의 종류에는 문자, 화상, 기하형 그래픽의 세가지가 있다. 또한 수정 가능 여부에 따라 수정 가능 형태(Processable Form), 배치완료 형태(Formatted Form), 수정가능 및 배치완료 형태(Formatted Processable Form)의 세가지로 구분된다. 이와 함께 문서내용을 구성하는 요소들의 표현 방법, 표현 속성, 영상화 방법도 문서내용을 결정하는 요인이 된다.

Formatted Form은 문서가 작성자의 의도대로 수신자에게 보이도록 하는 양식으로 문서 레이아웃과 표현에 관한 정보를 지정한다. 이 문서 양식의 단점은 제어 정보가 문서내에 나타나 있지 않기 때문에 수정 및 변경이 어렵다는 점이다.

Processable Form은 논리구조를 지정하고 있기 때문에 문서가 전달되었을 때 수신자가 작성자의 의도에 따라 편집, 재형식화 등의 처리를 할 수 있다. 이 양식은 문서 Profile, 특정 논리구조, 공통 논리구조 및 공통 레이아웃 구조를 지정한다. 따라서 이러한 양식의 문서는 쉽게 수정 변경될 수 있으며, 실제 레이아웃은 문서가 format될 때에만 결정된다.

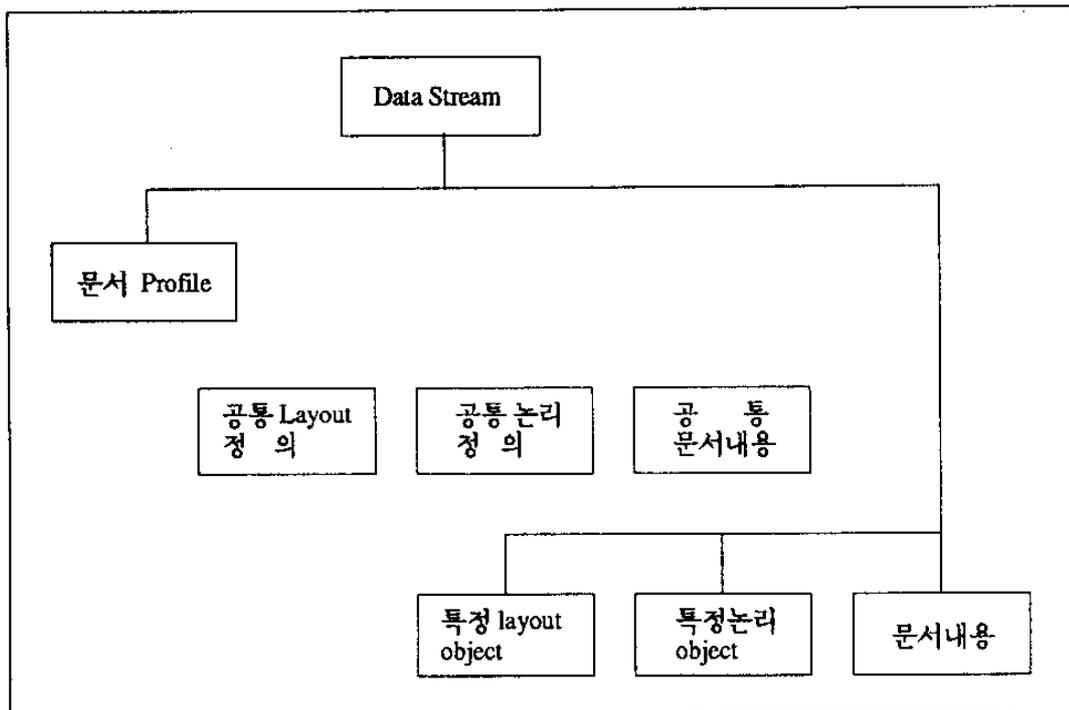
Formatted-processable form은 하나의 문서에 위의 2가지 양식을 통합한 형태이다. 즉, 논리구조와 레이아웃 구조를 모두 필요로 한다. 이 양식은 문서 Profile, 특정 논리구조 및 공통 레이아웃 구조를 포함한다. 문자형 내용구조는 문자형 내용을 일련의 문자, 빈칸, 그리고 제어문자로 구분하고, 사진형 내용구조는 2차원의 pixel의 집합으로 표현된다. 기하형 그래픽 내용구조는 선, 원, 호 등과 같은 요소, 그래픽, 차트 등을 ODA 문서에 포함시킬 수 있는 기능을 제공한다. 기하형 그래픽 내용구조의 목적은 기존의 그래픽 응용 패키지와 동일한 방법으로 이 기능을 제공하는 것이며, 기존의 그림을 ODA 문서에 통합시킬 수 있게 하는 것이다.

B.3.5.3 ODIF

ODIF는 ISO 8613에 따라 구조화된 문서를 교환하기 위하여 사용되는 데이터 스트림의 형식을 정의하고 구조화된 문서의 오브젝트, 내용, 스타일의 부호화를 정의하며, 교환된 문서에서 나타날 수 있는 성분의 표현을 정의한다.

데이터 스트림의 구조는 일련의 디스크립터와 텍스트 유닛으로 구성되어 있으며, 디스크립터는 문서 프로파일, 오브젝트 클래스, 오브젝트의 속성을 나타내는 기본 데이터

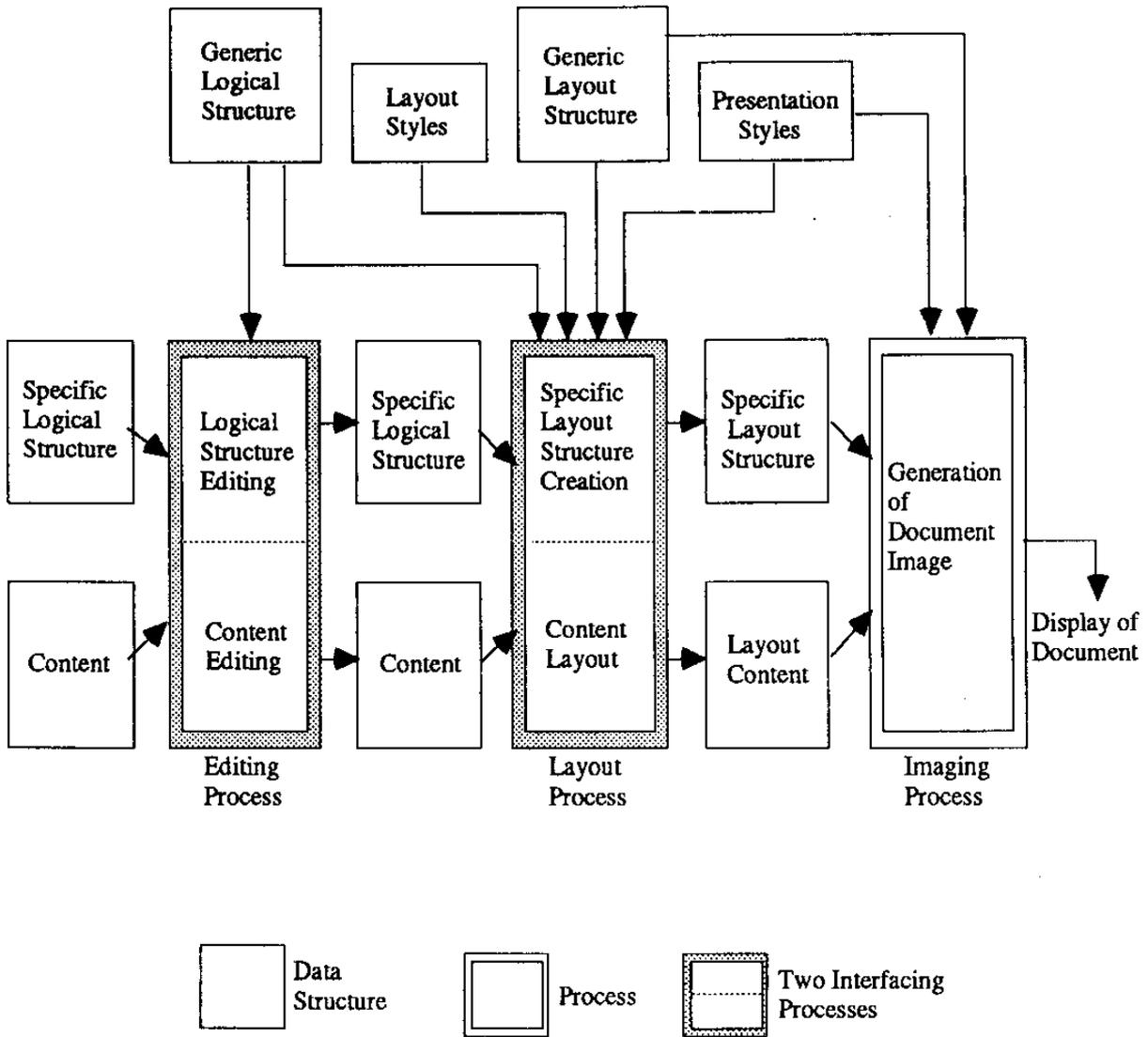
항목과 서브데이터 항목으로 이루어지는 복합 데이터 항목들이다. 텍스트 유닛은 content 부분을 나타내는데 이것의 속성을 표현하는 하나 이상의 기본 데이터 항목 세트로 구성된다. 텍스트 유닛은 교환 형식에 따라 데이터 스트림상에 나타나는 순서가 다르다. 데이터 스트림의 구조는 그림B.12와 같다.



그림B.12 데이터 스트림의 구조

B.3.5.4 문서처리 모델(Document Processing Model)

ODA 문서 표현모델에서 주요 부분의 하나는 사용자가 이 ODA를 통해서 어떻게 문서를 제작하고 수정하며 영상화 또는 인쇄하느냐 하는 것인데, 이는 PDA의 문서처리 모델에 의하여 그 작업 내용과 방법이 결정된다. ODA의 문서처리 모델은 크게 편집과정, 배치과정, 그리고 영상화 과정으로 구분되며 그 관계는 그림B.12와 같다. 편집 과정은 문서작성, 수정을 위한 작업으로 문서표현을 위한 여러 객체들을 다루는 과정으로 문서구조 편집, 문서내용 편집, 스타일 편집 등이 이 과정이다. 배치과정은 편집이 끝난 문서의 구체적인 배치구조를 결정하는 작업이며, 영상화 과정은 배치과정의 결과를 특정 표현기기에 맞게 고쳐 최종 영상화 작업을 하는 과정이다.



그림B.13 문서 처리 모델

B.4 멀티미디어 MHS의 설계

B.4.1 멀티미디어 시스템 환경

기존의 멀티미디어 시스템들은 주로 값비싼 워크스테이션과 I/O 장비를 이용하였기 때문에 일반적으로 널리 사용될 수 없었다. 그러나 PC의 성능이 향상되고 PC용 UNIX의 등장으로 PC를 개인용에서 Host의 기능까지 다양하게 사용할 수 있게 되었다. 또한 저가격 I/O 장비들의 출현으로 다양한 매체를 PC와 I/F하기 수월해지고 PC의 보급율도 가장 높으므로 멀티미디어 시스템을 구현하기 위한 주요 시스템으로 PC를 이용하는 것이 유리하게 되었다. 계속적인 증가가 예상되는 PC 환경에서 멀티미디어 처리 기능을 구현함으로써 사용자에게 보다 편리한 작업 환경을 제공할 수 있을 것이다.

멀티미디어를 처리하기 위해 PC는 다음과 같은 요구 사항을 만족해야 한다.

- 윈도우 기능이 제공되는 디스플레이가 요구된다.
- 그래픽 인터페이스를 위한 키보드와 마우스 같은 장비가 요구된다.
- 매체 처리를 위해 Image Scanner, Video Camera, Laser Printer 등이 요구되며, 장비에 대한 인터페이스가 요구된다.
- 멀티미디어 메시지는 50 - 250kbyte의 대량 정보를 가지므로 고속 네트워크 환경이 요구되며, 이에 대한 고속 어댑터가 요구된다.

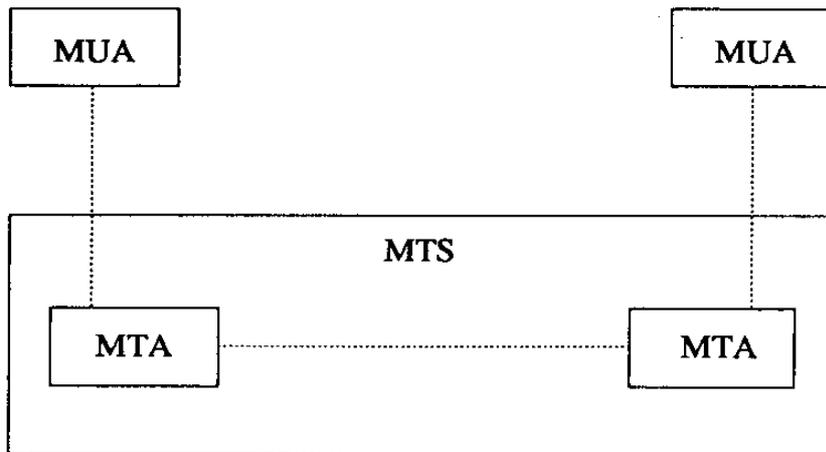
본 연구에서 멀티미디어를 처리하는데 있어서 사용하는 매체는 문자, 그래픽스, 화상만을 고려한다. 각각의 매체에 대한 처리는 표준안을 따르며 다음과 같다.

- 문자는 ASCII 코드를 사용한다.
- 그래픽스는 전송과 저장을 위해 CGM을 사용하며, GKS를 이용하여 그래픽스를 작성하는 것을 원칙으로 한다.
- 화상 데이터는 현재 널리 사용되고 있는 G3 FAX 코딩을 이용한다.
- 멀티미디어로 작성된 문서는 ODA 구조를 이용한 계층적인 구조를 가진다.

B.4.2 멀티미디어 시스템 모델

멀티미디어 시스템은 X.400 모델을 기초로 개방형 분산 구조로 설계하였다. 멀티미디어 MHS에서 요구되는 기능은 크게 정보의 store-and-forward 방식의 메시지 전송과 멀티미디어 정보의 처리 및 출력으로 나눌 수 있다. 그러나 기존의 MHS

모델은 구조화된 문서를 처리할 수 없기 때문에 멀티미디어 문서를 처리하기 위해서는 새로운 기능 모델이 요구된다. 따라서 본 연구에서는 멀티미디어 시스템의 기능 모델을 그림B.14와 같이 설계하였으며, X.400 MHS 모델의 MTA와 새로이 제안하는 MUA (Multimedia User Agent)로 구성된다. 여기서 MTA는 메시지에 대한 전송 관리를 담당하고 있으며, MUA는 멀티미디어로 작성된 문서를 처리하는 기능을 가진다.



그림B.14 멀티미디어 MHS 기능 모델

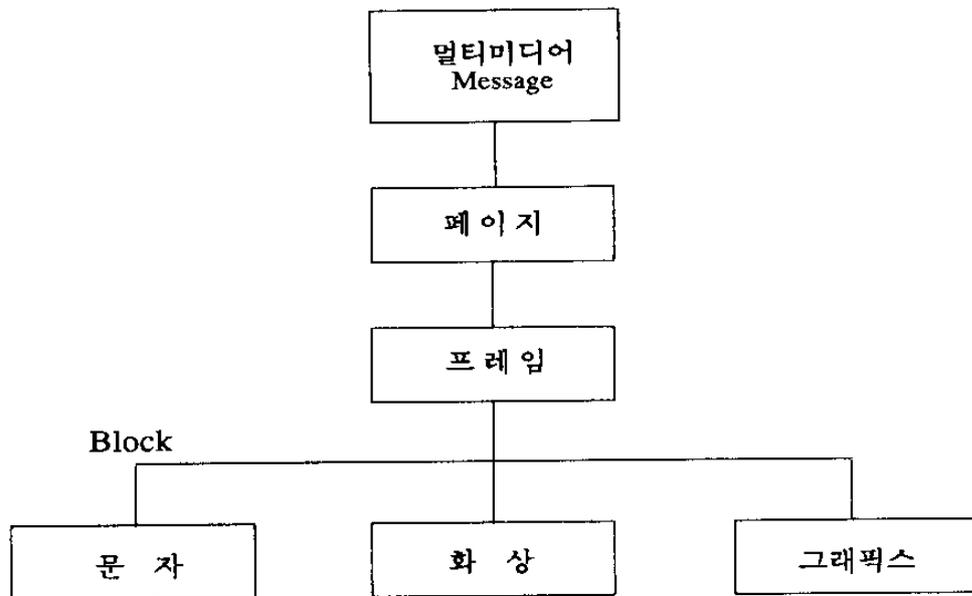
멀티미디어 시스템을 통해서 메시지를 전송하고자 하는 사용자는 우선 MUA와의 사용자 인터페이스를 이용하여 멀티미디어 메시지를 작성한다. 작성된 메시지를 송신측 MUA가 수신측 MUA로 전달하기 위해서 상호 인식할 수 있는 적절한 전송 구문으로 변환시켜 송신측 MTA로 전달하며, 다시 송신측 MTA는 이 메시지를 수신측 MTA로 제출하고 수신측 MTA는 이를 수신측 MUA로 배달한다. 여기서 MTA의 기능은 MUA에 대한 메시지의 제출과 배달, 연결된 다른 MTA로의 메시지 전송을 처리하는 것으로 메시지의 내용에는 관여하지 않는다. 수신측 MUA로 배달된 메시지는 수신자가 그 메시지를 보고자 할 때 수신측 사용자 인터페이스를 통해 수신측의 컴퓨터로 출력하게 된다.

B.4.3 멀티미디어 메시지의 구조

멀티미디어 메시지에 대한 처음 제안이 ARPANET에서 발표된 이후로 여러 문서 구조안이 발표되었는데, 본 연구에서는 ODA를 이용하여 문서를 표현하고자 한다.

따라서 문서를 작성시 ODA의 문서구조 형태로 자동적으로 구조화되도록 설계하여야 한다. ODA에서 제안한 문서구조는 앞에서 언급한 바와 같이 Formatted, Processable, Formatted & Processable의 3가지 형태를 가지고 있으나, 실제 MHS의 전송 문서 구조로는 Formatted 형태가 적절하다. 이에 따라 멀티미디어 메시지 구조는 ODA의 배치구조를 이용하여 그림B.15와 같이 하나 이상의 페이지로 구성하였다.

페이지는 메시지 내용부의 위치 및 표현 형식을 정하기 위한 직사각형 영역으로 프레임과 블록으로 구성된다. 프레임은 페이지내의 배치 경계를 결정하는 것으로 여러 개의 블록을 포함하며, 블록은 실제 메시지의 내용을 포함하는 기본 단위로 한 블록 내에는 한 종류의 표현 양식(문자, 그래픽스, 화상)을 포함한다.



그림B.15 멀티미디어 메시지 구조

B.4.4 멀티미디어 메시지 프로토콜

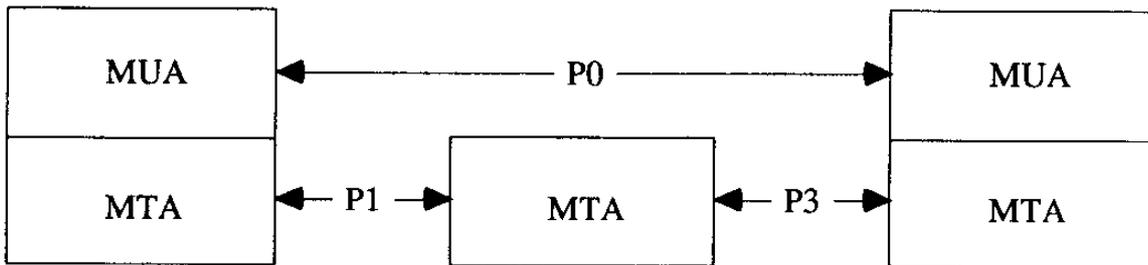
B.4.4.1 PO-IPM

일반적으로 메시지는 라우팅과 관련된 정보를 포함하는 Envelope와 전송하고자 하는 실제 정보를 나타내는 내용(Content)으로 이루어지며, 다시 내용은 표제(Heading)와

본체(Body)로 구성된다. X.400의 본체는 구조화된 문서를 처리할 수 없으므로 멀티미디어 메시지를 전송하기 위해서는 MHS에 구조화된 문서를 처리할 수 있는 새로운 본체 형태가 요구된다.

구조화된 문서는 사무 환경에서 교환되는 대부분의 정보의 구조 및 출력 양식을 표현하는데 적절하다. 만약 이러한 문서를 MHS의 P2 프로토콜로 전송하고자 한다면 IPM의 본체에 구조화된 문서를 포함시켜야 한다. 그러나 이 경우 IPM 표제와 IPM 본체에 포함시킨 구조화된 문서에 중복되는 정보가 있게 되므로 적절하지 않다. 따라서 구조화된 문서를 전송하기 위해서는 이를 처리할 수 있는 적절한 방법이 요구된다. 즉, 구조화된 문서를 IPM의 본체에 넣지 않고 그 문서 자체를 하나의 IPM으로 두고 이를 P0-IPM이라 새로이 정의한다.

그림B.16은 멀티미디어 MHS의 계층적 프로토콜 모델이다. 여기서 MTA-MTA간은 기존의 P1 프로토콜을 이용하며, MUA-MUA간은 새로운 프로토콜 P0를 정의한다. P0-IPM은 P2-IPM과 같이 MTA를 통해서 P1 프로토콜로 전송될 수 있다. 이 경우 P1 프로토콜의 내용 타입에 P2 외에 P0를 추가하여 새로운 IPM이라는 것을 MTA에게 알려주어야 한다.



그림B.16 멀티미디어 MHS 프로토콜

B.4.4.2 P0 프로토콜

멀티미디어 메시지 전송 구문(Multimedia Message Protocol Data Unit)은 Document-Profile-Descriptor, Layout-Object-Descriptor, Text-Unit로 구성 설계하였으며, 이를 ASN.1으로 기술하였다. Document-Profile-Descriptor는 작성한 멀티미디어 메시지에 대한

전반적인 정보를 포함한다. 즉, 메시지 작성자, 작성 시간, 메시지의 주제와 메시지 구성에 대한 정보를 가지고 있다.

Layout-Object-Descriptor는 배치 대상 타입과 본체로 되어 있으며, 배치 대상 타입은 설명자가 페이지, 프레임, 블록인가를 표시하고, 본체는 대상의 위치 속성과 블록일 경우 블록내의 내용의 타입을 표시한다. 대상의 위치 속성은 기준 위치(position)와 크기(dimension)로 구성되는데, 위치 속성은 대상의 시작점이고 크기 정보는 그 대상의 수평축과 수직축에 대한 범위를 기술하며, 위치값은 정수로 표현한다.

Text-Unit는 블록내의 내용과 이에 대한 속성을 표시한다. 내용에 대한 코딩 속성을 표시함으로써 내용의 정보가 어떤 코딩 타입으로 구성되었는가를 알려준다.

B.5 멀티미디어 에디터 설계

멀티미디어 에디터는 문자, 화상, 그래픽 등으로 구성된 멀티미디어 문서를 작성하고자 하는 사용자를 위하여 멀티미디어 문서의 논리구조 및 배치구조를 생성하고 편집할 수 있는 도구이다. 이러한 멀티미디어 에디터를 구현하는데 있어서 중요한 사항으로는 user-friendliness가 있으며, 또한 이러한 멀티미디어 에디터가 생성하고 편집하는 문서가 앞에서 설계한 문서구조와 일치하는 구조를 가져야 할 것이다.

B.5.1 설계 방침

본 연구에서는 멀티미디어 에디터를 구현하기 위해서 다음과 같은 설계 지침을 따른다.

- 에디터에서 생성된 문서는 계층적 구조를 가져야 한다.
- 사용이 간편해야 한다. 즉, WYSIWYG 인터페이스를 제공해야 하며, user-friendly하게 편집이 가능하게 하기 위해 간단한 키조작 및 마우스 입력이 가능해야 한다.
- 한 화면에서 모든 매체를 포함하는 문서를 수정할 수 있어야 하며, 편집과 동시에 화면상에 반영되어야 한다.
- 각 블록의 배치, 수정 및 내용 변경이 용이해야 한다.
- 최하위 계층의 오브젝트인 블록에는 한가지 매체만을 포함하며 다른 블록과 서로 겹칠 수 없다.

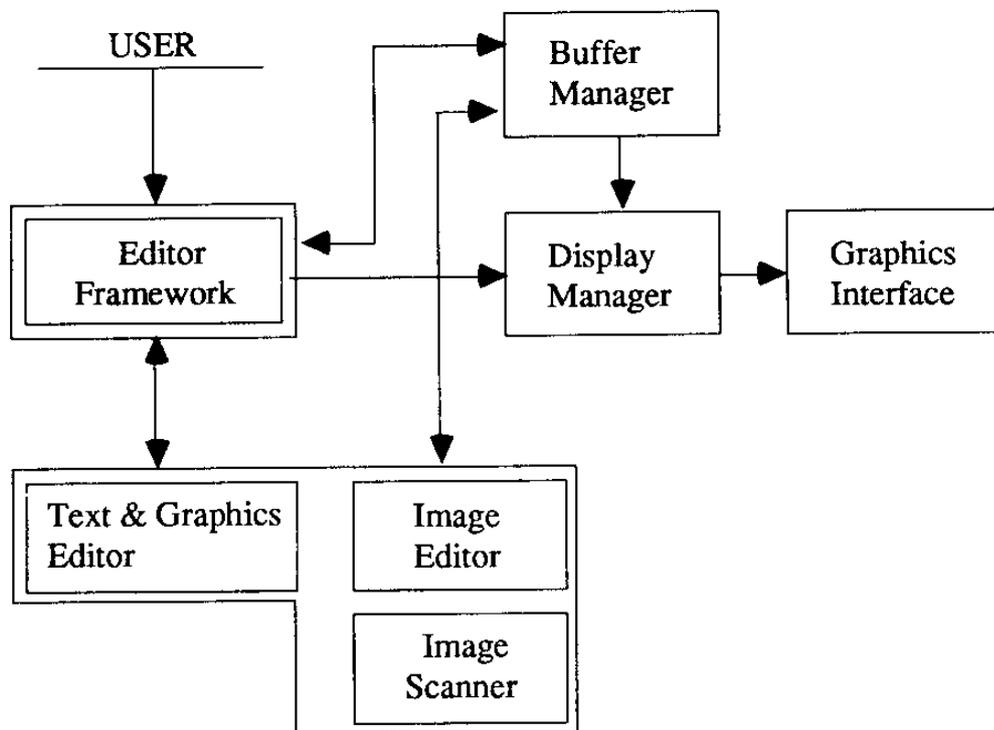
B.5.2 멀티미디어 에디터 구조

멀티미디어 에디터는 사용자와 에디터내의 각 모듈간의 인터페이스를 관장하는 에디터 프레임워크(Editor Framework), 텍스트, 그래픽 및 화상 등의 각 매체를 편집하는 매체별 에디터 모듈, 문서구조가 격납된 버퍼를 관장하는 버퍼 매니저(Buffer Manager) 그리고 화면의 디스플레이를 관장하는 디스플레이 매니저(Display Manager)의 4 모듈로 구성된다. 그림B.17은 이러한 에디터의 구조를 도시한다.

그림B.17에서 에디터내의 모든 모듈에 대한 관장은 에디터 프레임워크에서 하고 있다. 각 모듈은 에디터 프레임워크의 명령을 접수하면 이에 해당하는 동작을 수행하고 결과를 에디터 프레임워크에 보고한 뒤 에디터 프레임워크로부터 다음 명령을 기다린다. 따라서

에디터 프레임워크는 전체 에디팅 기능이 정확히 수행되도록 감시하고 에러가 발생했을 때 이를 복구하는 기능을 가진다.

에디터 프레임워크외의 모듈은 에디터 프레임워크의 명령을 받아 자신에게 주어진 고유의 기능을 수행하며, 수행 도중에 에러가 발생하면 에러의 상태를 에디터 프레임워크에 보고한 뒤 수행을 잠시 중단한다. 프레임워크 이외의 모듈의 모든 동작은 프레임워크에 의해 유발되므로 각 모듈은 프레임워크의 하위에 있다고 할 수 있다.

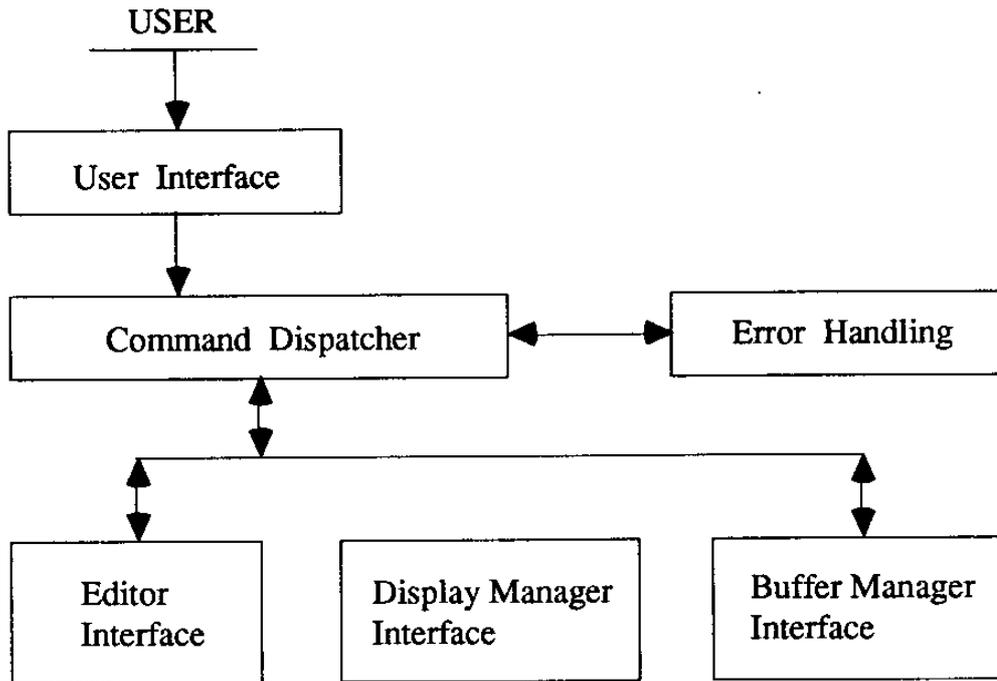


그림B.17 멀티미디어 에디터 구조

B.5.3 에디터 프레임워크

에디터 프레임워크는 사용자와의 인터페이스를 담당하고 각 모듈간의 인터페이스를 총괄 관장한다. 프레임워크의 동작을 유발하는 요인은 크게 두 가지로 볼 수 있는데, 하나는 사용자 요구에 의한 경우이고 다른 하나는 프레임워크 이외의 모듈이 자신의 고유 기능을 수행하다가 에러가 발생했을 때 이를 프레임워크에 보고하는 경우이다. 프레임워크의 대부분의 동작을 유발하는 원인은 사용자의 요구이다. 사용자의 요구는

키보드, 로케이터 등을 통하여 프레임워크에 전해지고, 프레임워크는 이러한 요구를 받으면 사용자의 요구를 분석하여 이를 처리할 수 있는 모듈로 분가하게 된다. 해당 모듈이 수행을 완료하거나 수행 도중 예러가 발생한 경우에는 다시 프레임워크가 개입하여 적절한 조치를 취한다. 그림B.18은 에디터 프레임워크의 구조를 나타낸다.



그림B.18 에디터 프레임워크 구조

B.5.4 버퍼 매니저(Buffer Manager)

버퍼 매니저는 현재 편집되는 문서구조를 관리하고, 파일로부터 이를 읽어들이며, 버퍼내의 문서구조를 파일로 저장하는 등의 역할을 한다. 버퍼 매니저는 프레임워크 및 각 매체 에디터 등으로부터 메시지를 받아서 메시지를 보낸 모듈이 요구하는 동작을 수행한다. 또한 디스플레이 매니저를 호출하여 현재 편집되고 있는 문서구조의 디스플레이되어야 할 부분을 화면에 디스플레이하도록 메시지를 보내는 일도 한다.

버퍼 매니저는 작업 버퍼(Working Buffer)를 운영하고 있는데 이 작업 버퍼에는 현재 다루어지고 있는 페이지내의 텍스트 블록, 그래픽 블록, 화상 블록이 격납된다. 작업 버퍼의 내용은 디스플레이 매니저에 의해 화면에 디스플레이되는 부분이며 각 매체

에디터의 작업 대상 구역이기도 하다. 작업 버퍼의 내용은 프레임워크의 요구에 의해서 주 버퍼에서 읽어들이지거나 주 버퍼로 옮겨진다. 각 매체 에디터는 이 작업 버퍼만을 대상으로 에디팅을 수행하는데, 버퍼 매니저는 각 매체 에디터로부터 메시지를 받아서 작업 버퍼내의 해당하는 내용을 변경한 후에 디스플레이 매니저에 메시지를 보내서 이를 화면에 반영하는 일을 한다. 따라서 디스플레이 매니저가 화면에 반영하는 내용 또한 이 작업 버퍼의 내용에 한정된다.

B.5.5 각 매체 에디터(Individual Media Editor)

각 매체 에디터는 프레임워크로부터 메시지를 받아서 버퍼 매니저내의 작업 버퍼를 변경하도록 버퍼 매니저에게 메시지를 보내는 일을 하게 된다. 텍스트 에디터는 프레임워크에 의해서 수행이 시작되면 키보드로부터 텍스트를 읽어들이 버퍼 매니저에게 보낼 메시지를 작성하고 작성된 메시지를 버퍼 매니저에게 전달함으로써 버퍼 매니저내의 작업 버퍼의 텍스트 내용 부분이 변경되도록 한다.

그래픽 에디터는 로케이터로부터 입력을 받아 버퍼 매니저에게 보낼 메시지를 작성, 전달하여 작업 버퍼의 그래픽 내용부가 변경되도록 한다. 화상 에디터는 이미지 스캐너를 다루는 Kernel내의 Device Driver와 인터페이스하면서 이미지를 입력하여 버퍼 매니저에게 전달하여 작업 버퍼에 저장한다.

B.5.6 디스플레이 매니저(Display Manager)

디스플레이 매니저는 버퍼 매니저로부터 메시지를 받아서 작업 버퍼의 내용을 화면에 디스플레이하는 일을 한다. 디스플레이 매니저는 버퍼 매니저가 관리하는 작업 버퍼의 내용이 변경되어 화면에 이를 반영할 필요가 있는 경우에 버퍼 매니저에 의해서 호출된다. 디스플레이 매니저는 Kernel내의 그래픽 H/W 드라이버와 인터페이스하고 있다. 디스플레이 매니저는 또한 프레임워크로부터 직접 메시지를 받아 화면상에 임의의 텍스트나 그래픽을 디스플레이할 수도 있다. 이 경우는 주로 프레임워크와 사용자간의 인터페이스 과정에서 사용자의 실수를 정정하기 위한 메시지를 디스플레이하거나 또는 각 매체 에디터가 작업 도중 에러를 프레임워크에 알린 경우 이를 사용자가 알아야 할 필요가 있는 경우인데, 이 경우는 프레임워크가 직접 디스플레이 매니저에게 메시지를 보낼 수 있다.

참 고 문 헌

<参考文献>

- [1] Proposed IEEE Standard, "FDDI subnet of a MAN," IEEE P802.6/D14, July 13, 1990.
- [2] "High Speed Optical Data Transfer Network," vol.I, Network Overview, Bellcore ST-ARH-00016, Issue No. 1, Sept. 11, 1989.
- [3] Collard J, "The Physical Layer for the BWN," Proc. of the IFIP TC 6/WG6.4 Second International Workshop on High Speed Local Area Networks II, Liege, Belgim, April 1988, pp. 47-52.
- [4] Hopper, A. et.al., "the Cambridge Fast Ring Networking System," IEEE Trans. on Comp., vol.37, No.10, Oct. 1988, pp. 1214-1223.
- [5] Greves, D. et.al., " The Cambridge Backbone Network," Proc. EFOC/LAN '88, Amsterdam, June 1988.
- [6] Apel, J, et.al., "Implementation techniques for LION," Proc. IFIP TC 6/WG 6.4 Second International Workshop on High Speed Local Area Networks II, lie'ge, Belgim, April 1988, PP. 207-221.
- [7a] American National Standard, "FDDI Token Ring Media Access Control(MAC)," ANSI X3.139, 1987.
- [7b] American National Standard, "FDDI Token Ring Physical Layer Protocol(PHY)," ANSI X3.148, 1988.
- [7c] Draft Proposed American National Standard, "FDDI Physical Layer Medium Dependent(PMD)," ASC X3T9.5, Rev 9, Mar. 1989.
- [7d] Draft Proposed American National Standard, "FDDI Token Ring Station Management(SMT)," ASC X3T9.5, Rev 5, May 1989.

- [8] CCITT Doc. AP IX-142-E, Study Group XVIII - Report R57, July 1988, G707-709.
- [9] Draft Proposed American National Standard, "FDDI Hybrid Ring Control(HRC)," ASC X3T9.5, Rev. 1.C, August 1988.
- 10] IEEE 802.6 Document 802.6-90/43, Minutes, IEEE 802.6 Working Group, Metropolitan Area Network, Denver, Jul. 1990.
- [11] C. Dennis Pegden, "Introduction to SIMAN™ with Version 2.0 Enhancements," Systems Modeling Corporation, Aug. 1984.
- [12] IEEE 802.6 Document 802.6-89, Minutes, IEEE 802.6 Working Group, Metropolitan Area Network, Ft. Lauderdale, FL, Nov. 1989.

주 의

1. 이 보고서는 과학기술처에서 시행한 특정연구 개발사업의 연구보고서 이다.
2. 이 연구개발 내용을 대외적으로 발표할 때에는 반드시 과학기술처에서 시행한 특정연구 개발사업의 연구결과임을 밝혀야 한다.