

IEEE 802.11e MAC 프로토콜 설계 및 구현

The Design and Implementation of IEEE 802.11e MAC Protocol

서덕규, 김도형, 이혁준

광운대학교 컴퓨터공학과

Key Words : 802.11, 802.11e, 무선 MAC 프로토콜, 설계, 구현

목 차

- I. 서론
- II. 802.11e MAC Protocol
 - II.1 EDCA
 - II.2 Block Ack
- III. IEEE 802.11e MAC S/W 구현
 - III.1 IEEE 802.11e MAC S/W 모듈별 동작
- IV. 결론
 - 감사의 글
 - 참고문헌

I. 서론

현재까지 무선 랜은 이동성과 쉽게 네트워크를 구성할 수 있다는 장점을 바탕으로 널리 사용되고 있다. IEEE 표준화 그룹은 802.11/a/b/g 등으로 발전시키며, 물리계층에서의 속도를 최대 54 Mbps까지 높여 왔다. 그에 반해서 MAC(Medium Access Control)에서의 변화는 없었다.

이후에 무선 랜 시장의 확대와 함께 QoS 지원에 대한 요구가 증가하면서 특히, 데이터 외에도 비디오 스트림이나 음성 데이터 등 전송 지연에 민감한 트래픽을 전송하기 위해 802.11e를 표준으로 채택하였으며, 4개의 AC(Access Category) Queue를 가지고, 차별화된 우선순위에 따라 전송권한을 주기 위해 EDCA(Enhanced Distributed Channel Access) 메커니즘을 도입하였다.

최신의 802.11n은 MAC과 물리계층을 발전시킴으로써 최대 600Mbps까지 전송속도를 높이기 위해 연구가 진행 중이다. 따라서 802.11n을 이해하기 위해서는 802.11e MAC에 대한 이해가 선행되어야 하는데, 802.11e의 EDCA와 Block Ack 등 대부분의 기능을 그대로 사용하고 있기 때문이다.

본 논문에서는 IEEE 802.11 표준에서의 Legacy MAC에 대한 SDL(Specification & Description Language)을 참고하여 구현된 Legacy MAC을 바탕

으로 802.11e에서의 EDCA와 Block Ack 기능을 추가 구현하는 과정을 기술하였다.

II. 802.11e MAC Protocol

IEEE 802.11 TG e는 IEEE 802.11의 DCF 및 PCF 프로토콜을 기반으로 하는 HCF(Hybrid Coordination Function)을 정의한다.

HCF는 DCF에 의해 제어되는 CP(Contention Period)와 PCF에 의해 제어되는 CFP(Contention Free Period) 동안 QoS 데이터를 전송할 수 있는 메커니즘으로, 경쟁 기반으로 하는 EDCA(Enhanced Distributed Channel Access)와 채널에 대한 무선 단말의 접근을 제어하는 HCCA(HCF Channel Access)로 구성되며 동시에 legacy MAC과의 호환성을 지원한다.

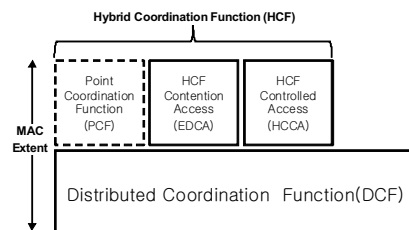


그림 1. MAC architecture

그림 1 은 802.11 DCF 서비스와 함께 802.11 PCF와 802.11e HCF를 지원하는 enhanced MAC 프로토콜의 구조를 보여준다.

802.11 DCF 전송방식은 중앙에서 매체접근을 제어하는 PCF 또는 HCF 전송방식과 함께 동일한 BSS 서비스를 지원 한다. 특히 QBSS에 존재하는 HC(Hybrid Coordinator)는 CP 동안에만 동작하는 DCF나 CFP 동안에만 동작하는 PCF와 달리, 채널이 CP 서비스를 지원하는 동안에도 경쟁 기반인 EDCA와 폴링 기반의 HCCA의 동작을 모두 지원한다.

HCF를 지원하는 QSTA(QoS STA)은 하나 이상의 채널 접근 메커니즘을 사용하여 무선 매체를 통해 전송을 시작하고 프레임 전송을 계속 유지할 수 있는 권한을 얻는다. 이러한 권한을 802.11e에서 TXOP (transmission opportunity)으로 정의되며, 전송의 시작 시점을 기준으로 하여 일정한 시간 동안 이루어진다. 802.11e enhanced MAC 프로토콜에서는 일반적으로 경쟁에 의해 획득된 EDCA TXOP과 controlled channel 접근 방식에 의해 얻어진 HCCA TXOP를 정의하고 있다.

또한 802.11e 프로토콜에서는 여러 번의 Ack 를 송수신 하는 과정을 한번의 Block Ack 프레임을 송수신으로 대체하여 채널의 효율성을 향상시키는 Block Ack 메커니즘을 제공한다.

1. EDCA

기존의 DCF 메커니즘은 모든 STA의 모든 트래픽이 동등한 매체 접근 권한을 가지기 때문에 QoS를 제공하는데 제약이 있다.

802.11e EDCA 프로토콜은 트래픽을 8개의 서로 다른 우선순위(UP : User Priority)로 정의한다.

그리고, 그림 2와 같이 각 UP를 우선순위에 따라 분류된 4개의 access categories (AC)를 정의하고 한다.

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest ↓ Highest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

그림 2. UP-to-AC mappings

EDCA에서 매체 접근 및 전송 권한은 EDCA-TXOP를 획득한 STA에 대해서만 허용된다.

각 AC에 따라 EDCA-TXOP를 다르게 적용되며, TXOP는 AIFS[AC], aCWmin[AC], aCWmax[AC],

TXOP limit와 같은 파라미터들이 QSTA인 경우 dot11EDCATable MIB, QSTA인 경우 dot11QAPEDCA Table에 저장된다.

각 AC는 이 파라미터들을 사용하여 서로 독립적인 AIFS를 갖으며, TXOP 전송을 하게 된다. DCF에서는 모든 트래픽에 대해서 동일한 DIFS를 갖는데 비해 EDCA에서의 AC마다 다른 AIFS를 갖으며, 우선순위에 따라 차별화된 매체접근 권한을 갖도록 한다.

EDCA에서 각 AC가 AIFS와 백오프를 통해서 매체 점유를 획득하게 되면 EDCA-TXOP를 얻을 수 있다. TXOP 동안에는 매체 점유를 획득한 STA이 이 기간 동안 경쟁 없이 다수의 프레임을 전송할 수 있으며, TXOP limit를 초과할 수 없다.

만일, BE, BK 경우와 같이 TXOP limit가 0인 경우에는 단 하나의 프레임 전송만 가능하다.

2. Block Ack

Block Ack 메커니즘은 여러 개의 Ack 를 하나의 프레임으로 통합시켜 보내는 방식으로 Immediate Block Ack 와 Delay Block Ack 두 가지가 있다.

Immediate Block Ack 는 송신자가 SIFS 간격으로 여러 개의 프레임과 Block Ack Request 프레임을 전송하면, 수신자는 그에 대한 응답으로 SIFS 시간 내에 Block Ack 를 전송하는 방식이다.

Delayed Block Ack 는 Immediate 방식과는 달리 수신자가 SIFS 시간 내에 Block Ack 로 응답하는 대신 일반적인 Ack 로 응답한 후, 후에 Block Ack 프레임으로 전송하는 방식이다. 송수신 양측은 Block Ack 메커니즘 사용을 위해 ADDBA Request/Response 를 이용하여 설정과정을 진행한다. 이때 각자의 장비가 지원하는 Block Ack 메커니즘 방식과 블록을 구성할 프레임들의 TID, 수용 가능한 버퍼 크기, 블록 시작 시퀀스 번호 등 Block Ack 메커니즘 사용에 필요한 파라미터 값들을 서로에게 알려준다. Block Ack 메커니즘은 필요치 않거나, 지원되지 않는 환경이라면, 어느 한쪽이 DELBA Request 프레임을 통해 Block Ack 메커니즘을 사용 중단을 상대방에게 알린다. 이러한 과정은 MAC 계층을 관리하는 MLME 모듈에 의해 진행된다.

III. IEEE 802.11e MAC S/W 구현

802.11e MAC S/W는 ANSI C를 이용하여 구현하였으며, SDL 기반으로 구현된 Legacy MAC을 기초로 하고 있다.

각 블록이 concurrent한 동작을 할 수 있도록 posix thread를 사용하였다. 또한, 블록 간 신호 전달은 ANSI C library에서 기본적으로 제공되는 condition variable

을 사용하였다.

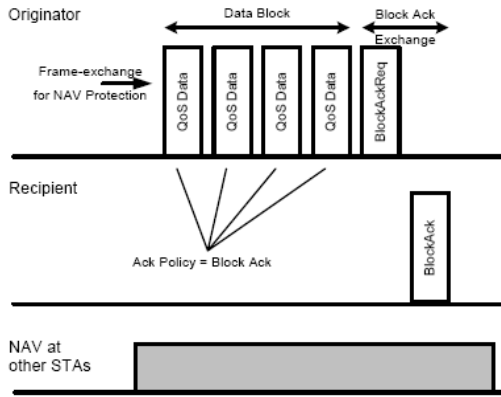


그림 3.Immediate Block Ack

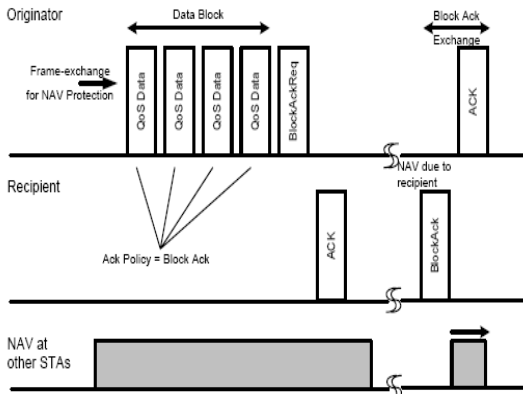


그림 4.Delayed Block Ack

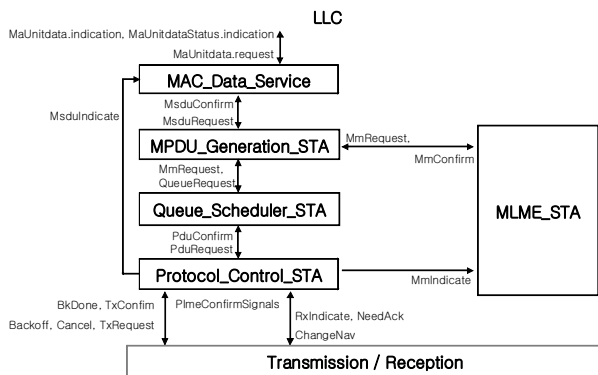


그림 5. 802.11e 설계 모듈

condition variable은 condition 이벤트가 발생하기 전까지 posix thread가 대기하고, 이벤트 발생 시 이후의 동작을 진행하는 library 함수이다.

1. IEEE 802.11e MAC S/W 모듈별 동작

1) MAC_Data_Service

MAC_Data_Service는 LLC로부터 데이터 전송 요청을 받고 올바른 데이터 요청인지 확인하고 데이터에 프

레이م 헤더를 붙여 Protocol_Control_STA 블록으로 전달하며, 수신하는 경우는 Protocol_Control_STA로부터 받은 프레임의 프레임 헤더를 제거하고, 데이터와 목적지 주소, 소스 주소등을 추출하여 LLC로 전달한다.

2) MPDU_Generation_STA

MPDU_Generation_STA는 MSDU_from_LLC로부터 프레임을 받아 필요한 경우, 프레임의 길이에 따라 하나 이상의 MPDU를 생성하는 fragmentation 과정을 수행하고, 생성된 MPDU를 전송하기 Tx_Coordination_sta에 MPDU를 전달한다.

Block Ack 메커니즘 설정을 위한 management 프레임에 대한 요청이 오면 MAC 헤더를 붙여 전달한다.

3) Queue_Scheduler_STA

Queue_Scheduler_STA은 MPDU_Generation_STA로부터 MPDU를 전달받아 MPDU의 TID에 따라서, 대응하는 AC 큐에 전달한다.

각 AC큐에서는 큐 내부의 MPDU 개수가 하나 이상인 경우, TXOP를 획득하기 위하여 독립적으로 Protocol_Control_STA으로 medium 획득을 위해 전송 요청을 한다.

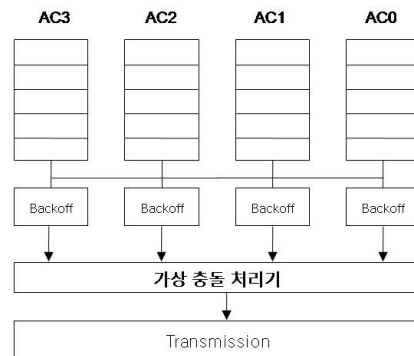


그림 6. EDCA 동작

4) Protocol_Control_STA

Protocol_Control_STA는 Queue_Scheduler_STA로부터 MPDU 전송 요청을 받아 QoS를 지원하는 MPDU에 대해서 EDCA 과정을 수행한다.

각 AC큐가 medium에 접근하기 위해 백오프를 시도하고, 만일 내부적으로 백오프 타임이 같아 충돌이 발생한 경우, 우선순위가 높은 AC의 MPDU에 대해서 전송권한을 부여하는 가상충돌 처리를 한다.(그림 4)

전송권한을 획득한 AC는 Transmission 블록으로 전송 요청을 한다. 또한, MPDU가 TXOP를 획득하기 위해 전송 트래픽 길이에 따라 RTS 전송 여부를 결정하고, 전송 실패한 프레임에 대해 재전송 기능을 수행한다.

수신하는 경우에는, 수신한 프레임이 RTS와 DATA

인 경우 CTS와 ACK 프레임 전송 요청 한다.

또한 Block Ack 메커니즘을 사용하는 송신자는 수신한 Block Ack 프레임 확인을 통해 전송이 성공여부를 판단하고 전송이 실패된 프레임에 대해서는 재전송을 한다. 이 모듈에서는 이러한 과정을 원활히 수행하기 위해 송/수신 프레임들을 기록하는 스코어보드와 실제 프레임을 임시로 저장하는 버퍼를 구현하였다. 스코어보드와 버퍼는 각 송/수신 스테이션의 주소와 TID 에 따라 프레임의 정보를 기록/저장한다. 이때 구현한 방식은 고정적인 메모리 공간을 사용하는 배열 방식이 아닌 각 환경에 따라 적용될 수 있으며 메모리 공간 사용을 최소화 시키는 Linked-List 방식으로 구현하였다.

5) MLME

MLME 는 Block ACK 메커니즘 설정을 위한 management 프레임 생성과 전송요청 그리고 프레임 수신에 대한 처리 동작을 수행한다.

V. 결론

본 논문에서는 SDL로 구현된 Legacy MAC을 바탕으로 한 802.11e MAC을 posix thread를 이용하여 C로 구현하였다. 직접 MAC을 구현함으로써 802.11e가 어떻게 동작하는지 확인할 수 있었고, 범용적인 언어인 C를 사용함으로써 이식성이 높다는 이점을 갖게 되었다.

802.11의 표준안에는 Legacy MAC에 대해서 SDL로 정의 되어 있기 때문에 구현상에 용이하였지만, 802.11e의 구현은, 표준안을 바탕으로 구현하였으나, 다소 추상적인 표현에 대해서는 구체화 하는데 어려움이 있었다. 또한 Legacy MAC을 기반으로 구현하였기 때문에 802.11e의 기능을 추가할 때, 기존의 SDL과 달라지는 부분에 대해서 어떻게 해결해야 하는지에 대해 많은 시간이 걸렸던 부분이다.

현재는 802.11e 구현의 안정화 작업과 함께 802.11n을 구현하기 위해 연구와 구현을 진행하고 있다.

감사의 글

본 논문은 서울시 산학연 협력사업(10560) 지원에 의하여 연구되었음.

참고문헌

[1]IEEE WG, "Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications." Revision of IEEE Std 802.11-1999, 2006