

# 통합된 구현 방식을 이용한 IEEE 802.11n MAC의 설계

준회원 정 철 호\*, 이 선 기\*, 종신회원 정 윤 호\*\*, 정회원 김 재 석\*\*\*

## Implementation of IEEE 802.11n MAC using Design Methodology

Chulho Chung\*, Sunkee Lee\* *Associate Members*, Yunho Jung\*\* *Lifelong Member*,  
Jaeseok Kim\*\*\* *Regular Member*

### 요 약

본 논문에서는 MAC-SAP (Medium Access Control-Service Access Point)에서 하향 100Mbps 이상의 처리 속도를 요구하는 IEEE 802.11n MAC의 구현을 위한 새로운 방안을 제안하고, 이를 이용해 설계된 MAC의 구조를 제시한다. 제안된 방식을 이용하면, 기존에 MAC 프로토콜의 검증 및 네트워크 기반 시뮬레이션 과정과 소프트웨어 및 하드웨어의 설계 및 검증 과정이 분리되어 진행되던 방식과는 다르게 하나의 통합된 코드를 사용하여 네트워크 시뮬레이션과 시스템의 설계가 밀접한 연관성을 가질 수 있다.

제안된 방식을 통해 구현된 IEEE 802.11n MAC은 상위계층 MAC과 하위계층 MAC의 두 부분으로 구성되며, 각각은 표준에 새롭게 추가된 기능의 효율성을 높일 수 있게 그 기능을 구분하여 각각 소프트웨어와 하드웨어로 설계되었다. 설계된 MAC은 ARM 기반의 FPGA 보드를 통해 검증되었다.

**Key Words** : WLAN, MAC, IEEE 802.11n, Design Methodology, Implementation.

### ABSTRACT

In this paper, we propose a design methodology of IEEE 802.11n MAC which aims to achieve the higher throughput of more than 100Mbps in downlink as measured at the MAC-SAP and present the implementation results of MAC using the proposed design methodology.

With our proposed methodology, different from the conventional design flow which has the separate codes for the protocol validation, for the network simulation, and for the system implementation, the unified code can be used for the network simulation and the implementation of software and hardware. Our MAC architecture is partitioned into two parts, Upper-layer MAC and Lower-layer MAC, in order to achieve the high efficiency for the new features of IEEE 802.11n standard. They are implemented in software and hardware respectively. The implemented MAC is tested on ARM based FPGA board.

### I. 서 론

무선 랜 (Wireless Local Area Network, WLAN)

은 이동성과 손쉬운 네트워크 구성이라는 장점을 바탕으로 널리 사용되고 있는 표준 기술 중 하나이다. DVD 재생, HDTV, 네트워크 저장 솔루션 등과 같

\* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었고 (IITA-2008-C1090-0801-0012), CAD Tool은 IDEC으로부터 지원받았음.

\* 연세대학교 전기전자공학과 정보통신용 SoC 설계 연구실 (chulho729@yonsei.ac.kr, yuvjjang@yonsei.ac.kr)

\*\* 한국항공대학교 항공전자 및 정보통신공학부 조교수 (yjung@kau.ac.kr),

\*\*\* 연세대학교 전기전자공학과 교수 (jaekim@yonsei.ac.kr)

논문번호 : KICS2008-10-471, 접수일자 : 2008년 10월 22일, 최종접수일자 : 2009년 3월 6일

이 무선 랜이 적용되는 응용 분야가 다양해짐에 따라 고속의 전송 속도 뿐만 아니라 더 빠른 처리 속도가 요구되고 있다. 이를 위해 IEEE 802.11 표준화 그룹은 IEEE 802.11a/b/g 등 개정안을 통해 최대 54Mbps까지 물리 계층의 전송 속도를 높여왔다<sup>[1]</sup>. 현재 활발히 표준화 작업을 진행 중인 Task Group N (TGn)은 유선에 비해 여전히 느린 통신 속도의 한계를 극복하기 위해 물리 계층의 전송 속도를 최대 600Mbps까지 높일 뿐만 아니라, Medium Access Control-Service Access Point (MAC-SAP)에서 하향 100Mbps 이상의 높은 처리 속도를 지원하는 것을 목표로 차세대 무선 랜 표준화 작업을 진행하고 있다<sup>[2]</sup>. 이와 같은 목표를 달성하기 위해, IEEE 802.11n 표준은 물리 계층과 MAC 계층 전반에 걸쳐 다양한 새로운 기법을 채택하고 있다.

본 논문에서는 더욱 빨라진 전송 속도와 처리 속도를 필요로 하는 IEEE 802.11n 시스템을 위한 MAC의 구현 방안을 제안한다. 기존의 구현 방식에서는 프로토콜의 검증 및 네트워크 시뮬레이션을 위한 코드와 시스템 설계를 위한 소프트웨어 및 하드웨어 코드가 다르기 때문에, 설계를 마치고 시스템을 구성하기 이전에 시스템의 성능을 평가하기 어려운 단점이 존재하였다. 그러나 제안된 구현 방식에서는 통일된 코드를 통해 성능 검증과 설계를 밀접하게 연관시켜 시스템 구성 이전에 시스템의 검증 및 성능 평가가 가능하게 하였다. 제안된 구현 방식을 사용하여 설계된 MAC 구조는 소프트웨어로 처리되는 상위계층 MAC (Upper-layer MAC, UMAC)과 하드웨어로 처리되는 하위계층 MAC (Lower-layer MAC, LMAC)의 두 부분으로 구성된다. 상위계층 MAC과 하위계층 MAC은 IEEE 802.11n 표준이 요구하는 성능을 만족시키고 표준에 새롭게 추가된 기능을 효율적으로 수행할 수 있게 그 기능을 구분하였으며, 각각은 C/C++ 언어와 Verilog HDL을 이용하여 설계 및 검증되었다.

본 논문은 다음과 같이 구성된다. II장에서는 IEEE 802.11n MAC 표준에 새롭게 추가된 집합 프레임 전송 및 Compressed Block Ack에 대해 설명한다. III장에서는 기존의 MAC 구현 방식을 설명하고, 기존 방식의 문제점을 살펴본다. IV장에서는 본 논문에서 제안하는 통합된 시뮬레이션 및 설계 방식에 대해 설명한다. V장에서는 제안된 구현 방식을 이용해 설계된 IEEE 802.11n MAC의 네트워크 시뮬레이션과 시스템 설계의 결과를 제시하며, 마지막으로 VI장에서 본 논문의 결론을 맺는다.

## II. IEEE 802.11n MAC 개요

IEEE 802.11n 표준은 다중 안테나 기술을 비롯한 물리 계층의 다양한 기법을 적용하여 기존 IEEE 802.11a/g에 비해 10배 이상 빠른 최대 600Mbps의 전송 속도를 지원한다. 물리 계층의 빨라진 전송 속도는 데이터의 전송 시간을 단축시켜 프레임의 길이를 짧게 만들며, 고정된 시간을 갖는 프리앰블, 프레임 송수신 과정에서 채널에 접근하기 위한 대기 시간인 Inter-Frame Space (IFS) 등의 프로토콜 오버헤드가 상대적으로 커지는 문제를 야기한다<sup>[3]</sup>. 이를 해결하기 위해 IEEE 802.11n MAC은 집합 프레임 전송 기법 (Frame Aggregation)을 채택하고 있다. 집합 프레임 전송 기법은 다수의 프레임을 하나의 프레임으로 합쳐 보다 긴 프레임을 생성하여 전송시키는 기법으로 Aggregated MAC Service Data Unit (A-MSDU)와 Aggregated MAC Protocol Data Unit (A-MPDU)의 두 종류로 구분된다.

A-MSDU는 동일한 수신 주소와 트래픽 ID를 갖는 다수의 MSDU들을 하나로 합치는 방식이다. A-MSDU는 서브프레임 헤더와 MSDU를 포함하는 서브프레임들로 구성되며, 그 구조는 그림 1과 같다. A-MSDU는 최대 3839바이트 혹은 7935바이트까지 확장될 수 있다.

A-MPDU는 동일한 수신 주소로 향하는 다수의 MPDU를 하나의 PHY Service Data Unit (PSDU)로 합치는 방식이다. A-MPDU는 MPDU 구분자와 MPDU를 포함하는 서브프레임들로 구성된다. A-MPDU는 A-MSDU 기법을 통해 생성된 MPDU를 선택적으로 포함할 수 있으며, 최대 65535 바이트까지 확장 가능하다. 또한, A-MPDU에 포함된 다수의 MPDU에 대한 수신 응답으로 Compressed Block Ack를 사용하며, 최대 64개의 MSDU에 대한 수신 정보를 포함한다. A-MPDU의 구조는 그림 2와 같다.

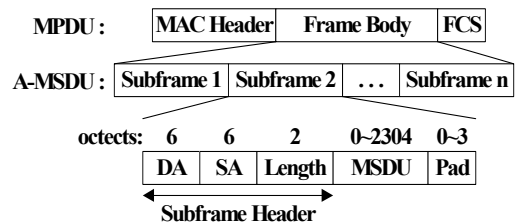


그림 1. A-MSDU 프레임 구조  
Fig 1. A-MSDU Frame Format

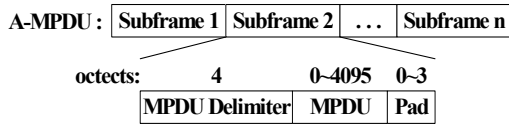


그림 2. A-MPDU 프레임 구조  
Fig 2. A-MPDU Frame Format

### III. 기존의 MAC 구현 방식

기존의 MAC의 구현 과정은 크게 두 가지 단계로 구분할 수 있다. 하나는 프로토콜을 연구하고 성능을 검증하는 과정이고, 다른 하나는 시스템의 사양을 결정하고 결정된 사양에 따라 시스템을 설계하는 과정이다. 기존 방식의 MAC 설계 흐름은 그림 3에 나타내었다.

Specification and Description Language (SDL)은 통신 프로토콜을 기술하고 검증하기 위해 사용되는 언어이다. SDL은 그림을 기반으로 하기 때문에 프로토콜을 추상화하고 검증하기 용이할 뿐 아니라, C/C++와 같은 언어로 변환이 가능해 빠른 설계 접근이 가능하다는 장점이 있다. 그러나 복잡한 구조의 알고리즘을 설계하기 어려우며, 프로토콜의 성능 검증은 지원하지 않는다.

프로토콜의 성능 검증은 NS-2나 OPNET과 같은 네트워크 시뮬레이터를 통해서 이루어진다. 네트워크 시뮬레이터는 다양한 네트워크 프로토콜과 어플리케이션 환경을 제공하기 때문에 설계하고자 하는 프로토콜을 이용하여 네트워크를 구성하고 성능을 분석하기에 용이하다. 일반적으로 네트워크 시뮬레이터에서 사용되는 프로토콜의 소스 코드는 프로토콜의 주요 기능 부분에 대해 단순 모델링 한 것으로 시스템 설계를 위한 코드와는 큰 차이가 있다. 따라서 기존의 MAC 구현 방식에서는, 시스템 설계 사양을 결정된 이후에 소프트웨어 및 하드웨어 코드를 새로 설계해야 한다<sup>[4][7]</sup>.

기존 방식에서는 프로토콜의 성능을 검증한 코드와 시스템 설계에 사용되는 코드가 동일하지 않기 때문에, 설계를 마치고 시스템을 구성하기 이전에 설계될 코드에 대한 성능을 평가하기 어렵다. 이와 같은 문제를 해결하기 위해서, 본 논문에서는 통일된 코드를 이용하여 성능 평가와 시스템 설계가 이루어 질 수 있는 구현 방식을 제안하며, 제안된 MAC 구현 방식은 다음 장에 자세히 설명한다.

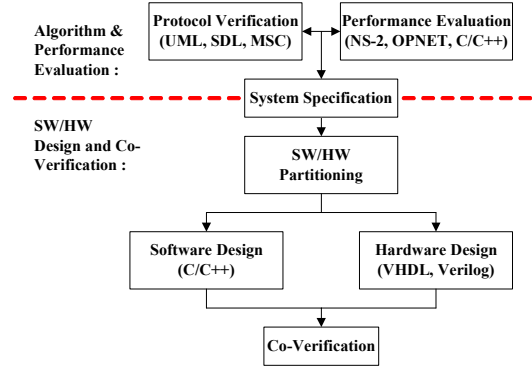


그림 3. 기존 방식의 MAC 설계 흐름도  
Fig 3. Conventional MAC Design Flow

### IV. 제안된 IEEE 802.11n MAC 구현 방식

제안된 MAC 구현 방식에서는 먼저 C/C++ 언어를 이용하여 네트워크 시뮬레이션 및 시스템 설계를 위한 통합된 코드를 설계한다. C/C++ 코드는 SDL로부터 변환된 코드를 이용할 수 있으나, 본 논문에서는 직접 설계한 IEEE 802.11n MAC 코드를 사용하였다. 설계된 코드는 하드웨어로 설계될 기능을 비롯한 MAC 프로토콜의 모든 기능을 포함하며, 네트워크 시뮬레이션 및 시스템 설계 과정에 특화된 기능도 함께 포함한다. 성능 검증 및 시스템 설계를 위해 필요한 수정 사항은 세부 절에서 설명한다. 설계된 코드는 시스템의 처리 속도를 반영하기 때문에 성능 검증 과정에서 보다 정확한 결과를 얻을 수 있으며, 그 결과에 따라 시스템의 설계 사양 및 상위계층 MAC과 하위계층 MAC의 기능 분할을 조정할 수 있다. 하위계층 MAC은 Verilog 코드로 변환되며, C/C++ 코드를 통해 생성한 테스트 벡터를 이용하여 검증된다. 이와 같이, 제안된 구현 방식에서는 동일한 코드를 이용하여 시스템의 처리 속도를 반영한 성능 검증을 할 수 있으며, 시스템 설계 사양의 결정 및 설계 과정이 밀접한 연관성을 갖고 이루어진다. 본 논문에서 제안하는 MAC 구현 방식의 전체 흐름도는 그림 4에 나타내었다.

#### 4.1 네트워크 시뮬레이션을 위한 고려사항

본 논문에서는 네트워크 시뮬레이션을 위해 이벤트 구동 방식의 네트워크 시뮬레이터인 NS-2를 사용하였다<sup>[10]</sup>. NS-2의 이벤트 스케줄러는 이벤트 시간 정보를 갖는 packet 구조와 타이머를 조정하여 정해진 시간에 관련된 함수를 호출하는 방식으로 시뮬레이션을 진행한다. NS-2 시뮬레이터에 설계된

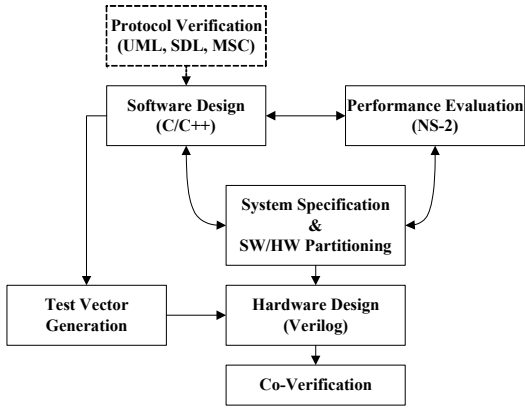


그림 4. 제안된 방식의 MAC 설계 흐름도  
Fig 4. Proposed MAC Design Flow

C/C++코드를 적용하기 위해서는 packet class와 타이머 등의 두 가지 사항이 고려되어야 한다.

#### 4.1.1. Packet Class

NS-2는 packet class를 이용하여 계층 간의 정보를 주고받는다. 일반적으로 packet class에는 이벤트 ID, 길이 및 전송 시간과 프로토콜 헤더 등의 정보를 포함하지만, 실제 데이터는 포함하지 않는다. 설계된 MAC 코드는 데이터 스트림의 처리를 수행하기 때문에, 이를 위해서는 데이터 배열의 포인터를 packet class에 포함시켜야 한다. 또한, 하나의 packet class에 다수의 MSDU 혹은 MPDU의 정보를 포함해야 하는 IEEE 802.11n의 집합 프레임 전송 기법을 지원하기 위해서는 각 MSDU 혹은 MPDU에 해당하는 packet class를 포함시켜야 한다. 수정된 packet class는 그림 5에 나타내었다.

NS-2에서 네트워크 계층 간의 연결은 packet class의 전달을 통해 이루어진다. 따라서 MAC 내부적으로 처리되는 MAC Management Protocol Data Unit (MMPDU)와 Control 프레임도 packet class를 생성하여 전달해야한다. packet class에 포함되어 있는 프레임의 길이와 전송 시간 정보를 갱신한 후, dwnntarget\_과 uptarget\_등의 함수를 이용

```

class Packet : public Event {
Packet*      mac802_11n_pkt[256];
Unsigned int  mac802_11n_numpkt;
Unsigned char* mac802_11n_data;
Unsigned int  mac802_11n_length; }
    
```

그림 5. 수정된 NS-2의 Packet Class  
Fig 5. Modified Packet Class of NS-2

해 다른 계층에 packet class를 전달한다.

#### 4.1.2. 타이머

NS-2는 이벤트 스케줄링 기법을 이용하여 타이머 기능을 제공한다. NS-2에서 제공하는 타이머 기능을 이용하면 소프트웨어만으로는 구현하기 힘든 Slot, IFS, Response timeout, Request To Send (RTS), Network Allocation Vector (NAV), Target Beacon Transmission Time (TBTT) 등과 같은 MAC 프로토콜 타이머들과 MAC/PHY 처리 시간 등의 시스템 특성을 반영하는 지연 시간을 구현할 수 있다.

### 4.2 시스템 설계를 위한 고려사항

C/C++ 코드를 시스템 설계에 적용하기 위해서는 목표로 하는 시스템의 사양, 제어 신호 및 데이터 입출력 등의 인터페이스와 테스트 벡터 생성에 대한 고려가 필요하다.

#### 4.2.1. 시스템 사양 및 인터페이스

본 논문에서는 ARM9 프로세서 기반의 임베디드 시스템에 IEEE 802.11n MAC을 구현하였다. 상위 계층 MAC은 ARM9 프로세서의 펌웨어를 통해 처리되며, AMBA (Advanced Microcontroller Bus Architecture) 인터페이스와 인터럽트를 통해 하위계층 MAC과 연동된다. 상위계층 MAC은 하위계층 MAC 내부에 존재하는 제어 레지스터를 통해 동작을 제어하며, 하위계층 MAC으로부터 인터럽트가 발생하면 인터럽트 정보를 분석하여 해당 함수를 실행시킨다. 하위계층 MAC의 내부 레지스터와 메모리에 대한 읽기/쓰기 동작과 인터럽트 처리 동작을 별도의 함수로 설계하면 C/C++코드에서 Verilog 코드로의 변환을 용이하게 할 수 있다.

#### 4.2.2. 테스트 벡터 생성

하위계층 MAC은 크게 데이터 처리와 제어의 두 가지 과정으로 분리할 수 있다. 데이터 처리 과정은 C/C++ 코드로부터 블록별 입출력 벡터를 생성하여 비교하는 방식으로 C/C++ 코드와 Verilog 코드의 동일성을 검증한다. 하위 계층 MAC의 제어부는 Finite State Machine (FSM)으로 구성되며, C/C++ 코드의 상태 천이 과정과 Verilog 코드의 상태 천이과정을 비교하여 두 코드의 동일성을 검증한다. 이때 상태 천이를 일으키는 조건을 별도의 함수로 설계하면 코드의 변환과 검증이 용이하다.

### V. 설계 결과 및 검증

#### 5.1 네트워크 시뮬레이션

본 논문에 제안된 방식을 통해 설계된 IEEE 802.11n MAC 코드의 전체 구조는 그림 6에 나타내었다. 설계된 C/C++ 코드는 물리 계층을 통해 송수신할 데이터 스트림을 처리하기 위한 PHY-SAP와 프레임의 전송 시간 계산 등을 포함한 PHY Sublayer Management Entity (PLME)의 일부 기능을 포함한다. MAC 세부 블록의 설명은 다음 절의 시스템 구조에서 자세히 설명한다.

설계된 IEEE 802.11n 코드의 성능을 검증하기 위한 네트워크 시뮬레이션 환경은 "Point to Point goodput Test for CC" 시나리오를 사용하여 진행하였다<sup>[9]</sup>. 이론적인 성능과의 비교를 위해 PHY의 전송 속도는 144.4Mbps (MCS15, CBW 20MHz, SGI)로 설정하였으며, 다음의 세 가지 경우에 대한 시뮬레이션을 진행하였다<sup>[8]</sup>.

- Case1: 패킷의 길이를 40us 주기로 125바이트~1500바이트 길이의 패킷을 생성시켜 25Mbps~300Mbps로 트래픽의 속도를 조절
- Case2: 패킷의 길이를 1000 바이트로 고정하고, 25Mbps~300Mbps로 트래픽의 속도를 조절
- Case3: 트래픽의 속도를 100Mbps로 고정하고, 패킷의 길이를 125바이트~1500바이트로 조절

그림 7 ~ 그림 9는 각각의 경우에 대한 MAC의 처리 속도를 이론적인 성능과 비교한 결과를 보여준다. 설계된 코드는 시스템의 처리 속도로 인한 지연

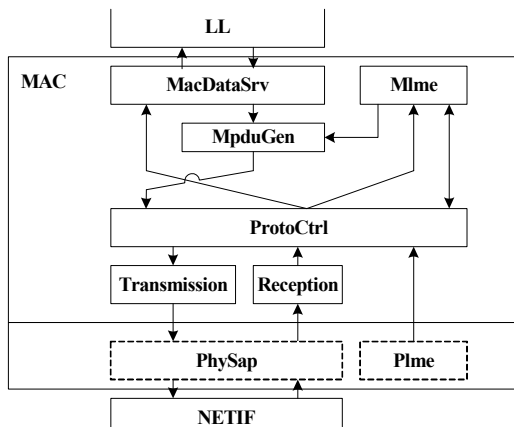


그림 6. IEEE 802.11n 코드의 구조  
Fig 6. Architecture of IEEE 802.11n Code

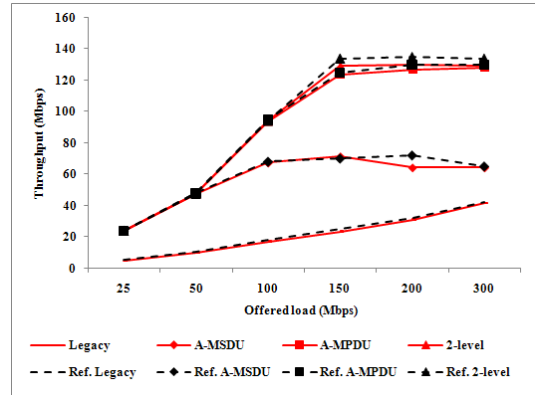


그림 7. MAC 처리 속도 (Case1)  
Fig 7. MAC Throughput (Case1)

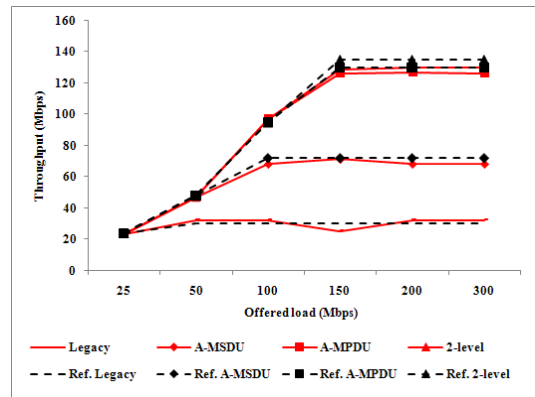


그림 8. MAC 처리 속도 (Case2)  
Fig 8. MAC Throughput (Case2)

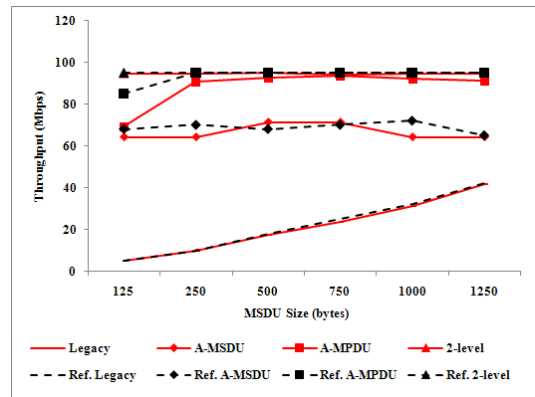


그림 9. MAC 처리 속도 (Case3)  
Fig 9. MAC Throughput (Case3)

시간을 반영하고 있기 때문에 이론적인 성능과는 조금 다른 결과를 보이는 것을 확인할 수 있다.

### 5.2 시스템 설계

본 논문에서 설계된 IEEE 802.11n MAC은 상위 계층 MAC과 하위계층 MAC으로 구분된다. 각 계층에서 수행되는 기능은 표준에서 요구하는 기능들을 효율적으로 수행할 수 있게 분류되어 있으며, 분류된 MAC의 기능들은 표 1에 정리하였다.

그림 10은 본 논문에서 제안된 방식을 이용하여 설계된 IEEE 802.11n MAC 시스템의 전체 구조를 보여준다. Mlme 블록, MpdGen 블록, ProtoCtrl 블록과 Reception 블록의 기능은 상위계층 MAC과 하위계층 MAC으로 나뉘어 수행된다.

MacDataSrv 블록은 LLC로부터 송신할 MSDU를 전달받고 물리 계층으로부터 성공적으로 수신된 MSDU를 LLC로 전달한다. 본 논문에서는 LLC로 연결하는 대신 네트워크 시뮬레이션에서 사용된 테스트 시나리오에 따른 벡터를 사용하여 기능을 검증하였다.

Mlme 블록은 인증-접속, 전력 관리와 링크 적응을 비롯한 MAC의 다양한 관리 기능과 이를 위해 사용되는 MMPDU를 생성한다. 또한, Mlme 블록은 MAC의 동작을 위한 설정 값 및 상태 정보를 저장한다. 하위계층 MAC에는 하드웨어 동작에 필요한 설정 값과 하드웨어의 상태 정보를 저장한다.

MpdGen 블록은 MSDU와 MMPDU로부터 MPDU를 생성하고 전송에 필요한 정보와 함께 전

송 큐에 저장한다. 전송 큐는 beacon을 저장하기 위한 bcnQ, management 프레임 저장하기 위한 txQ, 저전력 동작 중인 수신 단말로 전송될 프레임을 위해 사용되는 psQ와 트래픽 ID별 프레임을 저장하는 tidQ로 구분되며, 연결리스트의 형태로 존재한다. A-MSDU 기법을 사용하는 경우 MPDU의 저장 과정에서 A-MSDU의 생성 기능 여부를 판단하여 적용한다. 하위계층 MAC에는 연결리스트의 마지막 요소를 저장하는 FIFO가 존재한다. FIFO는 작은 용량의 순환 큐 구조로 설계하여 최대 64K바이트의 길이를 갖는 A-MPDU를 효율적으로 지원할 수 있게 하였다.

ProtoCtrl 블록은 채널 접근과 프레임 송수신에 대한 결정을 내리는 MAC의 핵심 기능을 담당한다. ProtoCtrl 블록은 채널의 상태를 확인하고 프레임의 송신을 위한 채널 접근을 제어하는 Enhanced Distributed Channel Access (EDCA)의 기능을 포함한다. EDCA의 기능을 지원하기 위해, 각 Access Category (AC) 별로 독립된 backoff 프로시저가 존재하며, backoff 프로시저 사이의 충돌을 제어하여 우선순위에 따른 전송 품질이 보장되게 하였다. 하위계층의 ProtoCtrl 블록은 전송 준비를 마친 프레임의 정보를 분석하여 RTS/CTS 여부, 분할 송신 여부 및 A-MPDU 기법의 사용 여부 등을 판단하여 해당 전송 과정을 수행한다. 또한, Ack와 Compressed Block Ack를 포함한 수신 응답도 이곳에서 이루어진다. 상위계층의 ProtoCtrl 블록에서는 전송 오류로 인한 재전송 여부를 결정하여 해당 프레임을 전송 큐에 다시 저장하거나 버린다.

Transmission 블록은 ProtoCtrl 블록에서 결정된 전송 큐에 저장되어 있는 프레임을 PHY로 전달한다. Frame Check Sequence (FCS)는 프레임의 전달과 동시에 crc 블록에서 계산되어 MPDU의 마지막에 추가된다. A-MPDU 기법이 사용되는 경우 MPDU 구분자를 포함한 A-MPDU 서브프레임을 생성하여 PHY로 전달한다.

하위 계층의 Reception 블록에서는 PHY로부터 수신된 PSDU를 분석하여 A-MPDU를 수신한 경우 먼저 MPDU 단위로 PSDU를 분리한다. 분리된 MPDU는 각각 오류 여부와 중복 여부에 대한 검사 과정을 거치며, 성공적으로 수신된 MPDU는 수신 버퍼를 거쳐 상위계층의 Reception 블록으로 전달된다.

Timer 블록은 Slot, IFS, Response timeout, RTS, NAV, TBTT 등과 같은 MAC 프로토콜 타이

표 1. 상위계층 MAC과 하위계층 MAC의 기능 분류  
Table 1. Classified features of Upper-layer MAC and Lower-layer MAC

상위계층 MAC
<ul style="list-style-type: none"> <li>- MMPDU 생성</li> <li>- MPDU 생성</li> <li>- A-MSDU</li> <li>- 재전송</li> <li>- 송수신 큐 관리</li> <li>- 분할 수신 프레임 재결합</li> <li>- 다중 속도 지원 및 link 적응</li> <li>- 인증-접속 및 전력 관리</li> </ul>
하위계층 MAC
<ul style="list-style-type: none"> <li>- 채널 접근</li> <li>- 프레임 송수신</li> <li>- 응답 프레임 생성 및 전송</li> <li>- RTS/CTS</li> <li>- 분할 송신</li> <li>- A-MPDU 생성 및 송수신</li> <li>- FCS 생성 및 검사</li> <li>- 중복 프레임 검사</li> <li>- 프로토콜 타이머 관리</li> </ul>

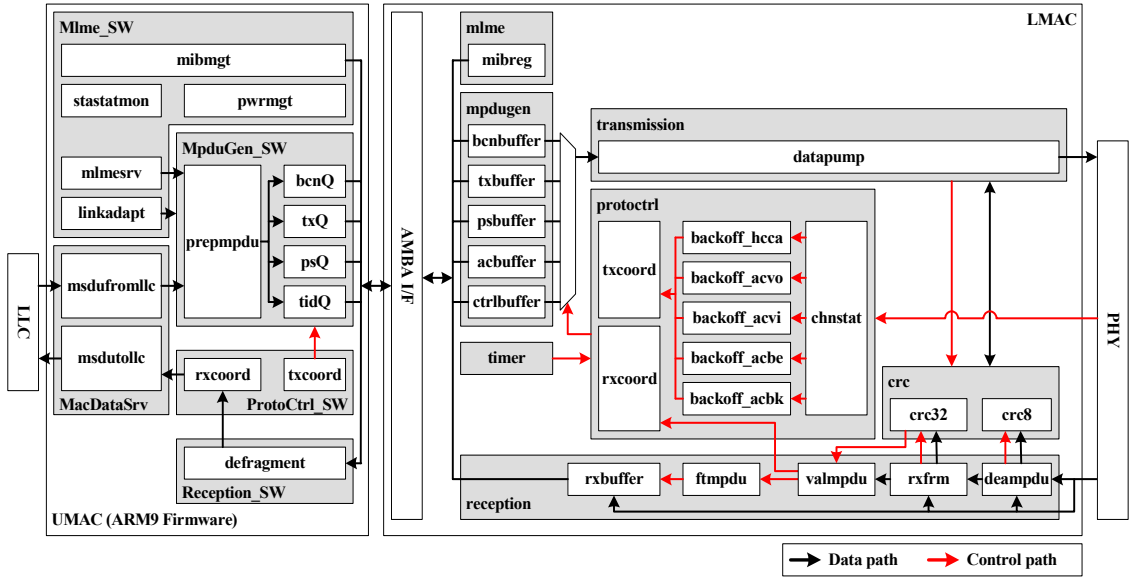


그림 10. IEEE 802.11n MAC 시스템 구조  
Fig 10. System Architecture of IEEE 802.11n MAC

머의 기능을 포함한다.

본 논문에서 설계된 하위계층 MAC은 Xilinx Vertex4 XC4VLX60를 타겟 디바이스로하여 합성하였다. 합성에는 Synplicity사의 Synplify Pro 9.2.4가 사용되었으며, 그 결과는 표 2와 같다. 표 2에 제시된 결과 중 gate count는 CMOS 0.18um 공정에서 Synopsys사의 Design Compiler를 이용하여 합성된 결과이다.

설계된 IEEE 802.11n MAC 시스템의 검증은 ARM9의 펌웨어에서 MSDU를 생성해서 상위계층 MAC으로 전달하는 방식으로 진행하였다. 시스템의 성능 측정은 앞서 네트워크 시뮬레이션과 동일한 네트워크 및 트래픽 환경에서 진행되었으며, Case1에서 A-MPDU를 적용하였을 경우에 대한 네트워크 시뮬

레이션 결과와 시스템의 성능 측정 결과는 그림 11과 같다. 네트워크 시뮬레이션의 성능 결과와 구현된 IEEE 802.11n MAC 시스템의 성능 결과는 오차범위(±3%) 이내에서 거의 일치함을 확인할 수 있었다.

## VI. 결 론

본 논문에서는 시뮬레이션 과정과 시스템 설계 과정에 동일한 코드를 사용하는 MAC 설계 방식을 제안하고, 제안된 방식을 이용하여 설계된 IEEE 802.11n MAC 시스템의 구조를 제시하였다. 제안된 방식은 기존의 MAC 구현 방식과는 달리 네트워크 시뮬레이션을 통한 성능의 검증과 시스템 설계가 밀접한 연관성을 가질 수 있게 해준다. 이를 통해 시

표 2. 설계된 IEEE 802.11n MAC 하드웨어의 합성 결과  
Table 2. Synthesis Report of the Implemented IEEE 802.11n MAC Hardware

Project Status	
Target Device	XC4VLX60
Target Clock Frequency	40MHz
Logic Utilization	
Total LUTs	18,403 (34%)
Dual Port RAMs (RAM16x1D)	64
Single Port RAMs (RAM32x2S)	34
Gate Count	164,146

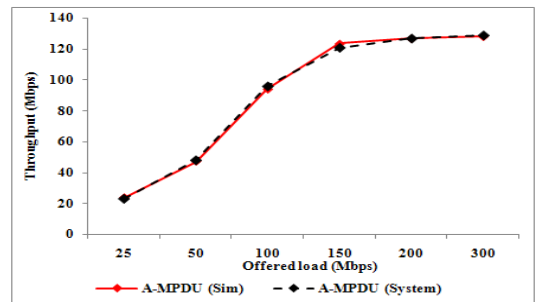


그림 11. 네트워크 시뮬레이션과 시스템 성능 비교 (Case1)  
Fig 11. Comparison between Network Simulation and System Performance (Case1)



시스템을 구성하기 이전에 성능을 평가할 수 있다. 설계된 IEEE 802.11n MAC 시스템은 표준에 새롭게 추가된 기능을 효율적으로 수행할 수 있게 상위계층 MAC과 하위계층 MAC으로 기능이 분리되어 있으며, ARM을 기반의 FPGA 보드를 통해 검증하였다.

참 고 문 헌

- [1] IEEE, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", *IEEE 802.11-2007*, 2007
- [2] IEEE, "Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", *IEEE 802.11n/D4.00*, 2008
- [3] Y. Xiao, "IEEE 802.11n: Enhancement for higher throughput in Wireless LANs", *IEEE Wireless Communications*, pp. 82-91, 2005
- [4] M. Sheu, M. Shieh, S. Liu, C. Dou, "An Efficient Hardware Design Approach from System-Level Specification", *Midwest Symposium on Circuits and Systems*, 1997
- [5] Y. Kim, H. Jung, H. Lee, K. Cho, "MAC Implementation for IEEE 802.11 Wireless LAN", *IEEE 4th International Conference on ATM and High Speed Intelligent Internet Symposium*, 2001
- [6] Y. Cai, F. Luo, H. Zhang, Z. Zhou, "Novel Design and Implementation of IEEE 802.11 Medium Access Control", *IEEE 10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communication*, 2004
- [7] 김유진, "무선 MAC 프로토콜의 FPGA/SoC 구현", 한국전자통신연구원, 2005
- [8] Skordoulis, D., Qiang Ni, Hsiao-Hwa Chen, Stephens, A.P., Changwen Liu, Jamalipour, A., "IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs [medium access control protocols for wireless LANs]", *IEEE Wireless Communication*, Vol 15, Issue 1, 2008
- [9] IEEE, "IEEE P802.11 Wireless LANs Usage Models", *IEEE 802.11-03/802r23*, 2004
- [10] "Network Simulator - NS2", online link, <http://www.isi.edu/nanam/ns/>

정 철 호 (Chulho Chung)

준회원



2003년 연세대학교 전기전자공학과 학사  
 2003년~2006년 (주)아이앤씨테크놀로지 연구원  
 2009년 연세대학교 전기전자공학과 석사  
 2009년~현재 연세대학교 전기

전자공학과 박사과정

<관심분야> WLAN/WPAN/WBAN MAC, 통신용 SoC 설계

이 선 기 (Sunkee Lee)

준회원



2007년 연세대학교 전기전자공학과 학사  
 2009년 연세대학교 전기 전자공학과 석사  
 2009년~현재 삼성전자 연구원  
 <관심분야> WLAN/WPAN MAC, 통신용 SoC 설계

정 윤 호 (Yunho Jung)

종신회원



1998년 2월 연세대학교 전자공학과 학사  
 2000년 2월 연세대학교 전기전자공학과 석사  
 2005년 2월 연세대학교 전기전자공학과 박사  
 2005년~2007년 삼성전자 책임 연구원

2007년~2008년 연세대학교 연구교수

2008년~현재 한국항공대학교 조교수

<관심분야> MIMO/OFDM 통신시스템, VLSI 신호처리, 모뎀SoC설계

김 재 석 (Jaeseok Kim)

정회원



1977년 연세대학교 전자공학과 졸업  
 1979년 연세대학교 전기전자공학과 석사  
 1988년 Rensselaer Polytechnic Institute, NY. 박사  
 1988년~1993년 AT&T Bell Lab.

연구원

1993년~1996년 한국전자통신연구원 책임연구원

1996년~현재 연세대학교 전기전자공학과 교수

<관심분야> 통신SoC설계, 고속멀티미디어IP설계