100Mbpsを超えるIEEE802.11n無線LAN用MACのハードウェア・ソフトウェア協調設計

ヘンドラ セティアワン[†]　Min-Li HUANG[††]　Jin-LEE[††]　Kwang-Eui PYUN[††]
Sin-Chong PARK[††]
長尾 勇平[†]　黒崎 正行[†]　尾知 博[†,†††]

† 九州工業大学大学院情報工学府情報システム専攻 〒820–8502 福岡県飯塚市川津680–4
†† 情報通信大学 大韓民国大田広域市儒城区文旨洞文旨路119
††† 株式会社レイドリクス 〒820–8502 福岡県飯塚市川津680–4 九州工業大学インキュベーション施設
E-mail: †{hendra,nagao,kurosaki,ochi}@dsp.cse.kyutech.ac.jp,
††{minli.huang,mygenie,kepyun,scpark}@icu.ac.kr

あらまし　IEEE802.11n規格は，従来のIEEE802.11a/g等の無線LAN規格に対してスループットの向上を目的としており，少なくとも100Mbps以上の最大実効スループットを実現する．本稿では，IEEE802.11n規格に準拠した100Mbps以上の実効スループットを実現するMACシステムを構築する．このシステムでは，ソフトウェアでの処理だけでなく，ソフトウェアとハードウェアを協調させ，かつ，ハードウェアでの処理を重点的に行うことにより，MACのオーバヘッドを最小化している．

キーワード　IEEE 802.11n, MAC, WLAN

Over 100Mbps High Throughput MAC for IEEE802.11n Wireless LAN with Hardware and Software Cooperative Design

Hendra SETIAWAN[†], Min-Li HUANG[††], Jin LEE[††], Kwang-Eui PYUN[††], Sin-Chong PARK[††],
Yuhei NAGAO[†], Masayuki KUROSAKI[†], and Hiroshi OCHI[†,†††]

† Department of Computer Science and Electronics, Kyushu Institute of Technology 680-4, Kawazu, Iizuka, Fukuoka, 820–8502, JAPAN
†† Information and Communication University 119, Munjiro, Yuseong-Gu, Daejeon, 305-732,KOREA
††† Radrix Co., Ltd. Incuvation Facility, Kyushu Institute of Technology, 680-4, Kawazu, Iizuka, Fukuoka, 820–8502, JAPAN
E-mail: †{hendra,nagao,kurosaki,ochi}@dsp.cse.kyutech.ac.jp,
††{minli.huang,mygenie,kepyun,scpark}@icu.ac.kr

Abstract　IEEE 802.11n is throughput enhancement of previous WLAN standard such as IEEE 802.11a and g. The modes of operation can be enabled to reach much higher throughputs with a maximum throughput of at least 100 Mbps. In this paper we introduce the high throughput MAC system that can reach more than 100 Mbps to support IEEE 802.11n standard. Our propose schemes are using 32 bits data bus interface at 75 MHz clock rate and migration some parts of MAC from software platform to hardware platform to minimize MAC overhead.

Key words　IEEE 802.11n, MAC, WLAN

1. Introduction

To support rich multimedia applications such as high definition television (HDTV, 20Mbps) and DVD (9.8Mbps), the physical layer (PHY) rate in such networks is expected to exceed 216 Mbps. The MAC layer, however, greatly restrains the performance improvement due to its overhead such as backoffs, distributed interframe space (DIFS), acknowledgment (ACK), short interframe space (SIFS) [1]. Moreover, it was well known that the conventional IEEE 802.11 MAC protocol was not enough to achieve improvement of throughput in accordance with increasing PHY data rate [2]. Some schemes such as [1], [3], [4] and [5] had been introduced for MAC layer implementation in WLAN system, but they can not achieve 100 Mbps yet.

The 802.11n amendment has introduced substantial enhancements in Wireless Local Area Network (WLAN) performance, efficiency and robustness of the previous 802.11 physical (PHY) and Medium Access Control (MAC) layers [6][7]. Its goal is to achieve 100 Mbps net throughput, after subtracting all the overhead for protocol management features like preambles, interframe spacing, and acknowledgments [7].

The 802.11n standard has introduced efficiency improvements in the MAC protocol in the form of frame aggregation, block acknowledgements and bursting [7]. Unfortunately, they can not implement directly to achieve 100 Mbps with power consideration. Strict timing constrains for larger frame and limitation of computation resources have big impact in reliable MAC system design.

In this paper, we implement distributed coordination function (DCF) known as carrier sense multiple access with collision avoidance (CSMA/CA) with the goal to provide priorities and traffic differentiation in the wireless access [8]. Thus, the same coordination function logic is active in every station in the basic service set (BSS) whenever the network is in operation [9].

To improve the QoS, we implement Enhanced Distributed Channel Access (EDCA) model. Actually EDCA models take as a basis the previous DCF model that extend them with the new features defined in the standard, such as different AIFS and different ACK policies [10]. However, due the nonlinear behavior of the EDCA random access MAC protocol, some processes require complex computations [10]. Then, rebuild them completely starting from another point of view are of interest in order to increase their accuracy or decrease the computational costs. [10]

In this paper we will discuss about improvement of MAC efficiency from both software and hardware implementation point of view. First MAC target design for IEEE 802.11n will be discussed in section 2. The detail architecture and implementation will explain in section 3. Then, section 4 discuss about simulation and the result that already achieved. Finally, how to improve the MAC layer throughput is discussed in section 5.

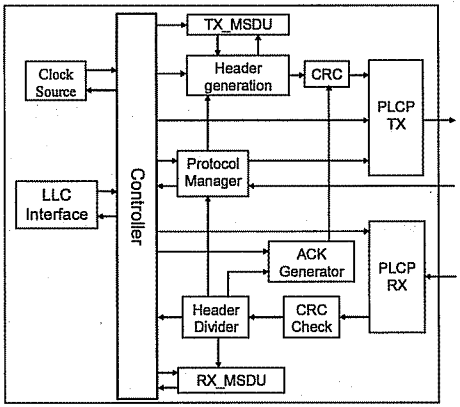

Fig. 1 MAC layer functional block.

Table 1 PHY-MAC System specification.

Standard	MIMO 2x2, IEEE 802.11n
Band	5 GHz
Bandwidth	20 MHz
PHY Max Data rate	144 Mbps
FFT	64 points
Encoding	Convolutional encoder
Coding rate	1/2, 2/3, 3/4, 5/6
Decoding	Viterbi Decoder
Constellation	BPSK, QPSK, 16 QAM, 64 QAM

2. MAC Specifications and Requirments

MAC layer manages and maintains communications between stations (radio network cards and access points) by coordinating access to a shared radio channel and utilizing protocols that enhance communications over a wireless medium. The main services provided by the MAC layer can be summarized as MSDU data delivery, time bounded services, security services (authentication, access control), management services (association service, power management). The overhead operations taken place at acknowledgment, interframe space (IFS), backoff procedure, header generation and header check. To increase efficiency process, we split some of parts MAC in some dedicated blocks as shown in Fig. 1.

The target of maximum MAC layer throughput depend on PHY layer specifications that will be achieved. The global specification is shown in Table 1 which the maximum PHY layer throughput is 144 Mbps. This is the highest throughput for 802.11n standard in 20 MHz bandwidth. Within Aggregation with Fragment Retransmission (AFR) and bit error rate 10^{-6} conditions, Tianji Li et.al [1] advise 70% MAC efficiency can be achieved. Minyoung Park [11] express MAC efficiency equation as

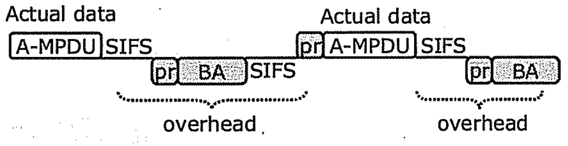

Fig. 2 MAC layer functional block.

$$\text{MAC Efficiency} = \frac{\text{MAC Throughput}}{\text{PHY Rate}} \qquad (1)$$

It mean for 144 Mbps PHY throughput, 100 Mbps even more MAC throughput can be achieved. According to Minyoung Park [11], MAC throughput can be derived as:

$$\text{MAC Throughput} = \frac{\text{Total MAC Payload}}{\text{Time Consumed}} \qquad (2)$$

Time consumed means the duration that consumed transmitting total MAC payload. It contains of real data and overload time duration (see Fig.2). Based on eqation above, to increase MAC throughput, we need to increase the total MAC payload and decrease the duration of overhead process. It means sending real data as many as possible in one packet that called aggregation. To support aggregation, high speed computation is exactly needed. How to implement high speed computation for MAC layer will discuss in the next section.

3. Architecture and Implementation

High speed computation for MAC layer needs both efficient computation in software and high speed device in hardware aspect. In this paper we make hardware-software partioning to increase computation efficiency. There is no unique solution for how to design MAC regarding what to implement in software or to develop in hardware. To support MAC throughput up to 100 Mbps, some hard time constraints between frames must be applied. After the reception of the last byte in the incoming frame the MAC only have a few microseconds before it must have started the transmission of the acknowledge, and this includes to make the decision whether to answer or not due to address decoding, checksum calculating and similar operations.

The most time critic operations occur when a reply frame is to be sent due to an incoming message. Possible replies can be: (1) acknowledge frame, if the incoming frame is of a type that requires this. (2) CTS (Clear To Send), if incoming is an RTS (Request To Send) frame. (3) Data frame, this can occur during CFP (Contention Free Period), i.e. when a centralized coordination function has control over the medium.

Since the timing-critical algorithms and an accurate timing not possible at the software level, reply functionality as be mentioned above are implemented in dedicated hardware platform [12]. If reply functionality is put in hardware it requires decoding and CRC calculation to be there to. Thus, some decoding modules and CRC calculation are also implemented in hardware platform. All other functionality that is not timing critical is executed in software. Hence, MAC implementation consisting of hardware and software that shown in Fig.3. FPGA Xilinx Virtex II pro has been chosen for target device before real ASIC implementation.

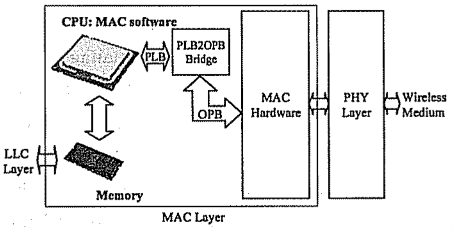

Fig. 3 MAC layer consist of software and hardware part.

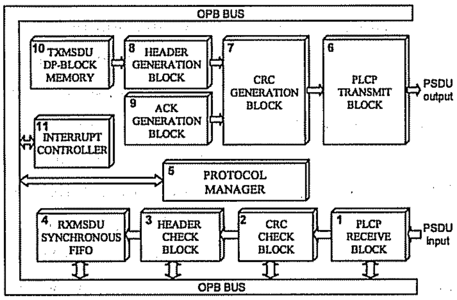

Fig. 4 MAC hardware block diagram.

MAC software has higher position than MAC hardware. It handles interface to logical link control (LLC), complex frame exchanges (e.g.: authentication and association), fragmentation, frame buffering and bridging, and other network management functions. MAC software is implemented using PowerPC (Performance Optimization With Enhanced RISC) that available in Xilinx FPGA board. SystemC language is used to develop MAC software model in PowerPC processor as well as dual port RAM is used for MAC-LLC interfacing.

MAC hardware architecture as shown in Fig.4 consists of some dedicated modules that can work parallel for timing efficiency. Using verilog MAC hardware model was implanted in FPGA. MAC hardware uses on chip peripheral bus (OPB) as main data bus to interface each others. Since MAC software use processor local bus (PLB), OPB to PLB bridge need for interfacing (Fig.3).

There are 11 modules in MAC hardware system shown in Fig.4 detailed below: (1) PLCP receive Block. PLCP Receive Block supports the interface between Medium Access Control (MAC) and Physical Layer (PHY). (2) CRC Check Block. CRC Check Block performs the cyclic redundancy code (CRC) check on frame check sequence (FCS) field in MPDU frame (data, control or management). (3) Header Check. Header Check Block checks the received MPDU frame's type/subtype, Receiver Address (RA) and extracts MAC header information from the MPDU frame. (4) RXMSDU Synchronous FIFO. The synchronous FIFO is a First-In-First-Out memory queue provided by Xilinx. It supports data widths up to 256 bits, and memory depths of up to 65,536 locations [13]. (5) Protocol Manager. Protocol Manager Block manages the transmission of MAC layer channel access protocols related to contention based channel access mechanisms, known as DCF for non-QoS data and EDCA for QoS data. (6) PLCP Transmit. PLCP Transmit Block manages the interface protocol between MAC layer and PHY layer. (7) CRC Generation Block. CRC Generation receives MAC header + MSDU (data, control or management) from Header Generation Block, or MAC header + MSDU (ACK or Block ACK) frame from ACK Generation Block. (8) Header Generation Block. Header Generation Block generates the MAC header for MPDU frame. (9) ACK Generation Block. ACK Generation Block generates control frames, such as ACK, CTS and Block ACK in response to received data or control packets. (10) TXMSDU DP (Dual Port) Block Memory. The dual-port Block memory for virtex-II Pro by Xilinx is composed of single or multiple Virtex-II 18 Kb blocks (SelectRAM-II). It supports data widths up to 256 bits, and memory depths from 2 to 1M word [14]. (11) Interrupt Controller. The OPB interrupt controller (v1.00c) is generated using Xilinx's CoreGen is composed of a bus centric wrapper, attaches to the OPB Bus and it can be used in embedded PowerPC systems (Virtex-II Pro devices) and in MicroBlaze soft processor systems.

MAC hardware system was built at register transfer level (RTL) in verilog or VHDL language. To verify the functionality of this system regarding to MAC-PHY interface, we made PHY RTL model as shown in Fig.5. Using this model, we can simulate transmitting proccess, receiving process, error and no error situations in PHY side. After RTL level verification is completed, next step is board level verification as discussed in the next section.

4. Board Level Verification

Regarding to verify MAC functionality at board level, Table 2 shows the parameters for FPGA board implementation. In one board there are two Virtex II FPGAs and two PowerPC processors that act as an access point (AP) and a station (STA). The architecture of AP and STA are same which is shown in Fig.6. For supervisoring and controlling, a PC are connected to each of them (see Fig.7). From the PC, we can control and monitor the high level MAC process such as beacon scanning, acknowledgment, number of transmit and receive data, and the content of MAC memory in AP and STA.

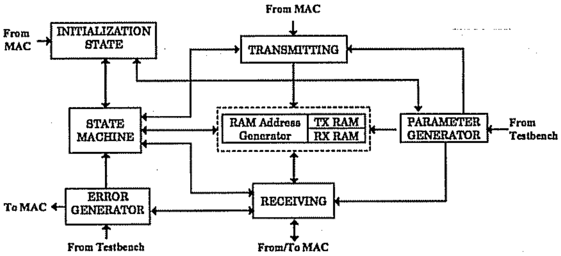

Fig. 5 PHY RTL model for MAC-PHY interface verification.

Table 2 Specification for board level verification

MAC Software	PowerPC PPC405
MAC software clock	200 MHz
CAD tool	Xilinx ISE version 9.1i
Target board	FPGA Vertex II pro 70-5
PHY-MAC I/F Clock	40 MHz
PHY-MAC Data Bus width	32 bits
MAC Hardware clock	50 MHz
OPB clock	50 MHz
PLB	50 MHz

The simulation result for MAC software indicates data generation each 21 clock of PLB bus. Thus, using 50 MHz PLB clock, MAC software throughput only achieves (50 MHz/21 cycles) × 8 bits = 19 Mbps. This value is below of the previous specification that must achieve at least 100 MHz.

At MAC hardware, compilation result using ISE 9.1i shows 10979 slices or around 33% of maximum Virtex II capacity that needed to implement MAC hardware. Since each module has difference task, they have difference timing performances as shown in Table 3. Although simulation use 50 MHz clock, actually they can achieve up to 75 MHz clock rate even more than 100 MHz for some modules. Each clock, MAC hardware can send or receive 8 bits data from OPB. Hence, the MAC hardware throughput can achieve 400 Mbps by 50 MHz OPB clock rate. This is already exceed the 100 Mbps MAC target throughput.

Since MAC hardware and MAC software have difference throughput, the overall MAC throughput will follow the slowest one. Thus, overall MAC throughput only achieves 19 Mbps using 50 MHz clock rate and 8 bits data bus width. Because our target is 32 bits data bus width as mention in Table 2, the MAC overall throughput automatically increase 4 times become to 114 Mbps. This is the maximum value that can be achieved using Virtex II since some modules restricted at 75 MHz. Furthermore, we can increase MAC throughput as long as we use faster devices such as Virtex

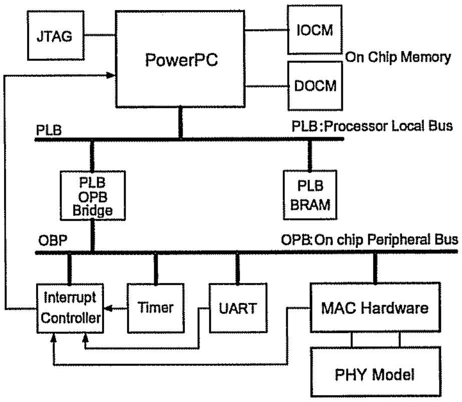

Fig. 6 MAC board level simulation.

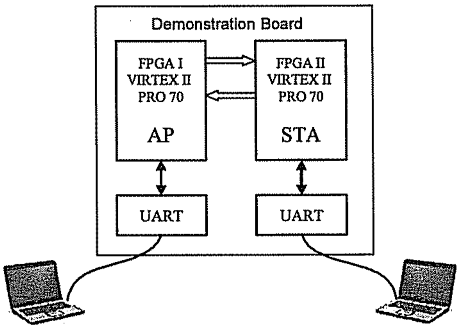

Fig. 7 MAC layer high level verification.

Table 3 MAC hardware maximum frequency result.

Module	Maximum frequency
Header Check	75.053 MHz
CRC Generator	87.856 MHz
ACK Generator	90.630 MHz
Protocol Manager	94.407 MHz
Header Generator	96.045 MHz
PLCP Receive	110.196 MHz
PLCP Transmit	112.633 MHz

IV or higher FPGA performance. Of course implementation on ASIC give higher throughput and power efficiency, significantly.

5. Conclusion

Migration from software platform to hardware platform can reduce the MAC overhead process to support MAC throughput more than 100 Mbps. Hardware and software cooperative design is limited by efficiency of MAC software computation. Hence, overall MAC throughput depand on how fast and efficient MAC software computation. In this design, we can implement MAC throughput 100 Mbps even more at 75 MHz clock rate and 32 bits data bus width at FPGA Virtex II. Further research need to get maximum throughput of MAC implementation in FPGA Virtex-4 and real ASIC.

Acknowledgement

This work was partly supported by a grant of Ministry of Education, Culture, Sports, Science and Technology (MEXT) Japan.

References

[1] Tianji Li, Qiang Niy, David Malone, and Douglas Leith, "A New MAC Scheme for Very High-Speed WLANs", Hamilton Institute, Proc. of the 2006 Int. Symp. on World of Wireless, Mobile and Multimedia Networks, pp. 171-180, June 2006.

[2] Yukimasa Nagai, Akinori Fujimura, Yoshihiko Shirokura, Yoji Isota, Fumio Ishizu, Hiroyuki Nakase, Suguru Kameda, Hiroshi Oguma, and Kazuo Tsubouchi 324Mbps WLAN Equipment with MAC Frame Aggregation for High MAC-SAP Throughput, Consumer Communications and Networking Conf. 3rd IEEE Vol.2, Las Vegas, Nevada, USA, pp. 656-660, Jan. 2006.

[3] Ananth Rao and Ion Stoica, "An Overlay MAC Layer for 802.11", Proc. of the 3rd Int. Conf. on Mobile Systems, Applications, and Services, Seattle, Washington, USA, pp. 135-148, June 2005.

[4] Qiang Ni, Tian-ji Li, Thierry Turletti, Yang Xiao, IEEE P802.11 Wireless LANs, AFR partial MAC proposal for IEEE 802.11n, doc.: IEEE 802.11-04-0950-00-000n, August 2004.

[5] Bechir Hamdaoui and Kang G. Shin, OS-MAC: An Efficient MAC Protocol for Spectrum-Agile Wireless Networks, IEEE Tranc.on Mobile Computing, vol. 7, no. 8, Aug. 2008.

[6] IEEE802.11TM-2007 edition, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", Standard, IEEE, June 2007.

[7] IEEE P802.11nTM/D3.00, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 4: Enhancements for Higher Throughput", Standard, IEEE, Sept. 2007.

[8] Chul Geun Park, Dong Hwan Han, and Seong Joon Ahn, Performance analysis of MAC layer protocols in the IEEE 802.11 wireless LAN, Journal Telecommunication Systems, Springer Netherlands, vol 33, no 1-3, pp. 233-253, Dec. 2006.

[9] IEEE P802.11 Wireless LANs, Joint Proposal: High throughput extension to the 802.11 Standard: MAC, doc.: IEEE 802.11-05/1095r5, Jan 2006.

[10] Boris Bellalta, Flow-level QoS guarantees in IEEE 802.11e-EDCA based WLANs, Ph.D dissertation, Universitat Pompeu Fabra, December 2006.

[11] Minyoung Park, Analysis on IEEE 802.11n MAC Efficiency, IEEE 802.11-07/2431r0, 2007. Available: https://mentor.ieee.org/802.11/file/07/11-07-2431-00-0vht-analysis-on-ieee-802-11n-mac-efficiency.ppt.

[12] Goran Panic, Daniel Dietterle, Zoran Stamenkovic, Klaus Tittelbach-Helmrich, "A System-on-Chip Implementation of the IEEE 802.11a MAC Layer", Proc. of the Euromicro Symposium on Digital System Design (DSD' 03), Belek-Antalya, Turkey, pp. 319- 324, Sept. 2003.

[13] Xilinx, Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet, Xilinx Product Specification, DS083 (v4.7), Nov. 2007.

[14] Xilinx, PLB Block RAM (BRAM) Interface Controller, Xilinx Product Overview, DS420 (v1.4.1) July 2003.