

A Novel Design of IEEE 802.15.3 MAC over UWB

Yu Cai, Zihua Guo, Richard Yao

Microsoft Research Asia
3F, Beijing Sigma Center, No.49, Zhichun Road
Haidian District, Beijing 100080, P.R. China

Abstract—In this paper, we introduce a novel design for the Medium Access Control (MAC) layer of IEEE 802.15.3, which is defined for a high-rate WPAN system. We split the 802.15.3 MAC into two layers: upper layer MAC and lower layer MAC, according to the different timing requirements. Thus, our MAC implementation includes two main modules: 1) a 802.15.3 network interface card (NIC) which is responsible for only direct transmission/reception and other timing critical functions such as protocol accelerator; 2) a driver in Operating System (OS) which is a flexible plug-in module and allows us to implement some non-timing-critical intelligent algorithms. The proposed approach has many advantages such as scalability, flexibility and low cost for chip vendors.

I. INTRODUCTION

Recently, Ultra-WideBand (UWB) technology has gained more and more interest in both academia and industry. One of the most exciting characteristics of UWB is that its bandwidth is up to 480Mbps+ which can satisfy most of the multimedia applications such as audio/video delivery in home networking and it can also act as a wireless cable replacement of high speed serial bus such as USB 2.0 and IEEE 1394. The IEEE standardization body has set up 802.15.3a task group to work towards a common standard for UWB physical layer technology; while one of the MAC layer supports for UWB will be based on IEEE 802.15.3 MAC part [1]. IEEE 802.15.3 MAC has some good features such as QoS and low power design [1]. So, it is of great importance for us to implement an efficient 802.15.3 MAC over UWB, which is the main purpose of this paper.

The whole UWB system contains UWB radio subsystem, base-band processor and MAC controller subsystem. Traditionally, the MAC controller performs the whole MAC functions and is implemented in hardware as a single chip on a NIC. However, our design is quite different. We split the MAC into two parts according to different timing requirements, namely, upper layer MAC (UMAC) which is responsible for non-timing-critical functions and lower layer MAC (LMAC) which needs to respond to events in real time. It is known that hardware implementation can save time while software implementation can be more flexible. So, in our system, LMAC is left in the NIC built in hardware, while UMAC is implemented in OS driver. The 802.15.3 based UWB system architecture of traditional design and our design are shown in Figure 1, from which we can see the difference clearly.

With the UMAC implemented in driver and LMAC in NIC, we can integrate them as a complete MAC for UWB.

Compared with the traditional design, our two-layer MAC design has the following benefits [2]:

- (1) Low cost: With only timing critical functions such as synchronization or beacon generation left in NIC, the powerful embedded CPU acting as MAC controller can be replaced by a less powerful and much cheaper MPU. Thus the cost of the UWB NIC can be greatly reduced.
- (2) Flexibility: It is more flexible to upgrade various management algorithms by only modifying the driver than replacing a new NIC as traditional design does.
- (3) Scalability: We can dynamically load MAC function modules depending on different NIC function requirements. So, it can be adaptive for different environments.
- (4) Testability: In addition to the upper layer MAC protocol, we can allow code extensions in the driver with useful statistical information which would be useful for further analysis, testing and debugging of the system.

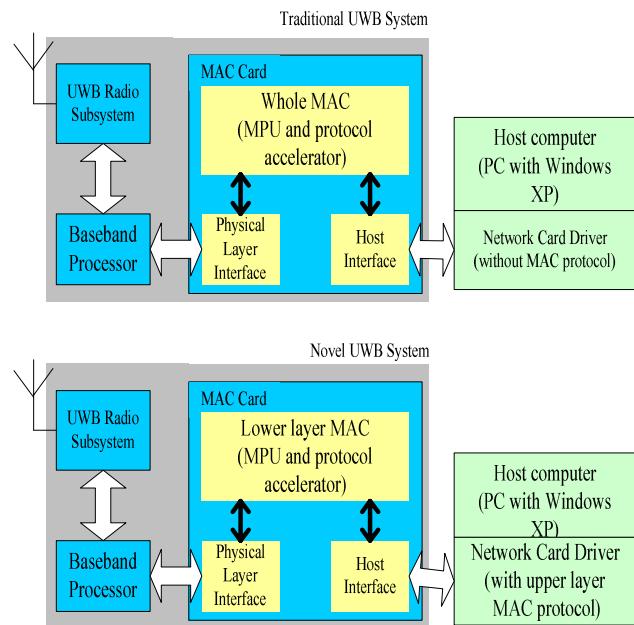


Figure1 Traditional and our proposed UWB system

The rest of the paper is organized as follows. Section II studies the 802.15.3 MAC functions in detail and splits them into UMAC and LMAC. Section III presents the UMAC architecture and integrates it into Windows operating system.

Section IV gives a hardware framework of LMAC which is implemented as an embedded system. Based on the materials above, we estimate the performance of the MAC of UWB and give the testing solution in Section V. Finally, the conclusions are drawn in Section VI.

II. MAC FUNCTIONALITIES PARTITION

As mentioned above, one of the key features of UWB is its bandwidth up to 480MHz+. So, dealing with high data rate is the fundamental requirement of our MAC design. In order to achieve such a goal, the MAC function splitting between UMAC and LMAC must be done properly. Otherwise, the Tx/Rx may not be able to handle frame transmitting/receiving timely. Also, the receiver's frame buffer may overflow and frames may be lost.

After the detailed study of 802.15.3 MAC, we can see that MAC layer consists of MAC data entity and MAC layer management entity (MLME). Most functions of the MLME are not timing critical and can be classified as UMAC. In MLME, functions such as starting, maintaining, stopping piconets and association/dissociation etc. can be delayed without performance deterioration if it is confined to a certain value. For example, if a DEV want to associate with PNC while the UMAC processing is delayed due to some reasons, the DEV's association can still be in success unless the delay is longer than association timeout. But the delay over PCI between host and NIC is usually much smaller than the association timeout, so it is tolerable. It is similar for other MLME functions. However, we must point out that the MAC synchronization in MLME is a bit different and it cannot be viewed as UMAC since DEV synchronization state may change quite often. Besides the MLME, some MAC data functions which is not timing critical can also be implemented as UMAC, for example the MAC protocol data unit (MPDU) generation can be put into UMAC because MPDU generation is to fragment the upper layer service data unit (SDU) and upper layer SDU is also generated in software.

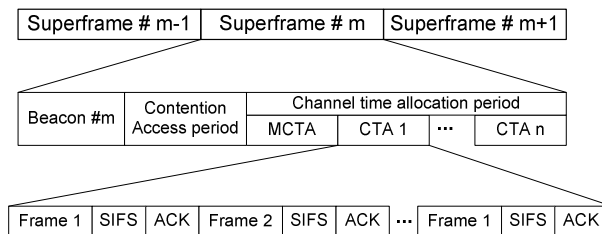


Figure 2 Superframe structure

Many of the MAC data functions, especially the transmission, reception and processing of beacon frame, command frame and acknowledge frame must be classified as LMAC and they should be implemented in hardware. In 802.15.3, the data frame, command frame are transmitted in the sequence of superframes and the superframe is shown in Figure 2. First, the PNC transmits the beacon frame. DEV parses the received beacon and extracts information such as timing synchronization, channel time allocation (CTA). The maximum superframe duration is 65536 us. Although the

host side OS may be able to handle beacon parsing, we still classify beacon parsing in LMAC to allow DEV allocated the first MCTA or CTA1 to prepare transmission in time. During the transmission with Imm-ACK, source DEV communicates with destination DEV, and every data frame will be acknowledged by an ACK frame. The inter-frame space is small between the data frame and the ACK frame is rather small. Thus, it is impossible for the host side OS such as Windows or Linux to deal with the event in such a short time. So the ACK frame must also be implemented in hardware to respond in time. Similarly, other timing critical functions should also obey such rules. From detailed analysis we find that frame transmitting, receiving and wireless channel control should be split into LMAC and implemented in hardware.

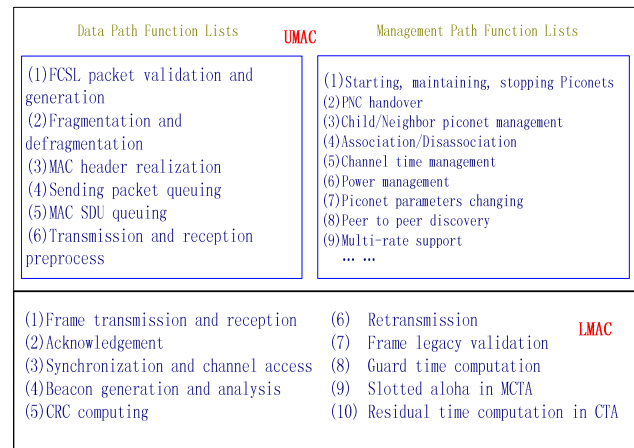


Figure 3 UWB MAC Partitioning

In summary, the 802.15.3 MAC partition for LMAC and UMAC are shown in Figure 3. Here, we put timing critical MAC functions into LMAC and those non-timing-critical functions into UMAC. UMAC is implemented as Intermediate Driver or protocol driver which is ported into Network Driver Interface Specification (NDIS) in Windows system [2]. LMAC includes task control and direct frame transmission and reception. In LMAC, the former is implemented as firmware running on embedded CPU controlling the whole LMAC and other hardware modules in the UWB NIC; while the latter is realized in logic design as the protocol accelerator attached to the on-chip bus of micro-controller unit (MCU) such as ARM, PowerPC. In the following section, we will focus on the detailed design and implementation of UMAC and LMAC.

III. UMAC IMPLEMENTATION

UMAC consists of partial MAC data path and management path. Data path is responsible for the functions such as MSDU reordering, fragmentation/defragmentation etc. Management path is responsible for various management functions such as power management, security, QoS scheduling, PNC handover etc. and it manages the state of the whole MAC. In our design, UMAC is mapped into

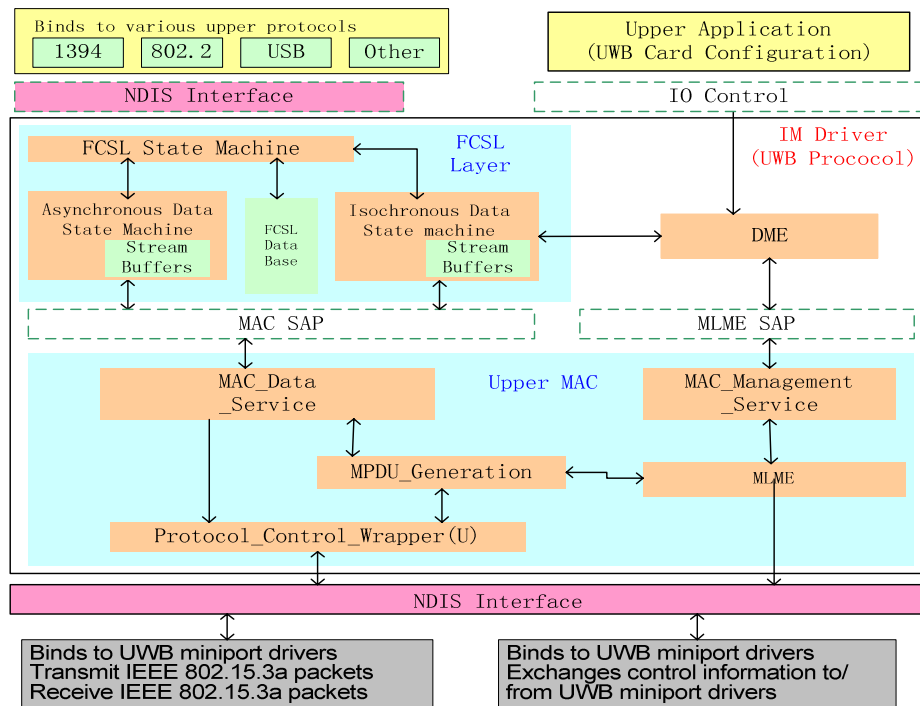


Figure 4: UMAC framework on OS side (Windows)

NDIS driver. Above the NDIS are the upper layer protocols and a convergence sublayer is thus required. These protocols can be wireless USB, wireless 1394, or TCP/IP which may be defined by WiMedia forum [6]. Below NDIS interface is UWB NIC miniport driver which read or write data and status from the UWB NIC. NDIS interface links upper application, IM driver and lower NIC miniport driver and integrates them into a whole software system. The interaction between them and their sub-components can be clearly seen in Figure 4 [3].

The 802.15.3 protocol implementation consists of three major parts, namely, UMAC, device management entity (DME) and frame convergence sub-layer (FCSL). DME is responsible for functions such as gathering layer dependent status from various layer management entities and setting the value of layer-specific parameters. It can receive I/O commands from upper layer and set or reset whole MAC properties. DME can also control the behavior of FCSL and UMAC.

UMAC communicates with LMAC which is left on the NIC and their communication is in the form of MPDU fragmented by MPDU-Generation sub-module. It transfers information from upper layer through the FCSL and the communication is in the form of MAC SDU. In our design, the UMAC is divided into five sub-components and their interaction and MAC data flow can be seen clearly from Figure 4. These sub-components are described in details as follows.

- **MAC Data Service:** Transfers and validates MAC SDU between upper FCSL and UMAC. It acts as data frame

interface.

- **MAC_Management_Service:** Transfers and validates MAC management SDU between DME and UMAC. It also maintains piconet information base (PIB) according to the command from DME. PIB contains the MAC status such as dynamic channel selection, transmitting power control etc. It can be set or reset by DME or adjust itself according to the management information received from MLME.
- **MPDU_Generation:** Detach frame received from lower layer to their respective path, that is, data frame to MAC_Data_Service and management frame to MLME. It also multiplex frame received from upper layer and management path to the same channel. In this block, fragmentation and defragmentation are done if necessary
- **MLME:** Handles different management functions, e.g., power management, security, starting/maintaining/stopping piconets, and so on. When a management frame from MPDU_Generation is received, it will process the frame and extracts information to modify PIB and also produce response frame such as association response frame. In our design, we can dynamically upload different management algorithms according to different environments.

Besides the above parts, Protocol_Control_Wrapper is also implemented in the IM driver as UMAC module for better connection between UMAC and LMAC. However, Protocol_control part which controls the frame transmission and reception is left on NIC as LMAC.

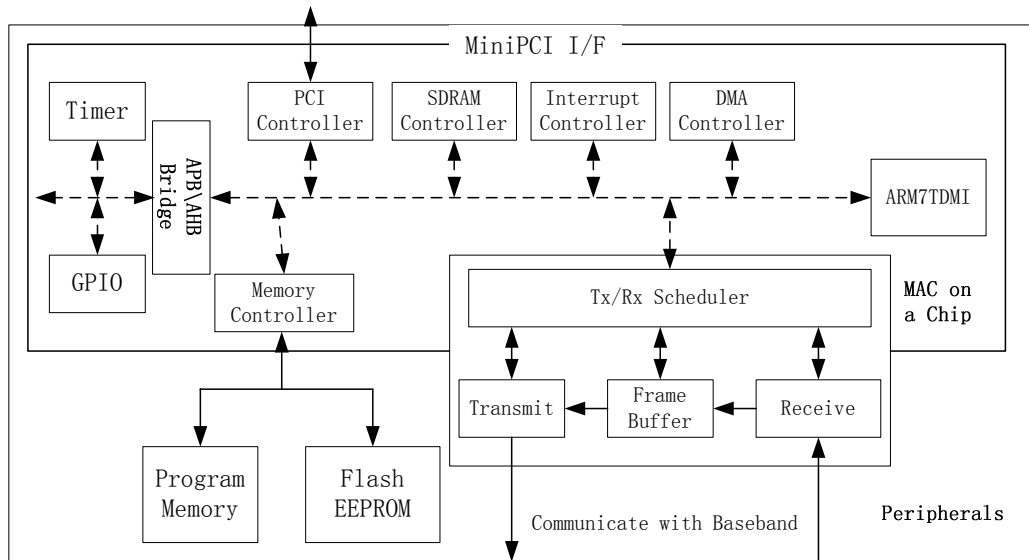


Figure 5 LMAC hardware architecture based on ARM

FCSL classifies upper SDU into asynchronous and isochronous streams and forwards them into their respective Service Access Point (SAP) with UMAC. In our design, the FCSL contains an important component, that is, FCSL database. This component maintains a classification parameter set to help FCSL state machine to analyze each frame entering the FCSL. The set includes class priority, system index and protocol specific parameters such as destination MAC address, source MAC address, etc. Each set entry corresponds to a specific stream index. When an upper layer 802.2 frame PDU enter FCSL, FCSL state machine will classify it to its corresponding stream index according to FCSL database by checking classification parameter set entry. At the same time, the source address and destination address are mapped to 802.15.3 source DEV ID and destination DEV ID under the help of DME. Then, FCSL delivers frame SDU to the MAC SAP in a request primitive. The request primitive's format is as follows [1]:

```
MaUnitData.request
{
    SourceAddress,
    DestinationAddress,
    RoutingInformation,
    Data,
    Priority,
    ServiceClass
}
```

When MAC SDU needs to be forward to FCSL, the FCSL state machine just multiplexes different stream under the help of FCSL database. In this case, the communication primitive is as follows [1]:

```
MuUnitData.indication
{
    SourceAddress, DestinationAddress
    RoutingInformation, Data,
    ReceptionStatus, ServiceClass
}
```

IV. LMAC IMPLEMENTATION

Figure 5 shows the architecture of LMAC which is based on an ARM micro-processor. Through PCI interface, LMAC communicates with UMAC which runs as host NIC driver. When accessing piconet based on 802.15.3, the frames sent are data frame, beacon frame, acknowledge frame (ACK) and command frame [1]. In figure 5, Tx/Rx scheduler is a set of hardware module which accelerates the transmission and reception of MAC MPDU under the control of firmware. It contains some status registers which may be read or set by firmware and they can trigger hardware actions to complete the frame transmission and reception.

The data frame is generated by UMAC and is fetched by the PCI controller from host memory when it needs to be transmitted. Then it is stored in the program memory and forms a first in first out (FIFO) queue. MAC firmware running on ARM [5] will do some preprocessing on the frame. Then firmware will transmit the first frame in the FIFO to Tx/Rx scheduler using the DMA engine. There, it is stored in the internal frame buffer in our protocol accelerator. The firmware running on ARM maintains a TDMA procedure and it monitors the timer. When it finds that its CTA arrives, the firmware will send enabling instruction to the transmission and reception scheduler. Then, the hardware will start to transmit the buffered frame in the internal buffer. The frame is sent to base-band and finally transmitted over wireless. Every time a frame is sent, the scheduler will inform the firmware of that and the firmware will do some processing and decide whether it will continue to send the next buffered frame or retransmit the previous frame. The processing also includes the computation of the residual time left in the CTA to avoid current communication going beyond the allocated CTA. When data frames are received, it

will be transferred to the UMAC in the host using the reverse path for transmission.

The frames other than data frames are quite timing critical and they need to be sent in a few microseconds, so their processing method is different from the data frame. For instance, when a data frame is received, the receiver has to respond with an ACK. ACK is a very simple frame and the transmission time is

$$\frac{FrameLength}{MacDataRate} \approx \frac{10Octets*8}{100M} \approx 1\mu s$$

and IFS which is between data frame and ACK frame is only a few microseconds in 802.15.3. So the total time left for processing is very limited. So, ACK frame generation should be implemented in pure hardware. In our design, ACK is generated by Tx/Rx scheduler that extracts information from previous data frame such as DestID, SrcID, Stream index to form an ACK. Another instance is the Beacon frame generation and reception. When a beacon frame arrives, the protocol accelerator will generate an interrupt to ARM and the interrupt routine will send certain instruction to it. Under the control of the firmware, Tx/Rx scheduler can extract all the information in beacon in real time and reset the status of the LMAC such as CAP starting time and duration. It can also modify information PIB in UMAC through firmware in ARM and PCI interface.

In summary, in our LMAC design, we treat data frame and other frames in different ways and this can meet the timing requirement and flexibility requirement.

V. PERFORMANCE TESTING

The performance of our design is now under estimation and simulation with the prototype described in Specification and Description Language (SDL). With SDL, we can simulate and validate the whole system in a formal way. Figure 6 illustrates the testing and experimental environment of the proposed novel MAC frame work. In order to test all MAC functions, we will totally build four device prototypes to form a piconet with one of them acts as piconet controller (PNC) and the other three act as normal DEVs. Each prototype in our testing network environment is a laptop equipped with a NIC and its corresponding driver based on our novel design. The LMAC is implemented in FPGA as a UWB NIC and we also install each laptop with a driver containing UMAC. Thus, the PNC can send beacon frame to devices to control the piconet and the devices can communicate in peer to peer mode under the control of the PNC. The testing framework can cover the fundamental 802.15.3 MAC functions.

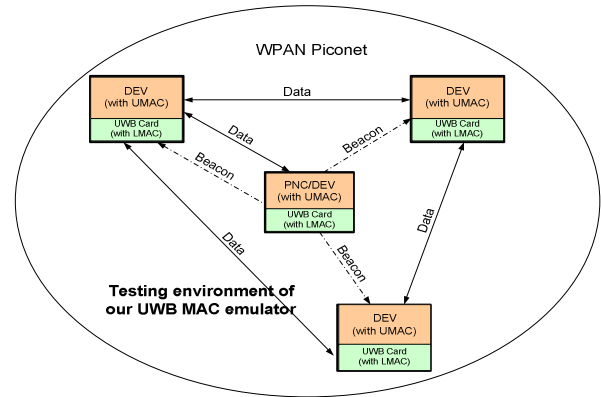


Figure 6 UWB System Testing Environment

VI. CONCLUSIONS

In this paper, a new two-layer based MAC design is proposed for 802.15.3 system over UWB. By considering the high speed requirements imposed by UWB, we partitioned the timing critical functions into LMAC and the non-timing critical functions into UMAC. The advantages of the design include flexibility, scalability and lower cost, etc. The detailed implementations of LMAC and UMAC are presented in this paper.

REFERENCES

- [1] IEEE Standard for Telecommunications and Information Exchange between Systems—LAN/MAN Specific Requirements -- Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN).
- [2] T. Aruna, B. Abhishev, Z. DKalin, "Microsoft, Native WiFi Architecture Specification."
- [3] A. Ganz, A. Savvides, Z. Ganz, "Media access control development platform for wireless LANs", in *Proc. Of IEEE ICECS '99*, Sept. 1999, pp.105 - 108.
- [4] T. Fujisawa, et.al., "A single-chip 802.11a MAC/PHY with a 32-b RISC processor" *IEEE Journal of Solid-State Circuits*, Nov. 2003, pp. 2001-2009.
- [5] <http://www.arm.com>
- [6] <http://www.wimedia.org>