國立交通大學

電子工程學系

碩士論文

移動性 WiMAX 系統：
跨層設計方法與軟體架構之系統觀點

Mobile WiMAX System:
A Systematic View on Cross-Layer Design Methodologies and Software Architecture

研究生：陳冠穎

指導教授：黃經堯　教授

中華民國九十六年十二月

移動性 WiMAX 系統：
跨層設計方法與軟體架構之系統觀點
Mobile WiMAX System:
A Systematic View on Cross-Layer Design Methodologies and Software Architecture

研究生：陳冠穎　　Student: Kuan-Yin Chen

指導教授：黃經堯　　Advisor: ChingYao Huang

國立交通大學
電子工程學系
碩士論文

A Thesis
Submitted to Department of Electronics Engineering
College of Electrical & Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in

Electronics Engineering

November 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年十二月

移動性 WiMAX 系統：
跨層設計方法與軟體架構之系統觀點

學生：陳冠穎

指導教授：黃經堯　教授

國立交通大學電子工程學系碩士班

摘　　要

於設計新一代的無線通訊系統時，設計者面臨到一連串的挑戰：標準本身的演進、上市時間、系統維護效率乃至於潛在的多系統整合的需求。也由於效能的需求愈來愈高，跨層（Cross-layer）運作在新一代的通訊系統中，扮演著重要的角色。若要達成諸多不同的設計目標，必須要有一個切合需要的 MAC 層系統架構。在本文中，我們提出一個模組化的 MAC 層系統架構與其設計方針。該架構是源自於一個包括有三個平面的概念模型；該架構能使得功能模組被快速地設計出來、系統維護更加簡易、跨層通訊更有效率且能適應於將來多系統整合的目標。

Mobile WiMAX System:
A Systematic View on Cross-Layer Design Methodologies and Software Architecture

Student: Kuan-Yin Chen

Advisor: ChingYao Huang

Department of Electronics Engineering

National Chiao Tung University

ABSTRACT

In designing novel wireless communications devices, designers are facing a number of challenges such as standard evolution, market timing, system maintenance and even multi-technology integration. Also, as the performance requirement becomes higher, cross-layer operation plays an important role in modern systems. To accomplish various design goals, the MAC layer architecture should be elaborately defined. In this article, we present a modularized MAC architecture and its design guidelines. This architecture is based on a three-plane conceptual model, and enables quick prototyping, easy maintenance, efficient cross-layer signaling and potential multi-technology integration.

誌謝

在完成本篇論文之際，首先要感謝的是我的指導老師黃經堯副教授。在就讀電子研究所的兩年半期間，承蒙黃教授不斷給予研究方向上的指導、督促與研究資源；在此向黃教授致上最誠摯的謝意。另外，亦感謝參與口試審核委員會的資策會李永定組長與電信系方凱田教授，與在計畫中合作的林大衛教授團隊。

感謝所有實驗室的成員，特別是兩年間與我合作計畫的曾理銓、李純孝、陳志展。感謝他們在兩年間的互相討論、砥礪與勉力務事，才能完成兩個年度的資策會計畫。該計畫是我研究所生涯的主軸，也是這篇論文的根源。

最後要感謝的是我的父母家人，在我出生以來，便無一日不養育我、支持我、鼓勵我成長。要感謝的人族繁不及備載，謹此這篇研究論文，獻給所有我身邊的人，謝謝！

陳冠穎

2007 年十二月十四日　謹誌於國立交通大學

目 錄

表目錄

圖　　目　　錄

符號與略語說明

略語	說明
BS	Base station
CID	Connection identifier
CS	Convergence Sublayer
DCD	Downlink channel descriptor
DL	Downlink
DPC	Data plane command
FCH	Frame control header
HO	Handover
MAC	Medium access control layer
MS	Mobile station
OFDM	Orthogonal frequency division multiplexing
OFDMA	Orthogonal frequency division multiple access
PHY	Physical layer
QoS	Quality of service
RRM	Radio resource management
UCD	Uplink channel descriptor
UL	Uplink
WiMAX	Worldwide Interoperability for Microwave Access

Chapter 1. Introduction

The demand for broadband wireless access (BWA), especially mobile and multimedia services, is continuously growing. Several technologies are developed to support new cell-based, mobile BWA environment. Among these proposals, the IEEE 802.16 Standard Family has been the most prominent candidate. However, the new standards, which has higher performance requirement and larger complexity than ever, bring new challenges in system design and implementation.

A system compliant to IEEE 802.16 standards is also called a “WiMAX (Wireless Interoperability for Microwave Access)” system, named after the sponsoring industry alliance, the WiMAX Forum. The standard family includes two main active documents:

- IEEE Standard 802.16-2004 for static BWA, which is officially published in October 2004. [1]
- IEEE Standard 802.16-2005 for mobile BWA, which is officially published in February 2006. IEEE Standard 802.16-2005 published as a revision and extension document of the previous one. [2]

IEEE 802.16 Standards include quite a few features that are different from previous cellular systems. The WiMAX system is designed based on a common MAC layer and an adaptive PHY layer. Four PHY modes are specified: Single-Carrier (SC), Alternative Single-Carrier (SCa), OFDM and OFDMA. In designing a mobile WiMAX system, designers tend to concentrate on the OFDMA mode due to mobility issues and throughput requirement. WiMAX is the first practical system that provides cellular services based on OFDMA technology. Another key feature of 802.16 PHY is link adaptation. The PHY transmission parameters, such as modulation-coding scheme (MCS), subcarrier usage and power, can be adjusted on a per-PDU, per-connection or per-user basis. [3]

To support adaptive PHY features such as link adaptation, MIMO & AAS and QoS levels, cross-layer operations are necessary. This is the key point that WiMAX system design emphasized more than any previous system designs. Traditionally, designers adopt layered models that prohibit direct communications between layers, to facilitate system development, and provide compatibility between products from different vendors. However, in WiMAX system design, cross-layer adaptation has to be taken into account, and traditional layered models are not perfectly suitable for new system implementation [4]. A new, modularized and rapid prototyping methodology must be developed.

Also, hard performance requirements and time-to-market issues bring great challenges to WiMAX software implementation. In a WiMAX system, there are very complicated QoS and link adaptation policies. To implement these policies, designers have to develop algorithms, and continuously modify them to keep interoperability. This requires rapid prototyping of control algorithms in the early design stage, and full modularization of each control mechanism to facilitate modification, replacement and case-selection. If the system software architecture is not carefully planned, designers may have to pay considerable time and manpower to maintain the system software.

For cross-layer signaling methodologies, there have been a number of discussions such as [5], [6], [7], [8] and [9]; most of them are developed from the viewpoint of layers higher than link (MAC) layer, and lacks considerations on implementation issues. Prior works and researches on overall WiMAX MAC architecture are rare. [10] proposed a simulator design compatible with standard 802.16-2005. [11] then developed an NS-2 simulation platform mostly based on the MAC architecture of [10]. [10] has a good overall discussion on the operation of the mobile WiMAX system, however, the architecture proposed is incomplete, and discussions on modularized design are rare. [11] has good system view on modularized system implementation, but does not emphasize on MAC/PHY cross-layer design.

This article describes a proposed software architecture with modularization, and focuses on MAC/PHY cross-layer implementation considerations. The rest part of this article is organized as follows: First, there is a brief technical overview of IEEE 802.16 standards. Second, the proposed overall architecture is illustrated. After that, the cross-layer design methodologies are discussed. Finally, there is a case study of 802.16 network entry as an illustration of overall system design.

Chapter 2. Mobile WiMAX System Review

The IEEE Standard 802.16 Family specifies the air interface of fixed and mobile WiMAX systems. The standard scope includes two layers: the MAC layer and the PHY layer. Specification of the MAC layer includes the security and convergence sublayer. The system architecture, optimizations and algorithm is not in the defined scope. A reference model of IEEE 802.16 PHY/MAC is shown in Figure 2-1.

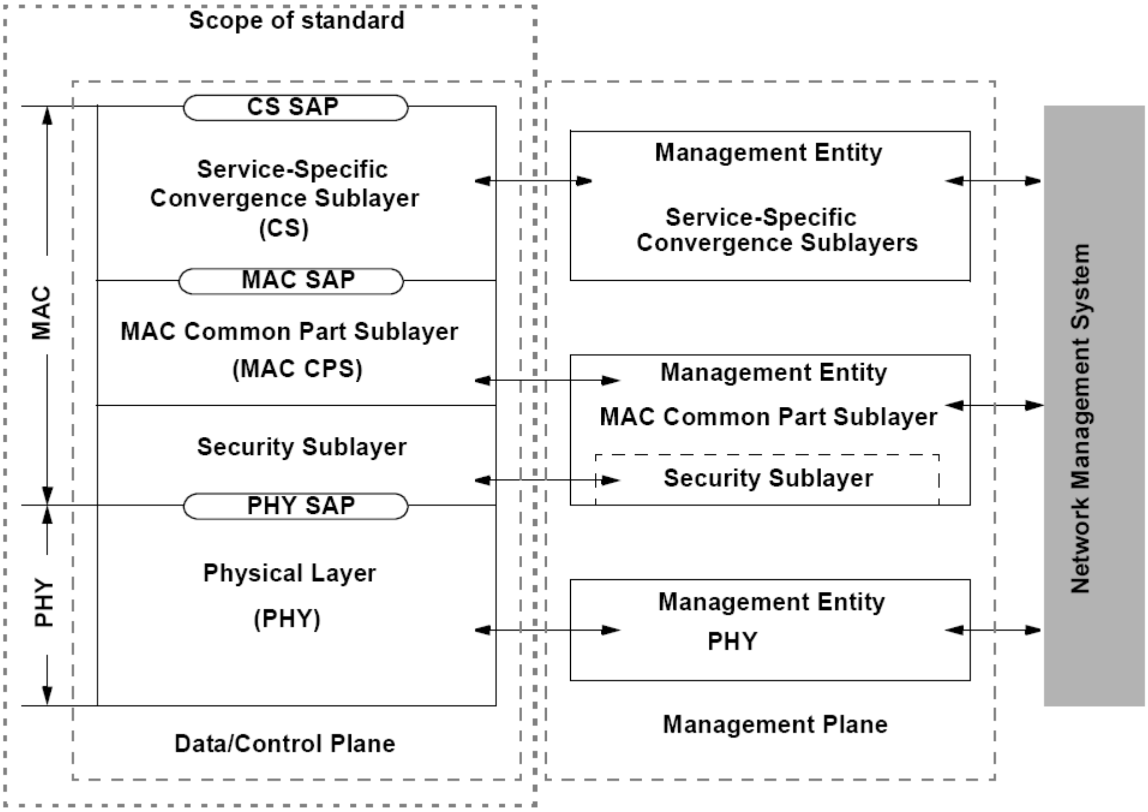


Figure 2-1: IEEE 802.16 PHY/MAC Reference Model [1] [2]

A WiMAX system is equipped with an adaptive PHY layer that supports multiple PHY modes, and allows PHY parameters such as modulation, coding and subcarrier allocation to be adapted on a per-connection or per-user basis [3]. A flexible common MAC layer enables different PHY modes, and succeeds various services from upper layers in an optimized manner.

Four PHY modes are specified in IEEE Standard 802.16. They are: WirelessMAN-SC, SCa, OFDM and OFDMA. WirelessMAN-SC and SCa are single-carrier modes intended for line-of-sight (LOS) operation of high transmission rate. The former operates in 10-55 GHz spectrum, and the latter is specified for sub-11GHz operation. WirelessMAN-OFDM and OFDMA are designed for non line-of-sight (NLOS) operation of client air interface, operation at sub-11GHz spectrum.

Compared with previous BWA systems, the MAC layer plays a more important role and is more complicated than ever. The MAC should be flexible in many ways:

– The MAC layer shall support both time and frequency domain duplexing, to accommodate unpaired spectrum and asymmetric data traffic.

– It shall also adapt its transmission control strategy, such as scheduling, coding-modulation and bandwidth allocation scheme, to meet different QoS levels.

– It may even support multiple network topologies, including point-to-point (PtP), point-to-multipoint (PMP) and mesh networking.

Practically, in the followings of this article we focus on OFDMA PHY mode and its related MAC operations only, since it is the usual case for mobility support. This section is organized as this: first we'll make a brief introduction to 802.16 PHY layer, then the 802.16 MAC layer is reviewed.

2.1. PHY Layer Technical Overview

We focus on only OFDM/OFDMA PHY in this technical overview. OFDM is a PHY modulation method that enables efficient data transmission over multiple frequency carriers. It uses a large number of closely-spaced subcarriers for transmission, while signals at adjacent subcarriers are orthogonally encoded and modulated such that inter-subcarrier interference is reduces and close spacing of subcarriers is allowed. The OFDM signal is generated by a mechanism illustrated in Figure 2-2.

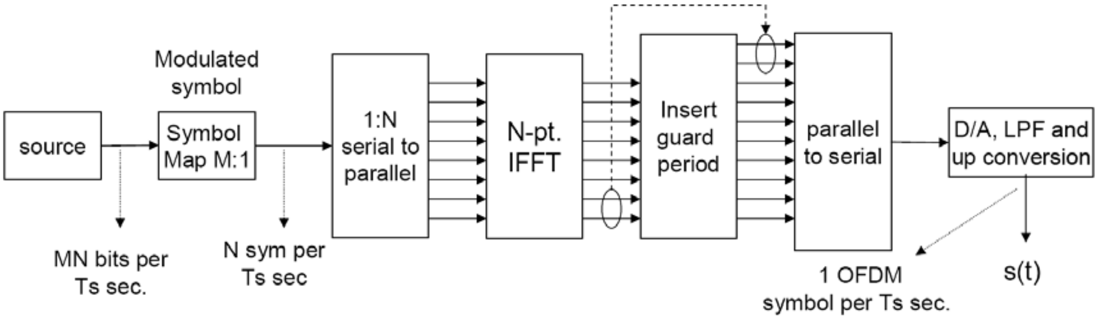


Figure 2-2: System Model for OFDM Signal Generation

The most important advantage of OFDM technology is that it is robust against inter-symbol interference and frequency selective fading caused by multipath transmission. It enables a narrow-bandwidth user to benefit from frequency diversity by spreading is subcarrier usage over a wide bandwidth. Different from traditional frequency division schemes, it eliminates guard band between two adjacent subcarriers with orthogonality, so the frequency efficiency is much better. Implementation of OFDM technology is also facilitated by a series of existing DFT/FFT DSP technologies.

However, OFDM technology pursues frequency efficiency at the cost of overheads and sensitivity: it needs to add cyclic prefix to time domain block to guarantee signal recovery, thus causing cyclic prefix overhead. Since subcarriers are densely spaced, the transceiver is very sensitive to frequency offset and resulted loss of orthogonality. Fortunately, this drawback is compensated by improving transceiver technology.

The OFDMA mode is a variant of OFDMA technology. It applies the same signal generating scheme as OFDM, but allocates radio resource in a 2-dimensional sense. The OFDMA mode divides the time domain into equal-duration OFDM symbols. Multiple users may share OFDM subcarriers in one OFDMA symbol time. Therefore, usage of subcarriers by users can be represented with “blocks” on a 2-dimensional map, as illustrated in Figure 2-3. OFDMA is more suitable than OFDM mode for mobility use due to its flexibility and promptness in radio resource allocation. However, this also makes the MAC layer more complicated for implementation.

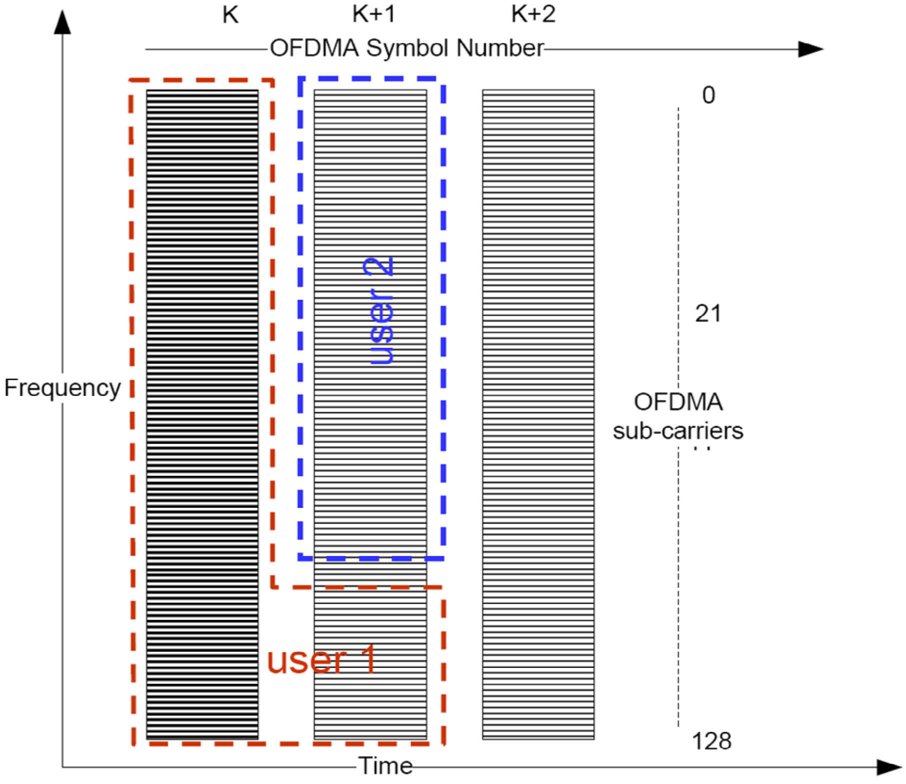


Figure 2-3: OFDMA Domains

2.1.1. OFDMA PHY Frame Structure

WiMAX supports both time division duplex (TDD) and frequency division duplex (FDD) modes, but the OFDMA mode supports only TDD. For TDD mode, uplink and downlink use the same spectrum, and a PHY frame is separated into downlink and uplink subframe in time domain. A Typical frame structure layout is shown in Figure 2-4. [1] [2]

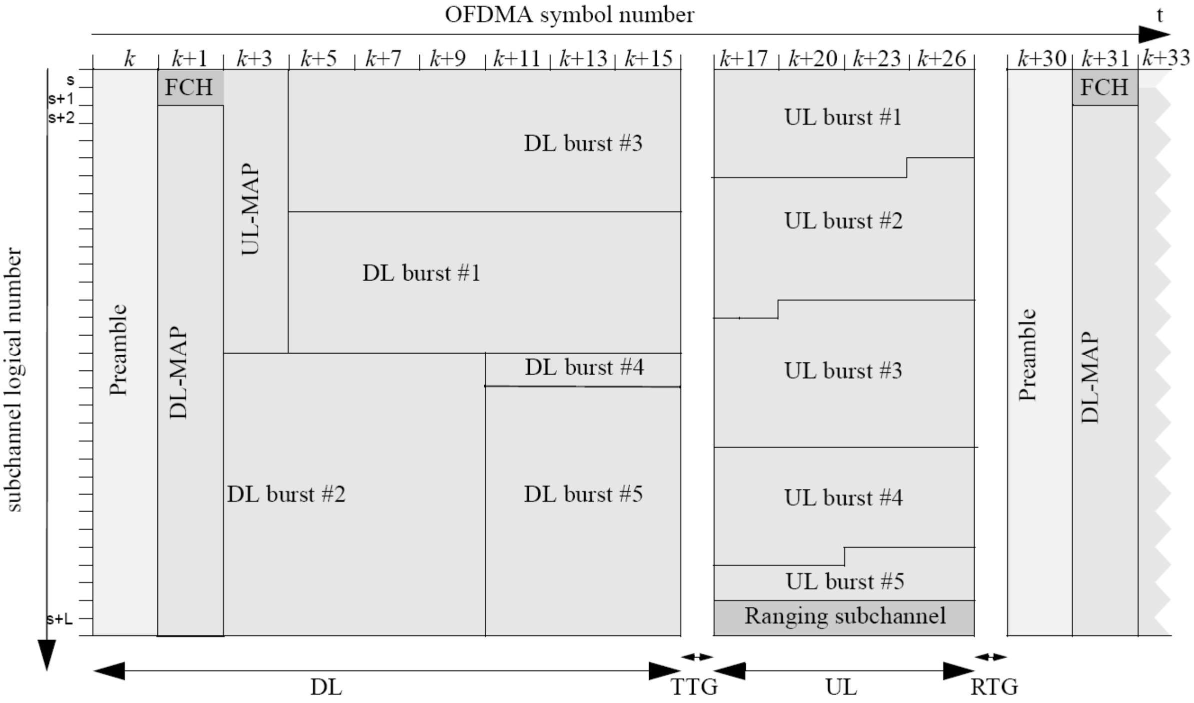


Figure 2-4: Typical TDD Frame Structure under OFDMA Mode [1]

An OFDMA TDD frame includes the following building blocks:

1. Preamble:

Each downlink frame starts with a preamble, which lasts for one OFDMA symbol and occupies the entire spectrum. The preamble is robustly modulated in BPSK across subcarriers by a PN code identifying BS cell/sector. MSs may acquire the system and maintain synchronization according to the preamble. Also the preamble enables MSs to estimate the channel and correct frequency/time offset.

2. Frame Control Header (FCH):

FCH contains Downlink Frame Prefix, which indicates the coding scheme and length of DL-MAP. MSs decode the DL-MAP according to information provided in FCH. The FCH uses the four logical subchannels following the preamble.

3. Downlink Map (DL-MAP):

The DL-MAP describes the DL subframe. By specifying subchannel and OFDMA symbol allocation to each user, the DL-MAP enables MSs to decode the DL subframe. Modulation of DL-MAP is fixed to QPSK, and the coding scheme is specified in the FCH.

4. Uplink Map (UL-MAP):

Similar to the DL-MAP, the UL-MAP describes the UL subframe, and namely bandwidth allocation among the served users. The UL-MAP is embedded in the first DL burst.

5. Downlink/Uplink Bursts:

DL/UL bursts contain data and messages to be transmitted by BS/MS. Each DL burst is mapped with a DL Interval Usage Code (DIUC), and the burst profile is provided in the DL-MAP. Similarly, each UL burst has a UL Interval Usage Code (UIUC), and its PHY characteristics are described in the UL-MAP.

6. Ranging Subchannels:

MSs use ranging subchannels to perform initial ranging, periodical ranging , handover ranging and bandwidth requests. Ranging is a process in which an MS adjusts its PHY parameters according to indicated by BS.

7. Transition Gaps:

Receiver mode and transmitter mode are separated by transition gaps to ensure proper operation. The gap from DL to UL subframe is Transmit Transition Gap (TTG), while the one from UL to DL is Receive Transition Gap (RTG).

2.1.2. Subcarrier Allocation

In OFDMA mode, there are three types of subcarriers:

- Data Subcarriers for data transmission.
- Pilot Subcarriers for coherent detection.
- Null Subcarriers for guard band.

The basic unit of user frequency allocation is subchannel, which is a combination of data subcarriers. Subcarriers can be allocated to subchannels distributed as shown in Figure 2-5, or adjacently as shown in Figure 2-6. For distributed subcarrier permutation mode, subcarriers within a subchannel are chosen pseudo-randomly, and therefore a user’s frequency usage is averagely distributed among the spectrum. This enables user to average inter-cell interference and avoid frequency-selective deep fading. For adjacent subcarrier permutation, subcarriers within a subchannel are adjacent in frequency domain. This helps user to take advantage of frequency-selective fading, and thus improve the throughput.

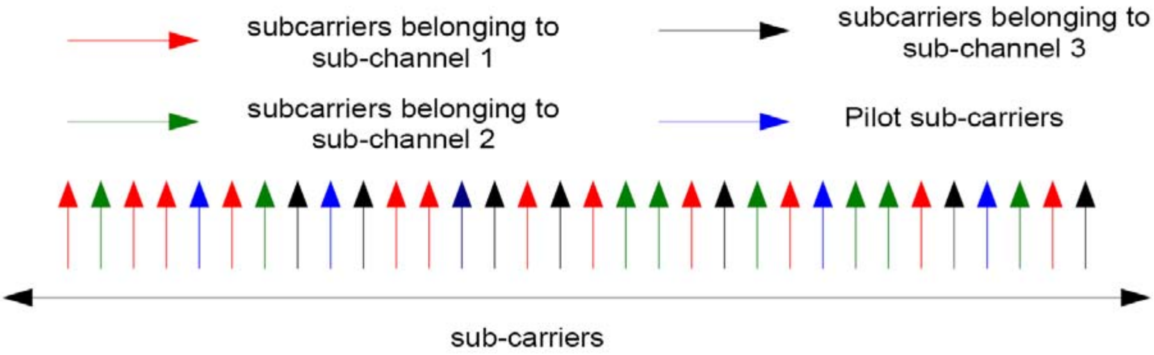


Figure 2-5: An Example of Distributed Subcarrier Permutation [3]

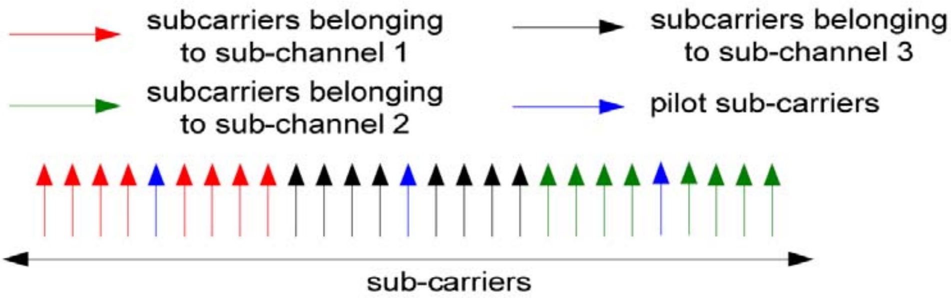


Figure 2-6: An Example of Adjacent Subcarrier Permutation [3]

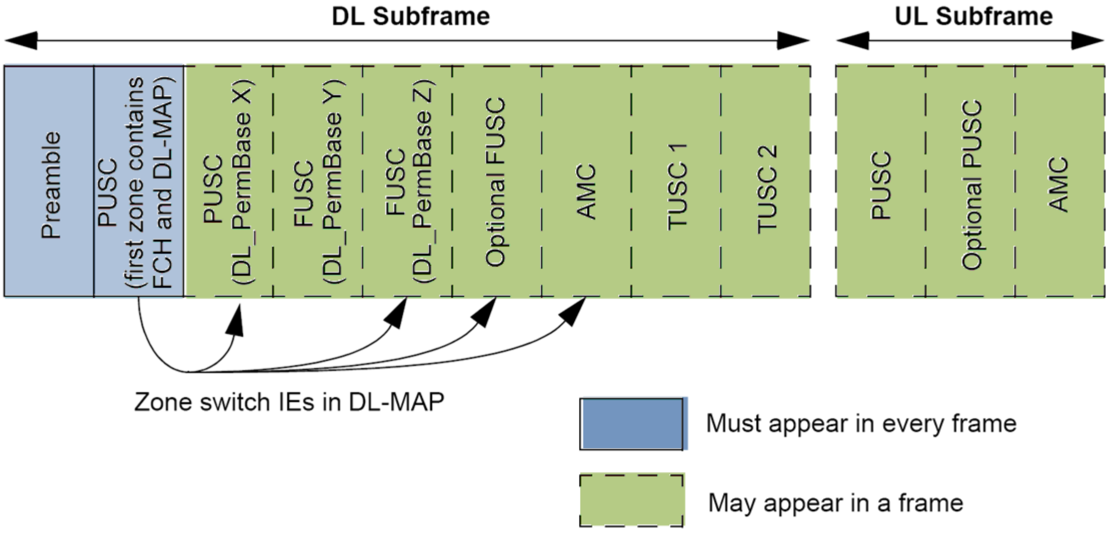


Figure 2-7: Relative Location of Permutation Zones within an OFDMA Frame. [1]

Figure 2-7 indicates relevant location of each permutation zone in a frame. Preamble and DL-PUSC zone are mandatory for every frame, while other permutation zones are optional. In standard 802.16, the following permutation method zones are defined:

1. Preamble Zone:

The whole spectrum is separated as three preamble carrier sets. Each sector within the BS cell selects one carrier set, to identify itself from neighboring sectors.

2. Full Usage of Subchannels (FUSC):

In FUSC mode, a sector uses up all available subcarriers, with distributed subcarrier permutation, to get maximum frequency diversity. However, it requires frequency reuse factor higher than 3. FUSC applies in downlink only.

3. Partial Usage of Subchannels (PUSC):

PUSC provides partial usage of subchannels, with distributed subcarrier permutation. This allows load sharing in the cell-based system. PUSC applies in both downlink and uplink.

4. Band Adaptive Modulation and Coding (Band AMC):

Band AMC provides partial usage of subchannels, with adjacent subcarrier permutation. Band AMC applies in both downlink and uplink.

2.1.3. Data Mapping

The minimum unit for data mapping is a data slot. A data slot is one subchannel wide, and its duration depends on the permutation mode, shown in Table 2-1.

Permutation Mode	Duration (in OFDMA Symbols)
DL-FUSC	1
DL-PUSC	2
UL-PUSC	3
Band AMC	1, 2, 3 or 6

Table 2-1: Slot Duration in Different Permutation Modes

A burst, or data region, is a two-dimension region which covers contiguous logical subchannels and OFDMA symbols. The sequences of downlink and uplink data mapping are shown in Figure 2-8 and Figure 2-9 [1] [2].

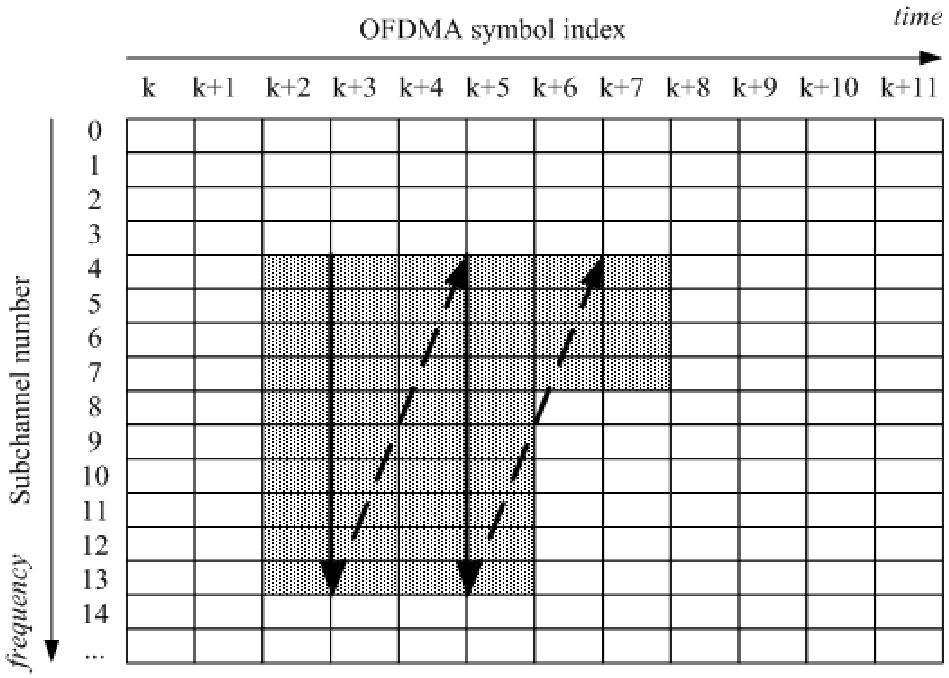


Figure 2-8: Data Mapping in DL-PUSC Mode [1]

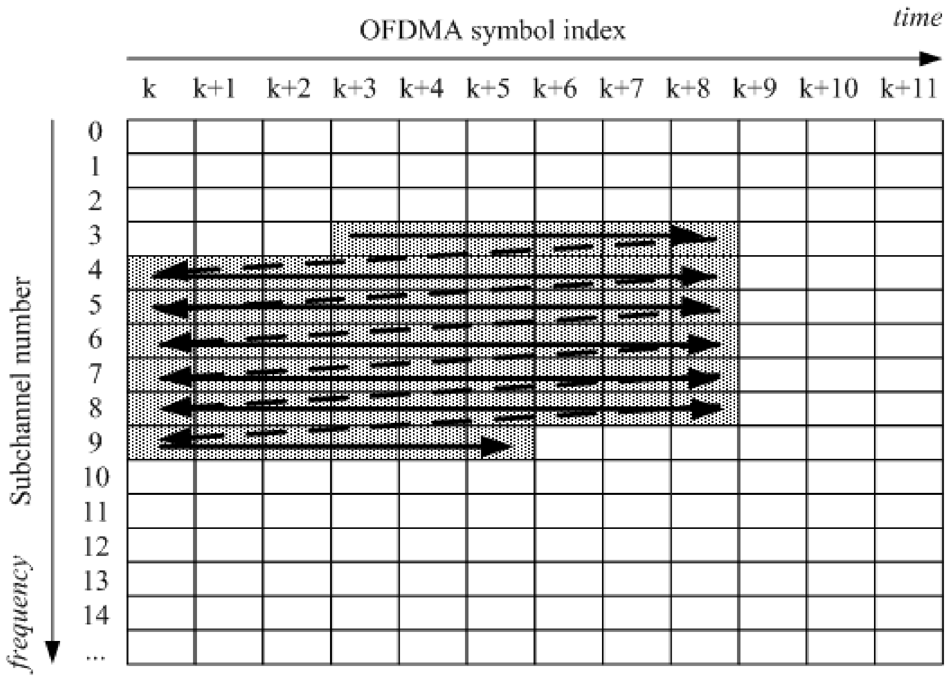


Figure 2-9: Data Mapping in UL-PUSC Mode [1]

2.1.4. Channel Coding and Modulation

Standard 802.16 allows each data burst to have an adaptive coding-modulation scheme. This enables the system to adjust robustness, according to channel quality, data size, transmission distance and purpose. Standard 802.16 specifies the following coding schemes for OFDMA mode: Tail-Biting Convolutional Code (CC), Block Turbo Code (BTC), Convolutional Turbo Code (CTC), Low Density Parity Check Code (LDPC), and Zero Tailed Convolutional Code (ZTCC).

Also, the system may adjust code rate under each channel coding scheme. Standard 802.16 specifies the following coding schemes for OFDMA mode: QPSK, 16-QAM, and 64-QAM. The available coding-modulation schemes are listed in Table 2-2.

Modulation Scheme	**Coding Rate**
BPSK	1/2
QPSK	1/2
	3/4
16-QAM	1/2
	3/4
64-QAM	2/3
	3/4

Table 2-2: Available Coding-Modulation Scheme in WiMAX System

In an OFDMA frame, coding-modulation schemes of some frame elements are fixed: The preamble is always BPSK modulated across subcarriers. The FCH is QPSK modulated, with 1/2 CC and repeats for four times. The DL-MAP is QPSK modulated, with 1/2 code rate under FEC mode specified in FCH. The procedures of channel coding and modulation are shown in Figure 2-10.

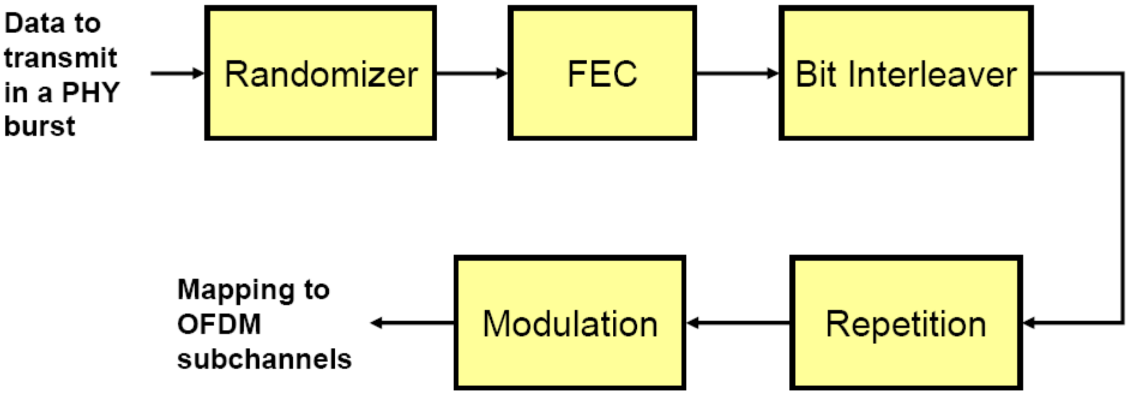


Figure 2-10: Procedures of Channel Coding and Modulation [3]

2.2. MAC Layer Technical Overview

The WiMAX system is equipped with a flexible MAC layer that is able to succeed various services from different upper layers, and support multiple PHY modes with elaborate link adaptation. The MAC is designed for very high bit rates support, and is able to provide suitable scheduling schemes with respect to real time, non-real time and best effort QoS requirements.

A characteristic of WiMAX MAC layer is that it is connection oriented. A connection is distinguished with a 16-bit connection identifier (CID). When performing network entry, an MS sets up multiple connections with the BS. The connections are created based on the services mapped to the MS, including broadcast, management and data transmission services. Each data connection is associated to a QoS level. Connections are dynamically added or dropped if services are initiated or terminated with the MS.

Under OFDMA mode, the MAC layer allocates radio resource in a 2-dimensional way. The OFDMA frame structure allows terminals to dynamically adjust burst profiles, i.e. common transmission parameters over a set of subchannels and OFDM symbols, according to the link conditions. This enables more flexible resource allocation among terminals and is more robust against air interface.

2.2.1. MAC PDU

MAC protocol data unit (PDU) is a data unit for protocol communication between the MAC layers of BS and MS. Basically, data traffic comes in the form of service data units (SDUs) from upper layers. The MAC layer tunnels upper layer traffics without knowledge of the payload content.

A MAC PDU basically includes three parts: the PDU starts with a 6-byte MAC header, followed by payload if exists. The PDU ends with a 4-byte CRC field if required by service flow. In [1], two basic MAC header formats are specified:

- Generic MAC Header: Used in MAC PDUs containing payload data. The generic MAC header indicates length, destination CID, encryption key and included subheader type of the PDU.
- Bandwidth Request Header: Used for requesting uplink bandwidth by MS.

Additional MAC header formats are specified in [2], including:

- BW Request & UL Tx Power Report Header
- BW Request & CINR Report Header
- Channel Quality Indication Channel (CQICH) Allocation Request Header
- PHY Channel Report Header
- BW Request & UL Sleep Control Header
- SN Report Header

As the specified payload size of a PDU may not perfectly match the incoming SDUs, the MAC layer shall perform packing/fragmentation over SDUs to fit the payload space. Each data fragment under packing/fragmentation should be attached with a packing/fragmentation subheader. Figure 2-11 gives examples of the ordering of data fragments and subheaders.

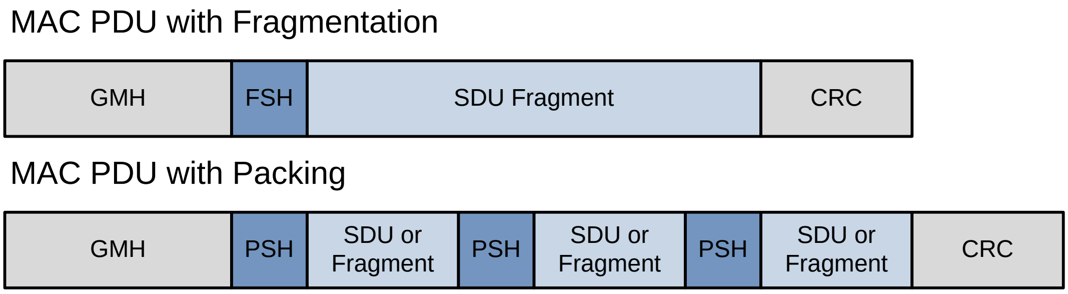


Figure 2-11: Examples of the Ordering of Payload and Subheaders in a MAC PDU

2.2.2. Network Entry

Network entry is the procedures for an MS to acquire, enter and register into the wireless network. The MS performs network entry procedures with the following procedures:

1. Scan for downlink channel. Establish PHY synchronization by decoding PHY preambles.
2. Establish MAC synchronization by decoding DL-MAP message in every frame.
3. Receive uplink and downlink parameters from UCD and DCD messages. Wait for initial ranging opportunity.
4. Perform initial ranging.
5. Perform periodical ranging.
6. Negotiate basic capability by exchanging SBC-REQ and SBC-RSP messages.
7. Authorization and exchange security key.
8. Establish IP connectivity by exchanging REG-REQ and REG-RSP messages.

9. Further upper-layer setups: establish Time of Day; transfer operational parameters, etc.

Figure 2-12 illustrates network entry procedures and involved message flows:

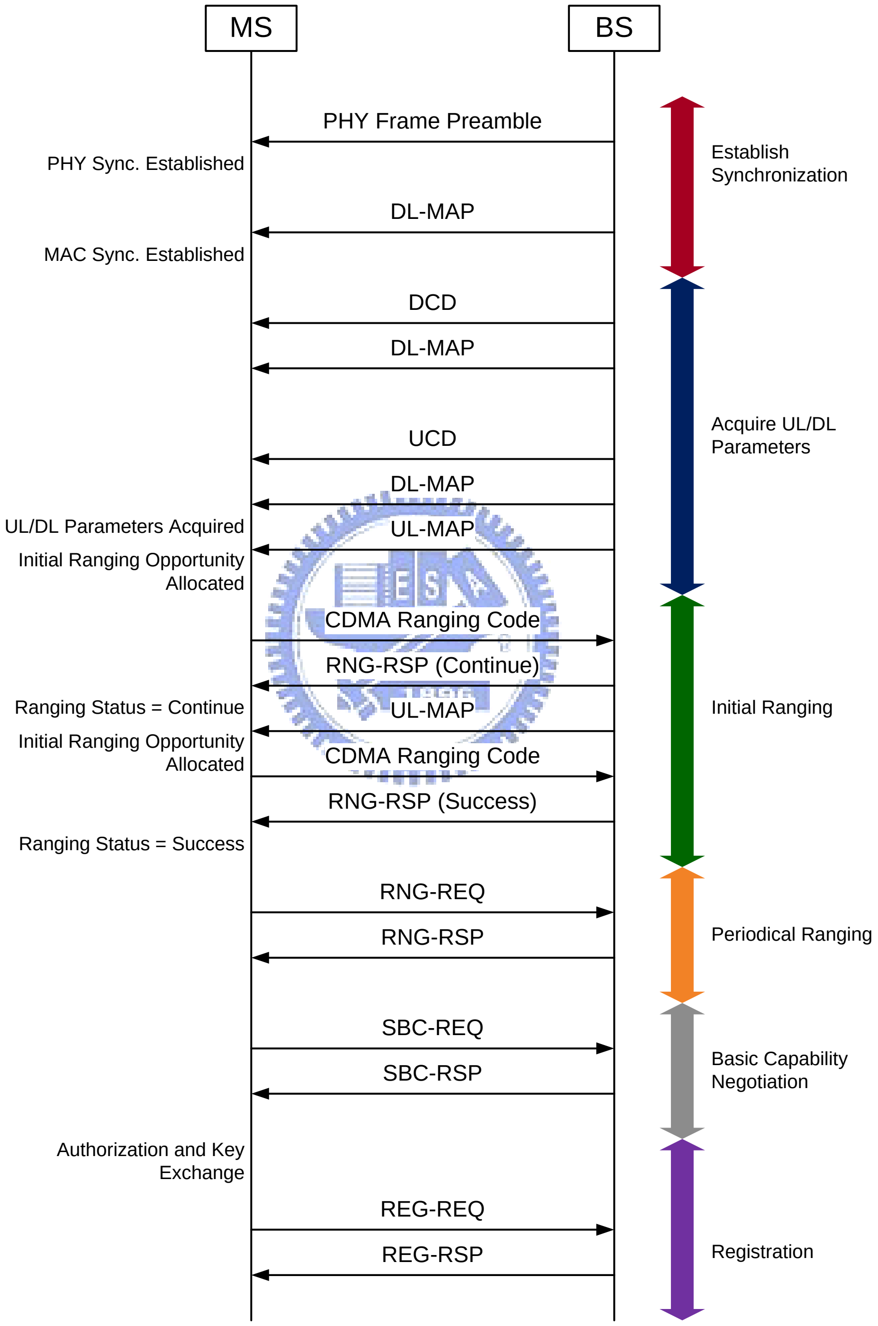


Figure 2-12: Network Entry and Involved Message Flows

2.2.3. Ranging and Initial Ranging

Ranging is a procedure for MS to gain access to the BS, and adjust its transmission parameters such as timing, power and frequency offset with respect to the BS. There are three types of ranging procedure defined in the WiMAX system:

- Initial Ranging for network entry.
- Periodic Ranging for periodic adjustments during normal operation.
- Handover ranging for BS release and a new network entry.

Ranging subchannel, which is allocated in the uplink subframe, is a region designated for ranging use. The relative location and size is specified in the UL-MAP message. The ranging subchannel consists of contention and contention-free region, allowing initial ranging users and periodic ranging users to perform different ranging mechanisms.

A user performs initial ranging following these steps:

1. BS allocates contention based ranging slots for initial ranging in the UL-MAP message. Backoff window size is defined therein. The minimum contention window size is 32.
2. The MSs randomly choose a CDMA ranging code, and a ranging slot to transmit the code. If the BS cannot successfully receive and decode the requests, it would double the contention window size in the next initial ranging opportunity, until it reaches maximum window size.
3. If the BS successfully receives and decodes the requests, it would broadcast a RNG-RSP message that advertises the received ranging code and ranging slot number. It would include in the RNG-RSP message all necessary adjustments (power, timing or even

frequency corrections) and a status field indication continue or success. If the status is continue, the broadcasted user shall continue adjusting its transmission parameters until the status is success.

4. Upon success, the BS would provide an anonymous BW allocation for the MS to send an RNG-REQ message. This allows the MS to perform further steps of network entry.

2.2.4. Data Service QoS Levels

When a data service is setup with the MS, its corresponding connection shall associate to a data service level. Selection of data service level is due to the QoS requirement of the service. There are five data service levels defined in a WiMAX system, each having different delay and rate requirements, and is tended for different applications in upper layers. The goal of scheduler design is to fulfill the needs of users with different data service levels.

The five service levels and their request/grant policies are:

1. Unsolicited Grant Service (UGS):

The BS offers real-time, periodic fixed-size grants to the user. The grant is preemptive, so the MS needs not continuously make bandwidth requests, thus eliminating the request overhead and latency. UGS is used for real-time data services with fixed packet size and periodic arrival, such as VoIP.

2. Real-Time Polling Service (rtPS):

This service offers real-time, periodic, unicast request opportunities to the involved user. It meets the service's real-time requirement while allowing the user to specify the desired bandwidth size. This service requires more request/grant overhead than UGS, but supports dynamic grant that optimizes radio resource efficiency. RtPS is used for real-time data service with variable-sized packets and periodic arrivals, such as MPEG stream.

3. Extended Real-Time Polling Service:

Only defined in [2]. Similar to rtPS but has more stringent requirements on handling jitter. This service is used for real-time data services with variable-sized packets and periodic arrivals, such as VoIP with silence suppression.

4. Non-Real-Time Polling Service (nrtPS):

The service offers unicast polls on a regular basis, assuring the involved user receiving request opportunity. This service is designed for delay tolerant non real-time services requiring a minimum data rate, such as FTP.

5. Best Effort (BE):

Offers no rate or latency guarantees at all. Used for delay tolerant, no rate guarantee services, such as short message service (SMS).

2.2.5. Mobility and Handover

Handover functionality enables the WiMAX system to support mobility. Handover is enabled by a series of MS/BS operations:

1. Network Topology Advertisement:

Before handover decision, the MS shall acquire network topology. First, the BS broadcasts network topology information using MOB_NBR-ADV message. This message is a compilation of DCD/UCD information of neighbor BSs. The MS then is able to synchronize to neighbor BSs without listening to them one by one.

2. MS Scan:

Upon receiving network topology, the MS then requests for scan periods using MOB_SCN-REQ/RSP message. When scan periods are allocated by the serving BS, the MS synchronize to the downlink of neighbor BSs, and evaluate received downlink signal quality for handover decision. During scan, the MS does not listen to the serving BS, so incoming data for the MS is buffered at the serving BS.

3. Association:

To obtain complete PHY information of a neighbor BS, the MS may perform optional association, i.e. handover ranging, with the neighbor BS. The information exchanged during handover ranging is kept at the neighbor BS, and may be reused for the future handover. The ranging process may be contention or non-contention based, and is much similar to initial ranging process.

4. Handover:

The MS keeps a candidate list and performs cell reselection. When the handover conditions are met, the MS would send a handover request to the serving BS using MOB_MSHO-REQ message. The BS would reply with MOB_BSHO-RSP message indicating if the handover is allowed or not. If handover is allowed, the MS may send a MOB_HO-IND message at any time signaling release of serving BS.

5. Network Re-entry:

The network re-entry procedures are much similar to the initial ranging ones performed at MS power-on. However, since the new serving BS may acquire information of the MS before handover through pre-negotiation or backbone network, the network re-entry procedures can be optimized. The new serving BS would indicate the MS to skip one or several network entry steps using the RNG-RSP message.

Figure 2-13 is an illustrative message flow diagram showing the message flow during an MS-initiated handover process. The message flow starts at network topology advertisement, and ends at target BS's RNG-RSP.

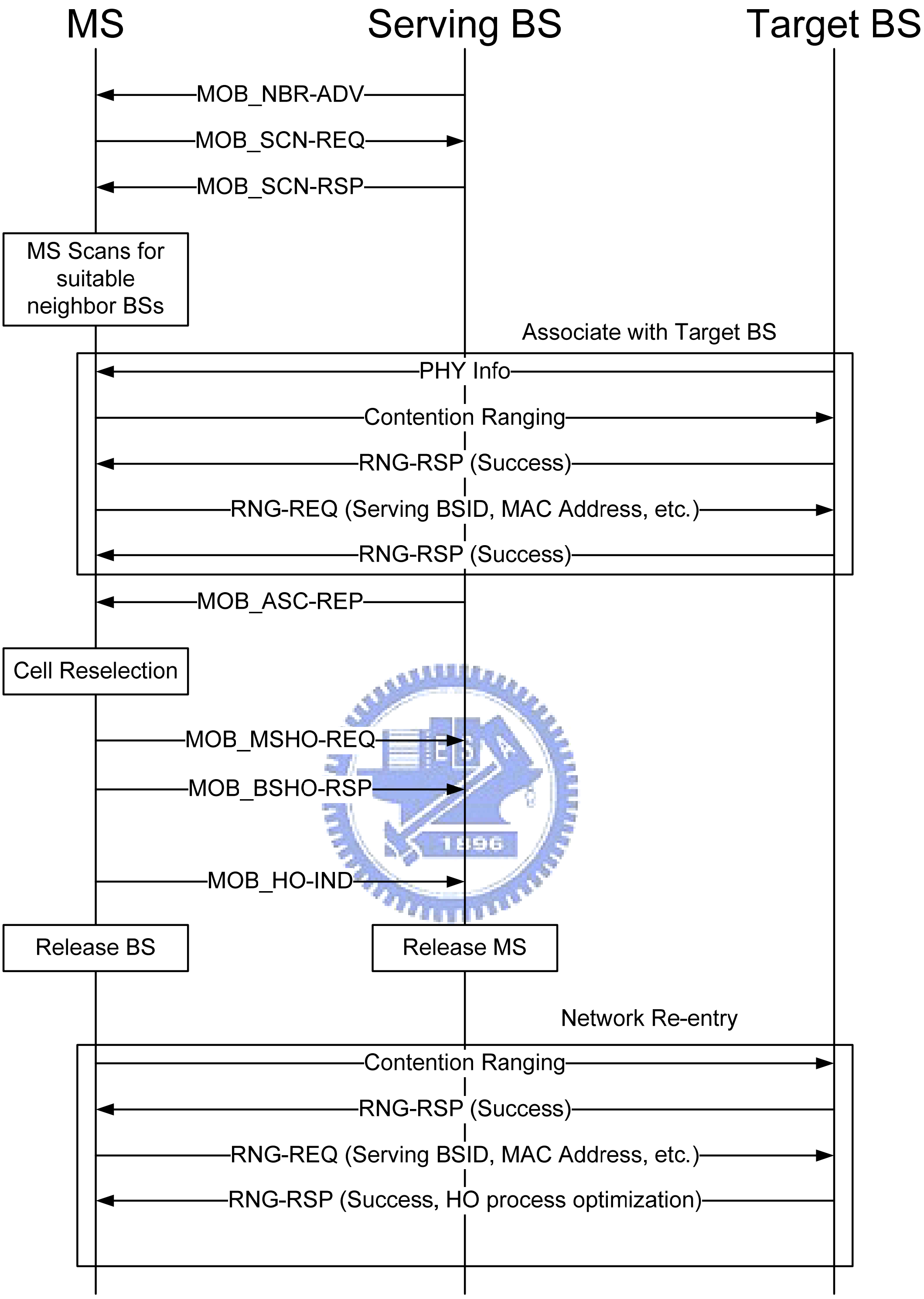


Figure 2-13: Handover Message Flow

Chapter 3. Proposed Overall MAC System Architecture

In this chapter, architecture of a mobile WiMAX system MAC layer is proposed. The proposed architecture includes the base station (BS) and the mobile station (MS) side. First, this article shall introduce the basic three-plane model that applies to either the BS side or the MS side. After that, there are graphs that illustrate the system architecture, and discussion of the main modules in it.

3.1. Basic 3-Plane Model

The system architecture is built based on a basic three-plane model, shown in Figure 3-1. The model consists of the following planes: the data plane, the control plane and the system database.

1. Data Plane:

The data plane deals with traffic data coming from upper layers or other traffic sources. It performs functionalities that directly access the data, including classification, buffering, packing/fragmentation, headers and concatenation.

The data plane is the executing part of the protocol layer. It handles incoming data according to the indications given by the control plane, and executes the control plane's commands by sending management messages to the air interface. Conformance to the standard depends on proper operation of the data plane

In practical designs, the data plane engages most of the software execution time. It requires careful implementation to meet the software timing restriction of the standard, or the system will fail to work properly.

2. Control Plane:

The control plane is in charge of controlling the operation of the MAC layer. Its operation mainly includes radio resource management (RRM), indications of data plane operation, and optimization of parameters.

The control plane is supported by various management entities. A management entity is basically a combination of some control mechanism, most of them operated by a state machine, and the algorithm inside it. In designing a complicated system like mobile WiMAX, the internal of a management entity requires continuous verification and modification. It is desired to develop a design methodology that enables the management entities to be rapidly prototyped, and easily modified with minimum intrusion to the protocol stack.

In practical designs, the control plane has relatively little influence to software execution time. The influence of the control plane is mainly related to the efficiency and throughput of the wireless system.

3. System Database:

The system database is the space for storing all parameters of the protocol stack. The environment and behaviors of network devices are abstracted to parameters and stored in the system database. The system database includes a collection of standard access interface functions. Entities that wish to access the system database has to call these functions, to prevent access conficts, thus guarantee that the parameters are correct and updated.

The system database has to be well-organized and modularized, to facilitate further maintenance of the system database and management entities that accesses the database.

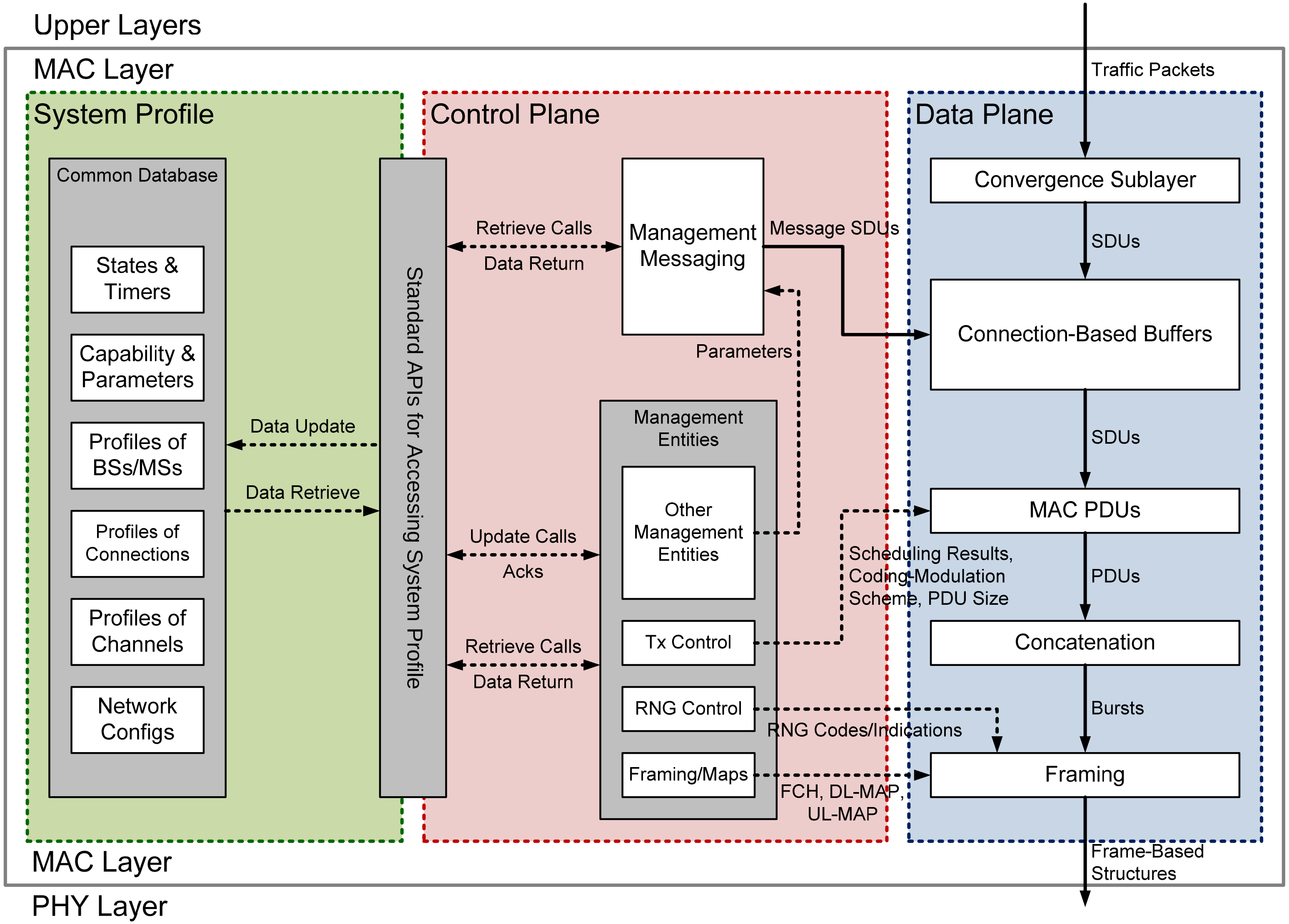


Figure 3-1: The Basic 3-Plane Architecture for Transmitter End

3.2. The Proposed System Architecture

The proposed system architecture follows the basic 3-plane architecture described in last section. As an example, the MAC design architecture of BS transmitter (downlink) side is shown in Figure 3-2.

The data plane deals with traffic from upper layers or generated by the MAC layer itself. The convergence sublayer (CS) resides on top of the data plane, which converts different format packets coming into uniformly formatted MAC SDUs. Then the SDUs are buffered in connection-based buffers. The PDU generator selects from these connections according to the scheduling results of the control plane, and performs necessary packing/fragmentation. The framing unit then concatenates PDUs into a frame information package containing multiple data bursts, and sends the frame information package to the PHY layer.

The control plane is the center of all control mechanisms and optimizations. A body that realizes control mechanisms is called a management entity (ME). MEs are the main constructing blocks of the control plane. If the control plane needs to send a message to the air interface, it triggers the messaging unit, which is in charge of gathering all necessary parameters in the message, and concatenates all message fields into a binary string.

The system database is a memory space that stores the parameters shared by the protocol layers in the device. The parameters are categorized into multiple objects, which are called profiles. The control plane may access system database through a well-defined standard interface set, while the data plane has no direct access to the system database since there’s no need for direct access, and access to the system database may cause complexities for hardware realization.

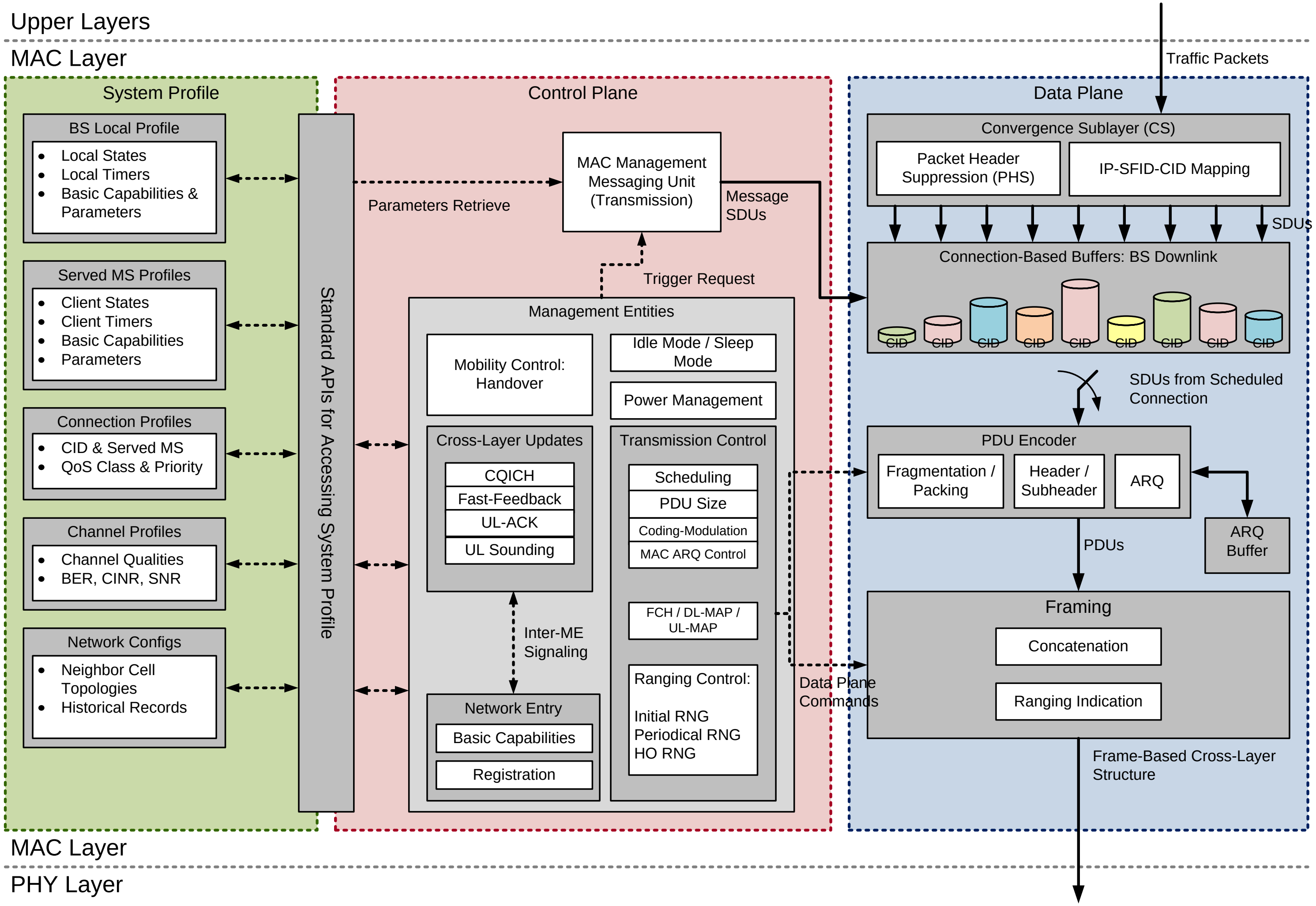


Figure 3-2: Proposed Architecture for BS Downlink

3.3. The Data Plane

3.3.1. Convergence Sublayer (CS)

The convergence sublayer resides on the top of the MAC layer data plane. It is the module that directly accesses traffic packets from/to upper layers. The operation of CS is shown in Figure 3-3. The CS performs the following functions:

1. IP-SFID-CID Mapping and Packet Classification:

The mobile WiMAX system is a connection-based protocol. Every service is given a service flow ID (SFID), and mapped to a MAC layer connection. Every MAC layer connection is labeled with its connection ID (CID). A mobile station engages multiple connections.

The CS at BS downlink reads the destination address and service type, and looks up the IP-SFID mapping table stored in the system database. It then accesses the system database again to look up the SFID-CID mapping table to get the corresponding connection. Since a user would request more than one service, an IP address may associate to multiple service flows, thus multiple SFIDs. SFID and CID are one-to-one mapped.

2. Service-Specific Processing and Packet Header Suppression (PHS):

The convergence sublayer accepts packets sourcing from different services. It has to reform different formatted packets into uniform format MAC SDUs; this requires service-specific processing. In Standard 802.16-2004, two types of service-specific sub-CSs: Cell CS for the Asynchronous Transfer Mode (ATM) services, and Packet CS for the IP-based services.

PHS is one of the service-specific processing functions. In PHS, repetitive portions of the IP packet/ATM cell headers are suppressed to save system bandwidth, before forming MAC SDUs. The suppressed headers are then restored at the receiver side.

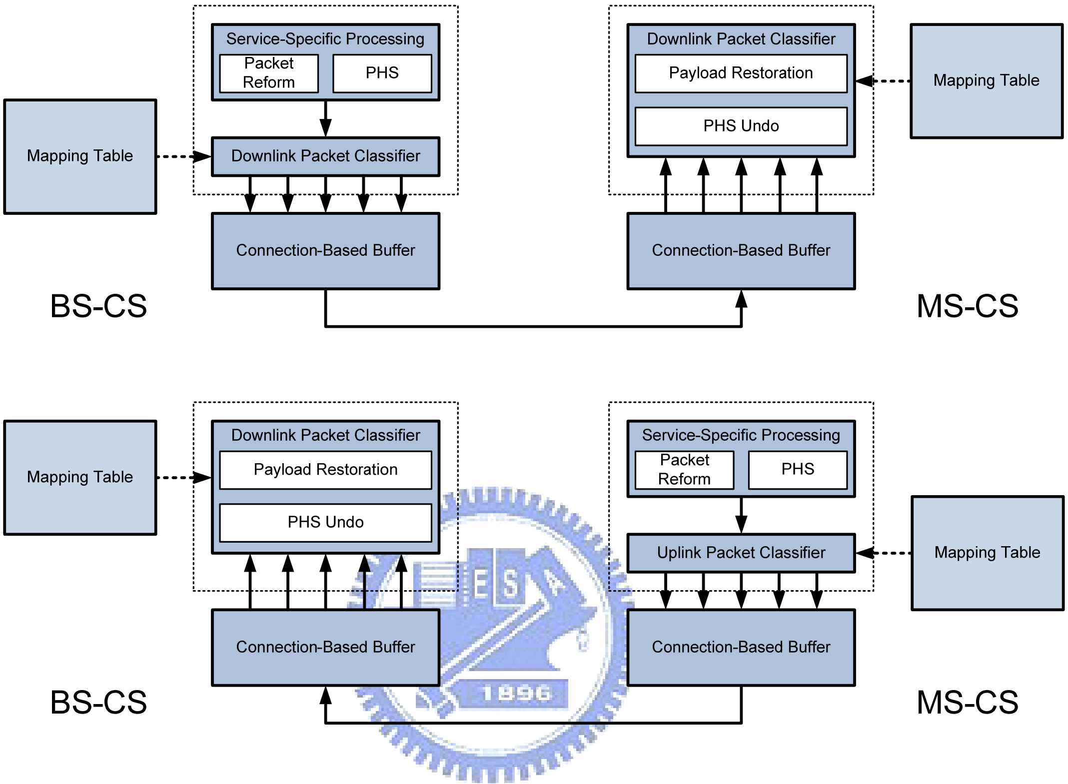


Figure 3-3: Operations of Convergence Sublayer

3.3.2. Connection-Based Buffer

Mobile WiMAX is a connection-based protocol; each connection is mapped to a specific service flow or management flow. Connection-based buffer is the buffer place for incoming MAC SDUs. At the transmitter end, packets are classified and reformed by CS, and dispatched to designated SDU queue. At the receiver end, the CS retrieves SDUs from designated SDU queue.

Connection-based SDU queues are distinguished by the connection ID (CID). For the mobile WiMAX system, the connection ID is 16-bit long, so there are a maximum of 65536 SDU queues in the BS architecture. There are three types of CIDs allocated between an MS and its serving BS:

- Basic CID: For management message types that are time-stringent.
- Primary management CID: For management message types that are less time-stringent.
- Transport CIDs: Dynamically allocated to data services.

Note that two CIDs are reserved for specific use and never allocated to MSs:

- CID 0 (all zeros): Initial ranging CID.
- CID 65535 (all ones): Broadcast CID.

Each connection is equipped with a SDU queue. Although not the usual case, SDUs arriving later than other packets in the same connection may have an earlier deadline. Therefore, data structure with priority consideration is necessary. Heap structure is recommended. Scheduling-related parameters, such as deadline, service type and priority shall be included in the SDU element structure. These parameters are given by the convergence sublayer. The SDU element shall also include an indicator that indicates the SDU is fragment/packing allowed or not. Accessed both by the CS, the PDU generator and the messaging unit in the control plane, the connection-based buffer shall be equipped with a set of standard access APIs: Element Insert, Element Retrieve and Element Delete.

3.3.3. PDU Generator

The PDU generator generates PDUs from queued SDUs. The PDU generator includes four functionalities:

- SDU selection according to scheduling result.
- Packing/Fragmentation.
- Header/Subheader.
- CRC.

The PDU generator receives indications from transmission control unit in the control plane, including selected CID returned by the scheduler unit, coding-modulation scheme returned by the MCS unit, and PDU size by the size decision unit. The relationship of the PDU generator and the transmission control unit are the function caller and the callee. The interface involved with the PDU generator is shown in Figure 3-4.

When the PDU generator retrieves SDU from the connection-based buffer, it always accesses the SDU that has earliest deadline in queue. When performing packing/fragmentation, it extracts the needed fragment from the SDU. If the SDU runs out, the PDU generator calls the element delete function and retrieves from the next SDU.

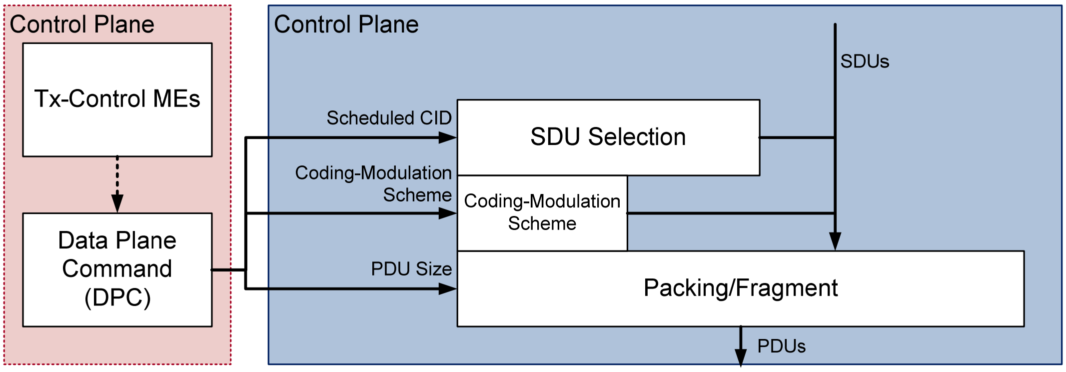


Figure 3-4: PDU Generator Architecture

3.3.4. Framing

The framing unit concatenates PDUs into data bursts, and integrates all necessary elements to form a frame-based structure. The MAC data plane shall communicate with the PHY layer through this frame-based structure. The architecture of framing unit is shown in Figure 3-5. The framing unit performs the following functionalities:

– Frame Elements Collection:

The framing unit shall receive data plane command (DPC) from the transmission control management entities in the control plane. The command is a data structure that collects frame control header, DL-MAP and UL-MAP and every element needed by the frame structure.

– PDU Concatenation:

The framing unit accesses the mapping unit in the control plane for mapping indication, and performs PDU concatenation according to the indication. The mapping unit shall be triggered periodically in every frame.

– Frame Structure Delivery:

After collecting all required units in the frame and the data bursts, the framing unit shall integrate them into a frame-based structure, and pass it to the PHY layer.

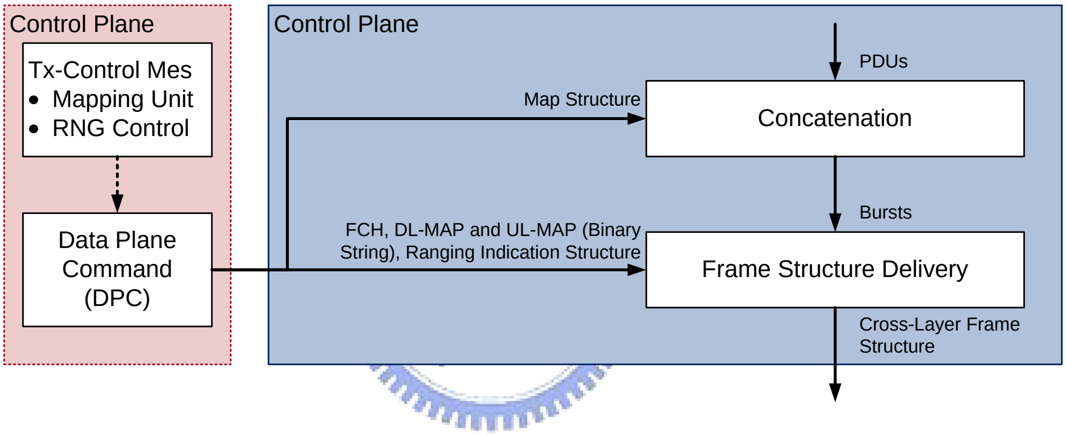


Figure 3-5: Architecture of Framing Unit

3.4. Control Plane

3.4.1. Management Entities (ME)

A management entity is a module that performs radio resource management. It is usually supported by a control mechanism, i.e. finite state machine, and its background algorithms. In practical design, an ME is a subroutine function that can be called or call other ME subroutines. Different MEs may communicate through predefined internal data structures.

In order to provide prompt and optimized management, the management entities shall permanently retrieve and update related parameters stored in the system data base. Therefore, a standard interface between management entities and the system data base shall be specified. The interface linked with a management entity depends on its functionality. There are five interface types involved in our proposed architecture of management entities:

1. Subscribe/Notification (SUB/NOT):

The SUB/NOT interface type exists between a management entity and the system database. This relationship is built if the former requires the latter to provide automatic notifications on parameter changes or periodical timeouts.

The management entity may subscribe to one or more parameters stored in the database. It has to specify the notification type in the subscription request:

– Notify on any change of the parameter.

– Notify if the parameter changes over some threshold.

– Notify periodically.

– Notify on a preset time spot.

2. Start Timer/Stop Timer/Timeout (STA/STO/OUT):

The STA/STO/OUT interface type exists between a management entity and the system database. This is a special interface type that enables the MEs to set or terminate timers in the system database. When the timer runs out, the system database shall also automatically issue a timeout notification to all involved MEs.

3. Retrieve/Return (RTV/RTN):

The RTV/RTN interface type exists between the management entity and the system database. This interface type is simply used for reading parameters from the system database. Note that the system database may give a "failed" return if it considers the parameter out-of-date and shall no longer be used.

4. Update/Acknowledgement (UPD/ACK):

The UPD/ACK interface type exists between the management entity and the system database. This interface type is simply used for writing parameters to the system database.

5. Trigger/Complete (TRI/CMP):

The TRI/CMP interface type exists between two MEs, or between an ME and the messaging unit in the control plane. The trigger request is a subroutine call. It passes to the callee a predefined data structure containing required parameters. After the finishing its work, the callee shall return a complete signal to the caller.

6. Data Plane Command (DPC):

The DPC interface type exists between an ME and the data plane. The DPC is a command data structure, including data plane related parameters, such as scheduling result, coding-modulation scheme, PDU size, DL-MAP/UL-MAP and ranging instructions.

An illustrative graph of an ME's interfaces is shown in Figure 3-6.

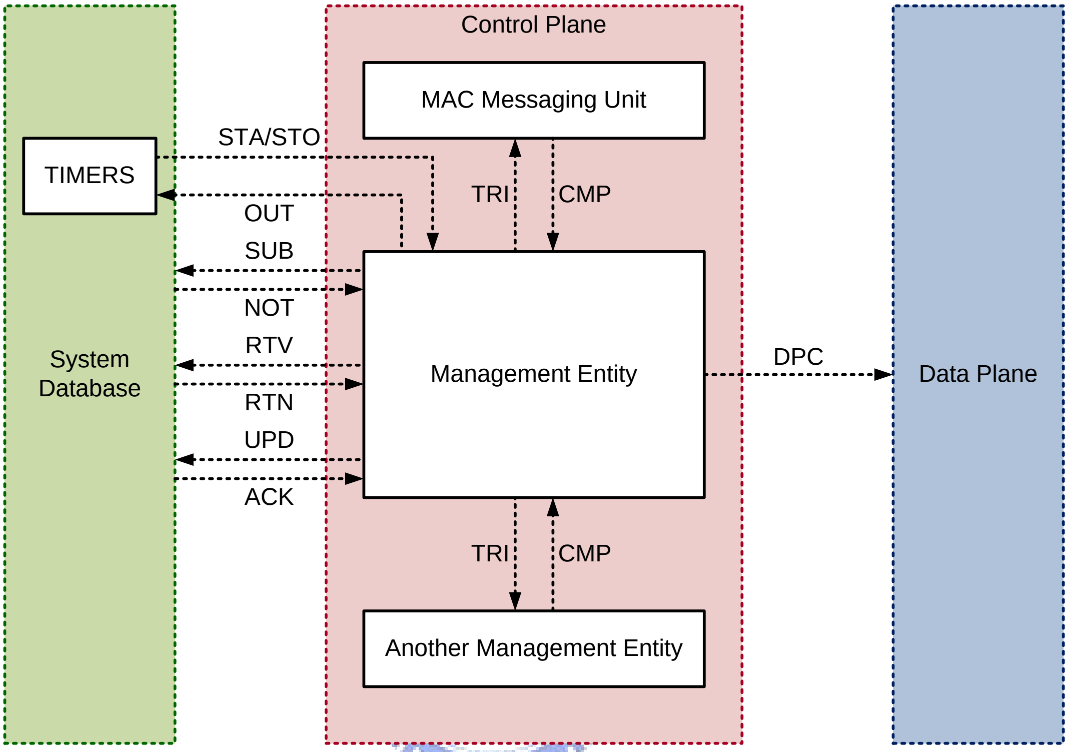


Figure 3-6: Interfaces Involved by a Management Entity

3.4.2. Transmission Control Management Entities

The Tx-control MEs is a subset of MAC MEs that directly control the activity of the data plane. The Tx-control MEs may trigger other MEs to make further optimization, or retrieve from the system database for necessary information. Transmission control entities include:

- Scheduler
- Coding-Modulation Scheme Optimizer
- PDU Size
- MAC ARQ Control
- Mapping (FCH, DL-MAP, UL-MAP)
- Ranging Control (Initial, Periodical and Handover Ranging)

The Tx-control MEs communicates with the data plane through a predefined data plane command (DPC). The DPC is a unified data structure that collects all necessary parameters and arguments for data plane operations. The relationship between Tx-control MEs and the data plane is illustrated in Figure 3-7.

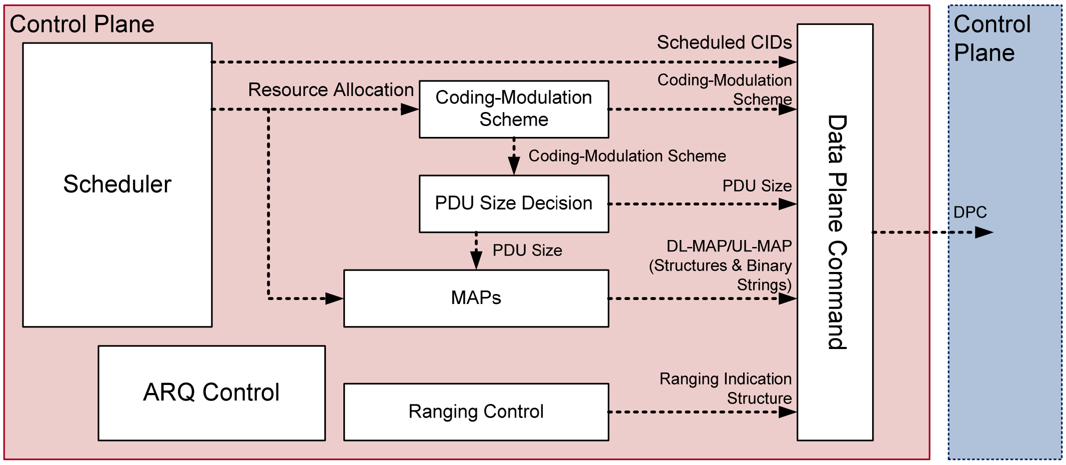


Figure 3-7: Tx-Control MEs and Relationship with the Data Plane

3.4.3. Messaging Unit

The messaging unit is the producer of MAC management messages (MMM). When an ME needs to transmit a MAC management message, it has to send a trigger request the messaging unit. The messaging unit produces message SDUs through the following procedure:

1. Retrieve and organize all required parameters.
2. Concatenate all message fields and type-length-value fields (TLVs) into a linked list.
3. Combines message fields and TLVs into a binary string.
4. Make the message SDU.
5. Insert the SDU into the connection-based buffer.

For the receiver end, the binary string is directly decoded into messages field and TLVs. The procedure and interface is illustrated in Figure 3-8.

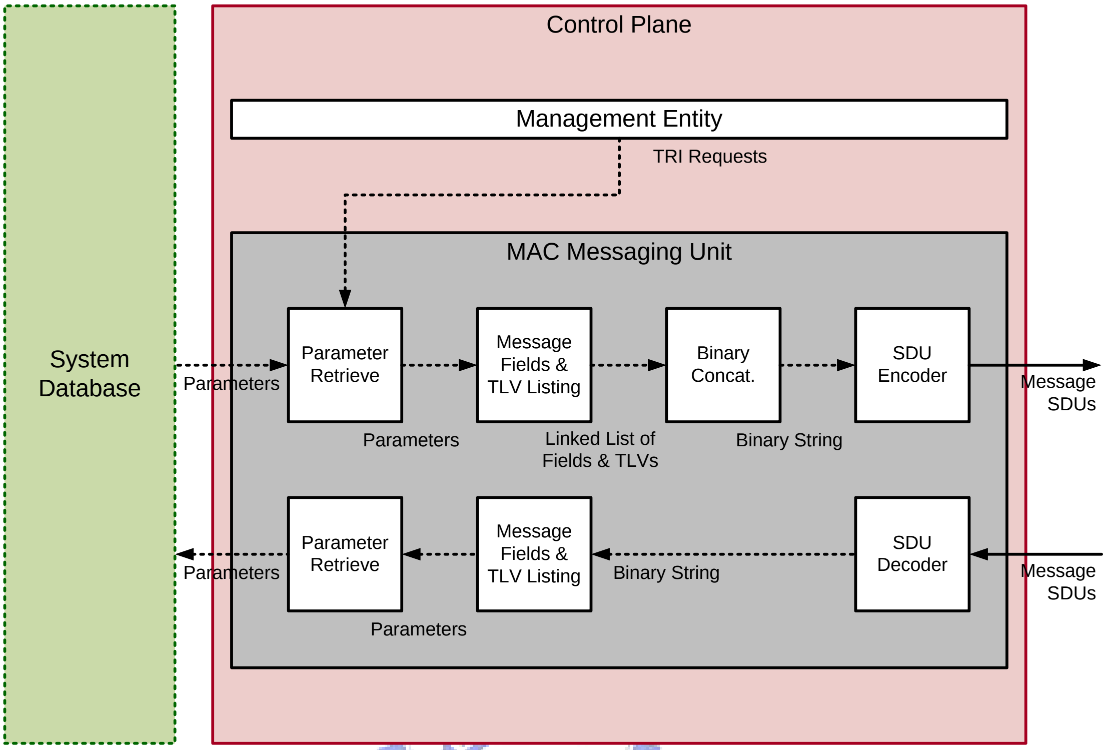


Figure 3-8: Internals of the Messaging Unit

3.5. The System Database

The system database is a storage place for all parameters used in system operation. The system database contains a number of profiles. A profile is a category that includes all parameters related to an object, or a specific use. The system database is accessed by every protocol layer on the device.

3.5.1. System Database at the BS Side

Note that the BS keeps a timer and a state profile in the BS local profile. It keeps a client timer profile and a client state profile for every MS served, too. The system database on the BS is suggested to be categorized as follows:

Profile	**Contents**
BS Local Profile	Parameters and arguments of the BS itself. A BS local profile may include: ● BSID ● Basic capability of the BS ● Network configuration of the BS ● Timer Profile: timers for the operation of the BS itself ● State Profile: FSM States for the operation of the BS itself
Served MS Profiles	Every served MS must register a served MS profile at the

	BS end. A BS may engage multiple served MS profiles. A served MS profile may include: ● User MAC address ● Basic capability of the MS ● Network configuration of the MS ● List of associated CIDs and their service types: basic , primary and transport ● Signal quality parameters measured of the BS-Served MS couple; ranging parameters ● Client timer profile ● Client states profile
Neighbor Set Profiles	The BS keeps these profiles when doing pre-negotiation with neighbor BSs. This profile type is dynamically declared and released on the BS's demand. A neighbor set profile may include: ● BSID ● Pre-negotiation information
Connection Profiles	Every connection associated to the BS must register a connection profile. A BS may engage multiple connection profiles, and a served MS profile may a list of multiple CIDs. A connection profile may include: ● CID ● Associated IP and SFID ● Service type (management, primary or transport), QoS level (UGS, rtPS, ertPS, nrtPs, BE), protocol

	(HTTP, FTP, MMS…etc.), priority, and any scheduling related parameters ● Queue load of the connection’s SDU buffer
Channel Profiles	Every subchannel must have a channel profile at the BS end. The number of channel profiles is fixed. A channel profile may contain the following elements: ● Band AMC channel profile ● Diversity channel profile
IP-SFID-CID Table	The BS must keep an IP-SFID-CID lookup table for its convergence sublayer to search quickly.
DCD/UCD Profiles	The BS may keep a number of preset DCD/UCDs distinguished by a configuration change counter. These profiles are defined so that they are ready for message unit to retrieve, and then convert to DCD/UCD messages. The DCD profiles should contain the following elements: ● Downlink Burst Profiles ● Information for the overall channel, which is in consistence with the channel profiles. The UCD profiles should contain: ● Uplink Burst Profiles ● Information for the overall channel, which is in consistence with the channel profiles ● Ranging Configurations

Table 3-1: BS Side System Database Categories

3.5.2. System Database at the MS Side

The system database on theMS is suggested to be categorized as follows:

Profile	**Contents**
MS Local Profile	Parameters and arguments of the MS itself. A BS local profile may include: ● User MAC address ● Basic capability of the MS ● Network configuration of the MS ● List of associated CIDs and their service types: basic, primary and transport CIDs. ● Timer profile: timers for the operation of the MS itself ● State profile: FSM States for the operation of the MS itself
Serving BS Profile	The MS keeps its serving BS's information in this profile. There should be only one serving BS profile at the MS. A serving BS profile may include: ● BSID ● Basic capability of the BS ● Network configuration of the BS ● Signal quality parameters measured between the serving BS and the MS ● Client state profile

	● Client timer profile
BS Profiles	The MS keeps these profiles whenever it needs to select among a number of BSs, e.g. performing network entry and cell reselection. BS profiles are dynamically allocated and released on the MS’s demand. A BS profile may include: ● BSID ● Measured signal quality and other channel quality parameters ● BS basic capabilities ● Other pre-negotiated information The MS shall keep four sets of BS profiles: ● Serving BS (with only one BS inside) ● HO candidate BSs ● Active set BSs ● Associated BSs The BS sets are simply a lookup table that contains pointers to each BS profile.
Connection Profiles	Every connection associated to the MS must register a connection profile for the reference of scheduling. An MS may engage multiple connection profiles. A connection profile may include: ● CID ● Associated IP and SFID ● Service type (management, primary or transport),

	QoS level (UGS, rtPS, ertPS, nrtPs, BE), protocol (HTTP, FTP, MMS…etc.), priority, and any scheduling related parameters ● Queue load of the connection’s SDU buffer
Channel Profiles	Every subchannel must have a channel profile at the BS end. The number of channel profiles is fixed. A channel profile may contain the following elements: ● Band AMC channel profile ● Diversity channel profile
DCD/UCD Profiles	The MS shall keep DCD/UCD with the latest configuration. When a DCD/UCD with new configuration arrives, the MS shall quickly replace the saved DCD/UCD profiles. The DCD profile should contain: ● Downlink Burst Profiles ● Information for the overall channel, which is in consistence with the channel profiles. The UCD profiles should contain: ● Uplink Burst Profiles ● Information for the overall channel, which is in consistence with the channel profiles ● Ranging Configurations

Table 3-2: MS Side System Database Categories

3.6. Design Considerations

As the complexity of modern BWA systems inflates, the requirement for system performance becomes more stringent, too. Also the maintenance of such a complicated software system is more difficult. Due to the constraints and requirements, there are a few rules to follow in practical design of a mobile BWA system. Some of them are illustrated in the follows:

1. Different Design Requirements at the BS and the MS:

Due to the difference in hardware capability and constraints on processing speed and power consumption, different design policies are suggested at the BS end and the MS end.

For the BS end, the memory space is almost unlimited. However, the processing speed requirement is high, since the BS may need to deal with highly repetitive works from multiple served MSs such as: packet convergence and classification, ARQ control, in-and-out handover, channel quality refresh…etc.

For the MS end, the limit of memory space in an embedded system is very strict. The number of devices the MS listens to is few: the serving BS and some active set BSs (mostly those of one-tier cells). The number of processed packets by an MS is several orders lower than that by a BS.

In view of this, it is suggested that the memory allocation policy at the BS end should be more static. The resource remain timer should set longer, and data structures shall not be declared and erased frequently. In contrast, the memory allocation policy at the MS end should be dynamic. Once unnecessary, the MSs shall release the memory spaces as soon as possible.

2. Object Oriented Design for System Database:

It is suggested to implement the system database in an object oriented manner. That is, the system database consists of building blocks that are mapped to real-world entities, such as BSs, MSs and channels. System database objects, e.g. BS profiles and MS profiles, should be dynamically declared and deleted.

This enables dynamic memory allocation, which enables embedded system operation. Object oriented design also minimizes memory access overhead when the system is going to move profiles from one set to another. When a management entity (ME) wishes to retrieve from the system database, it may access the entire profile for all needed parameters, so that repetitive retrieval to the system database is avoided.

Chapter 4. Proposed Cross-Layer Design

4.1. Concept of Cross-Layer Design

Layered architecture has been long the dominating design methodology of either wired or wireless communications systems. The basic principle of layered protocol stack is modularity, also called layer-independence [5]. Following the modularity principle, a layered protocol stack forbids direct communication between nonadjacent layers, while communication between adjacent layers is limited to procedure calls and responses [6].

Traditional layered architecture provides an adequate guideline for system implementation. Designers divide the whole system into several layers each assigned to a design team, and specify the function call flow and the related information exchange. This is suitable for most cases in wired systems, since the transmission channel is mostly stable, error probability is low, and error occurrences can be simply reasoned as congestion problems.

However, this is not the case for wireless systems. In wireless systems, the channel condition varies with time. Channel condition causes problems as well as network congestion does. Here's a well-known case of TCP packet errors. Designed for wired systems, a basic assumption of TCP protocol is that any packet error is caused by network congestion. However, in wireless systems the errors may be caused by bad channel condition. If the protocol stack has no knowledge of the channel condition, it may issue a false alert of network congestion, which simply makes things worse [7].

On the optimistic side, wireless environment also provides a way for opportunistic resource allocation [6]. One of the typical cases is the water-filling algorithm. The base station allocates more radio resource to the user experiencing better channel quality, thus makes use of multiuser diversity. Water-filling is the main scheme to boost overall transmission throughput, but guarantees neither minimum rate nor fairness to any user, which should be provided by upper layers in the protocol stack.

The upper layers of a protocol stack needs to derive knowledge of lower layers, either to provide a preventive mechanism from error, or an opportunistic resource allocation to gain better performance. The lower layers are the same. For example, the MAC layer has to know which type of service (HTTP, VoIP, SMS, multimedia streaming…etc.), in order to select a proper QoS class, and labels it to the corresponding MAC connection. Also, the PHY layer has to receive indications from the MAC layer to adapt its transmission parameters accordingly.

In view of this, although partially based on the layered architecture, designers tend to violate strict constraints of traditional layered architecture by introducing new cross-layer interfaces to new wireless systems. In the following part of this chapter, first we will discuss the design goal of a good cross-layer methodology. In the second place, prior works on cross-layer methodologies are presented. After that, existing cross-layer issues of WiMAX system design is listed. Finally, we come up with a proposed, mixed-type cross-layer methodology for WiMAX system design, and discuss its details.

4.2. Cross-Layer Design Goals

To facilitate quick and effective system design, cross-layer interface must be planned carefully. Previous researches such as [8] depict a few design goals for cross-layer feedback architecture. In this article, significant guidelines for cross-layer design are shown as follows:

1. Rapid Prototyping

In developing software for a communications system, it is necessary to build a baseline framework in the early stage. The framework provides compliance to the system specification, and secures interoperability. After this, designers shall develop new cross-layer optimizations, and add them to the code through a methodology that won't violate system compliance.

2. Modularization

The interface between the protocol stack and cross-layer optimizations shall be modularized. The cross-layer optimization modules shall have straightforward, configurable inputs and simple output formats. These leads minimum intrusion to the protocol stack codes. Through modularization, designers can maintain the system software more efficiently, and preserve system correctness.

3. Execution Efficiency

Every optimization function call brings overhead to system execution, while the computation inside the optimization is also a source of loading. For high-speed wireless communications systems, the performance requirements are hard, so unnecessary function calls have to be reduced.

4.3. Cross-Layer Design Topologies and Methodologies

There have been a number of researches targeting on cross-layer design lately. Most of them provide viewpoints only from several specific layers in the protocol stack, but not all of them. Also, most of the researches target on radio performance issues; discussions on implementation issues are weak.

In the following part of this chapter, we shall discuss prior works on cross-layer methodologies from a more holistic viewpoint, and concentrate on system implementation issues. First, cross-layer topologies are depicted. After that, there is a collection and re-interpretation of related proposals of cross-layer methodologies. Finally, we point out some open challenges for future cross-layer designs.

4.3.1. Cross-Layer Topologies

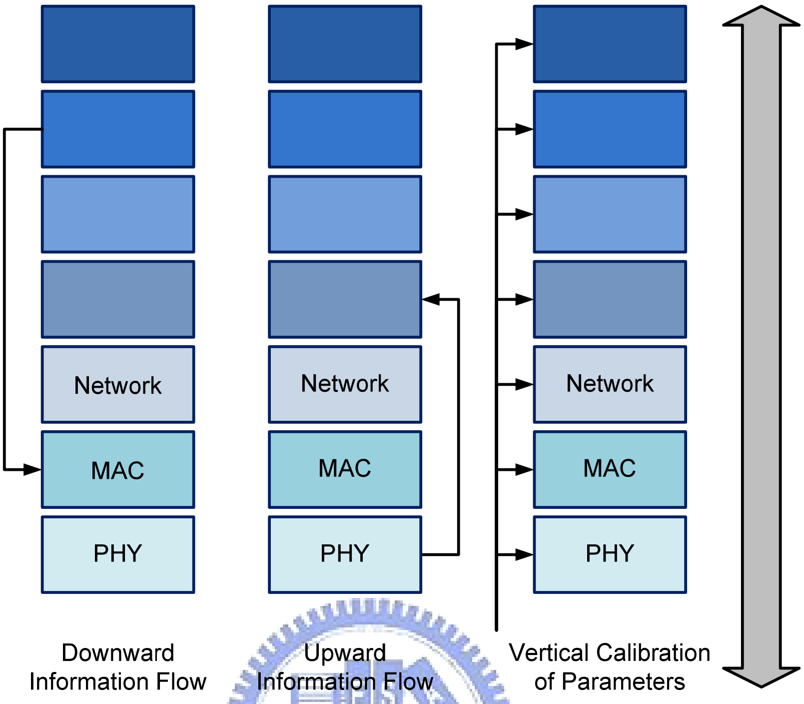


Figure 4-1: Cross-Layer Topologies [6]

In this article, the term “cross-layer topology” refers to the source-destination coupling of cross-layer signaling. The following discussion on topologies, illustrated in Figure 4-1, is a re-interpretation of the classification policy made in [6]:

1. Upward Information Flow:

An upward information flow occurs when a higher protocol layer requires information from a lower one. A well-known example is the TCP error resolution [7]. For wireless networks, TCP packet loss may be caused by either network congestion or bad channel condition. If the system considers only the congestion case, just as previous wired systems do, the TCP protocol would erroneously decreases its packet sending rate. This leads to throughput degradation. Therefore, an Explicit Congestion Notification (ECN) mechanism is proposed [12]. This mechanism is a cross-layer signaling method that distinguishes network congestion and channel error at the TCP sender, thus helps to avoid false decrease-rate decision.

Another example for the topology is Channel Quality Feedback (CQI) [10]. In a mobile WiMAX system, mobile stations report their measured CINR through CQI mechanism. When the BS PHY resolves CQI information from the Fast-Feedback Region located in the UL subframe, it reports the extracted average CINR value to the MAC layer. For OFDMA-based systems, the CQI information is essential for the MAC layer to make water-filling scheduling decisions. CQI information is also useful in BS-triggered handover, power management and other control mechanisms.

2. Downward Information Flow

A downward information flow occurs when a lower protocol layer requires information from a higher one. Downward information flow has been deployed throughout many wired or wireless system. A typical example of this is the packet delay requirement [6]. In a mobile WiMAX system, the MAC layer maps upper-layer packets to MAC layer SDUs. To support delay-sensitive services, the upper layers must inform the MAC layer of each packet's deadline. The upper layer may also include priority factors in the packets to provide reference to the MAC layer's scheduling mechanism.

Another example of downward information flow is the link adaptation scheme. In a mobile WiMAX system, the coding-modulation scheme of each PDU sent can be adjusted according to each user's condition. The MAC layer must include related indications with the PDU sent to the PHY layer.

3. Vertical Calibration of Parameters:

The vertical calibration topology refers to adjusting some parameters that are spanned across multiple layers in the protocol stack. For example, the bit error rate (BER) reflects the channel condition at each user’s end, and is measured at the PHY layer. This parameter is also referred by the MAC layer as reference to scheduling scheme, and layers above to make prompt decision on the packet sending rate.

In [6], this scheme is further classified into two: Static Vertical Calibration, and Dynamic Vertical Calibration. In the static scheme, parameters are set in a database at design stage, and left untouched throughout execution. In the dynamic scheme, parameters are retrieved and updated by multiple protocol layers in the stack, so designers have to plan very carefully to ensure the parameter is in a updated state, and causes less overhead for system execution.

Joint optimization across multiple protocol layers will be more and more important in next generation systems, and therefore the vertical calibration scheme will play an essential role in the future.

4.3.2. Cross-Layer Methodologies

In this article, the term “cross-layer methodology” refers to how the cross-layer signals are transferred, shared and updated, regardless of the source and destination. A number of prior proposals are surveyed, and re-classified according to their characteristics. Figure 4-2 is an illustrative graph of the three methodologies mentioned.

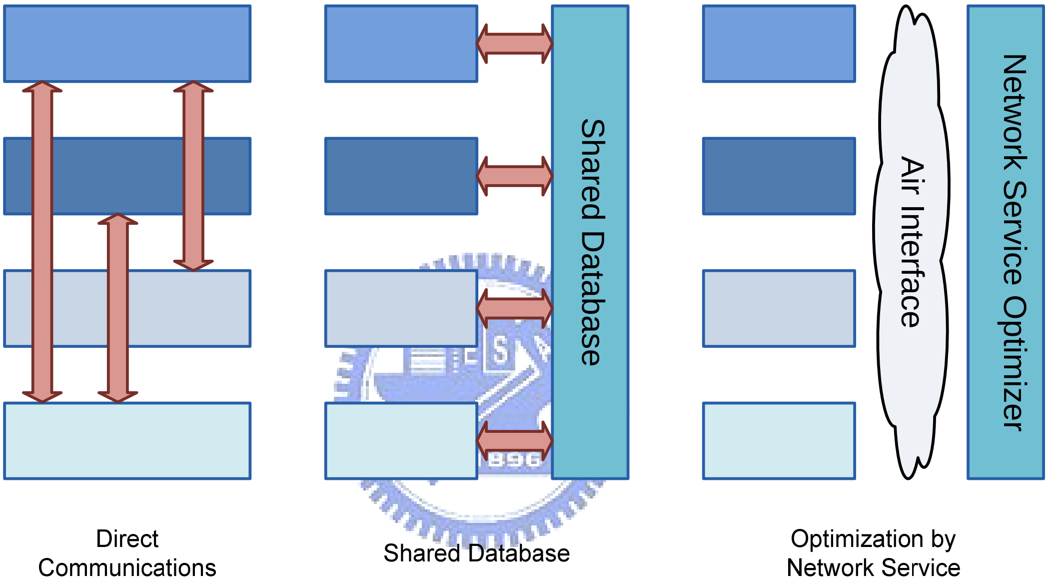


Figure 4-2: Cross-Layer Methodologies

4.3.2.1. Direct Communications between Layers

Direct communications requires well-defined protocols between two involved layers. Information may be embedded in piggybacked packet headers, or special message formats:

1. Signaling by Packet Headers

Limited amount of network layer information can be encoded in optional packet headers. In lower protocol layers, optional packet headers are decoded to derive network information inside. The packet headers play the role of “signaling pipe” spanned across multiple layers, and therefore the packet header scheme is also called “Interlayer Signaling Pipe”.

In [12], Explicit Congestion Notification (ECN) is transferred from the link/MAC layer to the network layer. The notification is made by setting a specific bit in the TCP packet header. In IPv6, an optional header format called Wireless Extension Header (WEH) is specified. The embedding of WEH in an IPv6 packet is shown in Figure 4-3.

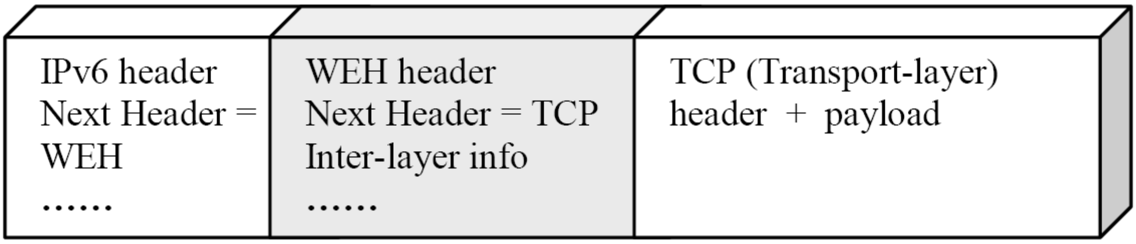


Figure 4-3: WEH Embedded in an IPv6 Packet Header [5]

WEH scheme has some disadvantages:

- Every protocol layer connected to the “signaling pipe” must be able to decode the extension header. Corresponding functionalities will cause significant changes to the protocol stack, and bigger code size.
- Decoding the packet at each layer will also cause considerable runtime processing overhead. This is especially undesirable for mobile device, which has limited amount of code memory and computing power.
- Information embedded in WEH is not accessible for layers higher than network layer,

since IPv6 packets cannot reach these layers.

- Finally, WEH contains only limited amount of parameters and indications. Packet header lacks flexibility; suits well only to Boolean indications, but not to complex parameters.

2. Signaling by Specific Messages

Internet Control Message Protocol (ICMP) is widely deployed in IP-based networks. Desired information is abstracted to parameters, and embedded in the message. Every layer can issue ICMP messages targeted to corresponding layers if convenient. The generation of ICMP messages is basically event-triggered.

This method transfer cross-layer signals by “punching holes in the protocol stack” [5]. Being a well-defined message structure, it seems to have better efficiency and flexibility than the WEH scheme. However, since ICMP messages are always embedded in an IP packet, every layer in the protocol stack must have the ability to process IP packets. This causes considerable code modification and execution overheads for the protocol stack.

Direct-communication signaling requires design coupling in every involved protocol layer. Involved layers must have the ability to process packets, and support a specific protocol. These schemes are suitable for external signaling, since the interoperability of any two devices is guaranteed through well-defined protocols. However, they are undesired in mobile host design; they cause bigger code sizes and excessive processing overheads.

4.3.2.2. Shared Database

In the shared database methodology, there is a common database accessible to every protocol layer. For mobile hosts, the common database is basically a part of system memory. In some researches such as [6], the common database is interpreted as a whole new protocol layer. In this article, it is viewed as an extra plane to the data/control plane system model.

This is an approach suitable for internal signaling within a network device. Also, it is particularly well suited to the vertical calibration topology shown in Figure 4-1. Also, this approach has better flexibility than the design coupling described in 4.3.2.1, since the database format can be defined and extended at will.

For direct cross-layer signaling, or so-called ad-hoc signaling scheme, additional cross-layer feedback caller and callee codes shall be added to any involved protocol layer pairings. The cross-layer feedback codes increases the difficulty of system implementation and maintenance, and may slow down the execution speed of the overall system due to complicated code references. It would also be harder to develop new optimizations, since rapid prototyping needs modularized code insertions. Share database approach in fact offers indirect cross-layer signaling that allows minimum intrusion to the protocol stack during implementation, and enables rapid prototyping and testing.

Design challenges that arise from shared database approach are: how to arbitrate database retrievals and updates by different layers. If the interface between the protocol stack and the common database is not well-standardized, the difficulty of system maintenance may increase awfully. Also, if the retrievals and updates are not well organized, the correctness of state variables and parameters is not guaranteed.

In [9], a cross-layer manager model, shown in Figure 4-4, is proposed. The cross-layer manager is a new plane that combines protocol states and managing algorithms. The protocol modules expose *events* and *state variables* to the manager. Events are notifications that trigger the algorithms inside the cross-layer manager. State variables allow the manager to query or modify the internal states of a protocol module.

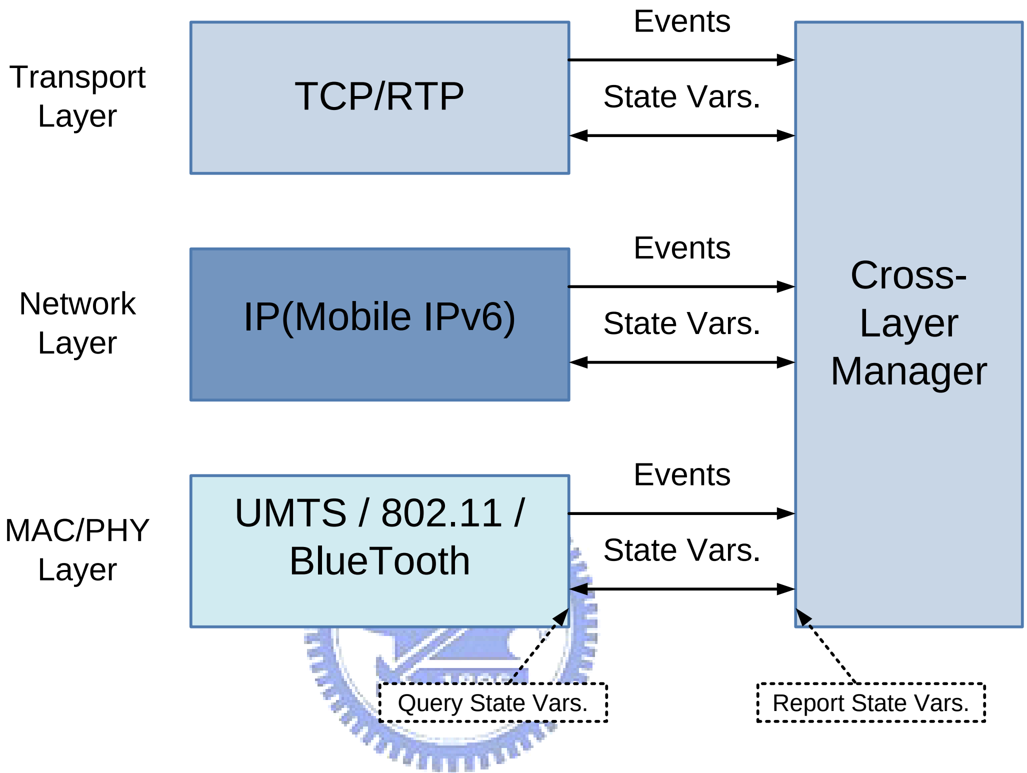


Figure 4-4: Interlayer Coordination Model Proposed in [9]

Discussions in [8] go deeper into software architecture issues. In [8], a cross-layer architecture called *ECLAIR* is proposed. This architecture divides the cross-layer system into two subsystems, shown in Figure 4-5:

– *Tuning Layer (TL)*:

The tuning layer provides an interface between the internals of a protocol and the optimizing algorithms. The tuning layer is basically a collection of API functions, including parameter retrievals, parameter updates and registrations. Each protocol layer is equipped with a corresponding tuning layer.

– *Optimizing Subsystem (OSS)*:

The optimizing subsystem contains the algorithms and data structures for cross-layer optimizations. The OSS plays the role of the "engine" of a cross-layer system, and executes concurrently with the protocol stack.

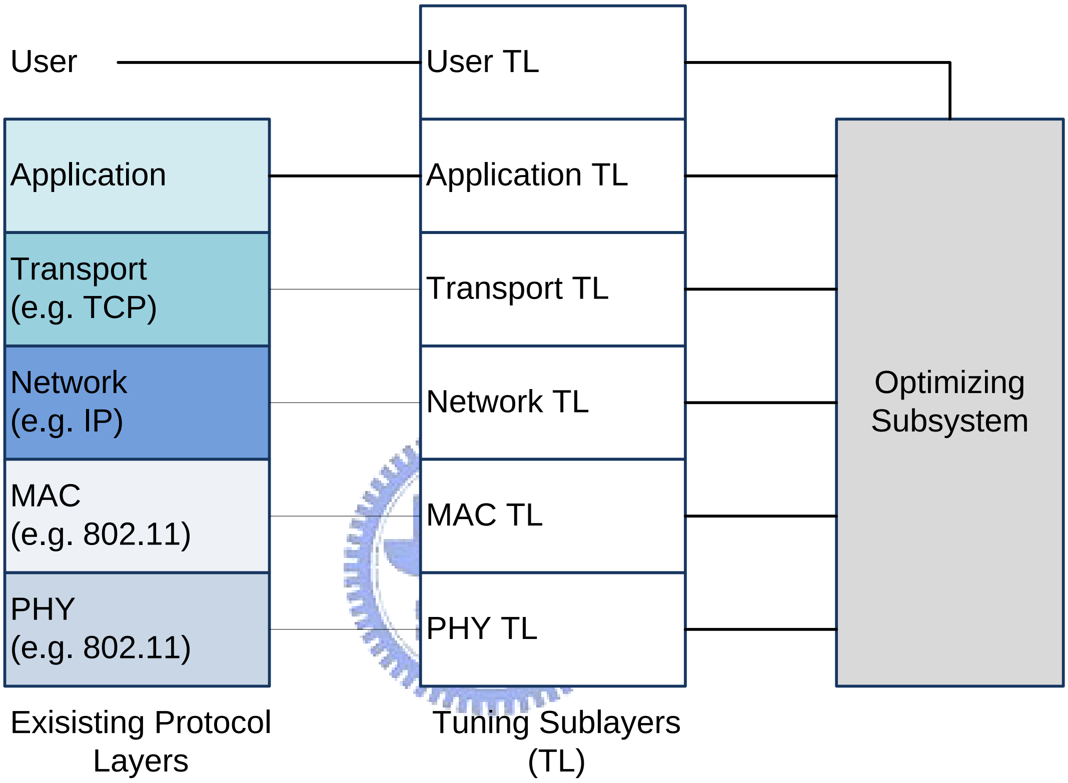


Figure 4-5: ECLAIR Architecture Proposed in [8]

The cross-layer architectures proposed in [9] and [8] are much similar: the architecture introduces least modification to existing protocol stack, a standardized interface based on parameter retrievals and updates, and a managing module that combines the data structure and algorithms. These architectures provide well-organized and standardized interface between optimization and the protocol stack, and the concept is adopted in this article.

4.3.2.3. Optimization by Network Service

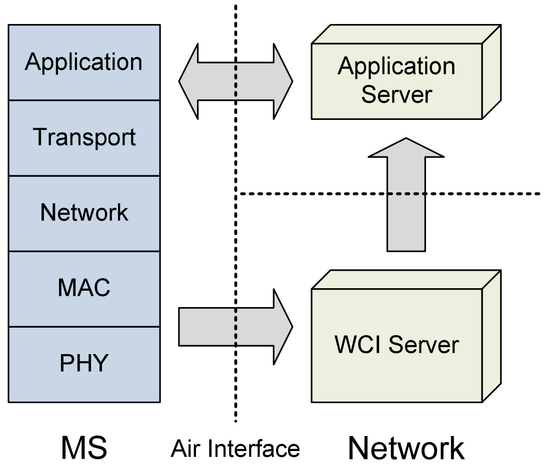


Figure 4-6: Concept Model of Network Assisted Optimization Service [5]

[13] proposed a specific network assisted optimization service called Wireless Channel Information (WCI). The concept model of WCI service is illustrated in Figure 4-6. The MS measures its local channel and link states and transmits the abstracted information to the remote WCI server, usually equipped at the BS. Interested upper-layer applications may access the WCI server for reports on the MS's conditions.

This optimization scheme is not an internal cross-layer signaling operated within a mobile host, but the concept is widely implemented in many existing wireless systems. Network assisted optimization has the following purposes: the condition report made by MS may help the BS in making optimizations. The upper layer applications may also access this information to make prompt adjustments. The MS may also make use of the BS's computing power instead of making computations on its own.

Network assisted optimization service is implicitly defined in mobile WiMAX system. There are some examples: CQICH, Fast Feedback Channels, Uplink HARQ ACK Channel and Uplink MIMO Sounding. However this scheme is carried out only at a periodical basis since air-interface signaling would introduce considerable signaling overhead and processing delay.

4.4. Proposed Cross-Layer Design Architecture

The proposed overall cross-layer design architecture includes three methodologies:

1. System database shared by the entire protocol stack.
2. Direct communication between two adjacent layers. For example, predefined cross-layer information package between two adjacent layers (MAC and PHY).
3. Conventional direct signaling approaches such as WEH and ICMP.

Each methodology has some design guidelines. For the system database methodology, the most important thing is to provide prompt and well-organized cross-layer information, with guaranteed correctness. For the predefined information package method, the cross-layer package should facilitate batch processing between layers. For the conventional protocols, conformance and convergence to existing protocols is required.

Also, the requirements of cross-layer design are different at the BS and the MS. For the MS side, all protocol layers are included in a machine. Therefore designers have more freedom to choose cross-layer methodologies when implementing an MS. Designers may prefer shared database approach more than WEH and ICMP. For the BS side, a BS may support only up to MAC or network layer. Layers higher than network layer may be implemented on remote servers. Therefore, when designing cross-layer interfaces higher than

MAC layer at the BS side, designers may have no choice but implement conventional upper-layer protocols such as ICMP or WEH.

In this article, we shall focus on the system database and the cross-layer information package methodologies. We shall discuss the standard interface of system database access, and an example of cross-layer information sharing. We shall also take a look at the data structure passed between two adjacent layers, and discuss the design considerations behind it.

4.4.1. Cross-Layer Signaling by System Database

4.4.1.1. System Database Standard Interfaces

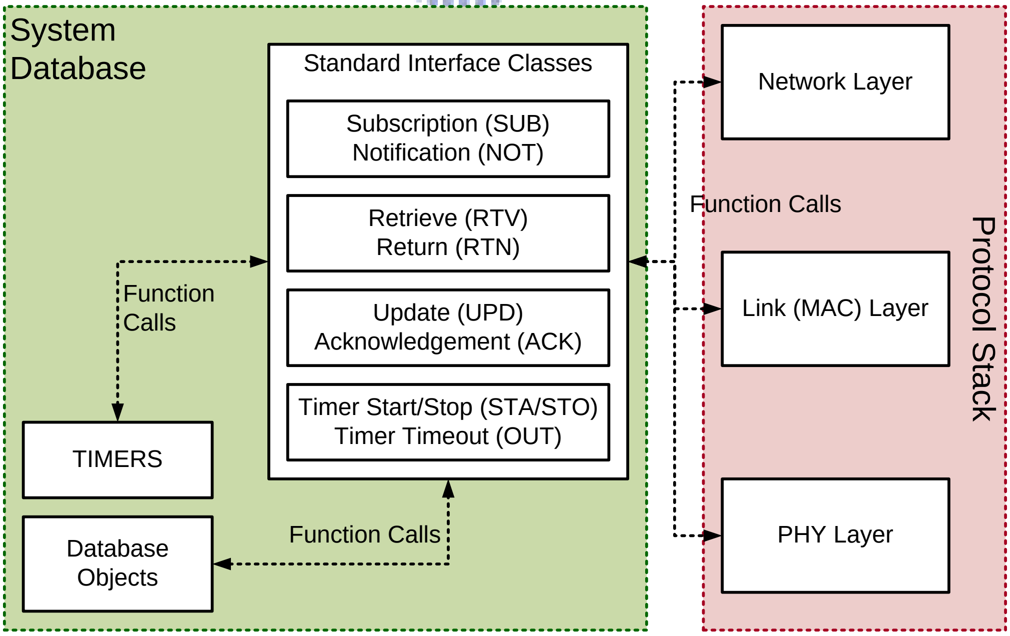


Figure 4-7: System Database, Protocol Stack and Standard Interfaces.

The system database is shared by all protocol layers within a machine. The major guideline to system database signaling is: ensure proper operation of the system database, and enable prompt information passing between layers. A series of standard interfaces should be defined. In the proposed cross-layer architecture, four types of system database standard interfaces are specified and incurred at every protocol layers. Figure 4-7 is an illustrative graph of the standard interfaces.

1. Retrieve/Return (RTV/RTN):

When a management entity wishes to get some information from the system database, it shall send a retrieve task to the system database. The system database then sends back a return. The retrieve task shall include the parameters to be retrieved. The return task should include signals that indicate the correctness of the parameter, and the content inside. To ensure correctness of parameters, the system database may return a "failed" indication if it considers the target parameter is out-of-date.

2. Update/Acknowledgement (UPD/ACK):

When a protocol layer entity wishes to write some information to the system database, it shall send an update task to the system database, indicating the parameters to be overwritten, and the updating value. The system database then sends back an acknowledgement. The system database may refuse updating the parameters in the acknowledgement if correctness is not guaranteed.

3. Subscribe/Notification (SUB/NOT):

This interface type is especially important for cross-layer signaling. The SUB/NOT interface is used if the protocol entity requires the system database to automatically report parameters on some conditions. In the subscribe task, a protocol entity shall indicate the parameter under subscription, and the subscription type, i.e. condition for the system database to send a notification. There are four notification types: notify on any change of the designated parameter; notify on change over some threshold; notify periodically and one-time notify on a designated time spot.

4. Timer Start/Stop/Timeout (STA/STO/OUT):

This interface type is similar to the SUB/NOT interface but is identified as an independent interface type. This interface type copes with timers only. In the timer stop task, the protocol entity shall specify if the timer shall reset or keep its value on timer stop. When a timer expires, the system database shall automatically send a timeout notification to the involved protocol entities.

4.4.1.2. Example for Cross-Layer Protocol by System Database

Here’s an operation example of how system database helps cross-layer signaling. Take cell reselection as an example. When an MS enters normal operation, its handover management entity (ME) would subscribe to the neighbor cell profiles in the system database. The PHY measurement section of MS would also periodically update the neighbor cell profiles.

The MS handover ME sends two types of subscription tasks to the neighbor cell profiles. One is periodical report of neighbor cell information; the other is event report when the neighbor cell signal quality reaches some threshold. The PHY measurement section may send an update task to modify multiple parameters in the neighbor cell profile.

When the system database detects that the signal quality of a certain neighbor BS falls below some threshold, it sends an automatic event notification to the handover ME, so that the handover ME may choose to drop the neighbor cell. When the MS decides to perform handover with an HO-IND message, it simultaneously starts the resource retain timer using a timer start task. When the resource retain timer runs out, the system database would send an automatic timer timeout notification to the handover ME, so that the ME may erase the prior serving BS’s information. The internal protocol between system database, PHY measurement section and handover ME is illustrated in Figure 4-8.

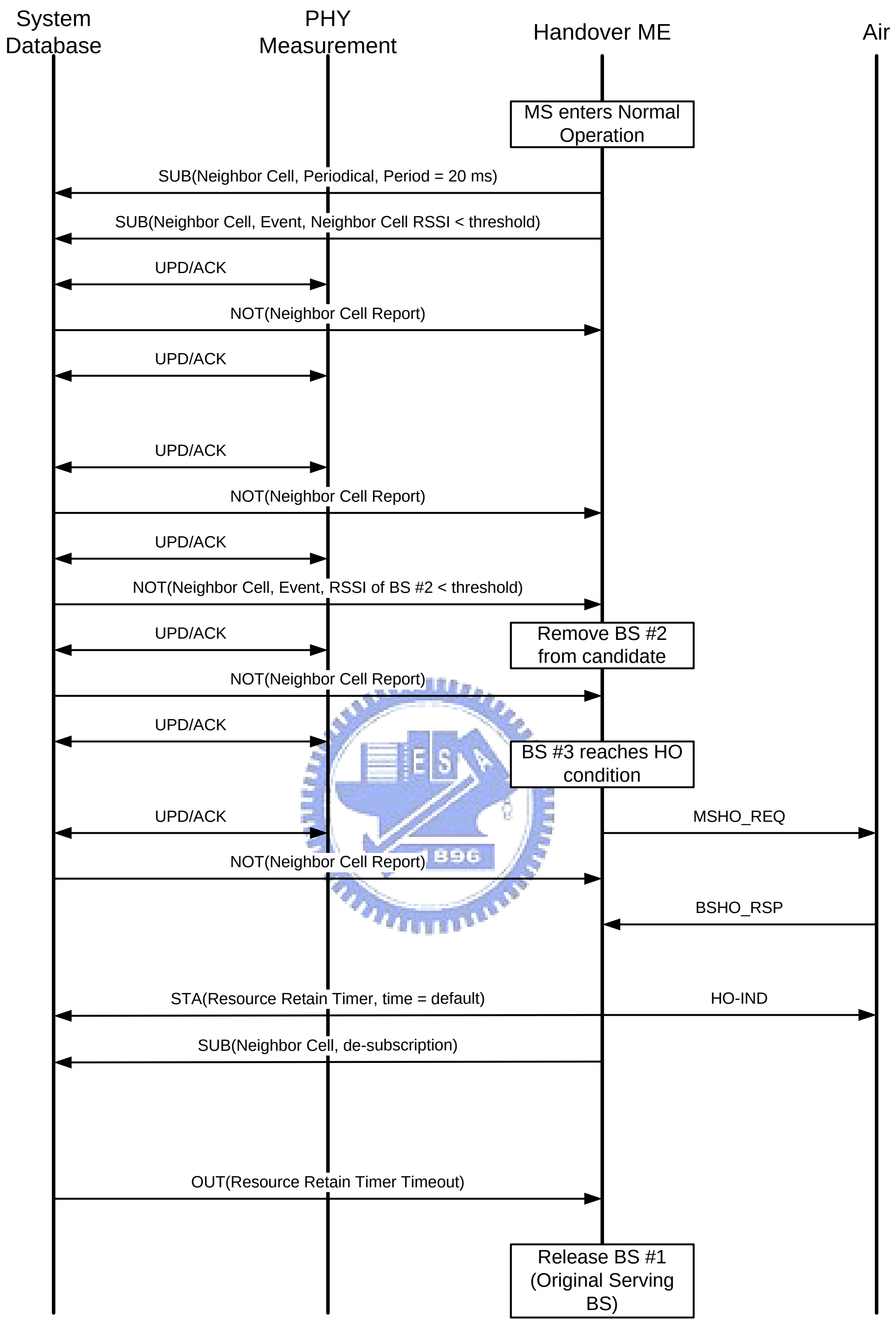


Figure 4-8: Internal Cross-Layer Protocol, with Handover Process as an Example

4.4.1.3. Design Considerations of System Database

For system database design at either BS or MS side, there are a few suggested design methodologies that facilitate rapid prototyping and overall system implementation:

1. Object Oriented Database Design:

In designing system database, it is suggested to define database objects that are mapped to real-world devices or entities. This facilitates system implementation and enables software to be executed more efficiently. A conceptual model of object oriented database design is shown in Figure 4-9.

Take the BS objects at the MS side for example. In order to perform cell reselection, fast BS switching (FBSS) or macro-diversity handover (MDHO), the MS shall continuously monitor the surrounding BSs and keep a number of BS sets, such as serving BS, handover candidate set, active set, and associated set.

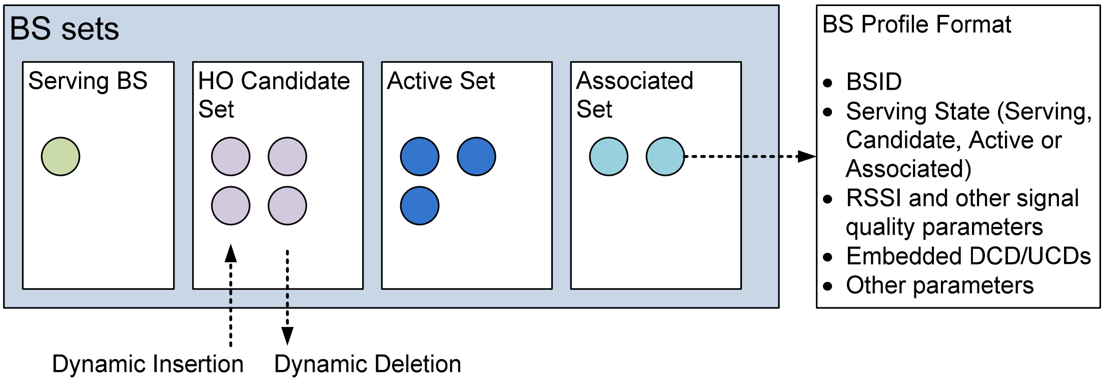


Figure 4-9: BS Sets as an Example of Object Oriented Database Design

It is suggested that the information of each BS is defined in a uniform data structure. When the MS detects the existence of a new BS, it may add a BS profile object at the handover candidate set. This is done in a dynamic manner using object-oriented programming languages such as C++. The MS may also dynamically drop a BS profile object at its will. When the MS wishes to upgrade or degrade a BS from the current set to another, since every BS is defined in the same format, the MS could simply move the object pointer to the target set instead of moving the whole data structure. This improves execution efficiency of the software. Some typical object formats are:

- BS Profiles.
- MS Profiles.
- Service Flow/Connection Profiles.
- Channel Profiles.
- Channel Descriptors (DCD/UCD), including a number of burst profiles.
- DL-MAP and UL-MAP, including a number of MAP IEs.
- Timers.

2. Standard Interface Task Stack:

Every standard interface indications, e.g. SUB/NOT or RTV/RTN, can be viewed as a task in the system software. It is suggested that these tasks are scheduled into a task stack. The main benefit is: through task stack implementation, the standard interface tasks are well scheduled and arbitrated. For example, when the system is scheduled to update and retrieve the same object in the system object simultaneously, it may perform update prior to retrieval, thus collision is avoided and correctness of parameters is guaranteed.

4.4.1.4. Benefits of Cross-Layer Signaling by System Database

The proposed system database design has a few benefits:

1. Adaptation to Real-Timer or Timedrop Based Environments:

For simulator design, in which software performance is usually not an issue, designers usually implement the simulator based on a global timedrop variable. That is, the simulator does not care for the real timing; it simply increments the global timedrop variable by a frame duration after it finishes a frame's workload. However, in practical system design, designers must refer to a real timer, usually embedded in the processor core.

The task stack implementation provides good adaptation both to real-timer or time-drop based environments. The standard interface tasks are scheduled into the task stack, thus updates and retrievals targeted at the same database objects are well arbitrated, and collisions are avoided. This is especially important in real-timer environments since the arrival time of two simultaneous tasks cannot be precisely predicted.

2. Minimum Intrusion to System Code:

The access interface is well defined in the system database. When a management entity wishes to access the system database, it needs only reuse the predefined interface functions, instead of adding a set of extra codes to the protocol stack. This is desirable since extra codes may make system maintenance more difficult, and large code size may cause problems in embedded systems.

3. Better Efficiency for Execution:

Object oriented database design enables dynamic add and delete of system database objects. When the MS wishes to move an object from a set to another, it can simply change each set’s lookup table for object pointers. This avoids a large number of memory accesses, and thus improves software performance.

4.4.2. Cross-Layer Signaling by MAC/PHY Information Package

4.4.2.1. The MAC/PHY Information Package

Mobile WiMAX is a frame-based system. Therefore, the major guideline for data passing between MAC and PHY layer is to facilitate batch processing within single frame duration. The information package basically includes two parts: binary raw data transmitted in the frame, and abstractions extracted from DCD, UCD, DL-MAP and UL-MAP.

- Binary Data:

This part includes all raw binary data transmitted in a frame, neither encoded by FEC nor modulated, including that of data bursts, DL-MAP, UL-MAP and FCH.

- Ready-to-Serve burst IEs:

This part is a series of burst IEs that indicates each data burst’s PHY parameters. Every burst IE is an aggregation of the burst’s burst profile (for downlink, in the DCD message) and MAP IE (for downlink, in the DL-MAP message). Therefore, for OFDMA PHY the burst IE shall include the following information related to a burst:

- Corresponding DIUC

- Coding-modulation scheme
- Repetition Coding Indication
- Involved CIDs
- Relative location in the frame (denoted by OFDMA symbol offset and subchannel offset)
- Two-dimensional size (denoted by number of OFDMA symbols and number of subchannels)
- Power boosting

- Ready-to-Serve Overall Channel Information

This part is a data structure containing a re-arrangement of overall channel encodings in DCD, UCD, DL-MAP and UL-MAP. This part includes:

- Parameters related to frame timing: frame duration, TTG, RTG
- Parameters related to frame frequency: center frequency, UL-allocated subchannels bitmap
- Parameters related to frame power: BS EIRP, power adjustment rule, $RSS_{IR,\ max}$
- Permutation base for BS preamble

- Ready-to-Serve Ranging Indications:

Ranging indications are included in the UCD message, but this part is extracted from the message as a data structure in order to facilitate PHY ranging operations. This part includes all necessary information for the ranging subchannel located in the uplink subframe, including:

- Initial ranging backoff start/end
- Periodical ranging backoff start/end
- Bandwidth request backoff start/end

- Initial ranging codes
- Periodic ranging codes
- Bandwidth request codes
- Location/Size of ranging subchannels

4.4.2.2. Design Considerations and Benefits

The guideline for MAC/PHY information package is to facilitate batch processing at the PHY layer. The PHY layer is usually realized with specific-application devices such as ASICs and DSPs, so the designer should take the hardware realization into account.

For the MAC/PHY information package, it is suggested to align the binary raw data of each burst in a continuous manner in the system memory, and indicate the memory allocation rule in the ready-to-serve burst IEs. This is because batch access to a whole section of system memory can decrease the number of memory access instructions, reduce the probability of cache miss and reload, and thus improve hardware performance significantly.

Also, the abstractions extracted from channel descriptors (DCD and UCD) and maps (DL-MAP and UL-MAP) provide good adaptations and re-configurability to hardware realization of PHY layer.

4.4.3. MAC/PHY Cross-Layer Architecture

Figure 4-10 is an illustrative diagram of the proposed downlink MAC/PHY cross-layer architecture. The PHY layer is equipped with an upper PHY module that receives binary raw data from MAC data plane, and channel/burst abstractions from MAC control plane. The cross-layer package is then distributed by the upper PHY module to requesting PHY modules, such as preamble, ranging unit and the main part of PHY layer – the data processing unit.

On the other side, measurements made by PHY measurement unit is updated to the system database using the standard interfaces define therein. The MAC layer also accesses the system database using the same set of standard interfaces, in order to preserve the modularity and well-definedness of the system architecture.

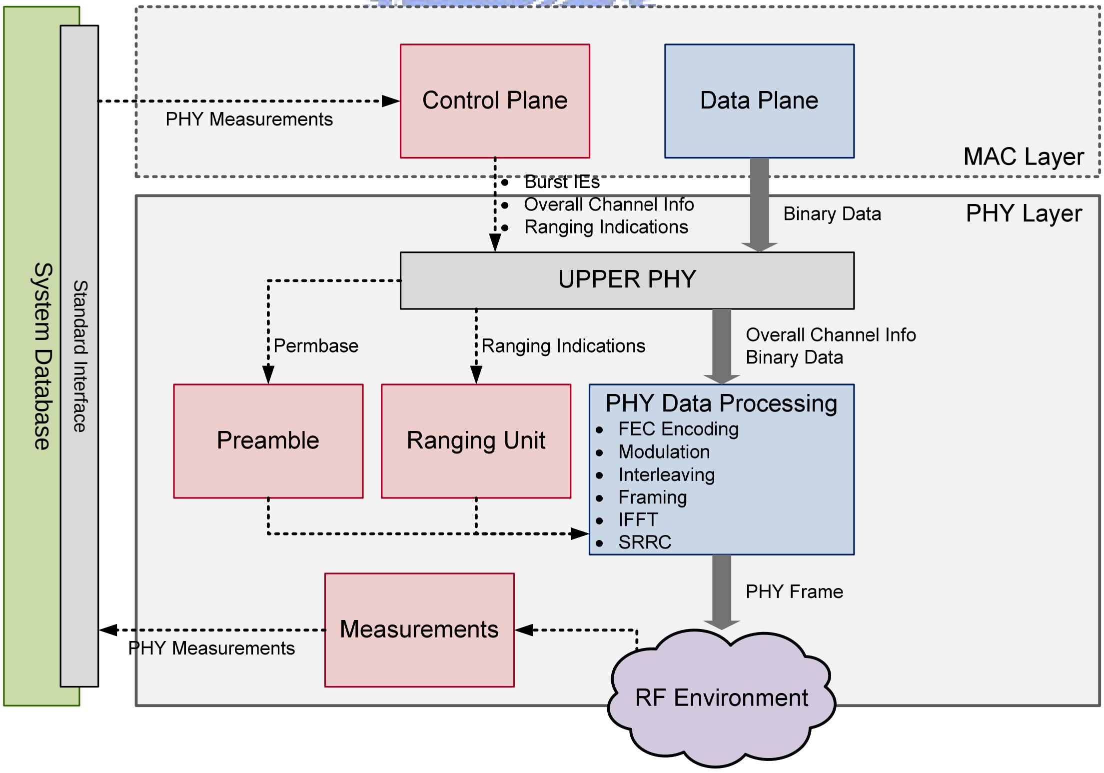


Figure 4-10: Proposed MAC/PHY Cross-Layer Architecture

Chapter 5. Case Study: Network Entry

5.1. Design Objectives

In this article, we present a mobile WiMAX design primitive that focuses on MAC layer implementation. The design objective is to establish a framework for:

– MAC 3-plane architecture and co-operation between planes.

– Design methodology for management entities and messaging unit.

– MAC/PHY cross-layer integration.

Practically we did not full-implement the mobile WiMAX system. Therefore, we use network entry process as an illustrative case study to the design details. This case is a good example for overall WiMAX architecture because it contains control-data plane cooperation, MAC management messages and PHY measurement feedback.

The following part of this section is organized like this: first, we come up with a thorough dissection of mobile WiMAX network entry procedures. Second, the design flow used in our system design is introduced. After that, there is a more detailed discussion on how the system is implemented.

5.2. Network Entry Procedures for Mobile WiMAX

Network entry is the applicable procedures for a new MS node, starting from power-on, to enter and register to the network [1]. The procedure can be roughly divided into the following phases:

1. Establish PHY/MAC Synchronization:

After power-on, the MS shall scan for suitable BS and downlink channel, and establish PHY synchronization with the selected BS. After PHY synchronization is established, the MS seeks for the DL-MAP message broadcasted in every frame. As long as the DL-MAP message is successfully received and decoded, the MAC synchronization is established.

2. Obtain Transmit Parameters:

After PHY/MAC synchronization is established, the MS shall search for the DCD to obtain downlink parameters. Then, the MS waits for a UCD message from the BS to retrieve a set of transmission parameters for a possible uplink channel. The MS is considered to have valid downlink/uplink parameters as long as it continuously receives the DL-MAP, DCD, UL-MAP and UCD messages. The MS has to return to synchronization phase if it fails to find these messages receivable within corresponding specified timeout periods.

3. Perform Ranging:

Ranging is the process of acquiring the correct timing offset and power adjustments such that the MS's transmissions can be correctly aligned to the PHY frame and received. The first ranging process performed after MS power-on is called initial ranging. Different from WirelessMAN-SC, SCa and OFDM PHY modes, in OFDMA mode, the MS performs contention-based initial ranging.

The procedure of contention-based initial ranging is described as follows: After PHY/MAC synchronization is established and the transmission parameters are received, the MS shall scan the UL-MAP message in every frame for an Initial Ranging IE. The MS shall than transmit a randomly selected CDMA code at the opportunity described in the IE. The BS then sends a RNG-RSP message with received CDMA ranging code. If the ranging status is "continue", the designated MS that shall make necessary correction and send randomly selected CDMA code in another opportunity. If the ranging status is "success", the BS shall allocate an uplink opportunity for the MS to send RNG-REQ message. The MS then continuously adjusts its transmission parameters using periodical ranging mechanism. The message flow diagram of initial ranging is shown in Figure 5-1.

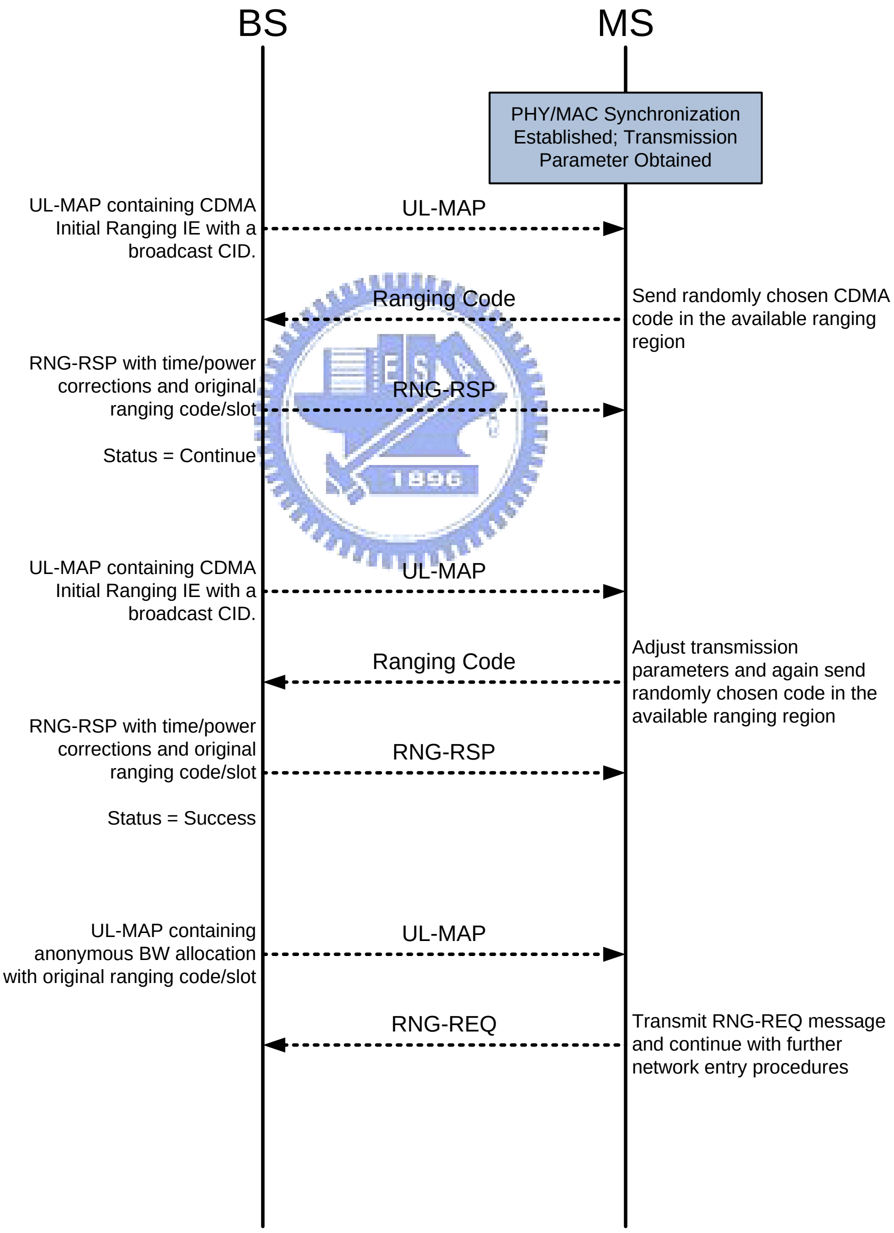


Figure 5-1: Message Flow for Initial Ranging/Automatic Adjustment in OFDMA Mode

4. Negotiate Basic Capabilities:

After ranging process, the MS then commences capability negotiation with the BS, in order to know the common capability parameters between the MS and the BS. The SS shall send a SBC-REQ message to the BS. The BS replies with a SBC-RSP message, indicating the intersection set of the MS's and the MS's capabilities. This is a relatively simple message exchange.

5. Authorization and Key Exchange:

The MS and BS exchanges authorization and key exchange information using PKM protocol. Due to practical problems, we didn't implement this part and consider this part out of research scope.

6. Registration:

Registration is the process by which the MS is allowed to enter the network and be managed by the network. In our research scope, this is the last step of network entry procedures. The registration process is a simple message exchange. The MS shall send a REG-REQ message. The BS then replies with a REG-RSP message, signaling the completeness of network entry procedures.

5.3. Software Development Flow

In Figure 5-2 an overall software development flow is illustrated. The overall system design starts with comprehensive study of system spec. After spec study, the designers build a rough basic 3-plane model that is applicable to cross-layer wireless system design.

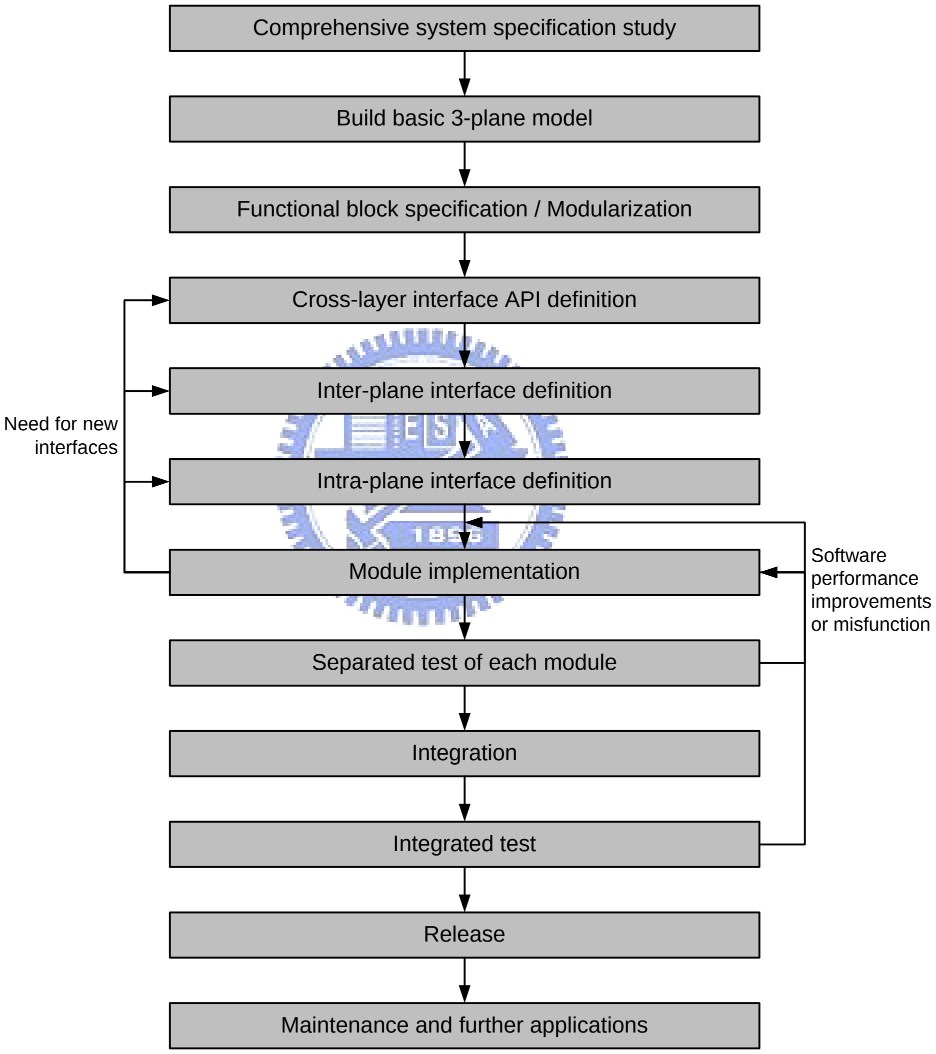


Figure 5-2: Overall Software Development Flow

Within each plane defined, the system is further analyzed and modularized according to functionality. Then the designers define interfaces between blocks and blocks. The interface definition includes three stages:

1. Cross-layer interface definition: specify cross-layer access to the common system database, and frame-based data structure between MAC and PHY.
2. Inter-plane interface definition: specify how the control plane MEs access the system database, how the MEs control the operation of the data plane, and how the message SDUs are inserted into the connection-based buffer.
3. Intra-plane interface definition: specify the function call relationship between MEs, and the trigger request between MEs and the messaging unit. For the data plane, the data structure of SDU/PDU/frame shall be well defined.

Designers then start module implementation, i.e. control mechanism design and coding. Each module shall be designed in a manner such that separated module test is allowed. There is a feedback mechanism among interface definition stage, implementation stage and separated test stage, to guarantee that the system functions properly and meets performance goal. Finally there’s an integrated test of the overall system. If there’s any need for performance improvement, the designers go back to the implementation stage and make necessary modifications within the module. A more detailed internal design flow of each plane is depicted in Figure 5-3.

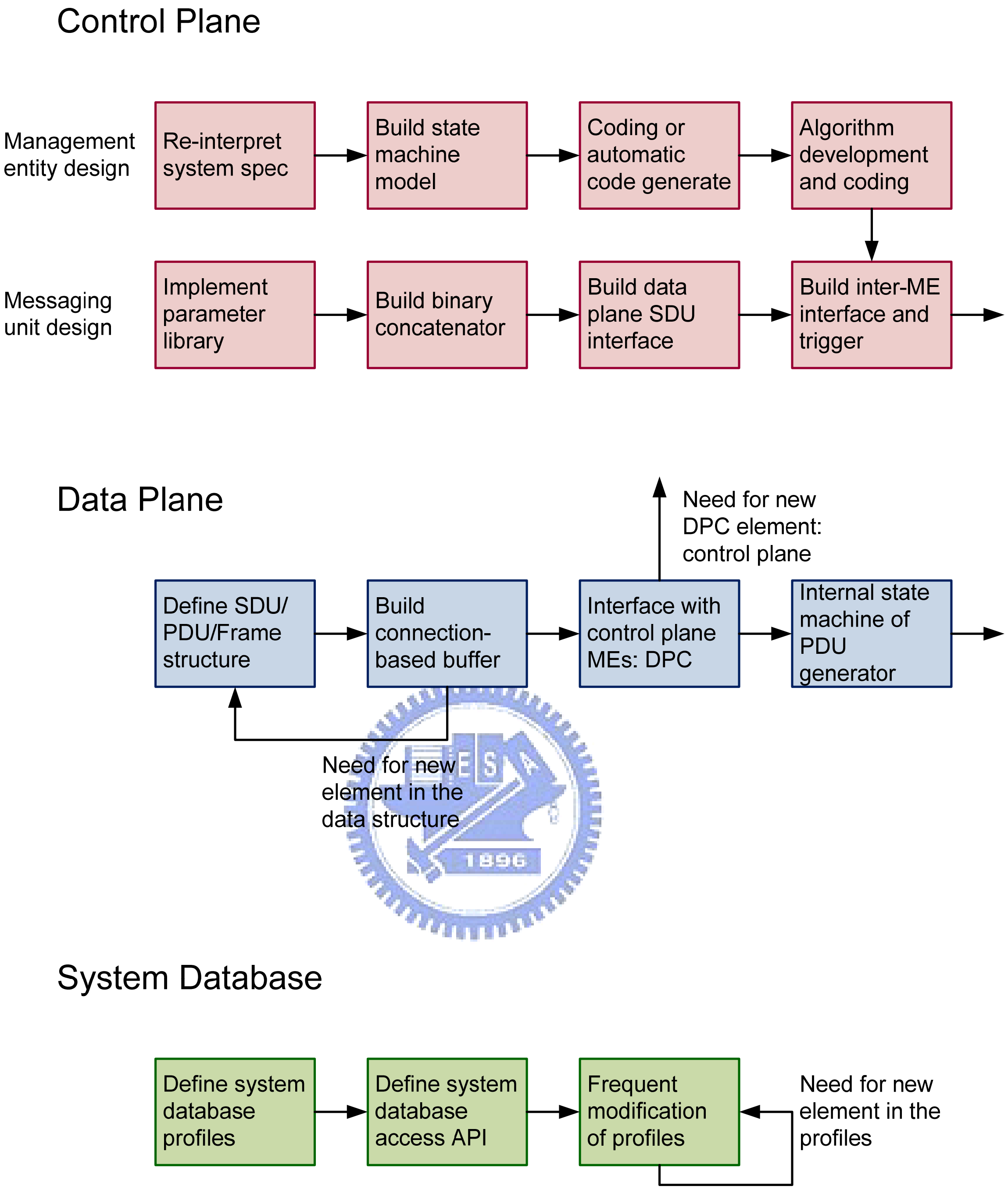


Figure 5-3: Detailed Design Flow of Each Plane

5.4. Design Details

5.4.1. Overview of the Design Primitives

The design primitive is an executable project written in C language. The source file structure contains a top module for overall execution. The execution includes both the BS side and the MS side, each having a top module and a list of supporting C source files. The execution is time-drop based, i.e. it doesn't involve a real timer, but increments the global system time-drop by constant frame duration after each frame's execution. The execution flow of the design primitive is illustrated in Figure 5-4, and the file structure is listed in Figure 5-5.

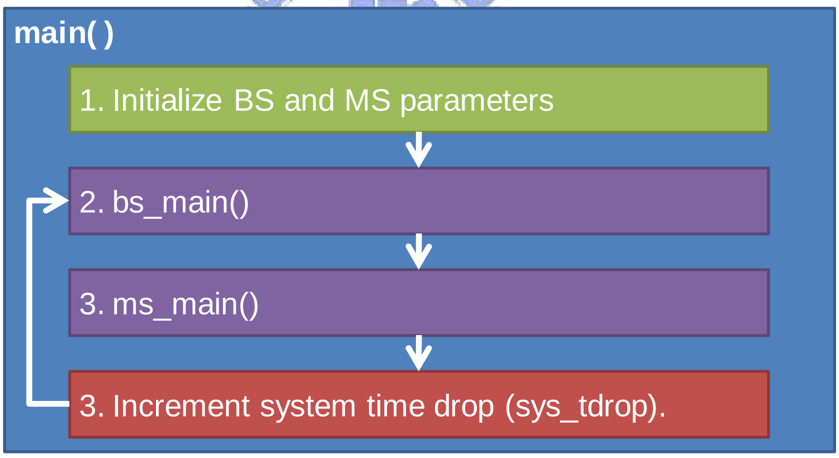


Figure 5-4: Execution Flow of the Design Primitive

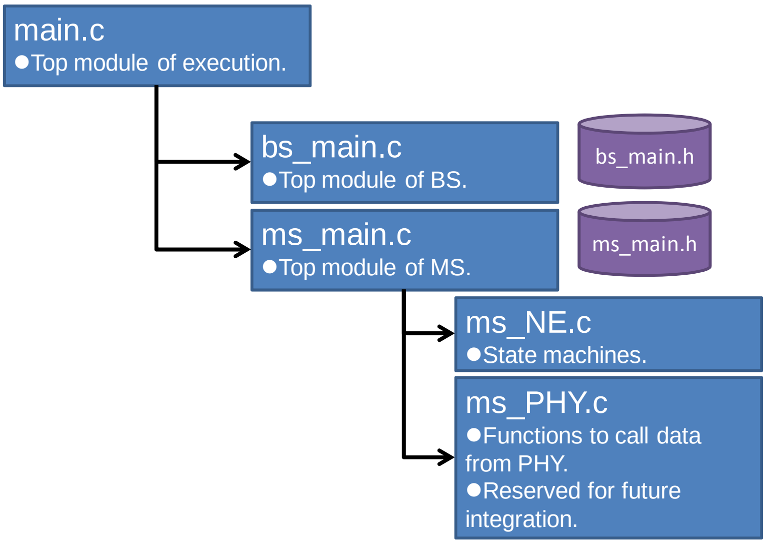


Figure 5-5: Source File Structure

5.4.2. Data Plane Design Example: PDU Generator

The PDU generator retrieves SDUs from the connection-based buffer, makes packing/fragmentation if necessary, and appends generic MAC header and subheaders to the produced PDU. The essential part of the PDU generator design is packing/fragmentation logic and the data plane command (DPC) interface.

The operation flow of the PDU generator is shown in Figure 5-6. The PDU generator receives DPC from the control plane, which includes a PDU’s type (generic PDU or BW request), source CID, allowed payload size and coding-modulation scheme. The compound DPC is prepared by the control plane MEs. Therefore, the data plane simply operates following the DPC indications, and doesn’t bother making any further decision from the DPC.

If the PDU type indicated by DPC is generic, then the PDU generator has to process PDU payload. After retrieving SDU from indicated CID, it shall check if the SDU is packing/fragmentation allowed. Note that message SDUs from broadcast and primary management CIDs are not allowed to be packed or fragmented.

The SDU element structure in the connection-based buffer shall also include a flag bit indicating the SDU fragmented or un-fragmented. The PDU generator then compares the residual size of the SDU with the payload size, along with the fragmented/un-fragmented flag, to decide taking packing, fragmentation or just-fit logic. The logic flow of packing/fragmentation is shown in Figure 5-7.

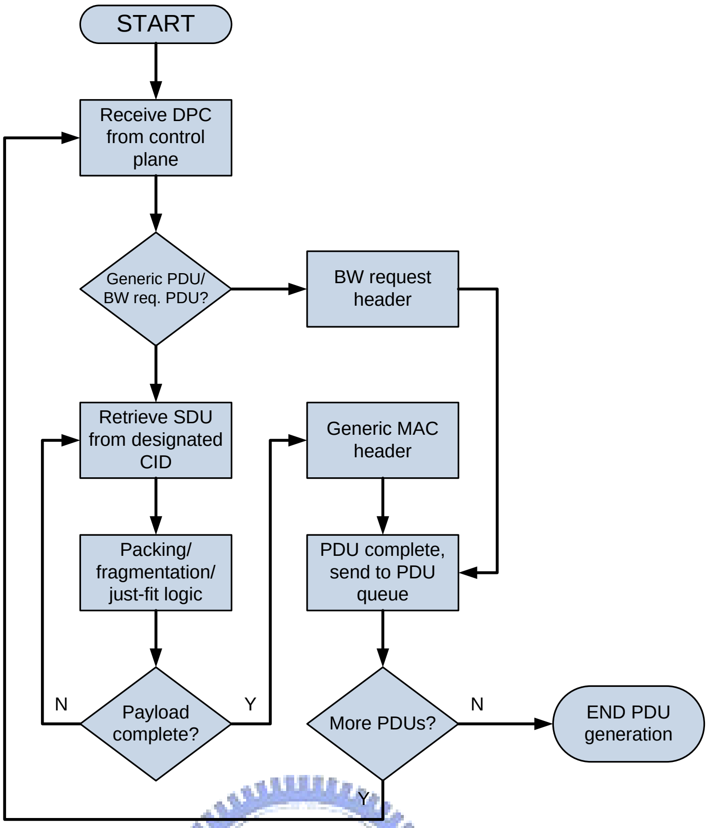


Figure 5-6: Operation Flow of the PDU Generator

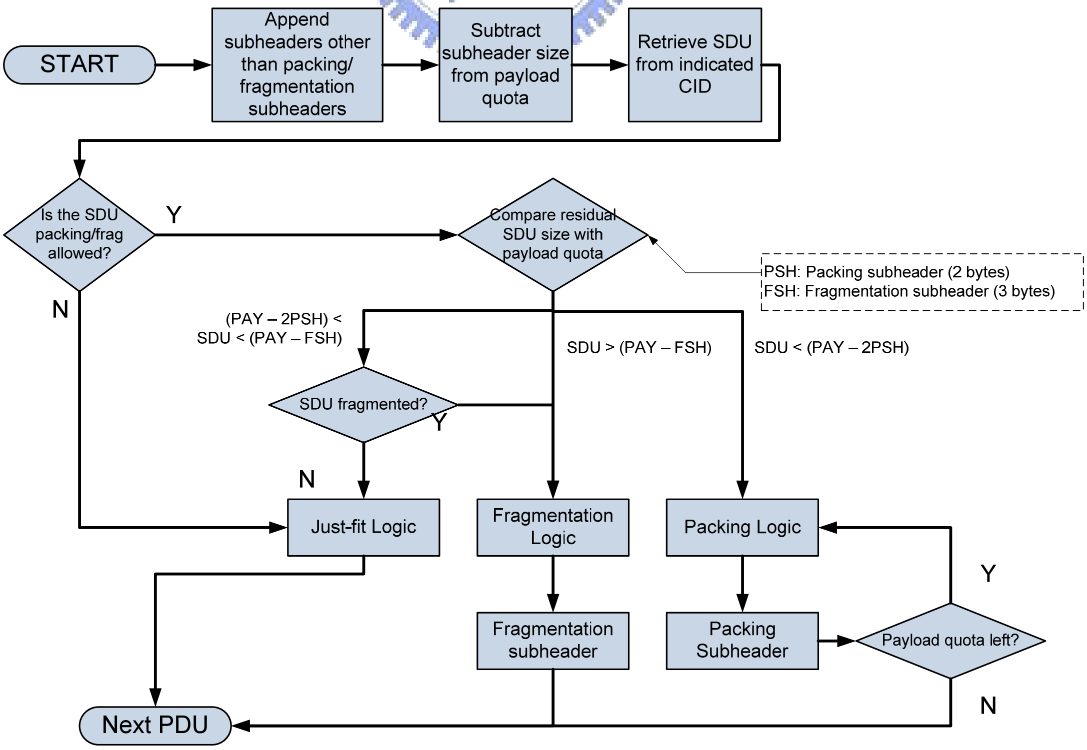


Figure 5-7: Packing/Fragmentation Logic Flow

5.4.3. Control Plane Design Example: MEs and State Machines

The management entities are the core components of the control plane. A management entity is basically a combination of three elements: state machine, the background algorithms and a set of standard interfaces with other components. In this design primitive, a management entity is wrapped up as a subroutine function. It can either be called or call other subroutines.

Take the MS side design of network entry as an example. The design flow starts from comprehensive study of the system spec. In standard [1] [2], a series of state diagram is specified. Although the transition logic is well-defined, the state variables and tasks on state transitions are not explicitly specified. The implementation details are not in the scope of the standards. Therefore, in practice, these state diagrams should be re-interpreted for further implementation. In our design primitive, the MS side keeps a three-level root state profile in the system database's local profile. Also each independent ME should keep a state profile in the MS local profile. The root state is illustrated in Figure 5-8 and Figure 5-9.

A re-interpreted state diagram example is shown in Figure 5-10. During system implementation, it is important to include the following annotations in the re-interpreted state diagrams:

- State Encodings: The root state machine encodings and state machine encodings of each ME involved.
- Flows: Entry and exit of each state.
- Entry/Exit Conditions: The condition description should be one of the standard interfaces described in 3.4.1.
- State Tasks: things to be done during this state, on entry to the state, or on exit from the state. The tasks should be one of the standard interfaces described in section 3.4.1.

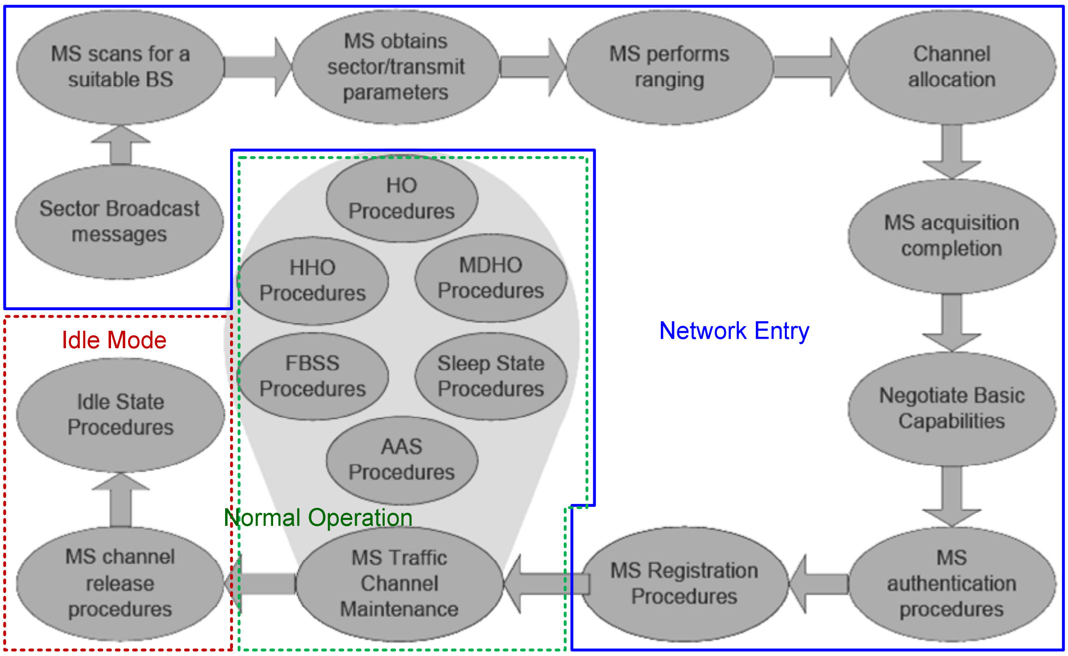


Figure 5-8: WiMAX Operation Procedures

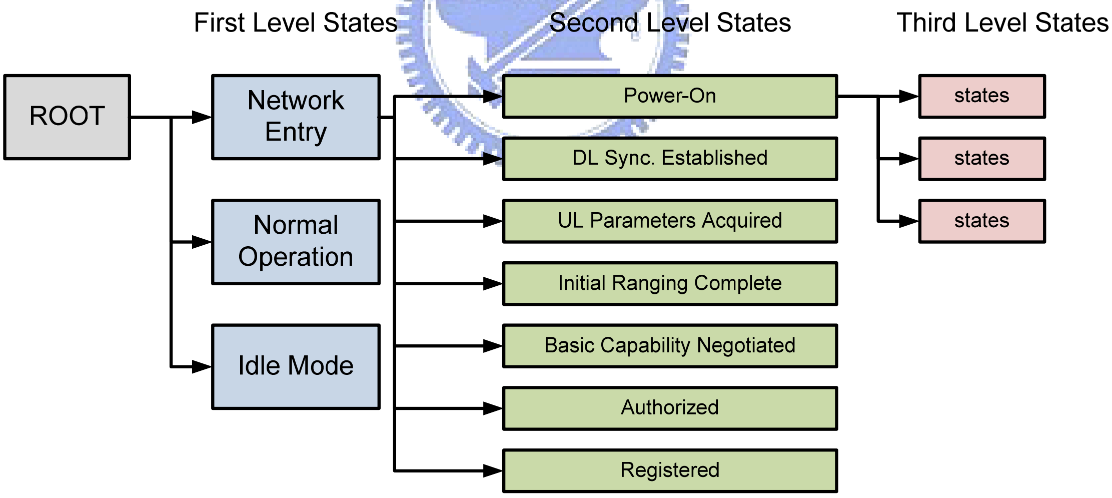


Figure 5-9: Root State Machine

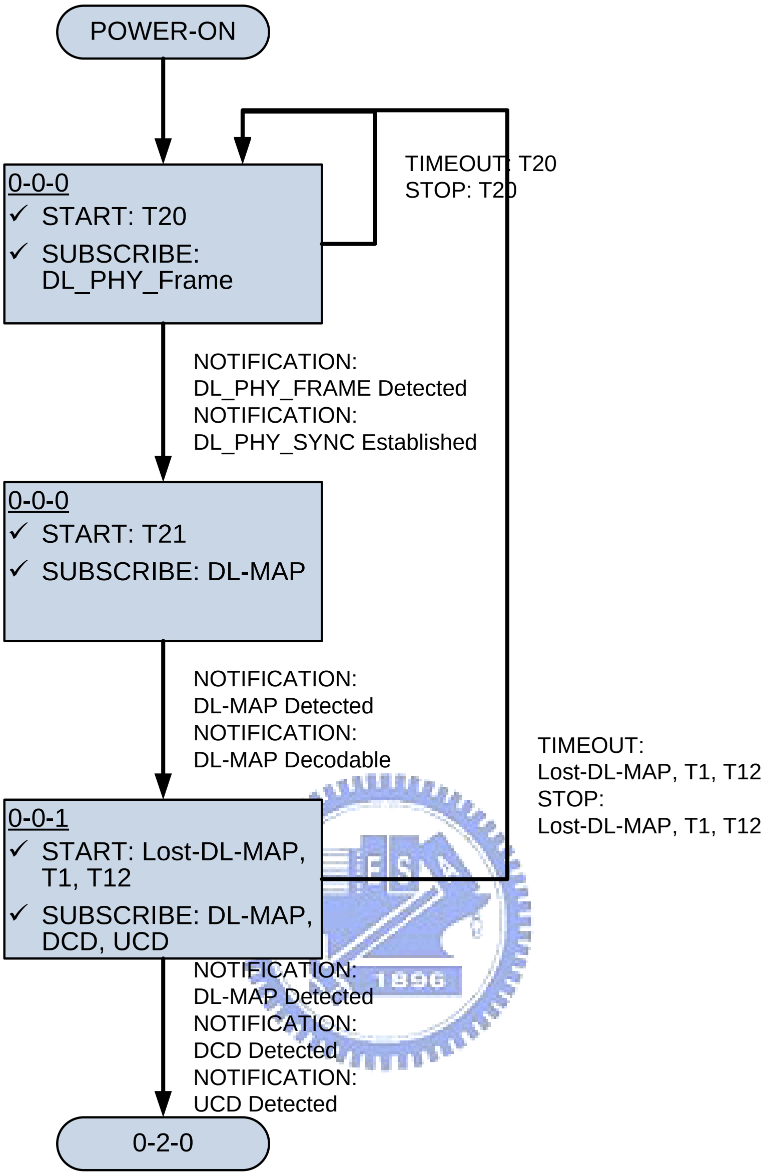


Figure 5-10: Example for Re-interpreted State Diagram

5.4.4. Cross-Plane Design Example: Messaging Unit

When a control plane ME requires sending a MAC management message, it triggers the messaging unit, and the process of making and sending a message shall be completed within the messaging unit. The operation of messaging unit includes the following steps:

1. Parameter Retrieve and Organization Function:

When sending a MAC management message, the ME triggers the parameter organizer function which is unique to the message type. The function identifies the linkage of system database items and message fields/TLVs. If necessary, the messaging unit triggers MEs to derive values for message fields/TLVs.

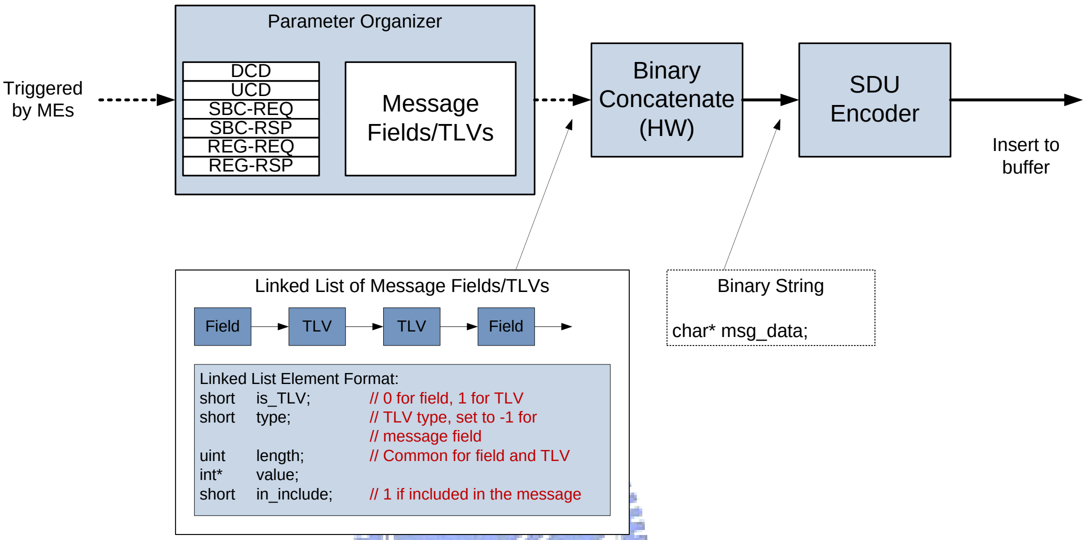
2. Message Field/TLV Listing

A message field/TLV may be included in or excluded from the message sent according to system configuration. The parameter organizer function is also in charge of the inclusion logic. The organizer function then concatenates message fields/TLVs into a linked list of an element.

3. Binary Concatenation Unit/SDU Encoder

The binary concatenation unit then transforms the linked list of message fields/TLVs into binary string. Padding-to-the-byte is performed if required. The binary string is then packed up as an SDU. The SDU encoder shall set up the sending priority and transport CID of the message SDU.

To boost up performance, it is suggested to perform binary concatenation and SDU encoding using hardware modules. In view of this, the messaging unit has to be equipped with multiple type-unique parameter organizer functions, and a common binary concatenation unit. It would facilitate HW/SW partition if the input elements to the HW binary concatenation unit, i.e. message fields and TLVs, are designed to be uniformly formatted. The operation of MAC messaging unit is illustrated in Figure 5-11.



Linked List Element Format:

```
short   is_TLV;          // 0 for field, 1 for TLV
short   type;            // TLV type, set to -1 for
                         // message field
uint    length;          // Common for field and TLV
int*    value;
short   in_include;      // 1 if included in the message
```

Figure 5-11: Details of Messaging Unit

5.4.5. Cross-Layer Design Example: Information Package

The PHY layer is equipped with an upper PHY module that aggregates binary data and channel/burst abstractions transmitted by the MAC layer. The binary data comes from the data plane in the MAC layer, while the channel/burst abstractions are generated by the MAC control plane.

Binary data and channel/burst abstractions together are referred to as the MAC/PHY cross-layer information package. The information package is frame-based, i.e. it contains all data and parameters required by PHY operation in single frame duration. The cross-layer information package includes:

1. Binary Data:

Binary raw data for data bursts, DL-MAP, UL-MAP and FCH. The raw data is neither encoded by FEC nor modulated. Binary data for each frame element shall be adjacently allocated in the system memory to improve PHY hardware performance.

2. Ready-to-Serve burst IEs:

Every frame elements shall have a burst IE mapped to it. The burst IE is an aggregation of burst profile in the channel descriptor (DCD and UCD), and MAP IE in the map (DL-MAP and UL-MAP). To sum up, the burst IE contains all required PHY parameters for a frame element's PHY transmission, including the DIUC, coding-modulation scheme, relative location in the frame and size of the two-dimensional allocation.

3. Ready-to-Serve Overall Channel Information:

A frame-based information package should embed a set of overall channel abstractions, including the time allocation, frame allocation and power management information of this frame. Also, the permutation base used in BS preamble shall be included.

4. Read-to-Serve Ranging Indications:

This part includes all necessary indications for the ranging subchannel, including ranging codes, backoff start/end and ranging subchannel allocation.

Besides the information package, the MAC and PHY layer also communicate through the system database. Measurements made by PHY is updated to the system database using a set of standard interfaces defined therein, while the MAC layer accesses these measurements in the system database using the same set of interfaces. Figure 5-12 is an illustrative graph of the two methodologies for MAC/PHY cross-layer signaling.

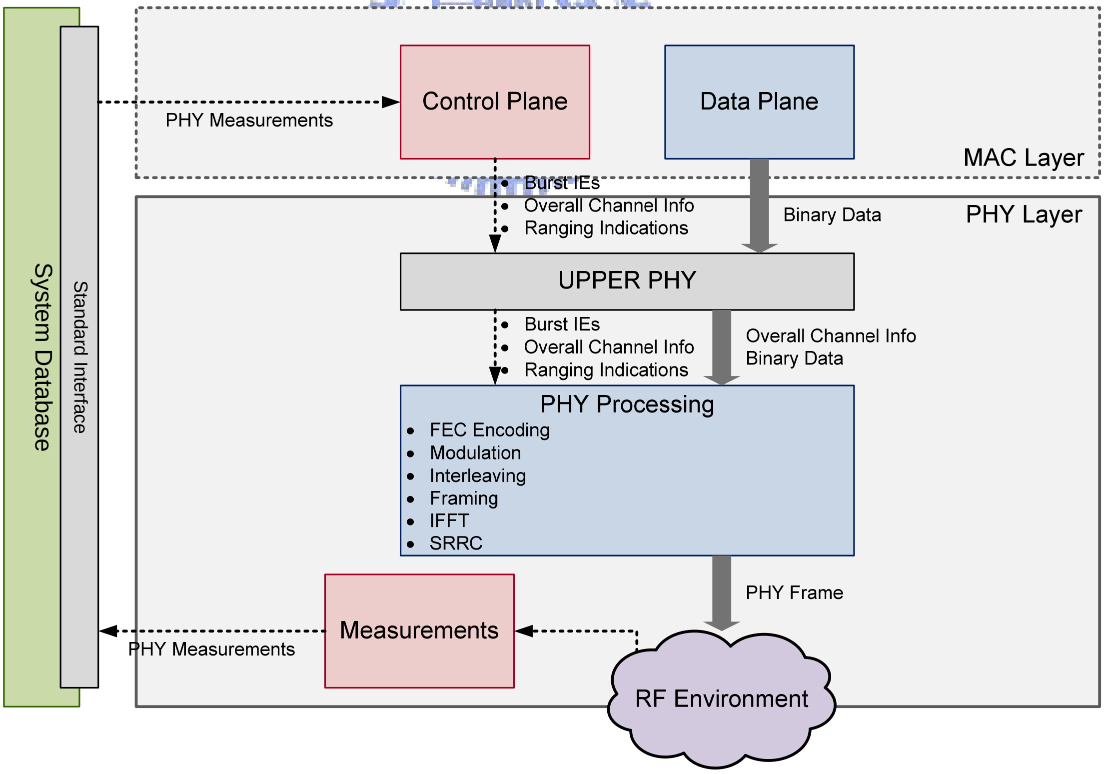


Figure 5-12: Two Methodologies for MAC/PHY Cross-Layer Signaling

Chapter 6. Conclusions and Future Works

In this article, a brief study of mobile WiMAX MAC layer implementation is presented. The discussion starts from a technical overview of the WiMAX system, then a proposed architecture of mobile WiMAX MAC layer, cross-layer methodology between MAC and PHY, and finally an illustrative case study of mobile WiMAX network entry.

In Chapter 3, the benefits of object-oriented system database design are depicted. In Chapter 4, we discuss the benefits of cross-layer signaling through system database, and how cross-layer information package facilitates batch processing in the PHY layer. Throughout this study, the most significant guideline for WiMAX MAC layer implementation is: modularized software design, and standard interface between any two modules in the system. Carefully-defined interface could help improving hardware performance, as well as enabling prompt information exchanging between different modules in the system.

Although this study provided guidelines for WiMAX MAC layer implementation along with a reference demo, it is still suggested to port the proposed architecture to conventional network simulator platforms, such as ns-2 and OPNET, as a basis for performance analysis. The most important reason is conventional platform may enhance reusability of the modules, as well as quick prototyping of new algorithms. Also, these platforms provide ready-to-use and realistic channel models.

Another future work worth doing is multi-system integration. Although this article focuses on mobile WiMAX implementation, the reference model illustrated in Figure 3-1 seems to be applicable to many wireless systems. To integrate multiple wireless systems on a device, it is necessary to reconfigure the MAC modules in order to save code size and provide easy maintenance. Modularized MAC design would lead a way to multi-system MAC integration and reconfiguration.

References

[1] IEEE 802.16 Working Group, “IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems,” IEEE Std. 802.16-2004, October 2004.

[2] IEEE 802.16 Working Group, “IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed Broadband Wireless Access Systems – Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1,” IEEE Std. 802.16-2005, Feb. 2006.

[3] Shawn Tsai et al., “Introduction to WiMAX Air Interface,” Radio Access Concepts & Standards Business Unit Access, Ericsson AB, Sept. 2006.

[4] Li-chuan Tseng, Kuan-yin Chen and ChingYao Huang, “Cross-Layer Design for IEEE 802.16-2005 System Using Platform-Based Methodologies,” International Workshop on System and Software Wireless SoC, Dec. 2007.

[5] Qi Wang and M. A. Abu-Rgheff, “Cross-Layer Signaling for Next-Generation Wireless Systems,” WCNC, vol. 2, Mar. 2003.

[6] V. Srivastava, “Cross-Layer Design: A Survey and the Road Ahead,” IEEE Communications Magazine, Dec. 2005.

[7] S. Shakkottai and T. S. Rappaport, “Cross-Layer Design for Wireless Networks,” IEEE Communications Magazine, Oct. 2003.

[8] V. T. Raisinghani and S. Iyer, “Cross-Layer Feedback Architecture for Mobile Device Protocol Stacks,” IEEE Communications Magazine, Jan. 2006.

[9] G. Carneiro, “Cross-Layer Design in 4G Wireless Terminals,” IEEE Wireless Communications, Apr. 2004.

[10] Taesoo Kwon et al., “Design and Implementation of a Simulator Based on a Cross-Layer Protocol between MAC and PHY Layers in a WiBro Compatible IEEE 802.16e OFDMA System,” IEEE Communications Magazine, Dec. 2005.

[11] Jenhui Chen et al., “The Design and Implementation of WiMAX Module for ns-2 Simulator,” ACM International Conference Proceeding Series; Vol. 202, 2006.

[12] S. Floyd, “TCP and Explicit Congestion Notification,” ACM SIGCOMM Computer Communication Review, vol. 24, Oct. 1994.

[13] Byoung-Jo Kim, “A network service providing wireless channel information for adaptive mobile applications: Part I: Proposal”, ICC 2001, June 2001.

作者簡歷

陳冠穎，民國七十二年出生於高雄市。民國九十四年自國立交通大學電子工程學系取得學士學位，並於同年進入同系碩士班就讀。作者從事之研究領域為無線通訊系統之架構設計與軟硬體實作。民國九十六年十二月取得碩士學位，學位論文題目為：「移動性 WiMAX 系統：跨層設計與軟體架構之系統觀點」。