Transcoding the System Information from the DVB-MHP to Blu-ray Format in Real-time

by

Zicong Mai

B. Eng., North China University of Technology, China, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES

(Electrical & Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

August 2007

© Zicong Mai, 2007
Abstract

The demand for interactive services in the broadcast world is rapidly increasing. The new-generation packaged medium, the Blu-ray system is designed to support many sophisticated interactivities. Since most interactive TV programs have the Digital Video Broadcast – Multimedia Home Platform (DVB-MHP) format, this study addresses how such programs could be played by the Blu-ray system in real-time.

Despite the fact that both the DVB-MHP and the Blu-ray standards are based on the MPEG-2 standard for video related applications and on the Globally Executable MHP (GEM) standard for Java-based interactivity, DVB-MHP and Blu-ray have several differences that make the two systems incompatible. One of the main challenges in realizing this compatibility is the conversion of the “system information” data. These data carry the information about the broadcasting programs and services. This thesis first analyzes the differences in the system information between the two standards, DVB-MHP and Blu-ray, mainly what information each standard requires, where they are stored, and how they are organized. We then propose methods that transcode this information efficiently, i.e., in real-time. Thus, to demultiplex the data needed for generating the Blu-ray system information, we propose a scheme that is 33% faster than the conventional method. To extract the needed information from the DVB-MHP video stream, we propose an efficient search algorithm. Finally, an improved transcoder structure is proposed. This transcoder only transcodes the incoming system information or its updated versions and at the same time keeps the rate of this information transmitted to the Blu-ray system the same as that in the DVB-MHP system.
Table of Contents

Abstract .......................................................................................................................... ii
Table of Contents ......................................................................................................... iii
List of Tables ................................................................................................................. v
List of Figures ................................................................................................................. vi
Acknowledgements ........................................................................................................ vii
1 Introduction ............................................................................................................... 1
2 Background ................................................................................................................ 4
  2.1 Digital Television Broadcasting ........................................................................... 4
    2.1.1 Features of Digital TV .................................................................................... 4
    2.1.2 DTV Broadcasting System ............................................................................. 6
    2.1.3 DTV Standards .............................................................................................. 9
  2.2 Interactive Television Technology ...................................................................... 10
    2.2.1 Interactive TV scenarios ................................................................................ 11
    2.2.2 Data Broadcast and ITV Middleware ........................................................... 12
    2.2.3 The DVB-MHP Standard and the Globally Executable MHP ....................... 15
  2.3 Blu-ray System .................................................................................................... 17
    2.3.1 Overview of the Blu-ray System ................................................................... 17
    2.3.2 Blu-ray High Definition Movie Mode ......................................................... 19
    2.3.3 Blu-ray BD-J Mode ....................................................................................... 20
  2.4 System Information and Video Sequence Metadata ........................................... 21
    2.4.1 MPEG-2 Transport Stream Source Packet .................................................. 21
    2.4.2 Program Specific Information/ Service Information (PSI/SI) Tables ........... 22
    2.4.3 Video Sequence Metadata .......................................................................... 30
3 Proposed System Information Transcoding Schemes .............................................. 34
  3.1 Introduction .......................................................................................................... 34
  3.2 PSI/SI Tables Supported by the Two Standards .................................................. 36
3.3 Transcoding Scheme for Program Association Table ........................................ 37
3.4 Transcoding Scheme for Program Map Table .................................................. 39
  3.4.1 Transcoding for the Blu-ray PMT fields .................................................. 40
  3.4.2 Transcoding to the Blu-ray PMT descriptors using DVB-MHP data ................. 43
3.5 Generating Blu-ray Selection Information Table Using a DVB-MHP stream ............. 47
3.6 General Architecture of the PSI/SI Transcoder ............................................. 50
3.7 Conclusions ...................................................................................................... 51

4 System Efficiency Improvement and Transcoding Results ................................... 53
  4.1 Introduction ..................................................................................................... 53
  4.2 Improved System Architecture of the PSI/SI Transcoder ................................. 53
  4.3 Fast Algorithm that Searches for Information Hidden in the Video Stream ......... 55
    4.3.1 Fast Search Algorithm for Typical DVB-MHP TS ................................... 56
    3.4.2 Best Search Algorithm for Non-typical DVB-MHP TS ......................... 59
  4.4 Efficient Demultiplexer for Our Transcoding System ................................... 63
  4.5 Results Generated from the PSI/SI Transcoder ............................................ 66
    4.5.1 Generated Program Association Table .................................................. 67
    4.5.2 Generated Program Map Table .............................................................. 69
    4.5.3 Generated Selection Information Table ................................................. 71
  4.6 Conclusions ...................................................................................................... 73

5 Conclusions and Future Work ............................................................................ 75
  5.1 Conclusions ..................................................................................................... 75
  5.2 Future Work .................................................................................................... 76

Bibliography ........................................................................................................... 77

Appendix A – List of Acronyms .............................................................................. 79
List of Tables

Table 2.1: Example of a PSI/SI table ................................................................. 24
Table 2.2: Component Descriptor ................................................................. 25
Table 2.3: Structure of the video sequence ............................................... 30
Table 2.4: Contents of the sequence_header() ......................................... 31
Table 3.1: Program Association Table ....................................................... 37
Table 3.2: Program Map Table ................................................................. 40
Table 3.3: Video stream_type in Blu-ray and DVB-MHP ........................... 42
Table 3.4: Selection Information Table ....................................................... 48
Table 3.5: Partial transport stream descriptor .......................................... 48
List of Figures

Figure 2.1: Illustration of digital television broadcast system ........................................... 7
Figure 2.2: Illustration of interactive TV scenarios ............................................................... 11
Figure 2.3: Overview of the software stack of DVB-MHP ...................................................... 16
Figure 2.4: Simplified structure of the Blu-ray system ........................................................... 18
Figure 2.5: Structure of the Blu-ray transport stream ............................................................ 20
Figure 2.6: The structure of the PSI/SI data in the transport stream ........................................ 28
Figure 2.7: The mechanism for demultiplexing a transport stream with PSI/SI tables ... 29
Figure 3.1: System architecture of the overall transcoding system ....................................... 35
Figure 3.2: Partial transport stream descriptor ........................................................................ 50
Figure 4.1: Optimized architecture of the PSI & SI transcoder ................................................. 54
Figure 4.2: Time comparison of the three different schemes that search for the information embedded in the video stream ......................................................... 58
Figure 4.3: The average search time using the three methods that search for different numbers of sequence_header() ................................................................. 59
Figure 4.4: Comparison of the schemes that search typical and non-typical DVB-MHP streams for needed information .............................................................. 63
Figure 4.5: Simplified architecture of the complete transcoding system ................................ 66
Figure 4.6: DVB-MHP Program Associate Table containing 11 programs ......................... 68
Figure 4.7: Generated Blu-ray Program Association Table containing only one program with the program_number 0x0D4C .............................................................. 68
Figure 4.8: DVB-MHP Program Map Table corresponding to the program with the program_number 0x0D4C .............................................................. 70
Figure 4.9: Generated Blu-ray Program Map Table ................................................................. 70
Figure 4.10: Generated Blu-ray Selection Information Table ................................................. 72
Acknowledgements

I would like to thank all those who make the completion of this thesis possible.

First of the foremost, I wish to express my deep and sincere gratitude to my supervisors Dr. Rabab Ward and Dr. Panos Nasiopoulos. Their support and guidance have been of great value in this thesis. Dr. Ward gave me an opportunity to study in Canada and work with her. Her extensive knowledge and logical way of thinking have offered me a new vision in conducting research. Dr. Nasiopoulos has been giving me excellent advice since the first stage of this study. This work cannot be completed without his encouragement and instructions.

I also want to express my warm thanks to the students in my lab. It is my great pleasure to work with these nice people who are always ready to help me at any moment. In particular, I am very grateful to Sergio Infante who had a lot of valuable discussions with me about the interactive TV project. Our lab administrator, Mehrdad Fatourechi has created a very comfortable working environment for us. The support from Colin Doutre, Qiang tang and Lino Evgueni Coria Mendoza was very helpful not only to my research but also to my life.

Lastly but most importantly, I also owe great gratitude to my parents. They brought me to this life, raised me and instructed me on many aspects. Now they morally and financially support me, their only child to study overseas without any complaining. To them I dedicate this thesis.
1 Introduction

Digital television (DTV) broadcast has been getting acceptance worldwide in recent years and is believed to completely replace the analog TV in the near future. However, traditional DTV programs can no longer satisfy all audiences who now are expecting more from the television. Interactive TV, which is considered as the merging of DTV and the Internet, has been gaining high popularity. At present, the Digital Video Broadcasting (DVB) standard is the most widely used television broadcasting system in the world. To meet the increasing demand for interactive services, the DVB standard has been recently expanded to support an open middleware system called Multimedia Home Platform (MHP) [1]. MHP is designed to provide functions appropriate for interactive TV contents. Recently, however, the new-generation packaged medium, the Blu-ray DVD system, has been introduced and the first Blu-ray player was released in 2006. Compared to the traditional DVD technology, Blu-ray offers advanced interactive features, high definition video quality and high storage capacity which is more than 5 times that of DVD [2]. In other words, Blu-ray offers the viewer more visual enjoyment as well as profound interactions with the content.

Both the DVB-MHP and the Blu-ray standards are based on the MPEG-2 standard [3] for video related applications and on the Globally Executable MHP (GEM) standard [4] for Java-based interactivity. However, DVB-MHP and Blu-ray have several differences that make the two systems incompatible. To enable a Blu-ray system to playback video and interactive contents transmitted by the DVB-MHP system in real-time, it is necessary to perform DVB-MHP to Blu-ray transcoding in real-time.

In addition to the video data, the interactive data and the transport stream packet header information, the metadata about TV programs should also be transcoded. These metadata are
known as system information and includes Program Specific Information (PSI) and Service Information (SI) data. PSI carries informative data used to demultiplex a transport stream, and SI provides information about programs to the user [3], [5]. For example, PSI and SI provides information relating data to a specific program, offers delivery information, and identifies the format of the transmitted multimedia data.

Blu-ray PSI is organized differently from DVB-MHP PSI even though they are both based on MPEG-2 PSI. Blu-ray SI is defined based on DVB-MHP SI, but Blu-ray introduces its own additional restrictions. Therefore, in order to play DVB-MHP contents on the Blu-ray system, it is necessary to convert PSI and SI from DVB-MHP to Blu-ray. Papers [6]–[9] discuss how to generate or store PSI & SI for broadcasting purposes, but very few studies describe how to transcode PSI & SI from one system to another. In transcoding PSI/SI to the Blu-ray system, one of the challenges is to analyze in detail all the differences between the DVB-MHP and the Blu-ray standards. Many efforts are also needed to develop effective schemes for generating Blu-ray PSI/SI using the existing DVB-MHP data. Furthermore, DVB-MHP PSI/SI is not sufficient to generate Blu-ray PSI/SI. We thus need information embedded in the DVB video stream. This is one of the challenging tasks.

In this thesis, we develop schemes to transcode the system information data in real-time from the DVB-MHP system to the Blu-ray system. The steps to realizing this are:

- Analyzing the similarities and the differences in the system information between the DVB-MHP and the Blu-ray standards;
- Proposing transcoding methods to ensure the desired compatibility by taking advantage of the existing similarities between the two systems;
• Developing schemes to allow real-time transcoding, including a) an improved architecture of the "PSI & SI transcoder", b) a fast algorithm that efficiently searches for the needed information embedded in the video stream, c) an efficient demultiplexing method that reduces the computational complexity of the transcoding system.

The rest of this thesis is organized as follows. In Chapter 2, we provide background information on digital television broadcast and the DVB standard, interactive TV technology and the DVB-MHP standard, the Blu-ray system, and the system information. The details of transcoding methods that convert the system information from DVB-MHP to Blu-ray are discussed in Chapter 3. In Chapter 4, we optimize the transcoding system to make the transcoding efficient. Finally, conclusions and possible future research are presented in Chapter 5.
2 Background

This chapter provides background information on 1) digital television (DTV) broadcast and the DVB standard, 2) interactive TV (iTV) technology and the DVB-MHP standard, 3) the two modes of the Blu-ray system, and 4) the system information (PSI/SI). In Section 2.1, we provide a description of the current DTV broadcasting system and the DVB standard. Section 2.2 provides the key technologies of interactive TV and an overview of the DVB-MHP standard. The high definition movie mode and the Java-based interactivity mode of the Blu-ray system are described in Section 2.3. In Section 2.4, we provide an overview on the system information and the metadata of a video sequence.

2.1 Digital Television Broadcasting

In this section, the basic background on digital television broadcast is covered. Section 2.1.1 discusses the four main features of digital TV and its advantages over analog broadcast. We describe the transport layer level of the DTV system in Section 2.1.2. In Section 2.1.3, we compare the main DTV standards today and provide more introductive information on the DVB standard.

2.1.1 Features of Digital TV

For a digital TV system, digital techniques are deployed to encode the video, audio, and the other data to be broadcast. These digital signals are sent to home receivers. Despite the fact that the transmission is still analog, digital television has far more advantages over analog TV.
DTV has more channels (programs) in one single physical channel (transport stream) than analog TV. In analog TV, only one single TV program can be transmitted in one physical transmission channel. However, in DTV, one transmission channel is capable of carrying several TV programs. Mainly two reasons lead to this. The first reason is that the bandwidth occupied by one typical DTV program is far smaller than that occupied by one analog TV program. Currently, most DTV systems encode a TV program applying standard-definition (SD) quality using about 4 – 5 Mbits/second [10]. An analog TV program offers about the same video quality as a SD signal but consumes much more bandwidth than DTV. The other reason is that DTV allows variable bit-rate coding while analog broadcast supports only constant bit-rate coding [10]. Encoding the TV contents with variable bit rate allows the optimization of the multiplexing.

Although there are more channels delivered in DTV, it is not a burden, in practice, to customize many programs based on regional needs. The reason for this is that adapting the programs is more convenient if done digitally than in analog mode. The DTV programs we receive at home can be international, national or provincial, and these programs may have to be customized in order to satisfy different viewers in different specific regions. For instance, this customization may be inserting local advertisements, regional weather and subtitles with different languages. Digital broadcasting enables such customization in an easy way by simply replacing the original digital signals with regional specific programming [10].

In addition to having more programs in a DTV system, digital broadcast offers better video and audio quality. For example, a DTV broadcast system using the North American standard is able to deliver up to around 19 Mbits/second in a 6-MHz terrestrial medium or around 38 Mbits/second in a 6-MHz cable channel [11]. This means that high definition
(HD) video and excellent quality audio can be transmitted with the current compression technologies. The resolution of the HD video varies from one standard to another, and the highest one is $1920 \times 1080$ pixels.

High-definition video and high-quality audio are attractive, but probably the most interesting thing for both viewers and the industry is the new applications and services that analog TV is not able to provide. Such applications may be non-interactive or interactive. For instance, statistics in a soccer game and extra information specific to a show can be viewed in non-interactive fashion. On the other hand, video on demand and TV e-commerce are good examples of interactive services. We should give credit to DTV in that it is the foundation and an enabler to interactive TV.

2.1.2 DTV Broadcasting System

In section 2.1.1, the chief attractive features of digital television broadcast are highlighted. The next concern is how the DTV broadcasting system works. In this subsection, we provide the system description of the DTV broadcast at its transport layer level.

Figure 2.1 shows the transmission model of a typical DTV broadcast system. The video and audio are encoded and compressed. The compressed streams are then multiplexed and inserted into a transport stream (TS) which is in the form of digital signals. The modulator converts the digital stream into a suitable analog waveform, which is to be sent via satellite, cable or terrestrial media. At the end of the delivery, our home device receives the digital signals and displays them on the television screen. Below, we describe the mechanism of this system in more detail.
The video and audio signals must be compressed before they are transmitted. Otherwise, the data rate is very high. For example, the data rate of a standard-definition video can reach as high as 106 Mbits/second [10], which is not acceptable by the transmission system. Most of the current DTV systems deploy the MPEG-2 compression scheme, but a
small number have started to try the H.264 compression standard. For both schemes, the raw output from the compression encoder is called the elementary stream (ES). Because an ES contains a complete piece of information about the video or audio, it is always very long and hard to manipulate. To resolve this problem, an ES is segmented into a number of packets called packetized elementary streams (PESs).

Video PESs and Audio PESs form one type of inputs to the multiplexer. The input PESs are usually from several DTV programs. Another source of inputs is the other data information, such as metadata for broadcast and interactive applications data. Inside the multiplexer, the PESs and the data are again segmented into smaller pieces, forming 188-byte packets. The output of the multiplexer is known as the transport stream (TS), and a 188-byte packet is called the transport stream (TS) source packet.

The modulator converts a digital transport stream into an analog wave that suitable for the transmission. There are several existing modulation methods. Different types of networks employ different methods. For example, in Europe, quadrature amplitude modulation (QAM) is used for cable networks, orthogonal frequency division multiplexing (OFDM) for terrestrial networks, and quadrature phase shift keying (QPSK) or binary phase shift keying (BPSK) for satellite networks [12].

After modulation, DTV signals can be delivered via terrestrial, cable, satellite or even IP-based networks. Each type of networks has its own advantages. Digital terrestrial broadcast provides services for home TV and is likely to make mobile services come true in the future [10]. A cable network automatically provides a return channel which is very essential for advanced interactive services. Satellite broadcast enjoys a wide area and allows a real shared network. In addition to the above three types of networks, the DTV industry has
presently been considering the IP-based network, which might eventually bring the convergence of the broadcasting and the Internet.

The digital receiver extracts the transmitted signals and sends them to a TV set. It also supports various other functions, such as demodulation, error correction, conditional access management, source decoding, interactive service control, and communication with the return channel [13]. The current receiver is in the form of a set-top-box that is connected to the television set. This means a user must acquire a new device with an additional remote control to access digital TV.

2.1.3 DTV Standards

The system described in Section 2.1.2 is generic to any DTV systems. However, due to technical, economical, political, and organizational issues, currently three main DTV standards exist in the world [14]. These are the Digital Video Broadcast (DVB) standard introduced by the DTV organization in Europe, the Advanced Television Systems Committee (ATSC) standard in North America and the Integrated Services Digital Broadcasting (ISDB) standard in Japan.

The three standards are different in their detailed technical approaches to broadcasting multimedia contents. Hence, one program produced by one standard cannot be run in a region which only supports a different standard. Among the three DTV standards, the DVB standard is most widely used in the world. All the countries in Europe as well as many countries in Asia, Africa, and South America deploy this standard.

The DVB standard is designed to support a large number of applications. At present, there are DVB specifications for satellite, cable, terrestrial, handheld terminals, and IP-based networks. Recently the goal of DVB is to combine the broadcast and the Internet and ensure
stability and interoperability [12]. In terms of the multimedia contents, DVB allows us to have a flexible choice of video and audio quality (including HD TV), to listen to radio program on TV, and to enjoy various interactive services.

In the DVB standard, MPEG-1, MPEG-2 and H.264 are used for the source coding of video, and a number of schemes (e.g., MPEG-2 and Dolby AC-3) are allowed for audio coding. The MPEG-2 standard was chosen for the construction of transport streams for non-IP-based networks. On the other hand, the Real-time Transport Protocol (RTP) is used for IP-based networks [12].

DVB MPEG-2 transport streams not only transmit video and audio data, but also other data information, including metadata used for broadcasting (i.e., Program Specific Information and Service Information) and data for other services (e.g., interactive services). Despite the fact that the data information is transmitted within an MPEG-2 transport stream (TS), four different data broadcast methods (i.e., encapsulation methods) are defined depending on the applications: data or object carousels, data streaming, data piping, and multi-protocol encapsulation [15].

2.2 Interactive Television Technology

As discussed in Section 2.1, probably the most attractive feature of digital TV is its interactive applications and services. This section discusses the interactive TV (iTV) technology and its standards. In Section 2.2.1 we provide possible scenarios of interactive TV today. Data broadcast and iTV middleware are presented in Section 2.2.2. Finally, we give a description of the most popular interactive TV standard, DVB-MHP, and its extension, the Globally Executable MHP (GEM) in Section 2.2.3.
2.2.1 Interactive TV scenarios

Figure 2.2: Illustration of interactive TV scenarios.

Figure 2.2 shows the scenarios of interactive TV services. Depending on the required technologies and the ways of interaction, all iTV services can be classified into three types of scenarios; these are enhanced broadcast, interactive services, and internet access [10].

In “enhanced broadcast”, only unidirectional channel is needed and the interactivity can be considered as push-mode services. The interaction occurs between the viewer and the home receiver, and it can be simply realized by using the data embedded in the transport streams. The very traditional interactive application, the electronic program guide (EPG) is a well-known example of enhanced broadcast. EPG makes use of Service Information which is sent periodically in the transport stream to provide program guides. Another typical example is the news report. Users can select their favorite news category or weather report.
Enabling “interactive services” requires a bi-directional channel. In other words, applications in this category can be pull-mode services. With the additional return channel, the viewer can interact with the broadcaster or a third-part server. The most typical example is video on demand (VOD). The movie selection guide can be embedded in the transport stream or acquired by the user via the return channel. After selecting a movie, the data about this movie are transmitted through the additional channel to the movie provider (a third-part server). Then the desired movie is delivered to the home device. The user can also playback the movie through the return channel. More and more new applications of this scenario have been emerging; these include program-specific quizzes, voting, gaming, and TV banking.

The last type of iTV is through “Internet access”. Internet access provides interaction not only between the user and the broadcaster, but also between the user and the Internet server. It means that the television could be used to browse websites, to check emails, and to enjoy all things that are done using a computer. Compared to the other two types of interactive TV, Internet access iTV requires a standardized IP connection and may need to support several internet protocols, such as Hypertext Transfer Protocol (HTTP), HTTP over Secure Socket Layer (HTTPS), and Simple Mail Transfer Protocol (SMTP).

### 2.2.2 Data Broadcast and iTV Middleware

In this sub-section, two issues are explored: data broadcast and iTV middleware. Data broadcast is an essential technology to transmit iTV applications while iTV middleware is the most important part in the home receiver.

In any interactive TV systems, we should be able to load data required by the application through the broadcast channel. The mechanism to resolve this problem is known as data broadcast. As discussed in Section 2.1.3, there are four ways (four encapsulation
formats) to broadcast data information defined by the DVB standard: data streaming, data or object carousels, multi-protocol encapsulation, and data piping. All these four approaches are designed based on the MPEG-2 transport stream structure and intended to encapsulate different types of data on top of this shared structure.

The data streaming approach encapsulates data into packetized elementary streams (PES), which functions the same way as encoding normal MPEG-2 video data for TV programs. With this encapsulation scheme, data streams can be asynchronous (i.e., no timing information), synchronous (i.e., conforming to a fixed-rate transmission clock) or synchronized (i.e., using time stamps for decoding and hence tied to other data type).

The second scheme is to use carousels which are designed for data requiring periodical transmission. The format of carousels is standardized by Digital Storage Media Command and Control (DSM-CC). Two types of carousels are supported, data carousels and object carousels. In data carousels, blocks of data with additional instructive information are sent to the receiver to allow the receiver gets the needed information. The concept of object carousels, on the other hand, is defined on top of data carousels and used for more complicated applications. The method of object carousels is the same as a standard file system structure in that it has the concepts of files, directories and streams [17].

Multi-protocol encapsulation is intended for carrying data which are only suitable for network protocols (e.g., IP traffic) in the broadcasting environment. With this scheme, we can use MPEG-2 transport streams to broadcast the data under other network protocols. For example in IP traffic, a stream of User Datagram Protocol (UDP) packets does not match the format of data carousels or object carousels, and it is also not suitable to carry it in a PES. In
this case, the stream of UDP packets can be fitted on top of an MPEG-2 transport stream using multi-protocol encapsulation.

Data piping is used to encapsulate data information for applications that are not supported by the other three data broadcast schemes. Data piping simply packs the needed data into transport stream source packets in a way based on the application type [16]. Practically speaking, this data encapsulation method is the least widely used one among the four approaches since the other three methods can satisfy almost all current interactive applications.

The above four data broadcast schemes play an important role in today's interactive services, but the schemes are used to encapsulate data for transmission. At the receiver end, however, the most critical part is the so-called interactive TV middleware.

Given an iTV application, there are two ways to run it on an iTV receiver. One is to compile it directly on the hardware platform. The other way is to allow the application to communicate with a virtual layer, and then the virtual layer communicates with the receiver operation system. The former method requires the application to be designed specific only to the hardware system. The second scheme makes it possible to design iTV applications in a generic way if the virtual layer inserted into a receiver is generalized. This virtual layer is known as middleware in the iTV world.

In general, middleware is defined as a virtual machine (VM) on top of a hardware system and serves as an abstract layer between the operating system and the applications running above it. In simple words, middleware is a relatively generic platform or application programming interface (API) that allows the receiver to access different iTV applications and services independent of the hardware system they are running on [10].
There are several technical advantages to develop interactive TV with middleware technology. Firstly, it paves the way for iTV receiver designers to hide important differences behind the hardware. Secondly, the middleware provides a standardized platform for iTV application developers regardless of various hardware designs used in the receiver.

Despite the fact that most current iTV middleware is Java-based solutions and many people consider it as a Java VM, iTV middleware is much more than just a software virtual machine. In fact, the iTV middleware must be able to manipulate many components specific to the broadcasting world, such as MPEG-related components. This is also one of the features that keep iTV middleware apart from any other programming environments.

### 2.2.3 The DVB-MHP Standard and the Globally Executable MHP

Middleware provides a generic platform for iTV applications and services. However, since different iTV standards define their own middleware specifications, an iTV application cannot usually work cross standards. There were a number of open middleware standards at the time the middleware solution called Multimedia Home Platform (MHP) [1] was introduced by the DVB standard. Nowadays, the DVB-MHP standard is the most popular standard for broadcasting interactivity and has been dominating the iTV world.

Similar to other middleware solutions, the DVB-MHP standard defines a number of models for various purposes. Examples of these models are the lifecycle model, the signaling model, the security model, and the graphics model. These models allow iTV applications to be loaded from DVB broadcasting networks and to be executed independent of the iTV receiver systems used [18].

All the above DVB-MHP models are designed jointly to support a large pool of applications and services. They can be grouped into three profiles: enhanced broadcasting,
interactive broadcasting, and Internet access [19]. Note that the profiles are defined in the same way as the three types of the iTV applications mentioned in Section 2.2.1.

![Diagram of software stack of DVB-MHP](image)

**Figure 2.3: Overview of the software stack of DVB-MHP.**

From the software aspect, the DVB-MHP platform is based on Java technology. In other words, on a DVB-MHP system, Java-based application programming interfaces (APIs) are used for manipulating iTV applications. Besides normal Java APIs, MHP APIs should include other APIs specific to digital broadcast, such as APIs for reading Service Information and APIs for controlling DSM-CC objects. Figure 2.3 shows one possible solution of software stack built in the DVB-MHP middleware [10]. It can be noted that a variety of components are created in addition to normal Java APIs.

In recent years, to avoid reinventing the wheel, the multimedia world (including broadcasting media and packaged media) has been showing an interest in adapting the DVB-MHP middleware in systems that do not conform to the DVB standard. The Open Cable Application Platform (OCAP) specification [20] of the American cable TV organization is one of the main driving-force behind adapting the DVB-MHP middleware. With the joint
efforts of many bodies, an adaptable middleware specification was finally introduced by the DVB standard and is named Globally Executable MHP (GEM) [4].

In its simple form, GEM can be viewed as a subset of MHP by removing a small number of elements which are specific to the DVB broadcast. Another important issue of GEM is that it is not a stand-alone specification and other standards should impose their own restrictions and add extra extensions to it. In spite of these restrictions and extensions, same as the MHP platform, GEM-based systems rely on the Java technology.

2.3 Blu-ray System

Blu-ray is the newest-generation packaged media format introduced recently. It supports high-definition video as well as advanced GEM-based interactivity. In Section 2.3.1, we provide an overview of the Blu-ray system. The two modes of the Blu-ray system: the high definition movie mode and the Java-based (GEM-based) interactivity mode are discussed in Section 2.3.2 and Section 2.3.3 respectively.

2.3.1 Overview of the Blu-ray System

The new packaged media data format is called Blu-ray because a blue-violet laser is applied to produce the disc, which is different from (the infrared laser) CD and (the red laser) DVD. With this brand-new technology, the capacity of the Blu-ray disc goes up to 25 GB for a single-layer disc and 50 GB for a double-layer one. This is much larger than the CD’s capacity (maximum 800 MB) and the DVD’s (4.7 GB).

In additional to the large capacity, Blu-ray enjoys many other innovations. One of them is that Blu-ray supports very high resolution video. Compared to the 720 × 480 resolution in traditional DVD technology, the video resolution in Blu-ray reaches as high as 1920 × 1080.
Another chief innovation of Blu-ray is its advanced interactive function that is never supported by any DVD system. The Blu-ray system takes advantage of the GEM specification and is designed to support Java-based applications which are always associated with profound interactivity.

Based on different presentation contents and the technologies in use, the application platform of the Blu-ray system can be categorized into two modes, high definition movie (HDMV) mode and Blu-ray Java-based interactivity (BD-J) mode. Figure 2.4 illustrates the simplified structure of the Blu-ray system [2], where the two modes are located under the “application environment”. In the following sub-sections, we explore the key concepts in the HDMV and the BD-J modes respectively.

![Figure 2.4: Simplified structure of the Blu-ray system.](image-url)
2.3.2 Blu-ray High Definition Movie Mode

The High Definition Movie (HDMV) mode can be viewed as an advanced extension of the traditional DVD model, but it offers lots of new features.

A main feature in this mode is its high definition video and good quality audio. As mentioned before, the video resolution can reach up to $1920 \times 1080$. Three types of video formats are supported: MPEG-2, H.264, and SMPTE VC-1. As for audio, besides being able to allow more audio streams (32 streams) in one title, the Blu-ray system supports a considerable number of formats, such as Linear PCM (LPCM), Dolby Digital Plus, Dolby Digital Lossless, DTS-High Definition.

In addition to its superior video and audio presentations, Blu-ray has advanced menu features and improved subtitle features, which can be considered as simple interaction. Unlike DVD, Blu-ray allows the menu presentations to be changed without interrupting content playback. In addition, full color HD buttons and animated menu transition effects are also supported by Blu-ray. Regarding subtitles, Blu-ray allows HD subtitles with animated full color images at up to the full video frame rate [2].

A very important concept in the Blu-ray standard is its transport stream (TS) which is designed for error-prone environments. In traditional DVD, multimedia contents are carried in program stream (PS) which is designed for error-free media. However, all Blu-ray video and audio should be encapsulated in its TS format. The Blu-ray TS is defined based on the MPEG-2 transport stream format and is also used to multiplex video and audio streams, but its structure is not the same as the normal MPEG-2 TS. Fig. 1.5 illustrates the structure of the Blu-ray transport stream [2], which consists of 6144-byte Aligned units, each Aligned unit
being constructed by 32 192-byte Blu-ray long source packets. The third row in Figure 2.5 shows the structure of the Blu-ray transport stream source packet [3]. The difference between the 188-byte MPEG-2 source packet and the 192-byte Blu-ray source packet is an extra 4-byte header, which contains the arrival time stamp and the copy permission information.

![Figure 2.5: Structure of the Blu-ray transport stream.](image)

**2.3.3 Blu-ray BD-J Mode**

The video and audio performance of traditional DVD seems satisfying enough for most viewers. In order to attract users to Blu-ray, new features are needed in addition to the improved video and audio quality. Hence, the Blu-ray standard introduced the BD-J mode which supports highly interactive applications. The BD-J mode is defined on top of the Globally Executable MHP (GEM) specification which uses Java-based technology. By deploying GEM in the Blu-ray system, the foundation for interoperability cross packaged media and broadcast media is built.

With the BD-J mode, the Blu-ray system is able to provide a large variety of interactive applications. Pre-loading applications into a Blu-ray title is a normal way to distribute interactive services. For example, along with the movie in a disc, users can also play games (e.g., solving a puzzle) and enjoy profound navigation of audio and video contents just by
using that disc. What is more important is that the BD-J mode is designed to enable access to
the Internet. For example, in addition to the languages pre-set in a disc, the BD-J applications
allow the user to download other languages for a movie title if the supporting information is
available on the web. To ensure security and protect copy rights through the network,
permissions are defined on the disc. These restrict the use of the Internet.

From the above, it is seen that the HDMV and the BD-J modes jointly construct an
advanced and attractive packaged media in the Blu-ray system.

2.4 System Information and Video Sequence Metadata

This section covers the background on the system information and the video sequence
metadata. To better understand these two topics, a brief overview of the MPEG-2 transport
stream (TS) source packet is provided in Section 2.4.1. In Section 2.4.2, we discuss the
format of the Program Specific Information/ Service Information (PSI/SI) tables, how the
PSI/SI tables are organized in a TS and how to demultiplex a TS using the system
information. The video sequence metadata are described in Section 2.4.3.

2.4.1 MPEG-2 Transport Stream Source Packet

At the broadcast head end, video, audio and data information (e.g., system information)
are multiplexed into a transport stream (TS), which is solely constructed from a long
sequence of 188-byte TS source packets. The TS source packet mainly comes from two
sources: packetized elementary stream (PES) and the data section. Video or audio data are
encapsulated in a PES while system information is carried in data sections. These two
sources are discussed in detail in the next two subsections.
Within a 188-byte TS source packet, at least 4 bytes are used as header and the rest as payload. Data in the payload can be video, audio or data information, but only one single type of data is allowed in one packet payload. On the other hand, the header which consists of a number of fields is used to identify the properties of the payload data. Among all the fields, three of them are very important in this thesis and discussed here. The first one is a 1-bit field called `payload_unit_start_indicator`. Setting this field to 1 is used to indicate that the current TS packet is the starting point of a PES. Another important field is the `PID` (Packet ID), which is used to identify the data type of the payload data. The `PID` value for a certain type of data may be pre-defined by the standard or assigned by the running system. For example, the `PID` value of the packets containing the Program Association Table (PAT) is always set to 0x0000, and the `PID` value of the packets carrying video data for a certain program is assigned by the transmitted Program Map Table (PMT). The third field mentioned is the `random_access_indicator`, which is a 1-bit field and should be set to 1 if the current packet contains information that aids random access [21].

### 2.4.2 Program Specific Information/ Service Information (PSI/SI) Tables

Compared to the analog television service, turning to a certain frequency in digital broadcast is not sufficient to select a desired program (channel). This is because in the transmission of DTV, multiple programs are multiplexed into one single transport stream (TS). In order to link packets in a TS to individual programs and identify the data type in the packet payload, we need extra information to define this mapping issue. On the other hand, we need the information of the physical DTV network to deal with the tuning issue. Moreover, to protect some pay-TV programs, their contents are scrambled or encrypted. Additional information should be provided to define this conditional access issue. All the
The above information should be transmitted to aid the receiver-end process. This information has been standardized by the MPEG-2 standard (part 1: systems) [3] and is known as the Program Specific Information (PSI).

The DVB standard is designed to offer profound DTV services. It transmits, for example, detailed information about TV programs to viewers. Thus, this kind of information is also standardized by the DVB standard and is called Service Information (SI). That is to say, in addition to PSI, DVB networks deliver SI as well.

PSI and SI share the same format and transmission mechanism. The collection of PSI and SI is called the system information or PSI/SI. By definition, PSI provides information mainly used to demultiplex a multi-program transport stream into different types of data streams (i.e., video, audio, PSI/SI data, and interactive application data) for a specific program. SI, on the other hand, is designed to offer informative data about broadcast services and programs.

In practice, the PSI/SI data are defined as a series of data bitstreams, known as tables. Different bitstreams (tables) are designed to provide different usages and each bitstream (table) is standardized by a unique syntax. In other words, a number of pre-defined bitstreams (tables) jointly realize the functionalities of PSI/SI. Similar to multimedia data, all PSI/SI tables are transmitted in a transport stream, but these tables are delivered periodically to aid random access.

2.4.2.1 Format of the PSI/SI Tables

The MPEG-2 and the DVB standards define a number of PSI/SI tables, such as the Program Association Table (PAT), the Program Map Table (PMT), the Conditional Access Table (CAT), the Service Description Table (SDT), and the Selection Information Table
(SIT). Each PSI/SI table (i.e., a stream of bits) follows a predefined structure and syntax, and every table can be considered to consist of two components, the basic fields and the descriptors. Table 2.1 shows the Program Map Table (PMT) [3] and illustrates these two components.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS_program_map_section()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>table id</td>
<td>8</td>
</tr>
<tr>
<td>section_syntax_indicator</td>
<td>1</td>
</tr>
<tr>
<td>reserved</td>
<td>1</td>
</tr>
<tr>
<td>section_length</td>
<td>2</td>
</tr>
<tr>
<td>program number</td>
<td>12</td>
</tr>
<tr>
<td>reserved</td>
<td>2</td>
</tr>
<tr>
<td>version number</td>
<td>5</td>
</tr>
<tr>
<td>current_next_indicator</td>
<td>3</td>
</tr>
<tr>
<td>section_number</td>
<td>1</td>
</tr>
<tr>
<td>last_section_number</td>
<td>8</td>
</tr>
<tr>
<td>reserved</td>
<td>3</td>
</tr>
<tr>
<td>PCR_PID</td>
<td>13</td>
</tr>
<tr>
<td>reserved</td>
<td>4</td>
</tr>
<tr>
<td>program_info_length</td>
<td>12</td>
</tr>
<tr>
<td>for (i = 0; i &lt; N; i++)</td>
<td></td>
</tr>
<tr>
<td>descriptor()</td>
<td></td>
</tr>
<tr>
<td>for (i = 0; i &lt; N2; i++)</td>
<td></td>
</tr>
<tr>
<td>descriptor()</td>
<td></td>
</tr>
<tr>
<td>CRC_32</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 2.1: Example of a PSI/SI table.

Basic fields are used to provide information about the table itself and to realize the basic function of the table. Each field has a fixed length of bits, and the fields in a table are strictly ordered. There are several common basic fields in all types of tables, such as table_id, the section_length, version_number, and CRC_32. The first 8 bits of a table must be table_id which is intended to identify the type of the table [3]. For instance, the values of table_id are set to 0x00 and 0x02 (the number after “0x” is in hexadecimal) respectively if the tables are PAT and PMT. The field section_length indicates the length of the current table in bytes. The
value of version_number changes if the current table (say, the current PMT) is updated from the previous one. The last 4 bytes of a table must be the CRC_32 field, which is used to verify the correctness of data transmitted in the table [3].

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Number of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>component_descriptor[]</code></td>
<td></td>
</tr>
<tr>
<td>descriptor_tag</td>
<td>8</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>8</td>
</tr>
<tr>
<td>reserved_future_use</td>
<td>4</td>
</tr>
<tr>
<td>stream_content</td>
<td>4</td>
</tr>
<tr>
<td>component_type</td>
<td>8</td>
</tr>
<tr>
<td>component_tag</td>
<td>8</td>
</tr>
<tr>
<td>ISO_639_language_code</td>
<td>24</td>
</tr>
<tr>
<td>for (i=0;i&lt;N;i++)</td>
<td></td>
</tr>
<tr>
<td>text_char</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2.2: Component Descriptor.

Descriptors are used in the PSI/SI to extend the definition of programs and define extra features [3]. For example, such descriptors provide detailed information (aspect ratio and frame rate) of the video in the program, indicate if the transmitted contents are scrambled, and provide the name of the program provider (say, HBO). We note that there are two “for” loops containing the element “descriptor()” in the PMT (see Table 2.1). Descriptor() in a table is a representative of a collection of descriptors used in the table. Different PSI/SI tables can have different types of descriptors. The types of descriptors in a table can be one or more than one. For instance, the descriptor() in the second “for” loop of PMT may represent three descriptors, `stream_identifier_descriptor()`, `component_descriptor()` and `mosaic_descriptor()`. Thus, the length of the representative descriptor() varies from table to table. In PMT, the length (in bytes) of the first “for” loop is defined by program_info_length, and the length of the second descriptor loop is identified by ES_info_length. Table 2.2 [5] displays `component_descriptor()`. Similar to a PSI/SI table, a descriptor is just a short stream.
of bits with predefined syntax. A descriptor has basic fields only. There are two common fields in all descriptors. The first 8 bits of a descriptor must be descriptor_tag, which is used to identify the type of the descriptor. The value of descriptor_tag, for example, of the component_descriptor() is fixed to 0x50. The field descriptor_length indicates the length of the current descriptor in bytes.

Several PSI/SI tables are very essential. These are Program Association Table (PAT), Program Map Table (PMT), Conditional Access Table (CAT), Network Information Table (NIT), and Selection Information Table (SIT):

• Program Association Table (PAT) is used to indicate the location of Program Map Table (PMT) in the transport stream by assigning a certain value to the Packet ID of the TS source packets containing PMT.

• Program Map Table (PMT) is mainly used to identify the locations (i.e., packet IDs) of the different elements (video, audio or data) in a transport stream and provide information that maps the video, the audio or the data in TS to a specific program [3].

• Conditional Access Table (CAT) is intended to provide information on the scrambling or encryption modes of contents in the transport stream.

• Network Information Table (NIT) is used to identify the physical network [5].

• Selection Information Table (SIT) is originally designed for storage media. It provides a summary of the Service Information in order to describe the streams used for storage [3]. Note that SIT is not broadcasted from the head end but generated by the DTV receiver.

To aid the random access to a TV program, the PSI/SI tables are transmitted periodically. The standards usually impose a lower bound on the frequency of the repetition.
The DVB standard, for example, recommends that PAT should be transmitted repeatedly at least every 100ms. Note that if they are delivered too often, then redundant overhead in the transport stream are introduced.

2.4.1.2 PSI/SI Tables in a Transport Stream

As discussed in Section 2.1.2, the data section is one of the two sources to create TS source packets. PSI/SI tables are transmitted using such data sections, and they are known as PSI/SI table sections. To provide a good explanation of how PSI/SI tables are organized in a transport stream, we compare below the structure of a video stream and PSI/SI tables at the transport layer level.

PSI/SI tables are transmitted in a similar fashion to the video contents in a transport stream (TS). Fig. 1.6(a) shows the structure of the video data in a transport stream. Note that the “H” in Fig. 1.6 denotes the TS header, followed by the payload. In broadcasting, a video elementary stream (ES) is the raw output of the video encoder, and each stream is segmented into packets that are known as packetized elementary streams (PESs). Then each video PES is divided into smaller packets which are inserted into the payloads of TS packets. Fig. 1.6(b) depicts how PSI/SI tables are organized in a transport stream. A set of PSI/SI tables is similar to a video elementary stream (ES). Each specific PSI/SI table will form a section which is just like a video PES. For example, a Program Association Table (PAT) section, a Program Map Table (PMT) section and a Service Description Table (SDT) resemble three video packetized elementary streams (PES). If the size of a section is large, then this section is divided into small packets, and inserted into the payloads of TS packets (see SDT section in Fig. 1.6(c)). On the other hand, if the section is small enough, then this section occupies only
one single TS packet (see PAT section in Fig. 1.6(b)). Therefore, in a typical transport stream, there are TS packets of video, audio, and PSI/SI tables (see Fig. 1.6(c)).

![Diagram of transport stream structure](image)

**Figure 2.6: The structure of the PSI/SI data in the transport stream.**

### 2.4.1.3 Using PSI/SI Tables to Demultiplex a Transport Stream

The demultiplexer uses the Packet ID (PID) to separate a transport stream (TS) into different sub-streams (e.g. video stream of Program A, audio stream of Program B, and the PMT of Program C). The Packet ID is located in the header of a TS packet and designed to identify the data (e.g., video of Program A, audio of Program B, and the PMT data of Program C) in that TS packet payload. A few PID values are pre-set by the DVB standard while others are assigned by the transmitted PSI/SI tables.
Figure 2.7 illustrates how a transport stream is demultiplexed using PSI/SI tables. The very first TS packet extracted by the demultiplexer must be the packet that contains Program Association Table (PAT). The demultiplexer is able to identify the PAT packet because its PID value is fixed to 0x0000 by the DVB standard. Once PAT is obtained, the demultiplexer starts reading its contents. PAT defines an individual PID value for each Program Map Table (PMT), and each PMT corresponds to a specific program. Now the demultiplexer knows the PID value of each PMT, so the second step in demultiplexing is to extract the packets which contain different PMTs (e.g., PMT of Program A and PMT of Program B).

<table>
<thead>
<tr>
<th>TS</th>
<th>PID</th>
<th>Payload Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>PAT</td>
<td></td>
</tr>
<tr>
<td>0x0100</td>
<td>Video in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0103</td>
<td>Audio 1 in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0108</td>
<td>Video in Program B</td>
<td></td>
</tr>
<tr>
<td>0x0109</td>
<td>Audio 2 in Program B</td>
<td></td>
</tr>
<tr>
<td>0x0103</td>
<td>Audio 2 in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0300</td>
<td>PMT for Program A</td>
<td></td>
</tr>
<tr>
<td>0x0108</td>
<td>Video in Program B</td>
<td></td>
</tr>
<tr>
<td>0x0301</td>
<td>PMT for Program B</td>
<td></td>
</tr>
<tr>
<td>0x0100</td>
<td>Video in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0109</td>
<td>Audio 1 in Program B</td>
<td></td>
</tr>
<tr>
<td>0x0102</td>
<td>Audio 1 in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0103</td>
<td>Audio 2 in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0106</td>
<td>Interactive Data in Program A</td>
<td></td>
</tr>
<tr>
<td>0x0108</td>
<td>Video in Program B</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAT</th>
<th>Program</th>
<th>PID for PMT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>0x0300</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>0x0301</td>
</tr>
</tbody>
</table>

<p>| PMT for Program A |</p>
<table>
<thead>
<tr>
<th>Data Type</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>0x0100</td>
</tr>
<tr>
<td>Audio 1</td>
<td>0x0102</td>
</tr>
<tr>
<td>Audio 2</td>
<td>0x0103</td>
</tr>
<tr>
<td>Interactive Data</td>
<td>0x0106</td>
</tr>
</tbody>
</table>

<p>| PMT for Program B |</p>
<table>
<thead>
<tr>
<th>Data Type</th>
<th>PID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>0x0108</td>
</tr>
<tr>
<td>Audio</td>
<td>0x0109</td>
</tr>
</tbody>
</table>

Figure 2.7: The mechanism for demultiplexing a transport stream with PSI/SI tables.

Program Map Table assigns the PID values to different types of multimedia data in the corresponding program. For example (see Figure 2.7), the PMT for Program A may assign the PID value of 0x0100 to the video of Program A, the PID value of 0x0102 to the first
audio (say, English) of Program A, the PID value of 0x0103 to the second audio (say, French) of Program A, and the PID 0x0106 to the interactive application data of Program A. By reading the information within different PMTs, the demultiplexer is capable of distinguishing different programs and identifying the data type in each program in a long transport stream.

2.4.3 Video Sequence Metadata

2.4.3.1 Format of the Video Sequence Metadata

```
video_sequence() {
    next_start_code()
    sequence_header()
    if (nextbitsQ == extension_start_code) {
        sequence_extension()
        do {
            extension_and_user_data(0)
            do {
                if (nextbitsQ == group_start_code) {
                    group_of_pictures_header()
                    extension_and_user_data(1)
                } else {
                    / ISO-TEC 11172-2 */
                }
                picture_header()
                picture_coding_extension()
                extensions_and_user_data(2)
                picture_data()
            } while (nextbitsQ == picture_start_code) ||
            (nextbitsQ == group_start_code) ||
            (nextbitsQ == extension_start_code)
        } else {
            / ISO-TEC 11172-2 */
        }
    } else {
        sequence_end_code
    }
}

Table 2.3: Structure of the video sequence.
After compression, the coded video data are organized as ordered sets of video bitstreams, which are also known as layers. There is only one single layer if the video is non-scalable. Each bitstream can be viewed as a structure of a syntactic hierarchy, and the highest structure is called the "video sequence", which includes one or a few coded frames [21].

A video sequence has several metadata units, which are used to provide information. These units indicate, for example, the frame rate and the scanning scheme (interlaced or progressive) of a video. The units include a video sequence header, a few extensions and several picture headers. The video sequence header is always followed by other metadata units in a video sequence. Table 2.3 shows the structure of the sequence header. It is noted that following sequence_header(), there are other metadata elements, such as sequence_extension(), group_of_pictures_header(), and picture_coding_extension().

<table>
<thead>
<tr>
<th>sequence_header()</th>
<th>No. of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequence_header_code</td>
<td>32</td>
</tr>
<tr>
<td>horizontal_size_value</td>
<td>12</td>
</tr>
<tr>
<td>vertical_size_value</td>
<td>12</td>
</tr>
<tr>
<td>aspect_ratio_information</td>
<td>4</td>
</tr>
<tr>
<td>frame_rate_code</td>
<td>4</td>
</tr>
<tr>
<td>bit_rate_value</td>
<td>18</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
</tr>
<tr>
<td>vbv_buffer_size_value</td>
<td>10</td>
</tr>
<tr>
<td>constrained_parameters_flag</td>
<td>1</td>
</tr>
<tr>
<td>load_intra_quantiser_matrix</td>
<td>8*64</td>
</tr>
<tr>
<td>if (load_intra_quantiser_matrix)</td>
<td></td>
</tr>
<tr>
<td>intra_quantiser_matrix[64]</td>
<td>8*64</td>
</tr>
<tr>
<td>load_non_intra_quantiser_matrix</td>
<td>1</td>
</tr>
<tr>
<td>if (load_non_intra_quantiser_matrix)</td>
<td></td>
</tr>
<tr>
<td>non_intra_quantiser_matrix[64]</td>
<td>8*64</td>
</tr>
<tr>
<td>next_start_code()</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.4: Contents of the sequence_header().
Each metadata element has a unique standardized structure, and a number of fields that are pre-defined. Table 2.4 illustrates the contents of `sequence_header()`. The horizontal and the vertical sizes of the video, for instance, can be found in `sequence_header()`, and they each occupies 12 bits immediately after the first 32 bits.

Note that all the video metadata elements are located within the video bitstream. In order to identify them, each element begins with a pre-defined unique start code, which is always 32 bits (4bytes). For instance, the start code of `sequence_header()` is 0x000001B3, and that of `sequence_extension()` is 0x000001B5. Using the start codes, we can find the needed metadata elements in a bitstream which contains only 0 and 1.

Among all the video metadata elements, `sequence_header()` is the most important one since it aids the random access to a video stream. Hence, in the broadcasting system, like PSI/SI tables, `sequence_header()` should be sent periodically.

### 2.4.3.2 Video Sequence Metadata in a Transport Stream

As explained in Section 2.1.2, in addition to data sections, packetized elementary streams (PESs) are another source of the 188-byte transport stream packets. Video data are contained in the PES.

The raw data after compression (e.g., compressed video data) are called the elementary stream (ES), which is segmented into a number of packets (whose size is much larger than 188 bytes) known as packetized elementary stream (PES). Then each PES is again divided into smaller packets, each having 188 bytes and known as a TS source packet.

Since the video sequence metadata elements (e.g., `sequence_header()`) are produced together with the raw output after compression and located also in the video ES, they are eventually distributed in TS source packets. Note that the locations of the video metadata
elements in a transport stream cannot be known a priori. Normally we identify them only by finding their start codes.
3 Proposed System Information Transcoding Schemes

3.1 Introduction

Despite of the fact that both the DVB-MHP and the Blu-ray standards are based on the MPEG-2 standard [3] for video related applications and on the Globally Executable MHP (GEM) standard [4] for Java-based interactivity, there are several differences that make the two systems not incompatible. In order to enable a Blu-ray system to playback, in real-time, video and interactive contents which are broadcasted by the DVB-MHP system, transcoding the data from the DVB-MHP system to the Blu-ray system is needed.

Figure 3.1 shows the system architecture of the overall transcoding system which converts the DVB-MHP contents to the Blu-ray compliant format. In digital video broadcast, a transport stream (TS) usually contains multiple programs and is denoted as multi-program transport stream (MPTS). Before performing the transcoding, the MPTS is first demultiplexed and separated into its video, audio, system information data and the file data (which is used for interactive applications) components. In addition to the video data, the interactive data and the transport stream packet header information, the metadata about TV programs should also be transcoded. The metadata are known as system information and include Program Specific Information (PSI) and Service Information (SI) data. PSI carries informative data used to demultiplex a transport stream, and SI provides information about programs to the user [3], [5].
Figure 3.1: System architecture of the overall transcoding system.

Blu-ray PSI is organized differently from DVB-MHP PSI even though they are both based on MPEG-2 PSI. Blu-ray SI is defined based on DVB-MHP SI, but Blu-ray introduces its own additional restrictions. Thus, transcoding PSI/SI from DVB-MHP to Blu-ray is needed in order to play DVB-MHP contents on the Blu-ray system.
In this chapter, the "PSI & SI transcoder" shown in grey in Figure 3.1 is discussed in detail. We analyze the compatibility in the system information (PSI/SI) of the two standards, Blu-ray and DVB-MHP, and propose methods to generate (in real-time) the Blu-ray PSI/SI from DVB-MHP data using the existing correlations between the two systems.

The rest of this chapter is organized as follows. The scheme that extracts PSI/SI tables which we use for transcoding from the DVB-MHP system is described in Section 3.2. In Section 3.3 and Section 3.4, the transcoding schemes for Program Association Table and Program Map Table are presented respectively. In Section 3.5, we develop a scheme that generates Blu-ray Selection Information Table using the transmitted DVB-MHP stream. In Section 3.6, the general architecture of the PSI/SI transcoder that converts from DVB-MHP data to Blu-ray PSI/SI tables is described. Finally, conclusions are presented in Section 3.7.

3.2 PSI/SI Tables Supported by the Two Standards

As explained in Chapter 2, PSI and SI data are defined as a series of data bitstreams which are named tables. The types of PSI and SI tables supported by the DVB and the Blu-ray standards are not identical. The Blu-ray transport stream is allowed to contain only 3 tables: Program Association Table (PAT), Program Map Table (PMT) and Selection Information Table (SIT). In contrast, DVB supports many more tables, such as Conditional Access Table (CAT), Network Information Table (NIT) and Running Status Table (RST).

Our task is to determine how to generate the necessary information to construct the PSI and SI tables of the Blu-ray system using tables supported by DVB-MHP. After analysis, we find that only PAT and PMT in DVB-MHP are useful in generating Blu-ray PSI/SI tables. Hence, in order to make the Blu-ray system compatible with the DVB-MHP contents in
terms of PSI and SI information, we propose to extract only PAT and PMT from the DVB-MHP PSI/SI tables. The other tables present in DVB-MHP are simply ignored.

Below we describe in detail the differences in the three tables between the two standards and develop transcoding methods for generating Blu-ray compatible PAT, PMT and SIT information.

### 3.3 Transcoding Scheme for Program Association Table

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>program_association_section()</code></td>
<td></td>
</tr>
<tr>
<td><code>table_id</code></td>
<td>8</td>
</tr>
<tr>
<td><code>section_syntax_indicator</code></td>
<td>1</td>
</tr>
<tr>
<td><code>reserved</code></td>
<td>1</td>
</tr>
<tr>
<td><code>section_length</code></td>
<td>12</td>
</tr>
<tr>
<td><code>transport_stream_id</code></td>
<td>16</td>
</tr>
<tr>
<td><code>reserved</code></td>
<td>2</td>
</tr>
<tr>
<td><code>version_number</code></td>
<td>5</td>
</tr>
<tr>
<td><code>current_next_indicator</code></td>
<td>1</td>
</tr>
<tr>
<td><code>section_number</code></td>
<td>8</td>
</tr>
<tr>
<td><code>last_section_number</code></td>
<td>8</td>
</tr>
<tr>
<td><code>for (i = 0; i &lt; N; i++)</code></td>
<td></td>
</tr>
<tr>
<td><code>program_number</code></td>
<td>16</td>
</tr>
<tr>
<td><code>reserved</code></td>
<td>3</td>
</tr>
<tr>
<td><code>if (program_number == 0) {</code></td>
<td></td>
</tr>
<tr>
<td><code>network_PID</code></td>
<td>13</td>
</tr>
<tr>
<td><code>}</code></td>
<td></td>
</tr>
<tr>
<td><code>else {</code></td>
<td></td>
</tr>
<tr>
<td><code>program_map_PID</code></td>
<td>13</td>
</tr>
<tr>
<td><code>}</code></td>
<td></td>
</tr>
<tr>
<td><code>CRC_32</code></td>
<td>32</td>
</tr>
</tbody>
</table>

**Table 3.1: Program Association Table**

Table 3.1 shows the format of Program Association Table (PAT) [3], which is used to identify PMTs of different programs by defining their packet IDs (PIDs). Note that there is a "for" loop in Table III. The number of iterations in the loop is usually equal to the number of programs in that transport stream plus 1. The extra iteration is executed when the `program_number` field is set to zero in the iteration. In this case, according to the MPEG-2 standard, `network_PID` is executed and indicates the PID value of Network Information
Table (NIT). Otherwise, if the value of the field `program_number` is not zero in that iteration, then the `program_map_PID` field is used to assign the PID value to PMT of a specific program.

Although the structure and syntax in both the DVB and the Blu-ray standards are the same, DVB conforms to the MPEG-2 standard, while the Blu-ray imposes the following additional constraints on the values allowed for some of the fields:

- The `program_number` field must be set to 0x0001 [2]. This is because the Blu-ray transport stream is allowed to contain only one single program.

- The value of the `network_PID` field must take the value 0x001F in Blu-ray [2]. Since `program_number` is always non-zero, this field is never visited by the demultiplexer. The reason for assigning a value to this field is that this fixed value 0x001F is used as the PID value of the Selection Information Table (SIT) supported by Blu-ray.

- The value of `program_map_PID` must be set to 0x0100 [2]. Because of the fact that only one program is allowed in Blu-ray, only one Program Map Table is needed in the transport stream. In such a case, the PID value for PMT can be fixed.

To generate a valid Blu-ray PAT using the existing DVB PAT, necessary modification should be considered. Several iterations are allowed in the “for” loop in a DVB PAT while only one iteration should be executed in Blu-ray since only one `program_number` is allowed. Thus, we propose to keep the first iteration in DVB PAT and remove all other iterations. This process is equivalent to shortening the length of the table.

After obtaining a table with a correct size, the values of some fields in the re-sized DVB PAT should be changed. In DVB, the value of `program_number` varies, but its value in Blu-ray is fixed, so we should change `program_number` to the fixed value 0x0001. The
value of `program_map_PID` is changeable in DVB, but in Blu-ray it should always be set to 0x0100. Moreover, we should also modify the `section_length` field (which assigns the number of bytes of the PAT section immediately after this `section_length` field) should be changed. Because only one iteration is allowed in the “for” loop, the size of the Blu-ray PAT should be fixed as 16 bytes. Since the `section_length` field and its previous fields jointly occupy 3 bytes, `section_length` should then be set to 13.

In addition to the above changes, the multi-program transport stream transmitted by DVB-MHP should be changed to a single-program transport stream as only one program is allowed in Blu-ray. The value of the packet ID in the header of TS packets that contain PMT should be also modified to 0x0100.

### 3.4 Transcoding Scheme for Program Map Table

Table 3.2 shows the format of Program Map Table (PMT). A PMT in a transport stream corresponds to a specific program, and the main function of PMT is to map different multimedia data in the TS packet payload to the corresponding program by assigning packet IDs [3]. Note that there are three “for” loops in Table I, and the second one from the top is called the stream loop. In general, the number of iterations of the stream loop is equivalent to the number of data types supported by the corresponding program. For example, if there are one MPEG-2 video, two AC-3 audios (say, in English and French), and one type of interactive application data, then four iterations should be executed in the stream loop. In each loop, the `stream_type` field identifies the data type (e.g., MPEG-2 video or AC-3 audio) and `elementary_PID` assigns the Packet ID to this data type.
Table 3.2: Program Map Table

The Blu-ray and the DVB specifications have the same Program Map Table structure\[2\], [3], [5]. However, the Blu-ray and the DVB standards impose different constraints on the values the fields and the descriptors are allowed to have. Below we show how to generate valid Blu-ray PMT fields and descriptors using the transmitted DVB-MHP information.

3.4.1 Transcoding for the Blu-ray PMT fields

The Blu-ray and the DVB specifications have the same Program Map Table structure\[2\], [3], [5]. However, the Blu-ray and the DVB standards impose different constraints on the values the fields and on the values the descriptors are allowed to have. Below we show how to generate valid Blu-ray PMT fields and descriptors using the transmitted DVB-MHP information.
The Blu-ray standard defines extra constraints on its fields in PMT while the DVB standard is mainly compliant with the MPEG-2 standard:

- According to the Blu-ray specification, the value of program_number must be set to 0x0001 [2]. This is because this field corresponds to the value of program_number in PAT. In Blu-ray PAT, the program_number must be set to 0x0001, so in PMT, this field must be fixed to this value.

- The Blu-ray specification fixes the value of PCR_PID to 0x1001 [2]. The PCR_PID field is used to assign the PID values of the packets that contain the Program Clock Reference (PCR) field for the program specified by program_number [3]. Since there is only one program allowed in Blu-ray, fixing the value of this field does not cause any confusion.

- According to the Blu-ray standard, the PID value of the packets that compose the primary video stream for that program is always set to 0x1011, and PID for other videos (i.e., secondary videos) should be numbered consecutively from 0x1B00 to 0x1B1F [2]. Again, there is only one program in Blu-ray, so the PID value of its primary video and the indexed PID values of the secondary videos can be fixed without causing any ambiguity.

- The stream_type field is used to identify the data type supported by the program. The video types supported by the two standards are different. Table 3.3 shows the values that this field may have and the comparison of video formats supported by the two standards [2], [3] and [12]. Both Blu-ray and DVB support MPEG-2 and H.264 video streams. The difference is that the DVB standard does not support the VC-1 video format while MPEG-1 video is not supported in the Blu-ray system.
### Table 3.3: Video stream_type in Blu-ray and DVB-MHP

In order to make DVB-MHP PMT compatible with that in a Blu-ray player, the values of some fields should be modified accordingly.

In the DVB system, different programs have different `program_number` values. We should fix this value to 0x0001. In a similar fashion in DVB, PID of the Program Clock Reference (PCR) may vary from one program to another within a certain range (0x0000, 0x0001, and 0x0010 - 0x1FFE) [2], [3], and [5]. For generating a Blu-ray PMT, we should also fix `PCR_PID` to 0x1001.

The `program_info_length` field, which specifies the number of bytes in the first “for” loop in PMT (see Table 3.2), also needs to be modified. To play broadcasting contents on the Blu-ray system, only two descriptors (the HDMV registration descriptor and the HDMV copy control descriptor) are required to appear in the first “for” loop, and they are 12 bytes in total. Hence, the `program_info_length` should be assigned the value 12.

After modifying `program_info_length`, we should consider `stream_type` and `elementary_PID` in the second “for” loop. As explained above, the video stream type supported by DVB are MPEG-1, MPEG-2 and H.264, but Blu-ray supports only MPEG-2 and H.264 video. Hence, if `stream_type` in DVB-MHP PMT is 0x02 (MPEG-2 video) or 0x1B (H.264 video), then in Blu-ray PMT, the value of `stream_type` should remain the same,
but *elementary_PID* of the primary video must be changed to 0x1011. For secondary video streams, we should change the number of their *elementary_PID*'s consecutively from 0x1B00 to 0x1B1F. On the other hand, if *stream_type* in DVB is 0x01 (MPEG-1 video), we must transcode the MPEG-1 video to one of other video formats supported in Blu-ray. After transcoding, the value of *stream_type* should be changed to a valid number according to Table 3.3.

The *section_length* field (which indicates the number of bytes of the whole PMT section immediately after this field) should be changed as well. It is the last field to be assigned a new value. After the number of iterations in the second “for” loop and the number of descriptors in each iteration are determined, the size of the second “for” loop can be calculated. Since we have already specified the size of the first “for” loop, now the size of the complete PMT table is determined. Thus, the value of *section_length* is specified accordingly.

In addition to the above modification, the PID value of the TS packets that carry the primary video should be changed to 0x1011. The PID value of the TS packets that carry a secondary video should also be modified according to the value of this video’s corresponding *element_PID*.

### 3.4.2 Transcoding to the Blu-ray PMT descriptors using DVB-MHP data

Descriptors are used to extend the definition of a program and provide information about the running environment of the program. For example, descriptors describe details (e.g., aspect ratio and frame rate) of the video in a program and provide information about the copy right control. Since DVB-MHP is a real-time broadcast system while Blu-ray is
intentionally designed for pre-packaged applications, descriptors defined by the two standards are very different.

Only 11 types of descriptors are supported by Blu-ray Program Map Table (PMT) [2]. Some are used to identify the format of a transport stream, some describe the features of a video stream, and some others control the copy right. However, in the DVB system, over 20 types of descriptors are permitted in PMT. What is more important is that only one of these descriptors is exactly the same in both standards.

Among the 11 descriptors allowed in Blu-ray PMT, 6 are used to play the video contents: \textit{HDMV\_registration\_descriptor()}, \textit{HDMV\_copy\_control\_descriptor()}, \textit{HDMV\_video\_registration\_descriptor()}, \textit{AVC\_video\_descriptor()}, \textit{VC-1\_registration\_descriptor()}, and \textit{AVC\_timing\_and\_HRD\_descriptor()}. The first two descriptors must be located in the first “for” loop in PMT, the others are located in the second descriptor loop (the third “for” loop).

We now discuss each of the 6 video-related PMT descriptors supported by Blu-ray and show how to generate them. It is found that it is difficult to generate two of these PMT descriptors, because unlike the others, the information in DVB-MHP PSI/SI tables is not enough to generate them. To overcome this difficulty we search for the desired information in the video stream.

\textit{VC-1\_registration\_descriptor()} is used to provide detailed information about the VC-1 video if it is running in the Blu-ray system. The DVB-MHP does not support VC-1 video, thus \textit{VC-1\_registration\_descriptor()} should not be present in the Blu-ray PMT after transcoding PMT descriptors.

\footnote{() appearing at the end of a term such as in \textit{HDMV\_registration\_des criptor() indicates that the existence of content in this descriptor.}
AVC_video_descriptor() must be present when the transport stream contains an MPEG-4 AVC still picture, and the DVB-MHP and the Blu-ray systems are fully compatible with regard to this descriptor. Therefore, our transcoder will maintain this descriptor if it is present in DVB-MHP PMT.

HDMV_registration_descriptor() is supported only by Blu-ray and is used to indicate that the architecture of the transport stream conforms to the Blu-ray standard [2]. This descriptor has three fields, descriptor_tag, descriptor_length and format_identifier. The values of all these three fields are fixed by the Blu-ray standard. Thus, in order to create HDMV_registration_descriptor(), we can simply assign these pre-defined values to its fields.

HDMV_copy_control_descriptor() is defined by Blu-ray and is designed to provide information about the copy protection of the program. DVB supports a similar descriptor called conditional_access_descriptor() which is also used for copy protection management. If conditional_access_descriptor() appears in DVB PMT, we can obtain the relevant information to generate Blu-ray HDMV_copy_control_descriptor(). Otherwise, if it is not present in DVB, we can assume that the program can be fully accessed, and this descriptor is then produced accordingly.

Generating HDMV_registration_descriptor() and AVC_timing_and_HRD_descriptor() is a more challenging task since the information in DVB-MHP PSI/SI tables is not enough to generate them.

HDMV_video_registration_descriptor() is supported by Blu-ray only and it must be present if the transport stream contains MPEG-2 or MPEG-4 AVC video. This descriptor provides detailed information of the present video by using three fields: video_format, frame_rate, and aspect_ratio [2]. Our task is to determine the values of these three fields.
using the transmitted DVB information. The information about frame_rate and aspect_ratio can be derived from certain DVB-MHP PSI/SI descriptors. However, the information about video_format which indicates the vertical resolution and the scanning method (interlaced or progressive) cannot be found in any DVB PSI/SI tables. Our analysis found out that this kind of information can only be found in the header information of the broadcasted video sequences. Since the information about frame_rate and aspect_ratio can be also obtained from the video header, our approach is to determine all these three fields using the video header instead of separately determining them from the video header and the DVB-MHP PSI/SI descriptors. There are several units in the header information of a video sequence, two of them are useful in our search for the desired information: sequence_header() and sequence_extension(). In DVB-MHP video TS packets, we first search for the following fields of sequence_header(): vertical_size_value, frame_rate_code and aspect_ratio_information[12]. If the value of aspect_ratio_information is 0x02 (meaning 4:3) or 0x03 (meaning 16:9), then it can be used directly for aspect_ratio in HDMV_video_registration_descriptor(). Otherwise, the aspect ratio of the video should be modified to 4:3 (with value 0x02) or to 16:9 (with value 0x03), and the value of this field should be changed accordingly. Regarding frame_rate_code, we can keep its value and apply it directly to frame_rate in the Blu-ray descriptor. After searching for sequence_header(), we should search for sequence_extension() in the TS packets. Within the sequence_extension() unit, the progressive_sequence field is useful for the generation. The value of video_format in HDMV_video_registration_descriptor() is determined jointly by progressive_sequence from sequence_extension() and vertical_size_value from sequence_header().
AVC\_timing\_and\_HRD\_descriptor() is defined only in the Blu-ray specification. Blu-ray PMT may contain this descriptor only when the Blu-ray transport stream carries an MPEG-4 AVC (H.264) video stream, with 3:2 pull-down information being sent by the Picture Timing Supplemental Enhancement Information (SEI) message [2]. In the DVB standard, there are no descriptors indicating the 3:2 pull-down information (used for modifying the frame rate). However, we can identify such information from the video sequence. In the header of a video sequence, there is a unit called picture\_coding\_extension(), within which the repeat\_first\_field field is located. If the value of repeat\_first\_field is set to 1 in a video sequence, then the 3:2 source is transmitted. Therefore, this descriptor is only generated if the value of repeat\_first\_field is 1.

It is challenging to efficiently search for the header information of a video sequence because the needed data are embedded in the TS packet payloads and have changeable locations. The fast search algorithm for such needed video information will be presented in Chapter 4.

3.5 Generating Blu-ray Selection Information Table Using a DVB-MHP stream

Table 3.4 shows Selection Information Table (SIT) [5]. SIT is intentionally designed for storage (such as recording a DTV program) and is used to provide a summary of Service Information. In DVB, SIT is not broadcasted from the transmission end. Instead, SIT is generated by the receiver-end device (e.g., a DVB set-top-box) if it is connected to a recording device (e.g., a VCR). In Blu-ray, however, SIT must be present in its transport stream. Hence, for the purpose of this study, SIT should be generated according to the relevant standards.
Syntax | Number of bits
---|---
selection_information_section() {
  table_id | 8
  section_syntax_indicator | 1
  DVB_reserved_future_use | 1
  ISO_reserved | 2
  section_length | 12
  DVB_reserved_future_use | 16
  ISO_reserved | 2
  version_number | 5
  current_next_indicator | 1
  section_number | 8
  last_section_number | 8
  DVB_reserved_for_future_use | 4
  transmission_info_loop_length | 12
  for(i=0;i<N;i++) {
    descriptor()
  }
  for(i=0;i<N;i++) {
    service_id | 16
    DVB_reserved_future_use | 1
    running_status | 3
    service_loop_length | 12
    for(j=0;j<N;j++) {
      descriptor()
    }
  }
  CRC_32 | 32
}

Table 3.4: Selection Information Table

In addition to assigning values to some of the fields based on the original standard [5], we should also generate SIT to conform to the constraints imposed by Blu-ray [2]. According to the Blu-ray standard, the value of service_id should be fixed to 0x0001 because this field is defined to have the same value as program_number of the current program. Furthermore, in Blu-ray, running_status should be set to 0, which indicates that the program is running.

Syntax | Number of bits
---|---
partial_transport_stream_descriptor() {
  descriptor_tag | 8
  descriptor_length | 8
  DVB_reserved_future_use | 2
  peak_rate | 22
  DVB_reserved_future_use | 2
  minimum_overall_smoothing_rate | 22
  DVB_reserved_future_use | 2
  maximum_overall_smoothing_buffer | 14
}

Table 3.5: Partial transport stream descriptor.
The Blu-ray standard also imposes restrictions on SIT descriptors. Only one descriptor, `partial_transport_stream_descriptor()` (see Table 3.5), is allowed in the Blu-ray system, and this descriptor must be present in the first "for" loop. No descriptors are allowed in the second "for" loop. Thus, the correct values must be found and assigned to the fields of this descriptor.

For `partial_transport_stream_descriptor()`, the `descriptor_tag` must be set to 0x63 [2]. Since there are 8 bytes immediately following the field `descriptor_length`, the value of the `descriptor_length` should be set to 8. The other fields including `peak_rate`, `minimum_overall_smoothing_rate`, and `maximum_overall_smoothing_buffer` are assigned based on the parameters of the Blu-ray system.

The `peak_rate` field denotes the maximum momentary TS packet rate, namely the size of the TS packet divided by the interval between the starting times of two consecutive TS packets [5]. From the Blu-ray specification [2], we know that:

\[
\text{Data rate of TS} \leq \frac{192}{188} \times 48 \times 10^6 \text{bits/sec.}
\]

Hence, the maximum momentary TS packet rate, i.e.

\[
\text{peak rate} = \frac{\text{packet size in bits}}{\text{time interval between two TS packets}}
\]

\[
= \frac{192 \times 8}{192 \times 8} = \frac{192}{188} \approx 50 \times 10^6 \text{bits/sec.}
\]

The `minimum_overall_smoothing_rate` field specifies the minimum smoothing buffer leak rate for the whole TS. According to the Blu-ray standard, the smallest leak rate is
$15 \times 10^6$ bits/second. Therefore, the minimum overall smoothing rate in SIT should be $15 \times 10^6$ bits/second.

The Blu-ray standard does not specify the maximum size of the smoothing buffer, so the value of the `maximum_overall_smoothing_buffer` field should be set to 0x3FFF, which means that the upper bound of the smoothing buffer size is undefined [5].

3.6 General Architecture of the PSI/SI Transcoder

![Diagram of PSI/SI Transcoder]

Figure 3.2: Partial transport stream descriptor.
From the above discussion, we are able to draw the conclusion that the information provided by DVB-MHP PSI and SI data is not sufficient for generating the corresponding Blu-ray PSI/SI data. In addition to DVB-MHP PSI/SI data, information which can be found in the DVB video stream is also needed to generate complete Blu-ray compatible PSI/SI. Figure 3.2 [22] describes the general architecture of the “PSI & SI transcoder”. The “video info extractor” module searches the DVB video stream for the useful information. The “PSI/SI extractor” module is used to pull out only the desired PSI/SI tables from the DVB-MHP PSI/SI, ignoring the tables that are not necessary for generating the Blu-ray PSI/SI data. In the “table modifier” module, the extracted PSI/SI tables are modified to conform to the Blu-ray standard. Finally, the “table generator” module processes the extracted video metadata and the modified PSI/SI data before combining them to generate the Blu-ray compliant PSI/SI data.

3.7 Conclusions

In this chapter, we have presented methods that generate the Blu-ray PSI/SI information using the incoming DVB-MHP stream. Our methods take advantage of the existing similarities between the Blu-ray and the DVB-MHP systems such that the least conversions are needed in the transcoding process (that makes the two systems compatible).

Our study has found out that only three PSI/SI tables, Program Association Table (PAT), Program Map Table (PMT), and Selection Information Table (SIT) are supported by the Blu-ray system. It is also found that only PAT and PMT in DVB-MHP are useful in generating the Blu-ray PSI/SI tables. Thus, our proposed scheme extracts only PAT and PMT from the receiving DVB-MHP PSI/SI tables. The other tables present in DVB-MHP are simply not used.
Our transcoding scheme generates Blu-ray PAT using the existing DVB-MHP PAT, and only basic fields in the table need to be changed. Since only one program is allowed in Blu-ray, the size of the generated PAT must be 16 bytes.

Our study found out that in order to generate Blu-ray PMT, the DVB-MHP PSI/SI data are not sufficient. Thus, information hidden in the DVB-MHP video stream is also needed. Our proposed scheme obtains the desired video information from two units in the header of a video sequence; these two units are \textit{sequence\_header()} and \textit{sequence\_extension()}.

The proposed scheme generates Blu-ray SIT according to relevant standards instead of using any DVB-MHP data. In addition to creating the basic fields and the descriptor based on the original PSI/SI standard, the constraints imposed by the Blu-ray standard are also considered.
4 System Efficiency Improvement and Transcoding Results

4.1 Introduction

In this Chapter, we propose three different schemes to make our transcoding system more efficient. The objectives are: i) to transcode the PSI/SI tables less frequently without affecting the repetition rate at which the Blu-ray system receives the generated tables; ii) to fast search for needed information (video headers) hidden in the video stream. Instead of searching every TS packet payload byte by byte, we use TS packet headers to find the payload where the video headers are located; and iii) to develop an efficient demultiplexing method for our transcoding system other than using the conventional demultiplexing approach. This chapter also shows the transcoded PSI/SI tables using the proposed schemes.

The rest of this chapter is organized as follows. The improved system architecture that transcodes the PSI/SI tables less frequently is described in Section 4.2. In Section 4.3, we develop a fast algorithm that searches for the desired information hidden in a video stream. An efficient demultiplexing method is proposed to enhance the efficiency for our transcoding system in Section 4.4. Section 4.5 shows the resulting tables generated by the optimized PSI/SI transcoder. The conclusions are presented in Section 4.6.

4.2 Improved System Architecture of the PSI/SI Transcoder

Due to the random access nature of broadcasting, PSI/SI tables are repeatedly transmitted in a periodical fashion in the DVB network. Despite the fact that Blu-ray is a packaged medium system, PSI/SI tables must also be present in its transport stream repeatedly at the same rate as that in DVB. Transcoding the tables every time they arrive is a
costly process and causes undesirable delay. Because of the fact that PAT, PMT and SIT are relatively static for a specific program in a certain time period, the use of a temporary storage that would store the PSI/SI tables right after they are transcoded can reduce the frequency of PSI/SI transcoding. Figure 4.1 describes the proposed structure of the “PSI & SI transcoder” with the storage module.

![Diagram of PSI & SI transcoder](image)

**Figure 4.1: Optimized architecture of the PSI & SI transcoder.**

The “PSI/SI content analyzer” in Figure 4.1 is used to check if the extracted PSI/SI table is updated. This analyzer sends a message to the communicator which decides on the
next step to go to. If the PSI/SI table is updated, then the "communicator" will send the extracted table to the "table modifier" module which modifies this extracted table according to the Blu-ray standard. After that, the modified table will be copied by the "table storage" module for later usage before it is sent to the output. On the other hand, if the extracted PSI/SI table remains the same as the previous one, the "communicator" will send a message to the "table storage" module which then outputs a pre-stored corresponding table without any transcoding process. Obviously, the "table storage" module makes the transcoding less frequently and creates free long intervals that can be used for other purposes, such as pre-transcoding PSI/SI tables for a different program.

The size of the storage buffer is small. In the worst case where there are up to 33 different video streams in one program, the maximum size of PMT is 1027 bytes [3]. Regarding PAT and SIT, their sizes are always fixed: 16 bytes for PAT and 28 bytes for SIT. In summary, the maximum buffer size needed is 1071 bytes, which is just a bit larger than 1KB.

4.3 Fast Algorithm that Searches for Information Hidden in the Video Stream

As explained in Chapter 3, DVB-MHP PSI and SI tables are not sufficient to generate the Blu-ray PSI/SI tables. The extra needed information should be extracted from the video stream. It is challenging to search for this type of information efficiently because the desired video data are hidden in the TS packet payloads and have changeable locations. Note that the needed information is used only for two PMT descriptors. To search for the needed information, we propose two algorithms, one for typical and another for non-typical DVB-MHP streams. These algorithms are explained as applied to
HDMV_video_registration_descriptor() and could be applied in the same way to AVC_timing_and_HRD_descriptor().

In Chapter 3, it is shown that the desired information could be found within two units in the header of a video sequence: sequence_header() and sequence_extension(). Each unit has a unique start code as an identifier in the bit stream (a video sequence is actually a bit stream). The start code for sequence_header() is 0x000001B3, and for sequence_extension() is 0x000001B5. The contents of sequence_header() and sequence_extension() follow their start codes immediately. Hence, a normal way to find the needed information in these two units is to first search all the payloads for the two start codes and then derive the needed contents following the start codes. Because there are many packets between two successive sequence_header()'s, a considerable number of payloads must be checked byte by byte (about 184 bytes for each TS packet) in order to find the next sequence_header(). Thus, the computational cost of this approach is high. Below we describe our proposed algorithms for searching the different types of the DVB-MHP transport streams.

4.3.1 Fast Search Algorithm for Typical DVB-MHP TS

Our proposed algorithm is a fast search algorithm of a hierarchical structure. It uses the TS packet header information to filter out packets not needed by our transcoding scheme. Two kinds of header information in TS packets can be used to identify the needed packets: the random access indicator or the payload unit start indicator.

The random access indicator indicates whether or not there is a random access point. If this field in a TS packet is set to 1, then the packet must carry information that aids the random access [3]. On the other hand, sequence_header() is used in a video stream to provide random access points in H.264 or MPEG-2 video. Thus, we propose to first check
the random access indicator in the TS packet header. If it is 0, then we skip that packet without checking any bytes in the payload. If it is 1, we search that packet’s payload for the start code of the `sequence_header()` which is immediately followed by the needed information. Since `sequence_extension()` is always located closely after `sequence_header()`, only a few operations are required to find `sequence_extension()`.

Similar to the random access indicator, the payload unit start indicator can also be used for the highest level search in the algorithm. If the payload unit start indicator is set to 1, it indicates that the current TS packet is the starting point of a PES. Because of the fact that the beginning of a video PES must contain a random access point which is indicated by `sequence_header()`, `sequence_header()` should be located at the starting point of a video PES. Hence, the above algorithm could also employ the payload unit start indicator to extract the needed TS packets.

This fast search algorithm can be applied to any typical DVB-MHP transport stream, which has both of the two following properties: i) the random access indicator is always set to 1 when there is a random access point; and ii) the payload unit start indicator is always set to 1 when there is a starting point of a PES. In practice, the typical DVB-MHP TS includes all the transport streams carrying H.264 encoded video and most of those carrying MPEG-2 encoded video.

Figure 4.2 shows for each of 8 programs (the horizontal axis) the normalized search time over 50 `sequence_header()`’s and their corresponding 50 `sequence_extension()`’s. The three search methods used are the hierarchical search using the random access indicator, the hierarchical search using the payload unit start indicator and the normal search using the start code only. 8 different programs (totally 3 different typical DVB-MHP transport streams)
are used in the evaluation. The search is designed to start only after a new PMT is obtained because the video information is only needed when generating a PMT. The time on the vertical axis is normalized by the time consumed by Program 8 using the random access indicator (the least time in the evaluation). We observe that the search using the start code of the sequence_header() consumes twice as much time as our proposed hierarchical search algorithm. It is also noted that the search using the random access indicator is always relatively faster than the one using the payload unit start indicator.

![Diagram](image)

**Figure 4.2:** Time comparison of the three different schemes that search for the information embedded in the video stream.

Figure 4.3 shows the average search time of the three methods for a different number of sequence_header(). Each point in the figure is the average search time for the 8 programs.
in Figure 4.2. The time is normalized by the average time used to search for 20 sequence headers. This figure further re-verifies that the hierarchy search algorithm saves 50% of the time compared to the straightforward method search for the start code. Moreover, we also observe that using the *random access indicator* in the search is always faster than using the *payload unit start indicator*. We hereby propose to use the *random access indicator* in our hierarchical search algorithm for typical DVB-MHP transport streams.

![Diagram](image)

Figure 4.3: The average search time using the three methods that search for different numbers of *sequence_header*.

### 3.4.2 Best Search Algorithm for Non-typical DVB-MHP TS

A non-typical DVB-MHP TS has one of the following two properties: i) the *random access indicator* is not always set to 1 even if there is a random access point in the current
packet; ii) the payload unit start indicator is not always set to 1 even if there is a starting point of a PES. Note that a non-typical DVB-MHP transport stream may only contain MPEG-2 encoded video. As explained, a TS carrying H.264 coded video is strictly defined by the DVB standard [13] and it is always a typical DVB-MHP TS.

In order for our search algorithm (described in the previous sub-section) to work in the presence of a non-typical transport stream, necessary modifications have to be made. Since in a non-typical TS the random access indicator or the payload unit start indicator may be sometimes set to 0 even if a sequence_header() is present in that packet payload, our hierarchical search algorithm may not be effective. This means that we have no clues as to which search algorithm (search using the random access indicator, the payload unit start indicator, or the start code only) should be applied when a TS is received. On the other hand, a multiplexer (encoder) processes its input stream in a certain pattern. We thus propose a modified algorithm that adds a “statistics gathering stage” where “enough” statistics about the presence of the indicators and start codes are gathered before a decision about the proper search method to be used is made. Below we describe the modified algorithm and how the algorithm decides when “enough” statistics are gathered.

The modified algorithm is as follows: start the search using the three methods (the start code and the two indicators). Stop the search when the transcoder receives 2510 video packets. If neither the random access indicator nor the payload unit start indicator is found, the “statistics gathering stage” is stopped and only the start code is then used to search the rest of the transport stream. On the other hand, if any indicators are found in the first 2510 video packets, the “statistics gathering stage” is continued. In this case, the “statistics gathering stage” is stopped when 12550 packets are received. At this point, we compare the
number of the *random access indicator* and the *payload unit start indicator* present within the 12550 packets (say, Tr and Tp respectively). If both Tr and Tp are greater than 4, then the *random access indicator* is used for future search. If Tr or Tp is less than 5 while the other is not, then the one which is greater than 4 is used for the future search. If neither Tr nor Tp is greater than 4, then the start code is used to search the coming packets. The reasons behind the strategy and the decision making in this algorithm are described below.

Two stop points are used in the above algorithm to stop the “statistics gathering stage”. At the point when 2510 video TS packets are received, a decision can be made as to whether searching using the start code is better than using the indicators; at the point when 12550 video TS packets are received, the statistics data are employed in choosing one of the three methods. The reasons behind choosing the number of TS video packets to be 2510 or 12550 are as follows. To aid random access to a video, a typical DVB-MHP TS should encode the *sequence_header()* at least once every 500 ms [13]. The bitrate of a DVB-MHP standard definition (SD) program is 3 – 5 Mbps. Assume that the average TS bitrate is 4 Mbps, and 90% of the TS are video packets. Thus we know that the *sequence_header()* is present at approximately every 1255 packets. That is to say, the *random access indicator* appears around every 1255 packets if it is a typical DVB-MHP TS with a SD video. Let's call 1255 packets “typical random access period”. From Figure 4.2 and Figure 4.3, the search time using the start code is around twice that using any one of the 2 indicators. Thus if the frequency of appearance of the *random access indicator* is lower than every $1255 \times 2 = 2510$ packets, we should search using the start code instead. Hence, the moment when 2510 video packets are received is chosen for the first stop point. The reason behind choosing 12550 (1255×10) packets for the second stop point is because in practice, the number of packets
which is equal to 10 times the “typical random access period” should provide how often the
two indicators are present in a TS.

The next thing is to choose a search method after each of the two stop points are
reached. At the first stop point (i.e., after the first 2510 video packets are received), if no
indicators are found, then we should stop the “statistics gathering stage” and use the start
code to search the rest of the TS. This is because the indicators appear at a frequency less
than once every 1225 packets. If we can find any indicators at the first stop point, then we
perform the “statistics gathering stage” until the second stop point is reached (i.e., after
12550 video packets are received). Based on our assumption, there should be 10 random
access points in the 12550 video packets. That is why we should search using the start code if
the number of appearance of the indicators is less than 5. Figure 4.2 and Figure 4.3 also show
that searching using the random access indicator is a bit faster than using the payload unit
start indicator. This is the reason behind the decision about choosing one of the two
indicators for the future search by comparing \( T_r \) and \( T_p \) in the modified algorithm.

The proposed algorithm can be adapted to a non-typical transport stream carrying high
definition (HD) MPEG-2 video. The only difference is the stop points. Since the average
bitrate of a normal HD program is around 20 Mbps, the first and the second stop points can
be set respectively at the moment when 12550 and 62750 video packets are received.

Figure 4.4 shows the search time for 9 programs (such news reports, interviews, and
movies) in typical and non-typical DVB-MHP streams with four different search methods.
The method using the random access indicator or the payload unit start indicator works well
for typical transport streams, but sometimes they fail to provide a fast search for non-typical
ones. On the other hand, our proposed algorithm guarantees that the search time is always the best possible search whether a typical or non-typical DVB-MHP stream is considered.

![Comparison of the schemes that search typical and non-typical DVB-MHP streams for needed information.](image)

**Figure 4.4: Comparison of the schemes that search typical and non-typical DVB-MHP streams for needed information.**

### 4.4 Efficient Demultiplexer for Our Transcoding System

In order to generate Blu-ray PSI/SI data, two types of data in a DVB-MHP transport stream must be demultiplexed: video data and PSI/SI data.

For demultiplexing DVB PSI/SI data, we use the conventional demultiplexing scheme presently used in broadcasting. The approach is to demultiplex PSI/SI data into PSI/SI table sections, which is similar to the process of demultiplexing video data into packetized
elementary streams (PESs). This is because it is better to make modifications directly on PSI/SI tables.

As for the video data, almost all broadcasting systems demultiplex the video part of a transport stream into packetized elementary streams or a video elementary stream (ES). Fully demultiplexing into PES’s or an ES requires the demultiplexer to analyze the TS header in detail and combine the relevant TS packets into PES’s or an ES. Such process is time-consuming. We thus propose to demultiplex the video part of the TS into TS packets instead. Demultiplexing into TS packets involves a simple process.

Below we analyze the complexity of both the normal demultiplex approach and our method.

The conventional demultiplexing process for a selected program contains three main steps:

i) Read the incoming byte in the transport stream until the value of this byte is 0x47 (the value 0x47 indicates this byte is the beginning of a 188-byte TS packet) [3], and then fetch the TS packet from the transport stream;

ii) Read the packet ID of this packet, and if this packet belongs to the selected program, then buffer this packet and go to step iii);

iii) Read the payload unit start indicator of this packet. If the indicator is 0, then combine the data of this packet with the previous ones to form a PES; if the indicator is 1, this packet should be the start of a new PES.

The computational complexity of each step is approximately the same as the others. As discussed, the PSI/SI part of our demultiplexing method can be regarded as a process that
involves the above three steps while demultiplexing video data using our method needs only the first two steps.

We consider a typical duration to construct a video PES. From Section IV (B), we know that for a typical SD DVB-MHP transport stream, the payload unit start indicator is present in a rate similar to that of the random access indicator which is once every 1225 packets (every 500 ms for a 4 Mbps TS). Thus, to form a typical PES, 1225 packets are needed. If we use the conventional approach to demultiplex the video data (and form a PES), this would require $3 \times 1255 = 3765$ steps. In contrast, only $2 \times 1255 = 2510$ steps are needed to demultiplex the video data using our proposed method. Next, we analyze the computational complexity of the demultiplexing of the PSI/SI data. According to the DVB standard, we can assume that PAT and PMT are transmitted every 100ms [5]. Practically speaking, we can also assume that one PAT needs one TS packet and one PMT needs two TS packets. So within 1225 TS packets, there are 30 (PAT and PMT) packets that need to be processed. For either the conventional or our proposed methods, it takes $3 \times 30 = 90$ steps to demultiplex the needed PSI/SI data. In summary, in a PES period, the conventional demultiplex approach takes $3765 + 90 = 3855$ steps to demultiplex the needed data while our proposed method takes only $2510 + 90 = 2600$ steps. Therefore, our demultiplexing method reduces the computational complexity by about 33% compared to the conventional approach.
This demultiplexing scheme reduces the computational complexity of the system not only in the process of the demultiplexing but also in that of the "header transcoder and remultiplexer" module (see Figure 4.5), which collects the outputs from the transcoders and forms a Blu-ray compliant transport stream. A remultiplexer performs inverse operations of its corresponding demultiplexer. Since our proposed demultiplexer does not de-packetize the video data, this module avoids re-packetizing them. Compared to using a conventional remultiplexer, our approach reduces the computational complexity by around 33%.

4.5 Results Generated from the PSI/SI Transcoder

The PSI/SI transcoder and the demultiplexer have been implemented using the above proposed schemes. In order to evaluate the results generated from the transcoder, the received and the transcoded PSI/SI table bitstreams are visualized, and contents between the
PSI/SI tables transmitted by DVB-MHP and the generated Blu-ray PSI/SI tables are compared. The analysis of the generated Program Association Table (PAT), Program Map Table (PMT), and Selection Information Table (SIT) are provided in the following subsections.

4.5.1 Generated Program Association Table

A DVB-MHP transport stream with 11 programs is used in the validation. 8 of the 11 programs contain video and audio while the other three contain audio only. The original DVB-MHP Program Association Table bitstream is visualized in the form of a table and the result is shown in Figure 4.6. Since only one program is allowed in PAT, the program whose program_number value is 0x0D4C and whose PMT PID value is 0x0104 is chosen for generating the Blu-ray PAT.

The DVB-MHP PAT bitstream is transcoded, and the Blu-ray PAT bitstream is generated. After changing the generated PAT from a bitstream to a table format, the contents of the table are shown in Figure 4.7. Compared to Figure 4.6, it is seen that the Blu-ray PAT contains information on only one program (only one set of program_number and program_map_PID found in the program loop). The values of program_number and program_map_PID have been modified to 0x0001 and 0x0100 respectively. The value of section_length has been changed to 13, which is the number of bytes of the PAT section immediately after this section_length field. Other fields in the generated PAT remain the same as those in the original DVB-MHP. In summary, the generated PAT maintains the necessary information from the DVB-MHP PAT and conforms to the Blu-ray standard.
Figure 4.6: DVB-MHP Program Associate Table containing 11 programs.

Figure 4.7: Generated Blu-ray Program Association Table containing only one program with the program_number 0x0D4C.
4.5.2 Generated Program Map Table

The DVB-MHP Program Map Table corresponding to the program (with the program_number value 0x0D4C in DVB-MHP) chosen by the Blu-ray PAT in Section 4.5.1 is used in the validation. The original PMT bitstream is displayed in a table format. Figure 4.8 shows only a part of the DVB-MHP PMT which is useful for our evaluation. It is noted that there are two iterations in the second loop, which means there are two types of applications. The value of the stream_type 0x02 indicates MPEG-2 video and 0x04 denotes ISO/IEC 13818-3 audio. Due to the purpose of our study, only the video part in this program is transcoded and the audio part is simply ignored.
Figure 4.8: DVB-MHP Program Map Table corresponding to the program with the
program_number 0x0D4C.

Figure 4.9: Generated Blu-ray Program Map Table.
The Blu-ray PMT bitstream is visualized and the resulting table is shown in Figure 4.9. Compared to Figure 4.8, the values of program_number and PCR_PID have been changed to 0x0001 and 0x1001 respectively. The program_info_length field has been modified to 12 in that the size of the first loop is 12 bytes. Two Blu-ray specific descriptors, HDMV_registration_descriptor() and HDMV_copy_control_descriptor() are contained in the first loop. On the other hand, only one single iteration is contained in the second loop because only one video stream is associated with this program (the audio stream ignored). The value of stream_type remains the same as the one in Figure 4.8 while elementary_PID has been changed to 0x1011 from 0x0203. The ES_info_length field is changed to 10 because the length of the descriptor(s) in this iteration is 10 bytes. Figure 4.9 shows that only one descriptor, HDMV_video_registration_descriptor() is contained in the first iteration of the second loop. To generate the video_format, the frame_rate and the aspect_ratio fields in this descriptor, we need information from the video stream. The value 0x02 of the field video_format means that the video of this program is interlaced and the vertical resolution is 576. The frame_rate field with the value 0x03 indicates that the frame rate of this video is 25 fps. The aspect_ratio value of 0x02 means the display aspect ratio of the video is 4:3. Overall, the generated PMT has extracted the desired information from the DVB-MHP data, and the contents of the generated PMT satisfy the Blu-ray standard.

4.5.3 Generated Selection Information Table

Selection Information Table is generated by assigning values to its fields and descriptors instead of it being transcoded from any DVB-MHP PSI/SI table. The generated Blu-ray SIT bitstream is exhibited in a table shown in Figure 4.10. The size of the SIT is fixed to 28 bytes. Thus, the value of section_length, which indicates the number of bytes in
the current table immediately followed by this field, is set to 25 (the first three fields of SIT occupy 3 bytes). It is noted that the table contains only one descriptor partial_transport_stream_descriptor() whose descriptor_tag (as a descriptor identifier) is 0x063. In the second loop, the service_id field has been changed to 0x0001 which is the same as the value of program_number in the generated PMT. The value of running_status has been fixed to 0x00 according to the Blu-ray standard. Since the second loop in Blu-ray SIT does not support any descriptor, the value of service_loop_length which indicates the size of the descriptors in the second loop has been set to 0. It can be seen that the Blu-ray SIT has been generated according to the discussion in Section 3.5 and conforms to the Blu-ray specification.

![Figure 4.10: Generated Blu-ray Selection Information Table.](image)
4.6 Conclusions

In this chapter, we have developed three schemes that make our transcoding system more efficient and implemented the PSI/SI transcoder based on the proposed schemes.

The first proposed scheme enables the incoming PSI/SI tables to be transcoded less frequently by improving the system architecture of the PSI/SI transcoder. Only the updated DVB-MHP PSI/SI tables are transcoded. In order to satisfy the repetition rate of the PSI/SI tables in the Blu-ray system, a storage module has been introduced at the end of the transcoder. The additional storage is as small as 1071 bytes, just a bit larger than 1KB.

Our second scheme is a fast algorithm that hierarchically searches for the desired information hidden in the video stream. Instead of searching every TS packet payload, the TS packet header information (i.e., the random access indicator and the payload unit start indicator) is used to find the payloads which contain the needed information. The proposed algorithm saves around 50\% of the time taken for searching typical DVB-MHP transport streams compared to the straightforward method that searches using the start codes. To search non-typical DVB-MHP TS, a modified algorithm is developed and guarantees that the search time is always the best possible search.

In our third scheme, an efficient demultiplexing method is proposed for our transcoding system. In our method, the video part of a transport stream is demultiplexed into TS packets and not packetized elementary streams. This results in saving almost 33\% operations compared to the conventional approach. Considering the whole transcoding system, our solution also reduces another 33\% operations in the re-multiplexing process.

This chapter also shows the Blu-ray PSI/SI tables generated from our implemented PSI/SI transcoder that deployed all the proposed schemes. It has been verified that the
resulting tables conform to the Blu-ray standard while they maintain the essential information of the original DVB-MHP data.
5 Conclusions and Future Work

5.1 Conclusions

In this thesis, a transcoder that efficiently generates Blu-ray PSI/SI tables using the broadcast DVB-MHP data is developed. Three main tasks accomplished are: 1) Analyzing the compatibility between the DVB-MHP and the Blu-ray systems of the system information data; 2) Proposing methods that transcode DVB-MHP data to Blu-ray PSI/SI tables with the least possible modifications; and 3) Developing schemes that make the transcoding system more efficient.

In Chapter 3, the differences in the system information between the DVB-MHP and the Blu-ray standards are discussed. Taking advantage of the existing similarities of the two systems, methods are proposed to generate the Blu-ray PAT, PMT and SIT with the least possible modifications of the incoming DVB-MHP streams. It is pointed out that DVB-MHP PSI/SI data are not sufficient to produce Blu-ray PSI/SI tables, and the information contained in the DVB-MHP video stream is also needed.

In Chapter 4, three schemes that enhance the efficiency of our transcoding system are developed and a PSI/SI transcoder that is based on the proposed schemes is implemented. Our first scheme improves the system architecture of the PSI/SI transcoder so that it transcodes the incoming tables the least frequently. The introduced storage module only consumes just over 1KB. The second scheme is a fast algorithm that hierarchically searches for needed information hidden in the video stream. The proposed algorithm enjoys a 50% time reduction compared to the normal search method. Based on this fast search method, we present a modified algorithm which always guarantees a best possible search for the desired
information even if a non-typical DVB-MHP stream is broadcasted. In our third scheme, an optimal demultiplexing method which enables a 33% time saving compared to the traditional demultiplex scheme is proposed. Considering the whole transcoding system, our demultiplexing solution also reduces 33% of the operations in the re-multiplexing process. The Blu-ray PSI/SI tables generated from our implemented PSI/SI transcoder have been verified to conform to the Blu-ray standard while they maintain essential information from original DVB-MHP data.

5.2 Future Work

The proposed PSI/SI transcoder generates only video-related information. If audio applications are considered, analysis of the differences in audio-related information between the two standards and transcoding approaches to realizing the compatibility could be done in a very similar way to this study. The search algorithm for video data could be modified to find the relation between the audio sequence headers and the TS packet headers.

Considering the whole transcoding system, the differences between DVB-MHP and Blu-ray do not only exist in the system information but also in the video coding data, the interactive application data and the transport stream packet headers. Transcoding approaches for those three components could be designed to realize full compatibility of the two systems. If all the transcoders are developed, the complete transcoding system could then be optimized as a whole.
Bibliography


# Appendix A – List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>ATSC</td>
<td>Advanced Television Systems Committee</td>
</tr>
<tr>
<td>BD-J</td>
<td>Blu-ray Java-based interactivity</td>
</tr>
<tr>
<td>DSM-CC</td>
<td>Digital Storage Media Command and Control</td>
</tr>
<tr>
<td>DTV</td>
<td>digital television</td>
</tr>
<tr>
<td>DVB</td>
<td>Digital Video Broadcast</td>
</tr>
<tr>
<td>ES</td>
<td>elementary stream</td>
</tr>
<tr>
<td>GEM</td>
<td>Globally Executable MHP</td>
</tr>
<tr>
<td>HD</td>
<td>high definition</td>
</tr>
<tr>
<td>HDMV</td>
<td>high definition movie (mode in Blu-ray)</td>
</tr>
<tr>
<td>ISDB</td>
<td>Integrated Services Digital Broadcasting</td>
</tr>
<tr>
<td>iTV</td>
<td>interactive television</td>
</tr>
<tr>
<td>MHP</td>
<td>Multimedia Home Platform</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MPTS</td>
<td>multiple-program transport stream</td>
</tr>
<tr>
<td>OCAP</td>
<td>Open Cable Application Platform</td>
</tr>
<tr>
<td>PAT</td>
<td>Program Association Table</td>
</tr>
<tr>
<td>PES</td>
<td>packetized elementary stream</td>
</tr>
<tr>
<td>PID</td>
<td>packet ID</td>
</tr>
<tr>
<td>PMT</td>
<td>Program Map Table</td>
</tr>
<tr>
<td>PSI</td>
<td>Program Specific Information</td>
</tr>
<tr>
<td>SI</td>
<td>Service Information</td>
</tr>
<tr>
<td>SIT</td>
<td>Selection Information Table</td>
</tr>
<tr>
<td>TS</td>
<td>transport stream</td>
</tr>
</tbody>
</table>
VOD

video on demand