VIDEO STREAMING OVER COOPERATIVE WIRELESS NETWORKS

by

Yi Liu
B.Sc., Simon Fraser University, Burnaby, BC, Canada, 2007

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Fall 2009

APPROVAL

Name: Yi Liu

Degree: Master of Science

Title of Thesis: Video Streaming over Cooperative Wireless Networks

Examining Committee: Dr. Thomas C. Shermer,
Chair

____

Dr. Mohamed Hefeeda, Senior Supervisor

____

Dr. Qianping Gu, Supervisor

____

Dr. Wo-Shun Luk, Examiner

Date Approved: December 10, 2009

Declaration of Partial Copyright Licence

Abstract

We study the problem of broadcasting video streams over a Wireless Metropolitan Area Network (WMAN) to many mobile devices. We propose a cooperative network in which several elected mobile devices share received video data over a Wireless Local Area Network (WLAN). The proposed system significantly reduces the energy consumption and the channel switching delay concurrently. We design a distributed leader election algorithm for the cooperative system and analytically show that the proposed system outperforms current systems in terms of energy consumption and channel switching delay. Our experimental results from a real mobile video streaming testbed show that the proposed cooperative system is promising because it achieves high energy saving, significantly reduces channel switching delay and uniformly distributes load on all mobile devices. Furthermore, we complement our empirical evaluation with a trace driven simulator to rigorously show the viability of the proposed cooperative system.

Keywords: mobile TV; video streaming; energy saving; channel switching delay; cooperative network; DVB-H

To my family with love.

"All religions, arts and sciences are branches of the same tree. All these aspirations are directed toward ennobling man's life, lifting it from the sphere of mere physical existence and leading the individual towards freedom."

— Albert Einstein

Acknowledgments

I am deeply indebted to my senior supervisor, Dr. Mohamed Hefeeda, for his continuous support, encouragement and guidance through my research. He patiently show me the way of doing research, never accepting less than my best efforts. Mohamed provided valuable insights, and a lot of help during my graduate career. This thesis would not have been possible without him.

I would like to thank my supervisor Dr. Qianping Gu and my thesis examiner Dr. Wo-Shun Luk for being on my committee and reviewing this thesis. I would like to thank Dr. Thomas C. Shermer for taking the time to chair my thesis defense. I would also like to extend my gratitude to the faculty and staff in the school of computing science at SFU, particularly Dr. Richard Zhang for what I have learned in his courses.

I would like to thank all my friends for their continuous support, specially: Shu Li who encouraged me to keep going when I thought I have given it all I can; Lan Yao who inspired me to strive beyond the ordinary, and gave me the confidence I needed to make it through life; and Siqi Fu who gave me lots of encourage and support right before the thesis defense.

I would like to thank all my colleagues at Network Systems Lab for their help throughout my academic work in SFU, specially: Cheng-Hsin Hsu who helped me with academic writing and mobile TV testbed setup. I am especially grateful to Cong Ly, Yuanbin Shen, Fei Gao, Ahmed Hamza and Shabnam Mirshokraie.

Last but certainly not least, I would like to thank my parents and grandma for their unquestioning love, care and support. I would not have made it without them. Their expectations have always propelled me to realize my full potential. This thesis is dedicated to them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we provide a brief introduction about video streaming in Wireless Metropolitan Area Networks (WMANs). Then, we introduce the energy saving and channel switching delay problems we address in this thesis and summarize our contributions. The organization of this thesis is given at the end of the chapter.

1.1 Introduction

Mobile TV, which allows users to watch TV programs on their mobile devices, is now starting to become a reality with the recent launch of 3G services and the significant consumer demand for multimedia content. Even though music and gaming are currently dominant sources of revenue for mobile devices, the mobile TV service is catching up quickly. Market research anticipates the number of mobile TV subscribers is expected to grow to as high as 500 million by 2011 with expected revenue of 20 billion Euros [4]. Mobile TV trials or commercial services have already been deployed in more than 40 countries [15]. For example, the trial DVB-H based mobile TV service in Moscow broadcast 9 TV channels [17] and in Iraq, nationwide free-to-air DVB-H based mobile TV service has been offered [16]. Furthermore, recent mobile devices in the market are equipped with fast processors, high resolution displays and multiple network interfaces so that they are powerful enough to natively receive, decode, and display videos sent over networks. The capabilities of the mobile devices as well as the users' desire to access multimedia content anywhere and at anytime have created a strong demand for mobile TV services. Supporting mobile TV natively is also an attractive selling point for mobile device manufacturers. In Fig. 1.1, we

Figure 1.1: Cell phones that support mobile TV natively.

Phone Model	**CPU**	**Screen**	**Network**
Nokia N96 [57]	266MHz	2.8-inch	DVB-H, WLAN 802.11g/b
LG KU950 [45]	Proprietary	2.4-inch	DVB-H, WLAN 802.11g/b
Nokia N92 [56]	220MHz	2.8-inch	DVB-H, WLAN 802.11g/b
Gygabyte gSmart [34]	520MHz	2.6-inch	DVB-H, T-DMB, WLAN 802.11g/b
Eten Glofiish [22]	533MHz	2.8-inch	DVB-H, T-DMB, WLAN 802.11g/b

Table 1.1: Technical specifications for cell phones that support mobile TV.

list several commonly used cell phones that support mobile TV natively and their technical specifications are listed in Table. 1.1.

Videos can be sent to mobile devices either over cellular networks or dedicated broadcast networks. Cellular networks cannot concurrently support a large number of mobile devices, as they are designed for unicast and have limited bandwidth. Supporting mobile TV services over cellular network lowers the overall capacity of the network for all users [33]. In addition, expanding a cellular network for higher capacity such as Multimedia Broadcast Multicast Service (MBMS) [35] to accommodate more users is expensive and cannot support many video channels due to limited bandwidth. In contrast, in a Wireless Metropolitan Area Network (WMAN), a base station can *broadcast* videos to support a large number of mobile devices within its coverage range. Streaming videos over WMANs is promising as it supports more mobile devices with lower cost on network infrastructure. We consider video streaming over WMANs in this work.

Streaming videos to mobile devices is challenging, because mobile devices are battery powered and have stringent energy constraints. Therefore, energy saving on mobile devices is important since lower energy consumption leads to longer watch time and increases the user satisfaction. Current video streaming systems use WMANs to broadcast videos, in which each mobile device *independently* receives and decodes the video data from the base station. That is, every bit of data is used at most once before being discarded. We explore the potential of better utilizing the received video data by sharing them among several mobile devices that watch the same video stream. More precisely, we propose a new video streaming system, in which several mobile devices receive and cache video streams broadcast in a WMAN *and* share the received data over a Wireless Local Area Network (WLAN). Such sharing is possible as WLANs are very popular and many access points are readily available. Furthermore, our proposed cooperative system supports the ad-hoc mode of WLAN in case there is no access point available. Another important factor in mobile TV network is channel switching delay. This is the time users need to wait when they switch the TV channels. Since normally users are used to flip through different TV channels quickly before they decide their target TV programs, a long channel switching delay is annoying to users and this bad user experience may turn users away from mobile TV services.

1.2 Problem Statement and Thesis Contributions

Our goal is to concurrently improve the energy saving metric as well as the channel switching delay of video streaming. This thesis studies this problem by using a WMAN and WLAN cooperative network.

1.2.1 Problem Statement

Current video streaming over WMAN uses the *time slicing* mechanism to save energy. That is to say, each TV channel is broadcast in bursts at a bit rate much higher than the encoding rate of that TV channel. Mobile devices can then receive a burst of traffic and turn off their radio frequency (RF) circuits until the next burst. One example of WMANs using time slicing is the Digital Video Broadcast-Handheld (DVB-H) standard. However, the time slicing mechanism increases the channel switching delay. Hsu and Hefeeda [38] show the existence of a tradeoff between the energy saving and channel switching delay because of the timing slicing scheme: increasing energy saving results in an increase in the channel switching delay, which may be annoying to users. This is because mobile devices save more energy by receiving data in larger, but less frequent, bursts. And to receive these bursts, users must wait for longer periods on average. Both the energy saving and channel switching delay are important to the mobile TV users. Instead of only improving one of these two parameters, we study the energy saving and channel switching delay reduction problem in video streaming and design a new strategy to improve both of them concurrently. More precisely, this problem can be stated as follows.

Problem 1 *Design an efficient algorithm to coordinate the operation of N mobile devices equipped with both WMAN and WLAN interfaces and form a cooperative network in order to optimize both the energy saving metric and channel switching delay. The algorithm should determine the role of each mobile device and manage the transmission of the video data over the WMAN and WLAN cooperative networks.*

1.2.2 Thesis Contributions

We propose a solution to the energy saving and channel switching delay problems of video steaming in WMAN and WLAN cooperative networks. Particularly, our contributions [37, 47, 48] can be summarized as follows:

- We design a distributed algorithm to elect a mobile device that is called *on-duty* to receive the data from WMAN and relays it over WLAN to other mobile devices [48]. Since other mobile devices can receive data bursts from this on-duty mobile devices over the WLAN, which consumes less energy compared to receiving bursts over the WMAN, the energy consumption is *reduced*. In addition, the delay of mobile devices switching to new video streams can be significantly reduced (almost *eliminated*), because these mobile devices can request an immediate burst transmission from the on-duty device using unicast.

- We provide a quantitative analysis of our algorithm as well as empirical experiments [47]. Our experimental results from a real testbed indicate that the proposed cooperative system: (i) achieves as high as 70% of energy saving gain, (ii) outperforms current systems with only three (or more) cooperative mobile devices, (iii) reduces channel switching delay by up to 98%, (iv) is robust against device failures and quickly reacts to network dynamics, and (v) uniformly distributes load on all mobile devices.

- We design and develop a real testbed for our empirical experiments [37].[1] This testbed is open-source. Our design for the testbed is modular with well-defined interfaces between the components. Therefore, different hardware/software components can be updated with minimal effects on the others, which can be useful for future generations of mobile TV systems.

- By sharing the data packets within the WMAN and WLAN cooperative network, we can improve both energy saving metric and channel switching delay. The trade-off between these two performance metrics in WMAN only network is successfully eliminated in our proposed cooperative network.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we briefly provide a background on mobile video streaming and summarize the related works in the literature. In Chapter 3, we present an overview of the proposed cooperative network. We then present our algorithm and analysis. We describe the setup of our testbed and present our experimental results in Chapter 4. We present simulation results in Chapter 5. We then conclude the thesis and outline potential extension of this work in Chapter 6.

[1] This is a joint work with my colleagues Cheng-Hsin Hsu and Cong Ly under the supervision of Dr. Mohamed Hefeeda.

Chapter 2

Background and Related Work

In this chapter, we briefly review the techniques used in mobile TV area. We then summarize previous works that are related to our work.

2.1 Background

2.1.1 Mobile TV Standards

There are several standards for Mobile TV networks, including MediaFLO [31], Terrestrial-Digital Multimedia Broadcasting (T-DMB) [7], Integrated Services Digital Broadcasting-Terrestrial (ISDB-T) [67], China Mobile Multimedia Broadcasting (CMMB) [8] and Digital Video Broadcast-Handheld (DVB-H) [25, 28, 43].

T-DMB [7] is an extension for the Digital Audio Broadcast (DAB) standard [23] by adding video broadcast services to the high-quality audio services offered by DAB. The development of T-DMB is supported by the South Korean government, and T-DMB is the first commercial mobile video broadcast service. ISDB-T [67] is a digital video broadcast standard defined in Japan for both fixed and mobile TV receivers. ISDB-T divides its bandwidth into 13 segments, where 12 of these channels are used for fixed HDTV and the remaining one is for mobile TV service. MediaFLO [31] is a video broadcast system developed by Qualcomm and the FLO forum [30]. MediaFLO is designed for video broadcast services to mobile receivers. However, it is not an open standard and the design details are not available to the public. CMMB [8] is a mobile television and multimedia standard developed and specified in China. It is designed for use on small screen devices. By using

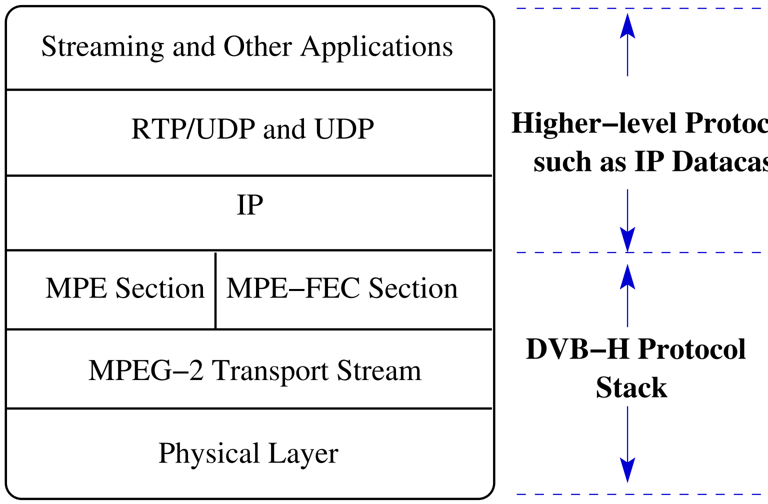

Figure 2.1: The DVB-H Protocol Stack.

both satellite and terrestrial cables, CMMB can provide 25 video and 30 radio channels with some additional data channels. DVB-H [28, 43] is an extension to the Digital Video Broadcast-Terrestrial (DVB-T) [44] standard tailored for mobile devices and it is an open international standard [25]. Among the above dedicated broadcast networks, only DVB-H and MediaFLO try to minimize the energy consumption of mobile devices by periodically turning their RF circuits off [14, 31]. We address DVB-H standard in this thesis.

2.1.2 Overview of the DVB-H Standard

The DVB-H standard defines protocols below the network layer and uses IP as the interface with the higher-layer protocols such as UDP and RTP. Another standard called IP Datacast [43, 53] complement DVB-H by defining a set of higher-level protocols for a complete end-to-end solution. DVB-H uses MPEG-2 as transport streams to carry data. The IP packets are encapsulated in these transport streams. Since both video stream data and singling messages are IP-based, many IETF-defined protocols are adopted by the IP Datacast standard for a complete mobile TV service. Above the network layer, IP Datacast uses the UDP protocol in its transport layer. There are two protocols on top of the UDP protocol for different content delivery: real-time streaming protocol (RTP) for multimedia streaming traffic and file delivery over unidirectional transport protocol (FLUTE) for file and meta-data transfer. IP Datacast also defines an XML-based Electronic Service Guide (ESG) that delivers TV channel information to users and TV channel initialization parameters to mobile TV receivers. More precisely, ESG uses the Session Description Protocol (SDP) to describe initialization parameters. The complete protocol stack [26] of video broadcasting over DVB-H networks is illustrated in Fig. 2.1.

DVB-H uses a physical layer compatible to the DVB-T, which employs Orthogonal Frequency Division Multiplexing (OFDM) modulation. DVB-H encapsulates IP packets using Multi-Protocol Encapsulation (MPE) sections to form MPEG-2 transport streams. Thus, data from a specific TV channel form a sequence of MPEs. In addition, broadcasting videos suffers from high data error rates caused by fading, shadowing, and interference. Since maintaining a large number of reverse connections to the base station for automatic repeat request (ARQ) is not feasible, DVB-H employs forward error correction (FEC) to mitigate the high data error rates. Thus MPEs are FEC-protected before transmitted over the wireless medium.

The two mandatory features for DVB-H are *time slicing* and *forward error correction*. We briefly describe them as follows, more details are provided in Section 4.1.1:

Time Slicing: To save energy of mobile devices, MPEs belonging to a given TV channel are transmitted in *bursts* with a bit rate much higher than the encoding rate of that TV channel. Thus, mobile devices can receive a burst of traffic and then turn off their RF circuits till the next burst. This is called *time slicing*. Time slicing can also enable smooth and seamless frequency handover.

Forward Error Correction: To improve the carrier-to-noise (C/N) ratio and Doppler performance in mobile channels and to improve the tolerance to impulse interference, the DVB-H standard applies Reed-Solomon (R-S) code and time interleaving in its link layer to protect IP packets in each burst transmission. This C/N improvement is reported to be equivalent to the antenna gain given by the spatial diversity [25].

2.1.3 WLAN Overview

Wireless Local Area Network (WLAN) is developed by the Institute of Electrical and Electronics (IEEE). It is similar to LAN but allows the connection of computers without any wires. WLAN provides increased mobility, flexibility, ease of use and also reduces the cost of setting up and maintenance. The most popular wireless network these days is based on the IEEE 802.11 standard. The IEEE 802.11 standard defines the first two layers of the network stack, namely Physical Layer and Link Layer for a LAN with wireless connectivity.

Name	**Operation Frequency**	**Max Data Transfer Rate**	**Release Date**
802.11n	5 GHz and 2.4 GHz	144 Mbps	September 2009
802.11g	2.4 GHz	54 Mbps	June 2003
802.11b	2.4 GHz	11 Mbps	October 1999
802.11a	5 GHz	54 Mbps	October 1999

Table 2.1: Major characteristics of the IEEE 802.11 family.

The most popular IEEE 802.11 standards these days are IEEE 802.11 b/g/n. We list the major characteristics of the IEEE 802.11 family in Table. 2.1 [40].

Even though an IEEE 802.11a can provide a maximum throughput of up to 54Mbps by using an Orthogonal Frequency Division Multiplexing (OFDM) based air interface in physical layer, its effective overall range is less than that of IEEE 802.11b/g networks. The maximum throughput of IEEE 802.11b is only 11 Mbps which is only about 1/5 of that in IEEE 802.11g. In addition, IEEE 802.11g is backward compatible with IEEE 802.11b. Our analysis and experiments are all based on IEEE 802.11g.

The IEEE 802.11g WLAN has two basic modes of operation. One is called infrastructure mode and the other is called ad-hoc mode. The main difference between the infrastructure mode and the ad-hoc mode is that there is no access point (AP) in the ad-hoc mode and the mobile devices in ad-hoc mode are connected via multi-hop instead of single-hop. Therefore devices in ad-hoc mode need to play the role of a router as well as a host. In one-hop clustering, every device is at most one-hop distance away from the central leader device called master device. As shown in Fig. 2.2 all network traffic is routed through the master device, and all other devices are no more than one hop from the master device. The master device is responsible to manage all its slave devices, which is similar to an AP in WLAN infrastructure mode.

2.1.4 Time slicing delay in WMAN

The nature of time slicing scheme introduces an additional delay when users switch channels. This delay is called time slicing delay. Since the video is broadcast in bursts from the base station. When users tune into a new video channel, they are more likely to encounter the sleep period of a WMAN cycle when no data is broadcast. As shown in Fig. 2.3, there are two possibilities when users switch channels. The first case is that the initial switching time falls between $t1$ and $t2$. The second case is that the initial switching time falls between $t2$

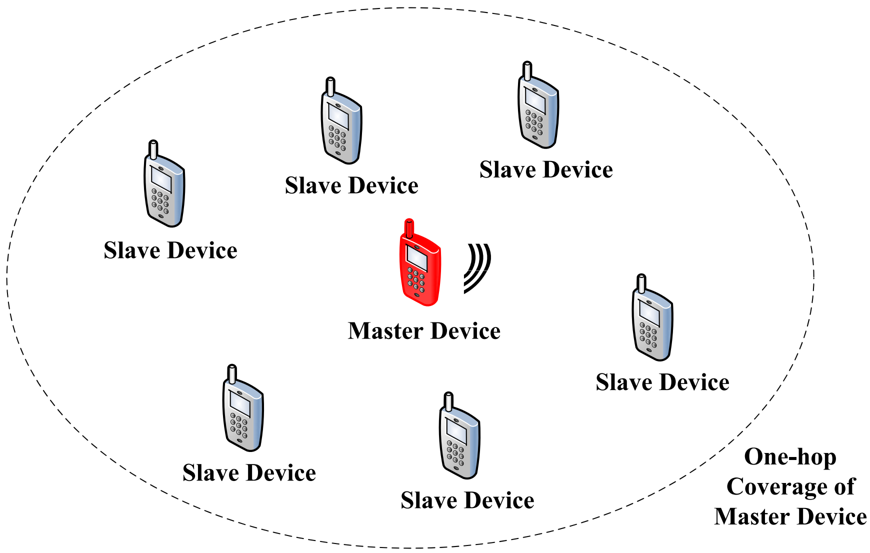

Figure 2.2: The proposed MANET.

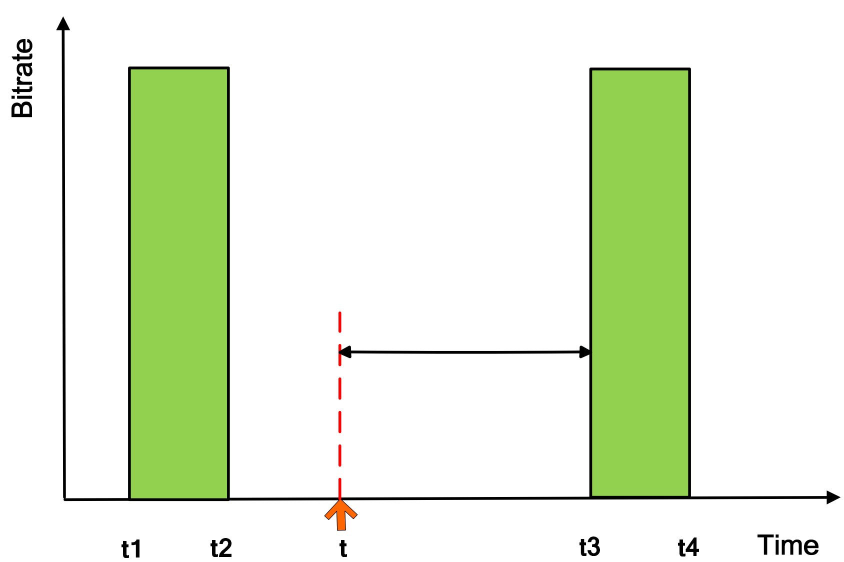

Figure 2.3: Time slicing delay in WMAN.

and $t3$. Obviously the possibility of the second case is much higher because the sleep period is much longer than the burst period. Depending on the switch time t, the user may have to wait up to the whole time slicing sleep period (from $t2$ to $t3$).

2.2 Related Work

Some previous works explore the benefits of using cooperative network between WMANs and WLANs. Pei et al. [60] develop an analytical model in the cooperative network to evaluate the call block probability and throughput performance. They also propose two cooperative load-balancing strategies in order to improve the resource utilization. Niyato and Hossain [54] propose a hierarchical bandwidth management scheme using game theory to achieve fairness and efficiency of radio resource allocation. The optimal burst size for WMAN and WLAN connections can be determined by their proposed two-level hierarchical model. Pan et al. [59] propose a quantitative evaluation of the effect of integrating WiMAX and WiFi networks, as compared to WiMAX or WiFi only systems. Their cooperative network is used to evaluate the VoIP traffic as well as the corresponding resource management scheme on WiFi to WiMAX handover. However, to the best of our knowledge, none of the previous works addresses the video broadcast over the cooperative WMAN and WLAN networks.

A number of works address the energy saving problems in WMANs only network. Due to the time slicing nature of WMANs, energy saving of mobile devices can be done by burst scheduling on the base station side. Yang et al. [71] address the effectiveness of the time slicing technique for given burst schedules and calculate the corresponding energy saving level. These two works do not solve the server side burst scheduling problem. Balaguer et al. [3] propose an energy saving strategy by not receiving some MPE-FEC (Multiprotocol Encapsulation - Forward Error Correction) sections once the sections received are good enough to reconstruct the data. Dropping some MPE-FEC sections can help mobile devices turn off their RF circuits earlier for additional energy saving. Hefeeda and Hsu [36] prove that the general burst scheduling problem with arbitrary bit rates of TV channels is NP-complete. They then propose a practical simplification of the general problem, which allows the bit rates of the TV channels to have power of 2 increments. The running time of this simplification is $O(S \log S)$ where S is the total number of TV channels. Hsu and Hefeeda [38] address the general scheduling problem where each TV channel can take any arbitrary bit rate, which enables finer-grained optimization by providing higher bit rate flexibility. All the above works are orthogonal to ours, because they assume all mobile devices receive video data from their WMAN interfaces only, while our work considers sharing video data over a WLAN in order to save energy *beyond* what can be saved by time slicing.

Some previous works address the channel switching delay problem in WMANs only network. Channel switching delay refers to the time period between when a user switches to a new video stream until his/her mobile device starts rendering that stream. Channel switching delay consists of several components [62, 65], including time slicing delay, frame refresh delay, decoder buffering delay and reception delay. Time slicing delay is a dominant component among them, which is the time period between the mobile device tunes to a new video stream until the first burst of that video stream arrives. We only consider time slicing delay in this work, and assume other delay components are constant as they are not impacted by the proposed cooperative system. Rezaei et al. [64, 65] propose to transmit the intra-coded frames in the beginning of the bursts to shorten frame refresh delays. Rezaei et al. [63] also propose an empirical method to determine the operational video quality under different combination of frame refresh delay and variable bit rate buffering delay. Hsu and Hefeeda [39] propose to control the channel switching delay by simulcasting each video over two streams. In contrast, our work uses unicast over a WLAN to almost *eliminate* the channel switching delay.

A number of works address the handover problem in WMANs only network. Ollikainen and Peng [58] propose a vertical handover approach for DVB-H networks, where each mobile device maintains a unicast connection over a Universal Mobile Telecommunications Systems (UMTS) network. Video streams are not only carried by the DVB-H network but also relayed by the cellular network, which enables a mobile TV device to quickly handoff to the unicast connection when the DVB-H signal is degraded. This auxiliary network can also be used for reducing channel switching delays. Unfortunately, maintaining a unicast connection for each mobile device for video traffic imposes tremendous load on cellular networks and streaming servers, and therefore is not scalable.

Some previous works address the multicast routing problem in WLAN ad-hoc mode. Bommaiah et al. [6] use virtual mesh links to establish a bidirectional shared multicast tree. The tree nodes only include the group senders and receivers. And unicast tunnels are used as the tree links. Each group has at least one core node to detect group members and setup the tree by flooding messages. Only core nodes incur processing and storage overheads while all other nodes do not need to support the multicast protocol. When network topology changes, as long as routes between tree nodes exist via mesh links, the tree structure does not need to be changed. Wu et al. [70] establish a shared delivery tree among all participating nodes. It does not require a separate unicast routing protocol. Each node in the network is assigned a multicast session ID. The node with the smallest multicast session ID becomes the root of the tree such that a delivery tree can join up the nodes participating in the multicast session. By utilizing the multicast session ID, nodes can adapt rapidly to changes in link connectivity. Broken links can be detected by a beaconing mechanism. Rapid repairs to broken links are performed locally without the need for any central controlling node. Both of these two works are complementary to our work.

Chapter 3

Proposed Cooperative System

In this chapter, we first present an overview of our WMAN and WLAN cooperative network. Then we present the detailed operations of the proposed distributed algorithm. We then analytically analyze the performance of the proposed algorithm from several angels.

3.1 Overview

We consider a video broadcast system with multiple mobile devices as receivers, where each mobile device joins a WMAN and a WLAN as illustrated in Fig. 3.1.

The base station of this WMAN concurrently broadcasts multiple video streams to all mobile devices. These video streams are sent in *bursts* to save energy. Burst transmission, however, increases the channel switching delay. As shown in Fig. 3.2(a) during the sleep time of a TV channel, no data is transmitted such that a new joiner of that channel needs to wait for a longer time to get the next burst data. This increases the channel switching delay for the new joiner. While the WMAN covers a much larger area compared to the WLAN, receiving video data from the WMAN may consume more energy than receiving from the WLAN. This is because a WLAN covers fewer mobile devices, and thus is less sensitive to path loss, shadowing, and interference among mobile devices. Therefore, broadcasting video streams over the WLAN achieves higher transmission rates than over the WMAN. As shown in Fig. 3.2(b), transmitting the same video stream over the WLANs at higher rates leads to shorter transmission time, and allows mobile devices to turn off the wireless receivers for longer time, thus reduces energy consumption.

As shown in Fig. 3.1, we propose to form a *cooperative* network among all mobile devices

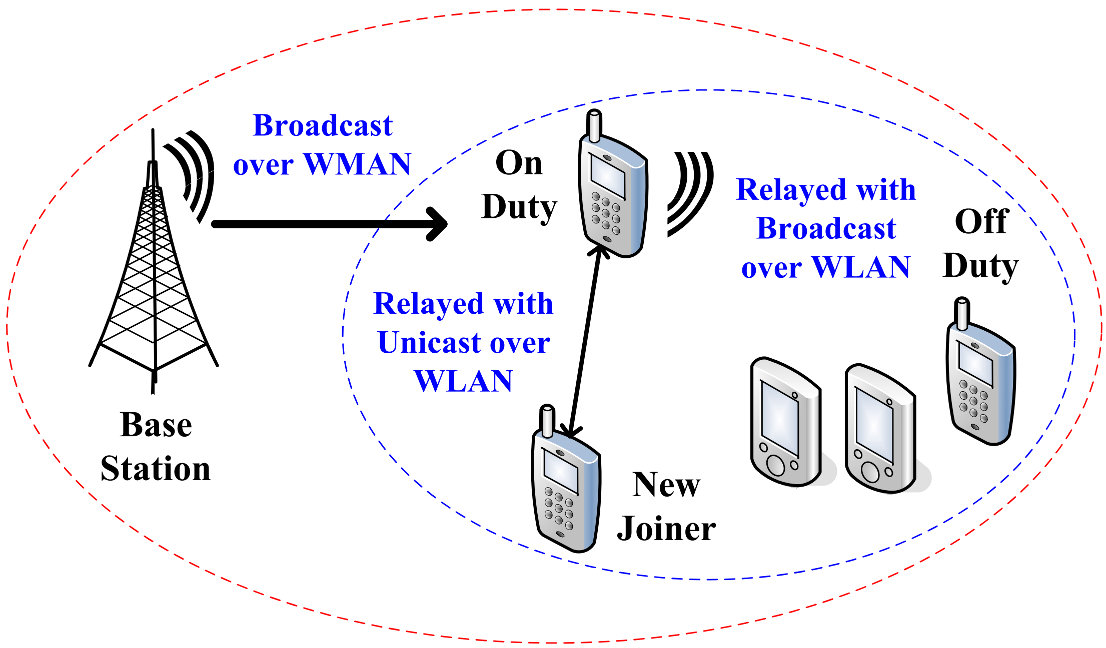

Figure 3.1: The proposed cooperative WMAN and WLAN video broadcast system. Mobile devices form a cooperative network over a WLAN. A device is elected to be on-duty on behalf of the group, which receives the video data from the base station over the WMAN then relays this data to other mobile devices over the much faster WLAN. A new joiner can contact the on-duty device to immediately receive video data without waiting for the next burst from the base station.

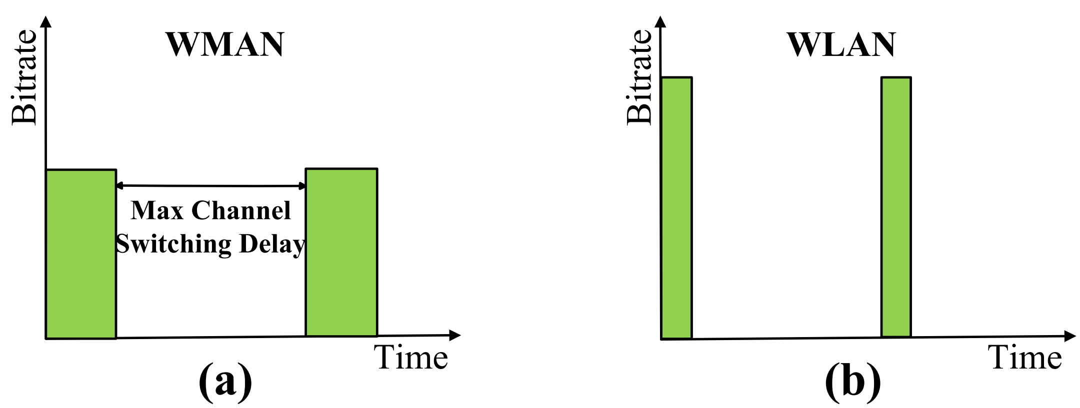

Figure 3.2: Burst Transmission over (a) WMAN and (b) WLAN.

that are viewing the same video stream. We let N be the size of the cooperative network. We then run a leadership protocol among these N mobile devices to elect one device that is *on-duty*. The on-duty device receives data bursts from the WMAN base station, and relays the data bursts to other mobile devices via WLAN. K *backups* devices are also elected, which monitor the status of the on-duty device and initiate a switch-over if the on-duty device fails or leaves the cooperative network. All other $N - K - 1$ devices are *off-duty* mobile devices, which receive the data bursts over the WLAN from the on-duty device. Last, the on-duty device offers the most recent burst to mobile devices that join this cooperative network. This allows new mobile devices to start playing out faster, as they do not need to wait for the next broadcast burst in WMAN.

Both the infrastructure mode and ad-hoc mode in WLAN are supported in our cooperative system. Many access points are readily available these days in places such as universities, airports, restaurants, and coffee shops. In addition, when no APs are available or are not freely accessible, the ad-hoc mode is used. For example, mobile users watching video streams while traveling on a bus or train could form a mobile ad-hoc network (MANET) to benefit from the proposed system to save energy and reduce channel switching delay. The support of the ad-hoc mode significantly lowers down the requirement of using our cooperative system. Additional operations such as unique IP assignment and multicast routing setup are required to form the cooperative group in MANET due to the lack of access point. We will discuss these extra operations in Section 3.3.

Our proposed cooperative system can apply to any WMAN standard that uses the time slicing technique to achieve energy savings on mobile devices. Some examples of WMANs using time slicing include the Digital Video Broadcast-Handheld (DVB-H) [28,43], MedioFLO [31] and Advanced Television Systems Committee-Mobile/Handheld (ATSC-M/H) [2] standards.

3.2 Operations of the Cooperative Network

In this section, we present the details of the operations needed to realize the proposed WMAN and WLAN cooperative system. These operations are employed in both the infrastructure and ad-hoc modes of the WLAN. Operations specific for the ad-hoc mode of the WLAN are presented in Section 3.3.

Election Algorithm

We propose a distributed algorithm to choose on-duty and backup devices. Each mobile device in our proposed cooperative system maintains a *contribution list* $C[n]$, which contains the contribution value of each device in the cooperative group. The contribution list is updated in each WMAN cycle by the mobile device. A cycle starts with the beginning of a burst and ends with the start of the next burst, which is in the order of few seconds. The contribution value is defined as the total amount of data that has been relayed over the WLAN. Therefore, only the contribution value of the on-duty device is updated in any cycle. This also means that the contribution lists on all devices will be the same, because they are all updated at the same time based on the reception of a burst of data from the on-duty device. In order to elect the on-duty and K backup devices, devices in the cooperative group sort their contribution lists. The device with the smallest contribution value is elected as the on-duty device. This device notifies others by broadcasting an ON-DUTY message. Moreover, among the remaining devices, K devices with the smallest contribution values are elected as the backup devices and each broadcasts a BACKUP message.

Since only the on-duty device has its contribution value updated in one WMAN cycle, all devices in the group only need to update one entry in their contribution lists. The on-duty device can easily update its contribution list. For off-duty devices, they receive the data from the on-duty device such that they update their contribution list entries based on the amount of data they actually received. For K backup devices, even though they do not receive the data from the on-duty device, they receive the data from WMAN base station which is the same as the on-duty device. K backup devices can calculate the size of the data received from WMAN to update the contribution value of the on-duty device in their contribution lists because the data received from WMAN is the same as the data relayed over WLAN.

This election algorithm is totally distributed and very simple; hence it can be implemented in practice. The algorithm also maintains the election truthfulness. A device that lies about its contribution value can be detected by other devices in the group since the contribution value is calculated independently by each device. The lier device has no control of contribution lists $C[n]$ of other devices. And the lier device once detected is kicked out from the cooperative group. In addition, the contribution list contains two other fields: join time and MAC address of each device. The join time is the *first* time a device participated in the cooperative group. In case that multiple devices have the same lowest contribution value, the device with the earliest join time will be chosen as the on-duty device, because the device that joined earlier has benefited from the cooperative network more than other devices. This scheme also does not penalize or discourage new joiners by making them on-duty right after they join. In case multiple devices have the same contribution value and join time, their MAC addresses are used to break the tie, since the MAC addresses are globally unique. The MAC address also provides another useful feature: it prevents devices from the whitewashing type of cheating. That is, a cheating device can participate in the cooperative group and save energy by receiving data from the on-duty device. When it comes its turn to serve as on-duty, it leaves the group. Then the cheating device joins a little latter with a new ID and a recent join time. Using the MAC addresses as IDs and the earliest join time a device appears in the cooperative group solve this cheating problem.

We note that the proposed algorithm provides a *practical* protection level against cheating that is sufficient for the proposed cooperative video streaming system over mobile devices with limited resources. This algorithm, however, may not be strong enough for other cooperative systems with more powerful devices. For example, a malicious device may spoof its MAC address and use a different (fake) address each time it joins the system. This MAC address spoofing may not be easily performed on mobile devices (e.g. cell phones), because most of them have light weight and secure operating systems (e.g. Symbian). In addition, the spoofing process and the continuous changing of the MAC address may impose high energy consumption on the malicious device, which may equal to or even exceed the energy spent in the legitimate cooperative process. Thus, a malicious device may be better off cooperating with other devices.

The proposed algorithm is also very efficient in terms of computation and communication. There is no computation in the algorithm except the periodic update and sorting of the contribution list at each device, which happens once in each WMAN time slicing cycle (few seconds). Also since the number of devices in the cooperative group is small (at most tens of devices), sorting the contribution list composes negligible CPU load on mobile devices. For communication overhead, there is only one message broadcast from the on-duty device to the cooperative group, and up to K messages from the backup devices. The messages are short as they only contain a single field indicating the message type, e.g., ON-DUTY or BACKUP. The pseudo code of our algorithm is presented in Fig. 3.3.

DeviceRole (K)

```
compare my contribution value C_i in the contribution list C[n]
if (my C_i is the smallest) {
  serve as on-duty device in next WMAN cycle
  broadcast on-duty message
  update my entry in my C[n]
}
elseif (my C_i falls into the least K ) {
  serve as backup device in next WMAN cycle
  broadcast backup message
  update my C[n] entry for on-duty device
}
else {
  serve as off-duty device in next WMAN cycle
  listen to on-duty message
  update my C[n] entry for on-duty device
}
```

Figure 3.3: Device Election Algorithm.

Receiving and Relaying Bursts

After one on-duty and K backup devices are elected, the cooperative system starts to transmit data. The on-duty and backup devices start to work slightly earlier before the WMAN burst period to make sure the cooperative group can be reliably organized such that devices can successfully receive the data from the WMAN base station. This is done by setting a timeout timer to receive the ON-DUTY messages announced by the current on-duty device. The timer for the ith backup device ($1 \leq i \leq k$) is set as $i \times \delta$, where δ is the timeout period. In the worst case, all on-duty and backup devices fail, then all off-duty devices need to turn on their interfaces to receive from the WMAN directly. This is achieved by setting the total preparation period to organize the cooperative group before the start of the WMAN burst to be $\Delta = (K + 1)\delta$. Therefore, if this period passes and none of the on-duty or K backup devices announces an ON-DUTY message, all devices will use their WMAN interfaces to obtain the video data. Therefore, by using this simple scheme, we can make sure that there is no data lost in our cooperative system even in presence of device failures. This is important to ensure high quality and continuous video playback.

The on-duty device turns off its WMAN interface after receiving the WMAN burst data and starts to send data packets via WLAN. After sending, the on-duty device switches to WLAN idle state to monitor potential new joiner requests. If there exists a new joiner during this period, the on-duty device forwards a fraction of the data packets to the new joiner based on the exact join time. Thus, the new joiner no longer needs to wait until the next burst period from the WMAN base station. Due to the high speed of the WLAN, the data packet exchange is fast and the channel switching delay is significantly reduced. The on-duty device finishes its duty after a WMAN time slicing cycle.

In addition, K backup devices are used to maintain a fail-safe mechanism. They receive data packets from the WMAN directly but do not forward the data via the WLAN. Their main task is to monitor the on-duty device status. If the on-duty device fails, one of the backup devices will take over the on-duty role and will start to forward data to all off-duty devices. For the remaining off-duty devices, they do not need to turn on their WMAN interfaces and only need to turn on their WLAN interfaces to receive data from the on-duty device. From the whole system point of view, for a particular video program, instead of all N devices turn on their WMAN interfaces, only $1 + K$ devices need to turn on the WMAN interfaces in our cooperative network. Therefore, the more users concurrently watching that video program, the more energy we can save in our cooperative network.

Handling Network Dynamics

Mobile devices may join, leave, or fail at any time during the cooperation period. Our cooperative system handles these dynamics as follows.

Device Joining. Instead of waiting for the next broadcast data burst sent from the WMAN, a newly joined mobile device can send a JOIN message to the on-duty mobile device. The on-duty mobile device, upon receiving the JOIN message, updates its contribution list by adding a new entry for the new joiner. Then the on-duty device transmits the *most recent* video data burst together with the latest contribution list information of the whole cooperative group to the newly joined mobile device using unicast. This allows the new joiner to start playing out immediately after joining the cooperative network. Thus reducing the channel switching delay. By receiving the most recent contribution list of the whole group, the new joiner can quickly take part in the election of the cooperative group in the following WMAN cycle. The on-duty device also notifies all existing members of the group such that they can all add an entry for the new joiner device. There are only three messages involved in the new join procedure. One message sent from the new joiner and received by the on-duty via unicast, another message replied from the on-duty and received by the new joiner via unicast and the third message broadcast from the on-duty to all devices.

Device Leaving. If an off-duty or backup mobile device gracefully leaves the network (i.e, not suddenly fails), a LEAVE message is sent to the on-duty device such that in the next cycle to elect on-duty and backup devices, the current leaving device will no longer be included. When the on-duty device receives the LEAVE message, it deletes the leaving device entry in its contribution list and broadcasts this leave information to all members in the cooperative group. All devices will thus know about the leaving event and will update their contribution lists. There are two messages involved in this procedure. One message sent from the leaving device and another message broadcast by the on-duty device. If the on-duty mobile device leaves the network, a LEAVE message is broadcast to the whole group such that all devices remove the leaving on-duty device entry from their contribution lists. The backup device with the least contribution value among all backup devices takes over and acts as the new on-duty device. It will then notify all group members by sending an ON-DUTY message. There are two messages involved in this procedure. One message broadcast to the whole group by the leaving on-duty device and another message broadcast by the new on-duty device.

Device Failing. In the proposed cooperative system, the on-duty device periodically broadcasts small ON-DUTY messages when it is not sending actual video data. This message is frequently broadcast during the WMAN cycle. The backup devices check the status of the on-duty device by monitoring the ON-DUTY messages. Whenever the on-duty device fails, the device with the least contribution value among all backup devices takes over and acts as the new on-duty device. Since backup devices receive and cache all video data bursts, the video streaming service is not interrupted by the failure of the on-duty device. The new on-duty device will inform the whole group to remove the failed on-duty device entry in their contribution lists. If an off-duty or backup device encounters failure, no message is sent. But this is not harmful to the cooperative system and the failed devices can be detected in future on-duty election periods. This is because if the failed devices are elected as on-duty or backup devices but do not send any messages as expected, they are treated as failed and their corresponding entries are automatically removed from the contribution lists of all devices in the group. This failure detection process is effective yet simple. Thus it is suitable for the proposed cooperative system in which devices are quite dynamic and failures are common.

Handling Time Synchronization

Mobile devices need to know the start of bursts such that they can properly schedule their wake up timers. The time synchronization problem is solved by the nature of the WMAN time slicing. The WMAN burst data packets are made of MPEG-2 Transport Stream (TS). With time slicing, MPEG-2 TS packets are sent in short bursts at the full bitrate of the MPEG-2 TS. In order for a receiver to wake up on time to receive the next burst data, the header of the TS packets contains a time parameter determining the time offset between the start time of the current burst and the next burst. Any data packets received on the receiver side contain such time offset information to wake up the receiver reliably in the next burst period. In our WMAN and WLAN cooperative system, even though off-duty mobile devices do not receive the relative time parameter directly from WMAN, this time parameter is forwarded by the on-duty mobile device via the WLAN to all off-duty mobile devices. This enables all mobile devices in the cooperative system to get the time parameter to determine the next wake up time in order to elect on-duty and backup mobile devices for WMAN reception. Therefore, the proposed cooperative system enables mobile devices to properly and efficient receive data busts on time. Our system does not employ any clock synchronization algorithms, which impose significant overhead especially if fine-grain clock synchronization is needed.

Initializing Cooperative Group

Whenever a new device comes to the cooperative network, it first listens to a default multicast IP address to determine if any on-duty device already exists. If there already exists an on-duty device, the new joiner contacts the on-duty device directly to receive data bursts and the group contribution list. If nothing heard after a period of time, the new joiner itself becomes the on-duty device and sends packets to the common multicast IP address to guide future coming devices. If several devices join the cooperative network at the same time, since they all have zero contribution value, the device with the smallest MAC address takes the on-duty role for the first WMAN cycle. Again, this group initialization scheme is simple and practically sufficient for our proposed cooperative video streaming system over mobile devices. More sophisticated, and much more expensive, protocols would have been needed if we were to consider malicious devices with powerful resources. For example, authentication schemes may be needed to only allow legitimate devices. Also encryption of some fields such as burst start time would prevent malicious mobile devices from free riding by passively listening on the multicast group without contributing anything back to the group. Encryption and decryption of a few fields (only few bytes) do not impose too much overhead. There is no need to encrypt the video data itself.

3.3 Additional Operations in WLAN Ad-hoc Mode

Additional operations are required in the ad-hoc mode to form the cooperative group. Since there is no AP available, devices form a mobile ad hoc network (MANET). We design a strategy to form a proper group in order to share the data received from WMAN. We note that if the mobile devices already form a MANET to support other distributed applications, our cooperative video streaming system can easily use this MANET and there will be no need to duplicate the operations described below. In most cases, however, the MANET does not exist. Thus we propose the following simple operations to support our proposed cooperative system in an efficient manner.

Assigning Unique IP Addresses. Different from using a Dynamic Host Configuration Protocol (DHCP) in the infrastructure mode, mobile devices in ad-hoc mode need a different way to get unique IP addresses to form a cooperative group. As every mobile device may move or leave the network at any time, a central DHCP server is not suitable in this kind of network. It is better to share the IP assignment work instead of overusing some particular mobile devices. Mobile devices watching the same video program form a cooperative group. Within each group, a master is needed to assign unique IP address to the new joiner, recycle IP address from the leaving device and manage data routing. We use the on-duty mobile device discussed above to serve as the MANET master device in order to handle the IP management task. We develop an IP management strategy in MANET to organize the cooperative group. Fig. 3.4 summaries this strategy.

As the figure shows, whenever a new device joins the MANET, it first listens to a default common multicast IP address to determine if any master device exists within its own short coverage range. If there already exists a master device, the new joiner becomes the slave device of this master and receives an IP address from it. If nothing heard after a period of time, the new joiner becomes the master device and sends packets to the common multicast IP to guide future coming slave devices. The master device can randomly choose an IP for itself from a block of IP addresses and generate a group ID based on its MAC address. Since the MAC address of each device is unique, this group ID is unique. After the master device starts to serve on MANET, it handles the IP management tasks within its coverage range. Whenever there is an IP request from a new joiner device, it replies with a packet containing the assigned IP, group ID, and the master device IP.

If there are no more IP addresses available, the master device will broadcast an IP verification message which contains both the master device IP and the group ID to all its slave devices. All alive slave devices with the same group ID, upon receiving this IP verification message, reply to the master device with their IP addresses to confirm their working status. The IP addresses of the devices that do not respond are collected back by the master device for future joiners.

Handling Movements and Failures. If a slave device leaves the group due to movement, it will send a NOTIFY message to the master device such that the master device can take back its IP address for future usage. If the master device leaves the group, it will send an ASSIGN message to one of the K backup devices in the group to take over the master role and work as master in the future. Since the backup device already has the current IP

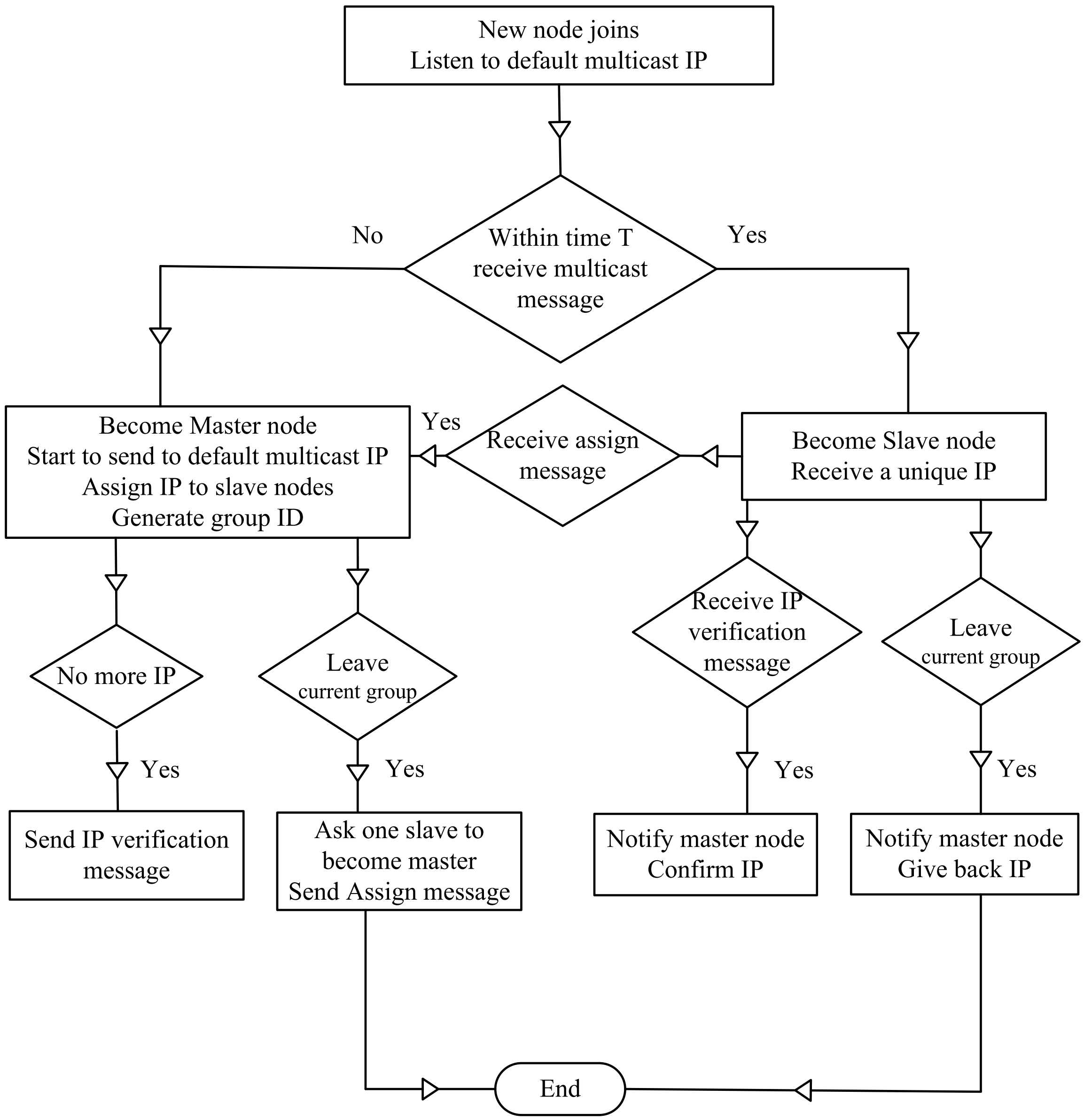

Figure 3.4: Address Management in the Ad-hoc Mode of the Proposed Cooperative System.

Parameter	**Description**
P_m^r	Power spent on WMAN receive
P_m^i	Power spent on WMAN idle
P_l^r	Power spent on WLAN receive
P_l^s	Power spent on WLAN send
P_l^i	Power spent on WLAN idle
T_m^b	WMAN receive period
T_m^i	WMAN sleep period
T_m^o	WMAN receive overhead
T_l^b	WLAN transmit period
T_l^i	WLAN sleep period
Δ	Cooperative group organization period

Table 3.1: Symbols used in the energy consumption analysis.

usage information, the selected backup device can seamlessly take over the work.

If the master device fails, no IP can be assigned to the new joiner. In order to address this issue, K backup devices periodically check the default multicast address to detect the status of the master device. If no multicast message is detected, one of the backup devices will take over the master role. Since no information is passed from the failed master device to the current one. The new master broadcasts an IP verification message to all existing slave devices to get the current IP address assignment.

Routing in MANET. Routing in MANET is a complex problem because the multicast tree structures that determine the routing table entries in MANET are fragile and experience lots of changes due to device movements. However, in our cooperative system, the multicast routing setup is easier as we only consider devices with low mobility *relative* to each other. For example, mobile devices on a bus are relatively static to each other while the bus is moving. The on-duty device can easily maintain a routing table after IP management. And due to the mobile devices in our MANET do not move much, some existing multicast protocols such as Protocol Independent Multicast (PIM) [10] and Distance Vector Multicast Routing Protocol (DVMRP) [9] can be applied in our cooperative network.

3.4 Algorithm Analysis

In this section, we analytically analyze the energy consumption of the proposed cooperative system and compare it against that of the current system. We use values obtained

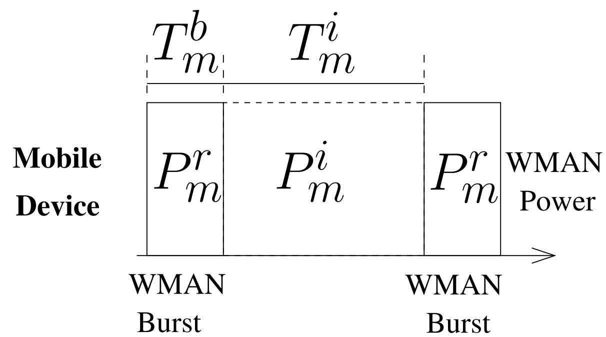

Figure 3.5: Burst transmission in current WMAN networks.

from actual devices to numerically analyze the derived equations under different network conditions. Table 3.1 contains all the symbols used in the analysis.

3.4.1 Analysis of Energy Consumption

Energy Consumption of Current Systems. We first study the energy consumption of mobile devices in current video broadcast systems, in which video streams are sent in bursts over the WMAN. Fig. 3.5 shows two bursts of the same video stream broadcast over a WMAN to mobile devices. Each mobile device receives a WMAN burst with length T_m^b sec, and then puts its WMAN interface into idle mode for T_m^i sec before reaching the next data burst. The WMAN interface has a receiving power consumption of P_m^r, and an idle power consumption of P_m^i. We note that the activation of the WMAN interface is not instantaneous because the WMAN interface needs to turn on its radio frequency (RF) circuit a little bit earlier to search for and tune into the broadcast frequency. We let T_m^0 sec be the WMAN burst overhead duration. Even though no data is transmitted during this period, the WMAN interface has to be on to get ready for the coming data. We compute the energy consumption of all mobile devices in the current system between two WMAN bursts as:

$$E_e = N\left(P_m^r(T_m^b + T_m^0) + P_m^i T_m^i\right) \text{ Joule.} \tag{3.1}$$

Energy Consumption of the Proposed Cooperative System. We next study the energy consumption of mobile devices in the proposed cooperative system, where each mobile device has a WMAN interface as well as a WLAN interface. Fig. 3.6 shows the video data transmission over the WMAN and WLAN. Notice that the main difference between the proposed cooperative system and the current system is that the on-duty mobile device

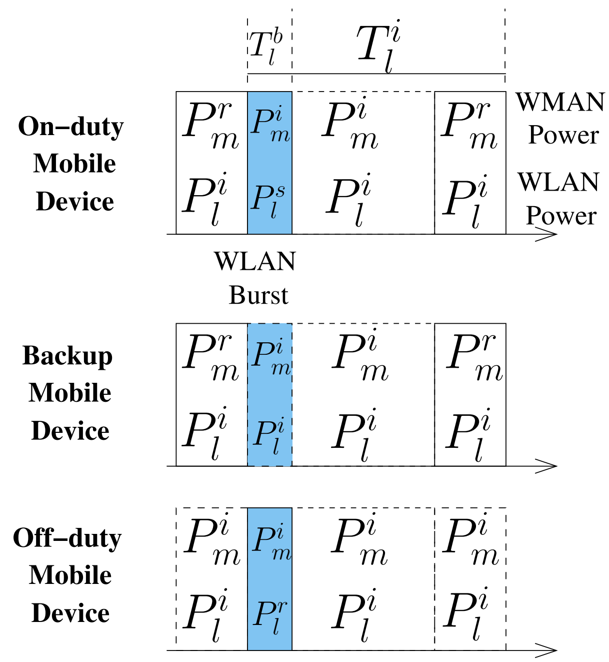

Figure 3.6: Burst transmission in the proposed cooperative WMAN and WLAN networks.

in the former relays the video data burst received from the WMAN over the WLAN to the off-duty mobile devices. These relayed bursts are shaded in Fig. 3.6. We let T_l^b sec be the burst time period and T_l^i sec be the idle time period in the WLAN. The WLAN interface has receiving power consumption of P_l^r, sending power consumption of P_l^s, and idle power consumption of P_l^i. In WMAN only network, all receivers experience the WMAN overhead T_m^0 sec. However, in the cooperative network, only the selected on-duty and backup devices need to spend energy on WMAN receiving preparation, which improves the energy saving metric. We also note that in our cooperative network, extra energy needs to be spent on the WLAN interfaces to organize the cooperative group a little bit earlier before the WMAN burst. As discussed in Section 3.2, we use Δ sec to denote this period. Using Fig. 3.6, we compute the energy consumption of on-duty, backup, and off-duty mobile devices between two WMAN bursts as follows:

An on-duty mobile device consumes:

$$P_m^r(T_m^b + T_m^0) + P_m^i T_m^i + P_l^s(T_l^b + \Delta) + P_l^i T_l^i \text{ Joule}$$

A backup mobile device consumes:

$$P_m^r(T_m^b + T_m^0) + P_m^i T_m^i + P_l^i T_l^b + P_l^i T_l^i + P_l^r \Delta \text{ Joule}$$

An off-duty mobile device consumes:

$$P_m^i T_m^b + P_m^i T_m^i + P_l^r T_l^b + P_l^i T_l^i \text{ Joule}$$

Then, we write the total energy consumption of the proposed system as:

$$\begin{aligned} E_c = & \Big(P_m^r(T_m^b + T_m^0) + P_m^i T_m^i + P_l^s(T_l^b + \Delta) + P_l^i T_l^i\Big) + \\ & \Big(P_m^r(T_m^b + T_m^0) + P_m^i T_m^i + P_l^i T_l^b + P_l^i T_l^i + P_l^r \Delta\Big) K + \\ & \Big(P_m^i T_m^b + P_m^i T_m^i + P_l^r T_l^b + P_l^i T_l^i\Big)\Big(N - 1 - K\Big) \text{ Joule.} \end{aligned} \tag{3.2}$$

Comparison of Energy Consumption. We derive the sufficient condition for the proposed cooperative system to outperform current systems. Observe that, since the energy consumption of the on-duty device is higher than that of the off-duty device, more off-duty devices lead to lower energy consumption. Therefore, we let N' be the minimum number of mobile devices for the proposed system to outperform current systems. Combining Eqs. (3.1) and (3.2), and rearranging the inequality, we get N' as:

$$N' = \left\lceil \frac{(P_l^s - P_l^r + KP_l^i - KP_l^r)T_l^b + (P_l^s + KP_l^r)\Delta + (K+1)(P_m^r T_m^b - P_m^i T_m^b + P_m^r T_m^0)}{(P_m^r - P_m^i)T_m^b + P_m^r T_m^0 - P_l^r T_l^b - P_l^i T_l^i} \right\rceil \tag{3.3}$$

Keeping N' small is important, otherwise the proposed cooperative system may lead to worse performance than current systems *if* there are very few mobile devices in the cooperative network.

Note that, in the above analysis, we did not consider the new joiner event. Since when a new joiner joins the cooperative group is a random event. Whenever a new joiner comes to the cooperative group, the on-duty device needs to spend extra energy to send the data to the new joiner. However, since the WMAN cycle is typically short (normally around 1 to 2 seconds), the possibility of a large number of new joiners come to the cooperative group within a particular short WMAN cycle is very low. Even if many new joiners come to the cooperative group, the on-duty device spends lots of extra energy to guide the new joiners, the contribution value of this on-duty device also increases a lot so that this on-duty device can stay as off-duty device for a much longer period due to its high contribution value accumulated in this particular WMAN cycle.

Description	Sample value
WMAN receive	400 mW
WMAN idle	10 mW
WLAN send	592 mW
WLAN receive	375 mW
WLAN idle	2 mW

Table 3.2: Sample energy usage datasheets from [12] and [61].

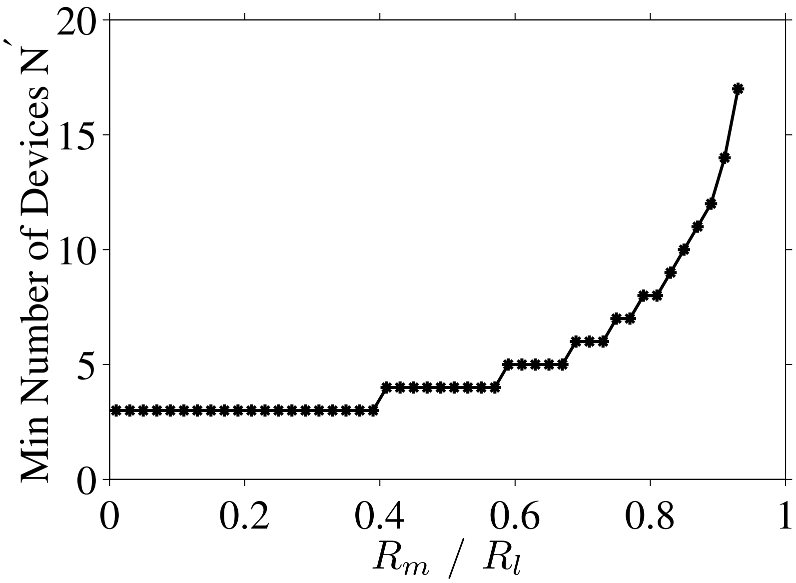

Figure 3.7: Minimum number of cooperative devices needed under different transmission speeds for WMAN and WLAN cooperative networks.

3.4.2 Numerical Analysis

To analyze the energy usage of WMANs and WLANs, we consider the data sheets released by a popular DVB-H chip manufacturer [12], which indicate that the recent WMAN chips have a receiving power consumption of 400 mW, and the power consumption drops to 10 mW in idle mode. For the WLAN energy consumption, we consider the data sheets released by the Philips Low Power 802.11 WLAN solution [61], which indicate that the lowest power consumption in the standby mode is less than 2 mW. The receiver power consumption in the IEEE 802.11g WLAN is 375 mW. The transmit power at 15 dBm is 592 mW for the 802.11g WLAN. These sample values are listed in Table 3.2. Multiple parameters affect the value of N' and we evaluate their impacts in the following.

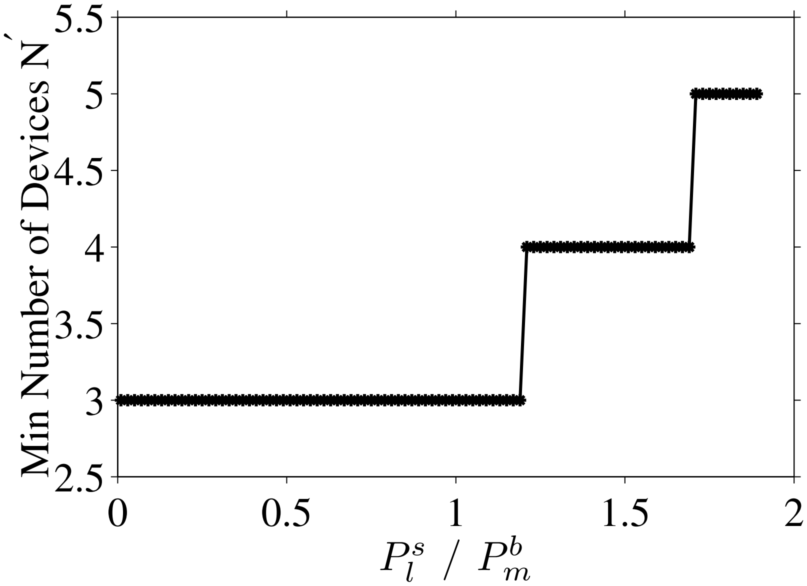

Figure 3.8: Minimum number of devices needed under different energy cost.

Transmission Speed Difference. The transmission speed difference in WMAN and WLAN can affect the values of T_m^b, T_m^i, T_l^b and T_l^i. We use R_m to denote the transmission speed of WMAN and R_l to denote the transmission speed of WLAN. To transmit the same amount of data, we have $R_m T_m^b = R_l T_l^b$. Since WMAN and WLAN use the same cycle period, we must have $T_m^b + T_m^i = T_l^b + T_l^i$. We vary the value of R_m/R_l and we use Eq. (3.3) to compute N' with the sample energy value listed in Table 3.2. We plot the results in Fig. 3.7. This figure shows that N' is fairly small for practical value of R_m/R_l. For example, when the WLAN rate is *only* two times faster than the WMAN rate ($R_m/R_l = 0.5$), the proposed cooperative system only needs 4 devices ($N' = 4$) to outperform current systems. This requirement is easy to meet in the real world.

Energy Consumption Difference in WMAN and WLAN. It is expected that hardware manufacturers will keep improving the energy consumption of their chips. P_m^b, P_l^s and P_l^r are the dominant energy parameters. We vary the value of P_m^b/P_l^s. We assume $P_l^r/P_l^s = 0.6$ which is similar to the sample ratio in Table 3.2. We also assume $R_m/R_l = 0.5$. We use Eq. (3.3) to compute N' with common network parameters and plot the results in Fig. 3.8. This figure shows that if the WLAN chip can reduce its energy cost level to about the same as WMAN chip, the proposed cooperative system only needs 3 devices ($N' = 3$) to

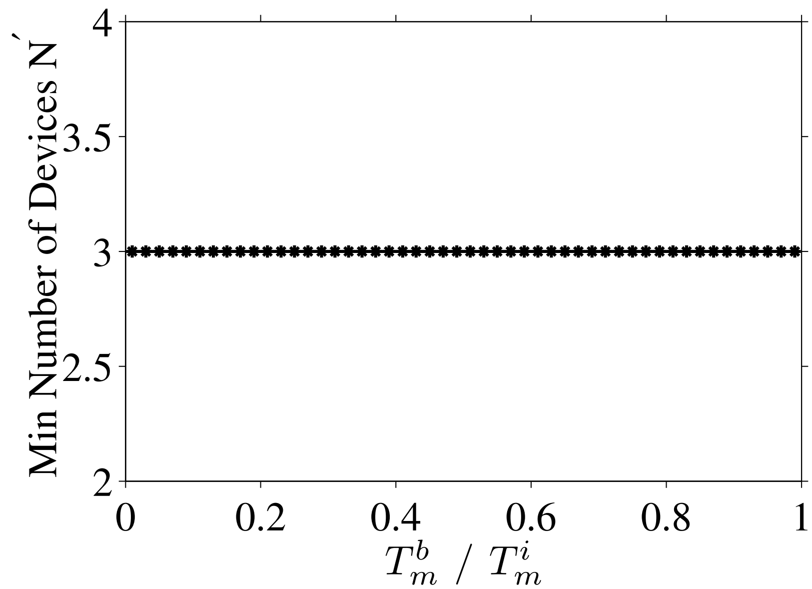

Figure 3.9: Minimum number of devices needed under different WMAN broadcast settings.

outperform current system. Even with existing technology that the WLAN chip consumes about 50 percent more energy than WMAN chip, only 4 devices are needed, which is also easy to meet in practice.

WMAN Broadcast Setting. The broadcast setting in the WMAN base station can affect the value of T_m^b and T_m^i, where a smaller T_m^b/T_m^i value can lead to more energy saving on the WMAN only system. In order to evaluate the impact of the WMAN base station setting on the N' value, we assume that the WLAN rate is three times faster than the WMAN rate ($R_l/R_m = 3$) which is easy to meet in practice. And we compute N' under different T_m^b/T_m^i by using Eq. (3.3) and Table 3.2. We plot the results in Fig. 3.9. This figure shows that N' in our cooperative system is independent from the WMAN broadcast setting. As long as there are 3 devices in the proposed cooperative system ($N' = 3$), it can outperform the current WMAN only system.

Chapter 4

Evaluation in the Testbed

In this chapter, we evaluate the proposed cooperative system in a mobile TV testbed. We choose the DVB-H standard to stand for WMAN and the IEEE 802.11g to standard for the WLAN. We first describe our DVB-H and IEEE 802.11g cooperative testbed setup [1] in Section 4.1. Then in Section 4.2, we design several experiments to measure the performance from several angles.

4.1 WMAN and WLAN Cooperative Testbed

4.1.1 Overview

The protocols we implement for the DVB-H and IEEE 802.11g cooperative testbed are shown in Fig. 4.1. Our implementation includes Electronical Service Guide (ESG) delivered via the File Delivery over Unidirectional Transport (FLUTE) protocol, time slicing schemes, Multi-Protocol Encapsulation (MPE) forward error correction computations and Program Specific Information/Service Information (PSI/SI) tables creation. We also design a web-based management tool to configure all parameters of the testbed easily. We describe the testbed architecture in the following.

[1] This is a joint work with my colleagues Cheng-Hsin Hsu and Cong Ly under the supervision of Dr. Mohamed Hefeeda.

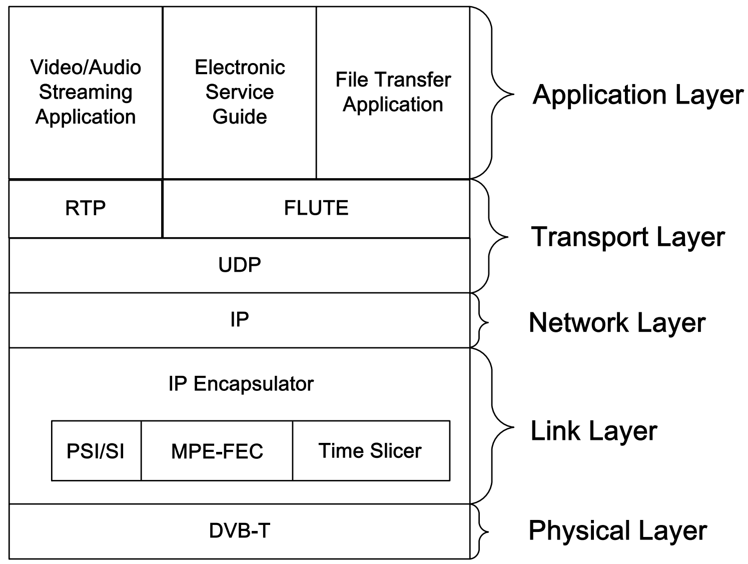

Figure 4.1: The protocols implemented in our mobile TV testbed.

ESG Delivery

The Electronic Service Guide (ESG), which is in the application layer, provides the detailed multimedia program information to users. In addition, ESG contains session information to help mobile devices to play the corresponding video streams. Before a user watches one video program, he/she needs to search and choose the corresponding ESG entry of that video program. ESG is also helpful for the DVB-H service provider to manage user subsystems such as billing and accounting. ESG is written as XML files and is transmitted to mobile devices using the FLUTE protocol. There are two kinds of ESG. One is from the DVB group called IPDC ESG [41], the other is from Open Mobile Alliance (OMA) group and called OMA BCAST ESG [5]. Our testbed implemented the IPDC ESG.

ESG operations include a bootstrap period and ESG acquisition period. There are two descriptors involved in the bootstrap process: ESGProviderDiscovery Descriptor and ESGAccess Descriptor. Both of them are delivered through a FLUTE session with a well known IP and port. The ESGProviderDiscovery contains a ProviderID which is unique and used to identify a particular ESG entry within the ESGAccessDescriptor. ESGAccessDescriptor contains information on how to acquire the corresponding ESG file. ESG acquisition starts after a successful ESG bootstrap. In addition, there exists an ESG update process that restores the latest version of the ESG information in a terminal after a period of time. We

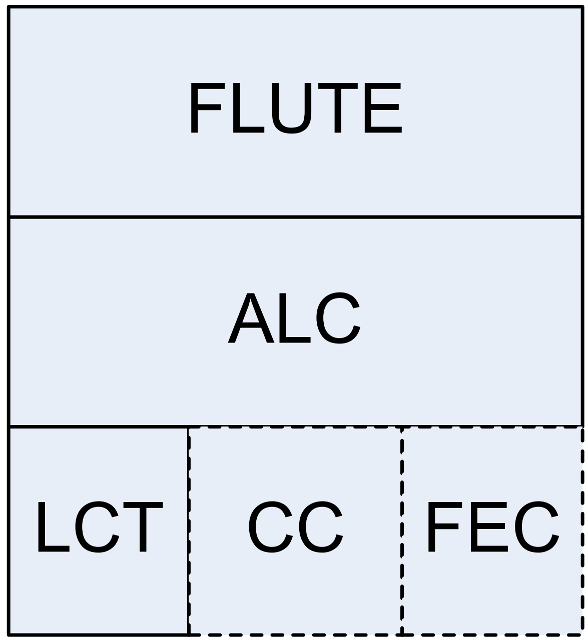

Figure 4.2: FLUTE Building Block Structure.

implement ESG as XML files in the testbed. We analyze some pre-recorded DVB-H transport stream and parse out the contained ESG files with the help of the dvbSam analyzer [18]. We also develop scripts to systematically create these XML files.

FLUTE Protocol

FLUTE is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. Flute sessions run over one (or more) Source-Specific Multicast (SSM) or Any-Source Multicast (ASM) channels and support both IPv4 and IPv6. The FLUTE protocol specifications build on the Asynchronous Layered Coding (ALC) protocol [49], which is the base protocol designed for massively scalable multicast distribution systems. ALC is a protocol instantiation of the Layered Coding Transport building block (LCT) such that ALC inherits the session concept of LCT [50]. ALC combines the LCT building block, a Congestion Control (CC) building block and a Forward Error Correction (FEC) building block [51] to provide congestion controlled reliable asynchronous delivery. The reliability is maintained by retransmissions and Forward Error Correction. LCT can provide transport level support for reliable content delivery and stream delivery protocols. The use of CC and FEC in FLUTE is optional. The FLUTE structure is shown

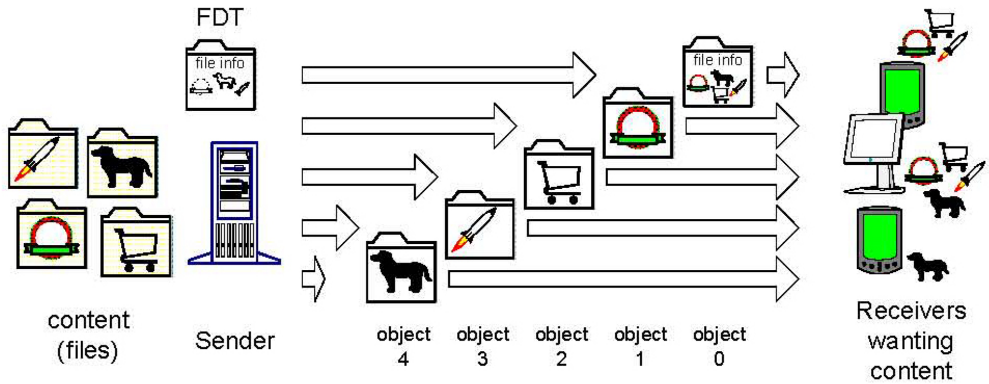

Figure 4.3: FLUTE Example from [26, pp. 27].

in Fig. 4.2.

In addition to inheriting the congestion control, session management and reliable transmission features from the ALC, FLUTE introduces a new concept called File Delivery Table (FDT) which is a set of file description entries for files to be delivered in the session to map the files to their attributes [32]. FDT provides all the attributes that are related to file transmission including file names, IDs, types, sizes and so on. Within the file delivery session the FDT is delivered as FDT Instances. One FDT Instance can describe all or part of the files for the FLUTE session. The FLUTE protocol recommends the transmission of FDT instances before any other files such that receivers can predetermine whether the coming files are needed to avoid receiving unnecessary files. Receivers maintain an FDT database based on the received FDT instances. At any time, the FDT instances in the receiver FDT database represent the current knowledge of the file transmission. A simple example of 4 files delivered by a session which are all described by one FDT Instance is shown in Fig. 4.3. Note that the FDT Instance is delivered before the set of files.

The DVB-H standard uses the FLUTE protocol as a Content Delivery Protocol (CDP). The FLUTE protocol is used for transmitting text, audio, video, and image files in order to download files required for the ESG construction and ESG update. The FLUTE protocol supports one-to-many data transmission without the need of any feedback. Thus, there is no limitation on the number of supported receivers, which is well suited in video broadcast networks. In order to transmit ESG files, we integrated an open-source FLUTE implementation called MAD-FLUTE [52] into our testbed. Because MAD-FLUTE is not targeted for DVB-H IPDC ESG, we customized it to make it compatible with the DVB-H IPDC standard [26]. For example, there is no congestion control problem in DVB-H network. Thus, we removed the congestion control function from MAD-FLUTE project. We enhanced the algorithm used in ALC FEC to match the requirements of the DVB-IPDC Content Delivery Protocol [27]. We also used the Session Description Protocol (SDP) to record the FLUTE session transmission parameters which is used by FLUTE receivers to receive FDT instances.

Time Slicing Scheme

DVB-H data is encapsulated in MPE sections and is delivered in bursts. With the help of the open-source FATCAPS DVB-H Encapsulator [29], we implement an IP Encapsulator that completes the header fields for the different protocols according to the DVB-H standard. This IP Encapsulator reads IP data from configurable sources and organizes the data in MPE frames. In the link layer, each MPE carries a time parameter called *Delta_t*. It signals the relative time from the start of current MPE section to the start of the next burst duration. The DVB-H standard also defines that *Delta_t* equals to zero means the end of the service. By using this *Delta_t* in MPE sections, time synchronization between the base station and the receivers is no longer an issue. Every MPE section contains its *Delta_t* value such that even if in bad network condition some MPE sections are lost, as long as at least one MPE section is received by the receiver, proper *Delta_t* value can be accessed by the receivers to achieve time slicing scheme. We implement the time slicing scheme in our testbed by setting *Delta_t* as an input configurable parameter together with the transmission parameters such as OFDM bandwidth, coderate, constellation and guard interval. Users can decide the *Delta_t* and transmission parameter values and our software application dynamically calculate the corresponding MPE burst rate, burst duration to fulfill the time slicing scheme. The output stream is calculated such that it saturates the output channel as defined by the transmission parameters (OFDM bandwidth, coderate, constellation, guard interval). Many null packets with PID 0x1FFF exist in the output stream, which can later be replaced by PSI/SI tables.

MPE Forward Error Correction Computation

The MPE-FEC frame is arranged as a matrix with 255 columns and a flexible number of rows, between 1 and 1024. There exist two parts within an MPE-FEC frame. The left part, which consists of the 191 leftmost columns, is used for IP datagrams and possible padding. The right part, which consists of the 64 rightmost columns, is dedicated for the parity information of the FEC code. Our software application collects 191 $\times$ numRows Bytes of data for each MPE-FEC frame and calculates the corresponding MPE forward error correction.

Program Specific Information/Service Information Tables Creation

The output of the IP Encapsulator contains many null packets with PID 0x1FFF, which are replaced by PSI/SI data. The PSI/SI, in the link-layer, provides signaling for carrying the network configurations. The PSI/SI in DVB-H complements the PSI/SI in DVB-T to support the IP packets transmission in DVB-H such as time slicing and MPE-FEC. It is organized as tables, where each table is encapsulated in one or more TS packets. The TS packets of the same PSI/SI table share a PID, which allows receivers to reassemble PSI/SI tables.

Several tables are defined in the DVB-H standard. For example the Network Information Table (NIT) carries tuning information such as frequency, bandwidth, and modulation scheme on a given network. The unique NIT table id is 0x40 with which is carried on PID 0x0010. The Program Association Table (PAT) provides the information about how many programs in a channel and the association of PMT PID and Program Number. The unique PAT table id is 0x00. The Program Map Table (PMT) identifies and indicates the location of the streams that make up each service. The IP/MAC Notification Table (INT) describes the availability and location of IP streams within a DVB MPEG2 transport stream. We use the open-source DVB-T based MPEG2 transport stream data generator [42] to help generate PSI/SI data. In addition, we expand this generator by adding the DVB-H specific descriptors as well as constructing INT table which only exists in DVB-H. We also create several configuration scripts to generate valid PSI/SI tables following the DVB-H standard. An example of the whole PSI/SI flow is shown in Fig. 4.4 [24, pp. 24]. All sample values shown in the figure come from our mobile TV testbed. A brief description of this figure follows.

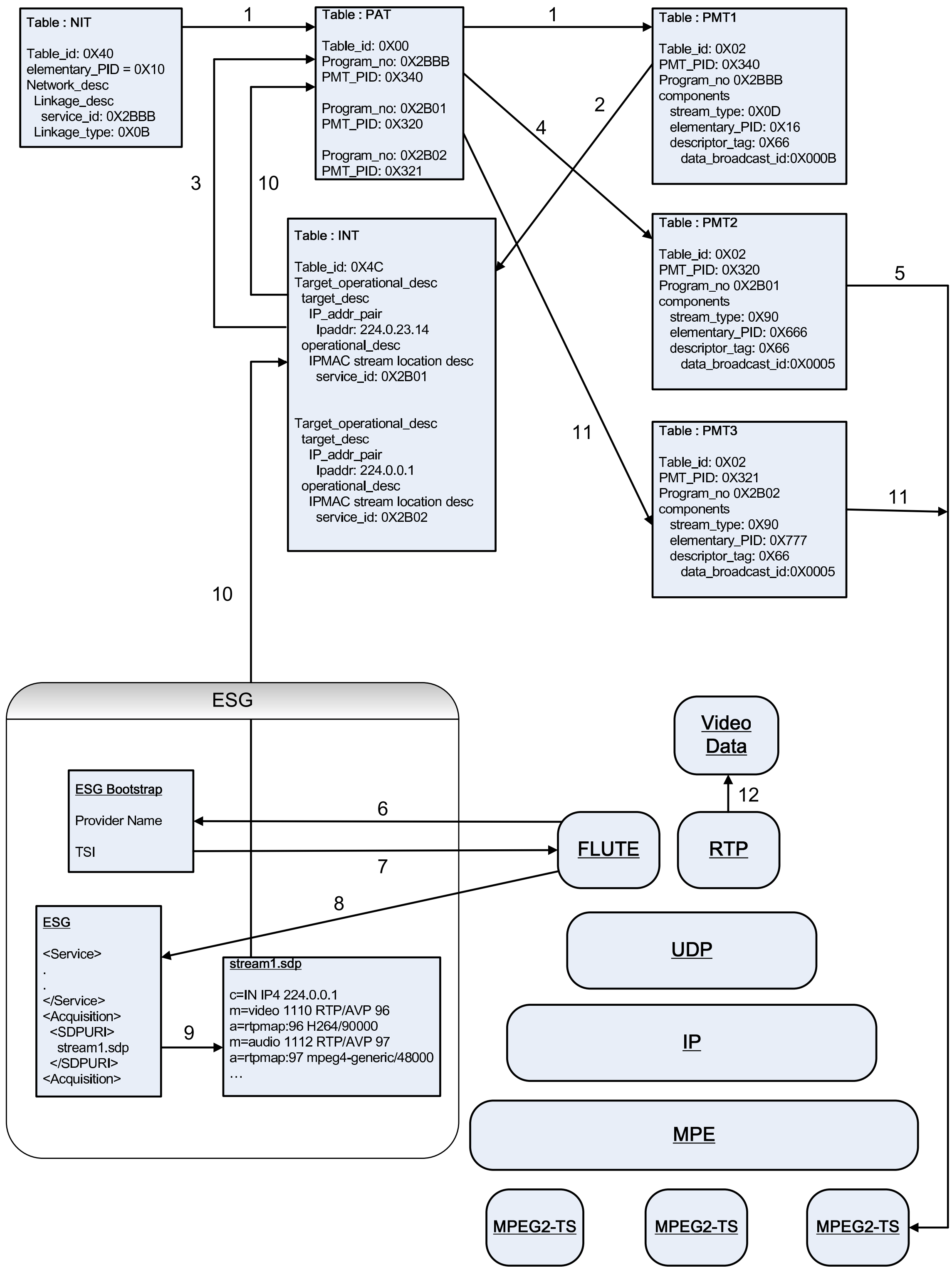

Figure 4.4: Flowchart describing the roles of PSI/SI tables in mobile TV networks.

1. At the beginning, we retrieve NIT by Table_id 0x40 and PID 0x10 from the given transport stream (TS). Within NIT network_descriptors, there is an entry called linkage_type with value 0x0B which means the TS contains INT. INT only exists in DVB-H streams. We then search for the entry called linkage_descriptor to get the value of its service_id. This value is used later on in PAT table to search for PMT_PID. In our example, the NIT service_id value is 0x2BBB such that we search the program_number entry of the PAT. There is a match with program_number entry value 0x2BBB in the PAT such that we can get the PMT_PID for the corresponding PMT table which is PMT1 with PMT_PID 0x340.

2. After retrieving PMT1, we first detect whether or not there exists two components called stream_type and descriptor_tag. If the value of the stream_type is 0x0D and the value of the data_broadcast_id entry inside descriptor_tag is 0x000B, it indicates an INT exists in an elementary stream. The value of the elementary_PID entry in the PMT1 shows the PID of INT such that we can retrieve INT with Table-id 0x4C and PID 0x16.

3. Now we are in INT. This table can guide us for ESG bootstrap process. Its ipaddr entry shows the common used ESG bootstrap IP, which is 224.0.23.14. We fetch the value of service_id under IPMAC_stream_location_descriptor and use this value 0x2B01 to find out the corresponding PMT_PID in the PAT. We search the program_number entry of the PAT. There is a match with program_number entry value 0x2B01 in the PAT such that we can get the PMT_PID for the corresponding PMT table which is PMT2 with PMT_PID 0x320.

4. Now we are in PMT2. We first check the value of the stream_type. Since its value is 0x90, it indicates that the current transport stream is an IP stream. We then check the value of the data_broadcast_id entry inside descriptor_tag that is 0x0005. This value indicates the current content is of MPE structure. The value of the elementary_PID in PMT2 0x666 can guide us to retrieve the ESG bootstrap data from the transport stream.

5. Receiver receives the specific transport stream with PID 0x666 and combined them all together in order to extract the final data. At this stage, we can retrieve ESG-Bootstrap.

6. After we get the ESG-Bootstrap. Because each object is associated with a unique Transport Object Identifier (TOI) within the scope of a FLUTE session, we can use the TOI to retrieve two files: ESGAccessDescriptor.bin and ESGProviderDiscovery.xml. The ProviderID in ESGAccessDescriptor file was extracted afterwards.

7. From the previous step, we can also get the Transport Session Identifier (TSI) value and set the TOI value to 0 in the FLUTE session. TSI is carried in the ALC/LCT header and is scoped by the source IP address. In addition, the (source IP address, TSI) pair uniquely identifies a session. Therefore we can retrieve FDT from TS.

8. At this step, we parse both the FDT file and the ESG data by the XML parser and retrieve some user oriented content so that users can interact with ESG.

9. At this step, we parse the SDP file in order to know what contents are in the TS so that can start to prepare the corresponding decoder to play the video content.

10. Within the SDT file, the value of attribute c=IN IP4 224.0.0.1 corresponds to the INT. In INT, based on the IP address 224.0.0.1 in target_descriptor, it finds the corresponding service_id 0x2B02 in the operational_descriptor. We can use this service_id to search in the PAT in order to find the corresponding PMT3 with PMT_PID 0x321.

11. PMT with PMT_PID 0x321 is found. Since its data_broadcast_id value is 0x0005 and its stream_type value is 0x90, it is an IP stream. So that we can retrieve the content from TS based on its elementary_pid value 0x777.

12. After knowing the PID of the IP streams, users can start to watch the video programs they selected on the mobile devices.

Web Interface for the Testbed

We design a web-based interface, written in PHP, to control the whole testbed. Using this interface, many parameters of the base station can easily be configured, e.g, Constellation, Bandwidth, Frequency, Guard Interval, Convolutional Rate and Transmission Mode. A snapshot of the interface is shown in Fig. 4.5. Another important function for the web interface is to configure the PSI/SI tables. As illustrated in Fig. 4.5, various IDs can be

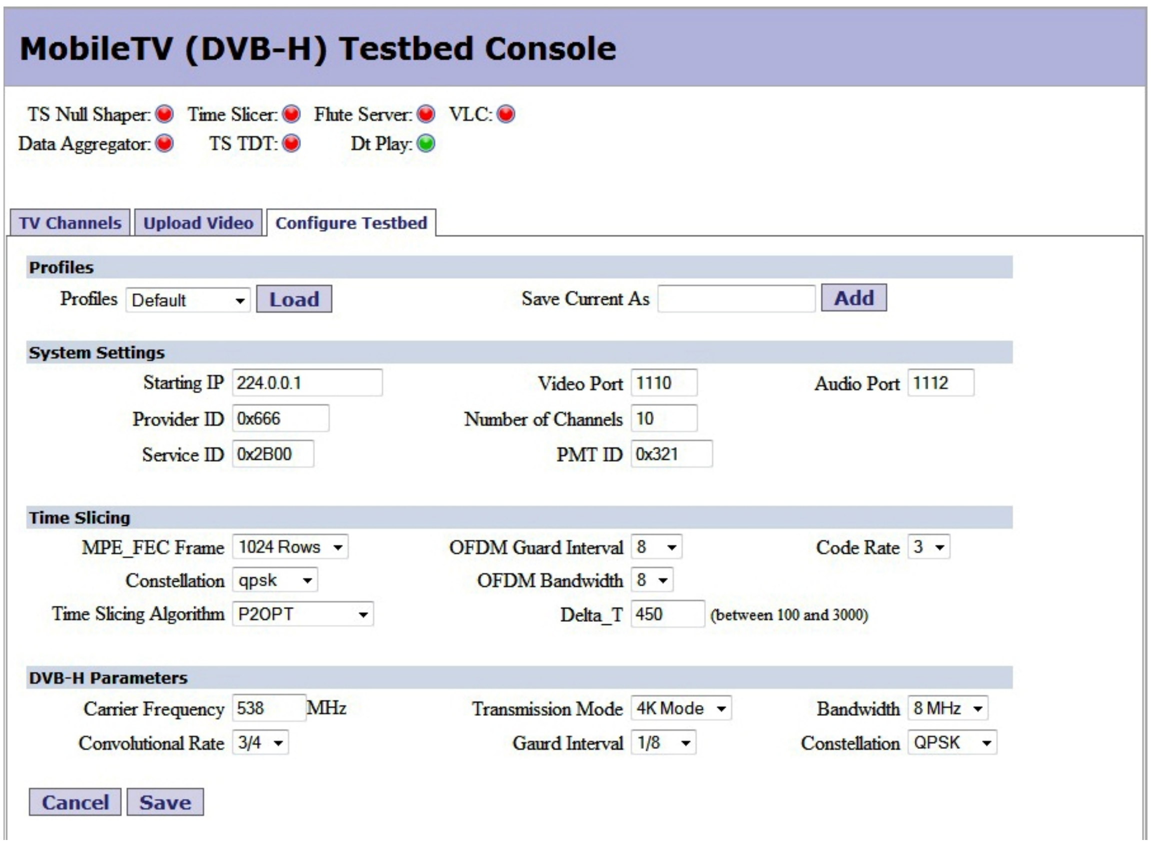

Figure 4.5: A snapshot of the web-based interface for managing the mobile TV testbed.

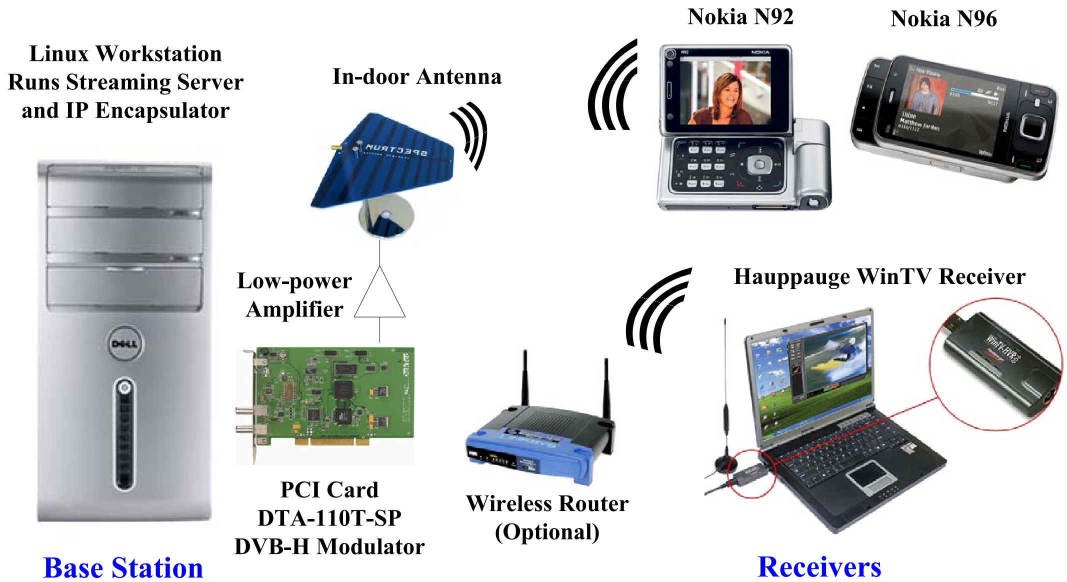

Figure 4.6: Real Device Based DVB-H and WLAN Cooperative Testbed.

configured to generate the customized PSI/SI tables. This web interface facilitates conducting various experiments, because it does not require researchers to modify and compile the source code of different components of the system.

4.1.2 Testbed Setup

We setup a DVB-H and WLAN cooperative testbed, which is shown in Fig. 4.6. This testbed generates valid DVB-H video streams and provides an IEEE 802.11g WLAN environment. It includes a base station, receivers, several data analyzers and management tools. We describe each component in the following. Table. 4.1 lists all hardware components used in the testbed.

WMAN Base Station. We use a commodity Linux PC as our base station, in which we install an RF signal modulator card [11] to implement the physical layer of the DVB-H stack. This modulator card can modulate the MPEG-2 packets and transmit DVB-H signals. The RF output level of the modulator, however, is quite low ($\sim$ -29 dBm) and can only reach up to 1-meter broadcast range for receivers with 6 dB gain antenna. Mobile devices typically have antenna gains much lower than 6 dB. We use the amplifier available from Enensys [21] to boost the signal to about 0 dBm, which gives us approximately 20-meter range for cell phones in our lab environment. The amplifier is connected to the DVB-H

Hardware Components	Usage
Linux PC	DVB-H base station
DTA-110T PCI based modulator card [11]	Provide physical layer of DVB-H stack
Enensys amplifier [21]	Boost the signal to about 20-meter range
LP49-DTV antenna [66]	DVB-H compatible antenna
Linksys WRH54G router	Provide WLAN environment
Nokia N96 and N92 cell phones	Verify DVB-H streams and quality
DiviCatch receiver [13]	Data analyzer of DVB-H steams
Hauppauge WinTV receiver [69]	Low cost USB based DVB-T receiver

Table 4.1: List of all hardware components used in the mobile TV testbed.

compatible antenna LP49-DTV [66]. This PCI card comes with a software tool to control the OFDM modulator. Several software programs are developed to organize the data in MPE FEC frame, compute the MPE forward error correction, properly handle the program specific information/service information (PSI/SI) for DVB-H services and transmit the data under the time slicing scheme discussed in Sec. 4.1.1.

WLAN Access Point. The IEEE 802.11g based wireless router Linksys WRH54G is used to provide the WLAN environment. WLAN encryption is enabled to make sure only our selected test client devices can use the wireless bandwidth.

Receivers. We use Nokia N96 and N92 cell phones to verify the output stream of our DVB-H testbed as well as assessing the video quality. We install the Nokia Energy Profiler [55] application on the cell phones and configure the profiler to take a measurement every 0.25 sec, which is the finest granularity it supports. By receiving the video stream broadcast from the testbed, we plot the energy used during a 20-sec time interval in Fig. 4.7. The fluctuations of power consumption when the mobile TV application is running is due to the time slicing mechanism. Instead of keep using the DVB-H radio component, the cell phone indeed turns off the DVB-H radio component after receiving the burst data. We also use several USB based low-cost WinTV-NOVA-T receivers from Hauppauge [69] in our testbed. Each WinTV-NOVA-T USB receiver is connected to a PC to receive the DVB-H steams. The WinTV-NOVA-T receiver is designed for DVB-T reception but can still receive DVB-H steams after our configuration and modification. More details on using these USB receivers are discussed in Section 4.1.3.

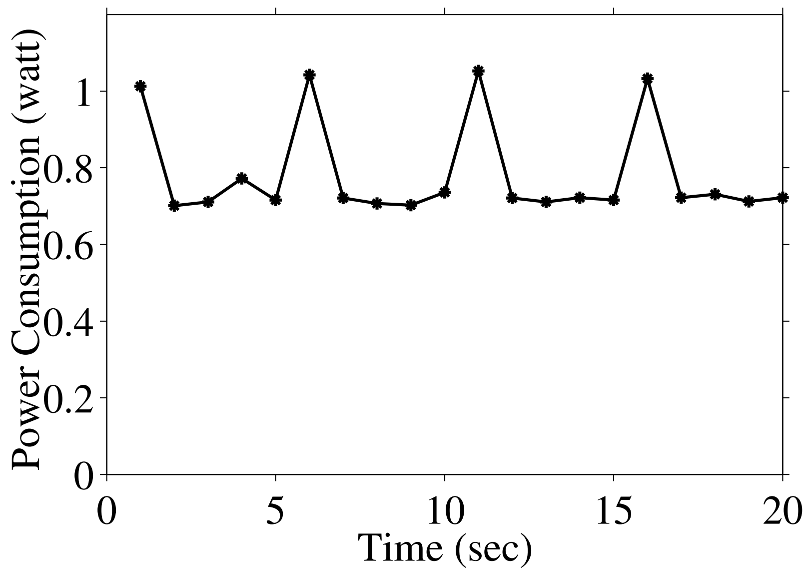

Figure 4.7: Energy consumption on Nokia N96 during DVB-H signal reception.

Data Analyzers. As the mobile TV application on the Nokia cell phone is proprietary, we can not use the cell phones to evaluate the performance metric in the testbed implementation. To address this shortcoming, we added the DVB-H Analyzer called DiviCatch available from [13] to the testbed. This analyzer can be attached to a PC via a USB port, and comes with a visualization software that runs on Microsoft Windows. Screen shot of the DiviCatch application is shown in Fig. 4.8. From this picture, we can clearly see that the video streams from our testbed follow the time slicing mechanism. The DiviCatch software also provides detailed real-time information on the RF signals, MPE frames, burst schedules, burst jitters and so on. One shortcoming of the DiviCatch analyzer is that it does not support ESG very well. To overcome this limitation, we use another DVB-H analyzer, called dvbSAM, available from Decontis [18]. dvbSAM, which is also a Microsoft Windows application, provides more details on ESG fragments and is more stable on ESG reception. It allows us to dump various information, such as PSI/SI tables and ESG fragments, in hexadecimal format to debug the testbed implementation.

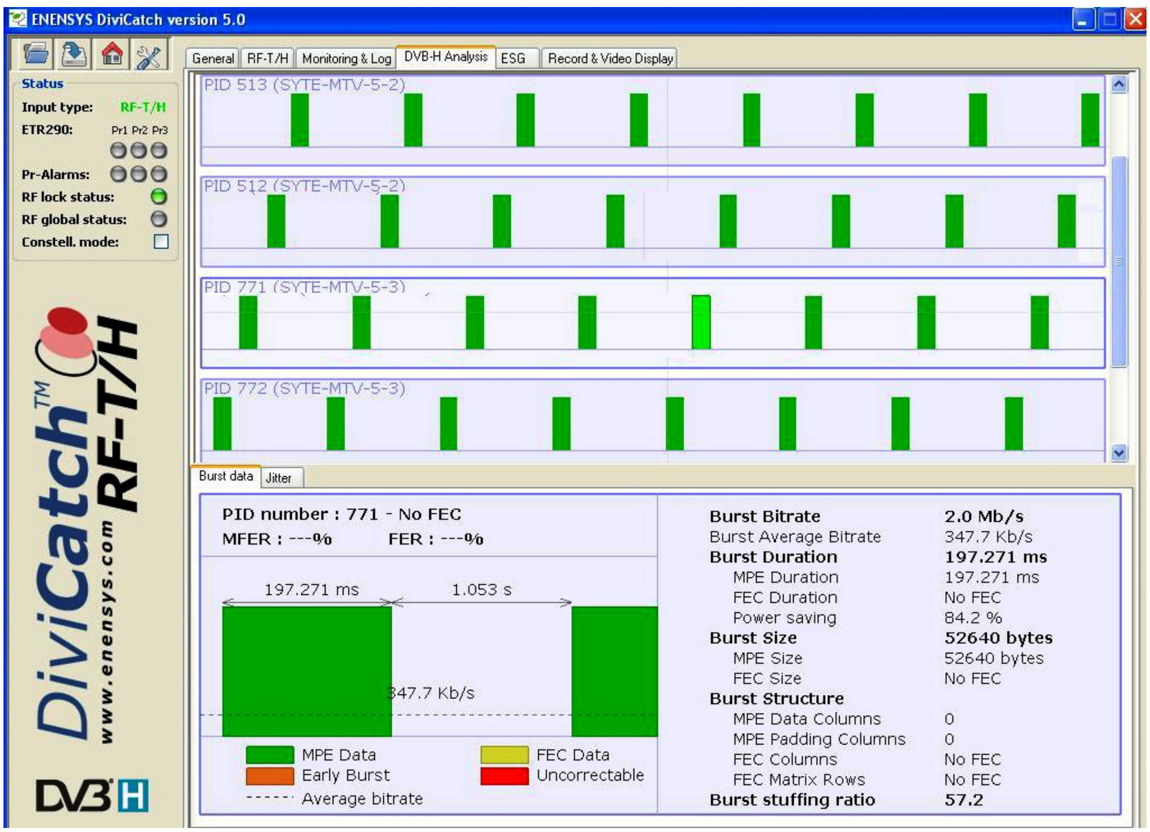

Figure 4.8: Screen shot of the DiviCatch analyzer showing the time slicing of four TV channels.

Tool	Usage
v4l-dvb [46]	Operate DVB-T USB receivers in Ubuntu Linux
dvbsnoop [19]	Debug and analyze the video stream information
w-scan [68]	Perform DVB-H signal scan of the DVB-H video stream
tzap [20]	Tune into the right frequency of the DVB-H video stream

Table 4.2: Software tools and packages used in the mobile TV testbed.

4.1.3 Algorithm Implementation

Since DVB-H is compatible with DVB-T, the DVB-T USB receivers can receive DVB-H data packets successfully. By installing the `v4l-dvb` tool provided by the Linux TV Project [46], the DVB-T USB receivers can operate in the Ubuntu Linux as a network interface. We install the `dvb-utils` [20] software package to have a collection of commonly used DVB related tools. In order to debug and analyze the video stream information, we install the `dvbsnoop` [19] software. In order to perform DVB-H signal scan of the DVB-H video stream, we install a tool called `w-scan` [68]. A channel configuration file can automatically be created after the scan. We then use the `tzap` tool from `dvb-utils` [20] to tune into the right frequency of the DVB-H video stream. We list all software tools and packages used in the testbed in Table. 4.2

We implement our proposed distributed cooperative algorithm on Linux. The corresponding Unified Modeling Language (UML) class diagram of our software application is shown in Fig. 4.9. We explain the details in the following:

Receiver class. The purpose of this class is to create a receiver that can transmit the data within the DVB-H and WLAN cooperative network. It loads the video channel configuration file and setups the virtual DVB-T network interface for the data transmission. Each line of the configuration file stands for a video stream channel, which provides the broadcast video stream PID, multicast IP and its Session Description Protocol (SDP) file name. In `loadConfig` method, we load the configuration file and create the data structure. In `execCommand` method, we execute several system command in the background to help setup the environment. Then in `init` method, virtual network interfaces are automatically created with unique local IPs and MAC addresses based on the content of the configuration file. New routing table entries are also created. Whenever the application finishes, the `uninit` method undoes the previous tasks to clean up and remove the virtual network interfaces.

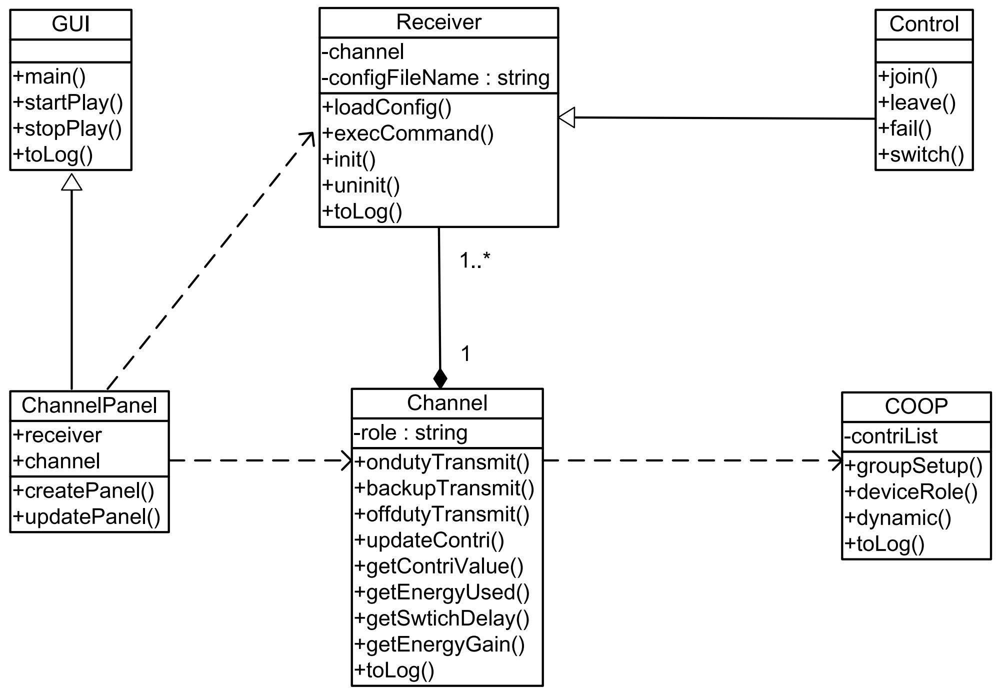

Figure 4.9: Class diagram of the proposed algorithm.

Control class. This class extends the Receiver class. The goal of this class is to control the behaviors of receiver devices of our experiments under different conditions. There are four kinds of actions for each receiver device which are failure, switch channels, leave, and new join. We create a method for each action. The `join` method emulates a new device joins the cooperative network. A JOIN message is broadcast to the cooperative group whenever the new join event happens such that the on-duty device can be informed to send data to the new joiner. The `leave` method emulates the device departure together with a LEAVE message broadcast to the whole cooperative group. The `switch` method, which is used to emulate a channel switching event. It is in fact a combination of the `leave` and `join` methods as the device first leaves the current video channel and then joins the new video channel. The `fail` method, which is used to emulate out of battery event, turns the device into the failure mode immediately without sending any message to the whole group. With the help of the Control class, we have the flexibility of designing different test cases.

COOP class. In this class, we implement our cooperative algorithm to decide the device role in each DVB-H cycle. The `groupSetup` method, called only once at the beginning of the cooperation, helps the receiver to self-form a cooperation group based on the current video channel it is watching. By exchanging multicast messages, the total number of devices, the MAC address of each device are recorded to setup the contribution list. Once the cooperative group is initialized, the on-duty device will help organize the group in the future. The `deviceRole` method is used to determine the device role in the coming DVB-H cycle. It compares the local contribution value with all other devices in the contribution list. If the local contribution value is the smallest, the device will serve as on-duty and start to broadcast ON-DUTY messages to the whole group. If the local contribution value is not the smallest but falls in the the least `K` values, the device will serve as backup and start to broadcast BACKUP messages to the whole group. If the local contribution value is not small enough, the device will server as off-duty and listen to ON-DUTY messages. Depending on the device role, the receiver device calls the methods in Transmit class for data transmission. In order to handle network dynamics, we use `dynamic` method to handle device join, leave or switch channel.

Channel class. In this class, depending on the device role, the receiver may need to receive from DVB-H directly if it serves as on-duty or backup. Otherwise it waits for WLAN data relay if it serves as off-duty. The `ondutyTransmit` method applies to on-duty device only. We uses the `libpcap` library to capture the video stream based on its PID from the DVB-H network. We set the filter to receive IP and UDP packets only. The filtered data packets are then relayed via WLAN to all off-duty devices in the same multicast group. The `backupTransmit` method applies to backup devices only. The backup device independently receives from the DVB-H network which is the same as on-duty device. After DVB-H data reception, the backup devices keep monitoring the on-duty device status in case the on-duty device leaves the cooperative group or encounters failure. The `offdutyTransmit` method applies to all off-duty devices for WLAN reception.

After data transmission, the `updateContri` method organizes the local contribution list. If the device serves as on-duty, we update the local contribution value in the contribution list based on the amount of data sent via WLAN. If the device serves as backup, we update the on-duty device entry in the contribution list based on the amount of data received via DVB-H. If the device serves as off-duty, we update the on-duty device entry in the contribution list based on the amount of data received via WLAN. In every DVB-H cycle, only the on-duty device entry of the contribution list needs to be updated. Furthermore, when we capture the IP packets of a specific DVB-H channel, each packet is associated with a receiving timestamp. Based on the timestamps, together with DVB-H burst information, we can get T_m^i, T_m^b, T_l^i and T_l^b for each cycle as shown in Eqs. (3.1) and (3.2). Precise channel switching delay can also be detected.

We design several methods such as `getEnergyGain`, `getContriValue`, `getEnergyUsed` and `getSwitchingDelay` to calculate the performance parameters. Detailed logs, which are implemented in `toLog` method, are collected which contain the device role in each round, the total contribution value, the energy saving gain, and the channel switching delay of the device (if exists). To measure the energy saving gain of our cooperative system compared to the DVB-H only system, we dynamically calculate the energy saving gain in each cycle based on Eqs. (3.1) and (3.2) together with the sample energy consumption values from DVB-H chip manufacture [12] and Philips Low-Power 802.11 WLAN solution [61].

GUI class. This class contains the main function and provides a button to call VLC media player for each video stream. Users can choose to play any available video channel to verify our implementation via `startPlay` method. We wrote the session description protocol (SDP) file for each video stream as the input for the VLC player. The path of these SDP files are included in the configuration file. `stopPlay` method is also provided for users to stop playing the video programs.

ChannelPanel class. This class extends the GUI class and displays each video stream as a JPanel to dynamically shows the performance parameters such as the number of packets received, the number of packets relayed, the current device role, channel switching delay, energy spent, the energy saving gain and so on. We update these information on the GUI every second.

4.2 Empirical Results

To conduct our experiments, we configure the testbed as follows. We setup an 8 MHz radio channel to broadcast four 5-minute long TV programs coded at 250 kbps. We use the QPSK modulation scheme together with the convolution coding rate at 2/3 and guard interval at 1/8. We have 4 WinTV-NOVA-T DVB-T receivers [69] in total. For the WLAN environment, we use the IEEE 802.11g based wireless router with encryption enabled to make sure only our selected receiver devices can use the wireless bandwidth. The receivers can take advantage of the router to use resource reservation-based protocol to reserve resources across the network during data transmission so that on-duty device can reliably relay the data via WLAN.

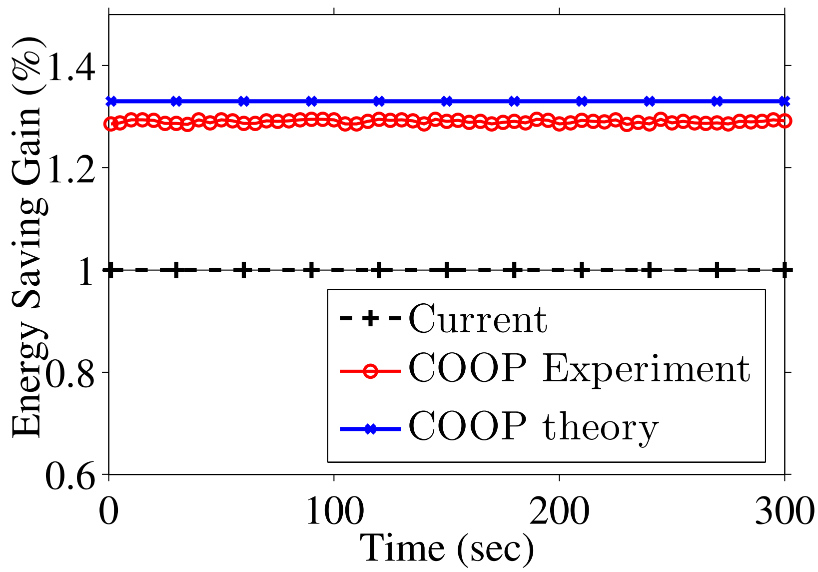

Figure 4.10: Energy saving gain in the testbed experiment.

Potential Energy Saving. We first study the potential energy saving gain that can be achieved by using the proposed cooperative system. We define the energy saving gain as $E_g = (E_e - E_c)/E_e$, where E_c is the total energy consumption of the proposed cooperative system, and E_e is that of current WMAN only system. We use 1 on-duty device, 1 backup device and 2 off-duty devices in the WMAN and WLAN cooperative system. We broadcast a 5-minute long video stream via the DVB-H testbed. Since we can get accurate T_m^i, T_m^b, T_m^0, Δ, T_l^i and T_l^b for each WMAN cycle as shown in Eqs. (3.1) and (3.2) on the receivers, after the video stream finishes, we can measure the total energy spent in the WMAN and WLAN cooperative system based on the sample energy consumption listed in Table 3.2. We broadcast the same video stream again from the DVB-H testbed and this time all 4 receivers are in the WMAN only mode. After the video stream finishes, we measure the total energy spent in the WMAN only system in the same way. We compare the experimental results against the results we get in theory by using Eqs. (3.1) and (3.2). The comparison plot is shown in Fig. 4.10. In theory, with 4 devices, we can save around 33%, and in the experiment we can save around 29%. This performance difference may due to the extra energy spent on turning on and off the network interfaces and handling all overhead messages composed in the real system. We have only four devices in our testbed, because of the limited hardware.

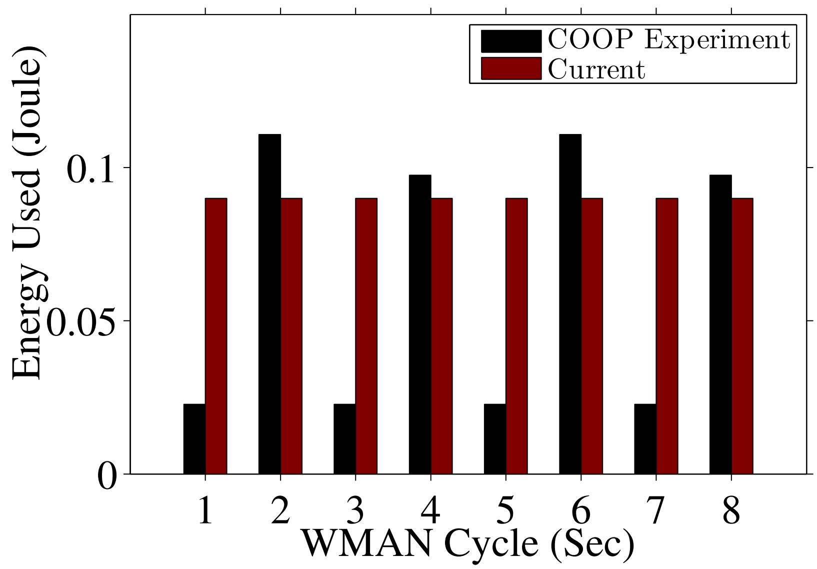

Figure 4.11: Energy consumption of one mobile device.

Yet, our experiments show a gain of up to 29% in the energy saving. The gain will be much larger with more cooperative devices.

Energy Consumption of Individual Mobile Device. Next, we analyze the energy consumption of individual client devices. Using the above setup, we present sample results of client 1; all other results are similar. We calculate the energy used by the client in both cooperative mode and WMAN only mode in each WMAN time slicing cycle and plot the results in Fig. 4.11. We draw two observations on this figure. First, in our cooperative system, because of different roles taken by the client device in WMAN cycles, its energy usage in each cycle is different. When the device is on-duty, its energy consumption is about 23% more than in WMAN only mode. When the device serves as backup, its energy consumption is about 8% more than in WMAN only mode. However, when the device is off-duty, it can save about 75% of its energy, leading to the overall energy saving gain. Second, the client device switches roles with 25% possibility of being on-duty, 25% possibility of being backup and 50% possibility of being off-duty. Knowing that the total number of devices used in this experiment is 4, these results demonstrate the fairness of our proper functioning algorithm. This is because the high energy consumption which corresponds to the device being on-duty occurs every second WMAN cycle. Similarly, the low energy

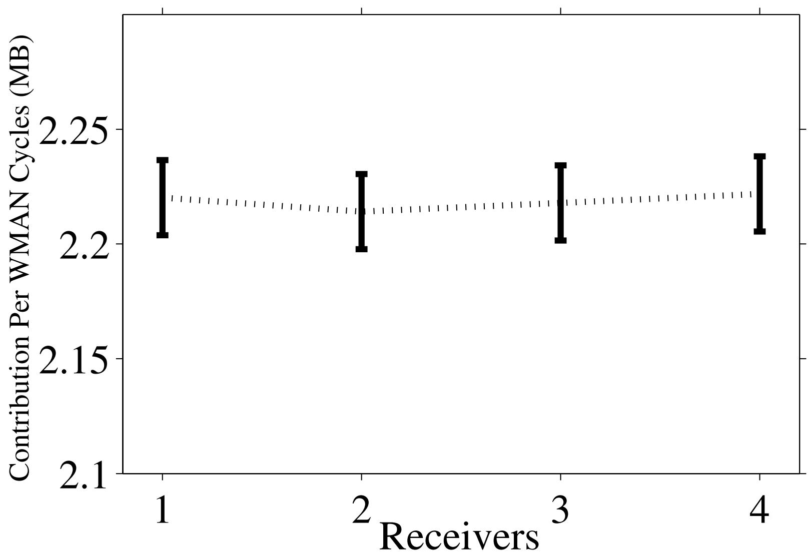

Figure 4.12: Contribution value per WMAN cycle of four receivers in testbed.

consumption which corresponds to the device being off-duty occurs every first and third WMAN cycles.

Fairness of the Proposed System. Finally, we consider the fairness of the proposed cooperative system in details. Fairness is important to mobile devices because they have stringent energy constraints, and any imbalance in the load may drive users away from the cooperative network. We measure the contribution of each device in the cooperative network throughout the whole experiments. The contribution value of a device is the number of bytes relayed by that device to other devices over the WLAN. The total contribution values for the four devices are 222.02 MB, 221.41 MB, 221.79 MB and 222.18 MB. These values confirm that all four receivers contributed almost equally during the experiments.

Next we compute the average contribution in each WMAN cycle for all devices. We also compute the 95% confidence interval, which is computed as the average value plus/minus $1.96 \times$ the value of the standard deviation for each device (1.96 is the ratio to determine 95% confidence interval). We plot the results in Fig. 4.12. The figure shows that for each device, the average contribution value per WMAN cycle falls into a small range. In addition, the range of each device is similar, which shows that the proposed cooperative system evenly balances the load on mobile devices.

Chapter 5

Evaluation using Simulation

In the previous chapter, we evaluate our algorithm in real mobile TV testbed. In this chapter, we analyze the proposed cooperative system using simulation. The simulation can support more devices than our testbed and it enables us to rigorously study the performance of the proposed system from several angles. Both the WLAN infrastructure mode and ad-hoc mode are evaluated.

5.1 WMAN and WLAN Infrastructure Mode Simulation

5.1.1 WLAN Infrastructure Mode Simulation Setup

We have implemented a trace driven simulator that concurrently supports the WMAN and WLAN. We have implemented the proposed cooperative system in the simulator. For comparisons, we have also implemented the current system, which only uses the WMAN for broadcast. For realistic simulations, we have studied a transport stream generated by a Nokia mobile TV base station. This transport stream consists of four video channels for five minutes each, where each channel is encoded at 450 kbps using H.264/AVC video codec and Advanced Audio Coding (AAC) audio codec [1]. We have developed a software utility to analyze the transport stream and create a log file for each video channel. The log files indicate the start time of each burst and its size. We have conducted the simulations with all four video streams. We present sample results of video stream 1; all other results are similar. We still use the sample energy consumption listed in Table 3.2.

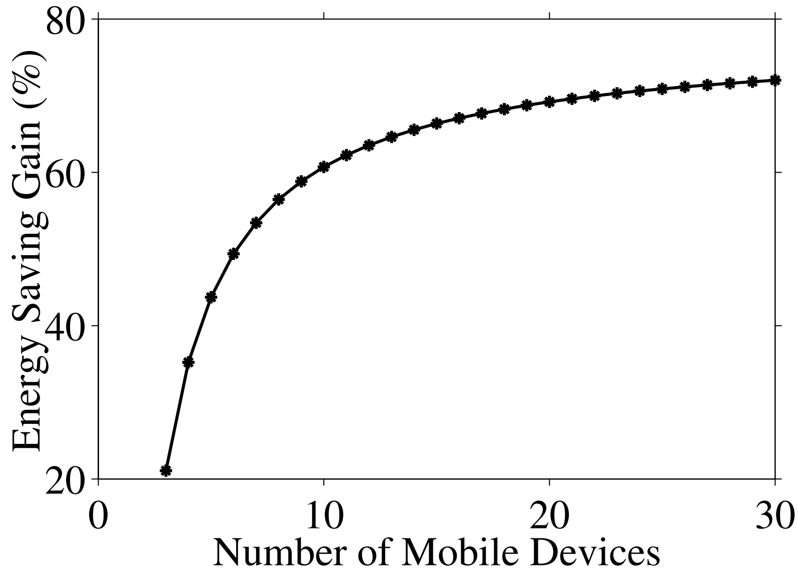

Figure 5.1: Potential energy saving gain in simulator.

5.1.2 Evaluation of WLAN Infrastructure Mode

We design several test cases to evaluate the performance under wide range system parameters.

Potential Energy Saving. We vary the total number of mobile devices from $N = 3$ to 30. We compute the average energy saving gain E_g and plot the results in Fig. 5.1. We draw two observations on this figure. First, the energy saving gain increases as the number of mobile devices increases and can be as high as 70% with a total of 30 devices.

Second, the proposed system outperforms the current systems if $N \geq 3$. For example, with only three mobile devices, the proposed cooperative system saves more than 21% energy compared to current systems. Note that the energy saving gain is relative to what can be achieved by the current system and not absolute. Since the current systems employ time slicing schemes to save energy, the absolute energy saving achieved by our proposed cooperative system will be substantial. This is improtant to prolong the battery life.

Impact of Backup Devices on Energy Saving. We turn on 30 mobile devices that view the same video stream for five minutes. We vary the number of backup devices from 1 to 6. We compute the average energy saving gain E_g under different K values and plot the results in Fig. 5.2. The figure shows that while more backup devices lead to

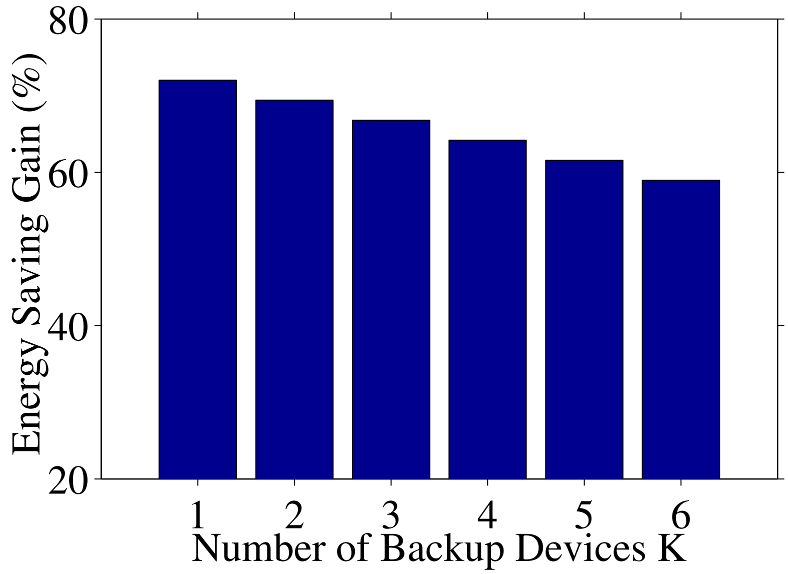

Figure 5.2: Implications of different number of backup mobile devices on energy saving.

higher robustness, more backup devices also reduce the energy saving gain achieved by the proposed system. Despite the *tradeoff* between robustness and energy saving, the energy saving gain achieved by the proposed system does *not* dramatically drop when the number of backup devices increases. As illustrated in this figure, even when devoting 1/5 of mobile devices to be backup devices, the energy saving gain is still higher than 58%.

Switching Delay Reduction. Next, we study the channel switching delay under user dynamics. We broadcast four video streams for five minutes. For each video stream we choose 1 on-duty mobile device and 1 backup mobile device. We turn on 120 mobile devices to watch one of the four video streams randomly. Initially, each video stream has 30 mobile devices. We generate random channel switching events using Bernoulli trials, and we set the probability of success in a way that users stay with one video stream for 30 sec on average. For each channel switching event, the mobile device randomly selects a new video stream other than the currently viewed one. We derive the channel switching delay by measuring the time difference between when a user decides to switch channel until he/she gets the data for the new video stream. We then compute the average channel switching delay of each mobile device, and we plot the CDF curves for the current and the proposed systems in Fig. 5.3. This figure illustrates that the proposed cooperative system effectively eliminates

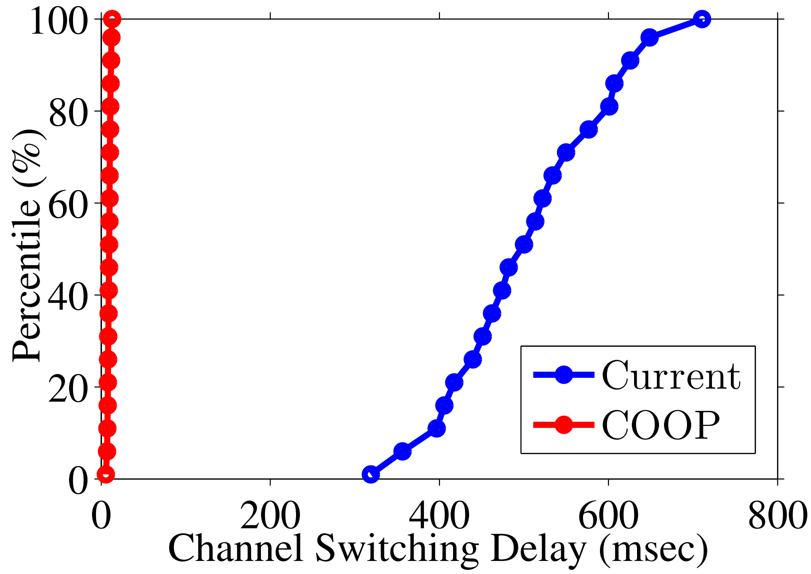

Figure 5.3: Channel switching delay gain.

the channel switching delay: from up to 700 msec in current systems to at most 13 msec in the proposed system. That is to say, our cooperative system can reduce the channel switching delay by up to 98%.

Impact of New Joiners on Energy Saving. We study the impact of energy saving when mobile devices dynamically join the cooperative network. Initially we turn on 3 mobile devices that view the same video stream for five minutes. We configure the system to have 1 on-duty and 1 backup mobile devices. After the first 30 seconds, we gradually add mobile devices to the group in order to emulate that new users choose to watch this video stream. More precisely, we insert a mobile device every 10 seconds such that by the end of the 5-minute video stream, we have 30 mobile devices in total. We compute the average energy saving gain E_g, and we plot the results in Fig. 5.4. We draw two observations on this figure. First, for the particular round when new device joins in the group, extra energy are used to provide the data packets to the new joiners. Second, after that round, the cooperative group can reorganize itself properly to achieve even better energy saving due to the bigger group size.

Fail-safe Mechanism under Device Failures. We study the effectiveness of the fail-safe mechanisms under mobile device failure. We turn on 30 mobile devices that view

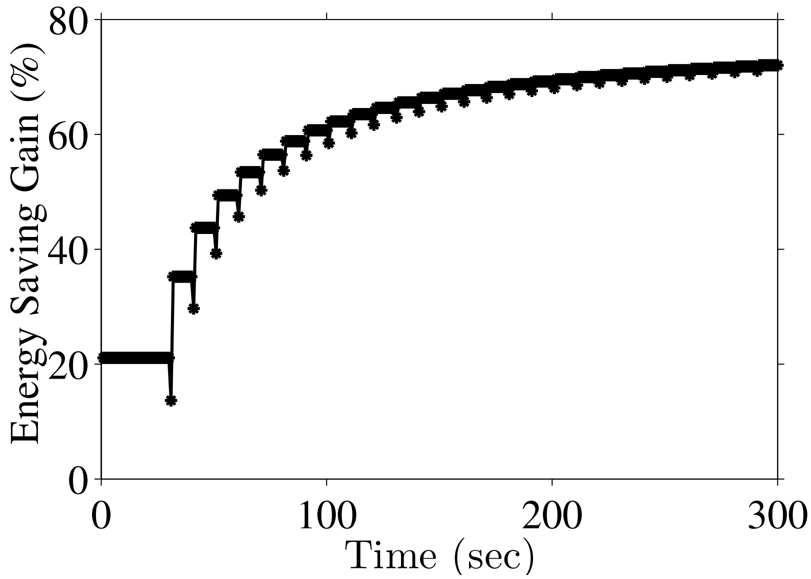

Figure 5.4: Energy saving gain under network dynamics: new mobile devices join.

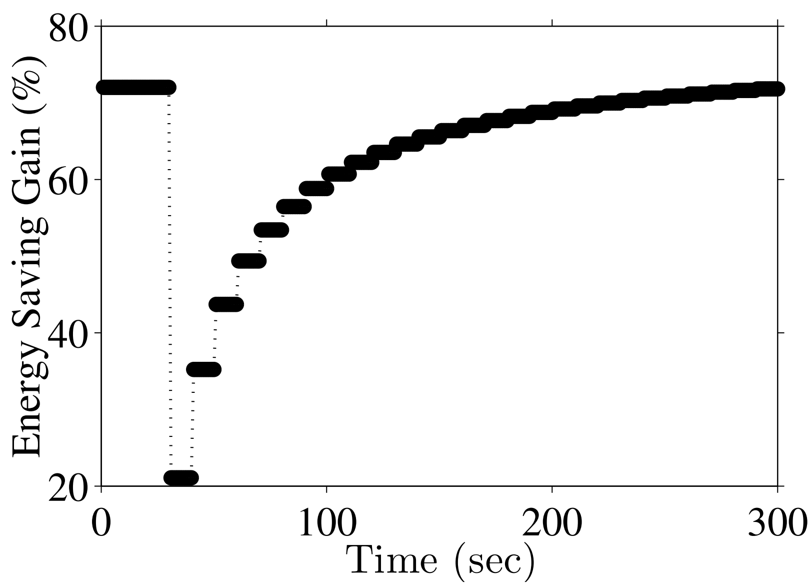

Figure 5.5: Energy saving gain under network dynamics: failed and rejoined mobile devices.

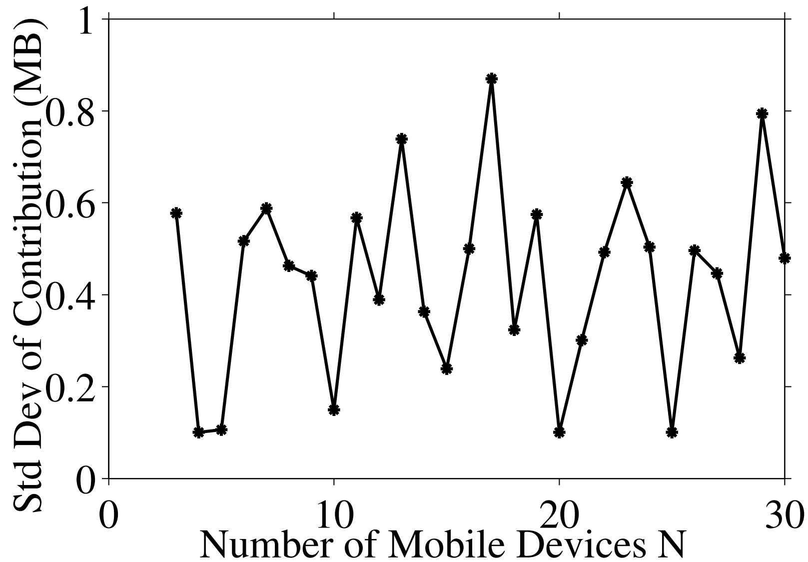

Figure 5.6: Fairness of the proposed system.

the same video stream for five minutes. We configure the system to have 1 on-duty and 1 backup mobile devices, and instruct the simulator to fail 90% of the mobile devices after 30 seconds. In addition, we gradually add these failed mobile devices back in order to emulate that the users either reboot their mobile devices or adjust the antennas for better reception. We insert a mobile device every ten seconds. We compute the average energy saving gain E_g, and we plot the results in Fig. 5.5. The figure shows that the proposed cooperative system is robust, because it survives a sudden loss of 90% of mobile devices, despite an energy saving hit of about 51%. The proposed cooperative system also quickly adapts to the network dynamics, because once the failed mobile devices rejoin the cooperative network, the energy saving increases and finally can reach the original level with the same number of devices in the group.

Fairness of the Proposed System. Next, we consider the fairness of the proposed cooperative system. We turn on N mobile devices that view the same video stream for five minutes. We vary the value of N from 3 to 30. We measure the contribution (in terms of number of bytes) of each mobile device. For each N value, we compute the standard deviation of the contribution among all mobile devices, and we plot the results in Fig. 5.6. This figure shows that the standard deviation of contribution is always less than 0.6 MB. The figure clearly shows that the proposed cooperative system evenly balances load on mobile devices, given that the total contribution is in the order of several hundred MBs.

5.2 WMAN and WLAN Ad-hoc Mode Simulation

5.2.1 WLAN Ad-hoc Mode Simulation Setup

The simulations in the previous section use the infrastructure mode of the WLAN in the proposed cooperative system. In this section, we evaluate the cooperative system under the WLAN ad-hoc mode. We implement another simulator to do this. We use the video stream from Nokia and the sample energy consumption values in Table 3.2. We setup a 10m $\times$ 10m area for all devices. The WLAN coverage for each device is limited to a radius of 2m. We assume that within this WLAN coverage, the WLAN transmission rate remains the same. Two devices that are more than 2m apart cannot cooperate directly. Each device can move with a speed of 0.2m/s in any direction within the defined area. We make this estimation as in real world users using WLAN ad-hoc mode to watch video programs on mobile devices usually are not moving fast while watching video programs.

5.2.2 Evaluation of the WLAN Ad-hoc Mode

We design several test cases to evaluate the algorithm performance.

Cooperative Group Size. We first study the dynamics of group formation and the average number of devices in each group as devices move and change their locations. We use N_g to denote the number of groups in the defined area and N_d to denote the total number of devices in the defined area. We calculate N_d/N_g every second. We configure the system to add 1 device per second to the center of the defined area until there are 30 devices in total. Each device can move at any direction with the speed of 0.2m/s. We plot the results in Fig. 5.7. The figure clearly shows the dynamics of the system, as the number of groups changes because of mobility. These results confirm that our cooperative system can properly function in the ad-hoc mode and it can adapt to user mobility.

Potential Energy Saving. Next, we study the energy saving gain in the ad-hoc mode. We initially put 3 devices in the center of the defined area. We configure the system to add 1 device per second to the center of the defined area until there are 30 devices in total. Each device can move at any direction with the speed of 0.2m/s. Every second, we calculate the

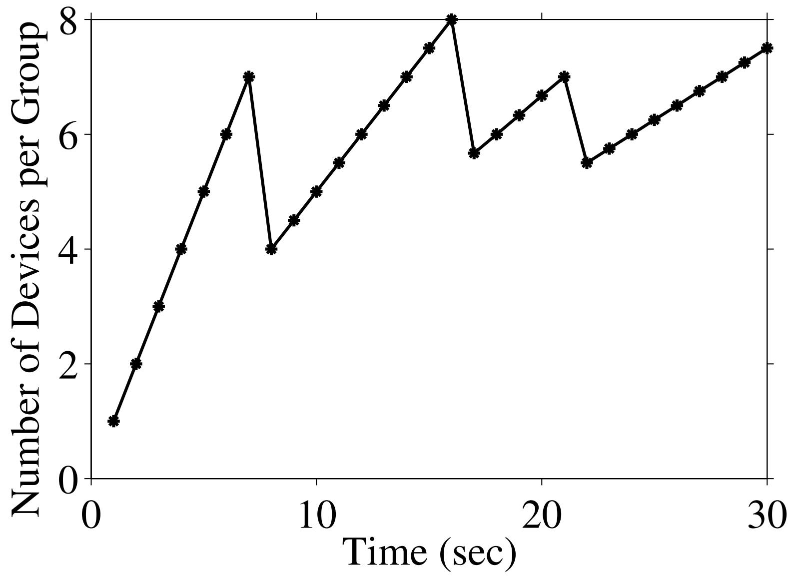

Figure 5.7: Average number of devices per group.

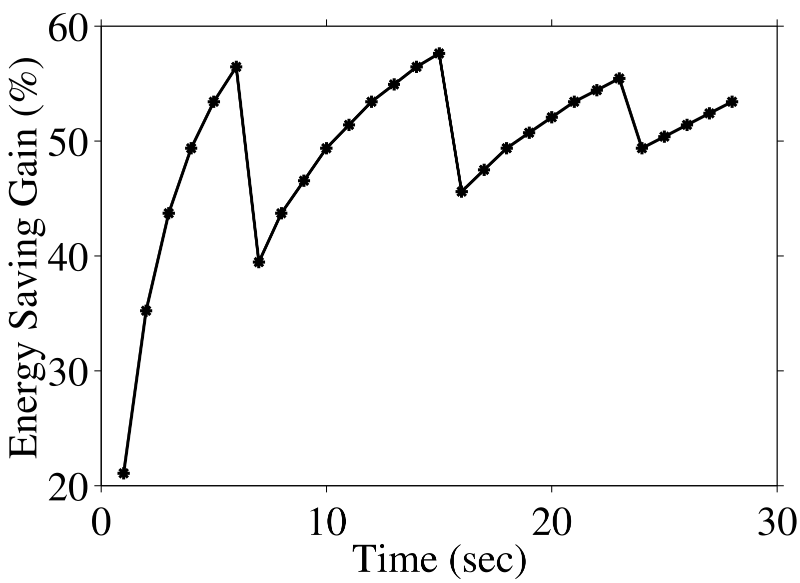

Figure 5.8: Energy saving gain in ad-hoc mode.

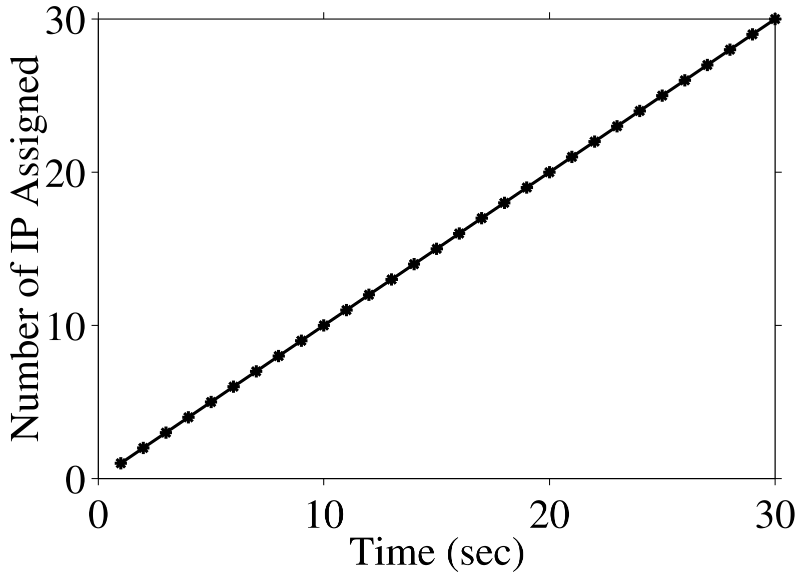

Figure 5.9: MANET IP management.

potential energy saving gain. In each cooperative group, we use 1 on-duty and 1 backup devices. We plot the results in Fig. 5.8. The figure clearly shows that in the ad-hoc mode, our cooperative system can still have energy saving gain as high as 57% with a total of 30 devices. Notice that the fluctuations in the energy saving are due to device mobility and re-formation of the cooperative groups.

IP Management. We want to verify that our IP assignment strategy can indeed perform well in the WLAN ad-hoc mode. We configure the system to add 1 device every second to the center of the defined area until there are 30 devices in total. Each device once added can move at any direction with the speed of 0.2m/s to emulate the user movement. We collect the total number of IP addresses each second and plot the results in Fig. 5.9. The figure clearly shows that the proposed IP assignment strategy indeed helps in managing IP addresses to form the cooperation groups. This is because whenever a new device is added, an IP is assigned fast. Finally all 30 devices get their own unique IP addresses.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize contributions of this thesis. Then, we briefly describe some future research directions to extend the current work.

6.1 Conclusions

We studied the problem of broadcasting video streams over a WMAN to many mobile devices. We have proposed a cooperative system, in which several mobile devices *share* received video data bursts over a WLAN. The proposed system reduces the energy consumption and significantly reduces the channel switching delay. We have analytically shown that the proposed cooperative system outperforms current systems in terms of energy consumption with only few cooperative devices.

We presented a simple distributed leader election algorithm to choose the on-duty and backup devices in the cooperative network. The proposed algorithm imposes very little computation and communication overheads. Moreover, the proposed algorithm enables truthful cooperation among devices because it distributes the load by rotating the on-duty role across all devices. The on-duty device is elected based on previous contributions, where the contribution of a device is not computed by itself but computed independently by all other devices based on the actual data received from the on-duty device. Therefore the room for a device to cheat or free-ride is limited.

We setup a DVB-H and IEEE 802.11g cooperative testbed to evaluate our proposed algorithm. We implement the full DVB-H protocol stack including ESG delivery via FLUTE protocol, time slicing scheme, MPE-FEC computation, and PSI/SI generation. For easier configuration, we also design and develop a web based GUI interface to control our testbed. Valid DVB-H streams are received and verified on Nokia cell phones as well as a USB based DVB-H stream analyzer. We also implemented our cooperative algorithm in the real receivers to evaluate the performance of our cooperative network.

Our experimental results from the real testbed showed that the proposed cooperative system is promising, because it: (i) achieves as high as 29% of energy saving gain with a total of 4 devices, and (ii) balances the work load properly among all cooperative devices. We also implemented a trace driven simulator that addresses larger networks, device failures, departures and mobility. Our simulations confirmed the viability of the proposed cooperative system and the significant energy saving potential that can be achieved by using our simple algorithm. The simulation results clearly showed that our cooperative system only requires three (or more) mobile devices to outperform the current WMAN only systems. With 3 devices only, our system outperforms the current systems by 21% in terms of enengy saving. With 30 mobile devices, it can outperform the current system by 70%. The results also showed the channel switching delay can be reduced by up to 98% (reduced from up to 700 msec to up to 13 mesec). Finally the simulation results showed that our cooperative system is robust under device failures and quickly reacts to network dynamics, as well as it uniformly distributes the load across all mobile devices.

6.2 Future Work

The work in this thesis can be extended in several directions. Some of them are summarized bellow for future research:

- The IEEE 802.11n standard has been officially released in September 2009. This standard provides much faster transmission speeds than that in IEEE 802.11g standard. The higher transmission speeds in WLAN will result in shorter relay period within the cooperative group such that more energy can be saved in our proposed cooperative network. Experimental and analytical analysis can be conducted to validate and quantify this intuition.

- Our open source testbed currently supports the Digital Video Broadcast-Handheld (DVB-H) standard only. However, the design of our testbed is modular and each component has well-defined interfaces. This enables the flexibility of updating different hardware and/or software components with minimal impacts. For example our testbed can be extended to support other Digital Video Broadcast standards such as Digital Video Broadcast-Terrestrial (DVB-T), Digital Video Broadcast-Cable (DVB-C) and Digital Video Broadcast-Satellite (DVB-S). This extension can help us exploring the benefits of integrating different standards for video steaming.

- User interactivity can be another research direction. In current WMAN only network, the feedback channel is based on cellular network and is very expensive to use. In our WMAN and WLAN cooperative network, we can use the WLAN network as the feedback channel such that we no longer need to occupy the expensive cellular network. In addition, WLAN has higher speed and is almost free. This extension can increase the popularity of Video on Demand (VOD) services and increase the revenue for mobile TV service providers.

- Handover strategy in our WMAN and WLAN cooperative network can be improved. Due to the nature of WMAN broadcast network, the WMAN base station has no information about its client devices such that in WMAN only network the mobile devices must make decisions themselves and perform the handover. With the help of the cooperative network, WLAN can be used to provide feedback to the WMAN base station to accelerate and improve the handover procedure.

- Support WLAN interface only mobile phones can also be a potential research direction. Even though several mobile devices support both WMAN and WLAN interfaces, the majority of mobile phones only support WLAN and can not receive mobile TV programs directly. By using our cooperative network and sharing the video within the WLAN, we enable WLAN only mobile phones to watch mobile TV programs. This can be an advertise feature for mobile TV service provider to attract more potential users and can result in more revenue in the long term.

Bibliography

[1] Nokia Vancouver Design Center. Cooperation with Nokia's engineers managing mobile TV base stations, December 2008.

[2] ATSC Mobile DTV standard overview, 2009. http://www.atsc.org/standards/a153.php.

[3] E. Balaguer, F. Fitzek, O. Olsen, and M. Gade. Performance evaluation of power saving strategies for DVB-H services using adaptive MPE-FEC decoding. In *Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'05)*, pages 2221–2226, Berlin, Germany, September 2005.

[4] Mobile TV predicted to be a hit, 2007. http://news.bbc.co.uk/2/hi/technology/6639249.stm.

[5] BCAST ESG technical specifications, 2009. http://www.openmobilealliance.org/Technical/release_program/.

[6] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade. Amroute: Adhoc multicast routing protocol. In *IETF manet, draft-talpade-manet-amroute-00.txt*, August 1998.

[7] S. Cho, G. Lee, B. Bae, K. Yang, C. Ahn, S. Lee, and C. Ahn. System and services of Terrestrial Digital Multimedia Broadcasting (T-DMB). *IEEE Transactions on Broadcasting*, 53(1):171–178, March 2007.

[8] CMMB technology overview, 2009. http://www.cmmb.org.cn/.

[9] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, pages 85–110, April 1990.

[10] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The pim architecture for wide-area multicast routing. In *IEEE/ACM Transactions on Networking*, volume 4, pages 153–162, April 1996.

[11] Dektec DTA-110T PCI modulator, 2009. http://www.dektec.com/Products/DTA-110T/.

[12] Evolution of DVB-T front-end receivers through integration, June 2007. http://www.dibcom.info/Images/Upload/pdf/whitePaper2_integration_MD_V2.pdf.

[13] Divi Catch RF-T/H transport stream analyzer, 2009. http://www.enensys.com/.

[14] DVB-H global mobile TV : FAQ, 2009. http://dvb-h.org/faq.htm/.

[15] Digital Video Broadcasting - Handheld (DVB-H) home page, 2009. http://www.dvb-h.org/.

[16] DVB-H Iraq mobile TV customer trial summary page. http://dvb-h.org/Services/services-iraq-mobision.htm/.

[17] DVB-H Moscow mobile TV customer trial summary page. http://dvb-h.org/Services/services-moscow-kentavr.htm/.

[18] dvbSAM DVB-H solution for analysis, monitoring, and measurement, 2009. http://www.decontis.com/.

[19] dvbsnoop home page, 2009. http://dvbsnoop.sourceforge.net/.

[20] DVButil home page, 2009. http://www.mythtv.org/wiki/Dvbutils/.

[21] Enensys 1-watt RF power amplifier, 2009. http://www.enensys.com/.

[22] Eten Glofiish technical specifications, 2009. http://www.phonegg.com/Eten/glofiish-V900/Eten-glofiish-V900.html/.

[23] Radio broadcasting systems: Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers. European Telecommunications Standards Institute (ETSI) Standard EN 300 401 Ver. 1.3.3, May 2001.

[24] Digital Video Broadcasting (DVB); DVB specification for data broadcasting. European Telecommunications Standards Institute (ETSI) Standard EN 301 192 Ver. 1.4.1, June 2004.

[25] Digital Video Broadcasting (DVB); transmission system for handheld terminals (DVB-H). European Telecommunications Standards Institute (ETSI) Standard EN 302 304 Ver. 1.1.1, November 2004.

[26] Digital Video Broadcasting (DVB); IP datacast over DVB-H: Content delivery protocol. European Telecommunications Standards Institute (ETSI) Standard EN 102 472 Ver. 1.1.1, November 2006.

[27] Digital Video Broadcasting (DVB); IP datacast over DVB-H: Content delivery protocols (CDP) implementation guidelines. European Telecommunications Standards Institute (ETSI) Standard EN 102 591 Ver. 1.1.1, October 2007.

[28] G. Faria, J. Henriksson, E. Stare, and P. Talmola. DVB-H: Digital broadcast services to handheld devices. *Proceedings of the IEEE*, 94(1):194–209, January 2006.

[29] FATCAPS project, 2009. http://amuse.ftw.at/downloads/encapsulator/.

[30] FLO forum home page, 2009. http://www.floforum.org/.

[31] FLO technology overview, 2009. http://www.qualcomm.com/common/documents/brochures/tech_overview.pdf.

[32] FLUTE Protocol, 2009. http://www.networksorcery.com/enp/rfc/rfc3926.txt.

[33] B. Furht and S. Ahson, editors. *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T, and MediaFLO*. Auerbach Publications, Boca Raton, FL, 1st edition, 2008.

[34] GSmart website, 2009. http://www.gigabytecm.com/.

[35] F. Hartung, U. Horn, J. Huschke, M. Kampmann, T. Lohmar, and M. Lundevall. Delivery of broadcast services in 3G networks. *IEEE Transactions on Broadcasting*, 53(1):188–199, March 2007.

[36] M. Hefeeda and C. Hsu. Energy optimization in mobile TV broadcast networks. In *Proc. of IEEE Innovations in Information Technology (Innovations'08)*, pages 430–434, Al Ain, United Arab Emirates, December 2008.

[37] M. Hefeeda, C. Hsu, and Y. Liu. Testbed and experiments for mobile TV (DVB-H) networks. In *Proc. of ACM Multimedia '08 Demo Session*, pages 995–996, Vancouver, Canada, October 2008.

[38] C. Hsu and M. Hefeeda. Time slicing in mobile TV broadcast networks with arbitrary channel bit rates. In *Proc. of IEEE INFOCOM'09*, pages 2231–2239, Rio de Janeiro, Brazil, April 2008.

[39] C. Hsu and M. Hefeeda. Bounding switching delay in mobile TV broadcast networks. In *Proc. of ACM/SPIE Multimedia Computing and Networking (MMCN'09)*, San Jose, CA, January 2009.

[40] IEEE802.11 background overview, 2009. http://www.ieee802.org/11/.

[41] IPDC ESG technical specifications, 2009. http://www.dvb-h.org/.

[42] JUSTDVBIT project page, 2007. http://krusty.cineca.it/vpweb/cgi-bin/blosxom.cgi/.

[43] M. Kornfeld and G. May. DVB-H and IP Datacast – Broadcast to handheld devices. *IEEE Transactions on Broadcasting*, 53(1):161–170, March 2007.

[44] U. Ladebusch and C. Liss. Terrestrial DVB (DVB-T): A broadcast technology for stationary portable and mobile use. *Proceedings of the IEEE*, 94(1):183–193, January 2006.

[45] LG KU950 technical specifications, 2009. `http://www.phonegg.com/LG/KU950/LG-KU950.html/`.

[46] Linux TV project, television with linux, 2009. `http://www.linuxtv.org/`.

[47] Y. Liu and M. Hefeeda. Video streaming over cooperative wireless networks. In *Proc. of ACM Multimedia Systems (MMSYS'10)*, Scottsdale, AZ, February 2010.

[48] Y. Liu, C. Hsu, and M. Hefeeda. On the benefits of cooperative video broadcast over WMANs and WLANs. In *Proc. of ACM Multimedia'09*, pages 901–904, Beijing, China, October 2009.

[49] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft. Asynchronous layered coding (ALC) protocol instantiation. *RFC 3450*, December 2002.

[50] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley, and J. Crowcroft. Layered coding transport (LCT) building block. *RFC 3451*, December 2002.

[51] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft. Forward error correction (FEC) building block. *RFC 3452*, December 2002.

[52] MAD-FLUTE project, 2009. `http://mad.cs.tut.fi/`.

[53] G. May. The IP Datacast system - overview and mobility aspects. In *Proc. of IEEE International Symposium on Consumer Electronics (ISCE'04)*, pages 509–514, Reading, UK, September 2004.

[54] D. Niyato and E. Hossain. A hierarchical model for bandwidth management and admission control in integrated IEEE 802.16/802.11 wireless networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'07)*, pages 3763–3767, Hong Kong, China, March 2007.

[55] Nokia Energy Profiler, 2009. `http://www.forum.nokia.com/`.

[56] Nokia N92 technical specifications, 2009. `http://europe.nokia.com/find-products/devices/nokia-n92/`.

[57] Nokia N96 technical specifications, 2009. `http://europe.nokia.com/A4797549/`.

[58] V. Ollikainen and C. Peng. A handover approach to DVB-H services. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME'06)*, pages 629–632, Toronto, Canada, July 2006.

[59] H. Pan, S. Yamano, and H. Tani. On the effectiveness of integrating WiMAX and Wi-Fi. In *International Journal of Communication Networks and Distributed Systems*, volume 2, pages 198–213, November 2009.

[60] X. Pei, G. Zhu, Q. Wang, G. Liu, and W. Yuan. Cooperative load-balancing strategy based on N-hops routing algorithm of ad hoc network in heterogeneous overlapping networks. *Annals of Telecommunications*, pages 597–606, November 2008.

[61] Philips BGW211 low-power 802.11 WLAN chip, April 2005. `http://www.mt-system.ru/documents/bgw211.pdf`.

[62] M. Rezaei, I. Bouazizi, and M. Gabbouj. Joint video coding and statistical multiplexing for broadcasting over DVB-H channels. *IEEE Transactions on Multimedia*, 10(7):1455–1464, December 2008.

[63] M. Rezaei, I. Bouazizi, V. Vadakital, and M. Gabbouj. Optimal channel changing delay for mobile TV over DVB-H. In *Proc. of IEEE International Conference on Portable Information Devices (PORTABLE'07)*, pages 1–5, Orlando, FL, March 2007.

[64] M. Rezaei, M. Hannuksela, and M. Gabbouj. Video splicing and fuzzy rate control in IP multi-protocol encapsulator for tune-in time reduction in IP datacasting (IPDC) over DVB-H. In *Proc. of IEEE International Conference on Image Processing (ICIP'06)*, pages 3041–3044, Atlanta, GA, October 2006.

[65] M. Rezaei, M. Hannuksela, and M. Gabbouj. Tune-in time reduction in video streaming over DVB-H. *IEEE Transactions on Broadcasting*, 53(1):320–328, March 2007.

[66] Spectrum LP49 DTV indoor antenna, 2009. `http://spectrum.co.kr/`.

[67] M. Takada and M. Saito. Transmission system for ISDB-T. *Proceedings of the IEEE*, 94(1):251–256, January 2006.

[68] W-scan home page, 2009. `http://edafe.org/vdr/w_scan/`.

[69] WinTV-NOVA-T USB DVB-T receiver, 2009. `http://www.hauppauge.co.uk/`.

[70] C. Wu, Y. Tay, and C. Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification. In *IETF manet, draft-ietf-amris-spec-00.txt*, November 1998.

[71] X. Yang, Y. Song, T. Owens, J. Cosmas, and T. Itagaki. Performance analysis of time slicing in DVB-H. In *Proc. of Joint IST Workshop on Mobile Future and Symposium on Trends in Communications (SympoTIC'04)*, pages 183–186, Bratislava, Slovakia, October 2004.