國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

以可適性媒體播放技術改善垂直換手過程之視訊品質研究

Improving the Performance of Video Streaming during Vertical Handover through Adaptive Media Playout

何雍淵

Yung-Yuan Ho

指導教授：謝宏昀 博士

Advisor: Hung-Yun Hsieh, Ph.D.

中華民國 99 年 7 月

July, 2010

國立臺灣大學（碩）博士學位論文
口試委員會審定書

以可適性媒體播放技術改善垂直換手過程之視訊品質研究

Improving the Performance of Video Streaming during Vertical Handover through Adaptive Media Playout

本論文係何雍淵君（R97942097）在國立臺灣大學電信工程學系、所完成之碩（博）士學位論文，於民國 99 年 7 月 29 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

謝宏昀（簽名）

（指導教授）

高榮鴻　　鄭振牟

[illegible]

系主任、所長　陳光禎（簽名）

誌謝

離完成論文只剩一步之遙，也得感謝各位一路上的陪伴和鼓勵，才會有如此小小成果得以呈現作為回報。

感謝　謝宏昀老師這兩年來的辛勤指導，讓我知道身為「研究生」應該要有的心態和那種執著那種熱情。讓我原本從「結果和預期差不多就好」的心態轉變成為「為趨勢而探索變因」，以及「有何方式可達到更好的結果」而努力。撰寫論文時，老師不厭其煩的和我討論從論文題目的擬定、章節的編排，到式子的推導、驗證其合理性，以及實驗結果的分析角度。能成為謝老師的學生我感到非常榮幸！

感謝各位口試委員撥冗來參加我的口試，並給我許多寶貴的意見，而使我的論文能更趨於完整。在此感謝各位教授的不吝指教！

感謝TONIC實驗室的各位成員們，謝謝你們在我做研究報告的時候提供我不錯一些想法和意見。Kkstar的精闢見解和分析幾可與謝神和Dexter匹敵，你沒留下來對TONIC的損失難以估計！Jin將自己的意見化為條列式的剖析，往往具有很高的參考價值！Tego你真的很酷，我一定要給你一個大姆指！KCT的程式功力之強足以自己開一家公司！還有Sun、Eddie、Maple、Ellen和Chien，有你們的加入，讓整個實驗室充滿生氣！

另外，也感謝我的摯友們：簡董、唐牛、老邦、RO男、重漢、小朱、成哲、阿德、佩佳等，感謝你們陪伴我度過研究所時光；感謝Facebook中一同玩Fishville和Fishbowl的板友們，以及感謝陪我分享快與不快的「熊熊」、「秋」和「影子」，對於妳們的鼓勵和關心余謹記於心！

最後，感謝爸爸、媽媽還有弟弟的體諒和照顧，有你們的扶持，使我在研究所生涯中沒有後顧之憂。有了這段研究所的磨練，未來進入職場，一定能更加得心應手，謝謝你們！

何雍淵@台灣大學電信研究所

中華民國九十九年八月十五日

摘要

近來，有越來越多的人透過智慧型行動裝置來使用視訊串流服務，包括隨選視訊以及視訊會議等。由於這些行動裝置通常支援雙模甚至多模網路介面，因此在使用視訊串流服務的同時,有可能會因為網路品質的變化而需要進行異質網路間的垂直換手，以切換至最佳的網路介面。在這樣的換手過程中，封包群體遺失或是延遲送達的現象會較傳統因無線通道下造成的問題來的嚴重,而使得視訊串流的品質大幅的下降。雖然相關文獻曾提出以「播放快取監視機制」和「接收速率監視機制」來做為控制播放速率的方式，但這些機制並不能有效解決在異質網路垂直換手過程中所遇到的問題。

有鑑於此，在本論文我們利用「可適性媒體播放」的技術，小幅度降低播放的速率來增加可播放畫面的比率，以改善封包少量、間歇性的遺失或是延遲之下的視訊品質。我們首先針對異質網路換手的特性，進行「換手時機」以及「換手期間」的預測，以獲得離換手所剩餘的時間和換手所需的時間。其次，我們將這些資訊跨層提供給「可適性媒體播放」模組，利用排程規劃執行該技術的時機和期間，並且設計不同的演算法來處理使用者的移動行為。我們使用 NS-2 來做為驗證可適性媒體播放技術的模擬平台，從模擬結果中可以發現，我們提出的技術可以有效的改善異質網路換手對視訊串流品質造成的影響。

ABSTRACT

Recently, there are more and more people using video streaming services through intelligent mobile devices, including on-demand video and video conference. Since these mobile devices often support the dual-mode or multi-mode network interfaces, it will switch to the best quality network while doing video streaming services by vertical handover. In the handover process, the burst packet loss or delay delivery will be more serious than the same events happened in ordinary wireless channels, and it will reduce the video quality obviously. Although the literature has proposed the "playout buffer monitoring mechanism" and "arrival rate monitoring mechanism" to control playout rate, they cannot effectively solve the problems occurred during vertical handover. According to this, we use "adaptive media playout" technology in this thesis. By slightly decreasing playout rate, it could increase the proportion of playable frame to improve the video quality in small, intermittent packet loss and delay network condition. Firstly, we refer the characteristic of vertical handover to predict "handover time" and "handover duration", to obtain the handover remaining time and the handover spending time. Secondly, we provide above cross-layer information to "adaptive media playout", then using the schedule to plan the technology execution time and duration. We also design different algorithms to handle the user mobility. We use NS-2 to be the simulation platform to verify the proposed adaptive media playout technology. From the simulation results, we could find out that the proposed technique can effectively improve the impact of the video streaming quality on heterogeneous network handover.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Recently, some advanced communication media continues be published and improved. Nowadays, more and more people are using broadband network to do many things like getting everyday news, talking with net-pals, playing on-line games, holding video conferences, doing distance educations, watching live ball games, etc. We can imagine that video streaming is one of the most successful and popular applications in communication world. There are many ways and products using video streaming technology, like IPTVs for ball games [2], web cameras for video conferences and distance educations, network cameras for surveillance systems, and even TV-guided missiles [3], etc. We can roughly separate the video streaming transmission ways from on-demand streaming and live streaming; the former has some popular applications to be examples like YouTube [4] and I'm Vlog [5], and the latter also has some popular examples like PPStream [6], hichannel [7], Gmail talk [8], etc. We will focus on video quality changing when using live streaming afterward.

The video quality becomes unstable obviously when audiences move and use some mobile devices to play live streaming, like cell phone and vehicle TV. The devices may support some types of wireless network or mobile telephony communications protocols to receive the streaming, like IEEE 802.16e [9], IEEE 802.11p [10], ETSI GSM [11], and 3GPP HSDPA [12]. Figure 1 is an example to describe a mobile node going through the several WiMAX 802.16e-2005 and WiFi 802.11b base stations signal range. If the mobile node (MN) has 802.11b and 802.16e such two network modes to switch, it is able to choose 802.11b mode to receive the streaming if the channel condition is better. When the MN finds out the received signal strength from 802.11b base station is lower than a threshold, the device will switch back to the 802.16e mode. The switching to another type of network procedure named vertical handover, and it will cause such as frame burst delay, buffer underflow and video interruption some serious problems.

Nowadays, there are some powerful techniques published to reduce the effect of network conditions on video quality. In Application Layer, loss concealment and error resilience two optional tools have already implemented in MPEG-4 standard [13]. In Transport Layer, some papers verify that using TCP-friendly to do the rate

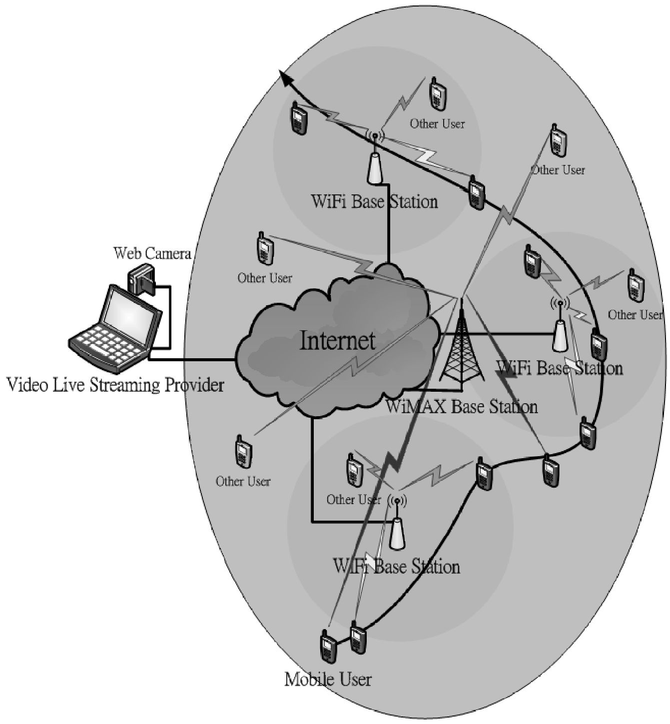


Figure 1: A sketch of WiMAX 802.16e and WiFi 802.11b heterogeneous handover

control will improve the video quality than using pure UDP. In routing protocol, AOMDV [14] is able to reduce re-routing time and losing link percentage. In MAC Layer, IEEE 802.11e standard [15] has already provided a differential service for multimedia using. Although there are many good thinkings trying to maintain the video quality as above, we could not guarantee that the whole network has already supported QoS techniques in realistic condition. Therefore, our objective is to keep high video quality by modifying the video provider and receiver only in Application Layer. We treat the lower layers as a "Black Box" and just outputting the parameters from it to judge the network condition, like the end-to-end delay, jitter, packet loss situation, etc. Some powerful techniques are already published to reduce the problem, such as loss concealment and error resilient mentioned before. About the codec-based techniques, SSC (Single Sub-stream Coding) and MSC (Multiple Sub-stream Coding) could adapt to the network condition; about the network-based techniques, transmission rate adaptation, frame rate adaptation, adaptive media playout, priority frame transmission are proposed.

Although there are many ways to improve the video quality, only few approaches are able to be used in live streaming condition, and few of them could reduce the video distortion from vertical handover. Adaptive media playout (AMP) is one of the efficient techniques, and it is suitable be applied in the network condition. The concept of AMP is adapting playout frame interval to improve frame recevied percentage before its playout deadline. About conventional AMP, it usually considers the buffer fullness to decide playout interval, but it is not suitable to fit in vertical handover case because it does not have enough time to postpone playout time when it finds out that the buffer becomes underflow due to vertical handover. For the reason, we refer to the original concept of AMP and then realizing it in vertical handover network condition.

To realize, we create two network acquirement parts to get the essential network information, which will be the input to AMP core decision part. Handover prediction time part is to get the remaining time and distance to handover by using statistical approach. Because we use propagation shadowing model to be our network model, handover prediction time part also has to get the network environment parameters before calculating. After getting the parameters, it will start to get the distance between the mobile node and the base station, and thus it could get the remaining time and distance to handover. The purpose of handover duration prediction part is to get the time spending during the handover. The part uses theoretical approach to get the handover duration, and the output will relate to the playout postponing time. We will detail the two parts in Chapter 3.

AMP core decision part is to decide AMP execution time, duration and strength, based on network acquirement parts. AMP execution duration decision is based on the output from handover duration part basically, to decide the playout postponing time requirement before handover to prevent video interruption. AMP execution time and strength are based on the output from handover time prediction part. To guarantee achieving the goal after considering the effects of shadowing network and some uncertain events, AMP core decision part uses a concept of frame storing schedule and checkpoint to adjust and update playout rate. We will detail the part in Chapter 4.

After our proposed AMP is accomplished, we will analyze and compare the algorithm formation with other AMPs and do a series of simulations. The simulation results show that our proposed AMP could improve the quality of video live-streaming over vertical handover scenario. We will detail the simulation and results in Chapter 5.

Finally, Chapter 6 concludes the thesis and the future work is depicted as well. Additionally, there is a VDoP minimization framework achieved by using roll-off approach in this chapter. VDoP is a kind of video quality evaluation tool to calculate varying of playout interval.

Chapter 2

BACKGROUND AND MOTIVATION

This chapter provides the background and motivation for the research. First, we give a general overview of the video streaming system, on-demand streaming and live streaming, and both of them are popular used in nowadays, and we will focus on live streaming case afterward. Then we introduce some network-based improvement techniques like the concept of AMP and roll-off. Next, we will discuss some phenomena happened in different network layers during vertical handover and some solving approaches. We will point out the shortcomings of the approaches, including the conventional AMP. We believe that our proposed AMP could obviously improve the video quality in vertical handover duration by listing some characteristics.

2.1 Video Streaming System

Video streaming is one of the network contents transmitted from server to client. Different from some ordinary data contents, video streaming server sends video packets which are separated from video frames. We could separate the sending, receiving and playout ways to three kinds of methods. Besides, the network stability requirement and time sensitivity of them are quite different. Then we will introduce two video quality evaluation tools which will be used in Chapter 5.

2.1.1 System Structure

When a client receives video packets from a server, it begins to assemble to frames and prepares to play, and there are three kinds of playing methods. The first method is very like traditional data file transmission; media player begins to play after receiver downloads whole video "file". The audience may have to wait for a long time before the media player begins to play-out, but this method can promise of the video quality because it is independent of network condition; in other words, the quality is locked by video file provider. This method is not mentioned anymore because the challenge to this territory is not attracted. The remaining two ways to play video streaming are on-demand streaming and live streaming. Some kinds of IPTV [2] is an examples of the former, and video conference is an examples of the latter. We consider live streaming such video transmission method to improve quality afterwards.

2.1.1.1 On-Demand Streaming

The most obvious difference between on-demand streaming and live streaming is that the time sensitivity of the former is less than the latter, and video server has such file to transmit. Different from traditional file downloading approach, receiver begins to play streaming before it finishes downloading whole file, and the progress of playing relates with how percentage the video file downloads. RTSP (Real Time Streaming Protocol, RFC 2326, [16]) is an existent protocol to make a rule in application level, to provide an extensible framework to enable controlled, on-demand delivery. Because of this protocol, we can press play, stop, pause bottoms to control video streaming. Not like live streaming, the changeable network condition does not seriously impact video quality when using on-demand streaming, at most audience does not feel comfortably when media player pauses playing because of playing procedure catches download procedure, and continues playing because of download procedure holds time to continue receiving video packets during media player pause playing.

2.1.1.2 Live Streaming

Previously, the only way to watch live baseball games was using TV if people did not go to the ball field, but now they have more than one choice to achieve the goal. Some streaming providers forward live streaming in Internet to let people be convenient to watch live game. Because the duration limitation from encoded to playout is much strict, live streaming is very sensitive to the network condition. For above reason, packet loss, delay, and jitter such network characteristics will impact the quality of video. Because the duration between event happens and event played is short, it is not very efficient to retransmit. In video conference case, it even only allows hundreds milliseconds delay from encoded, transmitted to playout, and thus the network condition becomes the main reason to cause unstable video quality.

2.1.2 Video Quality Evaluation Tool

There are some tools to measure video quality, and it means that how degree does the coding parameters and network condition impact the quality. The most common used tool is PSNR (Peak Signal-to-Noise Ratio), to compare every pixels between original and playout picture. And there is another tool named DoP, to measure the total latency and playout quality. The score of DoP is quite important to prevent the playout tool from just delaying the playout time to make frames received correctly but ignoring audience requirements and feelings in live streaming.

2.1.2.1 MSE and PSNR

We have already known that video is composed of pictures, so we can use the same way to measure video quality: one frame one score. When PSNR is higher, it means that the two pictures are more similar, and the difference between them is smaller.

PSNR uses MSE (Mean Square Error) to measure the error degree between them:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(I\left(i;\, j\right) - K\left(i;\, j\right) \right)^2, \tag{2.1}$$

where $I\left(i;\, j\right)$ means the value of the original pixels, $K\left(i;\, j\right)$ means the value of the compared pixels, and there are $m \times n$ pixels per frame. PSNR uses this value to be the noise and do the calculation:

$$PSNR = 10 \times \log \left(\frac{MAX_I^2}{MSE} \right). \tag{2.2}$$

Here, MAX_I is the maximum possible pixel value of the image. If the pixels we calculating are represented using 8 bits per sample, the value of MAX_I is 255. So the maximum value of MSE is 255^2 and thus PSNR is zero. There are many signals have a very wide dynamic range, so it is the reason that why PSNR is usually expressed in terms of the logarithmic decibel scale.

2.1.2.2 DoP and VDoP

There is a shortcoming of PSNR which just measures the difference between original and playout frames because it does not have concept of time when we consider live streaming. Audiences want both playout rate deviation as small as possible, in high video quality requirement. And that is the purpose of the DoP (Distortion of Playout) and VDoP (Variance of Distortion of Playout) [17]. DoP and VDoP derive from VoD (variance of discontinuity) [18], which captures the disruptive effect of underflow and of intentionally introduced gaps. DoP and VDoP are used to capture the effects on users perceived quality. In this thesis, the main target of DoP is to take the effects of playout rate deviation, underflow and packet loss for some reasons into consideration concurrently because we will use AMP to improve the video quality impacted by network, and we use the same definition as [19] which also use this tool to compare the purposed AMP with other AMPs.

The DoP is defined by[1]

$$DoP\,(i) = \begin{cases} \left|t^{RAN} + t_{\mathtt{j}} - \mathbb{T}\right|, \; if\ 1^{st}\ frame\ of\ a\ certain\ preroll\ is\ played, \\ \left|t_{\mathtt{j}} - \mathbb{T}\right|, \; if\ \mathtt{j}^{th}\ frame\ is\ played, \\ \mathbb{T}, \; if\ the\ frame\ is\ lost, \end{cases} \tag{2.3}$$

where t^{RAN} is a random variable between the range from buffer underflow to the first new playout, and $t_{\mathtt{j}}$ is the actual playout interval between $\mathtt{j}^{th}$ frame and $(\mathtt{j}+1)^{th}$ frame, and $\mathbb{T}$ means the normal playout duration. In this thesis, we set the value of $\mathbb{T}$ is 40 milliseconds when playout rate is 25 frames per second.

And then we could define the meaning of DoP and variance of VDoP:

$$\overline{DoP} = \frac{\sum_{\mathtt{j}=1}^{N^{LOSS}+N^{TOL}} DoP\,(\mathtt{j})}{N^{LOSS}+N^{TOL}}, \quad VDoP = \frac{\sum_{\mathtt{j}=1}^{N^{LOSS}+N^{TOL}} DoP^2\,(\mathtt{j})}{N^{LOSS}+N^{TOL}} - \left(\overline{DoP}\right)^2, \tag{2.4}$$

where N^{LOSS} is the number of frame loss and N^{TOL} is the number of playout frames.

If our proposed AMP were good enough, the main element to let $DoP\,(\mathtt{j})$ increase only remaining $|t_{\mathtt{j}} - \mathbb{T}|$. Because AMP delays time to prevent buffer underflow or hurries time to prevent buffer overflow, it will cause this value varying in contrast to normal playout way and worse VDoP, and it will become a main topic we have to solve.

In this thesis, we both want to get a low VDoP and high PSNR score at the same time, and our purposed AMP will consider about this two phases to make the control decision. We know that the way to prevent PSNR being low is that we have to let the number of received frames as many as possible, and the way to prevent DoP and VDoP being high is that we have to achieve a balance between the following goals. The first goal is to make the playout rate smooth and close to original playout rate. Another goal is to keep the media player always having frames to playout.

[1]The definition of DoP in [19] uses “packet” to be a unit, but actually it is more significant if using “frame” to be a unit.

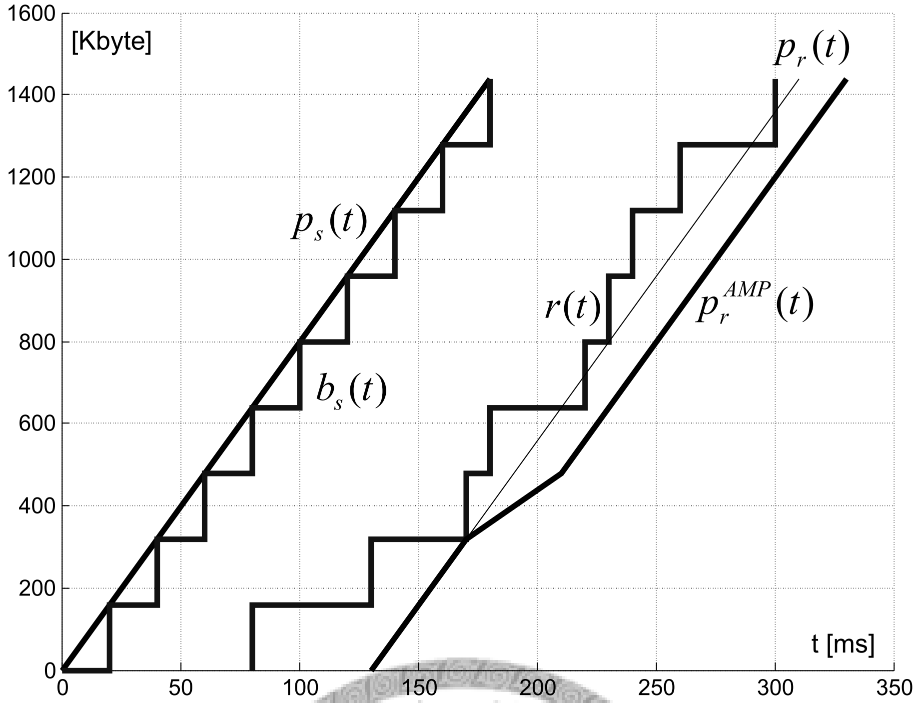


Figure 2: A Sketch of AMP profit: changing the third frame playout time to prevent frame dropped due to timeout [1]

2.2 Network-Based Improvement Techniques: AMP and Roll-off Function

2.2.1 Adaptive Media Playout

Adaptive media playout (AMP) is one of approaches to adapt changeable channel condition. The main idea of AMP is to change playout frame interval but it is an imperceptible change in audience feeling. The media player could use it to postpone buffer underflow happened when it playout slower, or to postpone buffer overflow happened when it playout faster. Figure 2 is a sketch of AMP which could explain the benefit of AMP. From [20], it indicates that from informal subjective tests, and it slows that people often would unnoticeable even if playout rate of video is up or down 25%. From the information above, it means that we could increase playout rate to 1.25 times the original rate or decrease playout rate to 0.75 times the original rate. If we set initial frame playout interval is $\mathbb{T}$ in default, AMP could adjust the playout interval from $^{4\mathbb{T}}/_{3}$ to $^{4\mathbb{T}}/_{5}$ and the audience could not notice speed change.

AMP control is useful in many cases and we are easily to imagine that it will get a good video quality even in bad channel condition if designed well, because it makes loss packet's have more time to let receiver get correct packet before its playout. So we could also image that if we add AMP element in our live streaming media player, we will get a much higher video quality and let our sight more comfortable comparing with other control approaches. AMP control approaches can be omnipresent, and it will reduce buffer overflow or underflow event when media player finds out the buffer has such trend by using its prediction mode, and then it uses AMP to reduce the phenomena.

2.2.2 Roll-off Function

To reduce VDoP, we use roll-off function to get the job. [19] wants to reduce the VDoP when using AMP mechanism, and it uses the quadratic function rather than the linear function to adjust the playout rates. The author refers to the number of packets in playout buffer to adjust the playout rate. Similarly, we use this way to create our roll-off function, which refer to the relation between $\mathtt{t}$ and $\mathtt{P}$. Below is an example of the roll-off function, which relates to AMP execution strength and the roll-off execution duration:

$$R(\mathtt{t}) = \mathtt{P} - \left(\frac{\mathtt{t}^{stop} - \mathtt{t}}{\mathtt{t}^{stop} - \mathtt{t}^{start}}\right)^2 \times \mathtt{P}, \tag{2.5}$$

which $\mathtt{t}^{start}$ and $\mathtt{t}^{stop}$ are the roll-off function start and stop execution time, and $\mathtt{P}$ is AMP execution strength. There is an obvious difference between [19] and our proposed roll-off function. The reasons and the results will show in Chapter 5. And in the extension part of the thesis in Chapter 6.1, we will design roll-off function for minimizing VDoP after we find out a large frame delay gap happened in some special events. We will keep the balance between frame storing efficiency and VDoP required by the MN.

There are some rules for our function design listed below:

- The playout interval has to strictly increasing or decreasing.
- AMP will store or release an additional frame exactly in the end of roll-off execution in default.
- The time duration of roll-off execution has to be within a range.

Next, we introduce and describe linear roll-off function simply here, which is a special case of polynomial roll-off functions:

Figure 3 is a sketch of linear roll-off function, which describes the relation between the function execution duration and AMP execution strength. Linear roll-off function

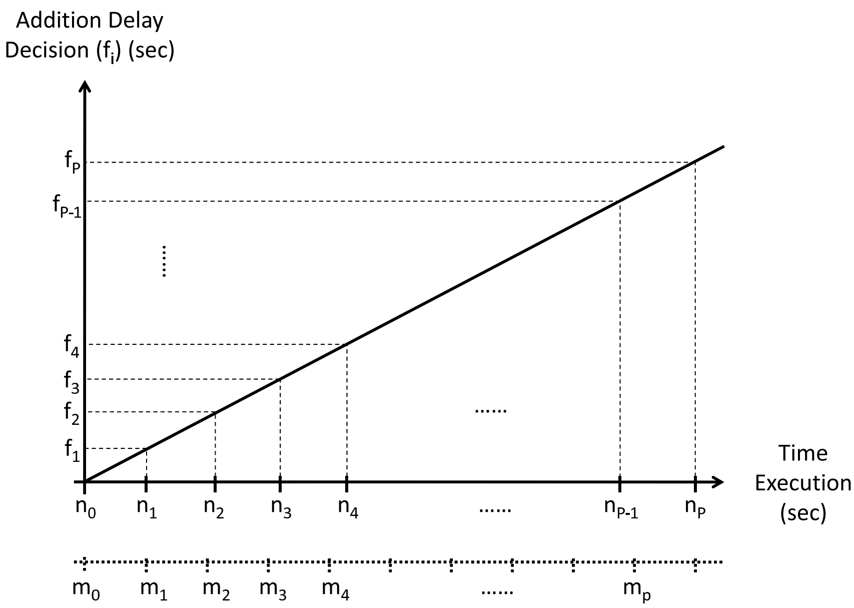


Figure 3: A sketch of linear roll-off function

will choose f_i to be the current AMP execution strength when it playout i^{th} frame from n_i to n_{i+1} time. We suppose that we begin to do the roll-off immediately to simply the situation after we receive and playout the $\mathtt{j}^{th}$ frame and set the received, playout time and additional playout delay be m_1, n_1, and f_1, respectively. Because f_1 is 0 in the beginning of roll-off execution, the playout time of the next frame (n_2) is not impacted by f_1 and thus n_2 is equal to m_2 again. We could get a rule that the place of n_{i+1} is impacted by n_i and f_i. In other words, when the MN in time n_i prepares to playout i^{th} frame, it will refer to f_i and decide the playout speed, and the decision will impact the playout frame interval and the next playout time (n_{i+1}), so that $n_{i+1} = \mathbb{T} + n_i + f_i$.We define $f_2 = \mathcal{K}$ to simplify the functions representation. Because the function is linear, the slope coefficient is all the same everywhere, and we could write $\frac{f_2}{n_2} = \frac{f_3}{n_3}$, so $\frac{\mathcal{K}}{\mathbb{T}} = \frac{f_3}{2\mathbb{T}+\mathcal{K}}$. We could use this relation to get $f_3 = 2\mathcal{K} + \frac{\mathcal{K}^2}{\mathbb{T}}$, and we also could use this method to get $f_4 = 3\mathcal{K} + \frac{3\mathcal{K}^2}{(\mathbb{T})^1} + \frac{\mathcal{K}^3}{(\mathbb{T})^2}$ and other f_i. Because there are some relations between each n_i, thus we could find a rule like:

$$n_i = \mathbb{T} \times i + \sum_{\eta=1}^{i} f_\eta, \tag{2.6}$$

and we already know that

$$m_i = \mathbb{T} \times i, \tag{2.7}$$

which means the i^{th} frame in roll-off zone received time in ideal case, so we could

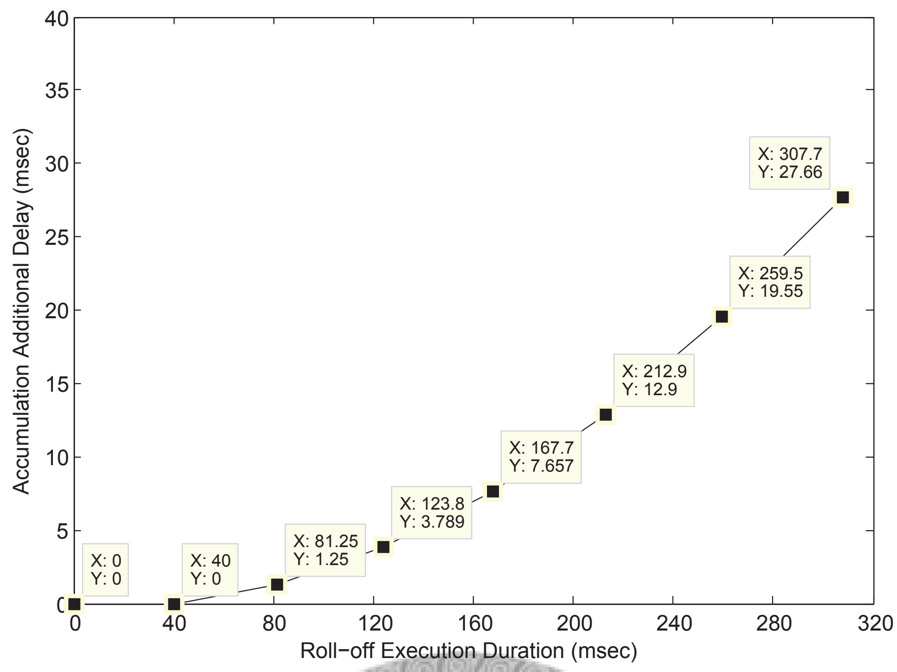


Figure 4: The relation between roll-off execution duration and accumulation additional received-to-playout delay

directly get the $DoP\,(i)$ in separate playout frames after knowing from f_1 to f_i.

Because we know that $n_i - m_i = \sum_{\eta=1}^{i} f_\eta$ and when $\sum_{\eta=1}^{i} f_\eta$ is equal to $\mathbb{T}$, the roll-off will finish and store an additional frame. We could plot the relation between the linear roll-off function execution duration and the "accumulation" additional delay between received and playout frame like Figure 4. From the figure, we could find out that it should not achieve the goal in the end of roll-off, neither we could not promise that it is the minimum VDoP. By using the similar works, we could get a suitable roll-off function in different kinds of situations.

2.3 Characteristic of Heterogeneous Handover

Nowadays, WLAN (Wireless Local Area Network) [21], MAN (Metropolitan Area Network) [22] and WAN (Wide Area Network) [23] overlap with each other in many cities. When a mobile user has at least two in three kinds of network interfaces, it has to face with heterogeneous handover between the different networks when the interfaces support this technique. Although many papers mention about the "seamless vertical handover" technique and declare that they achieve this goal between heterogeneous networks by considering some elements to enhance the handover performance, like using MIH[2] and mobile IPv6 (MIPv6), to do the handover [25–30]. Their goal is to reduce the handover delay and the packet loss, delay, during handover, by modifying MAC Layer and interaction between MIH and other network layers. Contrarily, we do not want to modify the network layers except Application Layer. We just monitor their behaviors to be our proposed AMP references and do a series of decisions to achieve our goal: preventing video interruption during handover procedure.

We suppose a mixed phenomena to design our AMP for build a more complex and true loss model. We replace simple packet loss model, like random uniform model and Gilbert-Elliott model used in most common AMP papers. We use propagation shadowing model to be our main packet loss reason in common situation mentioned in previous section, and we use heterogeneous handover condition to be our burst packet delay reason in special situation. The definition of bust delay is that several packets delay to receive, and the delay time is above several hundreds of millisecond or even several seconds. Our scheme is that, a mobile node is in WMAN/WWAN range and receiving video streaming all the time, and it goes through WLAN range and doing heterogeneous handover, considering shadowing network in the whole of situation.

We focus on WMAN/WWAN handover to WLAN and WLAN handover back to WMAN/WWAN two events, where will suffer from packet burst delay and link interruption between the transmitter and receiver. We hope that our proposed AMP scheme could reduce the handover impact and make audiences are not aware of video interruption. By using NIST WiMAX NS-2 module, we could simulate the scenario and reveal the phenomena during the handover like Fig. 5, which shows that the streaming interruption occurs when the mobile node handover from WiFi AP to WiMAX base station. From the figure, we could find out that there is no packet received in the handover duration.

2.4 *Some Representable Approaches to Solve Video Interruption during Handover*

Some papers have already discussed about how does handover impact video streaming and already had some solution approaches. We could divide the solution ways

[2]Media Independent Handover, also named IEEE 802.21, is used to enable handover and interoperability between heterogeneous network types including both 802 and non 802 networks. [24]

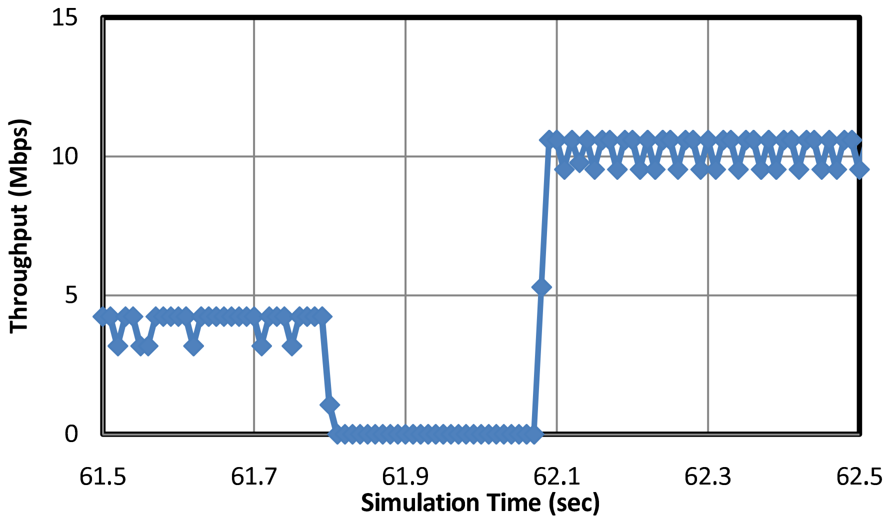


Figure 5: The throughput between CBR traffic server and mobile node while handover from WiFi to WiMAX by using hard handover approach

into two parts: one part is based on lower layers like Transport Layer, Routing Layer, MAC Layer, and another part is based on Application Layer, by monitoring the lower layers behavoirs.

Some solution ways based on network layers except Application Layer are listed below examples. Many papers have already discussed about how to achieve "seamless handover" to reduce packet loss, delay and jitter and thus we do not mention about the topic here. [31] modifies TCP-friendly rate control to get a higher QoS and throughput supports. [32] makes MIH collect statistics from Physical, Network and Application Layers, and update the video encoding parameters in real time.

Some solution ways based on Application Layer are listed below examples. [33] models handover spending time and feedback concept to adapt the playout rate during vertical handover; it estimates the required pre-buffering size by referring both handover duration and transient packet losses. [34] adapts the rate during vertical handover to provide uninterrupted streaming sessions to mobile terminals. [35] uses RSSI monitoring to predict handover time, which is an important information to decide playout rate in the thesis. In VoIP territory, [36] does a proper reconstruction of continuous playout speech is achieved by scaling individual voice packets using a time-scale modification technique, based on the Waveform Similarity Overlap-Add (WSOLA) algorithm. [37] uses AMP in audio packet, based on whether handover event is detected or not. [38] refers other layers information to decide AMP.

2.5 Conventional Adaptive Media Playout Mechanism

Conventional adaptive media playout mechanisms could reduce the user-perceived latencies causing for the characteristics of network, and it allows clients to buffer several packets to produce less delay to keep playout reliability [1]. AMP has some functions to prevent buffer overflow and underflow, and it also could reduce initial playout delay. It could observe current throughput, channel condition, and available bandwidth to estimate frame arrival rate and buffer state to decide playout speed, and thus it could conceal packet delay events to make the audiences unnoticed.

Although some good thinking about AMP already proposed, most of them still use simple models and cases to do the simulation and analysis results; [20, 39] have some special but not so reasonable hypotheses to simplify their simulation platforms and the simulation objects. For example, the number of packet segments is the same between different I-frames and P-frames from the former proposed, and the round-trip-time (RTT) is constant from the latter proposed; [40] adds wireless element to the simulation model and uses AMP to optimize video rate-distortion, but it still does not consider the characteristics of video streaming; [41] propose a concept of playout smooth to improve visual quality but cannot be used in live streaming. [33] estimates handover latency and packet loss by mathematical model, and it also uses feedback approach to do AM, but the model still has some spaces to improve.

The summary of the shortcomings is below, and our goal is to deal with the problem and design a better AMP module.

- Although packet delay and jitter is unpredictable in their network environment, AMP approach still could be replaced with other tools like FEC/ARQ, etc.

- Although some papers already propose handover prediction time and duration mechanisms based on model and the signal power, they do not consider the mobile node mobility, which will impact handover time.

- The simulation platform and environment settings are simple, and even some of papers just consider non real-time video streaming, so they do not set artificial end-to-end delay constraint.

AMP is able to solve some events happened in general network cases, like sporadic packets loss, jitter and delay. We also have already known that the conventional AMP and some approaches could not prevent video interruption event or reduce

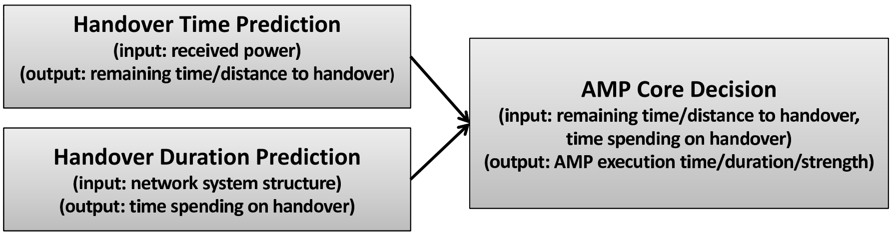


Figure 6: A sketch of proposed AMP

the duration efficiently in packet burst delay cases such as heterogeneous handover procedure, and the event certainly exists in normal network condition.

2.6 Proposed AMP Description

AMP is an efficient concept able to reduce packet loss, delay and jitter effects, because it could adjust video playout rate to postpone frame deadline. We could extend the concept and modify it to fit in the network environment.

We have already known the network environment will cause scattered packet delay due to shadowing network model and burst packet delay due to heterogeneous handover. Our proposed AMP has to aim at video interruption avoidance but also has to consider shadowing network effect. It has to add some additional functions to detect or predict the event happened time and duration to decide AMP execution time, strength and duration. Thus, we use handover time prediction part and handover duration prediction part to collect the information from network environment, and then we use AMP core algorithm part to decide the time to execute AMP and its execution strength.

The sketch of Proposed AMP is in Figure 6.

Chapter 3

VERTICAL HANDOVER TIME AND DURATION PREDICTION

From the previous chapters, we have already known that heterogeneous handover is one of the cases which would cause burst video packet delay while receiver getting live video streaming, and we also knew that AMP is one of efficient tools could reduce the effect. But we could not find an useful AMP to fit for this network condition. Therefore, we start to design our AMP which have powerful capability to face this situation and solve the problem. In this chapter, we propose a mechanism which could hold the essential handover information to be the input of core AMP decision part. We will detail AMP decision part in the next chapter.

Different from conventional AMP simulation environment, we use propagation shadowing channel network model replacing Markov chain to be the source of packet loss; we seriously discuss the effects of network and the receiver conditions on AMP execution strength decision.

In Section 3.1, we will do a series of jobs to predict remaining time and distance to handover. In Section 3.2, we will discuss the handover duration in different types of networks. In Section 3.3, we will construct a feedback mechanism between the media server and receiver to allow the former to get the handover information and delay frames transmission during the handover if needs.

We could use Figure 7 to show the simplified relation between two outer information gainer and AMP core decision. When MN enters WLAN AP signal range, the flow chart starts from the left side. The shading in the rectangle designates the beginning state; the backward diagonal lines in the rectangle designates handover time prediction in Section 3.1, the forward diagonal lines in the rectangle designates handover duration in Section 3.2, and the grid lines in the rectangle designates AMP decision in Chapter 4.

3.1 Handover Time Prediction

There are some parameters will be used in this section, already listed in Table 1. we will know about how to get the distance between the BS and MN from Section 3.1.1,

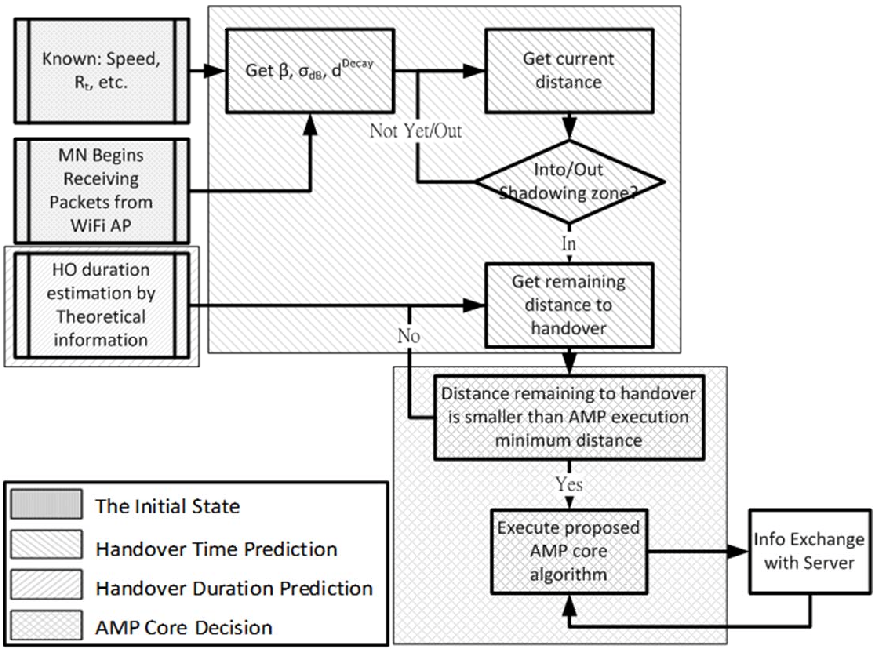


Figure 7: AMP whole flow chart

and then we will make a reason to do the handover time prediction. The objective of the handover time prediction part is to precisely estimate the MN remaining time and distance to handover. We focus on handover time prediction when the MN is in WiFi signal range, to predict when does MN handover from WiFi back to WMAN/WWAN [42].

Additionally, there is an important thing to mention before we starting this section. We have to give special emphasis to some figures when we use statistical viewpoint from samples. The source of these samples is caught from NS-2 MAC Layer, and we temporarily use CBR (constant bit rate) traffic to replace video traffic temporarily. When the MN and WiFi BS exchange MAC information with each other, there is also a recorder in MAC Layer copying the received power to a file, and it also records the source and destination of every event, and it is the main reason that the number of samples such many. The simplified simulation topology is in Figure 8.

3.1.1 Network Environment Settings

In this section, we will introduce the network environment settings and will be considered in the remaining sections.

We use propagation shadowing network model to be our network environment. We list some important parameters which be used in channel propagation model in

Notation	Description
$speed_{\mathtt{i}}$	: The MN "*efficacious*" speed between receiving $\mathtt{i}^{th}$ packet and $(\mathtt{i}+1)^{th}$ packet, related to $speed_{\mathtt{i}\parallel AP}$ ("*subjective*" speed) and $speed_{\mathtt{i}\perp AP}$ ("*invalid*" speed) (m/sec)
$d_{\mathtt{i}}^{RECV}$	: The distance between WiFi AP and MN after received $\mathtt{i}^{th}$ packet (m)
$d_{\mathtt{i}}^{ACT}$	: The actual distance between WiFi AP and MN after received $\mathtt{i}^{th}$ packet (m)
$d_{\mathtt{i}}^{STA}$	: The statistical distance between WiFi AP and MN after received $\mathtt{i}^{th}$ packet (m)
$d_{\mathtt{i}}^{CAL}$	: The simplification distance calculation between WiFi AP and MN after received $\mathtt{i}^{th}$ packet from $d_{\mathtt{i}}^{RECV}$ (m)
R_t	: The range of the MN able to receive packet transmitted by the WiFi AP (m)
$t_{\mathtt{i}}^{HO}$	: The remain time to handover after received $\mathtt{i}^{th}$ packet (sec)
$d_{\mathtt{i}}^{HO}$	: The remain distance to handover after received $\mathtt{i}^{th}$ packet (m)
$t_{\mathtt{i}}^{Decay}$	: The remain time when packet loss happens due to shadowing model settings after received $\mathtt{i}^{th}$ packet (sec)
$d_{\mathtt{i}}^{Decay}$	: The distance between WiFi AP and MN when packet loss happens due to shadowing model settings after received $\mathtt{i}^{th}$ packet (sec)
$t^{Duration}$	: Handover duration estimation (sec)

Table 1: Parameter settings for handover time prediction

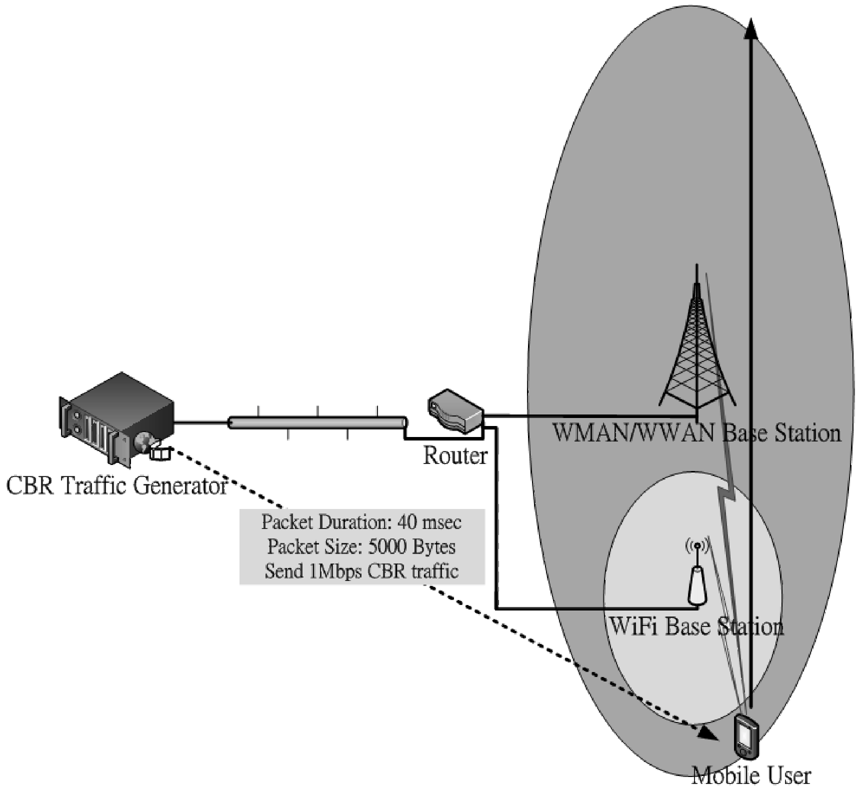


Figure 8: A Schematic of the MN receiving CBR traffic from media server through WiFi or WiMAX BS.

Table 2, and Table 3, [43].

We have already known that the log-normal shadowing model is like Equation 3.1:

$$\left[\frac{P_{r,i}(d)}{P_{r,i}(d_0)}\right]_{dB} = -10\beta\log\left(\frac{d}{d_0}\right) + \chi_{dB}. \tag{3.1}$$

Here χ_{dB} means the Gaussian random variable with zero mean and standard deviation σ_{dB}, or we could rewrite to $N(0, \sigma_{dB})$ replaced χ_{dB}. About the typical values of β and σ_{dB} used in some simulations are in Table 4. About Equation 3.1, we could separate it to the following elements:

$$P_{r,\mathtt{i}}(d) = Pr0 \times 10^{\frac{powerloss\_dB}{10}}, \tag{3.2}$$

where

Notation	Description
β	: Path loss exponent
dist0	: Reference distance, typically is 0 [44](=d_0) (m)
σ_{dB}	: Shadowing derivation

Table 2: Parameter settings in shadowing propagation model

Notation	Description
P_t	: Transmitted power signal strength (Watt)
$P_{r,i}(d)$	: Received power signal strength of i^{th} packet (Watt)
G_t	: Antenna gain of the transmitter
G_r	: Antenna gain of the receiver
L	: System loss
d	: Transmitter-Receiver (T-R) separation distance (m)
λ	: The communication wavelength (=c/f) (nm)

Table 3: Parameter settings in free space propagation model

$$\begin{cases} Pr0 = P_{r,\mathrm{i}}\left(dist0\right), \\ P_{r,\mathrm{i}}\left(d\right) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L^2}, \\ powerloss\_dB = avg\_dB + N\left(0, \sigma_{dB}\right), \\ avg\_dB = -10\beta\log\left(\frac{d}{d_0}\right). \end{cases} \tag{3.3}$$

In general, the receiver could get G_t, G_r,P_t, λ, L, and *dist0* such values to calculate *Pr0*, and they are constants if wireless nodes continuing communication with each other using a changeless communication frequency band.

We suppose that about the shadowing model in this simulation environment, all the parameter values are known by receiver except β, σ_{dB}, and d; β and σ_{dB} are two

Environment		β
Free space	:	2
Shadowed urban area	:	[2.7,5]
Line-of-sight	:	[1.6,1.8]
Obstructed	:	[4,6]

(a) Some Typical Values of β

Environment		$\sigma_{dB}(dB)$
Outdoor	:	[4,12]
Office, hard partition	:	7
Office, soft partition	:	9.6
Factory, line-of-sight	:	[3,6]
Factory, obstructed	:	6.8

(b) Some Typical Values of σ_{dB}

Table 4: Typical values of path loss exponent and shadowing deviation

constant unknown values [45]. Because here d is distance between the base station and the mobile node, and the mobile node may move or static at any time, so d is not a regular value. We know that d is an important value to decide handover remaining time. We will use statistical approach to get β, σ_{dB}, and d in the next section and in Section *5.3* simulation part.

In Figure 9, it is a sketch which shows the relation between the free space propagation model and the shadowing propagation model (or we can say that free space model is a special case of shadowing model which $\beta = 2, \sigma_{dB} = 0$). The x-axis is the node distance against with the transmitter, and the y-axis is the power distribution and it shows in logarithm.

After describing about some characteristics of handover due to shadowing propagation model, we know that the $P_{r,i}$ reductions is the main factor to let the MN doing the handover because it impacts the condition of AP beacon receiving. Some other factors will also impact the trend and distribution of $P_{r,i}$, like β, σ_{dB}, d, *speed*, etc, but the MN does not know any parameter values except *speed*, so here it starts to make a plan to get the remain handover time and distance. First, it tries to get the network environment values: β and σ_{dB} , and then it tries to get the distance between the WiFi AP and MN.

3.1.2 Getting β and σ_{dB} Using Statistical Approach

From Equation 3.2, we got the relation between the signal power and the distance between the transmitter and the receiver, and then we could arrange the equation to

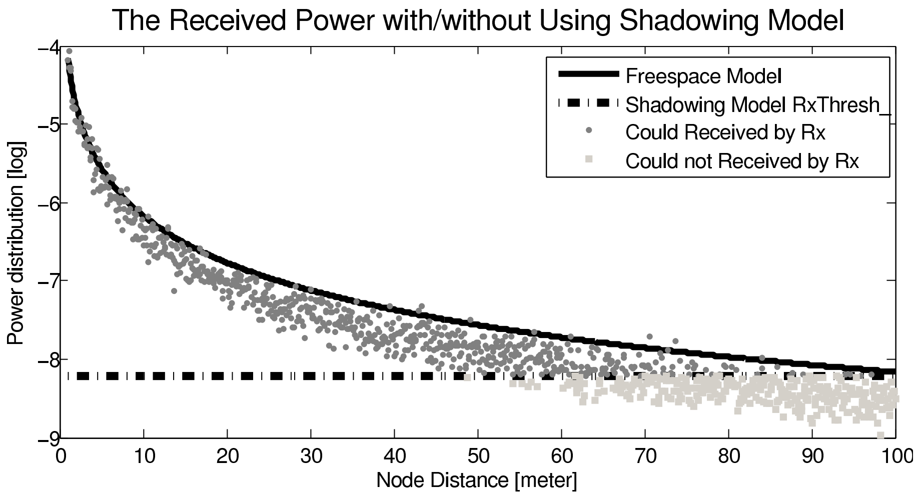


Figure 9: Received power is impacted/not impacted by shadowing network

Equation 3.4:

$$\log\left(P_{r,\mathtt{i}}(d)\right) = -\beta\log\left(d\right) + \log\left(Pr0\right) + 0.1 \times N\left(0, \sigma_{dB}\right). \tag{3.4}$$

It means that when we know the distance between the WiFi AP and MN (i.e., $d = d_{\mathtt{i}}^{ACT}$), we put the other known parameter values to Equation 3.4 and get the $\log\left(P_{r,\mathtt{i}}(d)\right)$. The receiver will check whether this value whether is larger than $\log\left(RX\_Thresh\right)$ or not to be able to receive. Here we define the WiFi AP transmission range (R_t):

$$R_t = 10^{\frac{-\log\left(RX\_Thresh\right)+\log\left(Pr0\right)}{\beta}}, \tag{3.5}$$

We also know that if the channel is not impact by shadowing deviation (the last element in Equation 3.4), we could get the corresponding, one-on-one $P_{r,\mathtt{i}}(d)$, and vice versa. Here we change the symbol in this special case from $P_{r,\mathtt{i}}(d)$ to $P_{\beta,\mathtt{i}}(d)$, which does not consider $N\left(0, \sigma_{dB}\right)$ element. So we get the conclusion that the actual $P_{r,\mathtt{i}}(d)$ and d will be in a range of:

$$\begin{aligned} &P_{\beta,\mathtt{i}}(d_{\mathtt{i}}^{ACT}) \times 10^{-0.1\times2\sigma_{dB}} \leq P_{r,\mathtt{i}}(d_{\mathtt{i}}^{RECV}) \leq P_{\beta,\mathtt{i}}(d_{\mathtt{i}}^{ACT}) \times 10^{0.1\times2\sigma_{dB}} \\ &\rightarrow d_{\mathtt{i}}^{ACT} \times 10^{\frac{-0.1\times2\sigma_{dB}}{\beta}} \leq d_{\mathtt{i}}^{RECV} \leq d_{\mathtt{i}}^{ACT} \times 10^{\frac{0.1\times2\sigma_{dB}}{\beta}}. \end{aligned} \tag{3.6}$$

by using three-sigma rule, respectively, where $d_{\mathtt{i}}^{RECV}$ means the probable result of distance after calculating when knowing the $\mathtt{i}^{th}$ packet received power and the value of path loss exponent which could be written as:

$$d_{\mathtt{i}}^{RECV} = 10^{\frac{-\log\left(P_{r,\mathtt{i}}(d)\right)+\log(Pr0)+0.1\times N\left(0,\sigma_{dB}\right)}{\beta}}. \tag{3.7}$$

So we know that there are about 95.45% of received signal ($P_{r,\mathtt{i}}(d_{\mathtt{i}}^{RECV})$) is in the bound between $P_{\beta,\mathtt{i}}(d_{\mathtt{i}}^{ACT}) \times 10^{0.1\times 2\sigma_{dB}}$ and $P_{\beta,\mathtt{i}}(d_{\mathtt{i}}^{ACT}) \times 10^{-0.1\times 2\sigma_{dB}}$. Because the expected value of $N(0, \sigma_{dB})$ element in this case is 0, we assume that we will collect enough number of $d_{\mathtt{i}}^{RECV}$ samples to do the statistic calculation and let Equation 3.7 could be written to 3.8:

$$d_{\mathtt{i}}^{CAL} = 10^{\frac{-\log(P_{r,\mathtt{i}}(d))+\log(Pr0)}{\beta}}. \tag{3.8}$$

And we could easy to find that the unknown parameters only remain β to get the $d_{\mathtt{i}}^{CAL}$. Now we suppose that the initial value of β is 2 from Table 4 in this condition. We continue using this method to collect the packets after $\mathtt{i}^{th}$ packet and get the series statistical distances ($d_{\mathtt{i}}^{CAL}$, $d_{\mathtt{i}+1}^{CAL}$,..., $d_{\mathtt{i}+N}^{CAL}$), and then we use SLRM (Simple Linear Regression Model) doing linear regression analysis to get the relation between the distance and the received time, and the result is like Equation 3.9[1] where $d_{\mathtt{i}}^{STA}$ is the calculated distance given $t_{\mathtt{i}}$. To simplify the regression calculation, we suppose that the speed of MN keeps a stable value except zero when sampling.

$$d_{\mathtt{i}}^{STA} = a_{\mathtt{i}} \times t_{\mathtt{i}} + b_{\mathtt{i}}. \tag{3.9}$$

From Equation 3.9, we could begin trying to plot the relation between the $t_{\mathtt{i}}$ and $d_{\mathtt{i}}^{STA}$ after knowing $a_{\mathtt{i}}$ and $b_{\mathtt{i}}$ values using linear regression. We also know that the slope coefficient means the mobile speed which the x-axis is time, y-axis is distance between WiFi AP and MN. When the slope is smaller than 0, it means that the MN is approaching to the WiFi AP, and when the slope is larger than 0, it means that the MN is leaving from the WiFi AP. We could use the difference between the speed[2] and $a_{\mathtt{i}}$ to modify β to let $a_{\mathtt{i}}$ value be closer to the speed and get the right series statistical distances ($d_{\mathtt{i}}^{STA}$, $d_{\mathtt{i}+1}^{STA}$,..., $d_{\mathtt{i}+N}^{STA}$).

The concept of adjusting β to let the speed slope and the linear regression slope almost the same is that, if the β value we give in ($d_{\mathtt{i}}^{CAL}$, $d_{\mathtt{i}+1}^{CAL}$,..., $d_{\mathtt{i}+N}^{CAL}$) is as the same as the β value the propagation model, after we do the regression analysis to except the Gaussian effect from ($d_{\mathtt{i}}^{CAL}$, $d_{\mathtt{i}+1}^{CAL}$,..., $d_{\mathtt{i}+N}^{CAL}$), and it should be the actual route the MN going through like Figure 10, and obviously it does not have any relation between β and the MN speed. But from the limitation of regression and the characteristics of distribution, we would not find a suitable β to be close with real value when the

[1]To simplify the calculation, we use least squares to solve the problem.

[2]About the speed when MN receives packet i, $speed_i$, we suppose that $speed_{i\|AP}$ is equal to $speed_i$ temporary, to simplify the distance calculation. When the MN is further away from WiFi AP, the value of $speed_{i\|AP}$ is closer to $speed_i$.

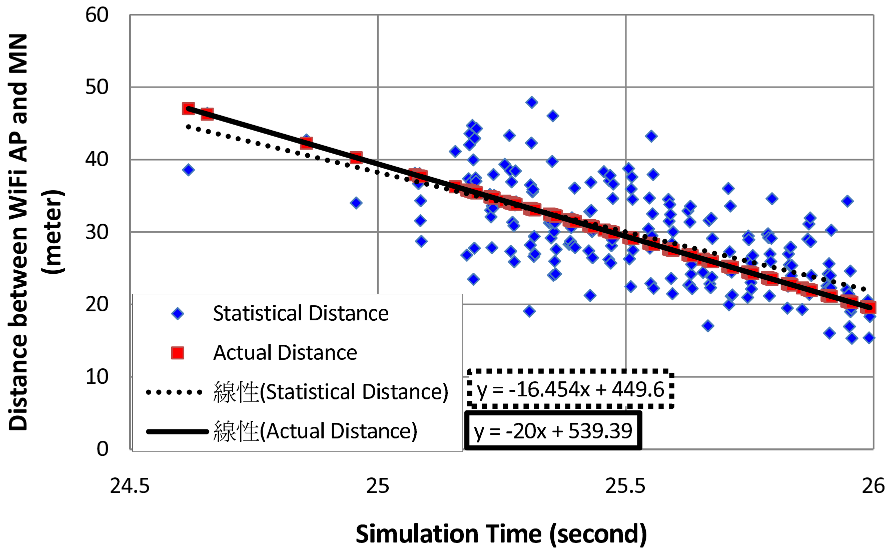


Figure 10: Slope comparison between the MN speed and the linear regression

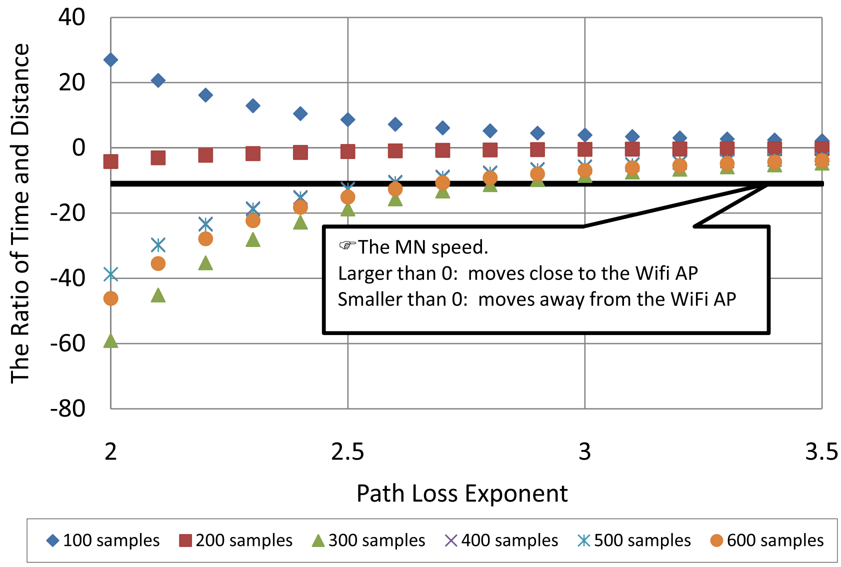


Figure 11: Path loss exponent guessing vs. the ratio of time and distance between WiFi AP and MN in different number of samples

Sample Numbers	β (Slope)	Between Actual β(%)
100	None	None
200	None	None
300	2.8(-11.3)	3.5%
400	2.6(-10.43)	3.5%
500	2.6(-10.74)	3.5%
600	2.7(-10.71)	0%

Table 5: Difference between calculated and actual β in different sample numbers

number of samples is too small, like Figure 11 showing. Figure 10 is a sketch of the slope comparison between the MN speed and the linear regression, and the diamond in the figure means the distance got from Equation 3.8, which $P_{r,\mathtt{i}}(d)$ and $Pr0$ are already known, and we give β a value as the same as actual one. The point line means the linear regression result got from diamond points. We care about the $a_{\mathtt{i}}$ of its linear regression equation, and it shows that the slope between the linear regression equation and the actual distance equation is close. The thick line means the current speed value, and the points means the regression line's slope in different number of samples when we guess β be a value. The result of the difference between calculated and actual β in different sample numbers shows in Table 5.

After calculation, we could use this β in other samples to get more precise value, until it achieves Equation 3.10. To reduce the calculation times, we set the β is accurate to 1 decimal place:

$$\min\left(\frac{speed_{\mathtt{i}} - a_{\mathtt{i}}}{speed_{\mathtt{i}}}\right). \tag{3.10}$$

In summary, we could use Figure 11 to indicate the β finding process and the relation between actual distance ($d_{\mathtt{i}}^{ACT}$), received distance ($d_{\mathtt{i}}^{RECV}$), calculated distance ($d_{\mathtt{i}}^{CAL}$) and statistical distance ($d_{\mathtt{i}}^{STA}$). $d_{\mathtt{i}}^{ACT}$ is the actual distance between WiFi AP and MN, and $d_{\mathtt{i}}^{RECV}$ should not equal to $d_{\mathtt{i}}^{ACT}$ when MN receives power signal in shadowing environment. $d_{\mathtt{i}}^{CAL}$ is the simplified element from $d_{\mathtt{i}}^{RECV}$ and then we use regression analysis to except Gaussian effect and get $d_{\mathtt{i}}^{STA}$ in every given sample time. If we have many samples to calculate, we think that the $d_{\mathtt{i}}^{STA}$ is almost as the same as $d_{\mathtt{i}}^{ACT}$.

Here we start to get the σ_{dB}. We could use the difference between $d_{\mathtt{i}}^{STA}$ and $d_{\mathtt{i}}^{ACT}$ to get the σ_{dB}. We suppose that the $d_{\mathtt{i}}^{STA}$ is similar with $d_{\mathtt{i}}^{ACT}$. We could divide $d_{\mathtt{i}}^{STA}$

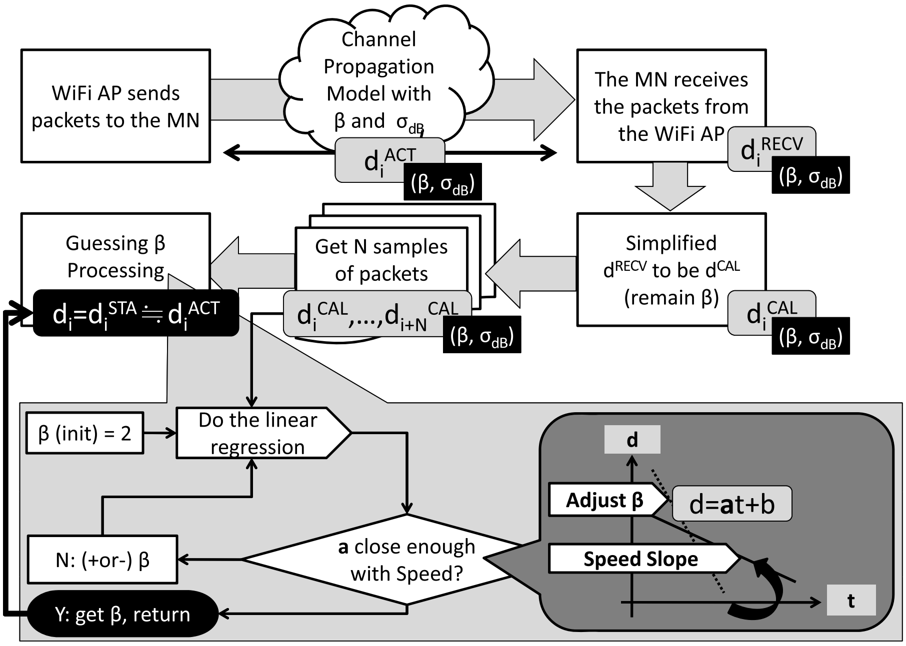


Figure 12: A simplified flow chart about getting β process

and $d_{\mathtt{i}}^{ACT}$ to be Equation 3.11, here β has already known.

$$\frac{d_{\mathtt{i}}^{STA}}{d_{\mathtt{i}}^{CAL}} = 10^{\frac{0.1\times N(0,\sigma_{dB})}{\beta}}. \tag{3.11}$$

From statistics, we known that in three-sigma rule [46], says that there are about 95.45% results are within $\pm 2\sigma_{dB}$ in this case, and we could use this rule to find a similar value to σ_{dB}. We have 95.45% confidence to say that if we have many samples from $N(0, \sigma_{dB})$ like $\upsilon_1,\ \upsilon_2,\ ,\ldots,\upsilon_{n-1},\ \upsilon_n$ and n is quite large, and thus they are satisfied that:

$$|\upsilon_i| \leq 2\sigma_{dB},\ i = (1,\ \ldots,\ n)\,. \tag{3.12}$$

It is same that there are about 95.45% samples fit above equation. So that if we have many pair of samples could be written as:

$$\left|10\beta\log\left(\frac{d_{\mathtt{i}}^{STA}}{d_{\mathtt{i}}^{CAL}}\right)\right|, \tag{3.13}$$

and we have 95.45% confidence to say that Equation 3.13 is equal or smaller than $2\sigma_{dB}$, and thus we could get the probable value of σ_{dB}, which equals in the last number of top 4.45% large value of the samples.

From Figure 13, it shows the difference between actual (using thick line to show) and statistical (using dotted line to show) σ_{dB}, and from Table 6 we reach a conclusion

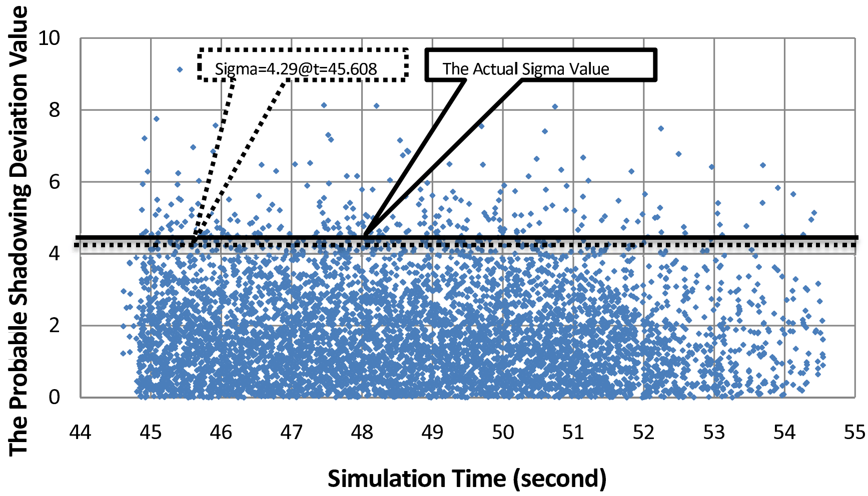


Figure 13: Difference between actual and statistical σ_{dB}

Sample Duration (second)	Probable σ_{dB}	σ_{dB} Difference (%)
0.50	4.318	4.04
1.00	4.410	2.00
1.50	4.367	2.95
2.00	4.367	2.95
2.50	4.273	5.04
3.00	4.286	4.75

Table 6: Relation between sample duration and probable σ_{dB}, and the difference between probable and actual σ_{dB}

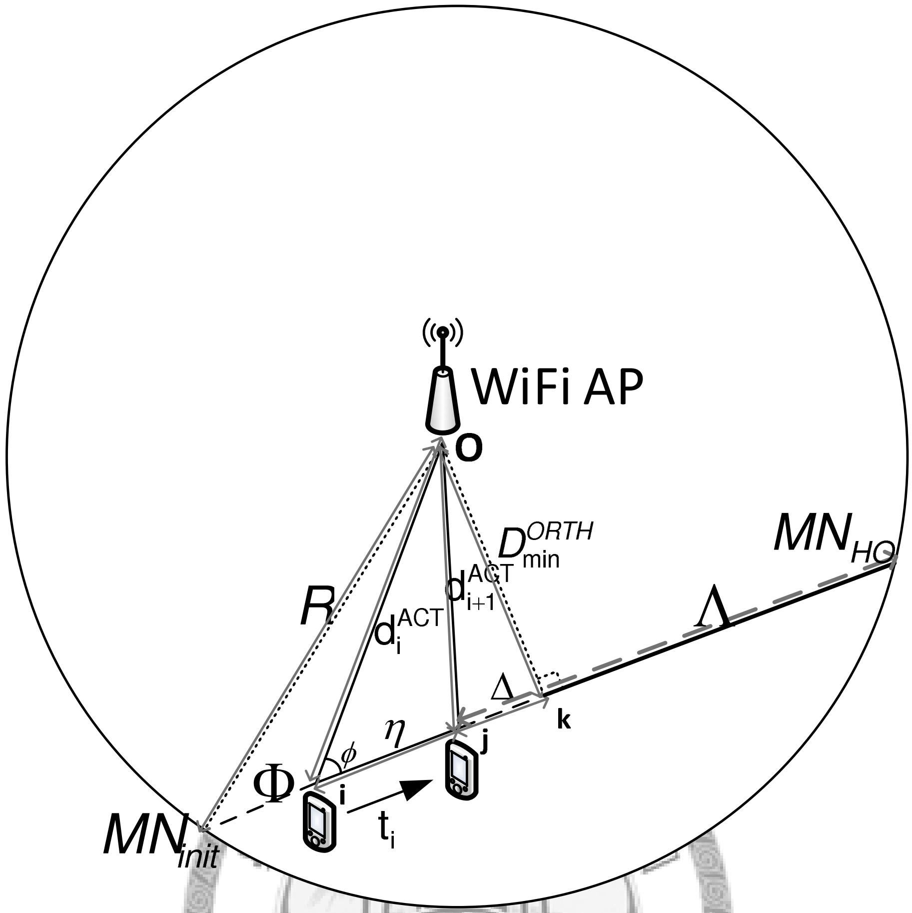


Figure 14: Handover distance time prediction by using geometric approach

that if we have enough samples and we could get the difference between them lower than 5%.

3.1.3 Using Geometry Model to Do the Handover Prediction

After we get β and σ_{dB}, we could start to predict the approximate handover time. From Figure 14, it shows the relation between the sample time and the $d_{\mathtt{i}}^{ACT}$. Because the number of samples requirement, the unit we do the distance calculation is I-frame, that is, when the MN gets an I-frame, it will do the distance calculation one time, so we replace the subscript of `i` to κ in default used by some parameters henceforth because we will do a calculation one time when we received κ^{th} I-frame to have much precise between the actual value.[3]

From Figure 14, the MN gets κ^{th} I-frame and gets the distance between WiFi AP,

[3]In fact, it is elastic to adjust the sample interval. In the simulation environment setup in Section 5.2, we will change another sample interval.

d_κ^{ACT}, and the MN gets $(\kappa+1)^{th}$ I-frame and also gets the distance between WiFi AP, $d_{\kappa+1}^{ACT}$, after t_κ time. So we focus on that we want to get the distance from the place when the MN gets $(\kappa+1)^{th}$ I-frame, to the radius of WiFi AP where causing the MN doing the handover ($\mathbb{D}^{HO}$), which are overlapped with the MN route (the overlapped point named MN_{HO}, and another overlapped point named MN_{init}) in the future, and then using this value divided by the current speed is the remain time to handover.

We define some parameters first: point i means the place when the MN gets the κ^{th} I-frame; point j means the place when the MN gets the $(\kappa+1)^{th}$ I-frame; point k means the event happened when the MN is closest to the WiFi AP; point O means the place of WiFi AP; t_i means the κ^{th} and $(\kappa+1)^{th}$ I-frames received interval; Φ means the distance between MN_{init} and i; η means the distance between i and j; Δ means the distance between j and k; Λ means the distance between j and MN_{HO}; R means the distance between O and MN_{init} or the distance between O and MN_{HO}, equals to $\mathbb{D}^{HO}$.

Now we discuss when the MN is closer to the WiFi AP, means that d_κ^{ACT} is larger than $d_{\kappa+1}^{ACT}$, and from the figure we want to get the distance between point j and point MN_{HO}, that is, Λ. We use geometric method to get the answer.

From trigonometric function, we get relations between the parameters:

$$\cos\phi = \frac{\left(d_\kappa^{ACT}\right)^2 + \eta^2 - \left(d_{\kappa+1}^{ACT}\right)^2}{2 d_\kappa^{ACT} \eta} = \frac{\left(d_\kappa^{ACT}\right)^2 + (\eta+\Lambda)^2 - R^2}{2\left(d_\kappa^{ACT}\right)(\eta+\Lambda)}, \tag{3.14}$$

and we could solve Λ by using quadratic equation calculation after known the value of $\left(d_\kappa^{ACT}\right)$, $\left(d_{\kappa+1}^{ACT}\right)$, η, and R, and finally we could get the remain distance (d_κ^{HO}) and the remain time (t_κ^{HO}) to handover by knowing that:

$$d_\kappa^{HO} = \Lambda,\ t_\kappa^{HO} = \tfrac{\Lambda}{speed_\kappa}. \tag{3.15}$$

In the next section, we will introduce a concept of "zone" to decide the occasion of AMP execution. Actually the MN does not need to monitor and calculate d_κ^{HO} and t_κ^{HO} anytime. By using this concept, we could simplify the job of the MN to reduce the processing time, and we will get the conclusion that the MN could begin to calculate d_κ^{HO} and t_κ^{HO} after the d_κ^{ACT} is larger than a value in the end of the section.

3.1.4 Events Partition Based on Received Signal

We separate the received power signal from 4 parts to describe. The beginning of first part is that it still has some packets dropped due to shadowing effect after

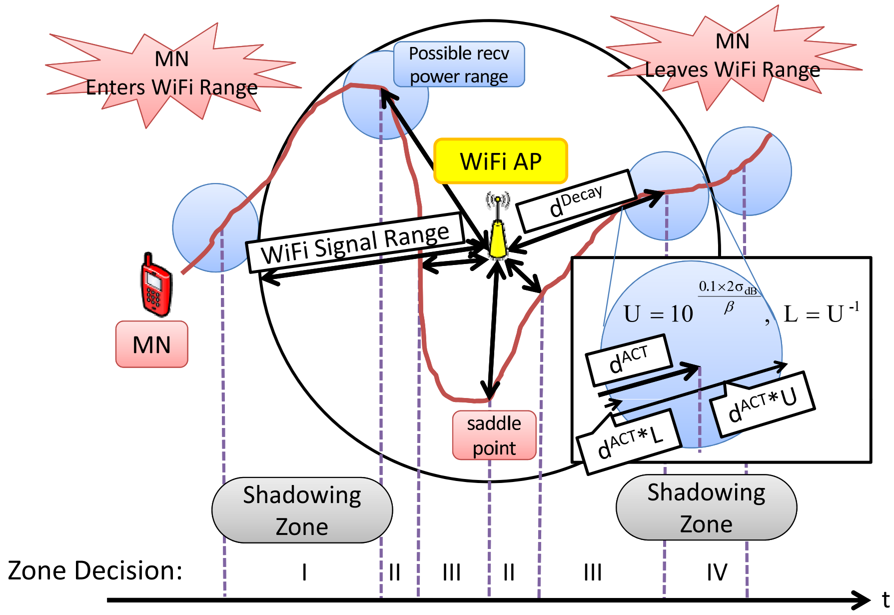


Figure 15: A Sketch to describe the relation between MN route and zone decision

the MN connects with WiFi AP when it enters the WiFi AP signal range, and the extremity of first part is that we have $\left(95.45 + \frac{4.55}{2}\right)\%$ confidence to promise that packets will not be dropped due to shadowing effect, and it is also the beginning of second part. The end of second part is that maximum signal occurred when the MN is closest to the WiFi AP and then it enters the third part where the trend of received signal is downward until the trend of received signal is upward again, and it enters the second part again. The end of third part and the beginning of fourth part is that when we have $\left(95.45 + \frac{4.55}{2}\right)\%$ confidence to promise that packets will not be dropped due to shadowing effect again, that is, the MN is going to enter the zone between d_{κ}^{Decay} and R_t, the end of fourth zone is that when the MN does the handover from WiFi back to WiMAX. Figure 15 is a sketch which describes a MN route and the zone decision, and there is a saddle point produced to represent the MN is back to second zone from third zone. To easy to representation, we plot the received power range of the MN is a 2-dimension and have a symmetric radius.

The fourth zone is a special zone which will quite impact the performance of frame storage and playout because we define it as: the received power at least $\left(\frac{4.55}{2}\right)\%$ percentage will less than RX_Thresh and will not able to be analyzed by the MN. We name the zone to be *Shadowing Zone*. We define the internal diameter of *Shadowing Zone* is d_{κ}^{Decay}, and the external diameter of *Shadowing Zone* is $\left(R_t \times 10^{\frac{0.1\times 2\sigma_{dB}}{\beta}}\right)$, where the definition of R_t is already in Equation 3.5. The definition of d_{κ}^{Decay} is that, if the distance between the WiFi AP and the MN is larger than it, then we claim that the packet loss probability is at least $\left(\frac{4.55}{2}\right)\%$ due to shadowing network.

After we know the definition of d_{κ}^{Decay}, we calculate d_{κ}^{Decay} now to know the distance to do the frame storage scheduling, and we want to send a notice to AMP when it only remain $\left(95.45 + \frac{4.55}{2}\right)\%$ percentage of received signal could be analyzed by the MN. We could refer Equation 3.6 and modify to:

$$
\begin{aligned}
&set: \ d_{\kappa}^{Decay} = R_t \times 10^{\frac{-0.1\times 2\sigma_{dB}}{\beta}} \\
&\rightarrow once\ d \times 10^{\frac{0.1\times 2\sigma_{dB}}{\beta}} \geq R_t, \ where\ d = d_{\kappa}^{STA} \doteqdot d_{\kappa}^{ACT} \\
&\rightarrow The\ MN\ is\ in\ Shadowing\ Zone.
\end{aligned}
\tag{3.16}
$$

So we know that when $d_{\kappa}^{ACT} \geq R_t \times 10^{\frac{-0.1\times 2\sigma_{dB}}{\beta}}$ is true, the packet loss will be occurred due to shadowing network, and the right side of above inequality is named d_{κ}^{Decay}, and it is the distance to separate the third and the fourth zone.

Actually, the MN still has enough distance to trigger the $d_{\kappa}^{HO}$and t_{κ}^{HO} calculation to decide whether beginning to schedule playout frames in fourth zone of this topology even in high speed (15m/s), after knowing β is between 2.7 and 5, σ_{dB} is between 4 and 12 by referring Table 4, Section 3.2, and Chapter 5. By calculation, it could allow the handover duration at least 800 milliseconds and be able to keep playing video streaming. Because the zone has such characteristics, we will decide whether to trigger the mechanism or not prudently after entering this zone.

3.2 Handover Duration Prediction

The function of the handover duration prediction is to precisely estimate the time duration while the MN doing the handover. It will impact the number of frames needed to store before the handover.

We could separate some fundamental elements which will impact the handover duration:

- The condition of WiFi AP beacon loss received by MN, which will impact the MN trigger handover mechanism time.
- The RTT between video streaming server and the MN, which will impact the response time to do the re-routing.
- The handover processing time, which includes some service re-assignment, and the time will be impacted by the channel condition.

About the first point of element estimation, we could create a model that in what condition will cause MN does not receive n sequential beacons if we have already known that the MN triggers handover mechanism after the timer is expired. And about the second element, we could measure RTT when the MN receives the video streaming and get the result. About the last element, we could refer some outcome of some testbed which the scenario is the same as ours.

There are many references mention as the heterogeneous handover in difference scenarios which will impact the handover duration. In [47], a wireless mesh network is considered, including 802.11e and 802.16e service points. It analyzes the routing changing and QoS requirement to decide handover, and using 802.11e module from TKN and 802.16e module from NIST, implementing in NS2. Finally it gets the handover delay is no more than 100 ms, and it claims that with proper buffer size control, the users using VoIP will be unnoticed. [48] uses different transmission priorities, experimenting the handover duration from 802.16e to 802.11e, UMTS to 802.16e, and gets the handover duration is between 7 to 18 msecs. [49] implements its handover decision model to embedded system, testing the handover delay between GPRS and WiFi and get the handover duration is between 5 ms to 60 ms. [50] considers a special scenario which has relay element between WiFi and WiMAX, and the handover duration is between 5 to 13 seconds when the transmission packet size is 1500 bytes. [51] proposes a modified SIP (Session Initiation Protocol) procedure to reduce IMS (IP Multimedia Subsystem) delay under WiFi to UMTS handover scenario, and the traditional outcome is about 610 msec. [52] presents a cooperative agent based approach for the VHO by using MATLAB to do the simulation. [53] sets the total handover time is 10 seconds when doing the simulation to prevent video interruption. [54] constructs a handover topology including 802.11 and 802.16 networks and uses four different decision location and using OPNET simulator in different types of traffic, and the handover duration is from tens of milliseconds to near 1 second. [55] uses NS-2 based on MIH and MIPv6 implementation to simulation the handover scenario from 802.11b to WiMAX network and gets the total need for handover preparation and actual handover is 136ms from its proposed handover procedure. [56] mentions that the standard NEMO (network mobility) handover delay is longer than 1.5 seconds by using mathematical analysis. [57] does the vertical handoff in integrated CDMA and WLAN systems and gets the time delay during the vertical soft handover are 80 msec and 270 msec, respectively.

In summary, we know that the different network scenario, the types of functions in core network, MN, BS already embedded will impact the outcome, and the range of handover duration from above is between several milliseconds to more than 10 seconds. Actually it could not be named 'live streaming' when the end-to-end delay is calculated by several seconds, although our proposed AMP still could deal with such long video interruption in handover duration, and the value of handover duration will impact on AMP execution time it needs at least.

After we decide the handover duration, it will become a constant to be one of the inputs of the third part of AMP mechanism. We set the parameter of handover duration is $t^{Duration}$, which means the video streaming interrupted duration, and in the duration we hope that the media player still has frames to playout to let the event passes unnoticed by users. We assume that in normal condition, the receiver will get $\left\lceil \frac{t^{Duration}}{\mathbb{T}} \right\rceil$ frames most in the duration, so we have to store the same frames before the handover event happens. Although we could set the playout speed in the handover duration as the same as the normal condition, we still let the media player keeps the playout duration in a slowest speed to prevent the handover duration prediction mistake until it re-connects with the server, gets the streaming from it again, and the number of frames in playout buffer is above a threshold.

3.3 The Exchange of Media Server and Playout Decision Modification Information

Although soft handover is already supported in some heterogeneous handover, which could only cause larger frame delay than normal condition which are sent during the handover, we still have to consider when BS does not support soft handover case and no redirecting packet sending functions in MAC Layer. It is the reason that why we have to construct a retransmission mechanism in media server Application Layer. In this part, we propose a cooperation concept about some information exchanges between media server and receiver, and adding some functions on media server to do the video streaming re-synchronize with receiver. We assume that the information they exchange is correct and does not impact and be impacted to the video streaming. Some important information sending from the MN to the media streaming server like: the remaining time to handover and streaming interruption, the handover duration, the handover types do the BSs support, etc. When the server gets the information, it will know the probable time of MN to start and finish handover, and it will pause the frames transmission for a while (i.e., $t^{Duration}$) then transmission to reduce frame loss. After it restarts to transmit the video frames, it will do the resynchronization with the MN to reduce the delay of a frame from transmission to playout. The media server has to consider the transmission rate to prevent traffic congestion after it knows the network type of the MN entering in after successfully handover.

Next chapter, we will propose a novel AMP executed in fourth zone after the MN knows the remaining time (t_{κ}^{HO}), distance (d_{κ}^{HO}) and handover duration ($t^{Duration}$) got from previous sections. We know that it is the third part of whole AMP, and it is also the most important part of three. In this part, AMP will decide the frame playout time during frame scheduling and set the goal which is that AMP has to store exact frames before handover.

Chapter 4

ADAPTIVE MEDIA PLAYOUT CORE ALGORITHM

In Section 2.2.1, we knew the definition of AMP and the boundary we could adjust in playout interval. In Section 3.1, we predicted remaining distance (d_{κ}^{HO}) and time (t_{κ}^{HO}) to did the heterogeneous handover under shadowing propagation by using statistical approach. In Section 3.2, we used theoretical viewpoint to know the heterogeneous handover duration. Now, in this chapter, we prepare to use the information got from the previous sections and do AMP decision. Additionally, there is an extension topic about how to minimize VDoP when executing AMP interesting to us, and we will discuss the optimization framework in Chapter 6.1.

There are some benefits and characteristics when we use the proposed AMP algorithm and summarized below:

- It uses distance-oriented to do AMP. When the distance between the MN and $\mathbb{D}^{HO}$ is closer, the number of frames it stores is closer to the goal.
- It could prevent some events like the MN getting away from WiFi AP suddenly or close again to WiFi AP, speed up or speed down.
- It considers the current speed of MN, d_{κ}^{HO}, and t_{κ}^{HO} during AMP for more precision.

We divide this part into three segments to convenient to discuss. Section 4.1 is the principle of AMP execution and the preparation of AMP execution. Section 4.2 is the events discussion after AMP started, and Section 4.3 is that we briefly introduce local VDoP minimization by using roll-off concept in some special events which are mentioned in second segment, and the detail of roll-off optimization function is in Chapter 6.1.

Table 7 below is the parameters definition which will be used in following sections, except roll-off function:

Notation		Description
$\mathbb{P}$	:	Execution strength, divide to $\mathbb{P}_{incr}$(AMP) and $\mathbb{P}_{decr}$(Inv-AMP) (%)
$\mathbb{P}_{incr}$	:	$\mathbb{P}_{incr} \in [\mathbb{P}_{incr,MIN}, \mathbb{P}_{incr,MAX}]$ (%)
$\mathbb{P}_{decr}$	:	$\mathbb{P}_{decr} \in [\mathbb{P}_{decr,MAX}, \mathbb{P}_{decr,MIN}]$ (%)
t_{min}^{AMP}	:	The minimum time for AMP storing enough frames(sec)
d_{min}^{AMP}	:	The minimum distance for AMP storing enough frames (m)
$t_{\mathtt{j}}^{playout}$	:	The $\mathtt{j}^{th}$ frame playout time (sec)
$t_{\mathtt{j}}^{recv}$	:	The $\mathtt{j}^{th}$ frame completely received time (sec)[1]
$\mathcal{F}$	:	Frame remaining to store, divided to $\mathcal{F}^{\mathbb{N}}$ and $\mathcal{F}^{\mathbb{F}}$
Section	:	An unit for scheduling frames to store
$\mathcal{S}$	:	The number of frames AMP needs to store ($\mathcal{S} \in \mathbb{N}$)
$\mathcal{J}$	:	The *Section* which the MN is/should be in, $\mathcal{J} \in (\mathcal{S}, \mathcal{S}-1, \ldots, 2, 1)$
t_{sec}^{AMP}	:	The time duration in a *Section* (sec)
d_{sec}^{AMP}	:	The distance in a *Section*(m)

Table 7: Parameter settings for AMP core decision

4.1 AMP Execution Principle and Execution Preparation

We here illustrate the principle of our purposed AMP algorithm first before we start to detail its content. About our proposed AMP, it uses a concept of *Section* to separate the schedule of frame storage. That is, the MN has to store indicated number of frames relating with the distance between the MN and the handover distance ($\mathbb{D}^{HO}$), no matter how other factors impacting the movement of MN. AMP will consider some elements to be the references of execution strength: the current speed of the MN, the distance and time remaining to do the handover and the difference between the time requirement of AMP to finish current *Section* and the actual remaining time evaluation. AMP will consider these elements to adjust its execution strength and store frames step by step.

After we know the principle of AMP, we first propose our AMP execution preparation. The range of AMP execution preparation is from collecting information to do some predictions, to the beginning of AMP procedure. The step of AMP execution preparation is below:

1. Calculate d_{κ}^{HO}, t_{κ}^{HO}, and $t^{Duration}$ from Section 3.2 and Section 3.1.3 and start to count the d_{κ}^{HO} and t_{κ}^{HO} backwards.

2. Get the current speed, $speed_{\mathrm{i}\|AP}$ and we assume that the current speed is similar with the nearest speed sampled from κ^{th} I-frame: $speed_{\kappa\|AP}$, and we will replace the symbol of the former to the latter.

3. From the above two steps, AMP could calculate the minimum distance and time the MN needs to store enough frames in maximum capability for it after knowing the limitation from [20].

4. Decide the number of *Sections* to be the steps of AMP execution, and the distance, time interval in every *Section*.

5. Calculate AMP strength (we set $\mathbb{P}_{incr}$ to represent AMP execution strength to slow the playout) in the beginning of execution (namely $\mathbb{P}_{incr,init}$).

6. Use roll-off concept to reduce VDoP when executing AMP at the first time.

We start to explain our propose steps from Step 3 in different sub-sections and ignoring first two steps which have been already mentioned in Section 3.1 and Section 3.2.

4.1.1 The Minimum AMP Execution Distance and Time Calculations

In the third place of AMP execution preparation, after AMP gets $t^{Duration}$ from Section 3.2, it will know the number of frames having to store at least to prevent the video interruption during the handover. The number of frames needed to store is $\mathcal{S} = \left\lceil \frac{t^{Duration}}{\mathbb{T}} \right\rceil$, after AMP knows the default playout frame interval is $\mathbb{T}$ seconds mentioned before. And then AMP could get the minimum execution time requirement, which means the boundary to execute in a highest strength:

$$t_{min}^{AMP} = \mathcal{S} \times \frac{1}{\mathbb{P}_{incr,MAX}} \times \mathbb{T}. \tag{4.1}$$

It means that when $t_{\kappa}^{HO} \leq t_{min}^{AMP}$, AMP should begin to be executed, no matter what $speed_{\kappa\|AP}$ the MN is. Because it knows that it still has enough time to store exact frames preparing for handover. The meaning of middle element of above equation is that AMP may save one more frame after it playout every $\frac{1}{\mathbb{P}_{incr,MAX}}$ frames, and the speed of frame storage is decided by the playout rate, which is defined the percentage of maximum AMP execution strength ($\mathbb{P}_{incr,MAX}$) to slow the playout speed. So when AMP knows that it has to store $\mathcal{F}$ frames, and it knows every $\frac{1}{\mathbb{P}_{incr,MAX}}$ received frames it could store one additional frame, so it has to receive $\mathcal{F} \times \frac{1}{\mathbb{P}_{incr,MAX}}$ frames and get the goal. At last, it knows the received frame interval is $\mathbb{T}$, so it knows the minimum execution time is t_{min}^{AMP} at least. Additionally, we could rewrite the middle element of the equation to $\frac{\mathbb{T}}{\mathbb{T}\times\mathbb{P}_{incr}}$ to fit in general cases:

$$t^{AMP} = \mathcal{F} \times \frac{\mathbb{T}}{\mathbb{T} \times \mathbb{P}_{incr}} \times \mathbb{T}. \tag{4.2}$$

AMP also could get d_{min}^{AMP} after we define t_{min}^{AMP}, and the function of d_{min}^{AMP} is below:

$$d_{min}^{AMP} = t_{min}^{AMP} \times speed_{MAX}. \tag{4.3}$$

To increase readability, we replace $\max\left(speed_{\kappa\|AP}\right)$ to $speed_{MAX}$ henceforth, means the maximum speed of the MN which could achieve 15m/s in the metropolis scenario. Because AMP does not know the $speed_{\kappa\|AP}$ of the MN in the future, it still could promise that no matter what speed of the MN, including $speed_{MAX}$, it always could store enough frames beginning from $d_{\kappa}^{HO} = d_{min}^{AMP}$ to the handover execution.

4.1.2 *Section* Environment Decision: $\mathcal{S}$, d_{sec}^{AMP}, and t_{sec}^{AMP}

In the fourth place of AMP execution preparation, AMP decides the number of *Sections* to be the steps of execution. We set *Sections* based on the number of MN storing frames it needs, one stored frame one *Section*. We could treat *Section* as a schedule and there are some checkpoints will be placed on it to ask whether AMP achieving the current job until now or not. So there are $\mathcal{S}$ *Sections*, and the distance interval in every *Section* $\mathcal{J}$ is

$$d_{sec}^{AMP} = \frac{d_{min}^{AMP}}{\mathcal{S}}. \tag{4.4}$$

We set $\mathcal{J} \in (\mathcal{S},\ \mathcal{S}-1, \ldots,\ 2,\ 1)$, and it means that the MN still needs $\mathcal{J}-1$ complete frames to store and now AMP is working in $\mathcal{J}^{th}$ *Section*. After AMP knows the distance interval in every *Section* $\mathcal{J}$, it also could know the time interval. To simplify the equation of time interval by assuming the changes in speed between each *Section* is small, we could define t_{sec}^{AMP} as:

$$t_{sec}^{AMP} = \frac{d_{sec}^{AMP}}{speed_{\mathcal{J}\|AP}}. \tag{4.5}$$

Because t_{sec}^{AMP} must not the same between every *Section*, and thus we change the representation of AMP execution duration from t_{sec}^{AMP} to $t_{\mathcal{J}}^{AMP}$.

4.1.3 $\mathbb{P}_{incr,init}$ Decision at the First Time of AMP Execution

Before we start to introduce the fifth place of AMP execution preparation, AMP has to know the time requirement to achieve the goal and the remaining time in every *Section* first. Equation 4.2 represents the former and Equation 4.5 represents the latter statement. AMP has to keep the balance by adjusting $\mathbb{P}_{incr}$when speed changing. When the MN enters the first *Section*, it has to set the initial $\mathbb{P}_{incr}$. Because here AMP just considers only one *Section* and storing one frame, so it could get: $t^{AMP} = t_{\mathcal{J}}^{AMP}$, $d_{min}^{AMP} = d_{sec}^{AMP}$, $\mathcal{S} = 1$, $\mathcal{J} = 1$. So it could get the relation between $\mathbb{P}_{incr,init}$ and the MN current speed $speed_{\mathcal{J}\|AP}$.

$$\begin{aligned}
&\because t^{AMP} = \tfrac{1}{\mathbb{P}_{incr,init}}\mathcal{F}\mathbb{T};\\
&\quad t_{\mathcal{J}}^{AMP} = \tfrac{d_{sec}^{AMP}}{speed_{\mathcal{J}\|AP}} = \tfrac{d_{min}^{AMP}}{speed_{\mathcal{J}\|AP}} = \tfrac{t_{min}^{AMP}\times speed_{MAX}}{speed_{\mathcal{J}\|AP}};\\
&\quad and \quad t^{AMP} = t_{\mathcal{J}}^{AMP}\\
&\therefore \mathbb{P}_{incr,init} = \tfrac{speed_{\mathcal{J}\|AP}}{t_{min}^{AMP}\times speed_{MAX}}.
\end{aligned} \tag{4.6}$$

4.1.4 The Use of Roll-off Concept to Prevent High Local VDoP

After AMP gets the value of $\mathbb{P}_{incr,init}$, it could continue its work to the last place of execution preparation. To achieve VDoP minimization, AMP will use roll-off function, not only in AMP execution preparation event. Because it is an additional job to achieve, we will describe our optimization framework in Section 4.3 and detail it in Chapter 6.1.

We will discuss some events happens during AMP execution first in Section 4.2, and then we analyze the events to decide whether they have to consider roll-off function or not, when we consider VDoP minimization.

4.2 AMP Adaptation Owing to Some Events

After AMP begins to work, it enters the $\mathcal{J}^{th}$ *Section* and store frames, and there are some missions which AMP has to deal with to get the goal. When AMP finds out that the checkpoint is updated, it will check it whether achieving the goal or not and then doing some suitable responses, such as changing $\mathbb{P}$ to catch new schedule requirement.

The routine tasks during AMP execution which are listed below:

1. Calculate and update d_{κ}^{HO}, t_{κ}^{HO}, and $speed_{\kappa \| AP}$ to know the current *Section* where the MN should be in, and the number of storing frames it should have.

2. Do some appropriate responses to catch the updated schedule after d_{κ}^{HO} renewed; use roll-off concept if it is essential.

3. Send the feedback information to the media server to know the remaining time to delay frames transmission.

And we will discuss them in separate sub-sections.

4.2.1 d_{κ}^{HO}, t_{κ}^{HO}, and $speed_{\kappa \| AP}$ Updates

Here we define remaining frame storage ($\mathcal{F}_{\mathtt{j}}$, measured when $\mathtt{j}^{th}$ frame playout) which is impact by $\mathbb{P}_{incr}$ and $\mathbb{P}_{decr}$. So the relation between them is: $\mathcal{F}_{\mathtt{j}+1} = \mathcal{F}_{\mathtt{j}} - \mathbb{P}_{incr}$ or $\mathcal{F}_{\mathtt{j}+1} = \mathcal{F}_{\mathtt{j}} - \mathbb{P}_{decr}$. $\mathcal{F}_{\mathtt{j}}$ is composed of integer part and fraction part. The integer part of $\mathcal{F}_{\mathtt{j}}$ is $\mathcal{J} - 1$ (represents $\mathcal{F}_{\mathtt{j}}^{\mathbb{N}}$, and it means that $\mathcal{F}_{\mathtt{j}}^{\mathbb{N}} \in \mathbb{N}$), and the fraction part means the unfinished part of $\mathcal{J}^{th}$ *Section* (represents $\mathcal{F}_{\mathtt{j}}^{\mathbb{F}}$). For example, in a AMP progress, $\mathcal{F}_{\mathtt{j}}$ is 3.5, so we know that $\mathcal{F}_{\mathtt{j}}^{\mathbb{N}}$ is 3 and $\mathcal{F}_{\mathtt{j}}^{\mathbb{F}}$ is 0.5, and it is in 4^{th} *Section*. Because it has a relation between $\mathcal{J}$ and $\mathcal{F}_{\mathtt{j}}^{\mathbb{N}}$ in given $\mathtt{j}$, so we will use one of them alternately afterward. We could write the finished part of $\mathcal{J}^{th}$ *Section* as $\frac{\left[\left(t_{\mathtt{j}}^{playout} - t_{\mathtt{j}}^{recv}\right) - (\mathcal{S} - \mathcal{J}) \times \mathbb{T}\right]}{\mathbb{T}}$. We could subtract the finished part from 1 to get the answer. AMP could use Equation 4.7 to know where *Section* does the MN place and what schedule the MN should catch. And then AMP could decide the execution strength when it finds out that the MN is updated to a new *Section*. AMP could add the concept of roll-off if it needs.

$$
\begin{aligned}
&For\ current\ Section:\\
&\mathcal{F}_{\mathtt{j}}^{\mathbb{N}} = \mathcal{S} - \left(\left\lfloor \frac{\left(t_{\mathtt{j}}^{playout} - t_{\mathtt{j}}^{recv}\right)}{\mathbb{T}} \right\rfloor + 1 \right), \quad \mathcal{J} = \mathcal{F}_{\mathtt{j}}^{\mathbb{N}} + 1,\\
&\mathcal{F}_{\mathtt{j}}^{\mathbb{F}} = \frac{\left[\left(t_{\mathtt{j}}^{playout} - t_{\mathtt{j}}^{recv}\right) - (\mathcal{S} - \mathcal{J}) \times \mathbb{T}\right]}{\mathbb{T}};\\
&For\ updated\ Section:\\
&\left(d_{sec}^{AMP} \times \mathcal{J}\right) \geq d_{\kappa}^{HO} \geq \left(d_{sec}^{AMP} \times (\mathcal{J} - 1)\right), \quad i.e.,\ d_{\kappa}^{HO} = \left(\mathcal{F}'^{\mathbb{N}}_{\mathtt{j}} + \mathcal{F}'^{\mathbb{F}}_{\mathtt{j}}\right) \times d_{sec}^{AMP}.
\end{aligned}
\tag{4.7}
$$

We could represent *Section Judgment(SJ)* to Algorithm 4.1.

Algorithm 4.1 *Section Judgment (SJ)*

start Section Judgment:
while (get $(\kappa+1)^{th}$ complete I-Frame) **do**:
  %Check whether got a new I-frame or not
update t_{κ}^{HO}, $\kappa \leftarrow (\kappa+1)$
update and decide $\mathcal{F}_{\mathtt{j}}$, $\mathcal{F}'_{\mathtt{j}}$ from Equation 4.7
  if $(\left|\mathcal{F}'^{\mathbb{N}}_{\mathtt{j}} - \mathcal{F}^{\mathbb{N}}_{\mathtt{j}}\right| \leq 1)$ **then**: Case 1 *%The MN still keeps in the same Section*
  else if $(\left(\mathcal{F}'^{\mathbb{N}}_{\mathtt{j}} - 1\right) > \mathcal{F}^{\mathbb{N}}_{\mathtt{j}})$ **then**: Case 2 *%Schedule falls forward*
  else if $(\left(\mathcal{F}^{\mathbb{N}}_{\mathtt{j}} - 1\right) > \mathcal{F}'^{\mathbb{N}}_{\mathtt{j}})$ **then**: Case 3 *%Schedule falls behind*
  else
  end if
end while

4.2.2 Trying to Catch the Updated Schedule, Considering to Use Roll-off Function

We separate five cases after updating the MN place and the responses to every case, and each case have their own response steps. The first case triggered when the MN still keeps in the same *Section* (called *SS*, the current and updated schedule are close), the second case triggered when the MN's current schedule is falling forward (called *FF*, AMP stores too many redundant frames in updated schedule's viewpoint), the third case triggered when the MN's current schedule is falling behind (called *FB*, AMP stores too little frames in updated schedule's viewpoint), the fourth case triggered when the MN in handover duration, and the fifth case triggered when the MN completely finishes doing the handover.

Furthermore, AMP will update *Section Judgment* to get the updated schedule which AMP should catch, once a new I-frame is received.

Case 1: The MN still keeps in the same *Section* The value of $\mathcal{F}_{\mathtt{j}}^{\mathbb{F}}$ becomes more important than other cases, relating with the actual progress of frame storage in every *Section*. Then we put this answer into Equation 4.2 and rewrite the equation:

$$t_{REQ,\mathtt{j}}^{AMP} = \mathcal{F}_{\mathtt{j}}^{\mathbb{F}} \times \frac{\mathbb{T}}{\mathbb{P}_{incr}}, \tag{4.8}$$

where $t_{REQ,\mathtt{j}}^{AMP}$ means time requirement to finish storing a complete frame when AMP execution strength is $\mathbb{P}_{incr}$. Contrarily, the actual remaining time to the end of this *Section* is:

$$t_{ACT,\mathtt{j}}^{AMP} = t_{\kappa}^{HO} - \frac{d_{sec}^{AMP} \times \mathcal{F}_{\mathtt{j}}^{\mathbb{N}}}{speed_{\kappa\|AP}}, \tag{4.9}$$

and the inequality $t_{REQ,\mathtt{j}}^{AMP} \leq t_{ACT,\mathtt{j}}^{AMP}$ should always be true anytime in this case.

AMP will check whether $t_{ACT,\mathtt{j}}^{AMP}$ is larger than $t_{REQ,\mathtt{j}}^{AMP}$ or not in current *Section*. In fact, t_{κ}^{HO} is a main element to have a chance to let $t_{ACT,\mathtt{j}}^{AMP}$ is larger than 1 or smaller than 0, rather than $speed_{\kappa\|AP}$. To keep $t_{REQ,\mathtt{j}}^{AMP} \leq t_{ACT,\mathtt{j}}^{AMP}$ being true but not too much, the mechanism will increase $\mathbb{P}_{incr}$ when the statement is false, or decrease $\mathbb{P}_{incr}$ when the gap between them is too large.

We could represent this case by using Algorithm 4.2.

Algorithm 4.2 Case 1: MN keeps in the same *Section* (*SS*)

δ: a small value to let $t_{ACT,\mathtt{j}}^{AMP}$ not far away from $t_{REQ,\mathtt{j}}^{AMP}$. ex: $\frac{1}{2}t_{\mathcal{J}}^{AMP}$
start Case 1:
while $((1+\mathbb{P}_{incr})\,\mathbb{T} == 0)$ **do**: *%Start to playout a new frame*
update $\mathtt{j} \leftarrow (\mathtt{j}+1)$
update $speed_{\mathtt{i}\|AP}$, $\mathcal{F}_{\mathtt{j}}$, t_{κ}^{HO}
calculate $\left(t_{REQ,\mathtt{j}}^{AMP} - t_{ACT,\mathtt{j}}^{AMP}\right)$ from Equation 4.8 and 4.9.
  if $(t_{ACT,\mathtt{j}}^{AMP} - t_{REQ,\mathtt{j}}^{AMP} \geq \delta)$ **then**: renew $\mathbb{P}_{incr}$ to fit $t_{ACT,\mathtt{j}}^{AMP} - t_{REQ,\mathtt{j}}^{AMP} \leq \delta$
  else if $(t_{ACT,\mathtt{j}}^{AMP} - t_{REQ,\mathtt{j}}^{AMP} \leq 0)$ **then**: renew $\mathbb{P}_{incr}$ to fit $t_{ACT,\mathtt{j}}^{AMP} - t_{REQ,\mathtt{j}}^{AMP} \geq 0$
  else then: keep $\mathbb{P}_{incr}$
  end if
start backward counter: $(1+\mathbb{P}_{incr})\,\mathbb{T}$
else
end while

Case 2: The MN's current schedule is falling forward (*FF*) There are some reasons to let the schedule fall forward, and this phenomenon would cause unmeaning frame storage and the time delay from frame transmitted to playout. AMP should catch up the new schedule but not too <u>active</u>. From this thinking, AMP sets $P'^{AMP}_{incr} = 0$, where $\mathcal{J}'$ is the *Section* after evaluated. To prevent high VDoP, AMP again uses the concept of roll-of function before updating AMP execution strength. AMP executes roll-off function until in the end of *Section* $(\mathcal{J} - 1)$ but spending more time to hope the update schedule could catch up the previous schedule. AMP uses the updated d^{HO}_{κ} in the roll-off duration to check whether the new schedule catches up again or not. So the previous schedule reduces the storing efficiency to $\left(\frac{1+F^{\mathbb{F}}_{\mathrm{j}}}{F'^{\mathbb{N}}_{\mathrm{j}}-\left(F^{\mathbb{N}}_{\mathrm{j}}-1\right)+F'^{\mathbb{F}}_{\mathrm{j}}}\right)$ times of previous. In other words, it extends time execution in the remain and next *Sections* to $\left(F'^{\mathbb{N}}_{\mathrm{j}} - \left(F^{\mathbb{N}}_{\mathrm{j}} - 1\right) + F'^{\mathbb{F}}_{\mathrm{j}}\right) \times t^{AMP}_{sec}$ and get two additional storage frames, and it will decrease frame storage efficiency, and then increase frame storage efficiency to get the goal. When AMP stores two additional frames before the new schedule catches up, the MN cancels AMP execution and back to normal mode and continuing monitoring the new schedule progress by updating t^{HO}_{κ} information. When it finds out the deviation of progress between the schedules larger than half of $\mathcal{S}$ or $d^{HO}_{\kappa} > d^{AMP}_{min}$, the MN starts to do Inv-AMP, which goal is to release stored frames. We set $\mathbb{P}_{decr}$ to be the symbol of Inv-AMP execution strength, contrasting with $\mathbb{P}_{incr}$ to be the symbol of AMP execution strength. The MN uses almost the same way to do Inv-AMP like the initial state of AMP mentioned in Section 4.1 and doing some modifications on Step 5. It begins to do Inv-AMP until it catches the new schedule. Once d^{HO}_{κ} is larger than d^{AMP}_{min} which is the $\mathbb{R}^{th}$ time event happened sequentially, Inv-AMP will release all the number of storage frames then using roll-off function to shutdown all AMP algorithm. $\mathbb{R}$ could be set by users and it is 5 in default.

We could represent this case by using Algorithm 4.3, and some procedures are in the behind of it.

Case 3: The MN's current schedule is falling behind (*FB*) It is quite serious when AMP finds out the schedule is falling behind, and it means that $t^{AMP}_{ACT,j}$ is not a logical value (i.e., smaller than 0). Contrary to Case 2 mentioned before, AMP should catch up the new schedule <u>actively</u> here, because it may cause that the MN does not store enough frames to face the handover. Thanks to AMP having elastic execution strength, AMP empowers itself to be able to increase the playout

Algorithm 4.3 Case 2: Schedule Falling Forward: Main Function

Δ: roll-off execution duration
update t_{κ}^{HO} when $\kappa \leftarrow \kappa + 1$
update $F_{\mathtt{j}}$, $F'_{\mathtt{j}}$, when $\mathtt{j} \leftarrow \mathtt{j} + 1$ anytime, after playout and calculating from Equation 4.7
$\delta \leftarrow \left|F'^{\mathbb{N}}_{\mathtt{j}} - F^{\mathbb{N}}_{\mathtt{j}}\right|$ *%The deviation between now and update schedule*
%the place where $\mathtt{j}^{th}$ *playout frame's contribution to playout delay*
start Case 2:
invoke **Schedule(5)** *%Slow the storing speed*
invoke **Checkpoint(1)** *%Check whether catching new scheduler*
invoke **Schedule(3)** *%If not pass, execute Inv-AMP roll-off*
invoke **Checkpoint(2)** *%Recheck whether catching new scheduler*
while ($\Delta == 0$) **do:**
execute Inv-AMP
invoke **Checkpoint(2)** *%Recheck while executing Inv-AMP*
$\Delta \leftarrow (1 + \mathbb{P}_{decr})\,\mathbb{T}$
start backward counter: Δ
end while *%Repeat Inv-AMP until pass Checkpoint(2)*
quit

interval up to $(1 + \mathbb{P}_{incr,MAX})\,\mathbb{T}$ to prevent the streaming interruption as well as it could. After AMP knows d_{κ}^{HO} is in $\mathcal{J}'^{th}$ *Section*, it wants to catch the new schedule when it is in the end of $(\mathcal{J}-1)'^{th}$ *Section*. In other words, it wants to produce $\left(F^{\mathbb{N}} - \left(F'^{\mathbb{N}} - 1\right) + F^{\mathbb{F}}\right) \times \mathbb{T}$ additional delay by spending $\left(1 + F'^{\mathbb{F}}\right) \times t_{sec}^{AMP}$ time. So there are $\left\lfloor \frac{\left(1+F'^{\mathbb{F}}\right)\times t_{sec}^{AMP}}{\mathbb{T}} \right\rfloor$ frames received in this duration, and the total gain of additional delay in this duration is $(\mathbb{P}_{incr} \times \mathbb{T}) \times \left\lfloor \frac{\left(1+F'^{\mathbb{F}}\right)\times t_{sec}^{AMP}}{\mathbb{T}} \right\rfloor$. AMP could compare with this outcome to the requirement of additional delay and calculate whether the inequality is true or not:

$$\left(F^{\mathbb{N}} - \left(F'^{\mathbb{N}} - 1\right) + F^{\mathbb{F}}\right) \times \mathbb{T} \leqq (\mathbb{P}_{incr} \times \mathbb{T}) \times \left\lfloor \frac{\left(1 + F'^{\mathbb{F}}\right) \times t_{sec}^{AMP}}{\mathbb{T}} \right\rfloor. \tag{4.10}$$

Because there are limit frames received in the duration, AMP will directly choose $\mathbb{P}_{incr,MAX}$ to be P'^{AMP}_{incr} without executing roll-off. AMP will use P'^{AMP}_{incr} until the schedule catches the new *Section* and then it uses roll-off to reduce AMP strength.

We could represent this case by using Algorithm 4.8.

After discussing the three cases above, we also have to care about the event when changing a case to another judging by *SJ* when t_{κ}^{HO} and d_{κ}^{HO} is updated due to high variable of network. In fact, we only have to discuss some special cases and know

Algorithm 4.4 Procedure: Scheduling

Procedure: Schedule(i)
update $\mathcal{J}$ from using Equation 4.7
update $\mathbb{P}$, $speed_{\mathtt{i}\|AP}$ based on Equation 4.6
switch case (i==1)
execute roll-off [Case I],
$\mathbb{P}_{incr,past} \leftarrow 0$, $\mathbb{P}_{incr,goal} \leftarrow \mathbb{P}_{incr}$, $\Delta \leftarrow 2t_{\mathcal{J}}^{AMP}$
switch case (i==2)
execute roll-off [Case II],
$\mathbb{P}_{incr,past} \leftarrow \mathbb{P}_{incr}$, $\mathbb{P}_{incr,goal} \leftarrow 0$, $\Delta \leftarrow 2t_{\mathcal{J}}^{AMP}$
switch case (i==3)
execute roll-off [Case III],
$\mathbb{P}_{decr,goal} \leftarrow \mathbb{P}_{decr,init}$, $\mathbb{P}_{decr,past} \leftarrow 0$, $\Delta \leftarrow 2t_{\mathcal{J}}^{AMP}$
switch case (i==4)
execute roll-off [Case IV],
$\mathbb{P}_{decr,goal} \leftarrow 0$, $\mathbb{P}_{decr,past} \leftarrow \mathbb{P}_{decr}$, $\Delta \leftarrow 2t_{\mathcal{J}}^{AMP}$
switch case (i==5)
execute roll-off [Case V],
$\Delta \leftarrow \left(F'^{\mathbb{N}}_{\mathtt{j}} - \left(F^{\mathbb{N}}_{\mathtt{j}} - 1\right) + F'^{\mathbb{F}}_{\mathtt{j}}\right) \times t_{sec}^{AMP}$
$\mathbb{P}_{incr,past}, \mathbb{P}_{incr,goal} \leftarrow \mathbb{P}_{incr}$, $TD \leftarrow \Delta$, $RAD \leftarrow 2$
% First decrease then increase AMP storage efficiency
%Change the frame storage efficiency and storing two additional frames
end switch
start backward counter: Δ
invoke **TimeUp** *%If time is up, update some parameters*

Algorithm 4.5 Procedure: Checkpoint(i)

Procedure: Checkpoint(i)
update $\mathcal{F}^{\mathbb{N}}_{\mathtt{j}}$, $F'^{\mathbb{N}}_{\mathtt{j}}$, t^{HO}_{κ}
if ($d^{HO}_{\kappa} \le d^{AMP}_{min}$) **then:** *%Schedule still in AMP execution range*
  if ($\delta \le 1$) **then:**
    %Already caught the new schedule and within AMP range
    if (**i**==1) **then: ;** *%Check after [Case V]*
    else then:
    invoke **Schedule(4)** and then **Schedule(1)**
    end if
  update $\mathbb{P}_{incr}$,$speed_{\mathtt{i}\|AP}$ and use new $\mathbb{P}_{incr}$
  quit
  else then: *%Still not catches the new schedule*
    if (**i**==1) **then:**
      if ($\mathbb{P}_{incr} \neq 0$) **then:**
      invoke **Schedule(2)** *and* then **Checkpoint(1)**
        %From AMP to 0 when timeout, and recheck again
      else if ($\delta > \frac{1}{2}\mathcal{F}$) **then:**
      continue *%to prepare for doing Inv-AMP*
      else then: *%Check after [Case III] and during Inv-AMP*
      Execute normal mode
      end if
    else then:
    continue *%Continue doing Inv-AMP*
else then:
%not in execution range, prepare to release whole stored frames by AMP
  if ($\mathcal{F}^{\mathbb{N}}_{\mathtt{j}} \ge \mathcal{F}$) **then:** *%No any stored frames by AMP*
    if ($\mathbb{P} \neq 0$ && **i**==2) **then:** invoke **Schedule(4)**
    else if ($\mathbb{P} \neq 0$ && **i**==1) **then:** invoke **Schedule(2)**
    end if
  Execute normal mode *%Cancel AMP, $\mathbb{P} \leftarrow 0$*
  quit
  else then: %still some frames stored by AMP
  invoke **OutOfRange** *%record the alert*
  end if
end if

Algorithm 4.6 Procedure: OutOfRange

ϑ: the number of sequential out of range alert times, based on user requirement
Procedure: OutOfRange
$IAMP[\kappa] \leftarrow 1$
if $((IAMP[\kappa] \bigcap \ldots \bigcap IAMP[\kappa - \vartheta]) == 1)$ **then:**
  if $(\mathbb{P}_{incr} \neq 0 \;\; \&\& \;\; \mathbb{P}_{decr} = 0)$ **then:**
  invoke **Schedule(2)**
  end if
  if $(\mathbb{P}_{incr} = 0 \;\; \&\& \;\; \mathbb{P}_{decr} = 0)$ **then:**
  invoke **Schedule(3)**
  end if
  while $(\Delta == 0 \;\; \&\& \;\; \mathcal{F}_{\mathtt{j}}^{\mathbb{N}} < \mathcal{F})$ **do:**
  %Still has some AMP frames to release
  %Not execute Checkpoint(2) anymore, only doing frame releasing
  Execute Inv-AMP
  $\Delta \leftarrow (1 + \mathbb{P}_{decr})\, \mathbb{T}$
  start backward counter: Δ
  end while
  invoke **Schedule(4)**
  Execute normal mode *%Doesn't have any AMP frames to release*
  quit
else
wait for d_{κ}^{HO} update
Checkpoint(2)
end if

Algorithm 4.7 Procedure: TimeUp

Procedure: TimeUp
while $(\Delta == 0)$ **do:**
update $\mathtt{j}$, $\mathcal{J}$, $\mathcal{F}_{\mathtt{j}}$, t_{κ}^{HO}, $speed_{\mathtt{i}\|AP}$
end while

Algorithm 4.8 Case 3: Schedule Falling Behind

Δ: roll-off execution duration
update t_{κ}^{HO} when $\kappa \leftarrow \kappa + 1$ anytime
update $F_{\mathtt{j}}$, $F'_{\mathtt{j}}$ when $\mathtt{j} \leftarrow \mathtt{j} + 1$ anytime, after playout and calculating from Equation 4.7
start Case 3:
if $(F^{\mathbb{N}} \leq F'^{\mathbb{N}})$ **then** *%Catch the new schedule*
update $speed_{\mathtt{i}\|AP}$ based on Equation 4.6
execute roll-off [Case II],
$\mathbb{P}_{incr,past} \leftarrow \mathbb{P}_{incr,M}$, $\mathbb{P}_{incr,goal} \leftarrow \mathbb{P}_{incr}$, $\Delta \leftarrow 2t_{\mathcal{J}}^{AMP}$
start backward counter: Δ
invoke **TimeUp** *%If time is up, update some parameters*
use new $\mathbb{P}_{incr}$ *%Restart new AMP*
else
$\mathbb{P}_{incr} \leftarrow \mathbb{P}_{incr,MAX}$
end if

how to transform from one case to another because it is not very hard to image that how to transform from one case to another from above introduction.

- *FF* in releasing frame step to *FB*: Although there is a large gap between them, it still has higher priority to catch the new scheduler. So AMP will spend $2t_{min}^{AMP}$ to transform from $\mathbb{P}_{decr}$ to $\mathbb{P}_{incr,MAX}$.

Case 4: The MN is in handover duration As mentioned before, to prevent the playout buffer underflow event happened due to handover duration prediction part mistake, AMP will set $\mathbb{P} \leftarrow \mathbb{P}_{incr,MAX}$ during the handover until the **Case 5** is triggered.

Case 5: The MN completely finishes doing the handover After the MN reconnects to the WiMAX BS and continuing receiving video streaming, it will receive many video frames retransmitted by the media server owing to cooperating with each other in hard handover case, or receive video packets retransmitted by the BS in soft handover case. About the mechanism, it will be mentioned in Section 3.3. When there are many frames in the MN's buffer, the MN will use Inv-AMP to release frames to reduce the duration of frame from transmitted to playout. Unless we consider the buffer constraint or the user requirement, Inv-AMP will be free to control the speed of frame release until the number of remain frames in buffer is almost the same as the normal case in a meaningful range.

4.2.3 The Feedback Information Transmission to the Media Server

In default, the media server Application Layer does not know the receiver status, including preparing to do the handover. To reduce frames loss in hard handover case, the media server could delay the frames transmission during the receiver doing the handover. It also has to tell to the server about the BSs handover supported types to let the server to decide whether delaying frame transmission in the handover duration or not. We have already mentioned the functions of the media server in Section 3.3.

4.3 The Concept of Proposed Roll-off Function Design to VDoP Optimization

When AMP/Inv-AMP adjusts the playout interval between frames, $|t_j - \mathbb{T}|$ will not be 0 anymore, which is a score to justify our proposed AMP is good or not. It is a trade-off between postponing frame receiving deadline and ignoring this truth, and both decisions will be given different DoP values. The proposed roll-off function could minimize VDoP after AMP inputs current network conditions and limitations, like roll-off execution duration, additional delay between frame received and playout requirement, current $\mathbb{P}$ and requirement $\mathbb{P}$ after finished roll-off, etc. The detail of proposed function will be discussed in Chapter 6.1, and here we just list and introduce the classification of different purposes of roll-off functions:

- [Case I]: Executed when normal mode transforms to AMP mode (i.e., $\mathbb{P} = 0\% \rightarrow \mathbb{P}_{incr}$)
- [Case II]: Executed when AMP mode transforms to normal mode (i.e., $\mathbb{P}_{incr} \rightarrow \mathbb{P} = 0\%$)
- [Case III]: Executed when normal mode transforms to Inv-AMP mode (i.e., $\mathbb{P} = 0\% \rightarrow \mathbb{P}_{decr}$)
- [Case IV]: Executed when Inv-AMP mode transforms to normal mode (i.e., $\mathbb{P}_{decr} \rightarrow \mathbb{P} = 0\%$)
- [Case V]: Executed in AMP mode, longer execution duration but limited frame storing efficiency (i.e., $\mathbb{P}_{incr} \rightarrow \mathbb{P}_{incr}$)

Chapter 5

SIMULATION RESULT AND ANALYSIS

In Chapter 3 and Chapter 4, we extended the concept of AMP to protect the video streaming from interrupted by heterogeneous handover. We created three main parts to be our proposed AMP mechanism: handover time prediction, handover duration prediction, and AMP core decision. In this chapter, we will employ simulations to demonstrate the superiority of the proposed AMP mechanism. We adopt NS2 (Network Simulator 2) to generate the topology and do our main simulation. In Section 5.1, we will introduce our modified video module which has much higher integrality and logic. In Section 5.2, we will define our scenario parameters and introduce some AMPs which will be compared with our proposed AMP. We will do a series of simulations and analyze the outcome in Section 5.3 and 5.4.

5.1 NS-2 Simulation Platform Modification

In order to build a nearly like real network on a personal computer, we use Network Simulator (NS-2) [58] version 2.33 to be our simulation platform. This version has not embedded in video module, different types of popular network modules except WiFi, and heterogeneous handover module yet, so the first job is to add the video module into NS-2. Some additional types of network modules like WiMAX module are already published and listed in [59]. About mobile network module, it is also already created by some organizations, so we could directly download the modules to use. However, it also easy to find some video modules but no one fit our requirement, so we will create our own video module by getting a concept from Smallko's video module [60][1].

The module reads a record file created by a *xvid_ encraw* converter which also exports video information while converting, like playout time, every frame size, number of frame, etc. All of the information within a frame is written into a row. When NS-2 sender Application Layer reads this file (i.e., trace file) which includes each frame information of video, it begins to package the specific information in order. The unit of Application Layer packages is frame, and in a preset time interval the frame will go

[1] Smallko combines *Evalvid* [61] and NS-2 to be *myEvalvid*.

through Transport Layer, which will divide frames into some small size packets to fit the preset UDP maximum fragment size requirement. After Transport Layer marks sequence number and size on packets, it will continue sending them to Network Layer. Transport Layer also records packets information into another output file. Then they will continue be sent down to lower layers and be got by receiver if they are not dropped on the transmission way. The goal of receiver side is trying to combine the separated packets back to complete frames when it starts to playout and receive video packets. It also records receiving packets situation and playout situation to respective files. There is a program to compare the difference between the sender and receiver files which record packets situation and then it produces a distortion video file which represents that an original video goes through the simulation network and received by the receiver.

In spite of already having some basic functions in this video module, we have to add some additional functions to let it be more systematic. Figure 16 is a sketch which simply shows the modified agents and the jobs they do. The figure shows the video streaming built from provider's Application Layer and segregated from frames to packets in Transport Layer and continuing sent to the lower layers. When Transport Layer of video streaming receiver gets the video packets, it forwards the packets to Application Layer. Application Layer has two main jobs when it gets video packets: one is to put packets into the buffer and another one is to acquire some essential information from packets and analyze. The detail of functions are show in Figure 17. The solid line in the figure means the execution direction and the solid-dot line between *Playout_req* and *Packet_buffer* means that *Playout_req* copies packets information in the buffer once they accord with some requirements from *Playout_req*. The solid-dot line between *Packet_buffer* and *Packet_buffer_update* means the latter will impact states of packets in playout buffer.

When *VideoAMPsink* receives video packets forwarded by *udpAMPsink*, it will record packets information into a file (*.log file in *Packet_rec* function), and they prepare to store to the buffer and waiting for playout. The buffer will decide whether it will store the packets or not based on remaining buffer size in *Packet_buffer* function and the playout time. There is another file to record packets dropped or stored status and reason (*.rec file in *Packet_rec* function). We separate dropped packets in Application Layer from two reasons: buffer overflow ($P_{i,j} + B_{now} > B_{max}$) or timeout ($Pts_{i,j} < Pts_{now}$), easily to distinguish the serious level. Here Pts means the playout time stamp, P means a certain packet size, and B means the playout buffer. *Packet_buffer* will trigger *Playout_bar* to playout once the buffer used is

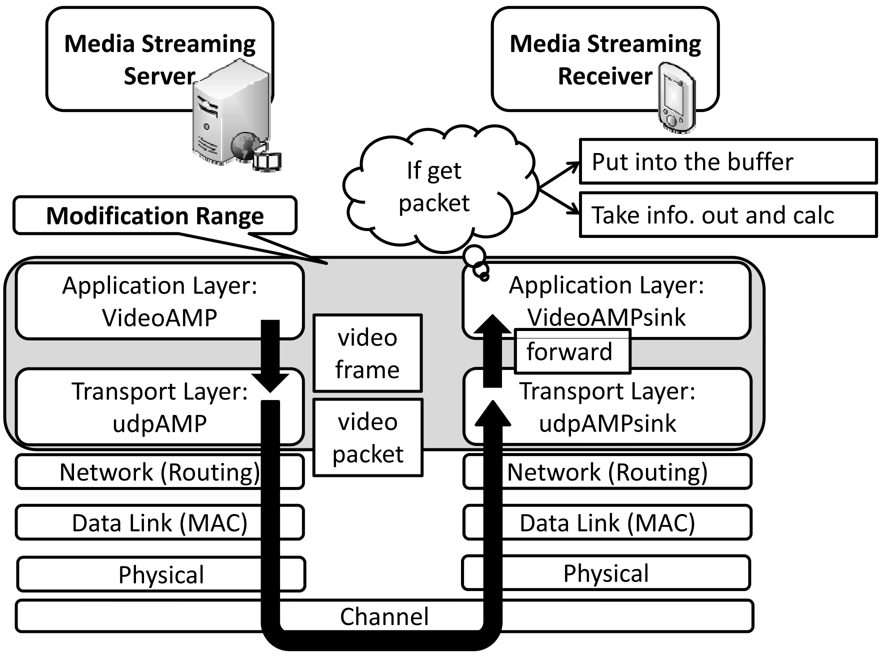


Figure 16: Network layers built in NS-2

larger than a value. It means that media player of the receiver starts to playout the video. *Playout_ bar* will trigger itself every $\mathbb{T}$ to be default playout interval when the playout rate is $1/\mathbb{T}$ fps. When *Playout_ bar* play the indicated frame at specific playout time, it will send a request to *Playout_ req* to search a frame which *Pts* is as the same as it wants. When *Playout_ req* gets the requirement from *Playout_ bar*, it will count the sum size of packets which fit *Playout_ bar* requirements. When the sum is same as frame size at the specific playout time required by *Playout_ bar*, *Playout_ req* will send the requirement success message back to *Playout_ bar*, with the frame size and the time stamp information, to let *Playout_ bar* write the information to the file. After playout procedure executed, *Playout_ buffer_ update* function updates the packets' situations of the buffer and the buffer size.

5.2 Simulation Environment Setup

In previous chapters, we have already known that the video streaming will be interrupted when the MN does the heterogeneous handover from WiFi AP back to WiMAX BS, so we will just observe the result in this duration. To simplify our simulation, we set the speed of the MN are stable in each simulation. About the characteristics of the video coding and the network settings are listed in Table 8, and the topology is similar to 1 but only discussing the MN leaves from a WiFi AP

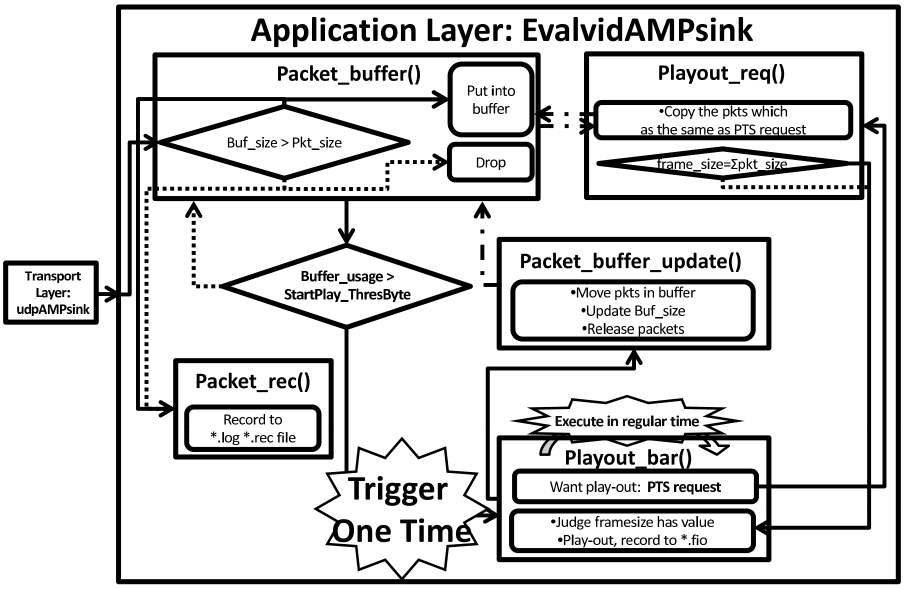


Figure 17: Functions in receiver Application Layer

signal range to WMAN/WWAN BS signal range (not limited in discussing WiMAX 802.15e-2005 BS) such event.

To test our proposed AMP, we let the MN only detect the channel and collect the received power to get the value of channel environment without executing AMP. We allow MN to execute AMP when d_{κ}^{HO} has been already smaller than d_{min}^{AMP}, so it should immediately adjust the playout interval when it becomes true.

Without considering all the retransmission behaviors, we will just present the outcome before handover, to observe how deadline postponing to frames playout when handover occurred, frame received status until handover, etc.

About the performance of our proposed AMP executed on video streaming, we use some representable evaluation ways to observe the outcome. We will compare our proposed AMP with **non-adaptive**, **conventional AMP scheme**, and **APTA scheme**. We will introduce the schemes in the following sub-Sections.

5.2.1 Conventional AMP Scheme

There are many papers design their $\mathbb{P}$ based on the number of remaining frames, and it is useful in some simple cases. They set that $\mathbb{P}_{incr}$ and $\mathbb{P}_{decr}$ relates to the remaining frames in playout buffer. The relation between the number of remaining

Notation		Description
Video sample name	:	foreman_cif repeat
Video coding format	:	MPEG-4
Coding/Transmission rate	:	1.5 Mbps
GOP (Group of Picture)	:	12 (IPPPPPPPPPPP)
Frame rate	:	25 frame per second
Max fragment size	:	1388 bytes
Buffer occupation and playout	:	30K bytes
Playout buffer size	:	3 MB
Random movement	:	OFF
Routing Protocol	:	DSDV
Propagation model	:	shadowing network
β (path loss exponent)	:	2.7
σ_{dB} (Shadowing derivation)	:	4.5
Transmission Power	:	0.6 Watt
Transmission Frequency	:	2.412e+9 Hz
RX_Thresh (received threshold)	:	2.12277e-10 Watt
Distance sample interval	:	0.2 sec
Section decision frequency	:	1 sec
MN probable speed	:	1 to 15 m/s
$t^{Duration}$	:	0.4 sec

Table 8: Simulation parameters

frames and $\mathbb{P}$ is like:

$$\begin{cases} \mathbb{P} = \mathbb{P}_{incr,MAX} \times \frac{\chi_L - \varsigma}{\chi_L} & , \varsigma\ remaining\ frames,\ \varsigma \in \mathbb{R}^+,\ \varsigma = [0, \chi_L) \\ \mathbb{P} = 0\% & , \chi_L\ to\ \chi_H\ remaining\ frames \\ \mathbb{P} = \mathbb{P}_{decr,MAX} \times \frac{\varrho - \chi_H}{\chi_L} & , \varrho\ remaining\ frames,\ \varrho \in \mathbb{R}^+,\ \varrho = (\chi_H, \chi_H + (\chi_L - 1)] \\ \mathbb{P} = \mathbb{P}_{decr,MAX} & , \varrho\ remaining\ frames,\ \varrho \in \mathbb{R}^+,\ \varrho = (\chi_H + (\chi_L - 1), \infty) \end{cases} \tag{5.1}$$

referred from [1], after we have already known that in normal condition the playout buffer always has χ_L to χ_H remaining frames based on long-term monitor. By definition and from some constraints, we know that in our simulation, χ_L is 3 and χ_H is 4 in normal environment, so the conventional AMP will keep the number of frames in playout buffer as best it can in the whole of simulation. Obviously, this scheme is not suitable to be used in this scenario because it does not have any element to predict handover, etc. Because the video type is live-streaming, and it is quite easy be impacted by channel, so we could imagine that even though it tries to keep ς not below than 3, it still has large probability to lower this value when the MN is in *Shadowing Zone* and closer to the edge of WiFi signal range.

5.2.2 APTA Scheme

APTA (Arrival Process Tracking Algorithm) is named by [19]. It refers **video frames** arrival rate and number of remaining **video frames** in playout buffer to decide $\mathbb{P}$ ([19] supposes that every packet could be treated as a complete, individual video frame). Like other papers, it separates the buffer status to be Safety Zone and Warning Zone. It also uses a quadratic function like 2.5 rather than the linear function to adjust the playout rates in Warning Zone, and it claims that it could reduce VDoP. Different from other papers $\mathbb{P}$ boundary setting, its range of $\mathbb{P}$ is much larger than others to prevent buffer overflow/underflow. Comparing with the conventional AMP mentioned before, we could imagine that it is more sensitive to channel condition, and it could do a better decision when it finds out that the arrival rate is lower than normal, although it does not know the phenomenon is caused by shadowing network. It should have less packet loss and buffer underflow before the handover than non-adaptive and conventional AMP schemes.

5.2.3 AMP Scheme without Section Schedule

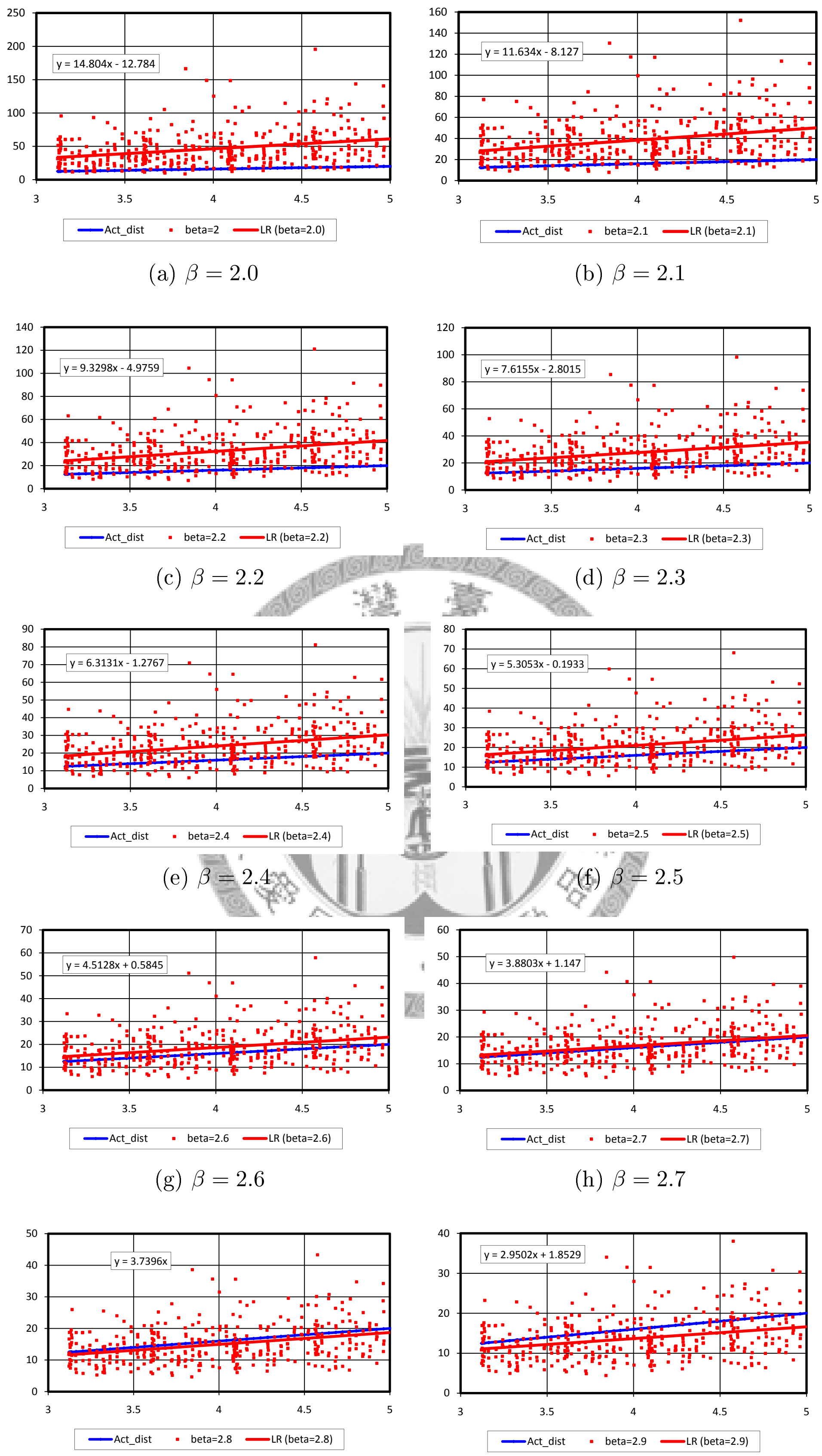
In Section 5.4, our proposed AMP will compare with the scheme which does not have the concept of *Section* schedule. The scheme is like our proposed AMP except

Section definition, so it knows when does the MN do the handover, the distance between the MN and WiFi AP, and the handover duration. About without considering *Section*, it means that it does not do any AMP behavior until the remaining time to handover less than Equation 4.1. After that, the MN executes AMP by using $\mathbb{P}_{incr,MAX}$ until handover finished or it finds out that the equation $t_{\kappa}^{HO} \leq t_{min}^{AMP}$ is excluded. Although from Equation 4.1 knowing that we still have enough time to store frames preparing to handover, it is uncertain about the condition of t_{κ}^{HO} because it relates to d_{κ}^{HO} and $speed_{\kappa}$, is variable. Thus we could imagine that after the scheme executing AMP the MN speeding up, it still could not store enough frames comparing with our proposed AMP, and contrarily, after the scheme executing AMP the MN speeding down or stop or turns its direction, it will store redundant frames. We also have to consider the deviation from distance measurement impacting on d_{κ}^{HO}, and it will cause the scheme executing too early or late. Comparing with above two schemes, it may have less video interruption duration during the handover.

5.3 Movement Speed Keeping in Stable

First we use 4 m/s speed of the MN to be an example. We could use the method mentioned in Section 3.1.2 to get the value of β and σ_{dB} like Figure 18 and Figure 19, which the values of received power is got from the receiver MAC Layer. The former is to search the β actual value used in this scenario, and we just show the search range from 2.0 to 2.9 and do the linear regression and get different linear relation between the time and the distance between the WiFi AP and the MN. From the figure we could find out that it is enough when we sample for 2 seconds to get β. The latter is the relation between the sampling time and the difference between the calculated and actual σ_{dB} and it shows that we sample for 5 seconds at least and get a closer value. After we get β and σ_{dB}, we could start to calculate and get the probable distance between the MN and the WiFi AP like Figure 20. The calculated distance means the distance is got from received signal, sampled for a while for collecting enough samples, and the actual distance is to compare with calculated distance, and the function of the updated distance is to prevent high variation of *Section* changing during AMP execution. The points would be the schedule's checkpoint to check whether AMP is already on the schedule or not. We have to get a balance between the degree of schedule accuracy and the packet storing efficiency.

From Figure 21, it shows the advantage of handover time prediction mechanism and the *Section* scheduling, so it will start to do AMP much earlier and more positive



(a) $\beta = 2.0$ (b) $\beta = 2.1$

(c) $\beta = 2.2$ (d) $\beta = 2.3$

(e) $\beta = 2.4$ (f) $\beta = 2.5$

(g) $\beta = 2.6$ (h) $\beta = 2.7$

(i) $\beta = 2.8$ (j) $\beta = 2.9$

Figure 18: The linear regression equations got from different β when the sample duration is 2 seconds

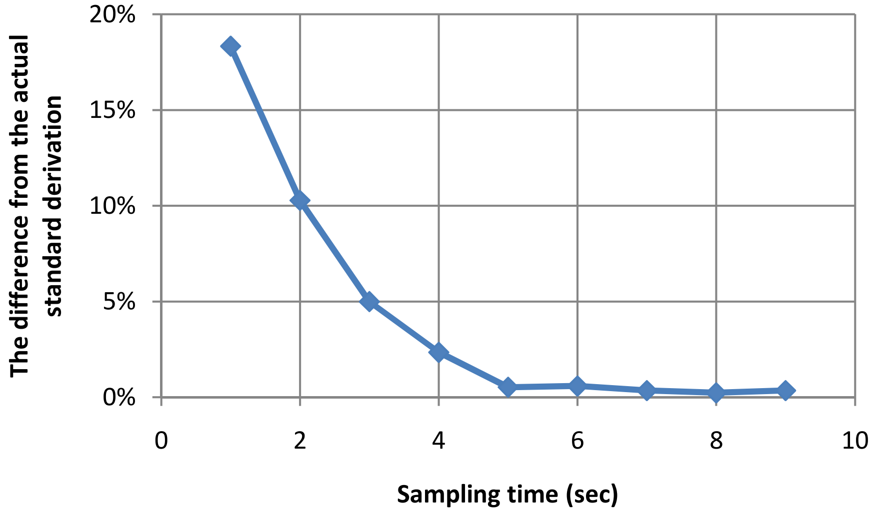


Figure 19: The relation between sampling time and the difference between calculated and actual σ_{dB}

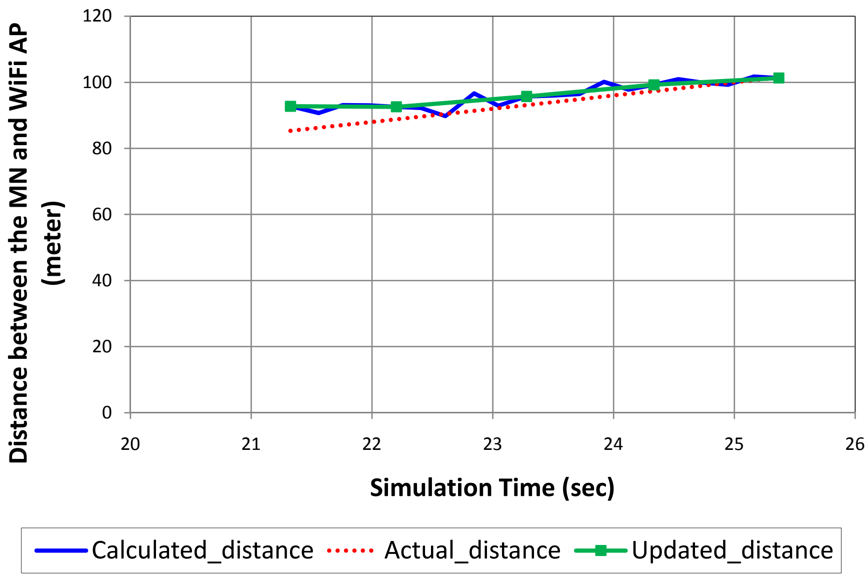


Figure 20: The Relation between the Actual Distance, Calculated Distance

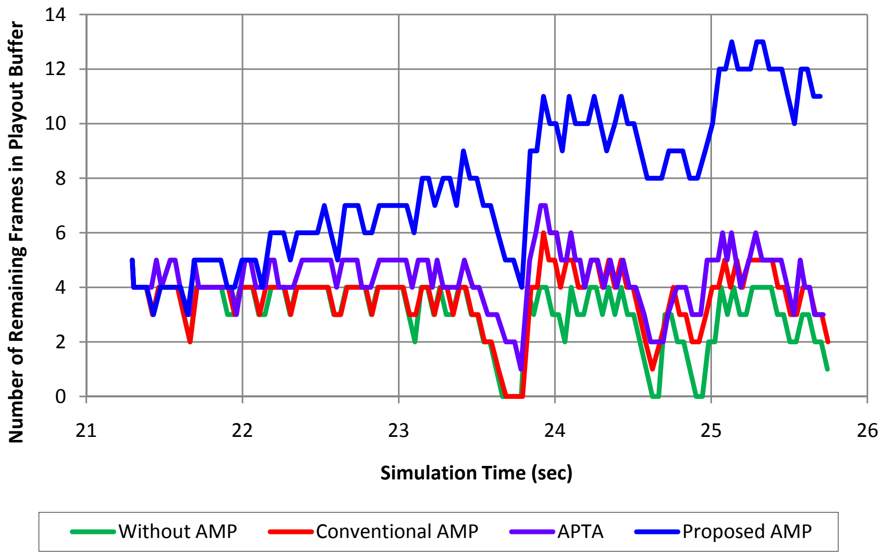


Figure 21: Number of remaining frames in playout buffer when a frame playout

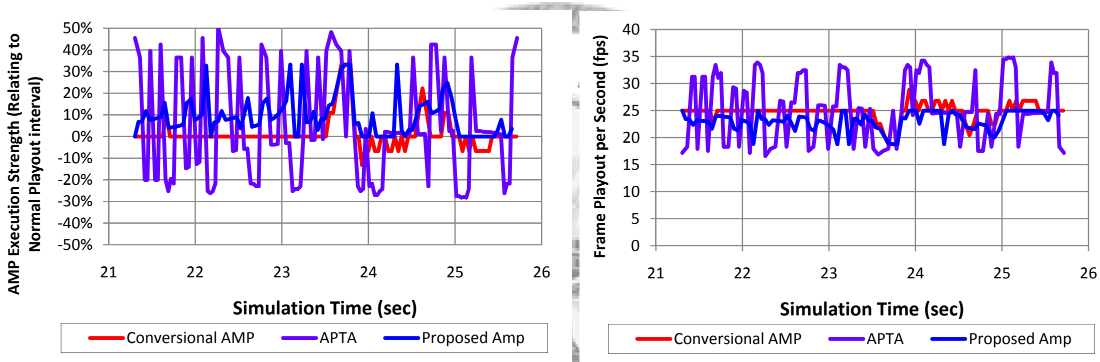
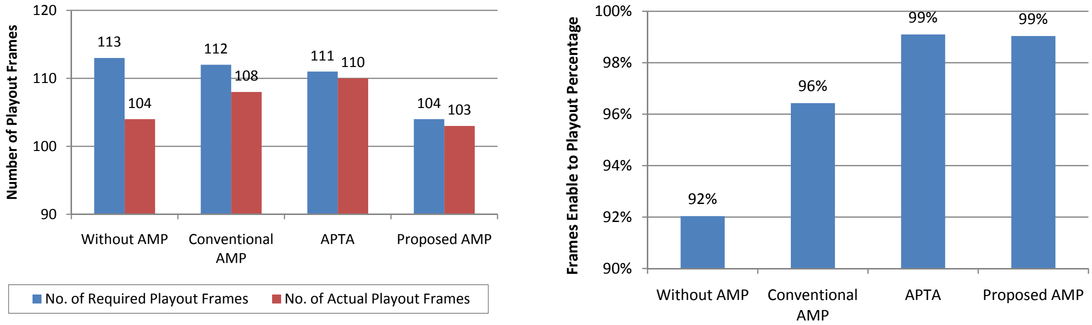


Figure 22: Relation between $\mathbb{P}$/fps and simulation time

than *Conventional AMP* showed in Figure 22. We could directly observe the last playout frame before handover and know that *Without AMP*, *Conventional AMP*, *APTA*, and *Proposed AMP* already stored 1, 2, 3 and 11 frames, respectively. *Proposed AMP* tries to overcome the efficiency of frame storing reducing due to playout and shadowing network, and the latter will cause the huge impact when the MN is close to the edge of WiFi AP signal range, while *Conventional AMP* only tries to prevent buffer underflow. *APTA* has a more number of frames storing and no buffer underflow event happening because it considers frame arrival rate and buffer status at the same time.

The reason why there are some obvious gaps happened in the figure is that, we set the GOP is 12. The value of GOP will impact the efficiency of frame storage when receiving I-frame based on the retransmission mechanism (REQ). The worse quality



(a) The number of required/actual playout frames

(b) The percentage of required/actual playout frames

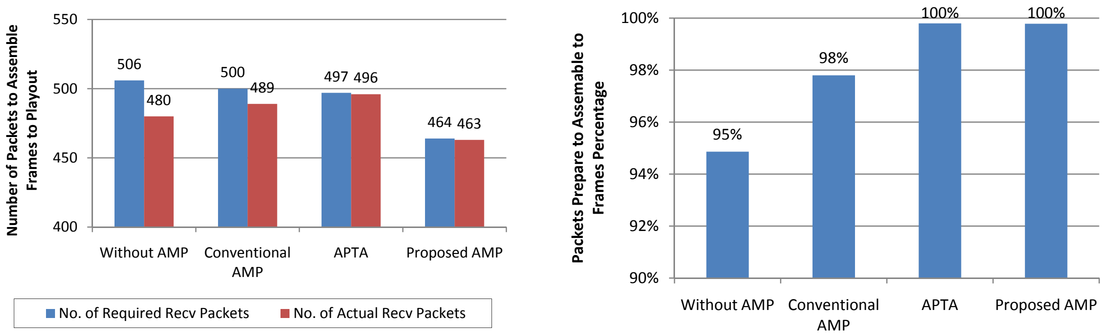
Figure 23: The number of completely received and playout frames

of power signal, the lower percentage of packet successfully received for one time, and the lower efficiency of frame storing.

The reason of why the curves overlapping percentage of *Without AMP* and *Conventional AMP* is high in the first half of time before handover but low in the last half of time is that, *Conventional AMP* tries to satisfy Equation 5.1 easily when the impact degree of shadowing network is light, to let the video packets successfully received without or seldom using REQ mechanism. And we could also explain that why in Figure 22 the *Conventional AMP's* curve is stable in the first half of time before handover. When the shadowing network seriously impacts the efficiency of frame storage, *Conventional AMP* will begin to exert itself, but its target is still to keep "enough" frames in the playout buffer, so it is the reason why the curve in Figure 22 is up and down when the MN is close to the edge of the WiFi AP signal range. From the figure, we also find out that the variation of *APTA* curve is large, although it has a better performance of frame storing. We think that the algorithm makes $\mathbb{P}$ easily to change to another value which is quite larger or smaller than previous $\mathbb{P}$ due to high sensitive of frame arrival rate, and it is the purpose in setting *Section* decision frequency to a suitable value in *Proposed AMP*.

Figure 23 and 24 show the condition of required and actual number of playout frames/packets before handover, and all types of AMP have more frames/packets be able to be playout than *Without AMP*, means it is meaningful to postpone frames deadline, especially in shadowing network case. Because of frame storing for handover, *Proposed AMP* playout less frames to get its purpose and it achieve the goal already showed in Figure 21.

The situation of every frame received percentage showed in Figure 25 is a good



(a) The number of required/actual packets (b) The percentage of required/actual packets

Figure 24: The number of packets preparing for assembling to frames

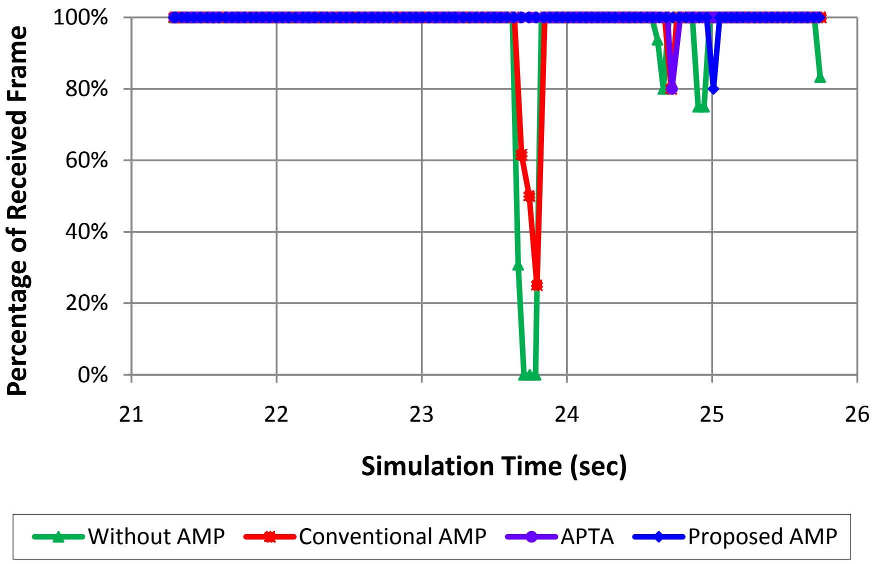


Figure 25: Percentage of received packets per playout frame

way to let us know the importance of frame deadline postponing. Because *APTA* and *Proposed AMP* both consider the channel condition to adjust their $\mathbb{P}$, they both have high packet received percentage. All of the three AMPs certainly work to prevent buffer underflow, and we could see the phenomena from about 23.57 to 23.72 sec simulation time. Because the media player has to wait for whole packets received (totaling 13 packets), separated from 60^{th} frame, and continuing receiving following packets, it will reduce frame produced efficiency, and the efficiency is also impacted by the shadowing network. Because it wastes too much time to collect all the packets to assemble to the frame, it also impact the several frames storing efficiency after 60^{th} frame, and it is the buffer underflow reason, showed in Figure 21 and 25. And in this duration we could find out that $\mathbb{P}$ in the three AMP schemes obviously rise to a high

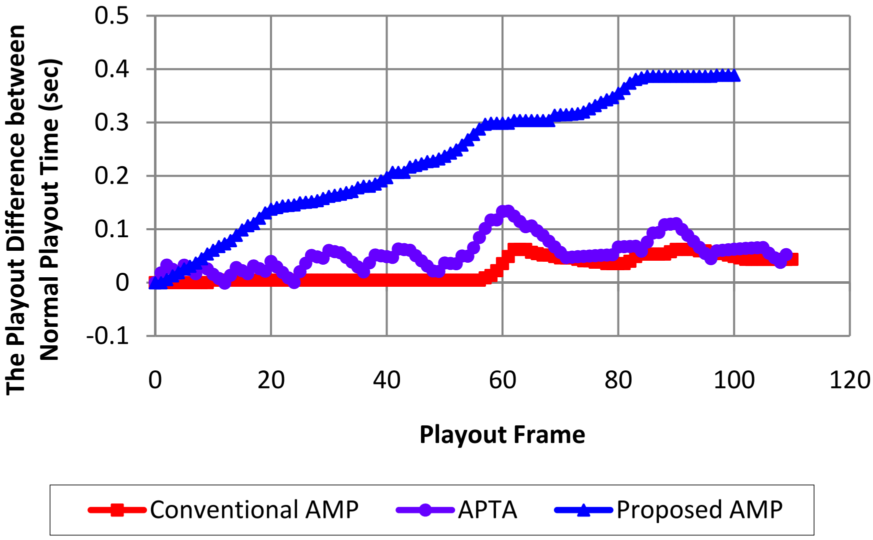
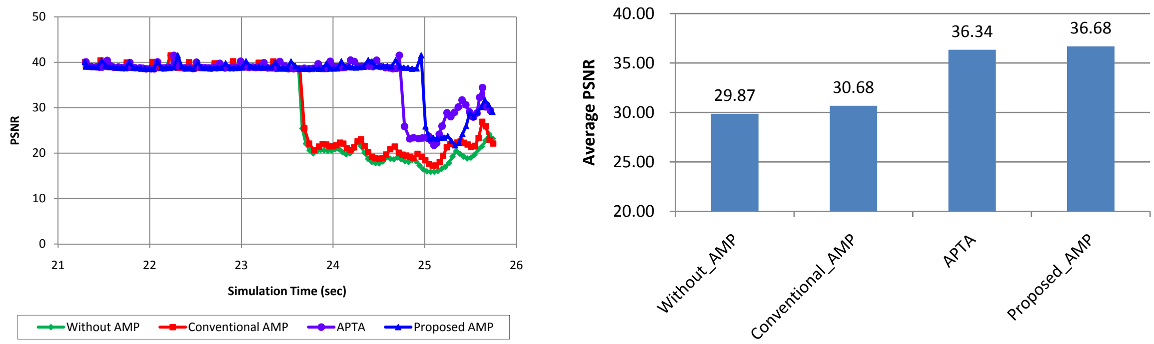


Figure 26: The playout difference of every frame between different schemes and normal scheme

value showed in Figure 22, but only *APTA* and *Proposed AMP* successfully prevent buffer underflow; *APTA* finds out that the frame arrival rate is low and adjusts $\mathbb{P}$ earlier than *Conventional AMP*; *Proposed AMP* finds out the frame storing efficiency could not catch the schedule's requirement if it keeps present $\mathbb{P}$, so it adjusts $\mathbb{P}$ to get the goal.

Figure 26 shows how deadline the different AMP schemes postpone to playout. From the figure, we can find out that at the handover time, the frame deadline of *Proposed AMP* postponing achieves about 400 msec, larger than *Conventional AMP* and *APTA* schemes about 50 msec. It is quite the same as the Figure 21 showing, due to AMP schemes using deadline postponement approach to store frames, and the approach could also make the video interruption be postpone. Our proposed AMP based on the characteristic to hope that the link could be re-constructed between the MN and video server before the video interruption, which is already postponed by AMP. From the figure, we could find out that *Proposed AMP* could support larger frame delay during the handover and has larger probability to prevent video interruption during the handover.

Figure 28 shows the different schemes video quality. We have to understand that we could not find a precise frame number to do the comparison because the characteristic of AMP. We sample the frames which the MN in handover duration, and the frames transmitted in the duration from the media server will be received successfully after the handover finished. From Figure 21 we know that video interrupted happens



(a) PSNR curve comparison (b) Average PSNR comparison

Figure 27: PSNR comparison

after handover begun in *Without AMP*, *Conventional AMP*, *APTA*, and *Proposed AMP* are about $\mathbb{T}$, $2\mathbb{T}$, $3\mathbb{T}$, and $11\mathbb{T}$, respectively. We have already known that it is no meaningful when they receive video frames which are already over playout deadline. So the conclusion is that, the video quality should become worse in handover duration in *Without AMP*, *Conventional AMP*, and *APTA* schemes. But at the same time, *Proposed AMP* could keeps the video quality even though it also does not receive any frames from the server. Here we compare the frames in different schemes playout in the handover duration. *Proposed AMP* is compared with 118^{th} and 117^{th} video frame playout by *Without AMP* and *Conventional AMP* schemes, respectively, which is already known that the frame is playout in the handover duration but received after the heterogeneous handover and thus the frame are over the playout deadline. About *Proposed AMP* case, it needs to playout 118^{th} frame after handover and it also successfully receives the frame. We are also known that at the same time our proposed AMP is playout 106^{th} video frame, but after we use YUVviewer [62] to get the frame we find out that it is not the same picture, so we show the probable pictures in Figure 28. The first two pictures are 117^{th} video frame of *Without AMP* and *Conventional AMP*, and the remaining pictures are *Proposed AMP* from 106^{th} to 118^{th} frames.

And Figure 27 shows the PSNR curve and the average values, and we could use the outcome to compare with Figure 25. A frame which is not received successfully will cause PSNR reduce, no matter what the mechanism we use, and we find out the the average PSNR of *Proposed AMP* and *APTA* schemes are much higher than other two.

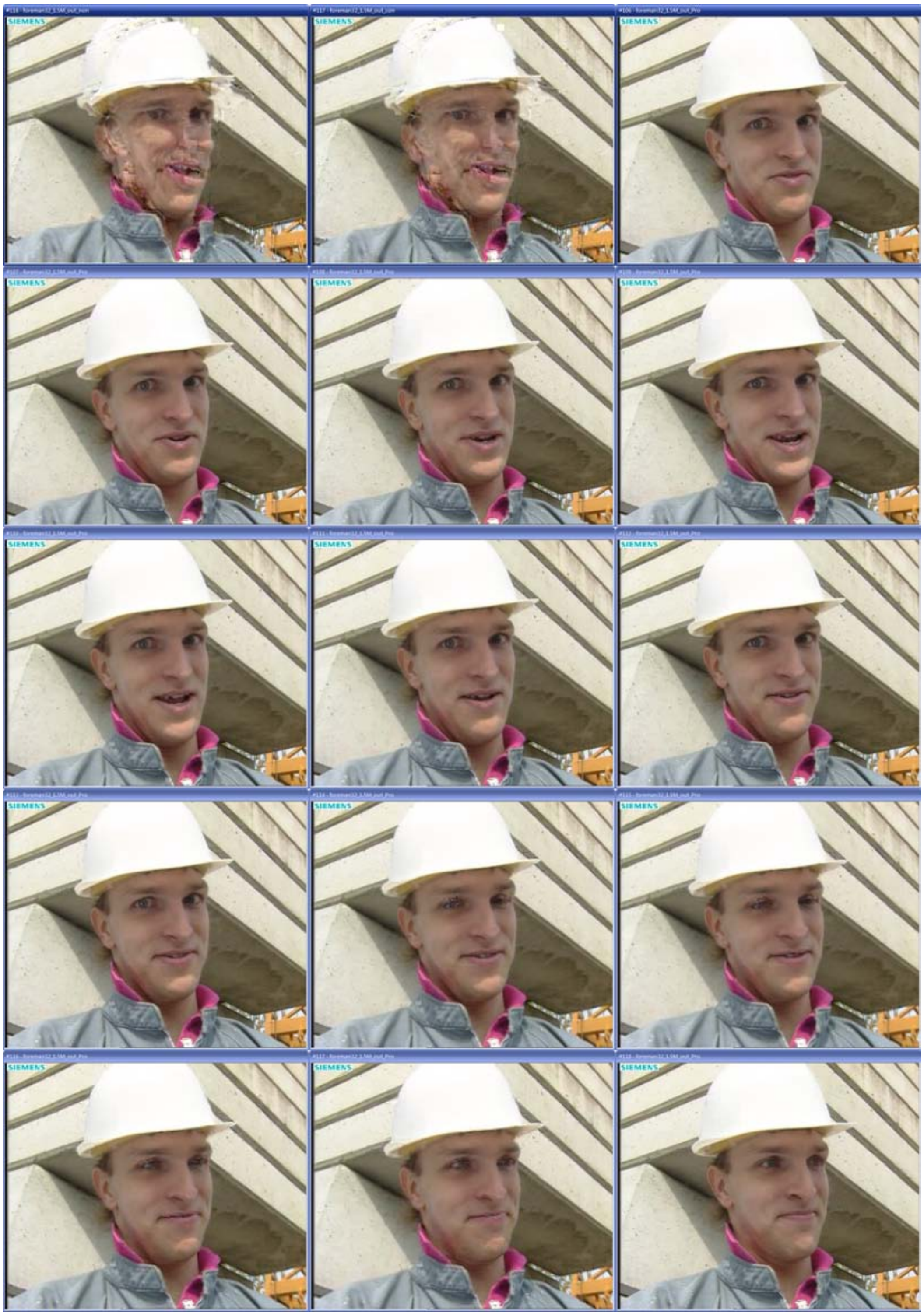
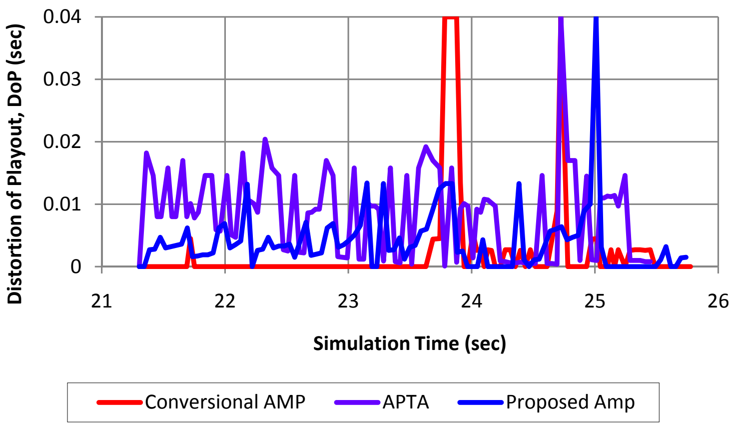
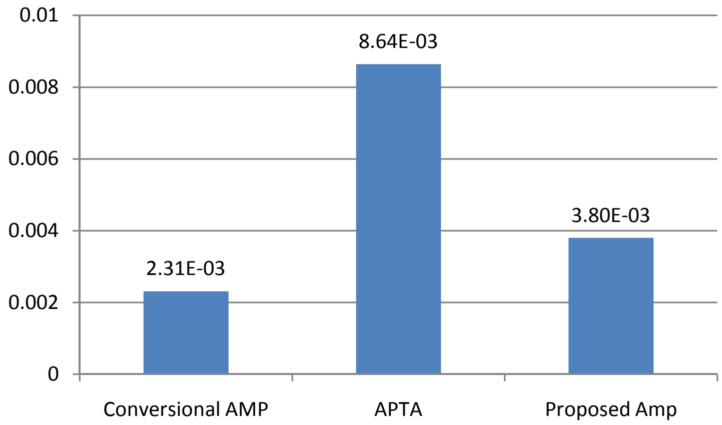


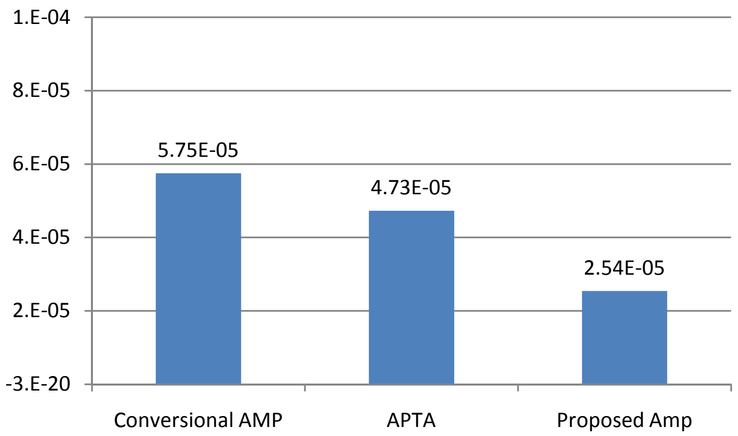
Figure 28: Frame quality comparison by visual inspection



(a) DoP curve comparison



(b) DoP value comparison



(c) VDoP value comparison

Figure 29: DoP and VDoP comparison

Finally, we will compare the value of DoP and VDoP using Equation 2.3, and the outcome of DoP and VDoP in different schemes showed by Figure 29. Here we redefine the definition of first sub-equation to be the same as the normal playout frame, because the decoder will still try to repair and conceal injured frames and playout, so we will not treat the frame's DoP particularly which is the next frame of injured frame.

We find out that although *Proposed AMP's* average DoP is worse than *Conventional AMP*, the former gets better VDoP than the latter eventually, even it does not do any roll-off behavior. And the high DoP and VDoP on APTA reason is mentioned before and easy to get; although *APTA* does not have less number of unable playout frames, the high playout interval variation and high $\mathbb{P}$ decision to make the score worse.

In summary, *Conventional AMP* is suitable to be used in non-live streaming, unpredictable piecemeal, short packet delay interval, so it does not have a plan to prevent "predictable" burst packet delay. Although *APTA* has its own frame arrival monitor to be one of main $\mathbb{P}$ decision elements, the algorithm still leaves some space to improve. Even though *APTA* has a worse performance in DoP and VDoP, *APTA* is a good way to detect the channel condition. Contrarily, *Proposed AMP* uses a series of mechanisms to create a storing frame schedule to prevent video streaming interruption from burst packet delay due to heterogeneous handover.

5.4 Accelerating Movement Speed before Handover

In our opinion, we could not guarantee that the MN keep its speed stably, so the remaining handover time would be change, and it is the purpose of *Section* definition. To show the concept of *Section* in our proposed AMP is useful and important, we change the mobile speed from a constant value to two values to compare with the scheme with almost the same as *Proposed AMP* except *Section* existence. The scenario becomes that: a MN keeps its speed in 4 m/s for a while, and after that it accelerate its speed to 10 m/s at the 24.6^{th} second and keeping the speed to handover. We adjust the "*Section* decision frequency (1 second in default)" to be as same as "Distance sample interval (0.2 second in default)" to increase *Section* updated frequency, so it means that every 0.2 seconds, AMP core decision part will receive d^{HO} from handover time prediction part and make its own $\mathbb{P}$ decision. Because we still have to consider the disparity between d^{ACT} and d^{STA}, it is hard to avoid the difference between the number of frame required storing and actual storing both on the

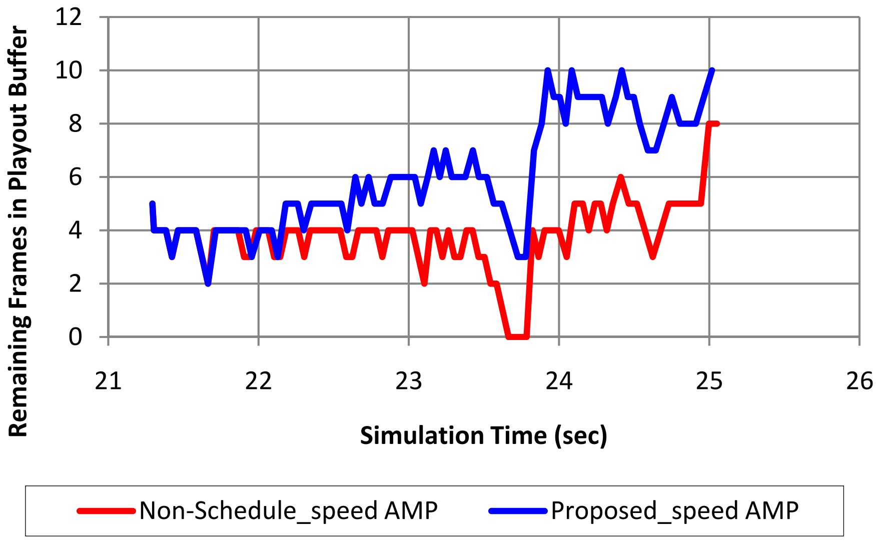


Figure 30: Number of remaining frames in playout buffer when a frame playout before handover

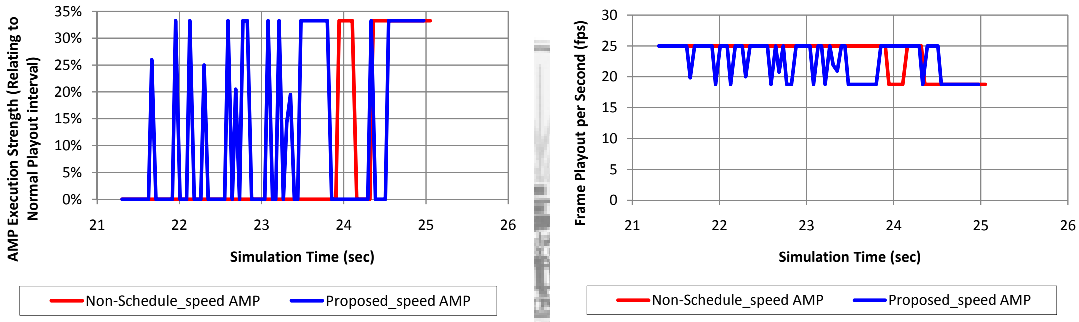
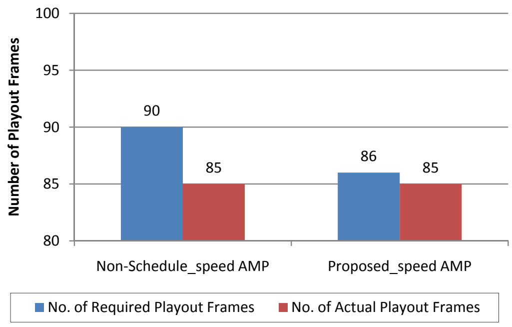


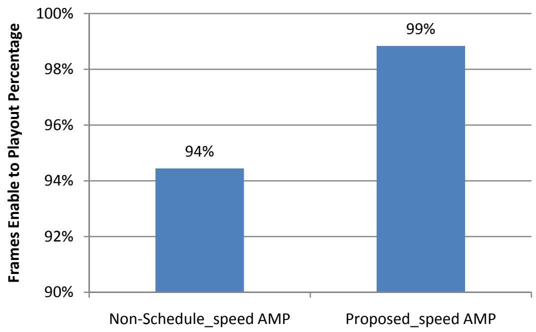
Figure 31: Relation between $\mathbb{P}$/fps and simulation time

two schemes, and adjusting the "*Section* decision frequency" may reduce the distance measurement impacting on AMP decision.

Figure 30 shows the number of frame storing before handover, and we know that about *Non-Schedule_speed AMP* case, it could not achieve the goal before handover. Here we already renamed from *Proposed AMP* to *Proposed_speed AMP* to compare with Non-Schedule_speed AMP which does not have a concept of *Section*. We could imagine that the worse result it would get if it executes AMP before the MN speeds up. Similarly, we also could imagine that it would also store more redundant frames when MN speeds down after it executes AMP, and the distance measurement deviation will have larger impact than *Proposed_speed AMP*. Although we know that *Non-Schedule_speed AMP* just needs to spend less time to store frames by calculation, it exists some risks when it start to store frames, such as distance measurement deviation



(a) The number of required/actual playout frames

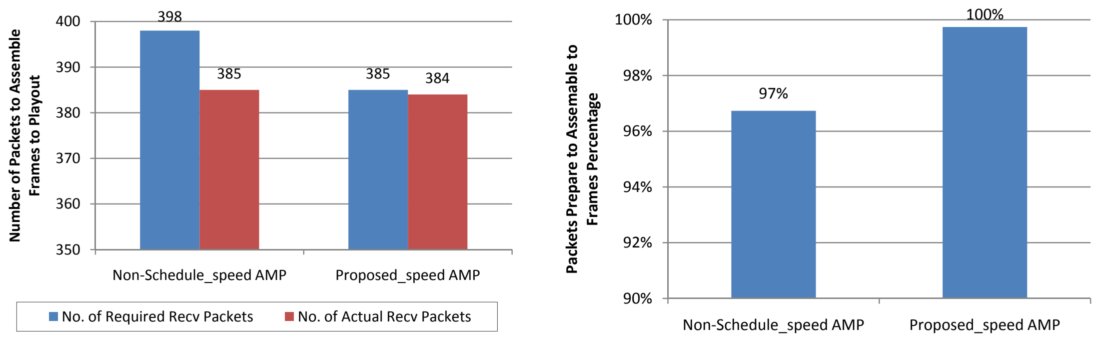


(b) The percentage of required/actual playout frames

Figure 32: The number of playout frames

which is showed in Figure 31, the efficiency of frame storing when it is close to the edge of the WiFi signal range, the degree of shadowing network effects becoming seriously impact the packet received condition. In the beginning of Section 5.3, we have already knew that the purpose of AMP is to overcome the shadowing network effect to prevent buffer underflow in *Conventional AMP* and *APTA* schemes, so we know that it should be execute a suitable AMP at least in this duration to prevent packet loss and buffer underflow events happen, not just concentrating on frame storing preparing for the handover. Furthermore, the large variation in $\mathbb{P}$ reason is that the higher *Section* updated frequency than last simulation. It will cause AMP having a higher frequency to be monitored its schedule and let the variance of $\mathbb{P}$ be higher, but it also makes *Proposed AMP* do a quickly response to adjust $\mathbb{P}$ to keep updated schedule and current schedule in the same or nearby *Section* and thus causing $\mathbb{P}$ be trembled. Figure 32, Figure 33, and Figure 34 show the shadowing network effects on packet received and frame playout.

Figure 35 shows the playout delay difference between *Non-Schedule_speed AMP*, *Proposed_speed AMP* with normal playout time. It is late for *Non-Schedule_speed AMP* when it finds out that there is not enough time to store frames then starting to execute AMP before speeding up. From the figure we also find out that *Non-Schedule_speed AMP* both two schemes pause in executing AMP due to distance measurement deviation, and the phenomenon becomes to a serious problem to reduce the number of storing frame. The problem makes a much large gap between the goal in *Non-Schedule_speed AMP* scheme, but only causing limited gap between the goal in *Proposed_speed AMP* scheme, and it could tell us that it is quite important to use the concept of *Section* in our proposed AMP.



(a) The number of required/actual packets (b) The percentage of required/actual packets

Figure 33: The number of packets preparing for assembling to frames

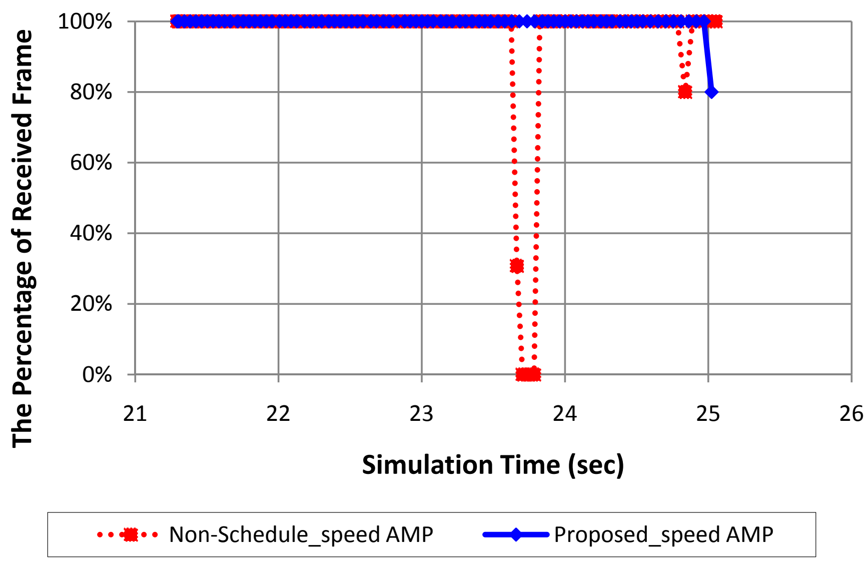


Figure 34: Percentage of received packets per playout frame

Figure 36 shows the outcome of PSNR and Figure 37 shows the outcome of DoP and VDoP. It is no doubt the PSNR outcome of *Non-Schedule_ speed AMP* occurred at about 23.7^{th} second because of without AMP execution and causing packet loss, and it also causes large local DoP to make the average DoP is worse than *Proposed_ speed AMP* scheme. Because *Proposed_ speed AMP* only has one frame unable to playout in the end of simulation, it becomes a main reason to make the outcome of VDoP is better than *Non-Schedule_ speed AMP*. It is not difficult to imagine the outcome of DoP and VDoP. Because none of them let packet loss happen and *Proposed_ speed AMP* stores more frames than *Non-Schedule_ speed AMP*, the DoP score the former is worse than the latter. The high variance of $\mathbb{P}$ impacts the VDoP score of

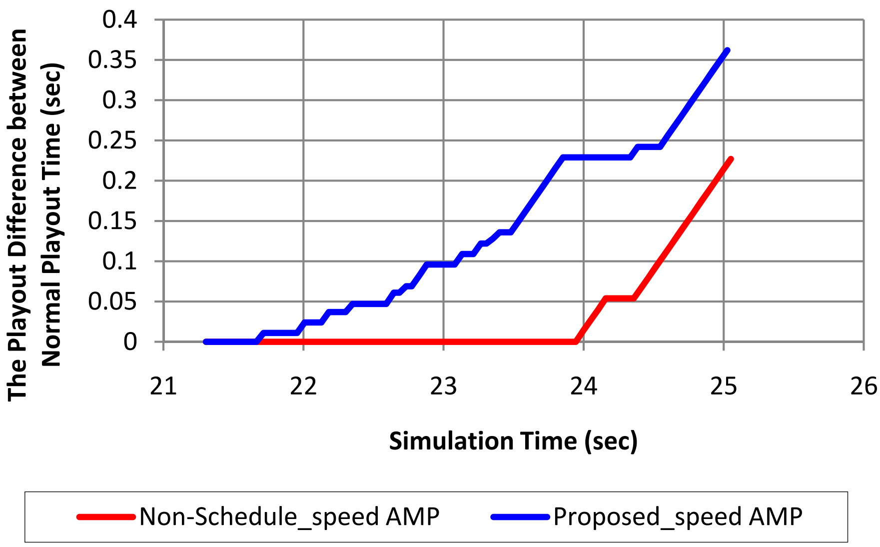
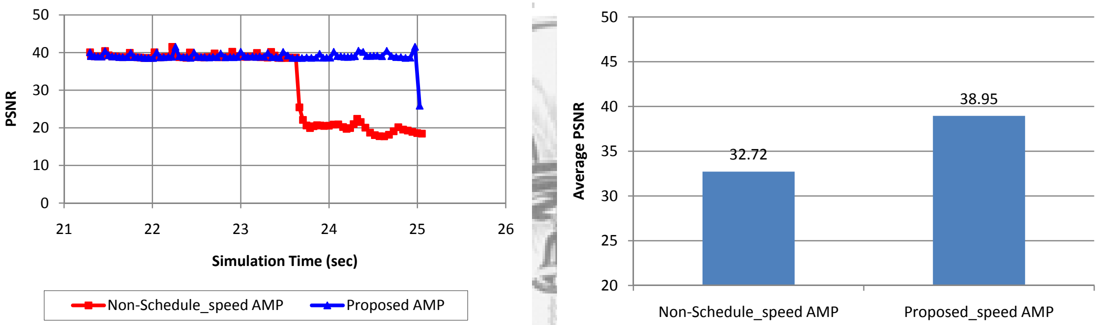


Figure 35: The playout difference of every frame between different schemes and normal scheme



(a) PSNR curve comparison (b) Average PSNR comparison

Figure 36: PSNR comparison

Proposed_ speed , and the square-wave-like curve impacts the VDoP score of *Non-Schedule_ speed AMP*. Here we do not show the outcome about PSNR because they both have no frame dropped.

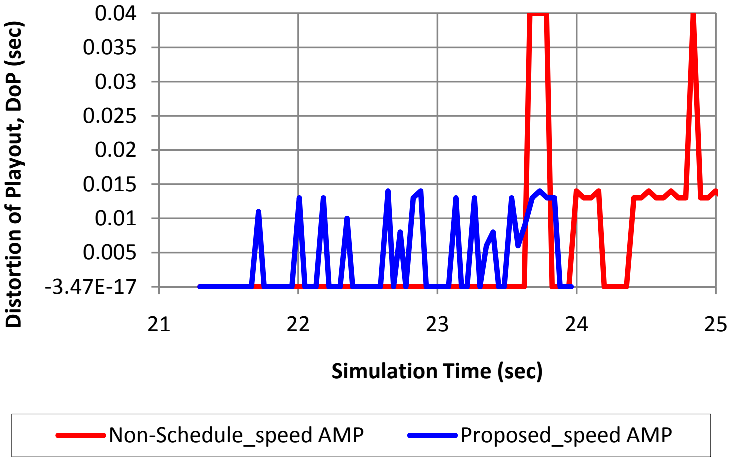
We could get a conclusion from a series of simulations above. Because our proposed AMP has handover time prediction, handover duration prediction parts, and it means that it could get some important information from outer environment to predict the handover occurrence time. Furthermore, AMP core decision part based on the concept of *Section* could create a schedule to check the progress of frame storing to prevent some events like speed up or down, etc. Although there are some benefits of our proposed AMP, it still has some trade-off and limitation.

The concept of AMP is to postpone frames playout time to prevent frames dropped due to over playout deadline, but it also changes the duration between transmitted and playout time ($t^{\varphi}_{\mathtt{j}}$), and it will make the live-streaming requirement harder to accomplish. Here we define that for all $t^{\varphi}_{\mathtt{j}}, \mathtt{j} = 1, \ldots$ could not exceed t^{φ} set by the user. This scenario is a good example to observe and explain the problem. If $t^{Duration}$ got from handover duration prediction part to AMP core decision is larger than t^{φ}, it should have no any approach to prevent video interruption. Some time-sensitive types of media could set t^{φ} to be a suitable value to get more comfortable on interaction after referring the network condition, like 400 msec set in [63]. Be remember that the parameter setting will seriously impact the human feeling and AMP efficiency.

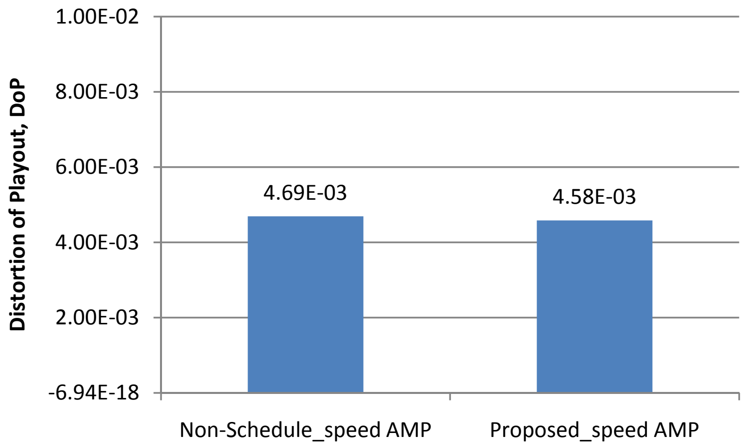
Our proposed AMP still has some space to improve and listed below:

- The way to make VDoP outcome to a much lower value. Although our proposed AMP gets a good score about VDoP due to low packet loss before the handover, the variation of $\mathbb{P}$ is still quite large and it will make the audiences uncomfortable. Due to above reason, we could use roll-off function to reduce VDoP or even getting a balance in frame storing efficiency because it has a trade-off between of them. In Chapter 6.1, we will use roll-off approach already mentioned in Section 2.2.2 to construct an optimization framework. We could use the optimized outcome to the algorithms in AMP core decision part and try to minimize the VDoP value.

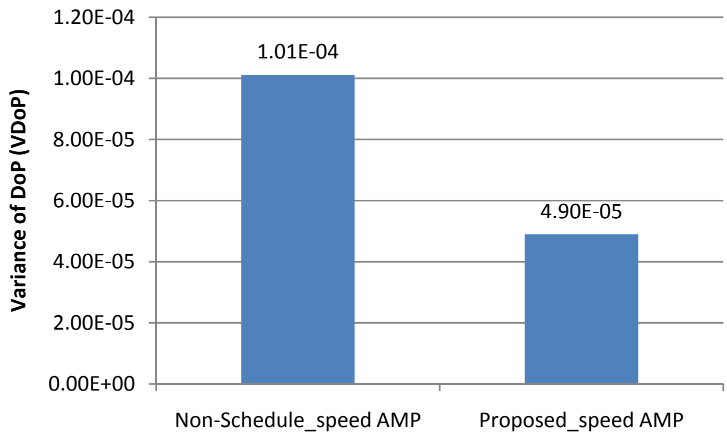
- The precision of d^{HO}, t^{HO}, and $t^{Duration}$, and it should have some ways to improve. Some subjects may useful for the situation like vehicle movement prediction. And we believe that if we could get some additional information from the MN besides packet received time, received power and current speed like GPS, it should also improve the precision of the outcome. About the improvement on handover duration prediction, we should analyze the structure of backbone networks in real case and time waste.



(a) DoP curve comparison



(b) DoP value comparison



(c) VDoP value comparison

Figure 37: DoP and VDoP comparison

Chapter 6

EXTENSION AND CONCLUSION

From the simulation result of Chapter 5, we find out that although our proposed AMP got a better VDoP than other cases, there are some rapid risen or dropped events happened in the DoP curve, and this phenomenon will impact VDoP score. Thus, in Section 6.1 and 6.2, we will use roll-off approach to let the curve risen or dropped slighter, considering the frame storing/releasing efficiency, and finally we could get a much better VDoP. Then we could get the conclusion from previous simulation results in Section 6.3, and some plans could be added are mentioned in Section 6.4

6.1 Roll-off Parameter Settings

In this section, we use another viewpoint to get a suitable roll-off function to minimize local VDoP in some special cases. First we discuss the profile of our proposed roll-off function execution to convenient to formulate our objective functions and constraints about VDoP minimization. To let the roll-off function more practical, we consider the situations of current MN and network environment to decide the function, like the MN current speed, the remaining time changing to another $\mathbb{P}$, the gap between the last and goal of $\mathbb{P}$, and the current *Section*. Because the problem is a non-linear programming, we will run the simulation after we get the objective functions and constraints and get the relation between the roll-off execution time and accumulation additional delay in different kinds of cases. At last we will get the relation between $\mathbb{P}$ and execution time to be our roll-off function.

Table 9 below defines some parameters about roll-off function, and we will explain the parameters in detail. We just use $\mathbb{P}_{incr}$ ignoring $\mathbb{P}_{decr}$ to represent simplify description if a parameter relates with AMP execution strength.

Different from using conventional roll-off function and trying to adjust coefficients to get the goal, we use top-down method to approach our objective. The meaning is that we first try to know the relation between each frames and their delays causing minimum VDoP in some conditions, and then we transform the result to roll-off function and get the goal.

Notation	Description
m_i	: Frame i received time, $m_{i+1} - m_i = \mathbb{T}$, $i \in [1, \mathcal{P}]$, $m_1 = 0$ (sec)
n_i	: Frame i playout time, $n_1 = 0$, $n_{i+1} = n_i + \mathbb{T} + f_i$ (sec)
f_i	: Additional delay, $f_i = i\mathcal{K} + \mathcal{X}_i$, $\left\|P_{past}^{AMP}\right\| \mathbb{T} \leq f_i \leq \left\|P_{goal}^{AMP}\right\| \mathbb{T}$ (sec)
t_i	: Accumulate additional delay,$f_i = t_{i+1} - t_i$,$t_1 = 0$,$0 \leq t_i \leq \mathbb{T}$ (sec)
t_{start}^{ROLL}	: The time to start to execute roll-off function, divided into $t_{start}^{ROLL,F}$ and $t_{start}^{ROLL,B}$ (sec)
t_{stop}^{ROLL}	: The time to finish to execute roll-off function, divided into $t_{stop}^{ROLL,F}$ and $t_{stop}^{ROLL,B}$ (sec)
P_{past}^{AMP}	: AMP or Inv-AMP execution strength in the end of last *Section*, divided into $P_{incr,past}^{AMP}$ and $P_{decr,past}^{AMP}$ (%)
P_{goal}^{AMP}	: AMP execution strength in the beginning of next or further behind *Section*, divided into $P_{incr,goal}^{AMP}$ and $P_{decr,goal}^{AMP}$ (%)

Table 9: Parameter settings for AMP roll-off function

Here we begin to describe our formulation of objective functions and their constraints in different cases. To simplify our objective and constraint, we do the following environment settings, and it is easy to extend the supposition to general cases. Before the execution of the roll-off function in every time, the scheduler already gave two *Sections* (namely, $2 \times t_{\mathcal{J}}^{AMP}$) to execute the function, and in this duration the MN will receive $\mathcal{T}$ frames by defining $\mathcal{T} = 2 \times \left\lfloor \frac{t_{\mathcal{J}}^{AMP}}{\mathbb{T}} \right\rfloor$. We hope the function will only playout $\mathcal{T}$ frames or already playout $(\mathcal{T} + 2)$ frames when the MN receives $(\mathcal{T} + 1)^{th}$ frame and enters a new *Section*; the former is $\mathbb{P}_{incr}$ case executed in AMP mode, and the latter is $\mathbb{P}_{decr}$ case executed in Inv-AMP mode. Both cases also have to achieve the playout rate what the next *Section* wants.

6.2 Optimization Function Formulation

By already known the fundamental definition and characteristics of m_i, n_i, and f_i and the meaning of Figure 4 in Section 2.2.2, we start to formulate our objective function and its constraints. The formulations below only describe the magnitude of the playout duration and accumulation addition delay duration without the direction, so it is the reason that they are all added absolute symbol.

The first case is the function when $\mathbb{P}_{incr,past}$ is smaller than $\mathbb{P}_{incr,goal}$:

$$
\begin{array}{ll}
\text{[Case I]} & \\
\text{Minimize} & Var\left(t_i\right),\ for\ i=1:1:\mathcal{T}, \\
\text{subject to} & \left|\left(n_{i+1}-n_i\right)\right| \leq \left|\left(n_{i+2}-n_{i+1}\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-2\right), \\
& \left|\left(n_{i+1}-n_i\right)\right| \leq \left(\mathbb{T}\times\left|\left(1+\mathbb{P}_{incr,goal}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
& \left|\left(n_{i+1}-n_i\right)\right| \geq \left(\mathbb{T}\times\left|\left(1+\mathbb{P}_{incr,past}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
& n_1 = t_{\kappa}^{HO} = t_{start}^{ROLL,F},\ n_{\mathcal{T}} \fallingdotseq t_{\kappa}^{HO} - 2\times t_{\mathcal{J}}^{AMP} = t_{stop}^{ROLL,F}, \\
& \left|\left(t_{i+1}-t_i\right)\right| \leq \left|\left(t_{i+2}-t_{i+1}\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-2\right), \\
& \left|\left(t_{i+1}-t_i\right)\right| \leq \left(\mathbb{T}\times\left|\mathbb{P}_{incr,goal}\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
& \left|\left(t_{i+1}-t_i\right)\right| \geq \left(\mathbb{T}\times\left|\mathbb{P}_{incr,past}\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
& \left|\left(n_{i+1}-n_i\right)\right| = \mathbb{T} + \left|\left(t_{i+1}-t_i\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
& t_1 = 0,\ t_{\mathcal{T}} = \mathbb{T}.
\end{array}
\tag{6.1}
$$

means that it will use a higher efficiency trend to store an additional frame .

And when $\mathbb{P}_{incr,past}$ is larger than $\mathbb{P}_{incr,goal}$, the objective function and constraints is like:

$$\begin{aligned}
&\text{[Case II]} \\
&\text{Minimize} && Var\left(t_i\right),\ for\ i=1:1:\mathcal{T}, \\
&\text{subject to} && \left|\left(n_{i+1}-n_i\right)\right| \geq \left|\left(n_{i+2}-n_{i+1}\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-2\right), \\
&&& \left|\left(n_{i+1}-n_i\right)\right| \geq \left(\mathbb{T}\times\left|\left(1+\mathbb{P}_{incr,goal}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
&&& \left|\left(n_{i+1}-n_i\right)\right| \leq \left(\mathbb{T}\times\left|\left(1+\mathbb{P}_{incr,past}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
&&& n_1=t_{\kappa}^{HO}=t_{start}^{ROLL,F},\ n_{\mathcal{T}} \doteqdot t_{\kappa}^{HO}-2\times t_{\mathcal{J}}^{AMP}=t_{stop}^{ROLL,F}, \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \geq \left|\left(t_{i+2}-t_{i+1}\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-2\right), \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \geq \left(\mathbb{T}\times\left|\mathbb{P}_{incr,goal}\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \leq \left(\mathbb{T}\times\left|\mathbb{P}_{incr,past}\right|\right),\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
&&& \left|\left(n_{i+1}-n_i\right)\right| = \mathbb{T}+\left|\left(t_{i+1}-t_i\right)\right|,\ for\ i=1:1:\left(\mathcal{T}-1\right), \\
&&& t_1=0,\ t_{\mathcal{T}}=\mathbb{T}.
\end{aligned} \tag{6.2}$$

means that it will use a lower efficiency trend to store an additional frame.

About the objective function and constraints in $P_{decr,past}^{AMP}$ and $P_{decr,goal}^{AMP}$ cases are below, and we have to restate that the time is count forward when using $\mathbb{P}_{decr}$, and $\mathbb{P}_{decr}$ has already defined a negative value:

$$\begin{aligned}
&\text{[Case III]} \\
&\text{Minimize} && Var\left(t_i\right),\ for\ i=1:1:\left(\mathcal{T}+2\right), \\
&\text{subject to} && \left|\left(n_{i+1}-n_i\right)\right| \geq \left|\left(n_{i+2}-n_{i+1}\right)\right|,\ for\ i=1:1:\mathcal{T}, \\
&&& \left|\left(n_{i+1}-n_i\right)\right| \geq \left(\mathbb{T}\times\left|\left(1+P_{decr,goal}^{AMP}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}+1\right), \\
&&& \left|\left(n_{i+1}-n_i\right)\right| \leq \left(\mathbb{T}\times\left|\left(1+P_{decr,past}^{AMP}\right)\right|\right),\ for\ i=1:1:\left(\mathcal{T}+1\right), \\
&&& n_1=t_{\kappa}^{HO}=t_{start}^{ROLL,B},\ n_{\mathcal{T}} \doteqdot t_{\kappa}^{HO}+2\times t_{\mathcal{J}}^{AMP}=t_{stop}^{ROLL,B}, \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \geq \left|\left(t_{i+2}-t_{i+1}\right)\right|,\ for\ i=1:1:\mathcal{T}, \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \geq \left(\mathbb{T}\times\left|1+P_{decr,goal}^{AMP}\right|\right),\ for\ i=1:1:\left(\mathcal{T}+1\right), \\
&&& \left|\left(t_{i+1}-t_i\right)\right| \leq \left(\mathbb{T}\times\left|1+P_{decr,past}^{AMP}\right|\right),\ for\ i=1:1:\left(\mathcal{T}+1\right), \\
&&& \left|\left(n_{i+1}-n_i\right)\right| = \mathbb{T}-\left|\left(t_{i+1}-t_i\right)\right|,\ for\ i=1:1:\left(\mathcal{T}+1\right), \\
&&& t_1=0,\ t_{\mathcal{T}}=-\mathbb{T}.
\end{aligned} \tag{6.3}$$

means that it will use a higher efficiency trend to release stored frames by using Inv-AMP mode, and

$$
\begin{array}{ll}
\text{[Case IV]} & \\
\text{Minimize} & Var\left(t_i\right), \ for\ i = 1:1:(\mathcal{T}+2), \\
\text{subject to} & \left|(n_{i+1}-n_i)\right| \leq \left|(n_{i+2}-n_{i+1})\right|, \ for\ i = 1:1:\mathcal{T}, \\
 & \left|(n_{i+1}-n_i)\right| \leq \left(\mathbb{T} \times \left|\left(1+P_{decr,goal}^{AMP}\right)\right|\right), \ for\ i = 1:1:(\mathcal{T}+1), \\
 & \left|(n_{i+1}-n_i)\right| \geq \left(\mathbb{T} \times \left|\left(1-P_{decr,past}^{AMP}\right)\right|\right), \ for\ i = 1:1:(\mathcal{T}+1), \\
 & n_1 = t_{\kappa}^{HO} = t_{start}^{ROLL,B}, \ n_{\mathcal{T}} = t_{\kappa}^{HO} + 2 \times t_{\mathcal{J}}^{AMP} = t_{stop}^{ROLL,B}, \\
 & \left|(t_{i+1}-t_i)\right| \leq \left|(t_{i+2}-t_{i+1})\right|, \ for\ i = 1:1:\mathcal{T}, \\
 & \left|(t_{i+1}-t_i)\right| \leq \left(\mathbb{T} \times \left|1+P_{decr,goal}^{AMP}\right|\right), \ for\ i = 1:1:(\mathcal{T}+1), \\
 & \left|(t_{i+1}-t_i)\right| \geq \left(\mathbb{T} \times \left|1+P_{decr,past}^{AMP}\right|\right), \ for\ i = 1:1:(\mathcal{T}+1), \\
 & \left|(n_{i+1}-n_i)\right| = \mathbb{T} - \left|(t_{i+1}-t_i)\right|, \ for\ i = 1:1:(\mathcal{T}+1), \\
 & t_1 = 0, \ t_{\mathcal{T}} = -\mathbb{T}.
\end{array}
\tag{6.4}
$$

means that it uses a lower efficiency trend to release stored by using Inv-AMP mode.

In fact, there are several additional mixed formulations and their constraints based on above, and one of them is useful in our scenario and already mentioned in Section 4.2. The [Case V] uses in condition that, if $\mathbb{P}_{incr,past}$ is similar to $\mathbb{P}_{incr,goal}$ but we want to store one or more frames when the roll-off duration is indicated and still achieving local VDoP minimization goal. We believe that the outcome could be better and simple to solve than the combination of [Case I] and [Case II]. Below is [Case V], including objective function and its constraints:

$$
\begin{array}{ll}
\text{[Case V]} & TD:\ time\ duration \quad RAD:\ required\ additional\ delay \\
\text{Minimize} & Var\left(t_i\right),\ for\ i=1:1:\mathcal{T},\ \ (\mathcal{T}\%2=1) \\
\text{subject to} & \left|(n_{i+1}-n_i)\right| \geq \left|(n_{i+2}-n_{i+1})\right|,\ for\ i=1:1:\left(\left\lceil\frac{\mathcal{T}}{2}\right\rceil-2\right), \\
& \left|(n_{i+1}-n_i)\right| \leq \left|(n_{i+2}-n_{i+1})\right|,\ for\ i=\left(\left\lceil\frac{\mathcal{T}}{2}\right\rceil\right):1:(\mathcal{T}-2), \\
& \left|(n_{i+1}-n_i)\right| \leq \left(\mathbb{T}\times\left|(1+\mathbb{P}_{incr})\right|\right),\ for\ i=1:1:(\mathcal{T}-1), \\
& \left|(n_{i+1}-n_i)\right| \geq \left(\mathbb{T}\times 1\right),\ for\ i=1:1:(\mathcal{T}-1), \\
& n_1=t_{\kappa}^{HO},\ n_{\mathcal{T}} \fallingdotseq t_{\kappa}^{HO}-TD, \\
& \left|(t_{i+1}-t_i)\right| \geq \left|(t_{i+2}-t_{i+1})\right|,\ for\ i=1:1:\left(\left\lceil\frac{\mathcal{T}}{2}\right\rceil-2\right), \\
& \left|(t_{i+1}-t_i)\right| \leq \left|(t_{i+2}-t_{i+1})\right|,\ for\ i=\left(\left\lceil\frac{\mathcal{T}}{2}\right\rceil\right):1:(\mathcal{T}-2), \\
& \left|(t_{i+1}-t_i)\right| \leq \left(\mathbb{T}\times\left|\mathbb{P}_{incr}\right|\right),\ for\ i=1:1:(\mathcal{T}-1), \\
& \left|(t_{i+1}-t_i)\right| \geq 0,\ for\ i=1:1:(\mathcal{T}-1), \\
& \left|(n_{i+1}-n_i)\right| = \mathbb{T}+\left|(t_{i+1}-t_i)\right|,\ for\ i=1:1:(\mathcal{T}-1), \\
& t_1=0,\ t_{\mathcal{T}}=RAD.
\end{array}
\tag{6.5}
$$

About the even case of $\mathcal{T}$, it is similar to the odd case above, so we do not enumerate it again.

About above objective functions and their constraints, the relation between f_i and t_i is that, t_i is equal to $\sum_{\eta=1}^{i} f_\eta$ got from Equation 2.6 and $n_i = \mathbb{T}\times i+t_i$. Additionally, the range of former is $\left[\left|P_{past}^{AMP}\right|\mathbb{T},\ \left|P_{goal}^{AMP}\right|\mathbb{T}\right]$, and the range of latter is $[0,\ \mathbb{T}]$. We do not consider the original delay between received and playout before the roll-off execution, so we set the initial accumulation additional delay t_1 is 0 in all cases. We also know that the purpose of [Case I] and [Case II] are to store an additional frame, so $t_{\mathcal{T}} = \mathbb{T}$, and the purpose of [Case III] and [Case IV] are to release and additional frame, so $t_{\mathcal{T}} = -\mathbb{T}$ to represent the phenomenon.

About the meaning of above functions and their constraints, we just focus on Equation 6.1 to be an example. The purpose of the function is to minimize local VDoP when AMP finds out that the gap between $\mathbb{P}_{incr,past}$ and $\mathbb{P}_{incr,goal}$ is quite large and prepare to use roll-off to reduce VDoP. The first constraint means that the playout interval becomes equal or longer than last playout interval, to increase frame storing efficiency. The second and third constraint means that the limitation of the playout interval, and the fourth constraint means the known values of the playout time. The fifth constraint means that the accumulation additional delay interval becomes equal or longer than last interval, and they also have their limitations shows in the sixth and seventh constraints. The eighth constraint means the relation between playout interval and accumulation additional delay interval, and the last constraint means the known values of the accumulation additional delay.

Besides, we could extend the optimization framework to some general cases by modifying $\mathcal{T}$, $t_{\mathcal{T}}$, $n_{\mathcal{T}}$, or other variables to become more elastic.

In summary, we could use a series steps to get the optimization playout frame schedule and thus we could get the profile of roll-off function:

1. Calculate and check $\mathbb{P}_{incr,goal}$ from Section 4.2 if some event happens. We also have to choose a right roll-off case to use.

2. Get roll-off execution duration, the least number of frames would receive in roll-off duration, the least additional delay requirement, $\mathbb{P}_{incr,past}$, and $\mathbb{P}_{incr,goal}$ to be the input of the optimization function.

3. Use optimization roll-off function to get the playout frame schedule.

It is a simple work to get the roll-off function after we know the relation between playout time (n_i) and accumulation additional delay (t_i). We just transform the t-n plot back to $\mathbb{P}$-n ($\mathbb{P}$ to playout time) plot and get the roll-off function like Figure 38 below, which number of playout frames are 10, number of received frames are 11, $\mathbb{P}_{incr,goal} = P^{AMP}_{incr,MAX}$, $\mathbb{P}_{incr,past} = 0\%$, [Case I], and it could promise of achieving local VDoP minimization, $F^{\mathbb{N}}_{\mathtt{j+9}} = F^{\mathbb{N}}_{\mathtt{j}} - 1$, etc.:

6.3 Conclusion

Conventionally, AMP is discussed and used in general network cases which have some scattered packet delay or burst packet received events. To prevent having no frame to playout and buffer underflow problems, or having too many packets cause buffer overflow, they use AMP to change frame deadline. But in our opinion, the solution way is to take a musket to kill a butterfly, because there are many other methods to achieve the same goal. However, there is no better way to solve burst frame delay during heterogeneous handover by using AMP.

In this thesis, we use a series of AMP mechanisms to prevent video interruption from heterogeneous handover duration by delaying the interruption happen event noticed by users based on buffer underflow. By predicting the handover happened time, knowing the range of handover duration, doing the frame storing scheduler, and cooperating with media server, AMP could guarantee that video frames will be

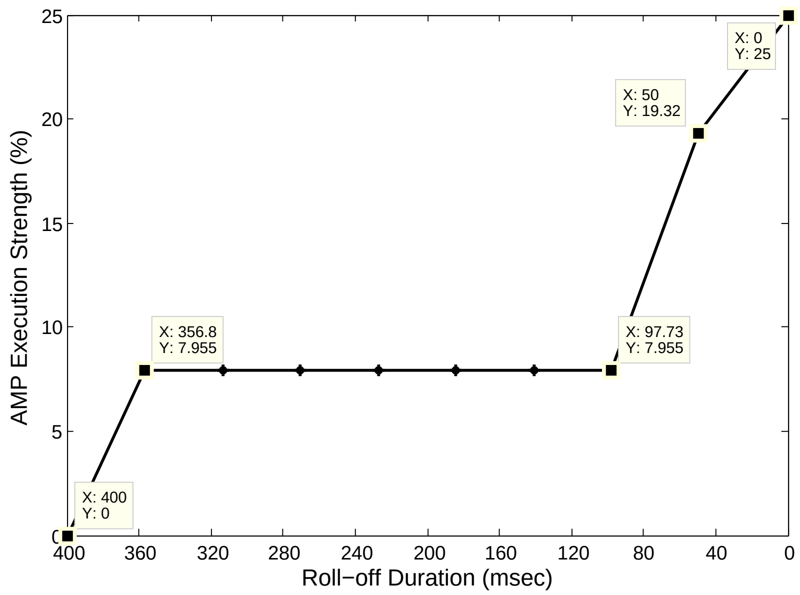


Figure 38: An example of $\mathbb{P}$-n plot

received again before the buffer underflow, to let users be unnoticed about video interruption, no matter what some improvements embedded in lower layers than Application Layer, and the MN route it drives. By experiments, we find out that the proposed AMP has a good plan to store additional frames preparing to handover, and defensing low storing efficiency due to shadowing network at the same time. By comparing with conventional AMP, our proposed AMP has lower percentage of frame and packet loss due to some reasons, higher PSNR than conventional AMP.

6.4 Future Work

In this thesis, we only use AMP to be our main method to achieve the goal, and there are some useful tools to improve video quality in some constraints like buffer limitations in media server and in playout buffer, end-to-end delay requirement, etc. Using adaptive coding rate scheme will increase the number of storing frames and reduce the impact on packet loss due to shadowing network and keep PSNR being a good value. Using frame priority transmission scheme could let the receiver gets important frames first (like I-frame and first several P-frames in GOP) if needed. Other tool like adaptive changing number of FEC/ARQ [64], VTSS [65], frame rate control [66, 67] jointed with AMP also could improve video quality. Furthermore, to get a higher precise of distance measurement degree, vehicle movement prediction is a good solution, and the$\mathbb{P}-t$ curve will also be smoother. The work can be implement in complex networks of NS-2 like VANET, or even in testbed devices to close the real world condition. Actually, for VANET topology construction, we could loading the maps and WLAN/WMAN/WWAN BS/AP positions based on some providers like [68–70] to closer to the truth. The subject is full of interest to us to bring more appealing and comfortable.

REFERENCES

[1] P. A. Chou and M. van der Schaar, *Multimedia over IP and wireless networks.* Academic Press, Mar. 2007.

[2] IPTV, "http://www.itu.int/itu-t/iptv/."

[3] A.-. MAVERICK, "http://www.pakdef.info/pakmilitary/airforce/asm/agm65.html."

[4] YouTube, "http://www.youtube.com/?gl=tw&hl=zh-tw."

[5] I'm Vlog, "http://www.im.tv/vlog/."

[6] PPStream, "http://www.ppstream.com/."

[7] hiChannel, "http://hichannel.hinet.net/."

[8] Google Voice and Video Chat, "http://google-voice-and-video.en.softonic.com/."

[9] 802.16e, "http://www.ieee802.org/16/tge/."

[10] 802.11p, "Status of project ieee 802.11 task group p wireless access in vehicular environments (wave)." http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm.

[11] GSM, "http://www.gsmworld.com/."

[12] HSDPA, "http://www.umtsworld.com/technology/hsdpa.htm."

[13] T. E. Fernando Pereira, *The MPEG-4 book.* Upper Saddle River, NJ : Prentice Hall, 2002.

[14] Y. Yuan, H. Chen, and M. Jia, "An optimized ad-hoc on-demand multipath distance vector(aomdv) routing protocol," in *Communications, 2005 Asia-Pacific Conference on*, pp. 569–573, 2005.

[15] 802.11e, "Ieee-sa standards board standards review committee (revcom)recommendations." http://standards.ieee.org/board/rev/905recomm.html.

[16] IETF, "Real time streaming protocol."

[17] N. Laoutaris and I. Stavrakakis, “Adaptive playout strategies for packet video receivers with finite buffer capacity,” *IN PROC. IEEE ICC*, 2001.

[18] M. Yuang, P. L. Tien, and S. T. Liang, “Intelligent video smoother for multimedia communications,” feb 1997.

[19] M. Li and S. Cheng, “Arrival Process-Controlled adaptive media playout for video streaming,” in *Proceedings of the 2nd International Workshop on Future Multimedia Networking*, (Coimbra, Portugal), pp. 26–37, Springer-Verlag, 2009.

[20] M. Kalman, E. Steinbach, and B. Girod, “R-d optimized media streaming enhanced with adaptive media playout,” in *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, vol. 1, pp. 869–872 vol.1, 2002.

[21] *Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure.* Hacker Friendly LLC, 2nd ed., 2007.

[22] “IEEE standard for local and metropolitan area networks: Overview and architecture,” 2004.

[23] S. McQuerry, *'CCNA Self-Study: Interconnecting Cisco Network Devices (ICND), Second Edition'.* Cisco Press, Nov. 2003.

[24] IEEE 802.21, “http://www.ieee802.org/21/.”

[25] A. Pontes, D. dos Passos Silva, J. Jailton, O. Rodrigues, and K. Dias, “Handover management in integrated wlan and mobile wimax networks,” *Wireless Communications, IEEE DOI - 10.1109/MWC.2008.4653137*, vol. 15, no. 5, pp. 86–95, 2008.

[26] S. Tran-Trong, S. Tursunova, and Y.-T. Kim, “Enhanced vertical handover in mobile ipv6 with media independent handover services and advance duplicate address detection,” in *KNOM conference 2008*, 2008.

[27] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, “An overview of ieee 802.21: media-independent handover services,” *Wireless Communications, IEEE DOI - 10.1109/MWC.2008.4599227*, vol. 15, no. 4, pp. 96–103, 2008.

[28] K. Taniuchi, Y. Ohba, V. Fajardo, S. Das, M. Tauil, Y.-H. Cheng, A. Dutta, D. Baker, M. Yajnik, and D. Famolari, “Ieee 802.21: Media independent handover: Features, applicability, and realization,” *Communications Magazine, IEEE DOI - 10.1109/MCOM.2009.4752687*, vol. 47, no. 1, pp. 112–120, 2009.

[29] T. Jun, Z. ying jiang, Z. zhi, Y. zhi wei, and C. zhi lan, “Performance analysis of vertical handoff in wifi and wimax heterogeneous networks,” in *Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on DOI - 10.1109/CNMT.2009.5374499*, pp. 1–5, 2009.

[30] H. Silva, L. Figueiredo, C. Rabadao, and A. Pereira, “Wireless networks interoperability - wifi wimax handover,” in *Systems and Networks Communications, 2009. ICSNC ’09. Fourth International Conference on DOI - 10.1109/ICSNC.2009.99*, pp. 100–104, 2009.

[31] J. Pyun, “Context-Aware streaming video system for vertical handover over wireless overlay network,” *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 1, pp. 71–79, 2008.

[32] M. Tsagkaropoulos, I. Politis, T. Dagiuklas, and S. Kotsopoulos, “Enhanced vertical handover based on 802.21 framework for real-time video streaming,” in *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, (London, United Kingdom), pp. 1–6, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[33] D. Lee, J. Kim, and P. Sinha, “Handoff-aware adaptive media streaming in mobile ip networks,” Feb. 2009.

[34] P. Stathopoulos, P. Papageorgiou, D. Kouis, S. Zoi, and N. Mitrou, “A rate adaptation scheme for media streaming over heterogeneous networks,” *International Journal of Wireless and Mobile Computing*, vol. 2, no. 2/3, pp. 173 – 191, 2007.

[35] P. Bellavista, A. Corradi, and C. Giannelli, “Adaptive buffering-based on handoff prediction for wireless internet continuous services,”

[36] Y. J. Liang, S. Member, N. Färber, and B. Girod, “Adaptive playout scheduling and loss concealment for voice communication over ip networks,” 2002.

[37] M. Benaissa, V. Lecuire, and F. Lepage, "An algorithm for playout delay adjustment for interactive audio applications in mobile ad hoc networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, pp. 524–529, 2002.

[38] M. Bernaschi, F. Cacace, G. lannello, and M. Vellucci, "Mobility management for VoIP on heterogeneous networks: Evaluation of adaptive schemes," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 9, pp. 1035–1047, 2007.

[39] E. Steinbach, N. Farber, and B. Girod, "Adaptive playout for low latency video streaming," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1, pp. 962–965 vol.1, 2001.

[40] B. Girod, M. Kalman, N. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," 2002.

[41] H. H. Chen, Y.-H. Yang, and M.-T. Lu, "Smooth playout control for video streaming over error-prone channels," in *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pp. 415–418, 2006.

[42] D. Buhalis and L. Pistidda, "Wireless applications in destinations," in *Information and Communication Technologies in Tourism 2009*, pp. 161–171, 2009.

[43] T. S. Rappaport, *Wireless Communications: Principles and Practice.* Prentice Hall, 2 ed., 2002.

[44] Q. Wu and C. Williamson, "Dynamic channel rate assignment for multi-radio wlans,"

[45] Ns3WirelessPhy, "http://www.nsnam.org/wiki/index.php/ns3wirelessphy."

[46] J. R. Fraenkel and N. E. Wallen, "How to design and evaluate research in education with powerweb," 2005-04-11.

[47] Y. Zhang, W. Zhuang, and A. Saleh, "Vertical handoff between 802.11 and 802.16 wireless access networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–6, 2008.

[48] D. Wright, "Maintaining QoS during handover among multiple wireless access technologies," in *Management of Mobile Business, 2007. ICMB 2007. International Conference on the*, p. 10, 2007.

[49] T. Ahmed, K. Kyamakya, and M. Ludwig, "Architecture of a Context-Aware vertical handover decision model and its performance analysis for GPRS - WiFi handover," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, pp. 795–801, 2006.

[50] R. Saeed, H. Mohamad, B. Ali, and M. Abbas, "WiFi/WiMAX heterogeneous seamless handover," in *Broadband Communications, Information Technology & Biomedical Applications, 2008 Third International Conference on*, pp. 169–174, 2008.

[51] H. Lee, B. Moon, and A. Aghvami, "Enhanced SIP for reducing IMS delay under WiFi-to-UMTS handover scenario," in *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST '08. The Second International Conference on*, pp. 640–645, 2008.

[52] A. Ahmed, Leïla Merghem-Boulahia, and Dominique Gaïti, "Cooperative agent based vertical handover scheme for heterogeneous networks," in *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*, pp. 410–415, 2010.

[53] H. Lin, K. Chu, W. Lee, and J. Lin, "Seamless handover for multimedia applications between heterogeneous wireless networks," in *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on*, pp. 985–988, 2009.

[54] L. Suciu, M. Benzaid, S. Bonjour, and P. Louin, "Assessing the handover approaches for heterogeneous wireless networks," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pp. 1–6, 2009.

[55] X. Wu and G. Nie, "Design and simulation of an enhanced handover scheme in heterogeneous mobile IPv6 networks," in *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, vol. 2, pp. 448–451, 2009.

[56] X. Chen, H. Zhou, Y. Qin, and H. Zhang, "NEMO-Based multiple interfaces scheme between overlay heterogenous access networks," in *Mobile Ad-hoc and Sensor Networks, 2008. MSN 2008. The 4th International Conference on*, pp. 103–109, 2008.

[57] J. So, "Vertical handoff in integrated CDMA and WLAN systems," *AEU - International Journal of Electronics and Communications*, vol. 62, pp. 478–482, June 2008.

[58] S. McCanne and S. Floyd, "ns network simulator."

[59] M. Majanen, P. Perala, T. Casey, J. Nurmi, and N. Veselinovic, "Mobile wimax handover performance evaluation," in *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, pp. 263–269, 2009.

[60] K. ChihHeng, "Ns2 learning guide, http://hpds.ee.ncku.edu.tw/ smallko/ns2/ns2.htm."

[61] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," 2003.

[62] YUVviewer, "http://ftp3.itu.ch/av-arch/jvt-site/software_tools/."

[63] H. Zhang, J. Zhou, Z. Chen, and J. Li, "Minimizing delay for video conference with network coding," in *ACM Special Interest Group on Data Communication (SIGCOMM)*, 2009.

[64] A. Moid and A. O. Fapojuwo, "A cross-layer framework for efficient streaming of h.264 video over ieee 802.11 networks," *J. Comp. Sys., Netw., and Comm.*, vol. 2009, pp. 1–13, 2009.

[65] E. Masala, D. Quaglia, and J. De Martin, "Variable time scale multimedia streaming over ip networks," *Multimedia, IEEE Transactions on*, vol. 10, no. 8, pp. 1657–1670, 2008.

[66] J.-G. Kim, Y. Wang, and S.-F. Chang, "Content-adaptive utility-based video adaptation," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 3, pp. III–281–4 vol.3, 2003.

[67] R. Ramanujan, J. Newhouse, M. Kaddoura, A. Ahamad, E. Chartier, and K. Thurber, "Adaptive streaming of mpeg video over ip networks," in *Local Computer Networks, 1997. Proceedings., 22nd Annual Conference on*, pp. 398–409, 1997.

[68] VMAX AP position search, "http://www.vmax.net.tw/coverage/."

[69] WiFly AP position search, "http://www.wifly.com.tw/nr/exeres/4720ba9f-9d2d-4b4f-a105-f900d90aaeee.htm."

[70] Qon AP position search, "http://www.qon.com.tw/hotsearch/."