# 國立交通大學

## 電子工程學系

## 碩士論文

無線通訊系統中具內容感知的視訊串流控制方法

**Content-Aware Controls for Video Streaming in Wireless Communication System**


研究生：莊孝強

指導教授：蔣迪豪　博士

黃經堯　博士


中華民國九十三年七月


E
S
A
1896

# 無線通訊系統中具內容感知的視訊串流控制方法

# Content-Aware Controls for Video Streaming in Wireless Communication System


研究生： 莊孝強　　Student: Hsiao-Chiang Chuang

指導教授: 蔣迪豪　　Advisors: Tihao Chiang

黃經堯　　ChingYao Huang

國立交通大學

電子工程學系

碩士論文


A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

July 2004

HsinChu, Taiwan, Republic of China

中華民國九十三年七月


E
S
A
1896

# 無線通訊系統中具內容感知的視訊串流控制方法

研究生: 莊孝強　　指導教授: 蔣迪豪 博士
黃經堯 博士

國立交通大學
電子工程學系 電子研究所碩士班

## 摘要

網際網路上的視訊串流技術已發展多年，其中多項壓縮、網路、及接收端處理的技術，已經能夠讓使用者在視覺上得到良好的效果；然而，少有文獻從端對端(end-to-end)的角度，探討無線網路中作視訊串流議題。本論文的主要研究目的有二，一在於將通訊協定堆疊納入考量之下，觀察在無線環境中作視訊串流，各項控制機制對於系統的影響。其二，由於頻寬的變化會造成接收端緩衝缺空的問題，將嚴重損害使用者的接收品質。本論文提出具內容感知的控制方法，此控制方法在解決緩衝缺空的同時，兼顧了人類的視覺品質，讓整個視訊串流的過程中，使用者能夠得到平順的播放品質。這項技術是基於 MPEG-4 漸進精細可調層次(Fine-Granularity Scalability)，作為資料的壓縮方式。此外，我們利用 ISO/IEC 21000-12 的測試平台(Test Bed)來評估這個控制演算法的實際視覺效果。為了驗證這套控制方法可適用於無線通訊系統之中，我們開發了一個無線行動通訊模擬平台，模擬在一般性的行動通訊架構之下，視訊串流內部的交互影響流程。實驗的結果證明了所提出的控制方法可以在不影響系統容量的前提之下，提供視訊串流的使用者，一個平順的播放過程。

# Content-Aware Controls for Video Streaming in Wireless Communication System

Student: Hsiao-Chiang Chuang Advisor: Dr. Tihao Chiang
Dr. ChingYao Huang

Department of Electronic Engineering &
Institute of Electronics
National Chiao Tung University

## Abstract

Internet video streaming technologies have been developed for years, and several techniques, located in the source coder, network nodes, and receiver processing, can be used to achieve excellent visual quality for the clients. This thesis proposes a content-aware control for video streaming in a wireless cellular system such as cdma2000. The main purpose of this control scheme is to provide uniform (smooth) quality video streaming over a wireless channel. To provide a capability of rate adaptation, we adopt the MPEG-4 Fine-Granularity Scalability (FGS) scalable video coding technique as the flexible format of the source bitstream. Besides, we utilize the ISO/IEC 21000-12 Multimedia Test Bed to evaluate the practical performance of the proposed scheme. Furthermore, we develop the system-level simulation environment to estimate the impact of the proposed control algorithm on a wireless cellular system. The simulation result shows that appropriate control is necessary for streaming video in the wireless communication system. The proposed control technique provides a smooth playback for video streaming application, while maintaining the system capacity unchanged.

# 誌謝

碩士班兩年的時光匆匆，似乎又到了離別的時刻。在人生最精華的兩年歲月裡，感謝上天讓我有這個緣分遇見你們，陪我一起度過這段難忘的時光。

首先，我要感謝蔣迪豪老師以及王俊能學長，讓我有機會能夠參與 MPEG-21 Testbed 的發展計畫，因為參與這項計畫，學習到許多相關的知識及背景，為我日後論文的完成紮下深厚的根基。同時要感謝的，是在我催生論文之時，像一盞明燈一直引領著我的黃經堯老師。這篇論文許多關鍵性的問題，都是在和老師一次又一次的討論之中，摸索找尋而得到了答案。感謝兩位老師總是要在百忙之中抽空，給予我適當的方向及適時的指導，我才能順利完成這篇論文。

此外，我也要感謝家揚及耀中兩位學長，在和你們一起作計畫的過程中，向你們學到不少東西，除了專業方面的知識之外，更有機會見識到強者我學長的行事作風，讓我學習到如何能更有效率地處理事情，以及面臨難題時應有的態度。

另外要感謝的，就是在實驗室的日子裡，一起共同努力的伙伴們，耀諄、子良、名彥、俊安、振韋、子翰、岳賢、汝芩、瑛姿、思浩，還有相處了五年的室友智勛、志松。縱使一個人走在孤單的畢業之路上，因為有你們的陪伴，使我不再感到孤單，這份心中的感動，對我而言，一定能夠成為畢生難忘的回憶。

最後，我要感謝我的家人，爸爸、媽媽、妹妹，和女友慧騏。因為你們在背後的默默支持，這一路走來，總是可以感受到來自於你們的關心及祝福，讓我的心中充滿著被關懷的溫暖。

能夠順利完成碩士論文，除了憑藉著自己的努力與決心之外，更因為有你們給我的支持、鼓勵及陪伴，在論文趕稿的苦悶日子裡，我才能更加堅定不移地繼續走下去。這篇論文，僅獻給最特別的你們。

莊孝強　謹誌

2004 年 7 月, Commlab, 交大, 新竹, 台灣

# Contents

# List of Figures

# List of Tables



# Chapter 1

# Introduction

## 1.1 Overview of Multimedia Streaming over Wireless Network

The 3G wireless communication technology has been developed for years. The advantages of 3G wireless technologies over the former technologies can be described from either from the system or the end-user points of view. With better wireless techniques, including power control, soft handoff, transmit diversity, and turbo coding, we can further improve voice capacity, data throughput, and battery life. Among these advanced features, the capability of high data throughput enables several applications, such as web browsing, network games, background download of e-mails, files, and multimedia streaming. Figure 1-1 shows the evolution roadmap of the data throughput provided by a distinct wireless service network.

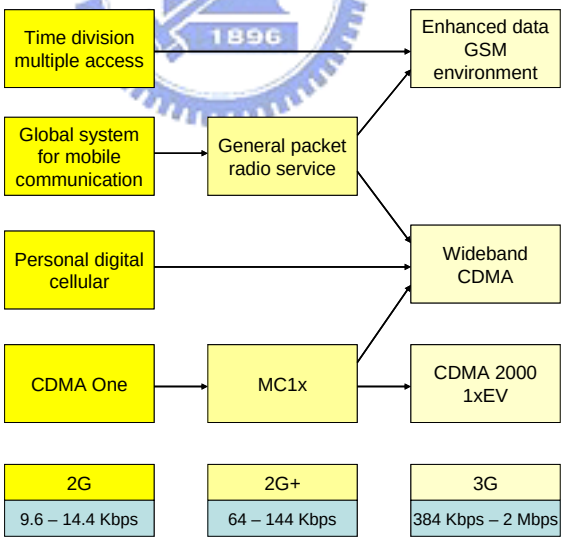



Figure 1-1. The evolution of data throughput from 2G to 3G wireless network [1]

However, these applications have diverse requirements with different parameters for wireless transmission. For example, web browsing requires the completeness of data while accessing the desired web page. Thus, the reliability of transmission is the key to this kind

of application. Take multimedia streaming for an example, most of the streaming service can tolerate the error in data integrity to some degree; instead, the delay issue is critically essential here because the multimedia data should be played back in a timely fashion. The 3G wireless technology such as [2] has defined different QoS classes with corresponding attributes to characterize the quality of service. In the QoS provisioning system, dissimilar types of data application could be of different importance, and hence more important applications could obtain higher priority to transmit with better quality, in terms of corresponding quality metrics.

Among all applications of multimedia transmission, one of most challenging services is the multimedia streaming (e.g. video/audio streaming or A/V simultaneous streaming). In the conventional multimedia streaming over the Internet, several issues such as bandwidth, delay, and loss are the key factors that influence the end-to-end performance. Besides, to transport multimedia data in a heterogeneous environment is also a challenging problem [3]. All-IP network architecture adopted by the future wireless data communication simplifies the heterogeneity problems. The existing protocols in OSI layer 4~7 could be efficiently re-used based on the IPv4 or IPv6 [4] network. Furthermore, some multimedia standards recommend that the specified applications should be transmitted at the IP-based network. For example, the real-time transmission of MPEG-4 streams should exploit the Real-time Transport Protocol (RTP) as the session-layer protocol, which is based on the User Datagram Protocol (UDP) at the transport layer and Internet Protocol (IP) at the network layer [5]. However, the heterogeneity problems still make the multimedia transmission difficult in term of terminal capabilities.

To provide good experience for each user, several control schemes, which could be categorized into congestion control and error control. The congestion control is used to reduce the packet loss and delay caused by network congestion. Usually, the connection path from the server to the client could be equivalently seemed as a virtual buffer. If the transmission traffic exceeds the available bandwidth, the residual parts of the traffic will be stored in the virtual buffer and wait to be sent to the destination in the following time slots. As the channel bandwidth varies with time, the resulting wait time would be changed respectively, and this causes the so-called delay/jitter problem. Once the amount of accumulated data is greater than the size of virtual buffer, the surplus data would be discarded, and this forms the major reason of the packet loss in the transmission over

Internet.

The packet loss is inevitable during the transmission of multimedia data, and the loss incurs different devastating effects such as quality degradation, decoding corruption, and out of synchronization. The purpose of error control is to mitigate the impairment caused by the packet loss. The error control could be done in two aspects: transport and compression layers [3]. The techniques such as retransmission and forward error correction (FEC) could be classified into the transport-based error control; while some post-processing schemes such as error resilient and error concealment belongs to the compression-domain error control.

As compared to the streaming over Internet, the multimedia streaming over wireless network suffers more limitations. One of the limitations primarily comes from the diverse nature between wire-line and wireless connection. Figure 1-2 illustrates a general scenario for multimedia streaming over wireless network. The associated protocol stacks are also shown in the lower part of the figure.

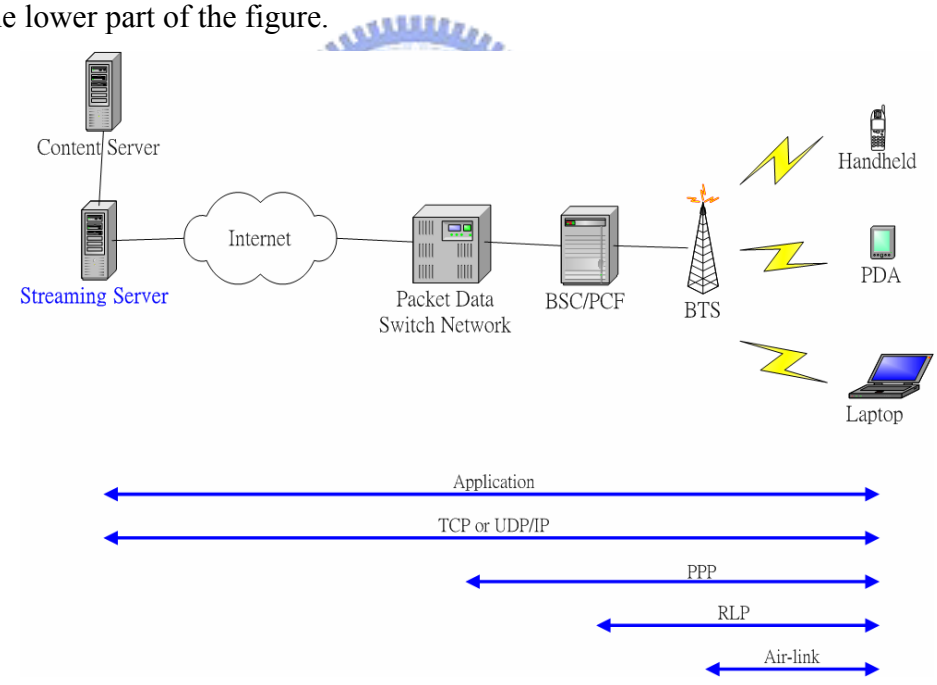



Figure 1-2. The general scenario for wireless multimedia streaming

We can see that multimedia streaming service requires not only the wireless connection between the mobile station (MS) and the base transceiver station (BTS) should be maintained, but the access to the Internet has also to be supported as well. The diversity of the Internet would much complicate the discussion of the multimedia streaming. In this

thesis, we focus on the last-mile transmission by assuming that the streaming server is located in front of the Packet Data Switch Network (PDSN). Without loss of generality, the terminology we use in this thesis is based on the cdma2000 1xEV-DO wireless network. Similar architecture appears in other wireless networks that support data transmission, such as GPRS and WCDMA. For example, the correspondence exists between the PSDN in cdma2000 1x system and the Serving / Gateway GPRS Support Node (SGSN/GGSN) in the WCDMA system, while both are interfaces to the packet data network.

The obstacles of wireless multimedia streaming are similar to the issue of streaming over Internet. In addition, the rapid variation of the wireless link quality, which is caused by fading, interference, and multi-path effect, makes the transmission environment more challenging. The bandwidth[1] changes more rapidly, the round-trip delay becomes larger, as well as the probability of random packet loss increases. In the same way, the aforementioned techniques of error control and congestion control could be employed to combat against these detrimental effects. Nevertheless, the radio resource is a limiting factor for these control schemes in a wireless cellular system. A determination of a control algorithm may not be appropriate for the use of wireless streaming due to this additional restriction. Hence, the introduction of buffers is important to transmit data over a wireless network. There are often buffers located in the (Base Station Controller) BSC/ (Packet Control Function) PCF and the MS, respectively, to absorb the variation of rate either at the transmitter or receiver side. One of the most important issues this thesis address is the impact of buffers on the end-to-end performance for streaming over a wireless network as described in Chapter 3.

Another major difficulty for multimedia streaming is the maintenance of inter-stream (e.g. video, audio, subtitles…) synchronization [6]. A typical solution to this problem is to synchronize the time stamps of the associated streams using a master clock, such as Simple Network Time Protocol (SNTP) [7]. However, the clock drift between the server and client still cause problems in synchronization during a long duration transmission. This thesis studies the issue of video streaming only because the synchronization is within the scope of another research field. In the following sections, we focus the discussion on the video streaming and its related topics.

---

[1] The term “bandwidth” is originally referred to the occupied spectrum. In this thesis, this term indicates the data throughput that data could be transmitted through the channel.

## 1.2 Motivation

For video streaming over wireless channel, there are limited literatures that address both the issues of source transmission and channel condition. Most of multimedia-related papers focus on the efficiency of the proposed technique on various types of hypothetical channels. On the other hand, majority of papers related to wireless communication discuss the end-to-end performance of different streaming architectures by assuming the distinct traffic model of the video streaming. This work addresses issues of both realistic source transmission and wireless channel. As to the source transmission, we adopt the MPEG-21 Testbed [8] as a realistic streaming server that packetizes each access unit of video bitstream into an IP packet. As to the transmission channel, we develop a cdma2000 1x-RTT based platform for simulating the air-link transmission of video content from the BTS to the MS. The realistic simulation of video streaming over wireless channel gives more valuable indication for a practical system design.

Due to the relatively rapid variation of physical channel for wireless communication, the effective transmission characteristics, such as available channel rate and error rate, also change quickly. The rapid variation of channel condition causes serious damage to the buffer in the receiving side, i.e., the outage (underflow and overflow) probability of buffer arises accordingly. Stockhammer et al. [9] explore the required pre-roll time and buffer size for video streaming via variable bit-rate wireless channels. They propose a buffer design by pre-determining the buffer size and pre-roll time for a specific bounded receiving process. However, the pre-determination of pre-roll time is inadequate for some streaming scenario, such as live-content streaming. Several channel-adaptive streaming techniques are proposed to solve this problem, such as Adaptive Media Playout (AMP), rate-distortion optimized packet scheduling, and channel-adaptive packet dependency control [10]. Among these techniques, the AMP could moderate the receiving buffer status without the involvement of the server, and this is a desirable feature since the manufacturer of a mobile station and the operator of cellular system are usually separated. Hence, the AMP-based buffer control is an attractive solution for the buffer outage during streaming in the modern cellular system.

However, an AMP-based buffer control causes noticeable artifact to the human perception, which is an undesirable side-effect of the buffer control. To this end, this wok aims to eliminate the quality degradation, which keeping the performance of buffer control.

## 1.3 Overview of Video Coding Techniques

For video streaming, there are principally two category of video coding technology that facilitates the transport of video content over IP-based networks. The first type is the non-scalable video coding scheme, which provides better coding efficiency but suffers more error-resilient problem in video streaming. Typical multimedia compression standards such as H.261, H.262 (also refers to MPEG-2), H.263 series (H.263 baseline, H.263+, H.263++), belongs to this category. Techniques such as pre-roll buffering and delay-constrained retransmission could be utilized to improve the error resiliency of the non-scalable coding scheme. Quaglia et al. [11] present an adaptive packet classification method to ensure the most important part of the bitstream could be transmitted to the client side, which also addresses the data integrity issue of the single layer coding scheme. To improve the rate adaptation capability of the non-scalable video, S-frame [12] and SP-frame [13] solution have been proposed to dynamically resolve the error-drift problem caused by the loss in the non-scalable bitstream. However, both these solutions suffer loss in coding efficiency to improve the bitstream robustness.

Real-time video encoding retains both the coding efficiency and the error resilience of the non-scalable coded bitstream for delivery video over network, provided that the feedback information for rate control is accurate enough. The source coder could dynamically adjust the coding rate by methods to increase/decrease the quantization step, retain/discard high-frequency Discrete Cosine Transform (DCT) coefficients, and increase/reduce frame rate. Nevertheless, the real-time encoding with adaptive rate control would increase the load of the streaming servers. Usually, each subscriber requires individual encoding thread to accommodate the corresponding network condition. Consequently, the solution of real-time encoding suffers complexity and cost problems, especially when the number of clients is large.

The second type of video coding scheme is the scalable video coding. There are usually three dimensions of the scalability, namely, SNR, temporal, and spatial. The layered scalable techniques given by the MPEG-2 coding standard provides stair-case rate adaptation capability [14]. Take SNR scalable coding as an example, higher visual quality could be obtained by receiving more layers of bitstream via the connection channel. However, the finite level of scalability limits the use of scalability in video streaming

application. Hence, the MPEG-4 coding standard addressed this issue and proposed the Fine-Granularity Scalability (FGS) [15] to provide bit-level flexibility for the rate adaptation. Figure 1-3 illustrates the concept of the layered and the fine-grained scalability.

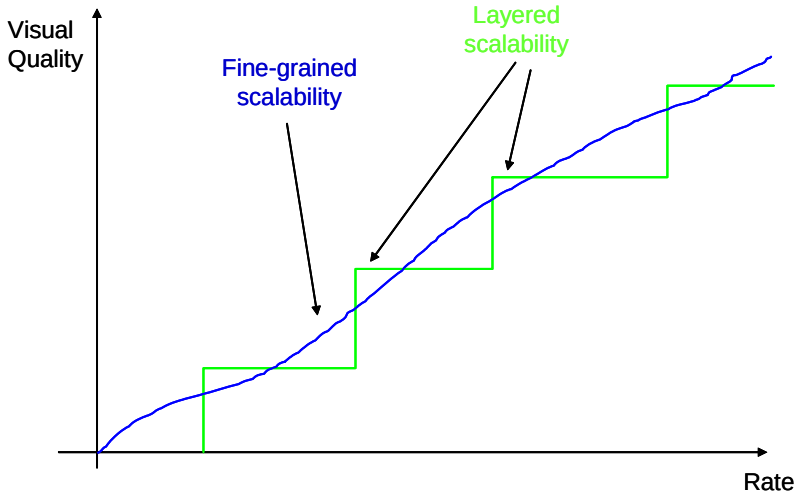



Figure 1-3. Illustration of the fine-grained and the layered scalability

As we can see, the fine-grained scalability could provide better bandwidth utilization than the layered scalability. The layered scalability may sometimes outperform the fine-grained scalability due to the non-scalable nature of the layered bitstream, and hence the coding efficiency of the associated layer would be higher. Detail comparison between these two kinds of scalability is discussed in [16].

Another kind of scalable video coding is the multiple descriptions coding scheme (MDC) [17], which was designed for path-diversity network such as IP-based network. There is no priority among descriptions and the reconstructed quality is based on the number of received descriptions. Therefore, in the packet network with path diversity, this coding scheme can be efficient in delivery of coded stream. As compared to the traditional layered scalable video coding scheme, the MDC coding scheme outperforms the layered coding scheme when packets are transmitted with unequal importance. Conversely, the layered coding has more benefits than the MDC coding when the rate-distortion (R-D) optimized packet scheduling are employed [18]. Some literatures have addressed the issue of transmission of MDC coded stream via wireless networks, and this thesis would focus on the scalable coding with the fine-grained scalable coding scheme.

All these scalable coding techniques could be exploited to provide better error resiliency and bandwidth utilization. The major drawback of the scalable video coding is the

coding efficiency problem. In most cases, the MPEG-4 FGS encounters a 2~3 dB loss in PSNR (Peak SNR), compared to the non-scalable MPEG-4 simple-profile coding scheme, at the same bit rate [19]. The coding efficiency problems of the scalable coding scheme have been investigated for years, and recently, the MPEG committee is preparing to collect proposals [20] on the state-of-the-art scalable coding techniques into the advanced scalable video coding standard (MPEG-21 part 13: scalable video coding).

The generic structure of using the scalable video coding technique in video streaming technique is depicted in Figure 1-4. A transcoder performs rate control to calculate the target transmission rate, following a rate shaper to truncate the enhancement-layer bitstream into a proper size for transmission. The coded-scalable bitstream under consideration could be generated by any scalable video coding algorithm that supplies bit-level scalability. Without loss of generality, this thesis focuses on the scalable video streaming using the MPEG-4 FGS coding scheme over wireless network. Similar results could be derived from the conclusion of this thesis in case that the advanced scalable video coding owns the same features with MPEG-4 FGS.

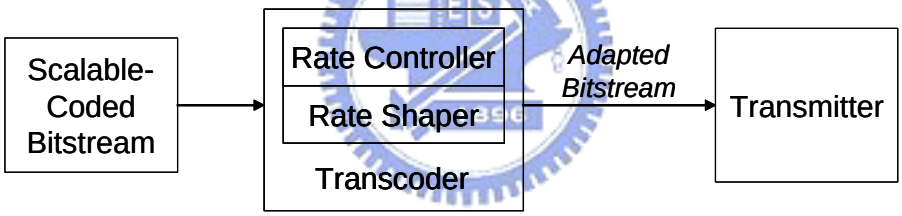



Figure 1-4. Generic structure for transmitting scalable-coded bitstream

To provide the capability of rate adaptation, in this thesis, we adopt MPEG-4 FGS as the scalable video coding scheme of interest, since the transcoding of FGS bitstream is relatively simple than other kinds of rate adaptation schemes. The scalable video coding is mainly used to adapt to the variation of effective channel bandwidth, i.e., the overall effect of channel throughput, including the channel bandwidth provided by the Internet and the cellular system.

## 1.4 Overview of Adaptive Media Playout (AMP) Algorithm

Adaptive Media Playout (AMP) is a receiver-based buffer control technique that adjusts the frame rate of the playback process, such that the probability for the event of buffer outage is minimized. Take buffer underflow for an example, the frozen picture

commonly seen causes serious damage to the perceptual quality, and the AMP scheme could prevent this situation by reasonably modulating the frame rate. Informal subjective tests have shown the reduction of playback rate up to 25% is unnoticeable [21]. Typically, the AMP-based buffer control could be divided into two steps. The first step is to determine the threshold for control activation. This step chooses a suitable threshold of buffer fullness to prevent potential buffer outage. The second step is to compute the playout rate based on the relationship between current buffer fullness and the predetermined threshold.

Yuang et al. [22] proposed a video smoother, which is based on a paradigm for threshold selection, and apply the selected threshold into an exponentially distributed service time to determine the next time for video frame play-out. With sets of fixed arrival-to-departure ratio, the effects of threshold selection on buffer outage are outlined. This paradigm of determining threshold gives some recommended threshold values for specified buffer outage parameters under the assumption of Poisson arrivals. The video smoother provides efficient buffer control, but it does not take the visual quality into account.

Laoutaris and Stavrakakis [23] address the issue of visual quality while supporting adaptive video playout. The receiving buffer is formulated as a M/G/1 queue and the buffer occupancy becomes a Markov chain. They merge the impact of buffer outage and buffer control into the proposed metric, Variance of Distortion of Playout (VDoP), to dynamically adjust the playout rate. Specifically, the control of playout rate is based on the threshold value determined by the dynamic threshold adjustment. An online detection algorithm for estimating optimal long-term value of threshold is proposed to provide a basis for the dynamic threshold adjustment. The value of threshold would be dynamically evaluated when the statistical behavior of arrival process is changed.

Kalman et al. gives an analytical result for various streaming environment such as archived streaming and live-content streaming [24]. The effect of packet retransmission, which is caused by the error-prone channel, is also included in the system model under consideration. Besides, the diverse delay-underflow tradeoffs for these streaming scenarios are also studied. This paper provides a sophisticated analysis on the channel model and the queuing model for client to clarify the relationship among pre-roll time, playout control, and buffer underflow. Experimental results show that significant performance improvements can be obtained in comparison to non-adaptive media playout.

Liang and Huang [25] proposed a content-based adaptive media player based on Perceived Motion Energy (PME), which implies the motion activity of a video sequence. They also proposed an AMP-based control in accordance with a distortion function, which combines both the distortion caused by control and the distortion incurred by the buffer outage. This control could provide better perceptual quality since the PME takes the motion activity into account.

This thesis proposed a model-based AMP control mechanism which addresses issues of both threshold and playout rate adjustments, which are based on the statistical assumption of both arrival and departure processes. The threshold and playout rates are computed to consider both the elimination of buffer outage and the smoothness of visual quality. The detail of the control algorithm will be stated in Section 3.3.

## 1.5 Thesis Organization

The organization of this thesis is described as follows:

Chapter 2 introduces the emulation platform of the MPEG-21 Resource Delivery Testbed, which is the basis of the experiment environment of this thesis. Some features of the Testbed are also qualified in this chapter.

Chapter 3 addresses the issue about wireless video streaming and states the architecture of the core control algorithm. The detail explanation of the proposed control scheme is included. A migration to the content-aware streaming will also be described in this chapter.

Chapter 4 shows some experimental result based on the proposed control algorithm. Both the end-to-end and the system-level performances are presented to justify the efficiency of the proposed control scheme.

Chapter 5 gives the conclusion and possible future works based on the effort of this thesis.

# Chapter 2

# Overview of MPEG-21 Resource Delivery Testbed and cdma2000 1x-RTT System

This chapter introduces the architecture of the MPEG-21 resource delivery testbed, which is the emulation platform for the experiment shown in this thesis. Some characteristics such as buffer configuration, rate control, traffic shaping, and packetization, will also be discussed. Some standard features such as Real Time Streaming Protocol (RTSP) / Session Description Protocol (SDP) control protocol would be skimmed only, as these standard features would affect the end-to-end performance lightly. Besides, we will describe the some streaming-related features of cdma2000 1x-RTT (Radio Transmission Technology) mobile system in this chapter, including the frame error rate, physical channel (FCH and SCHs) assignments, RLP retransmission, and etc. The physical-layer parameters are synchronized with the usage of [26], and some assumptions are also set in the same fashion. The mixed traffic system environment is also built to perform the system-level simulation.

## 2.1 Overview of MPEG-21 Resource Delivery Testbed

This section gives an overview to the MPEG-21 resource delivery testbed (abbreviated Testbed in the following text). The overall architecture is firstly introduced, while each component is explained in a functional fashion. This section would not touch the detail implementation because the MPEG-21 resource delivery testbed is just one example of the existing streaming system. Common features such as packetization, rate control, and rate shaping, of the testbed will be described.

### 2.1.1 Overall Architecture

The overall architecture of the testbed is shown in Figure 2-1. There are four major components in this streaming system: server, client, network interface, and network emulator. The server plays the role of a normal web server that waits for subscription for

multimedia content. Also, the server performs some sender-based controls that provide better quality of service for users. The client could be any subscriber that can make a request to the server for subscription, i.e., using the same protocols for negotiation during the streaming. The network interface defined in Testbed is the session-layer interface that uses RTP/RTCP for the real-time delivery of the content. The network emulator is used to emulate a channel whose behavior could be specified using some parameters.

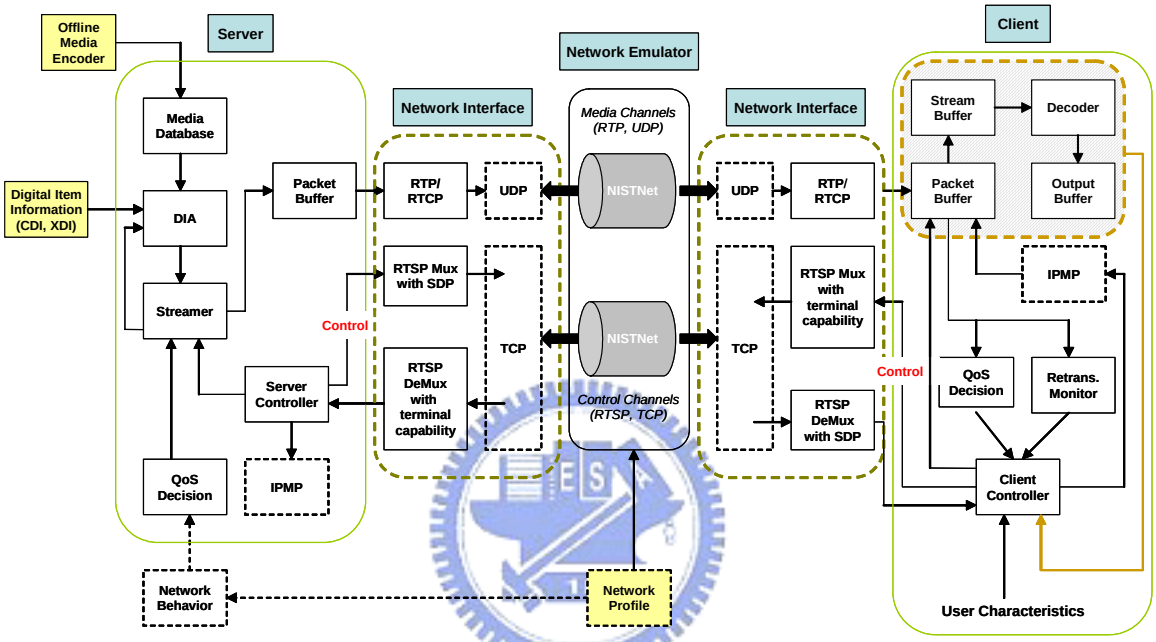



Figure 2-1. Overall architecture of the MPEG-21 Testbed [8]

### 2.1.2 Streaming Server

There are currently five functional blocks in the streaming server of the Testbed system. They are media database, Digital Item Adaptation (DIA), Streamer, QoS decision, server controller. The media source is firstly encoded by the offline media encoder and archived in the storage. The coded bitstream will then be managed by the media database. The media database supports the segmentation of the media into resource unit (in the terminology of MPEG-21 standard). The segmented bitstream will then be retrieved by the DIA engine, which performs resource adaptation according to the external description (Content Description Item (CDI) and conteXt Description Item (XDI)) and both static and dynamic description could be resolved. For video, the resource adaptation refers to the rate control

and rate shaping of the scalable bitstream. The streamer is primarily in charge of sending the adapted resource in a smooth way. To this end, the embedded leaky-bucket traffic shaping algorithm is employed at the streamer. The details of the bitstream segmentation, resource adaptation, and traffic shaping, are explained as following sub-sections. The functionality of each functional block in the server side is stated as follows：

- Offline encoder：Compress the source sequence into scalable bitstream
- Media database：Manage coded bitstream and segment bitstream into video packets
- DIA engine：Provide resource and description adaptation capability
- Streamer：Shape the transmission traffic in a smooth way
- QoS decision：Analytically compute the available transmission rate for each media based on the estimation of available channel rates

*Bitstream segmentation (Media Database)*

Due to the direct access to the bitstream, media database could retrieve some embedded information by parsing bitstream. During encoding process, some resynchronization markers are inserted into the bitstream to improve the error resiliency. Figure 2-2 illustrates the MPEG-4 FGS bitstream structure in term of the wide-sense resynchronization marker (Video Object Planes (VOP) start code, bit-plane start code (BPSC), and resynchronization marker (RM)).

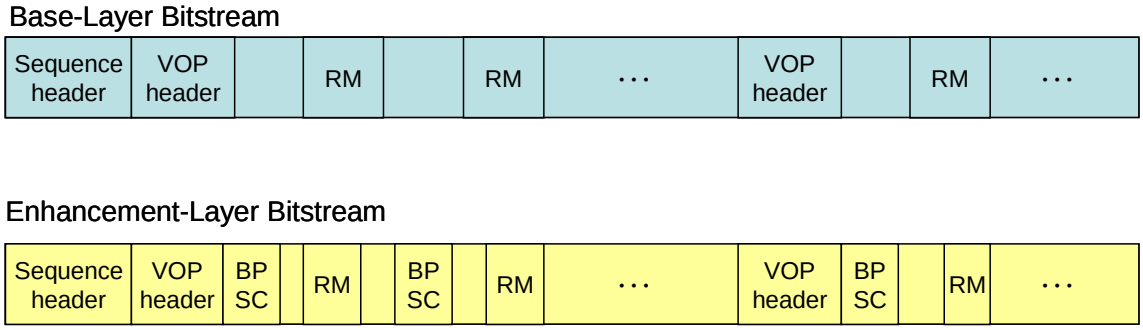



Figure 2-2. MPEG-4 FGS bitstream structure in terms of resynchronization markers

The bitstream segmentation of the MPEG-4 FGS structure is recommended in [27]. This bitstream segmentation could prevent the start-code emulation (i.e., incorrectly recognize the received symbol as another existing symbol due to error) because it prohibits

wide-sense resynchronization marker from being packetized into separate packets. This efficiently improves the error resiliency and the ease of finding next synchronization point for decoding, in case that packet loss occurs. As to the MPEG-4 FGS bitstream structure, the corresponding segmentation is depicted in Figure 2-3. There are additional five types of segmentation in the enhancement-layer bitstream owing to the utilization of bit-plane start code. Those segmentations shown in Figure 2-3 would not suffer any start-code emulation problem even if the packet loss rate is high. Note that each segment mentioned here is not necessarily a RTP packet. However, in the implementation of Testbed, no adaptive packetization is exploited and hence each segment is mapped into one RTP packet.

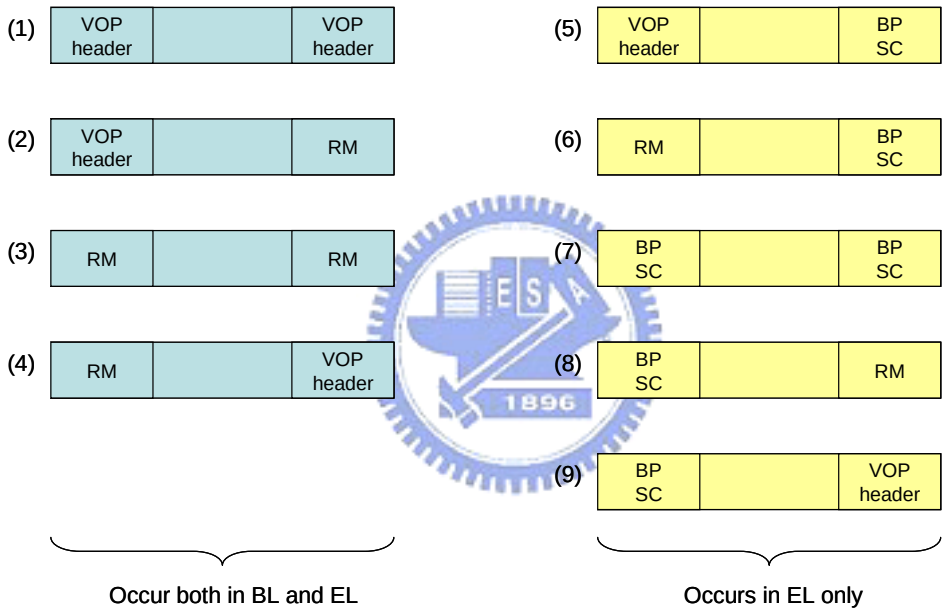



Figure 2-3. Representation of the bitstream segmentation for MPEG-4 FGS

*Resource Adaptation (DIA engine)*

The DIA performs resource adaptation based on the CDI and XDI descriptions. For video streaming, the XDI is generated according to the network profile to describe the available channel bandwidth in some time interval. The resulting XDI is fed into the DIA engine to do rate adaptation of the scalable-coded bitstream. The rate adaptation could be broken into two steps: rate control and rate shaping. The purpose of rate control is to compute the available bit budget for a resource unit (e.g. one VOP). The rate control for video is similar to that proposed in [28], which is based on a weighted average scheme. The

formula is shown as follows:

$$R_{vop}(t) = \frac{R(t) \cdot w_{vop}}{w_I I_{\sec} + w_P P_{\sec} + w_B B_{\sec}} \tag{2.1}$$

Where R(t) is the available channel rate at time t, $w_{VOP}$ is the weighting factor of the associated VOP type, and the $I_{sec}$, $P_{sec}$, and $B_{sec}$ is the number of VOP for each type in this second. The weighting factor $w_I$, $w_P$, and $w_B$ is set to 1, 1, and 0.6, respectively. The advantage of this rate control algorithm lies in its low complexity. However, the resulting visual quality depends on the selection of weighting factors for each type of VOP, while the quality varies among different type of VOP. Furthermore, better visual quality may be archived by R-D optimizing the streaming sequence, but this may require additional computational and/or storage complexity.

The computed bit budget is conveyed into the rate shaping module for truncating the enhancement-layer bitstream. Note here that enhancement-layer bitstream could be arbitrarily byte-aligned truncated except the symbol emulation should be avoided. For example, the last byte of the last resource unit is "0x00" in HEX, while the next resource unit is started with a VOP start code (0x000001b6). Then the concatenated bitstream would be "0x00000001b6…", and this is not a valid codeword in the MPEG-4 bitstream syntax.

*Traffic shaping (Streamer)*

The major purpose of traffic shaping is to keep the transmission as smooth as possible, either in the rate or number of packet sense, depending on the structure of the underlying network. No apparent impact would appear when the transmission rate is low. However, the packet loss would be resulted when the traffic is high during some time interval, provided that traffic shaping algorithm is absent. If the transmission rate is high and no traffic shaping is used, the transmission duration may be relatively short, compared to the total transmission time. This would cause the instantaneous transmission rate exceeding the tolerable transmission rate of the underlying network. Intuitive solution to this problem is to evenly distribute the data in streaming over the total transmission time. Typical solution to this problem is to use either the "Leaky Bucket" or "Token Bucket" algorithm. The Testbed adopts the leaky bucket solution for traffic shaping, and the concept is illustrated in Figure 2-4.

One can imagine that there is a buffer in the underlying network, i.e., in the lower layer

of the protocol stack. The buffer is empty initially, and the transmission drives the data piling up into the buffer with a speed of transmission rate. Once the buffer fullness reaches to specific value, data begins to “leak” from the buffer with a speed of sustainable rate. With a fixed buffer size, a leaky bucket model could usually be characterized with three parameters: *B* (buffer size), *R* (leak rate), and *F* (initial buffer fullness). These three parameters determine the performance of a leaky bucket model for transporting data in a smooth way. To avoid buffer overflow, the buffer size or the leak rate should be large enough. In contrast, the buffer fullness and leak rate should also be adjusted appropriately to prevent buffer from going underflow.

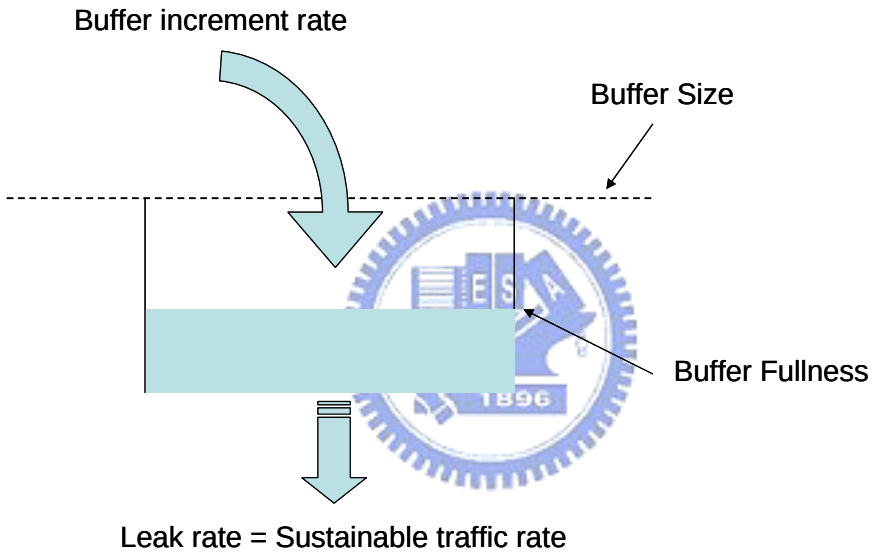



Figure 2-4. Representation of typical leaky bucket model

### 2.1.3 Client

The ultimate goal of the client-side processing is to playback the desired content smoothly. However, the limited bandwidth and random error of channel complicate the processing for good reconstruction. The buffer is especially important because it mitigates the pain from synchronization and variation of receiving rate. The Testbed designs two buffers in the client side: packet buffer and stream buffer. The packet buffer is used to receive packets from the network interface, while the stream buffer provides additional protection to prevent packet buffer from going underflow or overflow. The decoder in the

Testbed is a MPEG-4 FGS compatible decoder with error-resilient capability, i.e., the decoder could continue to decode received bitstream even if the bitstream is corrupted by loss. The error control used in this Testbed is to use the NACK-based retransmission mechanism. The retransmission monitor checks the packet buffer for the lost packet in some frequency, say, one check per second, to record the sequence number of the lost packets. The lost record would be sent from the client to the server via reliable control channel, and the lost packets will then be retransmitted to the client. Besides, the decoder follows an output buffer to moderate the decoding speed and playback. The functionality of each component in the client side is summarized as follows:

- Packet buffer：Smooth out the fluctuation of receiving rate
- Stream buffer：Provide additional protection for packet buffer from going overflow or underflow
- Decoder：Real-time decode the received bitstream with error-resilient capability
- Retransmission monitor：Monitor the packet-lost situation in the packet buffer periodically
- Output buffer：Moderate the decoding speed for smooth playback

*Buffer configuration*

The packet buffer plays a role of receiver buffer that prevents packet reception from duplication and out-of order. In the perspective of rate, the buffer pair regulates the receiving rate, which is strongly correlated with the channel delay/jitter, into the transmission rate of the scalable-coded video clip for smooth decoding. Figure 2-5 shows the interaction among packet buffer, stream buffer, and decoder. The notation “active” used here indicates that the component would continuously and concurrently activate its functionality.

In a normal operation case, we could assume that there are data both in the packet buffer and stream buffer. The data rate input to this buffer configuration at time $t$ is equal to the receiving rate, while the output data rate equals to the transmission rate at the time ($t$-$\Delta$), where $\Delta$ is the pre-roll time. When the network condition is pretty good, the transmission rate is equal to the available channel rate. In this case, the packet buffer is easily overflowed due to the high receiving rate. The packet buffer would immediately put the surplus data (oldest in time) into the stream buffer for decoding, and no packet loss would be incurred.

On the other hand, the packet buffer would suffer underflow in case that the channel condition turns to poor. In this state, the decoder will continue to decode the data stored in the stream buffer even if there is no incoming data at the same time. While the network condition improves, the stream buffer could be refilled with the incoming packet data. Note that this buffer design is primarily to protect the buffer from going underflow, for two reasons. One is that there are more chances for buffer underflow because the unpredictable channel condition is usually poor. Second, the use of scalable coding scheme would not overflow the buffer; the play-out rate would be equal to the receiving rate provided that the available channel rate is always sufficient.

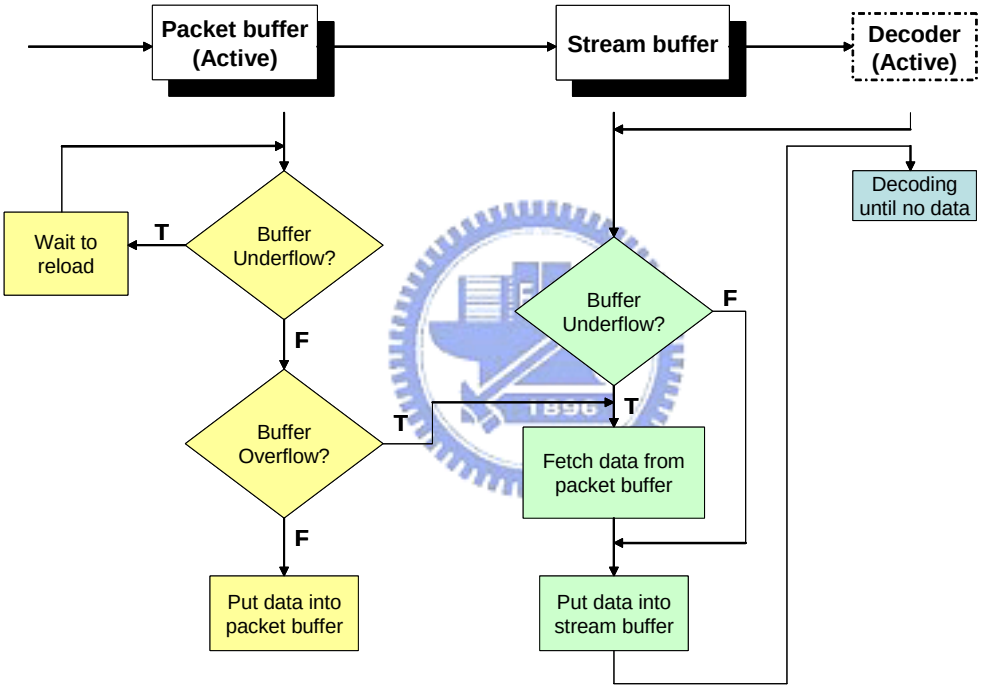



Figure 2-5. Interaction among packet buffer, stream buffer, and decoder in Testbed

### 2.1.4 Network Interface

The network interface used in Testbed is mainly to encapsulate the resource units into RTP packets for transmission. For video, each adapted resource unit is retrieved from the DIA engine by streamer, will then be sent to the integrated packet buffer. The integrated packet buffer consists of the buffer storing the packet data for retransmission and the network interface that is used for packet delivery. Once a resource unit arrives the integrated

packet buffer, a RTP packetizer encapsulates the resource unit as a payload and appends the associated RTP header to the payload, forming a valid RTP packet for video content delivery. The detail description of a formal RTP packet could be found in [29]. Figure 2-6 shows the protocol stack used in Testbed system. The delivery of video data is via the RTP channel, which is based on the best-effort UDP network without proving reliability. As we mentioned in Section 2.1.3, the error caused by packet loss would be recovered by retransmission request from the client to the server side. To make sure the retransmission request could be correctly delivered to the server side, the RTSP messaging is through the reliable TCP channel because the traffic used for retransmission is minor. Note that the retransmission may sometimes fail due to the timing constraint of continuous playback, i.e., retransmission of a packet takes effect only if the packet arrives before it is consumed for playback.

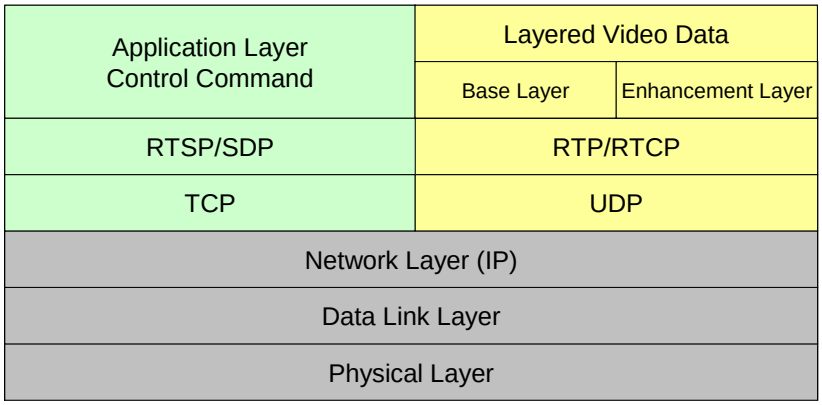



Figure 2-6. Protocol stack in Testbed streaming system

### 2.1.5 Network Emulator

The principal use of the network emulator is to produce a heterogeneous channel environment that a streaming server-client pair would often experience. The Testbed adopts a Linux-based, public domain network emulator named NIST Net, which is developed by the National Institute of Standard Technology, of the United States. In addition to the supply of heterogeneous channel environment, the controllable channel parameter is another significant feature such that the users could create their own characterized channel. Controllable parameters comprise the bandwidth, random packet loss probability, delay, jitter, and size of network queue. Users could adjust these parameters using a text file-based

network profile, and the minimum time interval between two consecutive parameter settings are set to 1 second.

The architecture of the NIST Net is illustrated in Figure 2-7. The logical operation that NIST Net performs is described in the following [30]. Once a packet reaches the computer installed NIST Net, a *packet intercept* code would activate and seize control of the IP packet type handler. All IP packets received by network device will be directly passed to the NIST Net module. *Packet matching* determines whether and how an incoming IP packet should be processed by the *packet processing*, which includes drop, duplication, and delay, of a packet. The processed packet will then be transferred to the Linux IP level code. The *fast timer* takes control of system real time clock and uses it as a timer source for *scheduling* delayed packets. The fast timer reprograms the clock to interrupt at a sufficiently high rate for precise delay processing to a packet.

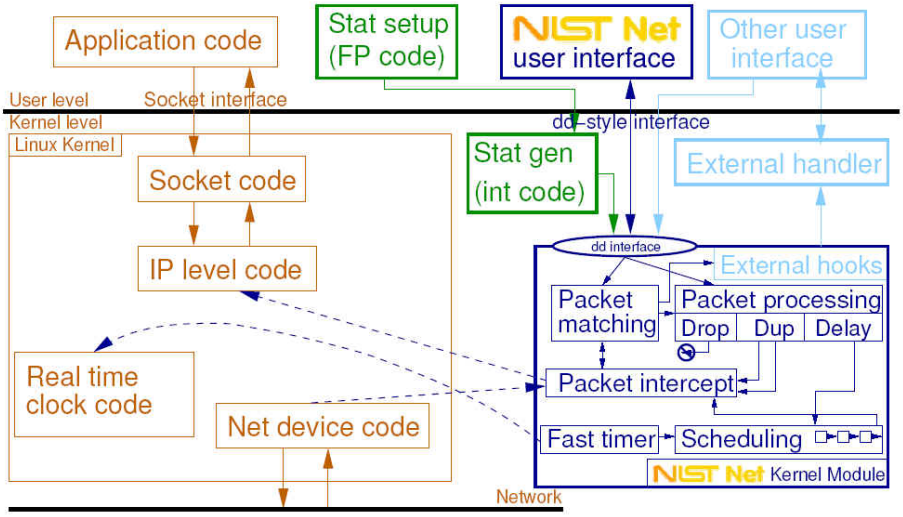



Figure 2-7. Architecture of NIST Net [30]

### 2.1.6 Summary

This section introduces the main architecture and operation of each functional block in the MPEG-21 Testbed resource delivery system. The Testbed is a real streaming system that users could create their own video content and deliver it over network. Some messaging mechanism, which would not significantly influence to the end-to-end performance, is touched on lightly. Besides, the network parameters are controllable, and the parameters

could be specified by network profile that is generated by any underlying network (both wire-line and wireless).

## 2.2 cdma2000 1x-RTT System

The wireless standards often cover large range of technical description that distributed both in the physical layer and the data-link layer. Nevertheless, this thesis aims at the video data transmission over this underlying network, especially for the streaming application of the scalable video stream. In this regard, we will discuss the layering structure of the cdma2000 1x-RTT system, and we would explain the impact of physical-layer attributes on the upper-layer parameters. The term "1x" defined here indicates the single carrier is utilized in the system deployment. The bandwidth variation could be directly derived from the rate assignment of the system, i.e., how many Supplementary Channels (SCHs) are assigned to a data user. In addition, the upper-layer packet loss probability could be computed from the frame error rate of the wireless link, by mapping the upper-layer packet into lower-layer frames. Furthermore, the delay and jitter caused by the wireless transmission could also be obtained from the frame error rate and the number of associated Radio Link Protocol (RLP) retransmission. The affect of lower-layer control such as power control would also be combined into the effect of the Medium Access Control (MAC).

### 2.2.1 Layering Structure

The cdma2000 standard [31] mainly defines the technology details of the link layer and physical layer. The 1x-RTT provides a flexible framework to be the foundation of other ISO/OSI upper layers (network layer, transport layer, session layer, presentation layer, and application layer). Figure 2-8 depicts the layering structure of the cdma2000 RTT in the abstract level. The link layer aims to support control mechanism and protocol interfaces for data services. Another important use of the link layer is to map the logical channels, including the signaling channel and the data channel, into the coding and modulation functions of the physical layer. To support services in cdma2000 (especially for data services), link-layer protocols could be further divided into two sub layers, consisting of

LAC layer and MAC layer, as shown in Figure 2-8.

*Link Access Control layer*

The function of LAC layer is to ensure that various types of traffic are transferred over the air interface according to their QoS requirements. This purpose is achieved by making use of ARQ-based protocols, such as ACK/NACK and sequence-numbering retransmission, to support different level of reliabilities. The degree of error-free guarantee could be made higher at the expense of added delay. Note that for services like circuit-switched voice, the LAC could be a null-functional block.

*Media Access Control layer*

The purpose of MAC layer is two-folded; one is to mitigate the contention issue among applications of a single mobile station, as well as competing among multiple mobiles. The MAC schedules its resources so as to ensure efficient utilization of bandwidth. The so-called resources include buffers, spreading codes, convolutional encoders, and so on. The MAC layer could be further divided into two sub layers, namely, physical layer-independent convergence function (PLICF) and physical layer-dependent convergence function (PLDCF).

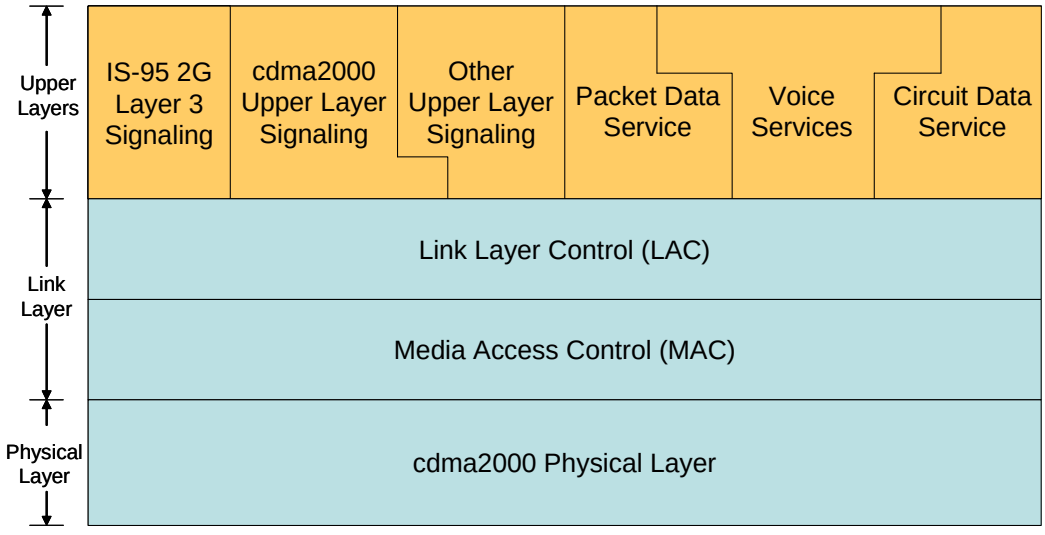



Figure 2-8. Layering structure of the cdma2000 RTT system

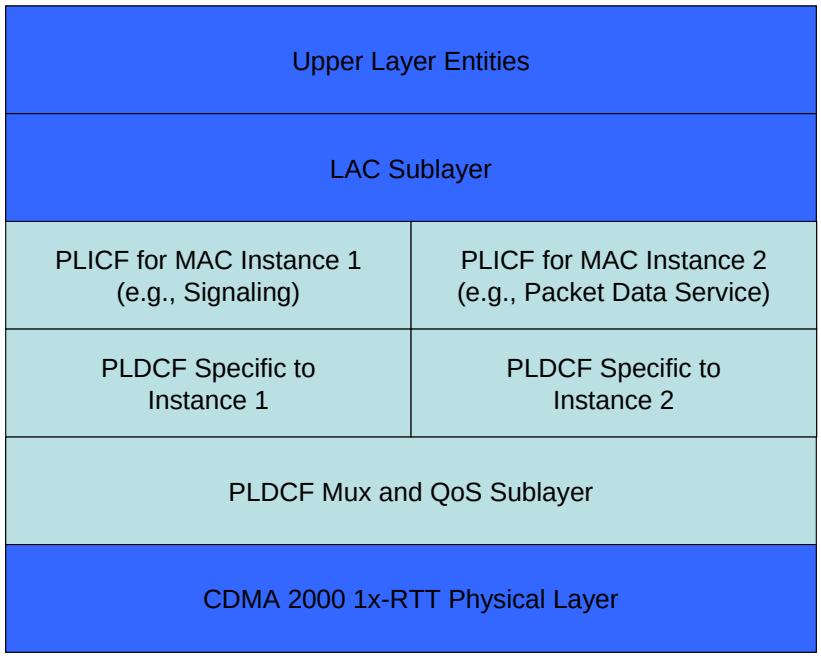



Figure 2-9. Detailed representation of the MAC Layer components

*Physical Layer Independent Convergence Function (PLICF)*

The PLICF is a component of the MAC layer that incorporates all MAC operational procedures and functions that are not unique to the Physical Layer. The PLDCF provides services to implement the actual communication activities in support of MAC layer service. The services that the PLICF uses are defined as a set of logical channels that carry various types of control or data information. Note that these logical channels do not map in a one-to-one manner to the physical channels in the Physical Layer. At a conceptual level, the PLICF can be integrated with any air interface by providing the appropriate PLDCF for that Physical layer. Examples of PLICFs that are defined for cdma2000 include Signaling PLICF, Packet Data PLICF, and Circuit Data PLICF.

*Physical Layer Dependent Convergence Function (PLDCF)*

The PLDCF performs three basic functions as follows. First, the PLDCF maps logical channels from the PLICF to the logical channels supported by the specific Physical Layer. Second, it performs multiplexing, de-multiplexing, and consolidation of control information with bearer data from the control and traffic channels from multiple PLICF instances in the same mobile station. Third, it implements the QoS capabilities, including the resolution of priorities among competing PLICFs (or mobile stations), and mapping of QoS requests

from PLICF instances into the appropriate Physical Layer service requests to deliver the desired QoS. The functions described above could be further divided into two sub-layers of the PLDCF. The Instance Specific PLDCF performs the first function, and the PLDCF MUX and QoS Sub-layer performs the last two functions.

*The Instance Specific PLDCF*

This sub-layer performs any required mapping of the simpler logical channels from the PLICF into the logical channels supported by Physical Layer. Besides, a more important feature is to provide any ARQ protocol that is Physical-Layer dependent and inappropriate for the LAC Sublayer to perform. For cdma2000, four PLDCF Non-reliable ARQ protocols are defined:

- Radio Link Protocol (RLP) – provides a highly efficient streaming service that makes a best effort to deliver data between peer PLICF entities. RLP supports both a transparent and a non-transparent mode of operation. In the non-transparent mode, RLP uses ARQ protocols to retransmit the missing data segment that were not delivered properly by the Physical Layer, and this mode may introduce some transmission delay. In the transparent mode, RLP does not retransmit the lost data segment but maintain byte synchronization between the sender and receiver and notify the receiver of the missing portions of the data stream.
- Radio Burst Protocol (RBP) – provides a mechanism for delivering relatively short data segments with best effort delivery over a shared access Common Traffic Channel. This capability is useful for delivering small amount of data without incurring the overhead of establishing a Dedicated Traffic Channel.
- Signaling Radio Link Protocol (SRLP) – provides a best effort streaming service for signaling information analogous to RLP, but optimized for the Dedicated Signaling Channel
- Signaling Radio Burst Protocol (SRBP) – provides a mechanism for delivering signaling messages with best effort delivery analogous to RBP, but optimized for signaling information and the Common Signaling Channel.

*The PLDCF MUX and QoS Sublayer*

The PLDCF MUX and QoS Sublayer coordinates the multiplexing and

de-multiplexing of code channels from multiple PLICF instances and implements QoS differences between those instances. This sub-layer also maps the data stream and control information onto multiple logical channels from different PLICF instances into requests for logical channels, resources, and control information from the Physical Layer. The PLDCF MUX and QoS Sublayer contain the following mapping functionalities:

- Functions that combine (and separate) traffic and/or control data on logical channels from multiple PLICF instances into (and from) logical channels supported by the Physical Layer.
- Three special Multiplexing/De-multiplexing Functions that perform the lowest level combination (separation) of logical traffic and signaling channel information into (from) physical channels that correspond directly to the code channels that the Physical Layer encodes and modulates (demodulates and decodes).

This sub-layer also provides the following QoS control functions:

- Consolidation of QoS requests from PLICF instances and maps the aggregated QoS requirement into the appropriate Physical Layer resource requests.
- Reconciliation of competing requests among mobile station in the base station.
- Coordination of the mapping and multiplexing/de-multiplexing functions in the PLDCF MUX and QoS Sublayer to deliver committed QoS to PLICF.

Note that the global management of QoS for cdma2000 system is anticipated to be accomplished using Upper Layer Protocols such as ReSerVation Protocol (RSVP). The PLDCF MUX and QoS Sublayer are designed to provide control interfaces and functional capabilities that effectively deliver the associated QoS requirement of Upper Layer Protocols.

### 2.2.2 Framing Structure

The cdma2000 system supports frame structures of 20ms and 5ms. The 20ms frame is mainly used for Fundamental Channel, Supplemental Channel, and Dedicated Control Channel. The Dedicated Control Channel is also provided by the 5ms frame in both Forward Link and Reverse Link. For a 20ms frame structure, 16 pairs of time slots (1.25 ms

per slot) have been allocated for transmission and reception. Figure 2-10 shows the framing structure of the cdma2000 with 20ms frame for a base station.

This framing structure could support various transmission rates for each RLP frame, by differentiating the modulation parameters. For video streaming application over cdma2000 1x RTT system, the Forward Fundamental Channel (F-FCH) and Forward Supplemental Channel (F-SCH) would be often utilized for high-speed data transmission. Table 2-1 and Table 2-2 show possible sets of modulation parameters to support various information rates (encoder input rates) for cdma2000 1xRTT Rate Set 1(RS1) F-FCH and F-SCH, respectively. We adopt the 9.6kbps F-FCH as the fundamental channel of interest. Note that the higher amount of data a RLP frame carries, the higher average frame error rate would be in each frame since less data redundancy could be used for error correction.

Table 2-1. F-FCH RS1 Modulation Parameters [31]

| | Data Rate (bps) | | | | |
|---|---|---|---|---|---|
| Parameter | 9600 | 4800 | 2700 | 1500 | Units |
| PN Chip Rate | 1.2288 | 1.2288 | 1.2288 | 1.2288 | Mcps |
| Modulation Symbol Rate | 9600 | 9600 | 9600 | 9600 | Sps |
| Walsh Length | 128 | 128 | 128 | 128 | PN Chips/Modulation Symbol |
| Processing Gain | 128 | 256 | 455.1 | 819.2 | PN chips/bit |

Table 2-2. F-SCH Modulation Parameters for Data Rates Derived from RS-1 [31]

| Chip Rate (Mcps) | Information Bits per Frame | Encoder Input Rate(kbps) | Code Rate | Puncturing | Modulation Symbol Rate (ksps) | Walsh Code Length |
|---|---|---|---|---|---|---|
| 1.2288 | 168 | 9.6 | 1/2 | None | 9.6 | 128 |
| | 360 | 19.2 | 1/2 | None | 19.2 | 64 |
| | 744 | 38.4 | 1/2 | None | 38.4 | 32 |
| | 1512 | 76.8 | 1/2 | None | 76.8 | 16 |
| | 3048 | 153.6 | 1/2 | None | 153.6 | 8 |
| | 6120 | 307.2 | 1/2 | None | 307.2 | 4 |

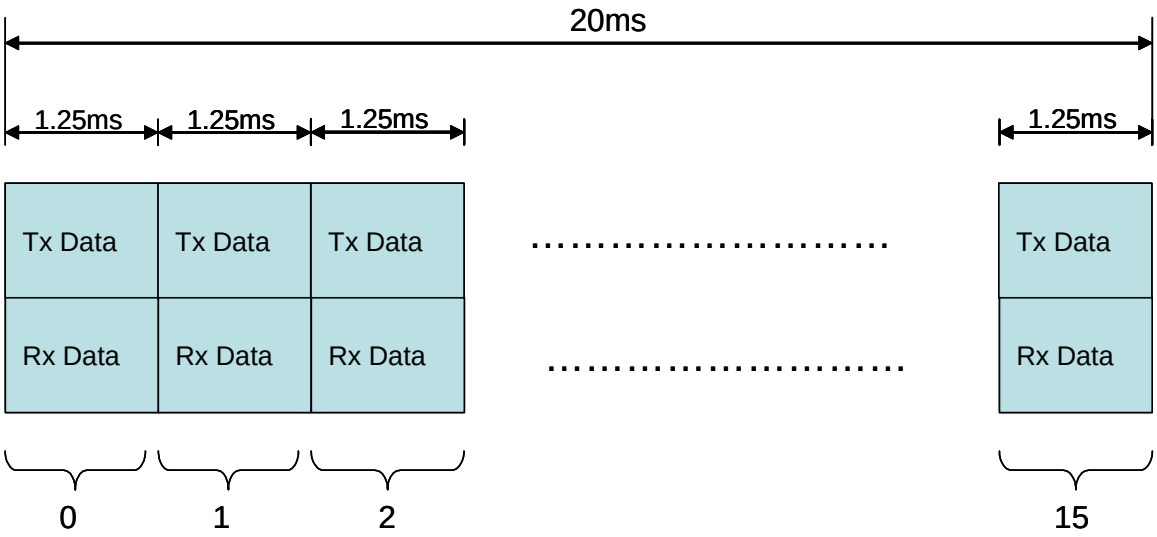



Figure 2-10. Framing structure of cdma2000 system with 20ms frame

For IP-based video streaming over cdma2000 1x-RTT system, each IP packet would be segmented into various number of RLP frames for transmission over the air link, depending on the IP packet size and the rate assignment for each RLP frame. Figure 2-11 shows this mapping relationship. Note that each IP packet would be segmented into an integer number of RLP frames. For each RLP frame, stuffing bits would often be inserted to the last RLP frame of an IP packet. One major reason for this mapping is to ensure the error resiliency in case that a comprised RLP frame is lost. However, stuffing bits would occupy available channel bandwidth and hence the effective transmission rate could be reduced.

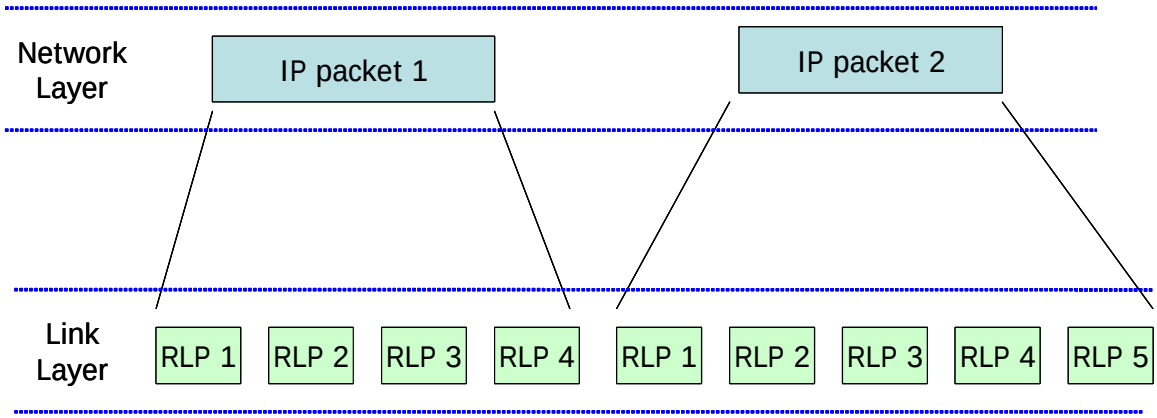



Figure 2-11. Segmentation of IP packets into RLP frames

For 20-ms RLP frames, the number of RLP frames per second is fifty. This implies that the maximum number of IP packets per second should not exceed fifty due to the limitation of available slots. Typically, for low-rate streaming, one video packet would contain one video frame (to reduce the waste on protocol overhead). Hence, the number of packets are approximately twice to the video frame rate due to the use of two-layer bitstream structure of MPEG-4 FGS coding. Adaptive packetization of source bitstream would be relatively important in streaming scenario of interest.

Once the channel error arises, the number of packet should be increased to scatter the loss of bitstream for the ease of post-processing in the client decoder (e.g. error resilient and error concealment decoding). Despite the increase of protocol overhead, the associated framing structure could only tolerate limited number of IP packets to be transmitted through the air link. The exceeding number of IP packet would cause larger end-to-end delay, because the framing structure could not handle the surplus packet within a specific time period. This gives another constraint for adaptive packetization to transmit video content in the IP-based scenario.

# Chapter 3

# AMP-based Buffer Control for Scalable Video Streaming over cdma2000 1x-RTT System

The objective of the proposed control is to prevent the buffer in the mobile station from overflow/underflow during the streaming process. The adaptive media playout algorithm could dynamically adjust the playback frame rate according to the buffer fullness. This mechanism could efficiently resolve the buffer outage but causes damage to the quality of service (QoS). The challenge of AMP-based control is how to protect buffer from outage while maintaining the desired QoS. This chapter initially describes the buffer dynamics of the uncontrolled system. By observing the buffer dynamics, the phenomenon could give us some hints for the buffer control algorithm design, and this would be stated in the second part of this chapter. In addition, the QoS metrics should also be merged into the buffer control design such that the QoS could be further improved.

## 3.1 Components Definition

From a global prospective, the end-to-end architecture of multimedia streaming involves all seven OSI protocol layers. Each component in this architecture may be located in various protocol layers to perform its own operation. A clear definition for each component is necessary for practical design and implementation. This section describes the buffer-related components in the mobile station and clearly explains the logical relationship among these components.

### 3.1.1 Link-layer Components

Figure 3-1 shows the link-layer buffering structure of the cdma2000 1x-RTT system for reconstructing the associated IP packet. The use of RLP queue is for RLP frame retransmission between the peer (BS and MS) RLP layer. An IP packet is announced as lost

if any one of the containing RLP frame is lost during the transmission. Since the retransmission of RLP frame introduces additional delay and reduction in throughput, the maximum number of retransmission for RLP frame should be chosen appropriately. Note that in an error-free channel environment, the IP packet should be received in a correct order. Nevertheless, the receiving of IP packet may be out of order in an error-prone environment due to the delay caused by RLP retransmission. This undesirable impact could be eliminated by fast response to the BS such that the retransmitted frame could be sent as soon as possible. In this thesis, we assume that the receiving of IP packet transmitted over the cdma2000 1x-RTT air interface would not be out of order.

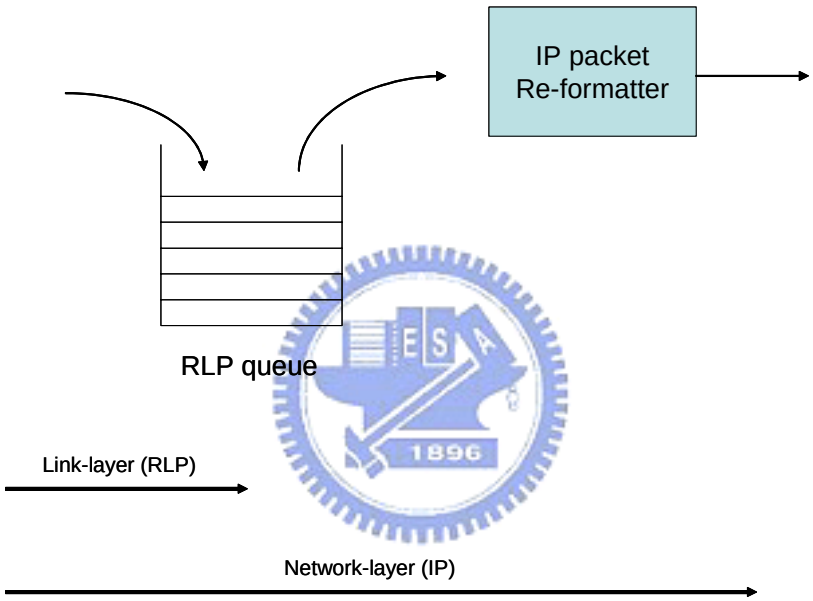



Figure 3-1. Link-layer buffering mechanism

### 3.1.2 Components in OSI Layers 3~7

This sub-section introduces those components in OSI protocol layer 3~7, of the mobile station. When an IP packet is reformatted and de-packetized, the corresponding UDP packet is reformatted. With the similar process, the UDP-RTP packet is de-packetized, while some timing information would be retrieved during the process of RTP de-packetization. This could be used to control decoder's action such as video pre-rolling. After all the protocol de-packetizations are performed, the information bearing data would be stored into the mobile station stream buffer and wait for decoding. Figure 3-2 shows the series of de-packetization process and associated protocol layers.

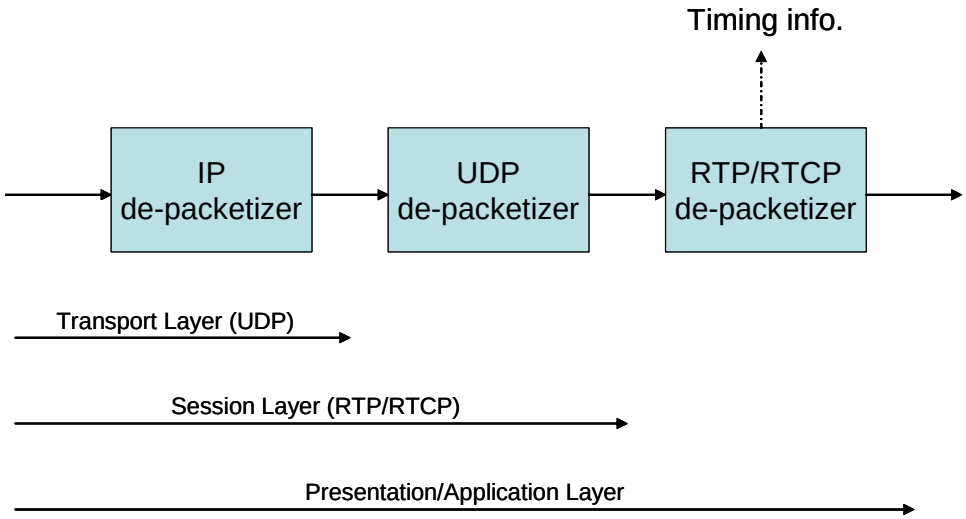



Figure 3-2. Representation of OSI protocol layers 3~7 for video streaming application

Note that for a general streaming application using RTP/RTCP, there are some traffic resulted from the accompanying RTCP feedback messaging. In the following discussion, we neglect the traffic caused by the RTCP signaling for simplicity. Besides, the buffer configuration for decoding MPEG-4 FGS-coded bitstream should resemble that of the MPEG-21 Testbed (sub-section 2.1.3), for practical implementation. Here we omit the interaction of two stream buffers, that is, we assume that the bitstream of both layers are stored in one compound buffer sequentially, and decoder could successfully obtain the separated bitstream for decoding. In practice, this could be done by exploiting a semantic parser for real-time parsing the output bitstream from the compound buffer, as shown in Figure 3-3. Also note that for this scenario, the streaming server should send RTP packets based on their time stamp information such that the sequential fetching behavior from the compound stream buffer in mobile station could result in correct (in the sense of decoding order) reconstruction of bitstream. The design of semantic bitstream parser is relatively simple because the base-layer VOP start code [32] and enhancement-layer VOP start code [15] are distinct.

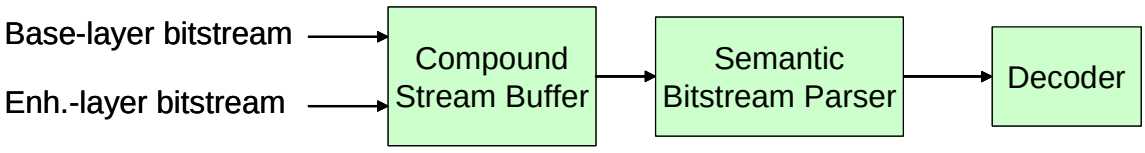



Figure 3-3. Simplified scenario for MPEG-4 FGS-coded bitstream buffering

## 3.2 Buffer Dynamics [33]

This section describes the internal behavior of buffering process during the video streaming experience. We explore some fundamentals from the link-layer simulation of transmitting real-case scalable video bitstream over the cdma2000 1x-RTT system. This simulation combines the use of MPEG-21 Testbed (C++ based) and a link-layer transmission platform (MATLAB based) to closely simulate the practical streaming over the cdma2000 1x-RTT system. Observation results could give system designers suggestions for designing the QoS control and protocols of similar systems.

### 3.2.1 Assumptions

*Rate Controls*

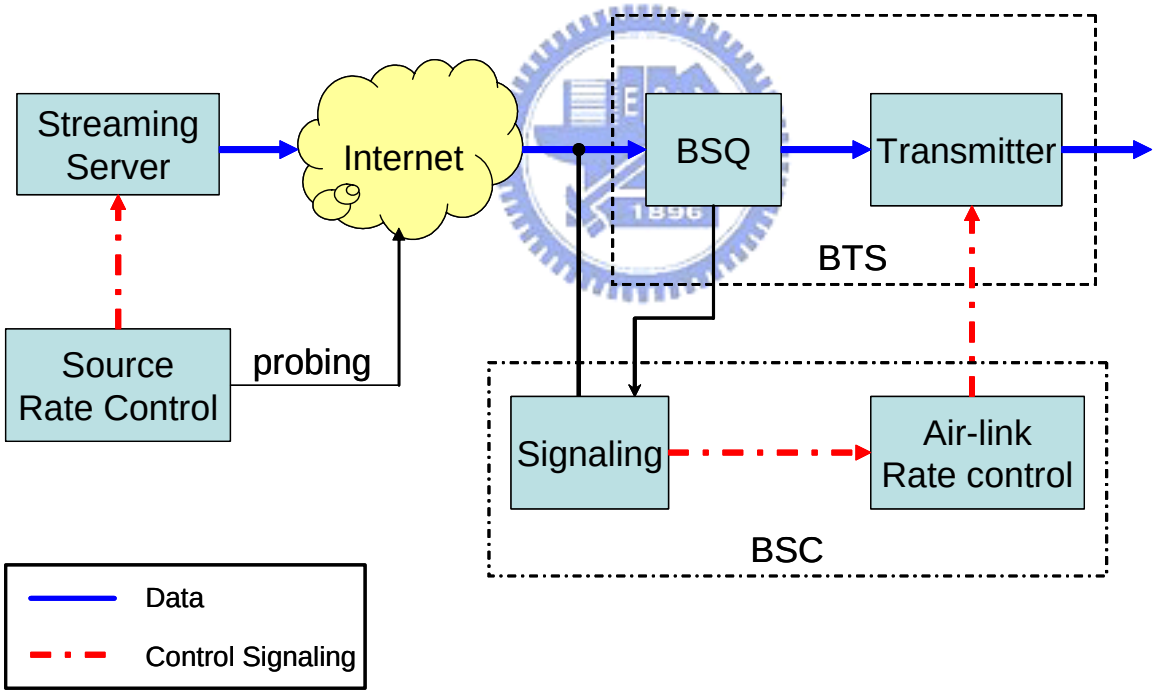



Figure 3-4. Two rate controllers in the transmission path for video streaming application

*Source Rate Controls*

There are typically two rate controllers in the transmission path for scalable video streaming, one is the source rate controller, and the other is the air-link rate control (rate assignment), as illustrated in Figure 3-4. The source rate controller should send out data

based on the channel condition of the subsequent network, including the Internet, base-station queue (BSQ), and available data throughput provided by the Base Station Controller (BSC). Typically, these two rate controllers are located within different devices in the network. If there are no negotiation protocols between source and air-link rate controllers to communicate with each other, or the round-trip delay is relatively large compared to the required response time, the behavior of source rate controller could be assumed to be independent from that of the air-link rate controller. The determination of source transmission rate would therefore be according to the result of probing the channel condition of Internet in the streaming server. We assume that the probed available channel bandwidth follows a uniform process, which is the upper bound of the highest data throughput that the cdma2000 1x-RTT system could support. This assumption may cause some inconsistency from practical streaming case; however, this simple assumption would give us fundamental observations for an end-to-end system perspective.

*Air-link Rate Assignment*

For the radio rate assignment, the data throughput that could be supplied by the system depends on two major factors: the first one is the amount of backlogged data in the BSQ, and the other is the available radio resource. The interaction among techniques of radio resource management such as call admission control, burst allocation, burst admission control, and rate assignment, would be rather complex. To simplify this problem, we assume that the rate assignment follows a Markov chain probability model with static transition probabilities. The assigned rate arises periodically and the corresponding RLP frame error rate is monitored. Once the current RLP frame error rate exceeds the target threshold, the assigned rate drops to lower bit rate levels instantly to ensure the transmission quality. Figure 3-5 depicts the state diagram that accommodates the state transition to the additive-increase multiplicative-decrease (AIMD) concept. Once steady state is reached, the probability of rate assigned to each level should approach a Gaussian distribution, where its mean should be close to the average source rate to support a reasonable streaming.

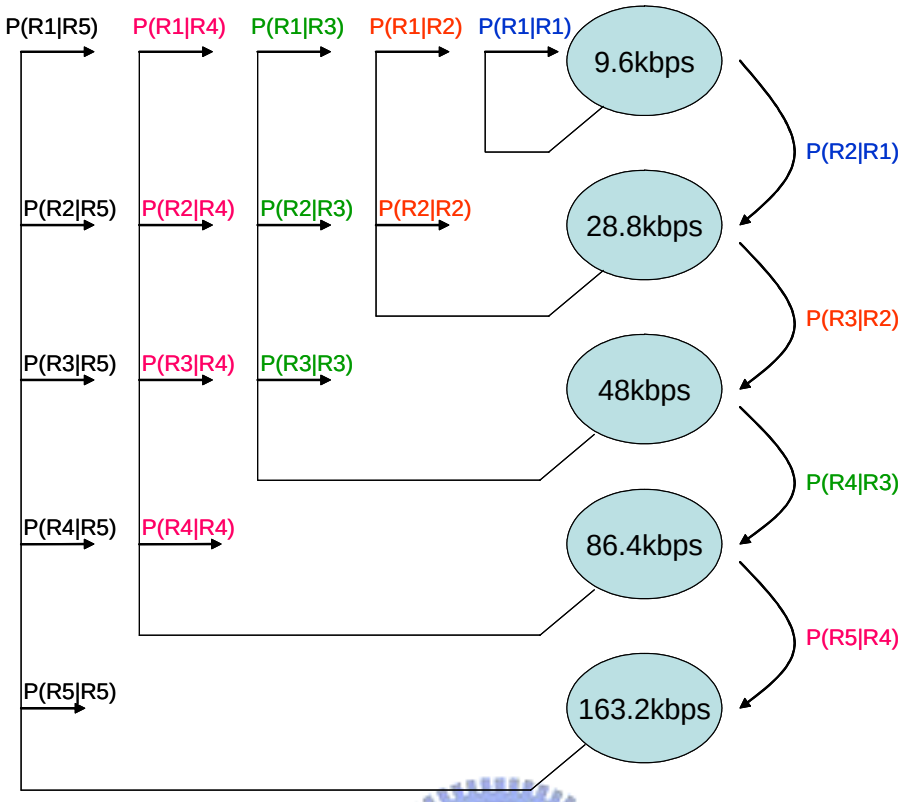



Figure 3-5. Markov-chain rate assignment for data transmission over air interface

*Buffer sizes*

There are mainly two buffers in the transmission path for video streaming. In this paper, we assume that the BSQ assigned for each video streaming user is large enough such that there is no overflow case for BSQ. As to the mobile-station buffer (MSB), the internal application memories in the state-of-the-art 3G handsets could be several megabytes. For example, there are 3.4MB internal dynamic memories in the Nokia 3650 handset [34]. However, this memory is shared for all applications in the handsets, such as phone books, messages, and photos. Therefore, the buffer size for video streaming application may vary.

Allocation of larger dynamic memory for streaming buffer could reduce outages since it lowers the probability of buffer overflow. Similarly, increase of the buffer size causes longer time for bitstream pre-rolling and reduces the probability of buffer underflow. However, a large pre-roll time makes video streaming less relevant because it degenerates to the case of offline file downloading which is insensitive to the channel variation. In addition, for real-time content such as live news, large pre-roll time incurs undesirable long latency

for video playback. Initially, we assume that the effective buffer size is 200kB for video streaming application in the heterogeneous multitasking environment of the Real-Time Operation System (RTOS) in a mobile station.

*Header Compression*

For low-rate video streaming, the protocol overhead occupies a significant portion of available channel bandwidth. Specifically, if the protocol stack of interest are RTP (as session-layer protocol), UDP (as transport-layer protocol), IP (as network-layer protocol), and PPP (as link-layer protocol), the size of overhead for each encapsulating video access unit would be 50 bytes. Assume that the packetization follows the recommendation of RFC 3016 [27], the total amount of overhead per second would reach several kilobits. Nevertheless, the typical per-session throughput for video streaming users of cdma2000 1x-RTT system is around 30~40kbps [26]. Table 3-1 compares the occupied percentage of overhead to the channel throughput for streaming MPEG-4 FGS video with and without header compression, at various frame rates. Obviously, the overhead occupies a large portion of channel throughput for low-rate streaming. Hence, the header compression is necessary for this kind of streaming scenario. Existing header compression technique such as Robust Header Compression (ROHC) [35] could provide both high compression efficiency and robustness to error. Furthermore, the standard development group such as 3GPP currently takes the issue about header compression into account while providing the packet switched streaming services [36] due to the similar reason.

Table 3-1. Comparison of occupied percentage of protocol overhead on the channel throughput among various frame rates

| Overhead / Frame rate | Without compression | | With compression (Compression ratio = 5) | |
|---|---|---|---|---|
| Channel throughput (kbps) | 30 | 40 | 30 | 40 |
| 10 fps | 26.7% | 20 % | 5.3 % | 4 % |
| 15 fps | 40 % | 30 % | 8 % | 5 % |
| 20 fps | 53.3 % | 40 % | 10.6 % | 6 % |

### 3.2.2 Fundamental Observations

Based on the assumptions for rate assignment made in the previous subsection, most cases of buffer outage are buffer underflow due to the limited radio resource allocation, i.e., the assigned data throughput for a single user. To observe the probability of underflow, we consider parameters that would introduce buffer underflow initially. Table 3-2 shows 4 parameters that would cause buffer outage. In the following text, we observe how these parameters including departure and arrival rates influence the dynamics of buffer fullness.

Table 3-2. Parameters that influence the probabilities of buffer outage

| Departure rate | | | |
|---|---|---|---|
| μ(t) ↑ | P(underflow) ↑ | μ(t) ↓ | P(underflow) ↓ |
| | P(overflow) ↓ | | P(overflow) ↑ |
| Arrival rate | | | |
| λ(t) ↑ | P(underflow) ↓ | λ(t) ↓ | P(underflow) ↑ |
| | P(overflow) ↑ | | P(overflow) ↓ |
| 3. MSB size | | | |
| B ↑ | P(underflow) | B ↓ | P(underflow) |
| | P(overflow) ↓ | | P(overflow) ↑ |
| 4. Pre-roll ratio | | | |
| Pr ↑ | P(underflow) ↓ | Pr ↓ | P(underflow) ↑ |
| | P(overflow) ↑ | | P(overflow) ↓ |

We first set the other two parameters fixed, with buffer size B = 200kB and Pre-roll time = 1 seconds. The source rate controller transcodes the FGS-coded bitstream into a truncated bitstream with average bit rate of 40kbps, while the average channel throughput provided by the system is set to 40kbps.

We could observe the buffer fullness F(t) over the whole streaming process in Figure 3-6. We could see that there are some buffer underflow events when the arrival rate is insufficient to sustain the continuous playback. Note that F(t) essentially depends on the

variation of channel throughput (data arrival rate) and the variation of source transmission rate (departure rate).

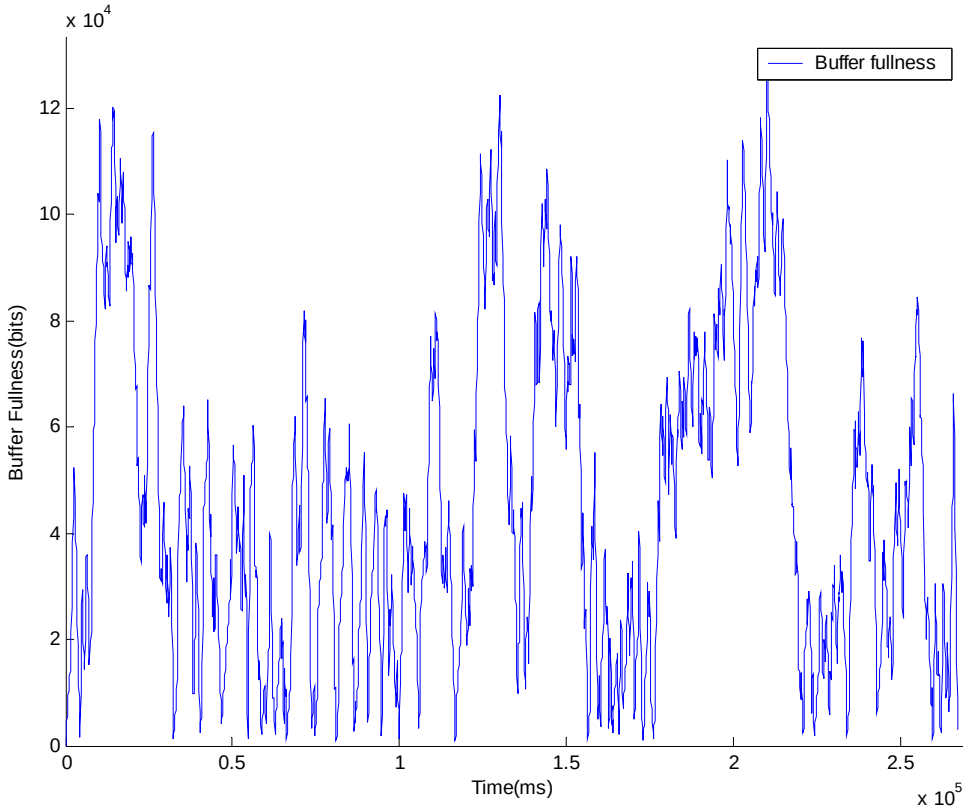



Figure 3-6. Buffer fullness over time during the streaming process

To point out the relationship between buffer fullness and probability of underflow, we study the probability of underflow in this streaming case as shown in Figure 3-7. Intuitively, if the buffer fullness is lower, there is higher probability that the buffer is underflow. Usually, the handling procedure of buffer underflow would halt the playback and wait for next data segment, with duration of pre-roll time. Here we assume that the buffer underflow occurs when the virtual decoder could not retrieve a complete video frame from the MSB at the specific time instant.

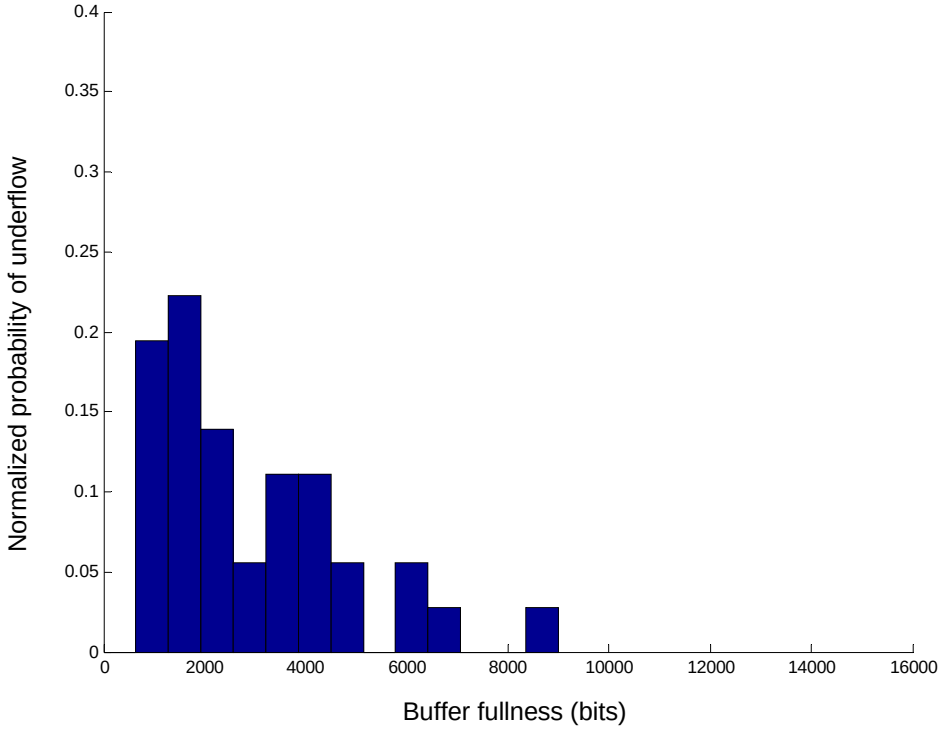
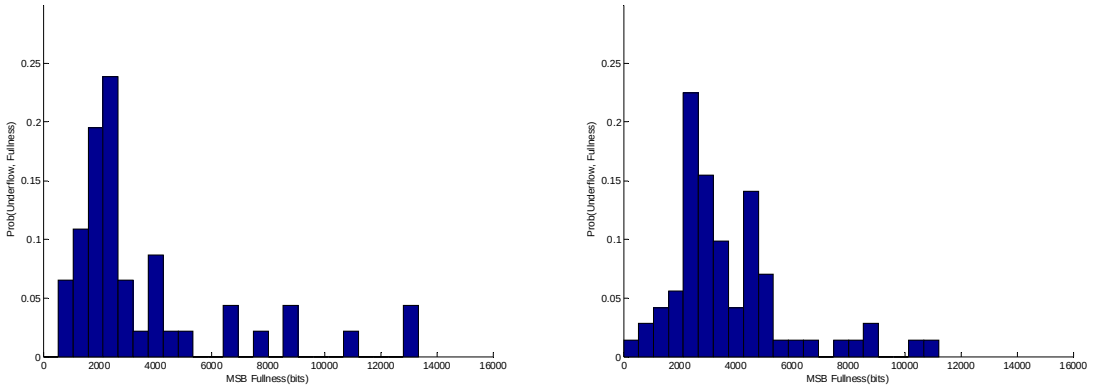



Figure 3-7. Probability of underflow vs. Buffer fullness

From Figure 3-7, we could discover that this probability distribution curve is similar to the shape of a Poisson distribution. However, there are only 36 underflow cases in this streaming case, and the observations (samples) are insufficient to estimate the random behavior of underflow. Figure 3-8 provides more cases for such distribution curves.




(a) (b)

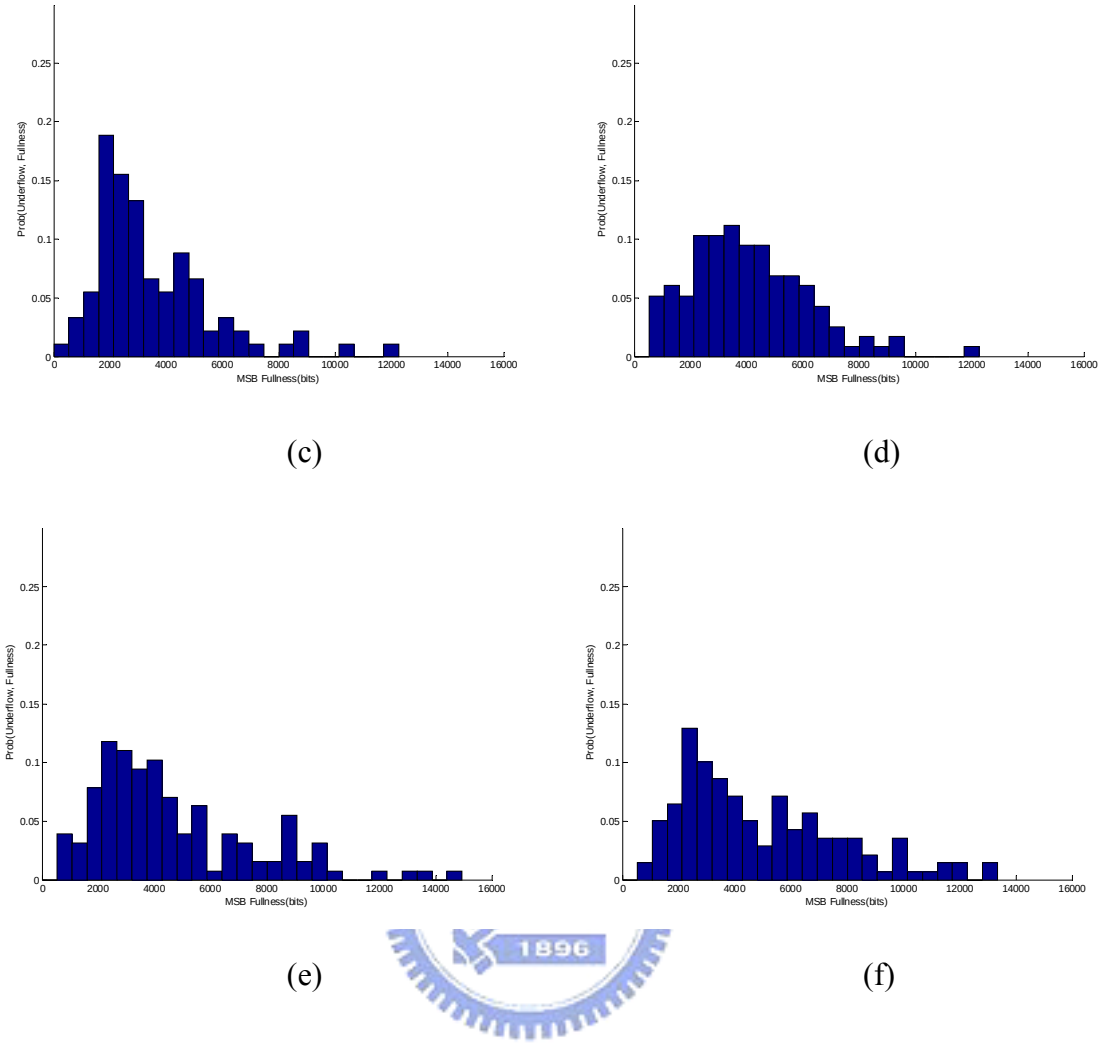



Figure 3-8. Probabilities of underflow vs. Buffer fullness at various source rates (average departure rate) (a)30kbps (b) 40kbps (c) 50kbps (d) 60kbps (e) 70kbps (f) 80kbps

From the observation, we could model the probabilities of underflow at different fullness using exponential distribution curve, which follows the following formula:

$$P(\text{underflow}, \text{fullness} = F) \cong \alpha \cdot \exp(-\alpha \cdot F) \tag{3-1}$$

The value of α represents that the probability of underflow when the buffer fullness is equal to zero, that is,

$$\alpha = P(\text{underflow}, \text{fullness} = 0) \tag{3-2}$$

This exponential curve could model the probability distribution of underflow event successfully, but it requires a scale factor β in the exponent to make the speed of decay faster. Therefore, Eq. (3-1) would become:

$$P(\text{underflow}, \text{fullness} = \text{F}) \cong \alpha \cdot \exp(-\beta \cdot \alpha \cdot F) \tag{3-3}$$

Note that for Eq. (3-3), the resulting exponential curve could not be a valid probability density function because the integration of Eq. (3-3) may not equal to 1. Additional boundary condition should be introduced to justify this probability distribution function.

Next, we will explore when the event of buffer underflow occurs in a realistic situation. The average source rate is set to 30kbps, and the average air-link channel throughput is set to 40kbps. The reason for this higher-rate assignment can be two-fold. One is to compensate for the cause of stuffing bits resulted from the mismatch between IP-packet boundary and RLP boundary, and which effectively increases the required rate for delivering the video content. The other reason is to clearly observe the impact of channel variation. Figure 3-9 shows the number of underflow occurs at various channel variation condition, where the various curves show distinct pre-roll times in unit of second. As expected, longer pre-roll time could endure more variation on channel throughput due to its fewer underflow counts. There are two ways to reduce the event of buffer underflow. One is to increase the pre-roll time, but this is not suitable for the live streaming. Another is to reduce the variation of channel rate, which requires a good radio resource management strategy provided by the system. The stability of channel rate assignment of a video streaming user is also an important factor to the QoS of the streaming experience. Note that it often takes more time to reload data of specific playback duration, and hence the count of underflow does not exactly reflect the total time of frozen playback.

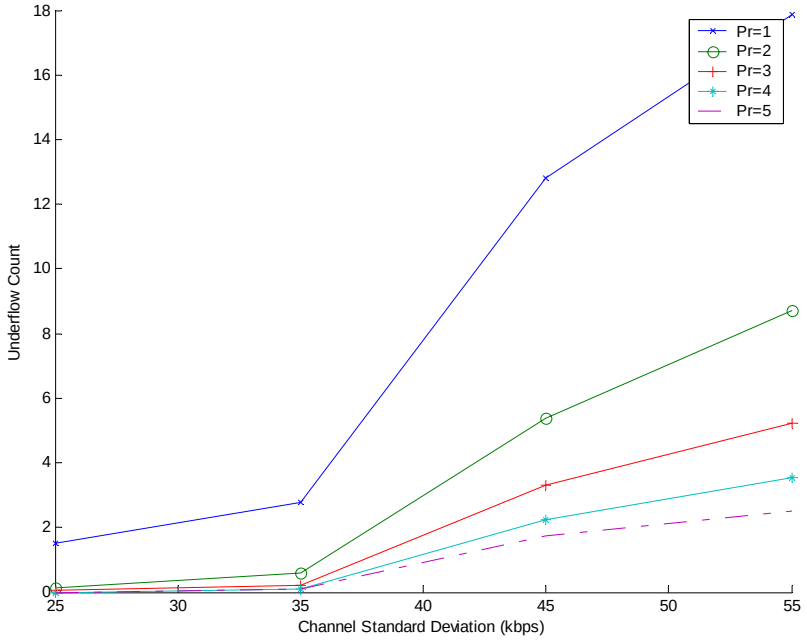



Figure 3-9. The impact of pre-roll time on the underflow

### 3.2.3 Mathematical Analysis

We could start the mathematical analysis for the probability of underflow by considering a buffering structure with size B, arrival data rate $\lambda(t)$, and departure data rate $\mu(t)$, as shown in Figure 3-10. We assume a random process $\Delta R(t) = \lambda(t) - \mu(t)$ and observe its influence on the buffer underflow. Note that the value of distribution boundaries of these bounded processes could be either estimated from the stored record of traffic statistics, or from the results of negotiation protocol such as MPEG-21 Digital Item Adaptation (DIA) [37] to make them as a priori knowledge for these random processes. For example, the maximum, minimum, and average arrival rate could be contained in the XDI description [37], and this information could be sent to the mobile station before streaming. For example, the maximum, minimum, and average departure rate, which represent $\mu_{max}$, $\mu_{min}$, and $E(\mu)$, could be contained in the XDI description [37], and this information could be conveyed to the MS before the streaming process starts.

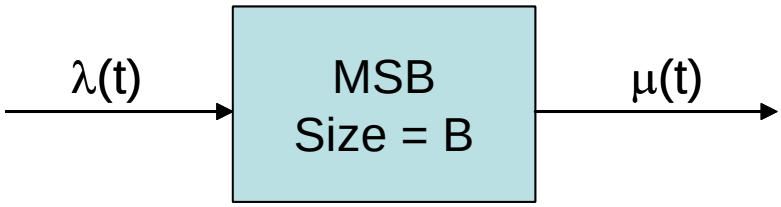



Figure 3-10. Simplified model for buffering

To find out the parameter used in the estimation of probability distribution of underflow defined in Eq. (3-3), we could start derivation from the following formula:

$$P(\text{underflow}, \text{fullness} = \text{F(t)}) = P(\Delta\text{R(t)} < \text{-F(t)}) = P(\lambda(t) < \mu(t) - F(t)) \tag{3-4}$$

If the pre-roll time is long enough to de-correlate the arrival and departure processes we can assume the arrival and departure processes as independent. Hence, the probability defined in Eq. (3-4) could be re-written as:

$$\begin{aligned} P(\lambda(t) < \mu(t) - F(t)) &= \sum_{i=1}^{\infty} P(\lambda(t) < \gamma_{i-1}, \mu(t) - F(t) = \gamma_{i-1}) \\ &= \sum_{i=1}^{\infty} P(\lambda(t) < (i-1)\cdot d\gamma, (i-1)\cdot d\gamma < \mu(t) - F(t) < i \cdot d\gamma) \\ &\approx \sum_{i=1}^{N} P(\lambda(t) < \gamma_{i-1}, \gamma_{i-1} < \mu(t) - F(t) < \gamma_i) \end{aligned} \tag{3-5}$$

where $d\gamma$ is an infinitesimal interval of the arrival or departure rate, and $\gamma_i$ is the boundary value for segmenting the arrival or departure curve into finite intervals. The approximation may lose some accuracy on the estimation for probability of underflow. However, the loss would be negligible when the number of intervals, N, is large enough. We could calculate the distribution for probability of underflow each time the fullness is changed. Nevertheless, the evaluation of distribution curve would cost a lot in the computational complexity since there is infinite number of levels for the buffer fullness. Moreover, the duration between two consecutive changes of buffer fullness could be too frequent. As a result, the frequency of distribution curve updating should be minimized.

To reduce the computational complexity of estimating distribution curve of buffer underflow, we could utilize the approximation function for underflow probability model defined in Eq. (3-3). Let the probability of underflow equal to $P_u(F)$, the probability density function, $pdf_u(F)$, will therefore be:

$$pdf_u(F) = \frac{P_u(F)}{\int_0^\infty P_f(f)df} = \alpha \cdot \beta \cdot e^{-\alpha \cdot \beta \cdot F} \tag{3-6}$$

To estimate these two parameters in the probability density function defined in Eq. (3-6), we can exploit the statistical relationship between the arrival and departure process. Comparing Eq. (3-4) with Eq. (3-3), we could find two boundary conditions for evaluating the parameters α and β in Eq. (3-3).

$$\begin{aligned} \alpha &= P(\text{underflow}, \text{fullness} = 0) = P(\lambda(t) < \mu(t)) \\ &\cong \sum_{i=1}^{N} P(\lambda(t) < \gamma_{i-1}, \gamma_{i-1} < \mu(t) < \gamma_i) \\ &= \sum_{i=1}^{N} P(\lambda(t) < \gamma_{i-1}) \cdot P(\gamma_{i-1} < \mu(t) < \gamma_i) \end{aligned} \tag{3-7}$$

$$\int_{\mu_{\max}}^{\infty} pdf_u(f)df = \int_{\mu_{\max}}^{\infty} \alpha \cdot \beta \cdot e^{-\alpha \cdot \beta \cdot f} df = \delta \Leftrightarrow \beta = \frac{-1}{\alpha \cdot \mu_{\max}} \cdot \ln(\delta) \tag{3-8}$$

, where δ represents the probability of underflow while the buffer fullness exceeds the maximum departure rate $\mu_{max}$, and it is reasonable to set it as a small number. As long as the Eq. (5) and Eq. (6) are both solved, we could obtain the exponential function to approximate the probability distribution curve of buffer underflow.

## 3.3 Model-based Adaptive Media Playout (AMP)

The adaptive media playout algorithm is based on the simple idea: adjust the frame rate such that the probability of buffer outage would be eliminated. However, the variation of playback video frame causes degradation on the subjective quality. The challenge of buffer control by exploiting AMP-based algorithm aims to optimize the visual quality while

maintaining the control performance (or vice versa). We propose an AMP-based buffer control in this section, while taking the temporal variation of visual quality into account. In the following subsections, we detail the proposed buffer control algorithm and some relevant AMP-based control algorithm.

### 3.3.1 Probability Model for Buffer Control

The model-based AMP buffer control mainly utilizes the parameter estimation using Eq. (3-7) and Eq. (3-8). The approximation function with these two parameters would be depicted in Figure 3-11.

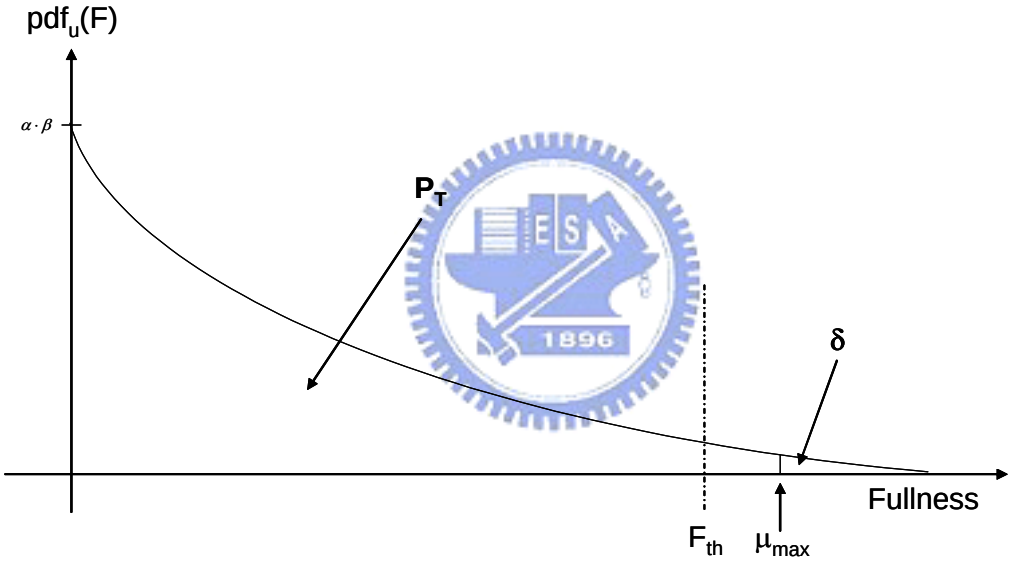



Figure 3-11. Approximation function for probability of buffer underflow

To perform the buffer control, we have to determine the activation threshold of buffer fullness so that power consumed by the control process is reduced. A quality factor $P_T$ could be chosen in advance to ensure that this control decreases the probability of underflow below $P_T$. Once this quality parameter has been determined, the threshold of fullness, $F_{th}$, could be computed from the approximation CDF function, as shown in Figure 3-12. As shown in Figure 3-11, the value of $F_{th}$ does not settle down until the associated covering area is equal to (or larger than) $P_T$. Note that the update for the approximation function should only be done when some of the boundary conditions are altered. For example, once

the maximum/minimum departure rate $\mu_{max}/\mu_{min}$ is changed, the departure distribution function would be modified accordingly. Therefore, we should re-compute the parameters for approximation function of underflow probability. Moreover, the number of segments N would also affects the accuracy of the buffer outage modeling, and which results in a tradeoff between the computational complexity and the precision of the modeling.

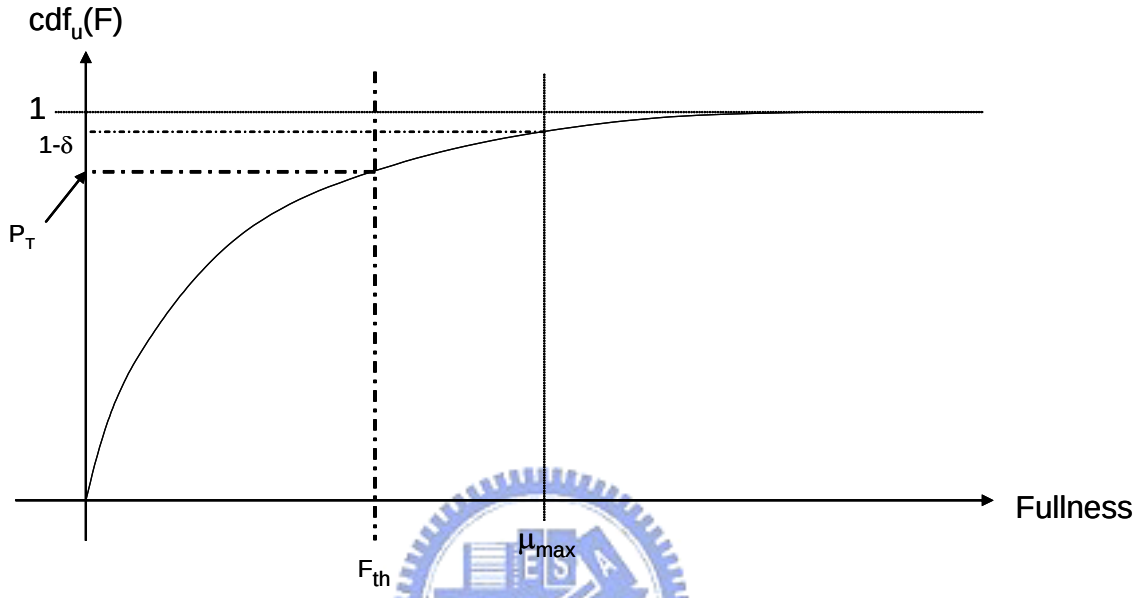



Figure 3-12. CDF of approximation function for computing threshold of activating buffer control algorithm

Based on the similar assumptions made in previous subsection, the steady-state arrival process should be analogous to a Gaussian process, which is bounded by the minimum and maximum air-link channel throughput during some period of time, in the cdma2000 1x-RTT system. Meanwhile we assume that the departure process is uniformly distributed, which is the worst-case assumption due to its high uncertainty. However, the departure process resembles triangular distribution more than a uniform distribution. Since the source rate controller support the rate adaptation from the application layer to the network layer, the protocol stack headers occupy partial available channel bandwidth. Besides, the base-layer bitstream has higher priority for rate adaptation. Thus, it is more probable for the receiver to obtain low-rate video content, and hence the departure process is more likely to become a triangular distribution. Note that we could estimate the true departure process instead of the blind assumption of uniform distribution, such that the parameters of approximation

function could be calculated more precisely.

Figure 3-13 depicts the merge of the assumptive distributions for arrival and departure processes. The arrival process is bounded by the maximum and minimum channel throughput supported by the system over a period of time. Take cdma2000 1x-RTT system for instance, the maximum channel throughput in 100ms would be 163.2kbps*0.1s = 16.32kbits, while the minimum channel throughput would be 9.6kbps*0.1s = 960bits. We could utilize the pre-determined system parameters of channel throughput to formulate the arrival distribution. In the meantime, the departure process is bounded by the source transmission rate, which could be determined by the detection of the backhaul network (i.e. public Internet).

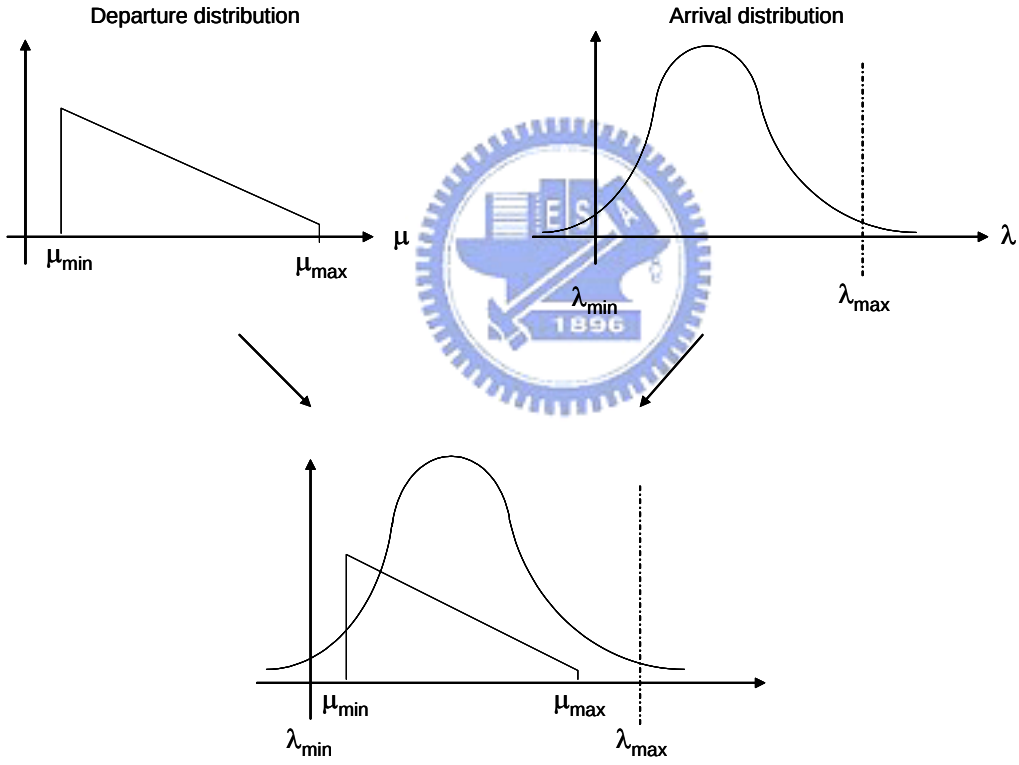



Figure 3-13. Merge of two bounded random process

Figure 3-14 illustrates the segmentation of distribution curve for computing the probability of underflow. We could obtain the distribution curve for probability of underflow each time the fullness is changed. Nevertheless, the evaluation of distribution curve would cost a lot in the computational complexity since there are infinite levels of

buffer fullness. Moreover, the duration between two consecutive changes of buffer fullness could be too frequent. As a result, the frequency of distribution curve update should be made as small as possible. The update of the model should only be done for some period of time to synchronize the inconsistency between the past and current statistics.

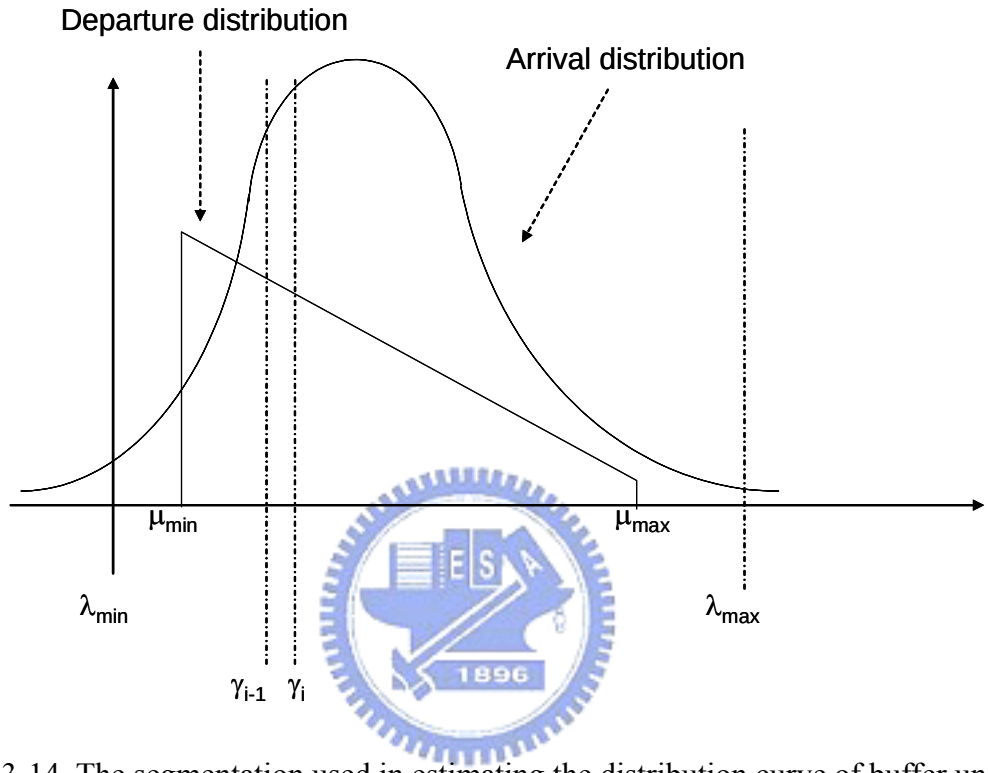



Figure 3-14. The segmentation used in estimating the distribution curve of buffer underflow

### 3.3.2 Objective Visual Quality in Temporal Domain

The AMP-based buffer control principally suffers the annoyance from the variation of playback speed. Hence, we have to also take the visual quality issues into consideration while preventing the buffer from going underflow.

*Mean discrepancy from original frame rate*

This metric is primarily used to indicate the average frame rate gap between the original streaming and the distorted streaming, which may be caused by overflow, underflow, or inadequate buffer control. Higher discrepancy implies that the playback process resembles slow-motion version of the original one (for underflow), in the average

sense. This metric could be formulated as follow:

$$MDFr = \frac{1}{N} \cdot \sum_{t=0}^{N-1} abs\left[OriginalFrameRate(t) - TrueFrameRate(t)\right] \tag{3-9}$$

*Standard deviation of frame rate*

The variation of frame rate describes the smoothness of the playback process. Large variation often causes more disturbances to the human perceptual quality. These metrics points to the long-term variation of the entire playback and implies one of the norm for quality-of-service. This metric could be expressed as the following formula:

$$FrStd = \sqrt{\frac{N \cdot \sum_{t=0}^{N-1} \left[TrueFrameRate(t)\right]^2 - \left[\sum_{t=0}^{N-1} TrueFrameRate(t)\right]^2}{N \times (N-1)}} \tag{3-10}$$

*Short-term standard deviation of frame rate*

Human perception prefers to perceive a smooth video playback in a period of time, i.e., a stable frame rate in some time span. This metric reflects the requirement of stable playback for the human perceptual quality, and which can be written as the following expression:

$$StStd = \sum_{i=0}^{M-1} Std\left\{TrueFR(i \times w), TrueFR(i \times w + 1), ..., TrueFR(i \times w + (w-1))\right\}$$
$$M = ceil(N / w) \tag{3-11}$$

, where the function *Std* computes the standard deviation of the frame rate inside a sliding window of time, the symbol *w* is the size of the sliding window, *N* is the overall playtime for the associated video streaming, and the value of *M* is the total number of computational sliding window.

### 3.3.3 Model-based AMP Buffer Control

In general, an AMP-based buffer control could be split into two parts, the first part is the threshold adjustment, and the second part is the frame-rate adjustment. The objective of threshold adjustment is to determine a proper activation threshold for which the associated

frame rate adjustment is applied. On the side, the principal purpose of frame-rate adjustment is to decide when the next video frame should be retrieved and displayed. The proposed architecture of the buffer control mechanism is depicted in Figure 3-15. The upper half of this plot is the normal path for uncontrolled playback, and the lower half stands for the attachment of the control mechanism. The buffer status monitor keeps tracks of the buffer-related information such as buffer fullness, arrival statistics, and departure statistics, for the control entities to adjust frame rate appropriately. The output of the control entity is the estimated playtime of the next video frame, and which is done by enabling/disabling the decoding process of the decoder. In the subsequent sub-sections, we will discuss the details of the proposed control algorithm.

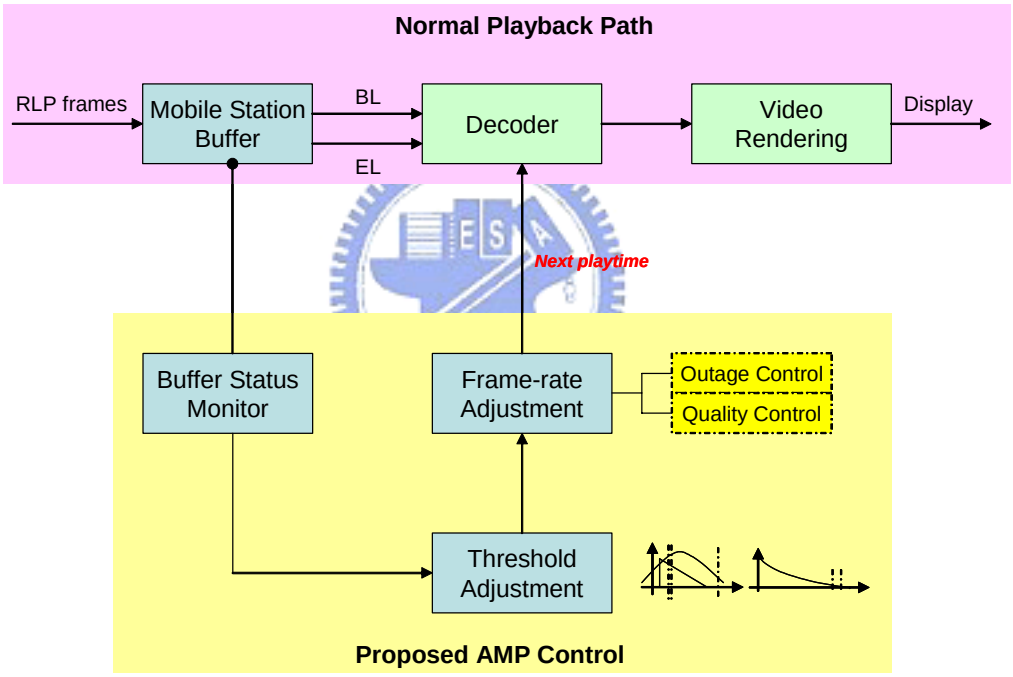



Figure 3-15. The proposed architecture of buffer control mechanism

To apply the control algorithm for the buffer controller, we determine the frequency of control activation in advance. There are in general two states for the control mechanism to operate. The first one the normal state, in which the control algorithm checks the current buffer fullness periodically and the threshold is updated according to the statistics. The second state is the potential outage state, where the control process is activated more frequently to prevent the buffer from underflow. Figure 3-16 shows the impact of the

frequency of the control mechanism. If the frequency of the control mechanism is too low, the probability of underflow would not follow an exponential function, but a uniform distribution. To keep the accuracy of the proposed control scheme, we set the period of the control mechanism equal to the duration of two consecutive video frames. Note that the observation time which is less than the original video frame duration could be important as well when buffer overflow situation appears. Besides, the adjusted frame rate need not be integer since the control mechanism determines only the playback time of the next video frame.

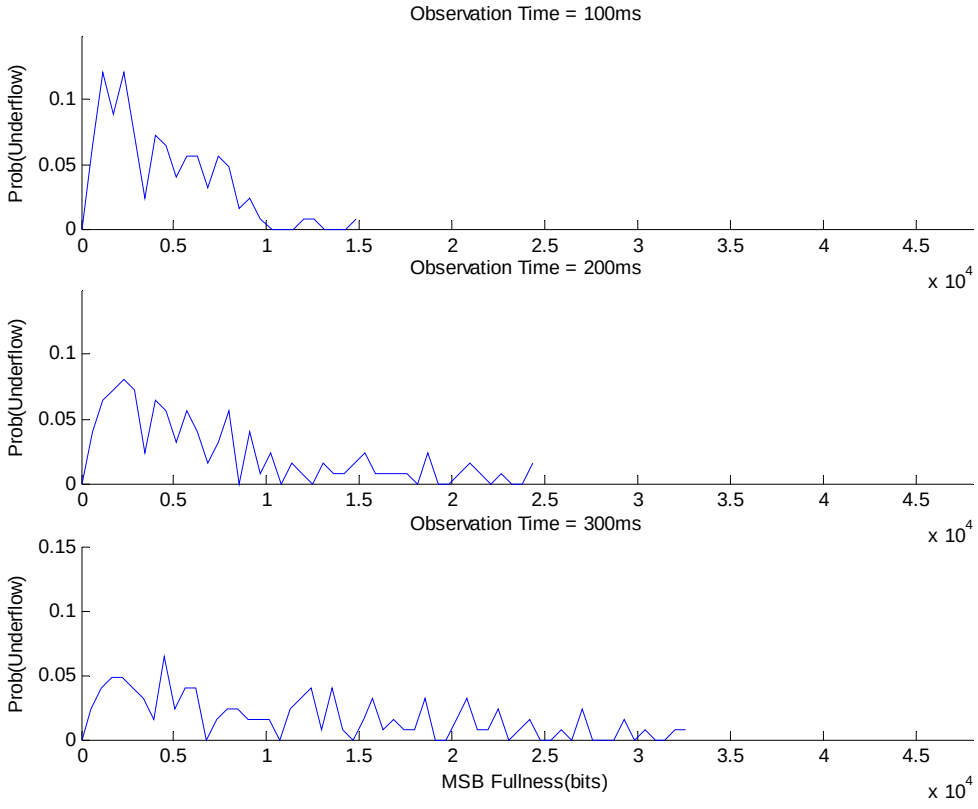



Figure 3-16. Illustration of different observation time

### 3.3.3.1 Outage Control

*Threshold Adjustment*

As mentioned previously, the threshold should be re-calculated when parameters for the hypothetical model has been updated. This could be regarded as a re-synchronization for a primary threshold adjustment. However, the resulting threshold may last for a long period

since the associated traffic parameters such as $\mu_{min}$, $\mu_{max}$, $\lambda_{min}$, and $\lambda_{max}$, change infrequently in the overall streaming process. We name the threshold computed by the model update as the "primary threshold" in this thesis. Based on the calculation of primary threshold, additional threshold adjustment should be introduced to minimize the impairment of AMP control on the visual quality.

For a dynamic threshold adjustment, we adopt a heuristic scheme based on the observation of the static threshold solution. Let us consider the influence of threshold values on the visual quality first. As we can see from Table 3-3, higher value of threshold value causes more discrepancy of the mean frame rate, which lengthens the overall duration of the streaming process. On the other side, the resulting playback suffers less jitter, which represents the variation of frame rate. As a result, we have to determine a proper value for the threshold such that we could obtain a good balance among all quality factors.

Table 3-3. Impact of threshold on visual quality

| | 32k | 48k | 64k | 80k | 96k | 112k |
|---|---|---|---|---|---|---|
| Q1 | 1.1592 | 1.1592 | 1.1951 | 1.1951 | 1.2308 | 1.2661 |
| Q2 | 1.8003 | 1.7306 | 1.6569 | 1.5813 | 1.595 | 1.5382 |
| Q3 | 0.7949 | 0.6981 | 0.68643 | 0.62084 | 0.57663 | 0.57125 |

Q1: Mean discrepancy of frame rate

Q2: Standard deviation of frame rate

Q3: Short-term standard deviation of frame rate

We take the trend of buffer fullness as a fundamental for prediction of underflow. The basic idea of this threshold is simple: Once the trend of buffer fullness is downward, the probability of underflow should arise, and vice versa. If the primary threshold is a low value, the corresponding variation of frame rate would be large since the control would be activated late. Hence, the threshold should arise when the trend of buffer fullness falls such that the AMP control could be activated earlier to improve the margin. The increase/decrease of threshold would be similar to a scaled "mirror" version of the buffer fullness. This extra adjustment would also reduce the jitter caused by the control with only primary threshold. Figure 3-17 illustrate the flowchart of the proposed threshold adjustment,

where the ΔF is the difference of buffer fullness between the fullness of current and last control check time instances, and k is a scaling constant.

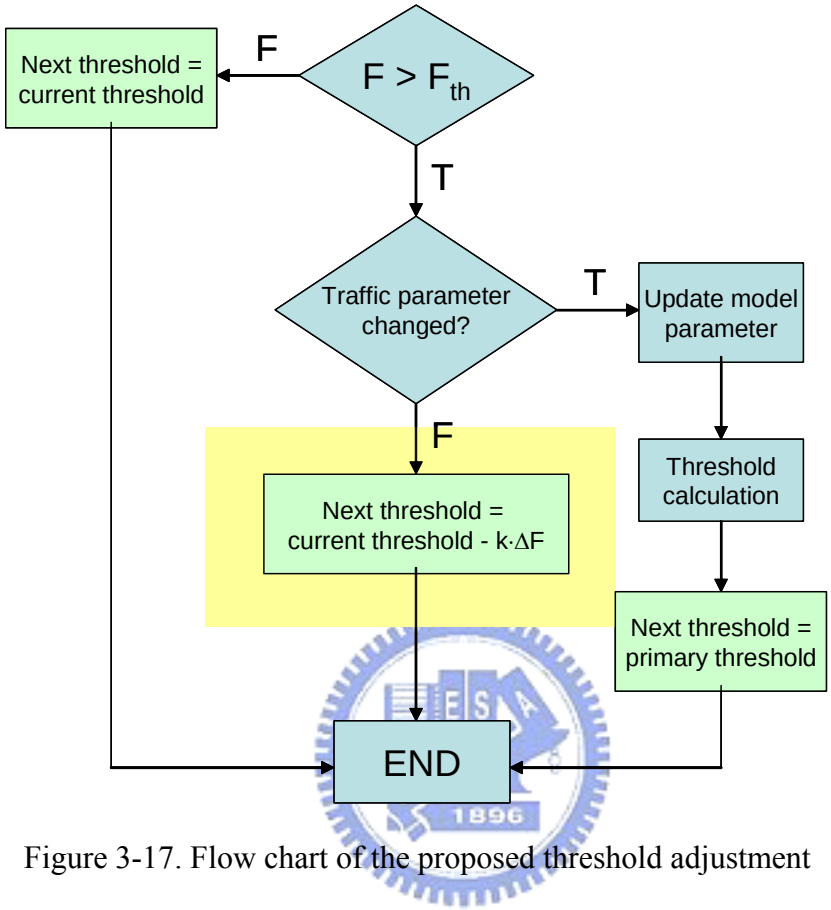



Figure 3-17. Flow chart of the proposed threshold adjustment

*Frame-rate Adjustment*

In this sub-section, we propose two types of frame-rate adjustment for the computation of the next playtime. The first type is the Stochastic-Approaching Adjustment (SAA). The basic idea of SAA is that the adjustment of frame rate is equivalent to the update of departure distribution. As shown in Figure 3-18, the probability density functions of various frame rates imply that the associated threshold would be lower if the adjusted frame rate is reduced. With a fixed control quality parameter $P_T$, SAA searches for the frame rate that the corresponding threshold is just lower than the current buffer fullness. Moreover, after we found the correct frame rate should be at, say, N fps and N-1 fps, a more precise adjustment could be achieved by linear interpolation. This may incur some imprecise result since the fractional frame rate between two consecutive integer frame rates is not distributed linearly. The disadvantage of SAA lies in its computational complexity. For example, when the

original frame rate is set to 10 fps, each model update have to compute 10 distributions instead of one, and which is actually a large computational burden due to the limitation of power in a mobile station.

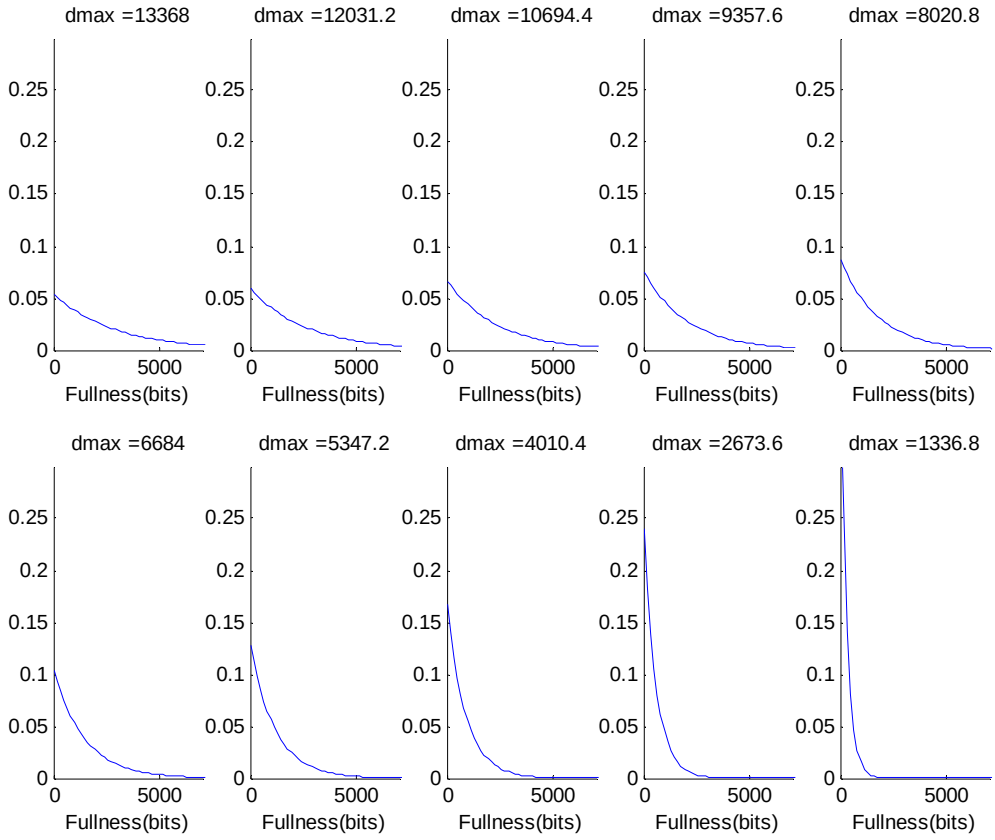



Figure 3-18. Distribution of various maximum departure rate at corresponding frame rate

The second type of frame-rate adjustment is the Content-Aware Adjustment (CAA). This control makes use of the syntax-level information about the video frame size. Owing to pre-rolling mechanism, the size of each video frame could be recorded while they are pre-rolled into the MSB. This information is quite valuable for estimation of the buffer fullness in the next control checkpoint. The estimated buffer fullness for next control checkpoint will be:

$$F_{next} = F_{now} + \left[ A_{next} \cdot \frac{fd_{now}}{20ms} - D_{next} \right] \tag{3-12}$$

,where the capital $F$ represents the buffer fullness, $A$ and $D$ respectively express the amount of arrival and departure data, and $fd$ is the interval between two consecutive video

frames. Here we assume that each RLP frame arrives every 20ms, and the value of $D_{next}$ could be acquired from the history record of the buffer status monitor. Note that the only unknown variable in Eq. (3-12) is the amount of arrival data, $A_{next}$, which could be predicted using simple estimators such as mean, median, or minimum. If estimated buffer fullness is less than some threshold of control activation, say, $F_{th}$, the CAA computes the suitable playtime for the next video frame using the inequality:

$$F_{next} \geq F_{th} \Leftrightarrow fd_{now} \geq \frac{F_{th} - F_{now} + D_{next}}{A_{next}} \times 20ms \tag{3-13}$$

### 3.3.3.2 Quality Control

Once the frame duration has been determined, we perform the stair-case frame duration adjustment to mitigate the effect of the variation of frame rate. The basic concept of this stair-case is that when a buffer status is stringent, it allows large variation on the frame rate to prevent the buffer from going outage. Afterwards, the resumption of frame rate should be smooth enough to decrease the variation of frame rate caused by the associated control. Figure 3-19 gives the conceptual illustration of the idea described above. One important factor to this quality control lies in the decision of the step of each control. Large step causes the acute variation of frame rate; conversely, small step makes the resumption of frame rate slower, which lasts the slow-motion duration of the playback. Therefore, inappropriate step incurs loss in subjective visual quality.

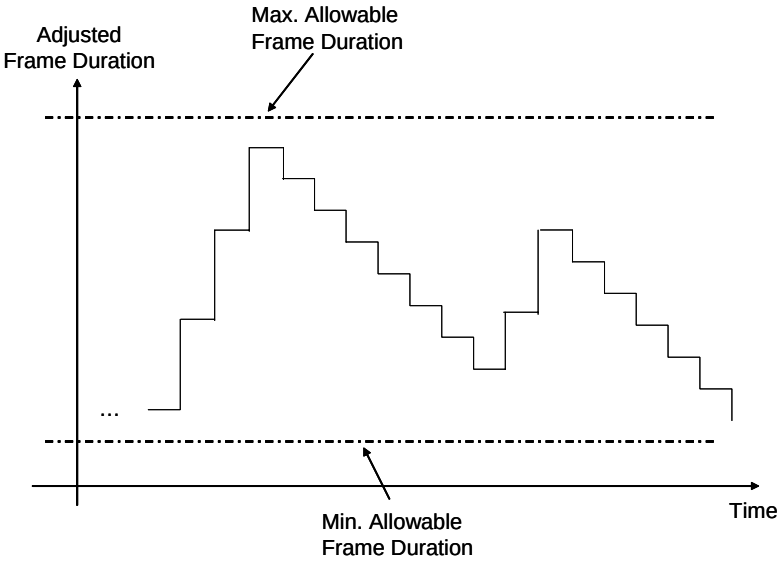



Figure 3-19. Smooth adjustment of frame duration

We perform a heuristic dynamic step adjustment for the staircase quality control. As stated earlier, the reduction in play speed within 25% would be unnoticeable [21]. Equivalently, the stretch of play duration could be 1.33 times of the current time span. The step size could be computed based on the following derivation. Let that the video frame duration of the current and next second be $D_C$ and $D_N$, a scale factor for tolerable variation of video frame duration is $S_T$, and the number of video frame that will be played next second is L. There will be a simple relationship between $D_C$, $D_N$, and $S_T$:

$$D_N = D_C \cdot S_T \tag{3-14}$$

For smooth change of step size, the following expression should hold:

$$step = \frac{D_N - D_C}{L} \tag{3-15}$$

The number of video frame in the next second will be:

$$L = floor\left(\frac{2000}{D_N + D_C}\right) \tag{3-16}$$

By Combining Eq. (3-14), (3-15), and (3-16), we could obtain the step size from the following formula:

$$step = \frac{(1 - S_T) \cdot D_C}{floor\left(\frac{2000}{(1 + S_T) \cdot D_C}\right)} \tag{3-17}$$

,where the unit of $D_C$ is in millisecond corresponding to the value of 2000. Moreover, the value of $S_T$ could be empirically determined for various type of sequence. For example, motion jitter would be more easily perceived for a high-motion sequence than for a slow-motion one. Hence, the value of $S_T$ should be lower for a high-motion sequence. A more proper selection of the value for $S_T$ could be done by automatically taking the motion information of a sequence into consideration.

Figure 3-20 illustrate the state diagram of the proposed control algorithm. There are three states for buffer controlling. The state 0 is used for distribution curve updating while either the arrival process or departure process alter its statistical characteristics. The state 1 is the state for normal operation of decoder. The state 2 is used to process the underflow

event with proper choice of playout frame rate. Each state transition may cause associated adjustment of playback frame rate. Note that the transition from state 2 to state 1 involves a constant N, where this parameter should also be model-relevant to efficiently achieve better performance on both control and visual quality. The proposed buffer control is principally founded on the probability models of the arrival and departure processes. With appropriate approximation on these two probability distributions, the resulting estimation of probability of buffer underflow would be relatively precise

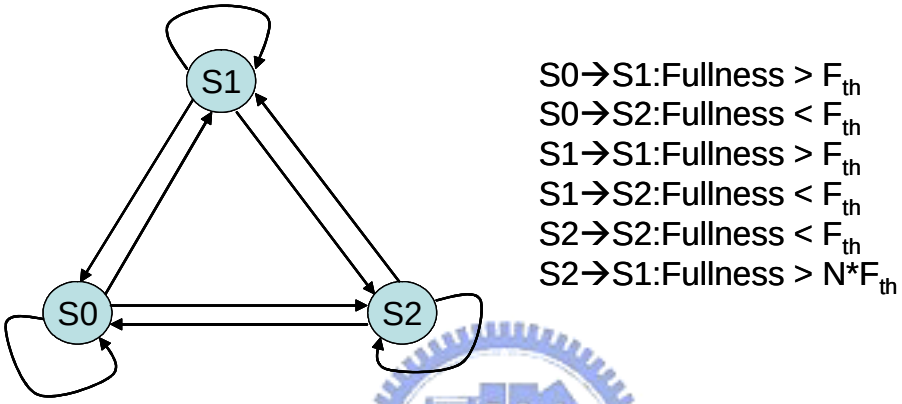



Figure 3-20. The proposed AMP-based buffer control mechanism

### 3.3.4 Summary

In this section, we derive an analytical formula for estimate the parameters for probability of underflow with various value of buffer fullness. The derivation is based on assumptions to the arrival and departure processes. The update of parameter should be done when the associated traffic parameters are changed during the streaming process, and it generates the primary threshold. Additional threshold adjustment could be done in a counter-fullness fashion to reduce the variation of frame rate, while the frame rate could be computed according to the result of threshold calculation.

# Chapter 4

# Experimental Results

In this chapter, we explore the performance of the proposed AMP-based control scheme. For higher data throughput, we also assume that some header compression techniques are employed, to reach a compression ratio of 5. Header compression becomes significant factor for evaluating the end-to-end performance for low-rate streaming, as discussed in Section 3.2.1.

The experiments employ Monte-Carlo method with 50 times of simulation. Furthermore, we assume that the rate assignment for each RLP frame could be done every 20ms, i.e., the transition between two states could take place every two consecutive RLP frames. Note that we support an air-link channel with average throughput of 40kbps instead of 30kbps. The reason of this higher-rate assignment can be twofold. One is to compensate for the cause of stuffing bits resulted from the mismatch between IP-packet boundary and RLP boundary, and which effectively increases the required rate for delivering the video content. The other reason is to clearly observe the impact of channel variation.

## 4.1 Experimental Scenario Design

Figure 4-1 illustrates the structure of the platform for simulation of link-layer transmission in this thesis. The MPEG-21 Testbed plays the role of a streaming server located in some node in the Internet, and it provides the source of video content from OSI layer 7 to layer 3 (add protocol headers gradually). Its output is the IP-packet profile, which consist of information about sequence number, time stamp, packet size, and associated video frame. Figure 4-2 gives an example of the associated packet profile. The Internet model currently is a null model, which could determine some transmission parameters of the network layer, such as packet loss rate, delay, and jitter. We assume that the backhaul network is an error-free environment, i.e., there is no packet loss in the backhaul network. Besides, we also assume that the delay/jitter is relatively small so that the base-station queue could absorb all the packet disorders. Therefore, all the IP packets are sent out by the

base station in the original delivery order of the streaming server.

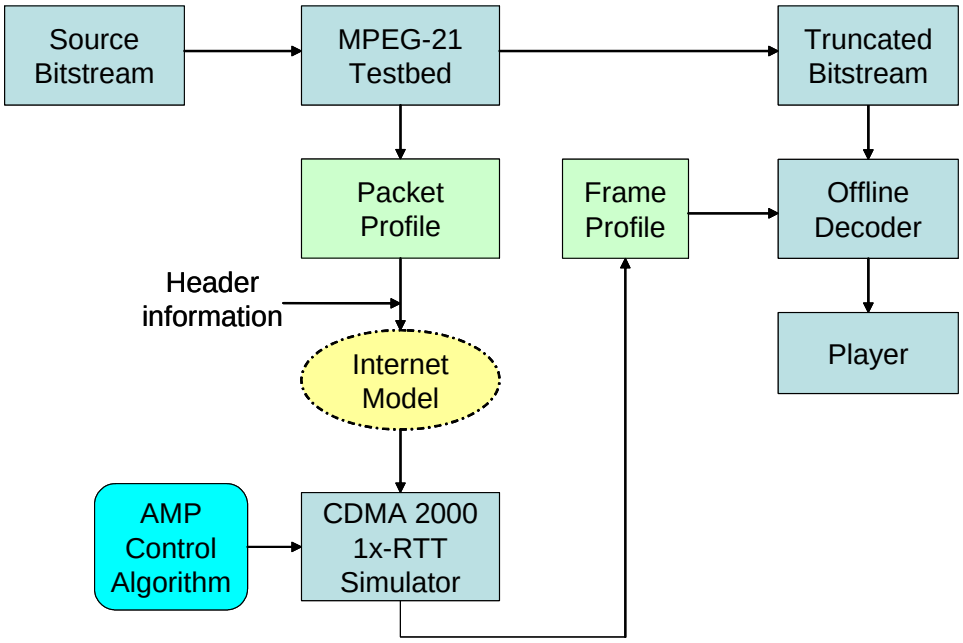



Figure 4-1. Platform built for link-layer simulation of realistic video content

| sn | ts | size | FrameNo |
|---|---|---|---|
| … | … | … | … |
| 760 | 38 | 177 | 386 |
| 761 | 38 | 192 | 387 |
| 762 | 38 | 256 | 388 |
| 763 | 38 | 259 | 389 |
| 764 | 39 | 256 | 390 |
| 765 | 39 | 264 | 391 |
| … | … | … | … |

Figure 4-2. An example of packet profile designed for the simulation platform

*cdma2000 1x-RTT simulator*

The air-link transmission behavior is provided by the cdma2000 1x-RTT simulator. Each IP packet (from packet profile) is manipulated by this simulator, which performs system-level simulation with time resolution of millisecond. Figure 4-3 illustrates the insight of the cdma2000 1x-RTT simulator. Specifically, each IP packet would be

segmented into several RLP frames by using the aforementioned air-link rate assignment strategy. Each RLP frame is transmitted over the air-link channel and received by the mobile station. In addition, the dynamics of the MSB is particularly focused by monitoring the influence of each received RLP frame on buffer fullness. A complete IP packet would be reconstructed once the associated RLP frames are received, while a video frame follows the similar reconstruction process with several IP packets. Moreover, the proposed buffer control algorithm is embedded into this simulator, and the control mechanism determines the playtime of each video frame, which is recorded in the frame profile. To visualize the impact of various control algorithms, the frame profile could be fed into the modified client side of the MPEG-21 Testbed, which possesses a realistic player for the playback of the streaming process.

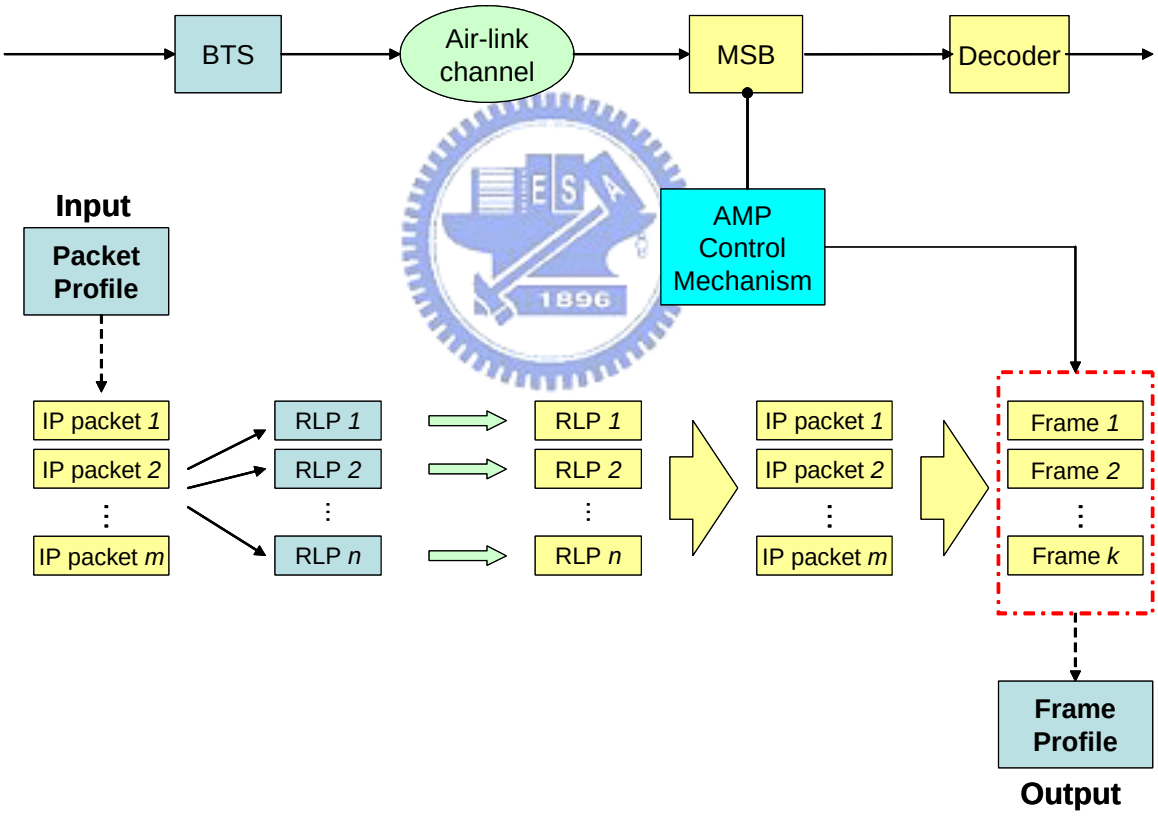



Figure 4-3. Transceiving behavior of cdma2000 1x-RTT simulator

*Prototyping system for visualization of AMP algorithm*

Figure 4-4 depicts the modified client-side architecture from the MPEG-21 Testbed (see Figure 2-1, client side) to visualize the effect of AMP-based control algorithm. The

inputs are the truncated bitstream from the streamer of MPEG-21 Testbed, and the frame profile from the cdma2000 1x-RTT simulator. Without modification, the decoder operates at its full speed of a thread to decode the incoming bitstream. The reconstructed video frames are then put into the output and wait for fetched by the player. Moreover, the fetching time of the player is controlled by a fast timer, which reads the frame profile and performs accurate timing control. The timing resolution of the fast timer could be at milliseconds, which is a sufficient precision that could support the proposed control mechanism in this thesis. Figure 4-5 shows the graphical user interface of the prototyping system. Once the truncated bitstream and the frame profile are input, a window would appear to emulate the playback process of the associated streaming. Moreover, this system could be utilized to evaluate the subjective quality of various AMP-based control algorithms, by either formal [38] or informal subjective assessment (e.g. simple mean opinion scoring).

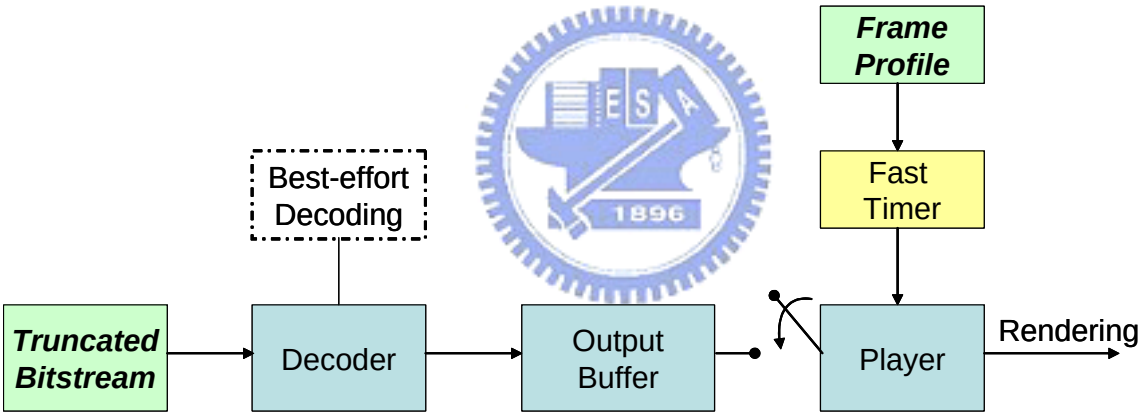



Figure 4-4. Architecture of the prototyping system for visualization of AMP-based control algorithms

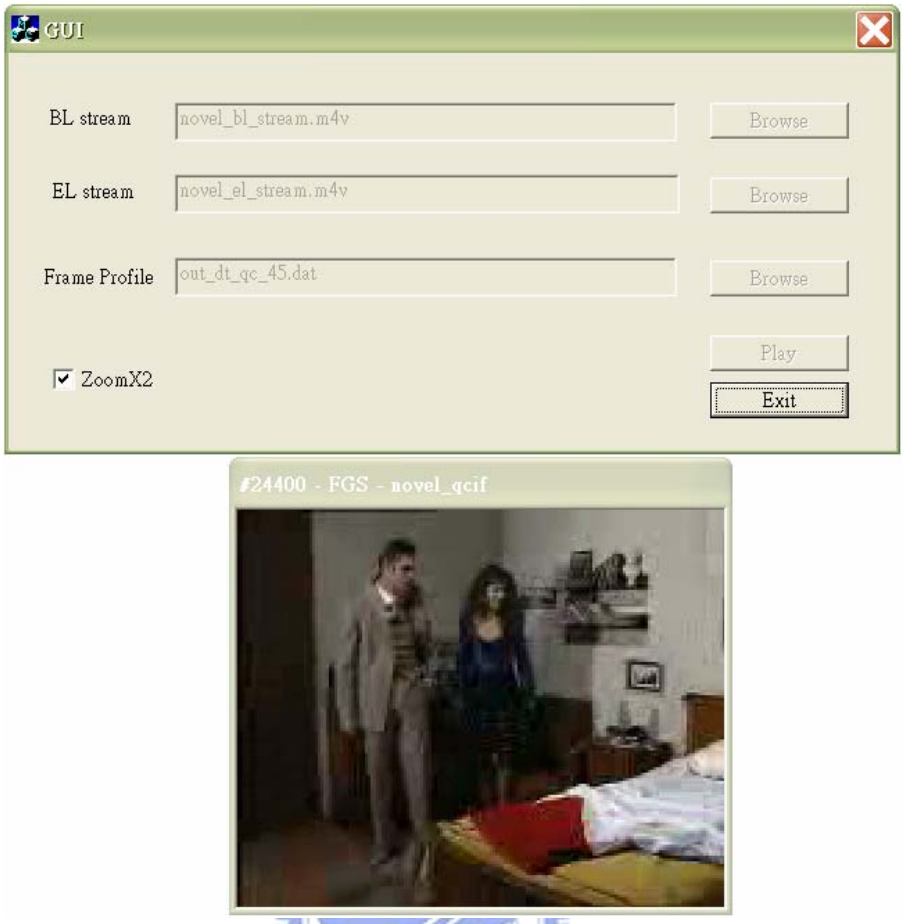



Figure 4-5. Graphical user interface of the prototyping system for visualization of AMP-based control algorithm

## 4.2 Experimental Results

Some common environmental parameters for the subsequent experiments are defined in Table 4-1.

Table 4-1. Some environmental parameters for the subsequent experiments

| Parameter | Value |
|---|---|
| $P_T$ | 0.99 |
| Header compression ratio | 5 |
| Buffer size | 200kB |
| Length of video clip | 216 sec. |
| Video format | QCIF (176x144) |

| | |
|---|---|
| Source frame rate | 10 fps |
| Pre-roll time | 1 sec. |
| Avg. source rate | 30 kbps |
| Avg. channel throughput | 40 kbps |

### 4.2.1 Comparison of Frame-rate Adjustment Schemes

Recall that we proposed two novel frame-rate adjustment schemes in section 3.3.3.1, namely, Stochastic-Approaching Adjustment (SAA) and Content-Aware Adjustment (CAA). To compare the performance of control, we adopt two other types of simple frame-rate adjustment schemes. The first type is a simple staircase adjustment, and the second type is a common linear adjustment. Specifically, the staircase adjustment increases/decreases frame rate when the current buffer fullness is less than a specific threshold. The linear adjustment simply maps the current fullness into a corresponding frame rate in a linear sense, whose concept is illustrated in Figure 4-6.

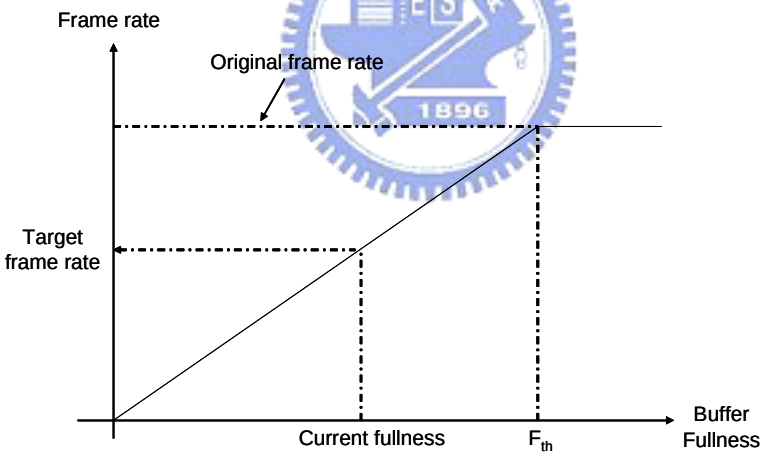



Figure 4-6. Illustration of linear frame-rate adjustment

We firstly compare the performance of the raw frame-rate adjustment schemes, i.e., we turn off the use of dynamic threshold adjustment and the quality control to observe the nature of the associated schemes. The metrics of performance are those stated in Section 3.3.2, and the short-term standard deviation is calculated with window size of 3 seconds. Figure 4-7 shows the mean discrepancy of frame rate, which implies a mean latency of the playback using some control scheme. The horizontal axis of the plot is the standard deviation of the channel throughput (abbreviated as channel_STD in the following text),

and whose mean throughput is defined in Table 4-1. As expected, a higher channel variation results in higher probability of buffer outage. We could discover that all the curves are nearly overlapping, and hence their latencies are almost the same during the whole streaming processes. This indicates that the system occupancy times are similar among these control algorithms, and which implies that the system capacity would not be reduced due to the introduction of control mechanism. An improper control scheme would lengthen the latency of the streaming, and hence increase the occupying time for a wireless system. This is an undesirable side effect of control scheme since the system resource of the associated user could not be released until the streaming is over, and which reduces the throughput provided by the system.

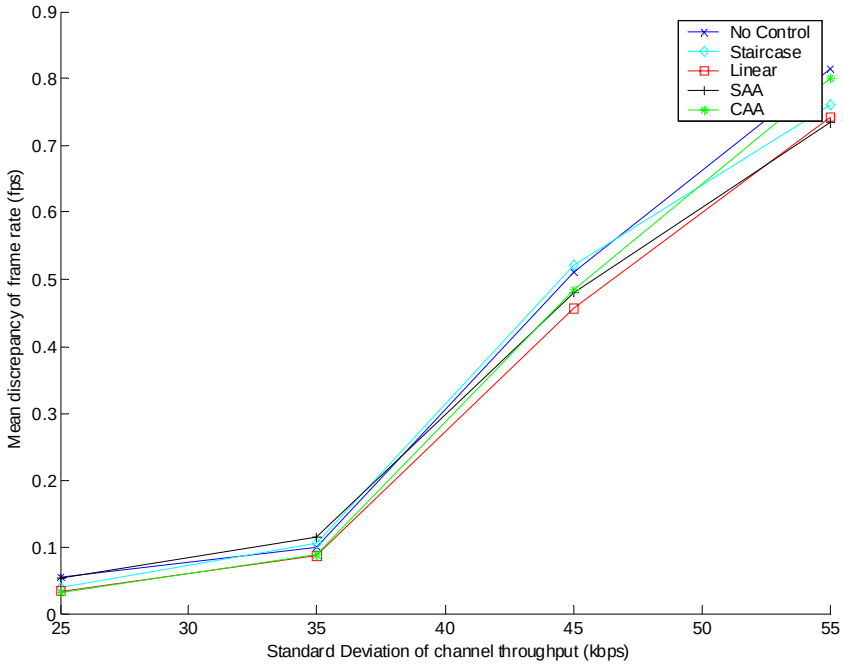
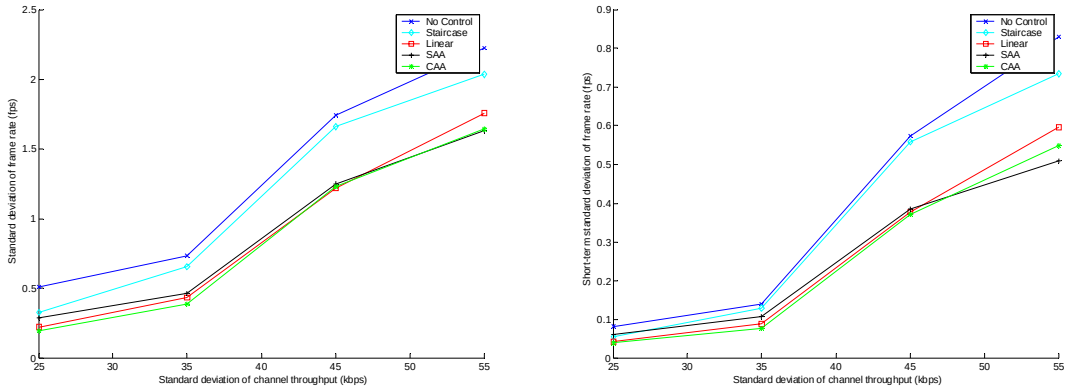



Figure 4-7. Mean discrepancy among various frame-rate adjustment schemes

To compare the performance of control in smooth video playback, we choose the long-term and short-term standard deviation of frame rate as an indication of smoothness. In general, a larger value of the standard deviation implies more motion jitters in the whole playback process, and vice versa. Figure 4-8 depicts the corresponding performance for each control schemes. We can see that the content-aware frame-rate adjustment could provide comparable quality to the sophisticated SAA frame-rate adjustment, because the content-aware scheme could prevent the fullness below some threshold at each control

checkpoint by the use of syntax-level information (size of each video frame). This makes the departure process become a deterministic process, provided that the pre-roll time is greater than zero. This prior knowledge simplifies the process of buffer fullness into a single random process (arrival process), and which is easier to estimate. The accurate estimation to the buffer fullness gives more precise result of the frame-rate adjustment, as Eq. (3-13) represents. Hence, the content-aware control outperforms other control algorithm with a low complexity.




(a) (b)

Figure 4-8. Comparison of (a) long-term and (b) short term standard deviation among various frame-rate adjustment schemes

Figure 4-9 shows the associated number of underflow among these control algorithms. As we can see from the figure, these algorithms resolve the buffer outage successfully, but the associated visual quality differs among these control schemes. The merit of a good frame-rate adjustment scheme lies in its treatment to the potential buffer underflow, since they share an identical threshold value.

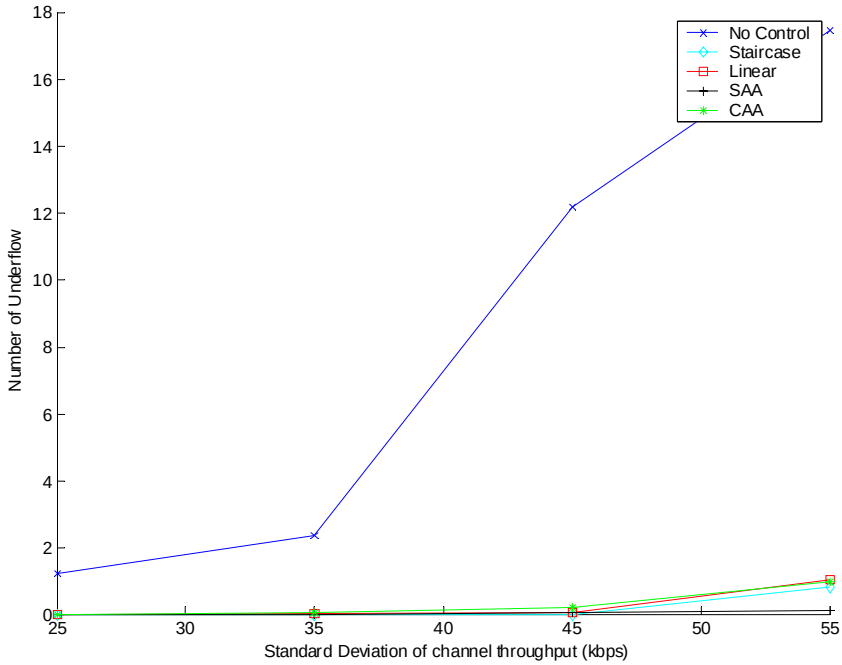



Figure 4-9. Comparison of underflow events among various frame-rate adjustment schemes

### 4.2.2 Performance of Pre-processing and Post-processing Tools

The pre-processing and post-processing tools we mentioned here is the dynamic threshold adjustment and the quality control proposed in Section 3.3.3.1 and 3.3.3.2, respectively. Since we have seen that the content-aware frame-rate adjustment could provide better quality among all the 5 control schemes, we choose the content-aware adjustment as the baseline. Figure 4-10 illustrates the framework of the proposed control mechanism, with highlight of processing tools. Since the primary objective of the proposed control mechanism aims to eliminate buffer outage, we set the outage control as the core of the control mechanism. The term “pre-“ and “post-” are used to modify the additional fine-tuning before and after the outage control.

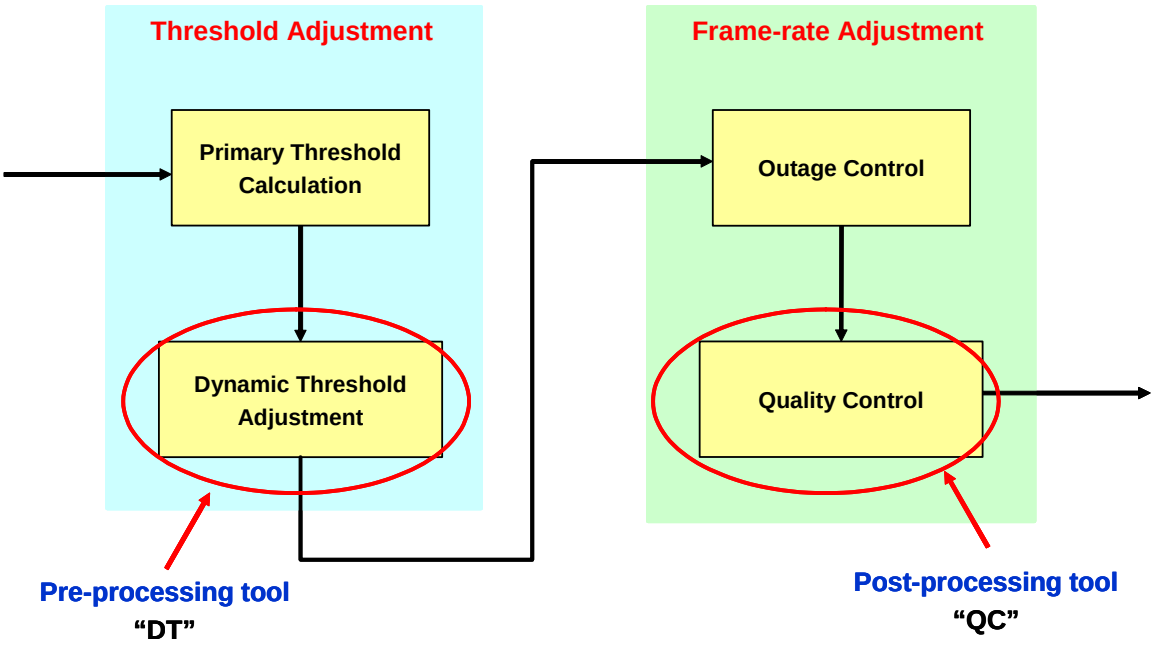



Figure 4-10. The framework of the proposed control mechanism

As to some parameters related to the post-processing tools, we choose the scaling factor k as 0.2, and the size of calculation window of the buffer trend is set to the length of pre-roll time (1 second in this case). Similarly, for the post-processing tool, the maximum allowable value of frame-duration increase (to prevent buffer from going underflow) is set to 40ms, and the value of $S_T$ is set to 0.75. The effects of the pre-processing and post-processing tools are shown in Figure 4-11. There are four symbols defined in Figure 4-11, the "Original" means that both tools are turned off, the "QC Only" means we only turn on the quality control, the "DT Only" stands for the activated dynamic threshold adjustment, and the "DT+QC" represents that both tools are applied.

Figure 4-11 shows the result of mean discrepancy among these four scenarios. Since the proposed control mechanism only reduces the playout speed once the potential buffer underflow is present, the longer duration of control activation would incur more latency. The dynamic threshold adjustment increases the probability of control activation by adjusting the threshold, which is designed for the early prevention of buffer outage. Besides, the quality control increases the duration of control activation due to the consideration of perceptual quality. Hence, we could conclude that both tools would lengthen the overall playback duration, as Figure 4-11 shows. However, both tools start to benefit from its

control features as the variation of channel throughput arises. Larger variation of channel throughput causes more events of buffer underflow. Both the extension of control duration (by QC) and the early activation of control (by DT) could resolve the increase of latency caused by the buffer underflow. In other words, the underflow would dominate the overall latency when the channel variation is large. This could be justified by exploring the corresponding average number of buffer underflow, as Figure 4-12 shows.

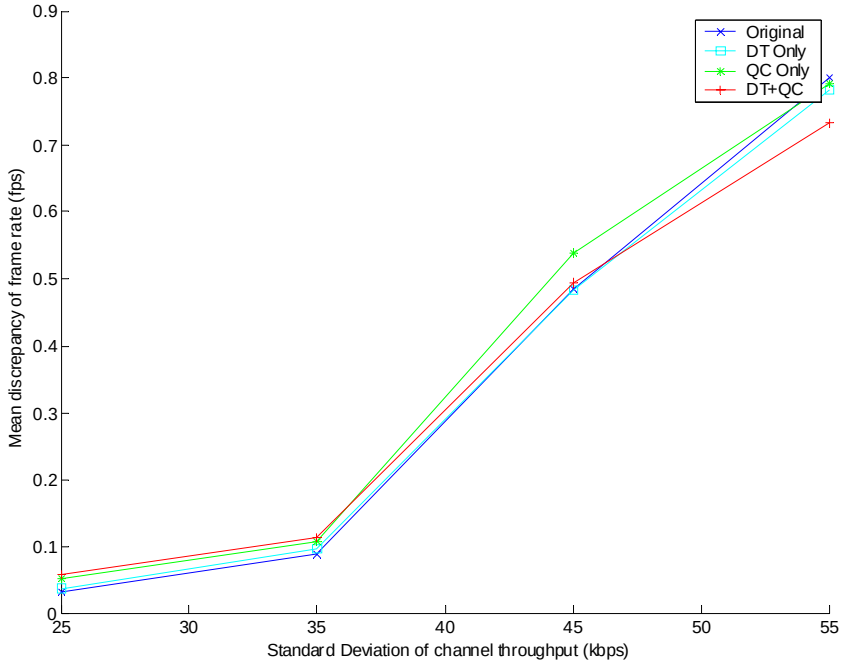



Figure 4-11. Comparison of mean discrepancies among various control scenarios.

As we can see from Figure 4-12, when the channel_STD equals to 55kbps, the original CAA frame-rate adjustment suffers about one buffer underflow in average, while the other three control scenarios incur buffer underflow below 0.5 times per streaming. Thus, the introduced latency from the buffer underflow exceeds the latency caused by the control of processing tools. Note that we could roughly estimate that the introduced latency by the quality control is equivalent to that incurred by 0.5 buffer underflow, for their mean discrepancies of frame rate are almost the same from Figure 4-11 at channel_STD = 55kbps.

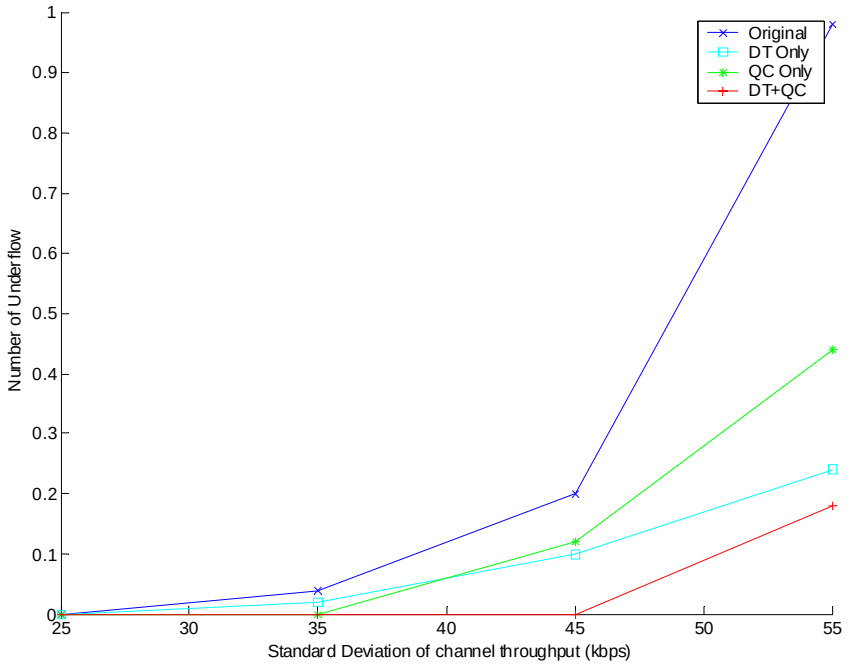



Figure 4-12. Comparison of underflow events among various control scenarios

Figure 4-13 show both the long-term and short-term variation of frame rate. Obviously, the processing tools suffers more quality degradation when the channel_STD is low (25~35kbps), since there are more ineffective control activated when QC or DT is turned on. Strictly speaking, the QC would suffer more quality degradation than DT does, because the DT could be released from the control activation more quickly when the buffer fullness arises. Hence, the discrepancies of long-term variation of frame rate between DT and QC diverge. On the other hand, the short-term variations of frame rate are not far off since the QC operates in a smooth way of frame-rate adjustment. Furthermore, both QC and DT benefits from the increase of channel_STD due to the contribution to the variation from underflow exceeding that from control. Note that the QC-only control scheme outperforms the QC+DT control scheme for the short-term variation of frame rate at channel_STD = 55 kbps, because the frequent on-and-off control activation caused by DT increases the effective short-term variation of frame rate in case of frequent buffer underflow. This represents that the short-term variation of frame rate is actually a suitable metric for measuring the smoothness of perceptual quality.

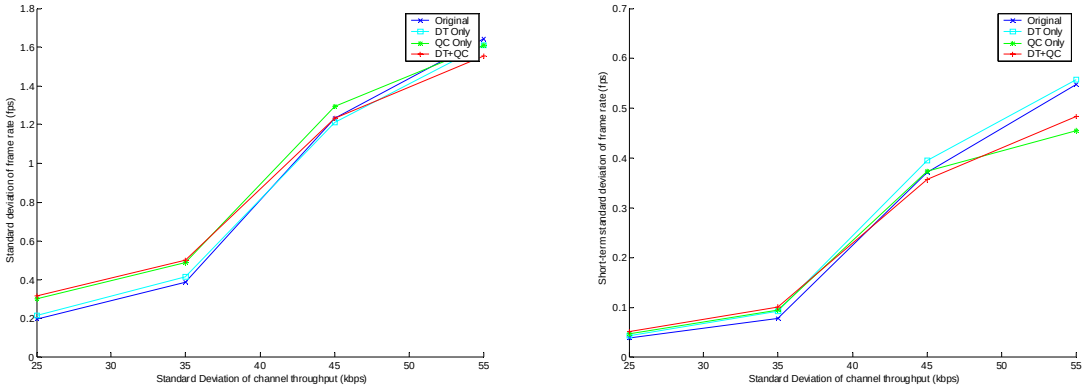



Figure 4-13. Comparison of (a) long-term and (b) short-term standard deviation among various control scenarios

Figure 4-14 visualizes the smooth playback process with the proposed AMP control mechanism with a sampling interval of 200ms. The simulation parameters are identical to that defined in Table 4-1. The early activation of proposed control slows down the playout speed before buffer underflow occurs. However, once the buffer underflow occurs due to the absence of control, the screen freezes for a long duration to reload the video content into the buffer. Conversely, with the presence of proposed control, the playback will continue with a slower speed, which maintains the smoothness of visual quality.

T = 3400ms  13th  17th

T = 3600ms  14th  19th

T = 3800ms  15th  19th

T = 4000ms  16th  19th

T = 4200ms  17th  19th

T = 4400ms  18th  19th

T = 4600ms  19th  19th

T = 4800ms  21st  19th

T = 5000ms  22nd  19th

T = 5200ms  23rd  19th

T = 5400ms  24th  19th

T = 5600ms  25th  21rd

T = 5800ms  27th  23rd

With Control Without Control

Figure 4-14. Snapshots of the playback process for streaming video sequence “container”. The accompanying number is the associated frame number of each video frame.

### 4.2.3 Summary

This section compares the performance of various type of AMP-based control. We compare the four types of frame rate adjustment in term of the three metrics described in Section 3.3.2. The content-aware control benefits from the syntax-level information and gives a comparable quality to the sophisticated SAA scheme. As a result, the content-aware control performs a good balance between the complexity and the control quality. The proposed control could eliminate the event of buffer underflow by setting a proper value of $P_T$ in advance.

Besides, the proposed dynamic threshold adjustment could provide a proper early activation of outage control, with little loss in perceptual quality. In addition, the proposed quality control of the frame rate adjustment could also lower both the long-term and short-term variation of frame rate, especially for the short-term variation of frame rate, and this means that the corresponding control incurs less jitter in the playback process. As a result, the switch of processing tools could depend on the probe of the channel characteristics and the user’s preferences.

# Chapter 5

# Conclusion

## 5.1 Contributions

This thesis integrates the whole architecture of video streaming over cdma2000 1x-RTT system. The use of MPEG-21 Testbed makes the experimental results more realistic for practical system design.

1. Study on the link-layer transmission behavior of video streaming over cdma2000 1x-RTT system, such that the system simulation is more close to a realistic one.
2. Analytical results could be further extended to the cdma2000-based systems such as 1xEV-DV or 1x-EV-DO, since they share similar framing structures.
3. The proposed control algorithm considers both sides of the control performance and the associated visual quality, and it is shown from the experimental results that the proposed control algorithm outperforms than other similar algorithms.

## 5.2 Future Works

The assumption on source rate control and radio resource assignment may be impractical for a realistic system. The source rate control could be revised by introducing the Internet delay and channel model to form a streaming traffic model in the base station. Besides, the air-link rate assignment should also take the radio resource management into consideration, especially for a mixed-traffic system. In addition, the impact of base station queue should be included such that the transmission behavior meets the practical scenarios. Moreover, the RLP frame error and the delay caused by RLP frame retransmission should be identified, too. With the presence of error, the impact of error-processing behavior of the decoder on the perceptual quality could be quantified at the same time. Eventually, all these studies could be applied to the future system to perform better control in the MAC layer while optimizing the video streaming application.

# References


[1] I. Elsen, F. Hartung, U. Horn, M. Kampmann, L. Peters, “Streaming Technology in 3G Mobile Communication Systems,” IEEE Computer, pp. 46 – 52, September 2001.

[2] 3GPP TS 23.107 v6.0.0, “Quality of Service (QoS) concept and architecture,” Dec. 2003.

[3] D. Wu, T. Hou, Y.-Q. Zhang, "Transporting Real-time Video over the Internet: Challenges and Approaches," Proceedings of the IEEE, vol. 88, no. 12, pp. 1855 -1875, Dec. 2000.

[4] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460, Dec. 1998.

[5] Y.-K. Lim, D. Singer, “ISO/IEC 14496-8 FDIS -- Carriage of ISO/IEC 14496 contents over IP networks,” MPEG02/N4712, Mar. 2002.

[6] D. Wu et al., “Streaming Video over the Internet: Approaches and Directions,” IEEE Transaction on CSVT, vol. 11, no. 1, Feb. 2001.

[7] D. Mills, “Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI,” RFC 2030, Oct. 1996.

[8] C.-N. Wang et al., “Scalable Multimedia Streaming Test Bed for Media Coding and Testing in Streaming Environments,” ISO/IEC SC29/WG11 M10298, Dec.2003.

[9] T. Stockhammer, H. Jenkač, and G. Kuhn, “Streaming video over variable bit-rate wireless channels,” IEEE Transaction on Multimedia, vol. 6, No. 2, April 2004.

[10] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, “Advances in channel-adaptive video streaming,” IEEE ICIP 2002, Rochester, NY, Sept. 2002.

[11] D. Quaglia and J. C. De Martin, “Adaptive packet classification for constant perceptual quality of service delivery of video streams over time-varying networks,” IEEE ICME 2003, Baltimore, Maryland, July 2003.

[12] N. Färber and B. Girod, “Robust H.263 compatible video transmission for mobile access to video servers,” Proc. IEEE International Conference on Image Processing, ICIP’97; 2:73-76; Santa Barbara, CA, USA.

[13] R. Kurceren and M. Karczewicz, "A Proposal for SP-frames," ITU-T SG 16 Doc.

VCEG-L27, Eibsee, Germany, Jan. 2001.

[14] ISO/IEC 13818-2, “Generic coding of moving pictures and associated audio,” recommendation H.262, March 1994.

[15] ISO/IEC 14496-2, “Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile,” ISO/IEC 14496-2/FPDAM4, July 2000.

[16] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, IEEE Transactions on CSVT., vol. 11, no.3 pp.301-317, March 2001.

[17] J. G. Apostolopoulos, “Reliable video communication over lossy packet networks using multiple state encoding and path diversity,” Visual communication and image processing (VCIP), Jan. 2001.

[18] J. Chakareski, S. Han, and B. Girod, “Layered coding vs. multiple descriptions for video streaming over multiple paths,” proceedings of the 11th ACM international conference on multimedia, pp.422-431, 2003

[19] W. Li, “Fine Granularity Scalability in MPEG-4 for streaming video,” IEEE ISCAS 2000, Geneva, Switzerland, pp299-302, May 28-31, 2000.

[20] ISO/IEC JTC1/SC29/WG11, “Call for Proposals on Scalable Video Coding Technology,” MPEG2003/N6193, Dec. 2003.

[21] M. Kalman, E. Steinbach, and B. Girod, “Adaptive playout for real-time media streaming,” IEEE ISCAS 2002, Scottsdale Arizona, May 2002.

[22] M. C. Yuang, S. T. Liang, and Y. G. Chen, “Dynamic video playout smoothing method for multimedia application,” Proc. IEEE ICC 1996, Dallas, Texas, June 1996.

[23] N. Laoutaris and I. Stavrakakis, “Adaptive playout strategies for packet video receivers with finite buffer capacity,” IEEE ICC 2001, Helsinki, Finland, June 2001.

[24] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay streaming over error-prone channels,” IEEE Transaction on CSVT, vol. 14, No. 6, pp. 841-851, June 2004.

[25] C. H. Liang and C. L. Huang, “Content-based adaptive media player for network video,” IEEE ISCAS 2004, Vancouver, Canada, May 2004.

[26] J. Huang et al., “Performance of a mixed-traffic cdma2000 wireless network with scalable streaming video,” IEEE Transaction on CSVT, vol. 13, issue 10, pp. 973-981, Oct. 2003.

[27] Y. Kikuchi et al., “RTP payload format for MPEG-4 audio/visual streams,” RFC 3016,

Nov. 2000.

[28] M. van der Schaar and H. Radha, “A hybrid temporal-SNR fine-granularity scalability for Internet video,” IEEE Transactions on CSVT, vol.11, no.3, Mar. 2001.

[29] H. Schulzrinne et al., “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550, Jul. 2003.

[30] M. Carson and D. Santay, “NIST Net: a Linux-based network emulation tool,” ACM SIGCOMM Computer Communication Review, vol. 33, issue 3, pp 111-124, July 2003.

[31] TIA TR 45.5, “The cdma2000 ITU-R RTT Candidate Submission,” v0.18, July 1998.

[32] ISO/IEC 14496-2, “Coding of Audio-Visual Objects, Part-2 Visual, Amendment 1: Visual Extension,” ISO/IEC 14496-2/AMD1, July 2000

[33] H. C. Chuang, C. Y. Hunag, and T. Chiang, ”On the buffer dynamics of scalable video streaming over wireless network,” IEEE VTC 2004-Fall, Los Angels, CA, Sept. 2004.

[34] http://www.nokia.com/nokia/0,8764,2275,00.html

[35] C. Bormann et al., “Robust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed,” RFC 3095, July 2001.

[36] V. Varsa and I. Curcio, “Transparent end-to-end packet switched streaming service (PSS); RTP usage model (Release 6), “ 3GPP TR 26.937 V6.0.0, March 2004.

[37] Anthony Vetro and Christian Timmerer, “ISO/IEC 21000-7 FCD, Digital Item Adaptation,” MPEG03/N5845, July 2003.

[38] ITU-T “Subjective Video Quality Assessment Methods for Multimedia Applications,” ITU-T Recommendation P 910, Aug. 1996.

# 自　　傳

莊孝強：西元 1979 年生於基隆市。2002 年畢業於台灣新竹的國立交通大學電子工程學系，之後進入該校電子工程研究所攻讀碩士學位。以無線網路視訊串流系統為論文研究主題。

Hsiao-Chiang Chuang was born in Kee-Lung in 1979. He received the B.S. degree in Electronics Engineering Department of National Chiao-Tung University (NCTU), HsinChu, Taiwan in 2002. His current research interests are wireless video streaming system.