ACROSS-PEER RATE ALLOCATION ALGORITHM IN PEER-TO-PEER NETWORKS

A Thesis

by

YANG SU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Mi Lu
Committee Members,	Henry D. Pfister
	Jennifer Welch
	Jiang Hu
Head of Department,	Chanan Singh

August 2013

Major Subject: Computer Engineering

ABSTRACT

We introduce a new across-peer rate allocation algorithm with successive refinement to improve the video transmission performance in P2P networks, based on the combination of multiple description coding and network coding. Successive refinement is implemented through layered multiple description codes. The algorithm is developed to maximize the expected video quality at the receivers by partitioning video bitstream into different descriptions depending on different bandwidth conditions of each peer. Adaptive rate partition adjustment is applied to ensure the real reflection of the packet drop rate in the network. Also the granularity is changed to the scale of atomic blocks instead of stream rates in prior works. Through simulation results we show that the algorithm outperforms prior algorithms in terms of video playback quality at the peer ends, and helps the system adjust better to the peer dynamics.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Lu, and my committee members, Dr. Pfister, Dr. Welch, and Dr. Lu, for their guidance and continuous support throughout the course of this research. Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience. I also want to extend my gratitude to my officemates, Leyi Yin, and Yoon Seak Yang, who have been my constant source of support and inspiration. In addition, I would like to thank all the members in Computer Engineering Group, who make me feel we are like a big family. Finally, thanks to my mother and father for their unconditional love and being very supportive of everything I do.

TABLE OF CONTENTS

Page

ABSTRACT ii

ACKNOWLEDGEMENTS iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

CHAPTER I INTRODUCTION 1

I.A. Motivation 1
I.B. Related Works 2
I.C. Introduction to Network Coding 4
I.D Introduction to Multiple Description Coding 6
I.E. Thesis Outline 7

CHAPTER II P2P VIDEO STREAM FRAMEWORK 8

II.A. Presumptions and P2P Implementation 8
II.B. Reed-Solomon Codes 9
II.C. Basic Rate Allocation Algorithm 9
II.D. Rate Allocation across Different Peers 12
II.E. Exponential Refinement on Video Sources 16

CHAPTER III INTEGRATION WITH NETWORK CODES 18

CHAPTER IV THEORETICAL ANALYSIS 20

IV.A. Performance Metrics 20
IV.B. Rate-Distortion Analysis 20

Page

CHAPTER V NS-2 SIMULATIONS AND RESULTS 24

V.A. Simulation Tool Setup 24
V.B. Simulation Environment Introduction 24
V.C. PSNR Simulation Results 27
V.D. Continuity Simulation Results 29

CHAPTER VI CONCLUSIONS AND FUTURE WORKS 32

REFERENCES 34

APPENDIX A 38

LIST OF FIGURES

FIGURE Page

1 Illustration on Network Codes 5

2 Bisection method to eliminate the Lagrange Multiplier.......... 11

3 Illustration on the across-peer rate allocation algorithm.......... 14

4 Test environment.......... 27

5 PSNR for different video bit rates
in Traditional scheme and our scheme.......... 28

6 PSNR for different video bit rates
in Mesh-based scheme and our scheme 29

7 Continuity for different video bit rates
in Traditional scheme and our scheme.......... 30

8 Continuity for different video bit rates
in Mesh-based scheme and our scheme 31

9 Adjust the probability profile 38

CHAPTER I

INTRODUCTION

I.A. Motivation

With rapid growth of Internet video services, there is an increasing demand for more reliable and faster video transmission through the networks. Service providers favor decentralized systems such as large content distribution, nowadays, and peer-to-peer networks arise to reduce the burdens of systems. However, some other issues come to emerge as well such as instability caused by rapid peer dynamics. In order to cope with the instability issue in P2P networks, multiple description coding can replace the retransmission scheme, which can result in heavy network congestion, to enhance the robustness of transmission. Another concern in transmitting packets through data networks is efficiency, which can be measured in the throughput of the network. Network coding can be adopted to improve throughput albeit sacrificing the computation resource and complexity.

Live video transmission has some unique characteristics, which other types of contents such as image and text do not have. It is required to be instantaneous in most cases, and have a large size, thus taking up larger bandwidth. These properties entail some special techniques for video transmission. Our work is inspired from [5] in which the paradigm of multiple description codes along with practical network coding was brought up for video multicast in lossless networks.

We extend the generic model [5] on combination of Multiple Description Codes and Network Codes to be applied in peer-to-peer networks. We propose the new rate allocation method in generating the multiple descriptions, in order to accommodate different network conditions of each user in peer-to-peer networks. We also adopt hierarchical network codes [2] to ensure the different priorities among different descriptions. By incorporating the new rate-allocation method and integrating multiple description codes with hierarchical network codes, the new scheme has been proved to improve the video playback performance through both theoretical analysis and simulation results.

I.B. Related Works

Network coding has been introduced to peer-to-peer networks since 2005. Network coding replaces the traditional block-exchanging scheme on the file sharing systems, which was mentioned in [7]. The basic idea is that the time required to distribute a file to all the peers can be shortened if all the users can linearly combine all the blocks received, and generate new blocks with random coding coefficients, thus eliminating the periodic block bitmap exchanges. However, it was argued that the computation costs might be increased since encoding and decoding of network codes involves in more processor utilization, memory usage, and disk access. As a result, the concerns for the plausible implementation attract more attention. To resolve the computation complexity issue, it was proposed [16] that the file can be divided into multiple generations, and network coding is performed only within each generation. Although this reduces the computation cost issue, the periodic information exchanges are still needed since network coding is performed only within each generation. At the same time, BitTorrent is a system designed to download the scarcest segment first by periodic segment exchanges, so that the balance of the system can be maintained. Because the computation complexity remains an issue to be resolved, network codes are not widely applied in the file sharing systems.

When it comes to the media distribution systems, different problems and concerns such as latency, arise in contrast to file sharing systems. Especially when the live streaming systems are used, all the users in the system need to be synchronized with respect to the video playback buffers. Literally, the segment availability bitmaps need to be exchanged with other peers periodically since the sliding window of buffer advances over time. Whether to reduce the buffer exchange information became an interesting topic in the research circle, and Wang et al [12] proposed a new live media streaming system named R^2, in which the random network coding is applied to enhance the performance. R^2 also adopts the similar method mentioned above: the content of the media is split into multiple generations, and network coding is performed within each generation. As a result, the communication overhead can be reduced because information is exchanged among peers at the granularity of a generation instead of a single segment.

Owing to the fact that latency requirements are essential in the live streaming systems, R^2 adopts the method that each peer randomly pushes the coded blocks within the same generation into the downstream peers until all of them receive complete information. Therefore, this avoids requesting explicit requests for the missing blocks. Through theoretical analysis, R^2 has been proved [13] to outperform other systems without network coding, especially when there is a large influx of users into the system within the short time.

I.C. Introduction to Network Coding

R. Ahlswede et al proposed the network coding mechanism in 2000 ([1]), which takes advantage of the network topology to enhance the throughput in the networks. It enables intermediate nodes to combine and encode the upcoming streams in the network, instead of only forwarding the packets. Therefore, each transmitted packet can be considered as a linear combination of all the packets that are available for the node. Furthermore, those encoded packets can be recombined by another node in the network to generate the new linear combinations. As a result, each receiver can recover the original information as long as it receives enough linearly independent packets ([3]).

A simple example can help illustrate the fundamental idea of network coding in Figure 1. Suppose that there are two sending nodes, which are S1 and S2, four intermediate nodes (A, B, D1 and D2), and two receiving nodes (T1 and T2) in the graph. The capacity of all the links is one unit. S1 wants to send the packet labeled "a" to both receivers, while S2 wants to send the packet labeled "b" to both receivers. Without network codes, there would be a conflict on the link A-to-B since only one packet can be transmitted through that link at a time. However, if the node A can perform the XOR operation on the packets it receives, and send the encoded packet "a XOR b" to the node B, both receivers can recover both "a" and "b" successfully.

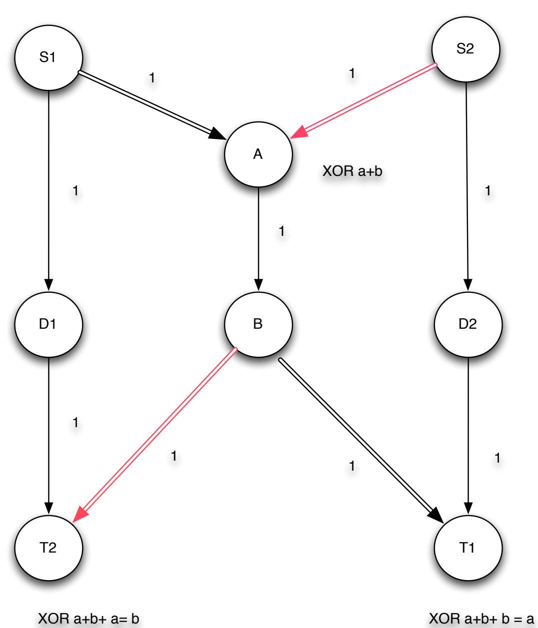

Figure 1: Illustration on Network Codes

In 2003, Ho et al [15] have further proposed the concept of random network coding, where a network code transmits on each of its outgoing links a linear combination of incoming packets over a finite field, with randomly chosen coding coefficients. In [6], the scheme using hierarchical network coding together with layered video coding is proved to improve the performance in P2P networks by adapting to the available bandwidth. However, it might lack flexibility when different users have different video quality requirements.

I.D. Introduction to Multiple Description Coding

To overcome the inevitable packet losses during the transmission in the networks, multiple description coding was invented at Bell Laboratories [3] to gain robustness of transmission by sacrificing some compression efficiency. Therefore, there is a trade-off between transmission failure rate and compression efficiency. In order to apply Multiple Description coding to networks, a correspondence between each description must exist to guarantee the transmission robustness between senders and receivers. The following conditions must be satisfied to make full use of MD coding: one or more users sometimes fail to receive one or more descriptions, and various quality levels are acceptable and distinguishable [4]. When it comes to video delivery, MDC becomes very promising for its stringent delay requirement, and this is also the reason why the multiple description transmission performs better than retransmission scheme if long delays are unacceptable [10]. A. Reibman et al [9] proposed a MD video splitting method based on the rate-distortion.

I.E. Thesis Outline

The rest of the thesis is organized as follows. Chapter II includes the complete framework of the new designed peer-to-peer system with the novel rate allocation algorithm for multiple description codes. Chapter III integrates the rate allocation algorithm with the network codes. Chapter IV covers the theoretical analysis of the performance of the system, and the metrics used to evaluate the performance. Chapter V discusses the simulation environment, and simulation results. Chapter VI concludes the thesis, and contains the conclusions together with future works.

CHAPTER II

P2P VIDEO STREAM FRAMEWORK

II.A. Presumptions and P2P Implementation

P2P networks can be implemented with two strategies: push methods, and pull methods [8]. Push methods construct multicast trees to connect all the peers, while pull methods build a directed graph to interconnect all the users to exchange the availability bitmaps periodically. Each of them has both advantages and disadvantages. Push methods can guarantee faster transmission speeds, but they are very vulnerable to sudden departure of active users. Pull methods are more impervious to peer dynamics, and easier for implementation, but can incur more latency during transmission because of updating the segment availability bitmaps.

Our system will adopt a scheme based on the combination of two methods mentioned above: push methods based on directed graph structures. Originally, push methods were used in the tree structures, while pull methods were used in the directed graph structures. The system pushes the packets from the upstream nodes to downstream nodes, after obtaining the accurate information on the missing segments of all the downstream peers.

II.B. Reed-Solomon Codes

Reed–Solomon (RS) codes [17] are a subset of BCH codes in the family of block codes. Owing to the excellent distance property, Reed-Solomon codes have been widely applied in so many different areas such as digital audio compression. Reed-Solomon codes can be defined as the q^m-ary BCH code of length $q^m - 1$. To construct a RS code of length $q^m - 1$ with the ability to correct t errors, we first have to find the primitive ($q^m - 1$)st root in Galois Field GF(q^m), and then we construct the cyclotomic cosets modulo $q^m - 1$ in GF(q^m). Because RS codes belong to the maximum-distance separable codes, an (n, k) Reed-Solomon code has minimum distance (n-k+1). To decode RS codes, there are many algorithms such as the Peterson algorithm, the Berlekamp-Massey Algorithm, and the Euclid's Algorithm.

II.C. Basic Rate Allocation Algorithm

The client sends a request, which includes both the rate constraint information and the requested video title to the server, and the server checks which sources possess the relevant video information, and decide which nodes to send packets to the client. Considering the rate constraint, the server also decides how many different descriptions are sent through the network. If the rate is low, few descriptions or even only one description is sent. Otherwise, more descriptions are sent through the network.

We model our peer-to-peer network as a directed acyclic graph G = (V, E, C), where V is the set of nodes, E is the set of directed link, and C is the set that contains the capacity associated with each link. The set of sources is denoted as {1, 2, … S}, and the set of clients is denoted as {1, 2, ... T}. We let p_{ij} denote the probability of receiving j packets out of M packets for the client numbered i. We here distinguish different clients for the reason that different clients might have different network conditions. We define R_j as the bit rate of the first j packets, and $D(R_j)$ as the rate-distortion function with respect to R_j. Therefore, we define the average distortion E(D) as

$$E(D) = \frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{M} p_{ij} D(R_j) \qquad (1)$$

From the observation of Eq(1), we can see that the formula resembles the expected distortion function in [7]. The rate allocation problem can be solved with typical convex approximation approach. Obviously, this problem is also subject to several constraints, which are shown as follows:

$$R_1 \leq R_2 \cdots \leq R_{M-1} \leq R_M \qquad (2)$$

$$\sum_{i=1}^{M} \frac{M}{i(i+1)} R_i \leq \sum_{i=1}^{S} R_i^* \qquad (3)$$

where R_j^* represents the total outgoing bandwidth of the source j. Because of the P2P system property that video data can be retrieved from different nodes, Eq(3) indicates that each peer must satisfy the bandwidth budget constraint respectively. All the outgoing rates from different sources still sum up to:

$$R_{total} = MR_1 + \frac{M}{2}(R_2 - R_1) + \cdots + \frac{M}{M}(R_M - R_{M-1}) \qquad (4)$$

As a result, we can follow the same procedure stated in [11] to get optimal rate allocation solution using Lagrange Multipliers method illustrated in Figure 2.

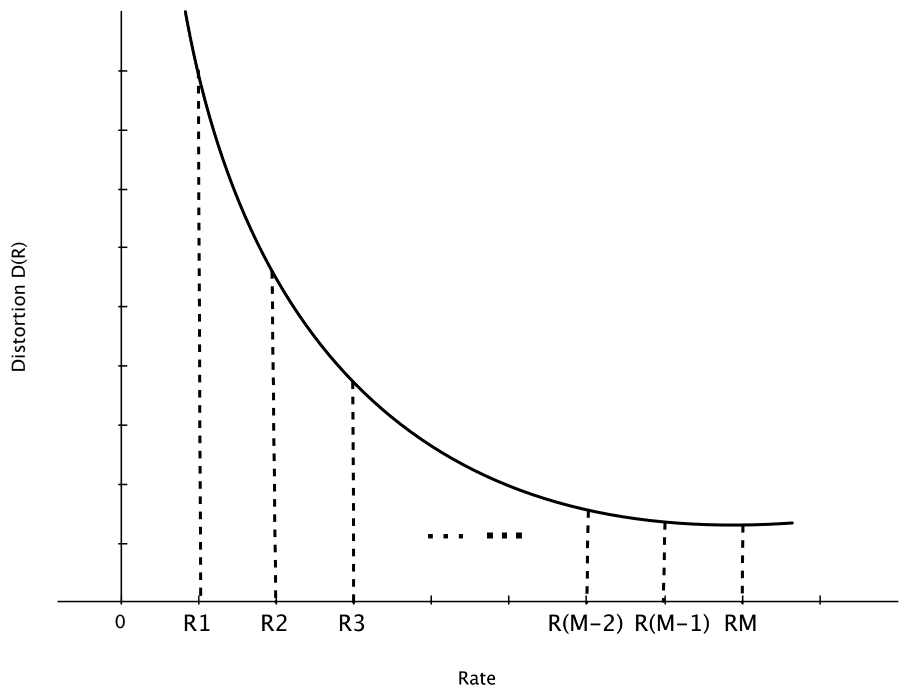

Figure 2: Bisection method to eliminate the Lagrange Multiplier

First, we define the Lagrange function as

$$L(R_1, R_2, \cdots R_M, \lambda) = \sum_{i=1}^{T}\sum_{j=1}^{M} p_{ij} D(R_j) + \lambda(R_{total} - \sum_{i=1}^{S} R_i^*) \qquad (5)$$

And in Eq(5), λ is the Lagrange multiplier. Then, we can get the partial derivative as to R_j, and make all the derivatives equal to zero:

$$\sum_{i=1}^{n} p_{ij} \frac{\partial D(R_j)}{\partial R_j} + \lambda \frac{\partial R_{total}}{\partial R_j} = 0 \tag{6}$$

Since we model the video as the CBR (Constant Bit Rate) source, we consider that each packet has the exactly same length L. Next we consider two extreme cases: If there were no limit for bandwidth budget, the rate allocation would be $R_1 = R_2 = \cdots = R_M$, ending up with maximizing the total rate, and λ would be zero; in contrast, if the bandwidth budget is so little that can be negligible, the rate allocation would be $R_1 = R_2 - L = \cdots = R_M - (M-1)L$, and λ would be a value larger than zero. Given the bandwidth budget, which definitely falls in between those two extreme cases, we can apply the bisection search method to find λ to match the total bandwidth budget.

II.D. Rate Allocation across Different Peers

The problem that remains to be solved is how to determine the rate allocation among different upstream peers. Because each upstream peer might have different bandwidth conditions, we have to periodically monitor the network condition for each peer, so that we can optimize the rate allocation across all of them. We apply the round-robin polling method to inquire about the updated bandwidth condition for each upstream peer, collect all the current bandwidth information. Within the server, we sum up all the bandwidth capacities currently possessed by each upstream peer, and name the summed capacity the system bandwidth capacity. Meanwhile, we maintain a max heap data structure in the server to serve as the repertoire of all the upstream nodes, which are sorted by the bandwidth capacity. Then we can feed the total bandwidth budget to the Lagrange Multipliers method in order to generate the optimal rate allocation from the perspective of the whole system.

Next, the server notifies each upstream peer about both the system-level rate allocation and the rate task each node should be in charge of. In order to determine the rate task for each independent upstream peer, we leverage the in-built extract-max() function of the heap structure to retrieve the node with the largest bandwidth in the heap from the time being, and to remove the node from the heap. After extracting the node, we get the link capacity the node can have at most, and assign the corresponding rate task that matches the capacity. Finally, we take a rate cursor to keep track of the current rate position, in order to avoid rate overlapping among different stream providers. Each time we allocate a rate task to a peer, the rate cursor is advanced on the scale by the amount of the rate task assigned to the peer. If the rate cursor exceeds the range of the total rate (or equal to the total rate budget), we exit from the algorithm, indicating the rate allocation is consummated (See in Figure 3).

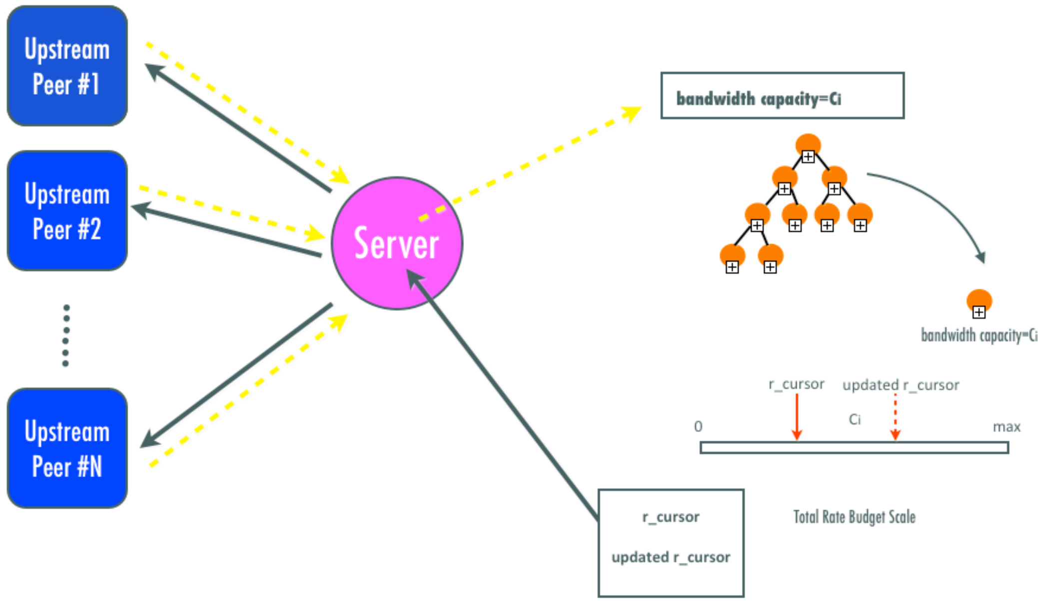

Figure 3: Illustration on the across-peer rate allocation algorithm

The complete algorithm is shown as below:

While *not at end of the video file* **do**

Server X sends polling requests to each node in the source Set {1,2,... S};

Each node S_i *in the source responds to the server with the feedback of the current link capacity* C_i*;*

Server X sums up all the link capacity $C = C_1 + C_2 + \cdots + C_S$*;*

Server X maintains the max-heap $H_{\max}$*;*

Server X performs the overall rate allocation (R1, R2, ... Rm) based on the Lagrange Multipliers method;

Initialize and reset the rate pivot r_curr = 0;

for *each_node i* **in** *Hmax* **do**

assign the rate task $r\_aug = \min(C_i, R_m - r\_curr)$ *to node i*

update r_curr to r_curr = r_curr + r_aug;

if r_curr > R_m*:*

r_curr = 0;

end

end

end

II.E. Exponential Refinement on Video Sources

Before we introduce the network codes into the system, there are still several issues to be resolved. In the algorithm mentioned above, we idealize the situation that each peer bandwidth perfectly matches the assigned task rate, which is not likely to happen in reality since application-level packetization requires a constant length of packet, which is, in most cases, not consistent with the given task rate. Meanwhile, in order to be compatible with network codes, we have to decide the fundamental coded block, which is the indivisible unit used to perform network codes. In our context, we name them the atomic block.

In order to tackle those issues, we have to refine the rate-partitioning scheme. Previously, we perform successive refinement with additive increase on each description. Instead, we perform successive refinement with exponential increase. In this case, the second layer remains divided to two equal parts, while the third layer is divided to four equal parts, instead of three parts. If we have M partitions in total, the last layer will be divided into 2^{M-1} parts. Each part in the last layer can be considered to be the atomic block. Therefore, the packetization problem is resolved, and atomic blocks are generated to be applied in network codes. Another significant improvement with exponential-increase refinement is that the complexity of Reed-Solomon codes is reduced, because only repetition codes are needed to generate replicas of the original parts for the fact that the number of divided parts in each layer is the power of 2.

By incorporating this new refinement method, we revise the previous across-peer rate allocation algorithm to the following one based on the atomic blocks:

while *:*

Server X sends polling requests to each node in the source Set {1,2,... S};

Each node Si in the source responds to the server with the feedback of the current link capacity C_i*;*

Convert each link capacity C_i *to the number of atomic blocks* a_i*;*

Server X sums up all the numbers of atomic blocks $A = a_1 + a_2 + \cdots + a_S$*;*

Server X maintains the max-heap $H_{\max}$*;*

Server X performs the overall rate allocation (R1, R2, ... Rm);

Convert rate allocation into block allocation (A1, A2, ... Am);

Set the block pivot cursor a_curr = 0;

for *each_node i in Hmax:*

assign the rate task $r\_curr = a_i L$*;*

update a_curr: $a\_curr = a\_curr + a_i$*;*

if a_curr > A:

break*;*

end

end

end

CHAPTER III

INTEGRATION WITH NETWORK CODES

The rate allocation strategy has to accommodate the issues caused by network codes, if network codes are introduced to the peer-to-peer systems. In the prior work [5], random network coding is used to generate M linear combinations of k linearly independent packets as $y_j = \sum_{i=1}^{M} f_{ij} x_i$, where f_{ij} coefficients are chosen from the Galois Field GF (2^q). Fixing up the appropriate size of the Galois Field can minimize the probability of obtaining linearly dependent combinations at the clients. However, when it applies to the lossy networks, loss of packets can aggravate the problem, increasing the failure rate that the client is unable to get enough linearly independent combinations. To alleviate the situation caused by network congestion and link failures, we use hierarchical network codes (HNC) instead. The typical scenario where hierarchical network codes are applied is that the source has a scalable video encoder, and can produce a base layer and several enhancement layers. To be applicable in our situation, the layers are labeled from 1 to M, with 1 being the most important and M being the least important. As a result, data from the 1st layer can be recovered with the highest probability at the client while the data from the M th layer can be successfully obtained with the lowest probability. To be compatible with hierarchical network coding, the rate allocation strategy has to satisfy another requirement that a RM -bit chunk should be partitioned into sections of equal size *k*, where the following equation should be met:

$$k = \gcd(R_1, R_2 - R_1, \cdots, R_M - R_{M-1}) \qquad (6)$$

As mentioned above, we use the term atomic block size to refer to the smallest coding unit in the scheme. The trivial value of k is 1 byte, and obviously it is not a plausible option. Considering that the encoding coefficients are included in the header, k should be neither too small nor too large for the purpose of efficiency issues, which will be discussed in the next chapter.

Then we can apply the hierarchical network coding of the encoded packets. Let $c_j = \frac{R_j}{k}$, and we can generate t packets with randomly generated coefficients:

$$\begin{cases} N_1 = f_1^1 x_1 + \cdots + f_{c_1}^1 x_{c_1} \\ N_2 = f_1^2 x_1 + \cdots + f_{c_1}^2 x_{c_1} + \cdots + f_{c_2}^2 x_{c_2} \\ \vdots \\ N_M = f_1^M x_1 + \cdots + f_{c_1}^M x_{c_1} + \cdots + \cdots + f_{c_M}^M x_{c_M} \end{cases} \qquad (7)$$

where f_i^j coefficients are randomly chosen from the non-zero elements of GF(2^q). As a result of this structure, packets in the first layer have larger possibility to be recovered than those in other layers, because we are able to decode data of the first layer by receiving c_1 linearly independent packets of N_1 type, while we only able to decode data of the second layer by receiving at least $c_1 + c_2$ linearly independent packets of either N_1 or N_2 type.

CHAPTER IV

THEORETICAL ANALYSIS

IV.A. Performance Metrics

We use the Peak-Signal-Noise Ratio (PSNR) to analyze the performance of the system, and it is commonly used to measure of quality of reconstruction of lossy compression codecs, and it is an approximation of human perception of video reconstruction quality.

Owing to the real-time property of streaming videos, we also introduce another measure, continuity [14], to evaluate the consistency of the video transmission. Y. Zhou, et al first brought up continuity in 2007 and it reflects the probability of the continuous playback.

IV.B. Rate-Distortion Analysis

If we assume that the rate of the failure caused by random network coding is so small that we can ignore this type of failure, we can bring up the following rate allocation strategy adjusted for the network-coding scenario. If we still define p_{ij} as the possibility that the client numbered i receives j out of M packets and assume that every packet has the same probability to fail to reach the client, the coefficients before the rate-distortion function D(R_j) should be changed from p_{ij} to $\hat{p}_{ij}$, which can be formulated as:

$$p_{ij} = p_{ij} + p_{i(j+1)}[1-(\frac{M}{j+1})^{1-c_{j+1}+c_j}] \qquad (8)$$

The reason why we have to adjust the possibility is that data in each layer have been encoded into $(c_{j+1} - c_j)$ packets, and only when we receive the correct combination of those packets can we decode into the original data, otherwise it is still considered to be the failure of decoding. Because of the property of Reed-Solomon codes, we can recover the *i* equal parts out of *n* chunks using parameter *(n, i, n-i+1).*

When compared with p_{ij}, $\hat{p}_{ij}$ is larger in value, which indicates that the low-rate distortion is more likely to happen than high-rate distortion. For now, we assume that the packet size and throughput have no influence on the loss rate of the network. However, when we take those factors into consideration, the correction coefficient $\alpha_j(k)$, which depends on k, should be placed in front of the original p_{ij} to adjust those probabilities. Therefore, the new expressions should be as follows:

$$p_{ij} = \alpha_j p_{ij} + \alpha_{(j+1)} p_{i(j+1)}[1-(\frac{M}{j+1})^{1-c_{j+1}+c_j}] \qquad (9)$$

If we denote the original probability mass function for each client as the column vector p^*, the adjusted probability mass function as the column vector p, and the part $[1-(\frac{M}{j+1})^{1-c_{j+1}+c_j}]$ in the expression above as $\beta_{(j+1)}$, we can rewrite it with matrix representation:

$$p^* = \begin{pmatrix} \alpha_1 & \alpha\beta & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \ddots & \ddots & & \vdots \\ & 0 & \ddots & \alpha_i\beta_i & \ddots & \\ \vdots & & \ddots & & \ddots & 0 \\ & & & \ddots & \ddots & \alpha_M\beta_M \\ 0 & & \cdots & & 0 & \alpha_M \end{pmatrix} p \qquad (10)$$

In order to achieve a better performance, we should reduce the values on the left upper part of the matrix while increasing the values on the right lower part.

First, we should select an appropriate value for the packet size k, which can determine the correction coefficients. As we mentioned above, we average packet loss rate is relevant to the packet size. In our system, we do not take into consideration the packet fragmentation and reassembly mechanisms below the application layer, and we assume that the lower layers can self-recover and restore the original packets during network transmission owing to independency between each network layer. As a result, we only have to decide how many descriptions the source encoder is about to generate, so that the atomic block size is determined as well. There is a tradeoff between complexity and performance: if we generate too many descriptions even though the network environment allows so, the computation complexity will be increased a lot; if we generate fewer descriptions, the flexibility to adjust to network condition will go down, thus degrading the performance. From empirical evidences, we usually choose the number of descriptions to be 4 or 8, by satisfying our additional constraint that it has to be a power of 2.

Next, we should guarantee that the rate allocation algorithm works in a fashion that each description is placed in the rate streaming in a sequential order among different peers, so that the maximum information can be transferred to the destination peers in the very poor network condition. By doing so, the entropy of information transmitted can be increased since distinct information comes before the repeated information.

CHAPTER V

NS-2 SIMULATIONS AND RESULTS

V.A. Simulation Tool Setup

We simulate the network environment using the ns2 simulation tool, and establish a simple encoder and decoder at the terminal, written in g++, to be integrated with network codes and the scalable multiple description codes. The recover stream system was used to generate scalable multiple-description streams from the source video files. Because the network codes are only applied at the terminals, we do not need to insert a new network layer to handle network code processing at the intermediate nodes. As a result, no additional modification on the current TCP/IP structure is applied. Meanwhile, a simple analyzer is mounted at each receiver, to collect the statistics about the received video stream.

V.B. Simulation Environment Introduction

The whole simulation system can break down into three parts: the rate-allocation server, the encoder/decoder at each peer, and the network simulated by ns2. First, the rate-allocation server handles the video stream request from downstream peers, determines the upstream peer group, which obtains the video information, and establishes connections between the upstream peers and the downstream peers. Another functionality the server maintains is that it collects all the network condition information from each upstream peer, runs the algorithm based on collected information, and decides the rate allocation among different upstream peers. Secondly, the encoder/decoder at each peer helps encode the source video file if it is an upstream peer, while the encoder/decoder decodes the received video stream if it is a downstream peer. The encoder/decoder is integrated with network codes and multiple description codes to leverage the network resources. Thirdly, we use the ns-2 tool to simulation the TCP/IP-based peer-to-peer network environment.

In our simulation, we establish 10 upstream peers initially, which are in charge of pushing video streams to the downstream peers. Simultaneously, we set up 20 downstream peers to establish connections from all 10 available upstream peers, so that those downstream peers receive video streams from different sources, and analyze the overall system performance (See Figure 4).

To simulate the dynamics of the peer-to-peer system, we introduce randomness to the join and departure of each peer, and each peer is loaded with a timer to keep track of how long it stays in the system. In order to do so, we have to set up a random variable generator in the tcl script, to both randomize the entry time of each peer and the duration of each peer in the system.

To simulate different bandwidth conditions for each peer in the system, we set up different link capacities for each upstream peer. To simplify the situation, we assume that the downstream peers have abundant downlink bandwidths. In addition, we assume that the transport network is functioning well enough not to be rendered as a bandwidth restriction to video transmission. We categorize all the up-stream peers into three different groups with different bandwidth conditions: 128Kbps, 512Kbps, and 1Mbps.

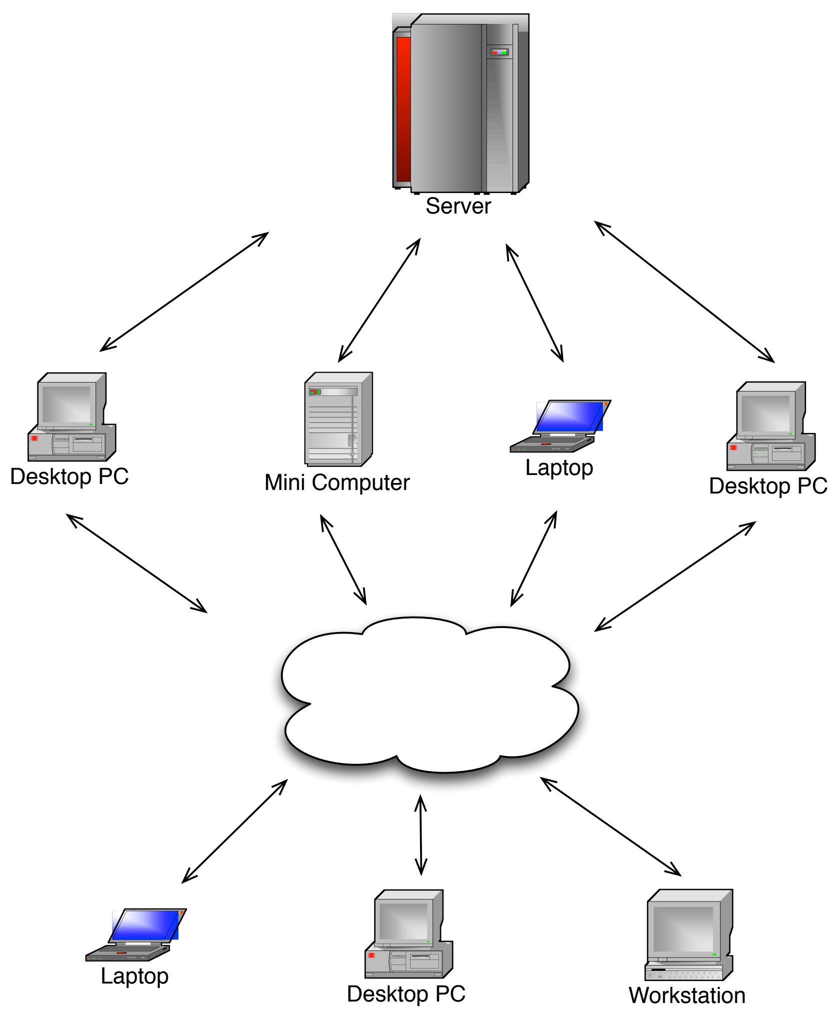

Figure 4: Test environment

V.C. PSNR Simulation Results

We compare the performance of our scheme with other existing schemes and observe the PSNR results within the same network environment. We select 7 video files (including QCIF, CIF and VGA formats) with different bit rates to examine the relationship between the PSNR and the bit rate under different schemes. We also set up other two schemes as the counterparts to compare the performance between different schemes: the Traditional scheme in which only pull-based method is used; and the Mesh-Based Push Method in which the Network Codes are applied. All the methods are tested and compared within the exactly same network configuration, and errors caused by randomness can be eliminated. Figure 5 shows the comparison between the Traditional scheme and our scheme, and Figure 6 shows the comparison between the Mesh-Based Push Method and our scheme.

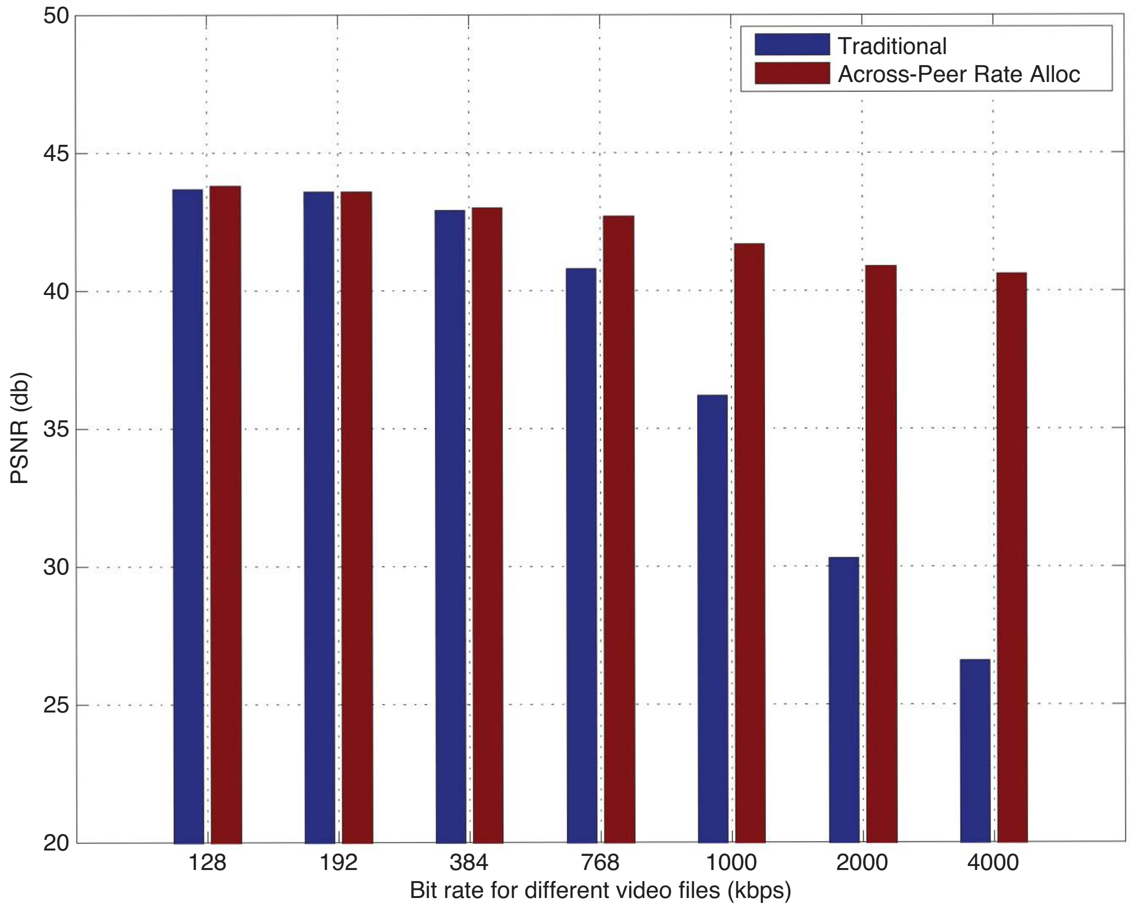

Figure 5: PSNR for different video bit rates in Traditional scheme and our scheme

From the simulation results, we can see that the network coding does not produce obviously better performance than the one without network coding when the bit rate of the video file is low. When the bit rate becomes high, we can see that the network coding starts to take effect to improve the video transmission in the network, and therefore the video stream can be better recovered within the terminals. Meanwhile, we can also observe that the across-peer allocation scheme can help increase PSNR by 7.8db at 4Mbps bit rate when compared with the Mesh-Based Push Method. This agrees with our discussion above because our scheme can utilize the network resource at the best, based on the collected bandwidth information from all the upstream peers, especially when the traffic is very large.

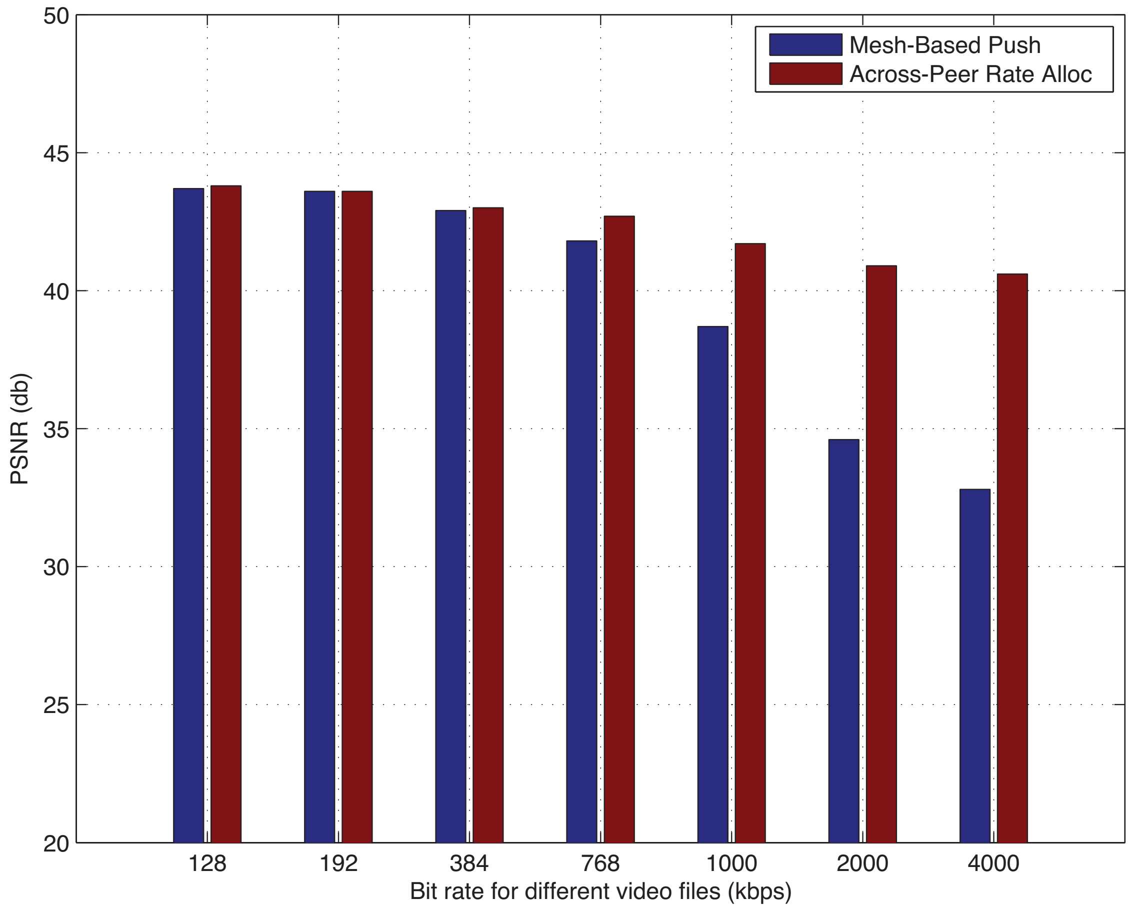

Figure 6: PSNR for different video bit rates in Mesh-based scheme and our scheme

V.D. Continuity Simulation Results

We also adopt continuity [14] as the new metric to evaluate the video playback quality, which is defined as the number of peers that have played the segment successfully in each time slot divided by the total number of active peers in the system. First, we compare our scheme with the Tradition Scheme. Observed by Figure 3, we can get that the new algorithm can perform better than the Tradition one. Similarly, we also compare our scheme with the Mesh-Based Pull Method with Network Codes. From the simulation results, we can see that the new algorithm can adapt to the network environment better than the prior scheme, thus yielding a better playback experience at the peer ends.

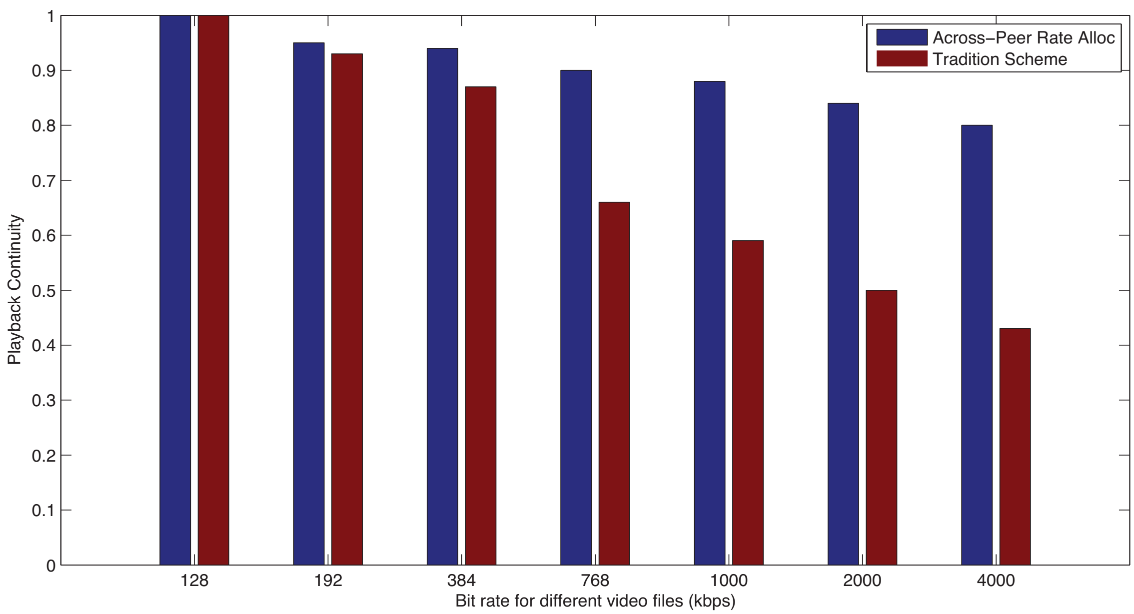

Figure 7: Continuity for different video bit rates in Traditional scheme and our scheme

From the analysis above, together with Figure 7 and Figure 8, we can see that our new algorithm helps improve the continuity of the video playback a lot, especially when the video rate is high when compared to the bandwidth condition. For instance, the continuity has been increase from 0.403 to 0.812 when the video rate is 4kpbs. As verified in the last chapter, the across-peer rate allocation can automatically detect the network condition, and choose the best scheme to leverage the network resources so that the transmission process is optimized.

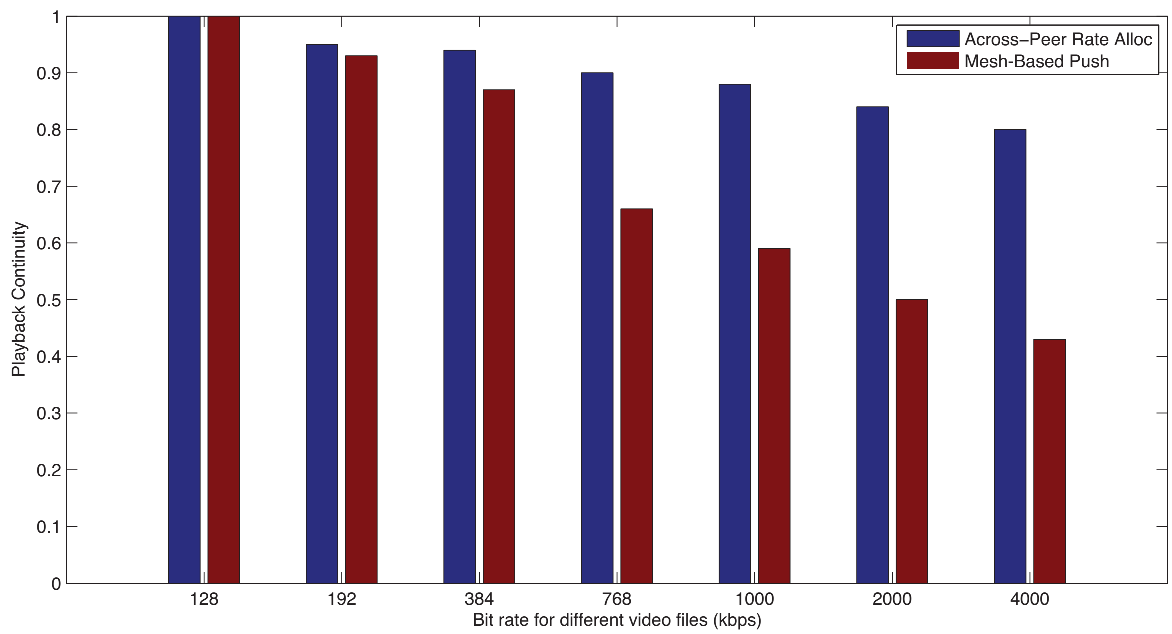

Figure 8: Continuity for different video bit rates in Mesh-based scheme vs. our scheme

CHAPTER VI

CONCLUSIONS AND FUTURE WORKS

In this thesis, a novel across-peer rate allocation algorithm is presented to improve the video transmission performance in peer-to-peer networks. As distinguished from previous works, we proposed the algorithm that monitors all the upstream peers in the system, and collects bandwidth information from each peer periodically, so that we can dynamically allocate to each upstream node the rate task that matches the real-time network condition. As a result, adaptive rate partition adjustment is applied to ensure the real reflection of the packet drop rate in the network. Another contribution is that the granularity of network transmission unit is changed to the scale of atomic blocks instead of streaming rates in previous works, so that integration with network codes can be coped with in a better way.

Through both the theoretical analysis and simulation results, we can show that the new algorithm outperforms prior algorithms in terms of two video performance metrics: PSNR and continuity. PSNR results indicate that video playback quality has been improved a lot at the peer ends, especially when the network load is huge; Continuity results shows that the new algorithm helps the system adjust better to the peer dynamics.

In this thesis, we model the video stream as the constant bit rate source. However, this is not always true in reality. In future, our work can be expanded into the situations where varied bit rate sources are applied. Also as there is a trend that peer-to-peer networks are on a decline in contrast to rise of other types of distributed networks, we can migrate the current model to other types of distributed systems such as cloud computing. Owing to the fact that peer-to-peer networks are sharing a lot similarities with other distributed systems and that we do not take much advantage of the peer-to-peer architecture, it is very promising that this algorithm can be successfully applied in other distributed systems.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," Information Theory, IEEE Transactions on, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] K. Nguyen, T. Nguyen, and S.-c. Cheung, "Peer-to-peer streaming with hierarchical network coding," in Multimedia and Expo, 2007 IEEE International Conference on, pp. 396–399, IEEE, 2007.

[3] V. K. Goyal, "Multiple description coding: Compression meets the network," Signal Processing Magazine, IEEE, vol. 18, no. 5, pp. 74–93, 2001.

[4] C. Wu, B. Li, and S. Zhao, "Diagnosing network-wide p2p live streaming inefficiencies," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 8, no. 1, p. 13, 2012.

[5] A. K. Ramasubramonian and J. W. Woods, "Multiple description coding and practical network coding for video multicast," Signal Processing Letters, IEEE, vol. 17, no. 3, pp. 265–268, 2010.

[6] C. Gkantsidis, J. Miller, and P. Rodriguez, “Comprehensive view of a live network coding p2p system,” in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 177–188, ACM, 2006.

[7] C.Gkantsidis and P.R.Rodriguez, “Network coding for large scale content distribution,” in INFOCOM2005.24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4, pp. 2235–2245, IEEE, 2005.

[8] C. Feng and B. Li, “On large-scale peer-to-peer streaming systems with network coding,” in Proceedings of the 16th ACM International Conference on Multimedia, pp. 269–278, ACM, 2008.

[9] A. Reibman, H. Jafarkhani, Y. Wang, and M. Orchard, “Multiple description video using rate-distortion splitting,” in Image Processing, 2001. Proceedings. 2001 International Conference on, vol. 1, pp. 978–981, IEEE, 2001.

[10] P. Correia, P. A. Assuncao, and V. Silva, “Multiple description of coded video for path diversity streaming adaptation,” Multimedia, IEEE Transactions on, vol. 14, no. 3, pp. 923–935, 2012.

[11] R.Puri and K.Ramchandran, “Multiple description source coding using forward error correction codes,” in Signals,Systems,andComputers, 1999. Conference Record of the Thirty-Third Asilomar Conference on, vol. 1, pp. 342–346, IEEE, 1999.

[12] Y. Zhou, D. M. Chiu, and J. C. Lui, “A simple model for analyzing p2p streaming protocols,” in Network Protocols, 2007. ICNP 2007. IEEE International Conference on, pp. 226–235, IEEE, 2007.

[13] M. Wang and B. Li, “R2: Random push with random network coding in live peer-to-peer streaming,” Selected Areas in Communications, IEEE Journal on, vol. 25, no. 9, pp. 1655–1666, 2007.

[14] Y. Zhou, D. M. Chiu, and J. C. Lui, “A simple model for analyzing p2p streaming protocols,” in Network Protocols, 2007. ICNP 2007. IEEE International Conference on, pp. 226–235, IEEE, 2007.

[15] Ho, Tracey, et al. "On randomized network coding." Proceedings of the Annual Allerton Conference on Communication Control and Computing. Vol. 41. No. 1. The University of Illinois at Urbana-Champaign; 1998, 2003.

[16] Chou, Philip A., Yunnan Wu, and Kamal Jain. "Practical network coding." Proceedings of the Annual Allerton Conference on Communication Control and

Computing. Vol. 41. No. 1. The University of Illinois at Urbana-Champaign; 1998, 2003.

[17] Wicker, Stephen B. Error control systems for digital communication and storage. Vol. 1. Englewood Cliffs: Prentice Hall, 1995.

APPENDIX A

CONVERGENCE OF THE PROBABILITY PROFILE

Given the initial probability distribution profile $(p_1, p_2 \cdots p_M)$, in which p_i indicates the probability that i packets out of M packets are received. Following the flow chart given in Figure 9, the final output distribution profile would be $(p_1^*, p_2^* \cdots p_M^*)$.

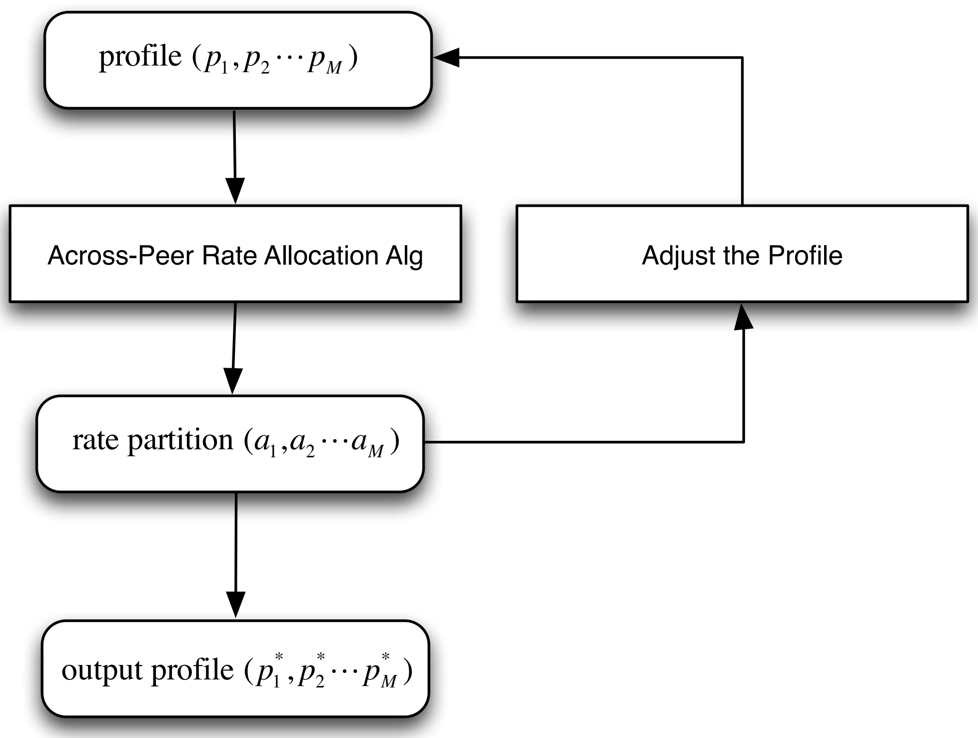

Figure 9: Adjust the probability profile

We have to prove that the adjustment process is convergent.

PROOF:

For any probability profile $(p_1, p_2 \cdots p_M)$ after rate allocation, each probability element belongs to one of the partition groups $P = (A_1, A_2 \cdots A_i)$. There are two different cases for two adjacent probability elements in the profile:

1) If two adjacent probability elements lies in the same partition group, then $p_i = p_j$ according to the property of network codes. Then on the rate-distortion curve, these two probability elements will merge into a single point. Likewise, all the probability elements in the same partition group will merge into a single point on the rate-distortion curve. As a result, the total number of working point set will be equal to the size of partition groups: $card(working\ set) = card(P)$.

The process of rate-allocation algorithm will form the steady partition if $\frac{dD(R_i)}{dR_i} \leq \frac{dD(R_{i+1})}{dR_{i+1}}$. When the size of the working set is equal to size of partition groups, this condition satisfies. As a result, the system yields a steady output distribution profile.

2) If two adjacent probability elements lies in the boundary of two partition groups $p_i \in A_m$ and $p_j \in A_n$, then $p_i > p_j$ according to the property of network codes that the entire block is decodable when sufficient coded blocks are received to form the linearly

independent coefficient full-rank matrix. Then on the rate-distortion curve, these two probability elements reside in different points. If $\frac{dD(R_i)}{dR_i} \leq \frac{dD(R_j)}{dR_j}$ is satisfied, the partition will be the same as the previous one, and the output profile is generated; if $\frac{dD(R_i)}{dR_i} \leq \frac{dD(R_j)}{dR_j}$ is not satisfied, p_j will be forcibly excluded in the working set. Equivalently, we can consider that p_i and p_j merge into the same point. Then this reverts to the situation depicted in the first case.