UNIVERSITY OF MINNESOTA

This is to certify that I have examined this bound copy of a master's report by

HUNG QUANG NGO

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Jaideep Srivastava
Name of Faculty Adviser

Signature of Faculty Adviser

Date

GRADUATE SCHOOL

ON OPTIMAL FRAME TRANSMISSION ORDER OF CONTINUOUS MEDIA STREAMS

A REPORT
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

HUNG QUANG NGO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

MAY 1999

ACKNOWLEDGMENTS

I would like to thank a few people, without whom this work is far from completion.

First of all, my sincerest thanks to professor Jaideep Srivastava, for pulling me off the ground into his group of fine people when I was a first year graduate student, for his helpful suggestions and guidance throughout my first two years at the Univesity. I have learnt a lot from his always clear and simple style of presentations.

Second of all, I truly appreciate what professor Dennis Stanton has done to me. He introduced me to the beauty of Combinatorics and gave me great inspiration to work on combinatorial problems. Even though the problem I solved here is rated at most 2+ out of 5 in his scale, I would not have been able to solve it without him.

The third person I would like to thank is my collaborator and also my very good friend – Srivatsan Varadarajan. Working with Vatsan has always been fun and productive, partially due to his hardwork and sincerity.

I would also like to thank the other member of the commitee – professor Ding Zhu Du.

DEDICATION

To my parents, who gave me everything I have today ...

ABSTRACT

With the growing popularity of the Internet, there is an increasing interest in using it for audio and video transmission. Periodic network overloads, leading to bursty packet losses, have always been a key problem for network researchers. In a long-haul, heterogeneous network like the Internet, handling such an error becomes especially difficult. Perceptual studies of audio and video viewing have shown that bursty losses have the most annoying effect on people, and hence are critical issues to be addressed for applications such as Internet phone, video conferencing, distance learning, etc. Classical error handling techniques have focused on applications like FTP, and are geared towards ensuring that the transmission is correct, with no attention to timeliness. For isochronous traffic like audio and video, timeliness is a key criterion, and given the high degree of content redundancy, some loss of content is quite acceptable. In this report we introduce the concept of *error spreading*, which is a transformation technique that takes the input sequence of packets (from an audio or video stream) and permutes them before transmission. The packets are un-permuted at the receiving end. The transformation is designed to ensure that bursty losses in the transformed domain get spread all over the sequence in the original domain. Perceptual studies have shown that users are much more tolerant to a uniformly distributed loss of low magnitude. We then describe a continuous media transmission protocol based on this idea, and validate its performance through an experiment performed on the Internet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the phenomenal growth of multimedia systems and their applications, there have been numerous research efforts directed at providing a *continuous media (CM)* service over varying types of networks. With the boom of the Internet, continuous media like audio and video are using the Internet as the principal medium for transmission. However, the Internet provides a *single class best effort* service, and does not provide any sort of guarantees [BT98]. A network characteristic of special concern to this report is transmission errors, and specifically the dropping of data packets. Packets are dropped when the network becomes congested, and given the nature of this phenomenon, strings of successive packets are often dropped [Bol93, Pax97], leading to significant bursty errors [YCV96].This bursty loss behavior has been shown to arise from the drop-tail queuing discipline adopted in many Internet routers [PFTK98].

Handling bursty errors has always been problematic, especially since no good models exist for its prediction. On the other hand, most applications, especially realtime Multimedia applications, do not tolerate bursty error, making it imperative that they be handled in a good manner. Perceptual studies on continuous media viewing have shown that user dissatisfaction rises dramatically when bursty error goes beyond a certain threshold [WSNF97, WS96, WVP$^+$98]. This is especially so for audio, where the threshold is quite small, and hence this issue is quite pressing for applications like Internet phone.

These observations point quite solidly to the need for development of efficient mechanisms to control bursty errors in continuous media streaming through networks. Redundancy is the key to handling packet loss/damage in standard communication protocols. There are two main classes of schemes, namely the *reactive* schemes and the *proactive* schemes.

Reactive schemes respond by taking some action once transmission error has been detected, while pro-active schemes take some action in advance to avoid errors . A protocol such as TCP is reactive since the receiver sends a feedback to the sender upon detecting an error, in response to which the sender will re-transmit the lost or corrupted packet. The reaction can be initiated by the source or the sink. *Source initiated reaction* occurs in schemes based on feedback combined with retransmission like [BT98, BTW94, RR93]. The feedback control can be based on stream rate [ZSKT96, MJV96], bandwidth [GJS96], loss/delay [BT98] and a wide variety of network QoS parameters [Tow93, AFKN94, RTP94]. *Sink initiated reaction* occurs in reconstruction based schemes like [WL98, ABLS94]. Coding data in an error correcting manner before transmission is a pro-active scheme where corrupted packets can be reconstructed ([HBK98] and *Forward Error Correction Codes* [BG, BT96]).

Each of these classes of schemes requires extra bandwidth, for feedback and retransmission in the first, and for extra bits in the second category. This need for extra bandwidth can exacerbate the problem, especially since network congestion is the principal culprit for the bursty errors. One more approach that has been proposed, is to fulfill the real time needs of CM applications with other services like *RSVP and RTP*, which offer varying degree of performance guarantees [ZDE+93, SCFM94]. Services like RTP/RSVP require that some resource allocation and/or reservation mechanism be provided by the network [BT98]. Since these mechanisms are not yet widely deployed in the Internet, our focus has not been on them.

Recent work ([SJBG98, ZNAT99]) has proposed schemes where the overall characteristics of the data being transmitted can be used to control the transmission error. For example, [ZNAT99] has proposed selectively dropping video frames on the sender side, based on a *cost-benefit analysis* which takes into account the desired Quality of Service (QoS). This is quite effective in a LAN (senders are known and cooperative) or the Internet using RED gateways where during congestion, the probability that the gateway notifies a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth through the gateway [FJ93]. While drop-tail queuing discipline is still adopted in many routers [PFTK98], this scheme may not be directly applicable yet.

In this report we propose a new type of scheme for handling bursty errors, which we call *error spreading*. A key advantage of this scheme is that it is not based on redundancy, and hence requires absolutely no extra bandwidth. The main idea is that we do not try to reduce overall error, but rather tradeoff bursty error (the *bad error*) for average error (the *good error*). Perceptual study of continuous media viewing [WSNF97, WS96, WVP+98] has shown that a reasonable amount of overall error is acceptable, as long as it is spread out, and not concentrated in spots. A similar approach has been taken by [YCV96]. But they have used a random scrambling techniques with redundant reconstruction for audio, and as stated by them they have not investigated the buffer requirements. We have established clearly the bounds and the relationship between buffer requirement and user perceived quality in a bounded bursty network error scenario.

In this report we make several contributions. First, we formulate the problem of error handling in continuous media transmission as a tradeoff between the user QoS requirements, network characteristics, and sender resource availability. Second, we provide a complete analytical solution for the special case where the network errors are bounded. While this solution may be of actual use in some specialized networks, e.g., a tightly controlled real-time network, its principal use is in providing important mathematical relationships that can be used as the basis of protocols for general networks. Next, we use this analysis to develop such a protocol for networks where there is no bound on the error. Finally, we present results of an experimental evaluation that illustrates the benefits of the proposed scheme.

This report is organized as follows: chapter 2 formulates the problem and chapter 3 presents a mathematical analysis of the bounded network error case. Chapter 4 presents the transmission protocol, while chapter 5 describes its evaluation. Chapter 6 concludes the report.

Chapter 2

Problem Formulation

This chapter briefly discusses the content based continuity QoS metrics introduced in [WS96]. Then, we define our problem based on this metrics.

2.1 QoS metrics

For the purpose of describing QoS metrics for lossy media streams, CM stream is envisioned as a flow of data units (referred to as logical data units, or LDUs, in the uniform framework of [SB96]). In our case, we take a video LDU to be a frame, and an audio LDU to consist of 8000/30, i.e. 266 samples of audio[1]. Given a rate for streams consisting of these LDUs, we envision that there is a time slot for each LDU to be played out. In the ideal case a LDU should appear at the beginning of its time slot. This report uses only the content based continuity metrics proposed in [WS96]. Issues on rate and drift factors (discussed in [WS96]) are not considered. Note also that we shall use the term LDU and frame interchangeably, since *frame* is very commonly used in Multimedia community.

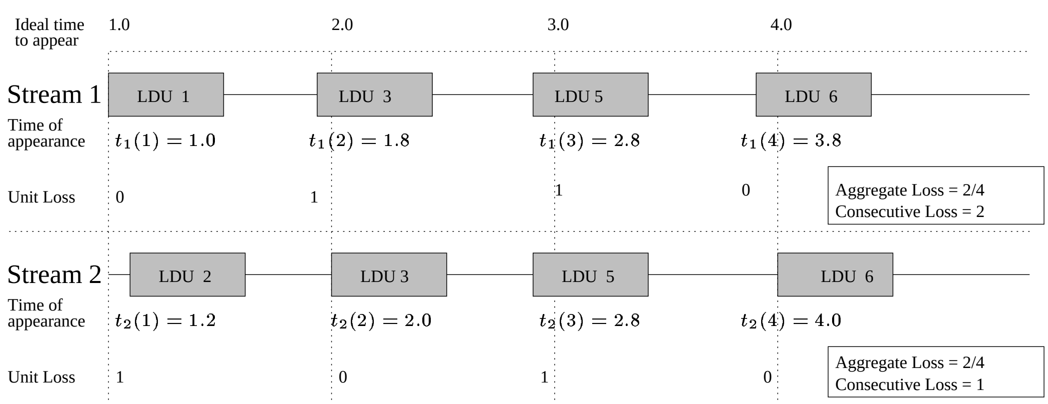

Figure 2.1: Two Sample Streams Used to Explain the QoS Metrics

The above figure is from [WS96]. Ideal contents of a CM stream are specified by the

[1] SunAudio has 8-bit samples at 8kHz, and an audio frame constitutes of 266 such samples equivalent to a play time of one video frame, i.e. 1/30 seconds.

ideal contents of each LDU. Due to loss, delivery error or resource over-load problems, appearance of LDUs may deviate from this ideal, and consequently lead to discontinuity. The metrics of continuity are designed to measure the average and bursty deviation from the ideal specification. A loss or repetition of a LDU is considered a unit loss in a CM stream. (A more precise definition is given in [WS96].) The aggregate number of such unit losses is the *aggregate loss (ALF)* of a CM stream, while the largest consecutive non-zero loss is its *consecutive loss (CLF)*. In the example streams of Fig. 2.1, stream 1 has an aggregate loss of 2/4 and a consecutive loss of 2, while stream 2 has an aggregate loss of 2/4 and a consecutive loss of 1. The reason for the lower consecutive loss in stream 2 is that its losses are more spread-out than those of stream 1. Note that the metrics already takes care of losses (both consecutive and aggregate) that arise due timing drifts [WS96].

In a user study [WSNF97] it has been determined that the tolerable value for consecutive video frame losses were determined to be two frames. For audio this limit was about three frames.

2.2 Problem Statement

One of the most important factors that affect the quality of a CM stream is the *Consecutive Loss Factor* [WS96] (CLF). Network often lose frames in bursts, alternating between lossy burst and successful burst [YCV96, Bol93, Pax97, PFTK98]. This often causes unacceptably high CLF. Our objective is to reduce CLF given the same network characteristics. The main idea is *loss spreading*, or distributing consecutive loss over some time period.

For example, suppose we sent a sequence of 17 consecutive video frames numbered 1 to 17. During transmission, a network bursty error of size 7 occurs which causes the loss of frames numbered 7 to 13, as shown in the first row of Table 2.1. This causes the stream to have a CLF of $7/17$.

Now suppose we permute this sequence of frames before transmission so that consecutive frames become far apart in the sequence, the CLF can be reduced significantly. To illustrate this idea, consider the frame transmission order shown in the second row of Table 2.1. With exactly the same bursty error once again consecutive frames are lost, except this time they are consecutive only in the permuted domain. In the original domain these are spread far apart.

Clearly, if the 17 frames were sent in this order, we would have had a CLF of only $1/17$. Table 2.1 summarizes our example by giving three sequences and their corresponding CLFs. The first sequence is the natural order of frames, the second is the permuted order, and the third is the un-permuted order observed at the receiver's side. The third sequence was presented to show how the loss has been spread out over the original sequence. The boxed numbers represent lost frames.

This example motivates the following problem.

	Frame sequence	CLF
In order	01 02 03 04 05 06 [07 08 09 10 11 12 13] 14 15 16 17	7/17
Permuted	01 06 11 16 04 09 [14 02 07 12 17 05 10] 15 03 08 13	1/17
Un-permuted	01 [02] 03 04 [05] 06 [07] 08 09 [10] 11 [12] 13 [14] 15 16 [17]	1/17

Table 2.1: An example of how the order of frames sent affects CLF

Bursty Error Reduction Problem (BERD)

- **Objective :** to reduce the bursty error, i.e. CLF, to an perceptually acceptable level (by spreading it out over the stream).
- **Input parameters:**
  - m is the sender's buffer size, in terms of LDUs. m is determined by the sender's operating environment and its current status.
  - p is the upper bound on the size of a bursty loss in the communication channel, within a window of m LDUs (we relax this assumption in chapter 4).
  - k is the user's maximum acceptable CLF.
- **Output :** a permutation function π on $S = \{1, 2, 3, \ldots, m\}$ which decides the order in which a set of m consecutive LDUs must be sent. Moreover, the system is expected to give the lower bound k_0 which is the minimum CLF that can be supported in this constrained environment.
- **Assumption :** two consecutive bursty loss are at least m LDUs apart.

Figure 2.2 visualizes how the solution space for a particular value of p would appear. The boundary of the curve is essentially what we found. Above it is the feasible region, where intuitively if we increase m, then k should still be the same or less. There is a typical trade off between buffer size m and CLF k. The greater m is, the less k we can support but also the greater memory requirement and initial delay time. Given m_0, line $m = m_0$ cuts the boundary curve at k_0 at or above which we can support. Conversely, given k_0, line $k = k_0$ intersects the curve at m_0 at or above which the buffer size must be in order to support k_0.

There are several points worth noticing. Firstly, we deal only with data streams that have no inter-frame dependency such as Motion-JPEG or uncompressed data streams (audio, video, sensor data, ...). The reason for this is that this allows us to consider every

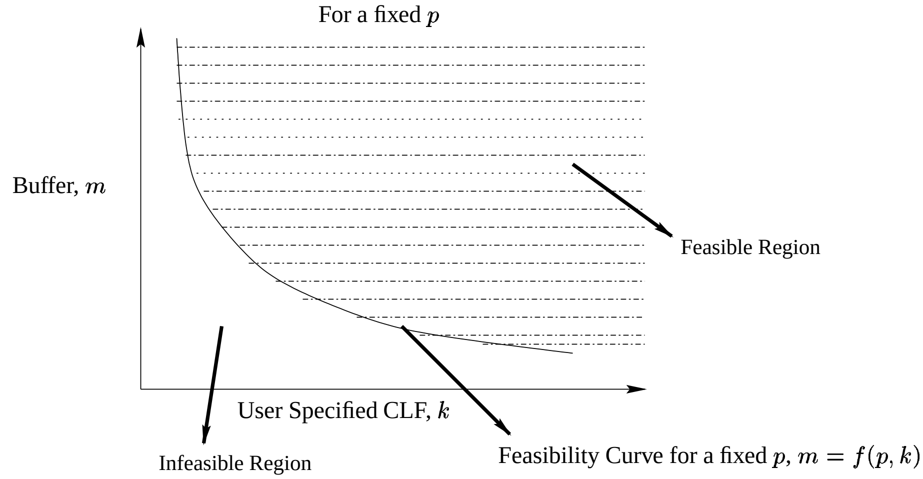

Figure 2.2: Part of Deterministic Solution Space

frame to be equally important; thus, we can permute the frames in any way we would like to. Secondly, the frames in these types of streams have relatively comparable sizes. For example, a sequence of MJPEG frames only has a change in size significantly when the scene switches. So, no matter if it is us who send the frames by breaking them up into equal size UDP packets or it is the transport layer interface (TLI) which does so, a consecutive packet loss implies a proportional consecutive frame loss. Finally, to satisfy our assumption that two consecutive lost windows are at least m frames apart, closer lost windows can be combined and consider to be a larger lost window.

Chapter 3

Bounded Network Error Case

In this chapter, we will discuss the case where the bound p of continuous network loss is known. For convenience, we first state the problem in purely mathematical terms and establish some notations to be used throughout the proof.

We are given positive integers m and p. Let S_m denotes the set of all permutations acting on $[m] = \{1, 2, \dots m\}$. For any permutation $\pi \in S_m$, the sets $W_i^\pi = \{\pi_i, \pi_{i+1}, \dots \pi_{i+p-1}\}, 1 \leq i \leq m$ are called the *sliding windows* of size p (of π), where the indices are calculated modulo m, then plus 1. Thus, when $1 \leq i \leq m-p+1$, $W_i^\pi = \{\pi_i, \pi_{i+1}, \dots \pi_{i+p-1}\}$, and when $m-p+1 < i \leq m$, $W_i^\pi = \{\pi_i, \dots, \pi_m, \pi_1, \dots, \pi_{i+p-m-1}\}$

For any pair of integers k and l such that $1 \leq k < l \leq m$, let $[k, l]$ denotes the set $\{k, k+1, \dots l\}$. Let $\{c_i^\pi\}_1^m$ be the sequence of integers defined as follows.

$$c_i^\pi = \begin{cases} \max\{|[k,l]|,\ [k,l] \subseteq W_i^\pi\} & \text{if } 1 \leq i \leq m-p+1 \\ \max\{|[k,m]| + |[1,l]|, & \text{if } m-p+1 < i \leq m \\ \quad [k,m] \subseteq \{\pi_i, \dots \pi_m\}, & \\ \quad [1,l] \subseteq \{\pi_1, \dots \pi_{p+i-m-1}\}\} & \end{cases}$$

Let $C^\pi = \max\{c_i^\pi | 1 \leq i \leq m\}$. Then k_0 is defined to be

$$k_0 = \min\{C^\pi, \pi \in S_m\}$$

Our objective is to find k_0 as a function of m and p. Moreover, we also wish to specify a permutation π so that $C^\pi = k_0$.

Informally, when $1 \leq i \leq m-p+1$, c_i^π is the maximum number of consecutive integers in W_i^π. While if $m-p+1 < i < m$, c_i^π is the sum of two quantities x and y, where x is the length of the longest consecutive integer sequence in $\{\pi_i, \dots, \pi_m\}$ which ends in m, and y is the length of the longest consecutive integer sequence in $\{\pi_1, \dots, \pi_{i+p-m-1}\}$ which starts at 1. The reason for this is that suppose we apply our permutation to two adjacent buffers of size m, we would like our permutation to also deal with the case where the network loss burst occurs across these two buffers.

The value of k_0 and permutation π depends tightly on the relationship between p and m. Section 3.1 presents the case where $p \leq \frac{m}{2}$, in which we have more freedom to choose our permutation. Section 3.2 discusses the case where $\frac{m}{2} < p < m$. Lastly, section 3.3 summarizes our work on the bounded error case by a theorem and an algorithm which gives us a *good* permutation given m and p.

Remark 1 *If p is known and m is fixed, then $k_0 = 0$ when $p = 0$ and $k_0 = m$ when $p \geq m$.*

Throughout this chapter, we assume m and p were given, unless specified otherwise.

3.1 Simple case

Lemma 1 *If $0 < p \leq \frac{m}{2}$ then $k_0 = 1$*

Proof : Since $p > 0$, we have $k_0 \geq 1$. So, to prove $k_0 = 1$ it is sufficient to specify a permutation π on $S = \{1, 2, 3, \ldots m\}$ so that $C^\pi = 1$. To avoid possible confusion, we would like to point out that for any $\pi \in S_m$, $\pi(i)$ specifies the position of i in the permuted sequence where i is sent to, while π_i is the number at position i in one line notation of π. For example, if $\pi = 4\ 1\ 3\ 6\ 2\ 5$ then $\pi_1 = 4$, $\pi_2 = 1$, but $\pi(1) = 2$ and $\pi(2) = 5$.

We consider two cases based on the parity of m as follows.

- **Case 1 : m is odd**

  Let $m = 2m' + 1$, $p' = \min\{j, j \geq p \wedge \gcd(m, j) = 1\}$. Since $p \leq \frac{m}{2}$, we have $p \leq m'$. Moreover, $\gcd(m, m') = 1$, hence $p' \leq m' < \frac{m}{2}$. We now construct a permutation π such that $C^\pi = 1$. Let π be defined as follows.

  $$\pi(i) = ((i-1).p' \bmod m) + 1,\ 1 \leq i \leq m \tag{3.1}$$

  We first need to prove that π is indeed a permutation. As $\gcd(m, p') = 1$, p' generates the group Z_m of integers modulo m ($\mathrm{Z}_m = \{0, 1, \ldots m-1\}$). Thus the sets $\{0, 1, \ldots m-1\}$ and $\{0, p', 2p' \bmod m, \ldots (m-1)p' \bmod m\}$ are identical. So (3.1) defines a valid $\pi \in S_m$.

  Secondly, we need show that $C^\pi = 1$. Let $\pi = \pi_1, \pi_2, \ldots \pi_m$ in one line notation. If $C^\pi \geq 2$ then there exists i and j, $1 \leq i < j \leq m$ such that **either** $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same sliding window W_k^π for some $1 \leq k \leq m - p + 1$ **or** $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same sliding window W_k^π for some $m - p + 1 < k \leq m$.

  Note that by definition of π, we have $i = ((\pi_i - 1).p' \bmod m) + 1$ and $j = ((\pi_j - 1).p' \bmod m) + 1$, thus,

  $$j - i = (\pi_j - \pi_i).p' \bmod m \tag{3.2}$$

  – If $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same W_k^π for some $1 \leq k \leq m-p+1$, then we must have $j - i < p$. Moreover, $|\pi_i - \pi_j| = 1$ and (3.2) imply $j - i$ is either p' or $m - p'$. As stated earlier, $p \leq p' < m/2$, so $m - p' > p' \geq p$, making $j - i < p$ impossible.

- Otherwise, suppose $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same sliding window W_k^π for some $m - p + 1 < k \leq m$. Notice that we must have $m - j + i < p$ for both π_i and π_j to be in W_k^π. Moreover, $\{\pi_i, \pi_j\} = \{1, m\}$ and (3.2) imply that $j - i$ is either p' or $m - p'$. So $m - j + i \in \{p', m - p'\}$, and similar to the previous case, this makes $m - j + i < p$ impossible

In sum, we have just shown that $c_i^\pi = 1, \forall 1 \leq i \leq m$, so $C^\pi = 1$. Notice that the choice of p' could have been any integer between p and $m/2$, as long as $\gcd(m, p') = 1$. In particular, $p' = m'$ clearly works. However, we have chosen p' to be the minimum of these because we would like the permuted sequence to be spreaded out in a "better manner". To illustrate this, consider the case where $m = 17$ and $p = 5$. Our choice of p' is 5 in this case, thus in two line notation we have

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 1 & 8 & 15 & 5 & 12 & 2 & 9 & 16 & 6 & 13 & 3 & 10 & 17 & 7 & 14 & 4 & 11 \end{pmatrix}$$

While if we choose $p' = m' = 8$, the permutation becomes

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 1 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 & 17 & 15 & 13 & 11 & 9 & 7 & 5 & 3 \end{pmatrix}$$

Although this permutation does satisfy $C^\pi = 1$, it has the alternative numbers being too close to each other, thus we might end up in a situation where we lose every other frame for a while and then a chunk of successfully arrived frames. This clearly creates worse affect perceptually on the media stream comparing to the previous choice of p'.

- **Case 2 : m is even**

  If $\{j, p \leq j < \frac{m}{2} \wedge \gcd(m, j) = 1\} \neq \emptyset$ then we can just use exactly the same permutation as when m is odd.

  If the set is empty, let $m = 2p' \geq 2p$, so $p' \geq p$. As the previous case, we seek a permutation π so that $C^\pi = 1$. Let π be defined as follows.

$$\pi(i) = p'.(i \bmod 2) + \left\lceil \frac{i}{2} \right\rceil,\ 1 \leq i \leq m \tag{3.3}$$

  If i is even, then $\pi(i) = \frac{i}{2}$, namely all the even numbers will be placed from the first position to the $\left(\frac{m}{2}\right)^{th}$ position in increasing order. When i is odd, $\pi(i) = p' + \frac{i+1}{2}$, so all the odd numbers will be placed from the $(\frac{m}{2}+1)^{st}$ position to the m^{th} position. It is clear from the above observation that π is a proper permutation on S.

We are left to prove that $C^{\pi} = 1$. Write $\pi = (\pi_1, \pi_2, \ldots \pi_m)$ in one line notation. Firstly, notice that for any $1 \leq i < j \leq m$, we have

$$\begin{aligned} i &= p'.(\pi_i \bmod 2) + \left\lceil \frac{\pi_i}{2} \right\rceil \\ j &= p'.(\pi_j \bmod 2) + \left\lceil \frac{\pi_j}{2} \right\rceil \end{aligned}$$

So,

$$j - i = p'.((\pi_j - \pi_i) \bmod 2) + \left\lceil \frac{\pi_j}{2} \right\rceil - \left\lceil \frac{\pi_i}{2} \right\rceil \tag{3.4}$$

Similar to case 1, if $C^{\pi} \geq 2$ then we consider two subcases :

– If $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same W_k^{π} for some $1 \leq k \leq m - p + 1$ then we must have $j - i < p$. Moreover, since $|\pi_i - \pi_j| = 1$ and $j > i$, it must be the case that π_j is odd and π_i is even. Combining with (3.4), we have

$$j - i = p' + \left\lceil \frac{\pi_j}{2} \right\rceil - \left\lceil \frac{\pi_i}{2} \right\rceil \geq p' \geq p$$

This makes $j - i < p$ impossible.

– Otherwise, if $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same W_k^{π} for some $m - p + 1 < k \leq m$, then we must have $m - j + i < p$ for both π_i and π_j to be in W_k^{π}. By (3.3), $\pi(1) = p' + 1$ and $\pi(m) = p'$, so it must be the case that $j = p' + 1$ and $i = p'$. Thus, $m - j + i = m - 1 \geq p' \geq p$, contradiction ! $\square$

Example: Let $m = 16$ and $p = 8$, then

$$\pi = \begin{pmatrix} 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \end{pmatrix}$$

3.2 More complicated case

Lemma 2 *If* $\frac{m}{2} < p < m$ *then* $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$

Proof : In order to prove that $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$, we will first prove that

$$k_0 \geq \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 \tag{3.5}$$

then specify a permutation π so that $C^{\pi} = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Let $\sigma = (\sigma_1, \sigma_2, \ldots \sigma_m)$ be any permutation on $S = \{1, 2, \ldots m\}$. Let $Y = y_1, y_2, \ldots y_{m-p}$ be the sequence obtained from sorting $\sigma_1, \sigma_2, \ldots \sigma_{m-p}$ in increasing order. Intuitively, Y is the complement of W^{σ}_{m-p+1} with respect to S.

Now, for convinience let $y_0 = 0$ and $y_{m-p+1} = m + 1$. Let $S_i = \{x : y_{i-1} < x < y_i, x \in S\}$, where $1 \leq i \leq m - p + 1$. S_i is simply the set of numbers between y_{i-1} and y_i in S.

$$y_0 = 0 \underbrace{1\ 2\ 3\ \ldots}_{S_1} y_1 \underbrace{\ldots}_{S_2} y_2 \cdots \underbrace{\ldots}_{S_i} y_i \ \cdots \ \underbrace{\ldots}_{S_{m-p}} y_{m-p} \underbrace{\ldots\ m}_{S_{m-p+1}} m + 1 = y_{m-p+1}$$

It is clear that

$$\sum_{i=1}^{m-p+1} |S_i| = p$$

Since $S_1 \uplus S_2 \uplus \ldots \uplus S_{m-p+1} = W^{\sigma}_{m-p+1}$, where $\uplus$ denotes disjoint union, it must be the case that $c^{\sigma}_{m-p+1} \geq |S_i|, \ \forall i \in \{1, 2 \ldots, m - p + 1\}$. Thus, we have

$$p = \sum_{i=1}^{m-p+1} |S_i| \leq \sum_{i=1}^{m-p+1} c^{\sigma}_{m-p+1} = (m - p + 1).c^{\sigma}_{m-p+1}$$

This implies

$$C^{\sigma} \geq c^{\sigma}_{m-p+1} \geq \left\lceil \frac{p}{m-p+1} \right\rceil$$

this inequality holds for all $\sigma \in S_m$'s, consequently

$$k_0 \geq \left\lceil \frac{p}{m-p+1} \right\rceil \tag{3.6}$$

Inequality (3.6) is not as tight as (3.5). To prove that (3.5) holds, we need to consider two cases.

$S - W^{\sigma}_{m-p+1}$	W^{σ}_{m-p+1}	
$\sigma_1, \sigma_2, \ldots \sigma_{m-p}$	$\sigma_{m-p+1}, \ldots \sigma_p$	$\sigma_{p+1}, \ldots \sigma_m$
W^{σ}_1		$S - W^{\sigma}_1$

Table 3.1: Pictorial illustration of $W^{\sigma}_1, W^{\sigma}_{m-p+1}, S - W^{\sigma}_1$ and $S - W^{\sigma}_{m-p+1}$

Case (i) : $(m-p+1) \not| \, p$. In this case, $\lceil \frac{p}{m-p+1} \rceil = \lfloor \frac{p}{m-p+1} \rfloor + 1$, so (3.6) implies (3.5).

Case (ii) : $(m-p+1) \mid p$. In this case, let $y = \lceil \frac{p}{m-p+1} \rceil = \lfloor \frac{p}{m-p+1} \rfloor$.

Suppose $k_0 = y$ and let $\sigma = \sigma_1, \sigma_2, \ldots \sigma_m$ be a particular permutation on S so that $C^{\sigma} = y$. From the definition of C^{σ}, for any $1 \leq i \leq m-p+1$, we must have $c^{\sigma}_i \leq y$. Consider two special sliding windows $W^{\sigma}_1 = \{\sigma_1, \sigma_2, \ldots \sigma_p\}$ and $W^{\sigma}_{m-p+1} = \{\sigma_{m-p+1}, \sigma_{m-p+2}, \ldots \sigma_m\}$. Table 3.1 illustrates the situation.

We have $2p > m$, so $p > m - p$. Hence, there is no overlapping between $S - W^{\sigma}_1$ and $S - W^{\sigma}_{m-p+1}$. From the previous analysis, $k_0 = y$ holds if and only if $|S_i| = k_0, \forall i \in \{1, 2, \ldots, m-p+1\}$. So it must be the case that

$$y_i = i.(k_0 + 1), \forall i = 1, 2, \ldots m - p \tag{3.7}$$

because if $\exists i, |S_i| < k_0$, then

$$\begin{aligned} m &= \textstyle\sum_{i=1}^{m-p+1} |S_i| + (m-p) \\ &< (m-p+1).k_0 + (m-p) \\ &= (m-p+1).\frac{p}{m-p+1} + (m-p) \\ &= m, \text{ contradiction!} \end{aligned}$$

The previous analysis was done based on the window W^{σ}_{m-p+1} containing the last p elements of σ. So we have $\{y_1, y_2, \ldots\ldots y_{m-p}\} = S - W^{\sigma}_{m-p+1}$. Notice that exactly the same result holds if we do the analysis with respect to W^{σ}_1, in which case the y_i's are also determined by (3.7). So, we have the following :

$$\{1(k_0+1), 2(k_0+1), \ldots, (m-p)(k_0+1)\} = S - W^{\sigma}_{m-p+1} = S - W^{\sigma}_1$$

This is impossible since $(S - W^{\sigma}_1)$ and $(S - W^{\sigma}_{m-p+1})$ are disjoint. Consequently $k_0 \neq y$, so $k \geq y + 1 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Finally, to show that $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$, it is sufficient to specify a permutation π on S such that $C^\pi = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Before specifying π, we need to observe some facts. Let $q = m - p$, $r = \left\lfloor \frac{p}{q+1} \right\rfloor$, and $t = \left\lfloor \frac{m}{r+2} \right\rfloor$, then $q \geq 1$ because $q = m - p > 0$. Moreover, since $p > m - p = q$, we have $p \geq q + 1$. Thus $r \geq 1$ and we can write p as :

$$p = \quad (q+1)r + r', \quad 0 \leq r' \leq q$$

In addition, $m = p + q = (q+1)r + r' + q \geq qr + q + r \geq r + 2$, so $t \geq 1$ and we can write m as :

$$m = \quad (r+2)t + t', \quad 0 \leq t' \leq r + 1$$

Now, we specify π in one line notation by considering 2 cases as follows.

Case (i) : $t' = r + 1$.

We have

$$\begin{array}{rrcl} & m & = & p + q \\ \Rightarrow & (r+2).t + (r+1) & = & (q+1).r + r' + q \\ & & = & (q-1).(r+2) + r + r' - q + r + 2 \\ \Rightarrow & (r+2).t + (r+1) & = & (q-1).(r+2) + (r' - q) + 2.r + 2 \\ \Rightarrow & (r+2).t & = & (q-1).(r+2) + (r' - q) + r + 1 \end{array}$$

Moreover, since $r' \leq q$, we get

$$\begin{array}{rrcl} & (r+2).t & \leq & (q-1).(r+2) + r + 1 \\ \Rightarrow & t & \leq & (q-1) + \frac{r+1}{r+2} \end{array}$$

t and q are natural numbers, so

$$t \leq q - 1 \tag{3.8}$$

We are ready to describe our π now. Notice that we would like C^π to be $r+1$, making $k_0 = r + 1$ as desired. First we define 2 special arrays $A = \{a_i\}_1^{t+1}$ and $B = \{b_i\}_1^{t+1}$ as follows.

$$\begin{array}{rcll} a_i & = & 1 + (i-1).(r+2), & 1 \leq i \leq t+1 \\ b_i & = & (r+1) + (i-1).(r+2), & 1 \leq i \leq t+1 \end{array}$$

It is obvious that $1 \leq a_i \leq m$ and $1 \leq b_i \leq m$, thus A and B are subsequences of S. Let $C = \{c_i\}_1^{m-2.(t+1)}$ be the increasing sequence of numbers in S but not in A

and B. Notice that if we write numbers $1, 2, \ldots m$ in increasing order, we have the following picture :

$$\begin{array}{ccccccccccccccc} & & b_1 & & & & b_2 & & & & & & & & b_{t+1} \\ 1 & .. & r+1 & .. & r+3 & .. & 2r+3 & .. & 2r+5 & .... & 1+t(r+2) & .. & m \\ a_1 & & & & a_2 & & & & a_3 & & a_{t+1} & & \end{array}$$

and the rest of the numbers (the dots in the picture) contribute sequence C. Finally, here is our π in one line notation :

$$\pi = (\underbrace{a_{t+1}, a_t, \ldots a_1}_{A}, \underbrace{c_{m-2.(t+1)}, \ldots\ldots\ldots\ldots, c_1}_{C=S-A-B} \underbrace{b_{t+1}, b_t, \ldots b_1}_{B})$$

To prove that this is a valid permutation, it suffices to observe that the sequences $\{a_i\}$ and $\{b_i\}$ don't intersect, since $a_i \equiv 1 \pmod{r+2}$, while $b_j \equiv r+1 \pmod{r+2}$.

We are left to prove that for every $1 \le k \le m$, W^π doesn't contain more than $r+1$ *consecutive* integers, where as usual, when $m-p+1 < k \le m$ we interprete *consecutiveness* a little differently.

Notice that $\forall i \in \{1, \ldots, t\}$, the numbers of integers between a_i and a_{i+1} exclusively is $r+1$, which we shall call the *internal distance* between a_i and a_{i+1}. Similarly, the internal distance between b_i and b_{i+1} is also $r+1$. Now, consider the first sliding window W_1^π of size p of π :

$$\pi = \overbrace{\underbrace{a_{t+1}, a_t, \ldots a_1, c_{m-2.(t+1)}, \ldots\ldots}_{W_1^\pi \longrightarrow}, \ldots, c_1}^{m-(t+1)\ numbers}, b_{t+1}, b_t, \ldots b_1$$

This window can not contain any b_i's, since from (3.8) we have $|W_1^\pi| = p = m-q \le m-(t+1)$. As we have noticed earlier, the internal distance between b_i and b_{i+1} is $r+1$ for all $i \in \{1, 2, \ldots t\}$, thus it is easy to see that $c_1^\pi \le r+1$. We will prove that this property holds for all sliding windows W_k^π by induction on k, where $1 \le k \le m-p+1$, namely $\forall k \in \{1, 2, \ldots m-p+1\}, c_k^\pi \le r+1$.

- Base case : $c_1^\pi \le r+1$ as discussed.
- Suppose we have $c_k^\pi \le r+1$ for some $1 \le k < m-p+1$.
- Let's look at W_{k+1}^π. If W_{k+1}^π does not contain any element of B then $c_{k+1}^\pi \le r+1$ holds trivially. Otherwise, suppose W_{k+1}^π does contain some set of more than $r+1$ consecutive integers. Let i be the least integer such that $b_i \in W_{k+1}^\pi$. It

is clear that W_k^π doesn't contains b_i, because intuitively we have just moved W_k^π one step to the right of π and added b_i into W_k^π to obtain W_{k+1}^π. The only way for W_{k+1}^π to contain a set of more than $r+1$ consecutive integers is when this set contains b_i, since by induction hypothesis, just before this point, $c_k^\pi \leq r+1$. Additionally, since the size of W_{k+1}^π is p, it's easy to see that if $b_i \in W_{k+1}^\pi$ then $a_j \notin W_{k+1}^\pi, \forall j \geq i$.

However, if b_i is contained in some set X of more than $r+1$ consecutive integers, then X has to either contain a_i or a_{i+1} because $a_i < b_i < a_{i+1}$ and the internal distance between a_i and a_{i+1} is $r+1$. Contradiction !

So W_{k+1}^π does not contain any set of more than $r+1$ consecutive integers. In other words, $c_{k+1}^\pi \leq r+1$. □

We also have to show that

$$c_k^\pi \leq r+1, \text{ when } m-p+1 < k \leq m \tag{3.9}$$

- If $r=1$, then $m = 3t+2$; thus, $2(t+1) = m-t \geq p+1$. Consequently, none of $W_k^\pi, m-p+1 < k \leq m$ contains both $a_1 = 1$ and $b_{t+1} = m$. (3.9) follows trivially.
- If $r > 1$, then $c_1 = 2$ and $c_{m-2(t+1)} = m-1$. Moreover, clearly $c_{m-2(t+1)} \notin \{\pi_k, \dots \pi_m\}$ and $c_1 \notin \{\pi_1, \dots \pi_{p+k-m-1}\}$ whenever $m-p+1 < k \leq m$. Thus, for these values of k, c_k^π is at most 2, which is at most $r+1$, which completes our proof. □

Example: Let $m = 17$, $p = 9$. Thus, $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 = \left\lfloor \frac{9}{17-9+1} \right\rfloor + 1 = 2$. $r = 1$ and $17 = 3.5+2$, so $t' = 2 = r+1$ in this case. The sequences A and B are as follows. $A = 1, 4, 7, 10, 13, 16$. $B = 2, 5, 8, 11, 14, 17$. And lastly, applying our scheme gives us permutation

$$\pi = \begin{pmatrix} 16 & 13 & 10 & 7 & 4 & 1 & 15 & 12 & 9 & 6 & 3 & 17 & 14 & 11 & 8 & 5 & 2 \end{pmatrix}$$

Case (ii) : $0 \leq t' < r+1$.

We have

$$\begin{aligned} m &= p+q \\ \Rightarrow \quad (r+2).t+t' &= (q+1).r+r'+q \\ &= q.(r+2)+r+r'-q \\ \Rightarrow \quad (r+2).t &= q.(r+2)+(r'-q)+r-t' \end{aligned}$$

Moreover, since $r' \leq q$, we get

$$\begin{array}{rrcl} & (r+2).t & \leq & q(r+2)+r-t' \\ \Rightarrow & t & \leq & q+\frac{r-t'}{r+2} \end{array}$$

t and q are natural numbers, so

$$t \leq q \tag{3.10}$$

Similar to the case (i), we define 2 special arrays $A = \{a_i\}_1^t$ and $B = \{b_i\}_1^t$ as follows.

$$\begin{array}{rcll} a_i & = & t'+1+(i-1).(r+2) & 1 \leq i \leq t \\ b_i & = & i.(r+2) & 1 \leq i \leq t \end{array}$$

It is obvious that $1 \leq a_i \leq m$ and $1 \leq b_i \leq m$, thus A and B are subsequences of S. $A \cap B = \emptyset$ since $(r+2)$ divides a_i and does not divide b_j for all i and j. Let $C = \{c_i\}_1^{m-2t}$ be the increasing sequence of numbers in S but not in A and B. If we write numbers $1, 2, \ldots m$ in increasing order, this time we have the following picture :

$$\begin{array}{ccccccccccccccc} & & & & b_1 & & & & & & b_t & & \\ 1 & .. & t'+1 & .. & r+2 & .... & t'+1+(t-1)(r+2) & .. & t(r+2) & .. & m \\ & & a_1 & & & & a_t & & & & \end{array}$$

and the rest of the numbers (the dots in the picture) contribute sequence C. π in one line notation is as follows.

$$\pi = (\underbrace{a_t, a_{t-1}, \ldots a_1}_{A}, \underbrace{c_{m-2t}, \ldots\ldots\ldots\ldots, c_1}_{C=S-A-B} \underbrace{b_t, b_{t-1}, \ldots b_1}_{B})$$

The proof goes almost the same as in case (i). Due to $t \leq q$, W_1^π does not contain any of the b_i's. Straightforward induction shows that when $1 \leq k \leq m-p+1$, none of W_k^π contain more than $r+1$ consecutive integers. On the other hand, when $m-p+1 < k \leq m$, we always have

$$c_1 \notin \{\pi_1, \ldots, \pi_{p+k-m-1}\} \text{ and } c_{m-2t} \notin \{\pi_k, \ldots, \pi_m\}$$

If $t' = 0$, then $c_1 = 2$ and $c_{m-2t} = m-1$, so $c_k^\pi \leq 2 \leq r+1$. If $t' > 0$, then $c_1 = 1$ and $c_{m-2t} = m$, so $c_k^\pi = 0$. $\square$

For example, let $m = 17, p = 12$. $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 = \left\lfloor \frac{12}{17-12+1} \right\rfloor + 1 = 3$. $r = 2$ and $17 = 4.4+1$, so $t' = 1 < r+1$ in this case. The sequences A and B are as follows. $A = 4, 8, 12, 16$. $B = 2, 6, 10, 14$. Applying our scheme gives us permutation

$$S' = (16\ 12\ 8\ 4\ 17\ 15\ 13\ 11\ 9\ 7\ 5\ 3\ 1\ 14\ 10\ 6\ 2)$$

3.3 Summary and Benefits of the Bounded Error Case

The following theorem summarizes our work on the deterministic cases.

Theorem 1 *If p and m are both determined, then*

- $k_0 = 0$ *when* $p = 0$ *and* $k_0 = m$ *when* $p \geq m$
- $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$ *when* $0 < p < m$.

Proof: since if $0 < p \leq \frac{m}{2}$ then $\left\lfloor \frac{p}{m-p+1} \right\rfloor = 0$, this is immediate from the preceding lemmas and remark. Also note that if the desired k_0 is given, these formulas allow us to find the minimum buffer size m_0 to achieve k_0.

Algorithm $calculatePermutation(m, p)$ is a permutation generator which generates permutation π with $C^{\pi} = k_0$ on input m and p. Notice that it takes only linear time.

calculatePermutation(m, p)

```
if p ≤ 0 or p ≥ m then
  output the identity permutation
end if
if p ≤ m/2 then
  M ← {j, p ≤ j ≤ m/2 ∧ gcd(m, j) = 1}
  if M ≠ ∅ then
    p' ← min{j, j ∈ M}
    for i ← 1 to m do
      π(i) ← ((i − 1)p' mod m) + 1
    end for
  else
    ** M = ∅, m must be even **
    p' = m/2
    for i ← 1 to m do
      π(i) ← p'.(i mod 2) + ⌈i/2⌉
    end for
  end if
else
  q ← m − p
  r ← ⌊p/(q+1)⌋
  t ← ⌊m/(r+2)⌋
  t' ← m mod (r + 2)
  if t' = r + 1 then
    for i ← 1 to t + 1 do
      a_i ← 1 + (i − 1).(r + 2)
      b_i ← (r + 1) + (i − 1).(r + 2)
    end for
    C ← {1, 2, ... m} − {a_i} − {b_i}
    C is extracted in increasing order.
    for i ← 1 to t + 1 do
      π(a_{t+2−i}) ← i
    end for
    for i ← t + 2 to m − (t + 1) do
      π(c_{m−i−t}) ← i
    end for
    for i ← m − t to m do
      π(b_{m−i+1}) ← i
    end for
  else
    ** i.e. 0 ≤ t' ≤ r **
    for i ← 1 to t do
      a_i ← t' + 1 + (i − 1).(r + 2)
      b_i ← i.(r + 2)
    end for
    C ← {1, 2, ... m} − {a_i} − {b_i}
```

C is extracted in increasing order.
for $i \leftarrow 1$ to t **do**
  $\pi(a_{t+1-i}) \leftarrow i$
end for
for $i \leftarrow t+1$ to $m-t$ **do**
  $\pi(c_{m-i-t+1}) \leftarrow i$
end for
for $i \leftarrow m-t+1$ to m **do**
  $\pi(b_{m-i+1}) \leftarrow i$
end for
end if
end if

Benefits of solving the bounded error case

The assumption that p is known can be envisioned in future networks where some sort of QoS guarantees are provided, such as ATM, Internet2, etc. . More importantly, it gives us a rigid background to solve the unbounded error case.

Chapter 4

Unbounded Network Error Case

4.1 Feedback based permutation adjustment protocol

Our protocol is a simple feedback based protocol. Some CM systems use TCP/IP for communication [HK92]. But it has been shown in [Smi94] that CM applications based on TCP are unstable when the real time bandwidth requirements fall below available bandwidth. Thus in this protocol, we use the UDP communication model (like [Smi, SRY93]). We dynamically use the solution provided in the deterministic cases as a mechanism for the non-deterministic scenario presented here. We assume that m, the buffer size, is known in advance by both client and server. This can also be part of a initial negotiation.

At the server side, a buffer of size m is kept. Server permutes frames (actually frame indices) based on current set of parameters, then initiates transmission of the frames in the buffer. Server changes the permutation scheme based on client's feedback. The permutation scheme changes only at the start of the next buffer of frames.

At the client side, the client waits for a period of $m/frameRate$ (time needed for the client's buffer to be filled up) and calculates consecutive network loss for this buffer window. The client keeps track of the previous window's estimated network consecutive loss and sends its next estimation back to the server. It sends feedback (ACK) in a UDP packet. Note that ACK packet is also given a sequence number so that out of order ACK packets will be ignored. The server makes decision based on the maximum sequence numbered ACK.

Given a buffer of size m, initially the server assumes the average case where $p = \left\lfloor \frac{m}{2} \right\rfloor$. Denote p_i as the actual consecutive network loss, and p_i^* as the estimated network loss in the i^{th} window. We use exponential averaging to estimate next loss. Suppose we are currently at the n^{th} window, p_n^* is determined by

$$p_n^* = \left\lceil \alpha . p_n + (1 - \alpha) . p_{n-1}^* \right\rceil$$

In this experiment, we have picked $\alpha = \frac{1}{2}$. This value turns out to work just fine, as shown in chapter 5. Whether or not there exists an optimal value for α is subject to further investigation. Basically, α measures how much weight we would like to give to the current network status. The larger α is, the less weight we give to the history of network behavior. p_n^* is rounded up because we want to assume the worse error.

4.2 Illustration of the protocol

Figure 4.1 illustrate an example of how client and server interact. $< j, \pi_i >$ is the time where server sends the i^{th} frame of the j^{th} buffer window. ACK_j containing the estimated p_j^* sent back by the client. By the time server gets ACK_j, it could be in the $(k-1)^{th}$ buffer window. So, it uses ACK_j for the k^{th} buffer window. Lastly, ACK_{j-1} is lost, so we have not used it for transmission of any of the buffer window subsequently.

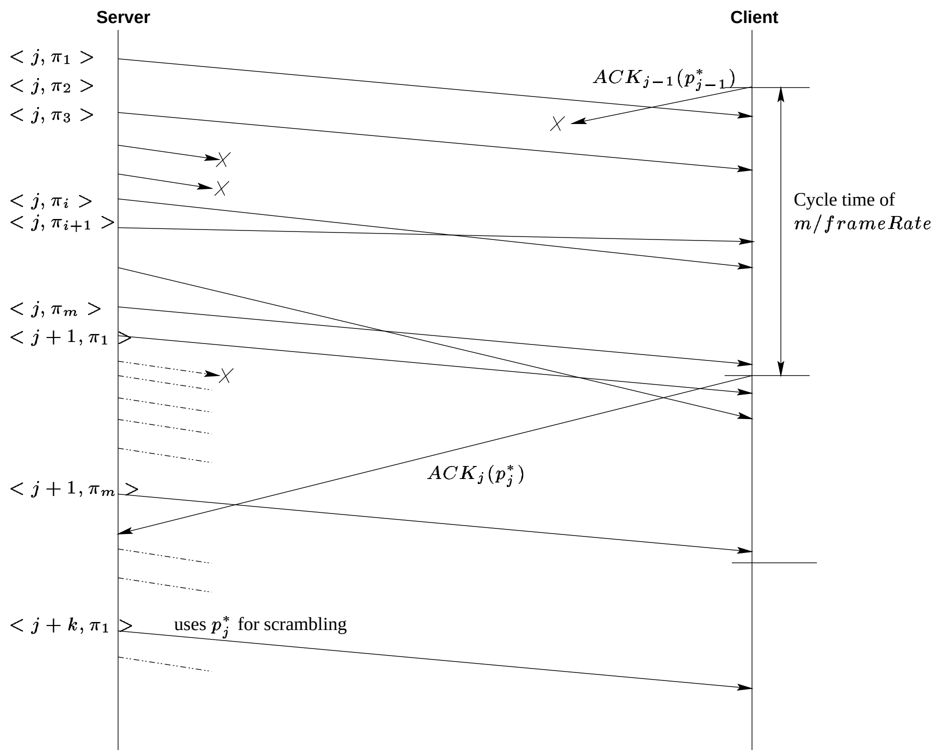

Figure 4.1: A sample session

Chapter 5

Experimental Evaluation

The following two sections presents the evaluation of our scheme in two scenarios. In one case the protocol described in section 4.1 has been implemented and tested over a long haul network. In the second case, we use a data set extracted from a real-time application such as *Internet Phone* and simulate our protocol. We show the reduction in CLF in both the cases. Our protocol has smoothed out CLF to be within the range of perceptually acceptable tolerance. Also, it adapts quite well with abnormality in network loss pattern. Moreover, *almost all* of CLF values are within the range of perceptual tolerance (see section 2.1. Thus this approach of of using End User QoS as a direct means to control Media Delivery shows a lot of promise. There are a number of extensions to the protocol which we have been and are currently looking into. These are briefly discussed in chapter 6;

5.1 Video Experiment : Actual Media Delivery over a Long Haul Network

We have conducted experiments of sending two MJPEG video clips over LAN and WAN. Due to limited space, only the result of WAN is shown here. However, the behavior of our protocol is the same in both cases. We transfered data from a UltraSparc 1 (rawana.cs.umn.edu) in Computer Science department, University of Minnesota to another SunSparc (lombok.cs.uwm.edu) in Computer Science department, University of Wisconsin, Milwaukee [1]. The experiment was conducted at 9:45am when network traffic is expected to be average. Both clips have resolution 512×384. Clip 1 frame sizes varies from 5276 to 36364 bytes with 9544 bytes as the median, 10845 bytes as the mean and 4450.7 is the standard variation. These numbers for clip 2 respectively are 5072, 34408, 10282, 10916 and 3642.8. Clip 1 contains 2607 frames and clip 2 contains 1736 frames. Our buffer window is of size 50. Three times "traceroute" [2] told us that the packets typically go through 14 hops in between. A sample traceroute session is as follows.

[1] Thanks to Mr. Thanh C. Nguyen at the Department of Computer Science, University of Wisconsin, Milwaukee for helping us in conducting this experiment

[2] Thanks to Mr. Luan V. Nguyen by the time was a system staff at the Department of Computer Science, University of Minnesota, for providing us with this traceroute result

```
 1  eecscix.router.umn.edu (160.94.148.254)  2 ms  1 ms  1 ms
 2  tc8x.router.umn.edu (128.101.192.254)  23 ms  4 ms  3 ms
 3  tc0x.router.umn.edu (128.101.120.254)  6 ms  1 ms  1 ms
 4  t3-gw.mixnet.net (198.174.96.5)  1 ms  1 ms  1 ms
 5  border5-hssi1-0.Chicago.cw.net (204.70.186.5)  11 ms  11 ms  29 ms
 6  core2-fddi-0.Chicago.cw.net (204.70.185.49)  11 ms  11 ms  11 ms
 7  core2-hssi-3.WillowSprings.cw.net (204.70.1.225)  13 ms  13 ms  15 ms
 8  core3.WillowSprings.cw.net (204.70.4.25)  310 ms  52 ms  123 ms
 9  * ameritech-nap.WillowSprings.cw.net (204.70.1.198)  245 ms  35 ms
10  aads.nap.net (198.32.130.39)  18 ms  18 ms  21 ms
11  r-milwaukee-hub-a9-0-21.wiscnet.net (207.112.247.5)  25 ms  22 ms  27 ms
12  205.213.126.39 (205.213.126.39)  19 ms  20 ms  23 ms
13  miller.cs.uwm.edu (129.89.139.22)  24 ms  25 ms  21 ms
14  lombok.cs.uwm.edu (129.89.142.52)  24 ms *  25 ms
```

Figure 5.1 shows the result. As can be seen from the figure, our scheme has done quite well smoothing network consecutive losses. In a few cases our CLF is 1 higher (clip 2) but that was due to rapid changes in network loss behavior and it is expected. Most of the time CLF is well below and also within tolerable perceptual limits (see section 2.1).

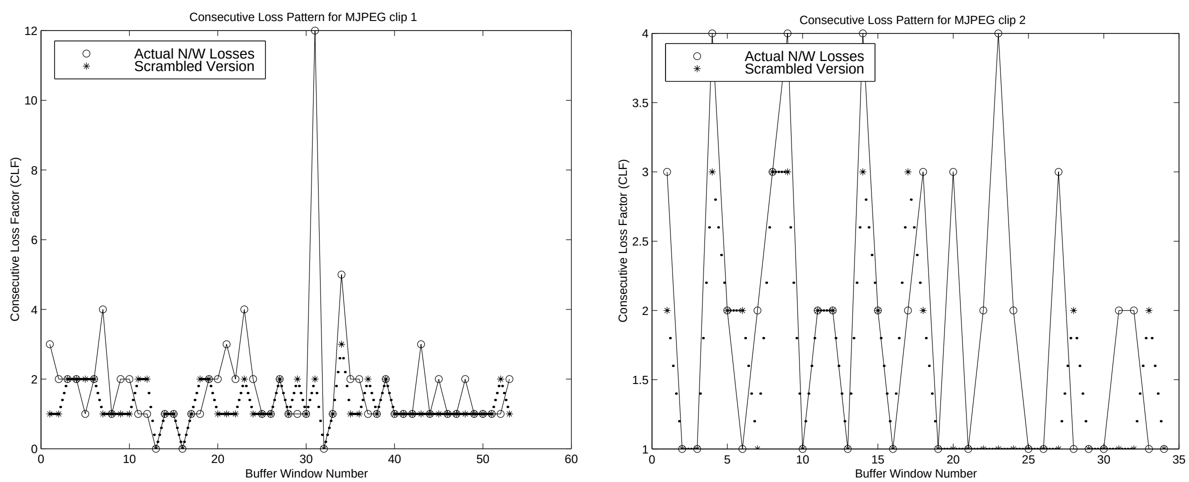

Figure 5.1: Performance of our protocol when transmitting video over long haul network

5.2 Simulation: Using data from a real time application like Internet Phone

The data [3] was collected for an *Internet Voice or Voice on Networks (VON)* application. The server is *vermouth.ee.umanitoba.ca (Canada)* and the Client is *rawana.cs.umn.edu (Minnesota, USA)*. *vermouth* and *rawana* is a SUN UltraSparc 1, running Solaris V2.6 and V2.5 respectively. Each host is on a 10 Mbps Ethernet (LAN). The transmission is over the

[3] Thanks to Mr. Difu Su, Computer Science Department, University of Minnesota, for providing us with the data set

Internet and the data set was collected on a Saturday, from 10 am to 2 pm. The two files presented here are of voice packets of sizes 160 and 480 bytes. As can be seen from figure 5.2 the actual CLF's of network losses are varying while the CLF based on our protocol always has lower CLF (in this case CLF=1, implying no consecutive losses).

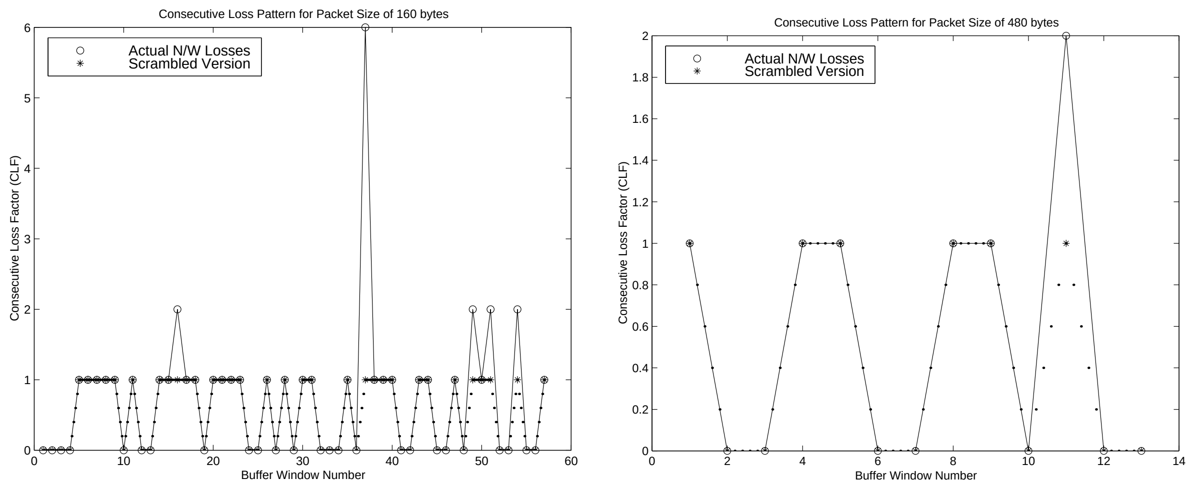

Figure 5.2: Performance of our protocol for a real-time application such as Internet Phone

Chapter 6

Conclusions

In this report we have addressed the problem of handling bursty losses in continuous media transmission. We formulated the problem in terms of a number of parameters including user QoS requirements, sender resource availability, and network loss behavior. We introduced the idea of *error spreading*, which takes spots of concentrated bursty losses and spreads it evenly over the entire stream. This makes the stream more acceptable from a perceptual viewpoint [WSNF97]. Our experiments over the Internet show that the scheme is quite effective.

Our ongoing work is addressing a number of issues. First, we want to develop an analytical formulation for the unbounded network error case. Second, we want to develop metrics which help us to choose which permutation to be used when the first level parameter (k_0) is not enough to break the tie. Finally, we want to extend this idea to groups of synchronized streams.

An extension of this work has already been done and is presented in Srivatsan's Master's report. In the report, he extends this idea to hanlde streams with inter-frame dependency such as MPEG and shows that error spreading could be used in conjunction with other existing protocols without changing the underlining protocol.

From a Combinatorial point of view, it would be interesting to answer the following question : *given* m *and* p*, how many* $\pi \in S_m$ *are there such that* $C^{\pi} = k_0$ *?* The answer to this is probably not so difficult to find, but it will be tedious with various boundary cases involved.

BIBLIOGRAPHY

[ABLS94] A. Albanese, J. Blomer, J. Edmonds M. Luby, and M. Sudan. Priority Encoding Transmission. *Proceeding of 35th Annual Symposium on Foundations of Computer Sciences, Santa Fe, New Mexico*, Nov 20-22 1994.

[AFKN94] R.T. Aptekar, J.A. Fisher, V. Kisimov, and H. Nieshlos. Distributed Multimedia:User Perception and Dynamic QoS. In *SPIE* , volume 2188. SPIE, 1994.

[BG] J. C. Bolot and A. V. Garcia. The case for FEC-based error control for packet audio in the Internet. *To appear in ACM Multimedia Systems*.

[Bol93] J. C. Bolot. End-to-End Packet Delay and Loss Behaviour in the Internet. *ACM SIGCOMM '93, Ithaca, NY*, 1993.

[BT96] J. C. Bolot and Turletti. Adaptive Error Control for Packet Video in the Internet. *Proceedings IEEE International Conference on Image Processing, ICIP'96, Laussane, Switzerland*, Sep 1996.

[BT98] J. C. Bolot and T. Turletti. Experience with Control Mechanisms for Packet Video. *ACM Communication Review, Jan '98*, 28(1), 1998.

[BTW94] J. C. Bolot, T. Turletti, and I. Wakeman. Scalable Feedback Control for Multicast Video Distribution in the Internet. *ACM SIGCOMM'94, London, UK*, pages 58–67, 1994.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug 1993.

[GJS96] W. C. Geng, F. Jahanian, and S. Sechrest. An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video. *ACM/Springer-Verlag Multimedia Systems Journal*, 1996.

[HBK98] R. Hasimoto-Beltran and A. Khokhar. Pixel Level Interleaving Schemes for Robust Image Communication. *17th IEEE Symposium on Reliable Distributed Systems, '98*, pages 455–460, Oct 20-23 1998.

[HK92] B. Heinrichs and R. Karabek. Tcp/ip supporting real-time applications: The predictive delay control algorithm. In *Second International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 45–47, 1992.

[MJV96] S. McCane, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. *ACM SIGCOMM '96*, pages 117–130, Sep. 1996.

[Pax97] V. Paxson. End-to-End Internet Packet Dynamics. *ACM SIGCOMM '97, Cannes, France*, 1997.

[PFTK98] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Evaluation. *ACM SIGCOMM '98, Vancouver, CA*, Sep 1998.

[RR93] S. Ramanathan and P. Venkat Rangan. Feedback Techniques for Intra-Media Continuaity and Inter-Media Synchronization in Distributed Multimedia Systems. *The Computer Journal*, 36(1):19 – 33, 1993.

[RTP94] S. V. Raghavan, Satish K. Tripathi, and B. Prabhakaran. On QoS Parameters and Services for MultiMedia Applications. Technical Report 3167, Department of Computer Science, University of Maryland, College Park, MD 20742 USA, 1994.

[SB96] Ralf Steinmetz and Gerold Blakowski. A media synchronization survey: Reference model, specification and case studies. *IEEE Journal on Selected Areas in Communication*, 14(1):5–35, 1996.

[SCFM94] H. Schulzrinne, S. Casner, R. Frederick, and S. McCane. RTP: A Transport Protocol for Real-Time Applications . *RFC 1889*, 1994.

[SJBG98] S. Sedigh, J. Joshi, A. Bashandy, and A. Ghafoor. Evaluation of Filtering Mechanisms for MPEG Video Communications. *17th IEEE Symposium on Reliable Distributed Systems, '98*, pages 449–454, Oct 20-23 1998.

[Smi] Brian Smith. Cyclic-UDP: A Priority Driven Best-Effort Protocol. http://www.cs.cornell.edu/Info/Faculty/Brian_Smith.html/Publications.

[Smi94] Brian Christopher Smith. *Implementation Techniques for Continuous Media Systems and Applications*. PhD thesis, University of California, Berkeley, 1994.

[SRY93] Brian Smith, Larry Rowe, and S Yen. A Tcl/Tk Continuous Media Player. In *Proceedings Tcl 1993 Workshop*, June 1993.

[Tow93] Don Towsley. Providing quality of service in packet switched networks. In L. Donatiello and R. Nelson, editors, *Performance Evaluation of Computer Communication Systems*, pages 560–586. Springer Verlag, 1993.

[WL98] B.W. Wah and D. Lin. Transformation-based Reconstruction for Audio Transmissions over the Internet. *17th IEEE Symposium on Reliable Distributed Systems, '98*, pages 211–217, Oct 20-23 1998.

[WS96] Duminda Wijesekera and Jaideep Srivastava. Quality of Service (QoS) Metrics for Continuous Media. *Multimedia Tools and Applications*, 3(1):127–166, September 1996.

[WSNF97] Duminda Wijesekera, Jaideep Srivastava, Anil Nerode, and Mark Foresti. Experimental Evaluation of Loss Perception in Continuous Media. *to appear in ACM - Springer Verlag*, July 1997.

[WVP+98] Duminda Wijesekera, Srivatsan Varadarajan, Shwetal Parikh, Jaideep Srivastava, and Anil Nerode. Performance evaluation of media losses in the continuous media toolkit. In *International Workshop on Multimedia Software Engineering , MSE'98, Kyoto, Japan*, 1998.

[YCV96] J. H. Yao, Y. M. Chen, and T. Verma. MPEG-Based Audio-on-demand experiment for the Internet. *Internetworking '96, Nara, Japan*, 1996.

[ZDE+93] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Network*, 5:8 – 18, 1993.

[ZNAT99] Z. L. Zhang, S. Nelakuditi, R. Aggarwal, and R. P. Tsang. Efficient Selective Frame-Discard Algorithms for Stored Video Delivery across resource Constrained Networks. *To Appear in IEEE INFOCOM '99*, 1999.

[ZSKT96] Z. L. Zhang, J. D. Salehi, J. F. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. *ACM SIGMETRICS*, May 1996.