# Retransmission in Distributed Media Streaming

Lin Ma
School of Computing
National University of Singapore
malin@comp.nus.edu.sg

Wei Tsang Ooi
School of Computing
National University of Singapore
ooiwt@comp.nus.edu.sg

## ABSTRACT

This paper considers the use of Automatic Request Request (ARQ) schemes in distributed media streaming. We analytically model three different ARQ schemes and derive effective packet loss rate and burst length of these schemes. Our model is verified through simulations and experiments over wide-area network. Our results show that retransmitting lost packet from senders other than the one who lost the packet could reduce effective loss rate and burst length. We also find that ARQ with a dedicated retransmitter outperforms other schemes if the retransmitter is chosen appropriately.

## Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols (Applications)

## General Terms

Experimentation, Performance, Measurement

## Keywords

Distributed Streaming, ARQ, Peer to Peer

## 1. INTRODUCTION

Traditionally, media on demand service uses a client-server model, where a single server is responsible for streaming prerecorded media files to the clients. Such single sender model suffers from two major drawbacks: (i) it is hard to scale the server to large number of clients, and (ii) the server is a single point of failure.

Distributed media streaming, or multiple-source streaming, has been proposed in recent years as a robust and scalable solution for streaming [11, 6, 10, 4]. Under this model, multiple servers storing the same media file collaboratively stream the media data to a client. Network bandwidth can be aggregated from these servers, and the network load is shared among them. Furthermore, with proper coding schemes, such as multiple description coding [3], failure or congestion at one server will not disrupt the service, but will only degrade received media quality at the client service, since the client can continue to receive data from the other servers.

Distributed media streaming presents many new and interesting research problems. Problems such as deciding which senders to use [4], or which packets should be sent by which senders have been studied recently [10]. However, the issue of error recovery, despite its importance as a fundamental problem in media streaming, has not been seriously looked into. This paper is part of our work that studies error recovery mechanisms in the presences of multiple senders. We analyze their effectiveness mathematically and experimentally, and suggest an effective error recovery scheme for use under the distributed streaming model.

The main contributions of this paper are as follows. We present a preliminary study of different ARQ schemes for distributed media streaming. We propose two variations of traditional ARQ schemes for distributed streaming, and show their effectiveness by mathematical analysis, simulations and experiments on wide-area network through PlanetLab. Our studies show that using an appropriately chosen dedicated sender to retransmit lost packets can reduce effective packet lost rate and reduce average packet loss burst length.

The rest of this paper is organized as follows. Section 2 presents existing works on distributed media streaming and related works on ARQ. Section 3 presents our model and assumptions. In Section 4, we describe the proposed ARQ schemes, and present their mathematical analysis in Section 5. In Section 6, we present simulation and experimental results. Finally, we conclude and introduce our extensions to this work in Section 7.

## 2. RELATED WORK

There are few existing work on distributed media streaming that incorporates error recovery schemes. Nguyen et. al. [7] and Golubchik et. al. [2] proposed distributed streaming model incorporated with Forward Error Correction (FEC). Rajaie and Ortega [10] implicitly apply ARQ in distributed streaming system for layered media. However, since their work do not focus on error recovery, the effectiveness of ARQ on loss recovery is not explored.

On the other hand, there is a large literature on using ARQ to recover packet loss using the traditional, single sender model.


*NOSSDAV'05,* June 13–14, 2005, Stevenson, Washington, USA.

Perkins et. al. discuss sender based recovery in multicast [9]. FEC and ARQ are both discussed. The authors suggest that ARQ works well in a low lossy environment and FEC performs better in non-interactive streaming with less overhead. While the survey points out the high overhead for ARQ, we must note that it is mainly due to the nature of multicast – whenever a packet is retransmitted, it is resent to the whole group. This overhead does not apply in distributed media streaming.

Papadopoulos and Parulkar's work [8] is one of the earliest that applies selective ARQ in continuous media streaming. As long as the round trip time is smaller than the time before the lost packet is to be played out, retransmission reduces loss drastically. Their evaluation reveals that retransmission copes well with bursty loss. Our work is a natural extension of this work for distributed streaming scenario.

Several recent works have studied the use of TCP protocol for streaming media (e.g. see [5]). Although our work recovers lost data using retransmission just as TCP, we focus on selection of different senders for retransmission, which cannot be achieved using TCP.

# 3. MODEL AND ASSUMPTIONS

We now present our model for distributed streaming and the assumptions that we made.

In this paper, we target non-interactive streaming of prerecorded media data. Thus, maintaining low end-to-end latency is not crucial. We assume that the client buffers sufficient amount of data in the playout buffer to allow us to use ARQ for error recovery.

In the discussion of ARQ schemes, we assume that only one retransmission is performed for every loss to simplify our analysis. This simplification is fair and is sufficient to show their effectiveness in loss recovery. We also assume that the time units between sending the lost packet and re-sending the ARQ packet is $\delta$. The value of $\delta$ is decided by the round trip time (RTT) of the channels. In order to be fair to all schemes, we assume that all senders have the same RTT in our analytical model.

We use a simple scheme to allocate data packets among the senders. We assign the sequence of data packets to the sender in a round robin manner. This packet allocation scheme, though simple, is effective as it minimizes packet loss burst length when any one of the senders fails or experiences congestion.

We assume that most of the time the senders send packets at a constant rate, and denote the time needed to send one packet as one time unit. This assumption allows us to use a two-state Gilbert model to model the state of the network, assuming that the state transition in the model occurs at discrete time unit. However, for senders sending both data packets and retransmission packets, we assume the retransmission packets do not delay data packets. When retransmission happens, bit rate is increased to send a data packet and a retransmission packet in one time unit. This simplification removes the cumulative delay of data packet caused by retransmission.

We further assume independent network channels among the senders. This assumption is most likely not valid on the Internet as many paths share the same links. We are currently extending our work to handle shared links between channels.

# 4. ARQ SCHEMES

We now present the different ARQ schemes that can be used under a distributed streaming model. ARQ performs error recovery through retransmission. When the receiver detects a packet loss, a request for that packet is sent back to the sender via a feedback channel and the packet is retransmitted by the sender. For simplicity only one retransmission is performed for every packet loss in this paper.

As the distributed streaming paradigm uses multiple senders to collaboratively send the media data, the receiver may choose to *retransmit packets using a different sender than the original sender whose data packet is lost.* This simple idea is the key to improve the effectiveness of ARQ schemes. Based on this idea, we propose an ARQ scheme for distributed streaming called *ARQ-D*. ARQ-D uses one of the sender as a dedicated retransmitter. Whenever the receiver detects a packet loss, an ARQ request is sent to the dedicated retransmitter, regardless of who the sender of the lost packet is.

Intuitively, ARQ-D is superior in performance than simple ARQ schemes. Since a channel that experiences packet loss is likely to remain lossy for a while, using another sender for retransmission will reduce the probability that the retransmitted packet is lost as well.

To evaluate the performances of ARQ-D, we term the original ARQ scheme as *ARQ-O*. Under the ARQ-O scheme, a receiver always asks for retransmission from the original sender of the lost packet.

We also propose an alternate scheme for ARQ called *ARQ-RR*, as a trade-off between ARQ-O and ARQ-D. Under this scheme, there is no dedicated retransmitter. The receiver, however, asks for retransmission from a different sender each time a packet is lost. The receiver rotates among the senders in a round-robin manner when requesting for retransmission.

# 5. MATHEMATICAL ANALYSIS

We now model these different schemes analytically and analyze their effective lost rate and expected burst length.

Let $n$ be the number of senders. Treating each sender as an independent channel, we denote the set of senders as $c_1, c_2, ..., c_n$. Each channel $c_i$ is modeled using a Gilbert model with parameter $p_i$ and $q_i$, where $p_i$ is the probability of transition from good state (denoted as 0) to bad state (denoted as 1) and $q_i$ is the probability of transition from bad state to good state (see Figure 1).

We introduce some additional notations as follows. Let $L_i$ be the average packet loss rate of channel $c_i$. $L_i$ can be computed from the Gilbert model and is given as $p_i/(p_i+q_i)$.

Given the Gilbert model, we can also compute $P_i(\delta)$, which is the probability of transition from a good state to bad state for channel $c_i$ after $\delta$ time units (or, equivalently, after sending $\delta$ packets). Similarly, we can compute $Q_i(\delta)$ as the probability that a channel $c_i$ goes from a bad state to a good state after $\delta$ time units. $P_i(\delta)$ and $Q_i(\delta)$ are analogous to $p_i$ and $q_i$ in Gilbert model in terms of good-to-bad and bad-to-good state transitions. These two probabilities can be computed as:

$$P_i(\delta) = L_i - L_i(1 - p_i - q_i)^\delta$$
$$Q_i(\delta) = 1 - L_i - (1 - L_i)(1 - p_i - q_i)^\delta$$

We denote the value $1 - P_i(\delta)$ as $\bar{P}_i(\delta)$. This value corresponds to the probability that the state is good after $\delta$

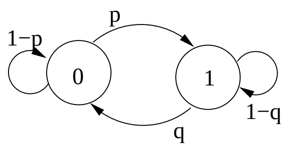


**Figure 1: Gilbert Model**

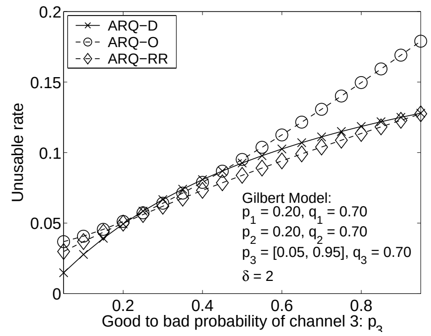


**Figure 2: Unusable Rate vs. $p_3$**

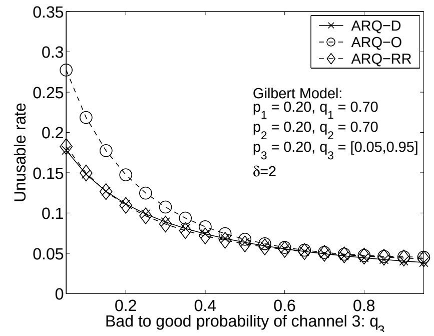


**Figure 3: Unusable Rate vs. $q_3$**

time units, given that the current channel state is good. Similarly, we use $\bar{Q}_i(\delta)$ to denote the probability of channel transiting from bad state to bad state after $\delta$ time units, i.e., $\bar{Q}_i(\delta) = 1 - Q_i(\delta)$.

## 5.1 Effective Loss Rate

Using $L_i$ and $\bar{Q}_i(\delta)$, we can now compute the effective loss rate of distributed streaming under different error recovery schemes. The effective loss rate, or *unusable rate* for short, is the probability that a packet is lost and cannot be recovered. Unusable rate reveals the average quality degradation of the received media.

We now consider the unusable rate of ARQ-based schemes. A packet is unusable if the packet is lost, and the retransmitted packet is lost as well. For simplicity, we only model the case for single retransmission here. The unusable rate is therefore computed as the probability that the data packet is lost, and the retransmitted packet is lost after time $\delta$.

**ARQ-D**: Without loss of generality, let $c_n$ be the dedicated retransmitter channel and the other $n-1$ channels be data channels. The probability that a data packet is lost is given by $\sum_{i=1}^{n-1} L_i/(n-1)$ and the probability that the ARQ packet is lost is $L_n$. The unusable rate of ARQ-D scheme, $V_{ARQ-D}$ is therefore given by

$$V_{ARQ-D} = \frac{L_n}{n-1} \sum_{i=1}^{n-1} L_i$$

**ARQ-O**: Since the data packet is sent on the same channel as retransmitted packet, the loss probability for data packet and retransmitted packet is correlated. We know that $L_i$ is the probability that data packet is lost on channel $c_i$ and $\bar{Q}_i(\delta)$ is the probability that a packet is lost in the same channel after time $\delta$. Assuming that retransmission occurs after time $\delta$, the unusable rate for that channel is therefore $L_i\bar{Q}_i(\delta)$. Averaging over $n$ channels, we have the expected unusable rate as

$$V_{ARQ-O} = \frac{1}{n} \sum_{i=1}^{n} (L_i \bar{Q}_i(\delta))$$

**ARQ-RR**: Under this scheme, the retransmitted packet is sent by different senders. With probability $1/n$, it is sent by the original sender. For a channel $c_i$, the probability that the data packet is lost is $L_i$, and the probability that its retransmitted packet is lost is $(\bar{Q}_i(\delta) + \sum_{j \neq i} L_j)/n$. The unusable rate is therefore given by

$$V_{ARQ-RR} = \frac{1}{n^2} \sum_{i=1}^{n} (L_i(\bar{Q}_i(\delta) + \sum_{j \neq i}^{n} L_j))$$

We have derived the average unusable rate for the three ARQ schemes as a function of Gilbert model's parameters $(p_i, q_i)$. We can now plot these functions.

For simplicity, only the case where $n = 3$ is considered. We vary the condition of channel $c_3$ (dedicated retransmission channel in ARQ-D scheme) and plot the unusable rate in Figure 2 and 3 with $\delta = 2$. Figure 2 shows that when the channel condition of $c_3$ is better then the other two channels, the ARQ-D scheme gives lower unusable rate. This behavior is expected since the probability of successful retransmission is higher in this scenario. As the probability of good-to-bad transition increases for channel $c_3$, the unusable rate for ARQ-D increases and become worse than ARQ-RR. We expected this trend as well, since ARQ-RR rotates among the channels for retransmission and for two out of three retransmissions, it chooses a better quality channel than channel $c_3$. An important observation from this figure is that, using an unrealistically bad channel under the ARQ-D or ARQ-RR schemes would still give lower unusable rate, compared to ARQ-O scheme. Figure 3 shows the results when we vary the bad-to-good transition probability of channel $c_3$. We can see that the burstiness does not differentiate the unusable rate of ARQ-RR and ARQ-D schemes much. Both of these schemes give lower unusable rate than ARQ-O schemes, especially when the channel is bursty.

Figure 4 and 5 plot the unusable rate as we vary the conditions of channel $c_1$. In these plots, we configure channel $c_3$, the dedicated retransmission channel, as a good quality channel with $p_3 = 0.1$ and $q_3 = 0.95$. These figures show that by using a good quality retransmission channel, we can achieve much lower unusable rate if we use ARQ-D scheme compared to ARQ-RR or ARQ-O schemes.

## 5.2 Expected Burst Length

To study the expected packet loss burst length, we further simplify our model to homogeneous channels. In other words, we use the same Gilbert model with parameter $(p, q)$ to model all channels. We also restrict our model to three channels only.

Despite these vast simplifications, the analysis for expected burst length is still quite complex. For each error recovery scheme, there are four cases to consider. A burst of packet loss, or *gap*, of length $m$ always starts with a usable packet,

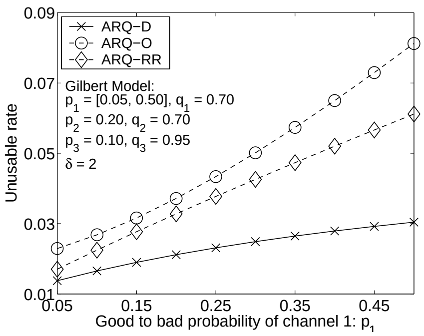


**Figure 4: Unusable Rate vs. $p_1$**

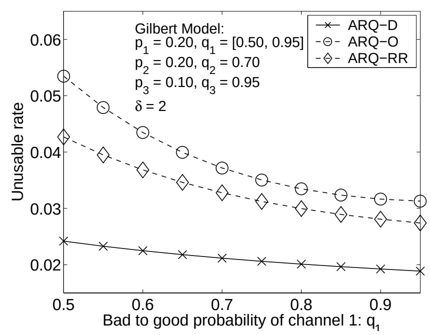


**Figure 5: Unusable Rate vs. $q_1$**

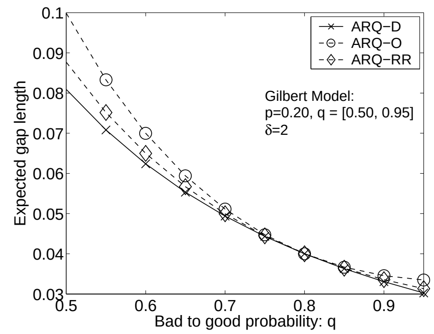


**Figure 6: Expected Gap Length vs. $q$**

followed by $m$ consecutive unusable packets and ends with another usable packet. A usable packet is either delivered, or lost but recovered. Thus, we have to consider the cases where the gap begins and ends with both delivered packets (Case 1), begins with a lost but recovered packets and ends with a delivered packet (Case 2), begins with a delivered packet and ends with a lost but recovered packet (Case 3) and begins and end with both lost but recovered packets (Case 4). We will analyze these four cases separately, and use $\alpha_i(m)$ to denote the probability that the burst length is $m$ for Case $i$. The probability of burst length $m$ occurring is thus $\sum_{i=1}^{4} \alpha_i(m)$.

While deriving the probability of occurrence of a gap of length $m$, we will only explain in details Case 1 in ARQ-D scheme, and list the equations of gap length for ARQ-O and ARQ-RR schemes without further explanations, as the derivation is similar.

**ARQ-D**: For Case 1, with $m = 1$,

$$\alpha_1(1) = L^2(1-L)(1-p)$$

$\alpha_1(1)$ is given as the probability that packet in channel $c_1$ is delivered, $1-L$, and a packet in channel $c_2$ is lost and not recovered, $L^2$, and the next packet in channel $c_1$ is delivered, $1-p$. This argument can be generalized for value of $m$ larger than 1, giving

$$\alpha_1(m) = L^3 p^2 (1-q)^{2m-3},\ m \geq 2$$

The probability for the other cases are given as:

$$\alpha_2(m) = \alpha_3(m) = L^3 q^2 (1-q)^{2m-2}$$
$$\alpha_4(m) = L^3 q^2 (1-q)^{2m-1}$$

**ARQ-O**: Similar to the analysis of ARQ-D, we compute $\alpha_i(m)$ for all four cases.

$$\alpha_1(m) = \begin{cases} (1-L)^2 L\bar{Q}(\delta), & m = 1, \\ (1-L)L^2\bar{Q}(\delta)^2(1-p), & m = 2, \\ L^3\bar{Q}(\delta)^m q^2 (1-q)^{m-3}, & m \geq 3. \end{cases}$$

$$\begin{aligned} \alpha_2(m) &= \alpha_3(m) \\ &= \begin{cases} (1-L)L^2\bar{Q}(\delta)Q(\delta), & m = 1, \\ L^3\bar{Q}(\delta)^m Q(\delta) q (1-q)^{m-2}, & m \geq 2. \end{cases} \\ \alpha_4(m) &= L^3\bar{Q}(\delta)^m Q(\delta)^2 (1-q)^{m-1} \end{aligned}$$

**ARQ-RR**: For ARQ-RR scheme, since the retransmitted packet is sent by the senders, in a round robin manner, we compute the loss rate of the retransmitted packet first. We denote this loss rate as $L'$.

$$L' = \frac{1}{3}(2L + \bar{Q}(\delta))$$

Using derivation similar to previous schemes, we have

$$\alpha_1(m) = \begin{cases} (1-L)^2 LL', & m = 1, \\ (1-L)L^2L'^2(1-p), & m = 2, \\ (1-L)L^2L'^m pq(1-q)^{m-3}, & m \geq 3. \end{cases}$$

$$\begin{aligned} \alpha_2(m) &= \alpha_3(m) \\ &= \begin{cases} (1-L)L^2L'(1-L'), & m = 1, \\ (1-L)L^2L'^m(1-L')p(1-q)^{m-2}, & m \geq 2. \end{cases} \\ \alpha_4(m) &= L^3L'^m(1-L')^2(1-q)^{m-1} \end{aligned}$$

The probability of different gap length is plotted in Figure 6 using the derived expressions with varying bad-to-good transition probability $q$ and $\delta = 2$. We can see that ARQ-D scheme gives shortest expected gap length. We omit the curve that shows the effect of Gilbert parameter $p$ on expected gap length as the differences among the schemes are too small to be interesting.

# 6. EXPERIMENTAL EVALUATION

To evaluate the ARQ schemes and verify our analysis, we implemented an RTP-based distributed streaming system for MP3 audio based on the LIVE.COM[1] media streaming library using the three proposed ARQ schemes for retransmission. We conducted experiments over PlanetLab for realistic network settings, and over our Intranet under controlled network environment.

For each experiment, the system streams a 31.8 second MP3 audio file, consisting of 1224 application data unit (ADU), packetized based on RFC3119 [1] using three senders. Each ADU is approximately 0.4KB, with one packet consists of 2 to 5 ADUs. ADUs are interleaved among the senders so that a lost packet from one sender will not caused consecutive ADUs to be lost. In our experiments, we measure unusable rate of ADUs and burst length of ADUs, as these metrics are more meaningful than unusable rate and burst length of packets.

## 6.1 Experiments over Intranet

[1]http://www.live.com/liveMedia

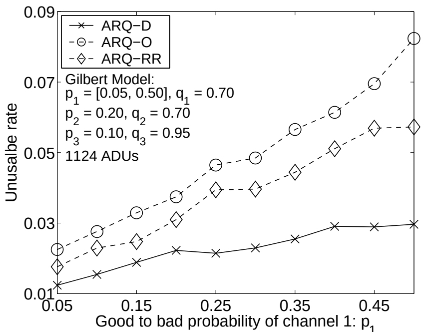


**Figure 7: Unusable rate vs. $p_1$ (Simulations)**

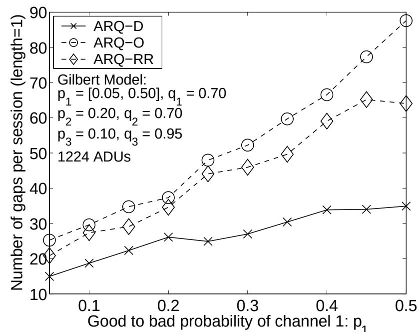


**Figure 8: Effect of $p_1$ on Number of Gaps with Length 1**

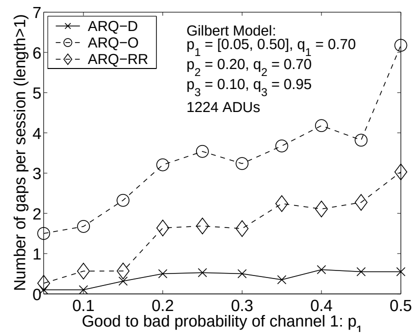


**Figure 9: Effect of $p_1$ on Number of Gaps with Length $> 1$**

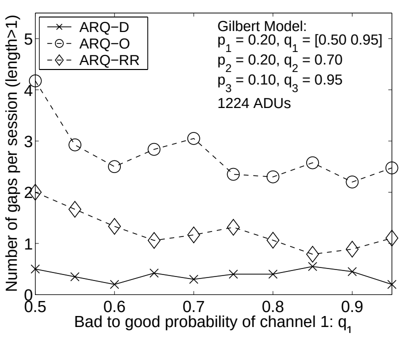


**Figure 10: Effect of $q_1$ on Number of Gaps with Length $> 1$**

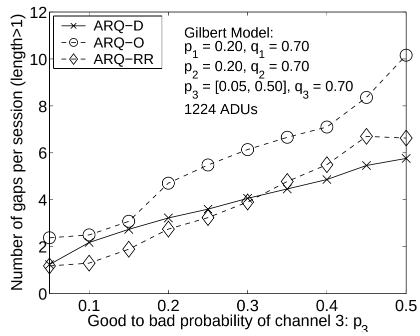


**Figure 11: Effect of $p_3$ on Number of Gaps with Length $> 1$**

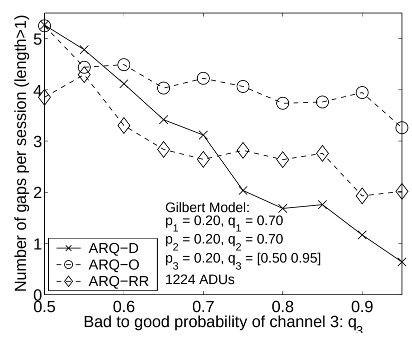


**Figure 12: Effect of $q_3$ on Number of Gaps with Length $> 1$**

We first present our results based on experiments over Intranet using simulated packet loss. Our goal is to further strengthen our observations since the analytical results obtained in previous section is based on simplifying assumptions such as homogeneous channels and fixed $\delta$.

Using the same Gilbert model parameters as in Section 5, we first verified our analytical results. Collected over 20 runs, our simulation results give very similar curves. One such set of curves, which corresponds to Figure 4 is shown in Figure 7.

Next, we study the effect of heterogeneous channels on burst length. We focus mainly on results for bursty loss with length larger than one, as we find that the results for gap length of one follows closely the behavior of the curves for unusable rate (e.g., see Figure 8).

Figure 9 and 10 show the number of gaps in ADU with gap length larger than one. They indicate that ARQ-D has fewer bursty losses compared to ARQ-RR and ARQ-O as we vary the condition of channel $c_1$. A more interesting observation can be found in Figure 11 and 12, which vary the condition of channel $c_3$, the dedicated retransmission channel. We can see in Figure 11 that even when channel $c_3$ is less lossy, using ARQ-D scheme leads to slightly more lengthy ADU gaps than ARQ-RR. The cause of this behavior is that, in our model, ARQ-D uses only two channels for data transmission while ARQ-RR uses all three. Thus, the probability of getting two consecutive losses is higher for ARQ-D. Figure 12 shows that when channel $c_3$ is bursty, ARQ-D can result in most number of ADU gaps. Again, this result can be explained by the fact that ARQ-D uses only two channels for data transmission. When the retransmission channel is bursty, probability of recovering from two consecutive data loss decreases. The number of gaps, however, drops rapidly as channel $c_3$ becomes less bursty.

## 6.2 Experiments over PlanetLab

Besides experiments under controlled environment within our Intranet, we conducted real experiments over PlanetLab, a wide-area test-bed for large scale distributed applications to see how the schemes performed under realistic network conditions. We use three remote senders plus one local receiver[2]. The measured loss rate of the channels are 13.34%, 11.60% and 12.34% respectively for $c_1$, $c_2$ and $c_3$. Due to the unpredictability of network conditions, we increase the number of runs per experiments to 50.

Figure 13 presents the average unusable rate of different error recovery schemes with 90% confidence interval. The PlanetLab test results show that under realistic network conditions, ARQ-D has the lowest unusable rate.

Figure 14 shows the average frequency of single loss and burst loss with length larger than 1, per session. The results from our PlanetLab experiments indicate that ARQ-O can result in long gaps, while ARQ-D achieves least number of gaps. We also observe that the performance of ARQ-RR does not differ much from ARQ-D. This observation suggests that in the case where channel conditions are unknown, ARQ-RR could be a good retransmission scheme. By re-

[2]planetlab2.ie.cuhk.edu.hk ($c_1$), planetlab2.cis.upenn.edu ($c_2$), planet1.cc.gt.atl.ga.us ($c_3$) and soccf-planet-002.comp.nus.edu.sg.

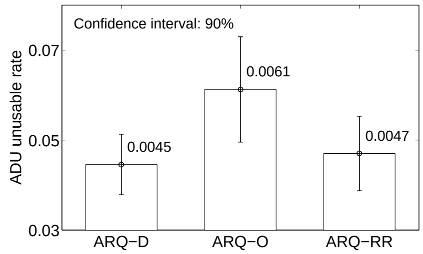


**Figure 13: Unusable rate (PlanetLab)**

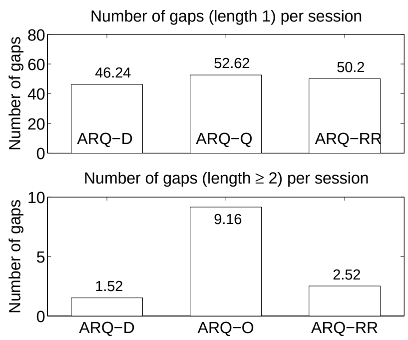


**Figure 14: Gaps per session (PlanetLab)**

questing a different sender for retransmission each time, the receiver experiences average channel conditions in the long run.

# 7. CONCLUSION AND FUTURE WORK

In this paper, we apply different ARQ schemes in distributed streaming system for non-interactive streaming of pre-recorded media. We propose two new schemes, ARQ-D and ARQ-RR, and compare them through (i) mathematical analysis, (ii) Intranet simulations and (iii) wide-area Internet experiments.

We conclude that the ARQ-D scheme, with a good quality sender selected as retransmitter, outperforms the other schemes in general, in terms of unusable rate and loss burst length. We also found that ARQ-RR is a suitable trade-off between ARQ-O and ARQ-D, and could be a good replacement for ARQ-D when channel conditions are unknown. Furthermore, ARQ-RR allows utilizing all three channels for data transmission and achieves better load distributions across all senders.

We are extending our work in a few ways. Firstly, we are studying how traditional Forward Error Correction (FEC) scheme can be used more effectively in distributed streaming environment. The pros and cons of sending FEC packets and data packets in the same channel versus different channels are being investigated. Secondly, we are studying how to adapt the different error recovery schemes under changing network conditions and without the assumption of independent channels. Finally, we plan to take bandwidth into considerations in the study of error recovery schemes.

# 8. REFERENCES

[1] R. Finlayson. A More Loss-Tolerant RTP Payload Format for MP3 Audio. RFC 3119, June 2000.

[2] L. Golubchik, J. C.S. Lui, T. F. Tung, A. L.H. Chow, W.-J. Lee, G. Franceschinis, and C. Anglano. Multi-path Continuous Media Streaming: What are the Benefits? In *Performance Evaluation*, volume 49, pages 429–449, September 2002.

[3] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, September 2001.

[4] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. PROMISE: Peer-to-Peer Media Stream-ing Using Collectcast. In *Proceedings of ACM International Conference on Multimedia (MM'03)*, Berkeley, California, USA, November 2003.

[5] C. Krasic, K. Li, and J. Walpole. The Case for Streaming Multimedia with TCP. In *Proceedings 8th International Workshop on Interactive Distributed Multimedia Systems (iDMS 2001)*, ancaster, UK, September 2001.

[6] T. Nguyen and A. Zahkor. Distributed Video Streaming over the Internet. In *Proceedings of SPIE Conference on Multimedia Computing and Networking*, San Jose, California, January 2002.

[7] T. Nguyen and A. Zahkor. Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks. In *Proceedings of INFOCOM 2003*, San Francisco, California, April 2003.

[8] C. Papadopoulos and G. Parulkar. Retransmission-Based Error Control for Continuous Media Applications. In *NOSSDAV'96*, Zushi, Japan, 1996.

[9] C. Perkins, O. Hodson, and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Audio. In *IEEE Network*, volume 12-5, pages 40–48, September/October 1998.

[10] R. Rajaie and A. Ortega. PALS: Peer-to-Peer Adaptive Layered Streaming. In *NOSSDAV'03*, Monterey, California, June 2003.

[11] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On Peer to Peer Media Streaming. In *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, July 2002.