Peer-Assisted Packet Loss Repair for IPTV Video Multicast

Zhi Li*, Xiaoqing Zhu†, Ali C. Begen† and Bernd Girod*
*Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA
†Cisco Systems Inc., 170 West Tasman Drive, San Jose, CA 95134, USA
*{leeoz,bgirod}@stanford.edu, †{xiaoqzhu,abegen}cisco.com

ABSTRACT

Emerging IPTV technology uses source-specific IP multicast to deliver TV programs to the end-users. To provide timely and reliable services over the error-prone DSL access networks, a combination of multicast forward error correction and unicast retransmissions is employed to mitigate the impact of impulse noise. In current systems, the retransmission function is provided by the Retransmission Servers.

We propose an alternative distributed solution where the burden of loss repair is partially shifted to the peer IP set-top boxes. Through the Peer-assisted Repair (PAR) protocol, we demonstrate how the packet repairs can be delivered in a timely, reliable and decentralized manner using the combination of server-peer coordination and coded redundant repairs. We show through analysis and simulations that this new solution not only effectively mitigates the bottleneck experienced by the Retransmission Servers, thus greatly improving the scalability of the system, but also efficiently deals with peer departures and other uncertainties.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; Network Protocols

General Terms

Design, Performance

Keywords

IPTV, error-resilient video, impulse noise, peer-to-peer, reliable multicast, scalability

1. INTRODUCTION

Advances in video and networking technologies have made the delivery of television over telephone lines a reality. Internet Protocol TeleVision (IPTV) service is delivered over a carefully engineered network infrastructure and thus capable of meeting stringent quality-of-service (QoS) requirements.

The deployment of IPTV over Digital Subscriber Line (DSL) links has its own challenges. DSL uses the wires of the telephone network, which are susceptible to various types of interference. Most devastating is impulse noise, which could be caused by AC power switches, motors or lightning strikes.

To mitigate packet losses on DSL links, current IPTV deployments integrate source-specific multicast (SSM) with an error-control mechanism combining multicast forward error correction (FEC) and unicast retransmissions [3]. Retransmitted packets are generated locally by the Retransmission Servers (See Section 3). Each Retransmission Server can only support a limited number of set-top boxes (STBs) lest they are overwhelmed by retransmission requests for error bursts that are not corrected by FEC. The new approach we are proposing in this paper can mitigate this Retransmission Server bottleneck.

This work assumes that DSL bandwidth and link delay are known to the service provider and do not change in the short term [4]. Under such guarantees, this work has the following design goals:

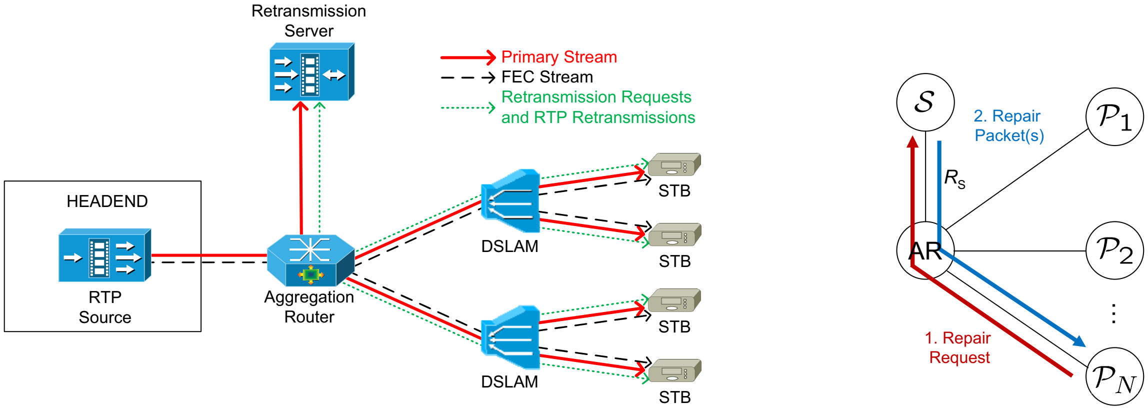
- Reliability. The mean time between artifacts (MTBA) is targeted at the industry benchmark for entertainment-grade IPTV service quality, *i.e.*, no more than one perceivable error in two hours. In terms of loss rate, we need to bring down the loss rate from a typical DSL level of 10^{-5} to less than 10^{-7} [3, 4].
- Timeliness. The playout latency is targeted at less than half a second to ensure a pleasing user experience during channel switching.
- Scalability. The loss-repair system is expected to scale to at least $10^4 \sim 10^5$ peer STBs.

1.1 Proposed Solution

We explore a new approach for achieving error resiliency in IPTV multicast – packet loss repair with the help of peer IP STBs. The main idea is to partially shift the repair task from the Retransmission Server to the nearby peer IP STBs receiving the same multicast data. Since each peer IP STB is now both beneficiary and contributor of repairs, the demand on the Retransmission Server can be dramatically reduced, allowing the Retransmission Server to scale more easily with the number of IP STBs served.

We propose Peer-assisted Repair (PAR) protocol, which coordinates the server and the peers to deliver the repair packets in a timely, reliable and decentralized way. In PAR, we provide a unified solution – coding of redundant repairs

MM'09 October 19–24, 2009, Beijing, China.



(a) (b)

Figure 1: (a) IPTV access network architecture with hybrid error control. IPTV primary and FEC stream are multicast to the IP STBs from the headend. Retransmission Server co-located with the aggregation router caches video streams and responds to retransmission requests from the IP STBs. (b) The abstract network topology considered and the server-assisted repair scheme. Only the retransmissions are depicted.

– to address various uncertainties in the network, including request losses, retransmission losses and peer departures. Such a solution can also be seamlessly integrated with FEC mechanisms, such as SLEP [15, 9]. Analytical and simulation results show that this solution only demands moderate upload bandwidth from each peer IP STB.

1.2 Overview

The rest of the paper is structured as follows. Section 2 discusses the related work. Section 3 gives an overview of the IPTV access network architecture and introduces the server-assisted loss-repair solution. Section 4 describes the details of the proposed peer-assisted loss-repair protocol. Section 5 analyzes various repair schemes presented in this study. Section 6 discusses the analytical and simulation results.

2. RELATED WORK

Multicast can be built at different layers of a network. Popular approaches include network-layer multicast (or IP multicast) [14, 13, 6, 18, 11, 7, 12, 8] and application-layer multicast [5, 2, 19, 10, 17, 21]. IP multicast requires network support whereas application-layer multicast does not require a specific infrastructure. In application-layer multicast, data packets are replicated at the end hosts, which logically form an overlay network. Media streaming over peer-to-peer networks, which has received considerable attention in recent years [19, 10, 17, 21], belongs to the later category. IPTV multicast that implements SSM at the network layer and unicast repair service at the transport/application layer can be generally considered in the former category.

Reliable multicast is a concept related to IP multicast. Reliable multicast protocols have been researched for a wide spectrum of applications, ranging from conferencing, network gaming, to file transfer and news feeds. Similar to TCP, reliable multicast seeks mechanisms for unfailing delivery of network packets. In the multicast scenario, achieving scalability by reducing the amount of feedback control traffic is an important concern. This issue has been addressed in either a hierarchical fashion, such as in Reliable Multicast Transport Protocol (RMTP) [13], or using multicast retransmission with feedback suppression [6, 12].

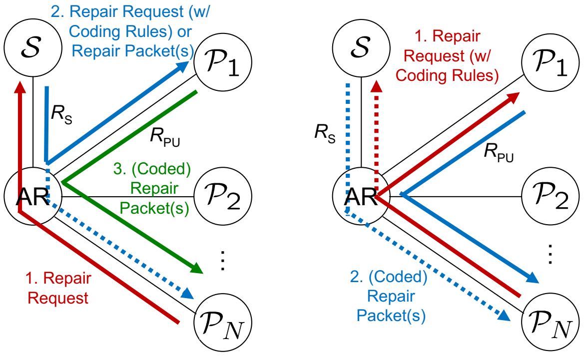
Scalable Reliable Multicast (SRM) is a protocol [6] implementing the "repair by any receiver" concept, which is similar in spirit to our PAR protocol. However, there are major differences between SRM and PAR. First, SRM is designed to minimize the coordination between the receivers, thus both the request and repairs are multicast to all the other receivers. PAR uses unicast repair instead. PAR provides more coordination among the receivers by leveraging the presence of a dedicated server. Timeliness is not an important concern in SRM, but it is of the essence in PAR. In SRM, group members maintain updates of status through low-frequency session messages, whereas in PAR, members promptly inform the dedicated server when joining/leaving the group and this information is promptly propagated to all the relevant members.

Pretty Good Multicast (PGM) [18] is a protocol that bypasses UDP and interfaces directly with IP. Designed with real-time applications in mind, timeliness is important in PGM. However, no precaution is taken for possible unavailability of repair – if no negative acknowledgment (NACK) packets are received by the time a transmit window times out, the data simply becomes unavailable for repairs.

Multicast File Transfer Protocol (MFTP) [11] targets bulk data delivery applications and has no low-latency requirement. MFTP breaks data into large chunks and aggregate repair requests for each chunk in NACK packets to reduce feedback implosion. A recent work named Cooperative Peer Assists and Multicast (CPM) [7] for the applications of video-on-demand (VoD) implements the bulk data transfer idea. To achieve low start-up delay, CPM proposes to prepopulate the beginning of each video to the local machines during off-peak hours. However, this is not an available option for IPTV applications. CPM also uses the dedicated server(s), but with the aim of constraining the latency. In contrast, the dedicated server is used for constraining the error probability in PAR.

3. SERVER-ASSISTED REPAIR

Figure 1(a) illustrates the IPTV access network architecture with a hybrid error-control system. IPTV primary stream, carried in a MPEG2 Transport Stream (MPEG2-



(a) (b)

Figure 2: Two versions of the peer-assisted loss-repair protocol: (a) Peer-assisted Repair with Centralized Tracking (PAR-CT) and (b) Peer-assisted Repair with Distributed Tracking (PAR-DT).

TS) [1] and encapsulated in Real-time Transport Protocol (RTP) [16] packets, originates from the RTP source in the headend. If FEC is employed, an FEC stream is also multicast from the source. Each IP STB may join one of many multicast sessions. Each group shares a primary stream and the corresponding FEC stream(s), which are delivered to the IP STBs through an aggregation router and DSL Access Multiplexer (DSLAM) over an SSM session. In the current server-assisted repair (SAR) scheme, the primary stream is cached by one or more Retransmission Server(s), co-located with the aggregation router. When interference on the DSL links (or on in-home links) corrupts the stream, the IP STB first attempts to recover the missing source packets by using the FEC packets. If this fails, it sends a request to the Retransmission Server, which then retransmits the requested packets over a unicast session.

In this work, we consider a simplified DSL access network topology. The aggregation router is connected with N peer nodes (or IP STBs) $\mathcal{P}_i, i = 1, \ldots, N$. Since the DSLAMs are layer-2 devices, we treat them as transparent in this model. Attached to the aggregation router is one Retransmission Server $\mathcal{S}$. Each peer node may join one of the K multicast sessions (*i.e.*, one group for each TV channel). Figure 1(b) gives an illustration of this network topology, as well as the SAR protocol. Note that only the retransmissions are depicted.

In SAR, the retransmission function is entirely provided by the local Retransmission Server. In our recent work [9], we showed that, for mitigating the impulse noise impact on the video quality, a limited available bandwidth for loss-repair is best utilized by reserving it entirely for retransmissions and omitting FEC altogether. However, in an SSM scenario, the Retransmission Server could be easily overwhelmed by the retransmission requests from all the downstream IP STBs. Consequently, the number of IP STBs each Retransmission Server can support is often limited by the Retransmission Server's capacity.

4. PEER-ASSISTED REPAIR

In PAR, the repair function is partially shifted to the peer IP STBs. The Retransmission Server only serves as the last resort when the Peer-assisted Repair is not available.

The PAR protocol is designed with *low latency* in mind, that is, unlike TCP or SRM [6], the system cannot afford to let the peer continue sending requests until the repair is received; instead, the system must ensure that the repair information is received in one retransmission with high probability. In addition, the communication between the peers does not rely on exchanging control messages between peers. Instead, the Retransmission Server promptly updates the status of each peer.

We propose two versions of the PAR protocol – PAR with Centralized Tracking (PAR-CT) and PAR with Distributed Tracking (PAR-DT). The two protocols differ from each other in (a) where the tracking table(s) are maintained, and (b) who takes the initiative to decide where to look for repairs. Compared to PAR-CT, PAR-DT demands less server bandwidth and reduces queuing delay at the server, but it is more difficult for PAR-DT to achieve tracking table synchronization. Figure 2 illustrates the two versions of the PAR protocol. Next, we discuss their details.

4.1 Tracking Peer Status

When a peer $\mathcal{P}_i$ joins a multicast session, the Retransmission Server $\mathcal{S}$ is promptly informed of this event via a special message [20]. During the session, $\mathcal{P}_i$ periodically sends messages to $\mathcal{S}$ to confirm that it continues to receive the multicast. If no such message is received for some time, $\mathcal{S}$ no longer considers $\mathcal{P}_i$ a receiver of the multicast. Also, when $\mathcal{P}_i$ leaves the multicast session, $\mathcal{S}$ is promptly informed.

In PAR-CT, $\mathcal{S}$ maintains a table that keeps track of the channel each peer $\mathcal{P}_i$ is watching.

In PAR-DT, $\mathcal{P}_i$ maintains its own tracking table that contains the set of peers $\{\mathcal{P}_j\}$ currently watching the same channel as $\mathcal{P}_i$. Whenever there is a status change, $\mathcal{S}$ is informed first, which, in turn, sends a message to update the tracking table of $\mathcal{P}_i$ if needed.

4.2 Repair Request Forwarding

In PAR-CT, when $\mathcal{P}_i$ experiences packet losses that cannot be corrected by FEC, it sends a repair request with the packet IDs to $\mathcal{S}$. $\mathcal{S}$ determines the number of redundant requests l needed to maintain a high probability of recovery. This is done either through simple heuristics or using the formulation given in Section 5. If coding needs to be applied across the repair packets, $\mathcal{S}$ also specifies the coding rules for each packet (See Section 4.3.2). $\mathcal{S}$ then selects l peers and forwards the retransmission requests, along with the coding rules if applicable; if not found, $\mathcal{S}$ replies to $\mathcal{P}_i$ with the requested packet itself. Notice that by forwarding the request instead of sending the repair packet, much Retransmission Server bitrate can be saved. Upon receiving the retransmission request, each $\mathcal{P}_j$ looks for the requested packets in its cache. If found, it responds to $\mathcal{P}_i$ with the requested packets. If necessary, coding is applied to generate the coded repair packets.

In PAR-DT, when $\mathcal{P}_i$ experiences packet losses, it first determines the number of redundant requests l needed. It also specifies the coding rules if coding shall be applied. $\mathcal{P}_i$ then selects l peers from $\{\mathcal{P}_j\}$ using its tracking table, and forwards the retransmission requests, possibly with coding rules. If $\mathcal{P}_i$ does not find a sufficient number of peers receiving the same multicast in its tracking table, it sends the repair request to $\mathcal{S}$. Either $\mathcal{S}$ or $\mathcal{P}_j$'s respond with the repair packets, similar to PAR-CT.

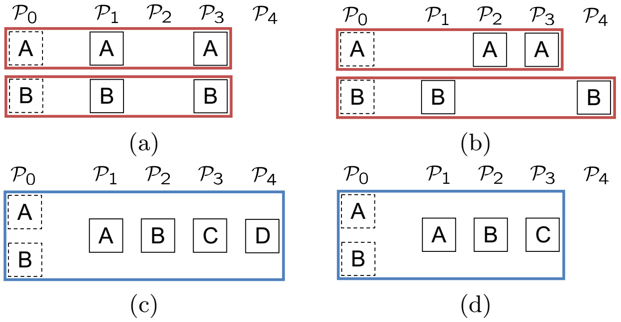


Figure 3: Examples of uncoded/coded redundant repairs. Two packets A and B are lost for peer $\mathcal{P}_0$, which has four available neighboring peers $\mathcal{P}_1$ to $\mathcal{P}_4$. The packet losses are independent for different peers but correlated for each individual peer. In (a) and (b), the retransmitted packets are uncoded. In (c), a systematic $(4,2)$ code is applied to A and B, generating retransmission packets A, B, C, and D. In (d), a systematic $(3,2)$ code is applied.

4.3 Redundant Repair Packets

In Peer-assisted Repair, peer departures could result in unserved requests. For example, when $\mathcal{P}_j$ changes channel (*i.e.*, leaves a multicast session), either $\mathcal{S}$ (in PAR-CT) or $\mathcal{P}_i$ (in PAR-DT) will still consider $\mathcal{P}_j$ available until their tracking tables are updated. Besides peer departures, other uncertainties could also lead to unserved requests. For example, due to imperfect synchronization, the requested packet may fall outside the cache window of $\mathcal{P}_j$. Redundant requests can address all these problems.

4.3.1 *Uncoded Case*

The simplest solution is to send redundant requests for each lost packet to multiple peers in $\{\mathcal{P}_j\}$, thus increasing the probability of $\mathcal{P}_i$ receiving at least one retransmitted packet. When not enough peers are available, retransmission can always be performed by $\mathcal{S}$. Note that the redundancy degree l should not be too large, otherwise the redundant packets will cause congestion in the downlink of $\mathcal{P}_i$. The uncoded case can be considered as a simple $(l,1)$ repetition code.

4.3.2 *Coded Case*

A more efficient solution is to apply erasure codes (*e.g.*, Reed-Solomon codes) across peers to recover a burst of packet losses. Either $\mathcal{S}$ (in PAR-CT) or $\mathcal{P}_i$ (in PAR-DT) computes the number of redundant repairs needed, specifies the coding rules in each request and sends out the requests to multiple peers as spreaded as possible. We use some simple examples to illustrate this idea.

Figure 3 illustrates simple examples where the peer $\mathcal{P}_0$ consecutively loses two packets A and B. $\mathcal{P}_0$ could send retransmission requests to peer $\mathcal{P}_1$ to $\mathcal{P}_4$. We assume that the packet losses are independent for different peers but correlated for each individual peer. In (a) and (b), $\mathcal{P}_0$ sends two requests for each lost packet. The resulting probability of unserved request is the same for both cases, but the occurrences are correlated for (a) whereas independent for (b). In (c), a systematic $(4,2)$ code is applied across A and B, generating packets A, B, C and D, which are then spreaded among all the neighboring peers. To achieve this, either the Retransmission Server $\mathcal{S}$ (in PAR-CT) or peer $\mathcal{P}_0$ (in PAR-DT) needs to specify in the request packets the explicit coding rule for the other peers. The repair succeeds if any two of the four packets are retransmitted and received by $\mathcal{P}_0$. The resulting probability of unserved request for case (c) is lower than (a) and (b). Codes of other rates can also be applied. In (d), a systematic $(3,2)$ code is applied to generate only one redundant packet to resist a burst of two packet losses.

Notice that only very little payload is needed to specify the coding rules. For example, in Figure 3 (d), the rule for $\mathcal{P}_1$ is “retransmit A”, for $\mathcal{P}_2$ is “retransmit B” and for $\mathcal{P}_3$ is “retransmit A XOR B”.

5. ANALYSIS

In this section, we analyze each repair scheme described. Throughout the analysis, we make the simplying assumption that the packet losses are independent for different peers. This assumption is based on the fact that the access network is single-hop and most of the interference occurs on the last-mile DSL links. There could be correlated losses as well (*e.g.*, a lightning strike), but their occurrence is rare[1]. In simulations, we also investigate the correlated case. For simplicity, we do not consider the loss of request and repair packets in the analysis.

Let p be the packet loss rate, as a function of the video packet duration T_{vp}, mean time between error burst T_{mtbb}, and the average burst duration T_{bst}. Denote the primary stream rate by R, and the budget for retransmission (*i.e.*, the repair packets) by α_{r}, expressed as percentage of the primary stream rate. Recall that N denotes the total number of peers, and K is the total number of channels. Denote the Retransmission Server bitrate as R_{S}, and the uplink bitrate at each peer as R_{PU}.

5.1 Server-Assisted Repair

For SAR, the expected number of requests received by the Retransmission Server is pN. The expected Retransmission Server rate is:

$$E[R_{\mathrm{S}}^{\mathrm{SAR}}] = pN\alpha_{\mathrm{r}}R. \tag{1}$$

5.2 PAR-CT and PAR-DT

For PAR-CT and PAR-DT, we analyze the expected Retransmission Server rate and the probability of unserved retransmission requests. We first find, for each multicast group, the probability of Retransmission Server directly handling the repair, and the probability of unserved request, both as a function of the group size. Then we compute the overall probabilities.

We assume a Zipf model for the user distribution among different channels, which has the following form:

$$f_z(k;s,K) = \frac{1/k^s}{\sum_{n=1}^{K}(1/n^s)} \tag{2}$$

where k is the popularity rank of a particular channel and s is the characterization exponent that determines the shape

---

[1] If losses are correlated, they can be more effectively repaired using multicast retransmissions. However, multicast retransmission would not have much impact on the Retransmission Server's bitrate.

Table 1: Simulation Parameters

Parameter	Default	Range
Number of Peers N	100	10 ∼ 1000
Number of Channels K	200	
Zipf Distr. Shape s	1	
Video Rate	5 Mb/sec	
Video Duration	30 sec	10 ∼ 50 sec
Playout Delay	400 ms	50 ∼ 550 ms
Peer-Router Link Delay	20 ms	
Router-Retran. Server Link Delay	0 ms	
FEC Budget α_{f}	0%	
Ret. Budget α_{r}	10%	
Avg. Burst Len. T_{bst}	8 ms	4 ∼ 16 ms
Mean Time btw Errors T_{mtbb}	2 sec	
Packet Loss Rate p	0.4%	0.2% ∼ 0.8%
Repair Packet Size	1375 B	
Request Packet Size	64 B	
Request Redundancy l	1	1 ∼ 3
Peer Departure Rate γ_{dept}	0	0 ∼ 0.2
Peer Uplink Bitrate m_{PU}	1	0 ∼ 3
Peer Downlink Bitrate m_{PD}	1	
Max. Peer Neighbors T	10	
% Correlated Peers cor	0%	0% ∼ 100%

of the distribution. The larger s is, the longer the tail. For program (*i.e.*, multicast group) k, the group size is

$$N_k = N \cdot f_z(k; s, K). \tag{3}$$

We now consider a multicast group of size n.

Each peer may depart the group without informing all the other peers or the Retransmission Server in time. Assume that a request is sent to an unresponsive peer with probability γ_{dept}. For simplicity, we only analyze the uncoded redundant repair case. Assume that, for each lost packet, the retransmission request is sent to l peers. If there are fewer than l available peers in the multicast group, the request is sent to the Retransmission Server instead, in which case we assume that the probability of an unserved request is 0.

Assume that, with uplink bitrate R_{PU}, each peer can serve m_{PU} retransmissions simultaneously, *i.e.*,

$$R_{\mathrm{PU}} = m_{\mathrm{PU}} \alpha_{\mathrm{r}} R. \tag{4}$$

There is a probability δ that the instantaneous demand on the uplink is beyond its capacity m_{PU}. We further introduce the practical limit that each peer can communicate with at most T other neighboring peers, and T neighboring peers can receive repair packets from that peer. Then δ can be expressed as:

$$\delta = \sum_{i=m_{\mathrm{PU}}}^{h} \binom{h}{i} \left(\frac{p}{h}\right)^{i} \left(1 - \frac{p}{h}\right)^{h-i} \tag{5}$$

where $h = \min(n-1, T)$. If the above case happens, at most $(h - m_{\mathrm{PU}})^{+}$ peers cannot be served. Assume this worst case.

Three possibilities may cause a peer unresponsive to a request – the peer has left the multicast session, the peer's corresponding packet is also lost, or the peer's uplink is congested. The probability that a peer is unresponsive to a request, denoted by p_{nr}, is thus:

$$\begin{aligned} p_{\mathrm{nr}} = {} & \gamma_{\mathrm{dept}} + (1 - \gamma_{\mathrm{dept}}) p \\ & + (1 - \gamma_{\mathrm{dept}})(1 - p) \delta \frac{(h - m_{\mathrm{PU}})^{+}}{h}. \end{aligned} \tag{6}$$

For a peer in a multicast group with a size larger than l, the probability of having a unserved request is p_{nr}^{l}. If the group size is no larger than l, the request is handled by the Retransmission Server. Overall, a peer's probability of unserved request in a group of size n, denoted by $\epsilon_{\mathrm{req}} \mid n$, is

$$\epsilon_{\mathrm{req}} \mid n = \begin{cases} 0, & n \le l \\ p_{\mathrm{nr}}^{l}, & \text{otherwise} \end{cases}. \tag{7}$$

The overall probability of unserved request is

$$\epsilon_{\mathrm{req}} = \sum_{k=1}^{K} (\epsilon_{\mathrm{req}} \mid N_k) \, f_z(k; s, K). \tag{8}$$

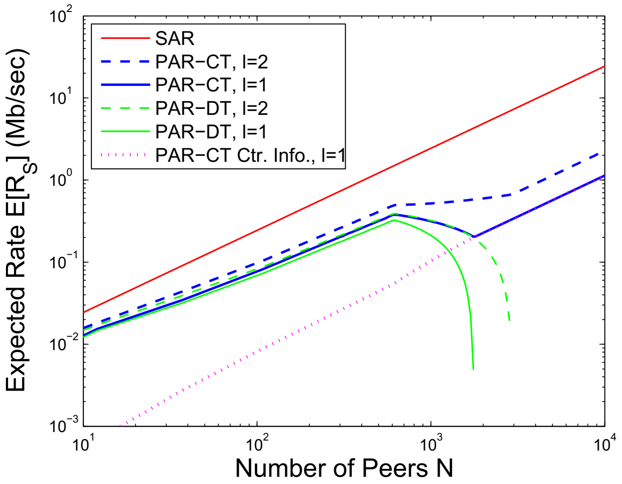


Figure 4: Analytical result: expected Retransmission Server rate $E[R_{\mathrm{S}}]$ versus the number of supported peers N, for (1) SAR with redundancy $l = 1$, (2) PAR-CT with $l = 1, 2$, (3) PAR-DT with $l = 1, 2$ and (4) PAR-CT control information.

For PAR-CT, the expected Retransmission Server rate includes the request packets forwarded by the Retransmission Server and the repair packets directly sent by the Retransmission Server. The expected number of repairs directly sent by the Retransmission Server is $p \sum_{k, N_k \le l} N_k$. Then the expected Retransmission Server rate for PAR-CT is:

$$E[R_{\mathrm{S}}^{\mathrm{PAR-CT}}] = p \alpha_{\mathrm{r}} R \sum_{k, N_k \le l} N_k + l p \alpha_{\mathrm{c}} R \left(N - \sum_{k, N_k \le l} N_k \right) \tag{9}$$

where α_{c} is the budget for control information (*i.e.*, the request packets), expressed as percentage of the primary stream rate.

For PAR-DT, the expected Retransmission Server rate is computed as:

$$E[R_{\mathrm{S}}^{\mathrm{PAR-DT}}] = p \alpha_{\mathrm{r}} R \sum_{k, N_k \le l} N_k. \tag{10}$$

6. PERFORMANCE EVALUATION

We evaluate the performance of the proposed PAR schemes through a set of comprehensive simulations and compare them with the analytical results. A proof-of-concept packet-level simulation has been implemented. The details of the

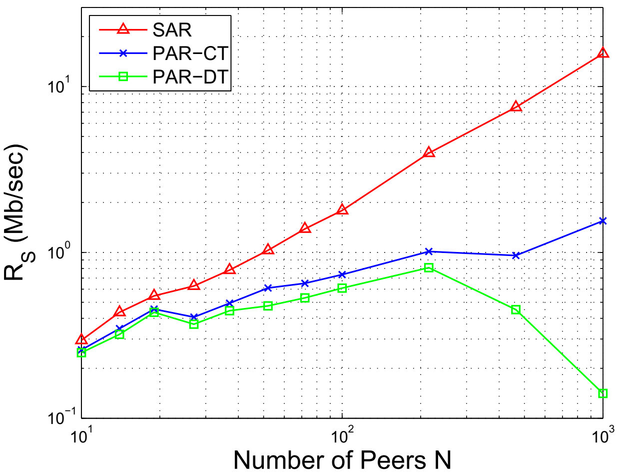


Figure 5: Simulation result: Retransmission Server rate $R_{\rm S}$, for PLR 3e-4, as a function of the number of supported peers N, for (1) SAR, (2) PAR-CT and (3) PAR-DT.

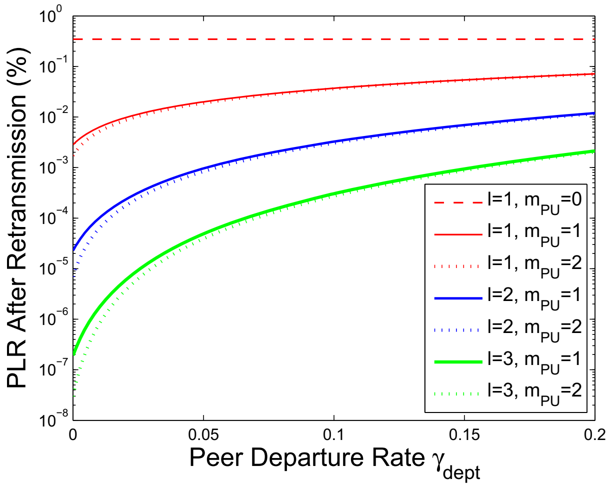


Figure 6: Analytical result: packet loss rate (PLR) after retransmission as a function of peer departure rate $\gamma_{\rm dept}$, for request redundancy $l = 1, 2, 3$ and peer uplink rate $m_{\rm PU} = 0, 1, 2$ (for PAR-CT and PAR-DT).

simulation and the parameters used are described in Section 6.1. In the rest of the section, we discuss various simulation results, including the demand on the server and peer bandwidths, resistance to peer departures, performance under different channel conditions and playout delays.

6.1 Simulation Settings

To allow simulations of up to 1000 peers, we have made some simplifications. Only the retransmitted packets are simulated. Since the primary and FEC streams are encoded with constant rate, we can assume that dedicated bandwidth has been allocated for forward transmission of these packets, and thus there is no interaction between them and the retransmitted packets. First-in-first-out queues buffer the packets for the server, peer uplinks and peer downlinks. The channel follows a two-state Markov model, whose transition probabilities are determined by $T_{\rm bst}$ and $T_{\rm mtbb}$. By picking proper values of $T_{\rm bst}$ and $T_{\rm mtbb}$, we simulate an environment with a packet loss rate (PLR) in the order of 1e-3. [2] For the simulation that applies coding across repair packets, we use Reed-Solomon codes across packets of a single burst loss. We use a code of average rate $2/3$, which introduces less redundancy than the $(2, 1)$ repetition code of the uncoded case. To simulate the correlation of loss pattern among the peers, we pick a proportion of peers randomly and make their packet losses fully correlated whereas losses for the rest are independent. Table 1 lists the simulation parameters used.

6.2 Retransmission Server Rate

We first measure the Retransmission Server rate, which reflects the burden on the central Retransmission Server. In Figure 4, we analytically compare the expected rate of SAR, PAR-CT and PAR-DT. As expected, the rate of SAR grows linearly with N. The rate of PAR-CT consists of two portions – control information (*i.e.*, request packets) and repair packets. For PAR-DT, only the repair packets count towards the rate.

The repair packet portion of rate exhibits an interesting behavior – as N increases, it grows until it reaches a maximum and then drops. This can be explained as follows. As N grows, the number of multicast groups increases, contributing to the overall rate. The number of peers within each group also increases, thus the availability of peers that possess the requested repair packets becomes higher, alleviating the demand on the Retransmission Server. The expected rate reaches its maximum when each multicast group has at least one peer. In Figure 4, we also plot the case when the peers send redundant requests ($l = 2$). Noticeably the rate is still much lower than the SAR case.

In Figure 5, we plot the Retransmission Server rate obtained in simulations. Note that in Figure 4 the y-axis is the *expected* bitrate, which is lower than the bandwidth for a PLR of 3e-4 in Figure 5. Nevertheless, we observe a qualitative agreement of the analytical results and the simulations.

6.3 Peer Uplink Rate

Compared to SAR, PAR-CT and PAR-DT require additional uplink bitrate from each peer. Figure 6 plots the PLR after retransmission versus the peer departure rate $\gamma_{\rm dept}$. We observe that no matter how much redundancy is introduced, it is sufficient to have the peer uplink bandwidth match the downlink bandwidth reserved for repair packets. The reason is that in DSL link, packet losses tend to be bursty but sparse. The chance that a peer is burdened by simultaneous repair requests from more than one peer is small. This result is confirmed in Figure 7, where it is shown that further increasing $m_{\rm PU}$ above 1 does not help reduce the PLR.

6.4 Effect of Uncoded Redundant Repairs

We measure, in the uncoded case, how much redundancy l would help mitigate most unserved requests due to peer de-

---

[2] A realistic DSL loss rate is at the level of 1e-5, but this is a rather unrealistic condition for simulations, since it would take too long to generate meaningful results. Therefore, we chose PLR 1e-3 in the simulation instead.

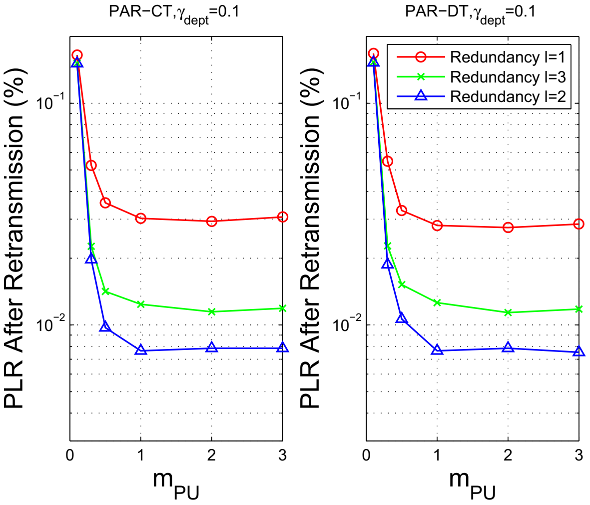


Figure 7: Simulation result: PLR after retransmission as a function of the peer uplink rate m_{PU}, for (a) PAR-CT and (b) PAR-DT with request redundancy $l = 1, 2, 3$. The results are measured at $\gamma_{\mathrm{dept}} = 0.1$. R_{S} is set such that at $m_{\mathrm{PU}} = 1$, the measured PLR is at least 3e-4 for all schemes.

partures. In Figure 8, we plot the PLR as a function of the Retransmission Server rate R_{S}, for $l = 1, 2, 3$ at $\gamma_{\mathrm{dept}} = 0.1$. When R_{S} is small, apparently R_{S} is not enough to support sending two or three copies of requests, thus the PLR is prohibitively high. When R_{S} is large enough, however, we observe that increasing l does help reduce PLR. This confirms the analytical results presented in Figure 6.

In Figure 7 and Figure 8, we observe that the performance at redundancy $l = 3$ is no better than the $l = 2$ case. This contradicts our analytical result in Figure 6. The reason is that the redundant packets from multiple peers congest the peer downlink which has only 10% of the video bandwidth allocated for repair packets. This can be seen in Figure 9, where the PLR drops if we increase the downlink bitrate m_{PD} beyond 1, suggesting that the downlink is no longer congested.

6.5 Effect of Coded Redundant Repairs

The result of applying R-S codes across the redundant packets is shown in Figure 8. We observe that although the redundancy $l = 3/2$ is less than for the $l = 2$ uncoded case, the coded scheme could further reduce the PLR in the presence of peer departures. Applying coding across packets is a more efficient solution to alleviate the impact of peer departures.

6.6 Varying Playout Delay

We verify the delay characteristics of each scheme discussed. We select an Retransmission Server bitrate enough to support all schemes, and vary the preset playout delay. Figure 10 illustrates the PLR after retransmission as a function of the playout delays. The result suggests that PAR-CT and PAR-DT require additional playout delay of about 40 ms than SAR. This is not unexpected. A simple analysis could show that for each retransmission, PAR-CT and PAR-DT traverse two more hops than SAR. Notice that,

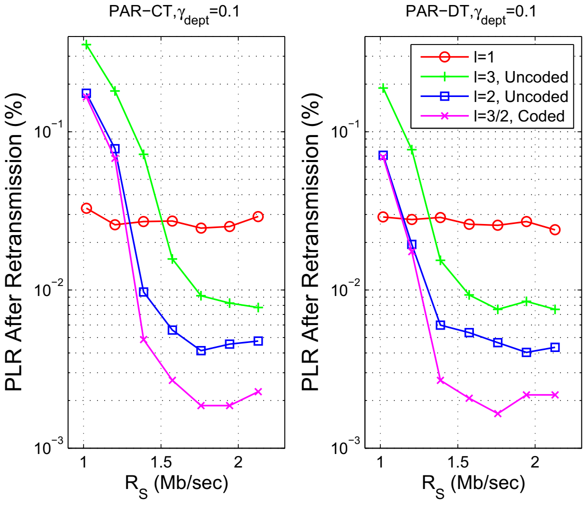


Figure 8: Simulation result: PLR after retransmission as a function of Retransmission Server rate R_{S} for (1) PAR-CT and (2) PAR-DT at request redundancy l=1,2,3. The results are measured at $\gamma_{\mathrm{dept}} = 0.1$.

compared to the target delay of 500 ms, the additional 40 ms is a rather short delay. Therefore, we could consider the additional delay as a small and manageable cost of PAR.

6.7 Effect of Correlated Losses

PAR is most efficient when the packet losses among peers are independent. This is because independence minimizes the probability of requesting repair packets from peers that lost the same portion of the stream. Figure 11 plots the PLR after retransmission at different correlation degrees. For some percentage of peers, the losses are fully correlated whereas for the rest, they are independent, are we vary the mix. The plot shows that the PLR becomes quite significant when 20% or more peers are that correlated. However, note that in a practical DSL system, in only very rare events (*e.g.*, lightning strikes) the losses are correlated. Nevertheless, we are currently investigating a solution to instantaneously switch to multicast repair mode when correlated losses occur.

7. CONCLUSIONS AND FUTURE WORK

We have presented a new error resiliency solution for IPTV multicast. The Peer-assisted Repair solution draws context from conventional reliable multicast, but with the support of specific network infrastructure, it is designed with simultaneously meeting reliability, low latency and scalability in mind. Through detailed analysis and simulations, we show that this solution is effective in reducing the burden on the Retransmission Server, thus allowing the system to scale easily to support a large number of IP STBs. In addition, it only requires modest additional uplink bitrate at each peer.

This distributed solution is most effective when the downstream packet losses show an uncorrelated pattern in the spatial domain. In DSL links and wireless home networks, impulse noise is mainly the result of power switches and mo-

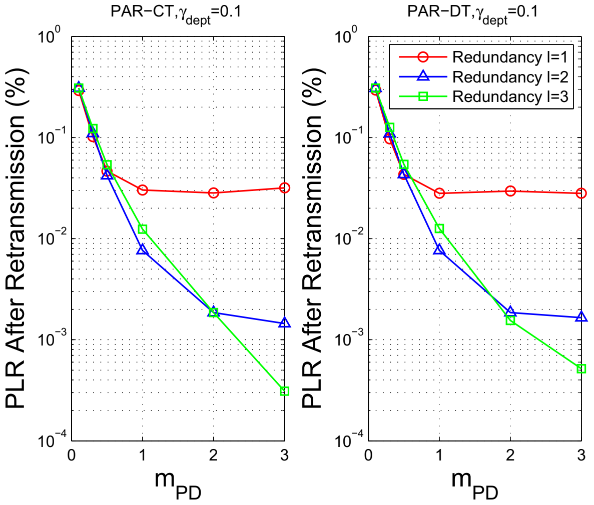


Figure 9: Simulation result: PLR after retransmission as a function of the (normalized) peer downlink bitrate m_{PD}, defined as $m_{\mathrm{PD}} = R_{\mathrm{PD}}/(\alpha_{\mathrm{r}} R)$, for (a) PAR-CT and (b) PAR-DT with request redundancy $l = 1, 2, 3$. The results are measured at $\gamma_{\mathrm{dept}} = 0.1$. R_{S} is set such that at $m_{\mathrm{PD}} = 1$, the measured PLR is at most 3e-4 for all schemes.

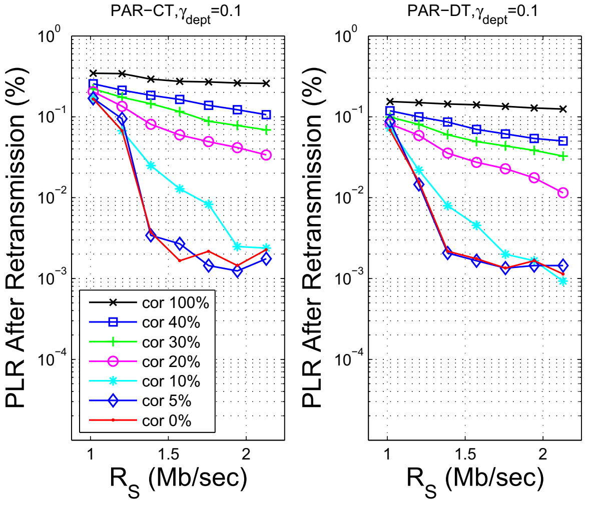


Figure 11: Simulation result: PLR after retransmission as a function of Retransmission Server rate R_{S} at different percentage of correlated peers.

tors, which are usually local and independent events. For correlated losses (in the spatial domain) caused by events such as a lightning strike, multicast repair may be a more effective solution. Combining Peer-assisted Repair with multicast repair into a single framework and designing the optimal switching point are interesting and important future research directions.

In this paper, we have assumed that the residual bandwidth on the data plane was the limiting factor for the repair services to scale up and the control plane processing resources were sufficiently available. In deployments where the

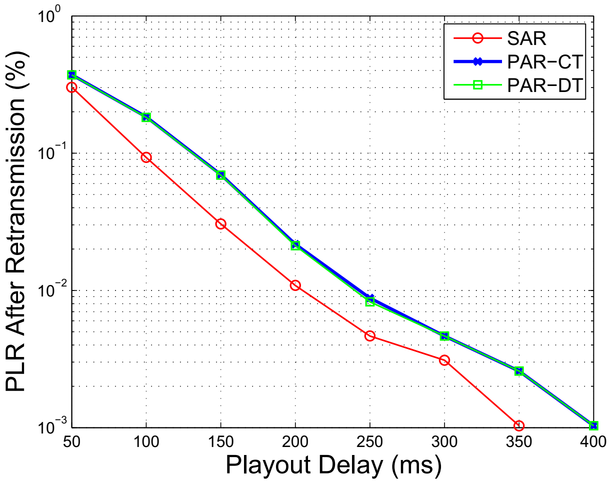


Figure 10: Simulation result: PLR after retransmission as a function of playout delay for (1) SAR, (2) PAR-CT and (3) PAR-DT.

opposite is true, distributed repair/tracking schemes such as PAR-DT may become more appealing than PAR-CT, where the server needs to coordinate among the peers by sending substantial amount of control information. How to further alleviate server processing load would be another interesting research direction.

The proposed solution has a great potential to be extended. One of its strengths is the ability to seamlessly integrate with forward protection schemes, such as simple FEC or content-aware SLEP. In the future work, we will investigate more advanced source and channel coding schemes in combination with Peer-assisted Repair and analyze their performances. Another natural extension is to accommodate fast channel switching [20] in the Peer-assisted Repair framework.

8. REFERENCES

[1] ISO/IEC 13818-1, Information technology: Generic coding of moving pictures and associated audio information: Systems. Dec. 2000.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM*, pages 205–217, 2002.

[3] A. Begen. Error control for IPTV over xDSL networks. In *Proc. Consumer Communications and Networking Conference, 2008. (CCNC 2008)*.

[4] Broadband Forum Technical Report. TR-176: ADSL2Plus configuration guidelines for IPTV. Available online: http://www.broadband-forum.org/technical/download/TR-176.pdf.

[5] Y.-H. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Measurement and Modeling of Computer Systems*, pages 1–12, 2000.

[6] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for

light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, December 1997.

[7] V. Gopalakrishnan, B. Bhattacharjee, K. Ramakrishnan, R. Jana, and D. Srivastava. CPM: Adaptive video-on-demand with cooperative peer assists and multicast. In *Proc. INFOCOM 2009*, Rio de Janieiro, Brazil, April 2009.

[8] M. Lacher, J. Nonnenmacher, and E. Biersack. Performance comparison of centralized versus distributed error recovery for reliable multicast. *IEEE/ACM Transactions on Networking*, 8(2):224–238, April 2000.

[9] Z. Li, X. Zhu, A. Begen, and B. Girod. Forward and retransmitted Systematic Lossy Error Protection for IPTV video multicast. In *Proc. Packet Video Workshop 2009*, Seattle, WA, May 2009.

[10] J. Liu, S. Rao, B. Li, and H. Zhang. Opportunities and challenges of peer-to-peer Internet video broadcast. *Proceedings of the IEEE*, 96(1):11–24, January 2008.

[11] K. Miller, K. Robertson, A. Tweedly, and M. White. Starburst multicast file transfer protocol (MFTP) specification. Available online: http://tools.ietf.org/html/draft-miller-mftp-spec-03.

[12] J. Nonnenmacher and E. Biersack. Scalable feedback for large groups. *IEEE/ACM Transactions on Networking*, 7(3):375–386, June 1999.

[13] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya. Reliable multicast transport protocol (RMTP). *IEEE Journal on Selected Areas in Communications*, April 1997.

[14] B. Quinn and K. Almeroth. RFC 3170: IP multicast applications: Challenges and solutions. Available online: http://www.ietf.org/rfc/rfc3170.txt.

[15] S. Rane, P. Baccichet, and B. Girod. Systematic lossy error protection of video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(10):1347 – 1360, Oct. 2008.

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 3550: RTP: A transport protocol for real-time applications. Available online: http://www.ietf.org/rfc/rfc3550.txt.

[17] E. Setton, P. Baccichet, and B. Girod. Peer-to-peer live multicast: A video perspective. *Proceedings of the IEEE*, 96(1):25–38, January 2008.

[18] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, R. Edmonstone, and L. Vicisano. RFC 3208: PGM reliable transport protocol specification. Available online: http://www.ietf.org/rfc/rfc3208.txt.

[19] D. Tsang, K. Ross, P. Rodriguez, J. Li, and G. Karlsson. Advances in peer-to-peer streaming systems. *IEEE Journal on Selected Areas in Communications*, 25(9):1609–1611, November 2007.

[20] B. VerSteeg, A. Begen, T. VanCaenegem, and Z. Vax. Internet draft: Unicast-based rapid acquisition of multicast RTP sessions. Available online: http://tools.ietf.org/id/draft-ietf-avt-rapid-acquisition-for-rtp-01.txt.

[21] X. Zhang, J. Liu, and T.-S. P. Yum. Coolstreaming/DONET: A data-driven overlay network for peer-to-peer live media streaming. In *Proceedings of IEEE Infocom*, 2005.