

Evaluating the Utility of FEC with Reliable Multicast

Dan Li and David R. Cheriton
Computer Science, Stanford University

Abstract

Forward Error Correction (FEC) has been proposed as a technique for implementing efficient reliable multicast (RM). However, FEC incurs costs in encode/decode delay and implementation complexity. How much benefit is provided relative to these costs and how dependent is the benefit on the specific RM protocol?

In this paper, we evaluate the benefits of FEC for RM, considering both proactive and reactive use with three RM recovery techniques: duplicate avoidance, limited scope multicast and subcast. Our simulation-based results indicate that FEC provides little benefit for an efficient RM protocol like OTERS and introduces extra delay for multi-point streaming data applications.

1 Introduction

Providing efficient reliable multicast (RM) support for large-scale multi-point applications is a challenge, especially when the application requires low delivery latency in addition to bandwidth-efficient delivery. Such applications (so-called streaming data applications)¹ include web cache invalidation, live stock-quote distribution, interactive Internet gaming and shared whiteboards.

Among the challenges is the problem of *feedback implosion*. With thousands or even millions of receivers, the probability of packet losses in the audience at any moment becomes so high that the source receives negative acknowledgments (NAKs) and performs retransmissions constantly, multiplying bandwidth consumption and causing more severe congestion and losses. An extra lossy receiver can trigger repeated retransmissions and slow down the multicast session even for the less lossy receivers. This is referred to as the problem of “*crying baby*” [21]. RM recovery tech-

niques addressing these problem typically trade off between latency and bandwidth. For example, delaying or restricting retransmissions [11, 15, 16] can reduce the recovery traffic but also increase the recovery latency.

Forward Error Correction (FEC) [2] is an appealing approach to avoid this feedback implosion. Several FEC-based protocols require no receiver feedback (RMDP [6], Digital Fountain [3], and Fcast [8]), while some send feedback only at the end of a *pass* (MFTP [1]). FEC has also been integrated with feedback-based general-purpose RM protocols, called *hybrid ARQ I* and *II* [9, 10, 20, 7]. More recent work [4, 5, 13, 14, 17] shows FEC efficiently recovers *temporally correlated*² losses, thus reducing the RM recovery overhead.

However, *proactive* use of FEC, i.e. on the original multicast data, requires for efficiency that the data be blocked and encoded in large transmission groups to reduce its bandwidth consumption overhead. This behavior delays packet transmission, making it less suitable for delay-sensitive applications. *Reactive* use of FEC, i.e. as part of the recovery mechanism, complicates the RM protocol significantly and can delay recovery as well.

In this paper, we evaluate the utility of FEC with reliable multicast, considering both its proactive and reactive use with three RM recovery techniques, namely *duplicate avoidance (DA)*, *limited scope multicast (LSM)*, and *subcast*. We perform event-driven simulations on transit-stub network topologies that are similar to those found in the real world, and employ *bandwidth-latency product* (or *BLP*) to assess the performance tradeoff between the delivery traffic and latency. Our experiments show that, among these techniques, (1) the more efficient the original design is, the less improvement FEC integration can offer, and

¹ *Streaming video/audio applications* can tolerate a small amount of loss and therefore do not require RM. However, fast recovery of key segments (e.g., the I frames) in the presence of a replay buffer may still be preferred.

² For example, instead of retransmitting individual lost segments, one segment of parity data is transmitted either proactively with the original data or in response to a NAK. Receivers can use the parity segment to recover any single loss in a range of data segments over which the parity segment is computed. Because these data segments are often transmitted consecutively, the parity segment is said to be able to repair *temporally correlated* packet losses.

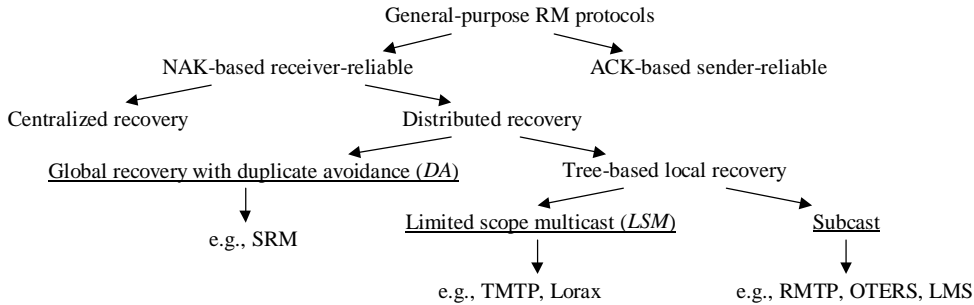


Figure 1 Classification of RM protocols

(2) subcast *without* FEC achieves the best overall performance.

The rest of the paper is organized as follows. Section 2 introduces the RM recovery techniques. Section 3 describes the integration of proactive and reactive FEC with RM protocols. Section 4 presents the simulation methodology. Section 5 analyzes the performance of FEC integration. Section 6 discusses related work, and Section 7 concludes the paper.

2 Background of reliable multicast

Reliable multicast protocols can be classified by the recovery mechanisms they use, as illustrated in Figure 1. The right branch of this classification, *ACK-based sender-reliable* transport, corresponds to conventional reliable transport as used in TCP, for instance. It is also called ARQ (Automatic Repeat reQuest). Simple ARQ does not scale to a large multicast receiver set [21, 22], prompting the use of the left branch of techniques, namely *NAK-based receiver-reliable* protocols. Here, each receiver is responsible for detecting losses and sending *NAKs* to request retransmissions. Scalability is further improved by using *distributed recovery*, where multiple retransmitters (in addition to the source) repair packet losses of nearby group members. We briefly describe three major classes of protocols within this NAK-based distributed recovery design space represented in Figure 1.

SRM [11] employs *global recovery with duplicate avoidance*. In SRM, any receiver may multicast a NAK to the multicast group and respond to a NAK with a multicast repair. Before transmitting a NAK or repair, the sender delays a period during which the transmission is suppressed if an identical one arrives. The delay is a randomized function of the round trip time (RTT) between the data source and the sender of the NAK (or repair). This suppression process is referred to as *duplicate avoidance (DA)*.

Another class of RM protocols employs *tree-based local recovery* [15, 16, 18, 33]. A subgroup hierarchy is constructed among multicast receivers and rooted at

the source. Members close to each other form a subgroup and elect a *designated receiver (DR)* for the subgroup. A DR may be a member of a higher-level subgroup. Only DRs retransmit NAKs and repairs are forwarded only within each subgroup.

To form the subgroup hierarchy, *TMTP* [15] and *Lorax* [16] rely on a technique called *expanding ring search (ERS)* which employs the IP time-to-live (TTL) field to discover nearby group members. Retransmissions are *limited scope multicast (LSM)*, i.e., multicasting to the host group with a TTL value equal to the subgroup's TTL radius, which restricts the delivery to the subgroup (working only in source-specific multicast routing domains). NAKs are either LSM with DA or direct unicast to the DR.

As an alternative to ERS, subcast is used by some tree-based local recovery protocols, including *RMTP* [33], *OTERS* [18], and *LMS* [23], for subgroup formation and/or packet retransmission. *Subcasting* is multicasting a packet over a subtree of the multicast delivery tree for a multicast group. In particular, a retransmission can be subcast by the DR to the requester's subtree, confining the delivery to just the subtree in which the loss was reported (Figure 2). Subcast requires support in the network. For instance, in *OTERS*, subcasting is built on IP encapsulation [24] and IGMP traceroute [25], with security extensions that involve router changes but impose no additional state and little processing overhead.

In later sections, we evaluate the integration of FEC with SRM, *TMTP-uni*³, and *OTERS*.

3 Integrating FEC with RM protocols

FEC can be integrated with the three main RM protocols, as described in this section. Figure 3 illustrates

³ *TMTP-uni* directly unicasts NAKs to the DR while the original *TMTP* design multicasts NAKs using LSM and DA. We did not use *TMTP-uni* for all parameter combinations. Besides, the SRM simulation studies the behavior of multicast NAKs with DA.

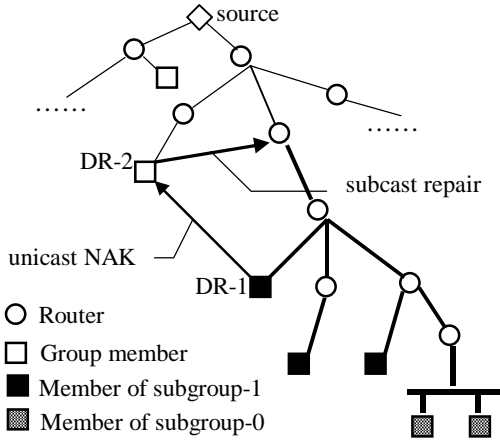


Figure 2 Subcast retransmission. DR-1 unicasts a NAK to its own designated receiver – DR-2 – after detecting a packet loss. DR-2 responds with a repair to DR-1's subtree, assuming DR-2 received this packet. Bold links indicate the path of the retransmission, which reaches all the descendant subgroups.

the basic operation of *FEC erasure correction* [27, 28, 3].

3.1 Reactive FEC

In *Reactive FEC*, i.e., *hybrid ARQ II* [9], retransmissions are FEC packets instead of individual lost segments. A NAK is sent for a TG (transmission group) when a packet in a later TG arrives while this TG is still incomplete. The NAK indicates the number of losses in the TG, not the sequence numbers of the lost segments.

In principle, a retransmitter should pick the maximum number of losses indicated in all the NAKs it receives for one TG and multicast that number of FEC packets to the requesters. However, NAKs do not usually arrive in synch and the retransmitter needs to respond to them promptly and therefore individually. Instead of sending the full amount that a NAK requests, the retransmitter subtracts the amount that has been sent recently from the requested amount and sends this smaller amount in the hope that the recently sent ones may reach the requester shortly after the requester sent out this NAK.

In a tree-based RM protocol, when a DR itself is recovering a TG, it cannot respond to NAKs for the TG. Therefore, the DR records them. Once the DR recovers the full TG, it immediately encodes the TG and responds to the outstanding NAKs as follows. In TMTP-uni, the DR transmits the maximum requested amount to its subgroup. In OTERS, the DR subcasts to each requester's subtree the requested amount in excess of what the DR has requested for itself (because the sub-

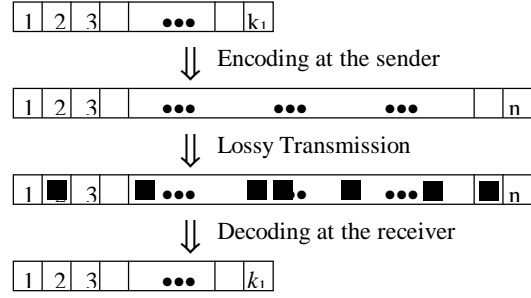


Figure 3 FEC erasure correction. The top row is the original transmission group (TG), consisting of k_1 equal-size segments. These segments can be reconstructed from any k_2 of the n encoded segments (the so-called FEC packets). k_1 and k_2 are usually equal and denoted as k . Black squares mark the segments that are lost during the transmission. The transmission is reliable with up to $n - k_2$ packet losses.

cast repairs sent from a higher-level DR can reach all the descendant subgroups).

3.2 Proactive FEC

In *proactive FEC*, i.e., *hybrid ARQ I* [10] (also referred to as *layered FEC* [4]), the sender transmits extra FEC packets immediately following the original data of each TG. Retransmissions are still necessary to ensure reliability but are fewer because proactive FEC effectively reduces the end-to-end loss rate.

Instead of sending a fixed number of FEC packets proactively [4], our study employs an *adaptive* proactive FEC scheme similar to that devised by Kermodé [14]. The number of FEC packets to send proactively is decided by the *redundancy estimation*, which is initially zero and incremented by the maximum number of losses that NAKs report for a TG. To track the loss-rate fluctuation, the redundancy estimation decays by half for each TG if there is no NAK for the previous TG (indicating that the redundancy for the previous TG is sufficient). Thus the redundancy estimation forms an exponential moving average of the actual amount of repairs needed for each TG. In SRM, only the source proactively transmits FEC packets (because otherwise receivers would transmit to the same group, resulting in duplicates and/or delays). In TMTP-uni and OTERS, both the source and the DRs proactively transmit FEC packets. In TMTP-uni, a DR keeps track of the redundancy estimation of its entire subgroup. In OTERS, a DR keeps the redundancy estimation for each subgroup member (because packets are subcast to each one's subtree).

3.3 General effect of integrating FEC

As the TG size k increases, the delivery traffic de-

creases and the delivery latency increases [32]. TG size 1 corresponds to no FEC integration.

There are two sources of traffic reduction. First, there are fewer repairs because one FEC packet can recover any loss in its TG for multiple receivers, while without FEC all the lost segments have to be retransmitted. Second, FEC reduces NAKs from one per missing segment to one per incomplete TG.

The prolonged delivery latency reflects the time to recover from losses. A packet loss is detected as soon as there is a sequence number gap. However, the receiver does not send NAKs to recover the missing packet until the beginning of the next TG. Moreover, a DR cannot retransmit until it completely receives the requested TG (possibly via recovery itself). Therefore, the longer delivery latency reflects the TG transmission delay.

Proactive FEC can improve the delivery latency over reactive FEC. However, it may consume more bandwidth because it is difficult to obtain appropriate *redundancy estimation*. Due to the heterogeneous connectivity among receivers, any amount of redundancy may be insufficient to some receivers while unnecessary to others. Temporal fluctuations of loss rates also cause the history-based estimation to be too large (raising the recovery traffic) or too small (leaving reactive FEC to make up the difference) over time.

The simulation in Section 5 confirms the above intuitive analysis. Moreover, it demonstrates that FEC's effect differs dramatically when integrated with different RM recovery techniques.

4 Simulation methodology

Three RM protocols were simulated with reactive FEC alone (called *SRM*, *TMTP-uni* and *OTERS*) and with both reactive and proactive FEC (called *SRM-pro*, *TMTP-pro* and *OTERS-pro*), all using the NS network simulator [29].

The simulations focus on the protocols' behavior in the network and do not attempt to model the end-station processing such as FEC encode/decode and storage overheads. Their addition to the delivery latency is relatively small compared to the transmission and recovery delay over the network [28], and likely to be less significant in the future as memory and processor cycles become even cheaper.

We assume that (1) k_1 and k_2 are equal and denoted as k , and (2) n is sufficiently large so that no FEC packet is transmitted twice. Thus every packet a receiver gets is not a duplicate and can contribute to the erasure correction.

4.1 Parameters and performance metrics

A simulation specifies five parameters: (1) protocol type, (2) TG size k , (3) membership density d , (4) link packet error rate (PER) p , and (5) average length of a burst of losses b .

A simulation tracks two quantities: (1) the *delivery latency* — from the time a segment is sent by the source to the time it is reliably delivered to all the receivers, and (2) the *bandwidth consumption* — average traffic required for reliably delivering one payload segment to all the receivers. Both quantities are normalized by that of the loss-free case.

The bandwidth consumption includes (1) *payload* — the original data stream, (2) *request* – NAKs, and (3) *repair* – proactive or reactive FEC packets transmitted (beyond the original k). Session management messages are not included because they are not a per-payload-segment cost and not affected by FEC.

The metric of bandwidth consumption is *packet-hops* (or *ph*), which tracks the number of forwards by routers rather than the number of distinct packets generated by end-stations. It reflects the traffic load on the network rather than that on the end-stations.

Last, to quantify the overall performance (the trade-off between bandwidth and latency), we introduce the *bandwidth-latency product* (or BLP) — the product of bandwidth consumption (in packet-hops) and delivery latency (in seconds).

4.2 Simulation environment

The simulation is driven by a packet stream from a Constant Bit Rate (CBR) source to a multicast group at the rate of 100 packets per second and with a total of 448 packets (chosen to be always a multiple of the TG size). The multicast routing protocol is DVMRP [30]. Receivers join the multicast group at time 0 and start to exchange *session messages* to form the subgroup hierarchy (in TMTP-uni and OTERS) or to measure RTTs (in SRM). The data stream starts at time 0.2 second and overlaps with the subgroup formation, which takes 300 to 600ms. Receivers also send heartbeats (a form of session messages) every second with random skews. The overlap and the heartbeats allow more realistic evaluations under the assumption that the network and the membership may change on a per-second basis, which may trigger the rebuilding of major parts of the subgroup hierarchy.

Ten transit-stub topologies are generated by the GT-ITM internetwork topology generator [31]. Each has 600 nodes, including 3 transit domains and 72 stub domains. Links inside a stub domain are 100Mbps with 1ms average queuing delay, resembling fast

Protocol Name	Bandwidth Consumption (in ph)				Delivery Latency	BW-latency Product	Session Messages
	Payload	request	Repair	total			
OTERS	525	27	84	636	325 ms	207	9 kilo-ph
TMTP-uni	525	31	596	1,152	330 ms	380	31 kilo-ph
SRM	525	1,394	4,681	6,600	425 ms	2,805	368 kilo-ph

Table 1 Performance of the RM protocols without integrating FEC ($k = 1$, $d = 10\%$, $p = 1\%$, $b = 2$)

Ethernet in campus networks. Links connecting stub and transit domains are 45Mbps with 15ms delay, resembling T3 lines from campus to the regional network. Links inside a transit domain are 155Mbps with 8ms delay, resembling OC3 lines of an ISP. Inter-transit-domain links are 155Mbps with 80ms delay, resembling OC3 lines between ISPs. All link delays adhere to an unbounded exponential distribution with 20% average variation. The simulation results are averages of the results obtained on these topologies.

We implement bursty link losses modeled by a *discrete Markov chain* $\{X_n\}$ with a countable state space I and stationary transitions P [12], where $X_n \in I =$

$\{0, 1\}$, $P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} = \begin{bmatrix} u_0 & 1-u_0 \\ 1-u_1 & u_1 \end{bmatrix}$. The n -th packet is lost if $X_n = 0$ and forwarded if $X_n = 1$. It follows that the average link PER $p = \frac{1-u_1}{2-u_0-u_1}$ and the

average burst length $b = \frac{1}{1-u_0}$. Given p and b , the

Markov chain has $u_0 = 1 - \frac{1}{b}$, $u_1 = \frac{1-p-p/b}{1-p}$. The bursty loss model degenerates to the random loss model if $u_0 + u_1 = 1$, i.e., if $b = \frac{1}{1-p}$ (≈ 1 if $p \ll 1$).

5 Performance analysis

We study FEC with RM under a set of typical parameters, namely group membership being 10% of the population, link PER being 1%, and average burst length being 2. Appendix at <http://www-dsg.stanford.edu/Publications.html> shows that the same conclusion holds under more extreme conditions, i.e., dense membership ($d = 50\%$), high link PER ($p = 5\%$), and random losses ($b \approx 1$).

5.1 Without FEC

Table 1 displays the performances of the three protocols without FEC integration. It shows that OTERS outperforms TMTP-uni, which in turn outperforms SRM, from both the bandwidth and latency aspects.

OTERS and TMTP-uni both have low request traffic because they use unicast NAKs. OTERS incurs less

repair traffic than TMTP-uni because subcast repairs are more efficient than LSM repairs. One drop at an intermediate router causes losses at all the receivers downstream from the point of drop, called *spatially correlated losses*. In TMTP-uni, such losses tend to scatter into multiple subgroups (controlled by TTL radius) and trigger multiple repairs, while in OTERS one repair subcast from the point of drop recovers all correlated losses. Subcast repair also has lower delivery latency because it reaches every receiver in a subtree and avoids relaying between DRs.

SRM consumes more bandwidth for two reasons. First, SRM uses global multicast for NAKs and repairs while TMTP-uni and OTERS localize them. Second, even with DA, a receiver may generate duplicates when another receiver's NAK (or repair) is propagating and has not reached this receiver. Moreover, DA intentionally delays the NAKs and repairs to avoid duplicates, resulting in a high latency for SRM.

The session messages are measured for the entire transport session. SRM has more session messages, again because of global multicast. OTERS has fewer because only DRs and new members generate subcast session messages, while in TMTP-uni every group member sends session messages using LSM. Moreover, the TTL in a LSM packet does not prevent the packet from being (wastefully) forwarded toward receivers that are further away than the TTL and dropped only at the border of the subgroup.

5.2 With FEC

Figure 4 plots the normalized bandwidth consumption vs. the normalized delivery latency, under the same condition as in Table 1 except that the TG size k varies from 1 to 112. It shows that a large TG size increases the delivery latency regardless of the RM protocol to which FEC is applied. Conversely, the level of bandwidth reduction heavily depends on the RM protocol, i.e., $SRM \gg TMTP\text{-uni} \gg OTERS$. This behavior arises because the more recovery traffic in the non-FEC case, the more of those NAKs and repairs are redundant and may disappear in the FEC cases.

Figure 5 plots the corresponding normalized BLP. Figure 6 plots the request traffic vs. the TG size and Figure 7 plots the repair traffic.

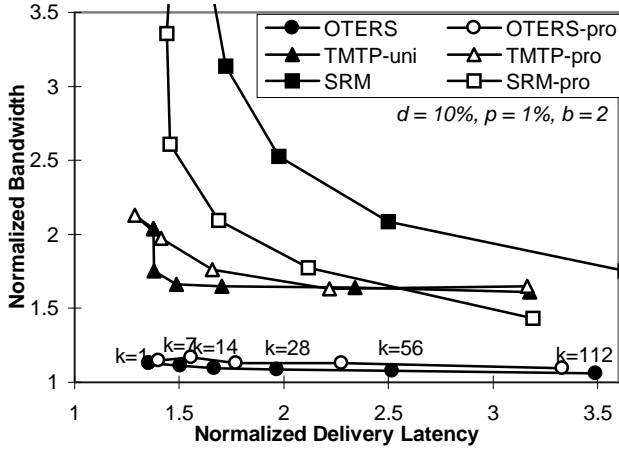


Figure 4 Bandwidth Consumption vs. Delivery Latency
 Both x and y axes are normalized by the value of the loss-free case. Nodes on each curve, from right to left, are for $k = 112, 56, 28, 14, 7$ and 1 respectively.

5.2.1 FEC and subcast-based recovery. Figure 5 shows that OTERS achieves the best BLP *without* FEC. FEC reduces little of its recovery traffic while significantly raising its delivery latency. Figures 6 and 7 show that the only reduction to OTERS’s traffic is from NAKs, which is at the direct cost of delaying the NAK and the recovery process.

FEC barely reduces OTERS’s repair traffic because OTERS leaves few inefficient cases for FEC to improve. One subcast retransmission can repair an entire subtree’s losses that are caused by one packet drop at the root of the subtree, i.e. the spatially correlated losses. According to studies such as those by M. Handley [35] and MFTP [1], Mbone losses show strong spatial correlation rather than independence.

Only leaf losses may be temporally, but not spatially, correlated. OTERS recovers these losses via unicast repairs.⁴ Such unicast repairs, on one hand, cannot be improved by unicasting FEC repairs and, on the other hand, are often more efficient (in terms of packet-hops) than multicasting FEC repairs to all the receivers.

The only inefficient case that FEC may enhance is when the loss is independent to a DR yet a subcast repair is still triggered. However, OTERS minimizes such cases by choosing a subgroup’s DR to be the least lossy and/or the closest to the subroot. Figure 7 shows that the reduction to OTERS repair traffic is less than 5% for TG sizes under 56.

5.2.2 FEC and duplicate avoidance. Figure 5 shows that FEC significantly improves SRM’s BLP, by up to

⁴ In most cases a leaf receiver is not a DR and requests only unicast rather than subcast retransmissions.

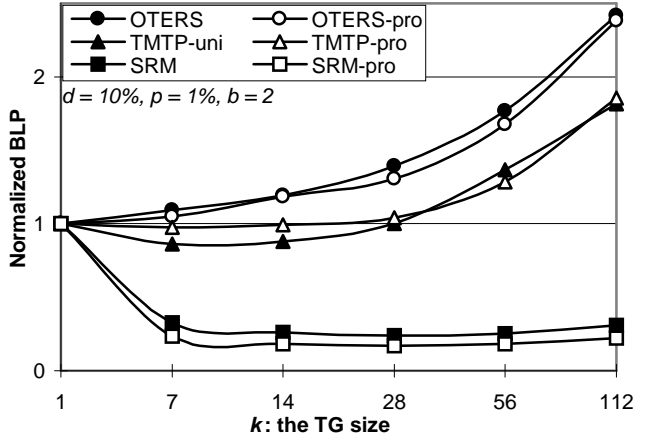


Figure 5 Normalized BLP vs. TG Size
 Each protocol’s BLP values are normalized by the BLP of that protocol in the non-FEC case, i.e., when $k = 1$.

83% over plain SRM, reducing both the traffic and the latency.

Figure 6 shows substantial reduction to SRM’s request traffic, because of SRM’s DA process. In plain SRM, when a group member receives a NAK that does not match any sequence number of its own missing packets, it becomes a potential retransmitter, a source of duplicate repairs. And it still needs to send NAKs for its own missing packets. Under FEC, not only does this receiver not contend for retransmission but also its own NAK is suppressed if the amount it has lost is no more than the amount reported in the NAK it receives.

Similarly, Figure 7 shows significant reduction to SRM’s repair traffic. In plain SRM, a group member may receive multicast repairs that do not match the sequence numbers of its own missing packets and thus are useless to its loss recovery. With FEC, these useless repairs may contribute to erasure correction and hence obviate further retransmissions, reducing both the traffic and the latency.

Figure 4 shows that SRM-pro is more effective than SRM with reactive FEC alone, because proactive FEC preempts the DA process. Thus it eliminates more NAKs as well as bypassing the DA delay. So SRM-pro at $k = 7$ has a delivery latency even lower than the non-FEC case. However, when $k > 14$, the TG transmission delay dominates and prolongs the delivery latency. And the BLP curve is almost flat, indicating that reducing the traffic merely proportionally increases the latency.

5.2.3 FEC and limited scope multicast. Figure 5 shows that FEC only moderately improves TMTP-uni’s BLP, by less than 14%, and worsens it when the

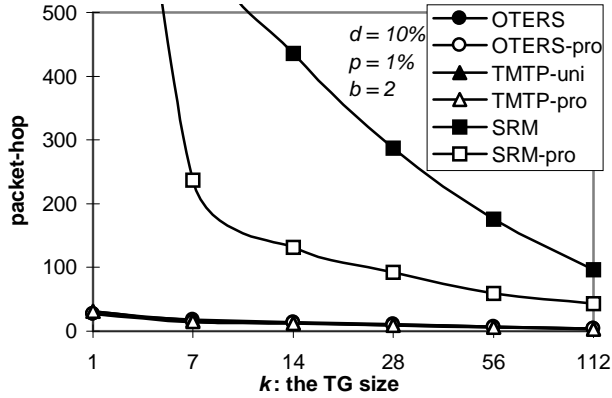


Figure 6 Request Traffic (NAKs) vs. TG Size

TG size is over 28. This is because TMTP-uni employs a subgroup hierarchy for local recovery. The same FEC mechanism that reduces SRM’s traffic applies to TMTP-uni, but at a smaller scale — subgroups. Thus Figure 7 shows moderate reduction to TMTP-uni’s repair traffic. Moreover, Figure 6 shows little reduction to its NAKs because, unlike SRM, TMTP-uni unicasts NAKs so there are few duplicates. So does OTERS.

TMTP-pro does not outperform TMTP-uni as SRM-pro does because TMTP-uni has no DA process. Figure 7 shows that TMTP-pro’s repair traffic exceeds that of TMTP-uni, especially when the TG size is small, because the smaller the TG, the more inaccurate the redundancy estimation. Figure 4 shows that the delivery latency of TMTP-pro is better than that of TMTP-uni with reactive FEC alone, but at the cost of traffic increase. Consequently, in Figure 5, TMTP-pro’s normalized BLP is never below 1.

5.3 Tradeoffs in RM recovery techniques

Reliable multicast design involves various tradeoffs that come down to a fundamental tradeoff between bandwidth and latency. FEC is one classic example. Duplicate avoidance another example. A longer DA timeout suppresses more duplicates but also proportionally raises the recovery latency. TMTP-uni and OTERS avoid the DA delay by constructing a subgroup hierarchy and unicasting NAKs directly to the DR. TMTP-uni’s tradeoff lies in the radius of limited scope multicast. A smaller subgroup radius can avoid NAK implosion and better limit the scope of multicast repairs, thereby using less bandwidth. Unfortunately, a smaller subgroup results in less sharing of multicast repairs. Besides, the subgroup hierarchy is deeper and thus propagating a repair down the DR hierarchy takes longer. Subcast-based recovery does not have this effect because repairs are subcast directly to all

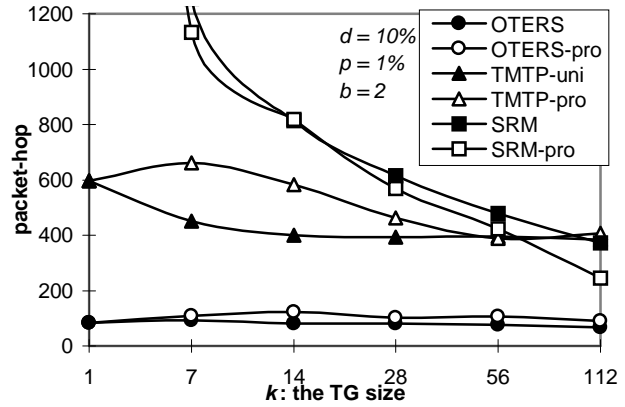


Figure 7 Repair Traffic vs. TG Size

the subgroups underneath a subroot.

Because of these design tradeoffs, both DA and LSM generate a substantial amount of redundant recovery traffic. FEC can make use of the redundancy and improve protocols that employ these techniques. Subcast-based recovery, however, complies with and takes advantage of the multicast loss pattern. Hence it does not benefit from FEC in any significant way.

6 Related work

Among previous studies on FEC, some focus on the centralized model where only the source generates and multicasts FEC packets, while some examine the distributed model where DRs employ FEC for loss recovery. Our work belongs to the distributed model.

6.1 Distributed model

Nonnenmacher *et al.* [17] analytically studied distributed and centralized reactive FEC. It concludes that distributed recovery with FEC outperforms that without FEC in terms of the bandwidth cost. Second, distributed FEC outperforms centralized FEC in both the bandwidth cost and the completion time. Third, applying FEC to an already distributed recovery scheme does not yield as much gain as applying FEC to a centralized recovery scheme.

Our study differs from [17] in the following ways. First, while [17] compares generic RM protocols (centralized and distributed schemes), our study goes a step further to examine three specific recovery techniques within the distributed RM design space. We show that specific design choices heavily affect how much they benefit from FEC.

Second, we track the delivery latency because it is a major requirement of streaming data applications, while [17] only measures the completion time, which large TG sizes hardly increase. Moreover, [17] does not show the completion time of the distributed

scheme without FEC, nor compare it to that with FEC.

Third, we employ event-driven simulation on transit-stub network topologies while [17] employs analytical models and a three-tier multicast delivery tree with the DRs placed at ideal locations. While [17]’s approach simplifies and generalizes its findings, our detailed simulation and realistic internetwork topologies provide additional insights. For instance, our simulation shows that, although LSM and subcast both use subgroups to localize recovery, LSM leaves more room for FEC to enhance while subcast does not.

Gemmel [13] and Kermode [14] studied FEC with SRM. [13] employs centralized reactive FEC. [14] extends SRM with proactive FEC and employs administratively scoped multicast for local recovery. They show that applying FEC to SRM reduces the traffic load of plain SRM by several orders of magnitude. However, applying FEC to already localized SRM reduces the traffic less significantly. In comparison, we not only study SRM but also study and compare TMTP-uni and OTERS. While [14] simulates a “hybrid mesh tree” topology with 112 nodes, we use more and larger network topologies and try to average out the randomness in the simulation.

6.2 Centralized model

Sakakibara *et al.* [32] model hybrid ARQ II on a broadcast channel with a fixed frame rate and negligible propagation delay (compared to the frame rate). They show that a larger k value yields higher throughput, especially for large error rate and large number of receivers. However, the average transmission delay increases in proportion to k . Similar evaluation methods and conclusions are presented by many other studies such as [9, 10].

Nonnenmacher *et al.* [4] studied FEC integration with a centralized NAK-based RM scheme using an analytical model where the source and the receivers are connected via a star topology with disjoint links. They show that reactive FEC scales better than a centralized scheme without FEC. Reactive FEC reduces the number of transmissions by the source, at the expense of coding at the end-systems. The source is the bottleneck if the data is not pre-coded. The proactive FEC scheme in [4] is static and does not adapt to the loss rate fluctuation. [4] shows that this form of proactive FEC can be worse than that without FEC and even worse in the presence of bursty losses. [4] also points out that FEC saves more bandwidth recovering independent losses (vs. shared losses).

Our study differs from them in that we focus on distributed RM techniques. Moreover, we take a different

angle to the bandwidth cost — from the network’s perspective (the number of forwards), while much previous work is from the source’s perspective (the throughput or the number of transmissions at the source). For example, [5] points out that the FEC gain (at the source) is higher if there are fewer shared links in the multicast distribution tree. However, from the network’s view, it is advantageous to have more shared links because the network is more productive, performing less forwards for each multicast packet.

7 Conclusion

Forward error correction (FEC) for reliable multicast (RM) provides limited benefit with an efficient RM protocol and detracts from performance with multi-point streaming applications. For instance, with the OTERS protocol, FEC provides marginal reduction in recovery traffic yet significantly increases the delivery latency. OTERS efficiently recovers spatially correlated losses via subcast, leaving few inefficient cases for FEC to enhance. Other less efficient protocols, such as SRM and TMTP-uni, show far more benefit but not sufficient to make them competitive with OTERS. In particular, OTERS *without* FEC provides low latency delivery and 6 to 13 times less recovery traffic (in 600-node topologies) compared to SRM and TMTP-uni *with* FEC. It also avoids the encode/decode complexity and the associated delay, processing and storage overhead. Without FEC, OTERS also outperforms TMTP-uni, which in turn outperforms SRM, in *both* bandwidth and latency.

With OTERS, efficient RM is dependent on network support for subcasting. Subcasting can be supported by simple, stateless extensions to routers so is straightforward to add [18]. However, subcasting is not currently widely supported, and its absence seems like the primary motivation to rely on more complex RM techniques such as FEC. As more multicast applications are deployed in the Internet, we hope to see the benefits of subcasting more widely recognized, causing it to be widely supported and eliminating the need for complicating RM protocols with FEC.

Acknowledgement

The authors would like to thank Armando Fox, Craig Partridge, Lorenzo Vicisano, Pablo Molinero and members of the Distributed Systems Group for their valuable support and discussions.

Bibliography

- [1] K. Miller, K. Robertson, A. Tweedly and M. White,

- “StarBurst Multicast File Transfer Protocol (MFTP) Specification”, work in progress, <ftp://ietf.org/internet-drafts/draft-miller-mftp-spec-03.txt>
- [2] S. Lin and D. J. Costello, “Error Control Coding: Fundamentals and Applications”, Prentice-Hall, 1983.
- [3] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashu Rege, “A Digital Fountain Approach to Reliable Distribution of Bulk Data”, ACM SIGCOMM98 Conference. Sept. 1998. Computer Communication Review (Oct. 1998) vol.28, no.4 p. 56-67
- [4] J. Nonnenmacher, E. W. Biersack, and Don Towsley, “Parity-based loss recovery for reliable multicast transmission”, IEEE/ACM TRANSACTIONS ON NETWORKING, Aug. 1998. vol.6, no.4, p. 349-61
- [5] J. Nonnenmacher and E. W. Biersack, “Reliable multicast: Where to use FEC”, Proceedings of 5th International Workshop on Protocols for High Speed Networks. Oct. 1996. (PpHSN '96) p. 134-48
- [6] Luigi Rizzo and Lorenzo Vicisano, “A Reliable Multicast data Distribution Protocol based on software FEC techniques (RMDP)”, 4th IEEE Workshop on High-Performance Communication Systems (HPCS'97) (1997) p. 115-24
- [7] J. P. Macker, “Reliable multicast transport and integrated erasure-based forward error correction”, Proceedings IEEE MILCOM, November 1997, p. 973-7 vol.2
- [8] Eve Schooler and Jim Gemmell, “Using Multicast FEC to Solve the Midnight Madness Problem”, Microsoft Research Technical Report, MSR-TR-97-25, September 1997.
- [9] J. J. Metzner, “An improved broadcast retransmission protocol”, IEEE Transactions on Communications, June 1984. vol.COM-32, no.6, p. 679-83
- [10] S. Lin, D. J. Costello and M. J. Miller, “Automatic-repeat-request error-control schemes”, IEEE communication magazine, 22(12):5-17, 1984.
- [11] S. Floyd, V. Jacobson, C. Liu, S. McCanne, L. Zhang, “A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing”, IEEE/ACM Transaction on Networking, Dec.1997 vol.5 no.6 p.784-803
- [12] David Freedman, “Markov Chains”, Holden-Day, Inc., San Francisco, 1971
- [13] Jim Gemmell, “Scalable Reliable Multicast Using Erasure-Correcting Re-sends”, June 1997, Microsoft Research Technical Report, MSR-TR-97-20, June 1997.
- [14] Roger G. Kermode, “Scoped Hybrid Automatic Repeat Request with Forward Error Correction (SHARQFEC)”, SIGCOMM'98, September 1998. Computer Communication Review (Oct. 1998) vol.28, no.4 p. 278-89
- [15] R. Yavatkar, J. Griffioen and M. Sudan, “A Reliable Dissemination Protocol for Interactive Collaborative Applications”, Proc. ACM Multimedia'95 Conference.
- [16] B. Levine, D. Lavo, and J.J. Garcia-Luna-Aceves, “The Case for Concurrent Reliable Multicasting Using Shared Ack Trees”, Proc. ACM Multimedia'96 Conference.
- [17] Jorg Nonnenmacher, Martin Lacher, Matthias Jung, Ernst W. Biersack and Georg Carle, “How bad is reliable multicast without local recovery?”, INFOCOM98.
- [18] D. Li and D. R. Cheriton, “OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol”, Proceedings Sixth International Conference on Network Protocols. Austin, TX, USA 13-16 Oct. 1998. p. 237-45
- [19] I. S. Gopal and J. M. Jaffe, “Point-to-multipoint communication over broadcast links”, IEEE Transactions on Communications, vol. COM-33, no.3,pp-232-240, Mar. 95.
- [20] R. H. Deng, “Hybrid ARQ Schemes for Point-to-multipoint Communication over Nonstationary Broadcast Channels”, IEEE transactions on communications, vol. 41, No. 9, September 1993.
- [21] H. Holbrook, S. Singhal and D. R. Cheriton, “Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation”, Proc. SIGCOMM'95, Computer Communication Review (Oct. 1995) vol.25, no.4 p. 328-41
- [22] B.N. Levine and J.J. Garcia-Luna-Aceves, “A Comparison of Reliable Multicast Protocols,” Multimedia Systems (ACM/Springer), Vol. 6, No.5, August 1998.
- [23] C. Papadopoulos, G. Parulkar, and G. Varghese, “An Error Control Scheme for Large-Scale multicast Applications”, Proceedings IEEE INFOCOM'98 Conference on Computer Communications, March 1998. p. 1188-96 vol.3
- [24] C. Perkins, “IP Encapsulation within IP”, RFC 2003, October 1996.
- [25] W. Fenner, and S. Casner, “A “traceroute” facility for IP Multicast”, Internet Draft <draft-ietf-idmr-traceroute-ipm-02.txt>, November, 1997, work in progress.
- [26] H. Holbrook and D. R. Cheriton, “IP Multicast Channels: EXPRESS support for large-scale multicast applications”, Proc. SIGCOMM'99, Sept. 1999, also <http://www-dsg.stanford.edu/holbrook/express>.
- [27] R. E. Blahut, “Theory and Practice of Error Control Codes”, Addison Wesley, MA, 1984.
- [28] Luigi Rizzo and Lorenzo Vicisano, “Effective erasure codes for reliable computer communication protocols”, ACM Computer Communication Review, April 1997, vol.27, no.2, p. 24-36
- [29] UCB/LBNL/VINT Network Simulator - ns (version 2), <http://www-mash.cs.berkeley.edu/ns/>
- [30] D. Waitzman, C. Partridge and S.E. Deering, “Distance Vector Multicast Routing Protocol”, RFC1075, Nov. 1988.
- [31] K. Calvert, and E. Zegura, “GT Internetwork Topology Models (GT-ITM)”, <http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/>
- [32] K. Sakakibara and M. Kasahara, “A multicast hybrid ARQ scheme using MDS codes and GMD decoding”, IEEE transactions on communications, 43(12):2933-2939, December 1995.
- [33] J. Lin and S. Paul, “RMTP: A Reliable Multicast Transport Protocol”, Proceedings of IEEE INFOCOM '96. Conference on Computer Communications. p. 1414-24 vol.3
- [34] UCB/LBNL/VINT Network Animator, <http://www-mash.cs.berkeley.edu/nam/>
- [35] M. Handley, ISI, “An Examination of Mbone Loss Distributions”, <http://north.east.isi.edu/mbonemon/>, 1998.