Dynamic Activation and Deactivation of Repair Servers in a Multicast Tree*

Per-Oddvar Osland†, Sneha Kumar Kasera‡, Jim Kurose, Don Towsley

Computer Science Department
140 Governor's Drive
University of Massachusetts
Amherst, MA 01003-4601
E-mail: {osland, kasera, kurose, towsley}@cs.umass.edu
CMPSCI Technical Report TR1999-56
January 2000

Keywords: reliable multicast, repair services, dynamic activation/deactivation

Abstract

Server based local recovery approaches are efficient in providing Reliable Multicast in a best effort network. In this paper we investigate how Repair Servers should be placed in the multicast tree spanning source and receivers in order to minimize costs and reduce packet delivery latency. We consider a cost function that accounts for the cost of transmission and buffering and processing at the RS. The cost-optimal placement of buffering resources depends on parameters such as link loss pattern and relative buffer cost. We also consider the case when link loss behavior changes over time. An adaptive, distributed, and autonomous policy is proposed based on on-line estimation of packet loss, which makes decisions for activation and deactivation of Repair Servers. The policy shows good performance in a simple case study.

1 Introduction

Many applications require the reliable delivery of data from one sender to multiple receivers. These include news publishing, teleconferencing, and distribution of software and financial information. Providing reliable delivery in a best effort network that exhibits packet loss, e.g. the Internet, require a reliable multicast transport protocol. Designing such a protocol which makes efficient use of network resources and provides low latencies is a challenging problem.

The most basic way a receiver can recover from packet loss, is by sending a NAK to the sender, which in turn retransmits the lost packet. There are however some aspects that show this approach as unsuitable:

- *NAK implosion:* if the same packet is seen as missing by several receivers simultaneously, the sender will become congested handling NAKs.
- *Loss path multiplicity:* a packet sent from the sender is likely to be lost on at least one link. Due to the branching in the tree this will probably affect several receivers, and the sender may transmit each packet at least once until obtained by all receivers.

*This material was supported in part by DARPA under Grant No. N6600L-97-C-8513. The first author was also supported by NFR (The Norwegian Research Council) under Grant No. 132388/431.
†Currently with Deptartment of Telematics, NTNU, N-7491 Trondheim, Norway. E-mail: peroo@item.ntnu.no
‡Currently with Bell Labs, Lucent Technologies, Holmdel, NJ, USA. E-mail: kasera@research.bell-labs.com

- *Retransmission scooping:* due to the branching in the tree, a loss will often affect multiple receivers, but not a large fraction of them. Packet restoration by multicasting from the sender results in wasted bandwidth to the receivers not requiring it. Unicasting the requested packet to each of the receivers that sent a NAK can result in inefficiently used bandwidth.

- retransmission from the sender can result in large delays in packet delivery.

Repair Services is an approach that can reduce resource utilization and latency. The idea is to temporarily store data in buffers at Repair Servers located close to the receivers. Lost packets may then be obtained from these buffers rather from the sender. However, Repair Services consume resources, and it is important to reduce their use while still obtaining much of their benefits.

In this paper we consider the problem of where to locate Repair Servers and how and when to activate them. Suggestions for how to provide reliable and efficient multicast include local recovery schemes, end-to-end based schemes, and server based schemes [1, 2, 3, 4, 5, 6]. To our knowledge, none of the server-based approaches have considered the problem of dynamic activation and deactivation Repair Server facilities.

This paper is organized as follows: In the next section we briefly motivate the need for an adaptive strategy for reliable multicast that accounts for changes in link loss. We argue that dynamic allocation of Repair Servers is a suitable method. Section 3 describes the protocol used. In Section 4 we derive an analytical expression for evaluating the performance of the protocol, as well as cost functions encompassing transmission, buffering, and processing costs. A case study is presented in Section 5. In Section 6 we suggest a policy for dynamic activation/deactivation of Repair Servers according to changing link loss in the multicast tree, and finally conclusions are listed in the last section.

2 Enabling reliable multicast with Repair Servers

Providing Repair Services in the network is a promising method for obtaining reliable multicast [1, 3]. The key idea is to store packets at Repair Servers close to the receivers. By retrieving lost packets from one of these buffers rather than from the sender, fast and efficient packet restoration is achieved.

In this section we present an approach where a Repair Server (RS) may be associated with routers in the multicast tree structure. This extends the earlier analysis of Kasera et al [3], where Repair Servers were colocated with routers at the edge of the backbone network. Receivers are connected to these routers through tail links. It was assumed that no loss occurs within the backbone, only at the tail links [7]. The extension in this paper is motivated by loss data collected by Handley [8] which show that link loss also occurs at upstream links within the multicast tree structure. To take account for this, an *Active Error Recovery* protocol [9] has been developed with the ability to associate a Repair Server with nodes in the multicast tree, depending on the loss pattern in the network.

A Repair Server is associated with a node (router) in the multicast tree. The RS offers buffering; it holds a window of the most recent packets in the multicast stream. It also offers processing power, e.g. for NAK suppression. A receiver restores a lost packet from the closest upstream RS or from the sender. If the RS no longer holds the packet, it obtains the packet from it's closest upstream RS or directly from the sender.

We investigate how RSs should be distributed in a multicast tree in order to obtain reliable multicast with a minimum use of bandwidth and buffer capacity. A RS allocation specifies the set of routers that have active RSs associated with them. To evaluate the quality of a specific RS allocation, we define a cost function that expresses the cost associated with sending one packet from the sender to all receivers. This cost function, which will be presented in Section 4.4, accounts for transmission and buffering and processing at the RS.

2.1 The static problem

We begin by assuming the link loss probabilities and the population of receivers to be invariant. In this context, we investigate different placements of RSs in order to find an optimal solution for that specific set of system parameters, i.e. the allocation of RSs that minimizes the cost for transmission and buffering and processing at the RS.

2.2 The dynamic problem

In real life, link loss probabilities will change over time. Furthermore, the multicast tree structure will change due to routing changes and as receivers join and leave the multicast session. Consequently the RS allocation should change during the lifetime of the session in order to adapt to these changes. Thus changes in the RS allocation could occur frequently suggesting the need for a decentralized RS allocation strategy. A policy for reliable multicast must be distributed, adaptive, and scalable. The Repair Servers must act autonomously and determine individually when to activate and deactivate. This is a veritable challenge as the only information available to the Repair Servers is the Multicast routing table (which resides in a router with multicast capabilities), the stream of packets from the sender, and the stream of NAKs from the receivers. An adaptive activation policy should base it's actions on measured packet loss rates. This calls for an online measurement of packet loss. When designing an estimation technique (see Section 6), we must know what properties of the link loss pattern we actually want to observe. Hence there is a need for describing the dynamics of the link loss probability.

Link loss dynamics

Analyses of end-to-end Internet loss measurements show that the temporal loss pattern is complex [10, 8, 7]. A very coarse model is constructed by decomposing the end-to-end loss into loss due to user activities, which cause long term changes, and losses due to application, protocol, and network mechanisms, resulting in short term variations in link loss. Paxson [10, Ch. 15] and Handley [8] show plots of loss rate versus time that indicate significant changes in losses over intervals of several minutes. The graphs reflect user activity during office hours and evening, which increase traffic in specific areas of the network, and consequently lead to higher loss rate on heavily loaded links. At a finer time-scale one also observes huge variations in loss probability. It has been shown that end-to-end packet loss occurs in bursts of length less than 500 milliseconds or less [8, 7]. This burstiness is explained by the fact that loss occurs due to buffer overflow at network nodes. It has also been observed that loss bursts occur periodically at 30 second intervals. This effect has not been fully explained other than attributing it to periodic effects in the network itself [8]. Although these results hold for end-to-end patterns, we assume that a single link will exhibit similar characteristics: link loss probabilities can coarsely be described by long term changes (significant over a period of several minutes), and rapid changes over several hundred milliseconds. It is not feasible, and probably even not desirable, to detect and react to losses that occur and vanish during less than a second. We rather design our policy to identify and adapt to the long-range trends in link loss dynamics.

One of the aims of this paper is to design a *policy*[1] for autonomous activation and deactivation of Repair Servers associated with router in the network. There are several aspects with such a policy that have to be investigated:

- is it possible to design a distributed algorithm that will result in a stable RS allocation following a change in the link loss pattern?
- if it is possible to design such an algorithm, how fast will the RS allocation transit between stable allocations?
- will the resulting allocation provide close to optimal performance?

In Section 6 we suggest a policy and investigate how well it performs with respect to these items.

3 Protocol description

In this paper we assume a Repair Server based reliable multicast protocol similar to the Active Error Recovery (AER) protocol. Here only the most important features are described, see the AER web-site [9] for details.

Consider a multicast session with a single sender and a fixed number of receivers. Packets flow from the sender to the receivers over a multicast tree $\mathcal{T}$ with nodes $\mathcal{N}$ and links $\mathcal{L}$, exemplified in Figure 1. We assume

[1] The terms *policy* and *algorithm* will be used interchangeably

the topology is fixed, i.e. different routing alternatives are not considered. Furthermore, for the time being we assume that the number of receivers and the link loss probabilities are constant.

All routers have Repair Server (RS) facilities, but only some of them will be activated at a given time. The active RS at router $n \in \mathcal{N}$ will in average buffer B_n packets in the multicast stream. Further multicast protocol details:

- when a packet is received for the first time, it is multicast downstream and stored at the RS
- there are three kinds of packets: ordinary *multicast packets* and *retransmitted packets* (aka *repair packets*) which travel downstream to the receivers, and *NAK*s which go upstream
- when a receiver or a RS detects a missing packet in the downstream multicast flow, it sends a NAK to the sender (or the upstream RS). A pending NAK is kept until the repair packet arrives. If the packet has not arrived within a given timeout, a new NAK is sent
- when a RS receives a NAK from below, the requested packet is copied from the buffer and multicast. A *local recovery failure* occurs if the requested packet does not exist in the RS's buffer. If the RS has a pending NAK for this packet, no further action is needed. Otherwise the RS sends the NAK upstream and keeps a pending NAK until the repair packet arrives
- when a RS receives a repair packet for multicast packet no k, it first checks if it has a pending NAK for this packet. If this is the case, the repair packet is stored in the buffer and multicast downstream. If not, the repair packet is simply discarded

From the Sender's perspective, the closest RSs in the tree act as receivers, and the subtrees below these RSs are not seen by the Sender. With respect to buffer management, we assume that

- packets are stored in a FIFO manner in the order of arrival. Consequently, packets in the buffer may not have consecutive sequence numbers (retransmitted packets distort the order)
- since the RS has limited buffer capacity available, the stored packets will only remain in the buffer for a certain time. The intention is to store each packet long enough to enable a certain number of retransmissions to the RS's subtree. This will be further elaborated in Section 4.3

4 Model and analysis

Let the nodes be numbered in a breadth-first manner, the sender being node 1. Nomenclature:

$\mathcal{T}(\mathcal{N},\mathcal{L})$	the entire tree with nodes $\mathcal{N}$ and links $\mathcal{L}$
$\mathcal{T}(n)$	subtree rooted at node $n \in \mathcal{N}$ with nodes $\mathcal{N}(n)$ and links $\mathcal{L}(n)$, considering nodes with activated RS as leaf nodes. In Figure 1: $\mathcal{N}(1) = \{1-5, 16-18\}$, $\mathcal{N}(3) = \{3, 6-10, 19-21\}$, $\mathcal{N}(4) = \{4, 11-15, 22-25\}$
$\mathcal{R}$	set of routers including the sender. In Figure 1: $\mathcal{R} = \{1,2,3,4,5,6,11\}$
$\mathcal{S}$	set of routers with RS capabilities activated. In Figure 1: $\mathcal{S} = \{3,4\}$
$\mathcal{C}(n)$	children of node n, $\mathcal{C}(n) = \{m \mid (n,m) \in \mathcal{N}\}$
$\mathcal{P}(n)$	parent of node n, $(\mathcal{P}(n), n) \in \mathcal{N}$
Δ	expected packet interarrival time at the sender
$\Delta'(n)$	at node n, $\Delta'(n)$ is the expected upstream NAK interarrival time for a specific packet
$R(n)$	$R(n)$ = number of transmissions of a packet from $\mathcal{P}(n)$ until accepted by all nodes in $\mathcal{N}(n)$, $n > 1$. $R(1)$ = number of transmissions of a packet from the sender until accepted by all nodes in $\mathcal{N}$.
F_n	Probability distribution function of $R(n)$: $F_n(i) = \text{Prob}[R(n) \leq i]$
q_n	link loss probability at link $(\mathcal{P}(n), n)$, $n \in \mathcal{N}$
p_n	effective loss probability at link $(\mathcal{P}(n), n)$ as seen from n's parent node $P(n)$, counting losses on the link $(\mathcal{P}(n), n)$ and losses in the subtree $\mathcal{T}(n)$ which result in retransmissions over the link $(\mathcal{P}(n), n)$
$L(n)$	number of links a packet travels in $\mathcal{T}(n)$ when multicast from n. $L(n) \leq \lvert\mathcal{L}(n)\rvert$ with equality if $q_m = 0$ for all links $(\mathcal{P}(m), m)$ such that $m \in \mathcal{N}(n)$
c_b	unit buffering cost (per RS per packet stored)
C_b	total buffering cost (per multicast packet originally transmitted from the sender)
c_t	unit transmission cost (per packet per link traveled)
C_t	total transmission cost (per multicast packet originally transmitted from the sender)
c_{RS}	fixed RS processing cost (per RS per multicast packet originally transmitted from the sender)
ε	probability of failure in packet recovery from a RS
B_n	expected buffer required at RS_n (# packets)
I_n	$I_n - 1$ = minimum number of retransmissions of a packet from RS_n needed to ensure probability of local recovery $\geq 1 - \varepsilon$

4.1 Approximation of effective link loss

The RS colocated with router n will store enough packets to ensure that the probability of recover failure at RS_n is less than or equal to ε, $0 \leq \varepsilon < 1$. The expected buffer occupancy, B_n, will be calculated in Section 4.3. Now assume that RS_n is active. When a packet is sent on link $(\mathcal{P}(n), n)$, the effective link loss is $p_n \leq q_n + (1 - q_n)\varepsilon$. The first term corresponds to the packet being lost on the link. In the second term the packet gets to the RS (the probability for this is $1 - q_n$). However, if a packet is lost repeatedly in the subtree below RS_n, the RS will finally have to ask for retransmission by sending a NAK to $\mathcal{P}(n)$. Emission of NAK due to recovery failure at RS_n will occur with probability $\leq \varepsilon$, and $\mathcal{P}(n)$ will perceive it as link loss. In the further calculations we shall use the upper bound, i.e. $p_n = q_n + (1 - q_n)\varepsilon$. When a packet is sent downstream to a node that does not have an active RS, the effective link loss is simply q_n. Hence

$$p_n = \begin{cases} q_n + (1 - q_n)\varepsilon & \text{if } \text{RS}_n \text{ is active} \\ q_n & \text{else} \end{cases} \quad (1)$$

4.2 Distribution of $R(n)$

The derivations in this section are inspired by Bhagwat et al [11]. $F_n(i)$ is the probability that at most i transmissions of a packet from node $\mathcal{P}(n)$ to n are needed before this packet is received by all receivers in

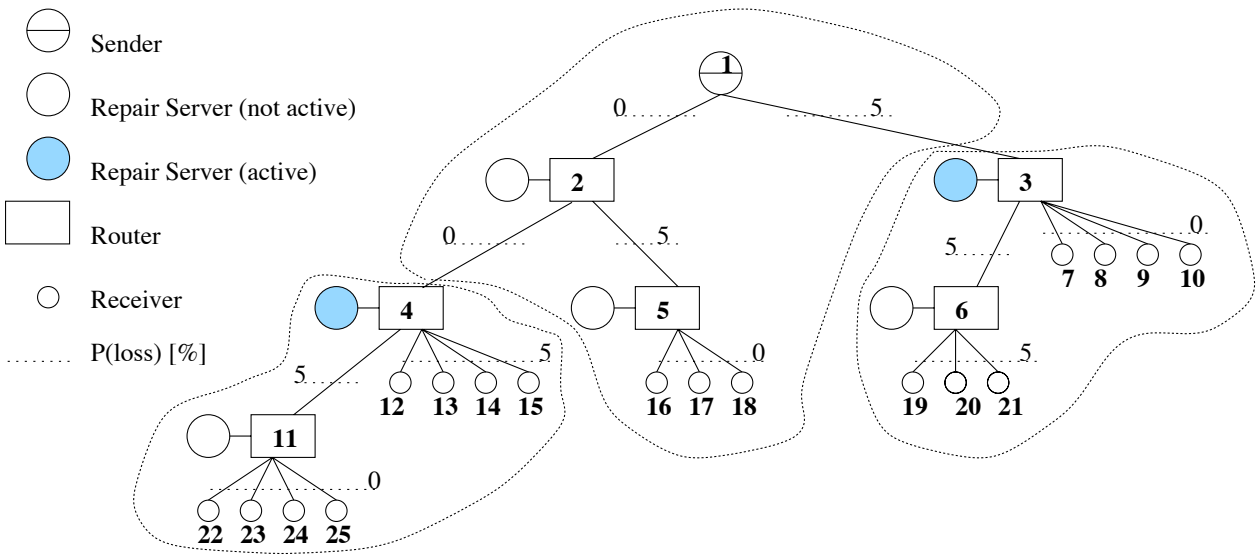


Figure 1: Example multicast tree

$\mathcal{N}(n)$, $n > 1$. $F_1(i)$ is the probability that at most i transmissions of a packet from the sender are needed before this packet is received by all receivers in $\mathcal{N}$.

Case 1: n is a leaf node.

We have

$$F_n(i) = Prob[R(n) \leq i] = 1 - p_n^i \tag{2}$$

where p_n^i is the probability that none of i transmissions of a packet from $\mathcal{P}(n)$ reach node n.

Case 2: n is not a leaf node.

$F_n(i)$ can be computed in terms of p_n and $F_k(i)$, $k \in \mathcal{C}(n)$. Assuming a packet is sent i times from $\mathcal{P}(n)$ to n, let $A_n(i)$ be the number of successful transmissions. $A_n(i)$ follows a binomial distribution:

$$P[A_n(i) = i - u] = \left(\begin{array}{c} i \\ u \end{array} \right) p_n^u (1 - p_n)^{i-u}$$

Accounting for independent losses at links leading to nodes in $\mathcal{C}(n)$, we get

$$P[R(n) \leq i \mid A_n(i) = i - u] = \Pi_{k \in \mathcal{C}(n)} F_k(i - u)$$

We remove the condition to obtain:

$$\begin{aligned} F_n(i) &= \sum_{u=0}^{i-1} P[R(n) \leq i \mid A_n(i) = i - u] \; P(A_n(i) = i - u) \\ &= \sum_{u=0}^{i-1} \left(\begin{array}{c} i \\ u \end{array} \right) p_n^u (1 - p_n)^{i-u} \; \Pi_{k \in \mathcal{C}(n)} F_k(i - u) \end{aligned} \tag{3}$$

In our analysis we will compute $F_n(i)$, where $n \in \{1\} \bigcup \mathcal{S}$ is the sender or a router with an active RS. The above formulas apply with $p_n = 0$.

The expected number of transmissions of a packet from n is

$$E[R(n)] = \sum_{i=0}^{\infty} (1 - F_n(i)) \tag{4}$$

If n is a router with an activated Repair Server, it has a finite buffer. Then there is a constraint, say $I-1$, on the number of retransmissions that can be provided by the Repair Server:

$$\begin{aligned} E[\#\text{ retransmissions provided by the RS}] &= \sum_{i=0}^{I} i \cdot (F_n(i) - F_n(i-1)) - 1 \\ &= I \cdot F_n(I) + \sum_{i=1}^{I-1} i \cdot F_n(i) - F_n(0) - \sum_{i=1}^{I-1} (i+1) \cdot F_n(i) - 1 \\ &= I \cdot F_n(I) - \sum_{i=0}^{I-1} F_n(i) - 1 \end{aligned} \tag{5}$$

The additional number of retransmissions (which could not be resolved by using the RS) needed to ensure reliable multicast in the subtree under n is

$$\begin{aligned} E[\#\text{ additional retransmissions}] &= E[R(n)] - E[\#\text{ retransmissions provided by the RS}] - 1 \\ &= \sum_{i=0}^{\infty} (1 - F_n(i)) - \left[I \cdot F_n(I) - \sum_{i=0}^{I-1} F_n(i) - 1 \right] - 1 \\ &= I(1 - F_n(I)) + \sum_{i=I}^{\infty} (1 - F_n(i)) \end{aligned} \tag{6}$$

4.3 Buffer usage

We assume each RS has infinite buffer capacity available, but due to nonzero buffer cost, only a certain amount will be used. Since packet interarrival times are random, the actual buffer occupancy will vary. In this section we determine B_n, the expected buffer required in order to ensure

$$F_n(i) \quad \geq \quad 1 - \varepsilon, \quad 0 \leq \varepsilon < 1 \tag{7}$$

In other words, RS_n will in average store B_n packets to achieve Prob[packet recovery at a RS] $\geq 1 - \varepsilon$.

Let I_n be the smallest integer i that satisfies requirement (7), By using (3) we see that I_n is found as the smallest i that satisfies

$$\sum_{u=0}^{i-1} \left(\begin{array}{c} i \\ u \end{array} \right) p_n^u (1 - p_n)^{i-u} \, \Pi_{k \in \mathcal{C}(n)} F_k(i-u) \quad \geq \quad 1 - \varepsilon$$

Then I_n is also the smallest number of transmissions of a packet (first transmission from the sender + $I_n - 1$ retransmissions from RS_n) needed to achieve Prob[packet recovery at a RS_n] $\geq 1 - \varepsilon$. In the following we shall express B_n as a function of I_n. With expected downstream packet inter-arrival time equal to Δ and expected NAK inter-arrival time (request for a specific packet) equal to $\Delta'(n)$, the expected buffer size is

$$B_n = \max(I_n - 1, 1) \frac{\Delta'(n)}{\Delta} \tag{8}$$

This can be written as

$$B_n = \max(I_n - 1, 1) \, 2 \, d(n) \, K \tag{9}$$

where $d(n)$ is the maximum depth (in number of links) in the subtree rooted at node n, and K is a constant depending a number of factors such as congestion control mechanism at the sender and link rates. If we assume that all link rates are equal, and that the sender rate is constant (i.e. no congestion control mechanism is applied), Eq. (9) may be simplified by letting K be equal to 1. Note that when RS_n is active we require that a buffer of expected size $B_n = 2 \, d(n)$ will be allocated even if no retransmissions are needed ($I_n = 1$).

node	1	2	3	4	5	6	7	8	9	10	11	12	13
q_n[%]	-	0	5	0	5	5	0	0	0	0	5	5	5
node	14	15	16	17	18	19	20	21	22	23	24	25	
q_n[%]	5	5	0	0	0	5	5	5	0	0	0	0	

Table 1: Static link loss probability in case study

symbol	value(s)	
ε	$0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1$	Prob(recovery failure at local RS)
c_{RS}/c_t	0.5	RS processing cost over unit transmission cost
c_b/c_t	0 - 0.5	Unit buffer cost over unit transmission cost

Table 2: Parameter values for case study

4.4 Cost functions

To evaluate the usefulness of RSs, we associate costs with transmission, buffering, and processing in order to find the cost for multicasting one packet from the sender to all the receivers. This is presented below as C_{RS} for the case when RSs are used, and as C_{noRS} for the case when no RS is used.

Transmission cost

$$C_t = \sum_{n \in \{1\} \bigcup \mathcal{S}} E[R(n)]\, E[L(n)]\, c_t \tag{10}$$

Here $E[R(n)]$ is the expected number of transmissions from Repair Server n, or from the Sender if $n = 1$. and $E[L(n)]$ is the expected number of links traveled by a packet in subtree $\mathcal{T}(n)$ when multicast from node n.

Buffering cost

$$C_b = \sum_{n \in \mathcal{S}} B_n c_b \tag{11}$$

Fixed RS cost To account for processing costs associated with a Repair Server, we specify a fixed cost c_{RS} per active RS per packet originally transmitted from the sender

Total normalized cost per multicast packet originally transmitted from the sender is the sum of buffering, transmission, and fixed cost, divided by the unit transmission cost c_t:

$$C_{RS} = \sum_{n \in \{1\} \bigcup \mathcal{S}} E[R(n)]E[L(n)] + \sum_{n \in \mathcal{S}} B_n\, c_b/c_t + \sum_{n \in \mathcal{S}} c_{RS}/c_t \tag{12}$$

Total normalized cost without active Repair Servers per multicast packet originally transmitted from the sender can be found by using Equation (4) in the absence of any RS, i.e. with $\mathcal{S} = \emptyset$. We obtain

$$C_{no\ RS} = E[R(1)]\, E[L] \tag{13}$$

Since only transmission costs are present, this is the total normalized cost when not using Repair Servers. $E[L]$ is the expected number of links traveled by a packet when multicast from the sender.

5 Case study

In this section we assume static link loss probabilities as shown in Table 1. Other parameter values are shown in Table 2.

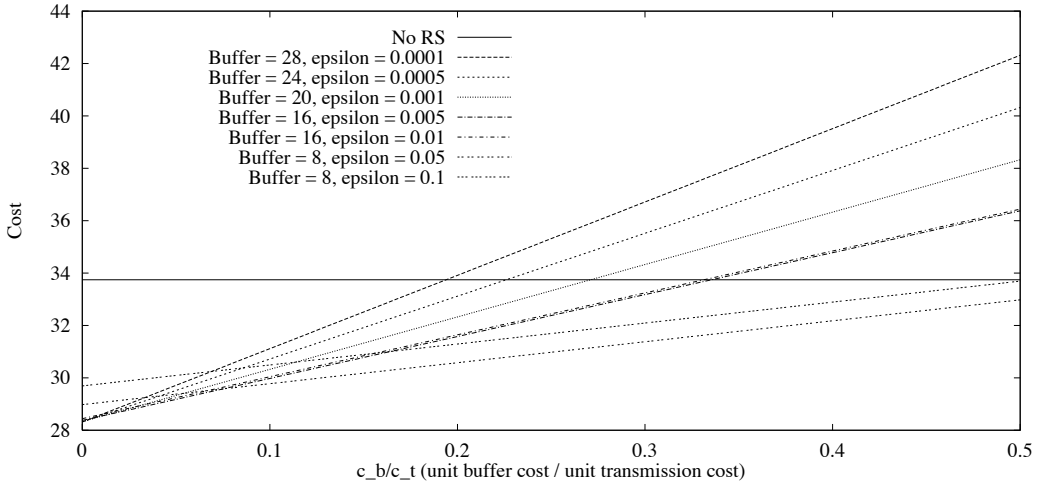


Figure 2: Cost for providing reliable multicast

5.1 Permanent RS allocation

In this section we present performance measures for the RS allocation in Figure 1, with only RS_3 and RS_4 activated. Figure 2 shows total normalized cost C_{RS}, the diagonal lines corresponding to various buffer sizes. The horizontal line is the total normalized cost $C_{no\ RS}$ for the case when no RS is active. We observe the following:

- using RSs is cost efficient as long as c_b/c_t is less than 0.2
- if we use RSs, a small amount of buffer is sufficient to significantly reduce costs, even when buffer is relatively expensive
- a larger buffer is better than a small buffer only when c_b/c_t is less than 0.05. However, a larger buffer gives a lower delay in failure recovery, a result not shown in the graph

Figures 3 and 4 show the expected buffer occupancy in the tree, $B_3 + B_4$, and the expected number of retransmissions due to local recovery failure, respectively.

- Figure 3 shows that the expected buffer occupancy in the tree increases as we decrease ε, the probability of local recovery failure. This correlation is intuitive since increased buffer at a RS makes it more likely to recover a packet from the RS. Figure 4 shows the relation between the expected buffer occupancy and the expected number of additional retransmissions (retransmissions that could not be resolved by using the RS). We see that the number of additional retransmissions, and thereby also the packet latency, decreases as the expected buffer occupancy increases.

5.2 Cost for different RS allocations

In this section we investigate all RS allocations possible in our multicast tree in order to find a cost-optimal configuration. $\mathcal{R} = \{1, 2, 3, 4, 5, 6, 11\}$ is the set of routers including the sender. Let U be a bit-vector of length $|\mathcal{R}|$:

$$
\begin{aligned}
U_1 &= 1 \\
U_i &= \begin{cases} 1 & \text{if } \mathcal{R}_i \text{ has an active RS} \\ 0 & \text{otherwise} \end{cases} \qquad i = 2..7
\end{aligned}
$$

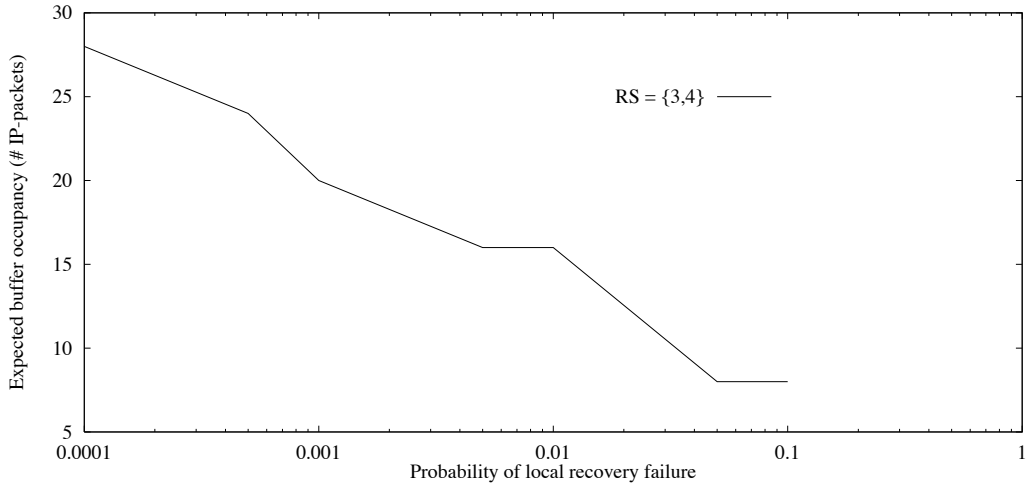


Figure 3: Expected buffer occupancy $B_3 + B_4$

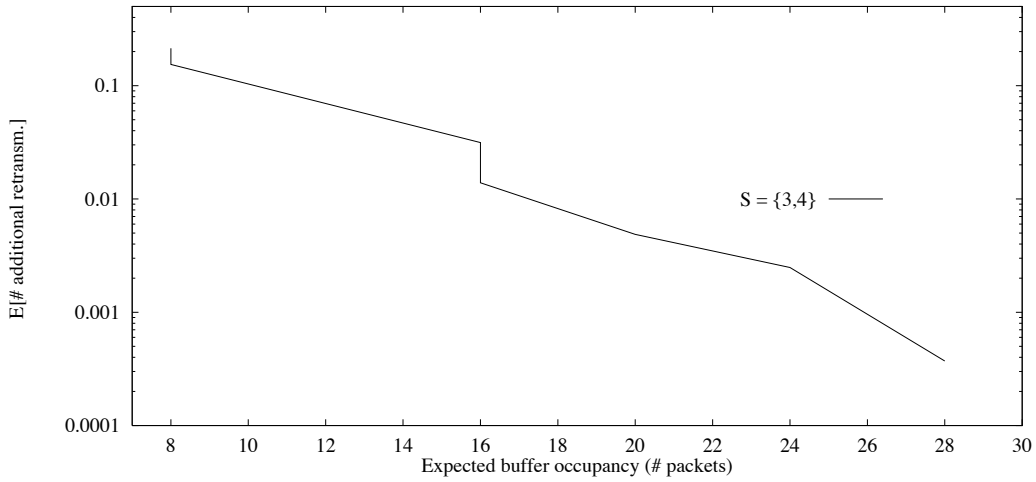


Figure 4: Expected number of additional retransmissions (retransmissions that could not be served by the local RS)

In our example we get $2^6 = 64$ different configurations. Furthermore let these configurations be indexed by

$$J_S = 1 + \sum_{j=2}^{7} 2^{j-2}\, U_j$$

For the tree in Figure 1 we have $U = \{1,0,1,1,0,0,0\}$ and $J_S = 7$. $U = \{1,0,0,0,0,0,0\}$ ($J_S = 1$) corresponds to the case that no RS is active, and $U = \{1,1,1,1,1,1,1\}$ ($J_S = 64$) to the case that all RSs are active.

Costs in the following three plots are calculated by means of (12) with $\varepsilon = 0.01$. Figure 5 shows the total cost for all RS allocations. A mapping table from x-axis value (J_S) to $\mathcal{S}$ may be found below the figure. The different curves in the each graph correspond to the fraction c_b/c_t which takes the values 0.01 (lower curve), 0.1, 0.2, and 0.3 (upper curve). Cost for the case with no RS is shown as a horizontal line. As may be seen, using Repair Servers is less costly than $C_{no\ RS}$ for most parameter values and RS allocations.

The sequence of RS allocations is permuted according to # active RSs in Figure 6. Costs for $\mathcal{S} = \emptyset$ are on the far left end of the graph, while costs for $\mathcal{S} = \{2,3,4,5,6,11\}$ are on the far right. As we read the graph from left to right, there is a weak trend indicating decreasing cost. However, the graph is very rugged and we cannot conclude that more active RSs necessarily reduce the total cost.

In Figure 7, the sequence of RS allocations is sorted according to cost for the case when $c_b/c_t = 0.1$. The lowest cost is achieved when $\mathcal{S} = \{2,4,6,11\}$. Since there is no loss below node 11, it seems paradoxical that this router should have an active RS. The answer lies in the fact that without node 11 in $\mathcal{S}$, the buffer at node 4 would have needed to be larger in order to cover a subtree of depth 2 rather than depth 1. Furthermore, since there is no loss below node 11, this node has no pending NAKs. Therefore it "filters" out retransmissions sent from node 4, and the overall transmission cost is reduced. As an illustration, assume a packet is lost only at node 15. With $\mathcal{S} = \{2,4,6\}$, this packet will be retransmitted from node 4 to 9 receivers: ($\{11-15, 22-25\}$), but if $\mathcal{S} = \{2,4,6,11\}$ it will only be retransmitted to 5 receivers ($\{11-15\}$).

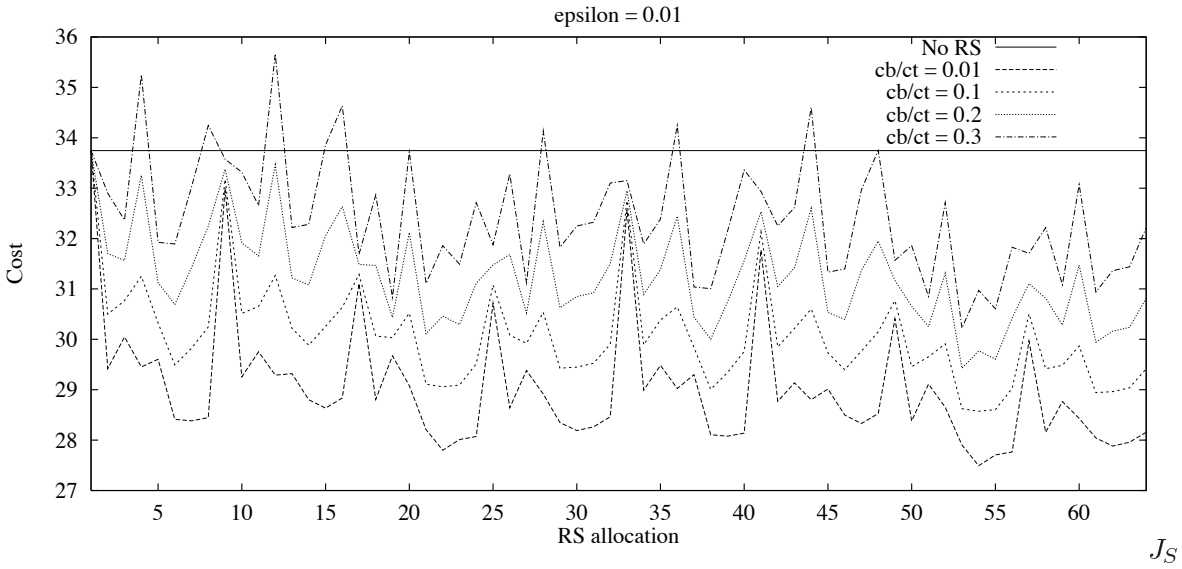


Figure 5: Cost vs RS allocation. Horizontal line represents cost without RS. Mapping from x-axis index to RS allocation, see table below

J_S	U	$\mathcal{S}$	J_S	U	$\mathcal{S}$
1	1, 0, 0, 0, 0, 0, 0		33	1, 0, 0, 0, 0, 0, 1	11
2	1, 1, 0, 0, 0, 0, 0	2	34	1, 1, 0, 0, 0, 0, 1	2, 11
3	1, 0, 1, 0, 0, 0, 0	3	35	1, 0, 1, 0, 0, 0, 1	3, 11
4	1, 1, 1, 0, 0, 0, 0	2, 3	36	1, 1, 1, 0, 0, 0, 1	2, 3, 11
5	1, 0, 0, 1, 0, 0, 0	4	37	1, 0, 0, 1, 0, 0, 1	4, 11
6	1, 1, 0, 1, 0, 0, 0	2, 4	38	1, 1, 0, 1, 0, 0, 1	2, 4, 11
7	1, 0, 1, 1, 0, 0, 0	3, 4	39	1, 0, 1, 1, 0, 0, 1	3, 4, 11
8	1, 1, 1, 1, 0, 0, 0	2, 3, 4	40	1, 1, 1, 1, 0, 0, 1	2, 3, 4, 11
9	1, 0, 0, 0, 1, 0, 0	5	41	1, 0, 0, 0, 1, 0, 1	5, 11
10	1, 1, 0, 0, 1, 0, 0	2, 5	42	1, 1, 0, 0, 1, 0, 1	2, 5, 11
11	1, 0, 1, 0, 1, 0, 0	3, 5	43	1, 0, 1, 0, 1, 0, 1	3, 5, 11
12	1, 1, 1, 0, 1, 0, 0	2, 3, 5	44	1, 1, 1, 0, 1, 0, 1	2, 3, 5, 11
13	1, 0, 0, 1, 1, 0, 0	4, 5	45	1, 0, 0, 1, 1, 0, 1	4, 5, 11
14	1, 1, 0, 1, 1, 0, 0	2, 4, 5	46	1, 1, 0, 1, 1, 0, 1	2, 4, 5, 11
15	1, 0, 1, 1, 1, 0, 0	3, 4, 5	47	1, 0, 1, 1, 1, 0, 1	3, 4, 5, 11
16	1, 1, 1, 1, 1, 0, 0	2, 3, 4, 5	48	1, 1, 1, 1, 1, 0, 1	2, 3, 4, 5, 11
17	1, 0, 0, 0, 0, 1, 0	6	49	1, 0, 0, 0, 0, 1, 1	6, 11
18	1, 1, 0, 0, 0, 1, 0	2, 6	50	1, 1, 0, 0, 0, 1, 1	2, 6, 11
19	1, 0, 1, 0, 0, 1, 0	3, 6	51	1, 0, 1, 0, 0, 1, 1	3, 6, 11
20	1, 1, 1, 0, 0, 1, 0	2, 3, 6	52	1, 1, 1, 0, 0, 1, 1	2, 3, 6, 11
21	1, 0, 0, 1, 0, 1, 0	4, 6	53	1, 0, 0, 1, 0, 1, 1	4, 6, 11
22	1, 1, 0, 1, 0, 1, 0	2, 4, 6	54	1, 1, 0, 1, 0, 1, 1	2, 4, 6, 11
23	1, 0, 1, 1, 0, 1, 0	3, 4, 6	55	1, 0, 1, 1, 0, 1, 1	3, 4, 6, 11
24	1, 1, 1, 1, 0, 1, 0	2, 3, 4, 6	56	1, 1, 1, 1, 0, 1, 1	2, 3, 4, 6, 11
25	1, 0, 0, 0, 1, 1, 0	5, 6	57	1, 0, 0, 0, 1, 1, 1	5, 6, 11
26	1, 1, 0, 0, 1, 1, 0	2, 5, 6	58	1, 1, 0, 0, 1, 1, 1	2, 5, 6, 11
27	1, 0, 1, 0, 1, 1, 0	3, 5, 6	59	1, 0, 1, 0, 1, 1, 1	3, 5, 6, 11
28	1, 1, 1, 0, 1, 1, 0	2, 3, 5, 6	60	1, 1, 1, 0, 1, 1, 1	2, 3, 5, 6, 11
29	1, 0, 0, 1, 1, 1, 0	4, 5, 6	61	1, 0, 0, 1, 1, 1, 1	4, 5, 6, 11
30	1, 1, 0, 1, 1, 1, 0	2, 4, 5, 6	62	1, 1, 0, 1, 1, 1, 1	2, 4, 5, 6, 11
31	1, 0, 1, 1, 1, 1, 0	3, 4, 5, 6	63	1, 0, 1, 1, 1, 1, 1	3, 4, 5, 6, 11
32	1, 1, 1, 1, 1, 1, 0	2, 3, 4, 5, 6	64	1, 1, 1, 1, 1, 1, 1	2, 3, 4, 5, 6, 11

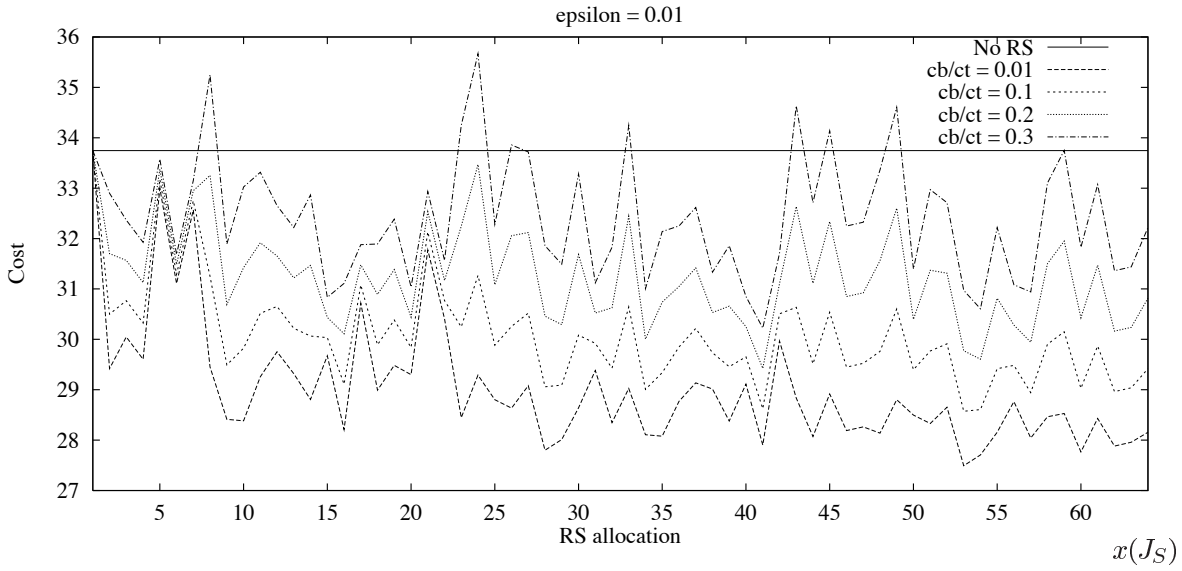


Figure 6: Cost vs RS allocation, which is permuted according to number of routers with active RS. Horizontal line represents cost without RS. Mapping from x-axis index to RS allocation, see table below

x-axis	J_S	U	$\mathcal{S}$	x-axis	J_S	U	$\mathcal{S}$
1	1	1, 0, 0, 0, 0, 0, 0		33	36	1, 1, 1, 0, 0, 0, 1	2, 3, 11
2	2	1, 1, 0, 0, 0, 0, 0	2	34	38	1, 1, 0, 1, 0, 0, 1	2, 4, 11
3	3	1, 0, 1, 0, 0, 0, 0	3	35	39	1, 0, 1, 1, 0, 0, 1	3, 4, 11
4	5	1, 0, 0, 1, 0, 0, 0	4	36	42	1, 1, 0, 0, 1, 0, 1	2, 5, 11
5	9	1, 0, 0, 0, 1, 0, 0	5	37	43	1, 0, 1, 0, 1, 0, 1	3, 5, 11
6	17	1, 0, 0, 0, 0, 1, 0	6	38	45	1, 0, 0, 1, 1, 0, 1	4, 5, 11
7	33	1, 0, 0, 0, 0, 0, 1	11	39	50	1, 1, 0, 0, 0, 1, 1	2, 6, 11
8	4	1, 1, 1, 0, 0, 0, 0	2, 3	40	51	1, 0, 1, 0, 0, 1, 1	3, 6, 11
9	6	1, 1, 0, 1, 0, 0, 0	2, 4	41	53	1, 0, 0, 1, 0, 1, 1	4, 6, 11
10	7	1, 0, 1, 1, 0, 0, 0	3, 4	42	57	1, 0, 0, 0, 1, 1, 1	5, 6, 11
11	10	1, 1, 0, 0, 1, 0, 0	2, 5	43	16	1, 1, 1, 1, 1, 0, 0	2, 3, 4, 5
12	11	1, 0, 1, 0, 1, 0, 0	3, 5	44	24	1, 1, 1, 1, 0, 1, 0	2, 3, 4, 6
13	13	1, 0, 0, 1, 1, 0, 0	4, 5	45	28	1, 1, 1, 0, 1, 1, 0	2, 3, 5, 6
14	18	1, 1, 0, 0, 0, 1, 0	2, 6	46	30	1, 1, 0, 1, 1, 1, 0	2, 4, 5, 6
15	19	1, 0, 1, 0, 0, 1, 0	3, 6	47	31	1, 0, 1, 1, 1, 1, 0	3, 4, 5, 6
16	21	1, 0, 0, 1, 0, 1, 0	4, 6	48	40	1, 1, 1, 1, 0, 0, 1	2, 3, 4, 11
17	25	1, 0, 0, 0, 1, 1, 0	5, 6	49	44	1, 1, 1, 0, 1, 0, 1	2, 3, 5, 11
18	34	1, 1, 0, 0, 0, 0, 1	2, 11	50	46	1, 1, 0, 1, 1, 0, 1	2, 4, 5, 11
19	35	1, 0, 1, 0, 0, 0, 1	3, 11	51	47	1, 0, 1, 1, 1, 0, 1	3, 4, 5, 11
20	37	1, 0, 0, 1, 0, 0, 1	4, 11	52	52	1, 1, 1, 0, 0, 1, 1	2, 3, 6, 11
21	41	1, 0, 0, 0, 1, 0, 1	5, 11	53	54	1, 1, 0, 1, 0, 1, 1	2, 4, 6, 11
22	49	1, 0, 0, 0, 0, 1, 1	6, 11	54	55	1, 0, 1, 1, 0, 1, 1	3, 4, 6, 11
23	8	1, 1, 1, 1, 0, 0, 0	2, 3, 4	55	58	1, 1, 0, 0, 1, 1, 1	2, 5, 6, 11
24	12	1, 1, 1, 0, 1, 0, 0	2, 3, 5	56	59	1, 0, 1, 0, 1, 1, 1	3, 5, 6, 11
25	14	1, 1, 0, 1, 1, 0, 0	2, 4, 5	57	61	1, 0, 0, 1, 1, 1, 1	4, 5, 6, 11
26	15	1, 0, 1, 1, 1, 0, 0	3, 4, 5	58	32	1, 1, 1, 1, 1, 1, 0	2, 3, 4, 5, 6
27	20	1, 1, 1, 0, 0, 1, 0	2, 3, 6	59	48	1, 1, 1, 1, 1, 0, 1	2, 3, 4, 5, 11
28	22	1, 1, 0, 1, 0, 1, 0	2, 4, 6	60	56	1, 1, 1, 1, 0, 1, 1	2, 3, 4, 6, 11
29	23	1, 0, 1, 1, 0, 1, 0	3, 4, 6	61	60	1, 1, 1, 0, 1, 1, 1	2, 3, 5, 6, 11
30	26	1, 1, 0, 0, 1, 1, 0	2, 5, 6	62	62	1, 1, 0, 1, 1, 1, 1	2, 4, 5, 6, 11
31	27	1, 0, 1, 0, 1, 1, 0	3, 5, 6	63	63	1, 0, 1, 1, 1, 1, 1	3, 4, 5, 6, 11
32	29	1, 0, 0, 1, 1, 1, 0	4, 5, 6	64	64	1, 1, 1, 1, 1, 1, 1	2, 3, 4, 5, 6, 11

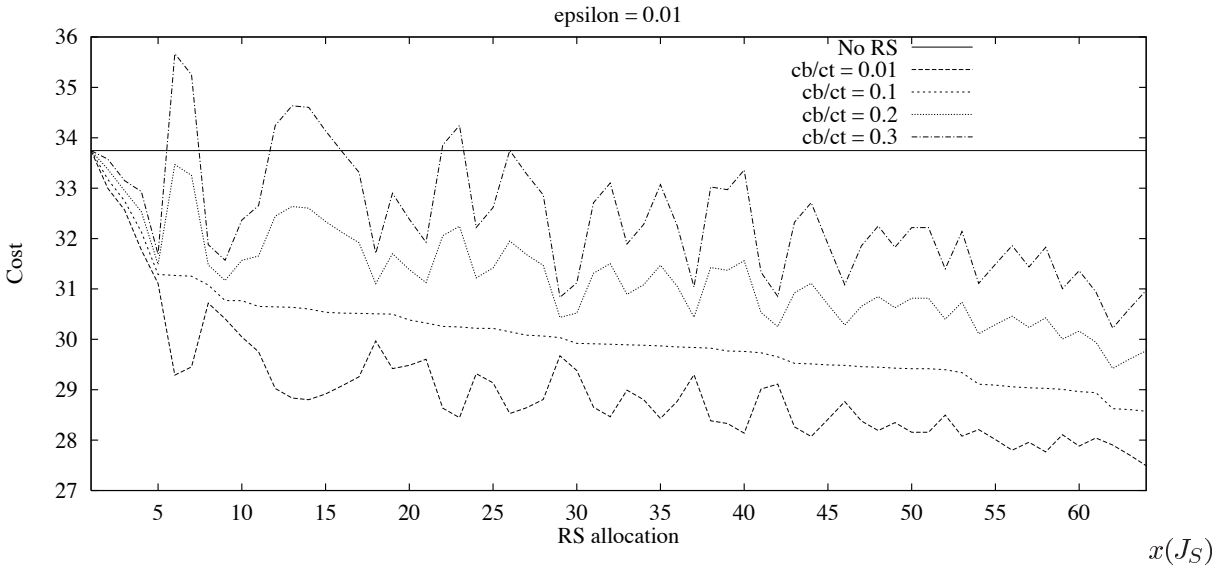


Figure 7: Cost. RS allocation is permuted according to cost for the series with $c_b/c_t = 0.1$. Mapping from x-axis index to RS allocation, see table below

x-axis	J_S	U	$\mathcal{S}$	x-axis	J_S	U	$\mathcal{S}$
1	1	1, 0, 0, 0, 0, 0, 0		33	34	1, 1, 0, 0, 0, 0, 1	2, 11
2	9	1, 0, 0, 0, 1, 0, 0	5	34	14	1, 1, 0, 1, 1, 0, 0	2, 4, 5
3	33	1, 0, 0, 0, 0, 0, 1	11	35	60	1, 1, 1, 0, 1, 1, 1	2, 3, 5, 6, 11
4	41	1, 0, 0, 0, 1, 0, 1	5, 11	36	42	1, 1, 0, 0, 1, 0, 1	2, 5, 11
5	17	1, 0, 0, 0, 0, 1, 0	6	37	37	1, 0, 0, 1, 0, 0, 1	4, 11
6	12	1, 1, 1, 0, 1, 0, 0	2, 3, 5	38	7	1, 0, 1, 1, 0, 0, 0	3, 4
7	4	1, 1, 1, 0, 0, 0, 0	2, 3	39	47	1, 0, 1, 1, 1, 0, 1	3, 4, 5, 11
8	25	1, 0, 0, 0, 1, 1, 0	5, 6	40	40	1, 1, 1, 1, 0, 0, 1	2, 3, 4, 11
9	49	1, 0, 0, 0, 0, 1, 1	6, 11	41	45	1, 0, 0, 1, 1, 0, 1	4, 5, 11
10	3	1, 0, 1, 0, 0, 0, 0	3	42	51	1, 0, 1, 0, 0, 1, 1	3, 6, 11
11	11	1, 0, 1, 0, 1, 0, 0	3, 5	43	31	1, 0, 1, 1, 1, 1, 0	3, 4, 5, 6
12	36	1, 1, 1, 0, 0, 0, 1	2, 3, 11	44	24	1, 1, 1, 1, 0, 1, 0	2, 3, 4, 6
13	16	1, 1, 1, 1, 1, 0, 0	2, 3, 4, 5	45	6	1, 1, 0, 1, 0, 0, 0	2, 4
14	44	1, 1, 1, 0, 1, 0, 1	2, 3, 5, 11	46	59	1, 0, 1, 0, 1, 1, 1	3, 5, 6, 11
15	28	1, 1, 1, 0, 1, 1, 0	2, 3, 5, 6	47	50	1, 1, 0, 0, 0, 1, 1	2, 6, 11
16	20	1, 1, 1, 0, 0, 1, 0	2, 3, 6	48	30	1, 1, 0, 1, 1, 1, 0	2, 4, 5, 6
17	10	1, 1, 0, 0, 1, 0, 0	2, 5	49	29	1, 0, 0, 1, 1, 1, 0	4, 5, 6
18	57	1, 0, 0, 0, 1, 1, 1	5, 6, 11	50	64	1, 1, 1, 1, 1, 1, 1	2, 3, 4, 5, 6, 11
19	2	1, 1, 0, 0, 0, 0, 0	2	51	58	1, 1, 0, 0, 1, 1, 1	2, 5, 6, 11
20	35	1, 0, 1, 0, 0, 0, 1	3, 11	52	46	1, 1, 0, 1, 1, 0, 1	2, 4, 5, 11
21	5	1, 0, 0, 1, 0, 0, 0	4	53	39	1, 0, 1, 1, 0, 0, 1	3, 4, 11
22	15	1, 0, 1, 1, 1, 0, 0	3, 4, 5	54	21	1, 0, 0, 1, 0, 1, 0	4, 6
23	8	1, 1, 1, 1, 0, 0, 0	2, 3, 4	55	23	1, 0, 1, 1, 0, 1, 0	3, 4, 6
24	13	1, 0, 0, 1, 1, 0, 0	4, 5	56	22	1, 1, 0, 1, 0, 1, 0	2, 4, 6
25	43	1, 0, 1, 0, 1, 0, 1	3, 5, 11	57	63	1, 0, 1, 1, 1, 1, 1	3, 4, 5, 6, 11
26	48	1, 1, 1, 1, 1, 0, 1	2, 3, 4, 5, 11	58	56	1, 1, 1, 1, 0, 1, 1	2, 3, 4, 6, 11
27	26	1, 1, 0, 0, 1, 1, 0	2, 5, 6	59	38	1, 1, 0, 1, 0, 0, 1	2, 4, 11
28	18	1, 1, 0, 0, 0, 1, 0	2, 6	60	62	1, 1, 0, 1, 1, 1, 1	2, 4, 5, 6, 11
29	19	1, 0, 1, 0, 0, 1, 0	3, 6	61	61	1, 0, 0, 1, 1, 1, 1	4, 5, 6, 11
30	27	1, 0, 1, 0, 1, 1, 0	3, 5, 6	62	53	1, 0, 0, 1, 0, 1, 1	4, 6, 11
31	52	1, 1, 1, 0, 0, 1, 1	2, 3, 6, 11	63	55	1, 0, 1, 1, 0, 1, 1	3, 4, 6, 11
32	32	1, 1, 1, 1, 1, 1, 0	2, 3, 4, 5, 6	64	54	1, 1, 0, 1, 0, 1, 1	2, 4, 6, 11

6 A policy for activation and deactivation of Repair Servers

The investigations so far in this paper have assumed time-invariant link loss probabilities. As discussed in Section 2.2, this assumption does not usually hold. Packet loss on a link exhibits temporal fluctuations, and this must be taken into account when designing protocols for reliable multicast. In this section we propose a distributed, scalable, and adaptive policy for RS activation and deactivation that uses on-the-fly estimation of packet loss.

Policy description

The protocol used is in accordance with the main properties of the AER-protocol as described in Section 3. For estimation of packet loss probabilities, some additional functionality is needed.

Every Repair Server, active or not, traces the flow of packets through it's corresponding router during a control interval of τ seconds. During control interval k, a RS keeps the sequence numbers of packets multicast from it's corresponding router (only transmissions, not retransmissions). The RS also maintains a log of NAKs from it's subtree. A NAK for a specific packet is counted only once in the log. Furthermore, only NAKs corresponding to packets that were transmitted from the router during control interval k, contribute to the loss count. At the end of the control interval, the RS estimates the probability of packet loss by means of exponential smoothing:

$$\hat{p}_{loss}^{k} \quad = \quad \alpha \, \frac{\#\text{ lost packets}}{\#\text{ transmitted packets}} + (1-\alpha)\, \hat{p}_{loss}^{k-1}, \qquad k \geq 1,\ \hat{p}_{loss}^{0} = 0 \tag{14}$$

where α is the smoothing parameter, $0 \leq \alpha \leq 1$.

As discussed in Section 2.2, we want to capture the long term dynamics that give significant variations over several minutes, but filter out the loss bursts occurring in less than a second. These requirements may be accomplished by letting τ be around 5 seconds. To make the policy more efficient, τ may be scaled up or down, within bounds, according to slow or fast changes in the estimated loss probability.

The threshold based activation/deactivation scheme is shown in Table 3. The RS will be activated if the estimated loss probability exceeds t_h, and will be deactivated if the estimate drops below t_l. Setting the threshold values t_l and t_h will be a major challenge when tuning the policy for optimal performance. These values may be application specific. The difference $t_h - t_l$ will prevent a RS from flip-flopping between activated and deactivated state. When the actual loss in subtree $\mathcal{T}(n)$ exceeds t_h or drops below t_l, the loss estimation at RS_n will need a finite number of control intervals, say j_n, to detect this change. j_n depends on the smoothing parameter α.

When the loss pattern in the multicast tree changes, the RSs will pass through several rounds of switching on and off before a stable allocation is obtained. This may be understood by the example in Figure 8. Assume all RSs are initially deactivated, and one of the tail links is lossy (Figure 8a)). Since none of the RSs can recover lost packets, NAKs are going upstream. At the end of first control interval, all RSs will detect downstream loss, and will consequently be turned on (Figure 8b)). During the 2nd interval the RS at node m will recover losses, and the RS at nodes k and l will not detect any loss. Consequently these RS will deactivate (Figure 8c)). Here it is assumed that changes in the loss pattern is detected by the loss probability estimation during one control interval. In general, $j_n < \infty$ control intervals are needed for RS_n to detect this change. Following the arguments used above for Figure 8, it will in general take no more than $2 \max_{n \in \mathcal{S}} j_n$ control intervals before the RSs regain a stable allocation. In our examples the policy needs 4-6 control intervals to transit between stable RS allocations.

6.1 Simulations

The system shown in Figure 1 was simulated to evaluate the usefulness of the RS activation/deactivation scheme in Table 3. Changes in link loss are assumed to happen over an interval of length 30 seconds or more, and the control interval is set to 4 seconds.

Link loss is imposed in 4 stages as shown in Table 5. The four stages constitute a dynamic link loss pattern that repeats every five minutes. Packets are lost according to a Bernoulli loss model.

```
# (*) tau_min = 2 sec;
# (*) tau_max = 30 sec;
tau     = 4 sec;
P(loss_prev_int) = 0;
t0 = time;
REPEAT {
   WHILE (time-t0 < tau) DO {
      Count transmissions and NAKs;
      }
   P(loss) =  a (# NAKs / # transmissions) + (1-a) P(loss_prev_int);
   IF (P(loss) > t_h)  Activate RS;
   IF (P(loss) < t_l)  Deactivate RS;
   t0 = time;
   # change control interval tau:
   # (*) IF ( RelChange(P(loss_prev_int),P(loss)) > 50% ) tau = Max(tau/2, tau_min);
   # (*) IF ( RelChange(P(loss_prev_int),P(loss)) < 10% ) tau = Min(2*tau, tau_max);
   P(loss_prev_int) = P(loss);
}

# (*) Has not been implemented for the time being
```

Table 3: RS activation/deactivation scheme

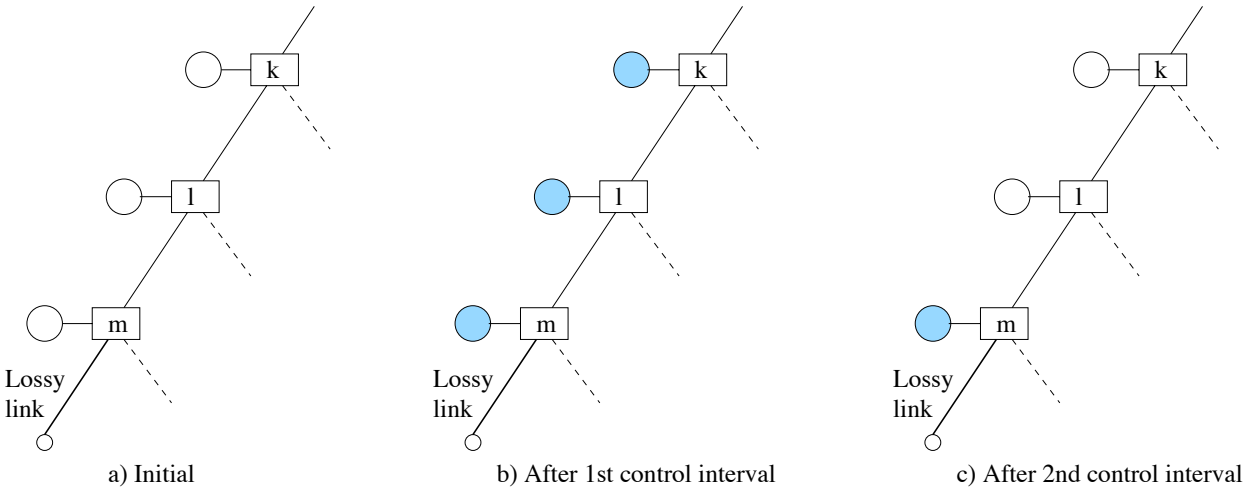


Figure 8: Activation and deactivation of RS based on estimation of downstream packet loss

symbol	value(s)	
t_l	0.07, 0.10	lower threshold in RS activation/deactivation algorithm
t_h	0.15, 0.20	upper threshold in RS activation/deactivation algorithm
τ	4 sec	control interval (constant)
α	0.5, 0.75	smoothing parameter in estimation of loss probability
	10 Mb/sec	link capacity (uniform over all links)
	1500 Byte	packet size

Table 4: Parameter values for simulation study

		node										
stage	duration [s]	1	2	3	4	5	6	7	8	9	10	11
1	60	0.0	0.0	5.0	0.0	5.0	5.0	0.0	0.0	0.0	0.0	5.0
2	90	0.0	10.0	2.5	10.0	2.5	2.5	10.0	10.0	10.0	10.0	2.5
3	30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	120	0.0	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5

		node													
stage	duration [s]	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	60	5.0	5.0	5.0	5.0	0.0	0.0	0.0	5.0	5.0	5.0	0.0	0.0	0.0	0.0
2	90	2.5	2.5	2.5	2.5	10.0	10.0	10.0	2.5	2.5	2.5	10.0	10.0	10.0	10.0
3	30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	120	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5

Table 5: Dynamic link loss probabilities in %

The issue of determining the buffer size at each RS has not been addressed in this section. The complication lies in the fact that after every reconfiguration of the RSs, the size of the different subtrees may change. Therefore RS_n must recalculate B_n rather frequently by using (8), and also reallocate buffer capacity. Since investigations of the policy's dynamic properties is the main scope of this section, a large fixed buffer is assumed to be available for every RS.

6.2 Discussion

Dynamic properties

The Figures 9 through 12 show that the policy reacts to changes in link loss as indicated in Figure 8. Consider e.g. the first 30 seconds in Figure 9b). Note that the loss pattern in this stage is as in Figure 1. RSs 2,3, and 4 become activated immediately, RS_6 shortly after. However, when RS_4 is active, RS_2 measures less loss (see Figure 9a)), and hence deactivates. The same occurs with RS_3. This is achieved without any explicit exchange of information between the RSs. In fact, the RSs communicate indirectly when measuring packet loss, and thereby gives the policy the advantageous property of avoiding situations where more RSs than necessary are active.

In transition between two loss stages, a time corresponding to 5 control intervals (20 sec) or less is needed for the Repair Servers to obtain a new, stable state. This phenomena can be observed in Figure 9b). The link loss pattern changes at $t = 300$ seconds, and system state changes from $\mathcal{S} = \{2, 3, 4\}$ to $\mathcal{S} = \{4, 6\}$ after approximately 20 seconds.

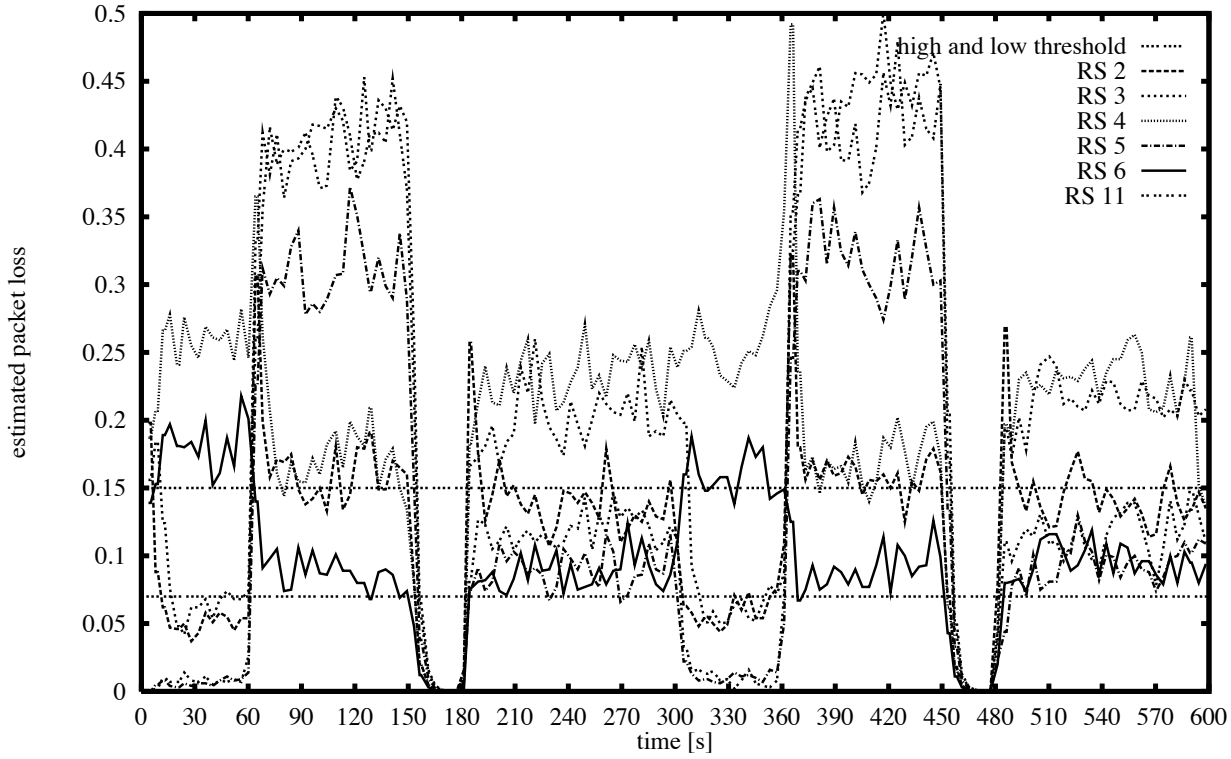
When it comes to the smoothing parameter, $\alpha = 0.5$ gives a more stable estimate and appears to be more suitable than $\alpha = 0.75$. However, choosing very low values for α will result in longer reaction time for the policy.

Cost considerations

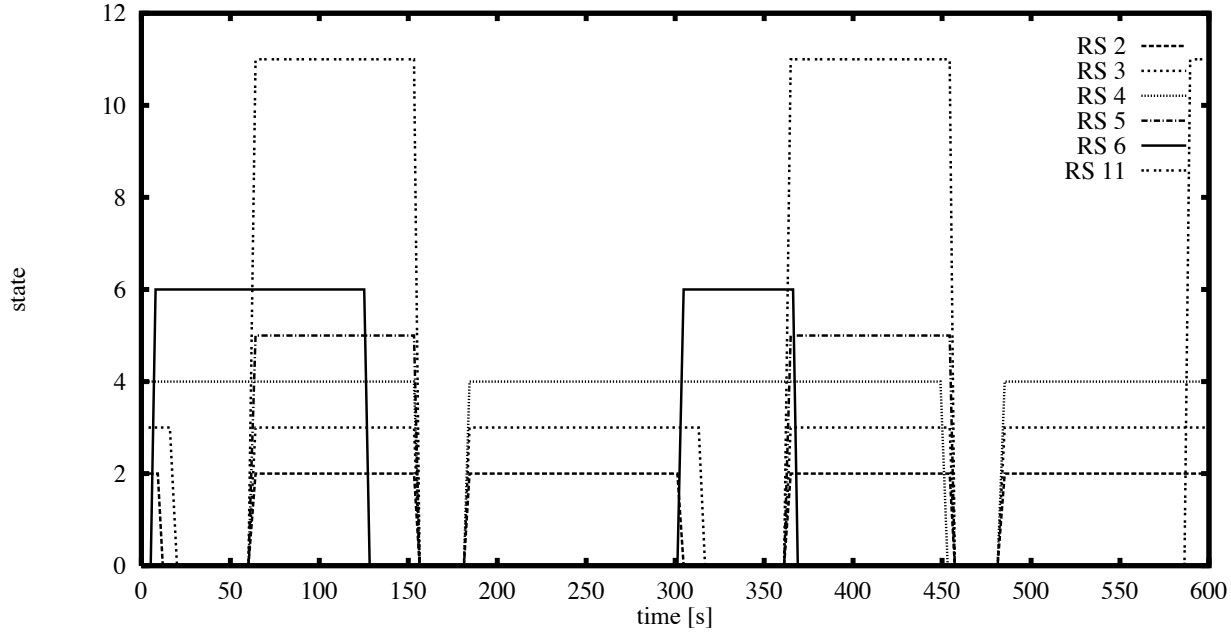
In stage 1 the link loss pattern is the same as in the case study in Section 5. Therefore cost considerations are only relevant for this stage. With threshold values equal to $t_l = 0.07$ and $t_h = 0.15$, the state $\mathcal{S} = \{4, 6\}$ is attained by the autonomous policy (see Figure 9b), time 0-60 sec). In the case when $c_b/c_t = 0.1$, the cost for this state is relatively close to the cost-optimal state $\mathcal{S} = \{2, 4, 6, 11\}$ (see Figure 7). In Figures 10 and 12 threshold values $t_l = 0.10$ and $t_h = 0.20$ have been used. RS allocations differ slightly from the previous case. In particular, during the first stage the state $\mathcal{S} = \{3, 4\}$ is attained. With $c_b/c_t = 0.1$ this state gives higher costs than with $\mathcal{S} = \{4, 6\}$, but it is still better than with no active RSs.

7 Conclusion and future work

Using Repair Servers (RS) is a promising method for efficient provisioning of reliable multicast. In this paper we have assumed that routers in a multicast tree may have a colocated RS. When activated, the RS will allocate buffering resources and store the most recent packets in the multicast stream.

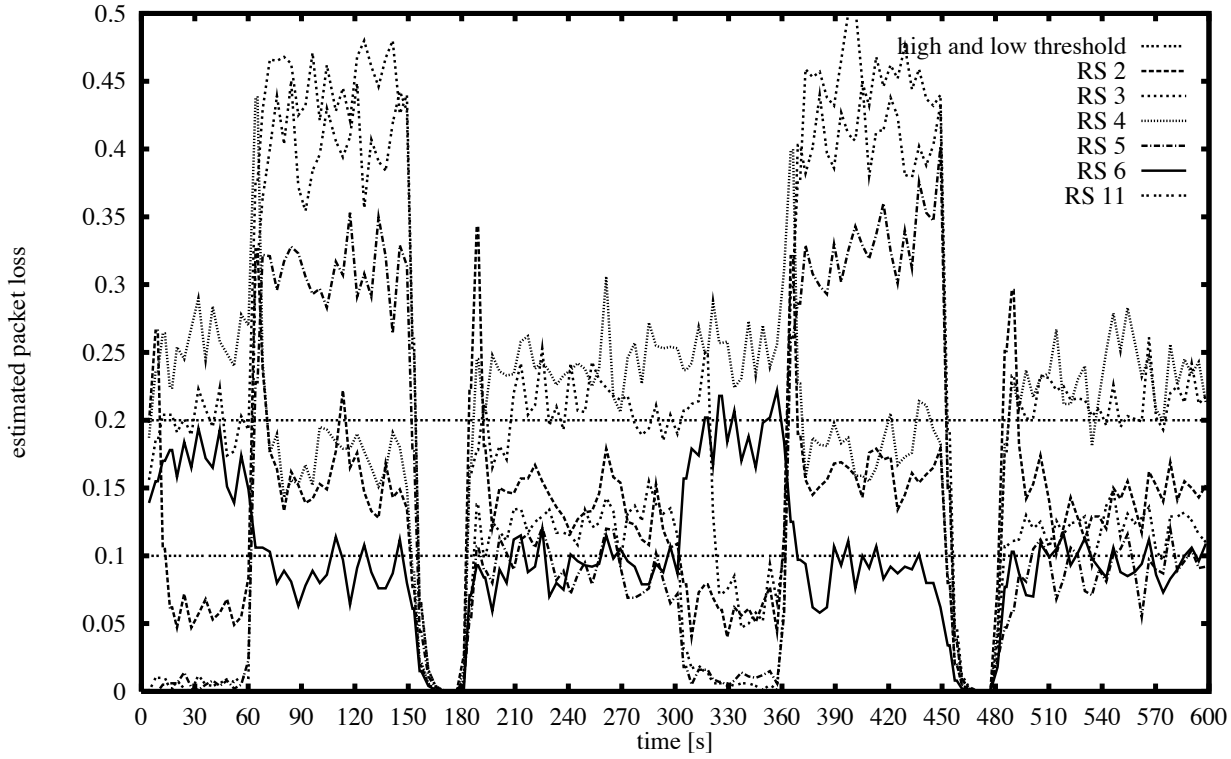


a) On-the-fly estimated downstream packet loss.

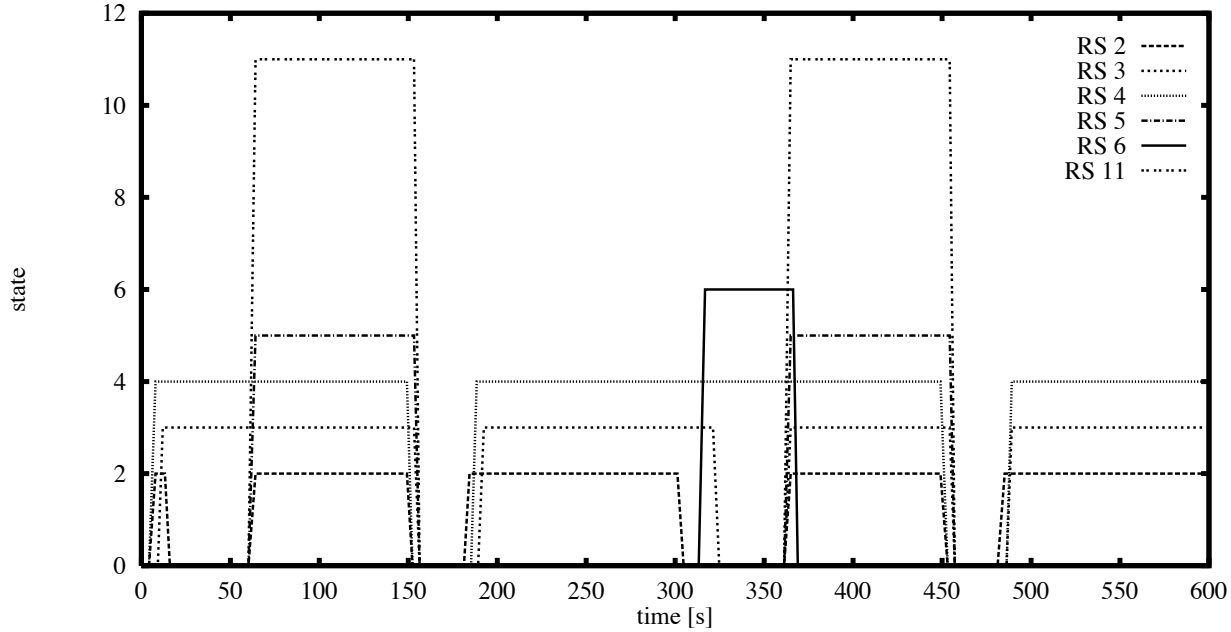


b) Dynamic RS activation and deactivation based on estimated downstream packet loss.

Figure 9: Parameters: $\alpha = 0.75,\ t_l = 0.07,\ t_h = 0.15$

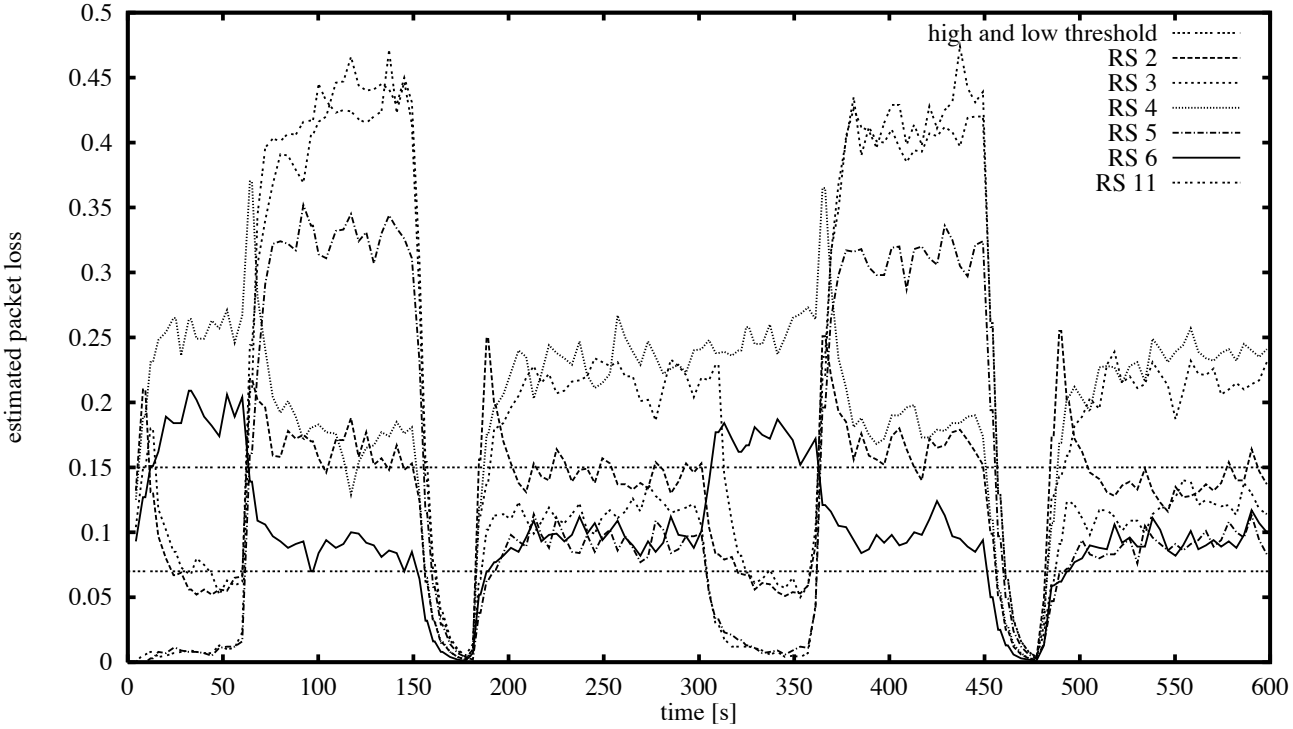


a) On-the-fly estimated downstream packet loss.

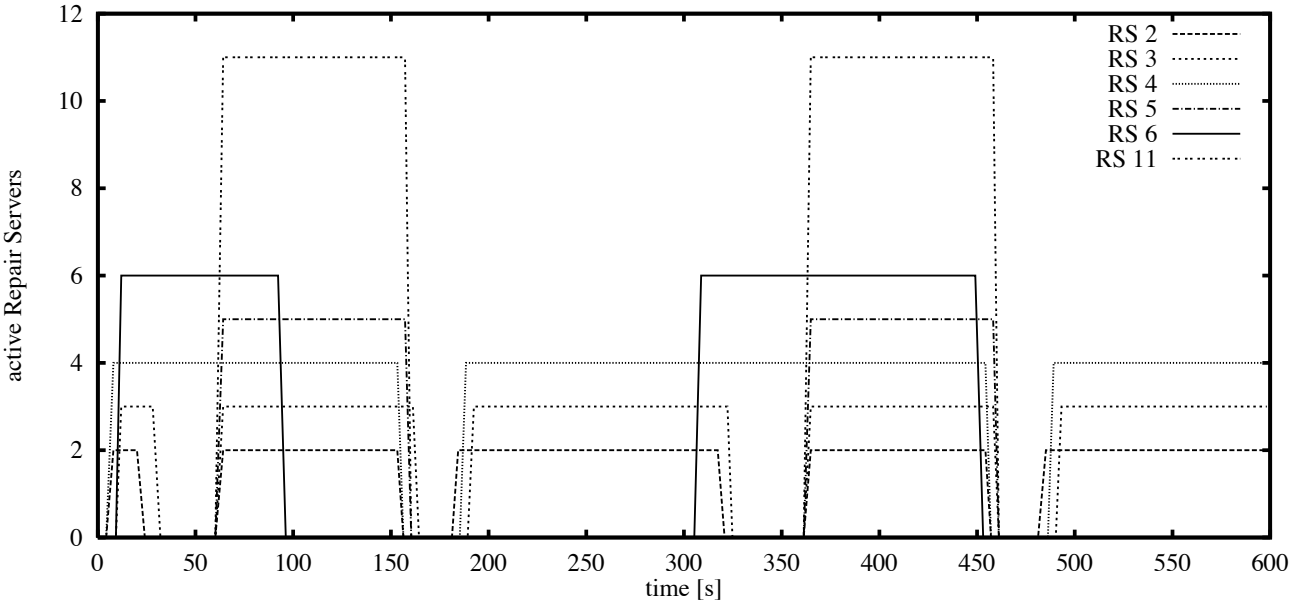


b) Dynamic RS activation and deactivation based on estimated downstream packet loss.

Figure 10: Parameters: $\alpha = 0.75,\ t_l = 0.10,\ t_h = 0.20$

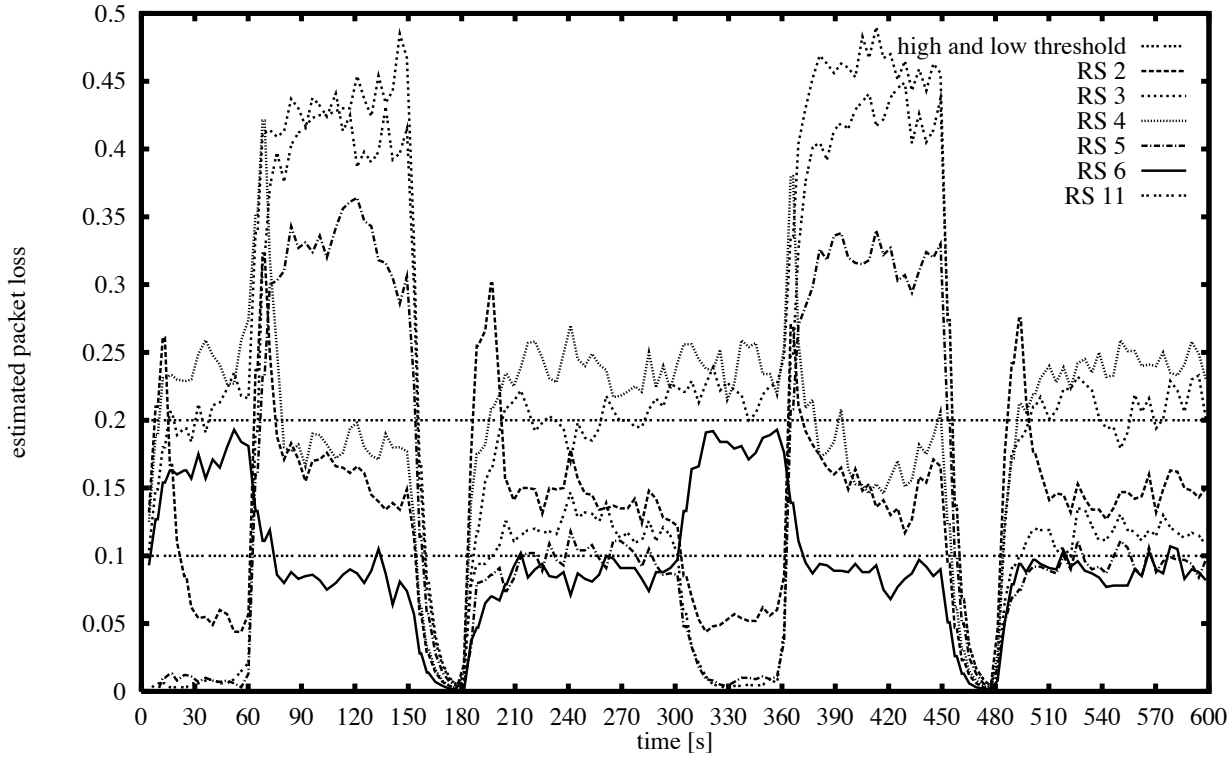


a) On-the-fly estimated downstream packet loss.

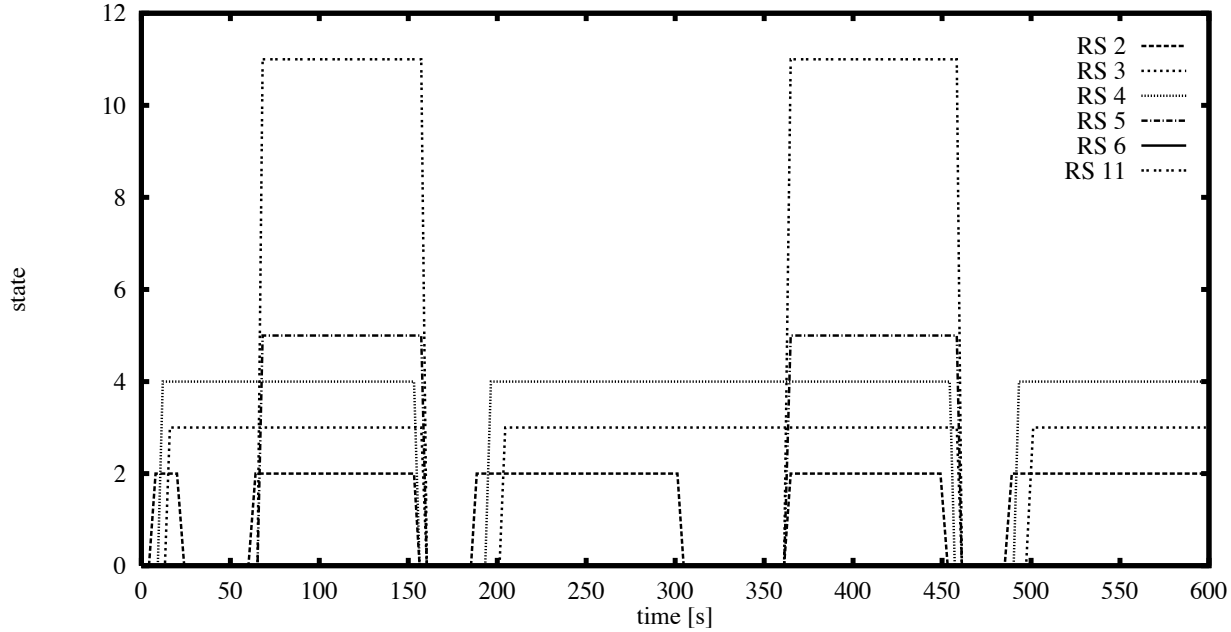


b) Dynamic RS activation and deactivation based on estimated downstream packet loss.

Figure 11: Parameters: $\alpha = 0.50,\ t_l = 0.07,\ t_h = 0.15$



a) On-the-fly estimated downstream packet loss.



b) Dynamic RS activation and deactivation based on estimated downstream packet loss.

Figure 12: Parameters: $\alpha = 0.50,\ t_l = 0.10,\ t_h = 0.20$

In a situation with known, constant link losses we have investigated which RSs should be active in order to provide cost effective reliable multicast. For the purpose of performance evaluation, an analytic cost function has been developed that accounts for bandwidth and buffer usage as well as processing costs at the RS. We have shown that the use of RSs reduces resource usage and latency in packet recovery. As was expected, it is beneficial to place Repair Servers immediately above lossy links. However, placing a RS over a loss-free subtree could also be useful. This is so because an active RS filters out packets sent to it's subtree, and thereby refrains from multicasting repair packets for which it has no pending NAKs. The optimal RS configuration which minimizes cost, to some extent depends on the relation unit buffer cost over unit transmission cost.

In a real life situation, link losses will be non-stationary, and hence the optimal placement of active Repair Servers will vary with time. An adaptive, distributed, and scalable policy for activation and deactivation of RS has been suggested. In a simulation study the policy rapidly activates and deactivates the appropriate RSs when the link loss pattern changes. The RS distribution attained by the policy is rather cost-efficient with respect to the cost measure presented earlier in the paper. One advantageous property with the policy is that it avoids situations where several RSs provide repair packets for the same NAKs. This is achieved because the RSs communicate indirectly through measurements of packet loss. Some few parameters, such as threshold values and length of the control interval, must be tuned for optimal performance.

References

[1] Hugh W. Holbrook, Sandeep K. Singhal, and David R. Cheriton. Log-based Receiver-Reliable Multicast for Distributed Interactive Simulation. In *Proceedings of SIGCOMM '95*. ACM, 1995.

[2] Don Towsley Dan Rubenstien, Sneha Kumar Kasera and Jim Kurose. Improving Reliable Multicast Using Active Parity Encoding Services. In *Proceedings of IEEE Infocom 1999*, volume 3, pages 1248–1255, New York, March 1999. IEEE.

[3] Sneha Kumar Kasera, Jim Kurose, and Don Towsley. A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast. Technical report, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA, July 1998.

[4] Sneha Kumar Kasera, Gisli Hjalmtysson, Don Towsley, and Jim Kurose. Scalable Reliable Multicast Using Multiple Multicast Channels. *IEEE/ACM Transactions on Networking*, 1999. Submitted.

[5] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya. Reliable Multicast Transport Protocol (RMTP). *IEEE Journal on Special Areas in Communications*, 15(3):407–421, April 1997.

[6] X. Rex Xu, Andrew C. Myers, Hui Zhang, and Raj Yavatkar. Resilient Multicast Support for Continuous-Media Applications. In *Proceedings of NOSSDAV'97*, 1997.

[7] Maya Yajnik, Jim Kurose, and Don Towsley. Packet Loss Correlation in the MBone Multicast Network. In *Proceedings of IEEE Global Internet Mini-conference, part of GLOBECOM '96*, pages 94–99, London, England, nov 1996. IEEE.

[8] Mark Handley. An Examination of MBone Performance. Technical report, USC/ISI, Department of Computer Science, University College London, January 1997.

[9] http://www.tascnets.com/panama/aer/index.html, 1999. Active Error Recovery (AER) web-page.

[10] Vern Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, Computer Science Division, University of California, Berkeley, University of California, Berkeley, CA 94720, USA, April 1997.

[11] Pravin Bhagwat, Partho P. Mishra, and Satish K. Tripathi. Effect of Topology on Performance of Reliable Multicast Communication. In *Proceedings of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies on Networking for Global Communication. Volume 2*, pages 602–609, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.