A Throughput Analysis of Reliable Multicast Protocols in an Active Networking Environment

M. Maimour and C. D. Pham
RESAM – UCB Lyon
ENS-Lyon, 46 allée d'Italie
69364 Lyon Cedex 07 - France
email:{mmaimour,cpham}@ens-lyon.fr

Abstract

Reliable multicast protocols have gained popularity with active services contribution where routers implement additional functionalities. Contributing mainly to feedback implosion problems, retransmission scoping and cache of data, these active protocols open new perspectives for achieving high throughput and low latencies on wide-area networks. In this paper, we define generic protocols with active services and present a throughput analysis of the various mechanisms proposed in existing protocols, i.e. ARM and AER. The main contribution consists in the analysis of the local recovery facility with different NACK suppression strategies and the potential of the subcast feature. Some insights on dimensioning active networks are also provided.

1 Introduction

Multicast is the process of sending every single packet to multiple destinations. Motivations behind multicast facilities are to handle one-to-many communications in a wide-area network with the lowest network and end-system overheads. In contrast to best-effort multicast, that typically tolerates some data losses and is more suited for real-time audio or video for instance, reliable multicast requires that all packets are safely delivered to the destinations.

Desirable features of reliable multicast include low end-to-end delays, high throughput and scalability. Meeting these objectives is not an easy task and reliable multicast is known to be a very difficult problem. Earlier reliable multicast protocols took the end-to-end solution to perform loss recovery. Most of them fall into one of the following classes: sender-initiated, receiver-initiated and receiver-initiated with local recovery protocols. In sender-initiated protocols, the sender is responsible for both the loss detection and the recovery [13]. These protocols do not scale well to a large number of receivers due to the ACK implosion problem. Receiver-initiated protocols move the loss detection responsibility to the receivers. They use NACKs instead of ACKs. However they still suffer from the NACK implosion problem when a large number of receivers have subscribed to the multicast session. In receiver-initiated protocols with local recovery, the retransmission of a lost packet can be performed by any receiver in the neighborhood [1], a designated receiver [10, 14] or a logging server [3] in a hierarchical structure.

All of the above schemes do not provide exact solutions to all the loss recovery problems. This is mainly due to the lack of topology information at the end hosts. Recently, the use of active network concepts [12] where routers themselves could contribute to enhance the network services by customized functionalities have been proposed in the multicast research community. Contributing mainly on feedback implosion problems, retransmission scoping and cache of data, active reliable multicast protocols open new perspectives for achieving high throughput and low latency on wide-area networks.

ARM (Active Reliable Multicast) [7] and AER (Active Error Recovery) [6] are two protocols that use active services. In both protocols, the main contribution of active services is a *best-effort* cache of data packets to permit local recoveries. However, they differ in the strategy adopted for solving the NACK implosion problem. ARM adopts a *global* suppression strategy: a receiver experiencing a packet loss sends immediately a NACK to the source. Active services in routers then consist in the aggregation of the multiple NACKs. In contrast, AER uses a *local* suppression strategy inspired from the one used by SRM and based on local timers at the receivers: prior to send a NACK packet, a receiver initiates a timer and waits for a random amount of time. If it receives a NACK for the same packet, it would cancel the timer, suppress its NACK and behave exactly as if it has sent the NACK itself. In this case, the active routers have in charge to indicate to the receivers that one of them has issued a NACK for a given packet. Implementation of this service can be done by multicasting to the other receivers the NACK packet. In addition, an active router in ARM would send the repair packet only to the set of receivers that have sent a NACK packet (subcast). In AER, the active router simply multicasts the repair packet to all its associated receivers.

Cache of data implemented as an active service permits the local recovery of loss packets. Kasera et al. [4] have analytically shown that protocols using local recovery exhibit better performances. They use an analytical framework originally proposed by Pingali et al. [11]. This framework was also adopted in the literature to perform other analysis of reliable multicast protocols [8, 5]. In [5], the authors have analyzed protocols that benefit from subcast facilities by using multiple multicast groups or channels. However local recovery and the global NACK suppression were not investigated.

None of these works have analytically investigated together the three active services: local recovery, NACK suppression strategies and subcast. One of our contributions in this field consists in the analysis of the local recovery with different NACK suppression strategies and the potential of the subcast feature. In an attempt to help for active networks dimensioning, we also address the constraints put on active routers processing power for achieving better performances than traditional non active approaches.

The rest of the paper is organized as follows. Our generic reliable multicast protocols with active support are presented in section 2. Section 3 presents the network model and the general assumptions associated to the processing cost analysis presented in section 4. Section 5 presents the numerical results and section 6 concludes.

2 Generic reliable multicast protocols

In this section, we describe 5 generic reliable multicast protocols from which the generic models are derived. These protocols are called S_1, S_2, S_2^s, S_3 and S_3^s. All these protocols benefit from the cache of packets at the active routers. However, they differ from each other in the strategy for the NACK suppression and whether subcast is used or not.

A NACK packet is assumed to follow the reverse path of the data packet. This assumption is essential for taking advantage of active services. This behavior can be implemented by using a dedicated routing service for NACK packets or to make a receiver know the identity of the active router to which it is attached [6].

Also, in order to limit the processing overheads of duplicate NACKs, the source and the active routers discard all similar NACKs for a given amount of time. For the analysis, we will assume that the "duplicate discard" period is well chosen. Therefore, no duplicate NACKs will trigger a new retransmission.

How to implement the subcast feature is not addressed in this paper. We will assume simply that active routers in the network store the soft state of the NACKs which are necessary to perform the subcast [5, 7]. It has been shown in [5] that active routers need only a few hundreds of bytes per multicast session to support the subcast functionality.

Finally, we will call *local group* the set of receivers linked to the same active router. If there are several active routers on the path, the nearest one, from the receiver perspective, is taken.

2.1 Description of protocol S_1

S_1 uses the global suppression of NACK packets. A receiver experiencing a packet loss sends immediately a NACK to the source. The active routers have in charge the aggregation of NACK packets in order to forward only one NACK to the source. S_1 has the following properties:

- the source multicasts data packets at the multicast address subscribed to by all the receivers.
- upon reception of a data packet, an active router stores the packet in its cache, if possible, and forward it downstream in the multicast tree.
- upon detection of a packet loss, a receiver sends immediately a NACK towards the source and sets a timer.
- upon reception of a NACK packet, an active router sends the corresponding repair packet, if available, to all the receivers composing its local group. Otherwise, it sends the NACK to the source.
- upon reception of a NACK packet, the source multicasts the repair packet to all receivers at the multicast address.

2.2 Description of protocols S_2 and S_2^s

S_2 uses the NACK local suppression strategy. The receivers wait for a random amount of time prior to send a NACK to the source. The goal is to generate only one NACK per group of receivers in the multicast tree. S_2 is similar to L1 described in [4] (which is itself a generic version of AER [6]). Both use NACK local suppression but S_2 performs this suppression only within the end receivers linked to a same active router. In contrast, L1 would also do local suppression within intermediate nodes located at the same level in the multicast tree. Our generic protocol does not use local suppression between nodes located at the same level in the multicast tree. The reason is not to overload the active routers. Like P2 and P3 [5], S_2 performs NACK suppression based on timers [1]. Whereas P2 and P3 do not perform any local recovery and adopt a strategy based on multiple group channels, S_2 benefits from the contribution of active routers [6] as will be described below:

- the source multicasts data packets at the multicast address subscribed to by all the receivers.
- upon reception of a data packet, an active router stores the packet in its cache, if possible, and forward it downstream in the multicast tree.
- upon detection of a packet loss, a receiver waits for a random amount of time. Only then it would send a NACK packet to the source and sets a timer.
- when a receiver is waiting to send a NACK, the reception of a similar NACK, from its active router, would make the receiver to cancel its NACK, set a timer and behave as if it has sent the NACK itself. If during the waiting time it receives the repair packet, then the receiver quits the recovery process.
- upon reception of a NACK from downstream, an active router multicasts the repair packet, if available, to its local group. Otherwise, it multicasts the NACK to both the source and its local group, excepting on the NACK incoming link. The desire behavior is to perform the NACK local suppression.
- upon reception of a NACK packet, an active router sends the corresponding repair packet, if available, to all the receivers composing its local group.
- upon reception of a NACK packet, the source multicasts the repair packet to all receivers at the multicast address.

S_2^s is identical to S_2 in performing a local NACK suppression among receivers of the same local group. However, it benefits from the subcast facility from the source. It is difficult for active routers that already perform the local NACK suppression to also implement a subcast service: they are unable to know the identity of the receivers that have experienced a loss since they do not receive all the generated NACKs in their local group. Therefore in S_2^s, receivers linked to an active router that performs local NACK suppressions do not benefit from the subcast feature whereas free receivers will benefit from the subcast directly from the source.

2.3 Description of protocols S_3 and S_3^s

Active routers that perform a global NACK suppression strategy as in S_1 can easily implement the subcast service since they know the identity of the receivers that have experienced a loss because they receive all the corresponding NACKs. Therefore we define here the protocol S_3 which is similar to S_1 in performing a global NACK suppression strategy but also implements the subcast service within active routers in addition to the NACK suppression service. We also define the protocol S_3^s that behaves as S_3 but also benefits from the subcast facility from the source.

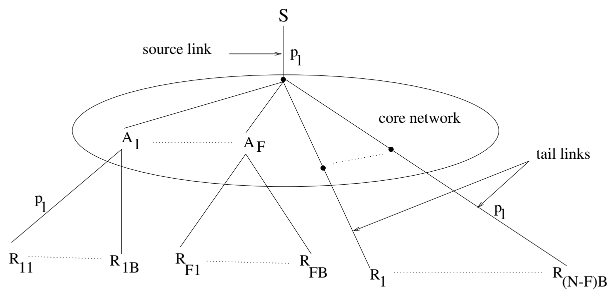


Figure 1. Network Model.

3 Network model and hypothesis

The network model used in the study is similar to the one used in [4]: one source multicasts data packets to R receivers through a packet network composed of a fast core network and several slower edge access networks. We will call *source link* the set of point-to-point links and traditional routers that connects the source to the core network. Similarly, a *tail link* is composed of point-to-point links and routers connecting a receiver to the core network (see Fig. 1). We only consider active routers at the edge of the core network. This is due to the fact that the core network is reliable and runs at a very high-speed. Adding complex processing functions inside the core network will slow down the packet forwarding functions.

Our network model differs from the one used in [4] as F routers among the N possible can be active, $0 \leq F \leq N$ (Fig. 1). Particular cases where $F = 0$ and $F = N$ will be considered in the study. Each active router A_i is responsible of B receivers $R_{i1}, \cdots, R_{iB}$ forming a local group. These receivers associated with the F active routers are said *linked*. The other receivers are said *free*.

For the loss model, we will consider that the core network is reliable, as mentioned previously. For the other links (the source link or the tail links), the loss probability is noted p_l. Therefore, the end-to-end probability of a packet loss perceived by a receiver is $p = 1 - (1 - p_l)^2$. The losses are assumed to be temporally independent and those at the tail links are assumed to be mutually independent. We will also assume that NACK packets are never lost.

For this analysis, we will consider that active routers are

Table 1. Notation

X^{S_j}	the total processing time per packet at the source for S_j protocols, $j = 1..3$.
Y^{S_j}	the total processing time per packet at a linked receiver for S_j protocols, $j = 1..3$.
$Y_*^{S_j}$, $Y_*^{S^s_k}$	respectively, the total processing time per packet at a free receiver for S_j and S^s_k protocols, $j = 1..3$ and $k = 2, 3$.
$A_i^{S_j}$, $A_i^{S^s_k}$	respectively, the total processing time per packet at an active router A_i ($i = 1..F$) for S_j and protocols S^s_k protocols, $j = 1..3$ and $k = 2, 3$.
Λ^ω_x	mean throughput achieved by node x with the ω protocol.
Λ^ω	mean throughput achieved by the ω protocol.
X_p, X_n	respectively, the processing time for sending a data packet and to receive a NACK packet.
Y_p, Y_n	respectively, the processing time for receiving a data packet and to send a NACK packet.
Y_t	time to process a timeout
X^a_p, X^a_n	respectively, the processing time for sending a data packet and to receive a NACK packet for an active router.
Y^a_p, Y^a_n	respectively, the processing time for receiving a data packet and to send a NACK packet for an active router.
M_S	number of retransmission of a packet at the source until all active routers and all free receivers have correctly received the packet.
M_R, M_A	same as M_S when $F = 0$, respectively $F = N$.
M_b, M_B	number of retransmission of a packet at an active router until one, respectively all, associated linked receivers have correctly received the packet.
M_a	number of retransmission of a packet at the source until an active router has correctly received the packet.
M_r	number of retransmission of a packet at the source until a receiver (linked or free) has correctly received the packet.

always able to perform local recoveries. Justifications of this assumption can be found in [9].

For S^s_2 and S^s_3 protocols, we assume that the subcast from the source is performed thanks to active routers located between the source and the F active routers and the free receivers. Since the role of these active routers is limited to the subcast service, they are never the bottleneck.

4 Throughput analysis

Similar to the approach taken by Pingali et al. in [11], our analysis uses the processing requirements of the protocols rather than the bandwidth requirements. The maximum throughput achievable by a protocol depends on the processing time per packet at the various nodes in the multicast tree. For example, the total processing time per packet at the source includes the initial transmission time, all the retransmissions, and NACK overheads necessary for achieving a correct reception of the packet by all the receivers.

The notation used in the study is listed in table 1. We will also use the notation $(x)^+$ to refer to $max(0, x)$. For space limitation, we will not be able to include the probability distribution and the mean of each random variable used in this paper. They can be found in [9].

4.1 Protocol S_1

We begin by analyzing the processing requirements at the source: a data packet is sent M_S time until all active routers and free receivers have correctly received it. The source receives for this packet F $(M_a - 1)$ NACKs from active routers and $(N - F)B$ $(M_r - 1)$ NACKs from free receivers. The processing time per packet at the source can therefore be written:

$$E[X^{S_1}] = E[M_S]\,E[X_p] + (F(E[M_a] - 1) + (N - F)B\,(E[M_r] - 1))\,E[X_n] \quad (1)$$

Replacing $E[M_a]$ and $E[M_r]$ by their expressions [9] gives:

$$E[X^{S_1}] = E[M_S]\,E[X_p] + \left(F\,\frac{p_l}{1 - p_l} + (N - F)B\,\frac{p}{1 - p}\right) E[X_n] \quad (2)$$

Note that the S_1 protocol with no active routers ($F = 0$) is identical to the N_1 protocol described in [11]. By setting $F = 0$ in eq. (2) we have a similar result as eq. (11) stated in [11]:

$$E[X^{S_1}] = E[M_R]\,E[X_p] + R\,\frac{p}{1 - p}\,E[X_n] \quad (3)$$

When all routers have active services, we find:

$$E[X^{S_1}] = E[M_A]\,E[X_p] + N\,\frac{p_l}{1 - p_l}\,E[X_n] \quad (4)$$

The processing time at a receiver depends on whether it is linked or free. A linked receiver receives a data packet M_B times with probability $(1 - p_l)$. It sends back $(M_r - 1)$ NACKs towards the source, and so processes $(M_r - 2)^+$ timeouts. This gives the following expression:

$$E[Y^{S_1}] = (1 - p_l)\,E[M_B]\,E[Y_p] + \frac{p}{1 - p}\,E[Y_n] + \frac{p^2}{1 - p}\,E[Y_t] \quad (5)$$

In contrast, a free receiver receives the data packet M_S times with probability $(1 - p)$. It sends $(M_r - 1)$ NACKs and processes $(M_r - 2)^+$ timeouts, giving the following equation:

$$E[Y_*^{S_1}] = (1 - p)\,E[M_S]\,E[Y_p] + \frac{p}{1 - p}\,E[Y_n] + \frac{p^2}{1 - p}\,E[Y_t] \quad (6)$$

If $F = 0$, meaning that all receivers are free, we have an expression similar to that of protocol N_1 found in [11]:

$$E[Y_*^{S_1}] = (1 - p)\,E[M_R]\,E[Y_p] + \frac{p}{1 - p}\,E[Y_n] + \frac{p^2}{1 - p}\,E[Y_t] \quad (7)$$

For an active router A_i, it receives packets like a free receiver: a data packet is received M_S times with probability $(1 - p_l)$. It forwards the packet M_B times to its B linked receivers until they correctly receive the packet. It receives B $(E[M_r] - 1)$ NACKs from its B linked receivers and sends back $(M_b - 1)$ NACKs to the source. These NACKs have been issued by the linked receivers due to a packet loss at the source link so we have:

$$E[A_i^{S_1}] = (1-p_l)\,E[M_S]\,E[Y_p^a] + E[M_B]\,E[X_p^a] + B\,\frac{p}{1-p}\,E[X_n^a] + \frac{p_l}{1-p_l}\,E[Y_n^a] \quad (8)$$

If all routers are active, we find:

$$E[A_i^{S_1}] = (1-p_l)\,E[M_A]\,E[Y_p^a] + E[M_B]\,E[X_p^a] + B\,\frac{p}{1-p}\,E[X_n^a] + \frac{p_l}{1-p_l}\,E[Y_n^a] \quad (9)$$

4.2 Protocols S_2 and S_2^s

For the analysis of the S_2 protocol, we will assume that the random waiting time of receivers are well chosen in order to generate only one NACK per local group.

In this protocol, both the source and the free receivers behave like in the S_1 protocol. Consequently, we have the same equation for them: eq. (1) and eq. (6) respectively. For a linked receiver, we have:

$$E[Y^{S_2}] = (1-p_l)\,E[M_B]\,E[Y_p] + \frac{E[M_B]-1}{B}\,(E[Y_n] + E[X_n]) + \frac{p^2}{1-p}\,E[Y_t] \quad (10)$$

The first term indicates how many times a linked receiver receives a data packet. The second and the third terms correspond respectively to NACKs that were sent and received by a linked receiver in the process of NACK local suppression. The last term represents the processing overhead for the $(M_r-2)^+$ timeouts.

An active router receives a data packet from the source M_S times with probability $(1-p_l)$. It sends the packets to its linked receivers M_B times. Thanks to the NACKs local suppression task performed by the receivers, the active router receives only one NACK per packet loss. In total, it receives (M_B-1) NACKs. The active router would forward the NACKs to the source and to the other receivers if it does not have the packet: that makes $p_l\ (M_B-1)$ NACKs where p_l is the loss probability at the source link. Finally we find:

$$E[A_i^{S_2}] = (1-p_l)\,E[M_S]\,E[Y_p^a] + E[M_B]\,E[X_p^a] + (E[M_B]-1)\,E[X_n^a] + p_l\,(E[M_B]-1)\,E[Y_n^a] \quad (11)$$

If all routers are active, we have:

$$E[A_i^{S_2}] = (1-p_l)\,E[M_A]\,E[Y_p^a] + E[M_B]\,E[X_p^a] + (E[M_B]-1)\,E[X_n^a] + p_l\,(E[M_B]-1)\,E[Y_n^a] \quad (12)$$

For S_2^s that benefits from the subcast from the source, only the processing cost at the active routers and the free receivers will change. Since they receive only once each data packet, we can deduce from (7) and (11) the following statements:

$$E[Y_*^{S_2^s}] = E[Y_p] + \frac{p}{1-p}\,E[Y_n] + \frac{p^2}{1-p}\,E[Y_t] \quad (13)$$

$$E[A_i^{S_2^s}] = E[Y_p^a] + E[M_B]\,E[X_p^a] + (E[M_B]-1)\,E[X_n^a] + p_l\,(E[M_B]-1)\,E[Y_n^a] \quad (14)$$

4.3 Protocols S_3 and S_3^s

In the S_3 protocol, the linked receivers benefit from the subcast feature so they receive a data packet only once. Therefore we have:

$$E[Y^{S_3}] = E[Y_p] + \frac{p}{1-p}\,E[Y_n] + \frac{p^2}{1-p}\,E[Y_t] \quad (15)$$

For the other nodes, we have the same results as for the S_1 protocol. In S_3^s, the free receivers behave as in S_2^s. For the active routers, we can deduce from (8):

$$E[A_i^{S_3^s}] = E[Y_p^a] + E[M_B]\,E[X_p^a] + B\,\frac{p}{1-p}\,E[X_n^a] + \frac{p_l}{1-p_l}\,E[Y_n^a] \quad (16)$$

For the other nodes, we have the same results as for the S_3 protocol.

5 Numerical results

In this section, we analyze the generic protocols S_1, S_2, S_2^s, S_3 and S_3^s. For each protocol, we first examine the overall throughput achieved by the protocol and then we focus on the throughput achieved separately by each node. In addition, we will first consider that the active routers have the same processing time as the receivers, then we will study the impact of varying the processing time at the active routers. The benefit of the subcast and the influence of the active routers density ($d = F/N$) in the model will be considered too.

The throughput Λ_x^ω achieved by node x under the protocol $\omega \in \{S_1, S_2, S_3, S_2^s, S_3^s\}$ is given by $\Lambda_x^\omega = 1/E[x^\omega]$ where $x \in \{X, Y, Y_*, A_1, \cdots, A_F\}$ and $E[x^\omega]$ is the mean processing time at node x under the protocol ω. Therefore, the overall throughput Λ^ω achieved by the protocol ω can be computed using $\Lambda^\omega = Min(\Lambda_x^\omega)$.

For the numerical evaluation, we take $E[X_p] = E[Y_p] = 500\ \mu s$, $E[X_n] = E[Y_n] = 85\ \mu s$ and $E[Y_t] = 32\ \mu s$. These values are those experimentally measured in [5]. For the mean processing time at the active routers, we set $E[X_p^a] = E[Y_p^a] = 500\ \mu s$ and $E[X_n^a] = E[Y_n^a] = 85\ \mu s$, identical to the processing time of the end nodes. In [7], the authors have measured a data packet processing time within an active router of about $4\ ms$. They argue that this processing time can be reduced at least by a factor of 10. The study will also present results when varying the processing power at the active routers. For the throughput computation the packet size is set to 1024 bytes.

We begin by examining the overall throughput achieved by the protocols S_1, S_2 and S_3. Fig. 2 shows the overall throughput achieved by S_1. We can see the benefit of active routers ($d \neq 0$) which permit a higher overall throughput, even with a high loss rate, and especially in the case of a large number of receivers. Although not shown, local or

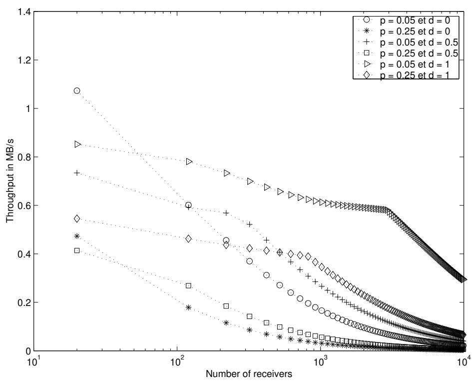


Figure 2. Throughput achieved by S_1

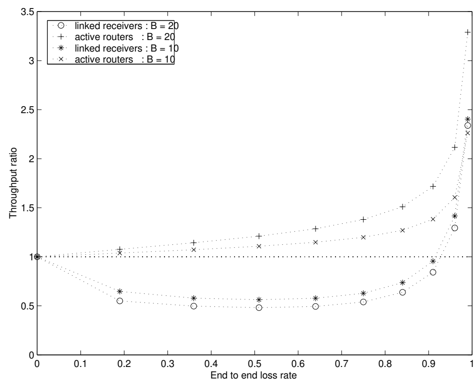


Figure 4. Linked receivers and active routers' throughput ratio: (S_2/S_3).

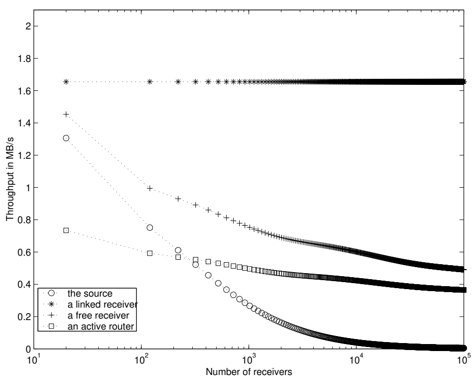


Figure 3. Throughput of the various nodes with S_1. p = 0.05, B = 10, 50% of active routers.

global NACK suppression strategies have very close performances in terms of the overall throughput for very low loss rates. However S_2 performs better than S_1 when the loss probability increases (see [9] for more details).

In order to know which nodes yield the minimum throughput, Fig. 3 plots the throughput achieved by each node in the protocol S_1 as a function of the number of receivers with 50 % of active routers and a loss rate of 5 %. We can observe that the minimal throughput in the system is first introduced by active routers. Then the source turns out to become the bottleneck. Although not shown, in S_3, the linked receivers achieve a higher throughput compared to S_1 which does not benefit from the subcast facility. S_2 presents a slightly higher throughput than S_1 mainly when p increases[9].

5.1 Local suppression vs global suppression

Local suppression appears to provide slightly higher performances in terms of the overall throughput. It has been also observed that S_2 performs better than S_1 when the loss rate increases. In order to deeply compare the two suppression strategies, Fig. 4 plots the ratio for S_2 and S_3 of linked receivers and active routers throughput as a function of the loss probability. Different local group sizes are used. S_2 performs better than S_3 at the active routers, mainly for high loss rates. In the local suppression strategy, NACKs are suppressed before they reach an active router. Thus justifying the benefit of this strategy for high loss probabilities. At the linked receivers end, we can see that for reasonable loss probabilities, S_3 performs better than S_2. This is because the linked receivers under S_3 benefits from the subcast service. In S_3, a linked receiver receives only once a data packet. However in S_2, a linked receiver could receive more than one copy of the same data packet. Moreover, it continues to receive NACKs from its active router every time a receiver in its local group has experienced a loss.

5.2 Benefit of the subcast

The subcast facility has the advantage of unloading the receivers and/or the active routers depending on whether we benefit from this facility from the source or not. To see the benefit of performing the subcast from the active routers associated to the linked receivers, Fig. 5 plots the throughput ratio at a linked receiver in S_3 and S_2. We can see that the subcast permits a higher throughput at the linked receivers in S_3. The gain obtained with the subcast depends on the local group size and the loss rate. These two parameters gives an idea on the number of the receivers that have experienced a loss.

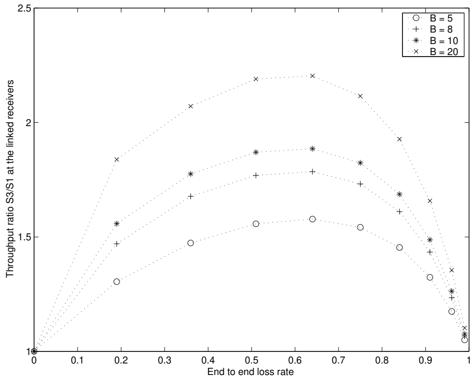


Figure 5. Benefit of the subcast for the linked receivers (S_3/S_2 ratio).

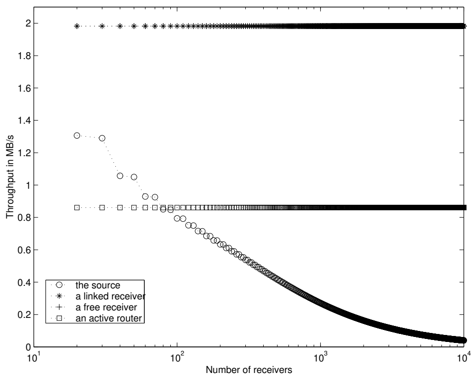
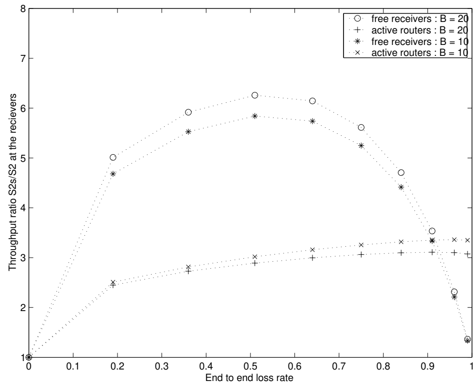
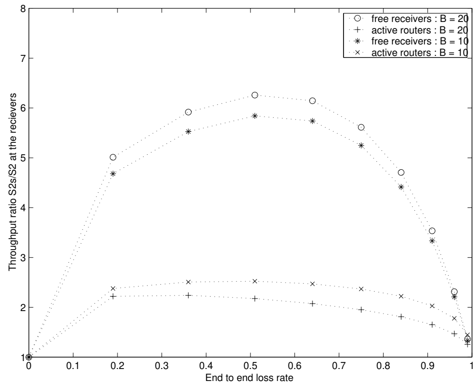


Figure 6. Throughput of the various nodes in S_3^s ,p = 0.05, B = 10, 50% of active routers.



(a)



(b)

Figure 7. Benefit of the subcast from the source (a) S_2^s/S_2 and (b) S_3^s/S_3.

In addition to the linked receivers, performing the subcast from the source allow to unload the active routers and the free receivers. Fig. 6 shows that the subcast gives a constant and higher throughput at all the nodes that benefit from this facility. Note that in S_3^s, a "linked receiver" and a "free receiver" achieve the same throughput. Both of them benefit from the subcast feature.

To show the benefit of the subcast from the source, Fig. 7a and 7b plot the throughput gain at the free receivers and the active routers as a function of the loss probability for S_2^s and S_2, and for S_3^s and S_3 respectively. Two different local group sizes are used. It is worth to mention that the number of free receivers ($(N - F)B$) is proportional to the local group size (B). At the free receivers side, we can achieve a gain of 5 or even 6 for a loss rate of 20 % and 50 % respectively. We notice also that similar to the case of the linked receivers in Fig. 5, the subcast is more beneficial in the presence of a large number of free receivers. Moreover, for very low and high loss rates, the subcast does not permit a noticeable gain. For the active routers, we can see that unlike S_3^s and S_3, S_2^s performs better than S_2 even for high loss rates. This is due to the fact that active routers in S_3^s receive all the NACKs generated in the local group which considerably increases with p. However, S_2 suppresses NACKs before they reach the active routers.

5.3 Active routers density

Fig. 8 shows the impact of the active routers density on the protocol's performances in terms of the overall throughput. The figure plots the overall throughput gain achieved by S_3 as the number of active routers is increased compared to the no active routers case. Several multiplicating factors to the active routers' processing power are applied. We can see that with the same processing time at the active routers and the receivers, the overall throughput can be an order of magnitude higher if all the receivers are linked. Most interestingly, if the active router's processing power is divided

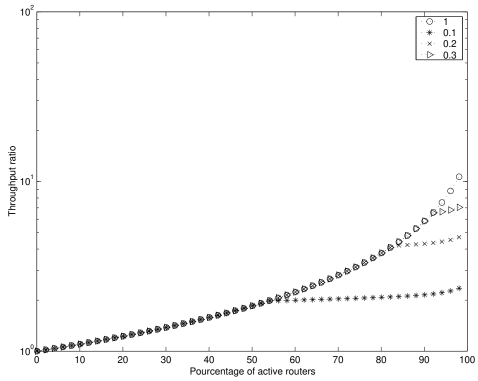


Figure 8. Gain in terms of the overall throughput achieved by S_3 when varying the active routers density. $p = 0.05$, $B = 10$ and $N = 1000$.

by 10, we can still double the overall throughput provided that 55 % of routers are active. Although not shown, in S_3^s, 55 % of active routers are sufficient to double the performances even when dividing the processing power by 20. S_1 and S_2 behave as S_3. S_2^s presents a slightly higher gain than S_3^s for low loss probabilities (eg. 0.05) [9].

6 Conclusions

Reliable multicast is a difficult problem. Contributions of active routers within the multicast tree can be used for performing additional functionalities such as cache of data, feedback aggregation or subcast. In order to evaluate the potential of these mechanisms, we proposed five generic analytical models derived from a corresponding protocol's description: S_1 that uses the global NACK suppression strategy, S_2 that uses local suppression instead, and S_3 that differs from S_1 by the subcast facility. S_2^s and S_3^s behave as S_2 and S_3 respectively but benefit from a subcast facility from the source. All five benefit from the cache of data.

Our analysis uses the processing requirements of the protocols to derive the achievable throughput at the various nodes in the multicast tree. It appeared that local suppression, under the assumption that the random waiting time at a receiver is well-chosen, requires less processing power than global suppression mainly for high loss rates. Therefore a dynamic scheme may show interesting results. We have also shown the benefit of the subcast feature which allows a higher and constant throughput. Thus unloading nodes that benefit from this facility, especially when it is applied from the source. Regarding the impact of the active router density on performances, all protocols have the same behavior: the achievable throughput increases as the number of active routers increases. Most interestingly, even with slower processing power at the active routers, increasing their number allows for more performances.

We are currently performing simulations in order to investigate deeper the processing and the buffering requirements of active reliable multicast. We are also in the process of implementing the proposed mechanism in a high-performance active environment called Tamanoir [2].

References

[1] S. Floyd, V. Jacobson, C.G. Liu. *A Reliable Multicast Framework for Light Weight Session and Application Level Framing*. Proc ACM SIGCOMM'95, pp342-356, Aug 1995.

[2] JP. Gelas and L. Lefevre. *TAMANOIR : A High Performance Active Network Framework*. Active Middleware Services, Kluwer Acad. Pub. ISBN 0-7923-7973-X, Aug 2000.

[3] H.W. Holbrook, S.K. Singhal, D.R. Cheriton. *Log-Based receiver-Reliable Multicast for Distributed Interactive Simulation*. Proc. of ACM SigComm'95, Vol.25, No.4, pp328-341.

[4] S.K. Kasera, J. Kurose and D. Towsley. *A Comparaison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast*. In Proc. of IEEE INFOCOM, San Francisco, Mar 1998.

[5] S.K. Kasera, G. Hjalmtysson, D. Towsley and J. Kurose. *Scalable Reliable Multicast Using Multiple Multicast Channels*. IEEE/ACM ToN, Jun 2000.

[6] S.K. Kasera, S. Bhatacharyya, M. Keaton, D. Kiwior, J. Kurose, D. Towsley and S. Zabele. *Scalable Fair Reliable Multicast Using Active Services*, IEEE Network Magazine's Special Issue on Multicast 2000.

[7] Li-Wei H. Lehman, S.J. Garland, and D.L. Tennenhouse. *Active Reliable Multicast*. Proc. of IEEE INFOCOM'98, San Francisco, CA, Mar 1998.

[8] B.N. Levine and J.J. Garcia-Luna-Acceves. *A Comparaison of Known classes of Reliable Multicast Protocols*. Proceedings of the international Conference and network Protocols (ICNP)'96, Oct 1996.

[9] M. Maimour and C. Pham. *A Throughput Analysis of Reliable Multicast Protocols in an Active Networking Environment*. Technical report, http://www.ens-lyon.fr/~mmaimour/Paper/TR/TR01-2001.ps.gz.

[10] S. Paul, K.K. Sabnani. *Reliable Multicast Transport Protocol (RMTP)*. IEEE Journal of Selected Areas in Communications, Special Issue on Network Support for Multipoint Communication, Vol.15, No.3, pp407-421, Apr 1997.

[11] S. Pingali, D. Towsley, and J.F. Kurose. *A Comparaison of Sender-initiated and Receiver-initiated Reliable Multicast protocols*. in Proc. of ACM SIGMETRICS'94, Vol.14, pp221-230, 1994.

[12] D.L. Tennehouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Winden. *A Survey of Active Network Research*. in IEEE Com Mag, pp80-86, Jan 1997.

[13] XTP Revision 4. 0. *Xpress Transport Protocol Specification*. XTP Forum, Santa Barbara, Mar 1995.

[14] R. Yavatkar, J. Griffioen, and M. Sudan. *A Reliable Dissemination Protocol for Interactive Collaborative Applications*. Proc' of the ACM Multimedia'95, Nov 1995.