A Strategy for Comparing Reliable Multicast Protocols Applied to RMNP and CTES

Arturo Azcorra Saloña
ETSIT, Univ. Politécnica de Madrid
Ciudad Universitaria s/n, 28040
Madrid, Spain
azcorra@dit.upm.es

María Calderón Pastor, Marifeli Sedano Ruiz
Facultad de Informática, Univ. Politécnica de Madrid
Campus de Montegancedo, 28660
Boadillla del Monte, Madrid, Spain
{mcalderon, msedano}@fi.upm.es

Abstract

An increasing number of networked Multimedia Applications do require a reliable multicast service to support the distribution of data to a potentially large number of receivers, where the receivers may be located in a sparse mode over a WAN network. Designing a reliable multicast protocol is certainly a challenging task, and the literature is abundant in proposals with different approaches to the problem. Because of the large number of proposals, and considering that none of them appears to be clearly superior to the others, it is required a quantitative performance comparison to find out their strengths and drawbacks. As the complexity of the algorithms and protocols is quite high, analytical approaches may be ruled out. A simulation approach may provide acceptably reliable results as long as it is built following the more or less well established guidelines. However, building up the model of the system to be simulated in a way that it assures fairness in the comparison of "competing" reliable multicast protocols requires a detailed study. In this paper, a strategy to compare different approaches to reliable multicast is presented. This strategy has been applied to a concrete case, by comparing two state-of-the-art protocols: RMNP and CTES.

1: Introduction

The growth that has been observed on Mbone and on other networks and technologies with multicast capability has led to the development of a range of multimedia multicast-based applications, such as distributed interactive virtual reality, whiteboard teleconferencing and distribution of financial and billing multimedia data. Many of these application require also that the multicast transfer service be also reliable, guaranteeing delivery of data sent from the source(s) to all the members of the group.

Multicast data communications based on Deering's IP multicast extension [5] has been widely available in the Internet. But the bearer service provided by IP does not fit the requirements of all applications. It offers a best-effort service leaving it up to the application to provide the required reliability. Designing a reliable multicast protocol for an application that can have several thousand receivers and these receivers are distributed over a wide area geographical (spanning one or more continents) is a quite different problem. In addition, the highly dynamic nature of the topology and population complicates the problem.

Different reliable multicast techniques have been found to be unacceptable because of their high inefficiency in the presence of either large groups or a high packet loss probability. Measurements [16] performed over Mbone have shown that between 38% and 72% of multicast packets sent, were lost by at least one receiver, and consequently it was required to retransmit them, at least once. This study shows that reliable multicast techniques have to be carefully analyzed under the target application conditions to actually ascertain their strengths and weaknesses in terms of performance and resource (network and end systems) usage.

The literature contains many proposals of reliable multicast protocols [2, 8, 10, 11, 14, 17], many of which contain interesting and promising ideas. To be capable of performing a solid comparative analysis of the different proposed solutions, it does not suffice to realize a qualitative comparison of advantages and disadvantages of each approach. What is required is to perform quantitative comparisons of the behavior of the different protocols in relation to the different aspects that are relevant in group communications: their scalability, their usage of network resources, their adaptability to dynamic membership changes, the processing overload on the end systems, the mean data distribution delay, etc.

A first approach to performing a comparison is to field test them over a real network. Unfortunately, this is not possible for the great majority of the proposals, as their are not implemented (an in many cases not even fully specified). Even in the cases when the implementation exists, it is highly difficult to build up a test case because it requires having access to a large number of end systems and the network, not only as a user, but also being able to measure system parameters such as CPU load, memory consumption, end to end delay and trunk load. Another approach is to base the comparison on system simulation. This requires also to have an implementation, if not of the complete protocols, at least of a significant set of features to be compared.

In either case, real or simulated, it is needed to take into account the great number of aspects related with the protocols themselves, such as the group members distribution, the supporting network, the selected topologies, and the test cases. This article aims at providing a set of guidelines to support a fair and solid quantitative performance comparison of different reliable multicast approaches. Section 2 presents these guidelines as a series of steps to be followed for the comparison. Section 3 presents an application of the guidelines to a concrete case of two state-of-the-art protocols that use different approaches to perform local recovery and recovery with restricted scope. Conclusions are presented on section 4.

2: Guidelines for comparing reliable multicast schemes

The guidelines presented here are based on simulation work performed along the last years to compare different proposed reliable multicast approaches. This might be polarized by the particular cases that have been studied, but it is still possible to draw some general conclusions. The following subsections present the steps to be followed.

2.1: Define clearly the characteristics that are to be compared

This requires a detailed analysis because its result will be the basis to construct the measurement model during the simulations, and it imposes relevant requirements over the simulator design. Some examples of characteristics frequently compared in reliable multicast protocols are:

- *Behavior of the protocol under a changes in the network topology*: This is a relevant factor in protocols that make use a control tree (for aggregation, local recovery, etc.).
- *Adaptability to dynamic membership changes*. Members joining and abandoning the group may require a modification of the control tree and possibly other state information.
- *Scalability in terms of group size and/or network diameter*.
- *Measured benefits of Local Recovery*. Local recovery discharges the source from performing all the retransmissions, but there are different approaches based on this idea.
- *Recovery with restricted scope (or recovery exposure or recovery isolation).*
- *Flow Control*. Controlling the packet rate in multicasting is complicated by the fact that the protocol must accommodate multiple receivers simultaneously. How this is performed can have significant impact on the overall performance.
- *Tolerance to different member characteristics:* Generic protocols must deal with the possibility that not all of the receivers have access to the same hardware and network resources.

The first step is then to clearly state what are the precise objectives of the comparison. Notice that for each mechanism or algorithm to be compared it is necessary to write the simulation code that models it. In this sense, it is much simpler to compare a subset than attempting to simulate the complete protocols. Of course, it is required to guarantee that when isolating a particular mechanism, the removal of other mechanisms does not affect the one under study. Notice that many protocol mechanisms cross-affect each other, and therefore it might be required on some cases to implement a broader subset than the one specifically under comparison. Those aspects that are not considered relevant for comparison, but are still needed for the simulator to run will be called the “common framework” of both protocols, and the intention is that a similar “common framework” be shared by them.

The result of this step will be a definition of the common framework and the subset of differential functionalities to be simulated for each protocol.

2.2: Network and application model

The objective of this step is to define the underlying network and group model, and the application characteristics and pattern to be used in the simulation. This is a critical step, because a wrong definition may render unfair comparison data, favoring the protocol that adapts better to the defined case. The definition should be based on a deep understanding of both protocols and on the specific characteristics to be compared defined in step 1. The model may be refined in five subaspects:

- Network technology, i.e., aspects such as connection oriented or connectionless service or dynamic topological changes permitted or not.

- Select the network topology(ies) to be used. It should be taken into account whether the protocol imposes some particular restriction on the underlying topology.
- Define the network parameters. Link bandwidths, propagation delay, loss probability, link recovery time, error rates, router processing capacity, router buffering restrictions, etc.
- The application communication model. For example, one-to-many (tele-lecture) or many-to-many (distributed interactive simulation.)
- The application traffic pattern of the source (s).

2.3: Protocol parameters

The objective of this step is to select the values of the different protocol parameters under simulation. This should be accomplished by following the protocol designer's guidelines on protocol configuration. As most papers do not provide any information on how to appropriately configure the parameters, in case of doubts it is recommended to test with a range of parameters. In respect to the "common framework", it must be guaranteed that the selected values match the profile of both protocols, or otherwise it would be necessary to reduce the common framework and increase the specific simulation code of each protocol.

2.4: Evaluation metrics

The metrics are directly related to the characteristics defined in step 1. It is the objective of this step to associate the desired characteristics to a set of measurable parameters. This will provide quantitative results from simulation executions, allowing to perform the judgment on the respective merits of the protocols in regard to each specific characteristic. Metric parameters that are frequently related to desired characteristics are the following:

- *Transfer delay*: average packet time from the source to **all** the members.
- *Network load*: measured as trunk line capacity consumed and router processing and buffer space required. This is a key factor to determine the scalability of a protocol.
- *End-system load:* this is aimed to metering aspects such as the processing complexity of the protocol, its storage requirements, or its avoidable operations (e.g. reception of duplicates).
- *Specific-system load:* this is restricted to those protocols that assign to given systems (end systems or intermediate systems) specific tasks to improve the global behavior of the communication. This, of course, imposes a load on these systems that should be evaluated.

When measuring the processing load introduced in a system by a given protocol, it has to be taken into account the different events that cause that load, and quantify them separately. An indicative list of such events is: received/sent packets, by type (ACK, DATA, JOIN,...), timer management, or driver interrupts. Different approaches to obtain a processing load figure from these data are just the number of occurred events, or use profiles of load taken for real systems [13,15] processing similar events, and weighting each event with its estimated profile.

2.5: Test case selection

The objective of test case selection is the definition of the parameters that will be modified, and its range of variation, to determine its effect over the protocol parameters. Some examples are the variation of the group size, the error rate or router failure rate. This is not a critical step, as this does not affect the design of the simulator as much as the definition of protocol parameters.

3: Case study: a comparison between RMNP and CTES

This section presents an application of the methodology presented along section 2. Among the many published protocols, there is a group considered particularly interesting. The common aspect to them is that they construct a control tree [4,10,11,14,17] in order to improve the behavior of flat end-to-end protocols. The objective of the control tree is mainly the support of two features: acknowledgment filtering and local recovery. Acknowledgment filtering is a technique oriented to solve implosion problems caused at the source by sender-based approaches to reliability. Local recovery aims to allow a faster and less resource-consumption of retransmission, by a) retransmitting from places closer to the member(s) missing the packet and b) restricting the scope of the retransmission to a subset of the group.

The objective of the present study is to compare the performance obtained from implementing local recovery based on two types of control trees. The first approach is based on building the control tree using a subset of the routers along the underlying multicast distribution tree. The second approach is based on building a control tree using a subset of the members of the group.

The advantage of using a subset of routers is that the criterion for restricted retransmission is simple and very effective: retransmit to the smallest subtree where the lost has been produced. When the tree is formed by a subset of the members it is necessary to define another criteria, such as using separate multicast groups for retransmission [12], using unicast retransmission under certain threshold [14], or using IP's TTL mechanism [17].

In order to compare both approaches to building the control tree, two generic protocols, derived from representative cases taken from the literature, have been selected. To analyze control trees based on routers, the protocol RMNP (Reliable Multicast Network Protocol) [2, 4] will be used. To analyze control trees based on members, a generic protocol called CTES (Control Tree based on End Systems) mainly based on [17] has been used. The definition of the protocols as defined for the study is contained in the following section.

3.1: Description of RMNP and CTES

3.1.1: The RMNP protocol

RMNP uses a sender-based approach, and uses network-located storing routers to improve the performance and scalability of the protocol. RMNP is designed to be supported on top of a multicast datagram network technology that used trees for data propagation to group members.

Building of the RMNP tree

Each RMNP connection has an associated control tree where the source is the root, the nodes are particular routers in the path, called Storing Routers (SRs), the members are the leaves. The RMNP tree is built during the connection establishment phase and it is matched to the distribution tree used by the underlying multicast routing sub-layer [3,6,7]. The routers that do have RMNP functionality and enough resources will become SRs. Over the lifetime of the connection, the RMNP tree grows and shrinks dynamically as a consequence of additions and deletions to and from the group membership.

Packet distribution

Multicast RMNP packets are sent to members using conventional IP multicast. As a consequence, the packets flow down the tree created by the underlying routing protocol. As RMNP is sender based, the source will wait until having received positive acknowledgment before deleting data from the buffer, and advancing the transmission window. However, in order to avoid ACK implosion, each SRs along the path will aggregate the ACKs, sending only one as a result of having received all the ones downstream from itself.

Packet acknowledgment and ACK aggregation

The acknowledgment process is initiated by the group members and it propagates through the RMNP control tree towards the source, using an aggregation process at the SRs belonging to the RMNP tree.

To perform the aggregation process a SR uses variable HAU, which contains the highest packet acknowledged to its RMNP father, this is, the sequence number contained in the last ACK sent. The SR also records in variables HAD_i the highest packet confirmed by its RMNP children, this is, the sequence number contained in the last ACK received from each of them. Whenever an ACK is received or sent, these variables are updated. An incoming ACK_n acknowledges all packets sent up, but nor including, sequence number *n* (accumulative acknowledgment).

Whenever an incoming ACK is equal, or greater than, a sequence number that has already been acknowledged by all the remaining RMNP children, then the SR will send upstream an ACK with the largest sequence number acknowledged by all downstream neighbors.

By means of this process the acknowledgments are propagated towards the source. When the source receives an acknowledgments from all its downstream neighbors, it actualizes its variables and frees up the associated buffers.

Error Control

Each SR (or the source) activates a timer for every packet that has been sent to its sons in the tree, and buffer the packet for an eventual retransmission. When an ACK is received from every son, the timer is stopped and the buffer is freed. In case any system of the RMNP tree detects a loss (sequence number control), it sends a NAK to its father in unicast mode. Sequence number control and NAK generation allows a faster recovery of packet loss.

Local loss recovery is implemented based on that both the source and the SRs do store the packets which are pending acknowledgment, and consequently packet retransmission is originated from the closest upstream SR (or the source). This serves also to implement restricted scope recovery, because the NAK is not propagated across the SR, and therefore the retransmission is concealed to the subtree rooted in the retransmitting SR. Simple reject is used because the combined effect of early loss detection and restricted scope retransmission did not appear to justify the complexity of selective reject.

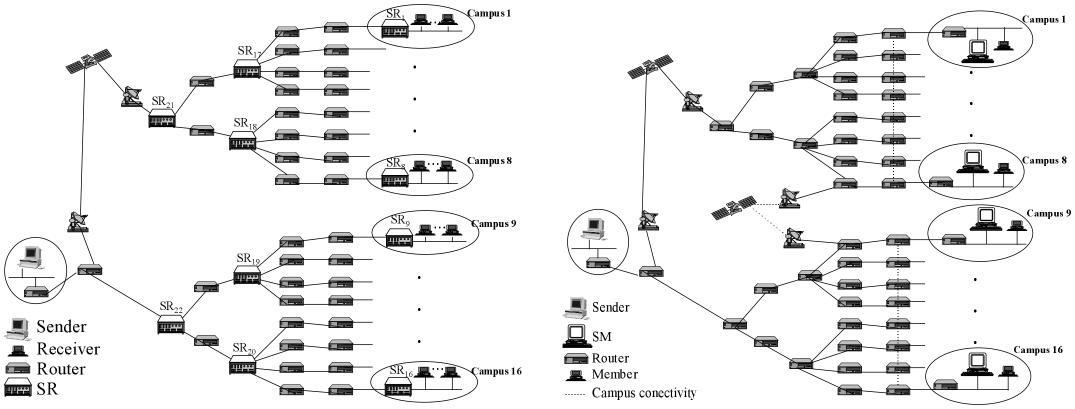

Fig. 1 Topologies used for simulations of RMNP (left) and CTES (right)

As time-out determination is a critical parameter, the protocol includes a procedure to automatically set up the appropriate value, based on a learning algorithm [4].

3.1.2: The generic CTES protocol

Packet distribution

In the same way that in RMNP, the multicast packets travel directly to all group members via standard IP multicast: packets sent by the source flow through the distribution tree associated to the source.

CTES organizes members into local regions and uses a Storing Member (SM) in each such region, so that the responsibilities of processing ACKs and performing retransmissions are distributed among several SMs and the source. SMs are organized hierarchically in a tree (CTES tree).

All members send periodically an ACK towards its father in the CTES Tree (SM or source). A difference with RMNP is that the SM sends a periodic ACK towards its father in the CTES tree without regard to having received or not the corresponding ACKs from its sons. Therefore, the source may receive an ACK from all its sons, while there are members that have not received that packet. This approach is also used in TMTP[17] and RMTP[14].

Error Control

The Generic CTES protocol uses recovery with restricted scope. Retransmissions have a restricted scope by setting the TTL value in the IP header to a “small” value. This approach is also used in TMTP[17]. Among the protocols that use a control tree built from group members, some of them follow a sender-based approach [14], while others are receiver-based [17]. As the objective of this work is to show how to compare a specific set of features (in this case, a member-built control tree and a router-built control tree) the generic CTES protocol presented here is sender-based, like RMNP.

In our generic CTES the source and each SM relies on periodic ACKs (from its immediate sons), timeouts, and retransmissions to ensure reliable delivery to its children. When a retransmission timeout occurs, the source or SM starts a retransmission process using multicast messages with a small value in the TTL field.

In addition the protocol defined uses NAKs to respond quickly to packet losses. When a member notices a gap in the sequence number, the member sends a NAK (unicast) to his father in the CTES Tree. When the source or SM receives a NAK starts a retransmission process using multicast messages with a small value in the TTL field.

When a SM receives a NAK and it does not have the requested packet, it waits until his father sends the packet to him before retransmitting the packet to his children. When an SM (or the source) has received a confirmation to a given packet from all of its sons in the CTES tree, it stops the timer and deletes the buffered copy of that packet.

3.2: Network and application model

The selected application model is one-to-many. The application delivers data according to an exponential distribution of inter-packet time, with an average of 0.2 seconds. The selected topology is formed by seventeen campus networks interconnected by a WAN network. The source is located in one campus network and the members are distributed along the remaining sixteen campuses. It is assumed that the distribution tree has been constructed by an underlying routing protocol that uses source trees, such as DVMRP [6].

Figure 1 depicts the topology of the source distribution tree for both RMNP and CTES. Notice that this tree is the same as the recovery tree in RMNP, but that is not

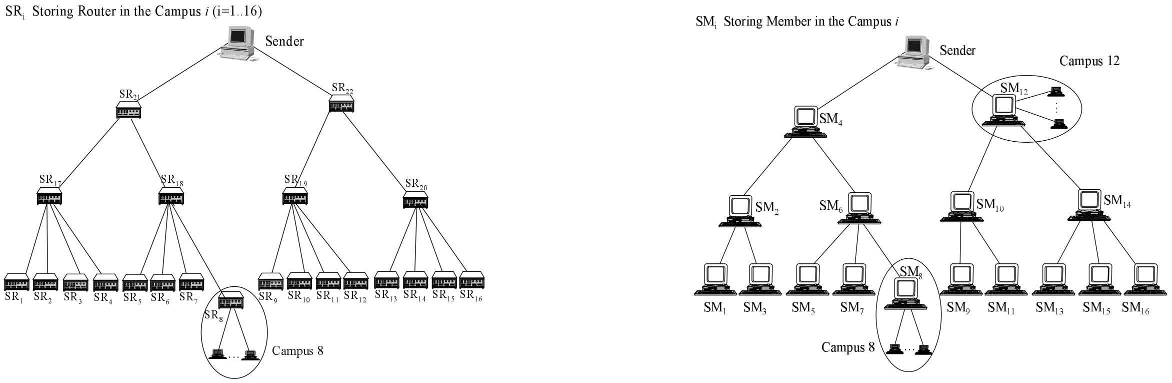
SR_i Storing Router in the Campus i (i=1..16)
Sender
SR_{21}
SR_{22}
SR_{17}
SR_{18}
SR_{19}
SR_{20}
SR_1 SR_2 SR_3 SR_4 SR_5 SR_6 SR_7
SR_8
SR_9 SR_{10} SR_{11} SR_{12} SR_{13} SR_{14} SR_{15} SR_{16}
Campus 8
SM_i Storing Member in the Campus i
Sender
Campus 12
SM_{12}
SM_4
SM_2
SM_6
SM_{10}
SM_{14}
SM_8
SM_1 SM_3 SM_5 SM_7 SM_9 SM_{11} SM_{13} SM_{15} SM_{16}
Campus 8

Fig. 2 Trees used by RMNP (left) and CTES (right)

the case for CTES. Data in CTES sometimes flows along the source distribution tree and sometimes along the SMs distribution trees (this is the case we have when a SM retransmits a packet)

In order to compare CTES in fair conditions, a number of links between the intermediate routers have been added to allow a reasonable built CTES and SMs distribution trees. The SMs distribution trees are not shown in this figure, but they indeed make use of the extra links. ACKs and NAKs travel in unicast mode along the CTES tree following the shortest path.

Transfer delay was assumed to 270 ms for satellite links and 12 ms for the rest of wide area links. Transfer delay within campus networks was fixed at 5µs. Error probability by WAN hop was assumed to be 0,5 % for message loss caused by buffer overflow or bit errors, and 0,01% by campus hop. The data rate is 2 Mbps in WAN links and 100 Mbps in campus links.

3.3: Parameters and structure of the protocols

Figure 2 shows the control trees for both RMNP and CTES. The RMNP only shows in detail the configuration for campus 8 because the structure is the same at all campuses: the members in one campus are the sons of the SR of that campus. The CTES tree has been built assuming also that there is a single SM at each campus, and all the members in the campus are sons of that SM.

The tree that connects the SMs is shown in the right side of the figure. Notice that the figure only shows the relation between SMs and only shows in detail campus 12 and 8. The selected TTL value for SMs is 6 for SMs that have at least one son which is itself an SM, and 1 for SMs whose sons are exclusively members.

Common Framework

PDU size is 1088 bytes (64 control plus 1024 data), and ACK and NACK size is 44 bytes. An important CTES parameter that has been included in the common framework is the periodic ACK interval (T_{ACK}). The merits of periodic ACKs vs. cumulative ACKs is not an objective of the comparison performed, and that is why it has been included in the common framework. It is considered that the periodic ACK mechanism is orthogonal to the way the control tree is built, and for this reason the control tree simulation comparison will render a precise result, not biased by other mechanisms that are different in the two simulations performed. The value assigned to T_{ACK} is 0.25 sec.

3.4: Evaluation metrics

Distribution delay is characterized by means of parameter **RMD**, defined as the average time elapsed since a packet is sent from the source until it has been correctly received by all the group members.

The load imposed on the WAN is estimated using two parameters: load on the routers and load on the lines. Load on the routers is characterized by parameter **CE**, defined as the average number of packets processed by one router quoted by the number of (new) packets send by the source. Load on the links is characterized by parameter **CLW**, defined as the average number of bytes transmitted over one link in the WAN, quoted by the number of bytes actually sent by the source. An important difference derived from the protocol approach is that the retransmitted packets do not follow the same routes (see Figure 1). This means that parameter CLW is not enough to actually estimate the differences in WAN link load, inasmuch as the number of hops of retransmitted packets is not the same. This aspect is characterized by accumulating the bytes transmitted through every WAN link (**T_{OCT}**) and normalizing it by the bytes sent by the source.

To estimate the potential to produce congestion at given places, the utilization factor of lines (ρ) is used. The simulator performs a periodic analysis (every 25 ms simulated time) of instantaneous link loads, accumulating the average.

Processing load on the source is estimated using parameters CF (CPU) and MF (memory). **CF** is computed by accumulating the number of received packets, the number of timers operated and the number of retransmission requests served. This total is quoted by the number of (new) packets sent by the source. Parameter **MF** is computed by periodically (every 25 ms simulated time) analyzing the instantaneous memory consumption of the source, accumulating the average.

Load on receivers is estimated using parameter **CR**, defined as the average number of packets received by one receiver (excluded SMs), quoted by the number of packets sent by the source. Notice that in the CTES protocol SMs have additional functions to that of a regular receiver, and because of this the load on SMs has a specific metric defined below.

Load on SRs (for RMNP) and SMs (for CTES) is estimated using parameters **C.SR** and **C.SM**. Both parameters are defined as the accumulation of received and processed packets, timers operated and retransmissions served, quoted by the number of packets sent by the source.

Memory on SRs and SMs is estimated using parameters **M.SR** and **M.SM**, computed by periodically (every 25 ms simulated time) analyzing the instantaneous memory consumption of the source, accumulating the average and registering the peak value. In addition to this, it has been computed the same parameter but restricted to SRs located within the WAN (**M.SR.W**). The rationale for doing it is that systems located in the campus are considered less critical for different reasons: first, SRs in the WAN will have to serve control trees from many different

	RMD (msec)		**CE**		**CLW**		**T_{OCT}**		**ρ**		**CF**	
	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES
64	444.95	978.36	3.106	2.689	1.092	1.375	64.475	101.775	5887	7075	3.111	3.044
192	441.18	980.44	5.246	2.705	1.092	1.388	64.442	102.762	5873	7163	3.075	3.067
400	443.43	977.92	8.738	2.703	1.092	1.391	64.474	102.986	5906	7156	3.094	3.039
592	441.63	983.60	11.987	2.658	1.093	1.365	64.517	101.076	5823	7139	3.086	2.996
800	442.03	979.56	15.509	2.719	1.092	1.415	64.417	104.720	5828	7332	3.100	3.039

	MF		**CR**		**C.SR**	**C.SM**	**M.SR**	**M.SR.W** **avg – max**	**M.SM** **avg - max**
	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	RMNP	CTES
64	4.520	4.488	1.000	1.574	6.533	5.969	0.30	1,01 - 10	3,41 - 20
192	4.485	4.920	1.002	1.592	12.373	13.031	0.30	1,01 - 10	3,39 - 18
400	4.531	4.873	1.005	1.591	21.901	24.537	0.30	1,02 - 10	3,42 - 19
592	4.474	4.964	1.008	1.566	30.763	34.521	0.30	1,02 - 9	3,41 - 19
800	4.479	4.919	1.010	1.617	40.375	46.353	0.30	1,01 - 10	3,42 - 17

Table 1. Effect of the group size

	RMD (sec)		**CE**		**CLW**		**T_{OCT}**		**ρ**		**CF**	
	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	CTES
Basic WAN	0.442	1.041	8.742	2.672	1.093	1.382	64.476	102.262	5814	7224	3.079	2.989
WAN$_1$	0.516	2.200	8.756	2.968	1.107	1.691	65.296	125.116	5862	9148	3.125	3.135
WAN$_2$	0.586	3.341	8.768	3.341	1.121	2.043	64.142	151.166	6109	11040	3.126	3.352
WAN$_3$	0.672	4.385	8.786	3.688	1.142	2.383	67.389	176.333	6160	13078	3.116	3.572
WAN$_4$	0.852	4.628	8.807	4.037	1.166	2.690	68.779	199.085	6276	14554	3.164	3.800
WAN$_5$	1.045	5.847	8.826	4.298	1.186	2.939	69.949	217.538	6425	15990	3.217	3.969

	MF		**CR**		**C.SR**	**C.SM**	**M.SR**	**M.SR.W** **avg - max**	**M.SM** **avg - max**
	RMNP	CTES	RMNP	CTES	RMNP	CTES	RMNP	RMNP	CTES
Basic WAN	4.462	4.964	1.006	1.579	21.911	24.157	0.30	1.01 - 10	3.42 - 18
WAN$_1$	5.353	5.903	1.006	1.909	21.942	24.149	0.44	1.49 - 14	3.68 - 25
WAN$_2$	6.421	6.765	1.006	2.286	21.972	24.818	0.61	2.07 - 19	3.92 - 32
WAN$_3$	7.456	7.720	1.006	2.649	22.015	25.160	0.78	2.64 - 23	4.26 - 38
WAN$_4$	8.847	8.527	1.006	2.986	22.064	25.999	0.96	3.29 - 28	4.46 - 40
WAN$_5$	10.521	9.554	1.007	3.256	22.103	26.305	1.14	3.89 - 32	4.75 - 46

Table 2. Effect of the WAN diameter

locations, whereas those located in the campus serve only the systems in the campus organization; second, as the WAN has slower links and higher loss probability, the load of SRs in the WAN will be much higher than those in campus, and therefore the average SR load is not representative.

Flow control aspects have not been metered, as they are not the objective of the simulation.

3.5: Test cases

The test cases have been aimed to ascertain the effect of varying group size and group diameter (in terms of delay and loss probability of the connecting WAN). The first set of tests was performed altering the number of group members between 64 (4 members per campus) to 800 (50 per campus). The network topology has been maintained. Table 1 shows the results derived from this test set.

Along the second test set, the number of group members has been maintained to 400 (25 per campus), while varying the WAN diameter. Table 2 shows the results derived from this second test set. The network topology has been maintained, and the effect of altering WAN diameter has been achieved by altering the network transit delay and the network loss probability.

Five cases have been under study, named WAN$_1$ (the smallest) to WAN$_5$ (the largest). The following table shows the quantitative modelization of WAN networks of different size, while maintaining the same topology.

	Parameter		
WAN type	***r***	t_p	$\overline{p}$
WAN$_1$ (1,5 times basic *WAN*)	9	18	0.007
WAN$_2$ (2 times basic *WAN*)	12	24	0.009
WAN$_3$ (2,5 times basic *WAN*)	15	30	0.012
WAN$_4$ (3 times basic *WAN*)	18	36	0.015
WAN$_5$ (3,5 time basic *WAN*)	21	42	0.017

r = *Router forwarding delay* (ms)
t_p = *WAN link propagation delay* (ms)
$\overline{p}$ = *Loss probability in a WAN link*

Table 3. Characterization of the different WANs parameters

In both test cases the source has generated a total of 15 Mbytes that has been successfully received by all members.

3.6: Analysis of results

The simulation results presented have been obtained using the MaRS (Maryland Routing Simulator) tool, which is an event-driven network simulation tool by Maryland University [1].

Many simulation executions have been conducted al-

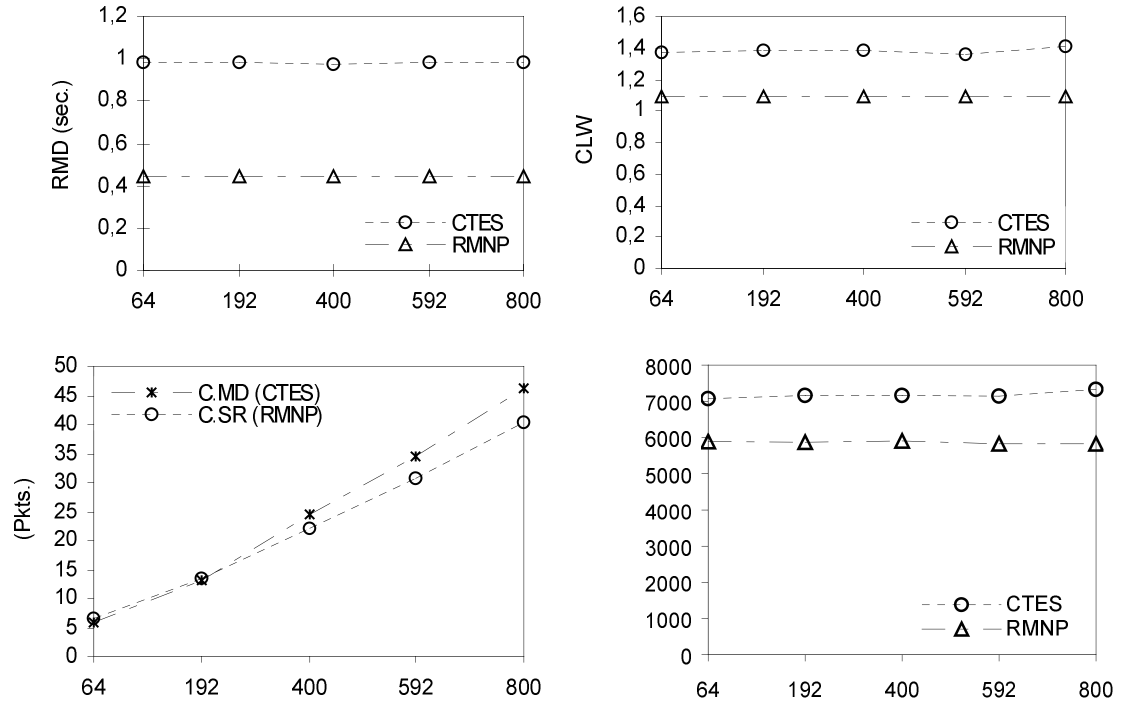

Fig. 3 Group size

tering different parameters (source traffic pattern, timeout values, ...). For conciseness reasons, it is not possible to present in this paper all the actual test cases and simulation results obtained, all of which give credit to the conclusions presented in this section. However, the results that are actually presented in the paper are considered as sufficiently representative so as to give the reader enough confidence in these conclusions.

Figure 3 show the effect of group size both for RMNP and CTES. It may be concluded that both protocols are successful in avoiding implosion problems at the source, and that both protocols scale in relation to group size. A more detailed analysis shows that:

1. The mean distribution delay (RMD) exposes a better behavior in RMNP than CTES. A possible cause for this is that RMNP performs intermediate sequence number control, allowing a faster loss detection, and that local retransmission along the underlying distribution tree guarantees lower reception delay than doing so along the CTES tree (notice that loss probability is higher in WAN links).
2. WAN bandwidth requirements are better in RMNP than CTES (around a 20% saving in transmission costs). A possible cause for this, similar to the previous effect, is that loss probability is higher in the WAN, and therefore retransmission in CTES transverse more WAN links.
3. Both RMNP and CTES expose acceptable utilization factor ρ values, although RMNP values are always better.
4. Processing load on control tree systems (SRs or SMs) is similar.
5. Average memory requirements are higher for SMs than for SRs. A possible cause may be that each SM that is not a leaf in the CTES tree has to wait for the ACKs from its sons located in distant campuses.

Figure 4 show the impact of WAN size both on RMNP and CTES. The behavior of RMNP is much better in terms of mean distribution delay, network load and processing load on SRs and SMs. The comparative behavior of RMNP improves as the WAN size increases.

A first cause for RMNPs superiority is that it is better adapted to network topology by using the underlying distribution tree as a pattern for the overlaid control tree. If for example a packet is lost half way the distribution tree, the probability of an SR having it is higher than an SM having it, and therefore the local recovery mechanisms may be used on more occasions. Another effect is that the TTL criterion used for locality is less restrictive than using a subtree.

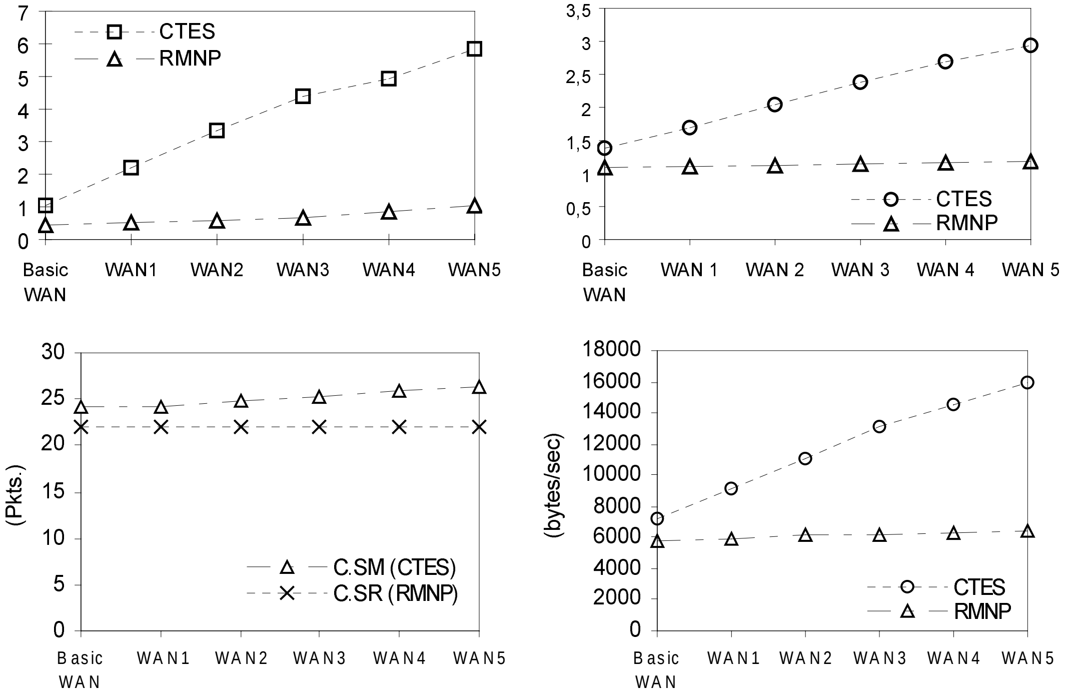

Fig. 4 WAN Diameter

Another cause is the ACK aggregation mechanism of RMNP vs. the direct acknowledgment mechanism of CTES. Performing direct acknowledgment has the objective of freeing buffer space, but has several drawbacks: first, impoverishes reliability, as the packet is deleted from the source before all members have received it, while the SM that actually has the copy might crash. This problem does not exist in RMNP. Second, as ACKs are also used to perform flow control, in CTES flow control is hop-by-hop instead of end-to-end, possibly causing saturation of part of the tree, without notification to the source implying buffer saturation on the corresponding SMs.

4: Conclusions and future work

Comparing in a rigorous way the performance of different reliable multicast protocols is a complex task that requires a very systematic approach. This paper has presented a first set of guidelines that still has to be refined to allow a more systematic application. Future work could be aimed to building up a set of network topologies, and overlaid distribution trees, over which any new proposal could be tested (following the approach used in processor benchmarks, that are based on a widely accepted set performance tests). Another area of work would be standardizing the set of metrics presented in this paper to allow the comparison of results. The isolation of specific characteristics to be compared, and its modelization to be simulated, remains as the point to be studied on a case by case basis

In relation to the concrete example that has been presented it may be concluded that using local recovery is an effective mechanism to allow the scalability of reliable multicast protocols. It has also been shown the superiority of control trees following the underlying distribution tree. It has to recognized the "de facto" difficulties on getting enough consensus and critical mass so as to provide this functionality at even a subset of routers (an R-mbone) as proposed in RMNP. But it is also possible to encounter difficulties in having one organization accepting the overload of becoming SM for the benefit of other organizations performance, as proposed in CTES.

In relation to the RMNP protocol, it should be considered the convenience of introducing a selective reject mechanism because of the predominance of solitary losses (a single lost packet preceded and followed by successful reception) as stated on studies performed over Mbone [16]. Another area of research is the provision of heuristics to perform an automatic selection of SRs across the WAN on a group by group basis.

References

[1] C. Alaettinoglu, U. Shankar, K. Dussa-Zieger and I. Matta. MaRS (Maryland Routing Simulator) - version 1.0 user's manual. *Technical Report, CS-TR-2687*, Department of Computer Science, University of Maryland, June 1991.

[2] A. Azcorra and M. Calderón. "A network-based approach to reliable multicast". *Proc. of the second workshop on Protocols for Multimedia Systems PROMS'95*, Salzburgo, Austria, pp. 393-404, October 1995.

[3] A.J. Ballardie, P.F. Francis and J. Crowcroft. "Core Based Trees". *Proc. of ACM SIGCOMM'93*, September 1993

[4] M. Calderón. "*Unificación de los protocolos de multipunto fiable optimizando la escalabilidad y el retardo*", Ph.D. Thesis, Facultad de Informática, Universidad Politécnica de Madrid, October 1996.

[5] S. Deering. "Host Extensions for IP multicast". *RFC 1112*, August 1989

[6] S. Deering. "*Multicast Routing in a Datagram Internetwork*". PhD Thesis, Stanford University, December 1991.

[7] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei. "An Architecture for Wide-Area Multicast Routing". *Proc. of ACM SIGCOMM'94*, 24(4):126-135, September 1994.

[8] S. Floyd, V. Jacobson, S. McCanne, C.G. Liu and L. Zhang. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing". *Proc. of ACM SIGCOMM'95*, pp. 342-356, August 1995.

[10] M. Hofmann. "Enabling Group Communication in Global Networks". *Proc. of Global Networking'97*, June 1997.

[11] H.W. Holbrook, S.K. Singhal and D.R. Cheriton. "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation". *Proc. of ACM SIGCOMM'95*, August 1995.

[12] S.K. Kasera, J. Kurose and D. Towsley. "Scalable Reliable Multicast Using Multiple Multicast Groups". *Technical Report TR 96-73*, University of Massachusetts, October 1996.

[13] J. Kay and J. Pasquale. "The Importance of Non-Data Touching Processing Overheads in TCP/IP". *Proc. of ACM SIGCOMM'93*, September 1993.

[14] J.C. Lin and P. Sanjoy. "RMTP: A Reliable Multicast Transport Protocol". *Proc. of IEEE INFOCOMM'96*.

[15] C. Partridge and S. Pink. "A Faster UDP". *IEEE/ACM Transactions in Networking*, 1(4):429-439, August 1993.

[16] M. Yajnik, J. Kurose and D. Towsley. "Packet Loss Correlation in the Mbone Multicast Network". *Proc. of IEEE Global Internet Conference*, December 1996.

[17] R. Yavatkar, J. Griffioen and M. Sudan. "A Reliable Dissemination Protocol for Interactive Collaborative Applications". *Proc. ACM Multimedia'95*, November 1995.