Dealing with Heterogeneity in a Fully Reliable Multicast Protocol

Moufida Maimour and Cong-Duc Pham
RESO-INRIA, LIP-ENS - France
email:{mmaimour,cpham}@ens-lyon.fr

Abstract— Many of the proposed multicast congestion avoidance algorithms are single-rate where heterogeneity is accommodated by adjusting the transmission rate as a response to the worst receiver in the group. Due to the Internet heterogeneity, a single-rate congestion control affects the overall satisfaction of the receivers in a multicast session. In this paper, we propose a multi-rate replicated scheme where some receivers (instead of the source) are designated to perform data replication for other receivers with lower capacity. To be more scalable and to minimize the bandwidth consumption due to data replication, the partitioning algorithm is performed on-the-fly by the routers depending on the feedback they receive. Neither a prior estimation of the receivers capacity is necessary nor a complex computation is required to execute our partitioning algorithm.

I. Introduction

The problem of reliability in multicast has been extensively studied and many proposals exist in the literature. However congestion appears to be the most common reason for packet loss in the Internet. Therefore, a reliable multicast protocol requires congestion control to be addressed. In a previous work, we proposed a framework for bulk data distribution with two components, a reliable protocol, DyRAM (Dynamic Replier Active reliable Multicast) [1] and a congestion avoidance protocol, AMCA (Active-based Multicast Congestion Avoidance algorithm) [2].

AMCA is a single-rate congestion avoidance protocol where heterogeneity is accommodated by adjusting the source rate in response to the most congested path in the multicast tree. This path is dynamically determined based on round trip time (RTT) variations estimated in a per-hop basis. Transmitting with a rate which matches the slowest receiver would limit the throughput of other receivers and thus their *satisfaction*. In a multicast session, a multi-rate mechanism can improve the receivers' satisfaction. In fact, receivers with different needs can be served at a rate closer to their needs rather than having to match the speed of the slowest receiver. In a multi-rate session, the multicast source can transmit at different rates either through a hierarchical scheme (layering) [3], [4], [5] or a replication-based scheme (destination set grouping, DSG [6]). In layered multicast, each receiver controls the rate at which it receives data, usually by using multiple multicast groups. The receivers join and leave groups depending on their path congestion state so the amount of data being received is always appropriate. In a replication-based scheme, the source splits the receivers into subgroups of similar capacities or *isolated rates*[1]. Afterwards, it sends a separate flow to every subgroup with the appropriate rate. Layering schemes provide more economical bandwidth usage than DSG schemes, however layering is more complicated and requires efficient hierarchical encoding/decoding algorithms and synchronization among different layers.

In order to avoid the complexity inherent to a layered scheme especially for the case of a fully reliable multicast, we propose to use a replicated scheme. However our scheme differs from the existing replicated schemes in the following main points (i) the partitioning of the receivers is performed by the routers instead of the source, (ii) the partitioning algorithm is executed on-the-fly depending on the RTT variations instead of the receivers' isolated rates which are very difficult to be estimated in the current Internet, and (iii) the data replication is no longer the responsibility of the source. More precisely, our solution consists in a distributed approach where some receivers (*replicators*) contribute in the replication of the data flow with an appropriate rate to other receivers of lower capacity. In this way, a *regulation tree* is built with the source as the root, the replicators as intermediate nodes and the remaining receivers as final nodes. In order to minimize bandwidth consumption due to data replication, the regulation tree is built with respect to the physical multicast tree. This is achieved by involving the routers in the regulation tree construction procedure. Every router performs a partition of its downstream links into subgroups and chooses a replicator for every subgroup formed. In addition to the construction of a regulation tree with a topology close to the physical multicast one, executing the partitioning algorithm at the routers instead of the source is more scalable since (i) every router performs a partitioning algorithm locally, (ii) there is no data replication at the source which still send only one data flow, and (iii) the transmission rate of the source is no longer dictated by the worst receiver in the whole multicast group.

The rest of this paper is organized as follows. Section II presents a general algorithm for the construction of a regulation tree. An overview of the DyRAM/AMCA framework is presented in section III. Afterwards, section IV applies the proposed algorithm for the extension of AMCA to support more heterogeneous receivers. Some simulation results are the object of section V, and section VI concludes.

II. Accommodating Heterogeneity through a Regulation Tree

In this section, we present our distributed algorithm for the construction of the regulation tree with some illustrative examples.

A. Background

Our approach relies on the partitioning of the receivers into subgroups with similar capacities so their overall satisfaction is maximized. Our adopted partitioning algorithm is based on the *relative RTT variation* (noted $\Delta\dot{\tau}$), defined as the ratio of a measured RTT variation to the periodicity of this measure. We have $\Delta\dot{\tau} = \Delta\tau/T$ where $\Delta\tau$ is the RTT variation and T is the period duration. In order to quantify the satisfaction of a receiver R_i in a multicast session, we use a utility function

[1] A receiver's isolated rate [7] is defined as the rate that this receiver would obtain if unconstrained by the other receivers in the group, assuming max-min link sharing.

defined as follows [7]:

$$U_i(r) = \frac{\min(r_i, r)}{\max(r_i, r)} \quad (1)$$

where r_i and r are respectively the isolated and the reception rate of receiver R_i. The receiver satisfaction is maximized when its reception rate is equal to its isolated rate, $U_i(r_i) = 1$. A receiver that receives data with a rate which is greater than its isolated rate could experience losses. In the opposite case, the receiver will also be unsatisfied since there are some unused bandwidth in its path to the source. In a similar way to [7], we define the utility function of a single-rate multicast session as the weighted sum of the individual utility values of receivers in the session:

$$U(r) = \sum_{i=1}^{n} \alpha_i U_i(r) \quad (2)$$

subject to $\sum \alpha_i = 1$ and $\alpha_i \in [0, 1], i = 1, \ldots, n$ where n is the number of the receivers in the multicast session. A multi-rate multicast session consists of one or more subgroups split from an original multicast session. The session utility function in this case is defined as the summation of the utility values obtained by all multicast subgroups, using the single-rate utility measure in each subgroup. More specifically, if a multicast session of receivers $\{R_1, R_2, \ldots, R_N\}$ is split into K subgroups $\{G_1, G_2, \ldots, G_K\}$ with different transmission rates $g_1, g_2, \ldots, g_K$, then

$$U(g_1, g_2, \ldots, g_K) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \alpha_{i,j} U_{i,j}(g_j) \quad (3)$$

subject to $\sum_{i,j} \alpha_{i,j} = 1$ and $\alpha_{i,j} \in [0, 1]$. We have $\sum_j n_j = N$ where n_j is the number of the receivers in subgroup G_j. $U_{i,j}(g_j)$ and $\alpha_{i,j}$ are respectively the utility function and the weight associated to the ith receiver of the jth subgroup. Since we are concerned with a fully reliable multicast, the transmission rate g_j for the receivers in subgroup G_j has to match the minimum rate of the subgroup isolated rates, i.e. $\forall j, \quad g_j = \min_{i \in G_j} r_i$

B. Regulation Tree Construction

The regulation tree construction is a distributed process where each router performs locally a partitioning of its downstream links into subgroups and designates a replicator for every subgroup formed. Algorithm 1 shows how a router contributes in building such a regulation tree, on-the-fly depending on the RTT variation feedback it receives. We do not show the transmission rate adjustment in this algorithm since it is the role of the source and the replicators. A replicator as will be seen, has to adjust its replication rate depending on feedback it receives from its children receivers in the regulation tree.

Initially, a router maintains a list P_0 that contains all of its downstream links. Every time, a downstream link l_j experiences a relative RTT variation ($\Delta\dot{\tau}_j$) greater than a given parameter b, the router creates a new partition P_i with all the links that experienced a relative RTT variation greater than a parameter a ($a < b$) and selects a replicator Rep_i for this subgroup. We note by $Rd(P_i)$ the set of direct receivers located downstream from the P_i links. The function $Best(P_i)$ returns for subgroup P_i, the identity of a receiver from $Rd(P_i)$ which has the highest estimated capacity[2]. For instance, when the subgroup P_i is split from P_0, then $Best(P_0)$ is elected as its replicator. Since this replicator may have been chosen for P_{i-1} in the previous split, $Best(P_i)$ is definitely elected as the P_{i-1}'s replicator (see algorithm 1). The properties and details of our partitioning algorithm are beyond the scope of this paper, a deeper study of this algorithm is provided in [8].

Algorithm 1 Regulation tree construction at a router

Require: $N > 1$ and $a < b$
  $P_0 \leftarrow \{l_j, j = 1, \ldots, N\}$, the set of all the links downstream
  $i \leftarrow 1$
  Periodically,
  if $\exists j, l_j \in P_0$ such that $\Delta\dot{\tau}_j > b$ **then**
    $P_i \leftarrow \{l_j \in P_0, \Delta\dot{\tau}_j > a\}$
    $P_0 \leftarrow P_0 - P_i$
    $Rep_i \leftarrow Best(P_0)$
    if $i > 1$ **then**
      $Rep_{i-1} \leftarrow Best(P_i)$
    end if
    $i \leftarrow i + 1$
  end if
  until no split is possible

In our approach, a router forwards the source data packets only on the P_0 links. Every time, a new subgroup P_i is formed and the corresponding replicator Rep_i is selected, the router notifies this latter to start performing data replication[3]. A replicator Rep_i sends its replicated data packets to the receivers of its corresponding subgroup P_i. Feedback from subgroup P_i are sent to their corresponding replicator Rep_i while those arriving on the P_0 links are forwarded to the source. In this way the source transmission rate is dictated by the worst receiver located downstream from the P_0 links. Once the regulation tree is constructed, the source (and every replicator) sends data to its children with a rate that matches their minimum isolated rate without of course exceeding its reception rate (or equivalently, its parent transmission rate).

C. Illustrative Examples

To illustrate our regulation tree construction algorithm, we consider a multicast session with seven subscribed receivers $\{R_0, R_1, R_2, R_3, R_4, R_5, R_6\}$ with respectively the following isolated rates $\{5, 6, 9, 11, 15, 19, 21\}$. All of these receivers are located downstream from the same router A (see Fig. 1). If we take $a = 0.01$ and $b = 0.26$ which correspond to $\rho = 0.8$, executing algorithm 1 produces $\{\{R_0, R_1\}, \{R_2, R_3\}, \{R_4\}, \{R_5, R_6\}\}$ as a partition with the regulation tree shown in Fig. 1a. The resulting subgroups are $P_0 = \{R_5, R_6\}$, $P_1 = \{R_0, R_1\}$, $P_2 = \{R_2, R_3\}$, $P_3 = \{R_4\}$ with respectively the reception rates, $g_0 = 19$, $g_1 = 5$, $g_2 = 9$, $g_3 = 15$, giving a session utility of 0.936 instead of 0.525 if no split is performed ($\forall i, \alpha_i = 1/7$). The selected replicator for the first split subgroup $P_1 = \{R_0, R_1\}$ is $Best(P_0) = R_6$. In the next split, when the second subgroup P_2 is formed

[2] A receiver capacity is evaluated using a metric that will be presented in the context of AMCA, in section IV-A.

[3] How this notification could be performed is addressed in section IV-A.

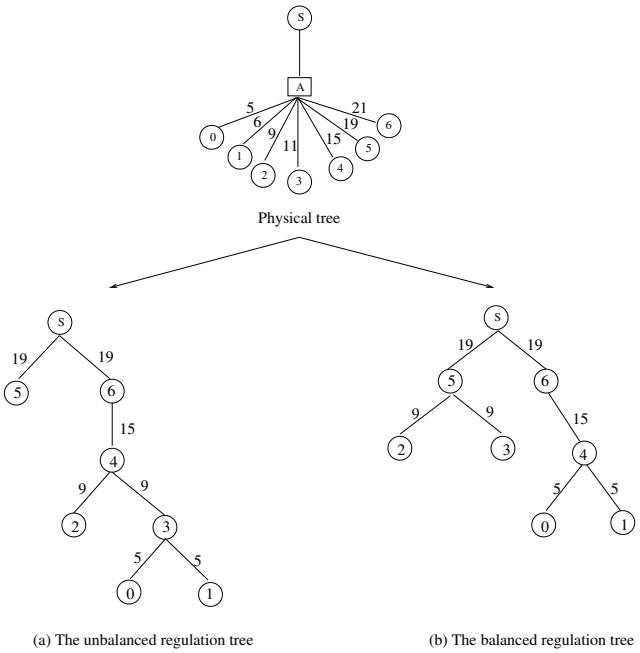

Fig. 1. Simple tree construction.

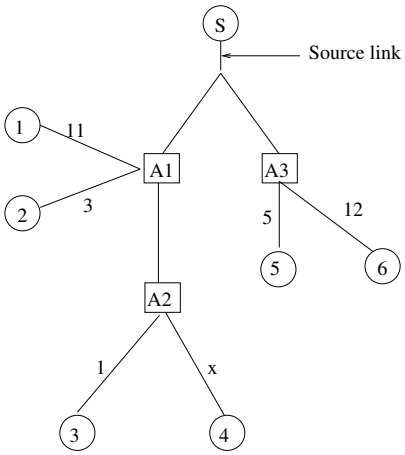

Fig. 2. Example with a hierarchy of routers

with $Best(P_0) = R_6$ as the replicator then the P_1 replicator (according to the proposed algorithm) has to be changed to $Best(P_2) = R_3$ which is definitely elected as a replicator for P_1. When $P_3 = \{R_4\}$ is split, its chosen replicator is $Best(P_0) = R_6$ and $Best(P_3) = R_4$ is definitely selected as a replicator for P_2 ...etc. We can note that the built tree is not balanced. For instance, receiver 5 could be the replicator for P_2 as shown in Fig. 1b. To have a more balanced tree, we can use another rule for selecting the replicators. In fact, when partition P_i is formed, a replicator is chosen using $Rep_i = Best(\cup_{k>i,k=0} P_k - \{Rep_k, k > i\})$

As a second example, we consider one source multicasting data to six receivers through three routers A_1, A_2 and A_3 (Fig. 2). A maximum bandwidth is given for every link in the multicast tree. The link (A_2,R_4) has a bandwidth of x which will be set either to 10 or 2 for the following. The main concern here is the behavior of router A_1 since it has two indirect receivers R_3 and R_4 in addition to two direct ones, R_1 and R_2. In what follows, a link will be designated by its downstream node in

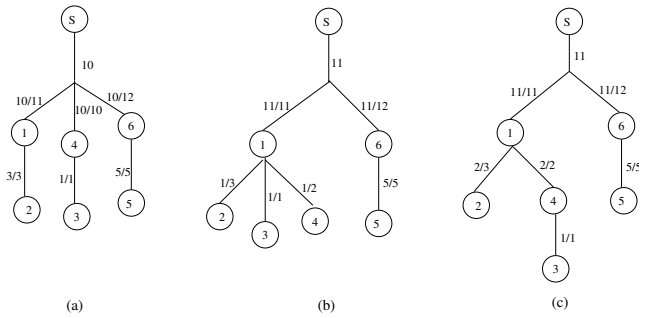

Fig. 3. Obtained regulation trees for: (a) $x = 10$, (b) $x = 2$ and $\rho < 0.5$, (c) $x = 2$ and $\rho > 0.5$

the multicast tree. For example the A_1 downstream links that lead to router A_2 and receiver R_1 are respectively noted A_2 and R_1. We also use $P_{i,j}$ to designate the ith partition of router A_j. Suppose that the maximum number of subgroups that could be built by a router is 2. If we set x to 10, then we get the following local partitions for the routers:

$$\begin{aligned} P_{0,1} &= \{R_1, A2\},\ P_{1,1} = \{R_2\} \\ P_{0,2} &= \{R_4\},\ P_{1,2} = \{R_3\} \\ P_{0,3} &= \{R_6\},\ P_{1,3} = \{R_5\} \end{aligned}$$

which gives the following overall partition seen by the source (Fig. 3a):

$$P_0 = \{R_1, R_4, R_6\}, P_1 = \{R_2\}, P_2 = \{R_3\}, P_3 = \{R_5\} \quad (4)$$

This partition (4) gives a utility value, $U_a = (1/1 + 3/3 + 5/5 + 10/10 + 10/11 + 10/12)/6 = 0.957$ instead of 0.301 if no partitioning is performed ($\forall i, \alpha_i = 1/6$). In this latter case the source would transmit data at a rate equal to 1 instead of 10. If we consider that the partitioning is performed by the source instead of the routers, then the optimal partition with two subgroups is $P_0 = \{R_3, R_2, R_5\}$ and $P_1 = \{R_1, R_4, R_6\}$ which gives a utility value of $U'_a = 0.713 < U_a = 0.957$. In order to get a utility value equal to $U_a = 0.957$, a source-based partitioning requires four subgroups. In this case, the source link will be loaded with $10 + 1 + 3 + 5 = 19$ instead of 10 since the source will send four flows with rates 10, 1, 3 and 5. This gives $19/10 = 1.9$ of additional consumed bandwidth at the source link.

If x is set to 2, then depending on ρ, we get respectively for $\rho < 0.5$ and $\rho > 0.5$:

$$\begin{aligned} P_{0,1} &= \{R_1\},\ P_{1,1} = \{R_2, A2\} \\ P_{0,2} &= \{R_3, R_4\} \\ P_{0,3} &= \{R_6\},\ P_{1,3} = \{R_5\} \end{aligned}$$

and

$$\begin{aligned} P_{0,1} &= \{R_1\},\ P_{1,1} = \{R_2, A2\} \\ P_{0,2} &= \{R_4\},\ P_{1,2} = \{R_3\} \\ P_{0,3} &= \{R_6\},\ P_{1,3} = \{R_5\} \end{aligned}$$

that gives respectively the following overall partitions:

$$P_0 = \{R_1, R_6\}, P_1 = \{R_2, R_3, R_4\}, P_2 = \{R_5\} \quad (5)$$

and

$$P_0 = \{R_1, R_6\}, P_1 = \{R_2, R_4\}, P_2 = \{R_3\}, P_3 = \{R_5\} \quad (6)$$

It is worth noting that when $x = 2$, the optimal source-based partition with two subgroups is $P_0 = \{R_2, R_4, R_3, R_5\}$ and $P_1 = \{R_1, R_6\}$ which gives a utility value of $U_b^{'} = 0.667$ instead of $U_b = 0.792$ or $U_c = 0.931$ for the partitions (5) (Fig. 3b) and (6) (Fig. 3c) respectively.

III. DyRAM/AMCA Framework: an overview

DyRAM/AMCA is a framework for reliable multicast that deals with both reliability (DyRAM) and congestion avoidance (AMCA). This section gives an overview of the DyRAM/AMCA framework before providing its extension to support more heterogeneous receivers. DyRAM uses a recovery strategy based on a tree structure constructed on a per-packet basis with the assistance of routers. It uses a receiver-based local recovery where receivers are responsible for both the loss detection (NACK-based) and the retransmission of repair packets when it is possible.

For the purpose of congestion control, AMCA uses the active networking technology to discover through a per-hop dialogue the available bandwidth along a multicast tree. The solution uses the RTT variations experienced by every branch to estimate the congestion situation in the multicast tree. Every receiver reports periodically a congestion report (CR) packet so the source can learn about congestion in the multicast tree. The CRs contain mainly information about the RTT variations used by the source to adjust the rate of the transmission. The physical multicast tree is used to aggregate appropriately the RTT variations at intermediate nodes before they reach the source. One structure which will be extended in the purpose of adding heterogeneity support to AMCA is the link state (LS) structure used to perform quick, accurate and optimized replier elections. An LS structure contains for every downstream link two fields (updated on the reception of the CRs and NACKs):

- lo, the sequence number of the last data packet received in order on this link, which corresponds to the last one received in order by all the receivers located downstream from this link.
- lr, the sequence number of the last received data packet by this link, which corresponds to the last received one by all the receivers located downstream from this link.

For more details about DyRAM and AMCA, the reader can refer respectively to [1] and [2].

A. Congestion Feedback Suppression/Aggregation

To achieve scalability, we use the physical multicast tree to hierarchically aggregate feedback at the intermediate nodes (routers). A suppression mechanism is performed on NACKs thus allowing just one NACK per loss to be sent upstream. Since we do not suppose that all the routers are active, more than one NACK for the same loss could be received by the source. To avoid reacting to duplicate NACKs, the source reacts to the first one and ignores the subsequent ones for a given period of time which depends on the multicast tree structure. Moreover, intermediate nodes aggregate the CRs they receive from their downstream links in one CR to be forwarded upstream. The aggregated CR contains the sequence number of the minimum last ordered and the maximum last received data packets among those reported by the CRs received from downstream.

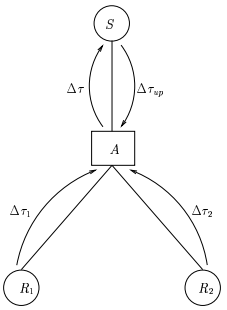

Fig. 4. CRs aggregation in a two-level multicast tree

In the DyRAM/AMCA framework, many timeouts are set depending on RTT measures. Moreover, the associated congestion avoidance algorithm requires the estimation of RTT variations. Instead of estimating directly the RTTs between the source and each receiver, we first begin by estimating the RTTs between every node and its parent in the multicast tree. Afterwards, every node will compute its RTT to the source by adding the RTT of its parent to the source and its own RTT to its parent. This computed RTT by a given node is then included in a special field of the data packets it forwards and will be used by its children in the multicast tree to compute their own RTTs to the source.

For the estimation of the RTT variations reported by the CRs and their aggregation, we illustrate our mechanism with the simple two-level tree depicted in Fig. 4. We consider two receivers R_1 and R_2 connected to the source S via one active router A. Receivers R_1 and R_2 send CRs to the active router with their respective RTT variations $\Delta\tau_1$ and $\Delta\tau_2$ (estimated using the last two RTT measures of their upstream links). Once these children CRs are received, the active router A sends its CR to the source with $\Delta\tau$ computed as follows:

$$\Delta\tau = \Delta\tau_{up} + Max(\Delta\tau_1, \Delta\tau_2)$$

where $\Delta\tau_{up}$ is the RTT variation of the source link (A, S). Using this method to aggregate the RTT variations, the source ends up by receiving the RTT variation experienced by the worst end-to-end path of the multicast tree.

B. Rate Adjustment

In AMCA, a minimum and a maximum rates r_{min} and r_{max} are set by the application. Any receiver that could not support the minimum transmission rate has to leave the multicast session. Initially, the source starts to send data packets with a rate equal to the minimum rate r_{min}. Then it tries to increase its rate if no congestion indication is received without exceeding the maximum rate r_{max}. The source uses the information fed back in the CRs and NACKs to update its rate. The RTT variation field of every CR is used by the source to compute the queue size variation per packet Δq_p during the previous period. We have:

$$\Delta q_p = \frac{\Delta\tau}{\Delta\tau + T} \quad (7)$$

where $\Delta\tau$ is the RTT variation experienced by the worst end-to-end path of the multicast tree. The rate regulation at the source is also based on another metric that estimates the number of data packets which are not acked yet by all the receivers Δq_a given by:

$$\Delta q_a = \max(0, N - (lo_{i+1} - lo_i)) \quad (8)$$

where N is the number of packets received during the period value T. lo_{i+1} and lo_i are respectively the values of the last ordered field of the newly and previously received CR. Note that Δq_a gives a measure of the difference between the emission and the reception rate.

The source, on the reception of a CR, extracts the current period (T) and the RTT variation ($\Delta\tau$). Afterwards, it computes the queue size variation per packet Δq_p and the number of data packets not acked yet Δq_a using (7) and (8). The aim is to maintain $\Delta q_p < \epsilon$, where ϵ is a positive number to be chosen in $[0, 1]$ and Δq_a in $[\alpha, \beta]$, where α and β are similar to the TCP-Vegas parameters. During a phase similar to the TCP slow start, the source continues increasing its rate by S/RTT_{max} (bits per second) every time it receives a CR that indicates $\Delta q_p < \epsilon$ and $\Delta q_a < \beta$. Increasing the rate by S/RTT_{max} where RTT_{max} is the maximum experienced RTT by the receivers, is equivalent to adding 1 to the congestion window for the largest end-to-end connection of the multicast tree. This behavior makes our congestion avoidance algorithm fair with TCP from the beginning of the multicast session. The source enters the congestion avoidance phase when it receives a NACK or a CR with $\Delta q_p > \epsilon$ or $\Delta q_a > \beta$. On the receipt of a NACK, the source reduces the rate by half. Subsequent NACKs are ignored during a period estimated by the difference between the current RTT to the source of the farthest and the closest receiver in the multicast tree ($RTT_{max} - RTT_{min}$). To avoid decreasing dramatically the rate because of isolated losses, we check the lo and lr fields of the NACKs and reduce the rate only if $lr - lo \geq 3$, otherwise we just retransmit the corresponding repair. During the congestion avoidance phase, on the reception of a CR with an RTT variation $\Delta\tau$ and a period T, the source updates its rate depending on the current values of both Δq_p and Δq_a. When these two measures do not indicate congestion ($\Delta q_p \in [0, \epsilon]$ and $\Delta q_a < \beta$), the rate is multiplied by γ chosen slightly greater than 1 (say 1.025). In the other cases, the rate is multiplied by $\gamma = T/(T + \Delta\tau)$. The rational behind this can be found in [2].

IV. AMCA Extension

In order to implement our regulation tree scheme, two approaches are possible. One approach could consist in the use of multiple multicast addresses, one per subgroup. Each replicator sends its data flow to a multicast address subscribed to by only its children in the regulation tree. In this case we need to implement a mechanism that would allow receivers (associated to a given replicator) to be aware of the multicast address on which the replicator is transmitting. For the extension of the AMCA protocol, we have adopted another approach based on the active networking technology. An active router maintains local information about the regulation tree and notifies the replicators to start the data replication. A replicator sends its flow toward its upstream active router which will forward every flow on the appropriate set of links. For practical considerations and ease of implementation, we chose to limit the number of subgroups maintained by an active router to 2. In this way, routers are not overloaded by the management of multiple subgroups. However, since every router performs locally its own partitioning procedure, the overall number of subgroups seen by the source can be much higher. For instance, if we consider a two-level multicast tree with N intermediate nodes (routers); if every router performs a two-subgroup partition of its downstream receivers, we will get $(N + 1)$ subgroups for the whole session.

A. Regulation Tree Construction within AMCA

An active router executes the partitioning algorithm proposed in section II and will split the set of its downstream links into two subgroups noted H and L which contain respectively the set of links with higher and lower capacity. Initially, and according to our partitioning algorithm, H contains all the downstream links, L is empty and the router is in the "initial phase". A router exits the initial phase when a partitioning is performed and a replicator is chosen. When a router receives a CR packet indicating a relative RTT variation greater than the b parameter, then it goes through the LS structure to determine if other links have experienced a relative RTT variation greater than the a threshold. If so all those links are moved from the H set to the L set. Once the two partitions H and L are built, the active router selects one link among those belonging to H as the replicator. Only direct receivers can be elected to act as a replicator by a router. This is why we have to be careful that the H set must contain at least one link leading directly to a receiver. To make a router aware of the links with direct receivers, the LS structure (introduced in section III) contains a $flag$ field set to 1 if this link leads to one direct receiver and to 0 otherwise. Moreover in order to be able to evaluate the relative RTT variation experienced by every downstream link during the partitioning phase, the LS structure will contain the RTT, the RTT variation ($RTTvar$) and the period ($period$) reported by the last CR received on each link. For the replicator election, we use the following metric to evaluate the capacity of the candidate links (those leading directly to a receiver):

$$weight = \kappa\ RTTvar + (1 - \kappa)\ \frac{1}{RTT \times (lr - lo + 1)} \quad (9)$$

where κ is a weighting parameter to be chosen in $]0, 1[$. $RTTvar$ varies inversely with the available bandwidth on the considered link and the RTT gives an idea on the potential replicator distance to the active router. As for $(lr - lo + 1)$, it can be seen as an estimation of the loss rate experienced by the receiver located downstream from this link. Information on $RTTvar$, RTT, lr and lo are retrieved from the CRs (eventually from NACKs) received on every link.

Since we have limited the number of subgroups to only two, we just need to maintain the identity of one replicator link in addition to the H and L contents for every subgroup. In order to make a replicator aware of its election, a data packet header is extended to contain an additional field ($IsRep$). Once a replicator is designated by a router, the first data packet received is forwarded on all the downstream links with the $IsRep$ field set to 1 only for the replicator link. Since an elected link

leads directly to the replicator, only this latter will receive a data packet with this field set to 1. The reason behind sending this first data packet to all the downstream links is to make the replicator aware of its election. On the other hand this data packet, when forwarded on the L links, will have its $rate$ field set to half the source current rate. This is because the replicator will begin the replication from the next data packet but with a transmission rate set to half the source current rate. This is important for setting the timeouts (especially for requesting repairs) at the receivers side.

Subsequent data packets received from the source are forwarded only on the H links. A replicator will retransmit every data packet (with $IsRep$ set to 1) it receives with an appropriate rate to its upstream router. This latter will retransmit the data packets received from downstream (from a replicator) only on the L links. For the receivers feedback, an active router forwards one aggregated CR based on the CRs received from the L links to the replicator. Similarly, the CRs received from the H receivers are aggregated into one CR to be sent upstream toward the source. The rate adjustment performed at the replicator side is the same as an AMCA sender (section III-B). One difference consists in the fact that there is no slow-start and the replicator enters immediately in the congestion avoidance phase. Since its upstream router forwards feedback from the L receivers to it, the replicator would achieve a transmission rate that matches the slowest receiver among those located downstream the L links. The source will send data packet with a rate that matches the slowest receiver located downstream the H links.

We must mention that a replicator has to perform the cache of data packets to be retransmitted to its children. A replicator removes data packets from its cache as soon as it receives the corresponding acknowledgments (piggybacked on the CRs).

B. The Regulation Process at the Replicator's Side

A receiver, when it receives a data packet that indicates its selection as a replicator, it initializes a timer for each subsequent data packet it receives in order to schedule its retransmission to a time which corresponds to its regulation rate. For the data packet with sequence number i (see Fig. 5), a timer will be set depending on the desired reemission rate, to δ_i computed as follows:

$$\delta_i = \Delta + \delta_{i-1} - x_{i-1}$$

where x_{i-1} is the interval between the arrival of the $(i-1)$th and the ith data packets. x_{i-1} is inversely proportional to the source transmission rate. δ_{i-1} is the timeout value for the previous data packet. Δ is the reemission interval which is inversely proportional to the reemission rate. The replicator has to update the regulation timeout values every time the reemission rate changes.

C. Partitioning Dynamics

We argue that our partitioning algorithm converges rapidly so the initial partitioning is not disturbed by receivers changing their isolated rates. After the initial partitioning was performed, if any link experiences an RTT variation where $|\Delta\dot{\tau}| > b$ for a sufficiently long period, then a decision to move this receiver to the other subgroup could be taken. Such a decision could be taken on the reception of K subsequent CRs with $|\Delta\dot{\tau}| > b$. K is a parameter to be set appropriately to avoid oscillations while avoiding performances degradation.

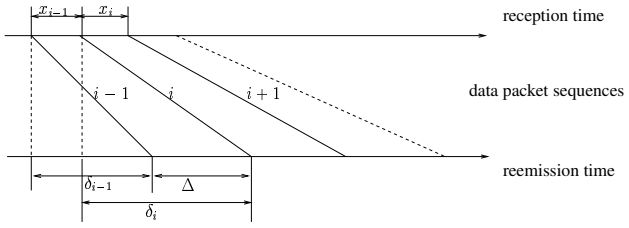

Fig. 5. setting the reemission timers for the ith data packet

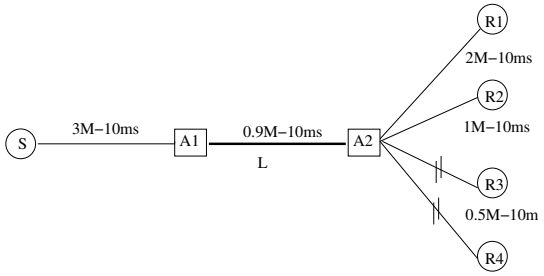

Fig. 6. The used topology

Moving a link from a subset to another does not require more than updating the L and H contents. The main concern is when the link leads to the replicator. If this link is the unique direct link in H, then a merge decision of the H and L subgroups can be taken. In this case, the router re-enters the initial phase and would eventually performs a new partitioning in the future depending on the network dynamics. Otherwise, the replicator is moved to L and a new replicator would be designated according to our capacity metric (9). Once done, the router will set the $isRep$ field to 1 when it forwards the subsequent data packet on the new replicator link. On the reception of this data packet, the old replicator knows that it is no longer the replicator but continues the replication of the standing packets. The new replicator, upon the reception of this data packet, would start its replication process as have been described.

V. Simulation Results

The AMCA congestion avoidance protocol has been extended to implement the regulation tree approach using ns-2.1b8 (network simulator [9]) and a set of simulations have been conducted. The original congestion parameters α, β, and N are respectively set to 1, 3 and 32. The partitioning parameters a, b and $\dot{\epsilon}$ are respectively set to 0.05, 0.2 and 0.25. The κ parameter is set to 0.5. For our simulations, we have adopted the topology of Fig. 6: one source S multicasts data packets to four receivers (with isolated rates $0.9Mbps$ for $R1$ and $R2$, $0.5Mbps$ for $R3$ and $R4$) through two active routers A_1 and A_2. The partitioning algorithm is enabled at the A_2 router in order to increase the satisfaction of the four receivers. The simulation is run for 100 seconds of the real system.

Fig. 7 shows the throughput achieved by the two subgroups $H = \{R1, R2\}$ and $L = \{R3, R4\}$ built by the active router A_2. We can see that both the H and L receivers obtain their isolated rates of $0.9Mbps$ and $0.5Mbps$. Fig. 8a shows the transmission rates of both the source and the chosen replicator

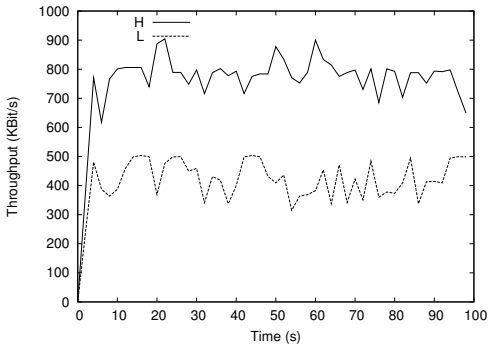

Fig. 7. The throughput achieved by the two subgroups with $r_1 = r_2 = 0.9$Mbps and $r_3 = r_4 = 0.5$Mbps.

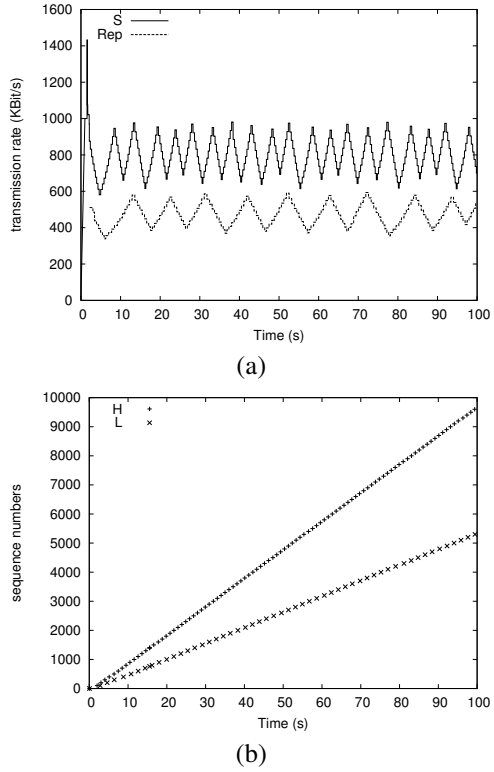

Fig. 8. (a) Transmission rate of the source and the replicator, (b) sequence numbers of data packets received by the H and the L receivers.

(here $R1$). We can see that the source achieves rapidly a transmission rate of $900Kbps$, that the partitioning is performed in less than 2 seconds, and that the replicator achieves approximately a transmission rate of $500Kbps$ which corresponds to the isolated rate of the L receivers. Fig. 8b shows the evolution of the data packet reception through their sequence numbers. At the beginning the two lines have the same slope. When the partitioning algorithm converges, one line keeps the same slope and the other achieves twice the original slope. Fig. 9 shows the same experimentation conducted with the same topology but with an isolated rate of $1Mbps$ for $R3$ and $R4$ and $2Mbps$ for the two others. Once again, the different receivers achieve a reception rate very close to their isolated rates.

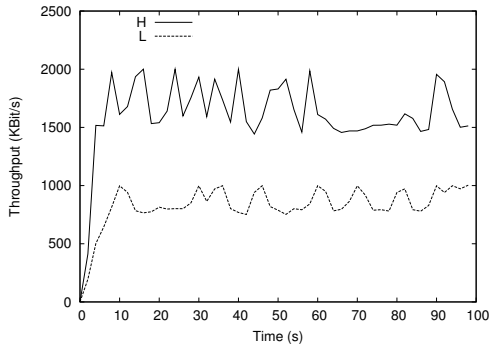

Fig. 9. The throughput achieved by the two subgroups with $r_1 = r_2 = 2$Mbps and $r_3 = r_4 = 1$Mbps.

VI. Conclusion

In order to accommodate receivers heterogeneity for multicast communications in the Internet, we investigated a new multi-rate congestion mechanism. In a distributed fashion, every router performs a partitioning of its downstream links into subgroups of similar capacities and designates for each of them a receiver (called replicator) to perform data replication with the appropriate rate. The adopted partitioning algorithm is executed on-the-fly depending on the RTT variation feedback received by the routers thus avoiding a prior estimation of the receivers' capacity. In addition to the construction of a regulation tree close to the physical multicast one, executing the partitioning algorithm at the routers instead of the source is more scalable.

To validate our approach, an extension of AMCA is proposed and evaluated with the DyRAM protocol. Some implementation issues have also been considered in the study. Preliminary simulation results show the rapid convergence of the partitioning process. As a future work, we plan to validate our approach with more complex topologies mainly to evaluate its dynamic behavior in the case of receivers changing their capacities over the time.

References

[1] M. Maimour and C. Pham, "Dynamic replier active reliable multicast (dyram)," in *Proc. of the 7th IEEE Symp. on Comp. and Comm. (ISCC 2002)*, July 2002.

[2] M. Maimour and C. Pham, "Amca, an active-based multicast congestion avoidance algorithm," in *the 8th IEEE Symposium on Computers and Communications (ISCC 2003)*, Kemer-Antalya, Turkey, July 2003.

[3] V. Jacobson, S. McCanne, and M. Vetterl, "Receiver-driven layered multicast," in *ACM SIGCOMM'96, Stanford, CA*, August 1996, pp. 117–130.

[4] X. Li, S. Paul, P. Pancha, and M.H. Ammar, "Layered video multicast with retransmission (lvrm): Evaluation of hierarchical rate control," in *INFOCOM'97*, 1997.

[5] L. Vicisano, L. Rizzo, and J. Crowcroft, "Tcp-like congestion control for layered multicast data transfer," in *Conference on Computer Communications (IEEE Infocom'98), San Fransisco, USA*, 1998, pp. 1–8.

[6] T. Jiang, M. Ammar, and E. Zegura, "On the use of destination set grouping to improve inter-receiver fairness for multicast abr sessions," in *IEEE INFOCOM'00*, March 2000.

[7] T. Jiang, M. Ammar, and E. Zegura, "Inter-receiver fairness: A novel performance measure for multicast ABR sessions," in *Measurement and Modeling of Computer Systems*, June 1998, pp. 202–211.

[8] Moufida Maimour and Cong-Duc Pham, "A rtt-based partitioning algorithm for a multi-rate reliable multicast protocol," in *6th IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC'03*, Estoril, Portugal, July 23-25 2003, in press.

[9] K. Fall, K. varadhan, and S. floyd, "Ns notes and documentation ucb/lbnl/vint. software and documentation available at http://www.isi.edi/nsnam/ns," July 1999.