A Grouping Scheme for Reliable Multicast Congestion Control

Harish Ramamurthy, Abhay Karandikar and Rahul Verma
Information Networks Laboratory
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India-400076
Phone no.: +91-22-576-7439
E-mail: {harish, karandi, rahulv}@ee.iitb.ac.in

Contacting author:

Abhay Karandikar

Address:

Information Networks Laboratory
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India-400076.

Phone no.: +91-22-576-7439

Fax no.: +91-22-572-3707

E-mail: karandi@ee.iitb.ac.in

Technical Areas:

Computer Communications
High-Speed Networks

A Grouping Scheme for Reliable Multicast Congestion Control

Harish Ramamurthy, Abhay Karandikar and Rahul Verma
Information Networks Laboratory
Department of Electrical Engineering
IIT-Bombay, Mumbai, India-400076.

Abstract— **Receiver heterogeneity severely constrains the performance of multicast congestion control schemes. In the typical TCP-like schemes employed for congestion control in reliable multicast, the performance is dictated by the slowest receiver. Intuitively, in the case where one bottleneck receiver is constraining the performance of other receivers, it would seem that isolating this receiver and carrying out a separate unicast conversation with it will improve the performance of the multicast transmission. In [1], we had formalized this notion of isolation and had proposed an algorithm to partition the receivers into disjoint groups based on their bottleneck bandwidth and RTT. The sender then carries out separate conversation with each sub-group formed. In our earlier formulation, an apriori knowledge of multicast tree was assumed and hence topology was not considered while forming groups. However, simulation results have shown that topology indeed plays an important role in the performance of grouping. In this paper, we extend our grouping formulation to include the effects of topology and formulate the problem of joint routing-grouping. The paper also discusses the various implementation issues like group formation, estimation of network parameters, of such grouping schemes. We present a scalable and efficient technique to estimate network parameters in a multicast scenario, through a receiver-driven hierarchical approach.**

Index Terms—**Multicast, Congestion control, Grouping, Multicast tree formation**

I. INTRODUCTION

MULTICAST congestion control is one of the fundamental issues in IP-multicast. Congestion control requires some feedback from the receivers or intermediate network elements so that the source can modify its data rate suitably. But due to the varying data transmission policies, it is not possible to come up with a "*one-size-fits-all*" multicast congestion control scheme. For example, the scheme will vary depending on whether the transmission is lossless (reliable data transfer e.g. file transfer) or whether it is lossy (e.g. video streaming). Receiver heterogeneity makes it even more difficult to come up with a scheme that suits all situations. The receivers can be heterogeneous in terms of their bottleneck bandwidths[1], round trip times, the quality of service (QoS) requirements, the packet error rates along their links, processing power etc.

The two primary approaches that have been studied in the literature for multicast congestion control are based on **single-rate** and **multi-rate** (layered) schemes. In single-rate schemes [2], [3], the source sends data to all the receivers at the same rate, the rate being determined by the bottleneck receiver[2]. This limits the scalability of such schemes because, in the presence of heterogeneous receivers, one single slow receiver can lead to low throughput and link utilization for the rest of the receivers. In multi-rate schemes [4], [5], data is transmitted at different rates (using, for example, a layered source coding) over multiple layers or streams. Depending upon their bandwidth capacities, the receivers can now subscribe to one or more layers. Though the multi-rate schemes address the problem of receiver heterogeneity effectively, but are suitable for applications like audio/video streaming only, *ie,* in scenarios where some packet loss can be tolerated. In other applications such as file transfer or software distribution where the data transfer has to be reliable, multi-rate schemes are not suitable.

Single-rate schemes are thus particularly important for scenarios where reliable data transfer is required. However, as mentioned earlier, receiver heterogeneity severely constrains the performance of such schemes. In [1], we have addressed this problem by considering a class of single-rate schemes that use multiple multicast groups rather than a single multicast group. We have termed such schemes as *single-rate, multiple groups* schemes. Currently in single-rate schemes, the sender transmits data to all the receivers in a particular multicast group at the same rate. However, suitably dividing the receivers (in a particular multicast group) into a set of disjoint groups and then carrying out separate (single-rate) conversations with each of the subgroups can help improve the overall performance of multicast transmission. For example, if one bottleneck receiver is constraining the performance of other receivers, intuitively it would seem that isolating this receiver and carrying out a separate unicast conversation with it will improve the performance of the multicast transmission. In a similar way, we can say that grouping together receivers that are *homogeneous* would help in improving the overall performance. This approach has been used by earlier authors [2], [3], [6] too, though in different contexts or different forms. For example, in [6] the authors use grouping for error control purposes. In this paper, we will consider grouping for congestion control purposes.

Traditionally, multicast communication is done using connectionless and unreliable data transfer protocols such as the User Datagram Protocol (UDP). However, various authors [2], [3] have recently proposed TCP-like schemes for multicast data

[1]By bottleneck bandwidth, we refer to the *minimum* bandwidth (along all links from the sender to the receiver) available to the receiver.

[2]By bottleneck receiver, we refer to the receiver that is slowest in terms of receiving data from the sender (and sending acknowledgments to the sender).

transfer and congestion control. Most of these schemes fall under the category of single-rate schemes. We refer to a TCP-like scheme as one that mimics TCP congestion control in some way. Multicast-extensions to TCP-like schemes can be used to achieve reliability in multicast transmissions in addition to providing end-to-end congestion control. In this paper, we will restrict our attention to reliable multicast scenarios, where multicast extensions to TCP-like schemes will be used for congestion control purposes. For such schemes, a grouping criterion that partitions the receivers based on their bottleneck bandwidths will improve the efficiency because all the receivers in a particular group can receive data at the same rate without adversely affecting the others. Further, as discussed in [7], TCP is unfair to receivers with different round trip times (RTT). The same problem will exist when dealing with any TCP-like scheme. Consider a scenario where two receivers (in the same multicast group) have the same bottleneck bandwidths but different RTTs. Since the acknowledgment from the receiver with greater RTT takes a longer time to reach the sender, the throughput of the other receiver is adversely affected. This gives the motivation behind grouping the receivers based on their RTTs.

In [1], we have addressed the problem of partitioning receivers into multiple groups based on their bottleneck bandwidths and RTTs. However, we had assumed that the multicast tree was known at the beginning. Thus, in a situation where there are two different topologies but with the same bottleneck bandwidth-RTT pairs, the scheme will result in the same grouping solution for both the topologies. This might not be a correct thing to do. As has been shown by simulations [8], [13], the topology indeed plays an important role in grouping.

In this paper, we study the effect of topology on the grouping scheme. We discuss the role of multicast tree formation on grouping and present the problem of joint routing-grouping. In this paper, we also look at some of the implementation issues of the grouping schemes like group formation, estimation of network parameters. The paper will not delve into exact details of the protocol messages exchanged by network elements, but will concentrate on how the network elements should co-ordinate with each other for grouping.

The rest of the paper is organized as follows. In Section II, we formulate the problem of joint routing and grouping that exploits topology. In Section III, we develop heuristics to solve the problem of joint routing and grouping. Section IV addresses some of the implementation issues faced by the grouping schemes. In Section V, we discuss issues like router support for multicast, which is required for the functioning of our scheme, and conclude the paper.

II. Multicast tree formation and grouping receivers

A multicast tree is a distribution tree used by the sender for sending data to a multicast group. Traditionally the multicast tree is formed by minimization of certain cost metric on the paths from the sender to the receiver. For example, DVMRP [9] uses hop count as the cost metric to construct the multicast tree. We argue that the multicast tree should be formed in conjunction with grouping. This will help improve the performance of receivers. Choosing the path for a receiver which will make it more similar to other receivers will improve the performance of multicast session. Consider the case when a receiver has two paths to the sender with different bottleneck bandwidths and RTT. Ideally, we shall choose the path which will further lower the cost of heterogeneity of receivers. The cost of heterogeneity, as defined in later sections, denotes the degree of *dissimilarity* among receivers in a multicast group.

Owing to different groups being formed, the sender has to send duplicate packets to each individual group. As a result, the congestion in the network might increase (especially at the source end) compared to the case when there is only one multicast group. Hence, a multicast tree formed in conjunction with grouping will improve the performance of multicast as the distribution tree will be set utilizing the common paths to receivers in the network. We now present the problem of joint routing-grouping.

A. Basic formulation

Let there be k receivers. The problem is to divide the receivers into n groups in such a way that each group is as *homogeneous* (we shall define homogeneity in the next sub-section) as possible. Note that on the one extreme, we can have as many groups as the number of receivers, *ie,* each group has exactly one receiver. This would mean that the sender carries a unicast conversation with each receiver. On the other extreme, we can have only one multicast group. As pointed out earlier, the performance in this case is dictated by the slowest receiver. This, therefore, suggests that we need to determine the best grouping that takes into account:

1) Cost due to heterogeneity of receivers in each group: For a particular grouping arrangement, say g ($g = \{g_1, g_2, \ldots, g_n\}$, where g_i is the i^{th} sub-group), let $C(g)$ denote the total cost due to heterogeneity of various receivers in each subgroup g_i of the grouping arrangement g.
2) Cost due to the number of groups: Let $C(n)$ denote the cost due to the number of groups.
3) Cost due to network routing: For a particular multicast tree T, the load on the network can be characterised by the cost $C(T)$.

The total cost of grouping $C_{tot}(n, g, T)$ can be defined as:

$$C_{tot}(n, g, T) = k_1 C(n) + k_2 C(g) + k_3 C(T) \qquad (1)$$

where k_1, k_2 and k_3 are some positive constants.

The problem of finding the optimal multicast grouping then reduces to finding a grouping arrangement g_{opt} with n_{opt} groups and T_{opt} tree that minimizes the total cost $C_{tot}(n, g, T)$. The solution to this problem depends on the determination and properties of the costs. We first attempt to capture these costs.

B. Cost due to heterogeneity of receivers

The heterogeneity of receivers can be quantified by using the grouping criteria. The grouping criteria provides a metric to characterise receivers, through network parameters. Network parameters help differentiate receivers and characterise the *similarity* of receivers. We now calculate the cost due to heterogeneity of receivers in terms of bottleneck bandwidth and RTT.

The cost due to heterogeneity can be similarly extended to other grouping criteria too. We begin with introducing the notion of homogeneity and heterogeneity of receivers.

1) Homogeneity in terms of the bottleneck bandwidths: Let us denote the bottleneck bandwidths of the receivers by b_1, b_2, ..., b_k and the round trip times by t_1, t_2, ..., t_k. Consider a group g_i. Let there be m_i receivers in the i^{th} group. Let their bottleneck bandwidths be $b_{i_1}, b_{i_2}, \ldots, b_{i_{m_i}}$. Let $d(b_i, b_j)$ denote the *distance* between the bottleneck bandwidths of receivers i and j. $d(.,.)$ is a non-negative real valued function such that

- $d(b_i, b_j) \geq 0$ for all b_i and b_j,
- $d(b_i, b_j) = 0$ if and only if $b_i = b_j$,
- $d(b_i, b_j) = d(b_j, b_i)$

We quantify homogeneity in terms of the distance function $d(b_i, b_j)$. We say that increasing (decreasing) homogeneity (heterogeneity) in each group is equivalent to decreasing the distance between various members in a group.

2) Homogeneity in terms of the round trip times: Similar to the distance function $d(b_i, b_j)$, we can have a function $l(t_i, t_j)$ that denotes the distance between RTTs of receivers i and j. The function $l(t_i, t_j)$ has similar properties as the function $d(b_i, b_j)$. Increasing RTT homogeneity in each group is equivalent to decreasing the distance $l(.,.)$ between various members in the group and vice-versa.

3) Receiver representation: We represent each receiver by its bottleneck bandwidth-RTT pair $\{b_i, t_i\}$. Thus, each receiver can be considered as a point on a two-dimensional plane whose X-axis and Y-axis are bandwidth and RTT respectively. Instead of denoting the receivers by the bottleneck bandwidth-RTT pair, we can also denote them by some transformation of the bottleneck bandwidth and/or RTT i.e. each receiver is denoted by $\{T_1(b_i), T_2(t_i)\}$ where T_1 and T_2 are some transformations (see Figure 1). As shown below, taking some suitable transformations may make the computation easier. The problem of grouping is then to partition the two-dimensional plane into disjoint grids that enclose points that are "close to each other".

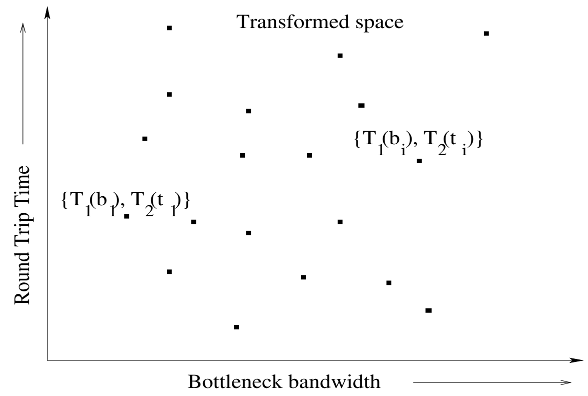

Fig. 1. Receivers as points on a 2-dimensional plane

We define

$$T_1(b_i) \equiv log_{10} \frac{b_i}{b_{min}} \tag{2}$$

where b_{min} denotes the minimum bottleneck bandwidth among all receivers. The above choice of $T_1(b_i)$ is based on the observation that the bottleneck bandwidths for two receivers can differ by several orders of magnitude (one receiver might have a bottleneck bandwidth of 10Kbps while another can have a bottleneck bandwidth of 1Gbps). Hence the function $log_{10} \frac{b_i}{b_{min}}$ is a convenient choice for $T_1(b_i)$ in such a scenario. We can define $T_2(t_i)$ in a similar fashion.

Note that the ranges of RTTs and bottleneck bandwidths may differ widely. For example, the RTTs may vary by a factor of 10 while the bottleneck bandwidths may vary by a factor of (say) 10^6. As a result the variations in RTTs may not be captured properly when the grouping is done. In order to circumvent this problem, we scale the RTT coordinates by a factor β where

$$\beta = \frac{log_{10} \frac{b_{max}}{b_{min}}}{log_{10} \frac{t_{max}}{t_{min}}} \tag{3}$$

Therefore we define

$$T_2(t_i) \equiv \beta * log_{10} \frac{t_i}{t_{min}} \tag{4}$$

Using the above definitions, the distance functions $d(b_i, b_j)$ and $l(t_i, t_j)$ now take the form

$$d(b_i, b_j) = \left\{ \begin{array}{ll} log_{10} \frac{b_i}{b_j} & \text{if } b_i \geq b_j \\ log_{10} \frac{b_j}{b_i} & \text{otherwise} \end{array} \right\} \tag{5}$$

$$l(t_i, t_j) = \left\{ \begin{array}{ll} \beta * log_{10} \frac{t_i}{t_j} & \text{if } t_i \geq t_j \\ \beta * log_{10} \frac{t_j}{t_i} & \text{otherwise} \end{array} \right\} \tag{6}$$

We define a function $f(d(b_i, b_j), l(t_i, t_j))$ that captures the distance between any two receivers i.e.

$$f(d(b_i, b_j), l(t_i, t_j)) \equiv \sqrt{A_{ij}^2 + B_{ij}^2} \tag{7}$$

where $A_{ij} = (T_1(b_i) - T_1(b_j))$ and $B_{ij} = (T_2(t_i) - T_2(t_j))$

$$f(d(b_i, b_j), l(t_i, t_j)) = \sqrt{|log_{10} \frac{b_i}{b_j}|^2 + |\beta * log_{10} \frac{t_i}{t_j}|^2} \tag{8}$$

$$f(d(b_i, b_j), l(t_i, t_j)) = \sqrt{d^2(b_i, b_j) + l^2(t_i, t_j)} \tag{9}$$

The cost due to heterogeneity of receivers increases as the distances $f(.,.)$ between receivers in a particular group increase. Intuitively, therefore, the partitioning of receivers into groups should be done in such a way that receivers "close to each other" in terms of distances are grouped together. Based on the above notion of distance between two receivers, we would also like to define the distance between two groups. A natural choice for the distance between any two groups is *that of the closest pair of receivers*, where only pairs consisting of one receiver from each group are considered.

C. Cost due to number of groups

As the number of groups increases:

- The number of multicast addresses to be used increases.
- There is greater overhead at the sender end which has to maintain additional states and perform signalling for each group.

The cost $C(n)$, thus should increase as the number of groups increases. Unfortunately, it is difficult to express this cost in a single figure of merit suitable for optimisation. Thus, we do not further take the issue of estimating the cost due to number of groups. Hence, we pose the joint routing-grouping problem as: Given that certain number of groups have to be formed, what is an optimal tree and grouping arrangement?

D. Cost due to network routing

As the sender carries out separate conversation with each group, it sends duplicate packets to the network, increasing the load on the network. We term such network load related costs as the cost due to network routing.

The multicast tree is formed by collating the set of paths from the sender to the receivers. Since at common paths no duplicate packets are sent, the multicast tree formed by utilising the maximum number of common paths will minimize the cost of network routing. The load on a network can be quantified in terms of the statistics of the arrival processes of the network queue. But due to the difficulty in quantifying the statistics, we will use a convenient but somewhat imperfect alternative to measure load at a link, in terms of average traffic carried by the link. We call the traffic carried by link (i, j) as the flow of the link, F_{ij}. We now formally state the costs.

Assuming a single sender and k receivers. Let
W = $\{g_i$, *ie,* set of sub-groups (refer Section II-A).$\}$
S_w = $\{$Set of receivers in w group.$\}$
P_s = $\{$Set of directed paths connecting the receiver s of group w from sender.$\}$
x_p = $\{$ A binary number to indicate whether a path p is used or not in the multicast tree. We call it the path flow variable. $\}$
c_w = $\{$Flow (data units/sec) of group w.$\}$

For a multicast tree, there should be a single path from the sender to every receiver hence the path flows must satisfy the following constraint:

$$\sum_{p \in P_s} x_p = 1, \forall s \in S \tag{10}$$

The flow on a link (i, j) is given by:

$$F_{ij} = \sum c_w x_p \tag{11}$$

The above summation is carried out for all paths p passing through link (i, j). In case there are more than one flows of the same group on a link, the network load due to the flow is taken into account once only. The cost due to network routing can be expressed as:

$$\sum_{(i,j)} D_{ij}(F_{ij}) \tag{12}$$

where D_{ij} is the function that relates the cost of using the link (i, j) with a flow of F_{ij} Here each D_{ij} is a monotonically increasing function. A frequently used formula in literature [10] is

$$D_{ij} = \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij} F_{ij} \tag{13}$$

where C_{ij} is the transmission capacity of link (i, j) measured in the same units as F_{ij} and d_{ij} is the processing and propagation delay. With this formula, the cost function becomes the average number of packets in the system based on the hypothesis that each queue behaves as an $M/M/1$ queue of packets.

III. Grouping: An optimization problem

Once the different costs are characterized, the joint routing grouping problem reduces to an optimization problem. The parameters to be optimized include the grouping arrangement or the set S_w for all $w \in W$ and the multicast tree, *ie,* the set of $\{x_p\}$. Since the cost due to the number of groups cannot be characterized properly, we assume that the number of groups to be formed is already known. For such fixed number of groups, we optimize the remaining cost by suitably changing the grouping arrangement and multicast tree. Optimizing the total grouping costs to determine the set of paths and grouping arrangement is a problem similar to that of finding steiner trees [11]. The steiner tree problem is a well known NP complete problem in graph theory. Heuristics are hence needed to solve the grouping problem. We now present the heuristics we have developed for solving the joint routing-grouping problem.

In our earlier formulation [1], we had partitioned the receivers into multiple groups based on their bottleneck bandwidths and RTT. However, we had assumed that the multicast tree was known in advance. We now extend our formulation to incorporate the effects of multicast tree formation too. As it is difficult to calculate the cost due to network congestion, the proposed heuristics will attempt to minimize the cost due to receiver heterogeneity only. The minimization of receiver heterogeneity can easily be accomplished by choosing a suitable multicast tree, as the bottleneck bandwidth and RTT for a receiver are characterized by the multicast tree.

We represent the receiver in a 3-dimensional plane of RTT, BBW and topology. The RTT and BBW planes are the same transformed planes as explained in Section III-B.3. The RTT and BBW planes were constructed for a specific tree arrangement. Now, we map the tree arrangement in the third dimension of topology. Consider a case where a receiver has two different paths (with different RTTs and BBW) to the sender. One tree arrangement will use one path and another may use the same or other path. Depending on the tree arrangement, the receiver will appear at different points in the 3d plane of RTT, BBW and topology. The plane of RTT and BBW illustrates the heterogeneity among receivers for a specific tree arrangement. Thus, cutting up a slice in the 3d plane of RTT, BBW and topology along the plane of RTT and BBW is equivalent to finding a multicast tree arrangement.

Once the multicast tree is known the grouping reduces to a problem of clustering receivers, if the number of groups to be formed is known. Several standard methods can be used to perform such clustering. We use the single-linkage [12] method to cluster receivers. The method proceeds in a hierarchical fashion, going from k groups each containing a single receiver to a single group containing k receivers. At each step, the groups with minimum distances[3] (defined as that of the closest pair of receivers) between them are merged into one larger group.

[3]The euclidean distance in the transformed space of RTT and BBW.

Finding an efficient grouping arrangement hence reduces to cutting a slice in 3d plane on BBW, RTT plane. This could be easily achieved using standard optimization techniques. Since the data set of parameters could be huge, we use a simulated annealing procedure to find the best tree arrangement.

We use the following parameters for the simulated annealing procedure:

- Temperature: We use an exponential cooling schedule of the form $T_{k+1} = \alpha T_k$. The initial temperature should be high enough to 'melt' the system completely and should be reduced to its 'freezing point' as the search for an optimum multicast tree progresses.
- System state: The system state is defined by the multicast tree arrangement, *ie.* a slice of topology in the 3d plane of RTT, BBW and topology.
- System energy: We denote the system energy by the heterogeneity of receivers in a specific group and heterogeneity of groups. To avoid situations where the multicast tree formed is of the least performance paths (i.e. of the lowest bandwidth paths), the system energy is adjusted with the mean performance of the receivers. The mean performance can be characterized in several ways. In our scheme, we use the mean of bottleneck bandwidths and RTTs of receivers to denote the mean performance of groups. These situations may indeed arise, as most of the receivers may have atleast one low performance path to the sender.

For our scheme, we start with a random tree arrangement. For this tree arrangement, the receivers are clustered into a fixed number of groups using the single linkage method of clustering points. The system energy is then computed by calculating the heterogeneity among receivers in a specific group and heterogeneity among different groups. The decision to accept or reject a solution is done using the standard probabilistic simulated annealing method. We then proceed by choosing another state (*ie.* another tree arrangement) and go on until we find a tree arrangement with system energy lower than the threshold of heterogeneity (which is set).

After having determined the best suitable multicast tree, we apply the single linkage method to cluster the receivers.

IV. Implementation issues: Estimation of Network parameters

One of the important issues concerning the proposed grouping scheme is the estimation of network parameters. Although several authors have proposed accurate techniques for the estimation of such network parameters in unicast scenarios, these schemes do not scale for multicast. The number of control messages exchanged is enormous making it non-scalable. Also, the estimation has to be carried out at different paths to the sender. The number of paths should be limited, for example by considering the n least hop paths to the sender, where n can be pre-decided.

We now present a hierarchical technique for efficient and scalable estimation of network parameters on one path. The basic idea is to avoid re-estimation of network parameters at common paths. We utilize the router support to achieve this.

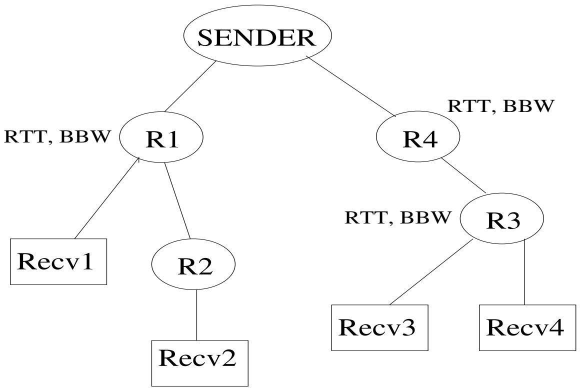

Fig. 2. Estimation of bandwidth and RTT

We now explain the scheme in context of our grouping scheme. Since our grouping scheme requires the estimation of bottleneck bandwidth and RTT, we explain the scheme for bottleneck bandwidth and RTT. Though we present the hierarchical scheme for bottleneck bandwidth and RTT, the scheme may be extended to estimate other network parameters too.

A. Hierarchical estimation of bottleneck bandwidth and RTT

The basic idea of the scheme is that the intermediate network elements store two states, namely the RTT[4] and bottleneck bandwidth (BBW)[5]. Until explicitly specified, in rest of the paper we shall use RTT and BBW to signify the RTT and BBW of the sender-receiver pair in consideration.

Consider the scenario as shown in Figure 2. Suppose receiver 1 (Recv1) wants to estimate its RTT and BBW. It sends a request packet (with contents that indicate that Recv1 wants to estimate the BBW and RTT) to the sender. When router R1 receives the request packet, it checks whether its RTT and BBW states for the sender are set or not. If the states are set (as is the case here), R1 sends two back to back packets with the information of RTT, BBW and the processing delay incurred at router R1, to Recv1. When these two packets arrive at Recv1, Recv1 notes the inter-arrival time between the two packets. This inter-arrival delay is then used to estimate the bandwidth up-to R1 using the packet pair technique[6]. BBW is then adjusted to the minimum of the two bandwidths (bottleneck bandwidth up-to R1 and bottleneck bandwidth between R1 and sender, which is sent by R1). RTT is computed by adding the RTT sent by R1 (i.e. RTT of the R1-sender path) and total delay occurred between the transmission and reception of packets at Recv1 and excluding the processing delay incurred at R1.

Now consider the case where Recv2 wants to estimate its RTT and BBW. Recv2 behaves similarly as in the above case (like Recv1) and sends a request packet to the sender. When router R2 receives this packet it checks whether the RTT and BBW states for the sender are set or not. As the BBW and RTT states are not set at R2, it sends a request packet to its

[4]RTT is defined as the two-way delay between source and destination.

[5]Bottleneck bandwidth is defined as the available bandwidth of link from source to destination.

[6]As two packets travel through a bottleneck they are spaced out in time and the spacing between packets is inversely proportional to the bottleneck bandwidth.

parent router (R1 here). R1 behaves similarly as in the above case (Recv1 estimation case) and sends two back to back packets with RTT, BBW and processing delay. When R2 receives these two packets (from R1) it estimates it's BBW and RTT and updates its states. After updating the BBW and RTT states, R2 sends two back to back packets with RTT, BBW and processing delay (incurred in estimating BBW and RTT from R2 to sender) to Recv2. Recv2 as explained in the case of Recv1, estimates the BBW and RTT similarly.

In cases where multiple receivers try to estimate the bandwidth and RTT (like at the start of session), only one request is forwarded to the parent. This helps in reducing a large number of redundant control messages that are exchanged, for example at the start of the session.

B. Salient Points

The proposed scheme being a hierarchical one, distributes the overhead of estimating RTT and BBW evenly over the network. The problem of feedback implosion is avoided, the intermediate routers and sender would at any time receive feedback only from their immediate children. In the proposed scheme, all calculations involved in the estimation of RTT and BBW are done at one point, avoiding the problem of clock synchronization. The RTT and BBW states set at the intermediate network elements are soft states that expire after periodic intervals of time. The expiry period or state timeout can be set according to application needs, *ie,* by considering how routinely the BBW and RTT need to be dynamically updated. But there exists a tradeoff in choosing an appropriate state timeout value. If the timeout is low, more number of control messages have to be sent to update the RTT and BBW, but the estimate will respond better to the network dynamics and vice versa.

In the conventional algorithms for RTT measurement, the receivers spend RTT amount of time to estimate the RTT. The same (*ie,* estimation time) applies for the schemes that estimate bandwidth using the packet pair technique. But in the proposed scheme, the time required to estimate RTT and bandwidth is always less than the RTT. In the best case, the time required would be equal to the delay between the receiver and its parent link and in the worst case, the time required would be RTT, when the request packet reaches goes upto the sender. Simulation studies of the scheme were carried out but owing to lack of space we present them in [13].

V. Discussion and Conclusions

In this section, we discuss various issues related to our proposed scheme. We also discuss some possible extensions that can bring further improvement in the overall performance of the scheme.

A. Multicast tree formation

The multicast tree formation has been an area of constant research. The inability to find an optimal multicast tree has led to the development of heuristics. Though several authors have proposed tree formation techniques by minimizing the cost functions of delay and hops, we propose to form the tree in conjunction with the grouping. In the present scheme, the heuristics that we have developed optimises the tree keeping grouping in consideration. We have ignored the cost of deployment in terms of network congestion. We are currently developing heuristics that take traffic congestion also into account. However, we would like to stress that even sub-optimal groupings (that might arise because of ignoring traffic congestion related costs) can lead to a substantial improvement in the performance as against without grouping. The preliminary simulation studies of the scheme have been carried out but owing to lack of space, we present them in [13]. The preliminary simulation results have show that there is a marked improvement in the performance of multicast session, in terms of receiver throughputs.

B. Router support

In the present grouping scheme, the estimation of network parameters requires the router support. We do recognize that such support from routers may not be justified, but they are essential to distribute the overhead of such estimations all over the network. The proposed scheme does not have any stringent requirements, as it just requires the routers to store a few finite number of states. Such simple services can be easily incorporated in routers using Generic Router Assist [14]. GRA is a functionality added to the routers to support simple filtering services that could be utilized for implementing a scalable and efficient multicast communication scheme.

References

[1] Rahul Verma, Abhay Karandikar and Saurabh Jain. "BRMG: Bandwidth and RTT based Multicast Grouping", ICACT'02, Feb 2002, Korea.

[2] Injong Rhee, Nallathambi Balaguru and George N. Rouskas, "MTCP: Scalable TCP-like Congestion Control for Reliable Multicast", IEEE INFOCOM'99, New York, March 1999.

[3] Luigi Rizzo, "pgmcc: a TCP-friendly single rate multicast congestion control scheme", SIGCOMM'00, Stockholm, Sweden, 2000.

[4] S. McCanne, V. Jacobson and M. Vetterli, "Receiver-driven Layered Multicast", ACM SIGCOMM'96, October 1996.

[5] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver, "FLIDDL: Congestion Control for Layered Multicast", in Proceedings of NGC 2000, pp. 71-81, November 2000.

[6] M. H. Ammar and L. R. Wu, "Improving the Throughput of Point-to-Multipoint ARQ Protocols through Destination Set Splitting.", in IEEE INFOCOM'92, pp.262-271, IEEE Communication Society, 1992.

[7] Bernard Suter, T. V. Lakshman, Dimitrios Stiliadis and Abhijit K. Chaudhary, "Buffer Management schemes for supporting TCP in gigabit routers with per-flow queuing", IEEE Journal on selected areas in communication, Vol. 17, no. 6, pp. 1159-1168, June 1999.

[8] Rahul Verma, "Issues in Best-Effort Delivery of Data for Unicast and Multicast Flows", Master Thesis (under Dual Degree programme), IIT Bombay. Available at: `http://www.ee.iitb.ac.in/uma/~harish/research`.

[9] T. Pusateri, Distance Vector Multicast Routing Protocol, Available at `http://potaroo.net/ietf/ids/draft-ietf-idmr-dvmrp-v3-10.txt`.

[10] S. Keshav, "An Engineering Approach to Computer Networking" Addison-Wesley, 1999.

[11] C. Diot, W. Dabbous and J. Crowcraft. "Multipoint communications: A survey of protocols, functions and mechanisms." IEEE Journal on Selected Areas in Communication, 15(3):277-290, April 1997.

[12] Brian S. Everitt, Sabine Landau and Morven Leese, "Cluster Analysis", Arnold Publishers, 2001.

[13] Harish Ramamurthy, "A Generalised Grouping Framework for Multicast Communication"., Technical Report, Available at: `http://www.ee.iitb.ac.in/uma/~harish/research`

[14] Generic Router Assist: Signaling Protocol Specification Available at `ftp://ftp-eng.cisco.com/speakman/draft-ietf-rmt-gra-signalling-00.txt`