Bandwidth and RTT based Multicast Grouping

Rahul Verma, Abhay Karandikar, Saurabh Jain
Department of Electrical Engineering, Indian Institute of Technology, Mumbai, India-400076
E-mail: {rahulv, karandi, saurabhj}@ee.iitb.ac.in

Abstract— **Receiver heterogeneity severely constrains the performance of multicast transmissions. Although several directions have been proposed to address the issue of heterogeneity, it is difficult to come up with a scheme which can be used for efficient data transfer in all scenarios. This paper studies the problem of efficient multicast in the presence of heterogeneous receivers. We propose a bandwidth and round trip time (RTT) based multicast grouping (BRMG) scheme where the receivers are partitioned into a set of disjoint groups and the sender carries a separate conversation with each of the groups. As discussed in the paper, such a grouping is ideal for single-rate multicast congestion control schemes. We treat the multicast partitioning as an optimization problem. The formulation can be easily extended to a variety of multicast scenarios. The paper also develops a simple algorithm to achieve a sub-optimal solution to the problem.**

Keywords—**Multicast, TCP, congestion control, homogeneity.**

I. Introduction

Many emerging applications in Internet such as multi-party video conference, distance learning, video-on-demand services, distributed databases, distribution of software, electronic news, distributed games etc. require the transmitted data to be received by multiple receivers. Multicasting provides an efficient way of transmitting data from a sender to a group of receivers. However, receiver heterogeneity severely constrains the performance of multicast transmissions. The receivers can be heterogeneous in terms of their bottleneck bandwidths, round trip times, the quality of service (QoS) they demand, the packet error rates along their links, processing power etc. Recent research ([1], [2], [3], [4], [5], [6]) in this area has addressed the problem of multicast transmissions to a set of heterogeneous receivers.

The design of a multicast communication scheme (for congestion control, scheduling, error recovery etc.) depends on a number of factors. For example, the scheme can vary depending on whether the transmission is lossless (reliable data transfer e.g. file transfer) or whether it is lossy (e.g. video streaming). Similarly, it can vary based on whether the data is generated on-the-fly or whether the data to be transmitted is stored data. Receiver heterogeneity makes it even more difficult to come up with a scheme which suits all situations. However, as discussed in the following sections, suitably dividing the receivers into a set of disjoint groups can help simplify a lot of problems. In addition to being scalable, such a division can be used for efficient data transfer in a wide range of multicast scenarios. In this paper, we propose a novel grouping scheme which partitions the groups based on the bottleneck bandwidths and the round trip times (RTTs). As discussed in the next section, such a scheme is ideal for single-rate multicast congestion control schemes. We treat the problem of multicast grouping as an optimization problem. The formulation can be easily modified to incorporate any other grouping criterion e.g. using packet error rate to divide the groups. The paper also presents a simple algorithm which can be used to achieve such a grouping.

The rest of the paper is organized as follows. Section II gives the motivation behind dividing groups based on their bottleneck bandwidths and round trip times. In Section III, we explain our proposed scheme (BRMG). In Section IV, we develop a simple algorithm for achieving a sub-optimal grouping and Section V concludes the paper.

II. Background and Motivation

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. The two primary approaches which have been studied for multicast congestion control are based on single-rate and multi-rate (layered) schemes. In single-rate schemes [7], the source sends data to all the receivers at the same rate and the rate is determined by the slowest receiver. One major disadvantage of such a scheme is that 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 ([5], [6]), data is transmitted at different rates (using, for example, a layered source coding) over multiple layers or streams. The receivers can subscribe to one or more layers based on their bandwidth capacity. These schemes effectively address the problem of receiver heterogeneity but are suitable only for unreliable data transmission like audio/video streaming where some loss can be incurred. In scenarios such as file transfer or software distribution where the data transfer has to be reliable, multi-rate schemes like Receiver-driven Layered Multicast (RLM) [5] are not suitable.

The problem of receiver heterogeneity in single-rate schemes can be solved to a large extent by appropriately grouping the receivers. This is particularly important in scenarios where reliable data transfer has to be achieved. Traditionally, multicast communication is done using connectionless and unreliable data transfer protocols such as the User Datagram Protocol (UDP). However, various authors ([7], [8], [9], [10]) have recently proposed TCP-friendly/TCP-like schemes for multicast data transfer and congestion control. One of the advantages that a TCP-like scheme offers is of reliable data transfer. (Note: In the rest of the paper, a TCP-like scheme refers to a scheme which is either TCP-based or uses some feedback from the receivers to modify the sender's transmission rate.) However, the main difficulty in employing a TCP-like scheme for reliable multicast transmissions is scalability. In a TCP-like scheme, all receivers have to send acknowledgements to the sender. But because of the potentially large number of receivers, the transmission of such acknowledgements from all the receivers to the sender is non-scalable and leads to wastage of network resources in terms of bandwidth and processing overhead at the sender and receivers. Further, as mentioned above, the presence of one slow receiver in the multicast group will result in low throughput for the other receivers also.

A good solution to overcome the problem of scalability and at the same time solve the problem of receiver heterogeneity is to group *similar* receivers based on some criterion. The sender then carries a separate conversation with each of these groups. This approach was initially studied in [1]. The authors use *packet error rate* as an approximate measure of similarity between the various receivers. In [10], the criterion of *packet loss rate* has been suggested for classifying the receivers. However, no algorithm has been outlined to classify the receivers.

Another issue that needs to be considered when grouping the receivers (with a TCP-like congestion control scheme) is of sending acknowledgments to the sender i.e. communication between the sender and various groups. In [7], the authors propose that such communication (sending acknowledgement) can be done by a group representative (*acker*) on behalf of all the receivers in the group. The *acker* is chosen as the receiver with the worst throughput among group members. However, the paper does not throw any light on the issue of forming various multicast groups.

A grouping criterion which partitions the receivers based on their bottleneck bandwidths will improve the efficiency of such schemes because all the receivers in a particular group can receive data at the same rate without adversely affecting the others. Further, as discussed in [11], TCP is unfair to receivers with different RTTs. The same problem will exist when dealing with any TCP-friendly scheme. Consider a scenario where two receivers (in the same multicast group) have the same bottleneck bandwidths but different RTTs. Since the acknowledgement from the receiver with greater RTT takes a longer time to reach the sender, throughput of the other receiver is adversely affected. This gives the motivation behind grouping the receivers based on their RTTs. Such a grouping criterion will be helpful in any scenario where the sender modifies its transmission rate (or window-size) based on acknowledgements from the receivers. We now explain our proposed bandwidth and round trip time based multicast grouping (BRMG) scheme.

III. Proposed Scheme (BRMG)

A. Basic formulation

We assume that the multicast tree is known at the beginning. Let there be k receivers. Assume that their bottleneck bandwidths and round trip times (defined as the two-way delay from the source to the receiver and back) are known. The assumption is not restrictive because these can be estimated using known techniques (discussed in a later subsection). Denote the bottleneck bandwidths by $b_1, b_2, \ldots, b_k$. Denote the round trip times by $t_1, t_2, \ldots, t_k$. The problem is to divide the receivers into n groups such that the receivers in a particular group are *homogeneous* (in terms of bottleneck bandwidths and round trip times). We quantify *homogeneity* formally in the next subsection.

The number of groups (n) can vary for a given number of receivers, k. On one extreme, there will be only one multicast group and on the other, each receiver will be placed in a separate group. We associate a *cost*, $C(n)$, with the grouping which increases as the number of groups, n increases. Thus, we have dual requirements:

1. To *increase* the *homogeneity* in the groups and
2. To *decrease* the *cost* due to the number of groups.

B. Notion of homogeneity

B.1 Homogeneity in terms of the bottleneck bandwidths

Consider a group g_i. Let there be m_i receivers in the group. Let their bottleneck bandwidths be $b_{i_1}, b_{i_2}, \ldots, b_{i_{m_i}}$. Let $d(b_j, b_k)$ denote the *distance* between the bottleneck bandwidths of receivers j and k. $d(.,.)$ is a non-negative real valued function such that

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

The function d(., .) can, for example, be of the form:

- $g(x,y) = |x-y|$
- $g(x,y) = (x-y)^2$
- $g(x,y) = x/y$ if $x > y$ and y/x otherwise
- $g(x,y) = \log(x/y)$ if $x > y$ and $\log(y/x)$ otherwise.

We quantify homogeneity in terms of the distance function $d(b_j, b_k)$. We say that increasing homogeneity in each group is equivalent to decreasing the distance between various members in a group. Define a *service differentiation factor*, $SDF(bw)$ (bw denotes differentiation in terms of bottleneck bandwidths):

$$SDF(bw) = \sum_{g_i \varepsilon G} \sum_{j,k \varepsilon g_i} d(b_j, b_k) \tag{1}$$

where $G = \{g_1, g_2, \ldots, g_n\}$, n is the total number of groups. Denote $\sum_{j,k\varepsilon g_i} d(b_j, b_k)$ by $SDF(bw)_i$ which is the service differentiation factor (bottleneck bandwidth) for group g_i.

B.2 Homogeneity in terms of the round trip times

Similar to the service differentiation factor $SDF(bw)$, we can define the service differentiation factor in terms of RTTs. We denote it by $SDF(rtt)$, that is,

$$SDF(rtt) = \sum_{g_i \varepsilon G} \sum_{j,k \varepsilon g_i} l(t_j, t_k) \tag{2}$$

where $l(t_j, t_k)$ is some measure of distance between RTTs of the receivers j and k. The distance function $l(.,.)$ has similar properties as the distance function $d(.,.)$. Denote $\sum_{j,k\varepsilon g_i} l(t_j, t_k)$ by $SDF(rtt)_i$ which is the service differentiation factor (RTT) for group g_i.

We define a *cumulative service differentiation factor*, SDF, which is function of $SDF(bw)_i$s and $SDF(rtt)_i$s, that is,

$$SDF = \sum_{g_i \varepsilon G} f(SDF(bw)_i, SDF(rtt)_i) \tag{3}$$

where the function $f(.,.)$ is directly proportional to $SDF(bw)_i$ and $SDF(rtt)_i$.

For a given number of groups (say n), there can be a number of ways ($\frac{1}{n!}\sum_{j=0}^{n}\binom{n}{j}(-1)^j(n-j)^k$) in which k receivers can be grouped. For each of these groupings we will have a corresponding SDF. Choose the grouping which gives the minimum SDF and denote the SDF as $SDF^*(n)$, that is, $SDF^*(n)$ denotes the minimum SDF when the number of groups is n.

Lemma 1: As the number of groups is increased, there always exists a grouping which has a lower (or same) SDF.

Proof: Consider n groups. Denote them by $g_1, g_2, \ldots, g_n$.
Case 1: There is atleast one group g_i for which any two receivers have different bottleneck bandwidths and/or RTTs.Then $SDF(bw)_i = \sum_{j,k\varepsilon g_i} d(b_j, b_k)$ and $SDF(rtt)_i = \sum_{j,k\varepsilon g_i} l(t_j, t_k)$. Assume that receivers r_1 and r_2 in group g_i have different bottleneck bandwidths and/or RTTs. Place the receiver r_2 (or r_1) in a separate group, g_{n+1}. The combined $SDF(bw)$ for the new group, g_{n+1} and the group g_i then becomes

$$SDF_{new}(bw)_i + SDF_{new}(bw)_{n+1} = SDF(bw)_i - \sum_{k\varepsilon g_i} d(b_{r_2}, b_k) \tag{4}$$

Now $\sum_{k\varepsilon g_i} d(b_{r_2}, b_k) \geq d(b_{r_2}, b_{r_1}) > 0$ because receivers r_1 and r_2 have different bottleneck bandwidths. Therefore the combined $SDF(bw)$ for groups g_{n+1} and g_i is less than the initial $SDF(bw)$ for group g_i. Similarly, if the RTTs of receivers r_1 and r_2 are different then the combined $SDF(rtt)$ for groups g_{n+1} and g_i will be lower than $SDF(rtt)_i$. Hence, SDF is lower when the number of groups is increased to $n+1$ as compared to the case when there are only n groups.
Case 2: There is no group g_i for which any two receivers have different bottleneck bandwidths and/or RTTs.
In this case, placing any receiver in a separate group results in the same $SDF(bw)$ and $SDF(rtt)$. So there exists a grouping with $n+1$ groups for which SDF remains the same.
From cases 1 and 2 we can infer that as the number of groups is increased, there always exists a grouping which has a lower (or same) SDF. ■

Theorem 1: $SDF^*(n)$ is a non-increasing function of n i.e. $SDF^*(n_2) \leq SDF^*(n_1)$ for $n_2 > n_1$.

Proof: Consider n groups. From Lemma 1, as the number of groups is increased to $n+1$, we can always find a grouping which has a lower (or same) SDF. This SDF will be greater than (or equal to) $SDF^*(n+1)$ (by definition of $SDF^*(n+1)$). Hence, $SDF^*(n) \geq SDF^*(n+1)$. This implies that for $n_2 > n_1$, $SDF^*(n_2) \leq SDF^*(n_2 - 1) \leq \ldots \leq SDF^*(n_1+1) \leq SDF^*(n_1)$. This completes the proof. ■

The above theorem is illustrated in Figure 1. For a given multicast tree, $SDF^*(n)$ is maximum when all receivers are in the same group i.e. $n = 1$. When all the receivers are placed in separate groups, each of the $SDF(bw)_i$s and $SDF(rtt)_i$s becomes zero and the $SDF^*(n)$ $(n = k)$ is minimum. Therefore as the number of groups increases, the groups become more homogeneous.

To incorporate another grouping criterion (e.g. packet error rate when considering a reliable error recovery scheme), we can introduce a similar service differentiation factor for packet error rate. The rest of the formulation remains the same.

C. Cost due to the number of groups

As the number of groups increases:

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

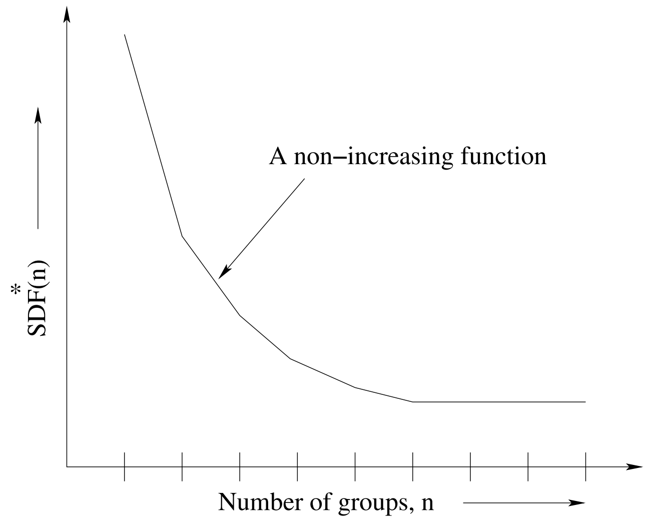

Figure 1. $SDF^*(n)$: A non-increasing function

- 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 multicast group.
  The *cost*, $C(n)$, thus increases (Figure 2) as the number of groups increases. The above factors can be captured by defining cost as:
  $$C(n) = \alpha n^{\beta} \tag{5}$$
  where n is the number of groups, α (a normalization factor) and β are certain (positive) constant parameters. The above definition of $C(n)$ is quite general and the parameters α and β can be chosen appropriately to suit the given scenario.

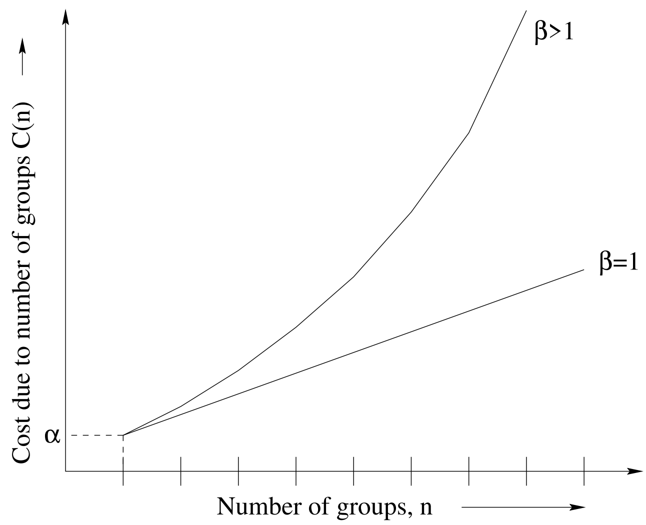

Figure 2. Possible choices for $C(n)$

D. Multicast grouping as an optimization problem

We pose the problem of grouping as an optimization problem. Let the total cost of grouping, $C_{tot}(n)$ be made up of two components:

1. Cost due to heterogeneity of receivers in various groups and
2. Cost due to the number of groups.

Let the cost due to heterogeneity of receivers in various groups be denoted by $C(SDF^*(n))$. We assume that this cost varies linearly as $SDF^*(n)$.

$$C(SDF^*(n)) = k * SDF^*(n) \tag{6}$$

where k is a positive constant. Note that as the number of groups increases, $C(SDF^*(n))$ decreases. Define $C_{tot}(n)$ as

$$C_{tot}(n) = h(C(SDF^*(n)), C(n)) \tag{7}$$

where $h(x,y)$ is of the form

$$h(x,y) = k_1 x + k_2 y. \tag{8}$$

k_1 and k_2 are some positive constants. The total cost of grouping can then be expressed as:

$$C_{tot}(n) = \alpha_1 SDF^*(n) + \alpha_2 n^{\beta} \tag{9}$$

α_1 and α_2 include the constants k, k_1, k_2 and α ($\alpha_1 = k_1 k$, $\alpha_2 = k_2 \alpha$). The objective function is then to find a grouping G with n_{opt} groups such that $C_{tot}(n)$ is minimized for $n = n_{opt}$.

E. Choice for various parameters and functions

We now discuss the issue of choosing appropriate parameters and functions in the above framework. The function $f(SDF(bw)_i, SDF(rtt)_i)$ can take the form $f(x,y) = p_1 x + p_2 y$. The parameters p_1 and p_2 determine the relative weights given to the homogeneity based on bottleneck bandwidth and RTT respectively. In a scenario where RTT does not play any role (e.g. CBR sources which do not require any acknowledgement) , we can choose $p_2 = 0$. Similarly, the parameters k_1 and k_2 (in function $h(.,.)$) determine the relative weights to be given to the two cost terms. Making k_1 (k_2) larger implies that homogeneity (number of groups) is the important factor in determining the grouping. For example, if the cost due to the number of groups carries a lower weight, a completely unicast scenario might turn out to be the optimum grouping.

F. Estimation of bottleneck bandwidths and RTTs

One important issue that needs to be addressed in the above formulation is that of estimating the bottleneck bandwidths and RTTs. One technique that can be used for estimating the bottleneck bandwidths is the *packet pair technique* ([12], [13]). The source sends two packets back-to-back, and they are separated at every intermediate router because of cross traffic. If the routers obey a round-robin-like scheme, the largest separation of the pair measures the bottleneck service rate for the source. The packet separation is reflected in the acknowledgment spacing, which can be measured at the source. Several algorithms ([14], [15]) have been proposed in literature for the estimation of the RTTs. The estimation of RTTs in a multicast environment has been discussed in [14]. The estimation is based on the assumption that there is some feedback (such as an acknowledgement) that comes back from the receiver to the source. The above schemes can be used for the estimation of bottleneck bandwidths and RTTs of the receivers before the actual transmission of data begins.

G. Static versus dynamic grouping

In the present version of the scheme, the grouping is static. Static grouping is justified when the network conditions do not change drastically. For example, if the average round trip times do not fluctuate much, the scheme will work efficiently. To achieve dynamic grouping, a periodic estimation of the bottleneck bandwidths and RTTs can be done and the receivers can be rearranged into different groups based on the updated network conditions. However, this poses the problem of synchronization of received data. We are currently working in this direction to extend the scheme for a dynamic grouping scenario.

IV. A SIMPLE SUB-OPTIMAL ALGORITHM FOR BRMG

The number of possible ways to arrange k receivers into n groups is approximately n^k. An exhaustive search algorithm that finds the optimal solution is impractical (especially for real-time applications). To simplify the problem of forming multicast groups, we visualize it in the following way:

Let each receiver be denoted by a bottleneck bandwidth-RTT pair $\{b_i, t_i\}$. (Note that two or more receivers can have the same bottleneck bandwidth-RTT pair.) Thus, each receiver can be considered as a point on a two-dimensional space whose x-axis and y-axis are bandwidth and time respectively (Figure 3). Instead of denoting the receivers by the bottleneack bandwidth-RTT pair, we can also denote them by some tranformation 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.

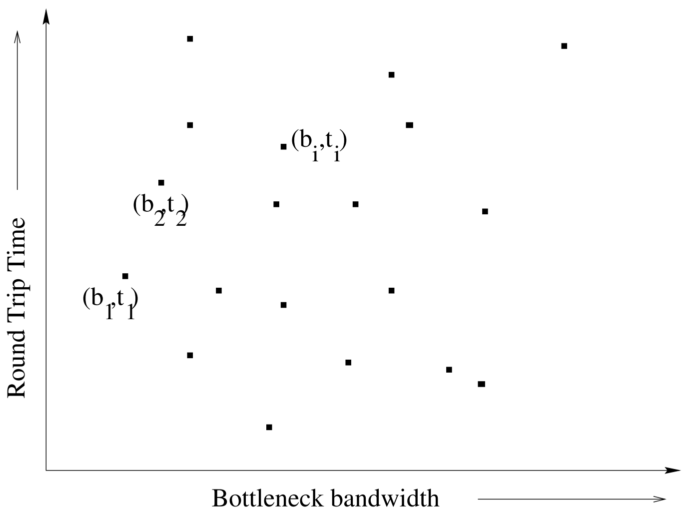

Figure 3. Receivers as points on a 2-dimensional plane

The service differentiation factors can be looked upon as the sums of distances between points (in the transformed space) in a particular group. The problem then is to divide these points (Figure 4) in such a way that the total cost, $C_{tot}(n)$ is minimized.

Next assume that we want to divide the receivers into n groups. We now describe a simple division criterion which gives a sub-optimal solution to the problem of dividing the receivers into a *maximum* of n groups. The division is performed in 2 steps:

1. Division based on bottleneck bandwidth:

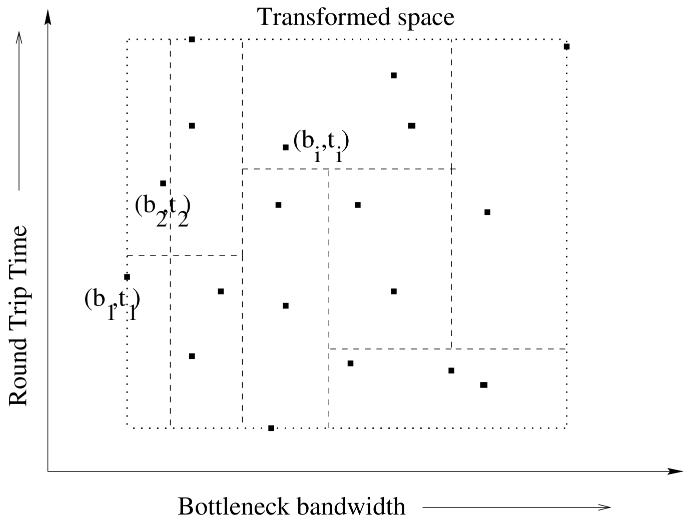

Figure 4. Division of receivers into a certain number of groups

Let $d_{ij} \equiv d(b_i, b_j)$. Define d_{ij} as

$$d_{ij} = \begin{array}{ll} log_{10}\frac{b_i}{b_j} & \text{if } b_i > b_j \\ log_{10}\frac{b_j}{b_i} & \text{otherwise} \end{array} \quad (10)$$

The above choice of d_{ij} 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 $log_{10}\frac{b_i}{b_j}$ is a suitable and convenient choice for d_{ij} in such a scenario. Let $G_1 = [\sqrt{n}]$ where [.] denotes the greatest integer function. We form a maximum of G_1 groups in the first step. Arrange the receivers such that $b_1 \le b_2 \le \ldots \le b_k$. Define D as

$$D = \frac{d_{1k}}{G_1} \quad (11)$$

d_{1k} is the maximum *distance* between the bottleneck bandwidths of any two receivers (Figure 5).

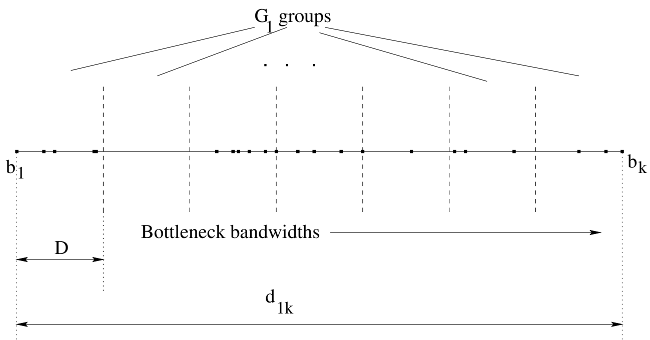

Figure 5. Division of receivers: Bottleneck bandwidth based (illustration of D and d_{1k})

Let b_i be such that $d_{1i} \le D$ and $d_{1(i+1)} > D$. Place the receivers $1, 2, \ldots, i$ in the first group. Next, let b_j be such that $d_{(i+1)j} \le D$ and $d_{(i+1)(j+1)} > D$. Place the receivers $i+1, i+2, \ldots, j$ in the second group and proceed similarly for the rest of the groups. To illustrate the above algorithm, Figure 5 modifies to the following figure (Figure 6).

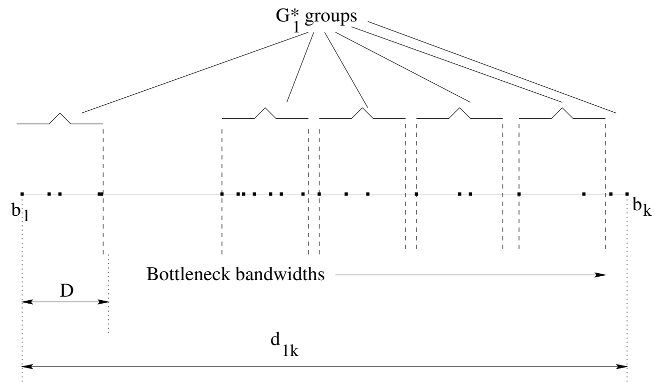

Figure 6. Division of receivers (Step 1): Bottleneck bandwidth based

Step 1 can thus be visualized as dividing the receiver space (Figure 3) into vertical stripes. After step 1, we will get G_1^* groups where $G_1^* \le G_1$. (Refer Figure 7 for the pseudo code.)

n: Maximum number of groups into which the receivers have to be divided.
Let $d_{ij} \equiv d(b_i, b_j)$
$G_1 = [\sqrt{n}]$
Arrange the receivers such that
$b_1 \le b_2 \le \ldots \le b_k$
d_{1k}: Maximum distance between the bottleneck bandwidths of any two receivers.
$d_{1k} = d(b_1, b_k)$
$D = \frac{d_{1k}}{G_1}$
$a = 1, b = 1, g = 1$
while $(b \le k)$ {
    while $(d_{ab} \le D$ and $b \le k)$ {
        $b = b + 1$
    }
    Place receivers $a, a+1, \ldots, b-1$ in group g
    $a = b$
    $g = g + 1$
}

Figure 7. Pseudo code: Division of receivers (bottleneck bandwidth based)

2. Division based on round trip time:
Let $G_2 = [\frac{n}{G_1^*}]$. In this step, we divide each group formed in step 1 into a maximum of G_2 groups. These divisions are based on RTTs and are done in a similar fashion as in step 1. The definition of $l(t_j, t_k)$ can be the same as that of $d(b_j, b_k)$, or some other distance metric (e.g. $l(t_j, t_k) = t_j/t_k$ if $t_j > t_k$ and t_k/t_j otherwise) can also be used. The total number of subgroups formed in each group will be $G_2^* \le G_2$ and thus the total number of groups formed will be less than (or equal to) $G_2 * G_1^*$ (which is $\le n$). This step can be visualized as dividing the receiver space (Figure 3) into horizontal stripes. Steps 1 and 2 can also be interchanged

i.e. the division based on RTTs can be done in the first step while the division based on bottleneck bandwidths can be done in the second step. Once such a grouping is done for various n, we can use $C_{tot}(n)$ to choose the best (least $C_{tot}(n)$) grouping.

V. Discussion and conclusions

We have proposed a bandwidth and round trip time based multicast grouping (BRMG) scheme. The scheme attempts to solve the problem of receiver heterogeneity in multicast networks and is particularly suited for single-rate multicast congestion control schemes. We are currently evaluating its performance by implementing it in a TCP-like congestion control environment. The simulation results will be reported separately. As mentioned earlier, the scheme can be useful in other scenarios too such as with CBR sources (division based on bottleneck bandwidth only). Our preliminary results with CBR sources show that in some scenarios the performance (in terms of packet loss rates and throughput) can improve drastically. In a heterogeneous environment, when there is only one multicast group, either the slow receivers suffer (high packet loss rates when the rate is high) or the fast receivers suffer (low throughput when the rate is low). Grouping the receivers results in a substantial improvement in loss rates while maintaining the same (high) throughput for individual receivers. This also means less duplicate packets from the source as compared to a completely unicast scenario.

The BRMG scheme provides an efficient way for reliable data transfer to a set of heterogeneous receivers. The problem of multicast grouping has been posed as an optimization problem. The paper also outlines a simple algorithm to achieve a sub-optimal solution to the problem. We are currently working towards extending the scheme for a dynamic grouping scenario.

References

[1] 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 Communications Society, 1992.

[2] Shun Yan Cheung and Mostafa H. Ammar, "Using Destination Set Grouping to Improve the Performance of Window-Controlled Multipoint Connections", Computer Communications Journal, Vol.19, pp723-736, 1996.

[3] V. Roca, "Packet Scheduling for Heterogeneous Multicast Transmissions", Protocols for High Speed Networks (PfHSN'99), August 1999.

[4] S. Bhattacharyya, J. Kurose, D. Towsley and R. Nagarajan, "Efficient Multicast Flow Control using Multiple Multicast Groups", Proceedings of IEEE INFOCOM'98, February 1998.

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

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

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

[8] L. Vicisano, L. Rizzo and J. Crowcroft, "TCP-like Congestion Control Scheme for Layered Multicast Data Transfer", IEEE INFOCOM'98, San Francisco, CA, Mar. 28-Apr. 1, 1998.

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

[10] V. Visoottivesh, T. Mogami, N. Demizu, Y. Kadobayashi and S. Yamaguchi, "M/TCP: The Multicast-extension to Transmission Control Protocol", ICACT'01, Muju, Korea, February 2001.

[11] Bernard Suter, T.V. Lakshman, Dimitrios Stiliadis and Abhijit K. Choudhary, "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.

[12] S. Keshav, "Packet-Pair Flow Control", IEEE/ACM Transactions on Networking, February 1995.

[13] S. Keshav, "An Engineering Approach to Computer Networking", Addison-Wesley, 2000.

[14] A. Basu and S. J. Golestani, "Estimation of Receiver Round Trip Times in Multicast Communications", RMRG Meeting, December 1998.

[15] P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", ACM Transactions on Computer Systems (TOCS), vol. 9, pp. 365–373, 1991.