Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 1421–1443
Computer Networks
www.elsevier.com/locate/comnet

Generalized multicast congestion control

Jiang Li [a], Murat Yuksel [b,*], Xingzhe Fan [c], Shivkumar Kalyanaraman [d]

[a] *Howard University, Department of Systems and Computer Science, 2300 6th Street NW, Washington, DC 20059, United States*
[b] *University of Nevada – Reno, Computer Science and Engineering Department, 171, Reno, NV 89557, United States*
[c] *University of Miami, Electrical and Computer Engineering Department, Room 411, McArthur Engineering Building, Coral Gables, FL 33124, United States*
[d] *Rensselaer Polytechnic Institute, Electrical Computer and Systems Engineering Department, 110 8th Street, Troy, NY 12180, United States*

Received 13 December 2005; received in revised form 24 July 2006; accepted 31 July 2006
Available online 30 August 2006

Responsible Editor: Nelson Fonseca

Abstract

Efficient multicast congestion control (MCC) is one of the critical components required to enable the IP multicast deployment over the Internet. Previously proposed MCC schemes can be categorized in two: single-rate or multi-rate. Single-rate schemes make all recipients get data at a common rate allowed by the slowest receiver, but are relatively simple. Multi-rate schemes allow of heterogeneous receive rates and thus provide better scalability, but rely heavily on frequent updates to group membership state in the routers. A recent work by Kwon and Byers, combined these two methods and provided a multi-rate scheme by means of single-rate schemes with relatively low complexity.

In this paper, we propose a new scheme called generalized multicast congestion control (GMCC). GMCC provides multi-rate features at low complexity by using a set of *independent* single-rate sub-sessions (a.k.a layers) as building blocks. The scheme is named GMCC because single-rate MCC is just one of its special cases. Unlike the earlier work by Kwon and Byers, GMCC does not have the drawback of *static* configuration of the source which may not match with the *dynamic* network situations. GMCC is *fully* adaptive in that (i) it does not statically set a particular range for the sending rates of layers, and (ii) it eliminates redundant layers when they are not needed. Receivers can subscribe to different subsets of the available layers and hence can always obtain different throughput. While no redundant layers are used, GMCC allows receivers to activate a new layer in case existing layers do not accommodate the needs of the actual receivers.

Keywords: Multicast; Congestion control; Multi-rate

1. Introduction

In multicast congestion control (MCC), satisfying demands of several heterogeneous receivers while maintaining scalable and efficient operation

[*] Corresponding author. Tel.: +1 775 784 6974; fax: +1 775 784 1877.

E-mail addresses: lij@scs.howard.edu (J. Li), yuksem@cse.unr.edu (M. Yuksel), fanx@miami.edu (X. Fan), shivkuma@ecse.rpi.edu (S. Kalyanaraman).

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.07.014

has been one of the major research problems. Researchers have developed various schemes that work effectively with different situations. *Single-rate* MCC schemes are simple and easy to deploy, but they only work well with small number of receivers or high number of receivers with less heterogeneity. In single-rate protocols such as ERMCC [2], PGMCC [3] and TFMCC [4], the source sends data to all receivers at a dynamically adjusted rate. The rate has to be adapted to the slowest receiver to avoid consistent congestion at any part of the multicast tree. Therefore, faster receivers suffer. Still, single-rate protocols have advantages because they are simple.

For cases where receivers are high in number or significantly different in their bandwidth and congestion circumstances, single-rate schemes do not scale. Hence, by adding more implementation complexity, *multi-rate* MCC schemes that are able to operate under a wider range of network conditions have been developed.

In multi-rate schemes, the source maintains several layers each with different transmission rate, and receivers subscribe to different subsets of these layers depending on their and network's bandwidth and congestion circumstances. In a multi-rate multicast session, each layer uses a separate multicast group address. In most multi-rate protocols, the sending rates in these layers are not fully adaptive. They are either static, such as in RLM [5] and PLM [6], or dynamic but are defined by a carefully designed pattern, such as in RLC [7], FLID-DL [8], FLGM [9], STAIR [10] and WEBRC [11]. Recipients have to increase or decrease their receiving rates by joining or leaving some groups.[1] To perform join and leave operations, they send IGMP messages to routers. Upon the receipt of these IGMP messages, routers update their multicast group states to allow traffic through (for join) or stop traffic forwarding (for leave), which allows adjusting throughput for receivers. To quickly react to congestion, these operations have to occur frequently. As a result, a large volume of control traffic is introduced into the network, and the routers are heavily loaded because all the rate control burden has been shifted to them. Moreover, according to IGMP [12], the join and leave operations (especially leave) need time to take effect. Therefore, the number of these operations is limited during a period and restricts the effectiveness of these multicast congestion control schemes. These schemes are also called receiver-driven schemes.

A concurrently proposed scheme SMCC [1] is a hybrid of single-rate and multi- rate multicast congestion control. It combines a single-rate scheme TFMCC [4] with the receiver-driven idea. In each layer, the source adjusts sending rate *within a certain limit* based on TFMCC, and receivers join or leave layers cumulatively according to their estimated maximum receiving rates using TCP throughput formula [13]. Since the flows in each layer are adaptive to network status, the number of join and leave operations are greatly reduced. The congestion control is more effective.

SMCC requires static configuration of the maximum sending rates for each layer. This requirement makes SMCC not capable of accommodating receivers with variant bandwidth circumstances. In the case when many or all of the receivers fall into the lowest layer, SMCC cannot provide new layers with smaller sending rates. Again, when many or all of the receivers subscribe to the very highest layer(s), then lower layers become redundant, thereby causing the scheme to spend extra effort to maintain those unnecessary layers.

In this paper we propose a new scheme GMCC that solves these problems while having the merits of SMCC. In the remainder of this section, we will briefly describe GMCC and summarize key contributions and properties of it. Then, in the rest of the paper, we will describe the details of GMCC, and show simulation results to demonstrate the performance of GMCC. In Section 3, we will provide a detailed explanation of GMCC components at the source and receivers. Finally, we will show our simulation-based performance evaluation of GMCC in Section 4, and conclude in Section 5.

1.1. Brief description of GMCC

The functions of the source and the receivers in GMCC can be decoupled into two main categories: intra-layer, and inter-layer. GMCC uses single-rate MCC to manage intra-layer activities at the source and the receivers. In particular, rate adaptation and congestion representative (CR) selection are totally left to the single-rate MCC scheme that is being used. Similarly, creation and management of feedback packets at the receivers are done by the

[1] Joining a layer is also called subscription, leaving a layer is also called unsubscription. In this paper we will use both sets of terms ex-changeably.

single-rate MCC scheme. Though we are using ERMCC [2] in this paper, GMCC allows usage of other single-rate MCC schemes. Because GMCC performs intra-layer functions by using a single-rate MCC as a building block, we will focus on inter-layer functions which are the main contributions of GMCC.

GMCC performs layer join and leave operations at receivers (see Fig. 4) by using statistical measures such as throughput attenuation factor (TAF). Similar to all earlier multi-layer schemes, GMCC allows only a predefined order of joining the layers, i.e. a receiver can join layers 1, 2 and 3 in sequence, but cannot join layers 1 and 3 without joining the layer 2 in between. Unsatisfied receivers join a new layer if they detect that they are significantly less congested than the CR for their highest layers. In particular, for a receiver *i* having a highest layer *j*, the receiver *i* joins a new layer if its TAF is significantly smaller than the TAF of the CR for layer *j*. GMCC does not allow join attempts and join decisions are made purely by the receiver thereby simplifying the operations significantly. Once the receiver joins a new layer it will start participating in the CR selection process for its new highest layer and maybe will be selected as the new CR. When a receiver in GMCC is selected as the CR of a layer, it checks whether or not it is the CR of its highest two layers. If so, then that receiver unsubscribes from its highest layer.

In order to dynamically adjust the number of layers GMCC performs *layer control* by activating or shutting down layers without setting a particular sending rate range for individual layers. In order to implement the layer control, GMCC leverages the fact that CR of each individual layer sends feedback packets regularly to help the source adapt the layer sending rates. In layer control, two operations can happen: (i) *activation* of a previously empty layer, and (ii) *deactivation* of a layer.

The activation operation happens only when a receiver joins a layer which did not have any receiver before. From regular CR statistics conveyed by the source, the newly joining receiver realizes that there is no CR for this layer and starts sending feedback thinking it is the CR of the layer. The source, then, figures out that there is a new receiver for the layer and activates the layer.

The deactivation operation takes place when the last receiver leaves the layer. Since it is the last receiver in the layer, it must be the CR of the layer. CR of each layer regularly sends feedback to the source for rate adaptation. Once the last receiver and the CR leaves a layer, the source will not receive these feedback packets. It will time out and ask receivers in the layer to elect a new CR, which will not occur since no other receiver is left in the layer. In that case the source will time out for the whole layer and stop sending the data packets thereby shutting down the layer.

1.1.1. Motivating example scenarios for GMCC

While earlier schemes like SMCC [1] fix the sending rate ranges of layers as well as the number of layers, GMCC provides the flexibility of varying them. This characteristic of GMCC is very useful for data streaming applications over highly heterogenous set of receivers, e.g. multicasting multimedia content to very large number of users located at different parts of the Internet. Applications over networks highly dynamic and heterogeneous congestion and end-to-end available capacity fit directly to the goals of GMCC, where adaptivity of sending rates is crucial.

We believe that GMCC-like schemes will be key to realizing Internet-wide large-scale multicasting applications such as Internet Protocol Television (IPTV) [14,15] and Akamai's global content delivery [16,17]. Any data streaming application with time constraints (e.g. real-time environmental data collection from one sensor to multiple sites) will be able to use GMCC. Though GMCC is not designed only for multimedia streaming, it is surely a crucial target application. While current multimedia streaming applications may have limits that range from a few 10 s of Kb/s (e.g. iPods [18] on 3G wireless channels) to a few 10 s of Mbps (e.g. HDTV [19] subscribers of IPTV), future applications may demand a larger dynamic range.

Fig. 1 shows the difference between SMCC and GMCC visually. For example, in SMCC, the lowest/base layer (i.e. Layer 1) is set to 1 Mbps maximum, the second layer is set to 2 Mbps maximum, and the *i*th layer is set to 2^{i-1} Mbps where *i* starts from 1. A receiver with an estimate of maximum throughput rate as 3 Mbps needs to join the lowest two layers. This static setting can negatively affect the performance of SMCC. With the settings above, consider the following scenarios:

- *Scenario 1*: All receivers have their estimated throughput rate below 1 Mbps. Some receivers are behind 0.1 Mbps bottlenecks, some behind 0.3 Mbps, and others behind 0.8 Mbps.

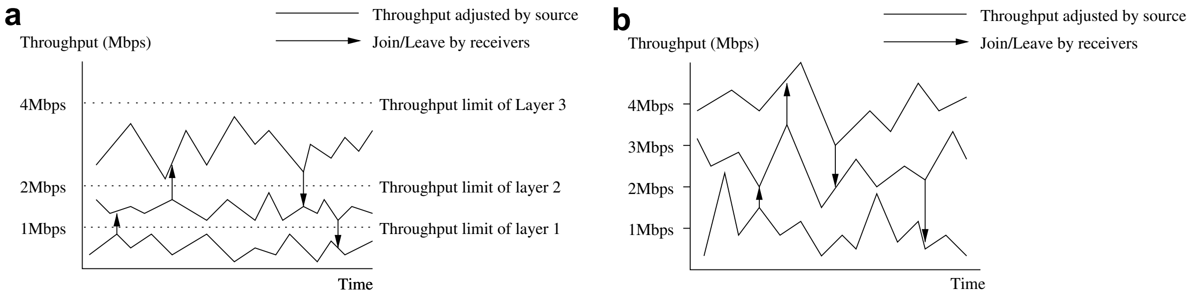


Fig. 1. Qualitative comparison of SMCC and GMCC: SMCC fixes the sending rate ranges of individual layers while GMCC allows flexibility in the number and sending rates of the layers. (a) SMCC overview (with per-layer throughput limit), (b) GMCC overview (no per-layer throughput limit).

- *Scenario 2*: All receivers have 100 Mbps bandwidth.

In Scenario 1, since there is only one layer for bandwidth less and equal to 1 Mbps, all these receivers have to receive at a single-rate. That means SMCC is degraded to a single-rate scheme under this situation. In Scenario 2, to fully utilize their bandwidths, the receivers will need to join eight layers in SMCC. Obviously, if the source is configured properly, only one layer is need. Therefore, seven layers are redundant due to the misconfiguration.

In the Scenario 1 above, receivers will be able to receive data at 0.1 Mbps, 0.3 Mbps and 0.8 Mbps, respectively, in GMCC. However, SMCC will only allow one layer and leave the receivers behind the 0.3 Mbps and 0.8 Mbps bottlenecks unsatisfied with a common transmission rate of 0.1 Mbps. Fig. 2a illustrates the scenario. In the second example Scenario 2, only one layer will be used in GMCC. However, as shown in Fig. 2b, SMCC will require receivers to join all the layers from 1 to 8 to satisfy the receivers with 100 Mbps bandwidth. This means that SMCC will require seven redundant layers.

Notice that when performing inter-layer operations (layer join/leave, layer control), GMCC uses the intra-layer CR selection process. This still does not make GMCC dependent on the particular single-rate MCC scheme being used, since all viable single-rate MCC schemes have one type of mechanism to track the slowest receiver. So, GMCC can leverage whatever the CR selection (or slowest receiver tracking) mechanism the underlying single-rate uses.

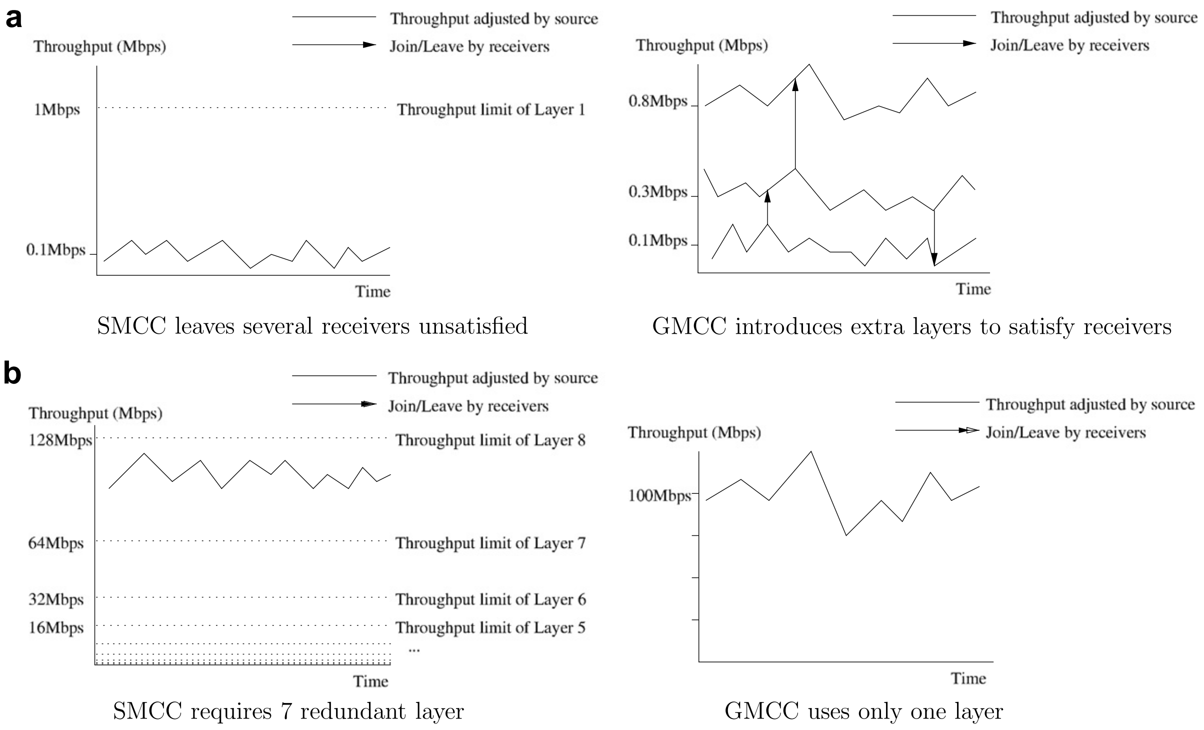
1.2. Key contributions

Major research problems in multi-rate MCC include *intra-layer* issues such as (i) proper rate adaptation of each layer, and (ii) selection and tracking of a representative receiver (i.e. slowest receiver); as well as *inter-layer* issues such as (a) minimizing the number of receiver join/leave operations to reduce the control traffic, and (b) accommodation of requirements and circumstances of numerous heterogeneous receivers with different bandwidth and congestion.

GMCC solves inter-layer management problems by introducing novel techniques to efficiently adapt the number and the sending rates of layers according to dynamic network situations. These novel techniques include (i) a highly sensitive statistical measurement of congestion, throughput attenuation factor (TAF), and (ii) a way to discover inter-layer rate allocation sub-optimality, probabilistic inter-layer bandwidth switching (PIBS). By means of these novel methods, GMCC successfully adapts the number of layers thereby eliminating redundant layers if they exist while not imposing any particular range for sending rates of individual layers. Furthermore, GMCC performs these inter-layer management operations while not hurting intra-layer management functionalities. So, GMCC decouples intra-layer operations from inter-layer operations completely and thereby allows any single-rate MCC scheme for individual layers not just the one used in this paper, i.e. ERMCC [2].

In brief, GMCC has the following advantages:

1. It is *fully* adaptive. The sending rate in each layer can be adjusted without rigid limits. Together with the automatically adjusted number of layers, it always allows heterogeneous receivers to receive at different rates.
2. The number of layers used is just enough to accommodate the differences among the throughput desired by receivers. No redundant layers are used.



SMCC leaves several receivers unsatisfied

GMCC introduces extra layers to satisfy receivers

SMCC requires 7 redundant layer

GMCC uses only one layer

Fig. 2. Qualitative comparison of SMCC and GMCC in two different scenarios. (a) Scenario 1: receivers are behind 0.1 Mbps, 0.3 Mbps, and 0.8 Mbps bottlenecks. (b) Scenario 2: all receivers have 100 Mbps bandwidth.

3. The source can control the overall throughput of a multicast session by limiting the number of layers to be used. In particular, if only one layer is allowed, GMCC works as a single-rate multicast congestion control scheme, which is the reason it is so named.
4. It is not coupled with equation-based rate control mechanism such as TFMCC. The rate control mechanism at source can be replaced by others based on representative (the most congested receiver).
5. It performs inter-layer control by
   (a) starting and stopping traffic within layers depending on whether there are receivers in the layers, and
   (b) Probabilistic inter-layer bandwidth shifting (PIBS) to discover suboptimal rate allocations to layers.

2. Related work

In multicast [20], the congestion control issue is complicated because we need to consider the congestion on a tree instead of that along a path. Intensive research has been conducted in this area, and researchers have proposed two categories of multicast congestion control protocols: single-rate and multi-rate.

2.1. Single-rate schemes

DeLucia and Obraczka's work in [21] is an early single-rate multicast congestion control scheme using representatives. It requires two types of feedback from receivers, congestion clear (CC) and congestion indication (CI). A fixed number of receiver representatives are maintained at the source. Whenever a CI is received by the source, if the sender of this CI is in the representative set, the representative is refreshed; if not, the sender will replace the representative that has not been refreshed for the longest time. Feedback from representatives is echoed by the source to suppress feedback scheduled at non-representative receivers. The source uses only the feedback from representatives to do MIMD (multiplicative increase and multiplicative decrease) rate adaptation.

The representative selection mechanism in that scheme is “simplistic” [21], but there is certain complexity involved in generating CC. The representative set is not guaranteed to include the slowest receiver, which means that the slowest receiver can be overloaded. Furthermore, it assumes that only a few bottlenecks cause most of the congestion. Based on this assumption, receiver suppression is the only mechanism for filtering feedback from receivers. In a heterogeneous network, where there may be many different bottlenecks and asynchronous congestion, the assumption may not be true. Consequently, the transmission rate may be reduced more than necessarily and stay very low or close to zero. This is known as the *drop-to-zero* problem.

PGMCC [3], TFMCC [4] and MDP-CC [22] are recent work also using representatives. Although they use different policies for rate adaptation, they all leverage the TCP throughput formula [13,23] for allocating the slowest receiver, i.e. the receiver with the lowest estimate TCP throughput according to the formula. Therefore, it is necessary for them to measure packet loss rate and RTT for all receivers.

PGMCC [3] keeps one representative as *acker*. The acker sends ACKs to the source which mimics the behavior of TCP. At the same time, NAKs with loss rate are sent from all other receivers. The PGMCC source measures RTT between itself and all receivers in terms of packet numbers, and compare the estimated throughput for updating acker. Due to the necessity of RTT measurement for all receivers, feedback suppression may have serious effect on PGMCCs performance. In fact, PGMCC does not provide a feedback suppression mechanism.

TFMCC [4] adjusts the rate according to the estimated rate calculated by the representative. RTTs are measured by receivers with a somewhat complex procedure. The sender needs to echo receiver’s feedback according to some priority order, and there is one-way delay RTT adjustment plus sender-side RTT measurement. TFMCC comes with feedback suppression which is an enhanced version of [24] and is probabilistic timer-based. Therefore, the total number of feedbacks is the function of the estimated total number of receivers, and additional delay is introduced into feedback.

MDP-CC [22] increases/decreases the transmission rate exponentially toward the target rate. Similar to TFMCC, the target rate is also calculated by the representative. In contrast to PGMCC and TFMCC, MDP-CC maintains a pool of representative candidates for representative update. As shown in that paper, maintaining multiple representative candidates requires much effort. MDP-CC can use probabilistic timer-based feedback suppression which has the same properties as that of TFMCC.

LE-SBCC [25] only requires single bit NAKs from receivers, and the source has three cascaded filters to filter receiver feedback before using it for rate adaptation. The computation complexity at the source is O(1). However, for n receivers, it needs O(n) states at the source, and network aggregation can also lead to performance degradation.

2.2. Multi-rate schemes

In multi-rate multicast congestion control, receivers may obtain different throughput rates. Ideally, data can reach each receiver at the rate that matches the condition of the path between the sender and the particular receiver. To realize such effects, it is a commonly accepted approach to use multiple simultaneous multicast groups (known as “layers”) for transmission. Based on some kind of metrics, each receiver independently and dynamically joins and leaves these layers during the course of a session. As the result, the sum throughput of joined layers as the session throughput varies from receiver to receiver.

In early multi-rate schemes, such as RLM [5], RLC [7], PLM [6], RLS [26] and MSC [27], the transmission rates of layers are fixed. Receivers join layers accumulatively, and leave in the reverse order. The throughput adaptation therefore totally depends on receivers’ join and leave actions. Some following multi-rate schemes, such as FLID-DL [8], Fine-grained layered multicast [9] and STAIR [10], still use fixed layer sending rates, but are more careful about the join and leave operations by receivers. By carefully designating the sending rates of layers and the order of join and leave, receivers emulate the increase and decrease of throughput more smoothly and thus achieve better performance.

Other more recent multi-rate schemes, such as MLDA [28], HALM [29], SMCC [1], LMMC [30] as well as our proposed scheme GMCC, allow the source to adjust the layer sending rates as well. The capability of adjustment at both sides (sender and receiver) allows for more adaptability to the network condition and yields better results.

Obviously, multi-rate schemes are more suitable for heterogeneous environments where they can utilize bandwidth more efficiently. Multi-rate schemes can potentially provide significantly higher scalability, in the price of design complexities such as close coupling with IGMP and aggregated multicast tree pruning [31]. Though, our proposed scheme *GMCC is purely end-to-end* and free of such design complexities.

3. Generalized multicast congestion control

The goal of multi-rate multicast congestion control is to fully utilize the available bandwidth on different paths between the source and receivers. One key issue is then how and when a receiver joins or leaves a layer to increase or decrease its total throughput rate. The second issue is how the source controls the throughput in each layer. The basic ideas of GMCC solutions to these issues are the following:

The goal of multi-rate multicast congestion control is to fully utilize the available bandwidth on different paths between the source and receivers. One key issue is then how and when a receiver joins or leaves a layer to increase or decrease its total throughput rate. The second issue is how the source controls the throughput in each layer. The basic ideas of GMCC solutions to these issues are the following:

- In each layer, the source chooses a most congested receiver as *congestion representative* (CR) and adjusts the sending rate of this layer according to the CRs feedback (Section 3.1.1).
- The source starts traffic in a layer when the first receiver joins and stops traffic in a layer when the last receiver leaves (Section 3.1.3).
- Each receiver joins layers cumulatively, and is allowed to be the CR of at most one layer.
- When a receiver detects that it is much less congested than the most congested receiver (i.e. the CR) in the highest layer it has joined, meaning it can potentially receive at a higher rate, it joins an additional layer *successively* (Section 3.2.2).
- When a receiver detects that it is the most congested receiver in more than one layer, which means it confines or can potentially confine the sending rates of more than one layer, it leaves the highest joined layer (Section 3.2.3).
- Receivers make decisions of join and leave based on statistics. Statistics can be used only if (1) At least a certain number of samples have been collected, and (2) Every layer has a CR.

As shown in the above ideas, it is important for a receiver to detect whether it is more congested than another. We propose to use throughput attenuation factor (TAF) for this purpose described in Section 3.2.1.

In the following subsections, we will describe major operations and components of GMCC at the source and the receivers. GMCCs functionalities are decoupled into intra-layer and inter-layer categories, and therefore we will present them in that manner.

3.1. GMCC source

As can be seen from Fig. 3, GMCCs source operations are composed of intra-layer activities like CR selection, rate adaptation, and data packet generation and handling; as well as inter-layer activities such as layer control, probabilistic inter-layer bandwidth shifting (PIBS), and maintenance of necessary layer statistics.

For *CR Selection*, the GMCC source participates in the messaging and maintains the actual list of

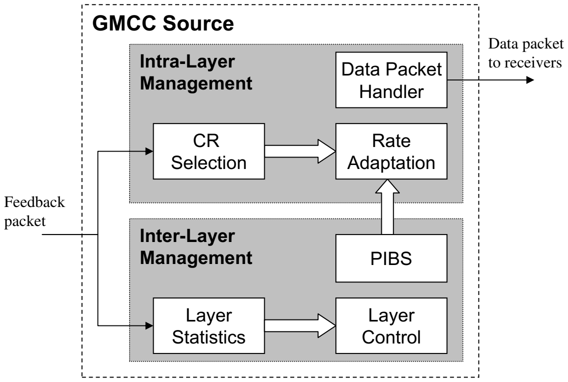


Fig. 3. Block diagram of major operations at the source in GMCC.

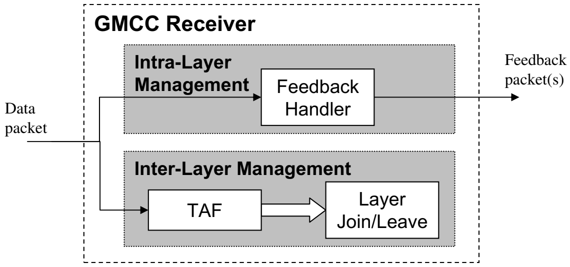


Fig. 4. Block diagram of major operations at a receiver in GMCC.

CRs pertaining to each layer. Depending on the single- rate MCC scheme being used, the CR selection mechanism can differ. However, in order for interlayer operations to work, GMCC requires the source to maintain the list of current CRs. The source piggy-backs the data packets for various control information to be conveyed to the receivers. *Data Packet Handler*'s main job is to piggy-back these intra-layer information on the multicast session's data packets.

Another per-layer information that needs to be stored at the source is the list of necessary *layer statistics* needed to decide about layer control operations. Such statistics include throughput attenuation factor (TAF) and its standard deviation for each layer. TAF values are measured at the receivers and the CR of each layer is expected to feed that value back to the source. We will describe measurement of TAFs in more detail later in Section 3.2.1.

3.1.1. Rate adaptation (intra-layer)

Given a layer with active receivers, the source chooses a most congested receiver (e.g. Receiver 2 in Fig. 7) in this layer as congestion representative (CR) and uses its feedback for rate adaptation.[2] When the CR detects packet loss, it sends feedback packets called congestion indications (CIs) back to the source that decreases the sending rate by half. To avoid reducing rate too much, we use the SMCCs method of smoothing RTT measurements, and the source decreases the sending rate at most once per SRTT (smoothed RTT). The samples of RTT are collected by the source at the receipt of CIs. The value of a sample is the time difference between the CI arrival and the departure of the data packet triggering the CI. As in SMCC, SRTT is calculated by exponential weighted moving average formula: SRTT = $(1 - \varepsilon)$ SRTT + ε RTT ($0 < \varepsilon < 1$, we use 0.125). At the absence of CIs, the sending rate is increased by s/SRTT each SRTT, where s is the packet size.

Notice that, even though SRTT method is the same as SMCCs, our rate adaptation methodology is profoundly different than SMCCs. In particular, in GMCC, there is no limit to the maximum or minimum sending rate of each layer as in SMCC. The sending rate in each layer can be increased or decreased to any level required for adaptation. Besides, other rate control mechanisms such as those in PGMCC [3] and TFMCC [4] can be used in place of the current one, as long as the transmission rate is controlled by the source based on the feedback packets from the most congested receiver.

3.1.2. CR selection (intra-layer)

To choose or update a CR, the source needs to compare the TAF statistics from receivers sent in by CIs. Given receivers i and j, and j being the current CR, let their TAFs be TAF_i and TAF_j. Also, let their average TAFs be Θ_i and Θ_j, and their TAF deviations be Θ_i^σ and Θ_j^σ, respectively. In order to constitute a confidence level on the difference of the two random variables TAF_i and TAF_j, we first assume that the difference of TAF_i and TAF_j fits to a Normal distribution. This assumption is based on the fact that summation/subtraction of multiple random variables obeys to the Normal distribution [32]. It is worthwhile to note that this assumption does not impose any constraints on the behavior of the individual random variables TAF_i and TAF_j.

Based on the above assumptions, we use the characteristics of the Normal distribution to construct a confidence level on the difference of Θ_i and Θ_j. So, given the two random variables, TAF_i and TAF_j, and N samples of each of them, a confidence interval for $TAF_i - TAF_j$ with confidence coefficient $1 - \alpha_2$ can be obtained by checking the following inequality:

$$\Theta_i - \Theta_j > \alpha_2\sqrt{\frac{\Theta_i^{\sigma^2} + \Theta_j^{\sigma^2}}{N}}, \tag{1}$$

where Θ_i and Θ_j are calculated from the N samples of the individual samples TAF_i and TAF_j, respectively. The right hand side of (1) corresponds to the standard deviation of TAF after N samples.

In order to be conservative and to bias the selection towards the current CR, we revise (1) by multiplying Θ_j with another positive constant $\alpha_1 > 1$:

$$\Theta_i - \alpha_1\Theta_j > \alpha_2\sqrt{\frac{\Theta_i^{\sigma^2} + (\alpha_1\Theta_j^\sigma)^2}{N}}. \tag{2}$$

Thus, if the following condition is satisfied, receiver i is chosen as the CR (When there is no CR yet, Θ_j and Θ_j^σ can be adjusted to make the condition always true.) This condition and those to appear later are all based on statistical inference [32]. In particular, we use confidence levels and Chebychev's principles to identify if a receiver is CR or not

$$\Theta_i > \alpha_1\Theta_j + \alpha_2\sqrt{\frac{\Theta_i^{\sigma^2} + (\alpha_1\Theta_j^\sigma)^2}{N}}. \tag{3}$$

[2] The concept of CR here is similar to the representative receiver in DeLucia and Obraczka's work [21] and TFMCC [4].

So, in (3), α_1, α_2 are configurable parameters, and can be used to bias the CR decision towards the current CR or a new receiver. In our simulations, α_1 is set to 1.25 since we want to bias toward the current choice of CR to avoid unnecessary oscillation, α_2 is set to 1.64 for a 90% confidence level.

Although the source needs to perform TAF comparison, it is not necessary for all receivers to send CIs to the source. In GMCC, receivers check the condition of (3) in advance. Only if the condition is true do they send CIs. The information of CR is broadcast to all receivers of that layer for the checking beforehand.

3.1.3. Layer control (inter-layer)

In any GMCC session, there is always a basic layer in which the source keeps sending packets subject to rate control. All other layers must be turned on (i.e. start traffic) or shut down (i.e. stop traffic) at right time to avoid bandwidth waste. Each GMCC session can limit the number of layers to be used. This number is configured at the source and broadcast to receivers periodically. Receiver subscriptions must not exceed this limit. Therefore, if only one layer is allowed, GMCC works the same as a single-rate scheme. The source can potentially control the number of layers to limit the throughput of the whole session.

To perform proper maintenance of the layers, GMCC source does two different operations:

- *Activation*: When a receiver joins a layer which did not have any receiver before, the source needs to start sending packets in this layer, i.e. activate this layer. Since this receiver can infer that there is no CR in this layer yet from the CR statistics conveyed by the source, it will send CIs. Upon the receipt of these CIs, the source realizes there is at least one receiver in this layer and therefore begins transmitting data. The receiver will be immediately chosen as the CR for rate adaptation need.
- *Deactivation*: If all receivers have left a layer, the source has to stop sending data in this layer, i.e. deactivate this layer. Each CR (one per layer) needs to send heartbeat packets once per RTT (known from the source) to the source to maintain its validity. If the source has not received any heartbeat packets from a CR for 8 RTTs, it will request CIs from receivers to choose a new CR. If after 4 RTTs, there is still no CR chosen, the source will set the sending rate to a very low level (e.g. one packet per RTT) and wait for another 20 RTTs.[3] The layer will be shut down if no response comes in during all these periods. In the above procedure, the second period is needed to avoid sudden rate decrease in case there are still other receivers in this layer. On the receiver side, to cooperate with the source, the receivers need to send back CIs to the CR once they know the previous CR is invalid. To reduce the total number of feedback packets, receivers may randomize their feedback according to their TAF value (e.g. the larger the TAF, the sooner CIs are sent). Once a new CR is chosen, its TAF statistics can be used by other receivers to suppress their feedback packets scheduled to send (Section 3.1.1).

3.1.4. Probabilistic inter-layer bandwidth switching (PIBS) (inter-layer)

Since the number of layers and their rates are all dynamic in GMCC, receivers have to employ careful layer join/leave decisions. In some cases, suboptimal rate allocation to some receivers occur; since they may not detect the available bandwidth and do not join new layers. In order to help receivers discover all available bandwidth and help them make the right layer join/leave decisions, we developed the following source-based technique called probabilistic inter-layer bandwidth shifting (PIBS). The essence of PIBS is to shift a small fraction of bandwidth of layer $i+1$ to layer i, so that receivers with highest layer i can discover more of the bandwidth available on their path and thus decide to join layer $i+1$.

Assume multiple layers (layer 1 to n, $n>1$) are used in a multicast session. Let the period between two consecutive rate reductions (in the same layer) be a rate control period (RCP). At the beginning of each RCP at layer i ($1 \leqslant i < n$), with probability ρ, the source decides that it will send data with an additional fraction δ. Otherwise, i.e. with probability $1-\rho$, it will send at the normal sending rate. More specifically, if it is determined that bandwidth shifting is going to be applied during an RCP with a normally calculated sending rate λ_i, the source will actually send packets at the rate of $\lambda_i + \min(\delta\lambda_i, \lambda_{i+1})$. At the same time, at layer $i+1$, the actual sending rate will be adjusted to $\max(0, \lambda_{i+1} - \delta\lambda_i)$ so that bandwidth is shifted to the

[3] All the numbers used here are tunable values.

layer i. Briefly, the source "shifts" some bandwidth from layer $i+1$ to layer i. Notice that we use min and max in the rate calculations to assure that no more than the whole sending rate of layer $i+1$ can be shifted to layer i. To avoid significant unfairness to non-GMCC flows, ρ and δ must be small (both are 0.1 in our simulations). Also, at any moment, no two layers are allowed to perform bandwidth shifting simultaneously.

This bandwidth shifting mechanism above helps receivers to make a better decision on whether or not joining to a new layer. In particular PIBS provides a direct solution to the Situation 3 described in Section 3.2.2. We now describe how exactly PIBS at the source helps receivers to resolve a suboptimal inter-layer rate allocation problem.

Given a receiver R in a GMCC multicast session, assume $\ell > 1$ layers go through the bottleneck on the path between the source and R. As it will be described in more detail later in Section 3.2.1, GMCC receivers measure magnitude of a congestion epoch by a metric called ITAF. According to the definition of ITAF, all receiver paths using the same bottleneck should be observing the same value for ITAF. So, all receivers subscribed to any of the ℓlayers passing through the same bottleneck must be observing the same ITAF value on average. Thus, we can conclude that, for any layer $i < \ell$, the average ITAF measured by R at layer i during bandwidth shifting periods (θ') should be approximately the same as that measured during periods without bandwidth shifting (θ). If this condition is satisfied (i.e. θ' is equal to θ on average), then R needs to join an additional layer. On the contrary, if θ' is larger than θ, it means shifting bandwidth to layer i cause more congestion, indicating that no layer above i goes through the same bottleneck, and that no action is necessary. The other possibility is that θ' is smaller than θ, which means bandwidth shifting is being done from layer i to layer $i-1$, and no action is necessary at R.

In order to compare θ and θ' robustly, we use statistical techniques. Assume R's highest joined layer is k, and the highest layer with traffic for the whole multicast session is L. If $k < L$, R will check the following condition at layer k once it has at least N samples for both θ and θ'

$$\theta - \gamma\sqrt{\frac{\sigma^2 + \sigma'^2}{N}} \leqslant \theta' \leqslant \theta + \gamma\sqrt{\frac{\sigma^2 + \sigma'^2}{N}}, \qquad (4)$$

σ and σ' are the standard deviations corresponding to θ and θ', respectively. If condition (4) is true, the receiver R will join layer $k+1$. Because, condition (4) means that θ and θ' are equal on average with a confidence level depending on the constant γ. γ is used to tune sensitivity of the decision to outlier samples of ITAF. From Chebychev's principles, when γ is 4, about 94% of the samples of ITAF will be within the range defined in (4), i.e. the error probability of not detecting the bandwidth shift will be 6%.

3.2. GMCC receiver

Receivers in GMCC are responsible for participating in both intra-layer and inter-layer decisions. GMCC receivers are expected to perform *Feedback Handling* pertaining to intra-layer decisions. Specifically, all receivers send feed-back for proper selection of CR, and also CR receivers send regular heartbeat feedback packets. These heartbeat packets from CRs are used by the source to adapt the sending rate of the layers and also to perform layer control, i.e. activation or deactivation.

3.2.1. Throughput attenuation factor (inter-layer)

The term *attenuation* is widely used in communications and system analysis where it refers to the reduction in signal strength when the signal passes through a particular medium. Similarly, we perceive the pipe from the source to the destination as a medium through which the source's sent traffic flows. Due to congestion, the sent traffic is reduced at the other end of the pipe and therefore *attenuates*. Fig. 5 shows the concept that the amount of traffic dropped due to congestion is the attenuation that the end-to-end throughput goes through.

Throughput attenuation factor (TAF) is a metric *measured at the receiver side* to indicate how congested the path to the receiver is. It comprises two parts, each describing a different aspect of congestion: Individual throughput attenuation factor (ITAF) and congestion occurrence rate (COR).

- *Individual throughput attenuation factor*: ITAF[4] is defined as

  $$1 - \frac{\mu}{\lambda}$$

  measured only in congestion epochs (A congestion epoch is an event when one or more consecutive packets are lost.[5]) μ is the instantaneous

[4] Table 1 includes all key acronyms and symbols.

[5] We assume that packet loss is due to congestion only.

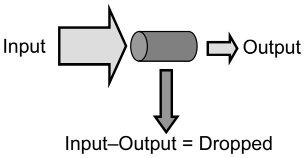


Fig. 5. Source's traffic attenuates due to congestion in the pipe towards the destination, and hence the receiver sees an attenuated throughput.

Table 1
Some key symbols and acronyms

Symbol	Meaning
TAF	Throughput attenuation factor that determines how congested the path to particular receiver
ITAF	Individual throughput attenuation factor that is measured at every instance of congestion epochs
COR	Congestion occurrence rate is the number of congestion epochs per unit time
Θ_i	Average TAF of receiver i
Θ_i^σ	Standard deviation of receiver i's TAF
θ	Average ITAF of a receiver's highest joined layer measured during periods without bandwidth shifting
θ'	Average ITAF of a receiver's highest joined layer measured during bandwidth shifting periods
N	Number of TAF/ITAF samples kept for calculation
J	Number of positive TAF/ITAF comparison results required to join an additional layer

output rate and λ is the rate of input generating this portion of output. It shows how much proportion of input is lost during an instance of congestion, and therefore indicates *how serious* this instance of congestion is. Though different measurement techniques can be developed, we measured ITAF in the following way in the implementation: Each data packet carries the instantaneous sending rate information, assumed to be λ_n for the packet of sequence number n. When a packet of sequence number n arrives, the receiver divides this packet size by the latest packet arriving interval and gets the instantaneous receiving rate μ_n. If the receipt of sequence number n indicates a packet loss, an ITAF is obtained as

$$1 - \frac{\mu_m}{\lambda_m},$$

where m is the received sequence number immediately prior to n.

The essence of ITAF is to measure strength of a particular congestion epoch. The reason behind using m in calculating ITAF is the fact that the very last packet received right before the congestion epoch starts indicates the most accurate output-to-input ratio (i.e. μ/λ), since that last packet is the one that passed through the bottleneck at its maximum output rate possible. Notice that the packet received right the loss burst, i.e. the packet with sequence number n, is less reliable since it might indicate a significantly higher or lower output rate information than what the bottleneck can sustain.

- *Congestion occurrence rate*: COR is defined as the reciprocal of the interval between two consecutive congestion epochs. For instance, if the loss of packet n and $n+i$ (where $i > 1$) is detected at time t_1 and t_2, respectively (with the packets from $n+1$ to $n+i-1$ received), then a sample of COR would be

$$\frac{1}{t_2 - t_1}.$$

COR shows how frequently congestion happens.

With ITAF and COR defined, TAF is the product of these two factors, i.e.

$$\text{TAF} = \text{ITAF} \times \text{COR}.$$

As shown in Fig. 6, TAF is calculated after several occurrence of loss bursts each of which causes an ITAF to be calculated. The larger TAF for a receiver, the more congested is the path to that receiver. Usually the samples of ITAF and COR change abruptly (e.g. due to bursty loss). Therefore, we collect a certain number[6] of ITAF and COR samples, average the samples and use the mean value for TAF calculation. That means, we compare congestion on an average sense. For more detailed analysis of TAF, please refer to our technical report [33]. In GMCC, each receiver measures its own TAF and maintains the mean Θ and standard deviation Θ^σ of the latest N TAF samples for the purpose of TAF comparison.

The meaning of TAF encompasses two dimensions of congestion answering the two questions: (i) "how strong is a congestion epoch?" (i.e. ITAF), and (ii) "how frequent do these congestion epochs occur?" (i.e. COR). Multiplication of these two important components become a "percent per second" unit, which represents a time-dependent attenuation. "attenuation" corresponds to percentage loss that a signal goes through if travels a particular medium. So, attenuation is represented with a unit of "percentage" and is used for fixed time-independent

[6] We used 30 in our simulations.

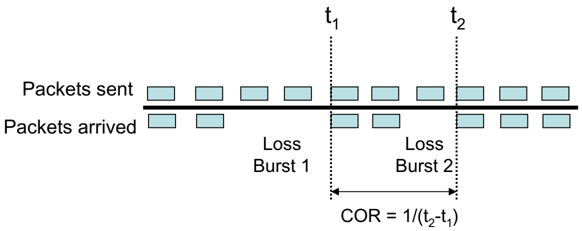


Fig. 6. ITAF and COR are calculated at each loss burst. However, TAF is calculated only after several (30 in our simulations) loss bursts to assure robustness of ITAF and COR samples.

mediums, e.g. water, air, concrete walls. However, loss behavior of an end-to-end path is heavily dependent on time. So, to make it more suitable to an end-to-end path, we multiply an individual instance of TAF (i.e. ITAF) by its occurrence rate. So, the multiplication, i.e. TAF, represents a time-varying loss percentage the end-to-end path is incurring to incoming traffic. Traditionally, congestion is measured by the drop rate, i.e. lost bytes per second. However, this is dependent on the incoming traffic flow's rate λ. What TAF does is to decouple congestion measurement from the traffic flow's rate and makes it a characteristic of the path itself. Indeed, the drop rate is just multiplication of TAF with the traffic flow's rate, i.e. $\lambda \times TAF$.

Also, TAF is implicitly dependent on RTT of crossing flows' paths. The COR part of TAF is implicitly dependent on RTT, as it is measured at every congestion epoch, the time difference of which is a function of RTT. More specifically, consider a bottleneck with capacity μ_b. Also consider two consecutive congestion occurrences/epochs on the bottleneck at times t_1 and t_2. Assuming that all flows (or most of them) on the bottleneck are employing an additive increase multiplicative decrease (AIMD) policy to adapt their sending rates. So, these flows will be increasing their sending rates with s/RTT per RTT, where s is the packet size. Right after the time t_1, the crossing flows will reduce their sending rates with a multiplicative factor (or they will reduce in some manner) to λ_{t_1}. Then, until the time t_2 the flows on the bottleneck will have to saturate a total capacity of $\Gamma = \mu_b - \lambda_{t_1}$ which can be reached in a time of $\Gamma/(s/RTT)$. So, the value of $t_2 - t_1$ will approximately be on the order of $\Gamma/(s/RTT)$. This identifies the relationship $COR \simeq s/(\Gamma RTT)$, which clearly indicates that TAF will be increasing as RTTs of the crossing flows decreases.

We chose to use TAF a careful rigorous study of it. For more detailed analysis of the TAF metric we refer the reader to our technical report [33]. The main reason behind our motivation to include a congestion metric like the TAF is to eliminate the requirement of RTT estimation in TCP-like congestion control schemes. This RTT measurement and estimation requires the source to *exchange* packets with every receiver periodically, which can yield to feedback implosion. TAF is an attempt to solve this problem by requiring the source to only *send* packets rather than exchanging them.

Also, it is notable that TAF is not the only possible congestion measurement methodology compatible with GMCC. As long as there is a metric which can successfully represent the congestion level of the source-to-receiver path, it can be used within the rest of the GMCC-like protocols.

3.2.2. Layer join (inter-layer)

Whenever a receiver enters a GMCC session, it subscribes to the basic layer of GMCC and stays there till it quits the session. Beyond this basic layer, the receiver must perform join operations to increase its total throughput rate at the right time. A receiver joins an additional layer *successively* when it detects that its throughput rate can be potentially increased. There are three situations, and we describe how the join decision is taken in each case.

3.2.2.1. Situation 1: Frequent congestion epochs. *Decision 1*: This case is suitable for those receivers that frequently detect congestion and thus gather enough samples for TAF measurement quickly. Assume we observe receiver *i*, and the CR is receiver *j*. When there is congestion *in the highest layer* that receiver *i* is in, it measures TAF. Once there are at least N TAF samples, it checks the following condition:

$$\Theta_j > \beta_1\Theta_i + \beta_2\sqrt{\frac{(\beta_1\Theta_i^\sigma)^2 + \Theta_j^{\sigma 2}}{N}}, \tag{5}$$

β_1 and β_2 are parameters. We are conservative about join, therefore we heuristically choose $\beta_1 = 2$, and $\beta_2 = 2.58$ for a 99% confidence level. If the condition in (5) is true for J consecutive times, the receiver will join an additional layer. $J \geqslant 1$ is another parameter controlling conservativeness of join operations and we use $J = 30$ in our simulations. The reason to use relatively small N for samples and J for TAF comparison results, instead of to use a single large N for samples, is that calculating the mean and deviation of a large set of

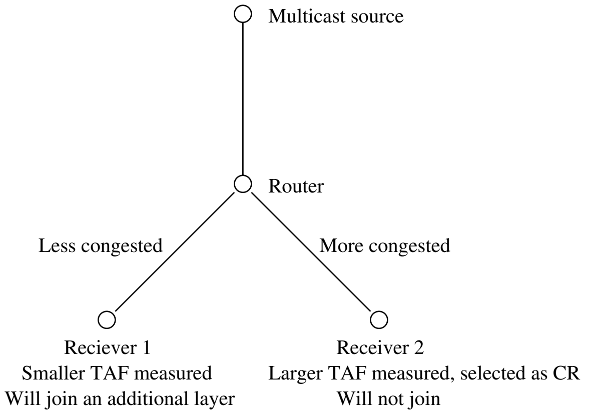


Fig. 7. A topology example for Situation 1 and 2.

samples is expensive. Meanwhile, this method can catch the dynamics of networks.

For example in Fig. 7, Receiver 1 is behind a less congested bottleneck and measures smaller TAF on average. Receiver 2 is behind a more congested link and have larger TAF on average. At some point, Receiver 1 will have detected that the condition in (5) has been true for J times, and decide to join an additional layer.

Although the TAF comparison in other layers can also stimulate the receiver to join more layers, restricting it in the highest joined layer has equivalent effect and simplifies the design.

3.2.2.2. Situation 2: Infrequent congestion epochs. Decision 2: If the congestion detected by a receiver is light, it may take a long time for this receiver to collect enough samples to make a join decision under Situation 1. The solution is to let receivers join under another situation.

When a CR gets a new TAF sample and updates its TAF statistics, it sends a CI to the source with new TAF information. The information is then broadcast to all receivers. When a non-CR receiver notices that the CR of its highest joined layer has updated TAF statistics, this receiver assumes that there is packet loss at this moment and calculates a test version of TAF using its current average ITAF value and the *hypothetical* packet loss interval. For example, a receiver has joined up to layer L. At time t_1 the receiver detects packet loss at layer L and calculates average ITAF as x, COR and then TAF. At a later time t_2 (no packet loss between t_1 and t_2) the receiver notices that the CR in layer L has updated TAF statistics. It then calculates a test version of average COR y using the sample $1/(t_1 - t_2)$, and computes a test version of TAF as xy together with the mean and deviation of TAF. Using this mean and deviation, it checks the condition in (5). Once there are J consecutive positive results, it joins layer $L + 1$ from layer L.

Note that the test version of COR and TAF are not accepted as permanent samples since they are not true samples. Once used, they are discarded. Consequently, the judging in Situation 1 will not be affected.

3.2.2.3. Situation 3: Multiple layers on a shared bottleneck. Decision 3: Still, there is a special case which cannot be dealt with by the solutions for Situation 1 and 2. Consider a topology in Fig. 8 containing two bottlenecks. The links L_x and L_y have ample bandwidth to avoid any congestion. At the beginning, R1 and R2 are both in only one layer. Therefore, only Bottleneck 2 can be fully utilized, and R2 will join a second layer. After that, Bottleneck 1 is also full. At a later moment, R3 enters the session. The congestion it detects will be approximately the same as that detected by R1. In consequence, R3 stays in only one layer, without knowing it can actually join an additional layer without increasing the congestion on Bottleneck 1. The reason is that the congestion generated by *intra-session flows* of other layers is not distinguished from that by *inter-session flows*, whereas the congestion of the former kind can actually be ignored in the context of deciding whether to join. In other words, when deciding to join/leave layers in a GMCC multicast session over a bottleneck shared with other unicast or multicast sessions, the congestion caused by other traffic sessions (i.e. inter-session flows) must be considered while ignoring the congestion caused by the traffic of the other layers of the GMCC multicast session (i.e.

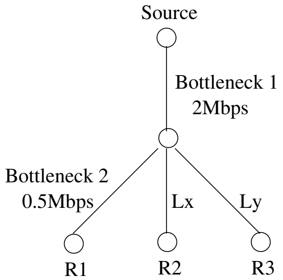


Fig. 8. A topology example where probabilistic inter-layer bandwidth shifting is needed (Situation 3).

intra-session flows). This problem also occurs in SMCC, but the paper [1] did not consider it.

A solution can be that, for the above example, sometimes we try to send more (e.g. 0.55 Mbps on average) in the first layer, while sending less in the second layer (e.g. 0.45 Mbps on average). If R3 does not see any increased congestion, it will know that a portion of the congestion is incurred by intra-session flows, therefore can join the second layer. Certainly this method should be carefully managed because sending more in a layer might cause more severe congestion on some paths. We described this technique in detail in Section 3.1.4.

3.2.2.4. Two exceptional cases. Even if a receiver decides to join under the three situations above, to prevent spurious join, there are two more cases to be checked before the join operation really occurs.

- *Case 1*: If any layer in the whole session does not have a CR yet, the join attempt should be canceled.
- *Case 2*: If a receiver is already a CR for some layer, or detects that it may become a CR in any of its joined layers, it also refrains itself from join. The detection is done by checking the following condition, assuming this receiver is i and the CR is j:

$$\Theta_i > \Theta_j + \omega\sqrt{\frac{\Theta_i^{\sigma^2} + \Theta_j^{\sigma^2}}{N}}, \tag{6}$$

ω decides confidence level, and we used 3.5 for 99.99%.

The first case above means that if in a layer there is no CR yet, the sending rate may not have stabilized. Either the sending rate has not been increased enough to fully utilize the available bandwidth, or the rate is still in the process of decreasing to adapt to the network situation. Under this vague situation, we cannot draw a conclusion whether it is appropriate or not for a receiver to join, and therefore have to wait. The second case shows that a receiver has the potential to become a CR in a layer. The reason of a receiver being CR is because that the total throughput rate of this receiver has matched its share of the bottleneck bandwidth. As a result, this receiver has to restrain the source from increasing the sending rate too much. Obviously, as long as the receiver is a CR or may become a CR, there is no more room for its throughput rate to increase.

It is worth mentioning that GMCC does not have "join attempt" as SMCC does. SMCC [1] performs additive increase join attempts only when a receiver wants to join the next successive (i.e. higher) layer. Specifically, a receiver attempting to join the next layer j keeps increasing its reception rate until it attains the throughput limit of the layer j. If no loss was detected during the additive increase of the receiver's reception rate, then the join takes place. Otherwise, the join attempt is ceased.

We believe that, in GMCC, since both the sending rates in each layer and the number of layers can be dynamically adjusted, as a multicast session goes on, the combination of sending rate settings and the choice of layer number will evolve to the extent that will accommodate the heterogeneity among the receivers, so that a join will not cause abrupt severe congestion. Moreover, omitting join attempts significantly simplifies the design.

3.2.3. Layer leave (inter-layer)

When a GMCC receiver is to leave a layer, it always unsubscribes from the highest joined layer first. Also, after a receiver joins a layer, it needs to wait for some time to allow the network to stabilize before performing any leave operation on that layer. This is achieved by collecting N more samples for TAF statistics in *all* joined layers before it checks whether to leave. Then, if the receiver is the CR, or satisfies the condition (6), in more than one layer, it leaves the highest layer it is in. The reason is the same as explained in the second exceptional case of join at the end of Section 3.2.2.

3.3. Decoupling inter-layer and intra-layer operations

A key contribution of GMCC is that it proposes a framework to decouple inter-layer functions from intra-layer ones. Inter-layer operations of GMCC imposes only two requirements to the underlying single-rate MCC scheme:

- The existence of a congestion representative (CR) with explicit feedbacks corresponding to congestion indications (CIs). Note that a CR with CIs is a basic component of all existing single-rate MCC schemes.
- A way of telling the underlying scheme's rate adaptation module that it needs to increase/decrease its rate with a small amount, in order

to do the probabilistic inter-layer bandwidth shifting (PIBS) in Section 3.1.4. This interfacing can be easily done by allowing inter-layer management to set/unset a particular flag for each layer, where setting corresponds to increase and unsetting corresponds to decrease the sending rate of the single-rate intra-layer scheme.

Given such underlying single-rate MCC schemes, GMCC can perform its operations. For example, in the deactivation methodology in Section 3.1.3, all the techniques that GMCC uses are based on observing the CIs of the CR for the layer to be deactivated. There is indeed no other information required except the RTT measurement, which is layer-independent. So, as long as there exists a CR, which is sending explicit CIs, the deactivation module of GMCC will work independent of the other functionalities of the underlying single-rate MCC scheme.

3.4. Number of layers

A key issue in GMCC is that it dynamically adjusts the total number of layers to adapt to the heterogeneity of receivers, while SMCC sets up the number of layers in advance. Thus on one hand, GMCC avoids too many redundant layers when some receivers leave and all the remaining receivers have the same bandwidth. By eliminating redundant layers in this scenario, it avoids unnecessary exchange of control information. Furthermore, GMCC increases the total number of layers when it detects new receivers joining and the differences between receivers' rates dramatically change. In this section, we theoretically show that the increase of total number of layers K in the second scenario helps to improve session utility in the optimization framework of [30]. We want to emphasize here that when K is fixed, GMCC can be modeled in a similar way as in [30]. However, since K is dynamically changing in GMCC, it will correspond to a series of optimization problems indexed by K, instead of just one as presented in [30]. Thus, beyond the optimization framework of [30], GMCC also dynamically searches for the optimal session utility along another dimension K. To completely model it, a cost function reflecting the load of exchanging control information per unnecessary layer has to be deducted from the overall utility function. In this paper, we just borrow the framework of [30] to show that the tuning of K does affect the session utility, which is one important stepping stone for our proposed scheme GMCC.

Consider a multicast media session with a partitioning of the receivers into K groups. Let P be the partitioning set and R be the set of receivers. Thus, the set $P = \{G_1|G_2|\cdots|G_K\}$ is a partitioning of the receiver set $R = \{1, 2, \ldots, N\}$ and decomposes set R into a family of disjoint sets. We assume that the receivers are numbered such that their isolated rates are in a non-decreasing order, i.e. $r_1 \leqslant r_2 \leqslant \cdots \leqslant r_N$ with ordered group rates $g_1 \leqslant g_2 \leqslant \cdots \leqslant g_K$, where g_k denotes the rate of group G_k. For the clarity of the presentation, when a new layer is added, we assume the new partitioning as $P' = \{G_1|G_2|\cdots|\underbrace{G_1^k|G_2^k}_{G_k}|\cdots|G_K\}$ with the group G_k divided into two groups, i.e. G_1^k and G_2^k. However, the conclusion holds for more general case. We consider a rational utility function for group G_k [30]:

$$IRFA_k = \sum_{i\in G_k} \frac{(2+a)r_i g_k}{g_k^2 + a r_i g_k + r_i^2}, \quad -2 < a < 2, \tag{7}$$

which is the approximation of the most widely accepted max–min fairness utility function [30]:

$$IRFA_k = \sum_{i\in G_k} F(r_i, g_k) = \sum_{i\in G_k} \frac{\min(r_i, g_k)}{\max(r_i, g_k)}.$$

The purpose of this approximation is to replace the non-continuously differentiable max–min fairness utility for receiver i of group G_k with a mathematically well-behaved function over the real numbers axis. The objective is thus to maximize the session utility:

$$IRFA_{\text{Total}} = \sum_{k=1}^{K} IRFA_k = \sum_{k=1}^{K} \sum_{i\in G_k} \frac{(2+a)r_i g_k}{g_k^2 + a r_i g_k + r_i^2}.$$

We see that the new partitioning yields higher session utility by observing the following:

$$\begin{aligned} \max_{g_k} \sum_{i\in G_k} \frac{(2+a)r_i g_k}{g_k^2 + a r_i g_k + r_i^2} &\leqslant \sum_{i\in G_k^1} \frac{(2+a)r_i(g_k^1)}{(g_k^1)^2 + a r_i(g_k^1) + r_i^2} \\ &\quad + \sum_{i\in G_k^2} \frac{(2+a)r_i(g_k^2)}{(g_k^2)^2 + a r_i(g_k^2) + r_i^2}. \end{aligned} \tag{8}$$

Note that in this optimization framework [30], when the number of groups is equal to the number of layers, i.e. $K = N$, it is easy to show that $g_k = r_i$ yields the maximum session utility $IRFA_k$ and thus

$IRFA_{\text{Total}}$. However, it is only true when the cost of exchanging the control information is not taken into account. Thus, a tradeoff exists between maximizing the session utility and minimizing the control traffic, which is also verified by the simulation results in Section 4. Thus, dynamic tuning the number of layers makes GMCC more adaptive to the network fluctuations and balance between utility and control cost.

4. Simulations

We have run several *ns*-2 [34] simulations to test the performance of GMCC. We used drop-tail routers with buffer size set to 20KB. We used TCP Reno for background traffic. By various simulation experiments, we have tested the following aspects of GMCC performance:

1. *Effectiveness of the adaptive layering*, to show that GMCC does not use redundant layers to satisfy heterogeneous receivers (see Section 4.1).
2. *Responsiveness to traffic dynamics*, to show how GMCC responds to dynamically changing competing traffic (see Section 4.2).
3. *Effectiveness of probabilistic inter-layer bandwidth shifting (PIBS)*, to show that the technique of PIBS is valid (see Section 4.3).
4. *Throughput improvement*, to show that GMCC can achieve good throughput for heterogeneous receivers (see Section 4.4).
5. *Large-scale scenarios*, to show that GMCC can achieve good throughput for very large number of (e.g. 1200) heterogeneous receivers (see Section 4.5).

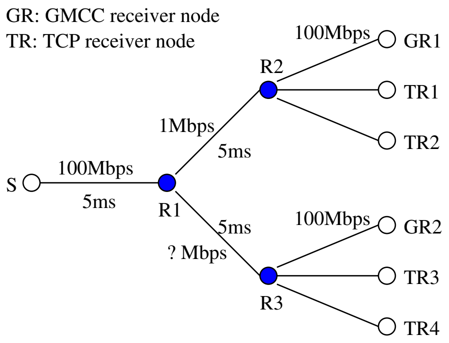


Fig. 9. Topology for layering effectiveness test (Section 4.1).

In the third simulation, we will also show that feedback packets of non-CR receivers can be suppressed efficiently, as described in Section 3.1.1.

4.1. Effectiveness of the adaptive layering

GMCC uses barely enough layers to satisfy heterogeneous receivers, as shown in the following simulations. In the topology of Fig. 9, four TCP flows go from node S to TR1, TR2, TR3, TR4, respectively. A GMCC session has S as the source and GR1, GR2 as the receivers. In the first simulation, the bandwidth of the link between R1 and R3 is set to 5 Mbps. In the second simulation, it is set to 10 Mbps. Obviously, in both simulations, with efficient layer settings, only two layers are needed, where GR1 subscribes to only one layer, and GR2 subscribes to both.

The throughput of the flows in these two simulations are shown in Fig. 10. GR2 joined an additional layer at 15.8th second and at 22.4th second

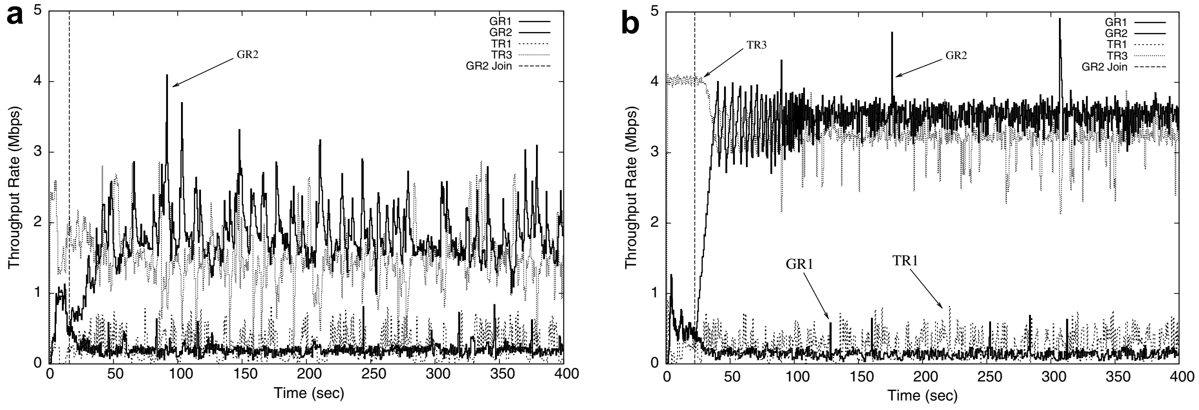


Fig. 10. Results of the effective layering tests on the topology of Fig. 9 (Section 4.1): GMCC can efficiently adjust the number of layers to two in both cases (a) and (b), and also tunes the sending rates of the layers appropriately for the 5 Mbps and 10 Mbps bottlenecks in (a) and (b), respectively. (a) Throughput when the link (R1,R3) is 5 Mbps, (b) throughput when the link (R1,R3) is 10 Mbps.

in the first and second simulations, respectively, and stayed in two layers till the end of simulations. In contrast, GR1 only subscribed to the basic layer. This conforms to the expectation above and shows that the GMCC does not use more layers than necessary. For comparison, consider SMCC with 1 Mbps, 2 Mbps, 4 Mbps limits for the lowest three layers. In the second simulation, since GR2's average throughput rate is above 3 Mbps, it will have to subscribe to at least three layers with some redundancy.

4.2. Responsiveness to traffic dynamics

There are two types of response to traffic dynamics. The first type of response is by the source that adjusts sending rates within layers. GMCCs rate adaption by source is almost the same as that in our single-rate work ERMCC [2]. Therefore, we omit the examination of source response to traffic dynamics here, and refer readers to [2]. The second type of response is by receivers by means of joining and leaving layers. It can be considered as a complementary measure of the first type response, since the latter is limited by CRs.

We used the star topology in Fig. 11 to test the receivers' responsiveness to the dynamics of crossing traffic on the bottleneck. A GMCC session has GS1 as the source node and R1, R2 as the receiver nodes. On each of the links of (R, R1) and (R, R2), there are six TCP competing flows at the beginning of the simulation. During the period between 100th and 200th second, five TCP flows on the link (R, R2) pause, leaving one TCP flow as the only competing flow.

As shown in Fig. 12, receiver R2 joined an additional layer at 135.412th second. After those five TCP flows pause, the link (R, R2) became much less congested than (R, R1). Therefore, this join operation is appropriate. There is 35-s gap between the pause and the join operation, though. That is relatively long because GMCC is conservative about join and therefore requires enough number of samples and positive TAF comparison results (see Section 3.2.2). However, GMCC is quicker when making decisions about unsubscription. In this simulation, R2 left the layer at 205.178th second. On the other hand, since there is no traffic dynamics on the link (R, R1), receiver R1 remains in one single layer.

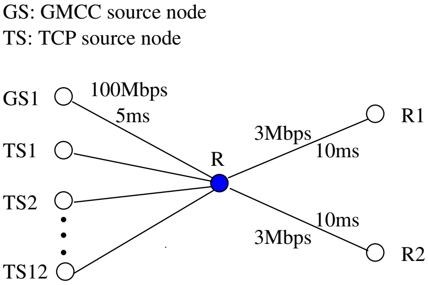


Fig. 11. Star topology for testing responsiveness to traffic dynamics (Section 4.2).

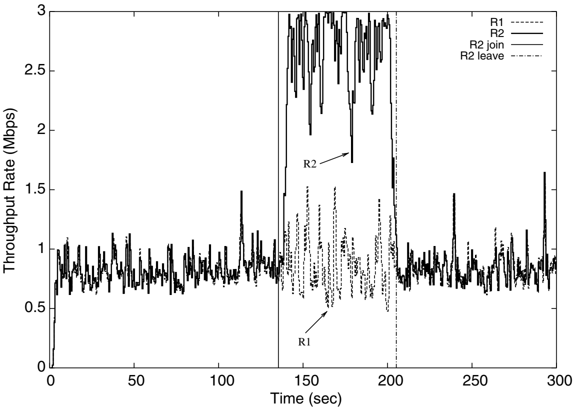


Fig. 12. Results of the tests for responsiveness to traffic dynamics (Section 4.2): GMCC performs well in adapting to varying background traffic. The two GMCC receivers R1 and R2 (in Fig. 11) obtain their optimal throughput. Also, R2 joins an additional layer when five TCP flows stop and thereby leaving extra bandwidth on the bottleneck of R2.

We notice that being in only one layer, R1 has average throughput of 0.83 Mbps that is more than the fair share of 0.43 Mbps (≈ 3 Mbps/7). The reason is because we used TCP Reno that suffers from unnecessary timeouts and therefore performance degradation when multiple packets are dropped from a window of data [35]. Meanwhile, since we used drop-tail buffer management on routers, packet losses are in bursts. In the future, we will explore the performance of GMCC using other flavors of TCP and other types of buffer management.

4.3. Effectiveness of probabilistic inter-layer bandwidth shifting (PIBS)

Recall that probabilistic inter-layer bandwidth shifting (PIBS) is a technique we developed in Situation 3 of Section 3.2.2 to distinguish the congestion incurred by intra-session flows from that by inter-session flows. This technique enables the receivers to join under some situations with shared bottlenecks. To verify that PIBS is a valid technique, we

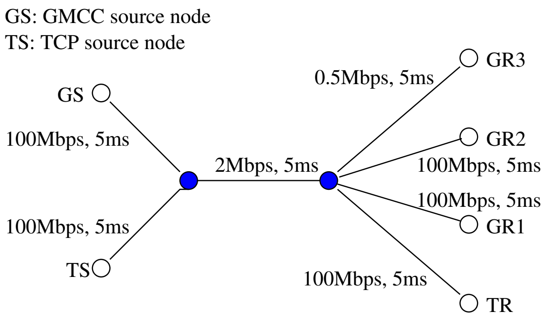


Fig. 13. Topology for testing probabilistic inter-layer bandwidth shifting (Section 4.3).

ran a simulation on the topology in Fig. 13. A TCP flow originates at TS and ends at TR as background traffic. The GMCC flows in a multicast session go from GS to GR1, GR2 and GR3. The 2 Mbps bottleneck is shared by all three GMCC receivers, and the 0.5 Mbps bottleneck only affects GR3. At the beginning of the simulation, only GR1 and GR3 are in the session. At 100th second, GR2 enters the session. Fig. 14 shows that in one simulation instance, GR2 subscribed to an additional layer at 170.146th second based on bandwidth shifting. Again, there is long delay because GMCC receivers need to collect enough samples before making decisions.

We noticed that in some other instances of this simulation, a join operation for another reason (in particular, under Situation 2 in Section 3.2.2) happened before the results of bandwidth shifting took effect, and the join operations triggered by bandwidth shifting were suppressed. This is not unexpected because the flows are dynamic and the comparisons in GMCC are all probabilistic. It is possible that during some random periods the condition in situation 2 becomes true and triggers a join operation.

We can also see how feedback suppression works in this simulation. As the CR in layer 1, GR3 sent 4424 feedback packets; as the CR in layer 2, GR1 sent 5448 feedback packets. Most of these packets are heartbeat packets, sent once per RTT of around 110 ms. GR2, since it is not CR at any time, only sent 2 CIs. Therefore, feedback from non-CR receivers is efficiently suppressed.

4.4. Throughput improvement

The topology in Fig. 15 contains six bottlenecks and is used to test how GMCC improves the throughput of heterogeneous receivers with relatively slight difference of expected throughput. All the links are of 5 ms delay. The bandwidths of the bottlenecks are from 1 Mbps to 6 Mbps. On each of them, there are two TCP flows as competing traffic. A GMCC session is held between the source GS and six receivers (GR1 to GR6). Simulation time is 600 s.

Fig. 16 shows the over time average throughput rate of all receivers. Over time average throughput rate at time t is defined as the total throughput through time t divided by the total run time. We can see that the six GMCC receivers do achieve different

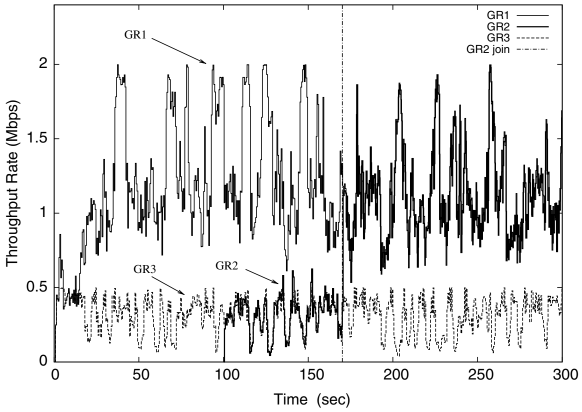


Fig. 14. Test results of PIBS (Section 4.3): Throughput of All GMCC Receivers. The technique of PIBS can exploit hidden available bandwidth. The GMCC receiver GR2 joins the multicast session at time 100 s and discovers the available bandwidth on its path at time 170.1 s by means of the PIBS method.

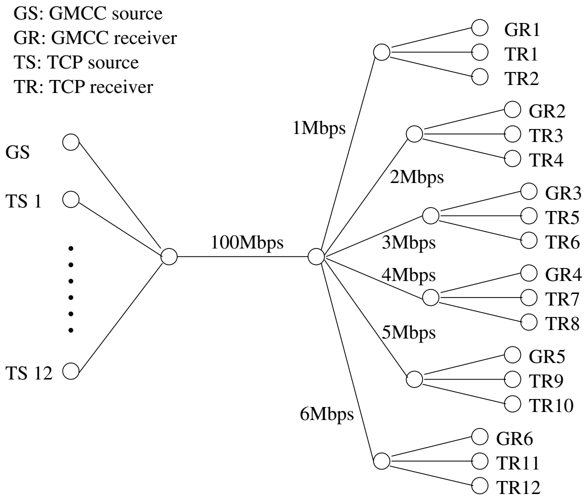


Fig. 15. Topology for testing throughput improvement (Section 4.4).

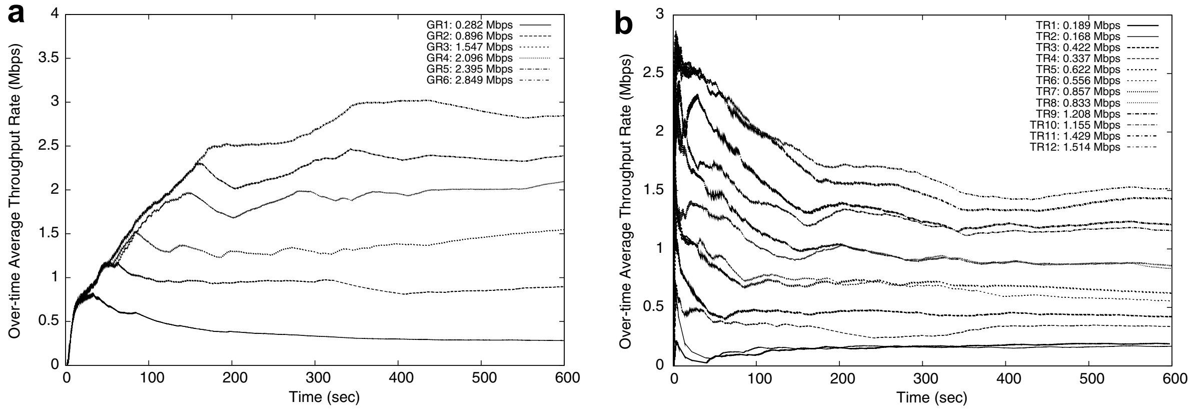


Fig. 16. Throughput improvement test result (Section 4.4): receiver throughput in the topology of Fig. 15. GMCC receivers behind different bottlenecks get different throughput matching the bottleneck capacity. (a) Over-time average throughput rate of GMCC receivers, (b) over-time average throughput rate of TCP receivers.

throughput rates, with GR6 being the highest and GR1 being the lowest. So, GMCC used six layers in this simulation experiment. Just to sketch a comparison, SMCC would only make three layers with 1 Mbps, 2 Mbps, and 4 Mbps layer throughput limits, which would be very inefficient for receivers GR3, GR5, and GR6 since they would have to adapt to the slowest of their highest layers thereby wasting 1 Mbps, 1 Mbps, and 2 Mbps, respectively.

Besides, there were only a few join and leave operations in this simulation. Compared to the previous multi-rate schemes where join and leave happen every RTT or so, GMCC clearly provides a great improvement. The number of join and leave operations of each receiver is listed in Table 2. Note that since join and leave are triggered by statistical comparisons, there were several oscillations that increased the operation numbers (e.g. for GR2).

In this simulation, GMCC receivers achieve higher throughput than TCP correspondents. The reason is that each flow in a GMCC layer is a single-rate congestion control flow independent of other flows. It competes for bandwidth like any other flow does. For example, when GR2 subscribes to two layers, there are then two TCP flows and two GMCC flows on the 2 Mbps bottleneck. The throughput of GR2 is the sum of both GMCC flows, and therefore can be approximately twice as much as each of the TCP flows. However, due to the limit by CRs in lower layers, assuming there are n TCP flows and m GMCC flows on a bottleneck, a receiver may not get the share of $m/(m+n)$. GR6 here is an example. Although what we observed for GMCC in this simulation is different from traditional TCP-friendliness concept, each GMCC flow within a layer still competes in a TCP-friendly manner. Another reason behind this result is the fact that we use a single-rate MCC scheme, ERMCC [2], which is not fully TCP-friendly due to its rate-based transmission unlike other schemes such as TFMCC [4] using TCP throughput formula to calculate their sending rates.

Table 2
Number of join and leave operations: the number of IGMP operations is small and incur very light control traffic

	GR1	GR2	GR3	GR4	GR5	GR6
Join	4	12	11	10	9	7
Leave	4	11	8	6	4	2

To achieve TCP-friendliness for the aggregation of all the layers' rates is non-trivial. Our scheme, GMCC, cannot achieve such "all-layers" TCP-friendliness, i.e. aggregate rate of all layers being TCP-friendly. GMCC provides "per-layer" TCP friendliness because of its reliance on underlying single-rate MCC schemes which are typically TCP-friendly. Many other scalable multi-rate schemes (e.g. [10,1,29,28,30]) have been able to achieve per-layer TCP-friendliness. However, not many studies have taken place towards attaining an all-layers TCP-friendliness. RLS [26] was an early proposal with this goal in mind, though all-layers TCP-friendliness was only achieved towards TCP connections having RTTs of approximately 1 s. Later, as an improvement to RLS, Coding-Independent Fair Layered multicast (CIFL) [36] offered TCP-friendliness at all-layers level. Future research could investigate how to use our GMCC framework of leveraging per-layer-fair single-rate MCC schemes to construct an all-layers-fair scheme as this paper does not have the answer.

4.5. Large scale scenarios

We have also run a large simulation by using the simulation engine ROSS [37] on the topology of Fig. 17, the same one used for ERMCC [2] large scale simulation. The background traffic on the last hops is generated by two single-receiver PGMCC flows (since its behavior is close to TCP), and the last hop is the only bottleneck on the path from the source to a receiver. There are 1200 receivers, each behind a different bottleneck. All the bottlenecks are divided into ten even groups, their bandwidths being from 0.2 Mbps to 1.2 Mbps with difference as 0.2 Mbps.

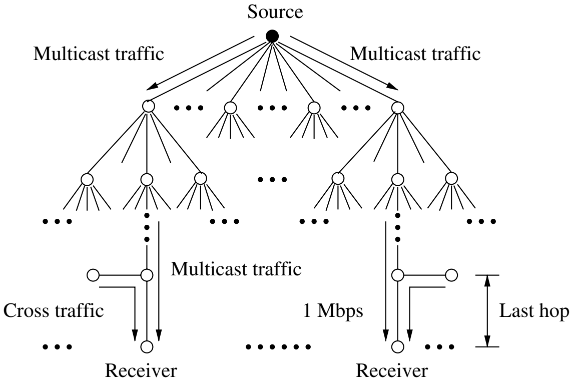


Fig. 17. Tree topology for large-scale simulations in ROSS [37].

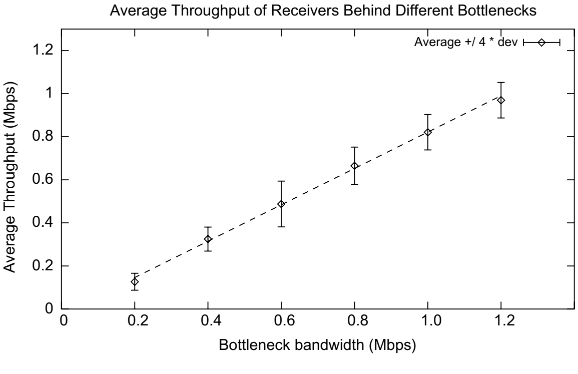


Fig. 18. Average throughput and deviation of different groups of receivers: different groups of GMCC receivers get different throughput proportional to their bottlenecks.

The simulation ran for 2000 s. The average throughput and the deviation of each group of receivers is shown in Fig. 18. The average throughput grows linearly with the bottleneck bandwidth, again showing that the multi-rate feature of GMCC is effective. The numbers of join and leave operations are in Table 3. (Group i is the group of receivers behind the bottlenecks of $i \times 200$ Kbps bandwidth.) Even in the most active groups, on average, each receiver has less than 15 join operations and much fewer leave operations (around 1) within 2000 s. Obviously, the volume is very light.

5. Conclusion and future work

We have presented a multi-rate multicast congestion scheme called GMCC. By combining single-rate congestion control and traditional multi-rate techniques (mostly joining and leaving layers by receivers) in a novel way, it provides a simple design for a perplexing problem of which most previous solutions are complicated. While having the merits of a similar previous scheme SMCC [1], it is *fully* adaptive and surmounts the limits posed by SMCCs required static configurations. A new technique called *probabilistic inter-layer bandwidth shifting* is proposed as the solution to a problem not mentioned in SMCC. Besides, the rate control mechanism at source can be replaced by other representative- based mechanisms.

There is still another potential problem of SMCC. Assume receiver R has joined a set of layers $\mathscr{L}$. In SMCC, R calculates its estimated throughput using TCP throughput formula [13] with the overall "loss event rate" [1,4] of all the layers in $\mathscr{L}$ as one of the parameters. Treatment of all layers with a single loss event rate values can causes performance degradation of some of the receivers. If the layers in $\mathscr{L}$ have different underlying multicast trees, the overall

Table 3
Number of join and leave operations in large scale simulations: the number of average per-receiver IGMP operations is very small, even at the presence of many receivers behind different bottlenecks

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Join	1913	2757	2986	2988	2896	2947
Leave	0	246	155	35	0	0

loss event rate can potentially be higher than the individual value of any single layer, since each layer can experience loss events at different links in the topology with different magnitudes. For example, let there be two different paths, P_1 and P_2, from the source to a receiver. Suppose that P_1 is used for layer 1, P_2 is used for layer 2, and that P_1 has much higher loss rate, while P_2 has very low loss rate. The receiver will join layer 1 over P_1 first, and its estimated throughput would be very low. Therefore the receiver will not consider joining layer 2. So, since in SMCC, receivers rely on estimated throughput to decide how many layers to join, and the source uses estimated throughput from receivers to control sending rates, the underestimation can degrade the performance for some of the receivers. On the contrary, GMCC does not have this problem since it does not rely on the estimated throughput in its decision-making of layer joins/leaves. However, it needs more careful exploration.

This paper presents the first step theoretical study of GMCC-like schemes. We note that intra-layer rate adaptation adjusts source sending rate in a similar way with AIMD, which has been shown to correspond to an optimization framework for unicast network [38]. Thus, our next question is whether GMCC also corresponds to any optimization framework. The answer is promising by observing the recent studies on optimization based rate control for multi-rate multicast network, e.g. [39–41]. In [39], the authors explored the problem of fair allocation of resources in multi-rate multicast networks and present a mathematical formulation of maximizing the "social welfare", i.e. sum of the utilities over all receivers, subject to the link constraints. Similarly [40], considered the networks which support both multi-rate multicast sessions and unicast sessions and presented a decentralized algorithm which enables the different rate-adaptive receivers in different multicast sessions to adjust their rates to satisfy some fairness criterion. In these studies, layer adding and dropping are not explicitly considered and they are implied in the rate adaptation for different receivers. Considering the fact that GMCC is fully adaptive without any rigid limits on the sending rates of each layer and restriction on total number of layers, we would conjecture that GMCC is the closest "discrete version" of those algorithms based upon the mathematical optimization model. Specifically, GMCC uses dynamic source rate control and considers price generations at links, and thus has the features of both [39] (which has the dynamic link algorithm) and [40] (which employs the dynamic source rate control).

GMCC is suitable to any data multicasting applications. However, GMCCs adaptively changing layer settings cause additional complexity in designing codecs for multimedia streaming applications. Specifically, receivers subscribing to a particular layer would like to receive the multimedia stream at a predefined quality for that subscription. This strict expectation levels do not fit to the adaptive rates of the layers. A solution could be to decouple subscription mechanism from the actual layers and deploy progressive source coding techniques [42,43] and wavelet coding techniques [44] to cope with adaptive sending rates of the layers. This issue clearly deserves further research.

Regarding its implementation, a GMCC source can control the throughput of a session by sending dynamic layer settings to receivers which will then adjust their subscriptions. How this works is still vague and deserves more investigation of viable subscription mechanisms capable of dealing with layers with adaptive transmission rates. We would also like to conduct simulations in more complex topologies and with different types of buffer management (e.g. RED) on routers.

Another important future research dimension is the investigation of the possible effects of the underlying single-rate MCC scheme on the overall operations and performance of GMCC-like schemes. Single-rate MCC schemes deal with quite a lot of issues in achieving a well-performing multicast [45], such as congestion representative (CR) selection, CR tracking, feedback suppression, and TCP window management in the case of window-based rate adaptation. In this paper, we used our single-rate MCC scheme, ERMCC [2], with its well-tuned parameters. It would be interesting to study the effects of having a single-rate MCC scheme without well-tuned parameters on the overall GMCC performance.

Acknowledgements

Authors would like to thank the anonymous reviewers for their invaluable comments and feedback on the paper. This work was supported in part by NSF Contract ANI9819112, ARO Contract DAAD19-00-1-0559 and grants from Intel and Reuters.

References

[1] G.-I. Kwon, J. Byers, Smooth multirate multicast congestion control, in: Proceedings of IEEE INFOCOM, April 2003.

[2] J. Li, M. Yuksel, S. Kalyanaraman, Explicit rate multicast congestion control, Computer Networks 50 (15) (2006) 2614–2640.

[3] L. Rizzo, PGMCC: A TCP-friendly single-rate multicast congestion control scheme, in: Proceedings of ACM SIGCOMM, 2000.

[4] J. Widmer, M. Handley, Extending equation-based congestion control to multicast applications, in: Proceedings of ACM SIGCOMM, 2001.

[5] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, in: Proceedings of ACM SIGCOMM, 1996.

[6] A. Legout, E. Biersack, PLM: fast convergence for cumulative layered multicast transmission schemes, in: Proceedings of ACM SIGMETRICS, 2000.

[7] L. Vicisano, L. Rizzo, J. Crowcroft, TCP-like congestion control for layered multicast data transfer, in: Proceedings of IEEE INFOCOM, April 1998.

[8] John Byers, Michael Frumin, Gavin Horn, Michael Luby, Michael Mitzenmacher, Alex Roetter, William Shaver, FLID-DL congestion control for layered multicast, in: Proceedings of NGC, November 2000.

[9] J. Byers, M. Luby, M. Mitzenmacher, Fine-grained layered multicast, in: Proceedings of IEEE INFOCOM, 2001.

[10] J. Byers, G. Kwon, STAIR: practical AIMD multirate multicast congestion control, in: Proceedings of NGC, 2001.

[11] M. Luby, V.K. Goyal, S. Skaria, G.B. Horn, Wave and equation based rate control using multicast round trip time, in: Proceedings of ACM SIGCOMM, 2002.

[12] W.C. Fenner, Internet Group Management Protocol, Version 2, RFC 2236.

[13] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, in: Proceedings of ACM SIGCOMM, 1998.

[14] IPTV World Forum. Available from: <http://www.iptv-forum.com>.

[15] Microsoft TV. Available from: <http://www.microsoft.com/tv>.

[16] H. Rahul, M. Kasbekar, R. Sitaraman, A. Berger, Towards realizing the performance and availability benefits of a global overlay network, in: Proceedings of Passive and Active Measurement Conference (PAM), 2006.

[17] B. Maggs, Global Internet content delivery, in: Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001.

[18] Apple iPod. Available from: <http://www.apple.com/ipod>.

[19] HDTV and The Digital TV Transition. Available from: <http://hdtvinfoport.com>.

[20] S. Deering, Host Extensions for IP Multicasting, RFC 1112, August 1989.

[21] D. DeLucia, K. Obraczka, A Multicast congestion control mechanism using representatives, in: Proceedings of IEEE ISCC, 1998.

[22] J. Macker, R. Adamson, A TCP friendly, rate-based mechanism for nack-oriented reliable multicast congestion control, in: Proceedings of IEEE GLOBECOM, 2001.

[23] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, ACM Computer Communications Review 27 (3) (1997).

[24] T.T. Fuhrmann, J. Widmer, On the scaling of feedback algorithms for very large multicast groups, Computer Communications 24 (5) (2001) 539–547.

[25] P. Thapliyal, Sidhartha, J. Li, S. Kalyanaraman, LE-SBCC: loss-event oriented source-based multicast congestion control, Multimedia Tools and Applications 17 (2–3) (2002) 257–294.

[26] I.E. Khayat, G. Leduc, Congestion control for layered multicast transmission, Networking and Information Systems Journal 33 (4) (2000) 559–573.

[27] M. Kawada, H. Morikawa, T. Aoyama, Cooperative interstream rate control scheme for layered multicast, in: Proceedings of Symposium on Applications and the Internet (SAINT 2001), January 2001, pp. 147–154.

[28] D. Sisalem, A. Wolisz, MLDA: a TCP-friendly congestion control framework for heterogeneous multicast environments, in: Proceedings of IWQoS, June 2000.

[29] J. Liu, B. Li, Y.-Q. Zhang, A hybrid adaptation protocol for TCP-friendly layered multicast and its optimal rate allocation, in: Proceedings of IEEE INFOCOM, June 2002.

[30] H. Yousefizadeh, H. Jafarkhani, A. Habibi, Layered media multicast control (LMMC): rate allocation and partitioning, IEEE/ACM Transactions on Networking 13 (3) (2005) 540–553.

[31] A. Fei, J. Cui, M. Gerla, M. Faloutsos, Aggregated multicast: an approach to reduce multicast state, in: Proceedings of IEEE GLOBECOM, 2001.

[32] W. Mendenhall, Introduction to Probability and Statistics, third ed., Duxbury Press, 1997.

[33] J. Li, S. Kalyanaraman, Using Average Attenuation Factor to Locate the Most Congested Path for Multicast Congestion Control, Rensselaer Polytechnic Institute, Computer Science Department. Available from: <http://www.cs.rpi.edu/~lij6/Research/papers.html>, Tech. Rep.

[34] The Network Simulator – ns-2. Available from: <http://www.isi.edu/nsnam/ns>.

[35] K. Fall, S. Floyd, Simulation-based comparisons of Tahoe, Reno, and SACK TCP, ACM Computer Communications Review 26 (3) (1996) 5–21.

[36] I.E. Khayat, G. Leduc, A stable and flexible TCP-friendly congestion control protocol for layered multicast transmission, in: Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems, in: Lecture Notes in Computer Science, vol. 2158, 2001, pp. 154–167.

[37] C.D. Carothers, D. Bauer, S. Pearce, ROSS: A high-performance, low memory, modular time warp system, in: Proceedings of 14th Workshop on Parallel and Distributed Simulation (PADS), 2000.

[38] H. Yousefizadeh, H. Jafarkhani, A. Habibi, Rate control in communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society 49 (1998) 237–252.

[39] K. Kar, S. Sarkar, L. Tassiulas, Optimization based rate control for multirate multicast sessions, in: Proceedings of IEEE INFOCOM, April 2001.

[40] S. Deb, R. Srikant, Congestion control for fair resource allocation in networks with multicast flows, IEEE/ACM Transactions on Networking (April) (2004) 274–285.

[41] N. Bonmariage, G. Leduc, A survey of optimal network congestion control for unicast and multicast transmission, Computer Networks (February) (2006) 448–468.

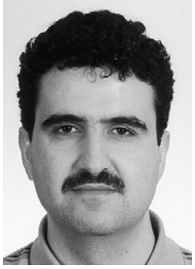
[42] J. Hagenauer, T. Stockhammer, C. Wei, A. Donner, Progressive source coding combined with regressive channel coding, in: Proceedings of 3rd ITG Conference Source and Channel Coding, 2000.

[43] J. Lu, A. Nosratinia, B. Aazhang, Progressive source-channel coding of images over bursty error channels, in: Proceedings of IEEE ICIP, 1998.

[44] H. Cai, G. Mirchandani, A new embedded image codec based on the wavelet transform and binary position coding, in: Proceedings of IEEE ICIP, 1996.

[45] K. Seada, A. Helmy, S. Gupta, A framework for systematic evaluation of multicast congestion control protocols, IEEE Journal on Selected Areas in Communications (December) (2004) 2048–2061.

Jiang Li is an Assistant Professor in the Department of Systems and Computer Science at Howard University, Washington, DC, USA. He received his B.S. and M.S. degree in Computer Science from the University of Science and Technology of China, Hefei, China in 1995 and 1998, respectively. In August 2003, he graduated from Rensselaer Polytechnic Institute in Troy, NY, USA with a Ph.D. degree in Computer Science. His research areas are in computer networking. In particular, he is interested in flow and congestion control, IP and application layer multicast, wireless networks, sensor networks, peer-to-peer networks, overlay networks and network security. He is currently conducting research on delay tolerant networks and multicast congestion control. He is a member of IEEE and ACM.



Murat Yuksel is currently an Assistant Professor at the CSE Department of The University of Nevada – Reno (UNR), Reno, NV. He was with the ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY as a Postdoctoral Research Associate and a member of Adjunct Faculty until 2006. He received a B.S. degree from Computer Engineering Department of Ege University, Izmir, Turkey in 1996. He received M.S. and Ph.D. degrees from Computer Science Department of RPI in 1999 and 2002 respectively. His research interests are in the area of computer communication networks with a special focus on experimental networking, such as wireless ad-hoc networks, large-scale network simulation and experiment design, mobile ad-hoc free-space-optical (FSO) networks, network economics, and performance analysis. He is a member of IEEE and Sigma Xi.

Xingzhe Fan received the B.E. and M.E. degrees from Tsinghua University, Beijing, China, and the Ph.D. degree from the Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY, in 1998, 2000, and 2004, respectively. He is currently a Visiting Assistant Professor in University of Miami, Miami, FL. His research interests are in nonlinear control and distributed optimization.

Shivkumar Kalyanaraman is an Associate Professor at the Department of Electrical, Computer and Systems Engineering at Rensselaer Polytechnic Institute in Troy, NY. He received a B.Tech. degree from the Indian Institute of Technology, Madras, India in July 1993, followed by M.S. and Ph.D. degrees in Computer and Information Sciences at the Ohio State University in 1994 and 1997, respectively. His research interests are in network traffic management topics such as congestion control architectures, quality of service (QoS), high-speed wireless, free-space optical networking, network management, multicast, pricing, multimedia networking, and performance analysis. His special interest lies in developing the inter-disciplinary areas between traffic management, wireless communication, optoelectronics, control theory, economics, scalable simulation technologies, and video compression. He is an member of the ACM and IEEE.