Distrust and Privacy: Axioms for Multicast Congestion Control

N.G. Duffield Matthias Grossglauser K. K. Ramakrishnan

AT&T Labs - Research, 180 Park Ave., Florham Park NJ 07932, USA

Abstract

Existing multicast control algorithms typically assume trust and free sharing of information among receivers. We believe that this is not tenable in the long term. We examine multicast congestion control under the constraints of distrust and privacy among receivers. We claim that despite these constraints, large-scale multicast congestion control in a heterogeneous environment is feasible.

We discuss three classes of solutions for the multicast congestion control problem, called *per-group feedback*, *selective participation* and *menu-adaptation*. For each class, we identify the limitations imposed by receiver distrust and privacy, and we argue that only a solution that integrates elements of each class can scale and adapt to heterogeneity in network and receiver characteristics. We outline such an integrated solution.

I. Introduction

A significant amount of work has gone on in the area of multicast congestion control that is based on cooperation and trust among all the participants. Algorithms enabling large scale multicast introduce a high degree of fate-sharing among mutually unknown parties. Furthermore, control protocols for managing multicast transport ignore important issues related to privacy. This Utopian picture of cooperation and mutual dependency for the success of multicast communication may not survive the commercialization of communication networks. As a result, we now look at the problem of multicast congestion control with a fresh set of axioms not involving any implicit assumptions of trust and privacy.

Trust would mean reliance on other participants for correctness and performance. For example, consider an electronic marketplace where information is multicast for efficiency reasons. A receiver may not share information with other receivers when needed. Even if it agrees to share the information, it may not do so in a timely or accurate manner. We believe an entity may be trusted to perform a certain action only if it is in its self-interest to do so.

Privacy means that a participant does not wish others to have any knowledge of his or her participation. For example, consider a corporation joining a multicast group. It may not wish other unknown entities to even know of its interest in the information that is multicast.

A prominent example of algorithms that implicitly assume trust and ignore privacy is the class of reliable multicast algorithms that rely on local repairs for efficient recovery from loss, and overcome the feedback implosion problem [14], [6], [10]. However, this requires mutual trust among receivers and reveals the identities of the receivers requiring and providing the repair.

In addition to taking a fresh look at the basis for the design of a multicast congestion control algorithm, we also observe that the set of algorithms that have been proposed so far have not been entirely successful in solving the difficult problem of multicast congestion control for sufficiently general applicability. In this submission, we propose an integrated set of three distinct mechanisms that together are likely to offer a generally applicable solution for the multicast congestion control problem. Furthermore, they satisfy our axioms of distrust and privacy for the design of a multicast congestion control algorithm.

The three mechanisms are:

- A per-group feedback mechanism that addresses the need of the members of a given group to have the right information to achieve efficient multicast.
- A selective participation mechanism that allows senders and receivers to selectively participate in a given group. For example, this allows a receiver to optimize the information it obtains in accordance with its capabilities, by joining or leaving groups based on its understanding of its utility.
- A menu-adaptation mechanism that allows senders to communicate appropriate information to its receivers. This is information shared by a sender with its receivers, in accordance with the principle of self-interest of an application in ensuring that its transmission is received by the participants in the most efficient manner feasible.

The next section describes existing proposals for multicast congestion control and observes their weaknesses. Subsequently, we provide a brief overview of our proposed approach to overcome these problems.

II. Overview of Multicast Congestion Control Solutions

We identify three general classes of solutions to the multicast congestion control problem.

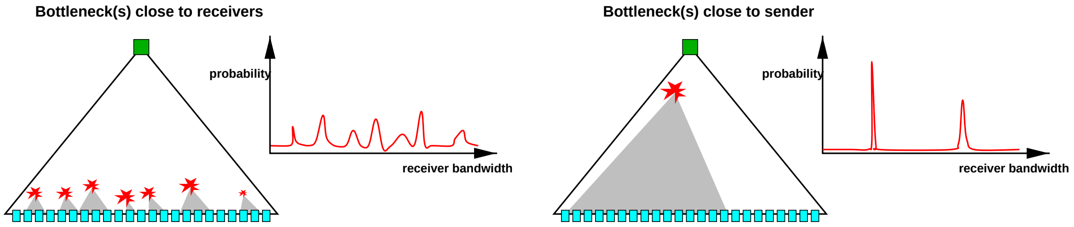
Bottleneck(s) close to receivers
probability
receiver bandwidth
Bottleneck(s) close to sender
probability
receiver bandwidth

Fig. 1: The location of the bottlenecks in the multicast delivery tree affects the distribution of the available bandwidth between the sender and the receivers. The scenario with multiple bottlenecks close to the receivers is problematic for per-group feedback. The scenario with a bottleneck close to the sender is problematic for selective participation.

- **Per-group feedback**, which has received the most attention in the research community, is an extension of the congestion control paradigm for unicast: the receivers provide feedback to the sender, from which the sender attempts to infer the properties of a "typical" path to a receiver. The sender adjusts its bandwidth in response, and possibly other parameters, such as the amount of forward error correction (FEC) information to use for error resilience.

As discussed earlier, we assume that receivers do not trust each other and do not want to reveal their participation in the group. This implies that they cannot cooperate to combine their feedback information in order to reduce the feedback processing load on the network and the sender, thereby eliminating the feedback implosion problem [5], [14]. RTCP uses multicast of receiver reports to estimate the group size to control the feedback load [8], [13]. Once again, this violates our axiom of privacy. Per-group feedback therefore requires a solution where receivers address their feedback solely to the sender, with possible network support to combine this feedback information when it flows upstream towards the sender.

Per-group feedback does not perform well if the path properties to the receivers are too heterogeneous. On the one hand, if we aim at satisfying all of the receivers in such a group, then the sending rate is effectively dictated by a single receiver. This is not acceptable in a large multicast group. On the other hand, if the sending rate is chosen according to some representative receiver instead of the worst receiver, then we potentially deny service to all of the receivers that have worse path properties. This number might be prohibitively large. A scenario where this occurs is illustrated in Figure 1: if multiple bottlenecks with different available bandwidths are located close to the receivers, then any choice of a typical receiver is suboptimal due to the large variance in the sender-receiver path properties.

- **Selective participation** reduces the degree of fate-sharing between receivers. Receivers have a choice of joining one among a set of multicast groups with different characteristics in terms of bandwidth requirements or error correction capabilities. A receiver sends a leave request to a group that has become unsuitable and a join request to the new group deemed suitable. The set of groups can either be *incremental* [9], used in conjunction with hierarchical coding, or mutually exclusive [3], where each group is self-contained. We note further that Selective participation is applicable only in selected situations where the receiver can still use a subset of the information sent, or when the sender actually transmits to multiple groups and each group is self-contained.

Solutions have been proposed that rely on receiver collaboration to decide which group or set of groups to join, by exchanging information about path properties from the sender to the receivers (this is called *shared learning* in [9]). Obviously, by doing so, receivers reveal their identity to each other, and group membership decisions are influenced by other receivers. This violates our non-cooperation assumptions. We must therefore assume that receivers solely rely on local observations of path properties to make group join and leave decisions.

Selective participation fails in the following scenario: if a multicast group has a bottleneck link close to the sender, i.e., shared by most or all of the receivers in the group (cf. Fig. 1), then a change in the bottleneck rate could only be accommodated by all of the receivers individually deciding to switch to a different group. Furthermore, for the switch to have an effect on the congestion at the bottleneck link, the "leave-group" signaling messages from all the receivers would have to be processed by the network to release the entire forwarding tree below the bottleneck. This is slow and poses a large burden on the signaling system.

- **Menu-adaptation** assumes that a "menu" of multicast groups represents the same information at different levels of quality. The control action consists of adjusting this menu to cover the path properties to all or to a majority of receivers. This adjustment is based on feedback from all the receivers of the session (the collection of groups). The operating points of the groups (e.g., their bandwidth or the amount of FEC information) are dynamically chosen to cover the diverse requirements of the receivers. Other control actions are the merging of several groups into one if their operating points become very similar over time, or conversely, the splitting of a group into several groups if the receiver characteristics within the group become too heterogeneous over time [4].

Solutions could be envisioned where receivers collude in a distributed manner to agree on an optimal menu. Again, this violates the non-cooperation assumption. Instead, the menu should be the responsibility of the sender, based on feedback from the receivers of all groups that make up the session.

Menu-adaptation on its own is not sufficient. It would amount to each receiver being subscribed to a fixed group, without a mechanism to switch groups on demand. Unless the path properties to all the receivers happen to be very stable, this would preclude a meaningful adjustment of the session menu. This is because the feedback of all the receivers in the session, regardless of their current group membership, are used collectively to determine the menu. If receivers cannot switch groups, this means that a receiver's feedback is likely to affect the wrong group.

We have discussed the three classes of solutions that can be brought to bear on the problem of multicast congestion control in a large-scale, heterogeneous setting, and the constraints imposed on the mechanisms because of our premise of distrust and privacy between receivers. In the next section, we outline a solution to this problem that comprises elements of per-group feedback, selective participation and menu-adaptation. Our goal is to demonstrate that scalable multicast congestion control does not require receiver cooperation. We note that previous work by Ammar et al. [4], [3], although not motivated by our axioms, embodies elements of our proposed solution framework.

III. Control Mechanisms and Protocols

Each of the three approaches described above require that the sender and the receivers are provided with sufficient information to control their operation. Our design premise excludes cooperation between the receivers; rather receivers are limited to reporting individual control information (such as their current menu selection, or measured packet loss) to the sender, but not to each other. Thus the sender has a distinguished role in collating reports from the receivers and inferring current network and adjusting its operation accordingly. The sender may communicate to the receivers aggregated information from the population of receiver that enables the receivers to enhance their control decisions.

The premise of non-cooperation between receivers has the consequence that although outward control from the sender can employ multicast, the return loop from the receivers must be unicast to the sender. Unless a mechanism exists in the network to aggregate this control information as it passes back up the multicast tree, the volume of such traffic must be controlled as the size of the multicast group increases, each receiver sending feedback at at (average) rate that is inversely proportional to the number of receivers in the group. In distinction with feedback mechanisms proposed in RTCP [8], our premise of non-cooperation between receivers excludes the possibility for receivers to infer the group size from multicast receiver acknowlegements; rather the feedback rate must be notified to the receivers by the sender, either explicitly or implicitly.

The detailed mechanism of per-group feedback used for rate control at short timescales is influenced by the particular transport employed. In ATM, for Available Bit Rate (ABR) multicast [1], [7], resource management (RM) cells are inserted at the sender in the multicast data stream every 32 cells. The RM cells contain a rate request by the sender; on the outward passage down the tree, network elements may adjust this rate downwards as dictated by their available resources. In the return passage up the tree, rates are aggregated by taking minima of rate from subtrees. This aggregation inherently scales for large groups; on the other hand the rate returned to the sender is the worst case rate over the group. In an IP network, Explicit Congestion Notification (ECN) [11] enables routers to notify congestion on their portion of an end-to-end path by setting a bit in the packet header. Feedback from the receivers to senders may be using a unicast channel. One way for the sender to control the feedback rate is to explicitly request feedback from receivers. An additional indicator that the sender sets on selected packets could be used to control the feedback rate by requiring that receivers return notification of congestion experienced in the forward path only on receipt of such an indication; or the response could be sent on a random timer in order to make feedback traffic less bursty. A variation of this approach is to encode the desired average feedback rate as an agreed function of the inter-request time. Whichever mechanism is used, the sender uses the distribution of congested paths in order to adjust the group rate. In addition, it may be feasible to exploit aggregation of the feedback information at branch points routers as suggested for multicast ABR, using the additional indicator set by the sender [1]. On seeing the indicator in a packet, the intermediate router collects feedback from its branches and sends back an aggregated indication of congestion up the multicast tree. A conservative sender response would be to adjust the rate downwards while any receiver suffers congestion and upwards when no receiver suffers congestion.

Additional feedback between sender and receiver at longer timescales facilitates menu-adaptation by the sender, and selective participation in groups by the the receivers. We envisage the feedback can be more detailed (and less frequent) than in per-group feedback; accordingly we assume the sender can notify a mean rate for a random timer for sender feedback. The sender uses feedback to optimize the menu. Opposing factors in the menu optimization are (i) coverage: having the set of group rates approximately match the set of rates at which receivers has obtain desired quality; and (ii) economy: limiting duplication of content over different groups. Receiver feedback can notify loss rates as in RTCP; or potentially a bitmap encoding the sequence numbers of packets received. It has recently been shown how the latter information can be used to reconstruct the logical multicast tree, bottleneck bandwidths [12] and associated loss rates on logical links [2]. Menu tracking of bottleneck rates near the source will be an effective method of maintaining coverage, and will decrease leave/join operations between groups for receivers behind such a bottleneck. In distinction, receivers behind bottlenecks closer to the leaves may expect slower or even no response in the menu to their consistent notification of congestion or loss to the sender. In order that the receiver act quickly according to whether a bottleneck is low or high in the tree, we propose that the receiver set a timer when loss increases beyond a given threshold. If the timer expires before the current group rate adjusts downwards, then the receiver should leave the group and join one with a lower rate.

IV. Summary

We believe that the practical issues of trust and privacy raised in this submission are important. Because the nature of multicast does not afford much control for a participant on who else is receiving the information being multicast, it is even more critical that algorithms for control do not implicitly violate these principles of trust and privacy. Further, as we observed, no one mechanism proposed in the literature for multicast congestion control is likely to be adequate for the general case. We believe the integration of the three schemes of Per-group feedback, Selective participation, and Menu-adaptation offers a promise of being the basis for a robust, and general solution for multicast congestion control. In addition, the approach we outline ensures that participant privacy is maintained, and the only principle that is relied upon is that of participant self-interest.

References

[1] ATM Forum Traffic Management Specification Version 4.0. ATM Forum Specification /af-tm-0056.000, ATM Forum, April 1996. available as: ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.pdf.

[2] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, and T. Bu. Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation. In *Proc. IEEE Infocom '99*, New York, March 1999.

[3] S. Y. Cheung, M. Ammar, and X. Li. On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. In *IEEE INFOCOM '96*, San Francisco, Calif., USA, March 1996.

[4] S. Y. Cheung and M. H. Ammar. Using Destination Set Grouping to Improve the Performance of Window-controlled Multipoint Connections. *Computer Communications Journal*, 19:pp. 723–736, 1996.

[5] J. Crowcroft and K. Paliwoda. A Multicast Transport Protocol. In *Proc. ACM SIGCOMM '88*, pages 247–256, August 1988.

[6] Sally Floyd, Van Jacobson, Steven McCanne, Ching-Gung Liu, and Lixia Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. In *Proc. ACM SIGCOMM '95*, August 1995.

[7] M. Grossglauser and K.K. Ramakrishnan. SEAM: Scalable and Efficient ATM Multicast. In *IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[8] R. Frederick H. Schulzrinne, S. Casner and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC 1889, available from: ftp://ftp.isi.edu/in-notes/rfc1889.txt*, January 1996.

[9] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. In *Proc. ACM SIGCOMM'96*, pages 117–130, Stanford, CA, Sept. 1996.

[10] Sridhar Pingali, Dow Towsley, and James F. Kurose. A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols. In *Proc. ACM SIGMETRICS '94*, pages 221–230, May 1995.

[11] K. K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. *RFC 2481, available from: ftp://ftp.isi.edu/in-notes/rfc2481.txt*, January 1999.

[12] S. Ratnasamy and S. McCanne. Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements. In *Proc. IEEE Infocom '99*, New York, March 1999.

[13] Jonathan Rosenberg and Henning Schulzrinne. Timer Reconsideration for Enhanced RTP Scalability. In *IEEE INFOCOM '98*, San Francisco, Calif., USA, March 1998.

[14] W. T. Strayer, B. J. Dempsey, and A. C. Weaver. *XTP: The Express Transfer Protocol*. Addison-Wesley (Reading, Mass.), 1992.