Multicast Congestion Control with Distrusted Receivers

Sergey Gorinsky, Sugat Jain, and Harrick Vin

Technical Report TR2002-25
Department of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124, Austin, TX 78712, USA

{gorinsky, sugat, vin}@cs.utexas.edu

September 5, 2002

ABSTRACT

Congestion control protocols rely on receivers to support fair bandwidth sharing. However, a receiver has incentives to elicit self-beneficial bandwidth allocations and hence may manipulate its congestion control protocol. Whereas the issue of receiver misbehavior has been studied for unicast congestion control, the impact of receiver misbehavior in multicast remains unexplored. In this paper, we examine the problem of fair congestion control in distrusted multicast environments. We classify standard mechanisms for multicast congestion control and determine their potential vulnerabilities to receiver misbehavior. Our evaluation of prominent multicast protocols shows that each of them is susceptible to attacks by a misbehaving receiver.

1. INTRODUCTION

Existing protocols for congestion control rely on receivers to support fair bandwidth allocation and assume that receivers always act according to the design specification. This assumption is not tenable in the Internet. While information sources and network providers have an interest in fair delivery of the information to all their clients, an individual client is interested in maximizing its own throughput. Thus, receivers have incentives to exceed their fair bandwidth shares at the expense of competing traffic. Moreover, open-source operating systems provide misbehaving receivers with means to manipulate congestion control protocols.

In unicast congestion control, the receiver notifies the sender about the congestion status. Based on this feedback, the sender adjusts its transmission. According to recent studies of TCP, a misbehaving receiver can abuse its feedback to inflate transmission and acquire an unfairly high throughput [5, 15]. In proposed solutions, the sender protects against the misbehavior by verifying the feedback correctness.

In comparison to unicast, multicast receivers have additional incentives to violate congestion control protocols: if a misbehaving receiver gains an unfair bandwidth advantage over other receivers in *the same* multicast session, the receiver secures an unfair edge over the entities interested in *the same* information. Nevertheless, we are not aware of any prior studies of receiver misbehavior in multicast congestion control.

Two differences between multicast and unicast are pertinent to congestion control:

- *Receiver Multiplicity.* If each multicast receiver reports its congestion status directly to the sender, the feedback from a large session can overwhelm the network or the sender. To avoid the feedback implosion, scalable feedback-driven protocols employ an additional mechanism to suppress or aggregate the feedback. Also, the sender of a scalable multicast session is not aware of the receiver identities.

- *Receiver Heterogeneity.* If a multicast session has receivers with heterogeneous capabilities, transmission at a single rate does not fully accommodate all the receivers. Some protocols compose a session from several multicast groups and assign the receivers to the groups according to the receiver capabilities. In such protocols, subscription to a multicast group constitutes a congestion control mechanism.

The additional mechanisms of feedback suppression, feedback aggregation, and group subscription are a source of *additional vulnerabilities* in multicast congestion control. For example, a misbehaving receiver of a multi-group session can acquire an unfairly high bandwidth by maintaining an unfairly high subscription. Feedback-driven multicast protocols also face new types of receiver misbehavior: the misbehaver can elicit an unfairly high transmission by failing to report or by suppressing legitimate reports from other receivers. Note that verification of feedback correctness at the sender does not protect against inflated subscription or incomplete feedback. Thus, *unicast-style protection does not solve the harder problem of multicast receiver misbehavior.*

In this paper, we examine distrusted environments where a multicast receiver can manipulate its congestion control protocol to elicit a self-beneficial bandwidth allocation. We classify existing mechanisms for multicast congestion control and determine their potential vulnerabilities to receiver misbehavior. Our evaluation of prominent multicast protocols shows that each of them is susceptible to attacks by a misbehaving receiver.

Note that the examined problem is different from denial-of-service attacks where a misbehaving receiver is not interested in exceeding its fair bandwidth share. Such a misbehaver enjoys a richer arsenal of disruptive actions. For example, a misbehaving receiver can waste bottleneck bandwidth by transmitting spurious data to legitimate or fabricated sessions. This wastage prevents well-behaving parties from delivering their data at fair rates. Since opportunities for purely destructive misbehavior are more opulent, denial-of-service attacks present a greater challenge.

The rest of this paper is structured as follows. In Section 2, we review multicast congestion control mechanisms. Section 3

Paradigms	Mechanisms	Protocols		
		Single-group	Feedback-free	Multi-group feedback-driven
Feedback-driven transmission adjustment	Feedback generation	RMTP, SAMM, TFMCC, pgmcc		DSG, SIM, MLDA
	Feedback aggregation	RMTP, SAMM		SIM
	Feedback suppression	TFMCC, pgmcc		DSG, MLDA
Group membership regulation	Group subscription		RLM, RLC, FLID-DL, WEBRC	DSG, SIM, MLDA
	Subscription synchronization		RLM, RLC, FLID-DL, WEBRC	DSG, SIM, MLDA

Table 1: Classification of multicast congestion control protocols.

presents our threat model. Section 4 evaluates existing designs experimentally. Section 5 analyzes our findings. Finally, Section 6 contains a summary of the paper.

2. CONTROL MECHANISMS

To be scalable, feedback-driven multicast protocols limit the amount of feedback to the sender. Aggregation and suppression are two alternative mechanisms for providing the sender with a brief summary of the session congestion status.

In *feedback aggregation*, receivers pass their reports up along the edges of a logical tree rooted at the sender. Internal nodes of the tree reduce the amount of the feedback by consolidating the provided information: each internal node gathers reports from its subtree, compiles their summary, and transmits a new report with the aggregated information towards the root. Various implementations of feedback aggregation have been proposed. Some protocols – such as RMTP [13] – build the aggregation tree entirely from receivers. Schemes like SAMM [19] aggregate feedback in routers or other network devices.

In *feedback suppression*, a receiver reports its status directly to the sender. Unlike feedback aggregation, this mechanism does not rely on intermediaries to generate new reports with aggregated information. Instead, feedback suppression filters out those reports that do not refine the current summary of the session congestion status. For example, in TFMCC [20] where the congestion summary is the fair rate for the slowest receiver, the sender multicasts its current summary to the session and thereby cancels reports from the receivers with higher fair rates. Multicast of the congestion summary is not the only implementation of feedback suppression. Some protocols – such as pgmcc [14] – suppress feedback at routers: a router discards reports that do not refine the feedback forwarded by this router earlier.

To address receiver heterogeneity, multicast protocols compose a session from several multicast groups. By joining and leaving the groups through IGMP [6], each receiver controls its level of participation in the session. In such multi-group protocols, *group subscription* becomes a congestion control mechanism. In fact, RLM [10], RLC [18], FLID-DL [1], and WEBRC [9] provide no feedback to the sender and control congestion through regulation of group membership.

Fairness of bandwidth allocation in a multi-group session depends on the ability of a receiver to converge to its fair subscription level. To facilitate this convergence, some multicast congestion control protocols incorporate a mechanism for *subscription synchronization*. Once again, there exist different implementations of this mechanism. In RLM, receivers coordinate their actions via so-called shared learning: before subscribing to a group, a receiver announces its intention to the other receivers. RLC and FLID-DL synchronize subscriptions through explicit signals from the sender: a receiver can add a group only upon an increase signal; increase signals are sent less frequently to receivers with higher subscription levels. Receivers in WEBRC coordinate their subscriptions by converging to rates derived from an equation for TCP-friendly throughput [12].

While group membership regulation and feedback-driven transmission adjustment constitute two different paradigms for multicast congestion control, they are not mutually exclusive. Combining these paradigms in one design improves fairness and efficiency of bandwidth allocation in heterogeneous multicast environments [4, 8]. DSG [2, 3], SIM [7], and MLDA [16] are multi-group feedback-driven protocols that adjust both membership and transmission rates of the groups.

Table 1 classifies the mentioned prominent multicast protocols with respect to their congestion control mechanisms.

3. THREAT MODEL

We define a threat as a general pattern of multicast receiver misbehavior that can reward the misbehaver with an unfair bandwidth advantage over other receivers in the network. To create our threat model, we examine multicast congestion control mechanisms and determine their potential vulnerabilities.

The paradigm of feedback-driven transmission adjustment engages multicast receivers in providing the sender with a summary of the session congestion status. The sender uses this information to adjust its transmission. By distorting the congestion summary, a misbehaving receiver can trick the sender into unfairly high transmission. After the inflated transmission forces well-behaving cross traffic to recede, the misbehaving receiver unfairly acquires the released bandwidth. This general attack of inflated transmission comes in various instantiations that exploit different vulnerabilities in the control mechanisms of the feedback-driven paradigm.

Feedback generation intrinsically resides in receivers: each receiver prepares and transmits reports about its congestion status. To distort the congestion summary, a misbehaving receiver can issue *incorrect reports*. This threat is analogous to receiver misbehavior in unicast congestion control [5, 15]. However, incorrect reports are not the only threat to feedback generation in multicast. *Failure to report* can also boost transmission by distorting the congestion summary.

In feedback aggregation, each internal node of the aggregation tree replaces incoming feedback with a smaller number of aggregated reports. If the aggregation tree consists of receivers, a misbehaving receiver inside the tree can issue *forged aggregated reports* that ignore or falsify information provided to the misbehaver by other receivers.

Feedback suppression uses a report from a receiver to filter out subsequent feedback that does not refine this earlier report. *Manipulation with feedback suppression* through a spurious report can also distort the congestion summary.

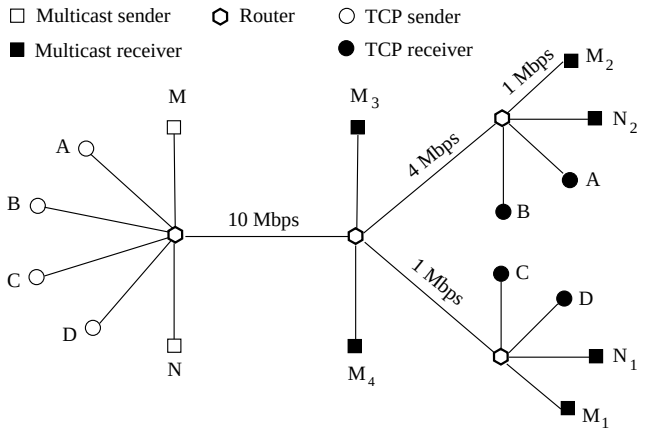

Figure 1: The network topology in our experiments.

In the paradigm of group membership regulation, group subscription allows a receiver to select its subscription level in a multi-group session. Since IGMP does not restrict multicast group membership, a misbehaving receiver can join those groups where transmission exceeds the fair rate for the misbehaver. The unfairly high subscription rewards the misbehaver with an unfairly high throughput after the competing well-behaving traffic recedes. Thus, *inflated subscription* poses a threat to fairness of multicast congestion control.

The mechanism of subscription synchronization coordinates actions of receivers to facilitate convergence to fair subscription levels. If a receiver's decision to join or to leave a group depends on information supplied by another receiver, a misbehaving receiver can manipulate the subscription levels of the others. By *preventing other receivers from subscription*, a misbehaving receiver keeps their subscription levels unfairly low and thus acquires an unfair bandwidth advantage over them.

To sum up the above discussion, we list the six threats of multicast receiver misbehavior: 1) *Incorrect reports*, 2) *Failure to report*, 3) *Forged aggregated reports*, 4) *Manipulation with feedback suppression*, 5) *Inflated subscription*, and 6) *Prevention of other receivers from subscription.*

In the next section, we use the proposed threat model to evaluate existing protocols for multicast congestion control.

4. EXPERIMENTS

4.1 Experimental Methodology

For each threat in our model, we evaluate one protocol from Table 1. Since our model defines threats with respect to control mechanisms, we select a representative protocol for a threat from the table row for the corresponding mechanism.

We use NS-2 [11] and conduct all our experiments in the same network. Figure 1 marks bottleneck links with their capacities. The capacity of each unmarked link is 100 Mbps. All the links have a delay of 10 msec and a buffer for two bandwidth-delay products. Multicast sessions M and N control congestion using the evaluated multicast protocol. Session M serves four receivers M_1, M_2, M_3 and M_4 that can misbehave. Well-behaving receivers N_1 and N_2 compose session N. Unicast sessions A, B, C, and D adhere to TCP Reno. Each sender transmits as much data as its protocol allows. The packet size in each session is 1000 bytes.

We run each simulation for 200 seconds. Unless we state explicitly otherwise, a misbehaving receiver starts its attack 100 seconds into the experiment. We measure throughput and loss rates for the misbehaver and other receivers. For reliable protocols, we consider only sequentially delivered data to compute the throughput. In unreliable protocols, the reported throughput reflects all delivered data.

4.2 Experimental Results

4.2.1 Incorrect reports in TFMCC

TFMCC [20] is a single-group protocol where each receiver uses an equation for TCP-friendly throughput to calculate its fair rate. The sender adjusts its transmission to the lowest of the fair rates reported by the receivers.

The slowest receiver can attack TFMCC by reporting an exaggerated rate and boosting the transmission. However, the misbehaver does not benefit if the inflated transmission swamps its bottleneck link and causes persistent heavy losses. Also, the misbehavior does not raise the transmission beyond the smallest rate reported by a well-behaving receiver. To profit the most from the attack, the misbehaving receiver can adjust the reported exaggerated rate and maintain the fastest transmission that does not result in congestion.

In our experiment, M_1 is the only misbehaving receiver. The fair rate for M_1 is 250 Kbps. The slowest well-behaving receiver M_2 has a fair rate of 1 Mbps. After 100 seconds, M_1 misbehaves by reporting a rate of 900 Kbps. Figure 2a shows that the attack rewards M_1 with a substantial throughput advantage over well-behaving receivers C, D, and N_1. Figure 2b presents the corresponding loss rates.

4.2.2 Failure to report in TFMCC

To attack TFMCC, the slowest receiver can also choose to be silent and boost the transmission to the smallest rate reported by a well-behaving receiver. If the inflated transmission overloads its bottleneck link, the misbehaver detects the persistent losses and discontinues the attack as disadvantageous. In comparison to incorrect reports, failure to report gives the misbehaver less control over the transmission. However, if the sender in TFMCC would verify the correctness of reported rates, this verification would ward off attacks based on incorrect reports but could not protect against missing reports. Thus, failure to report can spring more potent attacks.

As in the experiment above, M_1 is the only misbehaver. After 100 seconds, M_1 does not report to the sender. Guided by reports from M_2, session M increases transmission to 1 Mbps and subdues the well-behaving cross traffic. Figure 3 presents throughput and losses for receivers C, D, N_1, and M_1.

4.2.3 Forged aggregated reports in RMTP

RMTP [13] is a reliable protocol that marks data packets with sequence numbers. Each receiver specifies lost packets in its feedback. RMTP designates some receivers to aggregate feedback from other receivers. Every designated receiver also retransmits lost packets to its children in the aggregation tree. To control congestion, the sender monitors the highest reported loss rate. If this loss rate exceeds a threshold, the sender cuts its transmission to a minimum. While the losses stay below the threshold, the transmission rate grows linearly.

A designated receiver can attack RMTP by failing to relay loss reports from its aggregation subtree. If the ignored reports belong to the slowest receivers, the sender boosts its transmission. In comparison to own distorted feedback, forged aggregated reports reward the misbehaver more and punish the others harsher. In the above attacks on TFMCC, the misbehaver can raise the transmission up to the fair rate for the slowest well-behaving receiver. This increase can be small. In the attack on RMTP, the fastest receiver can govern the trans-