

Multicast Congestion Control with Distrusted Receivers

Sergey Gorinsky, Sugat Jain, and Harrick Vin

Laboratory for Advanced Systems Research
Department of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124, Austin, TX 78712, USA

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

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 misbehavior 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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NGC'02, October 23-25, 2002, Boston, Massachusetts, USA.

Copyright 2002 ACM 1-58113-619-6/02/0010 ...\$5.00.

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.

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 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 misbehavior 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.

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 misbehavior. The unfairly high subscription rewards the misbehavior 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.

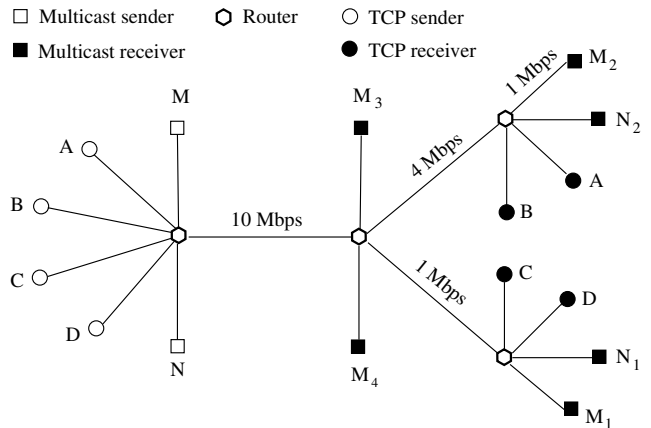


Figure 1: The network topology in our experiments.

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 misbehavior 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 misbehavior 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 misbehavior detects the persistent losses and discontinues the attack as disad-

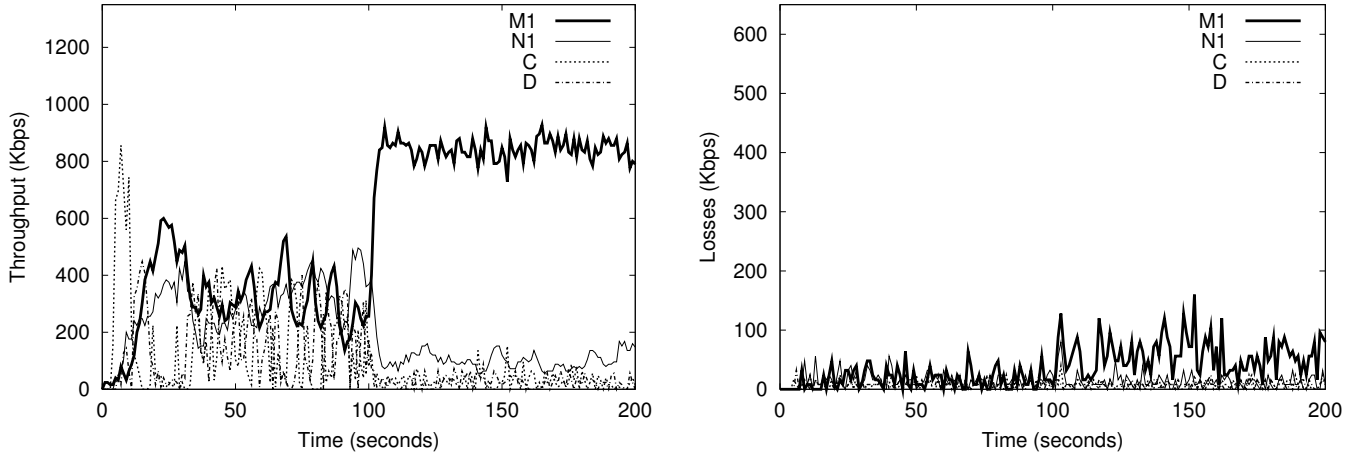


Figure 2: Incorrect reports in TFMCC.

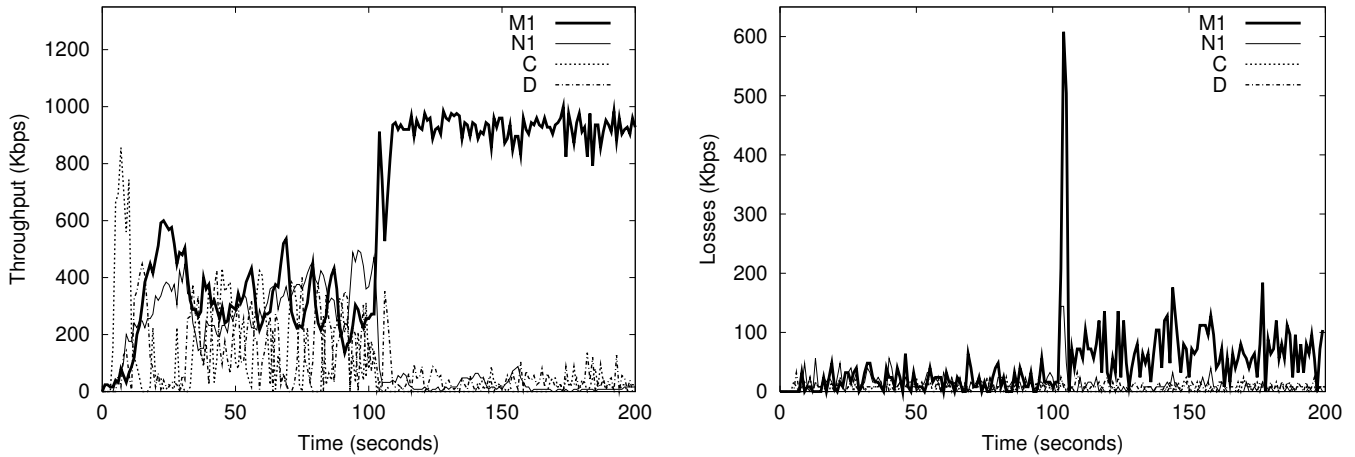


Figure 3: Failure to report in TFMCC.

vantageous. In comparison to incorrect reports, failure to report gives the misbehavior 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 misbehavior. 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 moni-

tors 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 misbehavior more and punish the others harsher. In the above attacks on TFMCC, the misbehavior 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 transmission by quenching the reports from the slower receivers. Furthermore, the inflated transmission can penalize the well-behaving receivers with heavy losses. To solidify the damage, the misbehavior can halt reliable delivery for the congested receivers by failing to retransmit the lost data.

We implemented RMTP following the description in [13]. In our experiment, designated receiver M_3 consolidates its

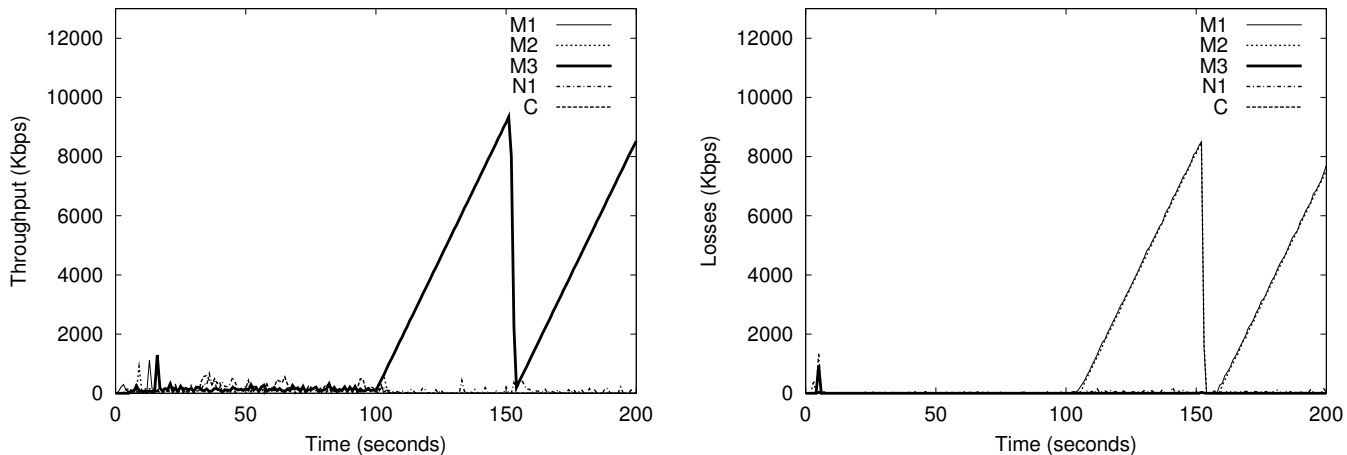


Figure 4: Forged aggregated reports in RMTP.

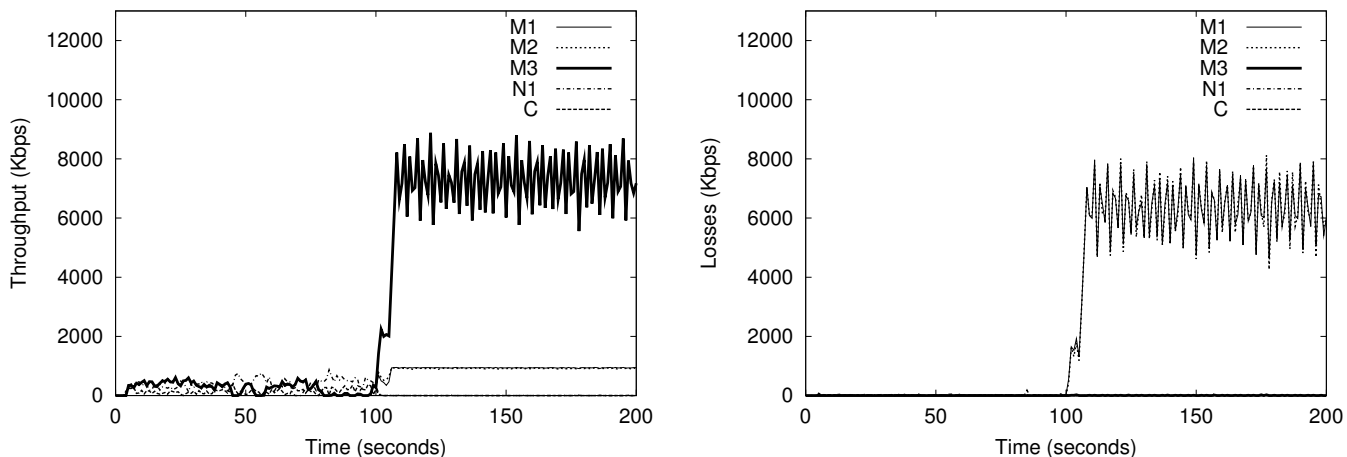


Figure 5: Manipulation with feedback suppression in pgmcc.

feedback with reports from M_1 and M_2 . The sender of M receives the aggregated feedback from M_3 and direct reports from M_4 . In session N , both receivers report directly to the sender. After 100 seconds, M_3 ignores loss reports from M_1 and M_2 . Figure 4 shows that this attack raises the transmission rate of M far above 1 Mbps and subdues well-behaving N_1 and C . Whereas M fills the 1 Mbps links with its data, M_1 and M_2 get skyrocketing losses and no throughput because M_3 does not retransmit lost packets to these receivers. Shortly after 150 seconds, the transmission rate of M saturates the 10 Mbps link, falls to a minimum upon a report from M_4 , and then inflates again.

4.2.4 Manipulation with suppression in pgmcc

pgmcc [14] is a single-group protocol that employs two types of feedback: NAK and ACK. Based on NAK feedback, the sender picks a receiver to represent the session. This receiver is called an acker and ideally should have the smallest fair rate. Based on ACK feedback from the acker, the sender adjusts its transmission. To support reliable multicast, the sender retransmits lost packets and controls the retransmis-

sion rate by a separate mechanism. Upon detecting a packet loss, a receiver transmits a NAK report that includes the sequence number of the lost packet, loss rate, and so-called lead parameter used by the sender to calculate the fair rate for the receiver. To avoid implosion of NAK feedback, pgmcc relies on feedback suppression at PGM routers [17]. For each sequence number, a PGM router forwards the first NAK report containing this number and discards subsequent reports with the same number. Feedback suppression does not interfere with the acker selection because slower receivers experience higher losses and transmit NAK reports more frequently. Thus, reports with the smallest fair rate have a good chance to reach the sender without being suppressed. Also, feedback suppression is likely to filter out NAK feedback from more capable receivers and thereby exclude them from being considered for the acker position.

A misbehaving receiver can attack pgmcc by issuing a spurious NAK report. To avoid suppression, the spurious report carries an exaggerated sequence number. The report also distorts the loss rate and lead parameter to trick the sender into calculating a tiny fair rate and selecting the mis-

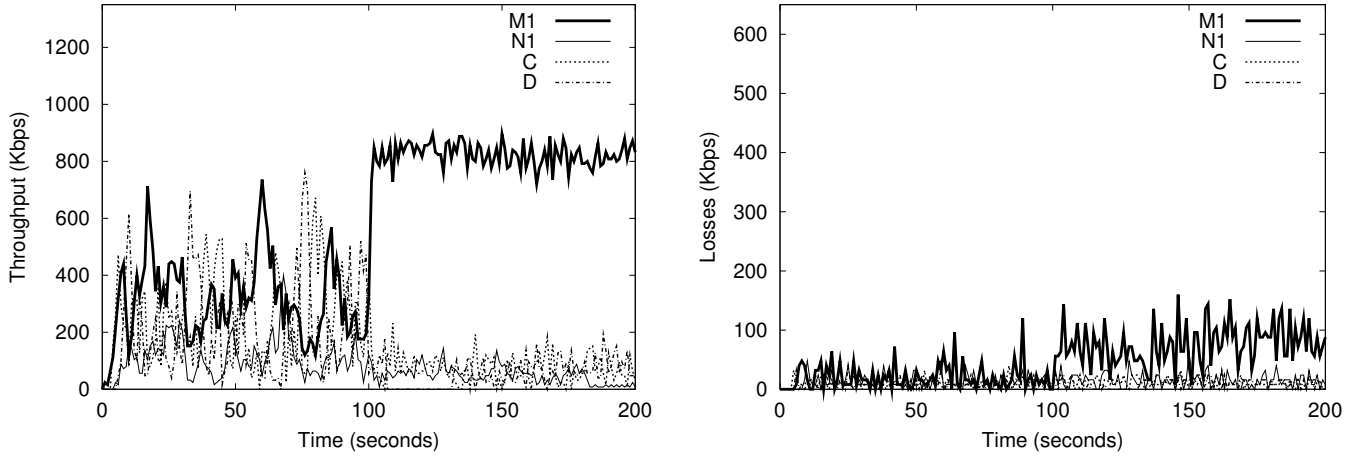


Figure 6: Inflated subscription in FLID-DL.

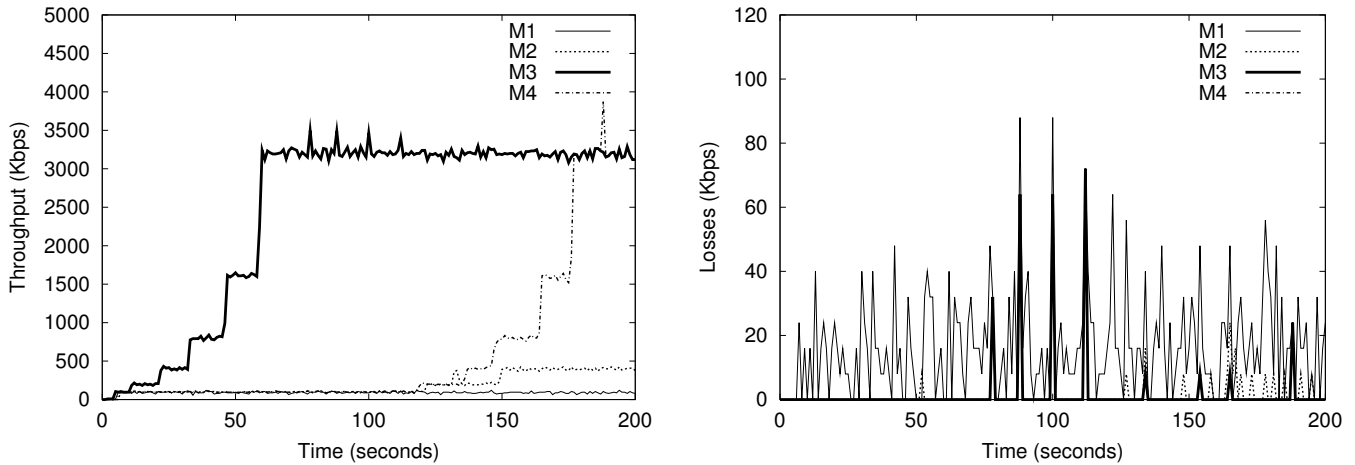


Figure 7: Prevention of other receivers from subscription in RLM.

behavior as the acker. Since the sender identifies the acker in each data packet, the misbehavior knows whether the attack is successful. After the misbehavior becomes the acker, its ACK feedback boosts the transmission to a level that can greatly exceed the smallest fair rate. Unlike in the above attacks where a receiver must be the slowest or designated to benefit from misbehavior, any receiver in pgmcc can fraudulently become the acker and inflate the transmission. Hence, this attack offers the highest probability of success.

In our experiment, all routers suppress NAK feedback. Upon receiving a data packet after 100 seconds, M_3 misbehaves by transmitting a spurious NAK report with 99.99% loss rate, lead parameter of 1, and sequence number $s+1000$ where s is the sequence number of the received packet. Since the sender is yet to transmit the packet requested in this NAK report, the report triggers no retransmission. However, the spurious NAK reports establish M_3 as the acker, and its correct ACK feedback inflates the sending rate of M beyond 8 Mbps. Figure 5 shows that the inflated transmission stomps throughput of well-behaving N_1 and C to zero and causes huge losses for M_1 and M_2 . Although M_1

and M_2 recover from the losses through retransmissions and maintain throughput of 1 Mbps, these receivers fall far behind M_3 in reliable acquisition of data.

4.2.5 Inflated subscription in FLID-DL

FLID-DL [1] is a multi-group feedback-free protocol where the sender encodes data into cumulative layers and uses a separate multicast group for each layer. Every receiver controls congestion by joining and leaving the groups of the session. Since the sender does not know the fair rates of the receivers, the default setting in FLID-DL uses a large number of multicast groups that cover – with a relatively fine granularity – the possible range of the fair rates.

To attack FLID-DL, a misbehaving receiver can join the layers with the cumulative transmission rate just below its bottleneck link capacity. The inflated subscription rewards the misbehavior with unfairly high throughput after the well-behaving cross traffic recedes. To detect the most beneficial subscription, the misbehavior can probe by adding a layer and keeping it only if the enhanced subscription does not cause persistent congestion.

Mechanisms	Threats	Vulnerable protocols		
		Single-group	Feedback-free	Multi-group feedback-driven
Feedback generation	Incorrect reports	RMTP, SAMM, TFMCC, pgmcc		DSG, SIM, MLDA
	Failure to report	RMTP, TFMCC, pgmcc		DSG, SIM, MLDA
Feedback aggregation	Forged aggregated reports	RMTP		
Feedback suppression	Manipulation with suppression	pgmcc		
Group subscription	Inflated subscription		RLM, RLC, FLID-DL, WEBRC	DSG, SIM, MLDA
Subscription synchronization	Prevention from subscription		RLM	

Table 2: Vulnerabilities of multicast congestion control protocols.

Each FLID-DL session in our experiment has the same parameter settings: the base layer is transmitted at 24 Kbps; data are encoded into 24 layers; the cumulative transmission rate grows multiplicatively with the factor of 1.3 per layer. After 100 seconds, M_1 joins 14 lowest layers of session M , maintains this inflated subscription, and raises its throughput to 800 Kbps. Figure 6 shows throughput and loss rates for receivers C , D , N_1 , and M_1 .

4.2.6 Prevention of other receivers from legitimate subscription in RLM

RLM [10] is also a multi-group feedback-free protocol where each group carries one layer of hierarchically encoded data. Every receiver maintains a join timer. When the join timer expires, the receiver adds the group that is immediately above its currently subscribed groups. To synchronize subscriptions, receivers rely on shared learning that sets the following rules:

- Before subscribing to a group, a receiver announces its intention to the other receivers.
- If a receiver observes losses shortly after subscribing to a group, the receiver drops the added group.
- When a receiver that is waiting to join a group receives a join announcement for a lower group, this receiver reschedules its join timer (to avoid derailing the announced join by the losses caused by its own join).

To attack RLM, a misbehaving receiver can periodically act as a newcomer. Its spurious announcements of joining the base layer prevent the other receivers from raising their subscriptions. Unlike the above attacks, this misbehavior gives the receiver an unfair edge over other receivers in the same session but not over receivers in a different session. Also, this attack succeeds only if it starts before the well-behaving receivers reach their fair subscriptions. However, the misbehavior can deflate these subscriptions by inflating its own. After the auxiliary misbehavior causes congestion and subdues the other receivers, the misbehavior can keep their subscriptions low. Thus, the receiver can combine these two attacks to maximize its benefits.

In our experiment, each RLM session encodes data into 7 layers, transmits the base layer at 100 Kbps, and doubles the cumulative transmission rate with each layer. Every second until the midpoint of the experiment, M_3 issues a spurious announcement of joining the base layer and thereby limits the subscriptions of M_1 , M_2 , and M_4 to this layer. After 100 seconds, M_3 stops its attack and allows the other

receivers of M to raise their subscriptions. Figure 7 presents throughput and loss rates for receivers M_1 , M_2 , M_3 , and M_4 .

5. ANALYSIS

Section 4 shows that each threat in our model victimizes at least one existing multicast protocol. Moreover, we observed that all the protocols from Table 1 are vulnerable to receiver misbehavior. Following the threat ordering in our model, Table 2 classifies the vulnerabilities of these protocols. Below, we discuss our findings in more detail.

Among the protocols in Table 1, SAMM [19] is the only feedback-driven design where a misbehaving receiver does not benefit from its failure to report. In SAMM, the sender transmits all layers of hierarchically encoded data to a single group. Every receiver reports its rate of raw data reception and a count of 1. Feedback is aggregated at routers or auxiliary network devices. An aggregation node reduces the number of reported rates to one per layer and enhances their counts with the counts of the ignored rates. The sender aligns its layer transmission rates with the reported rates.

The immunity of SAMM to failure to report comes from network support. All routers allocate the bandwidth of their links fairly among competing sessions and assign a larger forwarding priority to a lower layer within a SAMM session. At a bottleneck link, the SAMM session trims its rate to the fair share after the router discards the excessive higher layers. Whereas a misbehaving receiver cannot exceed its fair rate of raw data reception, feedback affects only the layer boundaries within this rate. Failure to report does not improve the alignment of the layer rates with the fair rate of the misbehavior. Hence, *the network can give receivers an incentive to supply feedback.*

Note however that SAMM is vulnerable to incorrect reports. By reporting inflated counts, a misbehaving receiver can elicit layer rates that match its capability exactly but are greatly unfair to other receivers in its session. Thus, *fair link scheduling is insufficient for comprehensive protection against multicast receiver misbehavior.*

Among the three protocols that use feedback aggregation, forged aggregated reports endanger only RMTP because SIM and SAMM aggregate feedback in the network. To protect receiver-based aggregation, a multicast protocol can employ feedback verification: if an aggregation node can detect that reports from its aggregation subtree are incorrect or incomplete, the protocol can curb the transmission to give receivers incentives to aggregate feedback properly.

Four protocols in Table 1 rely on feedback suppression but only pgmcc allows a misbehaving receiver to benefit from

manipulating this mechanism. Both pgmcc and TFMCC employ feedback suppression to provide the sender with the smallest fair rate. A misbehavior can deceive both protocols by reporting an even smaller rate. In TFMCC where the sender adjusts its transmission to the smallest reported rate, the misbehavior does not benefit from the deception. On the other hand, pgmcc uses the smallest reported rate to select the acker, and the same deception rewards the misbehavior with the acker position and an opportunity to inflate the transmission through correct ACK feedback. Thus, *if protection against receiver misbehavior relies on feedback verification, even the feedback that affects transmission indirectly should be verified.*

Two challenges complicate feedback verification in multicast. First, feedback can be presented in a compressed form such as a rate or an average. Second, not only the sender but also other receivers and network devices can react to feedback.

Let us now consider the paradigm of group membership regulation. Among the multi-group protocols in Table 1, only RLM allows a misbehaving receiver to subdue the subscriptions of other receivers in the same session. The rest of the protocols is immune to the threat because a receiver joins a group in these protocols without consulting with other receivers. Since RLC achieves the objectives of shared learning without network support, future protocols have *no reason to make subscription decisions dependent on information from other receivers.*

The ability of a receiver to join any multicast group represents a fundamental threat in distrusted multicast environments. All the multi-group protocols suffer from inflated subscription.

Protection against inflated subscription is a difficult task. Due to the scalability constraint, the sender can not track the receivers and their subscriptions. Since fair receiving capabilities can change frequently, enforcement of fair subscriptions should keep up with dynamic network conditions.

6. SUMMARY

In this paper, we examined the problem of fair congestion control in distrusted multicast environments. We classified standard mechanisms for multicast congestion control and determined their potential vulnerabilities to receiver misbehavior. Our evaluation of prominent multicast protocols showed that each of them is susceptible to attacks by a misbehaving receiver.

7. REFERENCES

- [1] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver. FLID-DL: Congestion Control for Layered Multicast. In *Proceedings NGC 2000*, November 2000.
- [2] S. Y. Cheung and M. H. Ammar. Using Destination Set Grouping to Improve the Performance of Window-controlled Multipoint Connections. *Computer Communications Journal*, 19:723–736, 1996.
- [3] S. Y. Cheung, M. H. Ammar, and X. Li. On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. In *Proceedings IEEE INFOCOM'96*, March 1996.
- [4] N.G. Duffield, M. Grossglauser, and K.K. Ramakrishnan. Distrust and Privacy: Axioms for Multicast Congestion Control. In *Proceedings NOSSDAV'99*, June 1999.
- [5] D. Ely, N. Spring, D. Wetherall, S. Savage, and T. Anderson. Robust Congestion Signaling. In *Proceedings IEEE ICNP 2001*, November 2001.
- [6] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, November 1997.
- [7] S. Gorinsky, K.K. Ramakrishnan, and H. Vin. Addressing Heterogeneity and Scalability in Layered Multicast Congestion Control. Technical Report TR2000-31, Department of Computer Sciences, The University of Texas at Austin, November 2000.
- [8] S. Gorinsky and H. Vin. The Utility of Feedback in Layered Multicast Congestion Control. In *Proceedings NOSSDAV 2001*, June 2001.
- [9] M. Luby, V.K. Goyal, S. Skaria, and G.B. Horn. Wave and Equation Based Rate Control Using Multicast Round Trip Time. In *Proceedings ACM SIGCOMM 2002*, August 2002.
- [10] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. In *Proceedings ACM SIGCOMM'96*, August 1996.
- [11] UCB/LBNL/VINT Network Simulator NS-2. <http://www-mash.cs.berkeley.edu/ns>, May 2002.
- [12] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings ACM SIGCOMM'98*, September 1998.
- [13] S. Paul, K. Sabnani, J.C. Lin, and S. Bhattacharyya. Reliable Multicast Transport Protocol (RMTP). *IEEE Journal on Selected Areas in Communications*, 15(3), April 1997.
- [14] L. Rizzo. pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme. In *Proceedings ACM SIGCOMM 2000*, August 2000.
- [15] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson. TCP Congestion Control with a Misbehaving Receiver. *ACM Computer Communications Review*, 29(5):71–78, October 1999.
- [16] D. Sisalem and A. Wolisz. MLDA: A TCP-friendly Congestion Control Framework for Heterogenous Multicast Environments. In *Proceedings IWQoS 2000*, June 2000.
- [17] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano. PGM Reliable Transport Protocol Specification. RFC 3208, <http://www.ietf.org/rfc/rfc3208.txt>, December 2001.
- [18] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proceedings IEEE INFOCOM'98*, March 1998.
- [19] B. Vickers, C. Albuquerque, and T. Suda. Source-Adaptive Multi-Layered Multicast Algorithms for Real-Time Video Distribution. *IEEE/ACM Transactions on Networking*, December 2000.
- [20] J. Widmer and M. Handley. Extending Equation-Based Congestion Control to Multicast Applications. In *Proceedings ACM SIGCOMM 2001*, August 2001.