Robust Congestion Control for Multicast: Challenges and Opportunities

Sergey Gorinsky, Sugat Jain, and Harrick Vin

Technical Report TR2003-02
Department of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124, Austin, TX 78712, USA
{*gorinsky, sugat, vin*}*@cs.utexas.edu*

January 2003

Abstract – *Trust is the foundation of most congestion control protocols developed and deployed in the Internet today. Unfortunately, with the growth of the Internet, the assumption of universal trust is no longer tenable. A communicating entity can misbehave to obtain a self-beneficial bandwidth allocation. Thus, design of congestion control protocols that are robust to such misbehavior has become an important research area. In this paper, we discuss the specific problem of designing robust congestion control for multicast in the presence of untrusted hosts. We examine IP and peer-to-peer instantiations of the multicast service. For both cases, we show that protection against host misbehavior is harder than in unicast and poses new research challenges. We outline possible solutions for designing robust multicast congestion control protocols. Further, we argue that intrinsically different design requirements imposed by untrusted environments point to the need for exploring an integrative alternative to the traditional layered network architecture.*

1. Introduction

Traditionally, network congestion control protocols have relied on trust and cooperation: protocols assume that each party always adheres to guidelines for sharing the network bandwidth fairly with competing traffic. Unfortunately, with the growth and commercialization of the Internet, the assumption of universal trust is no longer tenable. Different parties often have divergent interests. For example, an end host can be primarily interested in improving its own bandwidth allocation. Consequently, the end host has incentives to misbehave and acquire extra bandwidth at the expense of competing traffic. Furthermore, the widespread deployment of open-source operating systems provides end hosts with ample opportunities for such misbehavior. Consequently, design of congestion control protocols that are *robust* to such misbehavior is an important research area.

In this paper, we consider the problem of designing robust congestion control protocols for a *multicast service*. First, we review recent attempts at designing robust unicast congestion control protocols. Then, we focus on multicast services and argue that the design of robust congestion control protocols for multicast is fundamentally more difficult than for unicast. We substantiate our claims in the contexts of IP multicast and peer-to-peer multicast. We outline possible solutions for designing robust multicast congestion control protocols. Further, we argue that intrinsically different design requirements imposed by untrusted environments point to the need for exploring an integrative alternative to the traditional layered network architecture.

The rest of this paper is structured as follows. Section 2 reviews mechanisms for robust unicast congestion control with untrusted receivers. Section 3 explains fundamental differences between multicast and unicast. Section 4 shows impact of multicast receiver misbehavior. Section 5 outlines solutions for robust multicast congestion control. Section 6 concludes the paper with a discussion.

2. Unicast

In unicast congestion control, each receiver reports its congestion status to the sender. Based on this feedback or the lack of it, the sender adjusts its transmission rate.

Whereas traditional unicast protocols – such as TCP [1] – rely on the assumption of universal trust, Savage et al [12] consider an alternative trust model where information sources (i.e., senders) and network infrastructure (i.e., network links, routers, and servers) are

trusted but information consumers (i.e., receivers) may misbehave to elicit a self-beneficial bandwidth allocation. Figure 1 depicts this model by placing receivers outside a sphere of trust.

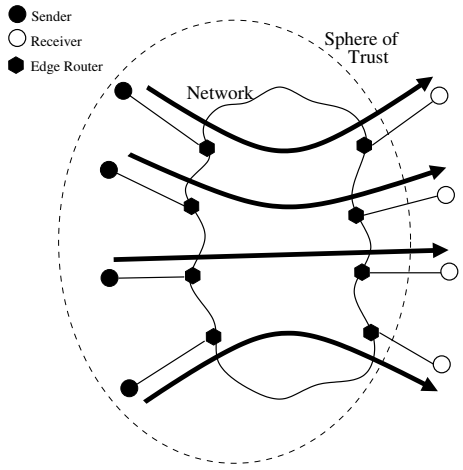

Figure 1: Unicast with untrusted receivers.

Studies of receiver misbehavior in TCP show that a receiver can abuse its feedback to inflate transmission and acquire an unfairly high throughput [7], [12]. In particular, the TCP receiver can benefit from issuing multiple acknowledgments upon receiving one packet as well as from acknowledging data segments that have not yet been received.

Proposed designs for robust unicast congestion control protect against the misbehavior by verifying the feedback correctness. The sender adds nonces to transmitted packets, and the receiver has to prove delivery of packets by including their nonces in its feedback.

3. Multicast versus Unicast

Multicast is a service for disseminating data to multiple receivers. A wide range of emerging distributed applications – such as dissemination of news and emergency alerts, multi-party interactive games, and video distribution – can greatly benefit from this service. The design of the multicast service and the degree of receiver involvement in congestion control are often governed by the following three considerations:

- *Scalability Considerations*. A scalable implementation of the multicast service cannot rely on direct unicast communication between the data source and each receiver. To disseminate data to a large population of receivers, the sender relies on a distribution hierarchy of intermediaries that duplicate and forward data to the receivers. Furthermore, if each multicast receiver reported its congestion status directly to the sender, the feedback from a large session could overwhelm the sender. To avoid the feedback implosion, scalable multicast protocols employ additional mechanisms to suppress or aggregate the feedback. Also, the sender of a multicast session is often not aware of the receiver identities.
- *Heterogeneity Considerations*. 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, receiver-driven group subscription constitutes a congestion control mechanism.
- *Deployment Considerations*. To minimize the dependence on router support, implementations of multicast services often employ receivers to perform control-path or data-path functions. In some designs, receivers themselves form the data distribution hierarchy (e.g., in peer-to-peer multicast [3]) while other solutions use receivers to address the feedback implosion problem (either by using receivers as intermediaries that aggregate feedback or by incorporating a feedback suppression mechanism at receivers).

For instance, feedback-free protocols for IP multicast [6] – such as FLID-DL [5] – control congestion via receiver-driven group subscription. Single-group feedback-based protocols – such as TFMCC [13] – rely upon receivers to implement feedback suppression mechanisms. Peer-to-peer multicast implementations [3], on the other hand, utilize unicast congestion control between each pair of communicating peers. However, they use receivers to form a data distribution hierarchy.

The increased involvement of multicast receivers in congestion control not only increases the forms of receiver misbehavior but also enhances the potency of the misbehavior. This makes the problem of dealing with untrusted receivers in multicast a much harder problem than in unicast. In what follows, we first describe a few experiments that demonstrate the impact of receiver misbehavior on multicast sessions and then discuss some approaches for designing robust multicast congestion control protocols for environments with untrusted receivers.

4. Impact of Receiver Misbehavior

Figures 2 and 3 depict, respectively, IP and peer-to-peer instantiations of the multicast service in the presence of untrusted receivers. In both settings, a misbehaving receiver can exploit multicast congestion control mechanisms to elicit self-beneficial bandwidth al-

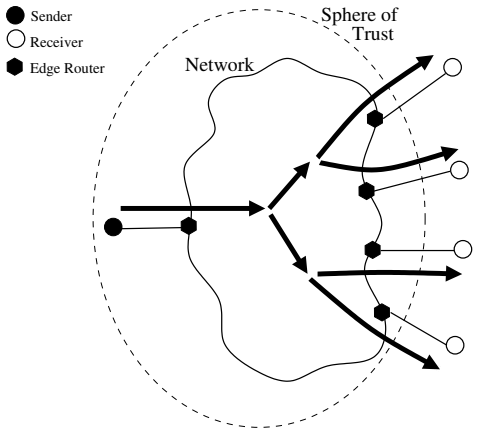

Figure 2: IP multicast with untrusted receivers.

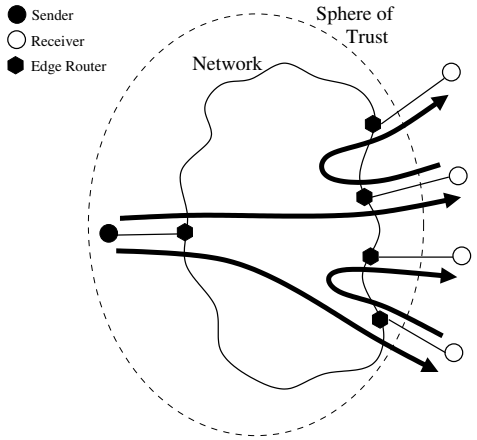

Figure 3: Peer-to-peer multicast with untrusted hosts.

location. We illustrate this with three examples. First, we consider a case where a feedback-based congestion control protocol relies on receivers to suppress feedback. Second, we consider a protocol where receivers use group subscription as a congestion control mechanism. Finally, we consider a peer-to-peer multicast service that builds the data distribution hierarchy from receivers.

We conduct experiments in NS-2 [10]. Figure 4 marks the bottleneck links of the simulated network 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 session M serves four receivers M_1, M_2, M_3 and M_4. Multicast session N has two receivers N_1 and N_2. Unicast sessions A, B, C, and D use TCP Reno. Each sender transmits as much data as its protocol allows. The packet

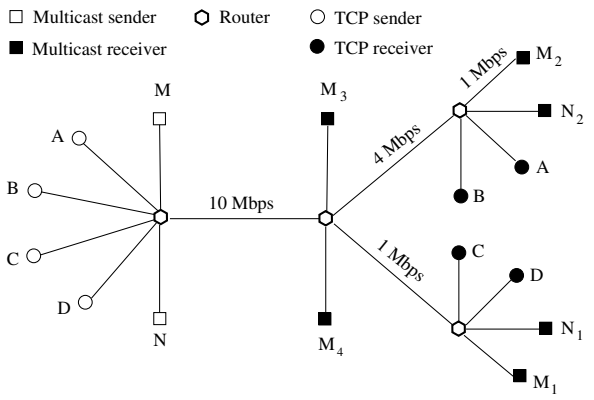

Figure 4: Network topology in our experiments.

size in each session is 1000 bytes. We run each simulation for 200 seconds. A misbehaving receiver from session M starts its attack 100 seconds into the experiment.

4.1. Attacks on Scalability Mechanisms

Consider TFMCC [13], a single-group congestion control protocol for IP multicast. TFMCC employs a receiver-based mechanism to suppress feedback. In TFMCC, each receiver uses an equation for TCP-friendly throughput to calculate its fair rate. The computed rate is then sent to the sender. To avoid feedback implosion, the sender multicasts the rate to the group. On receiving this announcement, a receiver sends a feedback to the sender *only* if its fair rate is lower than the announced rate. The sender adjusts its transmission rate to the lowest of the fair rates reported by the receivers.

The slowest TFMCC receiver can elicit a self-beneficial bandwidth allocation by suppressing its own feedback. Such a misbehavior results in the sender transmitting at the lowest reported rate, which is higher than the fair rate for the misbehaving receiver. In our experiment, the fair rates for receivers M_1 and M_2 are 250 Kbps and 1 Mbps respectively. After 100 seconds, M_1 misbehaves and does not provide feedback to the sender. Guided by reports from M_2, session M increases transmission to 1 Mbps. Figure 5 shows that M_1 gains an unfairly high throughput at their expense of subdued receivers C, D, and N_1.

4.2. Attacks on Heterogeneity Support

To address receiver heterogeneity, some multicast congestion control protocols group receivers according to receiving capabilities. FLID-DL [5] is such a multi-group protocol for IP multicast. To regulate congestion, every FLID-DL receiver joins and leaves the groups of its session by sending IGMP reports [8] to its local edge router. To attack FLID-DL, a misbehaving receiver can

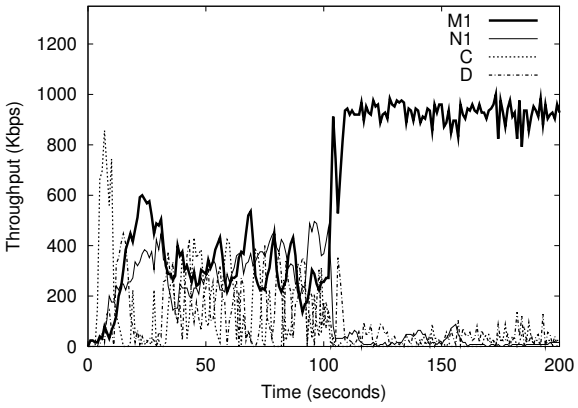

Figure 5: Suppressed feedback in TFMCC.

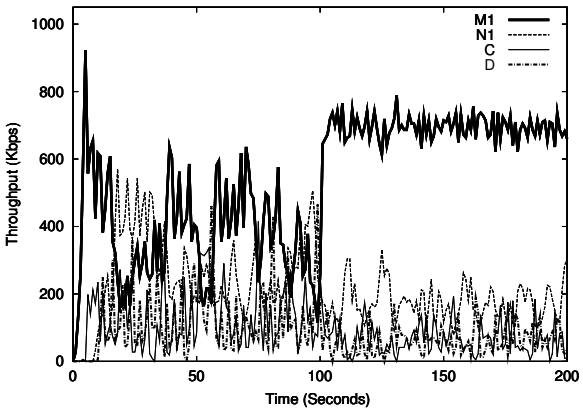

Figure 6: Inflated subscription in FLID-DL.

join the groups with the cumulative transmission rate exceeding the fair rate for the receiver.

In our experiment with FLID-DL, receiver M_1 misbehaves after 100 seconds and inflates its subscription in violation of the congestion control protocol. Then, as Figure 6 shows, M_1 enjoys an unfairly high throughput of 690 Kbps at the expense of subdued well-behaving receivers C, D, and N_1.

4.3. Attacks on Data Forwarding

The peer-to-peer instantiation of multicast requires no network support beyond unicast routing. Thus, congestion control can be done hop-by-hop and unicast-style protection techniques can be used to verify correctness of feedback along each hop.

However, many receivers in a large session obtain their data only after the data has traversed a number of untrusted intermediaries. Since an intermediary has a complete control over the amount of data it forwards to receivers down in the distribution hierarchy, a misbehaving host can forward the data at a lower than fair rate. Denial-of-service is not the only rationale for such misbehavior. By subduing the traffic to other receivers, the misbehaver reduces load on the network and thus can improve its own reception by acquiring the released bandwidth (e.g., when the bandwidth bottleneck for the misbehaving receiver is the semi-duplex wireless link connecting the receiver to the network).

5. Solution Directions

Section 4 showed that additional mechanisms needed for multicast congestion control are a source of vulnerabilities. We now outline possible directions for designing robust multicast services in the presence of misbehaving hosts.

5.1. Protection of Scalability Mechanisms

As the experiment with TFMCC shows in Section 4.1, scalability mechanisms – such as feedback suppression – offer misbehaving receivers opportunities to present the sender with an incorrect summary of the session congestion status. Robust congestion control designs should expose such manipulations. In particular, robust protocols should not allow silent membership in a multicast session. Each receiver *must* periodically prove its right for the multicast service by providing verifiable feedback. Due to the scalability considerations, the sender alone cannot carry burden of verifying all such feedback; assistance of other parties is required. Thus, there exists a fundamental need for integration of multicast congestion control with distributed group access control.

5.2. Protection of Heterogeneity Support

The experiment with FLID-DL in Section 4.2 shows that all feedback – including the feedback from receivers to edge routers – must be verified. One important consequence of this guideline in the context of IP multicast is a need for an alternative to IGMP. Edge routers must regulate access to multicast groups. Although there exist proposals for secure group access such as Gothic [9], these designs rely on receiver authentication and cannot enforce robust congestion control because the identity of a receiver does not reveal any information about its congestion status. Hence, the right to access a group should be a function of not only the identity but also the congestion status. Furthermore, since network conditions are dynamic, group access rights should also change over time.

A possible solution relies on dynamic congestion-dependent keys. The sender can periodically update the keys, partition them into components, and distribute the components among multiple packets so that a receiver

can reconstruct only the keys that the receiver is eligible to have according to the congestion control protocol. Thus, robust multicast with untrusted receivers needs an integrated approach to congestion control, dynamic group access enforcement, and secure key distribution.

5.3. Protection of Data Forwarding

To protect against misbehaving intermediaries in peer-to-peer multicast, a receiver *must* have some control over choosing the hosts on its route. This receiver-guided formation of the distribution hierarchy, however, faces the following challenges in untrusted environments:

- A misbehaving receiver may elicit a self-beneficial hierarchy at the expense of other receivers. For example, the slowest receiver can try to secure a direct unicast connection with the data source by displacing faster receivers to lower levels of the distribution hierarchy.
- Because of heterogeneous network conditions, it is difficult to determine whether the rate of data reception from a route is fair. Furthermore, disjoint routes to the same receiver can have different fair rates.

Due to the difficulties with detecting a misbehaving intermediary that forwards at a lower than fair rate [4], [11], an exciting research direction is a game-theoretic approach to forming the distribution hierarchy. Hosts in such optimization framework can choose their own last hop as well as advertise their services for forwarding the data to other hosts. Instead of detecting an unfair reception rate, a receiver tries to select its last hop to establish a route with the highest rate. To protect against false advertisements by misbehaving forwarders, a receiver can use its experience with chosen last hops to tune its degrees of trust in various hosts.

Regardless whether the "advertise-and-optimize" or "detect-and-punish" approach is chosen to tackle the forwarding misbehavior, a robust congestion control design for peer-to-peer multicast must involve receiver-influenced routing. Thus, features that traditionally belong to the transport layer (e.g., congestion control) has to be integrated with the functionality of the network layer (i.e., routing).

6. Discussion

In this paper, we considered the problem of robust congestion control for multicast in the presence of untrusted hosts. We examined IP and peer-to-peer instantiations of the multicast service. For both cases, we showed that protection against host misbehavior in multicast is harder than in unicast and poses exciting research challenges.

Our analysis of multicast congestion control supports recent assertions that robust Internet protocols should be designed based on non-traditional principles [2]. Although robust congestion control faces very different challenges in IP and peer-to-peer multicast, both cases reveal a common theme – a successful solution needs to bridge the gap between traditional network layers. For example, the paper presented evidence that a robust multicast service requires secure integration of routing, group access enforcement, and congestion control. These findings indicate a need for exploring an integrative alternative to the traditional layered network architecture.

References

[1] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.

[2] T. Anderson, S. Shenker, I. Stoica, and D. Wetherall. Design Guidelines for Robust Internet Protocols. In *Proceedings of HotNets-I*, October 2002.

[3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In *Proceedings of ACM SIGCOMM 2002*, August 2002.

[4] K.A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R.A. Olsson. Detecting Disruptive Routers: A Distributed Network Monitoring Approach. *IEEE Network*, 12(5):50–60, September/October 1998.

[5] 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.

[6] S.E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, December 1991.

[7] D. Ely, N. Spring, D. Wetherall, S. Savage, and T. Anderson. Robust Congestion Signaling. In *Proceedings IEEE ICNP 2001*, November 2001.

[8] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, November 1997.

[9] P. Judge and M. Ammar. GOTHIC: A Group Access Control Architecture for Secure Multicast and Anycast. In *Proceedings IEEE INFOCOM 2002*, June 2002.

[10] UCB/LBNL/VINT Network Simulator NS-2. http://www-mash.cs.berkeley.edu/ns, May 2002.

[11] R. Perlman. *Network Layer Protocols With Byzantine Robustness*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, May 1988.

[12] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson. TCP Congestion Control with a Misbehav-

ing Receiver. *ACM Computer Communications Review*, 29(5):71–78, October 1999.

[13] J. Widmer and M. Handley. Extending Equation-Based Congestion Control to Multicast Applications. In *Proceedings ACM SIGCOMM 2001*, August 2001.