Congestion Controlled Adaptive Lightweight Multicast in Wireless Mobile Ad Hoc Networks

Ken Tang[*] Katia Obraczka[+] Sung-Ju Lee[^] Mario Gerla[*]

[*]Computer Science Department, University of California, Los Angeles, CA 90095
[+]Computer Engineering Department, University of California, Santa Cruz, CA 95064
[^]Internet Systems & Storage Lab, Hewlett-Packard Laboratories, Palo Alto, CA 94304

Abstract

The use of contention-based MAC protocols combined with hidden terminal problems make multi-hop wireless ad hoc networks much more sensitive to load and congestion than wired networks or even wireless cellular networks. In such an environment, we argue that multicast reliability cannot be achieved solely by retransmission of lost packets as is typically done in wired networks with protocols such as SRM. We contend that in order to achieve reliable multicast delivery in such networks, besides reliability mechanisms, we must also consider jointly two components: reliability and congestion control. In this paper, we propose CALM, a congestion controlled, adaptive, lightweight multicast transport protocol and show that congestion control alone can significantly improve reliable packet delivery in ad hoc networks when compared to traditional "wired" reliable multicast protocols.

1. Introduction

Recent advances in portable computing devices and wireless communications technology have revamped the concept of mobile packet radio networks into what is called today *mobile ad hoc networks*. A mobile ad hoc network is defined as a multi-hop wireless network where all network components are capable of movements. Example applications include military battlefield scenarios, civilian disaster relief, emergency rescue, and exploration (e.g., hostile environments like space, under water, etc.) operations. The types of scenarios ad hoc networks target make group-oriented applications, such as teleconferencing and data dissemination services, one of the primary classes of applications. Multicast communication is certainly an efficient means of supporting group-oriented applications. In such an environment, we show that error control mechanisms alone do not lead to reliability, as is the case in wired networks. To achieve complete reliability, two complementary steps are required: congestion control and reliable delivery. In this paper, we focus on the *congestion control* facet (we will incorporate error and loss control techniques in our future work).

The remainder of this paper is organized as follows. Section 2 examines the motivation behind our research. Section 3 follows by describing the operations of CALM, a rate-based congestion controlled, lightweight multicast transport protocol. Simulation methodology and results are reported in Sections 4 and 5, respectively. Section 6 presents our concluding remarks.

2. Motivation

Reliable multicast transport protocols for **wired** networks have been a very active research area and the focus of the IETF Reliable Multicast Research Group (RMRG) [10]. Several reliable multicast protocols have been proposed, as surveyed in [6]. We hypothesize that the design choices underlying **wired** reliable multicast transport protocols are not adequate for **wireless ad hoc** network environments. Ad hoc networks are extremely sensitive to network load and congestion, even more so than wired shared-medium networks, such as Ethernet, because of the hidden terminal problem [8].

However, although reliable multicast is considered one of the fundamental technologies for enabling key applications in ad hoc networks, research in the area is minimal. A couple of exceptions are the Anonymous Gossip (AG) protocol [2] and the work on reliable *broadcast* by Pagani and Rossi [7] The former exhibits high recovery delay while the latter will likely degrade in dynamic ad hoc network scenarios where topology

Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)
1530-1346/02 $17.00 © 2002 **IEEE**

changes are frequent. In fact, simulation results reported in [7] indicate that the protocol does not perform well in the presence of high node mobility.

In our research, we investigate the effects of congestion control on reliable multicast communication in mobile ad hoc networks. We propose CALM (Congestion-controlled Adaptive Lightweight Multicast), a rate-based congestion control multicast transport protocol that adapts the source transmission rate to congestion in the network. We compare its performance against plain, unreliable UDP. We also include the SRM (Scalable Reliable Multicast) [3] protocol in our comparative study as an example protocol that employs error recovery, but not congestion control, to achieve reliable delivery.

3. Congestion-controlled Adaptive Lightweight Multicast (CALM)

CALM is a simple rate-based reactive congestion control multicast protocol designed for wireless ad hoc networks. As a reactive protocol, CALM transmits data packets at a rate specified by the application traffic until congestion arises. Congestion is indicated through negative acknowledgements (NACKs) sent back to the source by the multicast receivers that have not received a certain number of consecutive packets. Once the source is informed of congestion, it enters the congestion control phase. The source multicasts new "targeted" data packets, with each **new data packet** instructing a specific receiver to reply with an acknowledgement (ACK). Packets are multicasted to targeted receivers one at a time. The source then waits for an ACK from the targeted receiver or a timeout before moving to the next receiver. When all ACKs are received from the target receivers (an indication that the network is no longer congested), the source exits the congestion control phase and reverts to the initial application traffic rate. Thus, during the congestion control phase, the source clocks its sending rate based on the targeted receiver.

3.1. Detailed Description

Each multicast source maintains a *Receiver List*. Every time a source receives a NACK from a receiver that is not in the *Receiver List*, the receiver is added to the list. Nodes in the *Receiver List* are removed from the list when the source hears an ACK from them. In addition, the source keeps track of the end-to-end latency between itself and each receiver that sent NACKs (through a timestamp).

The source starts multicasting data packets at the rate specified by the application. Upon reception of NACKs, the source adds the receivers that sent the NACKs to the *Receiver List* and enters the congestion control phase. During congestion control phase, the source selects a receiver from the *Receiver List* and multicasts the next data packet with an indication in the header that instructs the target receiver to reply with an ACK. If the ACK is received from the receiver before a timeout (based on the measured end-to-end delay between the receiver and the source), the source presumes that receiver is no longer experiencing congestion. The receiver is removed from the *Receiver List* and the next node in the *Receiver List* is chosen for the next ACK reception. When all receivers in the *Receiver List* ACK the source, the source assumes the network is no longer congested and exits the congestion control phase reverting to the initial application traffic rate. If a timeout occurs before the ACK reception, the current target receiver remains in the list (moved to the end of the list) and the next receiver is selected. Thus, the source clocks its sending rate based on the response from the receivers.

A receiver unicasts a NACK to the source when it determines from the packet sequence numbers that it has not received N consecutive data packets from the source. In our experiments, N is chosen to be three since we do not want to perform congestion control for sporadic, occasional losses, such as the ones resulting from wireless medium transmission errors. Figure 2 depicts CALM's source state transition diagram.

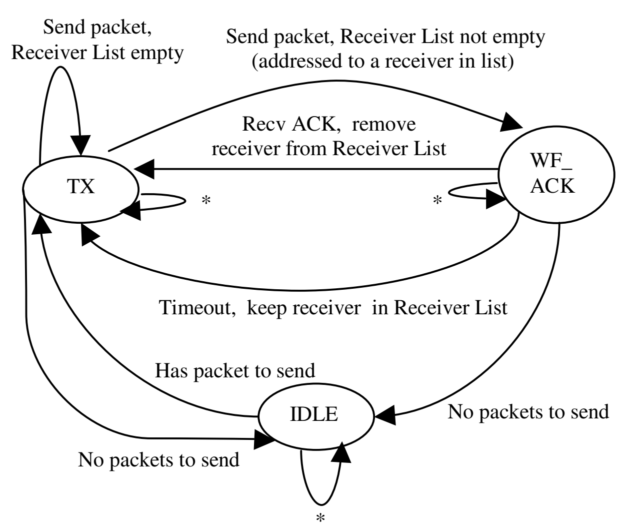

Figure 1. CALM state transition diagram at the source.

* In all states, when the source receives a NACK, the receiver associated with the NACK is added to Receiver List.

3.2. Observations

The proposed scheme reacts to every NACK sent to the source. As an alternate scheme, the receiver experiencing the longest latency can be used to trigger congestion control.

CALM does not attempt to multicast and/or suppress NACKs (as in SRM for example). Multicasting and suppressing NACKs has proven to work well in wired networks because packet losses are correlated; that is, packet losses observed by upstream nodes are also experienced by downstream nodes [3]. However, in ad hoc networks, losses are generally not correlated since there can be multiple paths to reach a destination. This is especially true in the case of using a mesh-structured protocol (e.g., On-Demand Multicast Routing Protocol [5], Core-Assisted Mesh Protocol [4]) as the underlying multicast routing protocol.

It is important to note that although using N consecutive packet losses as a criteria for sending NACKs applies well when high reliability is the goal, other alternatives are also possible. For instance, if an application operates on a less stringent reliability constraint, the receivers may opt to send NACKs only if the packet loss rate reaches a certain threshold or if the delay becomes intolerable. Other schemes may also apply. Thus, the triggering of NACKs and congestion control mechanism can be geared towards the deployed application.

Finally, it should be pointed out that CALM reduces the multicast delivery rate to what is acceptable to a set of slower, more congested receivers. This is appropriate in applications where all parties must have approximately the same quality of information to operate properly (e.g., audio and video resolution). In other applications, it may be more advantageous to selectively reduce the rate and degrade the video quality on congested paths by using layered encoding and selective dropping.

4. Simulation Methodology

To analyze the performance of CALM, we compare CALM against UDP and SRM running on top of the On-Demand Multicast Routing Protocol (ODMRP) [5]. UDP is chosen as it provides the basic multicast support without guaranteeing reliability; therefore, any reliable multicast algorithm should demonstrate improvements over UDP. SRM is elected because it employs error-, but not congestion control, to achieve reliable delivery. ODMRP is selected for this study as it is shown to perform well in ad hoc networks [5]. However, other ad hoc multicast routing protocols can also be used.

4.1. Simulation Platform and Parameters

We use the QualNet simulator [11], the commercial version of the GloMoSim [9], for performance evaluation. QualNet is a discrete event, parallel simulation environment implemented in PARSEC [1]. Table 1 through Table 4 list the parameters that are constant throughout our simulation. Parameters that vary are detailed in the appropriate sections.

Table 1. Simulation environment parameters.

Parameter	Value
Terrain	1500 *m* x 1500 *m*
Number of nodes	50
Node placement	Random
Multicast protocol	ODMRP
MAC protocol	IEEE 802.11 DCF
Bandwidth	2Mb/s
Propagation	Two-ray ground reflection
Maximum radio range	375 *m*

Table 2. SRM parameters.

Parameter	Value
C_1, C_2	2
D_1, D_2	1
Max repair request retry	4

Table 3. CALM parameters.

Parameter	Value
N	3

Table 4. ODMRP parameters.

Parameter	Value
Join Query refresh interval	3 seconds
Forwarding group timeout	3 seconds

The performance metrics we examine are packet delivery ratio, control overhead, end-to-end delay and total number of data packets received by multicast receivers. The packet delivery ratio is defined as the number of data packets received by the multicast members over the number of data packets the members are supposed to receive. This metric measures the effectiveness of a protocol (or how reliable it is). Control overhead is defined as the total number of all packets (data and control) transmitted by all the nodes in the routing and transport layer protocols over the number of data packets received by the multicast receivers. It is used to assess protocol efficiency. End-to-end delay measures the data packet latency from the source to the destination and evaluates the protocol's timeliness.

5. Results and Analysis

We present the performance of CALM by comparing it against UDP and SRM multicast under various ad hoc network scenarios by varying the traffic rate, number of receivers and mobility. As previously mentioned, the goal of comparing a wired reliable multicast transport (i.e., SRM) to CALM is to show that congestion control alone can outperform the use of pure error control techniques (without congestion control) in ad hoc network scenarios in achieving reliable delivery.

5.1. Traffic Rate

In the traffic rate experiments, we randomly choose five multicast sources and ten multicast receivers. All nodes are stationary in these experiments. We vary the "application driven" data packet interdeparture time at each source from 500 ms (2 packets per second) to 100 ms (10 packets per second). This is the maximum sending rate, i.e., the rate sources send if no congestion is experienced. During congestion, CALM lowers the sending rate based on its congestion control algorithm.

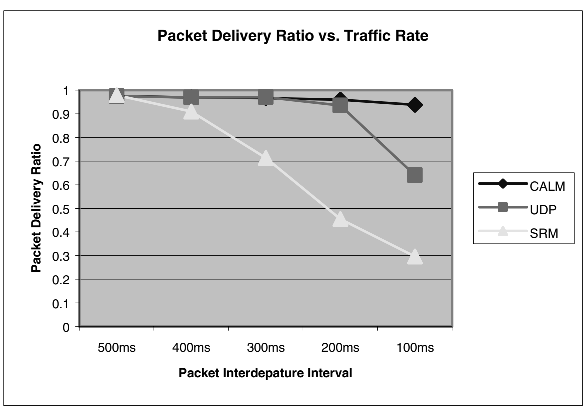

Figure 2. Packet delivery ratio as a function of traffic rate.

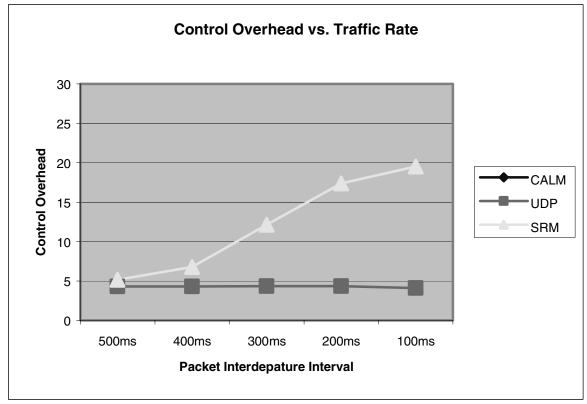

Figure 3. Control overhead as a function of traffic rate.

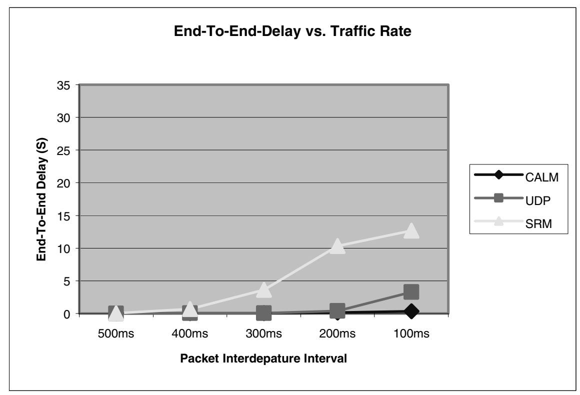

Figure 4. End-to-end delay as a function of traffic rate.

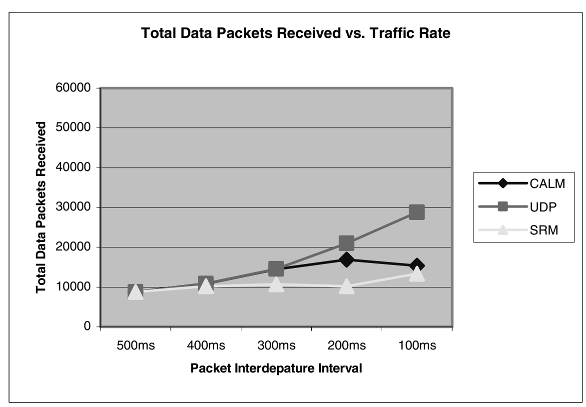

Figure 5. Total data packets received as a function of traffic rate.

We observe from Figure 2 that under light load (500 ms), SRM achieves near perfect reliability. However, as the traffic rate increases beyond 300 ms of interdeparture time, SRM performance starts to suffer and dramatically drops to approximately 30% packet delivery ratio. This degradation is due to the high contention the network experiences as the traffic rate and network load grow; high contention results in increased packet losses. This behavior is well known for CSMA protocols [8], which IEEE 802.11 is based on in the broadcast mode. Surprisingly, SRM performs even worse than UDP (which provides no reliability at all). As the network load increases, the packet loss grows and more SRM control messages are sent for recovery, as evident from Figure 3 (CALM and UDP exhibit approximately the same overhead, thus the CALM curve is hidden behind the UDP curve). SRM's poor performance is due to the fact that it attempts to recover the dropped packets through *repair request* and *repair*. The *repair request* and *repair* messages only add fuel to the fire; they simply contribute to more network congestion and therefore are counterproductive as the network is already saturated. Further analysis into the simulation statistics and trace files reveals that there are more packet drops from the queue maintained at each node as traffic rate increases under UDP and especially for SRM. CALM, on the other hand, shows resiliency to the network load by achieving a packet delivery ratio of 94% or higher under all traffic loads examined. CALM's robustness is due to congestion and rate control. Figure 5 shows that CALM achieves a throughput performance that is between UDP and SRM. CALM also outperforms both UDP and SRM in latency (Figure 4). SRM achieves the worst delay because of its lost packet retransmissions. Although neither UDP or CALM retransmit lost packets, CALM still provides better delay performance. As UDP does not adjust its sending rate, the queue size at each node is larger than in CALM, which leads to higher delay.

5.2. Number of Receivers

In these experiments, five sources are randomly chosen while the number of receivers is varied between 10 and 40. The packet interdeparture time is fixed at 500 ms for each source. Mobility is not considered. The results are shown below.

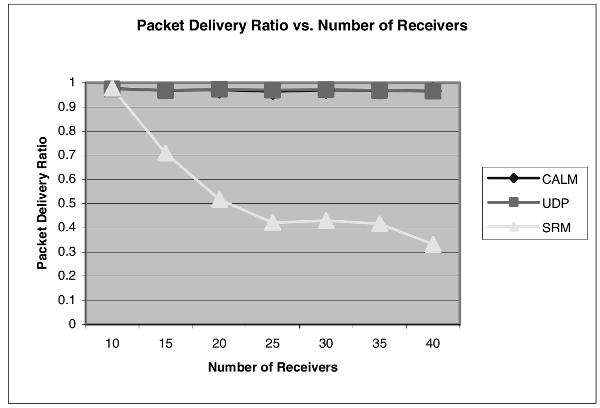

Figure 6. Packet delivery ratio as a function of number of receivers.

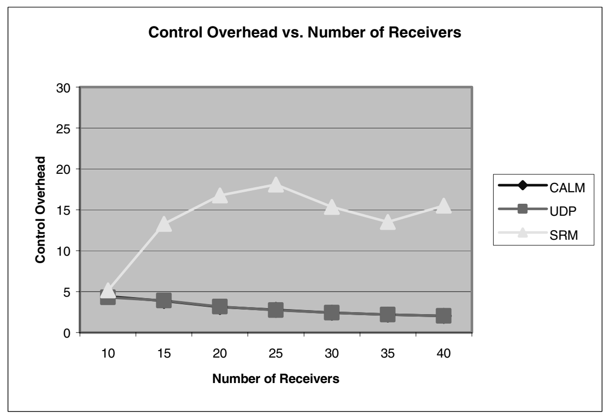

Figure 7. Control overhead as a function of number of receivers.

We observe that varying the number of multicast receivers has no impact on delivering data packets when using UDP. This finding is consistent with [5] where it was found that the performance of ODMRP under UDP is not affected by the multicast group size in the network due to the mesh created by ODMRP; the mesh allows for alternate routes, which leads to better reliability. CALM experiences approximately the same packet delivery ratio as UDP under all the metrics being investigated. Under this scenario, packets drops are rare, as evident by the high packet delivery ratio of UDP. Since the receivers in CALM transmit NACKs to the source only if `N` (chosen to be three) consecutive packet loss is determined, this scenario has little impact on CALM. However, the same is not true for SRM. We see that SRM performs worse as the number of receivers grows. As we have more receivers in the network, there is a higher probability that some data packets are dropped before reaching the destinations. Such condition is detrimental to SRM since it must use more control messages to perform recovery. The increase in the SRM control messages results in network congestion, packet drops, and poor network performance because it invokes further control message transmissions. Other metrics (not shown) in the number of receivers experiments are similar to those reported in Section 4

5.3. Mobility

In the mobility experiments, we randomly choose five sources and ten receivers, with a packet interdeparture rate of 500 ms. We vary the mobility speed from 0 m/s to 50 m/s. Based on [5], we expect that mobility would have little impact on UDP performance when using ODMRP because of the redundant transmission of the forwarding group mesh topology. The results confirm our expectation.

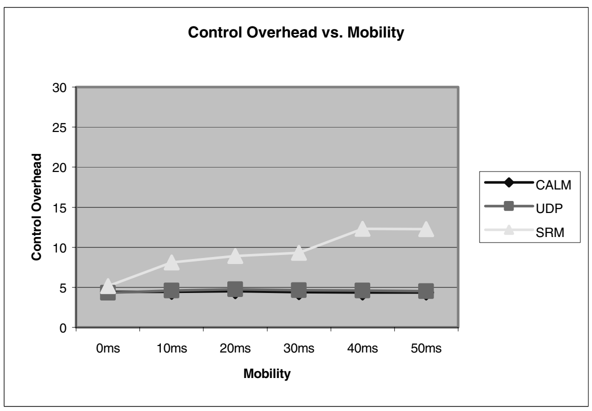

Figure 8. Control overhead as a function of mobility speed.

Mobility, similar to the number of receivers experiments, has little bearing on CALM and UDP. In addition to demonstrating worse packet delivery ratio (not shown), SRM displays the highest control overhead. CALM, however, competes favorably with UDP.

6. Conclusion and Future Work

We proposed CALM, a congestion control multicast transport protocol designed for ad hoc networks. Through simulation, we showed that by performing congestion control, CALM adapts well to the various network conditions observed in terms of achieving high packet delivery ratio and low end-to-end latency. We also demonstrated that SRM (designed to provide reliable delivery in wired networks) performed worse than CALM (and even UDP) in terms of packet delivery ratio, control overhead and end-to-end delay when deployed in ad hoc networks. The major offender is the SRM extra load introduced by the control packets needed to recover from losses.

Providing **congestion control**, we believe, is the **first step** in achieving **reliable multicast** in ad hoc networks. From the results in this paper, we note that CALM still does not guarantee 100% reliability. In general, some of the losses are unavoidable, as they may be caused by network by mobility or hidden terminal effects. Some form of end-to-end reliable retransmission mechanism specifically designed for wireless ad hoc networks is needed in conjunction with congestion control to achieve perfect reliability. We are currently continuing research in this direction.

7. References

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, “PARSEC: A parallel simulation environment for complex system,” *IEEE Computer Magazine*, vol. 31, no. 10, Oct. 1998, pp. 77-85.

[2] R. Chandra, V. Ramasubramanian, and K.P. Birman, "Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-Hoc Networks," *Proceedings of IEEE ICDCS 2001*, Mesa, AZ, Apr. 2001, pp. 275-283.

[3] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, “A reliable multicast framework for lightweight sessions and application-level framing,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, Dec. 1997, pp. 784-803.

[4] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, Aug. 1999, pp. 1380-1394.

[5] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols,” *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000, pp. 565-574.

[6] K. Obraczka, “Multicast Transport Mechanisms: A Survey and Taxonomy,” *IEEE Communications Magazine*, vol. 36, no. 1, Jan. 1998, pp. 94-102.

[7] E. Pagani and G.P. Rossi, "Providing Reliable and Fault Tolerant Broadcast Delivery in Mobile Ad-Hoc Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 4, no. 3, Aug. 1999, pp. 175-192.

[8] A.S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, New Jersey, 1996.

[9] X. Zeng, R. Bagrodia and M. Gerla, “GloMoSim: a Library for the Parallel Simulation of Large-scale Wireless Networks,” *Proceedings of PADS'98*, Banff, Canada, May 1998, pp. 154-161.

[10] http://www.ietf.org/html.charters/rmt-charter.html.

[11] http://www.scalable-networks.com.

1530-1346/02 $17.00 © 2002 **IEEE**