

# A Reliable, Congestion-Controlled Multicast Transport Protocol in Multimedia Multi-hop Networks

**Ken Tang**

Computer Science  
University of California  
Los Angeles, CA 90095

**Katia Obraczka**

Computer Engineering  
University of California  
Santa Cruz, CA 95064

**Sung-Ju Lee**

Internet Systems &  
Storage Lab (ISSL)  
Hewlett-Packard Labs  
Palo Alto, CA 94304

**Mario Gerla**

Computer Science  
University of California  
Los Angeles, CA 90095

## Abstract

*We consider the challenges of achieving reliable multicast in mobile ad hoc networks (MANETs). MANET environments can experience high packet loss rates because of node mobility and wireless signal interference. In addition, they are extremely sensitive to the network load due to the broadcast nature of the shared wireless medium. Thus, in designing a reliable multicast transport protocol, two components are essential – reliability and congestion control. We describe and evaluate the performance of the Reliable Adaptive Lightweight Multicast (RALM) protocol. We show through simulation that when subject to a wide range of network conditions (e.g., traffic load, number of multicast sources, and mobility), RALM achieves reliability, while exhibiting low end-to-end delay and minimal control overhead compared against those of UDP and SRM.*

## Keywords

Ad hoc networks, reliable multicast transport, congestion control, mobile computing, wireless networks

## INTRODUCTION

Mobile ad hoc networks (MANETs) [4] have no fixed infrastructure; in other words, there is no distinction between a host and a router. The types of scenarios targeted by MANETs (i.e., battlefield and disaster relief operations) make group-oriented multimedia services such as data dissemination and teleconferencing a key application domain. Multicast communication is one efficient means of supporting group-oriented applications. Despite the fact that reliable multicasting is important to the success of mission critical and multimedia applications in MANETs, surprisingly little work has been done in this area.

Reliable multicast for *wired* networks has been a very active area of research ([7][8][10][12]). One may consider applying them to MANETs. We argue that the design choices underlying wired reliable multicast transport protocols are not adequate for MANET environments; ad hoc networking protocols must handle node mobility and are extremely sensitive to network load and congestion because of the hidden terminal problem [14].

In the ad hoc domain, the work on reliable *broadcast* by Pagani and Rossi [9] addresses the problem of reliable atomic delivery of messages. While this protocol may work well in stable networks with low mobility and low failure rates, its performance will likely degrade in dynamic MANET scenarios where topology changes are frequent.

More recently, a gossip-based approach has been proposed to address reliable multipoint delivery. The Anonymous Gossip (AG) protocol [2] recovers from losses by having pairs of multicast group participants exchange information on messages they have received or lost. One potential problem with the AG protocol is the delay it takes for nodes to recover from losses.

In this paper, we focus on designing a multicast transport protocol that favors reliability and congestion control over throughput. Applications that are willing to trade throughput for reliability include military convert operations as well as search and rescue missions. To this end, we propose and evaluate the performance of the Reliable Adaptive Lightweight Multicast (RALM) protocol designed for MANETs. RALM employs TCP-like congestion and error control mechanisms. Through extensive simulations, we compare its performance in terms of reliability, cost, and timeliness against plain, unreliable (multicast) UDP, and the Scalable Reliable Multicast (SRM) protocol [3]. While UDP is an example of a protocol that employs neither congestion nor error control, SRM represents protocols that use solely error control. Our results demonstrate that RALM performs consistently better in achieving high packet delivery ratio and is favorable in regards to control overhead and end-to-end delay.

## RELIABLE ADAPTIVE LIGHTWEIGHT MULTICAST (RALM) PROTOCOL

RALM is a reliable congestion controlled protocol specifically designed for the multi-hop wireless ad hoc network. RALM performs TCP-like error and congestion control by picking one multicast receiver at a time in a round-robin fashion. The goal is to reduce control overhead while guaranteeing reliable multicast delivery.

RALM achieves reliability by guaranteeing reliable data delivery to one multicast member at a time in a round-robin order. The source chooses a multicast member node called the feedback receiver, to reliably transmit data to and in turn, the feedback receiver replies with either a positive acknowledgement (ACK) to indicate successful reception of a window's worth of data or a negative acknowledgement (NACK) to request retransmissions of lost data. RALM's congestion control uses the feedback receiver's response to adjust its window size. Similar to TCP, when losses are detected, the congestion window is halved; otherwise, it

increases linearly. There is also a slow-start phase in which the congestion window increases exponentially if no losses are detected.

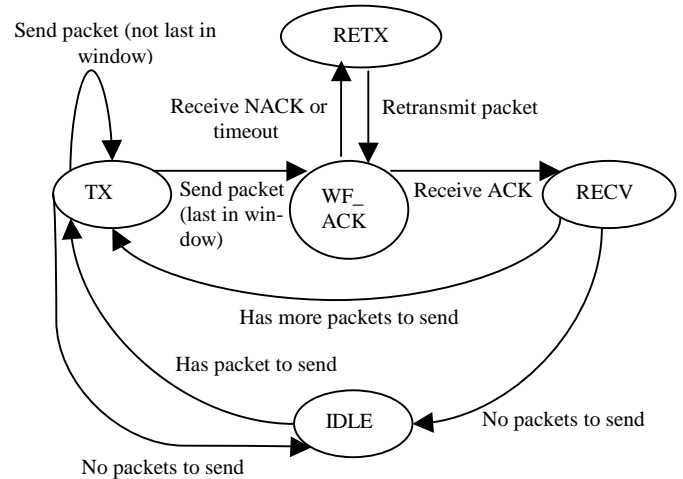
### Detailed Description

RALM assumes that multicast receivers are known to the sources either in advance or by some receiver discovery mechanism. In the case of mission critical applications, this is a reasonable assumption. For example, the commanding officer in a military special forces operation or the leader of an emergency search and rescue team will certainly have preexisting knowledge of the multicast receivers.

The source maintains a *Receiver List* that stores multicast member information. When the source has multicast data packets to deliver, it starts by choosing a node from the *Receiver List* to reliably transmit to. The source then sends the packet to the multicast group, with information in the packet header instructing the feedback receiver to reply via unicast with an ACK or a NACK, and the packet sequence number. All other receivers simply process the packet without sending a message to the source. Once the feedback receiver receives the packet, it determines whether any packets are missing. If so, it requests the source to retransmit the lost packet via unicasting a NACK to the source. Lost packets are requested one at a time until the feedback receiver has all up-to-date packets. The design philosophy behind retransmitting one packet at a time is to slow down the transmission of the source when congestion is detected. Note that since the source multicasts both new and retransmitted packets, all group members will receive the data packets. New packets are processed appropriately and duplicate packets are discarded. Retransmitted packets are multicast to the group since other nodes may also be missing these packets. Once the feedback receiver obtains all the packets, it unicasts an ACK to the source. Upon reception of the ACK, the source then chooses a new feedback receiver in a round robin fashion from the *Receiver List* to reliably transmit to, and repeats this process until the list is empty.

RALM incorporates a window-based congestion control mechanism similar to TCP to reduce overhead and increase efficiency. In RALM, the source keeps two variables: *cwin* and *ssthresh*. *cwin* records the current window size and *ssthresh* holds the slow start threshold, both of which are similar to the same variables of TCP. Initially, *cwin* is set to *cwin<sub>initial</sub>* and *ssthresh* is set to *ssthresh<sub>initial</sub>*. At the start of the protocol, the source transmits a window of packets (equal to *cwin*) to the feedback receiver. Unlike TCP however, the feedback receiver is instructed to acknowledge only the last packet in the transmitted window. In slow-start, when an ACK is received, the source exponentially increases its window size up to the maximum of *ssthresh* (with linear increase thereafter). When the source receives a NACK, it reduces *ssthresh* to  $\min(cwin/2, 2)$  and sets *cwin* to the resulting *ssthresh* value (i.e., fast recovery [13]). If a timeout occurs before the receipt of an ACK or NACK,

*ssthresh* is assigned to  $\min(cwin/2, 2)$  and *cwin* is reset to *cwin<sub>initial</sub>* (i.e., slow start [5]). Furthermore, exponential backoff is invoked to retransmit the unacknowledged packet. Note that only the last packet in the previous window is retransmitted (since it is the only packet required to be ACKed by the feedback receiver). If a receiver does not respond to the source after a certain number of retries, that receiver is skipped until the next round in the round robin cycle. Figure 1 illustrates the main algorithm of RALM.



**Figure 1. RALM state diagram. A receiver is initially selected and then later chosen in a round robin fashion during the RECV state.**

Note that RALM utilizes a TCP-like windowing mechanism. The two main differences between RALM's and TCP's congestion control are that first, one global window is maintained for all receivers, and second, only the last packet in the current window is required to be acknowledged. Using a global window assumes that all receivers experience similar traffic patterns and congestion characteristics. Alternatively, a window mechanism where the source maintains a separate window for each receiver can be utilized. With such a scheme, the sources can transmit to each receiver its window's worth of packets or transmit to each receiver the minimum window size among the receivers. The window mechanism of choice will depend on the applications target and the network environment. Also, the rationale for only acknowledging the last packet in the current window is based on the observation that ad hoc networks are extremely sensitive to network load and congestion. RALM reduces control overhead by not requiring the receiver to ACK every packet, but instead cumulatively ACK a window of packets.

### SIMULATION METHODOLOGY

We compare RALM against multicast (unreliable) UDP and SRM [3] operating on top of ODMRP (On-Demand Multicast Routing Protocol) [6]. UDP is chosen since it provides

the basic multicast support without guaranteeing reliability; therefore, any reliable multicast algorithm being considered should demonstrate improvements over UDP. SRM, as stated previously, is selected because of its strict use of error control without congestion control to achieve reliable delivery. We believe that in order to achieve reliability in MANETs, congestion control must also be considered.

### Parameters and Performance Metrics

We use QualNet [15], a discrete event simulation environment. In all experiments, 50 nodes are placed randomly in a 1500 m by 1500 m area. Constant bit rate (CBR) traffic is generated by the application, with each data payload being 512 bytes. RALM, UDP and SRM are used at the transport layer. In RALM, the values of 1 and 16 packets are chosen for  $cwin_{initial}$  and  $ssthresh_{initial}$ , respectively. Packets are multicast by ODMRP and unicast by AODV [11] with IEEE 802.11 DCF (Distributed Coordinate Function) [5] as the underlying MAC protocol. The channel capacity is 2Mb/s. Propagation is modeled using a two-ray ground reflection model where free-space path loss is used for near sight, and plane earth path loss is used for far sight. The maximum radio propagation range is 375 meters. When mobility is considered, the random waypoint model is used as the mobility pattern [1]. Simulation results are obtained from multiple runs and the results are averaged over the runs.

The metrics we examine are the packet delivery ratio, control overhead, and end-to-end delay. 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 and reliability of a protocol. Control overhead is defined as the ratio between the total numbers of data and control packets transmitted by the routing- and transport layer protocols and the number of data packets received by the 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.

### Results and Analysis

We subject RALM to a range of network characteristics, such as varying the network load, increasing the number of sources and changing mobility speed.

#### Traffic Rate

In the traffic rate experiments, all nodes are stationary. We randomly choose five multicast sources and ten multicast receivers. 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). Note that the traffic rate is simply the initial application traffic rate. Since RALM performs congestion control, the traffic rate under RALM will dynamically adjust, whereas the traffic rate will remain constant under SRM and UDP.

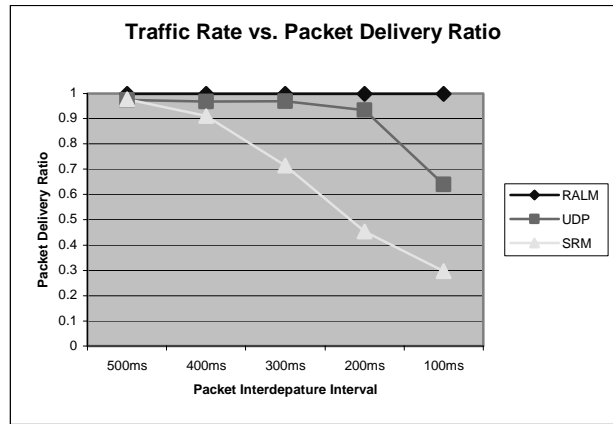


Figure 2. Packet delivery ratio as a function of traffic rate.

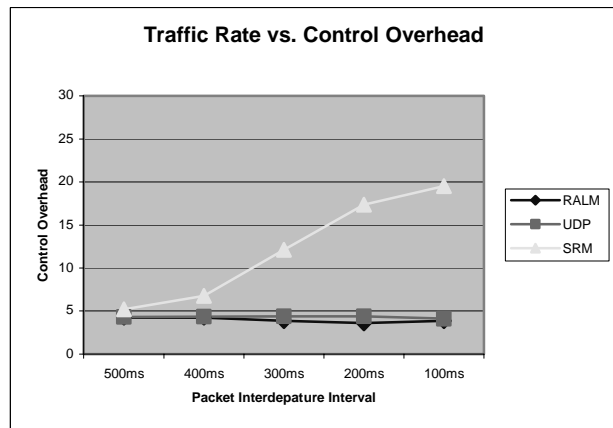


Figure 3. Control overhead as a function of traffic rate.

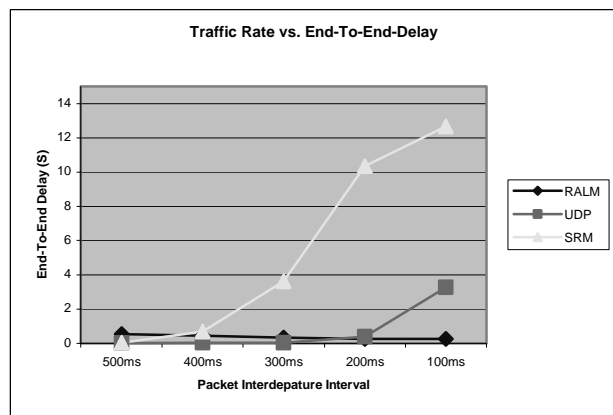


Figure 4. End-to-end delay as a function of traffic rate.

We observe from Figure 2 that RALM achieves 100% packet delivery ratio in all traffic rates we considered. UDP maintains a reasonable performance (above 90%) until the interdeparture time decreases to less than 200 ms. SRM experiences sharp degradation as we decrease the interdeparture time below 500 ms and drops to approximately 30%

packet delivery ratio under the highest load examined. 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 [14] that IEEE 802.11 broadcast mode is based on. The poor performance of SRM stems from its attempts to recover dropped packets by injecting additional control overhead through *repair request* and *repair* packets without performing any congestion control. In addition, the control overhead of multicasting *repair request* and *repair* packets in our simulation is extremely costly because the underlying multicast routing protocol must perform a flood search to locate multicast members. The *repair request* and *repair* packets are a major source of congestion that results in more data packet losses and retransmission requests (which in turn trigger more *repair request* and *repair* packets). This is evident from Figure 3. As a result, SRM performs even worse than UDP that does not provide reliability at all. RALM is able to combat increased network load through its implicit congestion control mechanism. Retransmissions due to NACKs or timeouts reduce the window and thus decrease the sending rate of the source.

Figure 4 shows RALM’s superiority to UDP and SRM in the latency metric. Although UDP maintains comparable delay with RALM for the most part, UDP starts to degrade after the network load exceeds 5 packets per second. RALM is able to maintain a constant latency due to its congestion control feature.

### Number of Sources

In the number of sources experiments, nodes are stationary. We randomly choose the number of multicast sources, ranging from 10 to 40. The number of multicast receivers is kept equal to the number of sources due to SRM’s restriction that all sources must also act as receivers. The packet inter-departure time is constant at 500 ms.

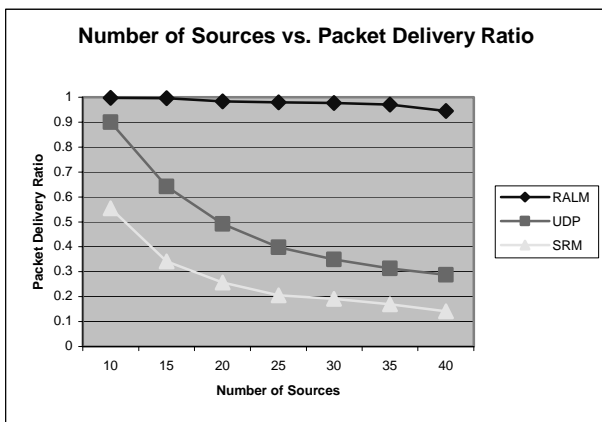


Figure 5. Packet delivery ratio as a function of number of sources.

In Figure 5, we observe that RALM is not able to attain perfect packet delivery ratio. This is because in all our experiments, the sources are continuously transmitting at a given rate of 2 packets per second. Due to the round-robin nature of RALM, it is possible that the sources cannot guarantee reliability to all receivers by the end of the simulation. Recall that in RALM, the source can only guarantee complete reliability to a receiver after the source has chosen the node to reliably transmit to. All other receivers (other than the chosen receiver) are then subject to packet loss until the source chooses them next. Since each source is not able to visit all the receivers to guarantee reliable data delivery before simulation ends, perfect packet delivery from all sources to all receivers may not be possible by the end of simulation. Note that this result stems from our simulation design, not the protocol deficiency. If we limit the number of packets sent by the source so that each source is able to visit each receiver by the end of simulation, complete reliability can be obtained.

### Mobility

We vary node speed from 0 m/s to 50 m/s using the random waypoint mobility model. We randomly choose five sources and ten receivers, with a packet inter-departure rate of 500 ms.

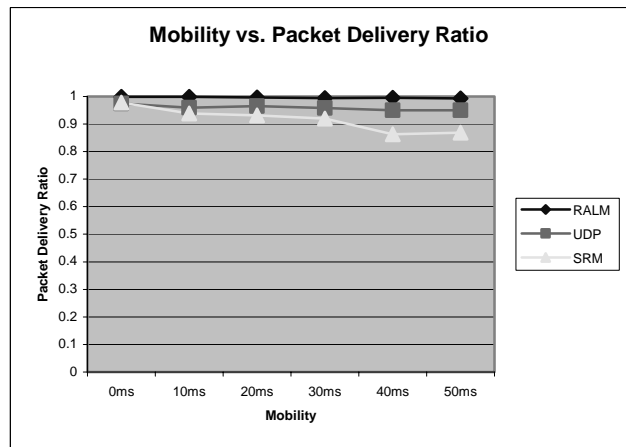


Figure 6. Packet delivery ratio as a function of mobility speed.

RALM demonstrates robustness to node mobility. We observe from Figure 6 that RALM achieves 100% packet delivery ratio under all the mobility speed examined. UDP and SRM were also able to obtain respectable reliability under all mobility scenarios, albeit at a lower packet delivery ratio than RALM.

### CONCLUSION

Mobile wireless ad hoc networks are highly dynamic because of node mobility, unreliable wireless links, and frequent outages. Moreover, they are sensitive to network load and congestion due to the broadcast nature of shared wire-

less medium. For ad hoc network protocol design, it is key that the protocols are adaptive and generate minimal control message overhead. In this paper, we proposed and evaluated the Reliable Adaptive Lightweight Multicast (RALM) protocol in multi-hop ad hoc networks. RALM is a transport protocol that uses a TCP-like approach to provide reliability and congestion control. RALM achieves reliability by guaranteeing reliable data delivery to one multicast member at a time in a round-robin order. For congestion control, RALM uses a window mechanism. The congestion window size adapts to congestion experienced by the feedback receiver.

## REFERENCES

- [1]. J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proceedings of the ACM MOBICOM'98*, Dallas, TX, Oct. 1998, pp. 85-97.
- [2]. R. Chandra, V. Ramasubramanian, and K.P. Birman, "Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-Hoc Networks," *Proceedings of the 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]. Z. Haas, M. Gerla, D.B. Johnson, C.E. Perkins, M.B. Pursley, M. Steenstrup, and C.-K. Toh (Ed.), *IEEE Journal on Selected Areas in Communications*, special issue on Wireless Ad Hoc Networks, vol. 17, no. 8, Aug. 1999.
- [5]. IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification," IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.
- [6]. S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 7, no. 6, 2002.
- [7]. J. Macker and W. Dang, "The Multicast Dissemination Protocol (MDP) Framework," *IETF Internet Draft*, draft-macker-mdp-framework-00.txt, Nov. 1996, work-in-progress.
- [8]. K. Obraczka, "Multicast Transport Mechanisms: A Survey and Taxonomy," *IEEE Communications Magazine*, vol. 36, no. 1, Jan. 1998, pp. 94-102.
- [9]. E. Pagani and G.P. Rossi, "Reliable Broadcast in Mobile Multihop Packet Networks," *Proceedings of the ACM MOBICOM'97*, Budapest, Hungary, Sep. 1997, pp. 34-42.
- [10]. S. Paul, K. K. Sabnani, J. C. Lin and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, Apr. 1997, pp. 407-421.
- [11]. C. Perkins and E. Royer, "Ad-Hoc On Demand Distance Vector Routing," *Proceedings of the IEEE WMCSA'99*, New Orleans, LA, Feb. 1999, pp. 90-100.
- [12]. T. Speakman, N. Bhaskar, R. Edmonstone, D. Farinacci, S. Lin, A. Tweedly, L. Vicisano, "PGM Reliable Transport Protocol Specification," *IETF Internet Draft*, draft-speakman-pgm-spec-03.txt, Jun. 1999, work-in-progress.
- [13]. W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *IETF RFC 2001*, Jan. 1997.
- [14]. A.S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, New Jersey, 1996.
- [15]. <http://www.scalable-networks.com>.