UNDERSTANDING THE INTERACTIONS BETWEEN UNICAST AND GROUP COMMUNICATIONS SESSIONS IN AD HOC NETWORKS

Lap Kong Law, Srikanth V. Krishnamurthy and Michalis Faloutsos *
Department of Computer Science & Engineering
University of California, Riverside
Riverside, California 92521
{lklaw,krish,michalis}@cs.ucr.edu

Abstract   In this paper, our objective is to study and understand the mutual effects between the group communication protocols and unicast sessions in mobile ad hoc networks. The motivation of this work is based on the fact that a realistic wireless networks would typically have to support different simultaneous network applications, many of which may be unicast but some of which may need broadcast or multicast. However, almost all of the prior work on evaluating protocols in ad hoc networks examine protocols in isolation. In this paper, we compare the interactions of broadcast/multicast and unicast protocols and understand the microscopic nature of the interactions. We find that unicast sessions are significantly affected by the group communication sessions. In contrast, unicast sessions have less influence on the performance of group communications due to redundant packet transmissions provided by the latter. We believe that our study is a first step towards understanding such protocol interactions in ad hoc networks.

1.   Introduction

Most routing protocol evaluations assume implicitly that only the protocol under consideration is deployed in the network. However, ad hoc networks are likely to support many types of communication such as unicast, broadcast, and multicast at the same time. Although the performance evaluation of a protocol in isolation can lend valuable insights on its behavior and performance, the protocol may have complex interactions with other coexisting protocols. These interactions may significantly alter the behavior of the protocols.

In this paper, we consider two families of protocols that are typically invoked by applications: unicast routing protocols and group communication protocols such as broadcast and multicast. While an assessment of the behavioral interactions between a large representative set of unicast routing and group communication protocols is beyond the scope of this paper, our attempt is to provide a fundamental understanding of such interactions by considering suitable candidate protocols from each set. Towards this, we consider a representative protocol from the unicast, broadcast and multicast routing. We choose the Ad-hoc On-Demand Distance Vector Routing (AODV) [10] as the representative unicast routing protocol. We consider the Simple Broadcast Algorithm (SBA) [9] as the candidate broadcast scheme and the On-Demand Multicast Routing Protocol (ODMRP) [5] as the candidate multicast protocol. The chosen protocols have been shown to be the elite members of their respective families [2] [11] [6].

The goal of this paper is to draw the attention of the community to the importance of cross protocol interactions. We discuss the possible effects that may arise when both types of protocols coexist in the network. Finally, we conduct extensive simulations to quantify the effects of the interactions. We find that the effects are indeed significant and should be considered in realistic simulations of complete systems.

We wish to point out that in a complementary effort [4], we evaluate the behavioral differences between broadcast and multicast considered in isolation in order to determine the suitability of each for performing group communications. In this paper, our objective is to examine the effects of broadcast and multicast on unicast sessions and vice versa.

The rest of the paper is organized as follows: In the next section, we provide a brief description of the candidate protocols chosen. In Section 3, we deliberate on possible interactions that may arise when unicast and group communication protocols coexist in a network; these deliberations provide a basis for the metrics that we choose when performing our simulation experiments. In Section 4, we first describe the simulation scenarios and parameters chosen; then we present and discuss the results from our simulations. Finally we conclude the paper in Section 5.

2.   Background

In the following paragraphs, we will briefly describe the candidate protocols that we choose for our evaluation. A description of the protocols in more detail can be found from the appropriate references cited. In addition, we present background that facilitates the discussions in the later sections.

Unicast: Ad hoc On-Demand Distance Vector Protocol (AODV) [10] is an on-demand routing protocol that builds routes only when needed. When a

source has packet to send but does not have a route, it buffers the packet in a temporary buffer and broadcasts a Route Request (RREQ) message. In order to perform the broadcast, the expanding ring search method is used. With this technique, a node iteratively searches for the destination in zones of increasing size (in terms of hop count) until the destination is found. Nodes that do not have the route to the destination, upon the reception of a unique RREQ message, forward it to their neighbors and update their route tables to set up a reverse route back to the source. Interim nodes that have a fresh route to the destination respond by means of a Route Reply (RREP) message directed to the originating source. As the RREP propagates back to the source, intermediate nodes set up forward pointers to the destination. Once the source receives the RREP, it may begin the transmission of packets from the temporary buffer. In AODV, a sequence number is used to prevent routing loops and nodes must use the information with the most up-to-date sequence number while making routing decisions. When a link along the active route breaks, the node upstream of the broken link propagates a Route Error (RERR) message towards the source to inform the source of the link failure.

Multicast: On-Demand Multicast Routing Protocol (ODMRP) [5] is a mesh based multicast protocol. When a multicast source has a packet to send and the multicast group members are yet to be identified, it floods a Join Query message in the network. The Join Query message is also periodically flooded to refresh group membership information and update routes as long as the source still has packets to send. When a node receives a Join Query message, it stores the source id and sequence number indicated in the message in its message cache; duplicate receptions of the same Join Query are discarded. If the message received is not a duplicate instance of a previous message and if the Time-to-live (TTL) value indicated in the message is greater than zero, the recipient node rebroadcasts the Join Query. When the Join Query reaches a multicast receiver, it creates a Join Reply message and broadcasts it to its neighbors. When a node receives a Join Reply, it checks if it is identified to be the next hop entry. If it is, the node is a forwarding node and the forwarding group flag is set. It then rebroadcasts its own Join Reply. Finally, the Join Reply reaches the multicast source and the routes are established. From then on, until information is further updated, a node will forward the packet only if it is in the forwarding group.

Broadcast: Simple Broadcast Algorithm (SBA) [9] is an intelligent broadcast protocol in the sense that it considerably reduces the number of rebroadcasts as compared with flooding. Furthermore, it has been shown in previous work [11] that SBA outperforms most of the other broadcast schemes such as the counter-based scheme and the location-based scheme. It reduces the effects of a broadcast storm [8] by using a simple technique that we discuss below in brief. SBA incorporates the exchange of periodic hello message be-

tween neighbors to enable the acquisition of local neighborhood information by each node. Each hello message contains a list of the one-hop neighbors of the broadcasting node and thus, finally, every node in the network will have its two-hop neighborhood information. The collected neighborhood information is used to decide whether or not a received data packet should be rebroadcasted. The decision is made by determining, by means of the neighborhood information table, if there exists any node that is not covered by previous broadcasts. If all the neighbors of the node are already covered, the node will not rebroadcast the packet; otherwise the node will schedule a time to rebroadcast the packet based on the number of neighbors that it has. The higher the number of neighbors, the sooner the node will rebroadcast the packet. This would therefore make nodes with higher degrees broadcast earlier than lower degree nodes. Thus, this can potentially enable the coverage of a large fraction of nodes with relatively few broadcasts.

We present a brief discussion on the IEEE 802.11 MAC and specifically the *Distributed Coordination Function* (DCF) since all protocols under our evaluation are affected by this protocol.

IEEE 802.11 Medium Access Control (MAC) [1] is the de-facto MAC layer standard for wireless networks and has been popularly considered for ad hoc networks. The fundamental access method with the DCF is known as *carrier sense multiple access with collision avoidance* (CSMA/CA). This protocol basically defines how each station may access the shared wireless channel. *Physical* and *virtual* carrier sensing are used to provide the status of the medium. In order for a host to transmit, it must sense that the medium is idle for a minimum specified duration before attempting to transmit. If the medium is busy, the host must defer its transmission and select a random backoff interval. The backoff interval counter decrements when a host senses that the medium is idle. As soon as the backoff counter reaches zero and the medium is determined idle, a host may proceed with its transmission. The aforementioned access mechanism defines the **basic access** method which is used for all broadcast, multicast and some of the unicast control packets transmissions. IEEE 802.11 MAC DCF also defines an additional mechanism which is used in conjunction with the basic access method to further avoid collisions. This additional mechanism is the **Request to send (RTS)/Clear to send (CTS)** exchange mechanism. Before any data packets are transmitted, a sender host should transmit a short RTS frame to announce its intention to transmit data. All the neighbors of the sender are expected to receive the RTS frame. The destination host upon the reception of the RTS replies with a CTS frame. Since both the RTS and CTS frame contain information that defines the period of time that the medium is reserved, the other neighbors of either the sender or the receiver preclude the initiation of any transmission in this period of time and thus avoid causing collisions. It is assumed as in prior work [2], that all

Table 1. Common Simulation Parameters (Fixed)

Number of nodes	100
Node speed	5m/s (constant)
Pause time	0 second
Simulation area	500m x 500m
Node transmission range	100m
Simulation time	60 seconds
Broadcast/Multicast group size	40% (40 group members)
AODV temporary buffer size	64 packets
Interface queue size	50 packets

the unicast data packets and some of the control packets are transmitted using this RTS/CTS exchange mechanism.

3. Issues that may arise when unicast and group communications protocols coexist

In this section, we qualitatively examine the various interactions between coexisting unicast and multicast sessions in an ad hoc network. The sessions compete with each of the network resources and in particular the wireless bandwidth. The exact nuances of the interactions do depend on the actual protocol design; we attempt to characterize certain generic aspects of the interactions via a few chosen metrics of interest.

3.1 Degradations in Packet Delivery Performance

Since, the coexisting protocols attempt to access the shared wireless channel at the same time, a higher number of collisions may be expected. This is especially the case with the control messages that are generated from both protocols since they do not employ the RTS/CTS reservation mechanism. Collisions happen when multiple hosts sense the medium to be idle and try to transmit at the same time or if there are hidden terminals that cause collisions at a receiver. Since all of the packets forwarded using the broadcasting and multicasting protocols are transmitted via the aforementioned **basic access** method, from the perspective of the unicast protocol, there is an increase in the number of broadcast packets that collide primarily with its control messages. Packet drops may occur since routes might not be established. For broadcast and multicast sessions, data packets may be lost directly due to collisions.

To elaborate on the effects of broadcast on unicast protocols, typically when a route to a destination is needed a protocol such as AODV invokes the flood (perhaps a modified version) of an RREQ message as mentioned earlier. In the presence of a group communication session these broadcast RREQ messages typically collide and are hence lost. In some cases, a route is not discovered although it actually exists. This would trigger additional route query attempts and thus would further increase the generated congestion. With AODV if four

consecutive route query attempts were to fail, the sending source would abort the session and drop the packets.

3.2   Increased Latency Effects

Note that a secondary effect of the congestion is that the MAC layer interface queues fill up quickly. Since the control messages have priority over actual data, the increase in the volume of control messages (repeated RREQ floods) may be expected to cause the actual data packets to experience increased (and potentially large) latencies in each queue.

The unicast packets are likely to be affected to a higher extent than the group communication packets. Because of the higher levels of redundancy built into the group communication protocols, they may be less affected by congested areas; packets that may actually be able to go through less congested areas to reach their intended destinations more quickly. Furthermore, since there are no explicit RTS/CTS exchanges, it is enough that only the sender perceive the channel to be free (as opposed to requiring both the sender and receiver to be free as in the case of the unicast connections). Thus, the group communication packets have a tendency to depart from their interface queues more quickly (once they get to the head of the queue) as compared with their unicast counterparts. Finally, note that there are no explicit attempts to retransmit a group communication session's packets. However, the MAC layer makes seven retransmission attempts for every unicast packet. Each attempt is made after an exponential back-off period from the previous retransmission attempt. Thus this retransmission procedure is expected to contribute immensely to the latency evolved by the unicast connections.

3.3   Increased Control Overhead

An consequence of congestion is an increase in control overhead. Control overhead includes all packet transmissions excepts the actual data packet transmissions. In other words, all the control messages transmitted for assisting with the functionality of a protocol account for control overhead. In an environment where contention levels are high, a great portion of the control messages are lost due to collisions. These control messages may or may not be retransmitted (depending on the type of the message). With a broadcasting protocol such as SBA, the control messages (Hello messages) are not retransmitted upon loss. For protocols such as ODMRP and AODV, control messages are retransmitted. If a route is not found, several additional attempts are made before the node aborts the query attempt. We call this type of protocols as **routing-dependent** protocols and the broadcasting protocol such as SBA as **routing-independent** protocols. Routing-dependent protocols are more sensitive to congestion since a complete route must be established before the route

Table 2. Parameters for the first part of the simulation

Number of unicast sessions	1,2,4,8
Unicast rate per session	5, 10, 20 packets/second
Unicast packet size	512 bytes
Broadcast/Multicsat packet size	512 bytes
Number of broadcast/multicast source	1
Broadcast/Multicast rate	5 packets/second

Table 3. Parameters for the second part of the simulation

Number of broadcast/multicast sessions	1,2,3,4
Unicast rate per session	2, 4, 8 packets/second
Unicast packet size	256 bytes
Broadcast/Multicsat packet size	256 bytes
Number of unicast sessions	2
Unicast rate	5 packets/second

discovery cycle ends. Therefore, the control overhead will increase when the contention level increases. On the other hand, routing-independent protocols do not require the construction of any forwarding structure and thus the control overhead is independent of the contention level.

4. Simulation Study

We conduct extensive simulations to quantify the effects of coexistence of two different types of protocols on the same network.

We use ns-2 simulator [7] in our evaluation. We vary parameters of interest to study the effects of interest under various scenarios. Other system parameters have been chosen to be fixed but the behavioral results from sample simulations for other values of these parameters were similar. The set of generic parameters used are listed in TABLE. 1. We divide the simulations into two parts. In the first part, we simulate scenarios to study the effects of unicast traffic on group communication performance. In the second part, we try to capture the effects of group communication on unicast session performance. The parameters that we vary in the first part of the simulation are listed in TABLE 2, and the parameters that we vary for the second part of the simulation are in TABLE 3. We repeat each simulation 30 times and for each run, we use different scenarios. In order to allow both types of protocols to run concurrently, we have made modifications to the ns-2 source code.

4.1 Simulation results

In the following, we present and discuss the simulation results that we obtain. In order to understand the mutual effects between protocols, we also conducted baseline simulations such that only the protocol of interest is deployed in isolation. We can thus compare the performance of each protocol in the presence of other coexisting protocols with that of the protocol in iso-
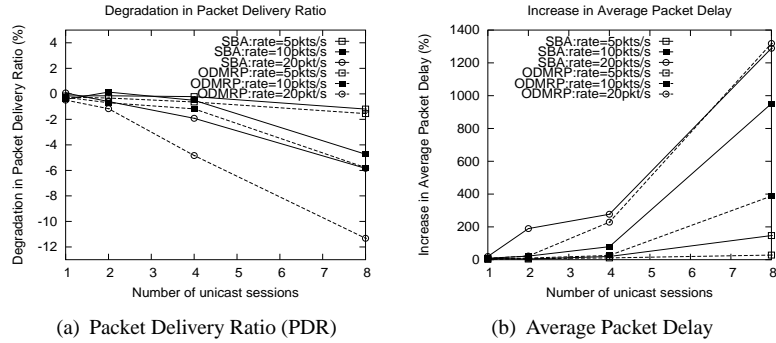
Figure 1. The differences in the performance of SBA/ODMRP in the presence of unicast traffic

lation. Most of the graphs presented reflect the comparisons and are usually represented as a percentage difference.

4.2 The effects of unicast protocol on the performance of group communication protocols

We first present the performance results that capture the effects of the group communication protocols in the presence of unicast protocol.

Packet Delivery Ratio. As seen from Fig. 1(a), the performance of both SBA and ODMRP degrade when the number of unicast sessions and the unicast rate increases. This is expected since the collisions on the physical channel increases when the background unicast traffic increases. Note that the packet delivery ratio of ODMRP decreases to a greater extent than that of SBA. The reason for this is that SBA has a higher rebroadcast redundancy than ODMRP and therefore the chance of successful packet delivery is higher in spite of congestion as discussed earlier in Section 3.1. The maximum offered load by all the unicast sessions together is 0.625Mb/s. This is a significantly high load given that the channel data rate is assumed to be 2Mbps. However, most of the results that we obtain demonstrate less than a 10% decrease and thus we do not regard these as a significant degradation.

Average Packet Delay. Although the packet delivery ratio with both SBA and ODMRP only decreases slightly, the average packet delay increases drastically as seen from Fig. 1(b). When the amount of unicast traffic is low, the average packet delay remains almost the same as that of the case wherein the group communication protocol is considered in isolation. However, when the unicast traffic increases, the delay increases significantly. When the number of unicast sessions is 8 and the unicast rate is 20 pkts/second, the average packet delay of both SBA and ODMRP data packets jumps by almost 1300% as com-
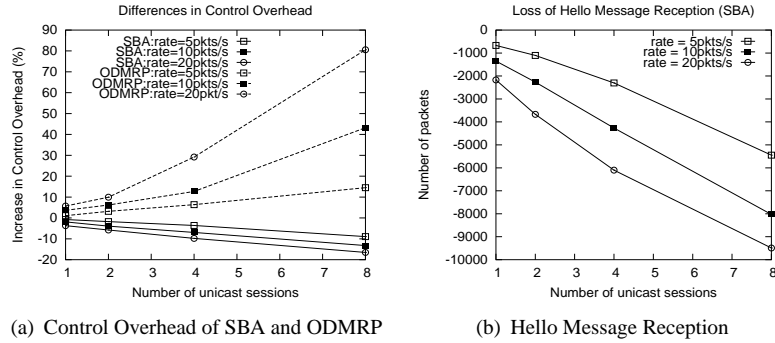
Differences in Control Overhead

Loss of Hello Message Reception (SBA)

(a) Control Overhead of SBA and ODMRP (b) Hello Message Reception

Figure 2.   The effect of hello message loss on the control overhead of SBA

pared to the case where no unicast traffic is present. The reason for such a drastic increase in average packet delay is that the data packets are caught up at the lower layer interface queue as discussed earlier in Section 3.2. The situation becomes worse when the amount of unicast traffic further increases as one might expect.

Control Overhead.   The control overhead of SBA and ODMRP are affected differently under the presence of unicast traffic. From Fig. 2(a), we see that the control overhead of SBA decreases as the amount of unicast traffic increases. In contrast, the control overhead of ODMRP increases drastically. Upon further investigation, we find that the decrease in control overhead of SBA is caused by the loss of hello messages. From Fig. 2(b), we see that the number of successful hello packet receptions decreases as the amount of unicast traffic increases. These losses of hello packets are mainly due to the collisions. As a result, the two-hop neighborhood information maintained at each node is incomplete and inaccurate. Therefore, on average, the size of the hello message (containing the one-hop neighborhood information) is smaller. This decrease in control overhead is significant, since state information is important for the effective functioning of the SBA protocol. For ODMRP, the control messages are retransmitted for a limited number of times. Because of this, the multicast structure is constructed successfully. Due to an increase in such retransmissions, the control overhead increases in ODMRP as the amount of unicast traffic increases, as we see in Fig. 2(a).

4.3   The effects of group communications protocols on the performance of the unicast protocol

In this subsection, we present the performance of AODV in the presence of broadcast or multicast traffic.
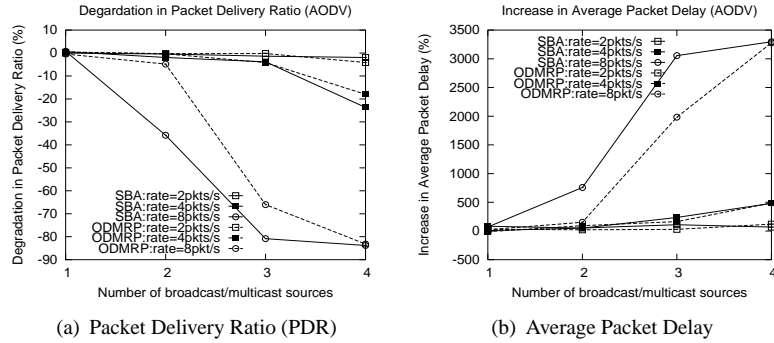
Degardation in Packet Delivery Ratio (AODV)

Increase in Average Packet Delay (AODV)

(a) Packet Delivery Ratio (PDR)　　　(b) Average Packet Delay

Figure 3. The differences in the performance of AODV in the presence of group communication traffic

Packet Delivery Ratio. From Fig. 3(a), we see that when the broadcast or multicast traffic rate is low (2 packets/s), the packet delivery ratio of AODV remains almost same as that in the case when it is considered in isolation. However, when the broadcast or multicast traffic increases, the packet delivery ratio of AODV decreases significantly. When there are 4 broadcast/multicast sources with a data rate of 8 packets/s, the packet delivery ratio of AODV differs by almost 90% as compared to the case where no group communication sessions are present. This tells us that the group communication traffic even at moderate amounts, degrade AODV performance significantly.

We investigate the reasons for the loss of packets with AODV. In Fig. 4(a) and Fig. 4(b), we represent the distribution of the AODV data packet drops when a varying number of broadcast/multicast sources with different data transmission rates are present in the network. At a transmission rate of 2 packets/s, only a small fraction of the unicast packets are being dropped (compared to the total of 600 packets) and more than half of them are due to link failures. We regard these as a normal packet drops since the link failures in this case are mainly caused by the node mobility. However, as the amount of broadcast/multicast traffic increases, the number of packet drops increases drastically. We notice that the number of packet drops at the interface queue upon link failure increases and accounts for a major portion of the total packet losses. This happens when there is a link failure and a large number of packets waiting in the queue rely on that link. Note that when the congestion increases, the number of link failures are also seen to increase. These are primarily due to "*false link failures*" and artifact of the IEEE 802.11 MAC protocol [3]. Here if the intended recipient of an RTS packet is within the sending range (interference range) of some other node, it does not respond to the sender of the RTS message with a CTS message. After seven consecutive attempts the sender deems the link to have failed although in reality it still exists. Notice that the aforementioned packet losses only happen with high amounts of broad-
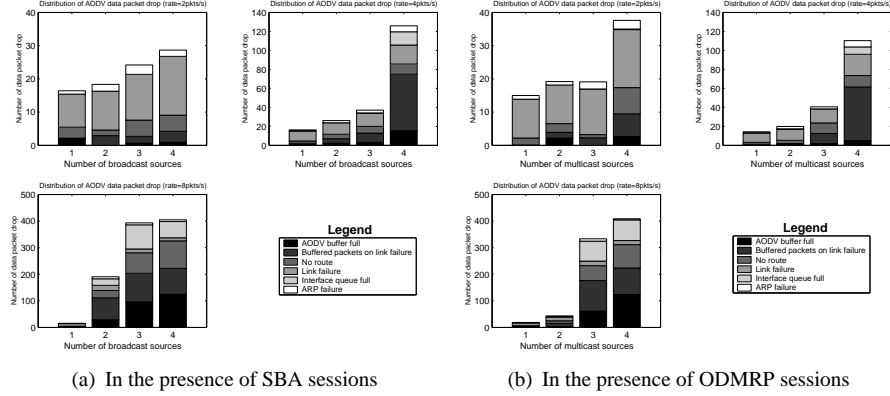
(a) In the presence of SBA sessions (b) In the presence of ODMRP sessions

Figure 4. The distribution of AODV data packet drop

cast/multicast traffic. This is in line with our discussion in the previous section that a high amount of traffic increases the contention level at the lower layer interface queue. At very high amounts of broadcast/mutlicast traffic (at a rate of 8 packets/s and $\geq$ 3 sources), packet losses increase drastically. This implies that at this level of background broadcast/multicast traffic, AODV route discovery attempts begin to encounter failures. A lot of packets are buffered in the AODV buffer temporarily without a route to the destination.

Average Packet Delay. In addition to the poor packet delivery ratio, the average packet delay also increases significantly (see Fig. 3(b)). When the background broadcast/multicast traffic rate is low (2 packets/s), the average packet delay remains steady with only a small delay increase as compared with the case of AODV in isolation. However, when the background broadcast/multicast rate increases, the delay increases drastically. When the broadcast/multicast rate is 4 packets/s, the delay rises by up to 500% and at a rate of 8 packets/s the delay increase is even up to 3250%! This effect is a combination of the delay incurred by the data packet while waiting in the interface queue as well as the delay due to waiting for the route to be found.

5. Conclusions

In this paper we attempt to understand the impact of coexisting unicast and group communication protocols. Specifically, we attempt to quantify the extent of performance degradations on one due to the other and discuss the underlying effects that cause such degradations. This is motivated by realistic networks wherein multiple applications, some of which may require unicast sessions while others require multicast/broadcast sessions, are likely to exist. To the best of our knowledge this is the first attempt to undertake such a study.

We discuss the possible effects of the unicast sessions on the coexisting group communication sessions and vice versa and perform extensive simulations to corroborate our reasoning and to quantify the effects. The effects are quantified in terms of the packet loss rate and the average incurred delay. We find that due to the inherent redundancy in group communication protocols, these sessions are affected to a lesser extent by unicast sessions than vice versa. The poor performance of the latter in the presence of the former is attributed to the increase in the number of false link failures, the increase in buffer delays which in turn cause packets to encounter link failures with increased probability (at the instance of which they are dropped), and due to collisions of route discovery query packets and the consequent failures of such queries. The group communication sessions also suffer from increased packet losses and delays. Broadcast schemes are likely to be more robust to route failures than multicast schemes since they are independent of routing.

References

[1] ANSI/IEEE std 802.11, 1999 edition.

[2] Broch, Josh, Maltz, David A., Johnson, David B., Hu, Yih-Chun, and Jetcheva, Jorjeta (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE MobiCom'98*, pages 85–97.

[3] Klemm, Fabius, Ye, Zhenqiang, Krishnamurthy, Srikanth V., and Tripathi, Satish K. (2004). Improving TCP performance in ad hoc networks using signal strength based link management. *Ad Hoc Networks Journal*. (to appear).

[4] Law, Lap Kong, Krishnamurthy, Srikanth V., and Faloutsos, Michalis. On evaluating the trade-offs between broadcasting and multicasting in ad hoc networks. (submitted to IEEE MILCOM 2004).

[5] Lee, Sung-Ju, Su, William, and Gerla, Mario (2002). On-demand multicast routing protocol in multihop wireless mobile networks. In *ACM/Baltzer Mobile Networks and Applications, special issue on Multipoint Communication in Wireless Mobile Networks*, volume 7, pages 441–453. Kluwer Acadmic Publishers.

[6] Lee, Sung-Ju, Su, William, Hsu, Julian, Gerla, Mario, and Bagrodia, Rajive (2000). A performance comparison study of ad hoc wireless multicast protocols. In *Proceedings of the INFOCOM'04*, pages 565–574.

[7] McCanne, S. and Floyd, S. Ns-2 simulator.

[8] Ni, Sze-Yao, Tseng, Yu-Chee, Chen, Yuh-Shyan, and Sheu, Jang-Ping (1999). The broadcast storm problem in a mobile ad hoc networks. In *Proceedings of the ACM/IEEE MobiCom'99*, pages 151–162, Seattle, Washington.

[9] Peng, Wei and Lu, Xi-Cheng (2000). On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the ACM MobiHoc'00*, pages 129–130, Boston, Massachusetts.

[10] Perkins, Charles E. and Royer, Elizabeth M. (1999). Ad hoc on-demand distance vector routing. In *Proceedings of the IEEE WMCSA'99*, pages 90–100, New Orleans, LA.

[11] Williams, B. and Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM MobiHoc'02*, pages 194–205.