Fast and Accurate Link Discovery Integrated with Reliable Multicast in 802.11

Daniel Lertpratchya
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

Douglas M. Blough
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

George F. Riley
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

***Abstract*—Maintaining accurate neighbor information in wireless networks is an important operation upon which many higher layer protocols rely. However, this operation is not supported in the IEEE 802.11 MAC layer, forcing applications that need it to each include their own neighborhood mechanism, creating redundancies and inefficiencies and failing to capitalize on potential synergies with other MAC layer operations. In this work, we propose to integrate link discovery and neighborhood maintenance with a reliable multicast extension to the IEEE 802.11 MAC. We show through simulations that our protocol adapts to neighborhood changes faster than traditional neighborhood maintenance mechanisms, thereby allowing MAC-layer multicast operations to achieve higher delivery rates. We also demonstrate that our protocol can quickly and reliably distinguish between unidirectional and bidirectional links. Traditional mechanisms assume links are bidirectional based on one-way reception of a short "hello" packet, which results in significant problems with higher-layer operations such as routing because of many unidirectional links being classified as bidirectional.**

I. Introduction

In addition to coordinating access to a shared channel among multiple devices, medium access control (MAC) protocols typically perform other functions. One function that is particularly important in wireless networks is link layer reliability. For example, the 802.11 MAC specification dictates that unicast frames be positively acknowledged and that transmitting nodes resend a frame if an ACK is not received. However, when it comes to the transmission of broadcast frames, the specification does not specify any reliability mechanism. Thus, standard 802.11 MAC broadcast is unreliable.

One of the difficulties of supporting reliable multicast at the MAC layer is that it requires the maintenance of neighborhood information so that the transmitter knows from which nodes it should receive ACKs. Absent this information, a MAC-layer multicast protocol can only broadcast a frame and assume that whoever the current neighbors are, they will successfully receive the transmission. This is the standard 802.11 MAC broadcast mechanism and it is inherently unreliable.

Prior proposals for reliable MAC multicast for 802.11, e.g. [1], [2], [3], [4], assume that neighborhood information is maintained by a separate protocol. Typically, it is assumed that nodes periodically broadcast special HELLO or beacon messages in order to notify other nodes of their presence. Section IV points out several problems with this approach. A rather obvious problem is that these special periodic messages are themselves unreliable and, therefore, there is no guarantee that neighboring nodes receive them correctly.

In this paper, we propose a new protocol, which integrates the functions of reliable multicast and neighborhood maintenance. Our protocol is compatible with the 802.11 MAC and provides a number of distinct advantages compared to the use of a separate protocol with periodic HELLO messages. Among its advantages are: 1) much faster recognition of new neighbors, 2) the ability to immediately distinguish between uni-directional and bi-directional links, 3) a unified and cleaner protocol design, and 4) the ability to efficiently gather neighborhood information for higher-layer protocols. In fact, concerning point 4, Kuhn et al. advocate MAC-layer support for reliable multicast and neighborhood maintenance as a basic building block for higher-layer protocols [5]. Numerous higher-layer protocols, such as routing and topology control [6], to name a few, require up-to-date neighborhood information. Maintenance of this information at the MAC layer is the most efficient solution and obviates the need for higher-layer protocols to implement their own link discovery and neighborhood maintenance mechanisms.

II. Related Work

In this section, we discuss related work in the field of neighborhood maintenance and reliable multicast. Previous work considered these two topics separately. To the best of our knowledge, our proposed protocol is the first one to offer an integrated solution to the two problems, which increases network efficiency and improves overall network performance.

A. Neighborhood Maintenance

An up-to-date neighbor list must be maintained in order for any reliable multicast protocol to work correctly. Without an up-to-date neighbor list, a node may miss nodes in its neighborhood or waste time trying to send packets to nodes that are no longer in the neighborhood. Three common approaches to maintaining neighborhood information are the

HELLO-based approach, the random access approach, and topology control.

In a HELLO-based approach, nodes rely on HELLO messages from other nodes to maintain their neighbor lists. Each node adds a new neighbor whenever it receives a HELLO message from a node not currently on the list, and removes a node from its neighbor list if it has not received any frame from that node within a HELLO timeout. Nodes use HELLO messages as heartbeat messages to prevent neighbor nodes from mistakenly removing them from their neighbor list when the nodes have no frames to transmit. Using HELLO messages consumes network bandwidth and incurs delays since a node is required to hear a HELLO message to detect a new neighbor.

Several studies on HELLO messages have been done. Chakeres and Belding-Royer [7] proposed that the characteristics of HELLO messages should be the same as that of data packets. Tan and Seah [8] and Stanze, et al. [9], proposed that HELLO message frequency should be dynamically adjusted according to the mobility in the network, where mobility is measured by changes in the one hop neighbors. Turnover-based adaptive HELLO protocol (TAP) [10] adjusts HELLO message frequency according to the current speed.

In a random access approach, nodes operate in different states. In birthday protocols [11], nodes randomly choose to enter a transmit, listen, or energy saving state in each time slot. A node transmits one beacon in the transmit state. If there is no collision, nodes in the listen state will receive the beacon and recognize the sender. Borbash, et al. [12], relax the requirement by allowing nodes to operate asynchronously. Vasudevan, et al. [13], considered an ALOHA-like neighbor discovery in a synchronous system and showed that improvement can be made if nodes have a collision detection mechanism. They also proved the expected time for a node to discover all neighbors. Being statistical in nature, these approaches converge over time to an accurate neighborhood view, but they therefore do not handle dynamic situations, e.g. networks with mobility.

The last approach to discover neighbors is topology control [6]. The goal of topology control is to dynamically adjust the transmission power of each node to maintain some property of the network. These properties often require the knowledge of neighbor nodes. Thus, neighbor discovery is often integrated as a part of topology control. However, these neighbor discovery mechanisms are not performed at the MAC layer. In a topology control proposed by Wattenhofer et al. [14], each node sends a beacon with growing transmission power until the number of neighbors exceeds the threshold, or the maximum transmission power is reached. In LMST protocol [15] and k-Neigh protocol [16], each node sends a beacon at the maximum transmission power to announce its presence. Every node that receives the beacon stores the identity and the estimated distance of the sender.

Unlike previously proposed neighborhood maintenance protocols that operate as external protocols, our proposed protocol is integrated into the MAC layer. Thus, higher layer protocols do not need to implement their own neighborhood maintenance mechanism. Integrating a neighborhood maintenance mechanism into the MAC layer results in a more up-to-date neighborhood information that provides higher reliability to multicast transactions. More accurate classification of unidirectional and bidirectional links can improve the performance of routing algorithms [17], [18].

B. Reliable Multicast

Two major approaches have been proposed to provide reliability for multicast frames at the MAC layer. One approach is using out-of-band signaling to provide feedback to a multicast sender. Examples include RBMAC [19] and BPBT [20]. To use out-of-band signaling, special hardware is required at each node, which may not be practical.

The second MAC-based multicast approach uses positive acknowledgment. Several protocols employing this approach have been proposed [1], [2], [3], [21], [22], [23]. BMW [1], MMP [2] and MWB [3] modified frame headers to include multiple receivers' addresses. In BMW, the source selects one multicast receiver to reply with an ACK in a round-robin fashion. In MMP and MWB, each receiver replies with an ACK or a CTS sequentially as determined by its position in the DATA frame or the RTS frame.

More recent work focused on reducing the overhead of using positive acknowledgement [21], [22], [23]. In SRM, an access point selects a leader among its receivers to send acknowledgement back. However, SRM cannot guarantee that all nodes received the multicast frame since only the leader sends an acknowledgement. In [22], [23], CTS and ACK frames are modified to include a pre-allocated DS-CDMA code and BPSK symbol, respectively. These codes were chosen such that the CTS and ACK frames can be received simultaneously. Thus, only one CTS slot and ACK slot are required. However, this approach requires the keys to be predetermined to prevent key collision and special hardware is required at each node.

A more recent IEEE 802.11aa draft, which is being proposed, provides a reliability to multicast transaction in an infrastructure network by using Block ACK mechanism. An access point transmits multiple multicast frames to its associated stations before sending a Request for ACK to instruct each station to acknowledge multicast frames. Block ACK can reduce the ACK overhead but also delays transmission of important neighbor information.

Our proposed protocol uses positive acknowledgment to provide reliability for multicast frames at the MAC layer and does not rely on out-of-band signaling. Our protocol differs from other protocols that use positive acknowledgements in that our protocol uses a frame size as a threshold for a four-way transaction. Previously proposed protocols either exclusively use a four-way transaction in all transactions, or use a four-way transaction to recover the loss. We show in Section IV that using a threshold-based approach results in a better bandwidth utilization.

III. Link Discovery Protocol and Reliable Multicast

We propose an extension for the IEEE 802.11 framework called Link Discovery with Reliable Multicast protocol (LDM). The proposed protocol has two main goals. One goal is to dynamically track nodes' neighbor sets within the MAC layer and the other is to provide reliability for MAC-layer multicast frames. To achieve the first goal, LDM provides a mechanism for devices to quickly recognize changes in their neighbor sets. Since many higher-layer protocols require neighborhood maintenance, supporting this capability efficiently at the MAC layer will both simplify the design of higher layers and eliminate potential redundancies in their execution. Thus, a unified MAC layer that supports both reliable multicast and neighborhood maintenance will streamline overall network performance. Before presenting our link discovery mechanism, we present the reliable multicast protocol it relies on, which is an enhancement of existing 802.11 reliable multicast protocols.

A. Basic Reliable Multicast Protocol

LDM modifies the default 802.11 frame headers to include additional receivers' addresses for a multicast transaction. Figure 1 shows the modified frame structure. LDM introduces a new field in the MAC header called Extended Control field. The Extended Control field is an 8-bit field where the least significant bit is called the "Join ACK" bit, and the next three bits are called the "Join ACK level".

Similar to 802.11 unicast, LDM supports both two-way and four-way transactions. LDM differentiates between two-way and four-way transaction by frame size. Figure 2 illustrates the two scenarios of LDM.

For a two-way transaction, the DATA frame is modified to include multiple receivers addresses. The ACK frame is modified to include the ACK sender's address. The multicast source selects as many receivers from its neighbor list as permitted by the maximum 802.11 frame size. Thus, the total number of addresses LDM can have in a transaction is limited by the data size. The multicast source splits a multicast transaction into multiple sub-transactions if it cannot support all neighbors in one transaction. The multicast source sets the Join ACK bit to 1 in the last sub-transaction, and to 0 in the other subtransactions.

A multicast source initiates a two-way transaction by sending a modified DATA frame. If the multicast source selects N receivers and sets the Join ACK bit to 0, the time after the DATA frame is divided into N ACK slots. If the Join ACK bit is set to 1, the time is divided into $N+1$ slots. All ACK slots are separated by SIFS. When a node receives the DATA frame, it checks if the DATA frame is addressed to itself or not. If the DATA frame is addressed to itself, the node schedules transmission of a modified ACK frame in a corresponding ACK slot according to its position in the DATA frame. If ACK frames from all N selected neighbors are received, the multicast source considers the multicast transaction completed. If ACKs from some receivers are not received, the source re-includes the missed-ACK receivers in a subsequent sub-transaction or re-starts the multicast transaction for the missed-ACK receivers if no additional subtransactions are scheduled. The source re-transmits to each failed receiver seven times before giving up.

For a four-way transaction, the RTS is modified to include multiple addresses and an 48-bit nonce. A multicast source initiates a four-way transaction by sending a modified RTS frame that includes N selected receiver addresses and an 48-bit nonce. The time after the RTS frame is divided into N CTS slots. When a node receives the RTS frame it checks if its address is present in the RTS frame or not. If the node address is included in the RTS frame, it schedules transmission of a modified CTS frame according to its position in the RTS frame. All nodes that received the RTS frame save the 48-bit nonce associated with the RTS frame.

If the multicast source receives at least one CTS, the multicast source schedules transmission of the DATA frame. The DATA frame is a standard 802.11 DATA frame with the address FB:FF:FF:FF:FF:FF in the Address 1 field to indicate that the DATA is the multicast frame. The DATA frame also has the previously generated nonce in the Address 3 field. The purpose of the 48-bit nonce is to match between RTS and DATA frames. The time after the DATA frame is divided into N slots if the Join ACK bit was set to 0 or $N + 1$ slots if the Join ACK bit was set to 1. Each multicast receiver that correctly received the DATA frame with the matched nonce schedules transmission of a modified ACK frame in an ACK slot according to its position in the RTS frame. If all ACK from N selected receivers are received, the multicast source considers the multicast transaction completed. If ACK frames from some receivers are missing, the multicast source re-includes the missed-ACK receivers in the subsequent transactions.

B. Link Discovery

Our main goal in designing LDM is to enable nodes to quickly recognize neighborhood changes and to eliminate the need for a separate neighborhood maintenance mechanism. Neighborhood maintenance typically involves the use of separate HELLO messages in a network. This HELLO mechanism, in addition to being wasteful of network bandwidth, has several other deficiencies we demonstrate in Section IV. In LDM, every frame that includes its sender's address and has the same transmission characteristics as a DATA frame serves the same function as a HELLO message. Every node has a countdown timer that counts from the last

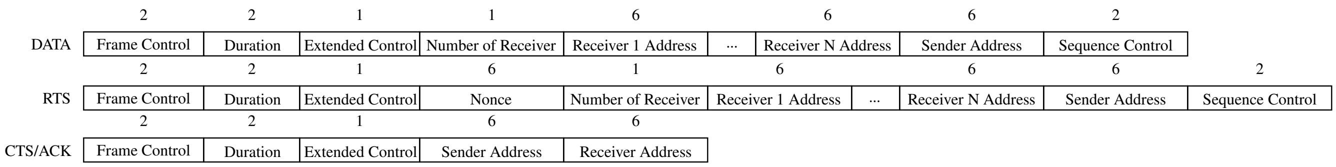

Figure 1. Modified headers

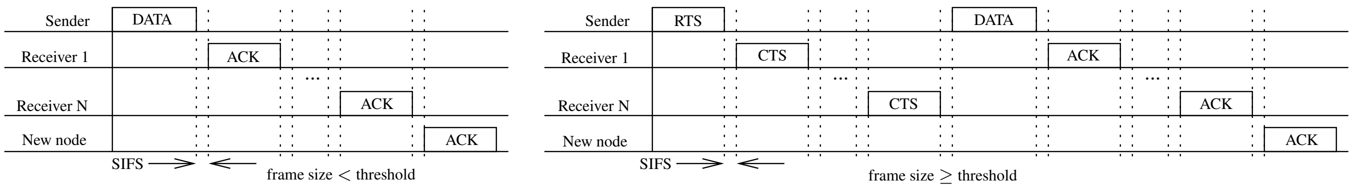

Figure 2. Link discovery with reliable multicast

time it sent a frame that can be treated as a HELLO message. If the countdown timer expires, the node sends out a data frame that serves as a HELLO message.

In our protocol, we distinguish between *incoming* neighbors and *bidirectional* neighbors. Node A considers node B as an incoming neighbor if A received frames transmitted by B. For node B to be considered a bidirectional neighbor by A, two conditions must be satisfied: 1) B must be able to receive frames sent by A, and 2) A must be able to receive ACKs from B. We do not assume that all links are bidirectional as Kotz, et al. [24] showed that the probability of an asymmetric link can be as high as 24 percent. Assuming all links are bidirectional can degrade the network performance when unidirectional links are present [17], [18].

To enable fast neighbor discovery and differentiation between unidirectional and bidirectional links, LDM provides an extra $(N+1)$th ACK slot for a new node to send an ACK frame called the Join ACK slot. If the node is not addressed in the DATA/RTS frames and the Join ACK bit is set to 1, the node randomly decides to send an ACK in the Join ACK slot with a probability that is indicated by the Join ACK level. Therefore, the new node is able to make its presence known to the sender as soon as it receives a DATA frame. The sender is also able to classify the new node as a bidirectional neighbor immediately since both bidirectional conditions are satisfied.

The Join ACK level maps to a probability value, which is used to reduce the chance of ACK collision when multiple nodes try to join at the same time. The sender adjusts the level according to what happened in the previous Join ACK slot. The sender assumes that the probability is too low if no transmission is detected during the previous Join ACK slot, and increases the probability level. If the sender detects a transmission, but failed to receive a frame, it assumes that multiple nodes are trying to join at the same time. The sender then decreases the probability level. If the sender correctly receives an ACK from a new neighbor, the sender does not change the probability level. The mapping between the probability level and the probability value, and how a sender adjusts the probability level, can be set to match a current network's condition.

Neighbor classification works as follows: consider two nodes A and B, A classifies B as an incoming neighbor if A receives a DATA frame or a HELLO from B that does not include A in the destination list. A then sends a Join ACK to B. After receiving a Join ACK from A, B classifies A as a bidirectional neighbor, because B is certain that its transmission can be ACKed by A. In the next transmission by B, it includes A in the destination list. Upon receiving the DATA from B, A can now classify B as a bidirectional neighbor since A knows that its Join ACK was correctly received by B. If a link from A to B is unidirectional, B will recognize A as an incoming neighbor since it can receive a DATA frame or a HELLO from A. In this case, B keeps track of the number of times it sends a Join ACK to A and it stops trying to join after seven attempts, at which time it classifies the link as unidirectional.

Two mechanisms are used to detect when a neighbor leaves the neighborhood: a retransmission limit and a timeout. A node keeps track of the number of retransmissions to each bidirectional neighbor. If the number of retransmissions is seven, the node considers the neighbor to be an incoming neighbor. A node removes an entry from its neighbor list if it has not received any frame from a neighbor within a timeout period.

C. Reliability and Scalability

LDM uses positive acknowledgments as a means to provide reliability. A sender includes a list of all intended receivers in an RTS or a DATA frame. Each receiver then replies with a CTS or an ACK sequentially according to its position in the RTS or the DATA frame. If a neighbor did not reply with an ACK, the source resends to that neighbor in a subsequent sub-transactions or in a separated transaction.

As illustrated in Figure 2, the transmissions of CTS and ACK from receivers one by one are time consuming. The time required is an increasing function of the number of multicast receivers. Although we do not set a limit on the number of multicast receivers, there are two factors that affect the maximum number of receivers in the multicast transactions.

First, the IEEE 802.11 specification imposes limits on the maximum frame size. Thus, the maximum number of addresses in the header depends on the size of the data. LDM is guaranteed to support at least three receivers since the original MAC header has four address fields. LDM uses one address field for the sender address and the remaining three address fields for three receivers. More than three receivers can be supported if the data size is smaller. If a hard limit is placed on the data size, the minimum number of receivers that can be supported in one frame can be increased.

The second factor is the overhead of the positive acknowledgement approach. One of the problems of using the positive acknowledgement approach is the ACK explosion problem. All multicast protocols that employ the positive acknowledgement approach experience this problem. We evaluate the efficiency of the protocol in Section IV-H.

IV. Performance Evaluation

We have evaluated our protocol performance through simulation. In this section, we provide details of the simulation environment, the assumptions, and the simulation results.

A. Simulation Parameters and Assumptions

We used ns-3.10 simulator to evaluate the performances of all protocols. We considered the physical interference (PI) model in this work. In the PI model, interference from all concurrent transmitters in the network, no matter how distant, is factored into the signal-to-interference-plus-noise ratio (SINR) value at the receiver, and the SINR value determines the probability that a transmission is successful.

We compare our protocol against existing 802.11 reliable multicast protocols, MMP [2] protocol and MWB [3], supplemented with HELLO-based neighborhood maintenance. All HELLO messages have the same characteristics as DATA frames [7]. Two variations of HELLO mechanisms were used: a simple HELLO message mechanism where all nodes send a HELLO message every one second and the timeout is two seconds, and the TAP protocol [10], where HELLO rate varies with node velocity.

For LDM protocol, the Join ACK probability value ranges from 0.125 to 1. The mapping function used in the simulation was $\Pr(Join) = \frac{1}{8} \times (1 + L)$ where L is the Join ACK level that ranges from 0 to 7. If a multicast source does not hear any transmission during the previous Join ACK slot, L is increased by one. If a multicast source detects a transmission but fails to receive the frame in the previous Join ACK slot, L is decreased by one.

Table I
Common simulation parameters

Parameter	Value
Deployment area	1000 m by 1000 m
Mobility model	Random waypoint [25]
Speed	v to $v+2$ m/s
Pause time	0 s
Simulation duration	600 seconds
Propagation loss model	Log-distance
Path-loss exponent	3
Device	IEEE 802.11g
Transmission power	30 mW
RTS/DATA	54 Mbps
CTS/ACK	24 Mbps
Application	2.5 Mbit/s On-Off
Packet size	$U(128, 1920)$ bytes
RTS threshold	1024 bytes

Common simulation parameters in Table I are used in all simulations, unless stated otherwise. All simulation results are averaged over ten simulation runs.

B. The Idealized Neighborhood Relationship

Under the PI model, the relationship between SINR and the probability of successful transmission is not a step function, where a transmission is always failed when SINR is below a certain threshold, and always successful when SINR is above this threshold. Since there is no threshold SINR in the PI model, there is no set maximum distance between transmitter and receiver and hence, the definition of which nodes are neighbors at any particular point in the simulation is not obvious. To evaluate how well the protocols maintain neighborhood information, we define an *idealized* neighborhood relationship between two nodes. Ideally, two nodes are considered neighbors if the distance between them maps to a specific SNR value or higher. The definition of the ideal neighbor distance is introduced as a means to evaluate how well the protocols maintain neighborhood information, but it does *not* affect the behavior of the protocols. In the simulations, a node considers another node as a bidirectional neighbor if it recognizes that there is a bidirectional link between them.

C. Evaluation Metrics

We evaluated two aspects of each protocol: neighborhood maintenance and reliability. To evaluate a protocol's ability to maintain neighborhood information, we added two metrics to ns-3. The first metric was used to record *neighbor add delay*, which is the difference between the time when a node recognizes a new neighbor and the time when the new neighbor actually moves within the ideal neighbor distance. If the node recognizes the new neighbor before the new neighbor moves within range, which is possible due to the

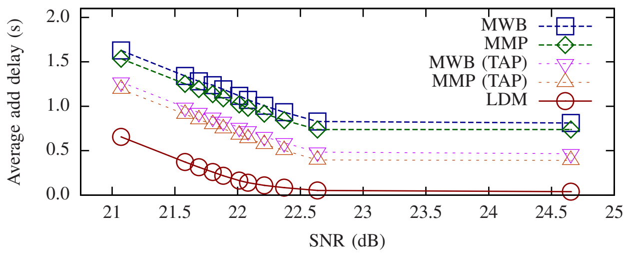

Figure 3. Add delay for each protocol under different SNR.

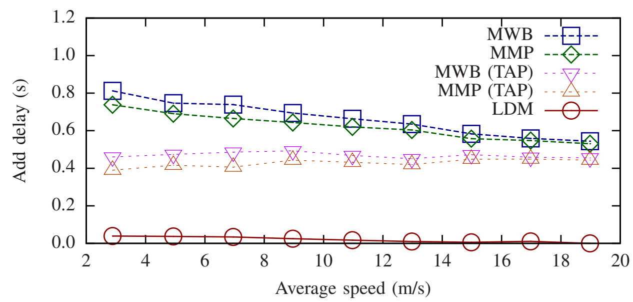

Figure 4. Average add delay of each protocol.

idealized definition of the neighbor distance, the neighbor add delay is set to zero.

The second metric was used to measure a protocol's ability to maintain neighborhood information in an environment where links' states are constantly changing. The second metric counts the number of transmissions resulting from a node that incorrectly classifies a unidirectional neighbor as a bidirectional neighbor. These transmissions waste bandwidth since the node is expecting an ACK from a unidirectional neighbor.

Multicast packet reception ratio (MPRR) was used to evaluate the protocol's reliability. Packet reception ratio for one receiver is defined as the total number of bytes received by that receiver divided by the number of bytes sent by a source during the time that the receiver was inside the ideal neighbor range from the source. MPRR is the average over the packet reception ratios for all ideal neighbors of the source.

To define the idealized neighborhood distance in the simulation, the add delay of the three protocols under different SNR values are reported in Figure 3. As seen from Figure 3, different ideal neighbor SNR (i.e. distance) yields different add delays. All subsequent simulation results are reported at the SNR value of 23.5 dB, which corresponds to an ideal neighbor distance of 41 meters under our simulation settings. Note that increasing or decreasing the chosen value by 1 dB has almost no impact on the add delay, meaning that the results are not very sensitive to this parameter value.

D. Speed of Link Discovery and Its Impact

In each simulation, forty nodes were placed randomly in the deployment area. Ten nodes were randomly selected as source nodes. One hop multicasts from the source nodes to nodes within their range were performed.

The add delays of different protocols are reported in Figure 4. Add delay was measured at the source nodes. We did not include non-source nodes in the evaluation since non-source nodes do not actively use their neighbor lists to transmit data. Therefore, a slight delay in maintaining a neighbor list does not affect the performance of those nodes. In a real network, where most nodes are active, either as original source nodes or as forwarding nodes, all active nodes act like source nodes from the MAC layer perspective.

The simulation results show that LDM is able to recognize a new neighbor that moves within its range significantly faster than other HELLO-based protocols. As can be seen from Figure 4, LDM has the shortest add delay among all three neighborhood maintenance mechanisms. MMP and MWB have longer add delays due to the nature of the HELLO mechanism; a node is required to receive a HELLO message from a new neighbor to recognize it, which results in some delay since HELLO messages are sent periodically. In addition, a lost HELLO message will further delay recognition of a new neighbor, since HELLO messages are sent unreliably without re-transmission.

The add delays of LDM are clustered closely around the mean whereas the add delays of MWB and MMP have higher variances. For instance, at the average speed of 2.86 m/s. LDM has an average add delay of 0.031 seconds with the standard deviation of 0.009 while MWB and MMP have average add delays of 0.812 seconds and 0.738 seconds and standard deviations of 0.31 and 0.27, respectively.

The multicast packet reception ratios of all the protocols are reported in Figure 5. As seen from the figure, LDM has the highest reliability among all the protocols. The reason for the difference between LDM and HELLO-based protocols is the ability of LDM to maintain better neighborhood information than the HELLO mechanism. MPRR decreased as average speed increased as maintaining up-to-date neighborhood information is more difficult when nodes are moving at higher speeds. In the worst case, where an ideal neighbor relationship lasts only briefly, nodes using a HELLO mechanism may not recognize the neighbor at all. TAP adjusts the HELLO message rate according to nodes' current speed to mitigate this problem, however, this also causes an increase in bandwidth consumption.

To study how the Join ACK probability affects add delay, the maximum Join ACK probability was varied from 0.2 to 1.0. The mapping function from Join ACK level to Join ACK value was set to: $\Pr(Join) = \frac{1}{8} \times P_{max} \times (1 + L)$. The average add delays at an average speed of 2.86 m/s are reported in Figure 6. The add delay of LDM increases as the maximum Join ACK probability decreases since the

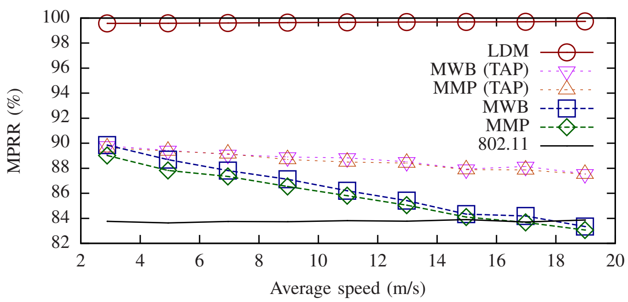

Figure 5. Multicast packet reception ratio for each protocol.

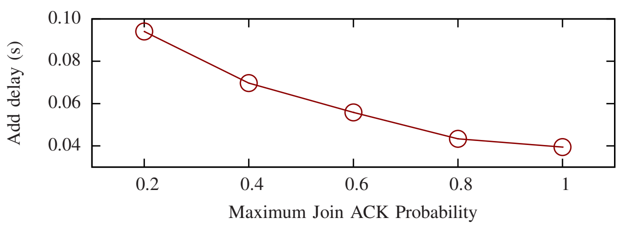

Figure 6. Add delay of LDM with different maximum Join ACK probabilities.

new node has lower probability to send a Join ACK to the source node. However, the average add delay of LDM is still significantly shorter than HELLO-based mechanisms.

E. Unidirectional vs. Bidirectional Links

In this section, we demonstrate that LDM has better ability to maintain neighborhood information than the HELLO mechanisms, particularly when links' states are frequently changing. In this simulation, 600 nodes were statically placed in the deployment area. Ten nodes were randomly selected as source nodes. A link between two nodes exists if they are separated by distance smaller than 41 m. Every link is either in a bidirectional state or a unidirectional state. The duration for the unidirectional state is uniformly distributed between 0 to 1 seconds. The duration for the bidirectional state is uniformly distributed between t_B to $t_B + 1$ seconds.

The total number of transmissions resulting from nodes that incorrectly classified neighbors as bidirectional are reported in Figure 7. Note that the y-axis is broken to better display the results. LDM has the lowest number of transmissions among all protocols. HELLO-based protocols have a higher number of false transmissions since nodes rely on HELLO messages and assume that links are bidirectional if a HELLO message is received. For instance, if a link between node A and node B is unidirectional from A to B, a HELLO message from A will be received by B. In this case, B will mistake A as a bidirectional neighbor. LDM does not assume that receiving a HELLO message from A indicates that the link is bidirectional and avoids this problem.

One possible approach to detect bidirectional neighbors in HELLO-based protocols is for the sender to include an incoming neighbor list in all HELLO messages. This approach was briefly mentioned in [26] although, to our knowledge, no existing implementations have adopted this technique. However, even if this technique is used, the delay in recognizing unidirectional links will be significantly higher than in our approach, because it could require several HELLO periods for the receiver on the unidirectional link to accurately determine that the sender of the link cannot receive its messages.

In the course of our experiments with the HELLO-based protocols, several other problems with their ability to distinguish incoming neighbors from bidirectional neighbors became apparent. First, in certain cases, the delay in recognizing a neighbor as bidirectional could be even higher than what was reported in Section IV-D. This problem arises in nodes that are not sending data (non-senders). The neighbor addition delay reported in Section IV-D was for the multicast sender, which is actually a best case for HELLO-based protocols. When nodes do not send data, the only opportunity for other nodes to recognize them as incoming neighbors is through their HELLO messages. Thus, when a non-sender node first moves into the neighborhood of another node, if the other node sends its HELLO message before the non-sender node sends its HELLO message, the other node will not yet have recognized the non-sender as an incoming neighbor and so it will not include the non-sender node in its neighbor list. Thus, when the non-sender node receives the first HELLO message from the other node, it will not recognize the other node as a bidirectional neighbor. Only after the non-sender sends its HELLO message will the other node recognize the non-sender as an incoming neighbor. The other node will then include the non-sender in its neighbor list in its *second* HELLO message. Thus, the delay for the non-sender to recognize the other node as a bidirectional neighbor could be as high as two HELLO message periods.

A second problem arises when two sender nodes move within range of each other. At the point of first moving within range, neither node is included in the other node's neighbor set, so their reliable multicasts will not be sent to each other. Furthermore, since both nodes have frames to send, they no longer send out HELLO messages. Therefore, A only recognizes B as an incoming neighbor and B only recognizes A as an incoming neighbor (A and B hear the transmissions from each other). Since A and B only recognize each other as incoming neighbors, they will continue to not include each other in their multicasts.

To try to address one or both of these problems, three possible options are: 1) nodes can continuously send out HELLO messages, 2) HELLO information can be piggy-backed onto DATA frames, or 3) nodes can assume that all links are bidirectional. Continuously sending out HELLO messages consumes more bandwidth. Piggybacking HELLO

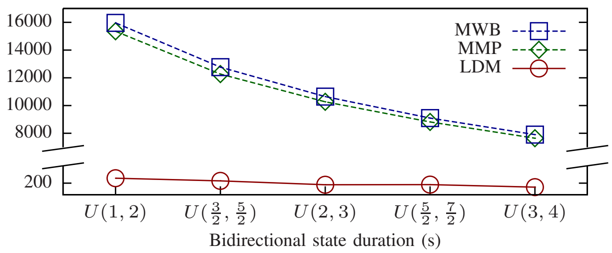

Figure 7. Total number of transmissions resulting from misclassification.

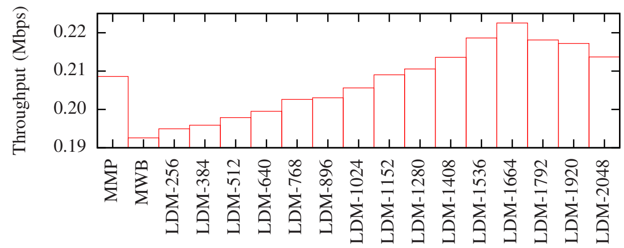

Figure 8. Average throughput for different protocols.

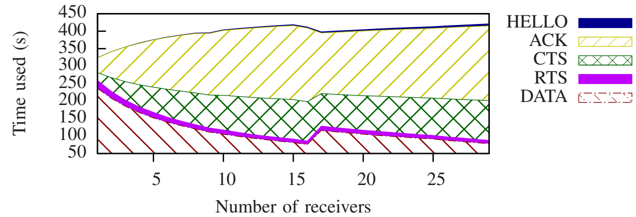

Figure 9. Transmission time used for different frame types.

information onto DATA frames requires all nodes to operate in a promiscuous mode. Finally, assuming that all links are bidirectional will result in errors in neighbor classification [24], [17], [18]. LDM does not suffer from these problems since nodes can send join ACK to each other if the link is bidirectional. If the link is unidirectional, say from A to B, B will receive frames from A but A will not receive Join ACK from B. Thus, B will recognize A only as an incoming neighbor while A will not recognize B at all.

F. Effect of the Application Traffic Model

The objective of the simulations reported in this section is to study the effect of different application traffic models on the performance of the LDM protocol. In this simulation, each source node has an on-off application. The duration of the on state is fixed to one second. The duration of the off state is uniformly distributed between t to $t+1$ seconds ($U(t, t+1)$). For LDM protocol, a HELLO message is sent if a node has not sent any DATA frame in the last one second.

The add delay of LDM increases from 0.031 s at the off state duration $U(1,2)$ to 0.058 s at the off state duration $U(4.5, 5.5)$. Since LDM relies on an ACK from the new neighbor to recognize its presence, the new neighbor must be recognized through an explicit HELLO message during the off state. Thus, the add delay of LDM increases as the off state duration increases. However, the delay is still smaller than it would be if a traditional HELLO mechanism was used, since the new neighbor can send an ACK to the HELLO message, which allows the HELLO sender to recognize the new neighbor as a bidirectional neighbor as soon as the ACK is received.

G. Threshold-based Transaction

In this simulation, we show that using frame size as a threshold for a four-way transaction results in better bandwidth utilization. In this simulation, 1000 nodes were statically placed in the deployment area. One hundred nodes were selected as source nodes. We varied the RTS threshold from 256 to 2048 bytes (LDM-*threshold*). The average throughput for each protocol is reported in Figure 8.

As seen from Figure 8, the RTS threshold in this case should be set to about 1664 bytes. Four-way transaction is useful when collision is likely. Setting the RTS threshold too low results in more transactions being protected by the RTS-CTS handshake than necessary. We note that selecting an appropriate RTS threshold depends on many factors, and thus should be left as a tunable parameter to be set by network administrators, which is the same practice recommended by the IEEE 802.11 standard for unicast frames.

H. Overhead of positive acknowledgement approach

Finally, we evaluate the overhead of a positive acknowledgement mechanism. A single source node is presented in this simulation. Varying number of receivers are placed within the transmission range of the source. The source node sends out packets at the rate of 30 Mbps. This simulation represents the *best*-case scenario where all receivers are within the transmission range of the source, and no contending application. The total time nodes used to transmit different frames are reported as stacked areas in Figure 9.

As seen from Figure 9, as the number of receivers increases, the total time used for transmitting control frames also increases. The total time used to transmit control frames exceeds the total time used to transmit DATA frames when more than four receivers are present. The increase in DATA transmission from 17 receivers to 18 receivers was due to the number of receivers' addresses is too large to be fitted into one transaction. In this case, the sender had to split the multicast transaction into two sub-transactions.

If a very large number of receivers are present and the multicast involves an access point and multiple receivers, an approach like the 802.11aa Block ACK can be used to reduce the overhead of the control messages.

V. Conclusions

We have proposed an extension to the IEEE 802.11 framework. The proposed protocol's goals are to provide an integrated MAC layer neighborhood maintenance with reliability multicast. Simulation results show that our proposed protocol is able to quickly recognize new neighbor nodes which results in higher reliability than protocols that rely on a traditional HELLO mechanism. Our protocol is also able to efficiently and quickly distinguish between unidirectional and bidirectional links. Future work includes attempting to extend the join ACK mechanism to unicast operations and reducing the overhead of the positive acknowledgement mechanism.

References

[1] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *Military Communications Conf.*, vol. 2, 2001, pp. 1008–1013.

[2] H. Gossain, N. Nandiraju, K. Anand, and D. P. Agrawal, "Supporting MAC layer multicast in IEEE 802.11 based MANETs: Issues and solutions," in *Int. Conf. Local Computer Networks*, 2004, pp. 172–179.

[3] S. Jain and S. R. Das, "MAC layer multicast in wireless multihop networks," in *Conf. Communication Systems Software and Middleware*, 2006.

[4] C. Campolo, C. Casetti, C.-F. Chiasserini, and A. Molinaro, "A multirate mac protocol for reliable multicast in multihop wireless networks," *Computer Networks*, vol. 56, no. 5, pp. 1554–1567, 2012.

[5] F. Kuhn, N. Lynch, and C. Newport, "The abstract MAC layer," in *Distributed Computing*, 2009, vol. 5805, pp. 48–62.

[6] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164–194, Jun. 2005.

[7] I. D. Chakeres and E. M. Belding-Royer, "The utility of hello messages for determining link connectivity," in *5th Int. Symp. Wireless Personal Multimedia Communications*, vol. 2, Oct. 2002, pp. 504–508.

[8] H. X. Tan and W. K. G. Seah, "Dynamically adapting mobile ad hoc routing protocols to improve scalability," in *The IASTED Intl. Conf. Communication Systems and Networks*, 2004.

[9] O. Stanze, M. Zitterbart, and C. Koch, "Mobility adaptive self-parameterization of routing protocols for mobile ad hoc networks," in *IEEE Wireless Communications and Networking Conf.*, vol. 1, Apr. 2006, pp. 276–281.

[10] F. Ingelrest, N. Mitton, and D. Simplot-Ryl, "A turnover based adaptive HELLO protocol for mobile ad hoc and sensor networks," in *15th Int. Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Oct. 2007, pp. 9–14.

[11] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. 2nd ACM Int. Symp. Mobile ad hoc networking & computing*, 2001, pp. 137–145.

[12] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 7, pp. 998–1016, 2007.

[13] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proc. 15th Annual Intl. Conf. Mobile computing and networking*, 2009.

[14] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. IEEE 20th Annu. Joint Conf. IEEE Computer and Communications Societies*, vol. 3, 2001, pp. 1388–1397.

[15] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," in *22nd Annu. Joint Conf. IEEE Computer and Communications.*, vol. 3, Mar. 2003, pp. 1702–1712.

[16] D. M. Blough, G. Resta, and P. Santi, "The k-neighbors approach to physical degree bounded and symmetric topology control in ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, pp. 1267–1281, Sep. 2006.

[17] R. Prakash, "Unidirectional links prove costly in wireless ad hoc networks," in *Proc. 3rd Intl. Workshop Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999, pp. 15–22.

[18] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proc. 2nd Int. Conf. Mobile Systems, Applications, and Services*, 2004, pp. 125–138.

[19] K. Yu and W. Choi, "A reliable multicast mac protocol using busy-tone for the IEEE 802.11-based wireless networks," in *Intl. Conf. Information Science and Applications*, Apr. 2011, pp. 1 –7.

[20] C.-Y. Chiu, E. H. kuang Wu, and G.-H. Chen, "A reliable and efficient MAC layer broadcast (multicast) protocol for mobile ad hoc networks," in *Global Telecommunications Conf.*, vol. 5, 2004, pp. 2802–2807.

[21] N. Choi, Y. Seok, T. Kwon, and Y. Choi, "Multicasting multimedia streams in IEEE 802.11 networks: a focus on reliability and rate adaptation," *Wireless Networks*, vol. 17, pp. 119–131, 2011.

[22] J. Kim, J. Jung, and J. Lim, "A reliable multicast mac protocol based on spread spectrum technique in wireless ad-hoc networks," in *Grid and Distributed Computing*, 2011, vol. 261, pp. 202–212.

[23] S. W. Kim, B.-S. Kim, and I. Lee, "MAC protocol for reliable multicast over multi-hop wireless ad hoc networks," *Journal of Communications and Networks*, vol. 14, no. 1, pp. 63 –74, Feb. 2012.

[24] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliot, "Experimental evaluation of wireless simulation assumptions," Dartmouth College, Tech. Rep., 2004.

[25] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *22nd Annu. Joint Conf. IEEE Computer and Communications*, vol. 2, 2003, pp. 1312–1321.

[26] C. E. Perkins and E. M. Belding-Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications*, Feb. 1999, pp. 90–100.