Abstract— In wireless ad-hoc networks, unidirectional links occur for several reasons: non uniform transmit power, non uniform background noise, and external interference. Several researchers have addressed unidirectional links and the associated unidirectional routing problem. The main focus has been so far on “unicast” routing; the consensus is that unidirectional links should be detected and avoided. In this paper, we consider the multicast case and derive a different conclusion: namely, it pays to exploit unidirectional links rather then avoid them. To prove the point, we select a popular ad hoc multicast protocol, On-Demand Multicast Routing Protocol (ODMRP) and introduce a slightly modified version, ODMRP-ASYM, that can handle unidirectional links. Specifically, ODMRP-ASYM reroutes the Join Reply packet when a unidirectional link is detected on the Join Query path. The option is invoked only when a unidirectional link is detected. The main advantages are: control overhead comparable with ODMRP even in highly asymmetric topologies; virtually no performance degradation in presence of unidirectional links (while ODMRP typically suffers up to 15% drop in delivery performance), and; 2-connectivity maintenance even if no bidirectional path exists between sender and receiver (in this case, unidirectional link avoidance strategies fail). Extensive simulation experiments demonstrate ODMRP-ASYM robustness to unidirectional links and superiority over conventional ODMRP.

Keywords— Ad hoc networks, Multicast, Routing, Simulation, Wireless networks, Unidirectional link.

I. Introduction and related work

Most routing protocols developed for wireless ad-hoc networks do not consider the existence of asymmetric links - namely, links with different characteristics (e.g., range, quality, etc) in the two directions. To define asymmetry, consider two nodes A and B, say, transmitting to each other. The link is asymmetric if the signal/noise ratio at A is different from that at B. Most common cause of this asymmetry is the imbalance in transmit power from A and B, but other causes may be the presence of external interference and high background noise at one of the stations. In some cases the asymmetry is intentional, i.e., it is planned, such as with directional antennas [2] and power (or topology) control algorithms [1][8][9]. In some wireless ad-hoc network scenarios, such as sensor networks with intrinsically limited resources, it is important to use power efficiently so that these networks can have longer lifetimes. Most power control protocols disallow [9] or disregard [1] asymmetric links, yet some retain asymmetric links [13]. For given transmit power conditions, the asymmetric link becomes a unidirectional link, i.e., a signal transmitted by A is received at B, but not vice versa. In the following we will assume this case, i.e., an asymmetric link is a unidirectional link. Asymmetric and unidirectional links are often found in the real deployment of wireless ad-hoc networks as shown in recent experimental studies [4][17].

Unidirectionality has an important impact on routing. For example, on-demand routing protocols that use a reverse path technique (e.g., AODV) to retrace the route back to the source will not work if they are faced with unidirectional links in the route discovery stage. This problem is referred to as the asymmetric/unidirectional link problem in routing. Recently, a number of researchers investigated the impact of unidirectional links on the performance of unicast routing protocols and developed new routing protocols to handle the problem [6][8][16].

The most common approach to tackle unidirectional links is to simply detect and avoid them. This approach is justified for unicast routing. In fact, data transmission in the IEEE 802.11 DCF MAC (the most common MAC today) in unicast mode requires an ACK handshake and often also RTS/CTS handshaking – this is feasible only on bidirectional links. This constraint is relaxed if the MAC is used in broadcast mode, which is the prevalent mode in multicast protocols. In this case, the avoidance of unidirectional links restricts the routing options and may in fact cause the failure of route discovery even when the topology is bi-connected, i.e., there is a (possibly unidirectional) path in each direction between any node pair. For instance, in [12] it was shown that the contribution of unidirectional links to connectivity is not negligible, and that their elimination may cause network partitions.

In this paper we consider multicast, thus we are not bound by symmetric links for MAC considerations. Our approach is to make use of unidirectional links instead of “avoiding” them. Previous approaches that attempt to utilize unidirectional links were mainly designed for “unicast” routing and are not efficient in the multicast context. For example, the two-way flooding scheme to find forwarding and reverse paths respectively as presented in [6] might face flood explosion if there is a large number of receivers.

The basic idea in our proposal is quite simple: suppose the Join Query procedure initially selects a path with a unidirectional link. During Join Reply a detour path is then found in the reverse direction so that the combination of the two paths reestablishes bi-connectivity. Our protocol can be implemented in on-demand multicast protocols such as ODMRP (On-Demand Multicast Routing Protocol) [7] and

This work is supported in part by ONR “MINUTEMAN” project under contract N00014-01-C-0016.
MAODV (Multicast AODV) [14]. As a first step, in the paper, we propose an extension of ODMRP called ODMRP-ASYM (ODMRP with ASYMmetric link support). We modify the Join Reply phase of ODMRP to find a detour path to deliver the Join Reply packet if the incoming path is unidirectional. Our protocol does not require to maintain the link status by exchanging periodic hello messages or other schemes. Rather, we detect unidirectional links only when we try to use them, by monitoring ACK failures.

Our scheme meets two important design goals: (a) achieve complete route discovery and; (b) incur no overhead if there is no unidirectional link in the network. To meet the first goal, we show that if a source-destination pair is bi-connected (and yet no bi-directional path exists), the scheme always finds a pair of unidirectional paths, one in each direction. Secondly, no extra overhead is introduced since the unidirectional link extension is activated only when one such link is detected; this occurs “on the fly” during path setup. This feature is important since the probability of unidirectional link is small in most applications. In contrast, the current solutions that “avoid” unidirectional links [8][16] pay a fixed overhead for unidirectional detection regardless of asymmetric links existence. The combination of aggressive use of unidirectional links and low overhead pays good dividends: simulation shows that ODMRP-ASYM considerably improves performance over ODMRP in networks with non-uniform transmit power, incurring only marginal overhead increase. To the best of our knowledge, ODMRP-ASYM is the first multicast routing scheme exploiting asymmetric links.

The multicast protocol lends itself ideally to the use of unidirectional links because of the IEEE 802.11 DCF operation in broadcast mode and the ability to deliver data packets to designated receivers even on one-way routes unlike unicast transmissions which require symmetric links for RTS/CTS support. In principle, our approach can also be applied to unicast routing protocols. The key condition is a MAC that does not require link bi-directionality. For instance an 802.11b MAC protocol that operates in broadcast mode (without RTS/CTS handshaking and ACK); a TDMA based MAC protocol with reserved slots, etc. In this case, one gives up link-level reliability. However, recent results [2][11] show how to regain link reliability by detouring acknowledgements and even RTS/CTS in case of asymmetric links. Another approach is to delegate recovery to the transport layer. We are currently exploring several such solutions for unidirectional unicast routing.

The rest of the paper is organized as follows: Section II describes our protocol; Section III presents simulation results, and; Section IV concludes the paper.

II. Supporting unidirectional links in wireless ad-hoc networks

In this section we extend ODMRP to address the unidirectional link problem in multicast routing. First we give a brief overview of ODMRP and then discuss ODMRP-ASYM.

A. Overview of ODMRP

In ODMRP [7], the data source establishes and updates group membership and multicast routes. ODMRP introduces the concept of forwarding group: a set of nodes which is responsible for forwarding multicast data, essentially forming a mesh structure between all senders and receivers. In ODMRP, a soft-state approach is taken to maintain multicast group members; no explicit control message is required to join or leave the group.

Similar to on-demand unicast routing protocols, Query and Reply phases make up the protocol. While a source has packets to send, it periodically floods member-advertising packets, called Join Query. The periodic floods refresh membership information and update the routes. The Join Query packet uses the MAC in broadcast mode.

Upon receiving a non-duplicate Join Query packet, every node in the network stores the upstream node address, i.e., reverse path learning, into the route table and rebroadcasts the packet to its neighbor nodes. When the Join Query packet reaches a multicast receiver, the receiver creates and broadcasts a Join Reply to its neighbors. This Join Reply packet is propagated all the way back to the source following the learned reverse path. Join Reply uses the unicast MAC version. Nodes on the reverse path become the forwarding group, i.e., have the Forwarding Group Flag set. During the data phase, Forwarding Group nodes rebroadcast the packets belonging to the associated Membership Group. For example, if the group consists of only two nodes (sender and receiver), ODMRP will flag nodes along the shortest path as forwarding nodes. These nodes will then deliver packets from source to destination virtually implementing unicast routing as a special case of multicast.

B. ODMRP-ASYM: An extension to ODMRP for asymmetric link support

ODMRP-ASYM is designed with the goals to achieve complete route discovery by utilizing unidirectional links and to detect such links on the fly preserving the on-demand nature of ODMRP. No additional overhead is introduced when there are no unidirectional links.

Figure 1: A network with asymmetric links

To understand how ODMRP-ASYM works, consider the example in Figure 1. There are two asymmetric paths between \( S \) and \( D \), namely \( S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \) and \( D \rightarrow C \rightarrow E \rightarrow A \rightarrow S \), and there is a loop which spans a part of
each path. In Figure 1, the loop is \{A, B, C, E\}. By construction, when two nodes are connected by two asymmetric paths, one in each direction, there must be one or more loops across these paths.

Suppose source S wants to create a path to destination D. In ODMRP, creating a path means *flagging* all the nodes along the path as forwarding nodes (e.g., set forwarding group flag on Node A, B, and C in Figure 1). Our solution, upon detecting a unidirectional link, will *discover* a detour loop and will flag nodes that are part of the path from S to D in the loop. To start, the Join Query packet is flooded from S into the network. As the Join Query packet proceeds toward D, pointers are set to the predecessors along the path. The flood reaches D through the path S→A→B→C→D. Upon receiving the Join Query packet, the destination D sends back a Join Reply packet using the reverse path. However, the reverse path D→C→B→A→S is obviously blocked at node B as there is no link in the direction B→A. At this point comes the novelty of our scheme. By a mechanism to be illustrated later in more detail, node B detects the failure to deliver the Join Reply to A and initiates the *Loop Discovery* procedure. Loop Discovery procedure will find the loop (B→C→E→A→B) and select the detour path (B→C→E→A) to reach source S. One can easily observe that our scheme does not introduce any extra overhead if there is no blocking, i.e., no unidirectional link.

**Loop Discovery**

The goal is to find an alternate path from the blocked node with no link to its upstream node (i.e., the node it received the Join Query from). A packet called Loop Discovery Packet (LDP) is flooded (with broadcast MAC) with TTL (Time-To-Live) set to the expected maximum size of the loop. In our experiments we set TTL = 6. In the worst case TTL = N, the number of nodes in the network. In practice, TTL can be increased after each failed attempt using an expanding ring search. If no loop is discovered with TTL = N, the source and destination are not bi-connected and the procedure quits. The LDP has a field which stacks the node IDs in the order they are visited, namely *IDlist*. For example, LDP collects “BCE” if the packet travels the path B→C→E. On reception of an LDP, each node adds its own ID number to *IDlist* and rebroadcasts. When a node detects that its ID number is already in *IDlist*, it has discovered a loop. Some of the discovered loops might be invalid. In this context, a loop is invalid if it does not help make progress towards the source (e.g., B→F→G→B in Figure 2). To discard invalid loops, we need a loop validation and selection scheme. For this purpose, LDP includes a field named *loopsumit*. (This field has another important role which will be described later.) The field *loopsumit* is basically a pointer to an ID in *IDlist*. A returned LDP is valid if *loopsumit* contains an actual pointer to an ID in *IDlist*. The update procedure is as follows. Each node learns its distance (i.e., hop count) from the source when it receives the Join Query packet generated by the source. The LDP field *hc2src* (hop count to source) is initialized with the distance to source of the LDP initiator. Each intermediate node compares its hop distance to *hc2src* when stacking its ID. If its hop distance to source is less than *hc2src*, it enters its stack pointer in the *loopsumit* field and updates the value of *hc2src* with its own distance. In the example, when node A receives the LDP, it enters its ID in the stack which now becomes “BCEA”. It then finds that its distance to S is less than the current *hc2src* value, and sets *loopsumit* = 4 and *hc2src* to its own distance.

This way, the LDP initiator accepts only loops in which one node is closer to source than the initiator itself. This guarantees convergence. Note that the LDP is not restricted to find an alternate path to an upstream node of the initiator. It rather finds a detour path through which a Join Reply packet can be delivered closer to the source.

Following a successful loop discovery (i.e., Upon receiving the first valid LDP packet), B carries out the *Loop Marking* procedure by circulating a Loop Marking Packet (LMP) around the loop. An LMP is made from the corresponding LDP: LMP has two fields, *IDlist* and *loopsumit*, which are copies of those in LDP. LMP is source-routed; it steps through the nodes taken from *IDlist* (in the order inserted when *IDlist* is originally constructed in LDP). The node pointed to by *loopsumit* (i.e., the node closest to source) restarts the Join Reply procedure towards the source. The nodes after *loopsumit* in *IDlist* are all flagged as forwarders. In our example, since there is no other node after A in the LMP’s *IDlist*, only A sets its forwarding flag on and send out the Join Reply packet which now reaches source S directly.

![Figure 2: Illustration of Loop Discovery Procedure](image-url)
search packet, it sends an ACK to B and restarts the Join Reply process. Convergence however will be slower as this scheme cannot take advantage of big leaps that expedite the route setup procedure.

**Detecting Unidirectional Links**

In the original ODMRP every Join Reply packet is acknowledged since the reliable delivery of the Join Query packets is critical for establishing forwarding paths. If a Join Reply is not acknowledged, it is retransmitted at most two times. We utilize this ACK scheme to detect unidirectional links. ODMRP-ASYM initiates the loop detection procedure if a Join Reply is not ACKed after the second retransmission.

**C. Refining loop discovery in ODMRP-ASYM**

The loop discovery is quite expensive because of flooding. In [12], it is observed that, given a unidirectional link B→A, the typical length of the reverse path, or detour around the B→A link is 2 hops. This property can be exploited to reduce flooding overhead. One straightforward strategy is to set the initial TTL = 3 (note: if the typical reverse path is 2 hops, the typical loop length is 3 hops). A more efficient optimization is obtained with a simple modification of the ODMRP-ASYM Join Reply handling rule. Namely, upon overhearing the second retransmission of the Join Reply packet by Node A, say, a neighbor of A will rebroadcast the Join Reply (without modifying its header) if the Join Reply “next node” address matches the address of its upstream neighbors (say, the neighbor from which it received the Join Query). Basically, the above rule requests the assistance of intermediate nodes to act as forwarders in case of unidirectional problems. Note, however, that the Forwarding Group Flag is not set on the rebroadcasting node. In the original ODMRP protocol a node cannot rebroadcast a Join Reply unless it matches its address. Figure 3 illustrates the new rule. A dot represents a node and the surrounding dashed line represents its radio transmission range. Let S and D be the source and the destination respectively. C transmit at high power such that C can reach A, but A cannot get to C.

In this topology, a Join Query will propagate S→C→A→D. In response to the Join Query, D generates a Join Reply. With the new rule, after C fails twice to return the ACK, node B takes over and rebroadcasts to C. In our example C then sends an ACK to A. With the original rule A has no choice but to initiate the Loop Discovery procedure; loop ABCA is found and the Join Reply is properly forwarded to S, albeit with more overhead than with the new rule. If the new rule fails to generate the ACK, Loop Discovery is entered. Note that the new rule introduces only minimal overhead (in case it also fails).

The example in Figure 3 illustrates another interesting property that differentiates our method from the “unidirectional link avoidance” method. With the avoidance method, the link C→A is detected as unidirectional link during the Join Query phase and is dropped from consideration. The Join Query must then propagate through intermediate node B, which gets flagged and will act as forwarding node. With ODMRP-ASYM, node B never gets flagged. As a result, during the data phase, our scheme forwards the packet from C to A directly. Node B remains silent. With the “avoidance” scheme, both B and A accept the packet from C, and forward it at the same time - to A and D respectively - causing extra contention and congestion.

**Figure 3: Illustration of the new Join Reply handling rule**

**III. Simulation results**

In this section, we study the performance benefits of ODMRP-ASYM through extensive simulation experiments using the QualNet simulation platform [15]. We assume that link asymmetry is caused by non uniform transmission power. In the first set of experiments the uneven power results from the K-NEIGH [1] power control protocol; in the second set, transmission power is randomly set to either a high or a low value. We compare ODMRP-ASYM to the original ODMRP which includes no handling of unidirectional links. At this writing, there is no other multicast protocol that deals with unidirectional links. In the future, we plan to retrofit ODMRP with the unidirectional link avoidance mechanism. Likewise, we plan to equip MAODV with both uni-link avoidance and uni-link exploitation. That will provide richer terms of comparison. The current comparison is important in that it tells us how much we stand to gain over off the shelf ODMRP implementations.

Since the ODMRP version used here for comparison DOES NOT provide, for unidirectional link removal, we must be aware of the consequences. In particular, if the Join Query arrives to a receiver on a “unidirectional path”, the Join Reply is blocked and that path does not get flagged as forwarding path. Thus, the receiver may never get its packets (although the intrinsic redundancy of the ODMRP mesh tends to alleviate this problem as well, especially if there are multiple sources and receivers).

The details of the simulation scenario are as follows. We randomly place 50 nodes within a square area. The size varies from 1.5x1.5 km² to 3.5x3.5 km² resulting in node density from about 22 to 4 nodes/km². A single multicast
We use the scenarios, the random mobility model is used. Over 10 runs of 360 second simulated time each. In mobile per second, to 10 group members. All results are averaged source generates constant bit rate traffic, four 512B packets node density when ratios of ODMRP-ASYM and ODMRP as functions of the decrease steeply when the node density goes less than 10 node/km². Below this density value the nodes are too sparse even for the maximum radio power to maintain connectivity.

Next, we study random power assignment. We use two-power model in which two different levels of transmission power are assigned to each node with equal probability. The two power levels used in simulations are 15dBm and 23dBm, which correspond to the transmission ranges of 376m and 597m respectively. The power level selection reflects typical transmission powers and ranges of commercial 802.11b radio cards.

Figure 5 shows that ODMRP-ASYM delivers again near 100% of the packets when connectivity is reached, confirming its ability to deal with unidirectional links resulting from random power assignment. In fact, ODMRP-ASYM’s packet delivery ratio is close to that of flooding, which provides an upper bound on delivery ratio of any routing/multicast protocol (at light offered load). ODMRP again fails to recover from unidirectional links and drops about 15% of the packet. By comparing Figure 4 and 5, we notice that the packet delivery ratio of ODMRP is about the same regardless of the source of the asymmetric links (K-NEIGH power control or random power allocation). Therefore, throughout the rest of this section we just limit ourselves to the two-power model.

Figure 6 demonstrate the performance of ODMRP-ASYM in a mobile scenario. Similar to the static scenario (Figure 5), ODMRP-ASYM does very well with unidirectional links. On average, ODMRP-ASYM delivers 15% more packets to receivers than ODMRP even in a mobile environment.

The overhead of ODMRP-ASYM is reported in Figure 7 and Figure 8. We first measure the normalized control overhead defined as the number of control packets issued divided by the total number of delivered packets. The
smaller the control overhead, the more efficient the scheme in the control phase. Figure 7 shows that the normalized control overheads of ODMRP and ODMRP-ASYM in the static scenario are very close, which leads to the conclusion that ODMRP-ASYM extends the support to asymmetric topologies with minimal extra control. The normalized forwarding overhead is defined as the ratio of the total number of packets forwarded to the total number of packets delivered. The smaller the forwarding overhead, the more efficient the scheme in the data phase. Figure 8 shows that ODMRP-ASYM has about 10% higher forwarding overhead than ODMRP. This is due to the fact that ODMRP drops packets on unidirectional paths, while ODMRP-ASYM does an extra effort and finds longer detour paths (hence extra overhead) to get packets through. The forwarding overhead is reduced as the network density gets higher as the average distance (i.e., the number of selected forwarding nodes from the source to each receiver) decreases.

Lastly, we measure performance as a function of the number of receivers in the multicast group in a static network with fixed density (12 nodes/km²). Larger number of receivers means more forwarding nodes and lower control overhead per node. Thus performance is expected to improve as number increases. Figure 9 and 10 confirm our hypothesis. In Figure 9, the delivery ratio of ODMRP-ASYM is 15% higher than that of ODMRP as seen before. In Figure 10, the normalized control overhead of ODMRP-ASYM becomes slightly higher than that of ODMRP as the number of unidirectional links increases with the number of receivers.

### IV. Conclusions

In this paper, we propose a new protocol, ODMRP-ASYM, which extends ODMRP to asymmetric topologies and unidirectional links. Our approach detours the Join Reply packet when a unidirectional link is detected. Using this approach, we manage to utilize unidirectional links as opposed to avoiding them as customary in unicast routing. The major contributions of this paper are the following: (1) ODMRP-ASYM maintains excellent performance even when unidirectional links are introduced by power control (while the conventional ODMRP suffers a 15% degradation); (2) the overhead introduced by ODMRP-ASYM is very modest with respect to conventional ODMRP especially when optimization techniques are used, and; (3) ODMRP-ASYM can establish 2-way connections between source and destination using two unidirectional paths if such paths exist. Property (3) puts ODMRP-ASYM ahead of any scheme that simply avoids unidirectional links, since such scheme would fail when bidirectional connectivity exists, but cannot be implemented via bidirectional paths.

### References