On the Effect of One-way Links on Route Discovery in DSR

Kulasekaran A. Sivakumar and Mahalingam Ramkumar
Department of Computer Science and Engineering
Mississippi State University, Mississippi State, MS 39762.
⟨sa151,ramkumar⟩@cse.msstate.edu

***Abstract*— One-way links are an unavoidable reality in wireless channels. The detrimental effects of one way links on many routing strategies, including the dynamic source routing protocol (DSR), have been investigated by many researchers. Apart from reducing the efficiency of DSR, we point out that disregarding possible one-way links also has some implications on the *security* of secure DSR extensions. Most secure extensions of DSR *rely* on the assumption that all links are bidirectional. We point out proactive strategies to indicate one-way links in route request (RREQ) packets to limit their use, with the intention of simultaneously improving the efficiency *and* security of secure DSR protocols.**

I. Introduction

The problem of efficient co-operative routing in mobile ad hoc networks (MANETs) [1] has received significant interest in the recent years. Most popular ad hoc routing protocols in the literature when originally proposed, did not take security issues into consideration. The problem of securing ad hoc routing protocols - or increasing their resilience to malicious nodes - has (deservedly) received much attention more recently.

It is well known that the choice and limitations of the MAC layer will have a significant effect on the efficiency [3], [4] and security [5] of any ad hoc routing protocol. Omnidirectional wireless broadcasts have the advantage of the ability to reach multiple nodes efficiently, with a single transmission. At the same time, efficient measures for reducing collisions, and ensuring reliable delivery of packets to *intended* nodes is more challenging. This is especially more so due to the inevitable presence of one-way links. Even if all transceivers have been manufactured to meet the same specifications, due to various reasons like differences in transmission range, receiver sensitivity, and aging, a node B may be out of the range of A's transmission, while node A may be within the range of node B.

The ill-effects of unidirectional links on the performance of various ad hoc routing protocols, and solutions to overcome this problem have received considerable attention [6] - [13] in the literature, which however assume that nodes co-operate with each other. On the other hand, *secure* routing protocols which consider the effect of non-cooperation (and include proactive steps to mitigate the detrimental effects of non-cooperation) have not factored in the possibility of one-way links.

In this paper we restrict ourselves to the dynamic source routing (DSR) protocol [2]. Though in its original incarnation DSR does not assume bidirectional links, all secure extensions of DSR in the literature assume that links are bidirectional. This assumption is *mandatory* for the security of such protocols as the integrity of established paths can be checked only if responses to route requests traverse through the same path (in the reverse order). Unfortunately, route requests (RREQ) that are flooded in the forward path, can reach "neighbors" that do not have a reverse path.

It is important to note that while medium access control (MAC) layer protocols like MACA (employed in 802.11) can avoid the use of unidirectional links by exchanging "request to send" (RTS) and "cleared to send" (CTS) packets, this is possible *only* for packets *unicast* from a node to a *single* destination (where the source sends an RTS probe and the destination replies with a CTS). RREQs, which are broadcast to all nodes within range, and the RTS / CTS packets themselves, are constrained to use carrier sense multiple access (CSMA), where nodes can transmit as long as they do not hear an ongoing transmission. In other words, RREQ packets (and RTS / CTS packets) can reach "neighbors" who do *not* have a *direct* reverse path to the source. Unless such "neighbors" refrain from forwarding the RREQ further, the route responses (RREPs) through those paths will fail. Unless the routing strategy *explicitly* addresses techniques to recover from such failures, the overall efficiency will suffer.

Apart from reducing the efficiency of DSR (by mandating repeated RREQs when the first one fails) ignoring the possibility of unidirectional links also has some implications on the *security* of route discovery process. We show that malicious nodes may be able to engineer attacks either by 1) taking advantage of unidirectional links, or even by 2) *pretending* that some links are unidirectional.

The specific contributions of this paper are three-fold: 1) a qualitative and quantitative (through simulations) study of the effect of one-way links on the efficiency of route discovery; 2) a discussion of the implications of one-way links on the security of route discovery, and 3) simple proactive approaches, amenable to existing secure DSR-like protocols, to mitigate problems arising due to one-way links.

In Section II of this paper we provide a brief overview of DSR, a brief summary of some secure extensions of DSR proposed in the literature. A qualitative discussion of the effects of one-way links on the efficiency and security of DSR is the subject of Section III. Proactive measures to overcome

1-4244-0572-6/06/$20.00 ©2006 IEEE

the limitations, and quantitative evaluation of efficiency, with and without such proactive measures, are presented in Section IV. Conclusions are offered in Section V.

II. The Dynamic Source Routing Protocol

DSR is an on-demand protocol where nodes find a route to desired destinations as and when required. The route discovery process starts by broadcasting a route request (RREQ) packet indicating amongst other things, the source S, the destination D, a unique sequence number and the maximum number of hops to which the RREQ packet may be flooded. The sequence number is used to keep the flooding in check - or to ensure that nodes do not re-broadcast the same RREQ multiple times.

Each node flooding the RREQ packet further, appends its ID / network address to it. When the RREQ packet reaches the destination (or some node which has the knowledge of a path to the destination) a route response (RREP) packet is initiated along the reverse path, as each hop is explicitly indicated in the RREQ.

A. Securing Route Discovery in DSR

Typical approaches to securing route discovery in DSR involve addition of cryptographic authentication to the DSR protocol. Cryptographic authentication employs security associations (SA) facilitated by key distribution schemes (KDS). Such SAs could be one-to-one (mutual or pair-wise authentication) or one-to-many (broadcast authentication).

Papadimitros [15] et al propose DSR-like secure routing protocol (SRP) where only the source and destination share a secret. Marshall et al [16] point out that SRP cannot avoid malicious behavior by intermediate nodes during the route establishment phase, as long as the (malicious) behavior is consistent in the forward and reverse path. They also suggest techniques to mitigate issues in SRP by employing promiscuous mode of operation [17] - [18]. In Ariadne, Hu et al [14] employ a per-hop hashing technique to prevent nodes from deleting upstream nodes from the path, and TESLA [19] for authentication of intermediate nodes, to prevent malicious insertion of nodes in the path. Kim et al [20] (SRDP) propose a general protocol for securing route discovery in DSR, where the primary deviation from Ariadne is that they strive to reduce the bandwidth overheads by *aggregating* signatures.

From the perspective of assuring the integrity of established routes, cryptographic authentication strives to ensure that it is not possible for nodes to

1) insert fictitious nodes in the path, and
2) delete nodes from the path

Obviously, cryptographic authentication alone is not sufficient to ensure that such paths can be *used*. Cryptographic techniques *cannot* prevent nodes from misbehaving once they are in the path. This calls for some assurances of trustworthiness of devices that have the capability to cryptographically authenticate themselves. In this paper, we shall not belabor on this issue any further. Insertion attacks are prevented by requiring nodes to append cryptographic authentication information when they modify RREQ packets before relaying them down-stream. Ideally, such authentication information will need to be carried over until the destination (or verified by all down-stream nodes, including the destination).

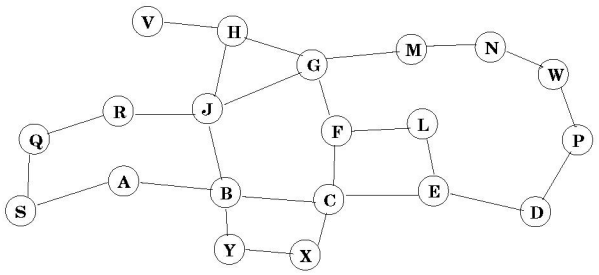

Fig. 1. Topology of an ad hoc subnet used for illustrations.

Prevention of deletion attacks, however is more challenging. Hu et al [14] proposed an elegant solution to address this problem, that makes use of *per-hop hashing* of a quantity known only to the source and destination (which will be discussed in the Section III).

III. Effect of Asymmetric Links on DSR

It is pertinent to point out here that in both SRP [15] and Ariadne [14], ensuring integrity of the path relies on the assumption that the RREQ and RREP take the same path. The unstated assumption is that nodes that do not have a direct reverse path to an upstream node (from which an RREQ was received) do not forward the RREQs further. In the absence of proactive strategies to realize this requirement, RREPs invoked in response to RREQs that include one-way links will fail.

While it may seem at first sight that the reduction in efficiency may be negligible if the fraction of such one-way links are low[1], we shall argue that this is not necessarily true. In this section we provide qualitative rationale for the reduction in efficiency (a more quantitative discussion, through simulations, is presented in Section IV).

Furthermore, we also argue that failing to take such one-way channels into account, has some implications on the security of the route discovery process. Thus, we argue, secure DSR extensions cannot simply afford to assume bidirectional links. Proactive approaches are required to identify such one-way links and warn downstream nodes to take appropriate action *before* RREQs are flooded.

A. Efficiency of DSR with One Way Links

Consider the topology of an ad hoc network shown in Figure 1. A RREQ from node S reaches D through the shortest path $P_{SD} = S \rightarrow A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$. While D may also receive the RREQ by S through other paths, typically node D would initiate a RREP along the shortest path P_{SD}. Let us assume that the link $B \rightarrow C$ is one-way (or $C \rightarrow B$ does not exist), resulting in failure of RREP. Two possible solutions in this case are

[1]For example, in the case of homogenous networks where all nodes are manufactured to meet the same specifications.

① C can notify failure to E, which could be passed on to D. Also note that if D has t neighbors, it can receive upto t RREQs for each request. If the destination also caches other RREQs it receives from S, the destination may then (after a time-out period) invoke another RREP along a different path. Alternately,

② it may be useful for the source and destination to simultaneously employ many parallel paths. Thus the destination D may invoke an RREP corresponding to many RREQs it receives (along different paths).

Once again, it would appear that the probability that *none* of t paths are usable, would be low enough to be ignored, especially if the number of one-way links is a *small fraction* of the total links. Unfortunately, even this is not true. For example, note that the path $S \rightarrow Q \rightarrow R \rightarrow J \rightarrow G \rightarrow F \rightarrow L \rightarrow E$ is also rendered unusable as E would have dropped the RREQ from L. The RREQ from L would have been *preempted* by the RREQ from C (corresponding to path P_{SD}, which turns out to be unusable). Ironically, shorter (faster) paths between two nodes are more likely to include possible one-way links (as each *hop* may be longer on an average), which could preempt good paths.

The end result is that even though *many good paths may exist*, it is likely that nodes *may not be able to discover any* of them, thus calling for a fresh RREQ from the source (after the original times out). If there is no significant change in the topology of the network (between the two requests) it is very much possible that P_{SD} *still remains the route* over which D initiates RREP. In other words, it is possible that S *repeatedly* fails in its attempt to discover a path to D, *even though several other paths exist*. What is required therefore, to be able to *recover* from a situation like this, is the ability of B to undertake some measure to ensure that C cannot forward the RREQ further (while other neighbors of B should).

B. Implications on Security

Ideally, in the scenario discussed earlier, (where $B \rightarrow C$ is one-way), node C should have recognized a priori that B is out of its range, and thus should not have forwarded the RREQ. However, if C is a malicious node, it can forward the RREQ without facing any risk of being *recognized* as malicious. Even under scenarios where nodes employ promiscuous mode to "keep tabs on neighbors," all that E can do (in the reverse path) is to verify that C *attempted* to send the RREP to B. However, as B is out of C's range, B does not recognize the attempt. A malicious C does not have to report failure to E.

The possibility of one-way links also provides attackers with an additional dimension of freedom - the ability to *pretend* that some of their links are one way. For instance, the link between B and C may actually be bidirectional, while C can pretend that it is out of B's range. This provides a mechanism for attackers to accomplish node deletion attacks, as explained below.

1) Per-hop Hashing in Ariadne: Thwarting node deletion attacks using per-hop hashing technique in Ariadne [14] is achieved by choosing a value $\beta_0 = f(K_{SD})$, where β_0 is a one-way function of a secret K_{SD} shared between the source and the destination (other known values in the RREQ like source, destination, maximum hop count and sequence number can also used for evaluating the public function $f()$ - what is important is that both S and D can determine β_0, while no other node can).

Along with RREQ, node S also broadcasts β_0 to its one-hop neighbors. Or node S broadcasts $(RREQ, (), \beta_0)$. Node A broadcasts $(RREQ, (A), \beta_1)$, where $\beta_1 = h(\beta_0, A)$. Node B in turn broadcasts $(RREQ, (A, B), \beta_2)$, where $\beta_2 = h(\beta_1, B)$. The destination (which can evaluate β_0) can ensure that the final β_5 value it receives in $(RREQ, (A, B, C, D, E), \beta_5)$ is consistent with the IDs of the nodes in the path. Note that in order to remove node B from the path, C needs access to the value $\beta_1 = h(\beta_0, A)$ that only *neighbors* of A are privy to.

In other words, the per-hop hashing technique that prevents node deletion attacks is based on the premise that nodes that are *not* neighbors, *cannot* hear the corresponding β_i value. More specifically, in the scenario where C pretends to be out of B's range, the value $\beta_2 = h(\beta_1 \parallel B)$ transmitted by B is known to C. Node C may also receive an RREQ along the path $S \rightarrow A \rightarrow B \rightarrow Y \rightarrow X$ from X, with a value $\beta'_4 = h(\beta'_3 \parallel X)$ where $\beta'_3 = h(\beta_2 \parallel Y)$. Now with access to β_2 over the one way link, a malicious C can remove Y and X or just X from the path[2] (which *cannot* be recognized by the destination).

In this case, dropping both X and Y is not in anyway useful for a malicious C. However if C drops X from the path (and advertises $B \rightarrow Y \rightarrow C$) there is still no way for any node to recognize C's maliciousness *even* by operating in the promiscuous mode. After all, from X's perspective, there *could* be direct path from Y to C. Thus during the reverse path, C can make a "sincere" attempt to unicast the packet to Y, knowing that the attempt will fail.

It is important to note that even if nodes use RTS / CTS exchanges for confirming bidirectional paths, and *individually* unicast RREQs[3] to neighbors, this still does not prevent node C from gaining access to the value β_2. Thus, unless proactive measures are taken to recognize one-way paths, node deletion attacks cannot be prevented.

C. Proactive Measures for Indicating One-way Paths

An example of such a proactive measure is for B to explicitly include a warning in the RREQ it forwards, that the link $B \rightarrow C$ may unreliable. The rationale used by B to conclude that link $B \rightarrow C$ is unreliable, will be discussed in the next section. The warning can be realized say by appending a special code λ and indicating such links in the RREQ forwarded by B, as $(S, A, [B\lambda C])$ (instead of just (S, A, B)).

In order to ensure that C cannot *delete* the warning we

[2]It may not be able to remove Y and leave X as X's authentication of a partial path (which will be checked by E) will include Y.

[3]Which is obviously undesirable especially in dense neighborhoods.

can still employ per-hop hashing where[4] $\beta_2 = h(\beta_1, [B\lambda C])$. When C receives such an RREQ, it is expected *not* to forward the RREQ further. Even if a malicious C disregards this instruction from B, downstream nodes will honor this request from B. Any RREQ containing both B and C (even if they are separated by many hops) will not be allowed to propagate.

Thus if the RREQ contains a warning $[B\lambda C]$, and includes *both* B and C will not be propagated further. Note that it is *not* sufficient just to inhibit RREQs that contain the *link* $B \rightarrow C$ as any RREQ of the form $(\cdots B, \cdots, C, \cdots)$ is susceptible to misrepresentations by C. Obviously, a node can easily indicate multiple links to avoid (say by appending $(S, A, [B\lambda C, X])$ to indicate both $B \rightarrow C$ and $B \rightarrow X$ could be one-way).

IV. Mitigating RREQ Failures

There are many reasons as to why a node B may suggest (by including a warning) that a link $B \rightarrow C$ or $B \leftarrow C$ as unreliable.

1 the link is actually one-way
2 the link is bidirectional, but
   a C is a malicious node which pretends that it is out of B's range,
   b B is a malicious node, disseminating misleading information,
   c both B and C or honest, but due to collisions in the channel they are not able to confirm the existence of bidirectional paths.

Obviously, if the link is truly one-way (B cannot hear C), B will never get to know the very existence of C - unless a neighbor common to both B and C can indicate this possibility to B [10] (however C can still realize that this is indeed the case, even without the help of a common neighbor and drop RREQs from B). Note that for three of the four reasons, viz., 1, 2a and 2b, mitigating damages calls for prevention of *propagation* of RREQs that include *both* B and C in the path.

A. Recognizing Unreliable Links

Several approaches for detecting and avoiding unidirectional links have already been proposed [7], [12], [13] in the literature. However, such techniques are based on the assumption that the nodes co-operate in a fair manner to detect possible one-way links. For instance in [7] nodes proactively advertise transmission power and receiver sensitivity to enable nodes receiving RREQs to determine if links could be one-way. Obviously, malicious nodes that seek to exploit one-way links can easily advertise wrong values or simply remain silent.

In [10] nodes maintain a neighbor table - for example when node B hears a transmission from a node Y, node Y is added to B's neighbor table. Thus in a scenario where $C \rightarrow B$ is one-way, C's neighbor table would include B while B's table would not include C. Nodes periodically advertise their neighbor tables. Thus a common neighbor of B and C can realize this discrepancy and warn B of the existence of C. Obviously, in the scenario where C does not want to reveal the fact that the link is one-way, C will not indicate B in its neighbor table.

1) Semi-active Approaches: Even without active transmissions exclusively for this purpose, it is possible for any node, say F (in Figure 1), to learn about its neighbors just by *listening* to transmissions from neighboring nodes. Thus node F (in due course), will learn that C, L and G are its neighbors. Furthermore, when the local traffic is low (thus preventing nodes from gathering information about neighbors), it is not a severe disadvantage to send probing messages to solicit responses from neighbors. In other words, instead of *mandatory* periodic messages to determine the link state information, nodes can rely on eavesdropping when traffic is high, and use active probes when traffic is low.

Thus, as in [10] each node can maintain a neighbor table. Whenever a node overhears a transmission from a new node, a row is created for the new node with zero scores and a timestamp. A neighbor C of F will get a positive score if 1) node F sends a broadcast message (for example, a RREQ) and is able to overhear C faithfully re-broadcasting the RREQ after inserting itself in the path; 2) F successfully unicasts a packet to C (with RTS / CTS, thus confirming that the link is bidirectional). On the other hand, the neighbor C could receive a negative score when 1) F is not able perform an RTS / CTS handshake with C, or 2) F does not hear C broadcasting an RREQ (with the same sequence number), or 3) F hears C broadcasting an RREQ with longer[5] path length.

In addition nodes can still use proactive advertisements of transmission power / receiver sensitivity as in [7], as this could still help in scenarios where nodes are not malicious (and this does not call for extra transmissions - just a small increase in bandwidth of each transmission). Additionally, the nodes can also take the received signal strength, and rate of fluctuation of signal strength into account for determining the weights of positive and negative scores.

B. A Pessimistic Approach

Whatever technique is used to classify the path to neighboring nodes as one-way or bidirectional, they will be susceptible to judgment errors, where some bidirectional links to be misclassified as one-way, and vice-versa. The links, as seen by any node, can be classified into three broad categories

1) confirmed bi-directional
2) more likely to be bi-directional, and
3) more likely to be one-way.

Note that while we can confirm bi-directional links by exchanging RTS / CTS packets, we can only hypothesize that some links may be one-way (depending on the positive and negative scores). However, it may be reasonable to assume that "most" links seen by any node will fall under the first category, especially in scenarios where the nodes taking part in the wireless network are *designed* to be homogenous.

[4]Note that in order to remove the warning C needs access to β_1 transmitted by A, and if C can indeed hear β_1 A would have appended a warning $[A\lambda C]$, which cannot be removed without the knowledge of β_0.

[5]For example, if F broadcasts RREQ with t nodes and hears C broadcast the RREQ with more than $t+1$ nodes in the path.

It is only the second and third categories that have to be analyzed further to improve the chance of making a right decision (classifying them as one-way or bidirectional). The specific strategies used for maximizing the *a posteriori* probabilities (or a MAP [22] estimate of whether any link is one-way or bidirectional, based on positive and negative scores) will depend on numerous factors including local network density, traffic, and node mobility. Unfortunately, algorithms that take into account a large number factors (which themselves may not be easily amenable to reasonably accurate estimation) to provide accurate MAP estimates can be complex.

However, we show that even with very simplistic strategies, a significantly higher success rate of RREQs (compared to the case where no proactive measures are taken) can be realized. More specifically, we consider a pessimistic approach in characterizing links. Links are deemed one-way unless proactively established as two-way. In other words, we group together the categories "more likely to be one-way" and "more likely to be bidirectional" into a single group, and declare them to be one way (or unreliable). However, we do *not* drop RREQ packets with warnings. Such RREQs just suffer additional *delay* before retransmission.

If node E receives an RREQ from some path containing both B and C, and if B has indicated a potential one-way path (by appending $[B\lambda C]$ in the path), the RREQ is delayed for some appropriate duration before further propagation. Meanwhile (during this delay period) if E receives RREQ for the same sequence number with a safe path (a path without warnings that affect the path), the earlier RREQ is ignored. Thus paths that could include one-way links are given lower preference over good paths during propagation of RREQs. However, as many links that carry the warning may still be bidirectional, they can still be used - or some RREPs that include links with relevant warnings may still succeed. Furthermore, paths that include multiple links with warnings are delayed more than paths that include single warnings (which are more likely to succeed).

C. Simulation Results

To obtain a quantitative estimate of the effect of such one-way links on the failure of RREQs and the effectiveness of different strategies for reacting to such warnings (ignore, drop, or delay), we carried out extensive simulations in a square region with unit edges, consisting of 200 randomly placed nodes. Though there are many reasons like small differences in transmission power, receiver sensitivity, aging and local noise level which could contribute to one-wayness of links, in our simulations we assumed that the range of each node was different. More specifically, it was assumed that the mean range is 0.1 units, but the actual range of any node is a value uniformly distributed between 0.09 and 0.11 units (10% swing from the mean).

In most of our realizations, each node had 5 neighbors on an average. Furthermore, the number of one-way links N_o were substantially smaller than the number of bidirectional links N_b.

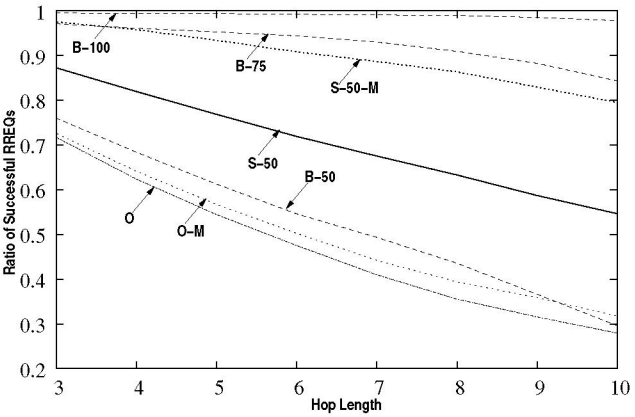

Fig. 2. Ratio of successful RREQs with different approaches.

More specifically, $N_b \approx 700$ and $N_o \approx 70$, on an average[6]. We simulated propagation of many route requests over different number of hops, choosing random source and destination nodes, for 10 different random realizations of the network. For each random realization we considered all possible node pairs[7] separated by a certain number of hops (ranging from 3 to 10), which have a link from the source to the destination, but not necessarily a reverse path. The RREQ propagation was modeled for 5 different cases

1) **B-100:** All bidirectional links are identified. Only bidirectional links are used for sending RREQ (the best case scenario).
2) **B-75:** Only 75% of the bidirectional link are identified and used for RREQ propagation. The remaining 25% are deemed one-way (though only a small fraction of them are actually one-way) and not used.
3) **B-50:** Only 50% of the bidirectional link are identified and used for RREQ propagation. The remaining 50% are deemed one-way and not used.
4) **O:** No proactive measure to identify one-way links. All links are used for RREQ propagation, and
5) **S-50:** Only 50% of the bidirectional link are identified and the remaining 50% are deemed one-way. However RREQs that contain paths deemed one-way (while many of which are actually bidirectional) are imposed additional delays.

In other words, for case O warnings are not employed, or alternately, warnings (if employed) are *ignored*. In cases B-100, B-75 and B-50 RREQs with relevant warnings are *dropped*. For case S-50 RREQs with relevant warnings are *delayed*.

The results are presented in Figure 2 terms of the ratio of successful RREQs to the total number of node pairs chosen. A total of over 250,000 RREQs (corresponding to randomly chosen node pairs) were simulated, ranging from

[6]The actual observed ranges were between 592-790 for N_b and 59-89 for N_o for over 1000 different realizations.

[7]Any (source, destination) pair for which the RREQ from source has a path to the destination.

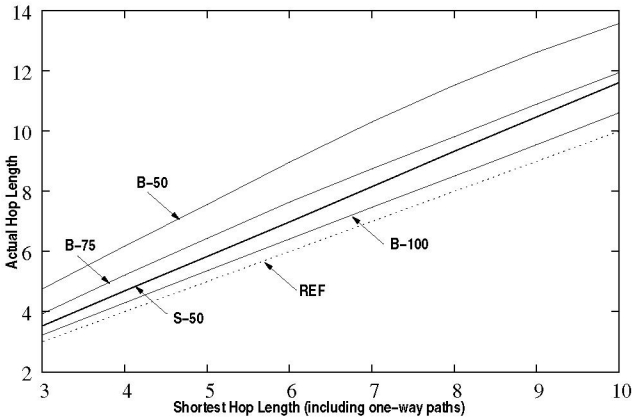

Fig. 3. Actual path lengths of successful RREQs.

25,000 to 40,000 paths for each hop length[8] In our simulations, collisions and node movement are ignored. We assume that the transmission delay is negligible. Thus all nodes at a distance of t hops from the source will relay the RREQ at time t units. However, with n nodes ready to send RREQ at time t we *process* the n RREQs in a random order, sequentially, among the n nodes.

1) Dropping RREQs with Warnings: Ideally, we would like to identify *all* bidirectional links (and thus ensure that all links identified as one-way are indeed one-way). In practice, the percentage of bidirectional links that can be identified will depend on numerous factors that perhaps cannot be reasonably modeled by simulations. It is for this reason we have opted to take the approach of illustrating three different scenarios where implies 100% (B-100), 75% (B-75) and 50% (B-50) of the actual bidirectional links are recognized as bidirectional. In a scenario where only $x\%$ of the bidirectional links are identified, the remaining $(100 - x)\%$ are classified as one-way (even though only a small fraction of them are actually one-way).

Obviously, decreasing x leads to lower network connectivity which will result in larger number of failing RREQs[9]. Clearly (for the particular choice of parameters), if we can identify 75% of the bidirectional links (and prohibit use of other links), we can still do reasonably well (not much lower than the best case scenario B-100). However, when x reduces to 50% many RREQs fail, as indicated by plot for B-50.

2) No Proactive Measures: Plot O in Figure 2 corresponds to the case where no proactive measure is taken to prevent use of one-way links (or scenarios where the possibility of one-way links, which exist, are ignored). In this case only a small fraction of RREQs (ranging from 71% for 3 hop paths to 32% for 10 hop paths) were successful, even while in almost every situation a good path *did indeed exist* (as is obvious from plot B-100).

We also considered an extension of scenario O, labeled O-M, where the destination responds (or invokes RREPs) to *all* RREQs that it receives. In this case, the RREQ is deemed successful even if *one* of the RREPs reaches the destination (or at least one path did not include one-way links). As can be seen from plot O-M, even under such a scenario the number of successful RREQs improve only marginally - supporting our intuition that good paths may be preempted by paths with one-way links. For the particular choice of parameters, even identifying and using only 50% of bidirectional links (B-50) performed better than not taking any proactive measure, thus clearly indicating the need for such measures.

3) Delaying RREQs with Warnings: Intuitively, if we can mitigate preemption of good paths by paths that are suspect, we can expect to do significantly better. This was the motivation for delaying RREQs with warnings, in scenario S-50. The plot labeled S-50 corresponds to the case where only 50% of the bidirectional links are identified (as in B-50). RREQs with warnings are delayed by a multiple of the hop period (to mitigate preemption). More specifically, an RREQ with w relevant[10] warnings is delayed by w hop periods, at each hop. As can be seen from plot S-50, a substantially larger fraction of RREQs succeed with this approach.

The plot labeled S-500-M (analogous to plot O-M) also takes into account success by responding to multiple RREQs. Note that while this approach did not yield substantial improvement from scenario O to O-M (due to preemption of good paths), as expected, the improvement in this case (S-50-M vs S-50) is indeed significant.

4) Path Lengths: Figure 3 indicates the actual path length of successful RREQs L_A for nodes which are L_0 (3 to 10) hops away, taking also one way links into account. The dotted line (labeled REF) where $L_0 = L_A$ serves as a reference. The best that we can do is B-100 where we identify all (100%) bidirectional paths. If we identify only 75% (B-75) obviously some RREQs will end up using longer paths. The situation gets worse for B-50. However the average path length for the strategy S-50 is even less than B-75. Thus apart from improving the success rate of RREQs, even the average length of successful RREQ paths are lower.

Obviously, proactive measures for indicating one-way links will increase the size of RREQ packets. If each node appends (on an average) n_w such warnings, the bandwidth of an RREQ packet that has traversed n_h hops will be increased by a factor $n_h n_w b$ bits where b is the number of bits in the ID of each node. In our simulations, for S-50 (and B-50) each node appends (on an average) about 3 additional IDs (or the effect is the same as increasing the size of the ID by a factor 4). This bandwidth overhead may be small compared to size of signatures appended by nodes. The advantages on the other hand viz., significant increase in the success rate of RREQs (and thereby eliminating the need for repeated RREQs) are

[8]The hop-length was defined on the basis on number of hops including one-way links. Thus in situations where one-way links are not used the actual path length will be longer (as is indicated in Figure 3).

[9]Note that we consider *all* node pairs that are connected - even with one-way links. Obviously, even if all bidirectional links are used, some pairs may not be connected anymore.

[10]The warnings are relevant only if the path includes the node identified by the warning(s).

compelling.

5) Rationale for the Simulations: The primary motivation behind our simulations is to demonstrate that unless proactive measures are employed to inhibit use of one-way paths, the result is significant reduction in the efficiency of DSR. We deliberately chose our parameters (number of nodes, range of each node and deviation from mean range) to result in a *small fraction* of one-way links, to highlight the point that even when the fraction of one-way links are *small*, the deterioration in performance can be *significant*. Note that while (on an average) 10% of the links are one-way, only half the links will affect RREQs as (on an average) half the links will inhibit *forward* propagation of RREQs[11]. Even while only 5% of the links contribute to failure of establishing paths, our simulations clearly demonstrate that the ill-effect of those links is still substantial enough, and cannot to be ignored.

V. Conclusions

The primary intentions of this paper are two fold: 1) to demonstrate that even when a very small fraction of links are unidirectional (as may be the case when devices taking part in a wireless ad hoc network are "homogeneous," by design) they can have a large detrimental effect on the efficiency of DSR, unless proactive approaches are taken to identify and mitigate the use paths that include one-way links, and 2) to show that ignoring one-way links can also affect the security of the route discovery process. It is essential, *especially* for secure DSR protocols, to take the possibility of one-way links into account, as they *rely* on bidirectional paths for providing assurances of integrity of established paths.

While proactive techniques for recognizing and mitigating the effect of one-way links in ad hoc routing protocols have received considerable attention in the literature, all such techniques depend on the assumption that nodes co-operate in a fair manner - or nodes do not send misleading information on purpose. We investigated techniques for determining "links to avoid" and cryptographically binding such blacklists in the RREQ messages. Fortunately while the reason for such blacklists may be because the link is truly one-way, or one of the nodes is malicious, the action to be taken is the same.

We further argued that while reliable algorithms for deciding if a warning is actually warranted could be complex (as they could depend on many factors, which themselves cannot be estimated), and hence impractical, even with very simple pessimistic approaches, we can realize significant gains in the success rate of RREQs. While all existing secure DSR protocols ignore the possibility of one-way links, the pessimistic approach suggested in this paper - declaring links to be one-way unless established to be bidirectional beyond reasonable doubt - can be very easily employed by such protocols to simultaneously improve *both security and efficiency*.

[11]While RREQs may loop around and propagate through such links in the reverse direction, such RREQs will not cause much damage as they are far less likely to *preempt* good paths.

References

[1] Web Link, http://www.ietf.org/html.charters/manet-charter.html

[2] P. Johanson, D. Maltz, "Dynamic source routing in ad hoc wireless networks," Mobile Computing, Kluwer Publishing Company, 1996, ch. 5, pp. 153-181.

[3] E. M. Royer, S-Ju Lee, C. E. Perkins, "The Effects of MAC Protocols on Ad hoc Communication Protocols," Proceedings of IEEE WCNC 2000, Chicago, IL, September 2000.

[4] R. R. Choudhury, N. H. Vaidya, "MAC-layer anycasting in ad hoc networks," ACM SIGCOMM Computer Communication Review, Volume 34 , Issue 1, pp 75 – 80, Jan 2004.

[5] S. Radosavac, N. Benammar J. S. Baras, "Cross-layer attacks in wireless ad hoc networks," Conference on Information Sciences and Systems (CISS-04), Princeton University, NJ, 2004.

[6] I.D. Aron, S.K.S. Gupta, "A Witness Aided Routing Protocol for Ad-Hoc Networks with Unidirectional Links," First International Conference on Mobile Data Access, LNCS **1748**, pp 24–33, 1999.

[7] Y-B Ko, S-J Lee, Jun-Beom Lee, "Ad Hoc Routing with Early Unidirectionality Detection and Avoidance," Personal Wireless Communications, Springer, 2004.

[8] M. K. Marina, S. R. Das "Routing performance in the Presence of Unidirectional Links in Multihop Wireless Networks," Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Jun. 2002.

[9] R. Prakash, "A routing algorithm for wireless ad hoc networks with unidirectional links," ACM/Kluwer Wireless Networks, Vol.7, No.6, pp. 617-625.

[10] S. Nesargi, R. Prakash, "A Tunneling Approach to Routing with Unidirectional Links in Mobile Ad-Hoc Networks," Proceedings of the Ninth International Conference on Computer Communications and Networks, 2000.

[11] R. Prakash, "Unidirectional Links Prove Costly in Wireless Ad Hoc Networks," Proc. of DIMACS Workshop on Mobile Networks and Computers, pp 15–22, 1999.

[12] V. Ramasubramanian, R. Chandra. D. Mosse, "Providing Bidirectional Ab- stracion for Unidirectional Ad Hoc Networks," Proc. of the 21st IEEE INFOCOM, June 2002.

[13] L. Bao, J.J. Garcia-Luna-Aceves, "Link state routing in networks with unidirectional links," Proc of IEEE ICCCN, Oct. 1999 Jun. 2000.

[14] Y-C Hu ,A Perrig,. D B.Johnson, "Ariadne:A Secure On-Demand Routing Protocol for Ad Hoc Networks," The 8th ACM International Conference on Mobile Computing and Networking, Atlanta, Georgia, September 2002.

[15] P Papadimitratos, Z. J.Haas, "Secure Routing for Mobile Ad Hoc Networks," Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference(CNDS 2002), San Antonio, Texas,2002.

[16] J.Marshall, V.Thakur, A.Yasinsac,"Identifying flaws in the secure routing protocol," Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference, 2003.

[17] S. Bucheggar. J-Y Le Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Spain,2002.

[18] S Marti, T. J. Giuli, Kevin Lai and Mary Baker,"Mitigating routing misbehavior in mobile ad hoc networks," Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston,2000.

[19] A. Perrig, R. Canetti, D. Song, D. Tygar, "Efficient and Secure Source Authentication for Multicast," in Network and Distributed System Security Symposium, NDSS '01, Feb. 2001.

[20] J. Kim, G. Tsudik, "SRDP: Securing Route Discovery in DSR," IEEE Mobiquitous'05, July 2005.

[21] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," INFOCOMM'99, 1999.

[22] A. Papoulis, S. U. Pillai, *Probability, Random Variables and Stochastic Processes,* 4th Edition, 2001, McGraw Hill.