Mobile Netw Appl (2012) 17:415–430
DOI 10.1007/s11036-011-0334-2

Secure Neighborhood Creation in Wireless Ad Hoc Networks using Hop Count Discrepancies

Thaier Hayajneh · Prashant Krishnamurthy · David Tipper · Anh Le

Published online: 28 July 2011

Abstract A fundamental requirement for nodes in ad hoc and sensor networks is the ability to correctly determine their neighborhood. Many applications, protocols, and network wide functions rely on correct neighborhood discovery. Malicious nodes that taint neighborhood information using wormholes can significantly disrupt the operation of ad hoc networks. Protocols that depend only on cryptographic techniques (e.g, authentication and encryption) may not be able to detect or prevent such attacks. In this paper we propose SECUND, a protocol for creating a SECUre NeighborhooD, that makes use of discrepancies in routing hop count information to *detect* "true" neighbors and *remove* those links to nodes that appear to be neighbors, but are not really neighbors. SECUND is simple, localized and needs no special hardware, localization, or synchronization. We evaluate SECUND using simulations and demonstrate its effectiveness in the presence of multiple and multi-ended wormholes. Lastly, we present approaches to improve the efficiency of the SECUND process.

Keywords secure neighborhood · wormhole attacks · route hop counts · ad hoc networks

1 Introduction

Neighborhood discovery in ad hoc and sensor networks is the process by which network nodes determine their neighbor nodes. Neighborhood discovery is a basic function that is a prerequisite to enabling multi-hop communication, access control, transmission scheduling, and other protocols for network operation. A short survey of neighborhood discovery in ad hoc networks is available in [1]. A conclusion in this paper is that securing the neighborhood discovery process is a difficult and open problem. In ad hoc networks the nodes typically try to discover their neighbors simply by broadcasting a neighbor discovery request. Each node that hears the request responds with a neighbor discovery reply. An adversary may try to thwart neighborhood discovery to disrupt the network operation by (a) preventing neighbors from discovering each other by jamming or (b) creating a false "neighbor relationship" between nodes that are not really in range of each other. The latter can be accomplished by spoofing neighbor discovery messages or by installing wormholes [2] in the network. This latter problem is the focus of this paper.

A wormhole (see Fig. 1) can be constructed by an adversary by simply copying all packets (signals) from one location (M_1) in the network and replaying them at another location (M_2) that is located several hops

This work was funded in part by the Army Research Office MURI grant W911NF-07-1-0318.

T. Hayajneh
Department of Computer Engineering,
Hashemite University, Zarqa 13115, Jordan
e-mail: Thaier@hu.edu.jo

P. Krishnamurthy (✉) · D. Tipper · A. Le
School of Information Sciences, University of Pittsburgh,
Pittsburgh, PA 15260, USA
e-mail: prashk@pitt.edu

D. Tipper
e-mail: tipper@tele.pitt.edu

A. Le
e-mail: atl13@pitt.edu

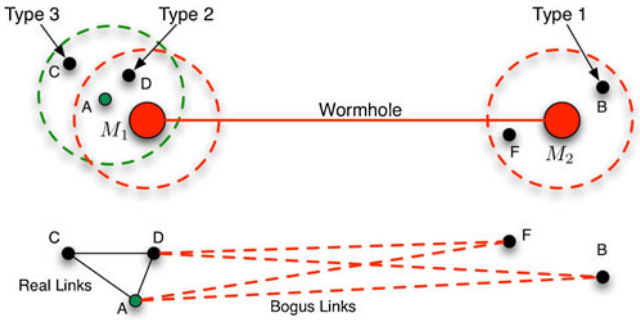

Fig. 1 Wormhole and types of neighbors

away. All the reply packets (signals) from location (M_2) will also be captured and replayed at location (M_1). Consequently, nodes that are located in M_1's range (e.g., A and D) will believe that they are neighbors to nodes that are located in M_2's range (e.g., B and F). In effect, the wormhole has created several bogus "direct" links between nodes in the network. In Fig. 1, the links $A-B$, $A-F$, $D-B$, and $D-F$ are bogus links. Multiple wormholes and multi-ended wormholes can worsen the situation. Standard security techniques such as encryption and authentication cannot detect the wormhole attack or identify the bogus links as the transceivers M_1 and M_2 can simply relay the encrypted or authenticated packets without changing them.

In this paper we consider the neighborhood discovery problem in ad hoc networks subject to wormhole attacks. Jamming attacks are not considered in this work. Most of the previous work (described in Section 2) on wormhole detection requires precise and accurate information about the location of nodes, the time of packet transmission and synchronization between nodes, or the use of special hardware (e.g., directional antennas). Further, the proposed protocols/algorithms must be applied between *all pairs of nodes* to detect the existence of a wormhole. In large networks with high node degrees, this can result in significant overhead and delay. Finally, detection of the existence of a wormhole is not sufficient. It is necessary to correctly identify the bogus links and distinguish them from real links between neighbors (i.e., effectively and efficiently remove the wormhole without adversely impacting the real links).

In this paper we propose SECUND,[1] a protocol for creating a SECUre NeighborhooD that can discover true neighbors, distinguish between true and purported neighbors, and detect and remove wormhole links if they exist in the network. Compared to other secure neighbor verification or discovery protocols, SECUND is simple, localized, needs no special hardware, localization, or synchronization. SECUND is based on principles developed in [3] where we presented a protocol called DeWorm to detect the existence of wormholes. DeWorm is an on-demand protocol that makes use of routing hop count discrepancies, determined by nodes along a route in a sliding fashion, to detect wormholes that may be somewhere along a route. SECUND also makes use of routing hop count discrepancies, but its goal is to efficiently check links between every pair of nodes for existence of wormholes and to remove only the tainted links to the extent possible. SECUND can also detect and remove two-ended and multi-ended wormholes (not considered in [3]).

Using hop count discrepancies to detect the existence of a wormhole exploits a basic characteristic of wormholes and consequently it needs no special hardware. The basic characteristic that is exploited is the fact that a wormhole must span multiple hops in order to have a significant impact on the network. Thus, routes from a source to a destination that avoid bogus links should be significantly longer than the routes that make use of bogus links. However, when this is independently employed by all nodes, there can be several false positives (i.e., a wormhole is flagged as existing when it actually does not). With SECUND, the number of false positives can be significantly reduced by *mutual* checks for existence of wormholes between pairs of nodes. The number of such checks can be drastically reduced (without significantly impacting the detection rate of wormholes or increasing the false positives) when nodes follow specific rules that enable them to *omit* such checks. When a wormhole is detected in the vicinity of a node, removing only those bogus links that have been created by the wormhole is challenging. SECUND uses a novel classification of different types of neighbors (see Section 3) to remove bogus links and a sequence of mutual checks to reduce the number of legal links removed.

To the best of our knowledge, SECUND is the first protocol that employs cooperation between honest nodes to reduce the overhead associated with detecting the existence of wormholes and creating a secure neighborhood. The heuristics suggested here to reduce overhead could be used by protocols that use protected location information (which is the straightforward way of detecting physical layer wormholes) to secure neighborhoods. This is also the first protocol, to our knowledge, that *explicitly* addresses the removal of

[1] In botany, the word "secund" refers to having elements on one side only—for example, leaves on one side of a branch—and not on both sides. The SECUND protocol ensures that the final list of neighbors of a node are those on the same side as the node, not the other side of a wormhole if it exists.

bogus links without removing legal links. Our simulations show that the proposed protocol can successfully detect and remove most wormhole links. Very few legal links are mistakenly removed. The cost associated with SECUND is the overhead measured by the number of route acquisitions (which we show is fairly low).

The rest of the paper is organized as follows. In Section 3, we describe SECUND in detail including the mutual cooperation between nodes to reduce the overhead of securing the neighborhood. Section 4 presents the results of a simulation based performance evaluation. Section 5 concludes the paper.

2 Related work

A survey of neighborhood discovery and neighborhood discovery protocols are presented in [1]. According to [1], researchers consider any wormhole defense mechanism as relevant to secure neighborhood discovery. Thus, here, we classify protocols for wormhole detection based on the approach they rely upon (even if they do not explicitly consider secure neighborhood discovery).

Location based approaches have the best ability to secure the neighborhood if the locations of nodes are securely exchanged and the general transmission range is known. In location-based approaches, a location-aware sender and receiver will securely exchange their location information. Then, in order to detect whether a wormhole connects them, the nodes will check if packets have traveled the distance between them using only a few hops and/or in a short time. In [4] end-to-end wormhole detection is proposed. Based on geographic information exchanged, the source node estimates the minimum hop count to the destination. The source compares the hop count value received from the reply packet with this estimated value. If the received value is less than that estimated, the corresponding route is marked as if a wormhole exists. Hu et al. [2, 5], suggested the use of geographical leashes to detect wormholes. A geographical leash requires each node to know its own location and all nodes to have loosely time synchronized clocks. The nodes need to securely exchange location information. A sender node can then ensure that the receiver is within a certain distance and detect discrepancies therein. Location based protocols usually require the nodes to be equipped with GPS or employ some other positioning technology. The problems with this approach are the need for having the hardware and/or infrastructure in place to accurately determine the positions of nodes and the fact that many positioning schemes may still not provide the required location accuracy in all environments (e.g., indoor and urban areas).

Time-based protocols, in general, are based on accurate time measurements or require the nodes to have tightly synchronized clocks. They work best with in-band wormholes that encapsulate packets between the wormhole nodes, but take the same number of hops otherwise. The idea here is that an in-band wormhole must cause noticeable delay for the traffic that passes through it. For instance, in [6], timing associated with existing MAC layer acknowledgments are used to detect a wormhole. In [7], the authors proposed a transmission time based mechanism (TTM) to detect wormholes. The protocol requires the computation of the transmission time between every two successive nodes along the established path during route setup. Time based protocols require some approximations as the node that is in charge of detection has to account for the processing and propagation delay times. Moreover, in ad hoc networks, the MAC protocol may also cause some unpredictable delays. More importantly, these protocols are not capable of detecting out-of-band physical layer wormholes. In such wormholes, a packet suffers only the propagation delay which could be small for wormholes using high-speed links. In [8], researchers showed that it is impossible to secure the neighborhood with general time-based protocols if adversarial nodes are able to relay messages with a delay below a certain threshold. Note that this threshold is what is typically used by such protocols to detect wormholes. A similar conclusion was also reached by Chiang et al. [9].

Distance bounding approaches use estimates of the physical distance between purported neighbors to ensure that such a distance is not longer than the maximum allowable distance (e.g., by using the farthest distance reachable by a node operating at its maximum transmission power). Many techniques have been used to estimate the distance between the nodes. Some researchers relied on the signal round trip time multiplied by the signal propagation time (speed of light) [1]. A secure neighbor verification protocol for wireless sensor networks is proposed in [10]. This protocol is distributed and relied on the estimated distance between nodes. They require each node to be equipped with a microsecond precision clock and two network interfaces: a radio-frequency and an acoustic interface. Other approaches also use some special hardware such as directional antennas [11], special RF [12], or ultrasound [13] to estimate such distance bounds. These protocols cannot be easily applicable to any ad hoc network because they add expense, complexity, and need for special customization. Moreover, some

of these protocols have their own specific weakness (e.g., uncertainty in location and varying propagation conditions) and cannot always ensure the detection of wormholes. Also it is sometimes possible for the attacker to use adversarial nodes that are equipped with the same hardware used by the network nodes to deceive a detection protocol.

Protocols that do not rely on location, timing, or tight synchronization can be classified into centralized and distributed approaches. Centralized approaches rely on gathering information such as statistics of node degrees and visual analysis of the network connectivity graph and processing them at a central entity. In [14], the network is reconstructed using multi-dimensional scaling and a wormhole that exists is detected by visualizing the anomalies introduced by the attack. Poovendran and Lazos [15] presented a graph theoretic framework for modeling wormhole links and derive the necessary and sufficient conditions to detect and defend against wormhole attacks. Qian et al. [16] presented a scheme to detect wormhole attacks based on statistical analysis. Centralized topology information was used in [17] to detect wormholes. The protocol looks for forbidden substructures in the connectivity graph that should not be present in a legal connectivity graph. This also requires the network to be highly connected. Detection requires a specific number of independent neighbors for the nodes connected through a wormhole to exist. It is always preferable to have the process for detecting and removing wormholes decentralized or distributed in ad hoc networks—centralized approaches are thus not very attractive.

Decentralized or distributed approaches include protocols that are based on connectivity and neighborhood information. These are the closest in scope to SECUND—our proposed protocol. Here nodes will exchange information such as node degrees or the list of one-hop and/or two-hop neighbors. Based on the collected information, the existence or not of a wormhole is determined. In general, the information should always be locally collected and/or disseminated, that is between a node and its one or at most two-hop neighbors. The node degree is used to detect wormholes in [18]. The assumption here is that the wormhole will increase the number of one-hop neighbors of a node and if this number is greater than some threshold (e.g., the average node degree) then there must be a wormhole. If however the wormhole connects a single node with another node that is far away, the node degree only changes by one and the wormhole will not be detected. Another possibility could be to place the wormhole between nodes that have a node degree less than the average which will prevent the wormhole's detection. But the damage to the network is comparable to any other wormhole. The protocol suggests an approximate removal process for a set of suspicious links that may however completely isolate some nodes from the network. In [19], an approach similar to [18] was presented. The assumption made is that the wormhole will significantly increase the number of one-hop neighbors. Nodes are assumed to be uniformly and densely deployed with no links changed or added. Each node will count the number of nodes that are two-hops away and the idea is that this number grows under a wormhole attack. In [20], the network topology is assumed to be static, links are assumed to be bidirectional, the topology is dense with every two neighbors assumed to have a common neighbor. The idea employed here is that the wormhole must change the topology structure of the network. The authors computed a so-called edge-clustering coefficient related to this change. A wormhole node is detected by one of its neighbors if that neighbor cannot reach one of the wormhole neighbors without using that node. However, it is possible to come up with many scenarios with wormholes that will not satisfy any of the necessary conditions for this approach to detect the wormhole. For instance, if the wormhole connects a group of nodes (≥ 2) with another group of nodes, which is the most common form of wormhole, then the protocol will not detect the wormhole.

3 SECUND

In this section, we describe how SECUND works. We will first describe the *detection* of the existence of a wormhole without regard to the percentage of false positives or the amount of overhead. Then we will consider the use of *mutual* checks to reduce the percentage of false positives and the use of *rules* to allow certain nodes to omit checks for wormholes, thereby reducing the overhead. Finally, we discuss the problem of *identifying types of neighbors* and *removal* of bogus links.

3.1 Network model and notation

We start by describing the network and the attack model used. Consider an arbitrary ad hoc or sensor network consisting of n nodes represented by the ordered set Q. Let the set of one-hop neighbors of a node A be N_A, that is, $N_A = \{A_1, A_2, \ldots A_{k_A}\}$, where k_A is the number of neighbor replies received by node A. For the discussion in this section we will assume the existence of a single two-end wormhole. The wormhole equipment $M_1 \leftrightarrow M_2$ is defined as two extra nodes M_1 and M_2

that are not part of the network, (i.e., not elements of Q). Here we assume a closed wormhole where M_1 and M_2 are not visible to their neighbors (i.e., they do not advertise their node IDs or MAC addresses) and that the wormhole is an out-of-band physical layer wormhole that uses a high speed link to connect M_1 and M_2. Detecting such wormholes is considered to be extremely difficult [21].

The set of one-hop neighbors of M_1 and M_2 will be N_{M_1} and N_{M_2}, respectively. Note that by definition, every node in N_{M_1} is connected to all the nodes in N_{M_2} via the wormhole and vice versa. Thus N_A, the one-hop neighbor set of node A includes nodes both within transmission range and on the other side of the wormhole if A is in the transmission range of the wormhole. Let $\hat{N}_A$ be the set of "true" one hop neighbors of A. Then $N_A^* = N_A - \hat{N}_A$ will be the set of nodes that are fake neighbors of A due to the wormhole. With reference to Fig. 1, $N_A = \{B, C, D, F\}$ and $\hat{N}_A = \{C, D\}$ and $N_A^* = \{F, B\}$. Clearly, $N_A^* = N_{M_2}$. The fake neighbors of A that are in N_A^* are classified as ***Type 1*** neighbors. The set $\hat{N}_A$ comprises of nodes that may also belong to N_{M_1}—these are called ***Type 2*** neighbors (e.g., D). The subset of nodes in $\hat{N}_A$ like C, that are not in N_{M_1} are called ***Type 3*** neighbors. Let the route from any node X to any node Y be R_{X-Y} and $|R_{X-Y}|$ be the length of the route in number of hops.

3.2 Detection of wormhole existence

Node A will first determine if it is in the vicinity (within the transmission range) of a wormhole The process used here is similar to that described in [3]. The basic idea here is to find alternative paths to a target node that do not pass through the wormhole. These alternative paths will be significantly different in length compared to the path that goes through the wormhole making use of the wormhole link—otherwise the wormhole will not attract large amounts of traffic. If node A is in the vicinity of a wormhole, one or more nodes in N_A will be on the other side of the wormhole. Suppose that $B \in N_A^*$ is a fake neighbor of A and that the wormhole is η hops long. If a node $X \in \hat{N}_A$ were to find a route to some neighbor of B that is not a neighbor of A (called the *target* T) avoiding all nodes $\in N_A^*$, such a route must be at least η hops long (since a route that goes through the wormhole has to include some node in $N_A^* = N_{M_2}$, the wormhole is avoided and the alternate route must be at least as long as the wormhole itself). Implementing this idea is not trivial since node A does not know the composition of N_A^*. So node X avoids using all nodes in N_A which will include all nodes in N_A^*. But X itself may be part of N_A^*. This makes it necessary for all nodes in N_A to repeat this process. Further, it is also possible that $T \in N_{M_1}$ since it is a neighbor of B which is in the vicinity of the wormhole. Thus, it could be closer to A than B. All of these are taken into account in the algorithm to detect existence of a wormhole. The basic algorithm is given in Figs. 2 and 3. A description of all the steps with some discussion is presented next.

SECUND steps

1. Node A will discover its one-hop neighbors by broadcasting a "hello" message. Cryptographic techniques (e.g., authentication) are used to prevent malicious nodes from sending fake replies.
2. Node A receives replies from its neighbors and verifies their authenticity. Neighbors could be elements of either $\hat{N}_A$ or N_A^*. At this time, A cannot distinguish between these sets.
3. Node A wishes to determine if B is a true neighbor. A asks B to provide its one-hop neighbor list N_B. We refer to B as the neighbor under examination (see Fig. 2).
4. Node A picks some node $\in N_B - N_A$ and marks it as the target node T.
5. Node A will ask all its one-hop neighbors (real and purported) to find the shortest route to T. Those routes R_{X-T}:
   (i) cannot be direct (must pass through another node) and
   (ii) must avoid the one-hop neighbors N_A and N_B of both A and B.
6. Nodes in N_A reply to A with the length $|R_{S-T}|$ of their shortest routes to the target node T.
7. Node A employs *Select(route)* (see Fig. 3) to select a route R_{S-T} that will be compared with a route that may pass through the wormhole, if a wormhole exists in the vicinity. The route that passes through the wormhole should be 3 hops—from the neighbor of A to A, from A to B and from B to T). Using *Select(route)* eliminates extremely long route outliers while ensuring that a route that is η hops longer than the wormhole route is not missed.
8. If the difference between the length of the selected route and the wormhole route is greater than a threshold η then SECUND declares there is wormhole in the vicinity. We discuss picking the right value of η next.

Selection of η The value of η is not known a priori, but while implementing security in the network, the

Detection (A,B)

1. A **finds** N_A
2. B **sends** N_B to A
3. A **chooses** $T \in \{N_B - N_A\}$
4. $\forall X \in N_A$, X **finds** R_{X-T} : $N_B \cup N_A \notin R_{X-T}$ and $|R_{X-T}| > 1$
5. $\forall X \in N_A$, X **sends** $|R_{X-T}|$ to A
6. A **determines** $Select(R_{X-T})$ and **computes** $|R_{S-T}|$
7. **If** $|R_{S-T}| > \eta + 3$ **then** A assumes B is connected to it through a wormhole

Fig. 2 The initial detection algorithm

administrative entity can decide what it should be. Typically, the longer a wormhole, the greater the damage it inflicts by attracting more traffic. With $\eta = 2$ even short wormholes (2 hops) can be detected. However, the simulation results will show that the number of false positives will be high. Using $\eta \geq 3$ reduces false positives but short wormholes (less than 3 hops) may escape detection. $\eta = 2$ or 3 provides the best tradeoff between detection rate and false positives as we see from the results later. We now consider some scenarios illustrating how SECUND works.

No wormhole In the example in Fig. 4 nodes A and B are real neighbors—there is no wormhole in this case. Node A wants to check if node B is a real neighbor. Node A picks node F, a neighbor of B but not a neighbor of A, as the target node and asks its neighbors C, D, and X to find routes to node F. The lengths of these routes will be 4, 2, and 5 hops. Note that the nodes have to avoid the one hop neighbors of nodes A and B in their routes to F. Node A will select one of these routes (for now let us assume it is the longest one of 5 hops). The route from X to F through A will be 3 hops. If $5 - 3 < \eta$ then node A will decide that node B is a real neighbor. In some cases, $|R_{X-T}| - 3 \geq \eta$ if the topology is sparse and there is a false positive.

Select(R_{X-T})

1. $\forall X \in N_A - \{B\}$
   sort $|R_{X-T}|$ from longest to shortest
   ${R_{X-T}}^L$ is longest, ${R_{X-T}}^1$ is shortest
2. $R_{S-T} = {R_{X-T}}^L$
3. for$(i = 1, i < L, i++)$
   $\{$ **if**$(|{R_{x-T}}^L - {R_{x-T}}^{L-i}| \leq \eta)$
   then $R_{S-T} = {R_{X-T}}^{L-i}$
   else break $\}$

Fig. 3 The route selection algorithm

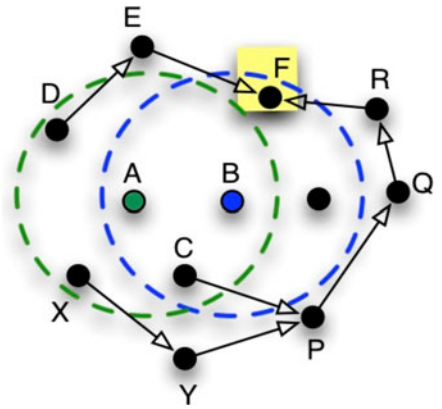

Fig. 4 Detection operation without wormhole

With wormhole Cases where nodes A and B are connected through a wormhole are shown in Figs. 5 and 6. In Fig. 5, $A \in N_{M_1}$ and it has at least one neighbor $B \in N_{M_2}$. The target node must be a neighbor of B but not of A. Thus, there are two possibilities for the target node in this case. An example of the first is node $F \in \hat{N_B} - N_{M_2}$ in Fig. 5. Neighbors of A avoid other nodes in N_A and all nodes in N_B when they try to reach F. Since all nodes in N_{M_2} are included in N_A and all nodes in N_{M_1} are included in N_B, all the wormhole links will be avoided. True neighbors of A will have routes to F that are longer than 3 hops by at least η and the wormhole will be detected. For instance, node D, which is a true neighbor of node A cannot use nodes B or X to reach the target node F and will use the long route shown in Fig. 5. In the second case, the target node is an element of N_{M_1} but outside the range of A (e.g., node E in Fig. 6). In this case, true neighbors of A will find short routes to E, but purported neighbors $\in N_{M_2}$ will have long routes to E (e.g., node X is Fig. 6). To conclude, in either case, some neighbor of A will report a route whose length exceeds $3 + \eta$ and the wormhole is detected. A detailed example and analysis of a grid network with a two-ended wormhole is provided in the Appendix.

Selection of route The algorithm used by node A to find R_{S-T} is $Select(R_{X-T})$ which is shown in Fig. 3.

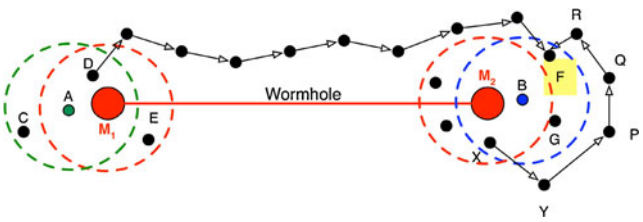

Fig. 5 Detection operation with wormhole—case 1

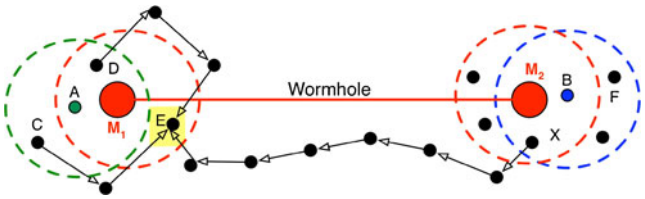

Fig. 6 Detection operation with wormhole—case 2

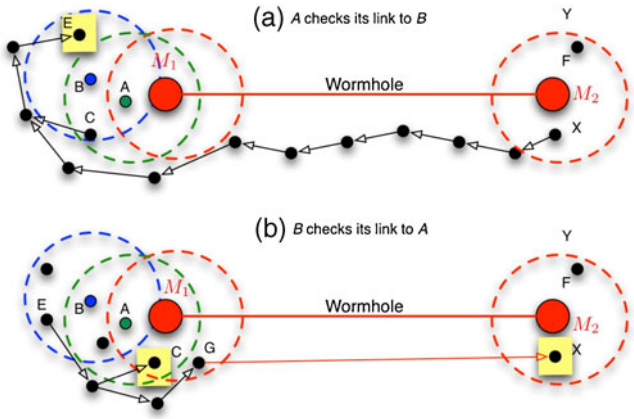

Fig. 7 False positives and mutual checks with type 3 neighbors

Node A creates a sorted list of route lengths from its neighbors to T (excluding replies from neighbors that do not have routes to the target node). Node A picks a route (if it exists) that is smaller than the longest route by *not more than* η. Otherwise the longest route is picked. It is the length of this picked route that is used to determine the existence of the wormhole. We have tested other methods for *Select*(R_{X-T}) (see [3] for a longer discussion). Using the longest route has a better detection rate especially for short wormholes but increases the percentage of false positives for randomly distributed networks. Using the average length of all routes reduced false positives but also reduced the detection percentage. The method in Fig. 3 provides the best performance.

3.3 Improving the detection process

We present approaches to reduce false positives and to reduce the overall number of checks that must be performed between supposed neighbors in the network.

Reducing false positives False positives can occur in two ways - first when there is no wormhole and the topology is sparse resulting a long route to the target and second when node A tries to check for the existence of a wormhole between itself and a Type 3 neighbor (see Fig. 1). In the latter case, nodes $\in N_A^*$ will find long routes to a Type 3 neighbor. The example below illustrates why this is the case.

Wormhole exists in the vicinity of A, but does not connect A and B As shown in Fig. 7a, when A checks its link to B for existence of a wormhole, a bogus neighbor of A such as $X \in N_A^*$ will find a long route to a target node (e.g., E). This will flag the link between A and B as being potentially corrupted by a wormhole resulting in a false positive. This problem exists when B is either a Type 2 neighbor or a Type 3 neighbor. As described below, this problem can be addressed through mutual checks if B is a Type 3 neighbor.

Mutual detection We have found that mutual detection (i.e., A checking if B is a true neighbor and B checking if A is a true neighbor) reduces the percentage of false positives significantly. A wormhole will be suspected to exist **if and only if** both A and B discover their links to be connected through a wormhole. When there is no wormhole, the target nodes for nodes A and B (see Fig. 4) when they perform mutual checks will be different. In most cases, even if A marks node B as connected through a wormhole, B will not or vice versa. Of course there will still be topologies where both A and B will still flag each other as connected through a wormhole, but this fraction is an order of magnitude smaller as shown in Section 4.

Mutual detection also helps when B is a Type 3 neighbor but A is in the vicinity of a wormhole. Even if node A in Fig. 7a flags its link to node B as bogus, node B will not flag its link to node A as shown in Fig. 7b. Note that B has no neighbors $\in N_{M_2}$ that would find a route to a target node such as C in Fig. 7b. Thus, if a neighbor of A such as $C \in \hat{N}_A$ is picked, routes to C will be short. If a target node is picked in N_A^* (for instance X), there will likely be nodes in N_{M_1} like G that are not neighbors of either A or B that will find short routes to X. Thus false positives are reduced and Type 3 neighbors are correctly identified most of the time. Exceptions will exist when the network is sparse. At this point, simply using the detection scheme will not be able to distinguish between Type 1 and Type 2 neighbors.

Reducing the number of checks If every node checks each of its neighbors to detect the presence of a wormhole, a large overhead and delay can result, especially in dense networks with high average node degree. The question then is whether every node must check links with all of its neighbors or some nodes can be exempt from applying the detection process. Without any formal proof, we argue that the following simple rules for checking the existence of wormholes can be adopted

eliminating a large number of unnecessary checks. The efficacy of these rules are verified by simulations in Section 4.

1. If node A checks its link with node B and no wormhole is discovered, node B need not check its link with node A.
2. If node A checks its link with node B and a wormhole is discovered, node B must check its link with node A. This ensures that false positives are reduced as discussed previously.
3. If node A has checked its link to node B AND vice versa, no wormhole is discovered, and any node C is a neighbor of *both* A and B, nodes A and B need not check their links with node C, and vice-versa.

The first two rules are based on the mutual detection process described above. The third rule is based on the following reasoning. The detection process simply determines the existence of a wormhole in the vicinity of a node A (see Fig. 7a and the related discussion). The fact that A was unable to find a wormhole when it checked its link with B AND vice versa implies that there is no wormhole in the vicinity of A or B. If C is a neighbor of both A and B, it cannot be on the other side of a wormhole since neither of the two nodes detected a wormhole.

One may pose the situation shown in Fig. 8 to be problematic, where node A has two neighbors X and $B \in N_A^*$. What if X checks its link with B and finds no wormhole? One of the characteristics of the detection process is that it cannot distinguish between Type 1 and Type 2 neighbors. Since both A and B are neighbors of X, when X checks its link with B, it will ask A to find a route to a target node (say G) and this will reveal the presence of the wormhole since A has to avoid all nodes in N_{M_1} and N_{M_2}.

More aggressive approaches may result in reducing the wormhole detection rates. For instance, we could exempt *all one hop neighbors* of nodes that did not detect a wormhole from checks. However, if the wormhole is only connecting exactly one node A with another node B located several hops away (see Fig. 9), both nodes A and B, may never check for the

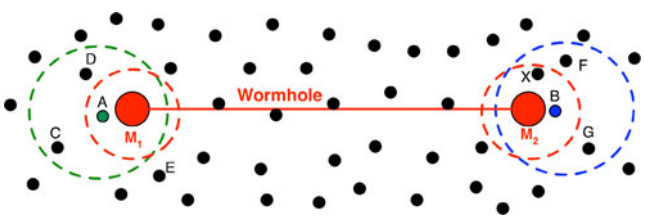

Fig. 8 Wormhole connecting single node with two nodes

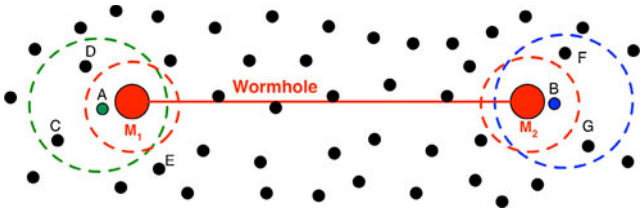

Fig. 9 Wormhole connecting single nodes

wormhole as one of their neighbors that is not within the wormhole range could have ruled out a wormhole. Similarly, it is not a good idea to exempt *all* other neighbors from checking links to a node if one of its neighbors did not detect a wormhole. For instance, in Fig. 9, if C did not detect a wormhole when it checks its link to A, which is true, then B may also not check its link to A even though there is a wormhole in between. By including the condition that B has to also be a neighbor of C, this possibility is averted.

3.4 Removal of bogus links

Detecting the existence of wormholes in the network is an important step. However, another crucial process is to remove the links created by the wormhole. Note that a wormhole that connects m_1 nodes $\in N_{M_1}$ with m_2 nodes $\in N_{M_2}$ results in $2m_1m_2$ bogus unidirectional links. Even if one of these links is not removed it may still cause damage by attracting traffic. Many of the available wormhole defense mechanisms ignore the removal of the wormhole connected links or use techniques that may remove many legal links. As previously described, the detection process *only* flags the existence of a wormhole. A link between both Type 1 neighbors and Type 2 neighbors will be flagged as corrupted by a wormhole as confirmed by mutual checks. However, mutual checks between Type 3 neighbors allows them to identify the fact that they are not connected through a wormhole. The challenge then is to distinguish between Type 1 and Type 2 neighbors to avoid removing legal links between Type 2 neighbors.

The algorithm *Removal()* used by nodes to decide the removal of the links is shown in Fig. 10. Only if a node A *detects* the existence of a wormhole when it checks the link between itself and node B, then all neighbors of A and B (i.e., all nodes in $N_A \cup N_B$) will use the algorithm in Fig. 10. The algorithm works as follows. Node A will ask all its neighbors that are not part of N_B to find routes to neighbors in N_B that are not part of N_A *one-by-one*. If at any point, routes are found to be very long (similar to *Detection()* in Fig. 2), the process stops. Then B is flagged as a Type 1 neighbor,

Removal(A,B)

1. $\forall X \in N_A - N_B$,
   For some $Y \in N_B - N_A$,
   X **finds** R_{X-Y} : no element in $N_A \cup N_B$ is part of R_{X-Y}
   then X **sends** $|R_{X-Y}|$ to A
2. A uses all R_{X-Y}s and employs $Select(R_{X-Y})$ to **find** $|R_{S-Y}|$
3. **if** $|R_{S-Y}| > \eta + 3$
   then A removes link to B and the process will stop
   else all X will repeat process for next $Y \in N_B - N_A$
4. **if** no $|R_{S-Y}| > \eta \; \forall Y \in N_B - N_A$,
   then link A to B will not be removed

Fig. 10 Wormhole removal algorithm

and the link is removed. If not, node B is flagged as a Type 2 neighbor.

Why does this work? Consider a Type 1 neighbor B of node A (see Fig. 11). Node A can have neighbors in $N_A - N_B$ that are on either side of the wormhole. This is also the case with node B. This is the reason why nodes in $N_A - N_B$ should find routes to nodes in $N_B - N_A$ one-by-one. Eventually, a long route is discovered. For example, node $H \in \hat{N}_A$ will find a long route to $K \in \hat{N}_B$ or node $X \in N_A^*$ will find a long route to $E \in N_B^*$. It is not sufficient to pick any one node in $N_B - N_A$ as it is possible that $N_A - N_B$ has nodes on only one side or the other of the wormhole. In the case of a Type 2 neighbor B (see Fig. 12), nodes in $N_B - N_A$ and $N_A - N_B$ are *both* constrained to be on the same side as nodes A and B. Any neighbor of A that belongs to N_A^* will also belong to N_B^* since they are both on the *same* side of the wormhole. Thus, routes from nodes in $N_A - N_B$ to nodes in $N_B - N_A$ will very likely be short (e.g., from node C to node D in Fig. 12).

A question that arises here is why do we not use this process for detection itself? The reason for not using this algorithm is that this results in a large number of false positives when there is no wormhole in the network. (That is, when only nodes in $N_A - N_B$ find routes to nodes in $N_B - N_A$ one-by-one, it is likely that some outlying long route exists flagging it as a wormhole). We observed that the false positives can be

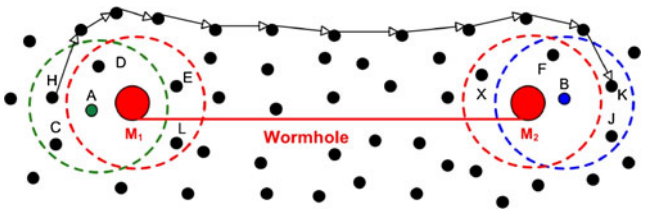

Fig. 11 Identification of type 1 neighbor

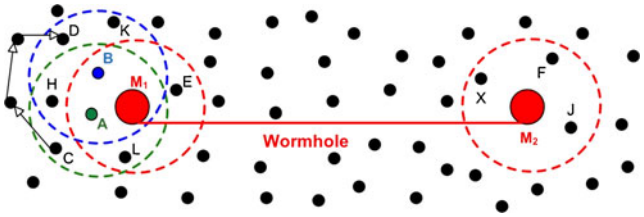

Fig. 12 Identification of type 2 neighbor

as large as 35% compared to 1% using the *Detection()* process mutually between nodes. Thus *Removal*() is employed only when a wormhole is detected and after mutual detection has been employed to eliminate Type 3 neighbors. We also observed from numerical results that using *Removal*() mutually between two nodes A and B can reduce the number of legal links removed.

3.5 Other issues

If a *critical node* (if this node is removed, the network will be partitioned) exists in the the network, then the *Detection*() process will not work. Figure 13a shows an example of a critical node in a network. Methods of detecting critical nodes and addressing this problem are discussed in [3] and [23]. Another situation that may result in a lower detection rate is when a node A has no neighbors in N_{M_2} and the target that is picked is a bogus neighbor of B in N_{M_1}. This situation has a probability that depends on the range of the wormhole and the fraction of neighbors of B that are not neighbors of A on either side of the wormhole. These two issues are related to sparse networks. We evaluate the performance of SECUND as a function of the average node degree in Section 4.

While the number of legal links removed will be small (as shown in Section 4), the impact of removing a very small number of legal links can be expected to be minimal. For example, in Fig. 13b, if the link from A to B is removed, node A may be able to use node C

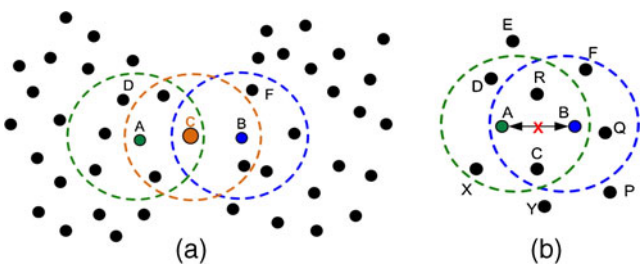

Fig. 13 **a** Critical node. **b** Removal of legal links

to reach Q without an increase in the number of hops. In some cases, a few additional hops may be required. Finally, while we have not explicitly described protocols that will use *Removal*() (and employ mutual checks and exchange of information), one can expect this to operate in a manner similar to the detection process.

4 Performance evaluation

4.1 Simulations

The important performance metrics for secure neighbor creation are: the percentage of correct wormhole detection, the percentage of false positives, the percentage of wormhole links removed, and the percentage of legal links removed by mistake. We have considered two different node distribution models: (1) grid distribution with perturbations and (2) random distribution. In the grid case, nodes are located in a perturbed 20×20 grid. The coordinates of each node x_i and y_j were randomly chosen using uniform random variables in the ranges $(100i - p100, 100i + p100)$ and $(100j - p100, 100j + p100)$, respectively, where p is the perturbation parameter and $i = 1, ...20$ and $j = 1, ...20$ (in our simulations, $p = 0.2$). For the random node distribution, the coordinates of the nodes (x_i, y_i) for $i = 1, 2, ...200$ were independently and randomly chosen in the range from 100 m to 2,000 m using a uniform [100–2,000] random number generator. As in [3], we also investigated SECUND with two link connectivity models: (1) the commonly employed unit disk graph (UDG) and (2) the quasi unit disk graph (quasi-UDG). The quasi-UDG connectivity model is described in [22]. Only results with the UDG are shown here for brevity. The results with the quasi-UDG connectivity model are very similar. To change the average node degree, the transmission range of the nodes was varied from 110 m to 160 m. The simulation model was programmed in C and the model uses the DSR routing protocol and node distribution models from ns-2. For statistical validation the simulations were repeated 50 to 100 times with confidence intervals of 95%. SECUND was evaluated for networks without any wormhole, with two-ended wormholes, and multi-ended wormholes.

Two-ended wormholes A two-ended wormhole connects groups of nodes from one location in the network to another group of nodes located several hops away (wormholes with different lengths are tested). As shown in Fig. 14, two separate wormholes are created in the network such that the ranges of the wormhole transceivers do not overlap. That is, each node in M_1's

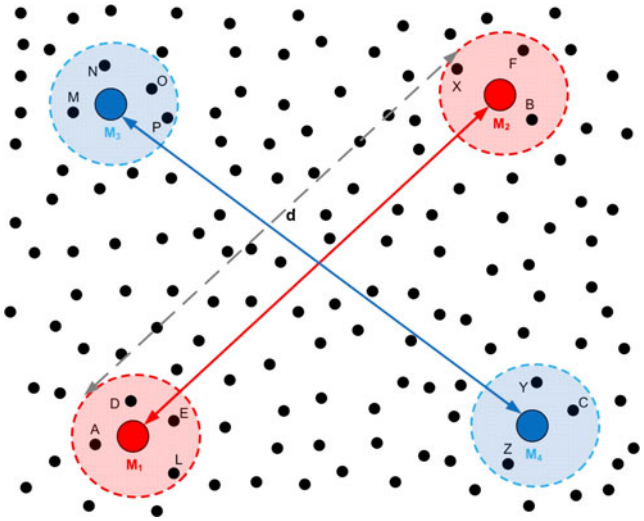

Fig. 14 Two-ended regular wormhole

(M_3's) range is only connected to every node in M_2's (M_4's) area and vice versa. Let m_i be the number of nodes in the range of wormhole transceiver M_i. The number of unidirectional links created by wormhole $M_1 \leftrightarrow M_2$ is $2m_1m_2$. Note that two nodes $A \in N_{M_1}$ and $B \in N_{M_2}$ have two links between between them $A \rightarrow B$ and $B \rightarrow A$. For the two wormholes shown in Fig. 14, the number of bogus links created is $(2 \times 4 \times 3) + (2 \times 4 \times 3) = 48$.

Multi-ended wormhole In this case the wormhole will be connecting nodes located in many different areas. In the example shown in Fig. 15, each node located near any wormhole transceiver will be connected to all nodes located at the other transceivers.

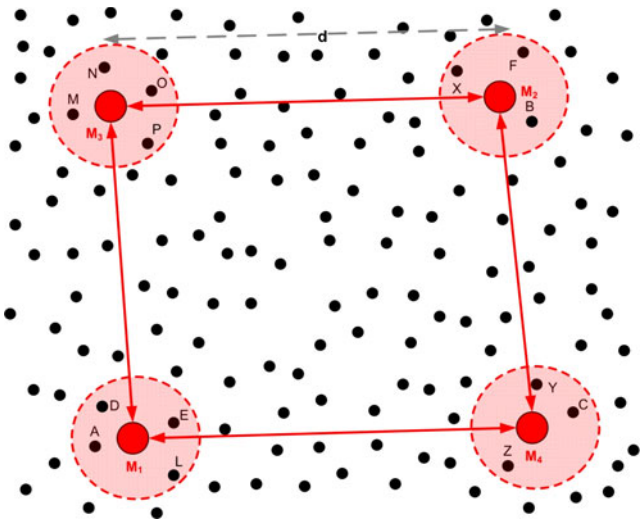

Fig. 15 Multi-end wormhole

Table 1 False positives without mutual checks

η	2	3	4	5
Grid	7.190	0.489	0.038	0.0172
Random	11.216	1.699	0.281	0.092

Table 3 Percentage of legal links removed by mistake

η	2	3	4	5
Grid	0.468	0.001	0	0
Random	1.033	0.061	0	0

For instance, every node in N_{M_1} will be connected to every node in N_{M_2}, N_{M_3}, and N_{M_4}. In general, for a n-ended wormhole the number of links created will be: $\sum_{i=1}^{n-1}\left[m_i \cdot \left(\sum_{j=1, j\neq i}^{n-1} m_{j+1}\right)\right]$. For the example in Fig. 15 the number of links created by the 4-ended wormhole is $4\ (3+4+3)+3(4+4+3)+4(4+3+3)+3(4+3+4)=146$.

4.2 Results without wormholes

We first simulated networks without wormholes and ran the *Detection*() and *Removal*() algorithms to determine the percentage of false positives and legal links removed by mistake. The false positive rate is determined by the fraction of instances where the *Detection*() process flags a wormhole as existing when it does not. We first look at the false positive rate when *without* mutual checks (i.e., A runs *Detection*() but B does not when A checks to see if B is a neighbor). Simulation results shown in Table 1 indicate that the false positive rate can be fairly high for small values of η. This is because it is possible for some nodes to only find routes to the target that are longer than the route through A and B by more than 2 hops. However, with $\eta = 3$ the percentage of false positives is less than 2%. When *Detection*() is run by both nodes (mutual checks), the false positive rates fall drastically as seen in Table 2. It is close to 0 for grid distributions of nodes and less than 0.2% for randomly distributed nodes for $\eta = 3$. More false positives occur with randomly distributed nodes since nodes may have relatively long routes to reach the target node.

The percentage of legal links removed by mistake (number of links removed/total number of links) when there is no wormhole, is a very important performance metric for secure neighbor discovery protocols. Table 3 shows the percentage of legal links removed by mistake for both grid and randomly distributed nodes for different values of η. The results show that SECUND removes none or very few legal links. Note that in this case there are no Type 1 or Type 2 neighbors which makes the process less complicated.

Table 2 Percentage of false positives with mutual checks

η	2	3	4	5
Grid	1.299	0.006	0	0
Random	2.884	0.196	0	0

4.3 Results with wormholes

In this section, we present simulation results when two-ended and multi-ended wormholes are present in networks. We first present results of the percentage of legal links removed—in comparison with results from the previous section. Next we present detection rates as a function of wormhole length and node degree.

4.3.1 Removal of bogus and legal links

We simulated wormholes with $d \geq 5$ hops and considered the fraction of legal links removed and the fraction of bogus links removed for different values of η. In the case of both grid and random distribution of nodes and for both two-ended and multi-ended wormholes, $\eta = 3$ provides almost 100% removal of bogus links and removal of less than 1% of legal links. Even for $\eta = 2$, the performance can be considered to be very good. As η increases to 5, the fraction of bogus links removed drops to around 80% with random node distributions and multi-ended wormholes where it becomes more difficult to distinguish between Type 1 and Type 2 neighbors for larger η values. These results are shown in Figs. 16 and 17. It is worth noting that traffic

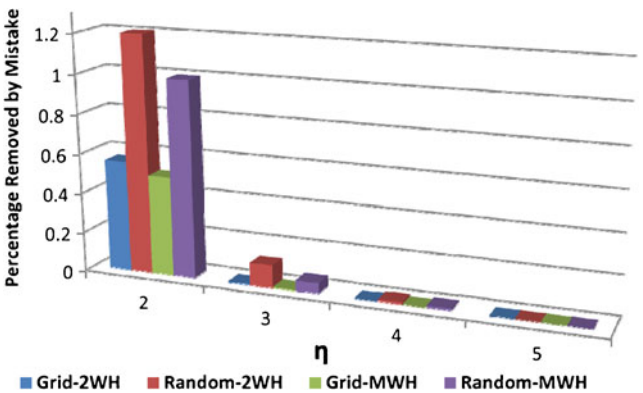

Fig. 16 Percentage of legal links removed with two and multi-ended wormholes

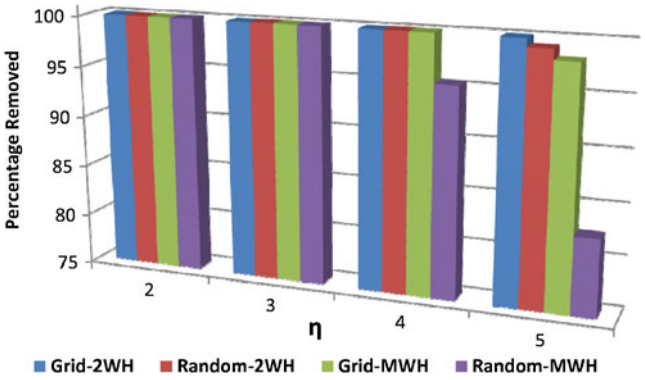

Fig. 17 Percentage of bogus links removed with two and multi-ended wormholes

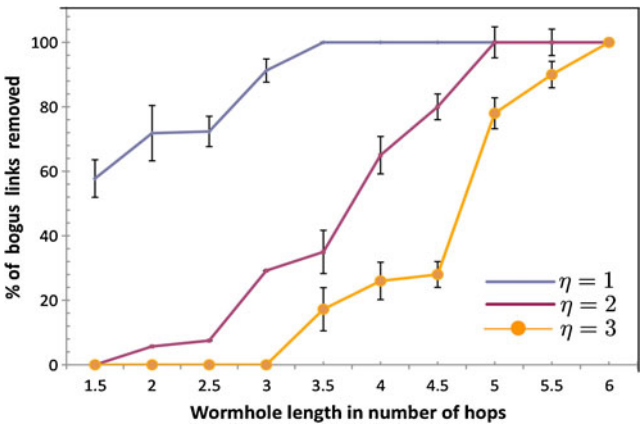

Fig. 19 Impact of wormhole length on wormhole removal

performance impact of the removal of legal links has been studied in more detail via simulation in [24].

4.3.2 Impact of wormhole length

Figure 18 shows the percentage of wormholes detected for different values of wormhole length starting from 1.5 hops till 6 hops, and for $\eta = 1, 2, 3$. The results confirm that η impacts the length of the wormhole that can be detected. With $\eta = 1$, any wormhole can be detected but the number of false positives will be extremely high. With $\eta = 3$, any wormhole longer than 4 hops will be certainly detected. Similarly, the impact of the length of the wormhole on the removal process is shown in Fig. 19. The results show that the removal process will be enhanced with longer wormholes. It it worth noting that in general, wormholes will only be a successful attack method if they capture a significant amount of network traffic which implies they should have a long hop count providing a significant shortcut through the network. In the literature [16] wormholes of lengths that are 6–10 hops are typically considered, thus SECUND should work well in practice with a low false positive rate.

4.3.3 Impact of node degree

With $\eta = 3$ fixed, we simulated networks with a variety of average node degrees. The average node degree was changed by changing the transmission range of the nodes from 110 to 160 m. Obviously, the larger the transmission range is, the more nodes there are that can be reached by a given node, and hence the higher the average node degree. A higher node degree provides more options for finding routes and improves the performance of SECUND in general. For example, the % of false positives performance of *Detection*() with and without mutual checks is shown in Fig. 20. With an average node degree of 5–6, the % of false positives is very small because it is unlikely that only outlying long

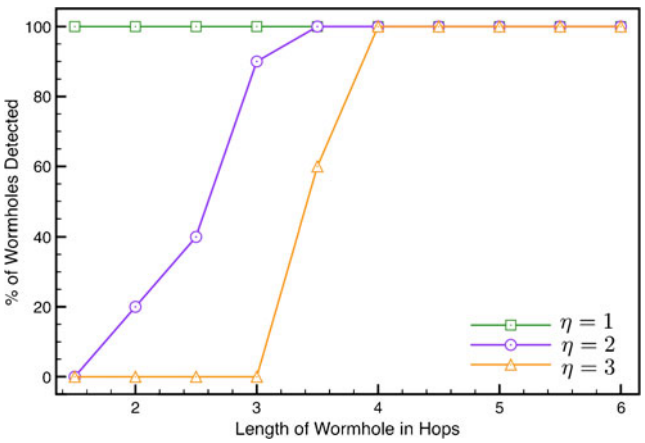

Fig. 18 Impact of wormhole length on wormhole detection

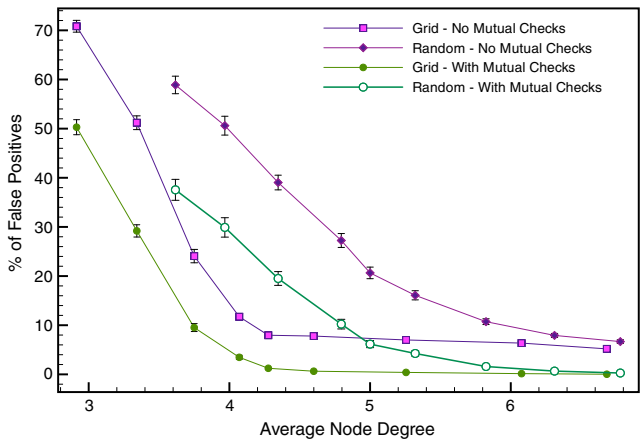

Fig. 20 False positives with detection with and without mutual checks

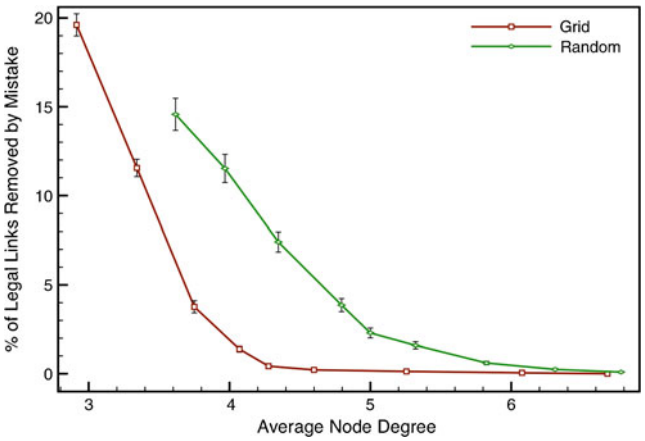

Fig. 21 Impact of node degree on links removed by mistake

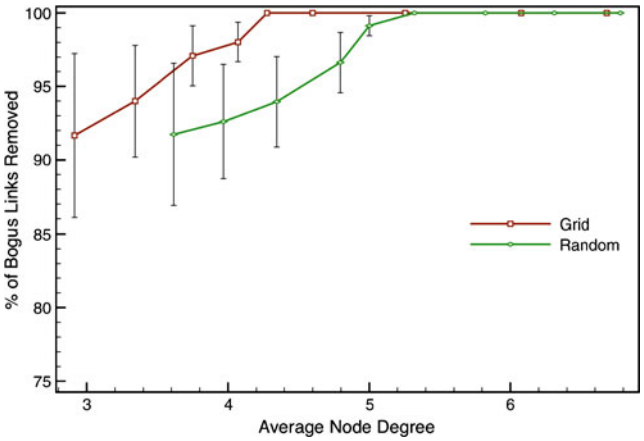

Fig. 23 Bogus link removal as a function of node degree

routes to target nodes exist. Mutual checks further reduce the false positive percentages as shown in Fig. 20. The impact of average node degree on the percentage of legal links removed by mistake for both grid and randomly distributed networks is presented in Fig. 21. The trend in this case is similar to the trends with false positives.

The impact of node degree on wormhole detection rates is shown in Fig. 22. For both grid and randomly distributed networks the results show that the detection process can detect wormholes successfully even for networks with very low node degree (3–3.5). Even lower node degrees (< 3) may result in nodes being unable to find alternate routes as required by SECUND. Similarly, an average node degree of 5 ensures that most bogus links are removed as shown in Fig. 23.

4.3.4 Overhead analysis

The number of *Detection*() operations performed by each node (for a given network topology, density, and degree) impacts the average number of route acquisitions each node has to employ to create a secure neighborhood with SECUND. These statistics were captured in our simulations (the average node degree in the numerical results reported here is 6.8).

Figure 24 shows the average number of times each node has to perform *Detection*() when mutual detection is used and after using the rules for cooperation in Section 3.3 to reduce the number of checks required. These two cases are labeled "With Mutual Checks" and "Reduced" respectively in Fig. 24. With $\eta = 3$, instead of running *Detection*() 6.8 times, each node runs it around two times after following the rules specified for improving the detection process. The number of route acquisitions that each node has to perform, shown in Fig. 25, falls from around 45 to 10, a savings in overhead of about 80%. Clearly, the rules for reducing the number of checks presented in Section 3.3 provide significant savings in overhead even without impacting performance metrics (e.g., % of false positives, % wormhole detection).

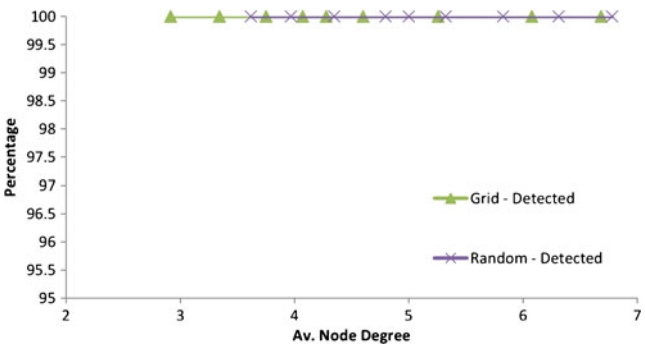

Fig. 22 Wormhole detection rates as a function of node degree

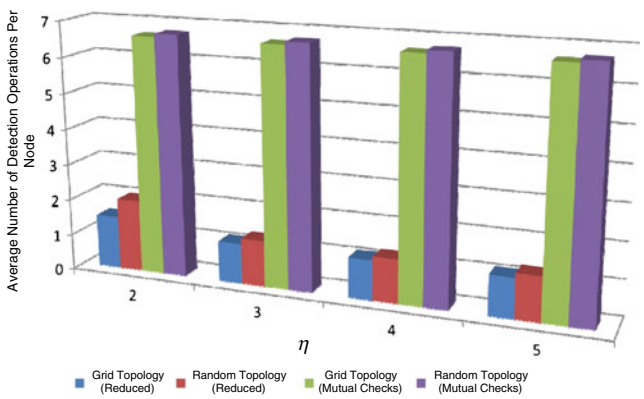

Fig. 24 Detection operations performed per node

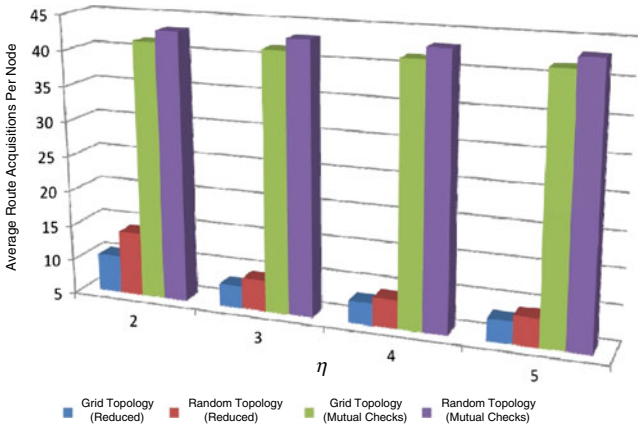

Fig. 25 Route acquisition per node

5 Conclusions and future work

In this paper, we present a localized protocol SECUND, that employs cooperation between neighboring nodes for creating a secure neighborhood in ad hoc networks. SECUND employs routing hop count discrepancies between neighbors to determine the existence of a wormhole and to remove bogus links created by wormholes. SECUND incorporates simple rules for cooperation that results in a small overhead in terms of the number of links checked for wormholes. It does not have special requirements such as location information, very high node degree, accurate synchronization between nodes, or special hardware. SECUND was tested through simulations for different distributions of nodes in networks and different types and lengths of wormholes. Under a variety of evaluated scenarios, SECUND demonstrated excellent wormhole detection rate and with few false alarms. Further, the protocol was show to be capable of removing most bogus links from the network while removing few if any legal links.

Acknowledgements This work was funded in part by the Army Research Office MURI grant W911NF-07-1-0318. The authors are grateful to the anonymous reviewers for their comments and suggestions to improve the paper.

Appendix: Detection example in a grid network

We illustrate how the detection process works with a grid network as on example. It is far easier to detect wormholes in grid networks if the topology is known a priori simply because of the increase in the number of neighbors. We use the grid network only for the purpose of illustrating the algorithm. The process works for random topologies and perturbed grids as well, as shown in Section 4. Consider Fig. 26 that shows a

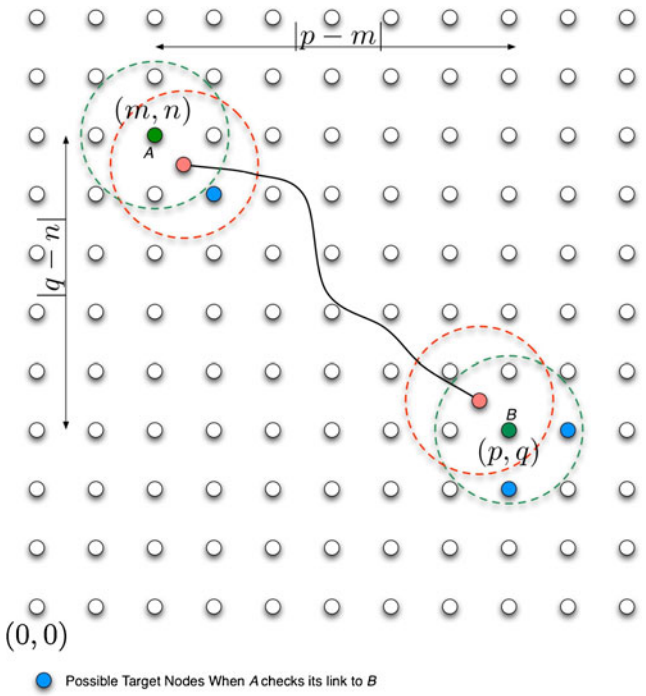

Fig. 26 Example of a grid network

grid network where the grid spacing is d. If we ignore d, we can use integer coordinates for nodes in two dimensions. In Fig. 26, the bottom-leftmost node has coordinates (0,0). We assume that the ranges of nodes are such that each node has exactly four neighbors (those that are at a distance d from a node) as shown in the figure. Let a wormhole exist with its endpoints having the the same range as regular nodes. Let end-point M_1 be located at coordinates $(m_{1,x}, m_{1,y})$ such that $m < m_{1,x} < m+1$ and $n > m_{1,y} > n-1$. Let the other end-point M_2 be located at coordinates $(m_{2,x}, m_{2,y})$ such that $p > m_{2,x} > p+1$ and $q < m_{2,y} < q-1$. Since m, n, p, q can be anything, this does not lose any generality although we assume $p > m$ and $q < n$ by at least a few hops in what follows. We can observe the following properties for this scenario.

1. Given a node with coordinates (m, n), it can only reach nodes at $(m, n-1)$, $(m, n+1)$, $(m-1, n)$ and $(m+1, n)$ when there is no wormhole.
2. The true shortest path between a node at (m, n) and a node at (p, q) has $|p-m| + |q-n|$ hops. In Fig. 26, $(m, n) = (2, 8)$ and $(p, q) = (8, 3)$. The shortest path between these nodes is $6+5 = 11$ hops.
3. The set N_{M_1} consists of nodes with coordinates (m, n), $(m+1, n)$,$(m, n-1)$,$(m+1, n-1)$. The set N_{M_2} consists of nodes with coordinates (p, q), $(p-1, q)$, $(p, q+1)$, $(p-1, q+1)$.

4. The wormhole physically spans a physical distance of $d = \sqrt{(m_{x,1} - m_{x,2})^2 + (m_{y,1} - m_{y,2})^2}$, but the minimum number of true hops between any node $\in N_{M_1}$ and any node $\in N_{M_2}$ is $|m - p| + 2 + |n - q| + 2$ hops. This is the shortest path from the node at $(m + 1, n - 1)$ to the node at $(p - 1, q + 1)$. In the example in Fig. 26, this is 7 hops.

Let us suppose that a node A at (m, n) is checking its link with a node B at (p, q) to see if a wormhole is in its vicinity. When the node A at (m, n) initially asks for its neighbors, it gets responses from nodes at: $(m, n-1)$, $(m, n+1)$, $(m-1, n)$, $(m+1, n)$, (p, q), $(p-1, q)$, $(p, q+1)$, $(p-1, q+1)$. If we identify nodes by their coordinates:

- $N_A = \{(m, n-1), (m, n+1), (m-1, n), (m+1, n), (p, q), (p-1, q), (p, q+1), (p-1, q+1)\}$
- $\hat{N_A} = (m, n-1), (m, n+1), (m-1, n), (m+1, n)\}$
- $N_A^* = \{(p, q), (p-1, q), (p, q+1), (p-1, q+1)\}$.

Similarly, $N_B = \{(p, q+1), (p, q-1), (p-1, q), (p+1, q), (m, n), (m+1, n), (m, n-1), (m+1, n-1)\}$. From this, we can see that $N_B - N_A = \{(p+1, q), (p, q-1), (m+1, n-1)\}$. The target node T that is picked by A belongs to this set, i.e., $T \in \{(p+1, q), (p, q-1), (m+1, n-1)\}$. Node A will ask the nodes in N_A to find routes to T avoiding nodes in N_A and N_B and having at least one intermediate node. We see there are three cases:

- Case 1: If $T = (p + 1, q)$, the lengths of routes will be as shown in Table 4 (assuming $p > m$ and $q < n$—otherwise absolute values of the differences will have to be used). Unless $p - m + n - q + 2 < \eta + 3$, the wormhole will most certainly be detected.
- Case 2: If $T = (p, q - 1)$, a similar tabulation of routes indicates that the longest route is still $p - m + n - q + 2$ hops long and a similar conclusion is reached. We do not tabulate the length of routes in this case.
- Case 3: If $T = (m + 1, n - 1)$, the lengths of routes will be as shown in Table 5. In this case, the wormhole is detected as long as $p - m + n - q - 3 > \eta + 3$.

Table 4 Lengths of routes to $T = (p + 1, q)$

Node in N_A	Length of route to T
$(m, n-1)$	$p - m + n - q$
$(m, n+1)$	$p - m + n - q + 2$
$(m-1, n)$	$p - m + n - q + 2$
$(m+1, n)$	$p - m + n - q + 2$
(p, q)	Not possible
$(p-1, q)$	6
$(p, q+1)$	2
$(p-1, q+1)$	5

Table 5 Lengths of routes to $T = (m + 1, n - 1)$

Node in N_A	Length of route to T
$(m, n-1)$	3
$(m, n+1)$	5
$(m-1, n)$	3
$(m+1, n)$	3
(p, q)	Not possible
$(p-1, q)$	$p - m + n - q - 3$
$(p, q+1)$	$p - m + n - q - 3$
$(p-1, q+1)$	$p - m + n - q - 4$

References

1. Papadimitratos P, Poturalski M, Schaller P, Lafourcade P, Basin D, Capkun S, Hubaux J-P (2008) Secure neighborhood discovery: a fundamental element for mobile ad hoc networking. In: Proc. of IEEE communications magazine
2. Yih-Chun H, Perrig A, Johnson DB (2006) Wormhole attacks in wireless networks. IEEE J Sel Areas Commun 24(2):370–380
3. Hayajneh T, Krishnamurthy P, Tipper D (2009) Deworm: a simple protocol to detect wormhole attacks in wireless ad hoc networks. In: In proceedings of the IEEE symposium on network and system security
4. Wang X, Wong J (2007) An end-to-end detection of wormhole attack in wireless ad-hoc networks. In: In proc. of international conference on computer software and applications
5. Hu YC, Perrig A, Johnson DB (2003) Packet leashes: a defense against wormhole attacks in wireless networks. In: In Proce. of IEEE INFOCOM
6. Eriksson J, Krishnamurthy SV, Faloutsos M (2006) Truelink: a practical countermeasure to the wormhole attack in wireless networks. In: Krishnamurthy SV (ed) Proceedings of the 2006 14th IEEE international conference on network protocols, 2006. ICNP '06, pp 75–84
7. Tran PV, Hung LX, Lee Y-K, Lee S, Lee H (2007) Ttm: an efficient mechanism to detect wormhole attacks in wireless ad-hoc networks. In: In Proc. of IEEE CCNC
8. Poturalski M, Papadimitratos P, Hubaux J-P (2008) Secure neighbor discovery in wireless networks: formal investigation of possibility. In: In proceedings of the ACM symposium on Information, computer and communications security
9. Chiang JT, Haas JJ, Hu Y-C, Kumar PR, Choi J (2009) Fundamental limits on secure clock. Synchronization and man-in-the-middle detection in fixed wireless networks. In: Proc. of IEEE INFOCOM
10. Shokri R, Poturalski M, Ravot G, Papadimitratos P, Hubaux J-P (2009) A practical secure neighbor verification protocol for wireless sensor networks. In: Proc. of ACM WiSec
11. Hu L, Evans D (2004) Using directional antennas to prevent wormhole attacks. In: Network and distributed system security symposium (NDSS), San Diego
12. Capkun S, Buttya'n L, Hubaux J-P (2003) Sector: secure tracking of node encounters in multi-hop wireless networks. In: Proceedings of the 1st ACM workshop on security of ad hoc and sensor networks. ACM, Fairfax, Virginia, pp 21–32

13. Sastry N, Shankar U, Wagner D (2003) Secure verification of location claims. In: Proceedings of the 2nd ACM workshop on wireless security. ACM, San Diego, CA, USA, pp 1–10
14. Wang W, Bhargava B (2004) Visualization of wormholes in sensor networks. In: Proceedings of the 3rd ACM workshop on wireless security. ACM, Philadelphia, PA, USA, pp 51–60
15. Poovendran R, Lazos L (2007) A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. Wirel Netw 13(1):27–59
16. Qian L, Song N, Li X (2007) Detection of wormhole attacks in multi-path routed wireless ad hoc networks: a statistical analysis approach. J Netw Comput Appl 30(1):308–330
17. Maheshwari R, Gao J, Das SR (2007) Detecting wormhole attacks in wireless networks using connectivity information. In: Gao J (ed) INFOCOM 2007. 26th IEEE international conference on computer communications. IEEE, pp 107–115
18. Hou Y-T, Chen C-M, Jeng B (2007) Distributed detection of wormholes and critical links in wireless sensor networks. In: Proc. of IIHMSP
19. Lee C, Suzuki J (2010) Swat: a decentralized self-healing mechanism for wormhole attacks in wireless sensor networks. In: Xiao Y, Chen H, Li F (eds) Handbook on sensor networks, chap 24. World Scientific Publishing. ISBN: 978-981-283-730-1
20. Znaidi W, Minier M, Babau J-P (2008) Detecting wormhole attacks in wireless networks using local neighborhood information. In: Proc. of IEEE PIMRC
21. Khalil I, Bagchi S, Shroff NB (2007) Liteworp: detection and isolation of the wormhole attack in static multihop wireless networks. Comput Netw 51(13):3750–3772
22. Kuhn F, Zollinger A (2003) Ad-hoc networks beyond unit disk graphs. In: Proceedings of the 2003 joint workshop on foundations of mobile computing. ACM, San Diego, CA, USA, pp 69–78
23. Kim T-H, Tipper D, Krishnamurthy P, Swindlehurst L (2009) Improving the topological resilience of mobile ad hoc networks. In: Proc. of IEEE design of reliable communication networks (DRCN 2009), Washington, DC, 25–29 Oct 2009
24. Kim T-H, Tipper D, Krishnamurthy P (2009) Connectivity and critical point behavior in mobile ad hoc and sensor networks. In: Proc. of IEEE symposium on computers and communications (ISCC '09), 5–8 July 2009