Local Connectivity Tests to Identify Wormholes in Wireless Networks

Xiaomeng Ban
Computer Science
Stony Brook University
xban@cs.sunysb.edu

Rik Sarkar
Computer Science
Freie Universität Berlin
sarkar@inf.fu-berlin.de

Jie Gao
Computer Science
Stony Brook University
jgao@cs.sunysb.edu

ABSTRACT

A wormhole attack places two radio transceivers connected by a high capacity link and retransmits wireless signals from one antenna at the other. This creates a set of shortcut paths in the network, and may attract a lot of traffic to the wormhole link. The link thus gains control of a large fraction of network traffic which opens the door for more dangerous attacks afterwards. In this paper we introduce a wormhole detection and removal algorithm based on local connectivity tests.

The basic idea is that the neighborhood of a wormhole contains two sets of nodes corresponding to two sides of the wormhole. The distance between these two sets is small when using paths that pass through the wormhole link, but is large when only regular network paths are considered. Thus we remove a small neighborhood that will contain potential wormhole links and check if a slightly larger neighborhood falls apart to multiple connected components. To accommodate spatial and temporal unpredictability of wireless communication links we abstract the network connectivity as an arbitrary graph so that the method does *not* assume any idealistic models (such as unit disk graph model). The algorithm uses purely local connectivity information, handles multiple wormhole attacks and generalizes to wireless networks deployed in 3D. It does not suffer from typical limitations in previous work such as the requirements of special hardware, communication models, synchronization, node density etc. In simulations, our method is seen to beat the state of the art solutions, in particular for cases where previous solutions experience poor performance.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms

Algorithms, Design, Theory

Keywords

Wormhole Attack, Wireless Networks, Network Security

MobiHoc '11, May 16-20 2011, Paris, France

1. INTRODUCTION

A wormhole attack to a wireless network [9] is to place two radio transceivers, connected by high capacity out-of-band wireless or wired links. Signals captured by one antenna are "tunneled" through the wormhole link to the other antenna, and replayed there. In the 'store-and-forward' scheme, the wormhole nodes copy the entire packet before transmittal through the wormhole link. In more sophisticated schemes, the wormhole can be launched *at the bit level* (the replay is done bit-by-bit even before the entire packet is received, similar to cut-through routing [14]) or at the *physical layer* [6] (the actual physical layer signal is replayed, similar to a physical layer relay [18]). Effectively, the wireless nodes near one wormhole antenna find out that they can directly communicate with the wireless nodes near the other antenna and would consider them as immediate neighbors. See Figure 1. A wormhole attack is easy to launch. It is independent of the MAC (medium access control) layer protocols and is also immune to cryptographic techniques. It does not require the adversary to break into the wireless nodes or understand the communication mechanisms employed by the network.

If the adversary only replays the signal faithfully, the presence of wormhole is of no harm or even beneficial as it enhances the network connectivity and creates short paths between otherwise far off regions. When the tunneled distance is larger than the transmission range in the network, nodes near the wormhole antennas find shorter, faster, and probably more reliable paths by tunneling through the wormhole. Wireless networks running any variations of shortest path routing will discover such paths and eventually make use of them to deliver data. For example, take a simple scenario where nodes are uniformly deployed in the domain with d nodes per unit area on average and the wormhole antennas are placed of distance k apart, roughly at least $\pi dk^2/8$ pairs of nodes will find shorter paths through the wormhole link. In another case when one radio transceiver is placed next to a data sink in a sensor network, the wormhole link provides shortcut paths to the sink for $\pi dk^2/4$ nodes. Therefore, a wormhole attack, in particular one with a long tunneling distance, will be able to attract a lot of traffic through the wormhole link. This puts the wormhole link at a powerful position than other nodes in the network and this allows the adversary to exploit this position in a variety of ways.

Since a wormhole attack fundamentally changes the network connectivity, by turning on and off the signal replay an adversary can suddenly create and destroy a large number of shortest paths in the network and upset most routing protocols. In on-demand routing protocols, a wormhole can attract the route request packet through the tunnel and later play denial of service attack by refusing to forward any packets. In routing protocols that periodically discover neighbors, the adversary can trigger frequent neighbor changes and

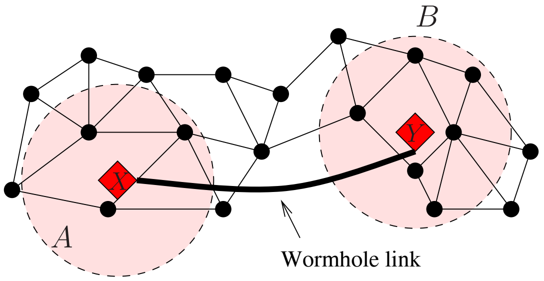


Figure 1. Demonstration of a wormhole attack. X and Y denote the wormhole nodes connected through a long wormhole link. As a result of the attack, nodes in Area A consider nodes in Area B their neighbors and vice versa.

paths changes, which consumes the node energy and communication bandwidth. Even when the wormhole does not shut down its replay scheme, the wormhole can be used to attract network traffic, and can then eavesdrop, maliciously drop packets, or to perform man-in-the-middle attacks. Traffic gathered this way can also help to break encryption and security mechanisms used in the network. Thus wormhole attack opens the door to many more malicious attacks. We measure the *impact* of a wormhole attack by the number of pairs whose shortest paths are affected by the wormhole attack. In this sense, a wormhole attack has larger impact/potentially more damages when the two antennas are placed relatively far away, as more traffic and more paths in the network are affected by the wormhole link. We call such a wormhole to be a 'long' one and it is of most interest to detect those long wormholes in the network.

In addition to messing up with the routing protocols, using wormholes an attacker can also break any protocol that directly or indirectly relies on geographic proximity. For example, target tracking applications in sensor networks can be easily confused in the presence of wormholes. Similarly, all localization algorithms that use network connectivity would fail or be confused by the alteration of the network topology due to wormhole links. This can have a major impact as location is a useful service in many protocols and application, and out-of-band location systems such as GPS are not always available.

1.1 Prior Work

In the literature a number of techniques have been proposed to detect wormhole attacks. These methods have their respective limitations, e.g., assuming additional hardware or explicit communication models or lacking the ability to single out wormhole links. We first review the prior work and then describe our approach using novel algorithmic techniques.

Methods using distance or timing analysis. Packets going through a wormhole take longer to reach the destination due to the delay in reception, transfer and retransmission at the other end. A number of schemes have tried to detect wormhole attacks by measuring packet traverse distance or time. Such methods are generally called packet leashes [2, 6, 8, 17]. The limitation of this method is that one needs to obtain the node location information using out-of-band mechanism such as GPS, or, extremely accurate globally synchronized clocks to bound packet propagation time. It is unclear whether the techniques can be carried out in low-cost hardware such as sensors. Even if so, such timing analysis may not be able to detect cut-through or physical layer wormhole attacks, as such replays can happen quite fast and cannot be detected easily.

Methods using special hardware. Using purely physical layer mechanisms one can prevent wormhole attacks such as those involving authentication in packet modulation and demodulation [8]. But such techniques require special RF hardware. Directional antennas can also be used to prevent wormhole attacks [7]. The requirement of special devices limits the use of such protocols.

Methods using special guarding nodes. A few protocols of this type [11, 12, 15] have been proposed that use special-purpose guard nodes with known locations, higher transmit power and different antenna characteristics, to attest the source of each transmission. The use of such special purpose guard nodes makes this approach limited in applicability.

Methods using neighborhood discovery. Since the placement of wormhole increases the local connectivity at the neighborhood of the wormhole nodes, one can use statistical approaches to detect the increase in number of neighbors and the decrease in lengths of shortest paths between all pairs of nodes due to wormhole presence [1]. A similar approach using statistical measurements of multi-path routing is used in [16]. Both schemes assume that the network is free of wormhole to start with and they are vulnerable if the attack is launched prior to such discovery.

A different approach examines the changes in the connectivity graph by the wormhole attacks and look for 'forbidden substructures' in the connectivity graphs that should not be present in a legal connectivity graph [13]. This approach however assumes fairly detailed knowledge of wireless communication model (i.e., a model that describes with some given confidence whether a link between two nodes should exist) and the performance deteriorates if such a model is lacking.

Methods using global network topology. The last family of work examines the global network topology. Essentially the wormhole attack drastically changes the network connectivity by 'gluing' links between the nodes near wormhole nodes. In [19], distance estimates between sensors are used to determine a "network layout" using multi-dimensional scaling (MDS) technique. Without any wormhole the network layout should be relatively flat. But the layout could be warped in presence of wormholes. Thus detecting whether the network can be embedded on a flat domain can tell whether wormhole attacks are present. This method is centralized and it does not identify nor isolate wormhole attacks.

Dong *et al.* [4] uses the local topological changes around the neighborhood of the wormhole nodes to detect the wormhole links. In particular, one takes a local k-hop neighborhood and see whether the 'boundary' has single or double cycles. Intuitively, the neighborhood that encloses a wormhole link will have two cycles and single cycle otherwise. The limitation of the method is that it requires relatively high node density to ensure that boundary detection algorithm works well, and relies on the local hop count metric being close to the Euclidean metric. They suggest using global topological properties to detect presence of wormholes in [3]. This idea has some merit for certain 2-manifolds, but do not carry over to actual networks, since real world network graphs are not surfaces.

1.2 Our Approach

In this paper we search for a detection method that is not limited to the various constraints as described earlier. The approach we use is to examine graph connectivity, and detect the fundamental connectivity changes a wormhole would introduce. This puts us into the family of protocols that test the network connectivity or global topological changes, such as those described in [3, 4, 13, 19]. Compared with these work, our method makes contributions in the following aspects.

Rigorous Definition of A Wormhole Attack. None of the previous connectivity based detection method has a rigorous definition of what constitutes a wormhole attack in the connectivity graph. Thus there is no provable results on detection ability and the algorithms rely on simulations to evaluate the performance. We introduce a rigorous definition of how a wormhole attack affects the network connectivity. Basically a wormhole would 'shortcut' the paths between two sets of nodes W_0, W_1 that can directly communicate with the two wormhole antennas respectively. Therefore, the wormhole attack introduces links between nodes in W_0 and W_1 and adds the full bipartite graph on W_0, W_1 to the existing topology. The length of the wormhole is dictated by the shortest hop count between nodes in W_0 and nodes in W_1 before the wormhole is introduced.

Guaranteed Detection of Wormhole Sets. All previous algorithms are conservative, in the sense that it is possible to report no wormhole while there is one even in the case of a long wormhole (connecting nodes that are far away in the original network). We consider the false negative to be more dangerous than false positive (that certain legal links are labeled as suspicious). When a false positive link is removed, a valid communication link is lost, but security is not compromised. A false negative, on the other hand, leaves the network insecure. We *prove* that our algorithm *guarantees* to detect all the nodes affected by the wormhole attack. Abstracting away some technical details, in our method we remove a local neighborhood around a node p and check whether a slightly larger neighborhood is connected. If not, p is considered as a suspicious node. We prove for all suitable parameters this simple test is *guaranteed* to identify all the nodes affected by a wormhole. By repeating the test for different sets of parameters we can also substantially reduce the number of false alarms. With the candidate sets, we include additional tests to verify that it is indeed a wormhole structure in our definition. Thus a wormhole set is provably and accurately detected.

Robustness to Different Communication Models and Dimensions. We remark that our detection algorithm looks at network connectivity alone. Thus the method applies to any general network settings. For example, the method does not require any assumption nor knowledge of the wireless communication models (as opposed to the method in [13]). It does not use any geometric intuition that relies on the network being embedded in the plane, as opposed to the methods in [3, 4]. The same algorithm works on networks deployed in 3D.

Scalability and Communication Costs. Our detection algorithm at a node p only uses information of a small bounded neighborhood of p. Thus naturally the algorithm is scalable to networks of large size. The communication cost for the test is low, dependent only on the network degree for each node.

We evaluated the detection performance (in terms of false positive and false negative) with connectivity based methods [3,4]. The results show that our method has better performance in detecting wormholes. In particular when the network model does not follow unit disk graph model the performance of other methods deteriorates substantially. Our method has slightly more false alarms but the detection of wormhole attacks is accurate.

In the following we first present the definition of a wormhole set, the threat model, and then describe the algorithm to detect nodes affected by a wormhole attack. We also discuss methods to eliminate false alarms and to detect multiple wormholes. We then present simulation results and comparisons with other connectivity based methods.

2. WORMHOLE DEFINITION AND LOCAL CONNECTIVITY TESTS

Our algorithm is to detect the anomalies in the graph connectivity. To start we first rigorously define what is the connectivity structure of a wormhole and then describe our algorithm.

2.1 Assumptions and Threat Model

In a wireless network communication links can possibly be directional. That is, A can send messages to B but not vice versa. In this paper we only consider the bidirectional links, as directional links do not support acknowledgement schemes. We assume that the transmission characteristics of the wormhole transceivers are the same as that of the other legal nodes in the network, to enable bidirectional communications.

We assume that the adversary can place wormhole nodes at arbitrary places in the network, and that these nodes are connected through a communication channel that is unobservable by other nodes. The wireless network can adopt efficient symmetric cryptographic schemes (as in [9]) to authenticate communication partners and protect the communication messages. The wormhole attacker simply sniffs traffic on one end and replays on the other end. That is, the attacker does not need to know the cryptographic schemes used in the network to fool the nodes to believe that they have a direct communication link. The wormhole transceivers also do not have identities. In fact, the wireless nodes are not aware of the presence of any special wormhole radios in the neighborhood and just hear about some messages in the air, that are possibly replay messages.

We assume a wireless ad hoc network in which the nodes are not compromised nor malicious. In particular, there is no Sybil attack [10], where a malicious node behaves as if it was a larger number of nodes, for example by impersonating other nodes or simply by claiming false identities. We will discuss the case of compromised or malicious nodes in the discussion section.

2.2 Wormhole Definition

We start with an unweighted communication graph $G = (V, E)$. A wormhole attack captures the signal in the air from one radio transceiver A and then broadcast from another radio transceiver B. As a consequence, all the nodes whose signal reach A and B respectively will think they have direct communication links. This creates a local structure of a full bipartite graph as a subgraph. The damage from a wormhole attack is defined as the number of pairs discovering shorter paths through the wormhole link. Thus, further away the two radio transceivers are, more damage is done by the wormhole attack. On the other hand, very short wormholes do not significantly modify connectivity and are not such a threat. Our wormhole definition captures this parameter by measuring the hop distance k between the nodes connected through a wormhole in the original network in absence of the wormhole. In this paper we assume that k is greater than a sufficiently large constant. All our tests will only use a bounded neighborhood of size determined by k around each node.

Definition 2.1. (k, τ)**-wormhole set.** *A set $W \subset V$ is a (k, τ)-wormhole set if it is a maximal disjoint union $W_0 \cup W_1$ for which the following conditions hold:*

1. *Each edge $(u, v) \in W_0 \times W_1$ is in E. That is, each node in W_0 is a neighbor of each node in W_1. Such edges are called* wormhole edges.

2. *$|W_0|, |W_1| \geq \tau$, that is, there are at least τ nodes whose signals are captured by the wormhole link on either side.*

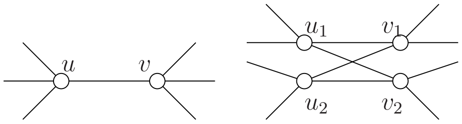


Figure 2. A legal network structure such as a bridge connecting two nodes on the boundary of a hole could also be identified as a 'wormhole' in our definition. However, the same graph structure can be generated by also placing wormhole antennas near u and v. Thus it is impossible to eliminate this case from our definition.

3. *Removing all wormhole edges $W_0 \times W_1$ increases the distance between W_0 and W_1 to be at least k, but does not disconnect any part of the network.*

The set W is said to *maximal* in the sense that no node can be added to it while keeping true to the conditions above. This definition implies that the diameter of W is at most 2. Sometimes we write a wormhole set simply as a k-wormhole to mean that τ is not relevant, or equivalently, $\tau = 1$.

We remark that in certain cases, legal links can be identified as a wormhole set. Consider a network with a 'bridge' connecting two nodes that are otherwise far apart in the network. Such a bridge or bridge like structure falls in our definition. See Figure 2. But such bridges could also be the result from a wormhole attack and there is no way to distinguish them from a real wormhole attack based on graph connectivity only. Thus, our tests will be on the aggressive side and also identify such structures, and report them for further investigation.

Finding a complete bipartite subgraph can be done in the centralized setting when the entire network topology is available. Eppstein [5] shows an algorithm that lists all complete bipartite subgraphs in a network with constant degree. The running time of the algorithm is linear in the size of the graph and exponential in the node degree. We will use this algorithm on local neighborhoods in the final stage of our algorithm to test that the wormholes detected have τ nodes on each side.

2.3 Local Connectivity Test

The idea in our test is to observe that a wormhole attack connects two sets of nodes that are otherwise far away in the graph, while the wormhole set itself is contained in a very small neighborhood. As a node near a wormhole expands its neighborhood, the neighborhood grows on two sides of the wormhole edges. Removing a small region around the node removes the wormhole and disconnects the neighborhood into two components.

Thus our local connectivity test is to check whether a neighborhood of a proper size will fall into multiple connected components. Since wireless communication has a lot of local spatial variations, checking the 1-hop neighborhood does not give reliable results. Thus we consider neighborhoods of different sizes. To be precise, we will introduce the following definitions.

Definition 2.2. **α-ball and $[\alpha, \beta]$-ring.** *An α-ball centered at node p, written as $B_\alpha(p)$ is the set of all nodes with distance at most α-hops from p. All the nodes that are within β hops from p but are more than α-hops away from p are called the $[\alpha, \beta]$-ring $N_{[\alpha,\beta]}(p)$. In symbols : $N_{[\alpha,\beta]}(p) = B_\beta(p) \setminus B_\alpha(p)$. α, β are integers satisfying $\beta > \alpha \geq 1$.*

To test for a wormhole, we first introduce a basic $[\alpha, \beta]$-ring-connectivity test, where α, β are integers satisfying $\beta > \alpha \geq 1$.

Definition 2.3. **$[\alpha, \beta]$-ring-connectivity test for node p.** *Consider the set of nodes $N_{[\alpha,\beta]}(p) = B_\beta(p) \setminus B_\alpha(p)$, and the subgraph in G induced by it. If this subgraph contains more than one connected components, the test returns true, and we say p is a k-wormhole candidate for all $k > 2\beta$. See Figure 3 for an example.*

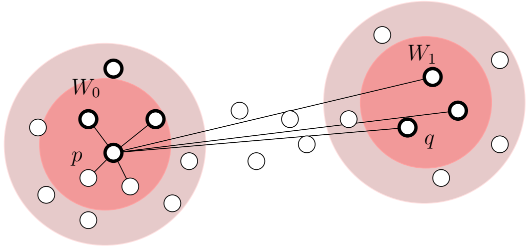


Figure 3. The thick circles represent the nodes within the wormhole range, those on two sides correspond to W_0 and W_1 respectively. The physical wormhole link is not shown since it is not visible in the network connectivity. The darkly shaded region denotes the ball $B_1(p)$, which includes all nodes in W_1. Thus removing $B_1(p)$ also removes all wormhole edges. The lightly shaded region denotes the ring $N_{[1,2]}(p)$. It has two components, one near W_0 and one near W_1.

Guaranteed Detection of Wormhole Sets. We show that if there is a wormhole, the $[\alpha, \beta]$-ring-connectivity test always detects it successfully. For now we consider the case that the network has just a single wormhole set. First we show that the connectivity test will surely label the nodes in a wormhole set.

Theorem 2.4 (Guarantee of detection). *Given a (k, τ)-wormhole set W, all the nodes in W will surely be detected by the $[\alpha, \beta]$-ring-connectivity test, given that $k > 2\beta$, $\beta > \alpha \geq 1$.*

PROOF. Consider a (k, τ)-wormhole set W. Without loss of generality, we take representative nodes $p \in W_0$ and argue that it must be labeled as wormhole candidate. Assume otherwise, then the subgraph induced by $N_{[\alpha,\beta]}(p)$ remain as a single connected component, after we remove the α-ball of p. Recall that all the nodes in W_1 are neighbors of p, thus removing α-ball of p with $\alpha \geq 1$ will surely remove all nodes in W_1. Thus all the wormhole edges are removed as a result. Intuitively the nodes in $N_{[\alpha,\beta]}(p)$ were originally reached from p through either the wormhole edges or not using any wormhole edges. After the wormhole edges are removed, these two sets naturally form disconnected components. We make this intuition rigorous in the following.

Consider the nodes in $N_{[\alpha,\beta]}(p)$. We define the set N_1 to be the nodes whose shortest paths to p go through nodes in W_1, and the set N_0 to be the nodes whose shortest paths to p do not go through nodes in W_1. We argue that the two sets are disjoint, and form disconnected components.

If the subgraph induced by $N_{[\alpha,\beta]}(p)$ has only one connected component, take a node $x \notin W$ in this subgraph. Since $x \in N_{[\alpha,\beta]}(p)$, x is within β hops from p. There are also two shortest paths that connect from p to x, one through the nodes in W_1 (denoted as P_1) and one not through the nodes in W_1 (denoted as P_0). These two paths, concatenated, form a cycle of length at most $2\beta + 1$. See Figure 4 for an example. We now argue that on path P_1 there can only be one node q from W_1, that is, the node immediately after p on P_1. Clearly, if there is another node $q' \in W_1$ further down the path P_1, then one can shortcut the path P_1 as p and q' are also neighbors. This will contradict with the fact that P_1 is a shortest path. Thus, removing the edges between W_0 and W_1 will still leave a path connecting p and q with total length 2β. This contradicts with the definition of a k-wormhole, where $k > 2\beta$. □

The parameters α, β can be varied. Our tests are *aggressive*, in

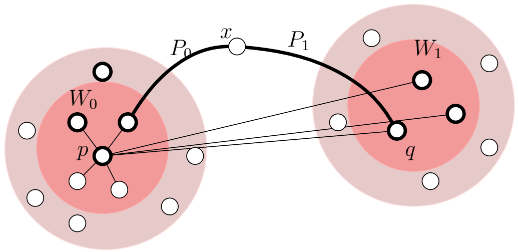


Figure 4. If $N_{[\alpha,\beta]}(p)$ has only one connected component, then there is a path connecting two nodes $p \in W_0$, $q \in W_1$ not using any wormhole edges with total length at most 2β.

the sense that a single wormhole attack will surely be identified for suitably small values of α and β. Thus detection is always guaranteed. Different parameters may introduce different type of false positives. For example, a small β is likely to introduce false positives – that is, certain nodes in sparse regions may be wrongly identified as a candidate because their small neighborhoods are naturally disconnected. But using a large β will show that it is actually not a real wormhole node, since the neighborhoods are connected by a slightly longer path. In our final algorithm we run multiple tests with different parameters and output the nodes that are labeled in *all* tests. We start with smaller values of α, β, and perform additional tests with larger values only on the nodes that are labelled as suspicious – as *wormhole candidates* – by the earlier tests. We take to be wormhole set the nodes that that are detected by all the tests up to a suitable value. Once a set of candidates are detected, we can remove the links connecting the candidates.

2.4 The Wormhole Algorithm

Based on the ring-connectivity test, we describe a simple distributed algorithm that identifies neighborhoods in a network as wormholes. Our goal is to detect wormholes of length k and greater. Since k must be greater than 2β, and β is at least 2, the minimum permissible value of k is 5.

Let us denote by $C_{[\alpha,\beta]}$ the set of nodes detected to be wormhole candidate by the $[\alpha, \beta]$-ring-connectivity test performed at each node in the network.

Algorithm: Connectivity Metric Test. The algorithm consists of performing the test on increasing values of (α, β) in lexicographic order, and performing subsequent tests only at nodes that are labelled candidates by all previous tests. More precisely, we select $\alpha = 1, 2, \ldots, \lfloor(k-3)/2\rfloor$. And for each α, we perform the test for $\beta = \alpha+1, \alpha+2, \ldots \lfloor(k-1)/2\rfloor$. Clearly, the result of the algorithm is a set of candidates

$$\bigcap_{\alpha=1}^{\lfloor(k-3)/2\rfloor} \left(\bigcap_{\beta=\alpha+1}^{\lfloor(k-1)/2\rfloor} C_{[\alpha,\beta]} \right).$$

What we have covered until now addresses the detection of some subgraphs whose presence have a large effect on the metric – the basic symptom of a wormhole. Condition 2 in our definition of wormholes requires that each side of a wormhole have a size τ. We now describe how to check for this threshold. For this, we make use of the algorithm in [5] that finds the maximal complete bipartite subgraphs in any graph. Note that this entire phase can be ignored for $\tau = 1$.

Algorithm: Test for τ Partitions. We take connected components of the subgraphs induced by the nodes detected as wormhole candidates after the connectivity metric test above. Let C be one such connected subgraph.

On the subgraph C, we apply the algorithm of [5]. Let B be the set of maximal complete bipartite subgraphs generated by the algorithm. We write as a pair (W_0, W_1) an element in B, where W_0 and W_1 are the two partitions of the bipartite graph.

On each such bipartite subgraph, we perform the following test. We consider a neighboring subgraph N that consists of nodes that are at a distance at most $\lfloor(k-1)/2\rfloor$ from all nodes in $W = W_0 \cup W_1$, but not the nodes in W itself. Let $N_0, N_1, \ldots$ be the connected components of N.

For any edge $(a, b) \in W_0 \times W_1$, if nodes a and b are neighbors to nodes of N, we check that these are in different components of N. For a graph that satisfies this condition, we check that $|W_0|, |W_1| \geq \tau$. If there is a complete bipartite subgraph that satisfies all these conditions, we have detected a wormhole $W = W_0 \cup W_1$.

Removal of Wormholes. One of the goals of detecting a wormhole is to be able to nullify it unobtrusively. We would like to retain the wireless nodes in action (thus keeping the sensing or computational capabilities of the nodes), but eliminate the high volume of traffic passing through the wormhole link that creates the wormhole effect. We do this by removing the edges $W_0 \times W_1$ in the bipartite graph.

Test for network connectivity. Once a wormhole has been detected and removed, we flood from any one node in it and ensure that the flood reaches all other nodes. This is to guarantee that the network remains connected as required by our definition.

Provable guarantee. Now we are ready to show our main result. The $[\alpha, \beta]$-ring connectivity test is guaranteed to label all nodes in a real wormhole, but may label some legal nodes incorrectly. Together with γ-partition test, the removal and the connectivity test, the false positives are removed so our detection precisely identifies a wormhole in our definition.

Theorem 2.5. *Any (k, τ) wormhole $W = W_0 \cup W_1$ is detected by our test. And, our detection is surely a (k, τ) wormhole.*

PROOF. To show the first claim that our test is effective, we simply need to show that in each of the succession of tests, a real wormhole set (W_0, W_1) is not eliminated. First, (W_0, W_1) is by definition a maximal bipartite graph. Therefore, it will be one of the graphs detected by [5].

Next we need to show that if $(a, b) \in W_0 \times W_1$, and a and b are neighbors to the neighbor set N, they are neighbors to different connected components of N. Suppose to the contrary that they are neighbors to the same connected component. Then there is a node $c \in N$ that is at a distance at most $(k-1)/2$ from both a and b. Thus, there is a path of length $k-1$ from a to b not passing through W. This contradicts the definition of a (k, τ) wormhole.

Finally, by definition, $|W_0|, |W_1| \geq \tau$. Thus every legitimate wormhole is detected by the test.

Now we show that our detections follow the wormhole definition. It is clear that our detection generates a bipartite graph (W_0, W_1) satisfying that each side has at least τ nodes. By the test of τ-partition, we see that without edges in the bipartite graph the nodes in W_0 and W_1 can only be connected by paths of length at least k. By the wormhole removal and connectivity test, the removal of the edges in the bipartite subgraph does not disconnect the network. Thus the detected structure precisely follows the definition of a wormhole. □

Scalability and Communication Costs. The detection method is naturally local and distributed. It is local in the sense that communication distances are bounded by a known parameter, and completely independent of the size of the global network. Each node only uses the connectivity information of the nodes within its β neighborhood, whose size just depends on the average network degree and not on any other property of the network. This makes the algorithm scalable to networks of any size.

For the test for τ sized partition, we aggregate the data about the set C and the adjoining components of N to a single node, and conduct the computation at that node. The algorithm from [5] can be computation intensive in a dense network, since its cost is exponential in degree. But note that we do this only at a few small neighborhoods we consider very likely to contain a wormhole. The overall cost for the network is therefore typically not large. Also, this step can be ignored for $\tau = 1$, which is the value we use in simulations and get very good results.

2.5 Discussions on Parameters

As shown in the previous section, our $[\alpha, \beta]$-ring connectivity test algorithm surely labels the nodes in a wormhole set. If we use the τ-partition test and the wormhole removal and connectivity test we precisely identify a wormhole. It is nice to have such theoretical guarantee but in practice one suggestion is to use the $[\alpha, \beta]$-ring connectivity test only, for the reason of simplicity and low communication requirement. In this way we do not lose any detection power but may identify some false alarms. In this section we discuss a few interesting cases and in particular how the parameters may influence the performance of the algorithm.

Effect of k. The user supplied parameter k essentially determines the sensitivity of the algorithm. A smaller value of k makes the algorithm more sensitive. It can detect smaller wormholes, but introduces a greater chance of false positives. A larger value of k may miss some smaller wormholes, but provides more reliable detection of the longer wormholes. Longer wormholes are more dangerous, since they introduce larger distortion to the graph metric and attract more traffic. Thus, in a sense the algorithm's accuracy automatically scales with the effect of the wormhole, or the danger posed by it.

The Influence of Parameters α, β. Recall that in our detection algorithm there are parameters α, β satisfying $k > 2\beta$, $\beta > \alpha \geq 1$. α is the size of the neighborhood around p to be removed. α is at least one. β must be at least one greater than α to allow a non-empty ring between the α hop and β hop.

While clearly a sufficiently long wormhole will surely be detected for many different combinations of these parameters, an intelligent choice of parameters can lead to fewer false alarms. Our final algorithm tries different sets of parameters and take the intersection of their candidate sets. Notice that our sufficiency proof guarantees that any real wormhole nodes will definitely pass all such tests so we will not miss any real wormholes. We discuss the influence of the parameters in the following.

When β is increased, the $[\alpha, \beta]$-ring has more nodes in it. For an example, take a look at Figure 6 (i). If $\alpha = 1, \beta = 2$, the ring has two nodes that are not connected. But if we increase β to be 3, the ring has three nodes in one connected component. It is also clear that the newly included nodes are always connected to the nodes already in the ring, so there will not be any newly connected emerged components in the ring. Increasing β will always reduce the number of false alarms. The issue is that β cannot be increased arbitrarily due to upper bound of k and higher communication/computation cost.

The parameter α works in an interesting way regarding false alarms. First, when α is small, there can be many false positives in a network that is not well connected. Take a look at Figure 6 (ii). In particular, when there are small 'dangling' nodes, these nodes may lead to identifications of some false positives. But increasing α can enclose all these dangling nodes inside the α ball and thus remove them. For a 'dangling' component with 'depth' of ℓ, using an $\alpha \geq \ell$ will include all dangling nodes inside the α ball and thus eliminate the false positives created this way. On the other hand, making α too big may remove a 'bridge' in the network and thus create falsely identified candidates. Take a look at Figure 6 (iii). A small α does not disconnect the bridge but a large α can fully remove the bridge and report p as a candidate (false alarm).

Eliminate False Alarms with τ. So far in our discussion we focused on the length of a wormhole, denoted by the parameter k, as the minimum hop distance between nodes in W_0 and W_1 once the wormhole edges are removed. Another parameter in a wormhole definition is the size of W_0 and W_1. A wormhole antenna takes all the signal it hears and broadcasts to the other antenna. Thus all the nodes within direct communication range of a wormhole antenna will be affected by the attack. In a case when the node density has a lower bound τ (i.e., an antenna placed at any location can hear from at least τ nodes), then it is clear that $|W_0|, |W_1| \geq \tau$. We can also use this property to eliminate the false alarms. This avoids identifying isolated edges that act as connection between otherwise distant parts of a sparse network.

2.6 Multiple Wormhole Sets

When the network has multiple wormhole sets, our $[\alpha, \beta]$-ring-connectivity test can also detect these wormhole sets if they are far away (and thus 'independently' alter the network connectivity) or too close (thus removing the α neighborhood will remove all related wormhole edges).

Theorem 2.6. *When there are multiple k-wormhole sets, the nodes in the wormhole sets are surely picked up by our $[\alpha, \beta]$-ring-connectivity test, given that $k > 2\beta$, $\beta > \alpha \geq 1$, and either one of the following conditions holds for each pair of wormhole sets W, W':*

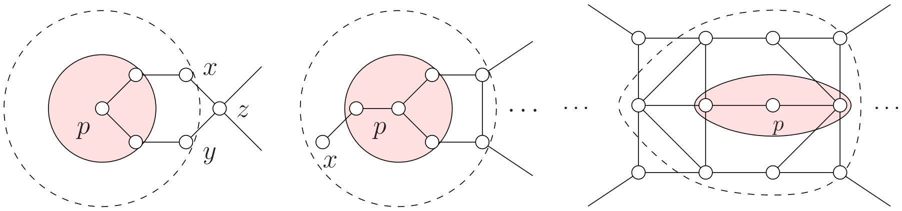
1. *The minimum hop distance between any two nodes that belong to different wormhole sets W, W' is greater than $\beta + 1$.*
2. *There are two nodes $p \in W$, $p' \in W'$ such that p, p' are within $\alpha - 1$ hops of each other.*

PROOF. In the first case, the two wormhole sets W, W' are far apart. Thus when we run the test at a node $p \in W$, all edges involved are within β hops from p. That means the existence of W' does not affect the test we run around p. Thus all nodes in W are still identified.

In the second case, the two wormhole sets are 'close'. Basically there is a node $p \in W$ and $p' \in W'$, p, p' are within $\alpha - 1$ hops of each other. Now if we do a test on any node $x \in W$, then all the nodes in W' are within α hops from each other. Thus the wormhole tests running on p will remove the wormhole edges of both W and W'. Thus the test will also turn out to label p as a candidate, since there cannot be any edges of W' that affect the results (i.e., decrease the number of connected components). □

The test for size τ of wormholes can be carried out as usual. In the second case, the detection of the complete bipartite graphs can help in identifying the fact that there are in fact two wormholes.

The case when our detection algorithm fails with multiple wormholes is when the multiple wormholes are carefully placed at a proper distance from each other such that they interfere. An example is shown in Figure 6. The removal of the α-ball around a node p does not leave the nodes in the ring in different connected components — as they can possibly be connected through another wormhole. In fact, in this case any single wormhole itself does



(i) (ii) (iii)

Figure 5. The α-ball is shown as the shaded region and the nodes within β-ball are within the dashed cycle. (i) If we take $\alpha = 1, \beta = 2$, p will be identified as a candidate since $x, y \in N_{[1,2]}(p)$ are not directly connected. But if we use $\beta = 2$, $N_{[1,2]}(p)$ has three nodes x, y, z and is connected. This way the false alarm for p is removed. (ii) p has a dangling path of length 2. For $\alpha = 1, \beta = 2$, the dangling node x is not connected with other nodes in the ring. Increasing α to be 2 will remove such dangling paths. (iii) Consider a bridge of 3 hops wide as shown in the figure. Consider a test at p with $\alpha = 1, \beta = 2$. The nodes in the ring are connected and thus p is not a candidate in this test. But if we increase $\alpha = 2, \beta = 3$, the entire bridge will be removed and the nodes in the ring will be disconnected. Thus large α will not necessarily reduce the number of false positives.

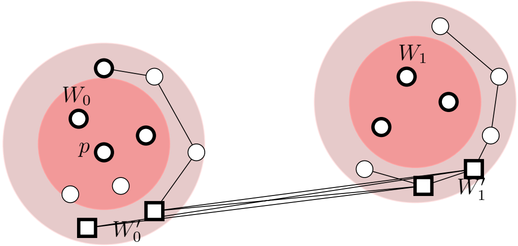


Figure 6. There are two wormhole attacks (W_0, W_1) and (W_0', W_1'), one on top of the other. Nodes in the second set are shown as squares. The edges after the removal of $B_\alpha(p)$ (darkly shaded region) are shown. The second wormhole connects what would have been the two components of $N_{[\alpha,\beta]}(p)$, which now appears to have one component and is not detected in connectivity tests.

not actually follow our Definition 2.1. The two wormholes interfere with each other such that the removal of edges from only *one* of them does not leave the nodes with long paths in the network. However, if the wormholes are long, that is, if k is large compared to the separation between W_0 and W_0', then removing a sufficiently large α-ball disconnects both wormholes, and detects a candidate. This property can be used to detect potential threats of multiple wormholes though it does not identify the wormholes precisely.

3. SIMULATIONS

3.1 Simulation Setup

We evaluated our algorithm using extensive simulations under various conditions, including different node distributions and density, radio models, positions of wormholes, and different test parameters.

Node Distribution. Two node deployment models are used in our simulations: grid with perturbation and random placement. In the model of grid with perturbation, the wireless nodes are placed on an $m \times n$ grid, each cell in the grid is a square with edge length d. Then each node with coordinate (x, y) will be perturbed around its initial position with displacement parameter p: its coordinate will be uniformly randomly drawn from the region $[x-pd, x+pd]\times[y-pd, y+pd]$. By varying p, we can get various node placements with different levels of regularities. In random placement, each node is assigned a coordinate uniformly randomly drawn from the network field. Random distribution typically has more irregularity than the perturbed grid distribution. In our simulations, we also extend both types of node placement strategies to three dimensional networks.

Radio Models. To determine links between nodes, we adopt both unit disk graph (UDG) and quasi-UDG settings. In the UDG setting, each pair of nodes u and v has an undirected link between them if and only if their distance is no greater than R, where R is the communication radius. Quasi-UDG adopts a more practical link generation model: each pair of nodes u and v will have a link if their distance is no greater than r. Besides, they will have a link with probability q if their distance is within $[r, R]$. In our simulation, we set $r = 0$ for quasi-UDG. By adjusting the parameters in UDG and quasi-UDG, we vary the average degree in the network from 6 to 20.

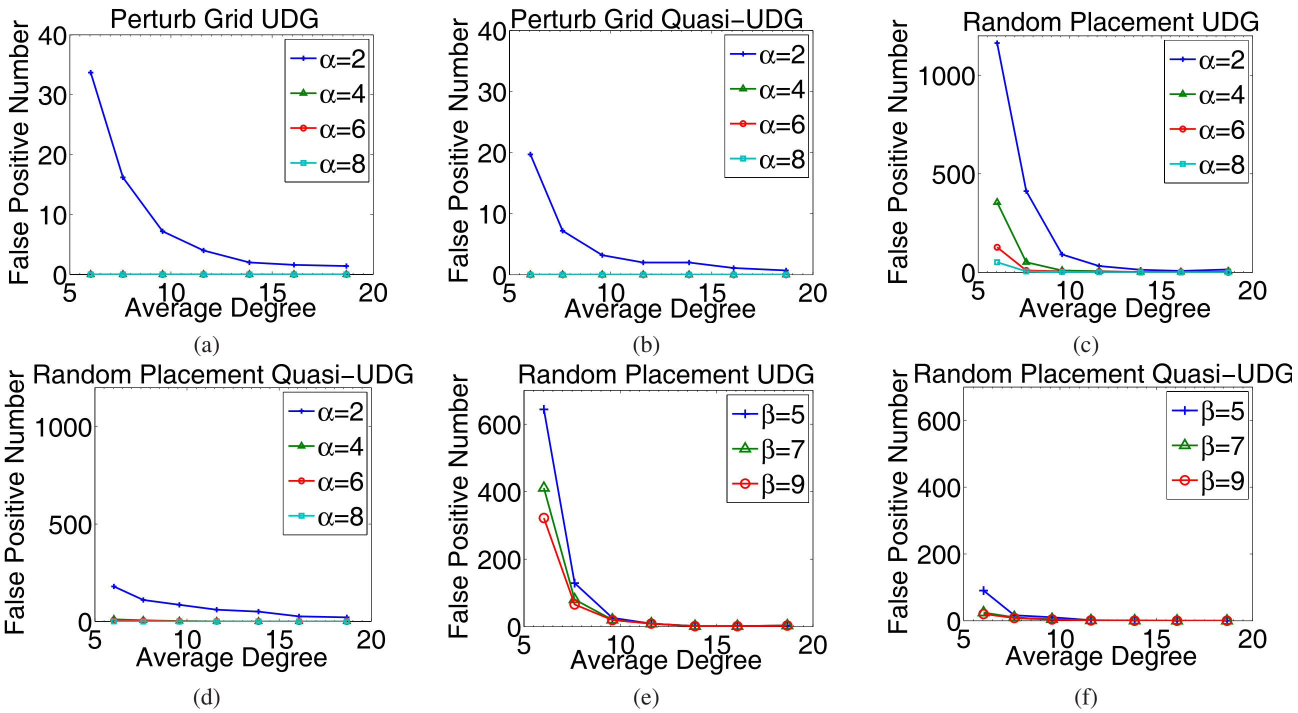
Wormhole Placement. The location of wormholes is a crucial factor in wormhole detection. The length of a wormhole is important: a wormhole is significant only when it is reasonably long. In previous work [4], the placement of wormhole antennas turns out to be another important factor: for the antennas being placed near the network boundary or sparse regions certain algorithms may experience deteriorating performance. Previous schemes did not tackle the case of multiple wormholes. Multiple wormholes detection is influenced by their relative positions. In our simulations, we vary the length of wormholes, put the antennas at different positions of the network, and change the relative positions of two or more wormholes.

3.2 Simulation Results

3.2.1 False Positive Rates By Ring Connectivity Tests

Our ring connectivity test guarantees to detect true wormhole nodes, which means that there are no false negatives. Our method may run for multiple rounds using different α, β parameters. For each round, we only test the candidates that have passed all previous rounds. We evaluate the number of false positive nodes in each round, by varying different setup parameters: node distribution, density, α, β, and radio models (UDG or quasi-UDG).

Influence of Node Distributions and Density. Figure 7 shows that in general there are much fewer false positives for networks with perturbed grid distribution than networks of uniform random distribution, since a network of perturbed grid is more regular. Second, with the same node deployment method and the same average degree, our detection methods have fewer false positive nodes on quasi-UDGs than UDGs. This observation is a bit counter-intuitive but confirms that our method does not rely on the communication models. In particular, on quasi-UDGs previous methods typically perform worse, especially for location based techniques. Figure 7 also shows that as the average degree grows, the number of false



(a) (b) (c)

(d) (e) (f)

Figure 7. The number of false positive nodes on a network with 5000 nodes. In the first four figures, we vary α to be $2, 4, 6, 8$ and take $\beta = \alpha + 2$. In the last two figures, we take $\alpha = 3$ and take β as $5, 7, 9$ respectively. (a) Perturbed grid with UDG model, perturbation ratio $p = 0.4$. (b) Perturbed grid with quasi-UDG model, $p = 0.4$. quasi-UDG radius $r = 0$,$q = 0.5$. (c) Random distribution with UDG model. (d) Random distribution with quasi-UDG model, $r = 0$,$q = 0.5$. (e) Random distribution with UDG model.(f) Random distribution with quasi-UDG model, $r = 0$,$q = 0.5$.

positive nodes drops very fast.

Effect of α and β. From Figure 7 shows that the increase of α and β reduces the number of false positive nodes. This resonates with our design idea in which we test the (α, β) parameters in lexicographic order, gradually removing false positives. Notice that we take the candidates that pass all tests, the number of false positives is very small.

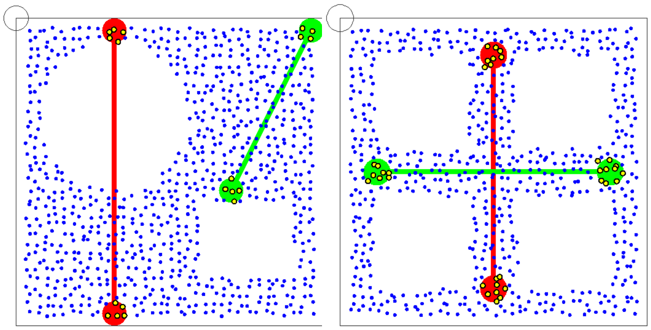


Figure 8. Example of wormhole placement, Network size is 1000, average degree is 6, $\alpha = 1$, $\beta = 3$.

Wormhole Placement. Certain schemes proposed earlier are extremely sensitive to the positions of wormholes. For example, the WormCircle method [4] divides the wormhole positions into different cases and under certain cases the detection rate is high, while in other cases, e.g. placing wormhole antennas on network boundaries, the detection rate is much lower. Our method is not influenced much by the wormhole placement. We show different scenarios in Figure 8. It shows that we can place wormhole antennas near the network outer boundary, or near holes, the detection is always effective and accurate.

3D Wireless Networks. A wireless Network may be deployed in 3D space, say, under water or in a multi-floor building. Most previous results would fail in 3D networks. The method that uses forbidden substructure in [13] can be extended to 3D, but would need very high node densities and detailed radio models. The WormCircle method [4] strictly assumes the underlying geometry to be two dimensional, and does not generalize to 3D at all.

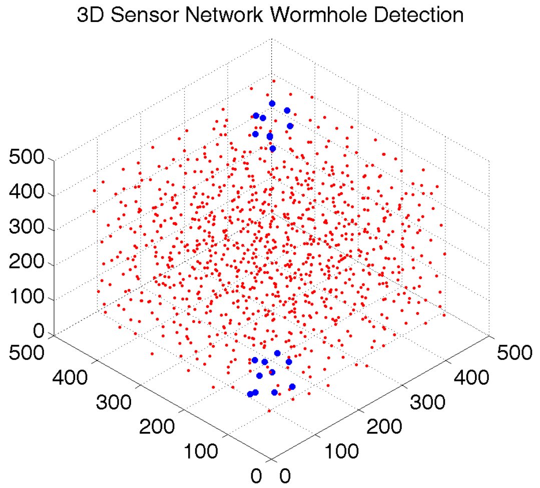


Figure 9. Wormhole detection in a 3D network. Network topology is formed by using a 3D grid with perturbation. The network has 1000 nodes. We use $\alpha = 3$, $\beta = 5$. The wormhole transceivers are located near a pair of diagonal corners and the nodes affected are accurately detected as highlighted in the figure.

Our method operates purely in terms of graph connectivity, without any dependency on the dimension of the network. Therefore it works naturally in 3D. Figure 9 shows an example of wormhole detection on 3-dimensional wireless network. The behavior of our method in a 3D network is similar to that on a 2D network.

3.2.2 *Communication Cost*

Our detection mechanism requires all nodes to participate initially, and the suspicious nodes participate more rounds using different parameters. For a test using parameter α, β, a node will need to gather the connectivity information for all nodes within β hops. While the nodes participating in more detection rounds will introduce higher communication cost, the number of participants is fairly small compared to the total number of nodes. Figure 10 shows the communication cost in terms of packets transmitted for each node on average for the entire detection process. There are a few interesting observations. First, the communication cost is smaller for networks built by a perturbed grid model than networks of randomly distributed nodes. This is because there are fewer false positives in a perturbed grid. Second, when the network density increases, obviously it would incur a higher cost to collect the connectivity in local neighborhood as there are more nodes. However, when the average degree increases, fewer nodes are marked suspicious in the first round, which leads to a decrease of communication cost in later rounds. The combination of the two factors shows the interesting trend of first increasing and then decreasing for the case of a network of random node distribution.

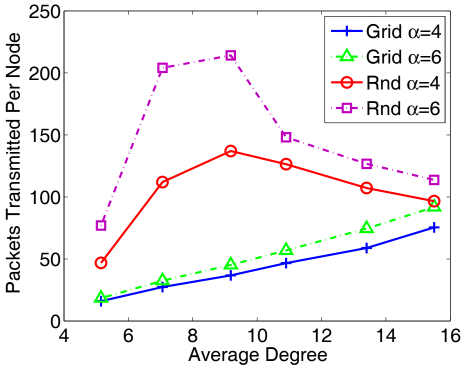


Figure 10. Communication cost in terms of packets transmitted. Network has 5000 nodes, $\beta = \alpha + 2$. Grid is perturbed grid with UDG, perturbation ratio $p = 0.4$. Rnd is node random placement with UDG.

3.2.3 *Multiple Wormholes*

When multiple wormholes are placed simultaneously, they may interfere with each other, making the detection harder. The interference of two wormholes depends on the relative positions of their antennas: as long as there exists at least one antenna which is far away from other antennas, those two wormholes will not affect each other in terms of detection. Figure 11 shows three scenarios. From top to bottom, in the first one the antennas of the two wormholes are far away from each other. In the middle, several wormholes share one antenna and the other antennas are far from each other. In both cases the wormhole nodes are well recognized. The last case is an interesting example where the second wormhole reduces the length of a previous existing wormhole. The wormhole nodes are detected for smaller values of α, β (left). But they are not detected when we use a larger set of α, β parameters (right).

3.2.4 *Comparison with Wormcircle*

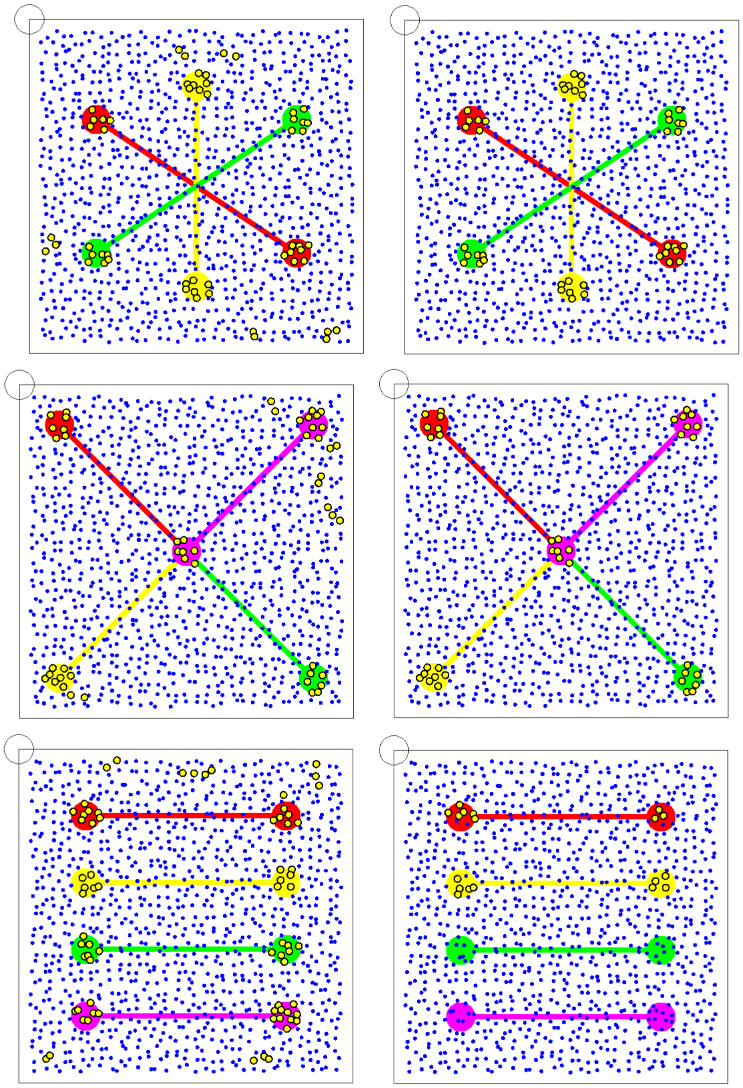


Figure 11. Multiple Wormholes. Left: $\alpha = 1$, $\beta = 3$; Right: $\alpha = 2$, $\beta = 4$.

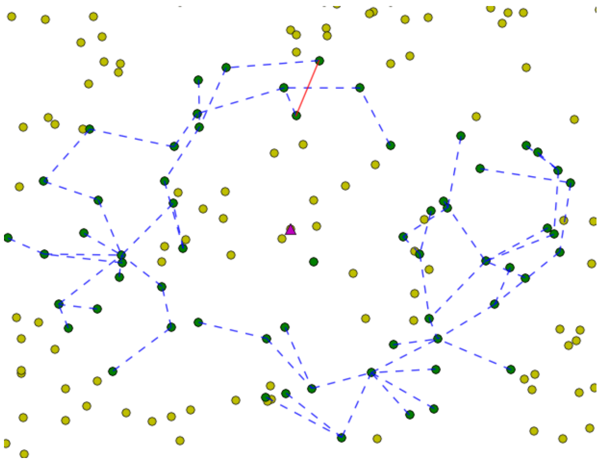


Figure 12. A wormhole detected by localized wormcircle at a regular node, in a quasi unit disk graph. The 3-hop ring has two components. Edges in dashed blue show the breadth first trees in the two cases. The red solid edge is detected as a cut edge, implying a long cycle in one of the trees and a false detection.

We compared the performance of our method with the Wormcircle algorithm [4].This algorithm is based on the idea that presence of a wormhole changes the geometry and topology of the ring of nodes at k-hops from a root node. Without wormhole, the k-hop ring should have the connectivity as a ring. If one antenna of the wormhole is less than k hops away from the root, then the set of k hop nodes will appear as two rings. The cut locus method of [20] is used to determine the topology of the k-hop band. The paper presents two different algorithms based on this principle.

The *basic Wormcircle* scheme starts with a designated root node in the network, and computes the breadth-first tree from this node. Next, it considers the connected components of nodes at k hops for each k. In a Euclidean or similar domain, each component resembles a circle. However, the connected component induced by the wormhole will have a smaller radius. In particular, the main connected component is expected to have a circumference of $2\pi k$, where the distant wormhole component will have a much smaller circumference. By comparing the circumference to $2\pi k$, a wormhole can be detected.

The *localized wormcircle* scheme takes a more topological approach. It computes a shallow breadth-first tree around every node and considers the k-hop ring. If the k-hop ring has two components and at least one of them resembles a circle, a wormhole is said to be detected. The circumference of the circle is not considered.

Avg Degree	False negative	False positive
6.3	60%	00%
7.7	50%	10%
9.0	50%	20%
10.3	40%	20%
11.5	30%	30%
12.8	30%	20%
14.1	30%	30%
15.3	20%	20%
16.8	20%	30%
18.0	20%	30%

Table 1. Wormcircle performance over 20 networks in each degree range. The first column shows the average degree of 20 networks. The false negatives show the percentage of cases that the algorithm failed to detect an actual wormhole, while false positives show the percentage of networks that did not have any wormhole but was erroneously detected to have one.

These methods depend heavily on the geometry of the network resembling a Euclidean plane. On graphs that are more general than that, they can fail frequently. The localized wormhole algorithm, while in some ways similar to ours, is still tied to the Euclidean geometry, and expects a circle as in that case. Our simulations show that if a network has significant *holes* or is not a unit disk graph, both these methods perform poorly.

Figure 12 shows a network constructed as a quasi unit disk graph. In the figure, the edge in red is detected as a cut edge. that is, it connects leaves of the same breadth first tree, such that the leaves are far apart within the tree itself. This method is used to confirm the presence of a circle. As seen in this example, in networks that are less geometric, this strategy can fail by detecting a cycle that does not resemble a circle at all. In our simulations and in [4] the localized method performs better than basic wormhole. Therefore we only present the results for localized wormhole in the following.

Table 1 shows the performance of localized wormcircle. We created a wormhole with end points 20 units apart in a region of diameter 40 units. Then we added nodes randomly and created networks in quasi unit disk model. We selected networks of different densities, and obtained 20 networks in each range. It is seen that wormcircle makes substantial errors in detecting wormhole. In comparison, our method detected presence or absence of wormhole correctly in all these cases.

In network structures with wormholes placed next to holes such as those in Figure 8, we find that wormcircle performs even more poorly. In these cases, the hole breaks the circular structure of the wormcircle. Thus it fails to detect the actual wormhole in all cases, though sometimes it detects wormhole at incorrect locations. Whereas our method is not affected in any significant way by the presence of holes.

3.3 Network Dynamics

In practice, wireless links may experience various types of dynamics, both temporal and spatial. Here we consider the setting that links fail randomly with a probability p. In our method, all nodes participate in the detection of wormhole region, but they may not enter the detection phase at the same time. Therefore, each node may have different view of the network topology due to potential dynamic link failures. When a single transmission fails, we may re-transmit and give up after a maximum K number of trials. In Table 2, we can see that when link failure rate is relatively low, our method still works fine on the tested networks. As failure rate grows, for random placement with UDG, the false positive node number increases dramatically, which makes our identification of wormhole infeasible. This can be understood since the network topology varies significantly and different nodes have very different views.

	0%	1%	5%	10%	15%	20%
Grid	0	0.03	0.05	0.13	0.27	0.46
Q-Grid	0.21	0.24	0.46	0.82	1.40	2.36
UDG	3.40	4.32	20.62	41.67	91.50	180.6
Q-UDG	0.11	0.20	0.32	0.37	5.33	27.17

Table 2. The average number of false positive nodes under random link failure. The network has 2000 nodes and average degree is 8. $\alpha = 5$, $\beta = 7$. The maximum number of retransmissions is 30. Grid is a network with perturbed grid distribution with UDG model, in which the perturbation ratio $p = 0.4$. Q-Grid is a network with perturbed grid distribution with quasi-UDG model, $p = 0.4$. quasi-UDG model uses $r = 0$, $q = 0.5$. UDG is a networrk of node random placement with UDG model. Q-UDG is a network with node random placement with quasi-UDG model, $r = 0$, $q = 0.5$.

4. DISCUSSION

4.1 Malicious Nodes

Our connectivity tests detect the bipartite subgraph introduced by the presence of wormholes. Notice that such connectivity change does not need the help of any compromised nodes. In the case when some nodes are compromised, a malicious node can choose not to cooperate with the local connectivity tests or report incorrect connectivity information. For example, the nodes that are within communication range of the wormhole antennas can choose not to report the edges faked by the wormhole link. However, not reporting the presence of the link faked by the wormhole attack would be equivalent to not imposing the attack to the network. That is, for the wormhole attack to truly alter the network connectivity and for such connectivity change to be observed and used by the honest nodes – to make any real damage — then the local connectivity tests can be executed to examine such possibilities.

However, a malicious node may impose sybil attacks and fake many node identities or even create phantom subgraphs. This will surely add to the detection difficulty. For example, a node x within the $[\alpha, \beta]$ ring of a node p may wrongly claim itself to be identical to a node near the other side of the wormhole antenna, thus causing the detection algorithm to fail. Since a sybil attack may create all kinds of incorrect graph structures we remark that the wormhole attack together with carefully positioned sybil attack may change the network topology in such a way that the wormhole links do not follow our definitions. Thus we defer the discussion of such combined, more sophisticated attacks to be the future work.

5. CONCLUSION

In this paper we examine the network connectivity and propose a local, distributed method to detect suspicious nodes. The method compares favorably with existing connectivity based methods. We believe this strategy can be improved further. For example, the multiple wormholes detection possibly can be improved by a more careful execution of connectivity and bipartite graphs test. The issue of eliminating false positives also remains open for closer investigation.

Acknowledgments We would like to thank the anonymous reviewers and our shepherd Dr. Konstantinos Pelechrinis for their valuable comments and suggestions that helped to improve this paper. Xiaomeng Ban and Jie Gao would like to acknowledge the support from NSF through CNS-0643687 and CNS-1016829. Rik Sarkar would like to acknowledge support from the German Research Foundation (DFG) through the research training group Methods for Discrete Structures (GRK 1408).

6. REFERENCES

[1] L. ButtyÃąn, L. DÃşra, and I. Vajda. Statistical wormhole detection in sensor networks. In *Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, volume 3813, pages 128–141, 2005.

[2] S. Capkun, L. ButtyÃąn, and J. P. Hubaux. SECTOR: Secure tracking of node encounters in multi-hop wireless networks. In *1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, October 2003.

[3] D. Dong, M. Li, Y. Liu, X.-Y. Li, and X. Liao. Topological detection on wormholes in wireless ad hoc and sensor networks. In *Proceedings of the 17th annual IEEE International Conference on Network Protocols (ICNP'09)*, pages 314–323, 2009.

[4] D. Dong, M. Li, Y. Liu, and X. Liao. Wormcircle: Connectivity-based wormhole detection in wireless ad hoc and sensor networks. In *ICPADS '09: Proceedings of the 2009 15th International Conference on Parallel and Distributed Systems*, pages 72–79, Washington, DC, USA, 2009. IEEE Computer Society.

[5] D. Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51(4):207–211, August 1994.

[6] J. Eriksson, S. Krishnamurthy, and M. Faloutsos. Truelink: A practical countermeasure to the wormhole attack. In *ICNP*, 2006.

[7] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Network and Distributed System Security Symposium (NDSS)*, 2004.

[8] Y. C. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM*, volume 3, pages 1976–1986, 2003.

[9] Y.-C. Hu, A. Perrig, and D. Johnson. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, 24:370–380, February 2006.

[10] N. James, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN '04, pages 259–268, New York, NY, USA, 2004. ACM.

[11] I. Khalil, S. Bagchi, and N. Shroff. MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks. In *Ad Hoc Networks*, volume 6, pages 344–362, May 2008.

[12] I. Khalil, S. Bagchi, and N. B. Shroff. LITEWORP: A Lightweight Countermeasure for the Wormhole attack in multihop wireless network. In *International Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, 2005.

[13] R. Maheshwari, J. Gao, and S. R. Das. Detecting wormhole attacks in wireless networks using connectivity information. In *Proceedings of the 26th Conference of the IEEE Communications Society (INFOCOM'07)*, pages 107–115, May 2007.

[14] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2):62–76, 1993.

[15] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *ACM Journal of Wireless Networks (WINET)*, 13, January 2005.

[16] L. Qian, N. Song, and X. Li. Detection of wormhole attacks in multi-path routed wireless ad hoc networks: a statistical analysis approach. *J. Netw. Comput. Appl.*, 30(1):308–330, 2007.

[17] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *ACM Workshop on Wireless Security (WiSe 2003)*, September 2003.

[18] A. Scaglione and Y. W. Hong. Opportunistic large arrays: Cooperative transmission in wireless multihop ad hoc networks to reach far distances. *IEEE Transactions on Signal Processing*, 51(8), 2003.

[19] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 51–60, New York, NY, USA, 2004.

[20] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 122–133, September 2006.