On the Resilient Overlay Topology Formation in Multi-hop Wireless Networks

Fei Xing and Wenye Wang

Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695, USA
fxing@ncsu.edu, wwang@ncsu.edu

Abstract. In this paper, we study the problem of how to design overlay topologies in multi-hop wireless networks such that the overlays achieve perfect resilience, in terms of all cooperative nodes included but misbehaving nodes excluded, and preserve the k-connectivity with high probability. To address this problem, we propose a new distributed topology control protocol called *PROACtive*. By using PROACtive, every node pro-actively selects its cooperative adjacent nodes as neighbors by mutually exchanging neighbor request and reply messages. As a result, the union of all neighbor sets forms a *resilient overlay* for a given network. Our analysis finds that the PROACtive protocol is light-weighted with the message complexity of only $O(m)$, where m is the number of links in the original network. Our simulation results validate the effectiveness of PROACtive and show that the overlays generated by our protocol preserve the k-connectivity with high probability ($> 90\%$) and low false positive ratio ($< 5\%$).

1 Introduction

Multi-hop wireless networks, especially mobile ad hoc networks, are more vulnerable to failures compared with wired networks due to nodal mobility and error-prone wireless channels. In addition, node misbehaviors, such as selfishness by refusing to forward packets of other nodes and maliciousness by launching Denial of Service (DoS) attacks, can also cause failures. For example, two DoS attacks, *Jellyfish* and *Blackhole*, were shown in [1] to have the network partitioning effect which degrades the network performance severely. In [2], a stochastic analysis on node isolation problem also shown that misbehaving nodes may damage the connectivity of mobile ad hoc networks substantially. Since misbehaving nodes may not provide connectivity to other adjacent nodes, existing routing protocols cannot cope with the failures caused by misbehaving nodes, which leaves the design of resilient multi-hop wireless networks an open and challenging problem in the presence of misbehaving nodes.

To enhance the resilience to misbehaving nodes, some efforts were made by using different approaches. Two techniques called *watchdog* and *pathrater* were proposed in [3] to identify misbehaving nodes and avoid them in routes. A credit-based system called *Sprite* was proposed in [4] to stimulate cooperation among

I.F. Akyildiz et al. (Eds.): NETWORKING 2007, LNCS 4479, pp. 1–12, 2007.

selfish nodes. In [5], a secure ad hoc routing protocol called *Ariadne* was presented to prevent attacks from tampering routing control messages by using symmetric cryptographic primitives. Multi-path routing scheme in [6] introduced redundancy to avoid single path failure caused by node failures or node misbehaviors. Nevertheless, the previous schemes are "passive" to misbehaving nodes since even a misbehaving node can be detected, it is very difficult to prevent it from being selected as intermediate relays for all paths.

In this paper, we study the problem of how to design overlay topologies in multi-hop wireless networks such that the overlays achieve perfect resilience, in terms of all cooperative nodes included but misbehaving nodes excluded, and preserve the k-connectivity with high probability (w.h.p.). Through the formation of resilient overlays, routing and data transferring can be performed upon cooperative platforms. Our contributions are mainly on two aspects.

1. A new distributed and localized protocol called *PROACtive* is proposed to generate a *resilient overlay* for a given network. By using PROACtive, every node pro-actively selects only cooperative adjacent nodes as its neighbors, which results in the exclusion of misbehaving nodes from the overlay.
2. The PROACtive protocol is shown to be light-weighted with the message complexity of only $O(m)$, where m is the number of links in the original network, and the overlays preserve k-connectivity w.h.p. ($> 90\%$) and low false positive ratio ($< 5\%$).

Note that our objective distinguishes itself from the existing topology control works [7,8,9], which usually focused on minimizing the energy consumption as well as keeping networks connected. For example, in [7], the *K-Neigh* protocol, based on distance estimation, was proposed to preserve the connectivity of static multihop networks, with efficient power consumption, by selecting K closest neighbors for each node. Nevertheless, our approach differs from the existing resilience-enhancing works in that we employ the topology control technique in the PROACtive protocol to connect cooperative nodes dynamically by mutual neighbor selections.

The remainder of this paper is organized as follows. In Section 2, we formulate the problem. In Section 3, we describe the details of the PROACtive protocol. In Section 4, we validate our approach by simulations, followed by conclusions in Section 5.

2 Problem Statement

In this section, we describe the system model and formulate the *perfect resilient overlay generation (PROG)* problem.

2.1 System and Threat Model

In this paper, we denote multi-hop wireless networks by $\mathcal{M}(\mathcal{N})$, where $\mathcal{N}$ is the set of nodes. All nodes are assumed to be distributed independently and uniformly on a two-dimensional plane, and they use omni-directional antennas with the same transmission radius r. For a pair of node u and v, they are called *adjacent* if the distance between them, denoted by $d(u, v)$, is no greater than r. When $d(u, v) > r$, u and v may communicate via multiple intermediate hops. It is known that the *geometric random graph (GRG)* [10], denoted by $G(N, r)$, is a graph in which N vertices are independently and uniformly distributed in a metric space with edges existing between any pair of vertices u and v if and only if $d(u, v) \leq r$. Thus, we use the GRG to model the underlying topologies of multi-hop wireless networks.

In order to identify the type of misbehaviors our work has targeted, we loosely classify the different types of misbehaviors in a multi-hop wireless network below, though the classification is not intended to be comprehensive. *(I)* Nodes participate in routing but not in data forwarding, like *Jellyfish* and *Blackhole*; *(II)* Nodes do not cooperate in forwarding control or data packets for others, like selfish nodes; *(III)* Compromised nodes, though appearing to be legitimate, malfunction maliciously; *(IV)* Malicious attacker nodes generate DoS traffic or signals, impersonate legitimate nodes, or tamper with routing messages. Our research focuses on misbehavior *(I)*; while our approach can also be applicable to address misbehavior *(II)* and *(III)*. Our approach, however, does not address misbehavior *(IV)*, which requires suitable authentication and privacy mechanisms. Further, colluding attacks are out of the scope of this work.

2.2 Problem Formulation

The objective of this work is to enhance the resilience of multi-hop wireless networks against node misbehaviors. As mentioned in Section 1, misbehaving nodes may undermine network connectivity and network performance. Here we take an example to look at the effect of misbehaving nodes on path reliability. For a path with h relay hops, let the probability of any relay node being failed (due to node mobility or energy depletion) be P_f, then the path reliability, denoted by R_P, can be presented by $R_P = (1 - P_f)^h$. While, if any relay node may also misbehave to disrupt communications, the representation of R_P becomes $R_P = (1 - P_f - P_m)^h$, where P_m is the probability of any node misbehaving. Then we can easily show that R_P can be significantly decreased by the route disruption effect resulting from misbehaving relays, and the negative impact is more exaggerative when the number of hops h increases.

Therefore, it has been an important issue in the design of resilient networks to "exclude" misbehaving nodes. For a multi-hop wireless network in the presence of misbehaving nodes, we call a connected subnet consisting its *all* and *only* cooperative nodes as a *perfect resilient overlay (PRO)*. If the routing and data transfer operations are conducted only on the induced PRO, then the communication between cooperative nodes is guaranteed to be resilient to misbehaving nodes. Here we formulate our problem as the *perfect resilient overlay generation (PROG)* problem, as follows:

Definition 1. PROG Problem*: Given a connected multi-hop wireless network $\mathcal{M}$ and a connectivity requirement k, generate a perfect resilient overlay $\mathcal{M}^-$ such that $\mathcal{M}^-$ is k-connected with high probability.*

To solve the PROG problem, we propose a distributed and localized protocol called *PROACtive*, by which every node can pro-actively select cooperative adjacent nodes as its *neighbors*, which will be described in detail right next.

3 PROACtive Protocol Design

In this section, we propose the *PROACtive* protocol as the solution to the PROG problem.

3.1 Basic Idea

In the PROACtive protocol, each node is assumed to be able to know whether its adjacent nodes forward packets for other nodes. For example, if wireless cards operate in *promiscuous mode*, a node can use the *Watchdog* method [3] to tell if its next-hop node drops packets instead of forwarding. By this way, a node u should quantitatively measure the *cooperativity* (borrowed from Biochemistry) of its adjacent nodes, which indicates the likelihood that a node performs normal network operations. Based on the obtained cooperativities, a node u can select neighbors by sending the soliciting messages called *Neighbor Request (Ngbr-Rqst)* to its adjacent nodes with *high* cooperativities. Once receiving an acknowledge message called *Neighbor Reply (Ngbr-Rqst)* from one of its adjacent nodes, say node v, then u knows that v has agreed to accept u as v's neighbor, so u can add v into its neighbor set. By this mutual neighbor selection process, a cooperative node can have a cooperative neighborhood easily; while a misbehaving node can hardly have any neighbors. As a result, the union of cooperative neighbor sets generates a perfect resilient overlay which excludes misbehaving nodes.

To satisfy the constraint of the k-connectivity, we refer to some results shown in a few recent literatures, which reveal the probabilistic relations between the connectivity $\kappa(G)$ and the minimum degree $\delta(G)$ of the graph G. It was proved in [10] (*Theorem 1.1*) that if N is sufficiently large, the GRG $G(N, r)$, obtained by adding links between nodes in the order of increasing length, becomes k-connected at the instant when it achieves a minimum degree of k, w.h.p.. In [11] (*Theorem 3*), it was shown that for a GRG $G(N, r)$

$$Pr(\kappa(G) = k) \approx Pr(\delta(G) \geq k) \tag{1}$$

holds if $N \gg 1$ and $Pr(\delta(G) \geq k) \approx 1$. The result was further verified by extensive simulations in [11], [12], and [13]. Moreover, it was shown in [2] that to achieve the k-connectivity in a multi-hop wireless network where misbehaving nodes present, a necessary condition is that each node should have at least k cooperative neighbors. This result implies that for a network $\mathcal{M}$, if let $\theta(\mathcal{M})$ denote the minimum number of cooperative neighbors of $\mathcal{M}$, then

$$Pr(\kappa(\mathcal{M}) = k) \approx Pr(\theta(\mathcal{M}) \geq k) \tag{2}$$

holds for the sufficiently large system size N. Therefore, in our protocol, each (cooperative) node should maintain at least k cooperative neighbors such that the k-connectivity is achievable in the overlay w.h.p.. In the next section, we describe the details of our approach.

3.2 PROACtive Protocol Details

As described briefly in Section 3.1, the essential idea of the PROACtive protocol is to build up cooperative neighbor sets, which is done by the mutual neighbor selections via *Ngbr-Rqst* and *Ngbr-Rply* message exchanges. In this section, we provide the detailed procedures of building up cooperative neighbor sets. For clarity of the description, we denote the adjacent nodes and cooperative neighbors of a node u by $Adj(u)$ and $Ngbr(u)$, respectively, and denote the cooperativity of a node u by $c(u)$. We will first discuss the procedure of querying potential neighbors as follows.

Once a node u knows the cooperativities of its adjacent nodes, u selects a node v of the highest cooperativity from set $Adj(u)$ as a potential neighbor, if v is not in set $Ngbr(u)$. Then u sends a *Ngbr-Rqst* message to v, indicating that u intends to add v to its neighbor set. If u receives a *Ngbr-Rply* message from v within a timeout, then u can add v into set $Ngbr(u)$; otherwise, u queries another adjacent node of the next highest cooperativity. Node u will continue the inquiries until k *Ngbr-Rply* messages from different adjacent nodes are received, which guarantees u with at least k neighbors. This procedure is summarized by **Algorithm 1** as follows.

Algorithm 1. Procedure of querying potential neighbors

```
Input: $k$, node $u$, and $Adj(u)$
Initiate $Ngbr(u) := \emptyset$,
create a temporary set $Temp(u) := \emptyset$,
create a counter $numRplyRcvd := 0$
$\forall v \in Adj(u)$, Measure $c(v)$
while ($numRplyRcvd < k$ AND $Temp(u) \neq Adj(u)$) do
    Select $v$ if $c(v) = \max\{c(w) : \forall w \in Adj(u) - Temp(u)\}$
    Send Ngbr-Rqst to $v$
    $Temp(u) := Temp(u) + v$
    if (Receive Ngbr-Rply from $v$) then
        $Ngbr(u) := Ngbr(u) + v$
        $numRplyRcvd := numRplyRcvd + 1$
    end if
end while
```

Next we discuss how a node processes the incoming neighbor requests. In our approach, each node, say u, can calculate its own threshold, based on the information available from its local environment, to decide if it should accept a querying node as its neighbor. We call this threshold as the *neighbor cooperativity threshold* and denote a node u's threshold by $c^*(u)$. For a node u, since u can measure the behaviors of its adjacent nodes quantitatively, u's threshold can be defined as the average cooperativity of its adjacent nodes, i.e.,

$$c^*(u) = \frac{1}{d} \sum_{\forall v \in Adj(u)} c(v), \text{ for } d = |Adj(u)|. \tag{3}$$

Hence, when u receives a *Ngbr-Rqst* message from one of its adjacent nodes v, it compares $c(v)$ to its threshold $c^*(u)$. If $c(v) \geq c^*(u)$, u replies v with a *Ngbr-Rply* message and adds v into u's neighbor set; Otherwise, u discards this neighbor request and replies nothing. **Algorithm 2** summaries the procedure of processing neighbor requests.

Algorithm 2. Procedure of processing neighbor requests

```
Input: node $u$, and $Adj(u)$
$\forall v \in Adj(u)$, Measure $c(v)$
Calculate $c^*(u)$ by Equation (3)
if (Receive Ngbr-Rqst from $v \in Adj(u)$) then
    if ($c(v) \geq c^*(u)$ AND $v \notin Ngbr(u)$) then
        Send Ngbr-Rply to $v$
        $Ngbr(u) := Ngbr(u) + v$
    else
        Discard Ngbr-Rqst
    end if
end if
```

By this mutual neighbor selection, a node with high cooperativity, in contrast to the nodes with low cooperativities, may receive multiple requests and immediate replies from adjacent nodes. Consequently, a resilient overlay topology can be constructed from the neighbor sets. We use **Algorithm 3** to summarize the perfect resilient overlay generation.

Algorithm 3. Generate a perfect resilient overlay $\mathcal{M}^-$

```
Input: $k$, and a multi-hop wireless network $\mathcal{M}$
Let $\mathcal{N}^-$ be the node set of $\mathcal{M}^-$, $\mathcal{N}^- := \emptyset$
for each $u \in \mathcal{M}$ do
    Build up $Ngbr(u)$ by using Algorithms (1) and (2)
    $\mathcal{N}^- := \mathcal{N}^- \cap Ngbr(u)$
end for
return $\mathcal{M}^-$ induced from $\mathcal{N}^-$
```

3.3 Cooperativity Measurement Scheme

To measure the cooperativity of a generic node, we investigate the characteristics of misbehaving nodes on the network layer. A selfish node, for selfish reasons such as saving energy, usually refuses to forward data packets for other nodes.

A malicious node can do anything such as dropping partial data packets at a random or periodic manner, or pretending to be adjacent to a node actually far-away from it, thus trapping all packets destinated to that node afterwards. Thus, dropping "transient" packets is one of the most common characteristic of most misbehaviors, especially the *Type-I* misbehavior mentioned in Section 2.1. This observation implies that for a node u, we can use u's packet drop ratio, denoted by q_{drp}, to measure u's cooperativity $c(u)$. Let $n_{fwd}(u)$ and $n_{drp}(u)$ denote the numbers of packets should be forwarded and dropped, then we have,

$$c(v) = 1 - q_{drp}(v) = 1 - \frac{n_{drp}}{n_{fwd}}. \tag{4}$$

We use an example in Fig. 1(a) to illustrate this method. In Fig. 1(a), every time node u asks node w to forward a packet to v, u increases a counter $n_{fwd}(w)$ by 1. If u cannot overhear w's forwarding after a timeout (e.g., *round-trip delay*), u increases another counter $n_{drp}(w)$ by 1. Moreover, when one of u's adjacent nodes, x, requires w to forward packets to v, u can record w's behavior as well. Based on the measurements from both "own experience" and "indirect observation", u can calculate w's cooperativity $c(w)$ by (4).

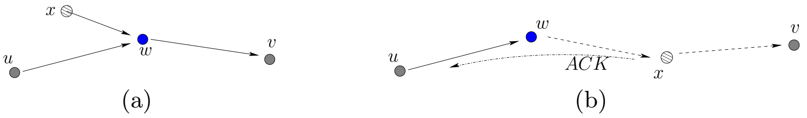


Fig. 1. Measuring cooperativity (a) by promiscuous mode (b) by ACKs

Notice that in our PROACtive protocol, the cooperativity measurement scheme is not limited to the technique that employs promiscuous mode only; it can also use other techniques such as close-loop feedbacks. For example, in Fig. 1(b), when u sends a data packet to w, it can piggyback an ACK request. Based on whether u can receive an ACK from one of w's downstream nodes, say x, u may tell if w has forwarded the packet successfully.

3.4 Features of PROACtive Protocol

In this section, we discuss some unique features of our approach. First, the "individual" threshold defined in (3) allows each node to reach a trade-off between *system* resilience and *individual* connectivity, compared to a global threshold. This is due to the fact that a node surrounded by nodes with relatively low cooperativities can hardly find enough neighbors although a relatively high global threshold can achieve a resilient overlay of only cooperative nodes. On the contrary, by using the individual threshold, for a node u with adjacent nodes of relatively low cooperativities, its neighbor cooperativity threshold can decrease accordingly. Thus u may still have enough "neighbors". Nevertheless, a global threshold can be also applicable for our protocol and more flexible decision policies in neighbor can be designed by combining the global threshold and individual thresholds.

Second, regarding the issue of neighbor set updating, our PROACtive protocol is able to deal with the dynamic topology changes due to node mobility. For instance, a mobile node can refresh its neighbor set when it detects a disconnection with its neighbor(s). If the topology is highly dynamic (e.g., mobility is high), then mobile nodes can keep the records of its neighbors to avoid frequent neighbor set updates. Further, the PROACtive protocol is able to deal with the dynamic changes of node behaviors as well due to the flexibility provided by our threshold design. For instance, the updating overhead can be also reduced by deleting a neighbor only if its cooperativity is below the minimum requirement for a specific application.

Third, our approach does not involve new security vulnerabilities and can avoid the false accusation problem. For instance, the cooperativity information measured by one node are not shared with others in our protocol, and the neighbor selection is only dependent on each node's own knowledge to its neighborhood. By this way, one node's cooperativity cannot be falsely rated to a low or high value by several other (might be malicious) nodes, which prevents any node from the false accusation. No information sharing also helps to avoid the complexity of deciding the actual cooperativity of one node when multiple different measurements are received. Nevertheless, the integrity of *Ngbr-Rqst* and *Ngbr-Rply* messages can be protected by traditional cryptographic techniques.

Finally, the PROACtive protocol is completely distributed and localized, which makes our approach more feasible to be implemented in a real scenario. Additionally, our protocol can be run locally, in an on-demand manner, whenever a mobile node detects a significant cooperativity change among its neighborhood. Further, our protocol is light-weighted in terms of the overhead of message exchanges. For example, given a wireless multi-hop network, in the worst case, every node should send either a *Ngbr-Rqst* or *Ngbr-Rply* message to each of its adjacent nodes in order to build up a neighbor set. This implies that to generate a resilient overlay, the total number of messages needed is no more than the number of links, denoted by m, in the original network. Therefore, the message complexity of our protocol is only $O(m)$.

Until now, the PROG problem, how to generate a perfect resilient overlay, has been solved by applying the PROACtive algorithm on given networks. We will evaluate the effectiveness of our solution by simulations in the next section.

4 Simulation Evaluations

To evaluate our PROACtive protocol, we performed a considerable body of experiments by using *NS2 v2.28* and *MATLAB v7sp3* tools. In our simulations, nodes are distributed randomly and uniformly in an area. Distinct node pairs randomly establish constant bit rate (CBR) connections, with packet size of 512 *bytes* and rate of 5 *packets/sec*, such that nodes can measure their adjacent nodes' coopertivities by the method described in Section 3.3. The AODV routing protocol is used. The connectivity requirement k is 3 for all simulations.

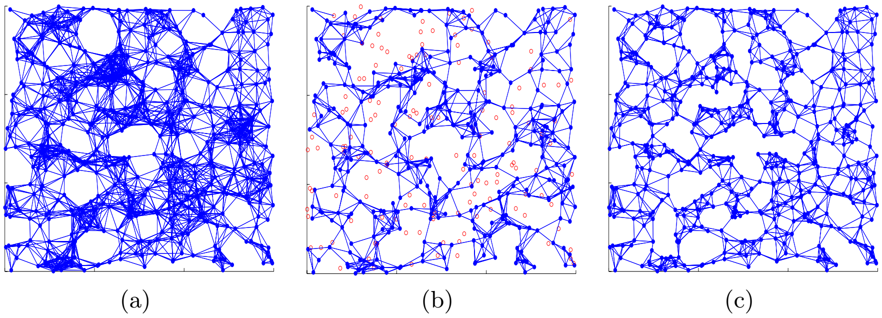


Fig. 2. The topologies of the underlying network and overlays generated: (a) no topology control, (b) applied with PROACtive, (c) *K-Neigh* with $K = 9$ (Phase I)

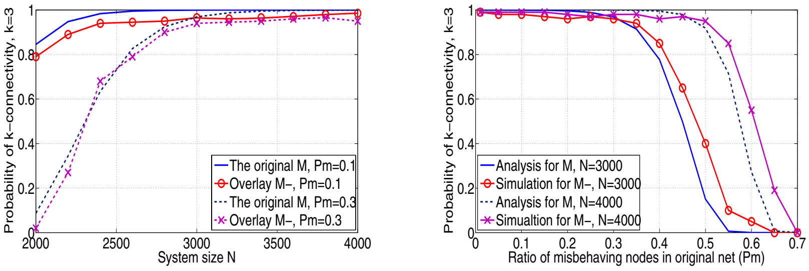
We show how the PROACtive protocol generates overlays first, then show the effectiveness of our protocol with regard to the k-connectivity preservation and the false positive (negative) rate.

4.1 Topology Generated by PROACtive Protocol

In this simulation, 500 nodes are distributed on a $1500\,m \times 1500\,m$ area at random with the same transmission radius of $150\,m$. Among 500 nodes, 150 misbehaving nodes drop packets to be forwarded or report false routes. Fig. 2(a) illustrates the network without applying any topology control, in which a pair of nodes are connected by a link as long as their distance is no larger than $150\,m$. Fig. 2(b) shows the network structure after applying our PROACtive protocol, in which cooperative and misbehaving nodes are represented by solid and hollow dots, respectively. From the figure, we can see that the overlay topology generated by PROACtive excludes most of misbehaving nodes, while containing almost all cooperative nodes. Though some links are removed in the overlay due to the mutual neighbor selection, the generated overlay is still connected. To highlight the feature of our protocol, the topology generated by the *K-Neigh* protocol (Phase 1 only, with $K = 9$) [7] is shown in Fig. 2(c), where we can see that all misbehaving nodes are included in the topology. This is because the neighbor selection in K-Neigh is only based on the distance between nodes, that is each node selects K nearest adjacent nodes as its neighbors.

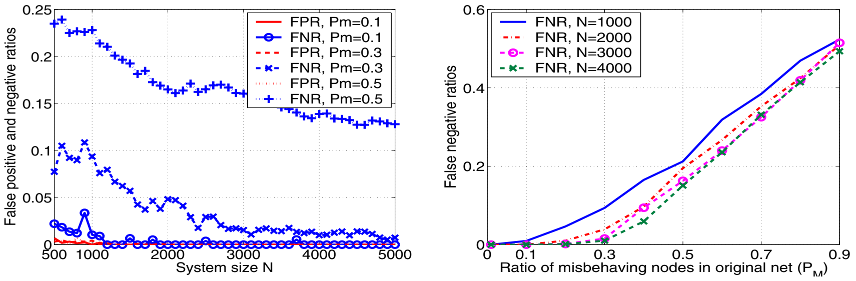
4.2 Preservation of k-Connectivity

One of the major tasks in our simulations is to verify that the k-connectivity should be preserved, w.h.p., for the overlays generated by PROACtive, when the original network is k-connected. To test whether the network is k-connected, we use breadth first search (BFS) to compute how many disjoint paths connecting two distinct nodes. In this simulation, the nodes are placed uniformly at random in a bounded region of $2000\ m \times 2000\ m$. The transmission radius is set to $100\ m$. The number of nodes N ranges from 500 to 5000 with an interval of 100,



(a) (b)

Fig. 3. The k-connectivity probabilities of the overlays generated, compared to those of the original networks: (a) against N, (b)against P_M



(a) (b)

Fig. 4. The false positive and negative ratios produced by PROACtive (a) against N, (b) against P_M (FNR only)

which makes the network placements representative for both sparse and dense networks. For every value of N, a certain number of nodes are misbehaving, whose ratio to the total, denoted by P_M, is ranging from about 1% up to 90% with a interval of 10%. The simulation results are shown in Fig. 3(a) and 3(b). For clarity, only the results for $2000 < N < 4000$ are shown, and we omit the part for $P_M > 0.7$ by the similar reason. From the two figures, we can see that the overlays generated preserve the k-connectivity with probability greater than 90%, when the original network is k-connected.

4.3 False Positive and Negative Ratio

As we described in Section 3.4, though the individual threshold helps nodes to reach a trade-off between system resilience and individual connectivity, it is possible that a cooperative node u cannot build up its neighbor set if $c(u) < c^*(v)\ \forall v \in Adj(u)$. In this case, we say u is a *false positive*. On the contrary, a misbehaving node u may have the chance to be added to another node v's neighbor set if v cannot have enough neighbors without adding u. In this case, we say u is a *false negative*. Since the perfect resilient overlay (PRO) is an overlay that contains *all* and *only* cooperative nodes of the original network, we can use two metrics, *false positive rate (FPR)* and *false negative rate (FNR)*, to evaluate the effectiveness of the PROACtive protocol in generating PROs. If let N_C and N_M denote the number of cooperative and misbehaving nodes in the original network, then the FPR and FNR can be calculated by $FPR = N_C^m/N_C$ and $FNR = N_M^c/N_M$, respectively, with N_C^m and N_M^c denoting the number of false positives and false negatives.

Our simulation results are reported in Fig. 4(a) and 4(b). In Fig. 4(a), the FPRs are very low ($< 5\%$) for all networks of different system size N as well as different P_M; however, the FNRs are more significant for small N than for large N. This indicates that relatively more misbehaving nodes are added into the overlay to keep it connected when the network is sparse. Another observation is that the FNRs increase significantly when P_M increases, which is further illustrated in Fig. 4(b), where the FNR for $N = 4000$ raises even up to 50% when $P_M = 0.9$. This is due to the fact that more false negatives are produced to keep enough neighbors for every node when many misbehaving nodes present. These observations show that the PROACtive protocol is more "conservative" in satisfying the k-connectivity constraint.

5 Conclusion and Future Work

In this paper, we proposed a distributed and localized protocol, PROACtive, to generate perfect resilient overlays which contain all and only cooperative nodes of the original wireless multi-hop networks. The PROACtive protocol has a light-weighted message complexity, $O(m)$, and the overlays generated achieve k-connectivity with high probability and low false positive ratio. The main advantage of applying our PROACtive protocol is that the resilient overlays generated essentially provide cooperative platforms for multi-hop routing and data transmission when misbehaving nodes are present. Based on the resilient overlays, new routing strategies and data aggregation schemes can be designed, which will be our next works. Further, more advanced cooperativity measurement schemes are also needed to be explored.

References

1. Aad, I., Hubaux, J.P., Knightly, E.W.: Denial of Service Resilience in Ad Hoc Networks. In: Proc. of ACM MobiCom '04. (2004) 202–215
2. Xing, F., Wang, W.: Modeling and Analysis of Connectivity in Mobile Ad Hoc Networks with Misbehaving Nodes. In: Proc. of IEEE conference on communication (ICC) '06. (2006) 1879 – 1884
3. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad hoc Networks. In: Proc. of ACM MobiCom '00. (2000) 255–265
4. Zhong, S., Chen, J., Yang, Y.R.: Sprite: A Simple, Cheat-Proof, Credit-based System for Mobile Ad-Hoc Networks. In: Proc. of IEEE INFOCOM '03. (Mar. 2003) 1987–1997

5. Hu, Y., Perrig, A., Johnson, D.B.: Ariadne: A Secure OnDemand Routing Protocol for Ad Hoc Networks. In: Proc. of ACM MobiCom '02, Atlanta, USA (Sep. 2002)
6. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. Mobile Computing and Communications Review (MC2R) **5**(4) (2001) 1–13
7. Blough, D.M., Leoncini, M.: The K-Neigh Protocol for Symmetric Topology Control in Ad Hoc Networks. In: Proc. of ACM MobiHoc '03, ACM Press (2003) 141–152
8. Shen, C.C., Srisathapornphat, C., Liu, R., Huang, Z., Jaikaeo, C., Lloyd, E.L.: CLTC: A Cluster-Based Topology Control Framework for Ad Hoc Networks. IEEE Transactions on Mobile Computing **3**(1) (2004) 18–32
9. Cardei, M., Wu, J., Yang, S.: Topology Control in Ad Hoc Wireless Networks Using Cooperative Communication. IEEE Transactions on Mobile Computing **5**(6) (Jun. 2006) 711–724
10. Penrose, M.D.: On k-connectivity for a Geometric Random Graph. Random Struct. Algorithms **15**(2) (1999) 145–164
11. Bettstetter, C.: On the Minimum Node Degree and Connectivity of a Wireless Multihop Network. In: Proc. of ACM MobiHoc '02, ACM Press (Jun. 2002) 80–91
12. Li, X.Y., Wan, P.J., Wang, Y., Yi, C.W.: Fault Tolerant Deployment and Topology Control in Wireless Networks. In: Proc. of ACM MobiHoc '03. (Jan. 2003) 117–128
13. Bettstetter, C.: On the Connectivity of Ad Hoc Networks. The Computer Journal, Special Issue on Mobile and Pervasive Computing **47**(4) (2004) 432–447