Computer Networks 55 (2011) 2608–2621

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Toward robust multi-hop data forwarding in large scale wireless networks

Fei Xing [a,*], Wenye Wang [b]

[a] Wireless Networking Business Unit, Cisco Systems, San Jose, CA 95134, United States
[b] Dept. of Computer and Electric Engineering, North Carolina State University, Raleigh, NC 27606, United States

ARTICLE INFO

Article history:
Received 27 September 2010
Received in revised form 15 April 2011
Accepted 22 April 2011
Available online 14 May 2011
Responsible Editor: J.C. de Oliveira

Keywords:
Multi-hop routing
Network robustness
Wireless networks
Performance evaluation

ABSTRACT

Design of robust network topology is an essential issue in large-scale multi-hop wireless networks since data packets are forwarded through intermediate nodes between source and destination, especially in the presence of non-cooperative nodes. Traditionally, topology design aims at generating network topology with high node degree, maximum throughput, and mitigation of malicious attacks. In this paper, we formulate a novel topology control problem as achieving optimal topology which maximizes network robustness against data forwarding distortion (DFD) in which a relay node may be compliant in route discovery, but drop or delay packets as non-cooperative nodes. Such node misbehavior can degrade network performance dramatically, without being detected by routing protocols and countermeasures. Therefore, we propose to design a network topology and data forwarding algorithms, namely PROACtive, in order to distribute data packets among cooperative nodes only, subject to k-connectivity constraint. Through analysis and simulations, we show that there exists a trade-off between achieving network robustness and k-connected with high probability (w.h.p.). By using distributed measurement schemes, data packets can be forwarded with low message complexity $\Theta(N)$, and improves network goodput significantly in different network scenarios.

1. Introduction

Successful data forwarding is critical to wireless multi-hop networks and is highly dependent on route discovery and path establishment in a reliable network. However, recent literature have shown that multi-hop path selections can be hindered by various node misbehavior [1–5], which means that wireless nodes, as autonomous entities, may behave non-cooperatively in routing. Fortunately, cryptographic-based schemes can protect malicious attackers to tamper with un-compromised routes, even though they may introduce routing loops, gray holes, or redirect routs [1]. Moreover, a selfish node may refuse to forward route request or reply messages for other nodes for the sake of saving its own energy [2]. Since purely cryptographic countermeasures are not effective against selfish members which may have owned the cryptographic keys, a few of credit-based schemes, e.g., [3,4], were proposed to stimulate selfish nodes to forward routing packets for others. Nevertheless, the non-cooperative nodes may try to hide their misbehavior under protocol-compliant route discovery and establishment. For example, malicious nodes launching *Jellyfish* (or *Blackhole*) attacks may conform to all routing procedures but reorder, delay, or drop packets once they are en-route, as studied in [5], which reduce network performance as well as induce network partitions.

In this paper, we aim to design network topology that is robust against aforementioned routing-compliant misbehavior, that is, malicious nodes generate and forward all control packets (e.g., *RREQ* and *RREP* in DSR or AODV) as defined by the routing protocol but drop all or just partial data packets to be forwarded. These misbehavior will be referred to as the *data forwarding distortion (DFD)* in the following context in that they cannot deliver data successfully by distorted operation. It is worthy noting that in the presence of DFD, paths can still be established, yet they may be useless

* Corresponding author. Tel.: +1 4088535356.
E-mail addresses: fexing@cisco.com (F. Xing), wwang@ncsu.edu (W. Wang).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.04.014

for data delivery if any intermediate node en-route behaves non-cooperatively in forwarding others' packets. In addition, if a misbehaving node launches Blackhole-alike attacks it may claim it is in the shortest or optimal path to the destination so that it can be included in new routes and thus fail re-route attemps. Clearly, secure routing protocols cannot answer the challenge imposed by DFD attacks since malicious nodes are all routing-compliant and can own cryptographic keys as well. Further, the limitations of existing credit-based solutions make them difficult to find a widespread acceptance in addressing DFD attacks. For example, the scheme proposed in [3] assumes a special temper-proof hardware, which could be costly for some terminal mobile devices; and the centralized security authority used in [4] can increase the complexity in the deployment and management of wireless networks.

Intuitively, we can mitigate the impact of DFD by preventing non-cooperative nodes from participating in routing based on reputation systems [2,6,7], where a system analyzes local and third-party data and rates each node. Nevertheless, quite a few challenges remain unsolved. For example, how to differentiate throughput decrease due to non-cooperative nodes from those resulting from network congestion and heavy traffic load [5], and how to quantify the cooperativity of individual nodes, as well as determining detection threshold for non-cooperative nodes. More importantly, network connectivity has been largely overlooked in previous reputation schemes, which is, however, a critical factor to sustain network performance. Therefore, the DFD problem remains an open and challenging problem for the design of robust wireless multi-hop networks. Motivated by the challenges aforementioned, we approach the DFD problem from the perspective of constructing a routing topology. More specifically, we strive to design a protocol-compliant misbehavior robust network, such that network operations on both control and data planes are distributed only between cooperative nodes and the network is *k*-connected[1] with high probability (w.h.p.) (e.g., >0.9).

Therefore, we first define a new metric called *robust space* to measure the maximum number of non-cooperative nodes that a network can sustain under the constraint that the network is *k*-connected with a certain probability. In other words, the problems we are solving are (1) given a random geometric graph with *N* nodes, how many non-cooperative nodes can be chosen while still providing *k*-connectivity with a given probability? (2) given the robustness space of a network, what are the conditions for a network topology such that *k*-connectivity can be satisfied under the limit of robust space? (3) given a network, how to design a network protocol that can discern non-cooperative nodes *and* maximize robust space such that a network is robust against non-cooperative nodes with DFD. As a result, the union of all cooperative neighbor sets forms a network topology which satisfies the connectivity requirement and maximizes the robust space. With the generated topology, the routing protocol can prevent non-cooperative nodes from participating in data relays by distributing control packets only among cooperative neighborhoods. Finally, our simulation results validate that after applying routing protocols (e.g., AODV) on the topology generated by the PROACtive protocol, the network goodput can be improved significantly in different network scenarios.

The rest of the paper is organized as follows. In Section 2, we describe system models and definitions. In Section 3, we present theoretical upper bound of number of non-cooperative nodes in a given network and identify conditions for robust network topology. In Section 4, we introduce the node cooperativity measurement scheme, which is a preceding step in the design of our protocol. In Section 5, we provide the design details of the PROACtive protocol. In Section 6, we evaluate the efficiency of our solution by simulations and compare with other solutions. Finally, conclusions are given in Section 7.

2. Network model and definitions

In this section, we present the network model, concept of robust space, and describe our design goal.

2.1. Network model

In this paper, we consider a wireless multi-hop network of *N* mobile nodes with the same transmission radius *r*. Let $\mathcal{N} = \{X_1, X_2, \ldots, X_N\}$ be independently and identically distributed (i.i.d.) random variables with uniform distribution over a 2-dimensional square with area *A*, where X_i $(1 \leqslant i \leqslant N)$ denotes the random location of node *i*. Then we can model wireless multi-hop networks by random geometric graphs $G_{\mathcal{N},r}$ [9], where $\mathcal{N}$ denotes the vertex (node) set with $|\mathcal{N}| = N$ and an edge exists between vertexes (nodes) *i* and *j* only if their Euclidean distance is no greater than *r*.

We consider in a network in which every node may be cooperative with probability p_C or non-cooperative with probability p_N at any time, which implies that $p_C + p_N = 1$. Cooperative nodes not only comply all routing rules but also relay all data packets for others at the best effort; while non-cooperative nodes launch the DFD attacks, i.e., they randomly drop data packets to be forwarded, though they are compliant to all rules in the route discovery. Further, we assume that the networking behavior of every node has no correlation with each other; in other words, the collusion of non-cooperative nodes is not considered in this paper. Note that this assumption has been implicitly used in other papers on reputation mechanisms as well [2,6,7]. With above assumptions, we can apply the original random geometric graph model with a Bernoulli node model in which an arbitrary node is non-cooperative with probability p_N and cooperative with probability $p_C = 1 - P_N$. Due to the ergodicity of our model, we know that at a given time, the number of non-cooperative nodes, denoted by N_n, can be approximated by $p_N \cdot N$ statistically as *N* is sufficiently large.

It is noticed that few mechanisms have been proposed to cope with the colluding misbehavior at the *MAC* layer (e.g., [10,11]; nevertheless, little study has been done to tackle the colluding DFD attacks on the *network* layer. Although we assume the independence of the behaviors of individual nodes to make our analysis tractable, we will

[1] The rigorous definition of *k*-connectivity is given in Section 2.2. Readers can also refer to [8](Chapter III.2 page 73) for details.

discuss the potential of our PROACtive protocol to mitigate the impact of colluding DFD attacks later.

2.2. Definitions

Non-cooperative nodes with DFD can hinder the communications between cooperative nodes even if the network is physically connected. Nevertheless, simply excluding non-cooperative with no consideration of network connectivity, as proposed by current reputation-based solutions, may even result in network partitions which can further degrade network performance. Therefore, we intend to find a network subject to k-connected topology which may be preferable for more reliable communications.

Let Ω be the sample space consisting of all the possible topologies G of a network. From graph theory, we know that a graph G is k-connected if it has at least $k+2$ vertices and no set of $k-1$ vertices separates it. Further, the connectivity of a graph G, denoted by $\kappa(G)$, is the maximal value of k such that G is k-connected [8](Chapter III.2 page 73). Note that the connectivity is usually referred to as vertex-connectivity, as used in this paper. Considering a multi-hop wireless is a dynamic system because wireless devices may be powered on and off, as well as node mobility, the connectivity of such a network can be treated as a random variable defined on Ω and the probabilistic k-connectivity of a network can be defined by $Pr(\{G \in \Omega : \kappa(G) = k\})$, or simply $Pr(\kappa(G) = k)$, for G as the geometric random graph model of the network. With above notations, we define the robust space as below.

Definition 1. Given a wireless multi-hop network, let its topology be represented by a random geometric graph $G_{\mathcal{N},r}$. For a connectivity preference $0 < \psi_0 < 1$, the *robust space*, which is the maximum number of non-cooperative nodes can be accommodated in $G_{\mathcal{N},r}$ such that the probability of $G_{\mathcal{N},r}$ being k-connected is no less than ψ_0, defined as follows

$$\Lambda(\psi_0, G_{\mathcal{N},r}) \triangleq \max\{N_n : Pr(\kappa(G_{\mathcal{N},r}) = k) \geqslant \psi_0\}, \quad (1)$$

where N_n denotes the number of non-cooperative nodes. In the succeeding sections, we also use N_n^* to substitute $\Lambda(\psi_0, G_{\mathcal{N},r})$ to make statements or calculations concise.

To achieve robust data forwarding in a multi-hop wireless network, our objective is to find an optimal network topology over which data can be forwarded successfully by minimizing the impact of DFD induced by non-cooperative nodes.

Definition 2. Given a physically k-connected wireless multi-hop network represented by $G_{\mathcal{N},r}$, design a topology G' such that as many as non-cooperative nodes can be excluded from G' subject to a probabilistic k-connectivity $0 < \psi_0 \to 1$. More specifically, the G' should fulfill the following requirements:

$$Pr(\kappa(G') = k) \geqslant \psi_0 \quad \text{and } G' = \underset{\text{all } G^-}{\operatorname{argmax}} \Lambda(\psi_0, G^-), \quad (2)$$

where G^- is any *subgraph* of $G_{\mathcal{N},r}$ in which the vertex set is a subset of $\mathcal{N}$.

Our approach takes three steps. First, we need to find the sufficient or necessary condition for asymptotic k-connectivity so that the topology can satisfy the connectivity requirement. Moreover, we need to know the theoretical bound on robust space in order to determine how many non-cooperative nodes can be excluded from the topology under a connectivity constraint. Second, we need to measure node behavior dynamics quantitatively such that we can differentiate non-cooperative nodes from cooperative ones. Third, we need to design a distributed and localized protocol which results in an optimal topology to be robust against DFD. In the next section, we conduct a theoretical analysis to complete the first step aforementioned.

3. Toward robust network topology with k-connectivity

In this section, we focus on two issues. First, given a random geometric graph with N nodes, how many non-cooperative nodes can be chosen while providing k-connectivity w.h.p.? Second, given the robustness space of a network, what are the properties or conditions for a network topology such that k-connectivity can be satisfied under the limit of robust space?

3.1. Probabilistic k-connectivity

The connectivity issue has been studied extensively recently [12–18]. For example, a critical transmission radius r_c is provided in [17] for k-connectivity as $\lambda\pi r_c^2 = \log N + (2k-3)\log\log N + f(N)$, where λ is the node density and $f(N)$ is an increasing function in N ($\lim_{N\to\infty} f(N) = +\infty$). This result provides a sufficient and necessary condition for a network to be k-connected and is very useful in topology design. Nevertheless, it is not applicable in real implementations since the global information, such as the network size N and node density λ, is normally unknown to individual nodes in a distributed system like large-scale wireless multi-hop network.

Alternatively, a powerful result on the probabilistic k-connectivity was proved by Penrose [12] that a geometric random graph G becomes k-connected when its minimum vertex degree, denoted by $\delta(G)$, becomes k w.h.p. as N goes to infinity, for any positive integer k [19, Theorem 6.1.2]. In particular, it is shown that.

Lemma 1 [13, Theorem 3]. *For a graph $G_{\mathcal{N},r}$,*

$$Pr(\kappa(G_{\mathcal{N},r}) = k) \approx Pr(\delta(G_{\mathcal{N},r}) \geqslant k), \quad (3)$$

when $N \gg 1$ and $Pr(\delta(G_{\mathcal{N},r}) \geqslant k) \to 1$.

This result has been verified by extensive simulations in [13,14,16], and [18]. Especially, it was shown in [14] that Eq. (3) holds even if N is in the order of 50 only and $Pr(\delta(G) \geqslant k)$ is not close to one.

However, in a multi-hop network nodes may be isolated logically from the network even if they have active neighbors. In other words, whether a node can establish reliable connections to other nodes depends on whether the node has *cooperative* adjacent nodes that operate normally on both control and data planes. Let $D_c(\omega)$ be the number of cooperative adjacent nodes of node ω, called the *cooperative degree* of ω, we define $\theta(G_{\mathcal{N},r})$ (or simply $\theta(G)$) as the *minimum cooperative degree* of a graph $G_{\mathcal{N},r}$, i.e.,

$\theta(G) \triangleq \min\{D_c(\omega), \forall \omega \in G\}$. Based on the above observation and Lemma 1, we have.

Lemma 2. *For a wireless multi-hop network represented by $G_{\mathcal{N},r}$, let μ be the average number of nodes in a node's transmission range. Suppose $N \gg 1$, then for any positive integer $k \geqslant 1$,*

$$Pr(\kappa(G_{\mathcal{N},r}) = k) \approx Pr(\theta(G_{\mathcal{N},r}) \geqslant k) \geqslant 1 - N\left(\frac{\Gamma(k, \mu \cdot p_C))}{\Gamma(k)}\right), \quad (4)$$

where $\Gamma(h) = (h-1)!$ and $\Gamma(h,x) = (h-1)!e^{-x}\sum_{i=0}^{h-1} x^i/i!$ are the complete and incomplete Gamma functions, respectively.

To understand the first approximation, we consider the fact that when N is sufficiently large, the uniform node distribution of the random geometric graph $G_{\mathcal{N},r}$ can be governed by a homogeneous Poisson point process with density $\lambda = N/A$. The second condition can be obtained by applying results in [Equation (40)] [16]. The modification is that the distribution of cooperative nodes also follows a homogeneous Poisson point process. As a result, according to the *Thinning theorem* [Theorem 9.15] [9], cooperative nodes can induce a new random geometric graph with the minimum (cooperative) degree as $\theta(G)$ and node density is $\mu \cdot p_C$, where p_C is the cooperative probability for a given node.

Further, when $Pr(D_c < k) = o(1/N)$, i.e., $N \cdot Pr(D_c < k) \to 0$, we have the following approximation

$$1 - N\left(\frac{\Gamma(k, \mu \cdot p_C))}{\Gamma(k)}\right) \approx \left(1 - \frac{\Gamma(k, \mu \cdot p_C))}{\Gamma(k)}\right)^N. \quad (5)$$

When $p_C = 1$, which means all nodes are assumed to be cooperative, the right hand side of (5) devolves into the special case reported in [13, Theorem 3].

Remark 1. Lemma 2 implies that the necessary condition for a network to be k-connected is that every node should have at least k cooperative adjacent nodes. Thus, (4) provides us a useful tool to design a k-connected topology w.h.p. in a localized and distributed algorithm (presented in Section 5). In addition, from (4), we know that $Pr(\kappa(G) = k)$ is a decreasing function in $p_N = 1 - p_C$, which implies that the more non-cooperative nodes a network has, the harder for the network to keep its topology k-connected w.h.p.. Next, we move onto the analysis of maximum non-cooperative nodes.

3.2. Analysis of robust space

Recall that in Definition 1 the robust space $\Lambda(\psi_0, G)$ of a (topology) graph G is defined as the maximum number of non-cooperative nodes that G can sustain with $Pr(\kappa(G) = k) \geqslant \psi_0$ By utilizing the lower bound of $Pr(\kappa(G) = k)$ in (4), we derive N_n^* in three cases: $k = 1$, $k = 2$, and $k \geqslant 3$. The main results are shown as follows.

Case-1: $\mathbf{k} = 1$. By solving $Pr(\kappa(G) = k) \geqslant \psi_0$, we have

$$N_n^* = \left\lfloor N\left(1 - \frac{1}{\mu}\ln\left(\frac{N}{1-\psi_0}\right)\right)\right\rfloor. \quad (6)$$

Case-2: $\mathbf{k} = 2$. From $Pr(\kappa(G) = k) \geqslant \psi_0$, we have

$$-\left(1 + \mu\left(1 - \frac{N_n^*}{N}\right)\right) \cdot e^{-(1+\mu\left(1-\frac{N_n^*}{N}\right))} = -\left(\frac{1-\psi_0}{N}\right)e^{-1}. \quad (7)$$

To solve N_n^* from above equality, we refer to *Lambert $\mathcal{W}$ function* [20], which is the function satisfying $\mathcal{W}(z)e^{\mathcal{W}(z)} = z$.

The branch satisfying $-1 \leqslant \mathcal{W}(z)$ is denoted by $\mathcal{W}_0(z)$, while the branch satisfying $\mathcal{W}(z) \leqslant -1$ is denoted by $\mathcal{W}_{-1}(z)$. By using $\mathcal{W}_{-1}(z)$, we have N_n^* solved as:

$$N_n^* = \left\lfloor N\left(1 + \frac{1}{\mu}\left(\mathcal{W}_{-1}\left(-e^{-1}\left(\frac{1-\psi_0}{N}\right)\right) + 1\right)\right)\right\rfloor. \quad (8)$$

Case-3: $\mathbf{k} \geqslant \mathbf{3}$. Let $z = \frac{1-\psi_0}{N}$ and $x = \mu\left(1 - \frac{N_n^*}{N}\right)$, then we use a *transcendental equation* as

$$e^{-x}\left(1 + x + \frac{x2}{2} + \cdots + \frac{x^{k-1}}{(k-1)!}\right) = z, \quad (9)$$

which is, however, cannot be solved by trivial solutions [21]. Thus we use a heuristic algorithm, summarized in Algorithm 1, to find the approximate value of N_n^*. The heuristic algorithm can be used to find N_n^* for $k = 1$ and $k = 2$ as well.

Algorithm 1: Calculate N_n^* for $k \geqslant 3$

Input: N, μ, k, ψ_0
1: $\Psi := 1$, $N_n := 10$
2: **while** ($N_n < N$ AND $\Psi > \psi_0$) **do**
3: $p_C = \frac{N - N_n}{N}$
4: $\Psi := 1 - N\left(\frac{\Gamma(k, \mu \cdot p_C))}{\Gamma(k)}\right)$
5: $N_n := N_n + 1$
6: **end while**
7: **return** $N_n^* := N_n$

From results in (6), (8), and Algorithm 1, we find that the robust space $\Lambda(\psi_0, G)$ is a function of N, μ, k, and ψ_0. In particular, for dynamic networks, we have an interesting observation that the robust space of a network can be improved by excluding some non-cooperative nodes. More precisely, given a network modeled by $G_{\mathcal{N},r}$, let $G'_{\mathcal{N},r}$ and $G''_{\mathcal{N},r}$ be two topologies built upon G with $N'_n < N''_n$ non-cooperative nodes, respectively, then we have $\Lambda(\psi_0, G') > \Lambda(\psi_0, G'')$. Therefore, we have.

Lemma 3. *Given a network modeled by $G_{\mathcal{N},r}$, let $\mathcal{N}_c$ be the set of cooperative nodes in G with $N_c = |\mathcal{N}_c|$. Let the topology containing all cooperative nodes only be denoted by $G^-_{\mathcal{N}_c,r}$, and let $G'_{\mathcal{N}',r}$ denote any topology containing $N'_n > 0$ non-cooperative nodes and N_c cooperative nodes, that is, $|\mathcal{N}'| = N_c + N'_n \leqslant N$. Then $\Lambda(\psi_0, G^-) > \Lambda(\psi_0, G')$ holds for any $0 < \psi_0 < 1$ and $k = 1$.*

Proof. When $k = 1$, by (6) and (1), we have

$$\Lambda(\psi_0, G^-) = N_c\left(1 - \frac{1}{\mu_1}\ln\left(\frac{N_c}{1-\psi_0}\right)\right),$$

$$\Lambda(\psi_0, G') = (N_c + N'_n)\left(1 - \frac{1}{\mu_2}\ln\left(\frac{N_c + N'_n}{1-\psi_0}\right)\right) - N'_n,$$

where $\mu_1 = N_c \cdot \frac{\pi r^2}{A}$ and $\mu_2 = (N_c + N'_n) \cdot \frac{\pi r^2}{A}$ and A is the area of the network. Consequently,

$$\Lambda(\psi_0, G') - \Lambda(\psi_0, G^-) = \frac{N_c}{\mu_1}\ln\left(\frac{N_c}{1-\psi_0}\right) - \frac{N_c+N_n'}{\mu_2}\ln\left(\frac{N_c+N_n'}{1-\psi_0}\right)$$
$$= \left(\frac{A}{\pi r^2}\right)\cdot\left(\ln\left(\frac{N_c}{1-\psi_0}\right) - \ln\left(\frac{N_c+N_n'}{1-\psi_0}\right)\right)$$
$$= \left(\frac{A}{\pi r^2}\right)\ln\left(\frac{N_c}{N_c+N_{n'}}\right) < 0.$$

This completes the proof. □

While we provide rigorous proof for the case of $k = 1$, similar proof can be done by using (8) for $k = 2$. Further, the same observation can be validated for $k \geqslant 3$ by numeric simulations using Algorithm 1.

Remark 2. The result of Lemma 3 implies that the robust space can be maximized when the generated topology contains *only* and *all* cooperative nodes of the original network. This finding also provides a new insight on the trade-off between eliminating non-cooperative nodes and connectivity constraint.

3.3. Properties of robust network topology

Toward the design of robust network against DFD, we have the following observations on the essential properties of an optimal topology.

Proposition 1. *Given a network $G_{\mathcal{N},r}$, if a topology graph G^- built upon G has the following properties,*

1. *every node in G^- has at least k cooperative neighbors,*
2. *the average node degree of G^-, denoted by $\mu(G^-)$, should at least scale with logN, and*
3. *G^- containsall and only cooperative nodes of G,*

then G^- is optimal in the sense of achieving the requirements in (2).

Proof (*Sketch*). The first property follows from Lemma 2 directly. The second property is due to the fact that (6) implies $\mu > \ln(\frac{N}{1-\psi_0})$, otherwise $N_n^* < 0$ and $Pr(\kappa(G) = k) < \psi_0$ even if G does not have any misbehaving nodes. This requirement on the node degree (for asymptotic connectivity) has been revealed in the literature (e.g., [17,15,22]). Following from Lemma 3, the third property then satisfies the robust space maximization requirement. □

Therefore, we have identified the properties of an optimal topology in terms of a network being robust against DFD subject to a connectivity constraint. Consequently, it is not difficult to design such a network by the following heuristic algorithm.

Algorithm 2: Generate the optimal robust topology

Input: a network $G_{\mathcal{N},r}$, k, and ψ_0
Output:the optimal robust topology G^-
1: Calculate N_n^* by using (6), (8), or Algorithm 1
2: **repeat**
3: Remove a non-cooperative node; *count* + +
4: **until** (*count* $\geqslant N_n^*$)
5: Adjust r so that $\theta(G^-) > k$ and $\mu(G^-) = \log N$

Nevertheless, Algorithm 2 can hardly be implemented in a real network since there is no such a centralized entity to make transmission range assignments and compute precise robust space, and no deterministic criteria to find non-cooperative nodes. Therefore, we need to design a *distributed* and *localized* protocol such that every node can select cooperative adjacent nodes as its neighbors toward robust data forwarding. A preceding step in the design our networking protocol, is to differentiate non-cooperative nodes from cooperative ones. For clarity, we describe measurement schemes in the next section and focus on the protocol details in Section 5.

4. Cooperativity measurement scheme

From earlier discussions, we have found that an optimal robust network should exclude all non-cooperative nodes. Therefore, an essential problem is how to differentiate such nodes from cooperative nodes in a distributed manner. In this section, we propose a simple yet effective scheme which uses *promiscuous mode* and *close-loop feedback* techniques to quantify node cooperativity.

4.1. Node cooperativity definition

A common characteristic of non-cooperative operation on the *data plane*, including unsuccessful data forwarding, is dropping transient packets, no matter whether non-cooperative nodes comply with the rule on the control plane or not. This observation implies that for a node ω, we can use ω's packet drop ratio to examine whether ω is actively forwarding packets for others. Let $c(\omega)$ denote ω's cooperativity, $n_{fwd}(\omega)$ and $n_{drp}(\omega)$ denote the numbers of packets forwarded and dropped, respectively, then we can define $c(\omega)$ as:

$$c(\omega) = 1 - \frac{n_{drp}(\omega)}{n_{fwd}(\omega)}. \quad (10)$$

One may argue that this definition can induce high false positives since cooperative nodes may also have high drop ratio due to various reasons. For example, if a cooperative node has too many neighbors, the chances of collisions or interferences are usually high, which may result in unsuccessful transmissions and dropped packets. If a cooperative node is overloaded, it may not be able to respond to newly incoming packets. In addition, packet losses can be caused by node mobility, network congestion, and even high bit error rate (BER) as well. Nevertheless, we find that the cooperative definition is more meaningful in this case, since a node with high drop ratio is not helpful to others even if it intends to be "cooperative".

It is worthy of pointing out that although all nodes are assumed to be randomly, independent, with non-cooperative probability p_N, the impact of being non-cooperative may not be the same. In particular, different DFD attackers can use various dropping rates based on different scenarios, e.g., they can drop more packets when BER is high. Thus, the definition of per-node cooperativity is quite adaptive to dynamic cooperativity of individual nodes. Also, the proposed cooperativity measurement scheme

does not differentiate the causes for dropped data. It, however, is able to capture a node's data forwarding capacity resulting from a wide variety of causes, such as non-cooperative behaviors, congestion, collisions, or high BER.

4.2. Technique I: promiscuous mode

The first technique to measure the cooperativity is the *promiscuous mode*, which has been used [2,6] to detect node misbehavior. When a wireless card operates in the promiscuous mode, it is able to monitor ongoing transmissions in its neighborhood since the MAC layer can pass all received traffic to the network layer rather than just packets addressed to itself. We use an example in Fig. 1 to illustrate the basic idea. Here every time node s requests an adjacent node ω to forward a packet to another node, say t, s increases a counter $n_{fwd}(\omega)$ by 1. If s cannot overhear ω's forwarding, it increases another counter $n_{drp}(\omega)$ by *1*. By this way, a node measures its adjacent nodes by its "own experiences", similar to the *Watchdog* [2].

An extension to this solution is to let nodes measure their adjacent nodes' behaviors even if they are not en-route to forward the packets. For instance, if one of s's adjacent nodes, u, requires ω to forward packets, s can record ω's behavior as well, which is called s's "direct observations". Based on the measurements from both own experience and direct observation, s can compute $c(\omega)$ by (10). Although this technique is simple and efficient, it may induce inaccurate estimations. Here are two typical examples.

- Case-1: In Fig. 1 node s hears ω's forwarding but the data packet is not received by t successfully, which can be caused by either transmission collision at t or insufficient transmission power of ω. In this case, the calculated $c(\omega)$ is *overestimated*.
- Case-2: If s is receiving while ω is forwarding, then there may be a collision at s. In this case, even the forwarded data packet is received successfully by t, node s may not hear this forwarding due to the collision. Consequently, node s can mistakenly increase $n_{drp}(\omega)$ by 1 so that $c(\omega)$ is *underestimated*.

To improve the accuracy of cooperativity measurement, we introduce a close-loop feedback technique next.

4.3. Technique II: two-hop-away feedback

The idea of close-loop feedback is using a *two-hop-away* ACK message as the feedback to indicate the success of forwarding. As illustrated in Fig. 2, when s sends a packet to ω, it piggybacks a feedback request in the packet to ask any downstream node of ω to send back an ACK. If s receives the ACK before a timeout, s considers ω having successfully forwarded the packet; otherwise, ω fails to do so. The technique does not necessary require an end-to-end

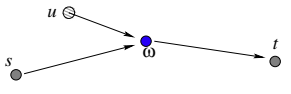


Fig. 1. Technique I: promiscuous mode.

Fig. 2. Technique II: two-hop ACKs.

feedback; instead, every node en-route can use the method to monitor the behavior of its next hop as long as the ACK is from *at least* two-hop away.

Note that when the aforementioned promiscuous mode is used simultaneously, node s can measure ω's behavior by listening to all feedback requests and ACKs passing through ω, which further increases the measurement accuracy. Nevertheless, it is worthy of pointing out that the accuracy brought by this technique is at the cost of extra communication overhead due to ACK messages. For example, let H be the number of nodes en-route that request two-hop away ACKs, q_{ack} be the frequency of requesting feedback ($0 \leqslant q_{ack} \leqslant 1$), then there will be $H \cdot q_{ack}$ ACKs to be sent in average for every (end-to-end) data packet delivery. To reduce this overhead, the feedback frequency q_{ack} can be set to a small value, e.g., 10%, so that the number of ACKs ($H \cdot q_{ack}$) is no much greater than the number of data packets delivered. When $H \cdot q_{ack} \approx 1$, then the communication overhead induced by sending ACKs will be merely similar to that for TCP connections. Another advantage of using a low feedback frequency q_{ack} is that it may help to avoid potential false accusations caused by ACKs with *TTL* = 1. For example, if s relies on two-hop ACKs only and u sends ACKs with *TTL* = 1, as shown in Fig. 2, then ω will be falsely accused by its legitimate operation of stopping to forward the ACKs. An evaluation on the communication overhead will be reported in Section 6.

4.4. Discussion

There are several possible factors that may be resulted from either *promiscuous mode* and *two-hop-away feedback* techniques.

- Inconsistent measurements: Note that different nodes may obtain different measurements to a same node because the traffic passing through the node may not be overheard by all its neighbors and measurements are not shared between each other in our scheme. Nevertheless, the effect of this discrepancy is constrained locally and does not affect cooperativity calculations and neighbor selections for the entire network.
- Asymmetric measurements: It is also very likely that a node ω finds one neighboring node ω' non-cooperative, while ω' considers ω cooperative based on their measurements. In this case, random geometric graph may be *asymmetric*. However, the ultimate goal of each node is to find its own cooperative neighbors to successfully forwarding packets. Therefore, such asymmetric links cannot affect the selection of forwarding nodes.
- Colluding attacks: The promiscuous method collects information only from a node's own experiences and direct observations, which is different from the *Watchdog* and *ALARM* [2,6]. Thus, although our method still has some inherited drawbacks listed above, it is robust to

colluding DFD attackers in the sense that the collusion of multiple DFD attackers will not impact the cooperativity measurement. However, the two-hop feedback method is not robust to the collusion of multiple DFD attackers, though it provides better accuracy. For example, if both ω and u are misbehaving nodes in collusion, u may generate faked ACKs for ω to send back to s. In this case, we can extend our two-hop feedback to a more robust *random-hop* feedback scheme, in which a feedback is requested from a random-hop away downstream node and the hop number is randomly chosen by the sender. This randomness can mitigate the impact of colluding misbehavior; while the treatment to more sophisticated colluding attacks is out of the scope of this paper.

5. PROACtive protocol design

To achieve robust data forwarding, we need to determine the maximum number of non-cooperative nodes can be sustained in a network (Section 3), to identify non-cooperative nodes (Section 4), and to design a network protocol for implementation. Here we present a new protocol called *PROACtive* to enhance the network robustness against protocol-compliant attacks, such as DFD. The basic idea of the PROACtive protocol is to allow every node to select adjacent nodes with high cooperativity as their *cooperative neighbors* [23]. Moreover, to satisfy the connectivity constraint, every node should select at least k neighbors; while to maximize the robust space, every node should exclude non-cooperative nodes from its neighbor set. As a result, the union of cooperative neighbor sets will generate a robust topology.

Before presenting the details, it is worthy noting implementation requirements for our design: (a) the protocol should be fully *distributed* and the topology formation should use every node's *local* information only; (b) the protocol should *preserve the connectivity* of generated topology w.h.p. ($\geqslant$0.9) if the underlying network is physically k-connected; (c) the protocol should avoid exclusion of cooperative nodes, called *false positive*, and inclusion of non-cooperative, called *false negative*; (d) the protocol should be light-weight in the computation and communication complexity and have a bounded convergence time; (e) the protocol should be *interoperable with routing protocols* for a graceful performance degradation.

Now we are elaborate how cooperative neighbors are selected by using both unicast algorithm (*PRO-UNI*) and broadcast algorithm (*PRO-BRO*).

5.1. Neighbor selection by unicast: PRO-UNI

In the *PRO-UNI* algorithm, neighbor selections are completed by exchanging the *Neighbor Request* (Ngbr-Rqst) and *Neighbor Reply* (Ngbr-Rply) messages. nodes. Let $Adj(\omega)$ and $Ngbr(\omega)$ denote the sets of adjacent nodes and cooperative neighbors of node ω, respectively. $Adj(\omega)$ can be obtained by exchanging similar *HELLO* messages defined in [24]; while $Ngbr(\omega)$ is empty initially and constructed during neighbor selections.

At first, node ω measures its adjacent nodes' cooperativity, and selects the node with the highest cooperativity, say ω', from $Adj(\omega)$ as a potential neighbor. Then ω sends a Ngbr-Rqst to ω', telling that it intends to add ω' to its neighbor set. If ω receives a Ngbr-Rply from ω', then ω adds ω' into $Ngbr(\omega)$; otherwise, ω queries another adjacent node with the next highest cooperativity. Node ω will continue inquiries until it receives k Ngbr-Rplys, which guarantees ω with at least k neighbors. Algorithm 3 summarizes the procedure.

Algorithm 3: Procedure of querying potential neighbors

Input: k, node ω, and $Adj(\omega)$
Output: $Ngbr(\omega)$
1: Initiate $Ngbr(\omega) := \emptyset$,
create a temp set $Tmp(\omega) := \emptyset$, create a counter $nRplyRcvd := 0$
2: $\forall \omega' \in Adj(\omega)$, Measure $c(\omega')$
3: **while** ($nRplyRcvd < k$ AND $Tmp(\omega) \neq Adj(\omega)$) **do**
4: If $c(\omega') = \max\{c(u) : u \in Adj(\omega) - Tmp(\omega)\}$
5: Send Ngbr-Rqst to ω'
6: $Tmp(\omega) := Tmp(\omega) + \omega'$
7: **if** (Receive Ngbr-Rply from ω') **then**
8: $Ngbr(\omega) := Ngbr(\omega) + \omega'$
9: $nRplyRcvd := nRplyRcvd + 1$
10: **end if**
11: **end while**

Next we discuss how a node processes incoming neighbor requests. In our protocol, each node calculates a cooperative preference, based on its local information to select neighbors. This preference is called *neighbor cooperativity threshold* and denoted by $c^*(\omega)$ for node ω. When ω receives a Ngbr-Rqst from ω', it compares $c(\omega')$ to its threshold $c^*(\omega)$. If $c(\omega') \geqslant c^*(\omega)$, ω replies ω' a Ngbr-Rply and adds ω' into $Ngbr(\omega)$; otherwise, ω discards this request and replies nothing. Note that $c^*(\omega)$ can be calculated by various means, among which a simple but effective way is to use the average of all adjacent nodes' cooperativity, as used in our simulation evaluations. Algorithm 4 summaries the procedure of processing incoming neighbor requests.

Algorithm 4: Generate the optimal robust topology

Input: node ω, and $Adj(\omega)$
Output: $Ngbr(\omega)$
1: $\forall \omega' \in Adj(\omega)$, Measure $c(\omega')$
2: **if** (Receive Ngbr-Rqst from $\omega' \in Adj(\omega)$) **then**
3: **if** ($c(\omega') \geqslant c^*(\omega)$ AND $\omega' \notin Ngbr(\omega)$)
4: Send Ngbr-Rply to ω'
5: $Ngbr(\omega) := Ngbr(\omega) + v$
6: **else**
7: Discard Ngbr-Rqst
8: **end if**
9: **end if**

One benefit of using this method to calculate the neighbor cooperativity threshold is that the average node degree of the generated topology is guaranteed to be on the order of log$N$ if the average (physical) node degree of the original network scales with log$N$. Recall that this feature can

achieve the second property of the optimal robust topology given in Proposition 1. Notice that the *PRO-UNI* algorithm may induce tremendous communication overheads, we continue to present the *PRO-BRO* algorithm which exploits the broadcasting technique to minimize the potential overhead.

5.2. Neighbor selection by broadcast: PRO-BRO

The cooperative neighbor selection can also be completed by a two-phase process in which each node broadcasts a neighbor request to multiple adjacent nodes and makes the decision based on received responses. In the first phase, each node broadcasts a neighbor candidate list containing k adjacent nodes with the highest cooperativity; in the second phase, based on received neighbor candidate lists, every node decides and publicizes its potential neighbor set. We use two new messages, named as the *Neighbor-Solicitation* (Ngbr-Sol) and *Neighbor-Advertisement* (Ngbr-Adv), to deliver the candidate list and potential neighbor set, respectively.

Assume that node ω finishes measuring cooperativity at time t_0 and obtains the neighbor candidate list at time $t_\omega^{(1)} \in [t_0, \Delta t]$, where Δt is considered to be on the order of milliseconds, which is the time to transmit one packet. For example, a conservative estimate of the average one hop traversal time for packet is set to 40 ms in [24]. Thus, we take $\Delta t = 0.1$ s in this protocol.

Then ω broadcasts a Ngbr-Sol message at time $t_\omega^{(2)} \in [t_\omega^{(1)} + \Delta t, t_\omega^{(1)} + \Delta t + \tau]$. Here the parameter τ is used to avoid contentions if multiple nodes accidentally broadcast at the same time, which is subject to an arbitrary probabilistic guarantee of no contention and is reasonable to be in the order of tenth of seconds for most topology scenarios [22]. The broadcast of Ngbr-Sol is intentionally delayed by Δt such that the first Ngbr-Sol is not sent until all nodes make decisions on their candidate lists. There are two reasons for this setting: first, a node has not found the candidate list should not process incoming Ngbr-Sol messages, which simplifies the implementation of the protocol; second, the candidate list is only dependent on individual nodes, which makes our approach unbiased to every node. This heuristic approach is presented in Algorithm 5.

Algorithm 5: Procedure of neighbor selection by broadcast

Input: node ω, and $Adj(\omega)$
Output: $Ngbr(\omega)$

1: $t = t_0$: node ω completes to measure cooperativity.

2: $t = t_\omega^{(1)} \in [t_0, t_0 + \Delta t]$: node ω selects k adjacent nodes with the highest cooperativity as *(neighbor) candidates* and records them, in the order of non-increasing cooperativity, into a list $\mathcal{L}_C(\omega)$.

3: $t = t_\omega^{(2)} \in [t_\omega^{(1)} + \Delta t, t_\omega^{(1)} + \Delta t + \tau]$: $\mathcal{L}_C(\omega)$ is broadcasted in a Ngbr-Sol message at time $t_\omega^{(2)}$.

4: $t = t_\omega^{(3)} = t_\omega^{(1)} + 2\Delta t + \tau$: node ω has received all candidate lists from its adjacent nodes. For a pair of nodes ω and ω', if $\omega \in \mathcal{L}_C(\omega')$ and $\omega' \in \mathcal{L}_C(\omega)$, which means both ω and ω' consider each other "cooperative" enough to be a neighbor, node ω adds ω' into its neighbor set $Ngbr(\omega)$ and vice versa. If $\omega \in \mathcal{L}_C(\omega')$ but $\omega' \notin \mathcal{L}_C(\omega)$, which means ω' considers ω as one of its neighbor candidates, then ω adds ω' into $Ngbr(\omega)$ if $c(\omega') > c^*(\omega)$ and does nothing otherwise. If $\omega' \in \mathcal{L}_C(\omega)$ but $\omega \notin \mathcal{L}_C(\omega')$, ω adds ω' in $Ngbr(\omega)$ *temporarily* since ω does not know whether ω' has accepted its solicitation or not.

5: $t = t_\omega^{(4)} \in [t_\omega^{(1)} + 3\Delta t + \tau, t_\omega^{(1)} + 3\Delta t + 2\tau]$: node ω updates its neighbor set in a Ngbr-Adv message and schedules to broadcast at random time $t_\omega^{(4)}$.

6: $t = t_\omega^{(5)} = t_\omega^{(1)} + 4\Delta t + 2\tau$: node ω has received all neighbor lists from its adjacent nodes. For any adjacent node ω', if $\omega' \in Ngbr(\omega)$ but $\omega \notin Ngbr(\omega')$, which means ω' does not accept ω''s solicitation in Step-4, then ω must delete ω' from $Ngbr(\omega)$. $Ngbr(\omega)$ is finalized.

5.3. Complexity and convergence

We first analyze the computation complexity of our protocol. In Algorithm 3, for node ω, the first two steps take at most $O(D)$ in time, where $D = |Adj(\omega)|$. Next, it takes at most $O(D)$ in time to select one neighbor candidate from $Adj(\omega)$. Since ω has to query at least k adjacent nodes and in the worst case all adjacent nodes need to be queried, the computation complexity is $O(D^2)$. In Algorithm 4, since ω needs to process at most D *Ngbr-Rqst*s, the computation complexity is $O(D)$. Thus, the overall computation complexity of the *PRO-UNI* algorithm is just $O(D^2)$.

In Algorithm 5, the second step is to select k "most cooperative" adjacent nodes to prepare a candidate list. A simple way to implement this operation is to sort $Adj(\omega)$ in the non-increasing order by the cooperativity of adjacent nodes, then select the first k nodes, which can be done in $O(D\log D)$. Other steps take at most $O(D)$ time in each operation. Thus, the computation complexity of *PRO-BRO* is $O(D\log D)$.

Compared with data processing, wireless nodes spend more energy in data communications. For the *PRO-UNI* algorithm, we know clearly that in the worst case, every node should send either a Ngbr-Rqst or Ngbr-Rply to each of its adjacent nodes in order to build up a neighbor set. Thus, the communication complexity of *PRO-UNI* is $O(N \cdot D)$. networks. For the *PRO-BRO* algorithm, to build up neighbor sets, every node sends exactly two messages, Ngbr-Sol and Ngbr-Adv. Thus, the communication complexity of *PRO-BRO* is only $\Theta(N)$, which is a significant reduce on the overhead.

Further, we show that the convergence time of Algorithm 5, denoted by T_{con}, is bounded. Here the *convergence time* is defined as the time between the first Ngbr-Sol sent and the last Ngbr-Adv received. In the last step, node ω should have

received all Ngbr-Advs by time $t_\omega^{(5)} = t_\omega^{(1)} + 4 \cdot \Delta t + 2\tau$, where $t_\omega^{(1)}$ is the time when ω sends the Ngbr-Sol. Since $0 \leqslant t_\omega^{(1)} \leqslant \Delta t$, the time when the last Ngbr-Adv is received is bounded by $T_{con} = 5 \cdot \Delta t + 2\tau$. With the default values for τ and Δt aforementioned, T_{con} is in the order of seconds, which is reasonable for most of network scenarios, compared to dynamics of network topology.

5.4. Discussion on PROACtive protocol

Now we discuss several features of the proposed solution regarding topology update, integration with routing protocols, and dynamic threshold in identifying non-cooperative nodes in countering DFD problem.

5.4.1. Topology update

Multi-hop wireless networks are more likely to be considered time-varying systems because network dynamics, such as changes in *link connection* and and *node behaviors*. In particular, a cooperative node may become selfish to save its energy or malicious after being compromised. Or a node is excluded from network topology because of low cooperativity at certain time, but it may participate in network operations again upon the cooperativity improvement. Usually, there are two commonly used methods for updating: *on-demand update* and *periodical update* [25]. The PROACtive protocol can adapt and combine both of update methods. When the periodical update is used, Algorithm 5 can be performed periodically with an update interval which can be either as long as several minutes or as short as several seconds, depending on specific applications. When network dynamics change frequently, either in link connection or node behavior, on-demand updates are required and they can be accomplished conveniently by using the unicast method described in Section 5.1. To use PRO-BRO method, minor modifications are needed. For instance, a node can broadcast a Ngbr-Sol with an updated candidate list if necessary, and all receivers simply reply Ngbr-Advs directly, with similar steps in Algorithm 5.

5.4.2. Integration with routing protocols

The reliable data delivery is achievable with the assistance of the routing protocol because control packets are only dispersed among the member nodes in the generated topology so that misbehaving nodes cannot involve in any route. For instance, when AODV is used, a source s initiates a route discovery to a destination t by sending *RREQ* packets to its neighbors only. When an intermediate node, say ω, receives a *RREQ* from s or another node ω', it checks whether the sender is in its neighbor set. If so, ω forwards the *RREQ* to its neighbors (when ω has no known route to t); otherwise, ω discards the *RREQ* packet. When *RREQ* packets arrive at t, t sends a *RREP* packet back to the sender. Since a backward path is generated without potential misbehaving nodes en-route, the data delivery between the source and destination can be accomplished via cooperative relays with the minimum impact of non-cooperative nodes.

5.4.3. Dynamic cooperativity threshold

Each node uses an individual dynamic cooperativity threshold, which allows each node to reach a trade-off between *network* resilience and *individual* connectivity, compared to the case using a global static threshold. For example, if a cooperative node can not find enough neighbors when surrounding nodes have relatively low cooperativity, the node can tune down its cooperativity threshold so that it may accept more neighbor requests.

5.4.4. False accusation avoidance

Our approach does not involve new security vulnerabilities and can avoid the false accusation problem because the cooperativity information measured by one node is not shared with others in our protocol and the neighbor selection is only dependent upon 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 others, which prevents any node from the false accusation.

Note that the cooperativity measurement scheme introduced in Section 4 should not be the only choice for our PROACtive protocol to make neighbor selections. In fact, any similar scheme can be used as long as non-cooperative nodes that launch DFD attacks can be determined. For example when dynamic power control technologies are available [26], a node may increase its transmission power so that it can find enough cooperative neighbors to maintain k-connectivity requirements. Therefore, the PROACtive protocol is fully distributed and localized scheme with low overhead, which makes our approach feasible to be implemented in a large-scale networks to achieve robust data forwarding.

6. Simulation evaluations

6.1. Simulation setup

To evaluate the performance of our solution as well as compare our design with other solutions, we implement the PROACtive protocol in the simulation tool *ns2* and make three modifications to the existing AODV module. First, the promiscuous mode is supported such that every node can measure others' cooperativity; second, *RREQ* and *RREP* messages are distributed only among cooperative neighbors such that path selections are controlled within the topology generated; third, the DFD attack is introduced by configuring nodes to drop data packets to be forwarded randomly.

The number of nodes is varied from 100 to 900 with the interval of 200, which makes our scenarios representative for both small and large scale networks. The default transmission radius is 100 m, which is suitable for both indoor and outdoor communications and supported by most off-the-shelf devices. The mobility model is the *Semi-Markov Smooth (SMS) model* [27], which provides the uniform node distribution and more realistic movement patterns. The default speed is uniformly distributed between 0 and 10 m/s and average pause time is uniformly distributed between 0 and 2 s to offer highly dynamic networks. Constant bit rate (CBR) is chosen for traffic and the sending

rate is set as 1 packet per second. In simulation, 100 sessions are constantly maintained to keep all nodes involved in networking operations. To investigate the performance of our solution against the DFD problem, we vary the percentage of non-cooperative nodes p_N, which can be considered a limiting probability of p_N for each individual node in large-scale networks. For better understanding its impact, it is also called *non-cooperative ratio* thereafter, from 0 to 80% with the interval of 10% for all simulations. At last, the results are averaged over multiple simulation rounds conducted with various random seeds. Table 1 summarizes the default simulation parameters.

The following three aspects are investigated: (i) *performance of PROACtive*, (ii) *network performance improvement*, and (iii) *effects of network dynamics*. Further, the performances of both *PRO-UNI* and *PRO-BRO* algorithms are evaluated and compared. Note that simulations are also conducted on the networks using the original AODV module to provide the baseline for comparison.

6.2. PROACtive performance evaluation

First, we evaluate the performance of PROACtive by examining the *k*-connectivity, false positive (negative) ratio, and communication overhead, as well as in comparison with another topology control scheme, namely $K-Neigh$ [22].

Before discussing simulation results, we illustrate a picture of topology generated. Fig. 3(a) shows a network without applying any topology control. Fig. 3(b) shows the topology generated by the PROACtive protocol, in which cooperative and non-cooperative nodes are represented by solid dots and circles, respectively. From the figure, we can see that the topology excludes most of non-cooperative nodes, while keeping most of cooperative nodes connected. To highlight the difference from other topology control protocols, the topology generated by the *K-Neigh* protocol (Phase 1 only, with $K = 9$) [22] is shown in Fig. 3(c). It is no wonder that all non-cooperative nodes are included in Fig. 3(c) because the neighbor selection in K-Neigh is only based on the distance between nodes.

6.2.1. Preservation of k-Connectivity

We use the DFS (depth-first-search) algorithm to calculate the maximum number of nodes that can be removed without partitioning the network. Then the probabilistic *k*-connectivity is calculated by the ratio between the number of *k*-connected topologies and that of all topologies randomly generated. In Fig. 4(a), we can see that the *k*-connectivity probabilities of generated topologies are beyond 0.9 when $N > 700$ for $p_N = 10\%$ and $p_N = 40\%$. Nevertheless, when $N < 500$, the *k*-connectivity probabilities for generated topologies and original networks decrease dramatically. Further, we observe that the *k*-connectivity can hardly be preserved if p_N is too high, e.g., $p_N > 30\%$, as shown in Fig. 4(b). Also, we see that the average (node) degree is reduced considerably in the generated topologies, as shown in Fig. 4(c), and it decreases slightly in the non-cooperative ratio p_N because there exists less chance to find enough cooperative neighbors. Recall that in Section 3.3 we pointed out that $\mu = \Theta(\log N)$ is a condition for connectivity. From Fig. 4(c), we can see that the average degree of generated topologies is asymptotically greater than log$N$ (given the original networks *k*-connected), which satisfies the second condition of the connectivity constraint and also implies the neighbor cooperativity threshold (See Section 5.1) is reasonable and effective.

6.2.2. False positive and negative ratio

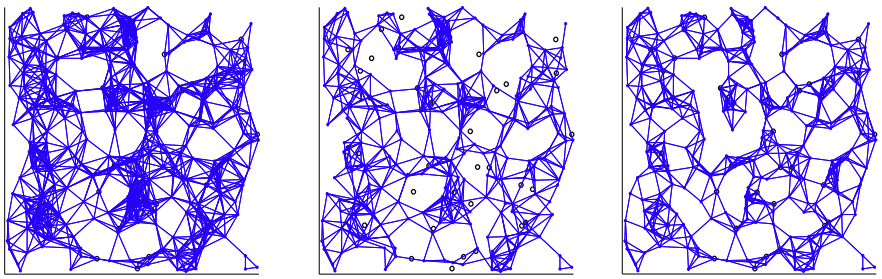
We use two metrics, *false positive ratio (FPR)* and *false negative ratio (FNR)*, to evaluate the effectiveness of the PROACtive protocol in classifying nodes. The simulation results show that the FPR is less than 5% for different network scales of N and non-cooperative ratio p_N, which indicates most of cooperative nodes can be included in the robust topologies after mutual neighbor selections; while the FNR is more significant than the FPR. For clarity, we only depict the FPR and FNR for $p_N = 20\%$ and $p_N = 40\%$ in Fig. 5(a) and observe that for fixed p_N both the FPR and FNR decrease as N increases. The main reason is that cooperative nodes have more choices in the neighbor selection when more adjacent nodes are available; while less non-cooperative nodes are added for maintaining connectivity, when the network is getting denser. Nevertheless, the FNR is quite significant when p_N is high as shown in Fig. 5(b). For example, for $N = 100$, the FNR raises even up to 47% as $p_N = 70\%$, which is due to the fact that more false negatives are produced to keep every node enough "neighbors". The high FNR can be explained by the plots shown in Fig. 5(c) as well, where the number of excluded non-cooperative nodes is shown to have a *sublinear* growth in the non-cooperative ratio p_N, especially when node density is comparatively small (e.g., $N = 500$). These observations are consistent with our analysis in Section 3 that there exists a trade-off between excluding non-cooperative nodes and maintaining network connectivity.

6.2.3. Communication overhead

In Fig. 6(a), we depict the numbers of PROACtive packets of both *PRO-UNI* and *PRO-BRO* algorithms, which are exchanged during neighbor selection with regard to network size N. As analyzed in Section 5.3, for *PRO-UNI* the communication overhead is significantly higher than that for *PRO-BRO*, especially when N is large. Moreover, when more non-cooperative nodes are present, it generates more Ngbr-Rqst and Ngbr-Rply messages for *PRO-UNI* to generate neighbor

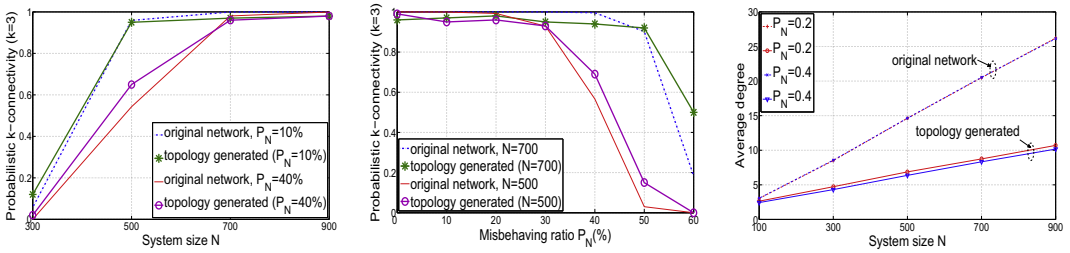
Table 1
Parameters for simulation.

Parameter	Setting
Simulation area	1000 m × 1000 m (default)
System size	500 (100,300,700,900)
Transmission range	100 m
Mobility model	SMS model (uniform placement)
Movement feature	Avg. speed 5 m/s, Pause time 1 s
Propagation	two-ray ground
MAC	IEEE802.11b DCF
Link capacity	11 Mbps (1 Mbps for broadcast)
Application	CBR (512 bytes)
Traffic load	100 connections, 1 packet per sec
Simulation time	200 s
Misbehaving ratio	[0,80%] with 10% interval



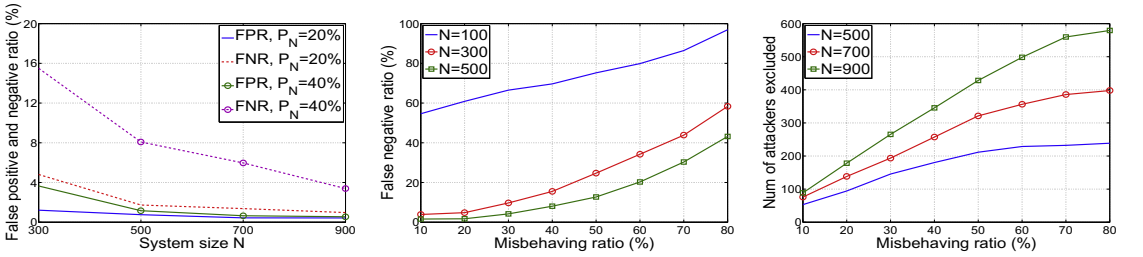
(a) No topology control (b) Topology generated by PROACTive (c) *K-Neigh* with $K = 9$ (Phase I)

Fig. 3. Topology generated in comparison with the original one (circles: non-cooperative, dots: cooperative).



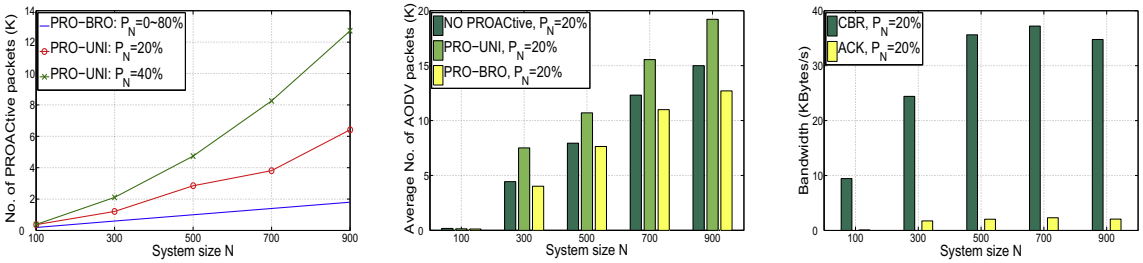
(a) k-connectivity vs. N (b) k-connectivity vs. p_N (c) Average degree vs. N

Fig. 4. The preservation of k-connectivity in the generated topology.



(a) FPR and FNR vs N (b) FNR vs p_N (c) Number of excluded non-cooperative nodes

Fig. 5. The false positive and false negative ratios in the generated topology.



(a) PROACTive message overhead (b) Routing overhead comparison (c) Feedback ACK overhead

Fig. 6. Communication complexity induced by applying the PROACTive protocol.

sets. On the contrast, the number for *PRO-BRO* approximates a linear growth on N, conforming to the analytical results.

Additionally, we investigate the overhead induced by our protocol on the AODV routing protocol, measured by the total number of *RREQ* and *RREP* messages sent per second. A surprising result is that the routing overhead can be reduced significantly when *PRO-BRO* is used, as clearly shown in Fig. 6(b). The reason behind this observation is quite simple: after a robust topology is generated, routing

traffic is carried *mainly* by cooperative nodes, which suppresses the number of routing packets relayed by non-cooperative nodes. Further, since the intermediate nodes en route are more likely to be cooperative, paths become more reliable which reduce the frequency of recovering paths and the number of *RREQ* packets in turn. However, when *PRO-UNI* is used, the routing overhead is increased considerably, especially when N is large. After a careful investigation to the current AODV module in *ns*2, we find that transmitting Ngbr-Rqst or Ngbr-Rply also needs AODV to find (one-hop) paths, which induces extra routing packets. Therefore, in terms of the communication overhead, *PRO-BRO* is more scalable than *PRO-UNI*.

The overhead induced by using the two-hop feedback technique is evaluated by using ICMP segments to implement ACKs. In the simulation, the feedback frequency q_{ack} is set to 10% and the size of ACK is 40 bytes. In Fig. 6, the average bandwidths of data and ACKs are reported, respectively, against different system sizes. Since the network of $N = 100$ is almost disconnected, the data bandwidth is significantly low and the bandwidth consumed by ACKs is almost negligible due to the high drop ratio and low average path length. As the system size increasing, we can see that the bandwidth consumed by ACKs will increase and become stable after the network is well-connected ($N \geqslant 500$). Thus, the overhead induced by ACKs is quite acceptable compared with the data bandwidth in terms of a low bandwidth consumption ratio less than 7%, which is very reasonable in most of real scenarios.

6.3. Network performance evaluation

The ultimate goal in design of robust networks is to improve network performance. Therefore, we study the DFD impact and demonstrate the improvement by using our solution with regard to *network goodput*, *data packet drop ratio*, and *average hop count*.

Network Goodput: In Fig. 7(a), the network goodput for a using pure AODV is drastically impaired as the non-cooperative ratio p_N increases. For example, the goodput drops dramatically from around 36 KBps to 10 Kbps when p_N increases from 0 to 20%. While, the goodput of the network using *PRO-BRO* remains above 34 KBps when $p_N < 20\%$ and degrades gracefully when $p_N > 20\%$. Notice when *PRO-UNI* is used only, the performance improvement is less significant. The main reason is that *PRO-UNI* induces comparatively higher volume of control packets which slacks the process of updating neighbor sets as well as suppresses the bandwidth for CBR traffic.

Data Packet Drop Ratio: The packet drop ratio against the non-cooperative ratio p_N is illustrated in Fig. 7(b), where we can see that the packet drop ratio in the network using pure AODV increases drastically up to more than 80% when $p_N = 20\%$. On the contrary, much less data packets are dropped in the topologies generated by PROACtive. For example, as shown in the figure, the data packet drop ratio keeps low as $p_N < 20\%$ and increases slowly until $p_N = 60\%$. When $p_N > 60\%$, since too many nodes are non-cooperative, even resilient topologies generated can no longer improve the performance. Thus the packet drop ratio increases quickly, but still lower than that in the network using pure AODV.

Average Hop-count: In Fig. 7(c), the average hop-count of the network using pure AODV decreases from 8.84 to 2.46 drastically as p_N increases up to 40%, which indicates that long paths are suffocated and finally only adjacent nodes can communicate when more non-cooperative nodes present. In the network applied with the PROACtive protocol, the average hop-count is kept as high as 6.85 for *PRO-UNI*, same as 7.27 for *PRO-BRO*, when $p_N = 40\%$. This effect does not mean that our solution degrades the performance, but instead it indicates that communications via longer paths are preserved for better connections.

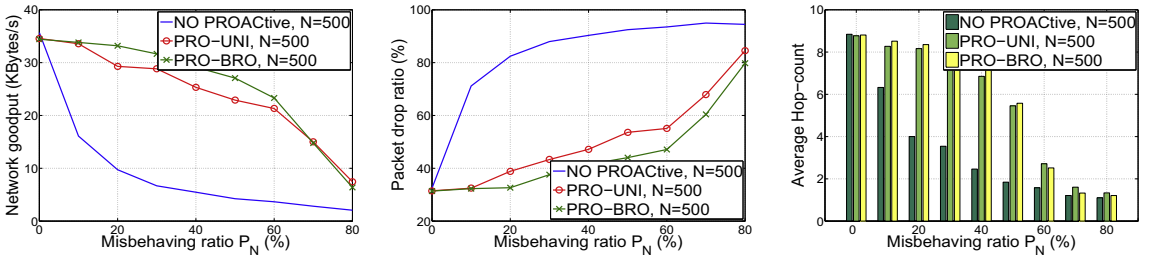
Therefore, for a well-connected network in the presence of DFD problem, generating robust topologies can improve network performance significantly and achieve a graceful performance degradation as the non-cooperative ratio increasing.

6.4. Impacts of density, scale, and mobility

Node Density: The node density μ refers to the average number of nodes in a node's transmission coverage. By fixing $N = 100$ and varying r from 100 m to 223.6 m, we obtain a series of scenarios with node density ranging from $\mu = 3.14$ (Sparse) to $\mu = 15.7$ (Dense). Fig. 8(a) shows the packet drop ratio for these two networks. Since the Sparse network is found actually disconnected, even without non-cooperative node, the packet drop ratio is as high as 80%. For the Dense network, the drop ratio is decreased by almost 50% as $p_N < 40\%$ when PROACtive is used. Recall that $\mu = \Theta(\log N)$ is an implicit condition for connectivity remarked in Section 3.2. This explains that our solution performs better in well-connected and dense networks than for sparse networks.

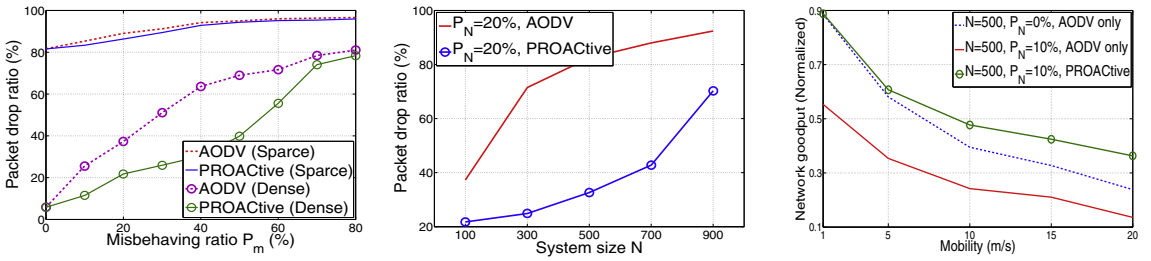
Network Scale: To examine the scalability of our protocol, we use the same node density ($\mu = 15.7$) and enlarge the network area A as the network size N increasing. In Fig. 8(b), the packet drop ratio is depicted against different network sizes. It is clear that the packet drop ratio is reduced tremendously by up to 65%, which means that PROACtive is quite scalable. Nevertheless, the packet drop ratio increases when the network size is increased. To explain this, we find that the path length between any source–destination pair increases as network scales up. As shown in [28], the path duration time is decreased exponentially to the path length, which implies that the more hops a path has the more likely the path is to break. Therefore, the paths in large networks are less reliable than those in small networks, which further induces a relatively higher drop ratio.

Node Mobility: To observe the impact of node mobility, we vary the average speed from almost stationary to 20 m/s, with pause time set as default. The non-cooperative ratio p_N is set to 10% for all networks with $N = 500$ and $r = 100$ m. Fig. 8(c) shows that the normalized network goodput (the ratio between the data received and sent) degrades significantly when mobility increases, can be improved by more than 20% when PROACtive is used. A surprising result is that the goodput in robust topologies with non-cooperative nodes is *even* higher than that in networks excluding



(a) Network goodput vs p_N (b) packet drop ratio vs p_N (c) Average hop-count vs p_N

Fig. 7. Network performance improvement by using PROACtive protocol.



(a) Impact of node density (b) Impact of network scale (c) Impact of node mobility

Fig. 8. Impact of node density, network size, and node mobility on the performance of PROACtive protocol.

non-cooperative nodes, depicted as the dotted line. This phenomena is credited to our well-defined dynamic cooperativity threshold scheme. For example, neither a malicious adjacent node nor a benign neighbor with high mobility can provide reliable link connections. Thus, a highly mobile node may also have low cooperativity and be excluded from topologies. When the non-cooperative ratio is small, e.g., $p_N \leqslant 10\%$, the increasing mobility becomes the dominant factor in neighbor and path selections. Consequently, the generated robust topology is composed of nodes providing more reliable connections for others, which improves the data delivery.

6.5. Comparison and discussion

There are three major components in PROACtive protocol: topology generation, cooperativity measurement, and signaling messages. In comparing PROACtive with other solutions, we have demonstrated generated network topology with $K - Neigh$ scheme[22], showing that the network topology generated by using probabilistic k-connectivity. Also, our approach distinguishes itself from all existing topology control works [29,22,30,26]. in that our objective is to generate robust topology against DFD attacks based on theoretical upper bound of robust space, while others are focused on minimizing energy consumption, reducing interferences, or improvement in network connectivity. The discrepancy in objectives may eventually yield different topologies.

Regarding cooperativity measurement, our scheme can easily be implemented in software, and does not require a *nuglet counter* in a tamper resistant hardware module [3]. In other similar reputation-evaluation systems, such as *Watchdog* [2], *CONFIDANT* [6] and *CineMA* [7], topological connectivity is not taken into consideration. A brief comparison between PROACtive and related solutions is reported in Table 2. More importantly, we designed several new signaling messages and demonstrated that our protocol has low signaling overhead. Furthermore, we have carried out extensive simulations to study the impact of a variety of network dynamics on network performance, which in turn demonstrate that the proposed PROACtive protocol can improve data forwarding in large-scale networks in the presence of non-cooperative nodes.

Table 2
Comparison between PROACtive and Watchdog/Confidant.

Solutions	Watch-dog	CONFI-DANT	PROAC-tive
Connectivity considered	No	No	Yes
Routing protocol specific	DSR	No	No
Misbehavior punished	No	Yes	Yes
Non-Cooperative ratio	40%	80%	80%
Network scale	10	50	100–900

7. Conclusions

In this paper, we tackled the DFD problem by achieving an optimal robust topology for a given wireless network. We first analyzed the trade-off between excluding non-cooperative nodes and maintaining k-connectivity w.h.p.. Then we proposed an integrated cooperativity measurement scheme to identify non-cooperative nodes so that they can be excluded in the generated network topology. Furthermore, we designed a new network protocol, PROACtive, to enable every node to build up its own coopera-

tive neighborhood dynamically. The simulation results validated that the PROACtive protocol can yield a robust network topology with a high probabilistic k-connectivity, low message complexity $\Theta(N)$, and low false positive ratio < 5%. With thorough simulations and examination, we demonstrate that PROACtive provides a cooperative platform for communications for both control and data planes, on which network performance, especially goodput, can be significantly improved.

References

[1] Y.-C. Hu, A. Perrig, D.B. Johnson, Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks, in: Proceedings of IEEE INFOCOM '03., 2003.

[2] S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating Routing Misbehavior in Mobile Ad hoc Networks, in: Proceedings of ACM MobiCom '00, 2000, pp. 255–265.

[3] L. Buttyan, J.-P. Hubaux, Stimulating cooperation in self-organizing mobile Ad Hoc networks, Mobile Networks and Applications 8 (5) (2003) 579–592.

[4] S. Zhong, J. Chen, Y.R. Yang, Sprite: A Simple, Cheat-Proof, Credit-based System for Mobile Ad-Hoc Networks, in: Proceedings of IEEE INFOCOM '03., 2003, pp. 1987–1997.

[5] I. Aad, J.-P. Hubaux, E.W. Knightly, Denial of Service Resilience in Ad Hoc Networks, in: Proc. of ACM MobiCom '04, 2004, pp. 202–215.

[6] S. Buchegger, J.L. Boudec, Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks), in: ACM MobiHoc '02, vol. 1, 2002, pp. 226 – 236.

[7] M. Frank, P. Martini, M. Plaggemeier, CineMA: Cooperation Enhancement in Manets, in: Proceedings of IEEE LCN '04., 2004, pp. 86–93.

[8] B. Bollobas, Modern Graph Theory, Springer, 1998.

[9] M. Penrose, Random Geometric Graphs, Oxford University Press, 2003.

[10] Y. Zhou, D. Wu, S.M. Nettles, On MAC-layer denial of service attacks in IEEE 802.11 Ad Hoc networks: analysis and counter measures, International Journal of Wireless and Mobile Computing 1 (3/4) (2006) 268–275.

[11] S. Radosavac, A.A. Cardenas, J.S. Baras, G.V. Moustakides, Detecting IEEE 802.11 MAC layer misbehavior in Ad Hoc networks: robust strategies against individual and colluding attackers, Journal of Computer Security 15 (1) (2007) 103–128.

[12] M.D. Penrose, On k-connectivity for a geometric random graph, Random Structures and Algorithms 15 (2) (1999) 145–164.

[13] C. Bettstetter, On the minimum node degree and connectivity of a wireless multihop network, in: Proceedings of ACM MobiHoc '02, ACM Press, 2002, pp. 80–91.

[14] X.-Y. Li, P.-J. Wan, Y. Wang, C.-W. Yi, Fault Tolerant Deployment and Topology Control in Wireless Networks, in: Proceedings of ACM MobiHoc '03, 2003, pp. 117–128.

[15] F. Xue, P. Kumar, The number of neighbors needed for connectivity of wireless networks, Kluwer Wireless Networks 10 (2) (2004) 169–181.

[16] C. Bettstetter, On the Connectivity of Ad Hoc Networks, The Computer Journal, Special Issue on Mobile and Pervasive Computing 47 (4) (2004) 432–447.

[17] P.-J. Wan, C.-W. Yi, Asymptotic Critical Transmission Radius and Critical Neighbor Number for k-Connectivity in Wireless Ad Hoc Networks, in: Proceedings of ACM MobiHoc '04, 2004.

[18] R. Hekmat, Ad-hoc Networks: Fundamental Properties and Network Topologies, Springer Netherlands, 2006.

[19] P. Santi, Topology Control in Wireless Ad Hoc and Sensor Networks, John Wiley and Sons Inc., 2006.

[20] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, D.E. Knuth, On the Lambert W function, Advances in Computational Mathematics 5 (1) (1996) 329–359.

[21] J.W. Harris, H. Stocker, Handbook of Mathematics and Computational Science, Springer, 1998.

[22] D. Blough, M. Leoncini, G. Resta, P. Santi, The k-neighbors approach to physical degree bounded and symmetric topology control in Ad Hoc networks, IEEE Transactions on Mobile Computing 5 (9) (2006) 1267–1282.

[23] F. Xing, W. Wang, On the resilient overlay topology formation in multi-hop wireless networks, in: Proceedings of IFIP/TC6 Networking 2007), 2007, pp. 1–12.

[24] C.E. Perkins, E.M. Belding-Royer, S.R. Das, URL <http://rfc.net/rfc3561.txt>.

[25] Y. Wang, W. Wang, X.-Y. Li,Distributed Low-Cost-Backbone Formation for Wireless Ad Hoc Networks, in: Proceedings of MobiHoc '05, 2005.

[26] S. Sorooshyari, Z. Gajic, Autonomous dynamic power control for wireless networks: user-centric and network-centric consideration, IEEE Transactions on Wireless Communications 7 (3) (2008) 1004–1015.

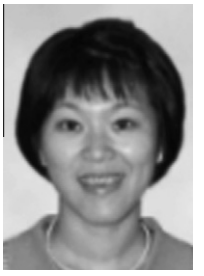
[27] M. Zhao, W. Wang, A unified mobility model for analysis and simulation of mobile wireless networks, ACM-Springer Wireless Networks 15 (3) (2009) 365–389.

[28] B.N.Sadagopan, F.Bai, A.Helmy, PATHS: Analysis of Path Duration Statistics and their Impact on Reactive MANET Routing Protocols, in: MobiHoc'03, Annapolis, MD, 2003.

[29] L. Li, J.Y. Halpern, P. Bahl, Y.-M. Wang, R. Wattenhofer, A cone-based distributed topology-control algorithm for wireless multi-hop networks, IEEE/ACM Transactions on Networking 13 (1) (2005) 147–159.

[30] M. Hajiaghayi, N. Immorlica, V.S. Mirrokni, Power optimization in fault-tolerant topology control algorithms for wireless mutli-hop networks, IEEE/ACM Transactions on Networking 15 (6) (2007) 1345–1358.

Fei Xing (S'06, M'09) received the B.S. degree in Telecommunication and Information Engineering, and the M.S. degree in Computer Science and Technology, both from Xian Jiaotong University, Xian, China, in 1999 and 2002, respectively. He earned the Ph.D. degree in Electrical and Computer Engineering at North Carolina State University, Raleigh, USA, in 2009. He is a senior software engineer with Cisco Systems. His research interests include resilient wireless networks design, mobile ad hoc networks and wireless communication systems. Before joining the doctoral program, he also worked as an engineer for FujiXerox Japan, Huawei Technologies, and Infineon Technologies, sequentially. He has been a member of IEEE since 2006 and a member of the Communication Society of IEEE since 2008.



Wenye Wang (M'98/ACM'99) received the B.S. and M.S. degrees from Beijing University of Posts and Telecommunications, Beijing, China. She also received the M.S.E.E. and Ph.D. degree from Georgia Institute of Technology, Atlanta, Georgia in 1999 and 2002, respectively. She is an Associate Professor with the Department of Electrical and Computer Engineering, North Carolina State University. Her research interests are in mobile and secure computing, network topology and architecture, and Smart Grid. Dr. Wang is a recipient of NSF CAREER Award in 2006. She is a senior IEEE member.