

Local Detection of Selfish Routing Behavior in Ad Hoc Networks

Bo Wang Sohraab Soltani Jonathan K. Shapiro
Pang-Ning Tan
Department of Computer Science and Engineering
3115 Engineering Building
Michigan State University
East Lansing, MI 48824
{wangbo1, soltanis, jshapiro, ptan}@cse.msu.edu

Abstract

Reputation mechanisms for detecting and punishing free-riders in ad hoc networks depend on the local detection of selfish behavior. Although naive selfish strategies based on dropping data packets are readily detected, more sophisticated strategies that manipulate ad hoc routing protocols present a greater challenge. In this work we develop a method to distinguish selfish peers from cooperative ones based solely on local observations of AODV routing protocol behavior. Our approach uses the finite state machine model of locally observed AODV actions to build up a statistical description of the behavior of each neighbor. We apply a series of well-known statistical tests to features derived from this description to partition the set neighboring nodes into a cooperative and selfish class. Simulation results for a non-mobile ad hoc network show that our approach can detect two different types of routing manipulation while maintaining a low rate of false positives.

1 Introduction

Ad hoc networks promise to provide wireless network connectivity in the absence of a network infrastructure by relying on end-hosts to forward data for each other. As ad hoc networks become more widely deployed, the cooperation of nodes in forwarding cannot be assumed a priori. Due to scarce power and access bandwidth a rational but self-interested node may attempt to avoid forwarding data for other nodes, even as it sends its own traffic through the system. Such free-riding behavior, if widespread, threatens the scalability of ad hoc networks.

Several reputation mechanisms have been proposed for encouraging cooperation in peer-to-peer systems including some specifically designed for ad hoc networks [3, 8]. Reputation mechanisms operate by identifying bad actors in a system and then isolating or punishing them. The threat of punishment can provide an incentive to cooperate if bad behavior can be detected with sufficient accuracy. Reputation mechanisms are built on some method of evaluating the cooperativeness of others based on direct observations. Some also involve the exchange of reputation information among nodes to improve convergence to a group consensus. Our work focuses on the first of these concerns—the local, or first-hand, assessment of reputation—which forms the foundation of any reputation mechanism. Without good local reputation assessment, additional exchange of reputation information may not improve detection accuracy, even if information is only exchanged with trusted nodes. Furthermore punishment is likely to be ineffective as an incentive if detection rate is low or false positives are high.

Previous work on detecting selfish behavior in ad hoc networks has largely focused on observing nodes' failure to forward data. The basic mechanism for detecting such failures was first proposed by Marti, et al, who dubbed the technique *Watchdog* [7]. Watchdog uses a so-called implicit acknowledgement obtained by overhearing a retransmitted packet. By promiscuously listening for an implicit acknowledgement, a node giving a data packet one of its neighbors can ascertain with reasonable certainty that the neighbor has also forwarded the packet.

Watchdog is quite successful at detecting dropped data—so successful that a selfish user wishing to evade detection would be well advised to seek alternate strate-

gies. In particular, a smart selfish node might forward all data packets it receives but take actions to minimize the chances of being asked to forward data in the first place. More sophisticated selfish behavior, therefore, is likely to involve manipulation of routing protocols.

In this work, we consider the problem of detecting selfish routing behavior based purely on local observations of the AODV routing protocol [9]. We employ a specification-based approach, which exploits the structure of the AODV protocol to extract useful features from the observed transmissions of neighboring nodes. We then apply statistical inference techniques to the data to classify neighboring nodes as either selfish or cooperative.

Our approach is similar in spirit to anomaly detection techniques from data mining since it defines misbehavior as a deviation ordinary behavior. Unlike anomaly detection, however, the model of normal behavior is based not only on statistical analysis of the observed data, but also on information about the routing protocol specification. Our technique requires no training data but instead compares observed behavior of multiple neighbors against each other, providing a basis for an on-line local reputation assessment algorithm.

The remainder of this paper is organized as follows: In Section 2 we review the operation of the AODV routing protocol and develop a finite state machine representation of locally observable node behavior. We present our detection algorithms in Section 3. In Section 4 we present the results of a preliminary simulation experiment designed to detect two types of selfish behavior in AODV routing. Section 5 reviews related work and we conclude in Section 6 with a discussion of potential directions for further study.

2 Specification-Based Approach

Consider the problem of an individual node in an ad hoc network—referred to hereafter as the local node—sharing wireless links with some number of neighboring nodes. The local node wishes to determine which, if any, of its neighbors is behaving selfishly. We assume that selfish nodes in this network attempt to manipulate the AODV routing protocol to minimize their chances of being included on routes for which they are neither source nor destination.

The AODV routing protocol works by propagating a route request (RREQ) message throughout the entire network by means of a flooding broadcast. Route reply (RREP) messages are unicast over a subset of the reverse broadcast paths, providing a request source with one or

State	Description
1:init	No RREQ observed
2:unexp RREP	Observed receipt of an "unexpected" RREP
3:rcvd RREQ	Observed receipt of RREQ
4:fld RREQ	Observed broadcast of RREQ
5:timeout RREQ	No activity observed after receipt of RREQ
6:rcvd RREP	Observed receipt of RREP
7:complete LRI	Observed forwarding of valid RREP
8: timeout RREP	No activity observed after receipt of RREP

Table 1. Interpretations of the states in Fig. 1.

more candidate routes. Nodes may cache the routes they discover and respond to route requests with cached information. We refer to the set of transmissions that flood a single RREQ message throughout the network along with the set of RREP messages transmitted in response as a *routing instance*.

Although no individual node is able to observe an entire routing instance, each node does see the subset of transmissions generated by itself and its neighbors. We refer to this subset as the *local routing instance* (LRI). The LRI includes transmissions sent by the local node itself as well as those sent by its neighbors, including messages overheard by the local node but not explicitly addressed to it. If we ignore data traffic and focus solely on AODV routing messages, we can think of the local node as observing a series of interleaved LRIs over the course of its lifetime in the system. This set of LRIs constitutes the data that must be examined for evidence of selfish routing behavior.

2.1 Finite State Machine Model

To impose additional structure on the data, the local node associates each transmission in a LRI with its sender and its receiver, or, in the case of its own outgoing broadcast RREQs, with multiple receivers.¹ The possible sequences of transmissions in a LRI are determined by the AODV protocol. We describe this set of possible behaviors using the finite state machine (FSM) description shown in Fig. 1. Table 1 gives the meanings of each state in the FSM.

The FSM in Fig. 1 describes the behavior of a single node with respect to a single LRI. Each transmission observed by the local node is recorded as a state transition in one or more neighbors' FSMs. It is important to emphasize that the FSM shown in Fig. 1 does not model the

¹In the case of RREQs broadcast by the local node itself, we assume such transmissions are received by all current neighbors, despite the possible absence of link-layer acknowledgements for broadcast transmissions.

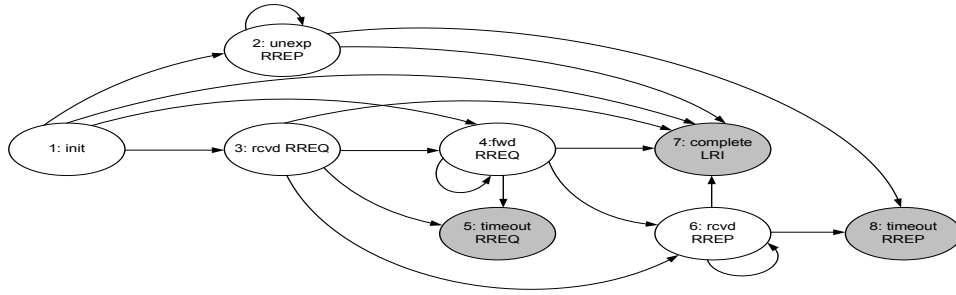


Figure 1. Finite state machine model of local AODV behavior. Final states are shaded.

exact behavior of a node participating in AODV routing. Instead, it models the node’s behavior *as observed by its neighbor, the local node*. For clarity in the following discussion, we refer to the node being observed by the local node as the *monitored node*.

Each LRI can be identified by the combination of source and destination contained a RREQ message. We denote the identifier for the k^{th} LRI observed by a node as (s_k, d_k) . Note that the combination (s_k, d_k) does not uniquely identify a LRI since a source can issue multiple route requests for the same destination. However, we find that this ambiguity has little effect on our analysis as long as subsequent re-issued route requests are suitably delayed so that there is only one active LRI for a given (s_k, d_k) at any point in time. In practice, AODV sources explicitly try to ensure such delays.

Prior to participating in an LRI—that is, prior to being observed to have received a RREQ for a particular (s_k, d_k) —the monitored node begins in the initial state 1. As the local node observes the monitored node’s behavior over the course of the LRI, it records a sequence of transitions from this initial state to one of three possible final states.

If the local node broadcasts a RREQ, it assumes that the monitored node receives it and records the transition $1 \rightarrow 3$ for that neighbor’s FSM. If the monitored node is observed to broadcast a RREQ, either the $1 \rightarrow 4$ or $3 \rightarrow 4$ transition is recorded, depending on whether the local node previously broadcast the RREQ. Transitions to timeout states occur when the local node fails to observe any additional activity for the LRI within a suitable duration. Transitions to final state 7 (complete LRI) occur when the monitored node is observed to forward a RREP.² It is in this state that the monitored node

²In this discussion, we have implicitly assumed that selfish nodes either drop messages or forward them without maliciously modifying message fields. To the extent that the local node can observe the modification of routing messages, the FSM can be readily extended to account for invalid forwarding actions, such as forwarding a RREP

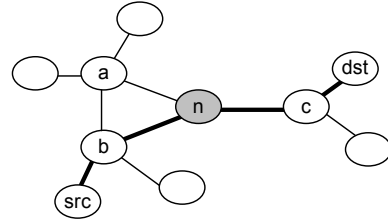


Figure 2. Example of the local routing instance observed by node n for the establishment of the route indicated with bold lines. For this example, all nodes are assumed to behave cooperatively.

becomes a candidate for inclusion on a route.

As an example, consider the LRI observed by a local node n during the discovery of a route from the source to the destination shown in Fig. 2. Table 2 shows the events observed by n and the corresponding state transitions in FSMs representing each of its three neighbors.

Upon reaching a final state, the FSM is considered complete and the local node stores the completed sequence of transitions X_k observed for the monitored node in the k^{th} LRI. These completed sequences are used to derive a matrix T containing an estimate of the probability of observing each state transition at least once during a LRI. We compute this probability as

$$T_{ij} = 1/N \cdot \sum_{k=1}^N \mathbf{1}(i \rightarrow j \in X_k)$$

where N is the total number of completed FSMs for the monitored node and $\mathbf{1}$ is an indicator function.

The storage requirements for computing T could be reduced by noting that each of the local node’s neighbors is implicitly in the initial state 1 for any LRI that

with an inflated hop count.

neighbor	event	state transition
a	a broadcasts RREQ	1→4
	n broadcasts RREQ	4→4
	timeout	4→5
b	b broadcasts RREQ	1→4
	n broadcasts RREQ	4→4
	n sends RREP to b	4→6
	b sends RREP to src (overheard)	6→7
c	n broadcasts RREQ	1→3
	c broadcasts RREQ	3→4
	c sends RREP to n	4→7

Table 2. Assignment of observed transmission events to neighbors and the corresponding state transition sequences for the three neighbors of node n in Figure 2

has not yet been observed. The local node therefore only needs to store information about FSMs that have transitioned out of the initial state but have not yet reached a final state. For any such uncompleted FSM, the local node must store the transition sequence X_k but may update T incrementally whenever a FSM reaches a final state and then discard X_k . Various probability estimation algorithms for this incremental update are possible, offering different tradeoffs between convergence and estimation bias.³

There are no transitions in the FSM that are inherently selfish. Timeout transitions will occur, for example, whenever RREP messages fail to follow the reverse broadcast path through the monitored node or due to channel errors during broadcasts. However, certain transitions can reasonably be considered more suspicious than others, especially when the corresponding value in the transition matrix is excessively large or unusually small. To capture this intuition, we introduce a weight matrix $W = [w_{ij}]$, where $w_{ij} = 1$ if transition $i \rightarrow j$ is considered to be suspicious and $w_{ij} = 0$ otherwise. We also introduce a type matrix $\Theta = [\theta_{ij}]$ where $\theta_{ij} = 1$ (resp. $\theta_{ij} = -1$) indicates that a high (resp. low) value of T_{ij} reflects selfish behavior.

2.2 Selfish Routing

We assume that the goal of a selfish node is to save as much power as possible while still evading detection. In processing routing messages, the node can choose among several actions to reduce its chances of being placed on a route. The most effective action is failing to re-broadcast RREQ messages, which ensures

³We defer a thorough exploration of incremental updating for future work.

that no RREP messages will ever be returned through the selfish node. Equivalently, the selfish node may re-broadcast suitably modified RREQ messages—reducing the request sequence number, for example. An alternative, somewhat more subtle approach would be to drop, delay, or modify RREP messages.

Described in terms of our FSM, all of a selfish node’s actions can be interpreted as attempts to avoid reaching final state 7, since any node reaching this state has both broadcast a valid RREQ and forwarded a valid RREP and is therefore a candidate for inclusion on a route.

3 Detection of Selfish Behavior

The conjecture of this work is that selfish behavior among the local node’s neighbors can be observed in the form of aberrant transition matrices. Described at a high level, our detection approach applies a series of statistical tests to attributes extracted from the set of transition matrices for all of the local node’s neighbors. The tests are designed to determine a threshold value of each attribute that separates cooperative neighbors from selfish ones.

In the following discussion, we let the local node’s neighbors form a set indexed by $1, \dots, n$. We also use the index 0 to designate the local node, itself. We use superscript notation (e.g. $T^{(u)}$) to associate a value with one particular neighbor.

3.1 Detection of Dropped RREQs

Consider two neighbors u and v and their transition matrices $T^{(u)}$ and $T^{(v)}$. To devise a statistical test for evaluating the degree to which u and v differ in behavior, we assume that each transition $i \rightarrow j$ has a Bernoulli distribution with parameter p_{ij} . Under this assumption, the transition matrix entry T_{ij} is the maximum likelihood estimate of the true p_{ij} , which we denote \hat{p}_{ij} .

The intuition behind our first statistical test is that if $\hat{p}_{ij}^{(u)}$ and $\hat{p}_{ij}^{(v)}$ have similar values, then u and v can be assumed to have behaved similarly with respect to the transition $i \rightarrow j$. We therefore form the null hypothesis

$$H_0 : p_{ij}^{(u)} = p_{ij}^{(v)}.$$

Defining $\hat{\Delta}_{ij} = \hat{p}_{ij}^{(u)} - \hat{p}_{ij}^{(v)}$ and using a Normal approximation, we reject H_0 if

$$\hat{\Delta}_{ij} > Z_{(1-\alpha)} \sigma_{\hat{\Delta}_{ij}}$$

where α is the error rate of rejecting H_0 when it is true, $Z_{(1-\alpha)}$ is the $(1 - \alpha)$ percentile in a standard normal distribution.

bution, and

$$\sigma_{\Delta_{ij}} = \sqrt{\frac{\hat{p}_{ij}^{(u)}(1 - \hat{p}_{ij}^{(u)})}{N^{(u)}} + \frac{\hat{p}_{ij}^{(v)}(1 - \hat{p}_{ij}^{(v)})}{N^{(v)}}}.$$

The hypothesis H_0 provides a test with respect to a single state transition. To measure of the difference in behavior between neighbors u and v , we compute the sum d_{uv} of transitions with $w_{ij} = 1$ for which H_0 is rejected. Formally,

$$d_{uv} = \sum_{i,j} \mathbf{1}(H_0 \text{ is rejected for } i \leftarrow j) \cdot w_{ij},$$

where $\mathbf{1}$ is an indicator function. We define the *difference matrix* $D = [d_{u,v}]_{u,v \in \{1, \dots, N\}}$, which shows the behavior differences between each pair of neighbors.

Although the matrix D is helpful for identifying different class of neighbor behavior, it is not sufficient to identify which neighbors should be classified as selfish. To capture the level of selfishness, we define a suspicion level of neighbor u as

$$S^{(u)} = \sum_{(i,j)} s_{ij}^u$$

where

$$s_{ij}^u = \begin{cases} \max_v T_{ij}^{(v)} - T_{ij}^{(u)} & \text{if } \theta_{ij} = -1 \\ T_{ij}^{(u)} - \min_v T_{ij}^{(v)} & \text{if } \theta_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Roughly speaking, $S^{(u)}$ measures the magnitude of the difference between neighbor u and the least suspicious neighbor for each weighted transition.

We currently set weight and type values heuristically based on our knowledge of the AODV routing protocol and on the type of selfish behavior being detected. For detecting dropped RREQs, we scrutinize the transitions out of state 3, setting $w_{34} = w_{35} = 1$, $\theta_{34} = -1$, $\theta_{35} = +1$ and all other weights and types to zero. These settings capture the idea that nodes who drop RREQs will be observed to time out more and re-broadcast less than normal nodes.

For every neighbor u we define an anomaly score

$$a^{(u)} = \sum_{v, v \neq u} S^{(u)} d_{uv}. \quad (1)$$

Without loss of generality, let us re-index the neighboring nodes so that the vector of anomaly scores is ranked according to the order statistics—i.e. $a^{(1)} \leq a^{(2)} \dots \leq a^{(n)}$. We next devise a method to determine a threshold

τ about which to partition the set of neighbors. Those neighbors with anomaly scores less than τ are classified as normal, while those with scores greater than τ will receive greater scrutiny and may be classified as selfish.

Selecting an initial threshold value τ is challenging. We choose a value by estimating the anomaly scores that neighboring nodes would compute for the local node itself and taking the maximum value. The rationale behind this approach is that a node should classify as normal any neighbor with a anomaly score lower than its own. Thus, initially we have

$$\tau = \max_v a^{(v,0)}, \quad (2)$$

where $a^{(v,0)}$ is the local node's estimate of its own anomaly score as computed by neighbor v . Producing local estimates of its own anomaly score requires that the local node track FSMs for itself from each neighbors perspective, effectively doubling the storage and computation requirements of the algorithm. We believe that more efficient choices can be made for the initial value of τ , but leave this question for future work.

Having selected an initial threshold, we first perform a Fisher Sign Test⁴ on τ . The null hypothesis H_0 of the sign test states that τ is the median of the distribution of anomaly scores. The test itself is non-parametric, making no assumptions about the underlying distribution of anomaly scores. If the test accepts H_0 , we raise τ to the next greater order statistic of anomaly scores. We perform this test iteratively until H_0 is rejected or we run out of values. If H_0 is never rejected, all neighbors are labelled normal; otherwise, neighbors with anomaly scores lower than τ are labelled normal. For anomaly scores exceeding τ , we apply additional scrutiny in the form of a heuristic designed to save neighbors whose scores only slightly exceed the threshold. We define two distances as follows:

$$D_1 = \tau - \tau^-, \quad D_2 = \tau^+ - \tau$$

where τ^- and τ^+ are the closest anomaly scores below and above τ , respectively. We label the neighbor with anomaly score τ^+ as selfish if $D_2 > D_1$. Otherwise, we label this neighbor normal and increase the threshold $\tau \leftarrow \tau^+$. We perform this test iteratively until $D_2 > D_1$, at which point all remaining unclassified neighbors are labelled selfish.

3.2 Route Reply Drop Detection

In the preceding section, we used features based on the relative frequencies of individual transitions as

⁴See, e.g., p.426 of [11].

the underlying features for statistical tests. This approach worked well for selfish nodes who drop RREQs, but works poorly for detecting dropped RREPs because many fewer events of interest are available in this case. In our simulations, for example, the number of RREQ-based transitions observed by the local node typically exceeds the number of RREP-based transitions by roughly a factor of ten. However, examination of the FSM in Fig. 1 shows that a neighbor forwarding a RREP always terminates in state 7 and one who drops RREPs always terminates in state 8. Furthermore, all transitions to final state 7 reduce the level of suspicion whereas all transitions to state 8 increase suspicion.⁵ Therefore, to detect dropped RREPs, we devise tests based on the total flow into final states 7 and 8, allowing us to gather better statistics for each measured feature.

We define $f_7^{(u)}$ and $f_8^{(u)}$ as the total flow into states 7 and 8, respectively, as derived from the transition matrix of neighbor u .

$$f_7^{(u)} = \sum_{i=1}^8 T_{i7}^{(u)}, \quad f_8^{(u)} = \sum_{i=1}^8 T_{i8}^{(u)}.$$

Observe that $f_7^{(u)}$ and $f_8^{(u)}$ are anti-correlated in the sense that $f_7^{(u)}$ tends to be low for selfish nodes and $f_8^{(u)}$ tends to be high. Thus, the difference $d^{(u)} = f_7^{(u)} - f_8^{(u)}$ is more likely to be positive if u is selfish.

Let the vector $d = d^{(1)}, \dots, d^{(n)}$ be the vector of differences for all of the local node's neighbors, re-indexed, as above, in ascending sorted order. We perform a Wilcoxon Signed-Rank Test⁶ on this vector to identify selfish nodes. Like the Fisher Sign Test, above, this test is non-parametric. The null hypothesis H_0 of the Signed-Rank Test states that the elements of d are symmetrically distributed about a median of zero. This test for symmetry is important because—as we observe in our simulations—negative values of $d^{(u)}$ for normal nodes tend to have greater in absolute value than the positive values for selfish nodes. Accepting H_0 implies that neighbors with positive $d^{(u)}$ values do not differ markedly from those with negative values. To be conservative, if H_0 is accepted, we classify all nodes as normal. If H_0 is rejected, we label a neighbor u selfish if $d^{(u)} > 0$ and normal otherwise.

⁵In contrast, for the case of dropped RREQs and final state 5, transition $3 \rightarrow 5$ is suspicious whereas $4 \rightarrow 5$ is not.

⁶See, e.g., p.403 of [11] or [15].

4 Evaluation

4.1 Simulation Setup

Using the NS-2 simulator [1] (version 2.27), we simulate a flat area of 670m by 670m with 50 randomly positioned stationary wireless nodes using AODV for routing. All nodes use a 2Mbps 802.11 radio with a transmission range of 250m. Each simulation lasts for 400 seconds. A short timeout value of 0.5 seconds triggers FSM transitions based on failure to observe an implicit acknowledgement—i.e. a forwarded or re-broadcast message. A longer timeout value of 3 seconds is used for transitions caused by failing to receive a RREP.

The communication model used in our simulation is an extension to the model proposed in [10]. Traffic is generated by constant bit rate (CBR) sessions with sources and destinations uniformly chosen from the population. The aggregate session arrival process is poisson with inter-arrival rate A and session durations are exponentially distributed with mean D . Each CBR session transmits at rate λ with a packet size of 512 bytes.

The total data traffic volume V in this network can be calculated according to Little's Law as $V = AD\lambda$. We fix V to be 60 packets/second and adjust parameters A , D , and λ to vary the relative amounts of control and data traffic. As observed in [10], the routing overhead increases with the average number of sessions in the network, which is given by $A * D$. We set A to be 3 second, D to be 60 seconds and λ to be 3 packets/second. These values are chosen to simulate realistic network operating conditions with high packet delivery ratio and tolerable routing overhead.

We assign 10% of the nodes, i.e. 5 nodes selected at random, to be selfish and perform simulations for three scenarios—(1) selfish nodes drop RREQs, (2) selfish nodes drop RREPs, (3) a mixed population with 3 RREQ-dropping nodes and 2 RREP-dropping nodes. RREQ-dropping nodes packet discard route request messages with a probability that can be controlled. For the simulations presented here, requests are dropped with 80% probability. RREP-dropping nodes always drop route replies and, in addition, always rebroadcast RREQs for cached routes instead of generating a reply from cache.

In this work, we perform all selfishness detection as a off-line post-processing step after each simulation. From each raw simulation trace we extract locally observable events for each node. We parse these localized traces to identify individual LRIs, generating transition matrices in the process. Finally, we apply our detection

tests. We can apply the RREQ and the RREP test individually or as a combined test. In the combined test, the set of nodes labelled selfish is the union of the sets returned by the individual tests.

4.2 Results

The performance of a detection algorithm can be characterized by any of several widely used measures. We choose to look at the detection rate—the fraction of selfish neighbors detected—and the false alarm rate—the fraction of normal neighbors incorrectly labelled selfish.

Scenario	Detection Rate	False Alarm
drop RREQ	0.6494	0.0376
drop RREP	0.7514	0.0023
mixed	0.7782	0.0439

Table 3. Detection and false alarm rates for the three simulation scenarios.

We simulated each of the three scenarios described above five times with randomized node positions, and session arrival times and session durations. For each simulation, we apply the detection methods appropriate to the type of selfishness exhibited by the population, performing a combined test when both types of behavior are present. Table 3 summarizes detection and false alarm rates computed over all nodes and averaged over five simulation runs. To get a more fine-grained view of how individual nodes perform, we plot in Fig. 3 the complementary CDF for detection rate and CDF for false positive rates seen by individual nodes from all simulation runs.⁷ For detection rate (resp. false alarm rate), good performance is indicated by the knee of the curve being as close as possible to the right-hand (resp. left-hand) side of the plot.

We observe that the test for dropped RREPs is quite strong. In the test for dropped RREQs, however, the moderate detection rate indicates that the test fails to detect some selfish nodes in some cases. This is not surprising given that our method is conservative in the sense that it requires the rejection of a null hypothesis to classify neighbors as selfish. We believe that this sensitivity can be tuned by adjusting the acceptance level for H_0 in this test, at the possible expense of increasing the false positive rate and intend to explore this tradeoff in the immediate future.

⁷These plots exclude approximately 10% of nodes who had no selfish neighbors.

5 Related Work

Marti, et al. first proposed the Watchdog mechanism to detect faulty and malicious behavior based on dropping data and combined it with a mechanism called Pathrater to avoid routes through such nodes [7]. Buchegger and Le Boudec extended the use of Watchdog-style detection to identify the selfish dropping of data and showed how second-hand reputation information from untrusted nodes could be safely incorporated to improve detection accuracy [2]. Our work differs from these approaches in our focus on detecting selfish routing behavior and our restriction to using only first-hand information.

Specification-based approaches have become popular in intrusion detection as a way to improve accuracy and reduce false positive rates compared to purely statistical anomaly detection techniques, which use no underlying domain knowledge. Sekar, et al. [12] introduced a general technique using FSM representations of protocols and applied it to detecting misbehavior in wired networks.

Tseng et al. have applied FSM-based techniques to the detection of malicious routing behavior in AODV [13]. Their approach relies on cooperative network monitors to aggregate observations at different locations. Huang and Lee applied anomaly detection [5, 4] and, more recently, specification-based techniques [6] to ad hoc networks for local detection of malicious behavior. Vigna, et al. use a signature-based approach to detect intrusions in AODV [14]. The signature of selfish behavior in this work turns out to be similar to that detected by watchdog.

Our work contrasts with [6] and [13] in that we do not attempt to identify specification violations or transitions to an "alarm state". Similarly to [12], we adopt the FSM description to derive a useful set of features from observed data for statistical analysis. In contrast with [13], we focus on developing a technique that can be implemented on an individual node based only on locally collected data to detect selfish route-avoiding behavior. Our work differs from all of the above work in its focus solely on selfish behavior, where the goal is not to prevent misbehavior completely, but to mitigate its performance impact at reasonable cost.

6 Conclusions and Future Work

In this work, we hypothesized that selfish behavior can be distinguished from cooperative behavior by comparing the statistical behavior of neighbors across mul-

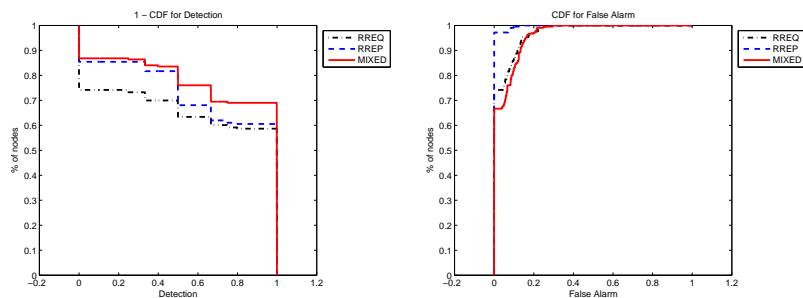


Figure 3. Complementary CDF for detection rate and CDF for false alarm rate.

multiple local routing instances. We have taken some first steps toward developing a robust detection technique based on this idea and have been able to detect simple strategies of dropping RREQ or RREP messages while maintaining a low false-positive rate.

There is significant room for improvement in our technique, however. In future work, for example, we plan to re-examine the use of heuristically chosen weights for detecting dropped RREQs. Furthermore, although we have focused on purely local detection, it is reasonable to believe that the performance of detection can be improved by making judicious use of data from other nodes. Finally, we have yet to consider the impact of mobility on our technique.

It is likely that more sophisticated selfish strategies will combine RREQ- and RREP-dropping probabilistically to reduce the detectable signature of either individual strategy. However, any such strategy must ultimately avoid terminating in final state 7, and will likely have a statistical signature that differs from that of cooperative behavior and can be detected by improved versions of our basic approach.

References

- [1] <http://www.isi.edu/nsnam/ns/index.html>.
- [2] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, 2002.
- [3] S. Buchegger and J.-Y. L. Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *2nd Workshop on the Economics of Peer-to-Peer Systems*, 2004.
- [4] Y. Huang, W. Fan, W. Lee, and P. S. Yu. Cross-feature analysis for detecting ad-hoc routing anomalies. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, 2003.
- [5] Y. Huang and W. Lee. A cooperative intrusion detection system for ad hoc networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, Fairfax, VA, 2003.
- [6] Y. Huang and W. Lee. Attack analysis and detection for ad hoc routing protocols. In *Proceedings of The 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, Sophia Antipolis, France, 2004.
- [7] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings MobiCom 2000*, pages 255–265, 2000.
- [8] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Sixth IFIP conference on security communications, and multimedia (CMS 2002)*, Portoroz, Slovenia, 2002.
- [9] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, 1999.
- [10] H. Pucha, S. M. Das, and Y. C. Hu. The performance impact of traffic pattern on routing protocols in mobile ad hoc networks. In *The Seventh ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, Venezia, Italy, 2004.
- [11] J. Rice. *Mathematical statistics and data analysis*. Duxbury Press, 2 edition, 1995.
- [12] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: A new approach for detecting network intrusions. In *ACM Computer and Communication Security Conference (CCS'02)*, Washington, DC, USA, 2002.
- [13] C. Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. A specification-based intrusion detection system for aodv. In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003.
- [14] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer. An intrusion detection tool for aodv-based ad hoc wireless networks. In *20th Annual Computer Security Applications Conference*, Tucson, Arizona, 2004.
- [15] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.