A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs

C. Zouridaki, B. L. Mark, M. Hejmo
ECE Dept., MS 1G5
George Mason University
Fairfax, VA
{czourida,bmark,mhejmo}@gmu.edu

R. K. Thomas
SPARTA, Inc.
5875 Trinity Parkway, Suite 300
Centreville, VA 20120
roshan.thomas@sparta.com

ABSTRACT

In mobile ad hoc networks (MANETs), a source node must rely on other nodes to forward its packets on multi-hop routes to the destination. Secure and reliable handling of packets by the intermediate nodes is difficult to ensure in an ad hoc environment. We propose a trust establishment scheme for MANETs which aims to improve the reliability of packet forwarding over multi-hop routes in the presence of potentially malicious nodes. Each node forms an "opinion" about each of the other nodes based on both first and second-hand observation data collected from the network. The opinion metric can be incorporated into ad hoc routing protocols to achieve reliable packet delivery even when a portion of the network exhibits malicious behavior. We present numerical results which demonstrate the effectiveness of the proposed trust establishment scheme.

Categories and Subject Descriptors

C.2.1. [**Computer-Communication Networks**]: Network Architecture and Design—*distributed networks,wireless communication*

General Terms

Algorithms, Performance, Reliability, Security

Keywords

Mobile ad hoc networks, Trust establishment, Routing

1. INTRODUCTION

The lack of infrastructure in a mobile ad hoc network (MANET) makes it difficult to ensure the reliability of packet delivery over multi-hop routes in the presence of malicious nodes acting as intermediate hops. To improve the reliability of packet delivery, we propose a trust establishment scheme, which we call Hermes[1], that enables a source node to route packets over more "trustworthy" intermediate nodes. In the proposed scheme, each node assigns a "trustworthiness" metric to each of its neighbor nodes based on direct observations of packet forwarding behavior. The concept of trustworthiness is extended to the notion of an "opinion" that a node has of any other node. The opinion metric can be applied in a various network settings to improve packet delivery performance. In particular, the opinion metric can be incorporated into ad hoc routing protocols to route packets on more "trusted" paths.

Our proposed trust establishment scheme makes use of a Bayesian approach similar to that used in [4]. In the Bayesian approach, trust values are computed under the assumption that they follow a beta probability distribution. The parameters of the beta distribution are estimated by accumulating empirical observations of packet forwarding behavior. A *trust* metric can then be derived from the parameters of the beta distribution. Our approach to trust evaluation differs from that in [4] in that we derive an additional parameter called *confidence*, which characterizes the statistical reliability of the computed trust metric.

The notion of maintaining two metrics, trust and confidence, is also considered in [18]. In [18], the trust and confidence metrics assigned to nodes are extended to paths via a semi-group approach. In contrast, we propose a new metric, called "trustworthiness," which combines the trust and confidence metrics in a manner that is tunable in terms of two parameters. The trustworthiness metric is used to formulate the more general "opinion" metric, which can be incorporated into routing protocols in a transparent manner. We present a windowing scheme to systematically expire old observation data in order to maintain the accuracy of the opinion metric.

The main contributions of this paper can be summarized as follows: (1) a scheme for evaluating trust and confidence with respect to packet delivery based on empirical observations; (2) a scheme for mapping trust and confidence into a "trustworthiness" metric and its extension to an "opinion" metric; (3) a windowing scheme to improve the fidelity of the opinion metric; (4) an approach to incorporate the opinion metric into ad hoc routing protocols to improve reliable packet delivery. We present simulation results to demonstrate the effectiveness of the proposed trust establishment scheme in distinguishing between malicious vs.

SASN'05, November 7, 2005, Alexandria, Virginia, USA.

[1]In Greek mythology, Hermes was the trusted messenger of the gods.

non-malicious nodes as well as in selecting the more "trustworthy" routes for packet delivery.

The remainder of the paper is organized as follows. Section 2 briefly reviews related work on trust establishment in ad hoc networks and sets the present work in context. Section 3 describes a methodology for evaluating trust between two neighbor nodes from first-hand observation data. We use the term *trustworthiness* to denote this notion of trust. Section 4 extends the trust evaluation scheme to a general pair of nodes. We use the term *opinion* to denote this extension of the trustworthiness concept. The issues involved accumulating trust information from first-hand observation data are treated in section 5. The application of the opinion metric to realize "trust-aware" ad hoc routing is discussed in section 6. Section 7 presents results from simulation experiments that demonstrate the key properties of the proposed trust establishment scheme. Finally, the paper is concluded in Section 8.

2. RELATED WORK

The objective of the present paper is to introduce a relatively complete and general conceptual framework for trust establishment with respect to reliable packet delivery. In principle, the Hermes framework can be applied to any ad hoc network. We remark that measures should also be taken to ensure the security of the trust establishment phase in conjunction with the routing protocol. For example, the exchange of trustworthiness values $T_{i,j}$ should be authenticated and protected by cryptographic primitives.

The topic of secure routing for MANETs has been studied extensively in recent years [20, 14], and many of the same mechanisms used to secure routing can also be applied to Hermes. In particular, Hermes could be applied to one of the secure routing protocols proposed in the literature [9, 8]. A full discussion of the use of cryptographic primitives to secure Hermes and trust-aware routing is beyond the scope of the present paper.

In recent years, there has been considerable interest in the topic of trust establishment for ad hoc networks. As mentioned in the Introduction, our proposed trust evaluation framework is based on a Bayesian approach similar to the one presented in [4]. A key difference, however, is that our framework incorporates the notion of statistical confidence associated with a trust value. The notion of confidence was proposed in [18] and a semi-ring approach was suggested to evaluate trust and confidence along network paths. In our approach, however, we map trust and confidence into a new metric, called "trustworthiness," which can more transparently be incorporated into network decisions such as route selection. Furthermore, our framework deals directly with the issue of collecting of evidence from the network.

In [17], a trust model is presented that allows the evaluation of the reliability of the routes, using only first-hand information. On the other hand, our approach to trust evaluation incorporates third-party information to derive the notion of an opinion that a given node has for any other node. The main idea of [1] is to bootstrap secure wireless communications via pre-authentication over a location-limited channel. As in [17], trust evaluation is based only on direct first-hand information.

The authors of [7] present a high-level framework for generation, revocation and distribution of trust evidence and demonstrate the significance of estimation metrics in trust establishment. They argue that a large body of trust evidence has to be generated, stored and protected across the network nodes, routed where needed and evaluated speedily to validate dynamically formed trust relations. A mechanism for trust evidence dissemination based on a model of ant behavior is proposed in [10] along the lines suggested in [7]. In contrast, our work focuses on developing metrics and mechanisms for establishing trust with respect to the objective of reliable packet delivery. In [13], a set of trust values are assigned to nodes in the network. The AODV routing protocol is modified such that a node applies different encryption keys to arriving packets depending on the trust value of the node and the security level required by the packet. However, the issue of how to compute the trust values assigned to nodes is not addressed.

In [5], a framework for stimulating cooperation in MANETs is proposed. The approach is based on a credit system for packet forwarding. The goal of collaboration is also pursued in [6], which proposes a trust management model, whereby each node carries a portfolio of credentials, which it uses to prove its trustworthiness. An autonomous trust establishment framework is proposed in [11, 2], which relies on the introduction of pre-trusted agents and a public key infrastructure. In [19], a trust framework is proposed for the purpose of establishing a set of group keys.

3. FIRST-HAND TRUST EVALUATION

In this section, we describe our approach to computing trust given a set of first-hand observations obtained from the network.

3.1 Bayesian Framework

In the Bayesian framework (cf. [4]), a random variable R, taking values on the interval $[0, 1]$, is associated with a given node. The random variable R represents a notion of trust and is assumed to follow a beta distribution. A realization of R is taken to be the trust value associated with the node. Since R is assumed to be beta distributed, trust is represented by the two parameters of the beta distribution.

The beta distribution is used because of its reproducibility property under the Bayesian framework. For a given node i, we define a sequence of random variables $R_1, R_2, \cdots$, where R_k characterizes the trust value at the sampling time k. For example, suppose that at time k, N_k network observations have been collected for a given node i. In particular, N_k is the number of packets that have been sent to the node i to be forwarded to other nodes. Let M_k be the number of packets actually forwarded by the node, out of the N_k packets that were sent to node i for forwarding at time k. Suppose a prior probability density function (pdf) for R_{k-1}, denoted by $f_{k-1}(r)$ is known. Then the posterior pdf of R_k (given that $N_k = n$ and $M_k = m$) can be obtained from Bayes theorem [15] as follows:

$$f_k(r) = \frac{f_k(M_k = m|r, N_k = n) f_{k-1}(r)}{\int_0^1 f(M_k = m|r, N_k = n) f_{k-1}(r) dr}, \quad (1)$$

where $f_k(M_k = m|r, N_k = n)$ is called the likelihood function and has the form of a binomial distribution:

$$f_k(M_k = m|r, N_k = n) = \binom{n}{m} r^m (1-r)^{n-m} \quad (2)$$

The prior pdf $f_{k-1}(r)$ summarizes what is known about the distribution of R_{k-1}. Under the assumption that prior pdf $f_{k-1}(r)$ follows a beta distribution, it can be shown that the posterior pdf $f_k(r)$ also follows a beta distribution. The beta distribution with parameters a and b is defined as follows:

$$beta(a,b) = \frac{r^{a-1}(1-r)^{b-1}}{B(a,b)} = \frac{r^{a-1}(1-r)^{b-1}}{\int_0^1 r^{a-1}(1-r)^{b-1}dr} \tag{3}$$

for $0 \leqslant r \leqslant 1$. In particular, if

$$f_{k-1}(r) \sim \text{beta}(a_{k-1}, b_{k-1}),$$

then given that $N_k = n_k$ and $M_k = m_k$ we have

$$f_k(r) \sim \text{beta}(a_{k-1} + m_k, b_{k-1} + n_k - m_k).$$

Therefore, $f_k(r)$ is characterized by the parameters a_k and b_k, defined recursively as follows:

$$a_k = a_{k-1} + m_k \text{ and } b_k = b_{k-1} + n_k - m_k.$$

At system initiation time ($k = 0$), there is no information for the network. Therefore, we assume that R_0 has the uniform distribution over the interval $[0, 1]$, i.e.,

$$f_0(r) \sim U[0, 1] = \text{beta}(1, 1),$$

which indicates our ignorance about the node's behavior at time 0.

3.2 Trust and confidence values

We define the trust value, t^k, assigned to a node at time k to be equal to the mean value $\mu(a_k, b_k)$ of the beta(a_k, b_k) distribution corresponding to the pdf $f_k(r)$ as follows:

$$t^k \triangleq \mu(a_k, b_k) = \frac{a_k}{a_k + b_k}, \tag{4}$$

for $0 \leqslant \mu \leqslant 1$. We define the confidence value, c^k, associated with the trust value t^k in terms of the standard deviation $\sigma(a_k, b_k)$ corresponding to the pdf $f_k(r)$ as follows:

$$\begin{aligned} c^k &\triangleq 1 - \sqrt{12}\ \sigma(a_k, b_k) \\ &= 1 - \sqrt{\frac{12 a_k b_k}{(a_k + b_k)^2 (a_k + b_k + 1)}} \end{aligned} \tag{5}$$

where $0 \leqslant c^k \leqslant 1$. A value of c^k close to one indicates high confidence in the accuracy of the computed trust value t_k, whereas a value close to zero indicates low confidence.

The definition of confidence value (5) captures the statistical dispersion from the mean value of the distribution, which corresponds to the trust value, as defined in (4). Note that the closer the kth beta probability distribution corresponding to $f_k(r)$ approximates a (shifted) Dirac function, the more confidence is placed in the trust value t^k, since the Dirac function indicates absolute certainty.

At system initialization time ($k = 0$), the trust value assigned to each node is given by $t^0 = \mu(1, 1) = 0.5$ which indicates our ignorance about the node's behavior. If we take the value 0.5 as the threshold that must be exceeded in order to consider a node to be trusted, then at time 0 a node is considered neither trusted ($0.5 < \mu \leqslant 1$), nor misbehaving ($0 \leqslant \mu < 0.5$). Note that the associated confidence value is $c^0 = 0$ according to (5). Fig. 1 shows how confidence grows when the number of observations grows, for different values of the parameters a and b of the beta distribution.

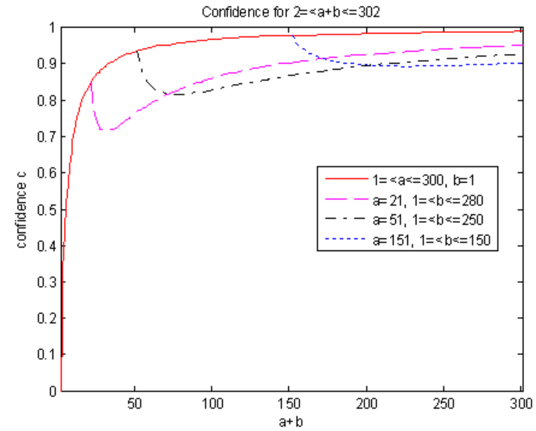

Figure 1: Confidence vs. number of observations.

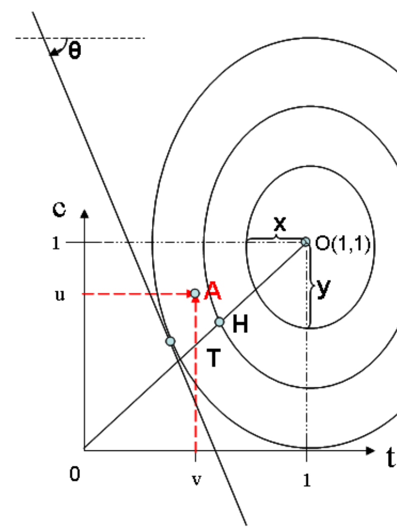

Figure 2: $(t, c)_{ij}$ relationship - Derivation of T_{ij}

3.3 Trustworthiness

As discussed in the previous section, at each time instant k a given node can be characterized by a pair (t^k, c^k). In particular, node i characterizes its trust in node j at time k by the pair $(t^k_{i,j}, c^k_{i,j})$. However, using a pair of values to describe the opinion of a node for another node makes comparisons between different nodes difficult. In particular, it is difficult for a given node to decide which of its neighbors is more "trusted" given the corresponding set of (t, c) values. This section presents a flexible method to transform the pair of values (t, c) into a single value T, which we call *trustworthiness*.

We note that given a pair (t, c) assigned to a node, the greater the value of c, the more the value of t can be considered as correctly reflecting the trust associated with the node. On the other hand, the smaller the value of c, the less the value of t should be considered as valid. This fact dictates that for large values of confidence, the trust value t should be weighted more than the confidence value c. Conversely, for small values of confidence, the trust value t should weighted less than the confidence value c.

Fig. 2 shows that the set of (t, c) values lies in the unit square region defined by $0 \leq t \leq 1$ and $0 \leq c \leq 1$. For example, the point A corresponds to the pair (u, v). In order to define trustworthiness, each pair (t, c) in the unit square must be mapped into a single value T. There are many ways to define the mapping from (t, c) to T. Fig. 2 illustrates the approach we have taken, which is based on considering the family of ellipses centered at the point $(1, 1)$, defined as follows:

$$\frac{(t-1)^2}{x^2} + \frac{(c-1)^2}{y^2} = 1, \tag{6}$$

where the pair of values of (x, y) defines the size and shape of the ellipse. The portion (if any) of the (x, y)-ellipse that lies in the unit square determines the set of (t, c) pairs that are mapped to a common value of trustworthiness T defined by

$$T(t, c) \triangleq 1 - \frac{\sqrt{\frac{(t-1)^2}{x^2} + \frac{(c-1)^2}{y^2}}}{\sqrt{\frac{1}{x^2} + \frac{1}{y^2}}} \tag{7}$$

We define a "default" value of trustworthiness

$$T_{def} \triangleq T(0.5, 0),$$

representing the trustworthiness value assigned to a node when its assigned trust and confidence values are $t = 0.5$ and $c = 0$, respectively. Thus, the value T_{def} represents ignorance about the trustworthiness of a node. The value T_{def} can be interpreted as a threshold of trustworthiness. If the trustworthiness of a node exceeds T_{def}, then the node is considered "trustworthy." Otherwise, the node is viewed as "untrustworthy."

The tangent line of a point (t, c) in the unit square lying on an ellipse with fixed parameters x and y, dictates the relationship between (t, c) and the trustworthiness value T. Let θ denote the angle between the tangent line and the t-axis. The value of θ lies in the interval $[-\pi/2, 0]$ and determines the mapping from (t, c) to T as follows:

- For $\theta = 0$, the value of t is ignored, i.e., $T = c$.
- For $-\pi/4 \leqslant \theta < 0$, the value of c weighs more heavily than the value of t in determining T.
- For $\theta = -\pi/4$ the values of t and c weigh equally in determining T.
- For $-\pi/2 < \theta < -\pi/4$, the value of t weighs more than the value of c .
- For $\theta = -\pi/2$, the value of c is ignored, i.e., $T = t$.

We now consider the impact of the choice of parameters x and y (i.e., the choice of ellipse) on the mapping of (t, c) to T. We will also refer to the x and y parameters as trustworthiness parameters.

- When $x > y$, the angle of the tangent to the ellipse at points (t, c) in the unit square takes values in the interval $(-\pi/4, 0]$ for the majority of the ellipse's points (within the unit square). This implies that that the confidence value has greater weight than the trust value for the majority of points on the ellipse.

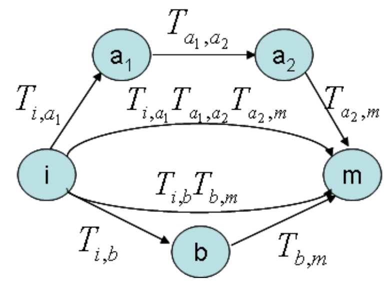

Figure 3: Example of opinion calculation for non-neighbors i and m.

- When $x = y = r$, the ellipse becomes a circle of radius r. The tangent line at the point $H = (t_H, c_H)$ in Fig. 2 has an angle of $\theta = -\pi/4$. At the point H, the values of t and c have equal weight in determining T, i.e., $T = (t + c)/2$. For all points (t, c) on the ellipse that lie below H (i.e., $c < c_H$), the value of c has a larger weight than the value of t in determining T. Conversely, for all points (t, c) on the ellipse lying above H, the value of t has a larger weight than c in determining T.
- When $x < y$, the angle of the tangent to the ellipse at points (t, c) in the unit square takes values in the interval $[-\pi/2, -\pi/4)$ for the majority of the ellipse's points (within the unit square). This implies that the trust value has greater weight than the confidence value for the majority of points on the ellipse.

The issue of choosing appropriate values for x and y is investigated further through computer simulations in section 7.

4. FORMULATION OF OPINIONS

We generalize the notion of trustworthiness to the concept of *opinion*, which incorporates second-hand trustworthiness values from third-party nodes. The propagation of trustworthiness information to form an opinion is similar to the concept of "recommendations" discussed in [4].

4.1 Definition of Opinion

We denote the opinion that node i has for node m by $P_{i,m}$. If node i and m are neighbors, the opinion that i has for m is set equal to the trustworthiness value, $T_{i,m}$, that node i has for m. Recall from section 3 that trustworthiness can be computed from first-hand observation data. In this case, the opinion is said to be *first-hand*. If node i and m are not neighbors, neither node can accumulate first-hand information about the other node's packet forwarding behavior. In order for node i to form an opinion about node m, it can make use of the trustworthiness values computed by neighbor nodes within the network. Here, the opinion is said to be *second-hand*.

To define the concept of second-hand opinion, we first define the notation of trustworthiness along a path. Suppose that i and m are neighbors of each other. Let R denote a path from i to m defined by

$$R = \{i = a_0, a_1, a_2, \cdots, a_{n-1}, a_n = m\},$$

where $n \geq 2$. In case the trustworthiness values associated with each of the links in R are at least T_{def}, we define the trustworthiness of path R simply as their product. However, suppose that one of the links, say (a_{j^*}, a_{j^*+1}) has an associated trustworthiness value less than T_{def}. Then we may infer that node a_{j^*} views node a_{j^*+1} as "untrustworthy." Hence, from the viewpoint of node a_{j^*}, the trustworthiness values reported by node a_{j^*+1} are irrelevant. This motivates the following general definition for the trustworthiness of path R:

$$j^* \triangleq \min\{\arg\min_{1\leq j\leq n-2}\{T_{a_j,a_{j+1}} < T_{def}\}, n-1\}. \tag{8}$$

$$T_R \triangleq T_{a_0,a_1} \cdot \prod_{j=0}^{j^*} T_{a_j,a_{j+1}} \cdot (T_{def})^{n-j^*-1}. \tag{9}$$

Given the concept of trustworthiness along a path, we define the opinion as the maximum trustworthiness value along all paths from the source to the destination node, assuming at least one path exists. If no path exists, we simply assign an opinion value of T_{def}. More formally, let $\mathcal{R}_{i,m}$ denote the set of paths from node i to node m. We define the opinion that node i has for node m as follows:

$$P_{i,m} = \begin{cases} T_{i,m}, & i \text{ and } m \text{ are neighbors,} \\ \max_{R\in\mathcal{R}_{i,m}} T_R, & \mathcal{R}_{i,m} \neq \emptyset, \\ T_{def}, & \text{otherwise.} \end{cases} \tag{10}$$

4.2 Computing Second-hand Opinions

When node i and m are not neighbors, the value of $P_{i,m}$ is obtained by computing the maximum value of the trustworthiness values with respect to each path from i to m. This computation can be carried out using a shortest path algorithm by defining a suitable set of edge weights for the network. Define the weight of the link from a node a to a neighbor node b as follows:

$$w_{a,b} \triangleq -\log(T_{a,b}), \tag{11}$$

where $T_{a,b}$ is the trustworthiness value that node a has for node b, computed using first-hand information. Note that since $T_{a,b} \in (0,1)$, the value of $w_{a,b}$ must be nonnegative.

PROPOSITION 1. *If i and m are not neighbors, and at least one path exists between them, then*

$$P_{i,m} = \exp(-d_{i,m}), \tag{12}$$

where $d_{i,m}$ is the length of the shortest path from i to m.

PROOF. The weight of a path $R = \{i, a_1, \cdots, a_n, m\}$ in the network is defined as the sum of the weights of the edges in the path:

$$w_R \triangleq w_{i,a_1} + w_{a_1,a_2} + \cdots + w_{a_{n-1},a_n} + w_{a_n,m}. \tag{13}$$

The length of the shortest path from node a to b is then given by

$$d_{i,m} \triangleq \min_{R\in\mathcal{R}_{i,m}} w_R. \tag{14}$$

Now it can easily be verified that

$$P_{i,m} = \exp(-d_{i,m}).$$

□

The mapping of the opinion computation to a shortest path problem is illustrated in Fig. 4. In MANETs, the computation can be performed in a distributed manner using a Bellman-Ford type algorithm [3]. Furthermore, the computation can be "piggybacked" relatively easily onto distance vector routing protocols such as AODV [16].

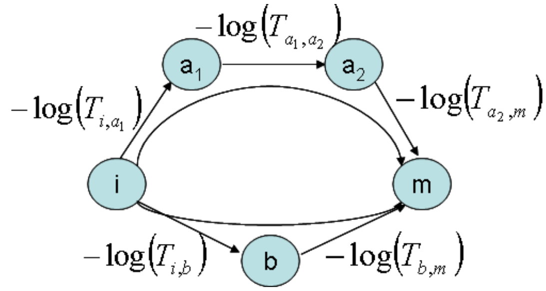

Figure 4: Opinion computation as a shortest path problem.

5. EVIDENCE ACCUMULATION

In the Hermes framework, a given node collects first-hand observation data with respect to each of its neighbors. The accumulation of observation data depends on the type of routing algorithm in place. We discuss how observation data can be collected in the case of source routing and distance vector routing. We also propose windowing mechanisms to systematically expire old observation data in order to maintain the responsiveness of the system.

5.1 MAC Layer Assumptions

The acknowledgements (ACKs) of the MAC layer (which are optional) are used to verify the successful reception of a packet through the wireless channel and address the hidden terminal problem. The MAC layer ACKs are sent by the destination to notify the source that the sent packet has been received. When a MAC layer ACK is not received, the source has to resend the unacknowledged packet.

5.2 Accounting for Malicious Behavior

5.2.1 Failure to Forward Packets

A given node X on a path forwards packets to the next or downstream node Y. Suppose that node Z is the next node after node Y on the path. Due to the broadcast nature of the wireless medium, node X could determine, for each packet it forwards to node Y, whether node Y fails to forward the packet to node Z. In order to do this, the MAC layer of a node must be modified to forward all received frames to the network layer. In this case, the node is said to be operating in *promiscuous* mode. Thus, node X should process, at the network layer, any packet received at the MAC layer from the wireless interface, whether or not node X is the MAC-level destination of the packet.

In our proposed scheme for accumulating observation data, each node operates in promiscuous mode. When a given node on a route, say node X, forwards a packet p to the next hop, say node Y, it increments a counter, $C_{X,Y}$, by one and starts a timer. The timeout value should be larger than the round-trip delay between node X and Y. If node X sees a packet from node Y that matches the packet p within the timeout period, then node X is assured that node Y correctly forwarded packet p to the next hop (i.e., node Z) and increments a counter, $F_{X,Y}$. Otherwise, if the timeout period expires, node X assumes that node Y did not forward packet p on to node Z. We point out that the penultimate node in the route, i.e., the node immediately upstream from the destination node D, does not expect node D to forward packets and hence does not follow this procedure. We limit our scheme to gathering statistics only for packets that a node has forwarded itself to ensure that valid information is recorded. On a route of n nodes (including the source and destination nodes), the first $n-2$ nodes accumulate evidence for their downstream nodes.

Note that the set of active traffic flows traversing node X and the neighbor set of node X change over time. Therefore, node X can potentially accumulate packet delivery statistics for every other node in the network. The set of values $C_{X,y}$ and $F_{X,y}$ for all other nodes y in the network forms a table of packet delivery statistics, which can be used to compute the first-hand trust and confidence values $t_{X,y}$ and $c_{X,y}$, respectively, according to the Bayesian framework discussed in section 3.1. The pair $(t_{X,y}, c_{X,y})$ can then be mapped to a trustworthiness value $T_{X,y}$, as discussed in section 3.3.

5.2.2 Misrouting of Packets

Node X could determine, for each packet it forwards to node Y, whether node Y correctly forwards the packet on to node Z. In order to do this, node X must operate in promiscuous mode. When node X, forwards a packet p to the next hop node Y, it increments the counter, $C_{X,Y}$, by one and starts a timer. The timeout value should be larger than the round-trip delay between node X and Y. If node X sees a packet from node Y that matches the packet p sent to node Z within the timeout period, node X is assured that node Y correctly forwarded packet p to the next hop and increments the counter $F_{X,Y}$. Otherwise, if the timeout period expires or if the packet was not forwarded to node Z, node X does not increment the counter $F_{X,Y}$. All nodes except the the penultimate and destination nodes follow this procedure.

5.2.3 Injection of Packets

A node injects packets when it sends new packets into the network and attributes them to a flow of another node. When a secure routing algorithm is implemented, it is impossible for a node to inject packets. Thus, a node cannot attempt the following attack: drop the legitimate packets and inject new packets in order to let its upstream node believe that it forwarded the packets it received for forwarding. In case the secret key of a node is compromised, packets can be injected by that node. This issue is beyond the scope of the Hermes framework.

5.3 Routing Protocol Considerations

In source routing protocols, e.g., DSR [12], each datagram at the network layer contains the entire list of nodes in the route from the source to the destination. Therefore, a node X can recognize whether its downstream node Y correctly forwards a packet p to Y's downstream node Z. Node X operates in promiscuous mode. When node X receives a packet q sent from node Y within the timeout period, node X examines the packet by comparing the source route listed in the datagram header with that of packet p as well as the destination field in MAC header to determine whether the packet is sent to the correct next hop. In case the received packet q matches packet q and is sent to node Z, node X

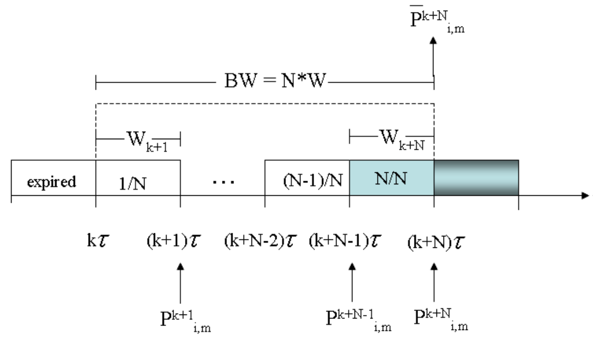

Figure 5: Averaging and observation windows.

is assured that node Y correctly forwarded packet p to the next hop.

In distance vector routing algorithms such as AODV [16], the header of a data packet contains information about the next hop and the number of remaining hops to the destination. Upon receiving a data packet, a node overwrites the next hop field and decreases the number of hops left to the destination by one. The observation scheme for source routing discussed above does not work for distance vector routing because a node that sends a packet to its downstream node for forwarding cannot determine whether the packet will indeed be forwarded, as the upstream node's identity does not appear in the new header of the packet.

A simple and efficient solution is to employ sequence numbers at the network layer to identify each data packet during the data forwarding phase. By checking the sequence number, a given node X can then verify whether its downstream neighbor node Y correctly forwarded a given packet p that was sent earlier by X. Nonetheless, node X does not know the identity of the node downstream from node Y. Therefore, node X can only see whether its downstream neighbor node Y correctly forwarded a given packet p, but does not know whether node Y misroutes the packet p to another node V. In this case, node V is responsible for forwarding the packet towards its destination. Thus, the packet might traverse a longer route to the destination node. A malicious node could misroute packets to a colluding node that drops the packets. We are currently investigating alternative ways of accumulating empirical evidence for AODV in order to avoid this type of attack.

5.4 Observation and Averaging Windows

We define an observation window over which a given node i collects first-hand observation data from its neighbor node j. At the end of the kth observation window, denoted by W_k, the trustworthiness value $T^k_{i,j}$, of node i for node j is calculated using the observations from W_k. We assume that each observation window is of length τ. Given the trustworthiness values $T^k_{i,j}$, the set of opinion values corresponding to window W_k, i.e., $\{P^k_{i,m}\}$ for any node m, can be computed. The computation of $P^k_{i,m}$ is assumed to take an additional τ_P time units after window W_k ends, during the first part of window W_{k+1}.

We present a sliding windowing mechanism to systematically expire old observation data in order to improve the accuracy of the opinion metric and maintain the responsiveness of the system. We introduce a sliding averaging window

BW_k, consisting of the N most recent observation windows, i.e.,

$$BW_k = \{W_{k-N+1}, W_{k-N+2}, \cdots, W_{k-1}, W_k\}. \quad (15)$$

The length of BW_k is $N\tau$ time units. During the averaging window BW_k, N opinion values are computed for each pair of nodes i and m (see Fig. 5):

$$P_{i,m}^{k-N+1}, P_{i,m}^{k-N+2}, \cdots, P_{i,m}^{k-1}, P_{i,m}^{k} \quad (16)$$

which correspond to the N observation windows contained in BW_k. We calculate a weighted average of the N opinion values computed during the window BW_k to obtain an *averaged opinion value*, $\bar{P}_{i,m}^k$. By applying a simple linear weighted averaging scheme, we define the averaged opinion at time k that node i has for node m as follows:

$$\bar{P}_{i,m}^k \triangleq \frac{2}{N(N+1)} \sum_{l=1}^{N} l P_{i,m}^{k-N+l}. \quad (17)$$

We remark that other averaging schemes, e.g., exponential averaging windows, may also be used to define the averaged opinion.

The proposed windowing scheme expires old observation data in a systematic manner. It is possible that during BW_k less observation data is accumulated from first-hand observations than during BW_{k-1}. The averaged opinion value $\bar{P}_{i,m}^k$ depends on the number of observation data collected in BW_k (see section 3). The use of the averaged opinion metric improves the stability of the system, since past information is taken into account.

6. TRUST-AWARE ROUTING

In this section, we discuss the application of the Hermes trust establishment framework to improve the reliability of packet forwarding in MANET routing protocols in the presence of malicious nodes.

6.1 Definition of Routing Opinion

Given a source node s, a destination node d, and a path $R = \{s, i_1, \cdots, i_n, d\}$ from s to d, we define the "routing opinion" that node s has for the route R as follows:

$$V_R \triangleq (\bar{P}_{s,i_1} \cdot \bar{P}_{s,i_2} \cdot \cdots \cdot \bar{P}_{s,i_{n-1}} \cdot \bar{P}_{s,i_n})^{1/n}. \quad (18)$$

According to (18), the routing opinion of s along route R is a function of the product of the (averaged) opinions that node s has for each node on the path R, except for the destination node d. The reason that $\bar{P}_{s,d}$ is not included in the product is that when node s chooses to communicate with node d, it implicitly trusts node d. The selection of a route entails a choice of intermediate nodes, not including node d, that lie on a path to d. In the definition (18) of routing opinion, the exponent $1/n$ is included in order to avoid excessively penalizing longer routes.

6.2 Route Selection

Given a source node s, a destination node d, a path $R = \{s = a_0, a_1, a_2, \cdots, a_{n-1}, a_n = d\}$, where $n \geq 2$, from s to d, link l of route R, $l \in R$, weight w of link l (see equation (11)), and the set of paths $\mathcal{R}_{s,d}$ from node s to node d, we define the route R^* on which node s chooses to send its data packets to destination node d as follows:

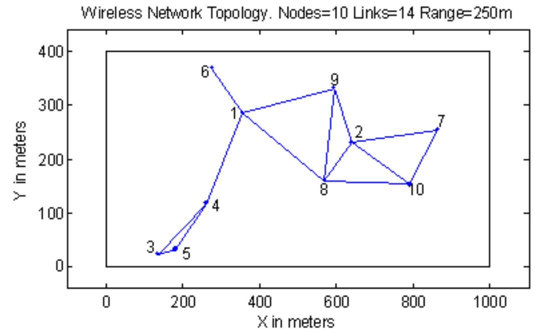

Figure 6: Wireless network topology.

$$w^* \triangleq \max_{R \in \mathcal{R}_{i,m}} \min_{l \in R} w_l. \quad (19)$$

$$\mathcal{C}_{s,d} = \{R \in \mathcal{R}_{s,d} : \min_{l \in R} w_l = w^*\} \quad (20)$$

$$R^* \triangleq \arg \max_{R \in \mathcal{C}_{s,d}} V_R \quad (21)$$

According to equation (21), node s chooses to send its data packets to destination node d on the route of maximum routing opinion, which is chosen among the routes in the set $\mathcal{C}_{s,d}$. The set $\mathcal{C}_{s,d}$ consists of the routes associated with the max-min link weight among all routes from s to d. The rationale for this is that any intermediate link on a route can be point of failure. Finally, the route of maximum routing opinion, among the routes of the maximum of the minimum link weight, is chosen by source node s to send its data packets to destination nodes d.

7. NUMERICAL RESULTS

The Hermes scheme was implemented and evaluated in Matlab. We present three simulation scenarios. The network topology shown in Fig. 6 is used for the simulations. Fourteen wireless links are formed among ten nodes that are randomly placed in a 1000 m by 400 m area. The wireless radio transmission range of the nodes is set to 250 m.

7.1 Trust, confidence, and trustworthiness

In the first simulation scenario, one traffic flow is established in the network from node 5 to node 7, along the path $\{5, 4, 1, 8, 2, 7\}$. Intermediate nodes 4, 1 and 8 forward 90% of the packets that they should be forwarding, whereas node 2 forwards only 20% of the packets received for forwarding. Node 5 sends 20 data packets during each observation window or "round" W.

Fig. 7 shows the trust and confidence values, $(t, c)_{5,4}$, that node 5 places on node 4 after 0, 1, 3, 10, and 30 windows, based on the direct observations of node 5. We note that node 5 forms a correct opinion about node 4, i.e., $(t, c) = (0.85, 0.75)$, even after a single round. Observe that the more observations node 5 makes for node 4, the more confident node 5 becomes about the trust value it assigns to node 4.

Fig. 8, 9, and 10 show the opinion values over time that node 5 places on node 4, 2 and 3, respectively, for different trustworthiness parameters, x and y. Node 4 is a "good" node, since it forwards 90% of the packets that should be

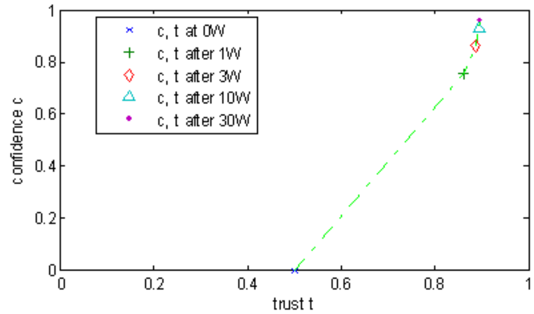

Figure 7: Confidence vs. trust of node 5 with respect to node 4 after 0, 1, 3, 10 and 30 windows.

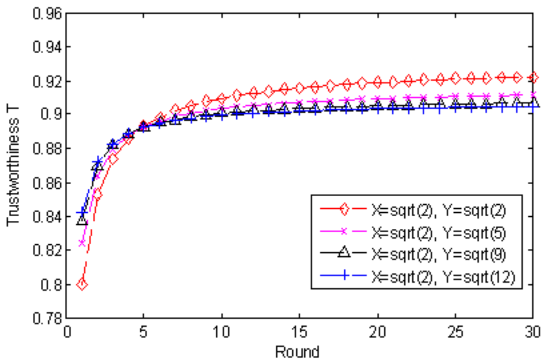

Figure 8: Opinion value $P_{5,4} = T_{5,4}$ for different trustworthiness parameter values.

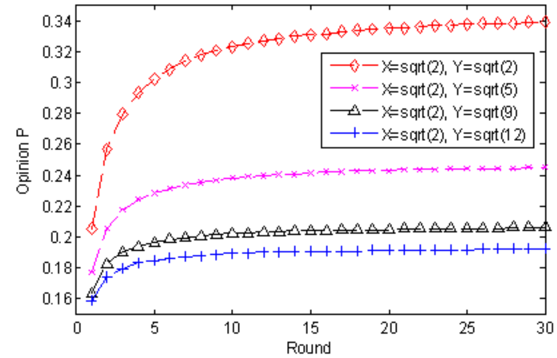

Figure 9: Opinion value $P_{5,2}$ for different trustworthiness parameter values.

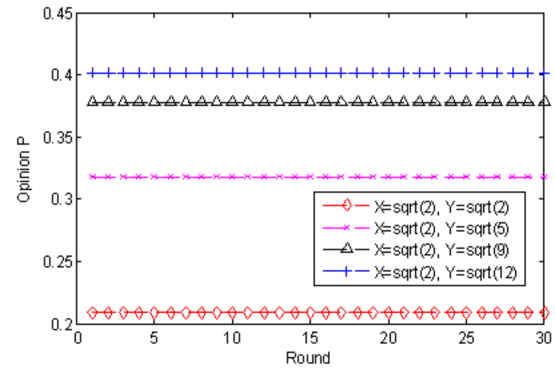

Figure 10: Opinion value $P_{5,3}$ for different trustworthiness parameter values.

forwarded. Node 2 is a "bad" node, since it forwards only 20% of the packets that should be forwarded. Node 5 has never interacted with node 3 and is ignorant about its behavior.

The simulation results show that the most appropriate values for the trustworthiness parameters are $x = \sqrt{2}$ and $y = \sqrt{9}$. Note that node 5 correctly assigns a trustworthiness value of 0.90 to node 4 and an opinion value of 0.20 to node 2 even after a small number of windows. A trustworthiness value of 0.38 is assigned to node 3 (see Fig. 10) and to all other nodes that node 5 is ignorant about. When the trustworthiness parameters are chosen as $x = y = \sqrt{2}$ (i.e., the ellipse becomes a circle), node 5 places an unreasonably high opinion value on node 2 and an unreasonably low trustworthiness on node 3. Note that when the trustworthiness parameters are set to $x = \sqrt{2}$ and $y = \sqrt{12}$, node 5 penalizes nodes 4 and 2 more than it should. We have found that the above observations concerning the parameter values x and y do not depend strongly on the actual trustworthiness values. Hence, we use parameter values $x = \sqrt{2}$, $y = \sqrt{9}$ to map trust and confidence to trustworthiness values in the remaining simulation experiments to be discussed. In practice, the parameters x and y could be tuned to the needs of a particular application.

7.2 Calculation of opinion

In order to demonstrate Hermes's ability to adapt to changes in the node behaviors, we simulate the network topology of Fig. 6 with the same flow as before, i.e., node 5 sends 20 packets per window for 30 windows. However, in the present scenario, the intermediate node 4 forwards 90% of the packets sent by node 5 in each window, node 8 forwards 90% of the first 100 packets sent to it and 20% of the remaining packets sent to it. Finally, intermediate node 2 exhibits malicious behavior by forwarding only 20% of the packets it receives. The trustworthiness parameters are set as follows: $x = \sqrt{2}$ and $y = \sqrt{9}$.

The opinions that node 5 places on the intermediate nodes over 30 windows when the window size is 20 is shown in Fig. 11. From Fig. 11, we can make the following observations:

1. Node 5 correctly computes an opinion for node 4 of value $P_{5,4} = T_{5,4} = 0.91$. The opinion node 5 has for node 4 is based on the direct observations of its packet forwarding behavior.

2. Node 5 computes an opinion for node 1 of value $P_{5,1} = 0.82 = T_{5,4} \cdot T_{4,1}$.

3. Node 5 detects the change in the behavior of node 8. At the end of window 5, node 5 calculates an opinion for node 8 of value $P_{5,8} = 0.75$. From window 6 onwards, the opinion value $P_{5,8}$ drops to 0.23. The change in the node behavior of node 8 is detected within one window.

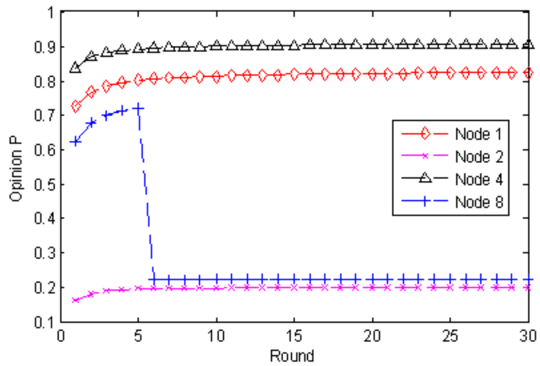

Figure 11: Opinion values P that node 5 places on the intermediate nodes when $W = 20$.

4. Up until the fifth window, node 5 considers node 8 "trustworthy" ($P_{5,8} = 0.75$) and accepts its recommendations for node 2. As a result, node 5 correctly assigns an opinion value of $P_{5,2} = 0.22$ to node 2, which always exhibits malicious behavior. From window 6 onwards, node 5 assigns a small opinion value to node 8, and does not accept its recommendations. The opinion value node 5 has for node 2 remains at 0.22.

5. Node 5 assigns the correct opinion values to the intermediate nodes after a single observation window.

7.3 Routing Opinion

In the third simulation scenario, five traffic flows are established in the network as follows:

- flow 1 along the path $\{7, 2, 8, 1, 4, 5\}$;
- flow 2 along the path $\{3, 4, 1, 6\}$;
- flow 3 along the path $\{4, 1, 8, 10\}$;
- flow 4 along the path $\{5, 4, 1, 9, 2\}$;
- flow 5 along the path $\{10, 2, 9, 1, 4, 3\}$.

Node 9 acts maliciously, forwarding only 20% of the packets it should be forwarding. All other nodes forward 90% of the packets they should be forwarding. The source node of each flow sends 20 packets per window over the course of 30 rounds.

Fig. 12 illustrates the opinion values, $P_{i,j}$, that node i places on node j with a gray-scale representation. A black color implies an opinion value of 0, whereas white represents an opinion value of 1, while intermediate values are represented by different shades of gray. Fig. 13 shows the corresponding numerical opinion values. One can verify that the source and intermediate nodes of these 5 flows have formed the correct opinion about the other nodes. Recall that node 9 is malicious, and is part of flows 4 and 5. Nodes upstream from node 9 in these two flows nodes, i.e., nodes 5, 4, 1, 10, and 2, have formed the correct opinion for it. The corresponding cells of the ninth column of Fig. 12 are the darker. The cells of value $T_{def} = 0.3784$ correspond to links between nodes that have never interacted.

We now investigate three different routing scenarios described as follows:

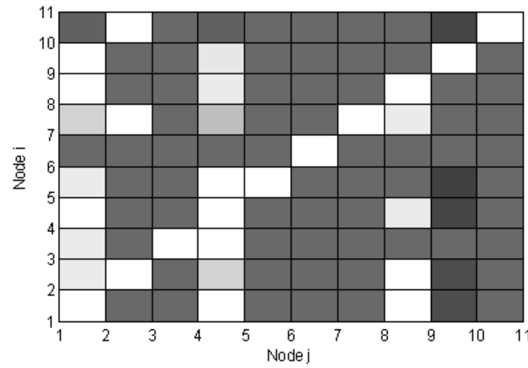

Figure 12: Opinion values $P_{i,j}$ in gray-scale.

Node i \ Node j	1	2	3	4	5	6	7	8	9	10
1	NaN	0.3784	0.3784	0.9071	0.3784	0.3784	0.3784	0.9076	0.2759	0.3784
2	0.8235	NaN	0.3784	0.7470	0.3784	0.3784	0.3784	0.9076	0.2759	0.3784
3	0.8228	0.3784	NaN	0.9066	0.3784	0.3784	0.3784	0.3784	0.3784	0.3784
4	0.9076	0.3784	0.3784	NaN	0.3784	0.3784	0.3784	0.8228	0.2504	0.3784
5	0.8228	0.3784	0.3784	0.9066	NaN	0.3784	0.3784	0.3784	0.2270	0.3784
6	0.3784	0.3784	0.3784	0.3784	0.3784	NaN	0.3784	0.3784	0.3784	0.3784
7	0.7465	0.9066	0.3784	0.6772	0.3784	0.3784	NaN	0.8228	0.3784	0.3784
8	0.9073	0.3784	0.3784	0.8231	0.3784	0.3784	0.3784	NaN	0.3784	0.3784
9	0.9001	0.3784	0.3784	0.8093	0.3784	0.3784	0.3784	0.3784	NaN	0.3784
10	0.3431	0.9066	0.3784	0.3431	0.3784	0.3784	0.3784	0.3784	0.2501	NaN

Figure 13: Opinion values $P_{i,j}$ in numerical values.

1. Node 2 does not initially start a session, but has been an intermediate node for one of the five previous flows. Then node 2 requests a route to node 1. The implemented protocol finds two possible routes: $R_1 = \{2, 9, 1\}$ and $R_2 = \{2, 8, 1\}$. From Fig. 13, we can see that node 2 has formed opinions for nodes 9 and 8 already. Node 2 calculates, using equation (18), the routing opinion values $V_{R_1} = (P_{2,9})^{1/1} = 0.28$ and $V_{R_2} = (T_{2,8})^{1/1} = 0.91$. The route with the highest routing opinion is chosen to route the packets to the destination node 1. Thus, node 2 successfully avoids the route that includes the malicious node 9.

2. Node 4 has established a session already. Now, node 4 requests a route to node 2. The routing protocol finds two possible routes: $R_1 = \{4, 1, 8, 2\}$ and $R_2 = \{4, 1, 9, 2\}$. From Fig. 13, we can see the opinions that node 4 has formed for nodes 1, 8, and 9. Node 2 calculates the following routing opinion values, using equation (18): $V_{R_1} = (P_{4,1} \cdot P_{4,8})^{1/2} = (0.91 \cdot 0.82)^{1/2} = 0.86$, $V_{R_2} = (P_{4,1} \cdot P_{4,9})^{1/2} = (0.91 \cdot 0.25)^{1/2} = 0.47$. Thus, route R_1 is selected to route packets from node 4 to node 2. This choice of routes successfully avoids the route that contains the malicious node 9.

3. Node 10 requests a route to node 9. The routing protocol finds two possible routes: $R_1 = \{10, 8, 9\}$ and $R_2 = \{10, 2, 9\}$. From Fig. 13, node 2 calculates the following routing opinion values, using equation (18): $V_{R_1} = (P_{10,8})^{1/1} = 0.38$ and $V_{R_2} = (P_{10,2})^{1/1} = 0.91$. In this case, route R_2 is selected.

8. CONCLUSION

We presented Hermes, a quantitative trust establishment framework for MANETs, which is designed to improve the reliability of packet forwarding over multi-hop routes in the presence of potentially malicious nodes. The framework defines two metrics, trust and confidence, which are computed using a Bayesian approach based on empirical first-hand observations of packet forwarding behavior by neighbor nodes. The trust and confidence metrics are mapped into a single "trustworthiness" metric, which can be tuned to the needs of the application by means of two parameters. The concept of trustworthiness is extended to the notion of an opinion that a given node has for any arbitrary node. The opinion metric can be incorporated into MANET routing protocols to improve the reliability of packet delivery. A windowing scheme for expiring old observation data improves the fidelity of the opinion metric.

Simulation results demonstrated the effectiveness of the Hermes framework in distinguishing among malicious and non-malicious nodes as well as in the selection of more "trustworthy" routes for reliable packet delivery. In ongoing work, we are investigating extensions to the Hermes framework in order to deal with the behavior of malicious nodes that selectively drop packets or propagate invalid trustworthiness information. We also plan to investigate the implementation of Hermes to realized trust-aware routing based on ad hoc routing protocols such as DSR and AODV.

Acknowledgment

The authors would like to acknowledge Prof. Kris Gaj for helpful discussions related to this work. This work was supported in part by the U.S. National Science Foundation under Grant No. CCR-0209049.

9. REFERENCES

[1] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. Symp. on Network and Distributed Systems Security (NDSS)*, 2002.

[2] J. Baras and T. Jiang. Cooperative Games, Phase Transition on Graphs and Distributed Trust in MANET. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, June 2004.

[3] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, Englewood Cliffs, New Jersey, 2 edition, 1992.

[4] S. Buchegger and J.-Y. L. Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In *Proc. 2nd Workshop on Economics of Peer-to-Peer Systems*, June 2004.

[5] L. Buttyan and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *Mobile Networks and Applications*, 8(5):579–592, 2003.

[6] L. Capra. Engineering Human Trust in Mobile System Collaborations. In *Proc. 12th ACM SIGSOFT Int. Symp. Foundation of Software Eng.*, pages 107–116, 2004.

[7] L. Eschenauer, V. D. Gligor, and J. Baras. On trust establishment in mobile ad-hoc networks. In *Proc. Security Protocols Workshop*, volume 2845, pages 47–66. LNCS, April 2002.

[8] M. Guerrero. Secure AODV. *ACM Mobile Computing and Communications Review*, 6(3), August 2002.

[9] Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proc. ACM MobiCom '02.* ACM SIGMOBILE, September 2002.

[10] T. Jiang and J. S. Baras. Ant-based Adaptive Trust Evidence Distribution in MANET. In *Proc. 2nd Int. Workshop on Mobile Distributed Computing*, March 2004.

[11] T. Jiang and J. S. Baras. Autonomous Trust Establishment. In *Proc. 2nd Int. Network Optimization Conf.*, 2005.

[12] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[13] R. K. Nekkanti and C. Lee. Trust based adaptive on demand ad hoc routing protocol. In *Proc. 42nd ACM Southeast Regional Conference*, pages 88–93, 2004.

[14] P. Papadimitratos and Z. J. Haas. Secure message transmission in mobile ad hoc networks. *Elsevier Ad Hoc Networks Journal*, 1(1), Jan/Feb/March 2003.

[15] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York, 1991.

[16] C. Perkins, E. Belding-Royer, and S. Das. Ad-hoc On-demand Distance Vector (AODV) Routing. *IETF RFC 3561*, July 2003.

[17] A. A. Pirzada and C. McDonald. Establishing trust in pure ad-hoc networks. In *Proceedings of the 27th Australasian Computer Science Conference (ACSC04)*, pages 47–54, January 2004.

[18] G. Theodorakopoulos and J. S. Baras. Trust Evaluation in Ad-hoc Networks. In *Proceedings of the 2004 ACM workshop on Wireless Security (WiSe '04)*, pages 1–10, 2004.

[19] M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya. Quantifying Trust in Mobile Ad-Hoc Networks. In *Proc. Int. Conf. Integration of Knowledge Intensive Multi-Agent Systems (KIMAS)*, 2005.

[20] L. Zhou and Z. J. Haas. Securing Ad Hoc Networks. *IEEE Networks Special Issue on Network Security*, November 1999.