Journal of Computer Security 00 (2006) 1–36 1
IOS Press

Hermes: A quantitative trust establishment framework for reliable data packet delivery in MANETs

Charikleia Zouridaki [a], Brian L. Mark [a,*], Marek Hejmo [a] and Roshan K. Thomas [b]

[a] *Department of Electrical and Computer Engineering, George Mason University, MS 1G5, Fairfax, VA 22030, USA*
E-mail: {czourida,mhejmo,bmark}@gmu.edu
[b] *SPARTA, Inc., 5875 Trinity Parkway, Suite 300, Centreville, VA 20120, USA*
E-mail: roshan.thomas@sparta.com

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. Using a Bayesian framework, each node assigns a "trustworthiness" value to each of its neighbor nodes based on direct observations of packet forwarding behavior. More generally, each node forms an "opinion" about each of the other nodes in the network, based on the set of trustworthiness values computed in 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.

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

*Corresponding author: Tel.: 703-993-4069; Fax: 703-993-1601.

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

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 [20]. 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 [20] 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 [9]. In [9], 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. The trustworthiness metric is used to formulate the more general "opinion" metric, which can be incorporated into routing protocols in a transparent manner. The opinion metric can also be applied as a criterion for access control in MANETs [10].

The main contribution of the paper is a quantitative scheme for establishing trust in an ad hoc network. The key components of the Hermes framework 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 maintain the responsiveness of the opinion metric; (4) a method 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. 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. An overview of the Hermes trust establishment framework is provided in Section 3. Section 4 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 5 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 6. In Section 7, the security properties of Hermes are discussed in terms of a probabilistic attacker model. The application of the opinion metric to realize "trust-aware" ad hoc routing is discussed in Section 8. Section 9 presents results from simulation experiments, which demonstrate the key properties of the proposed trust establishment scheme. Finally, the paper is concluded in Section 10.

2. Related work

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 [20]. A key difference, however, is that our framework incorporates the notion of statistical confidence associated with a trust value. In [9], the notion of confidence, together with a semi-ring approach to evaluate trust and confidence along network paths, was proposed. In our approach, however, the trust and confidence metrics are mapped to a new metric, called "trustworthiness", which can more transparently be incorporated into network decisions such as route selection. Furthermore, the Hermes framework addresses the practical issue of collecting of evidence from the network.

In [1], 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 in [6] is to bootstrap secure wireless communications via pre-authentication over a location-limited channel. As in [1], trust evaluation is based only on direct first-hand information.

The authors of [14] present a high-level framework for generation, revocation and distribution of trust evidence and demonstrate the significance of estimation metrics in trust establishment. The authors 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 [22] along the lines suggested in [14]. In contrast, our work focuses on developing metrics and mechanisms for establishing trust with respect to the objective of reliable packet delivery. In [19], 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 [12], 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 [13], 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 [11,23], which relies on the introduction of pre-trusted agents and a public key infrastructure. In [16], a trust framework is proposed for the purpose of establishing a set of group keys.

3. Overview of Hermes

In this section, we provide a high-level, conceptual overview of the Hermes trust establishment framework as illustrated in Fig. 1.

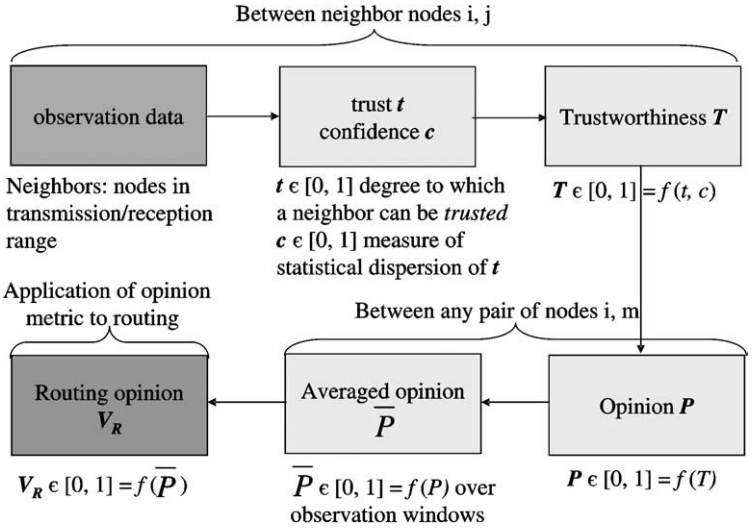

Fig. 1. Overview of Hermes.

3.1. Basic definitions and terminology

Two nodes are said to be *neighbors* if they are in the radio transmission and reception range of each other. Due to the broadcast nature of the wireless medium, a given node can collect first-hand information about the packet forwarding behavior of its neighbor nodes by snooping all received frames at the MAC layer and recording packet delivery statistics. The accumulation of observation data is discussed in Section 6. The packet statistics can be used to compute a pair of values associated with each neighbor node, which we call *trust* and *confidence*.

We define *trust* as a value in the range [0, 1], which indicates the degree to which a given node believes that its neighbor node behaves normally or can be "trusted" to forward the packets it receives for forwarding. The trust value is a statistic computed from first-hand observation data. A value of trust close to 0 implies that the neighbor is likely to be unreliable in forwarding packets, whereas a trust value close to 1 implies that the neighbor is likely to be reliable in its packet forwarding behavior. We define the *confidence*, c, associated with a given trust value t, as a value in the range [0, 1], which indicates a measure of statistical dispersion in the trust value. The pair of values (t, c) characterizes the degree of belief that the neighbor is reliable with respect to packet delivery and the statistical confidence in this belief. As discussed in Section 4, we apply a Bayesian framework to calculate trust and confidence values.

While it is useful to characterize empirical trust via the pair of values (t, c), it is often more convenient to characterize trust in terms of a single value in order to make decisions or take action with respect to the neighbor. Therefore, we define the concept of a *trustworthiness* value T, which is a function of the pair (t, c). The value

of T also lies in the range [0, 1]. The issue of how to map (t, c) into T is discussed in Subsection 4.3.

Thus far, we have defined the notions of trust, confidence, and trustworthiness only with respect to a node and its neighbor. We now define a more general notion called *opinion*. A given node i can formulate an opinion for any other node m in the network. If nodes i and m are neighbors, then the opinion that node i has for m equals the trustworthiness of node i for m. Otherwise, if i and m are not neighbors, the opinion that i has for m is determined as a function of the trustworthiness values along network paths from i to m.

3.2. Additional notation

Suppose that node i collects observation data related to the packet forwarding behavior of its neighbor j over a time window W. At the end of the window W, node i computes a pair of trust and confidence values with respect to j, which are denoted by $t_{i,j}$ and $c_{i,j}$, respectively. The pair of trust and confidence values can be mapped to a trustworthiness value denoted by $T_{i,j}$. The *opinion* that i has for another node m is denoted by $P_{i,m}$. In case $m = j$, we have that $P_{i,m} = T_{i,j}$. Otherwise, the computation of $P_{i,m}$ requires the propagation of trustworthiness values along network paths from i to m as discussed in Section 5.

More generally, the trust values are updated at the end of observation windows belonging to a sequence. Let W_k denote the kth time window. We denote by $t^k_{i,j}$ and $c^k_{i,j}$, the trust and confidence values that node i has for node j, respectively, computed at the end of window W_k. Similarly, $P^k_{i,m}$ denotes the opinion that node i has for node m at the end of W_k. An *averaged* opinion $\bar{P}^k_{i,m}$ can be calculated based on the opinion values computed in previous time windows (see Subsection 6.4). To apply the notion of option to improve routing, the notion of *routing opinion*, V_R, is defined to represent the opinion that a source node s has for a route R, and is used to select the most trustworthy route among a set of alternative routes from the source to the destination.

3.3. Authentication requirements

The Hermes framework can be applied, in principle, to any MANET. We remark that cryptographic primitives should be employed 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. Many of the same cryptographic mechanisms proposed for secure routing in MANETs [15,18,21,26] can also be applied to Hermes. In particular, Hermes could be deployed in conjunction with a secure routing protocol that already provides such mechanisms.

4. Trust evaluation from first-hand observations

In order to establish trust, raw data or observations must be accumulated from the network and transformed into numerical trust values. The accumulation of evidence from the observations is discussed in Section 6. In this section, we describe our approach to computing trust given a set of observations obtained from the network. Our approach is based on representing trust in terms of a probability distribution, in particular, the Beta distribution, which is updated using a Bayesian framework. The use of a Bayesian framework was also proposed in [20], but our approach to trust establishment is different in that two separate, but complementary trust metrics are considered: trust and confidence. Further, our approach addresses the issue of incorporating both trust and confidence into trust-based decisions by combining these metrics into a single metric, called *trustworthiness*.

4.1. Bayesian framework

In the Bayesian framework, 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 $R_1, R_2, \ldots$, 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 on 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 [2] as follows:

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

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

$$f(M_k = m|r, N_k = n) = \binom{n}{m} r^m (1 - r)^{n-m}. \tag{2}$$

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:

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

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 the system initiation (at 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.

4.2. 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 Dirac function, the more confidence is placed in the trust value t^k. A 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 ($0 \leqslant \mu < 0.5$). The associated confidence value is $c^0 = 0$ according to (5). Figure 3, shows how confidence grows when the number of observations grows, for different values of the parameters a and b of the Beta distribution.

4.3. 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. In this case, the trustworthiness metric should closely reflect the trust value, since the statistical confidence in this value is high. On the other hand, the smaller the value of c, the less the value of t should be considered as valid. If the value of confidence is low, then the reliability of the trust value should correspondingly be low. In this case, the trustworthiness metric should reflect the degree of uncertainty implied by the confidence value. Therefore, 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 be weighted less than the confidence value c.

Figure 2 shows that the set of (t, c) values lies in the unit square region defined by $0 \leqslant t \leqslant 1$ and $0 \leqslant c \leqslant 1$. 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.

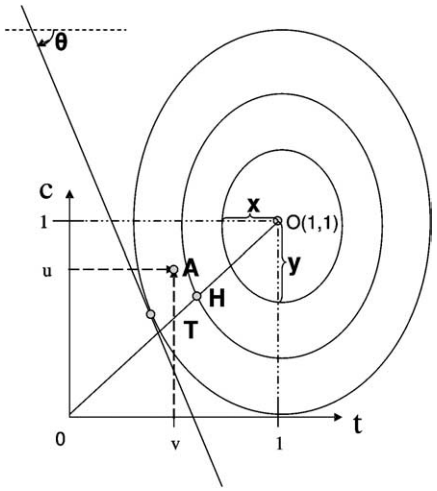

Fig. 2. Relationship between trustworthiness T and the trust-confidence pair (t, c).

Figure 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 defined by:

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

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 the confidence value has greater weight than the trust value for the majority of points on the ellipse.
- 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.

Intuitively, the parameters should be chosen such that $x < y$ in order to define an appropriate mapping from (t, c) to T. From Fig. 3, we see that after a sufficient number of observations, the value of confidence c grows to a large value. In this range of values of c, the trustworthiness value T should be determined primarily by the value of t. Ultimately, the goal of the Hermes scheme is to distinguish between two classes of nodes: (1) "good" nodes, which forward packets reliably; and (2) "bad" nodes, which do not forward packets reliably, together with nodes for which insufficient statistical evidence is available. As discussed above, the mapping (7) provides a concrete rule for combining the notions of trust and confidence into a single metric to allow a node to decide whether another node is "good". The performance of the proposed mapping in terms of convergence, false alarm, and missed detection rates is investigated through computer simulations in Section 9. We also remark that the choice of (x, y) could also be left up to the node that computes the trustworthiness value of the other node. As will be discussed in Section 8, each source node could implement its own policy in determining the trustworthiness of a path by making its own choice of (x, y).

5. Formulation of opinions

The concept of trustworthiness as defined in the previous section applies only to two neighbor nodes i and j. Due to the characteristics of the wireless access medium,

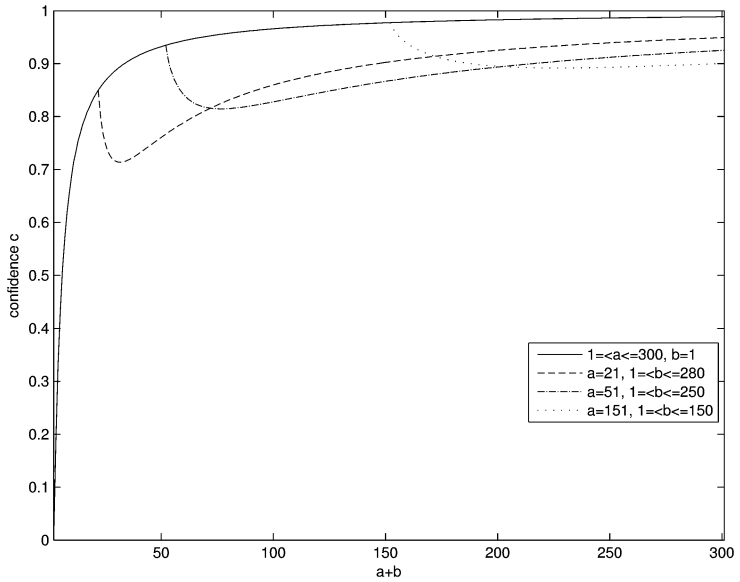

Fig. 3. Confidence as a function of number of observations.

node i can observe the packet forwarding behavior of node j and then compute the trustworthiness value $T_{i,j}$. However, for the purposes of routing or other network-related decisions, node i may need to form an opinion about an arbitrary node m, which may not be a neighbor of i. Therefore, 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 [20].

5.1. 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 4 that trustworthiness can be computed from first-hand observation data. 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.

Assume that i and m are not neighbor nodes. Let $R = \{i = a_0, a_1, a_2, \ldots, a_{n-1}, a_n = m\}$, where $n \geqslant 2$, denote a path from node i to node m. We can extend the concept of trustworthiness between two neighbor nodes to trustworthiness along a path from one node to another node. More precisely, we define the trustworthiness

of path R as follows:

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

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

If $T_{a_j,a_{j+1}} \geqslant T_{def}$, node a_j makes use of the trustworthiness values propagated by node a_{j+1}. Otherwise, node a_j simply sets the trustworthiness values of the downstream nodes to the value T_{def}. Thus, the Hermes scheme correlates the reliability of a node for trust propagation with its trustworthiness for packet forwarding. The impact of doing this on the security properties of Hermes is discussed in Section 7.4.

As an example of computing the trustworthiness of a path R according to (9), suppose that node i receives trustworthiness values $T_{i,a_1}, T_{a_1,a_2}, \cdots, T_{a_{n-2}}$, all of which exceed the value T_{def} and that $T_{a_{n-2},a_{n-1}} < T_{def}$. In this case, node i ignores the trustworthiness value T_{a_{n-1},a_n} that node a_{n-1} has sent. Then, the trustworthiness value T_R of the path R is computed as

$$T_R = T_{i,a_1} \cdot T_{a_1,a_2} \cdots T_{a_{n-2},a_{n-1}} \cdot (T_{def})^1.$$

Let $\mathcal{R}_{i,m}$ denote the set of paths from node i to node m. If the set $\mathcal{R}_{i,m}$ is empty, we define $P_{i,m} = T_{def}$, where T_{def} is the default trustworthiness value assigned to a node, when its assigned trust and confidence values are $t^0 = 0.5$ and $c^0 = 0$, respectively. That is, $T_{def} \triangleq T(0.5, 0)$. Otherwise, we define the opinion that node i has for node m by

$$P_{i,m} \triangleq \max_{R \in \mathcal{R}_{i,m}} T_R. \tag{10}$$

Since node i is not a neighbor of node a_j, it has to rely on the trustworthiness values computed by other nodes in order to form its own opinion about a_j. That is, node i must use *second-hand information* to form an opinion about m, as can be seen from (10). An example of opinion calculation for non-neighbors i and m is illustrated in Fig. 4. In this example, all trustworthiness values are greater than T_{def}, i.e., $T_{i,a_1}, T_{a_1,a_2}, T_{a_2,m}, T_{i,b}, T_{b,m} \geqslant T_{def}$. We now provide a definition for the opinion that any node i has for another node m as follows:

$$P_{i,m} = \begin{cases} T_{i,m}, & i \text{ and } m \text{ are neighbors,} \\ \max_{R \in \mathcal{R}_{i,m}} T_R, & \mathcal{R}_{i,m} \neq \emptyset, \\ T_{def}, & \text{otherwise.} \end{cases} \tag{11}$$

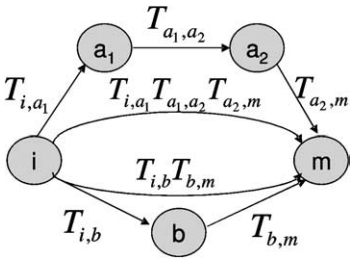

Fig. 4. Example of opinion calculation for non-neighbors i and m.

5.2. Opinions from second-hand trustworthiness

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:

$$w_{a,b} \triangleq -\log(T_{a,b}), \tag{12}$$

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*

$$P_{i,m} = \exp(-d_{i,m}), \tag{13}$$

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

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

$$w_R \triangleq w_{i,a_1} + w_{a_1,a_2} + \cdots + w_{a_{n-1},a_n} + w_{a_n,m}. \tag{14}$$

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

$$d_{i,m} \triangleq \min_{R \in \mathcal{R}_{i,m}} w_R. \tag{15}$$

Now it can easily be verified that

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

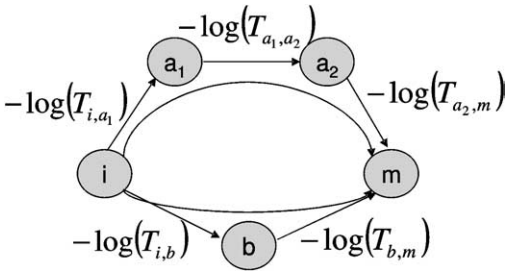

Fig. 5. Opinion computation as a shortest path problem.

The mapping of the opinion computation to a shortest path problem is illustrated in Fig. 5. In MANETs, the computation can be performed in a distributed manner using a Bellman-Ford type algorithm [7]. Furthermore, the computation can be "piggybacked" relatively easily onto distance vector routing protocols such as AODV [4].

5.3. Propagation of trustworthiness

The trustworthiness values can be carried by the control packets of the routing algorithm in order to avoid additional communication overhead. Therefore, the propagation of the trustworthiness values depends on the routing algorithm in place. Section 8 discusses how the trustworthiness values are exchanged in the signaling phases of the DSR and AODV routing algorithms, which are representative examples of source routing and distance vector algorithms, respectively.

6. Accumulation of empirical evidence

In the Hermes framework, a given node collects first-hand observation data with respect to each of its neighbors. The first-hand observations are used to compute trust and confidence values, which are in turn mapped to trustworthiness values. Recommendations are used by the given node to form opinions about a non-neighbor nodes. 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.

6.1. Physical and MAC layer assumptions

At the physical layer, we assume that the nodes are equipped with omni-directional antennas and that they transmit at a constant power level, i.e., no power control is used. We shall assume that acknowledgements (ACKs) at the MAC layer 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 hop to notify the source hop that the sent packet has been received. When a MAC layer ACK is not received, the source hop has to resend the unacknowledged packet.

6.2. Accounting for malicious behavior

6.2.1. Forwarding 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 correctly forwards the packet on 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 source 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.

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 4.1. The pair $(t_{X,y}, c_{X,y})$ can then be mapped to a trustworthiness value $T_{X,y}$, as discussed in Section 4.3.

In our proposed scheme for accumulating packet observation data, nodes maintain packet counters corresponding only to packets they have actually forwarded. The scenario of Fig. 6 illustrates a potential problem that may arise if a node attempts to accumulate information on packets that it did not forward. Here, node I wishes to send a packet to node F along the path $\{I, E, F\}$. Suppose that node E behaves correctly and forwards the packet to node F. In promiscuous mode, node A will hear the packet that node E forwards (since node E is in node A's radio reception

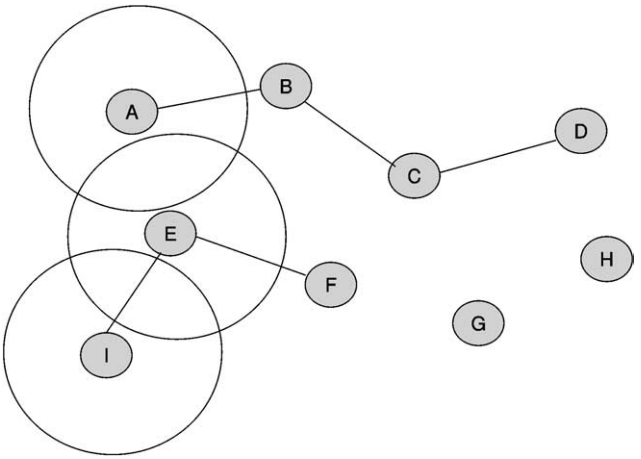

Fig. 6. Example topology showing active flows on paths $\{I, E, F\}$ and $\{A, B, C, D\}$.

range), but will not know how many packets node E received for forwarding, since node I is not in node A's radio reception range. Therefore, 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.

6.2.2. Misrouting 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 $F_{X,Y}$. We point out that the penultimate node in the route, i.e., the node immediately upstream from the destination node D, does not expect the destination node D to forward packets. Hence, the penultimate node does not follow this procedure.

6.2.3. Injecting 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 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. However, this issue is beyond the scope of this paper.

6.3. Routing protocol considerations

In source routing protocols, e.g., DSR [8], each datagram at the network layer contains the list of nodes in the entire route from the source to the destination. Therefore, a node X can recognize whether its downstream node Y correctly forwards a packet p on to its downstream node Z. Node X operates in promiscuous mode. When node X sees a packet sent from node Y within the timeout period, node X checks the packet by looking at the source route listed in the datagram's header to see whether the packet matches packet p, and the destination field listed in the frame's header to see whether the packet is sent to the correct next hop. In case the packet matches packet p, and is sent to node Z, node X is assured that node Y correctly forwarded packet p to the next hop.

In distance vector routing algorithms such as AODV [4], 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 presented earlier for source routing 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 which is the downstream node of 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 node V. In this case, node V is responsible of 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. In the future, we plan to investigate alternative ways of accumulating the empirical evidence for AODV in order to avoid this type of attack.

6.4. 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 $T_{i,j}^k$, 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 $T_{i,j}^k$, the set of opinion values corresponding to

window W_k, i.e., $\{P^k_{i,m}\}$ for any node m, can be computed. As indicated in Fig. 7, 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 propose 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. The sliding averaging window BW_k consists of the N most recent observation windows, i.e.,

$$BW_k = \{W_{k-N+1}, W_{k-N+2}, \cdots, W_{k-1}, W_k\}. \tag{16}$$

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. 8):

$$P^{k-N+1}_{i,m}, P^{k-N+2}_{i,m}, \cdots, P^{k-1}_{i,m}, P^k_{i,m}, \tag{17}$$

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

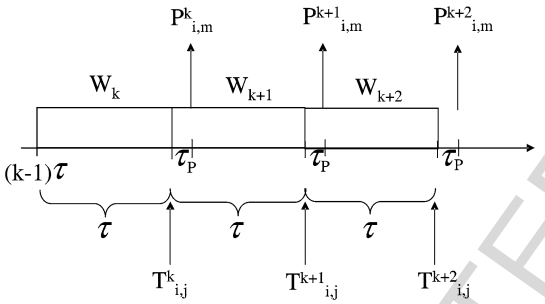

Fig. 7. Sequence of observation windows.

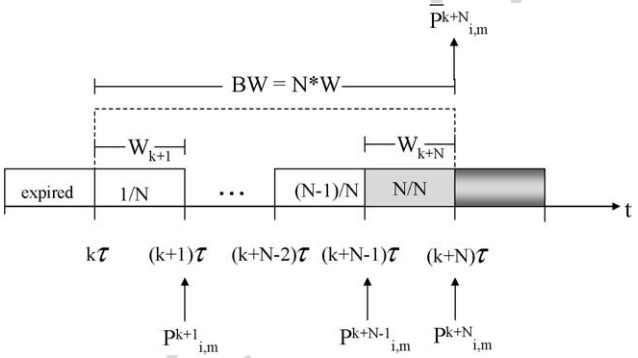

Fig. 8. Averaging window.

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:

$$\bar{P}_{i,m}^{k} \triangleq \frac{2}{N(N+1)} \sum_{l=1}^{N} l P_{i,m}^{k-N+l}. \tag{18}$$

We remark that other averaging schemes, e.g., exponential averaging windows, may also be used to define the averaged opinion. The use of averaging improves the stability of the opinion metric, since past information is taken into account.

7. Attacker model and security properties of Hermes

7.1. Attacker models for MANETs

Most of the attacker models for MANETs discussed in the literature are presented in the context of secure routing [17,24,27]. Attacks against routing protocols can be classified into passive and active attacks. The primary example of a passive attack is eavesdropping of routing control packets. Examples of active attacks against a routing protocol include: black or grey holes, creation of routing loops, wormhole attacks, gratuitous detour attacks (i.e., making a route appear shorter than it is), corrupting packets, fabricating packets, replaying packets, reordering packets, misrouting packets, and impersonating another user.

Many of the aforementioned attacks can be addressed using cryptographic mechanisms. For example, message authentication codes (MACs) can be applied to protect the integrity of control packets. The use of cryptography can also prevent impersonation attacks. Other attacks on routing can be mitigated using sequence numbers. For instance, the numbering of route request packets can be used to avoid the creation of routing loops. The route request and route reply phases of a routing protocol can be used to mitigate attacks such as black and grey holes and packet misrouting. Other mechanisms, such as packet leashes [25] have been proposed to address attacks such as wormhole attacks.

7.2. Attacks addressed by Hermes

The Hermes framework is intended to avoid a class of attacks on packet delivery in MANETs during the data transmission phase rather than the route discovery phase. The attacks on Hermes are considered to be committed by "insider" nodes who have succeeded in becoming part of active routes in the network. Such nodes are owners of valid cryptographic keys or key materials and are capable of authenticating themselves as authorized users of the network. Furthermore, these nodes have successfully passed the route discovery phase of a secure routing protocol.

As mentioned earlier, the integrity of message exchanges involved in the Hermes protocol should be protected by cryptographic primitives such as those used in secure routing protocols. We shall assume that this is the case and focus on insider attacks on packet forwarding and the propagation of trust information. The main attacks on packet forwarding to be considered in the attacker model include dropping, misrouting, and replaying data packets. As in secure routing protocols, sequence numbers can be used in conjunction with Hermes to avoid replay attacks. The main focus of the Hermes scheme lies in detecting packet dropping and misrouting attacks.

We remark that packet forwarding attacks can be launched even when a secure routing protocol is in place. A secure routing protocol aims to establish a route from a source node to a destination node containing only authorized or insider nodes. Once a route is established, nodes on the path are expected to forward packets correctly to the next hop. However, during the data transmission phase an insider node may consistently drop, misroute, or replay packets. The Hermes scheme attempts to identify such misbehaviors in terms of the trustworthiness and opinion metrics, but does not purport to distinguish between malicious or non-malicious misbehaviors. Non-malicious packet forwarding misbehavior may be due to such phenomena as network congestion, node mobility, or node malfunction.

To compute the opinion metric for non-neighbor nodes, the Hermes scheme relies on the exchange of trustworthiness information among nodes. Thus, an obvious attack on the Hermes scheme would be for nodes to propagate false trustworthiness information. In the current Hermes scheme, the trust that a node X has in the trustworthiness information propagated by another node Y is simply correlated with the trust that node X has for node Y with respect to packet forwarding. Consequently, an attack that cannot be resolved by the Hermes scheme is one in which node Y propagates false trust information to node X, yet forwards packet correctly. An extension to Hermes, which avoids such an attack is discussed in [5].

7.3. Probabilistic attacker model

The attack space covered by the Hermes scheme can be described more formally in terms of a probabilistic attacker model. The attacker model consists of two types of attacks: (1) incorrect data packet forwarding; (2) incorrect propagation of trust information. Note that we do not distinguish among the various types of data packet forwarding misbehaviors, i.e., packet dropping, misrouting, and replay attacks. Incorrect trust propagation refers to a node which propagates a trustworthiness value that is different from the value that it should compute if it were following the Hermes scheme. Thus, a node may propagate a trustworthiness value that is higher or lower than the value that a Hermes-compliant node would compute.

Let $\mathcal{N}$ denote the set of all nodes in the network. A network attack scenario, in steady-state, is specified by characterizing, for each node $i \in \mathcal{N}$, the probability B_f^i that the node performs incorrect packet forwarding and the probability B_t^i that the

node performs incorrect trust propagation, where $0 \leqslant B_f^i, B_t^i \leqslant 1$. More precisely, the network attack scenario can be represented by a set of three-tuples,

$$\mathcal{S} = \{(i, B_f^i, B_t^i) : i \in \mathcal{N}\}. \tag{19}$$

Let η_f and η_t denote, respectively, thresholds on the degrees of packet forwarding and trust propagation misbehaviors that can be tolerated in the network (e.g., one could set $\eta_f = \eta_t = 0.5$). A given node i can then be identified as an *attacker* if $B_f^i > \eta_f$ or $B_t^i > \eta_t$ (or both).

In the context of the Hermes scheme, which is primarily concerned with packet forwarding behavior, a *good* node is one for which $B_f \leqslant \eta_f$, whereas a bad node is one for which $B_f > \eta_f$. The Hermes scheme aims to identify the set of probabilities $\{B_f^i\}$ to a sufficient degree of accuracy to distinguish between good and bad nodes, based on both first-hand information from direct observations of packet forwarding behavior and second-hand information from other nodes. The probabilistic attacker model does not preclude the possibility that nodes may collude with one another. However, the Hermes scheme does not seek to identify collusions per se. Rather, the Hermes scheme is able to characterize the *effect* of a colluding attack as represented by an attack scenario (19).

7.4. Security properties of Hermes

The attacker model presented above is simple, but sufficient to characterize the main security properties of the Hermes scheme. Under Hermes, the opinion metrics $P_{i,m}$ should closely approximate the underlying attack scenario under steady-state conditions. That is, in steady-state we should have

$$P_{i,m} \approx 1 - B_f^m, \quad \text{for all } i \in \mathcal{N}. \tag{20}$$

For the Hermes framework to correctly distinguish the good nodes from the bad nodes, it is sufficient that

$$P_{i,m} \geqslant T_{def} \text{ if and only if } B_f^m < \eta_f, \quad \text{for all } i \in \mathcal{N} \tag{21}$$

hold in steady-state. The simulation results presented in Section 9 provide validation of the steady-state properties (20) and (21).

We point out that the current Hermes framework assumes that the trust propagation behavior of a given node is correlated with its packet forwarding behavior in the following sense. According to (9), the trustworthiness values propagated by a node i to a neighbor node j are effectively ignored by node j if the trustworthiness value $T_{j,i}$ is less than the default value T_{def}. Consequently, in terms of the probabilistic

attacker model, the Hermes scheme can correctly infer packet forwarding behavior under the following assumption:

$$B_f^i \geqslant B_t^i, \quad \text{for all } i \in \mathcal{N}. \tag{22}$$

In other words, the probability with which a node commits a packet forwarding misbehavior must not exceed the probability with which it commits a trust propagation misbehavior. To relax assumption (22), the Hermes scheme could be extended to maintain an opinion metric with respect to trust propagation, but such an extension is beyond the scope of the current paper.

Under the probabilistic attacker model and the assumption (22), the Hermes scheme is able to distinguish the good nodes from the bad nodes in a network scenario with high accuracy, as demonstrated through the simulation results presented in Section 9. In steady-state, the performance of the Hermes scheme with respect to a compliant node does not depend on the set of probabilities $\{B_f^i : i \in \mathcal{N}\}$ or the locations of the nodes. However, the convergence of the scheme is a function of the availability of first-hand observation data, which depends on the distribution and traffic volume of flows in the network. The probabilistic attacker model only characterizes steady-state behavior. To accommodate dynamic changes in the network attack in practice, the proper use of windowing as discussed in Section 6.4 is necessary to maintain the responsiveness of the Hermes scheme.

8. 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. First, we define the concept of "routing opinion", which is used to select among a set of alternative routes from a source to a destination node. Then we briefly discuss how the Hermes framework can be incorporated in the well-known MANET routing protocols DSR and AODV, respectively.

8.1. Definition of routing opinion

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

$$V_R \triangleq (\bar{P}_{s,i_1} \cdot \bar{P}_{s,i_2} \cdots \bar{P}_{s,i_{n-1}} \cdot \bar{P}_{s,i_n})^{1/n}. \tag{23}$$

This definition is illustrated in Fig. 9. According to (23), 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,

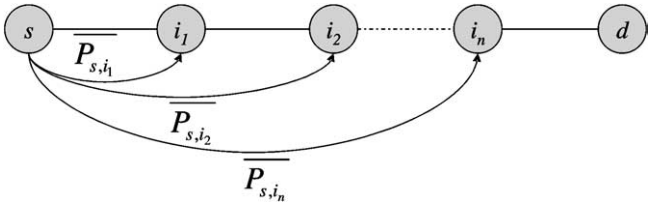

Fig. 9. Calculation of routing opinion.

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 (23) of routing opinion, the exponent $1/n$ is included in order to avoid excessively penalizing longer routes.

8.2. Route selection

Given a source node s, a destination node d, a path $R = \{s = a_0, a_1, a_2, \ldots, a_{n-1}, a_n = d\}$, where $n \geqslant 2$, from s to d, link l of route R, $l \in R$, trustworthiness T of link l, 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:

$$T^* \triangleq \max_{R \in \mathcal{R}_{i,m}} \min_{l \in R} T_l, \tag{24}$$

$$\mathcal{C}_{i,m} = \{R \in \mathcal{R}_{i,m} : \min_{l \in R} T_l = T^*\}, \tag{25}$$

$$R^* \triangleq \arg \max_{R \in \mathcal{C}_{i,m}} V_R. \tag{26}$$

According to equation (26), 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 of the maximum of the minimum link trustworthiness. 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 trustworthiness, is chosen by source node s to send its data packets to destination nodes d. Alternatively, the routing opinion metric could be used to choose the route probabilistically, i.e., a route would be chosen with probability proportional to the routing opinion. Such a randomized routing scheme would improve the performance of the Hermes scheme, as the flows would traverse a more diverse set of nodes, providing a richer set of first-hand observation data for computing the trust metrics. The competitive adaptive routing scheme proposed by Awerbuch et al. [3], employs a type of randomized routing except that the routing decisions are performed on a packet-by-packet basis rather than a per-flow basis.

8.3. Trust-aware DSR

DSR is a reactive ad hoc routing protocol based on source routing [8]. To incorporate the Hermes framework into DSR, each node computes trustworthiness values with respect to each of its neighbors. The routing opinion V_R that a node i has for a route R to node m is only computed when i receives the route reply (RREP) message sent by m in response to the route request (RREQ) message sent from node i to node m. Corresponding to each RREQ message, the destination sends a RREP message. As the RREP message propagates to the source i along a path R, it accumulates the link weights defined in Eq. (12) maintained by the nodes along the path. The source node i then determines its routing opinion, V_R, for route R to node m using Eq. (23). As specified by DSR, each node stores multiple paths to each destination. Our scheme requires the addition of a field indicating the trustworthiness of each route. Node i chooses to send its data packets to destination node m on the route R^* defined by Eq. (26) (see Section 8.2).

Recall from Eq. (23) that in order to calculate the routing opinion along a path from node i to node m, the opinion values for the intermediate hops are required. The computation of these opinion values is defined in Eq. (13). According to Eq. (23), the averaged opinion values, $\bar{P}_{i,m}$, should be used to compute the routing opinion, rather than the current opinion values, $P_{i,m}$. Since the source node can compute the opinion values, $P_{i,m}$ only when it receives a RREP message, it is problematic for the source node to compute the averaged opinion values. An alternative approach is for each of the nodes in the network to compute *averaged* trustworthiness values, $\bar{T}_{i,j}$, which are averaged over the past N observation windows, similarly, to the averaged opinion values. Then the averaged opinion value could be computed according to the following modified definition (cf. (10)):

$$\bar{P}^k_{i,m} \triangleq \max_{R \in \mathcal{R}_{i,m}} \bar{T}_R, \tag{27}$$

where (cf. (9))

$$T_R \triangleq \bar{T}_{a_0,a_1} \cdot \prod_{j=1}^{j^*} \bar{T}_{a_j,a_{j+1}} \cdot (T_{def})^{n-j^*-1}, \tag{28}$$

and (cf. (18))

$$\bar{T}^k_{i,j} \triangleq \frac{2}{N(N+1)} \sum_{l=1}^{N} l T^{k-N+l}_{i,j}. \tag{29}$$

Each source node can implement its own policy in determining the routing opinion of a path by making its own choice of (x, y) trustworthiness parameters (see

Section 4). The trustworthiness parameters can be piggybacked in the RREQ messages. Corresponding to each RREQ, the destination sends a RREP containing the trustworthiness parameters send by the source node in the RREQ. Each intermediate node should use the given trustworthiness parameters in computing the averaged trustworthiness values $\bar{T}_{i,j}$ of their downstream node.

8.4. Trust-aware AODV

AODV is a reactive ad hoc routing protocol based on distance vector routing [4]. As discussed in Section 5.2, the averaged trustworthiness values $\bar{T}_{i,j}$ can be propagated to calculate the averaged opinion values by means of a Bellman-Ford or distance-vector type algorithm. Since AODV is based on distance-vector routing, the propagation of the averaged trustworthiness values $\bar{T}_{i,j}$ can easily be incorporated into AODV via the route reply (RREP) messages. The averaged trustworthiness values $\bar{T}_{i,j}$ are computed as defined by Eq. (29). Then, each node i can compute its opinion $\bar{P}_{i,m}$ for every other node m in the network. The averaged opinion values $\bar{P}_{i,m}$ are calculated as defined by Eq. (27). The source node i determines its routing opinion, V_R, for route R to node m using Eq. (23). Node i chooses to send its data packets to destination node m on the route R^* defined by Eq. (26) (see Section 8.2).

As specified by AODV, each node maintains a routing table with the next hop and the hop count to each destination. Our scheme requires the addition of a third field indicating the trustworthiness of each route. When AODV is implemented, the destination nodes respond to the first RREP received, unless one arrives along a better path. Hence, more than one RREP may reach the source node, which can calculate the trust value of the returned routes and choose to route the packets through the most trusted one. AODV specifies that each source node stores one path for each destination. AODV could be extended so that source nodes stores all the returned paths, each associated with its trust value. As discussed in the case of DSR, each source node can implement its own policy in determining the routing opinion of a path by making its own choice of (x, y) trustworthiness parameters. The trustworthiness parameters can be piggybacked in the RREQ messages.

9. Simulation results

The Hermes scheme was implemented and evaluated in Matlab. We present three simulation scenarios. The network topology shown in Fig. 10 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. We remark that the simulation scenarios considered in this paper do not take into account the effect of mobility. Mobility will generally *improve* the performance of the Hermes scheme, since nodes will have more opportunities to gain first-hand information on all other nodes in the network. Larger simulation scenarios which take into account mobility are presented in [5].

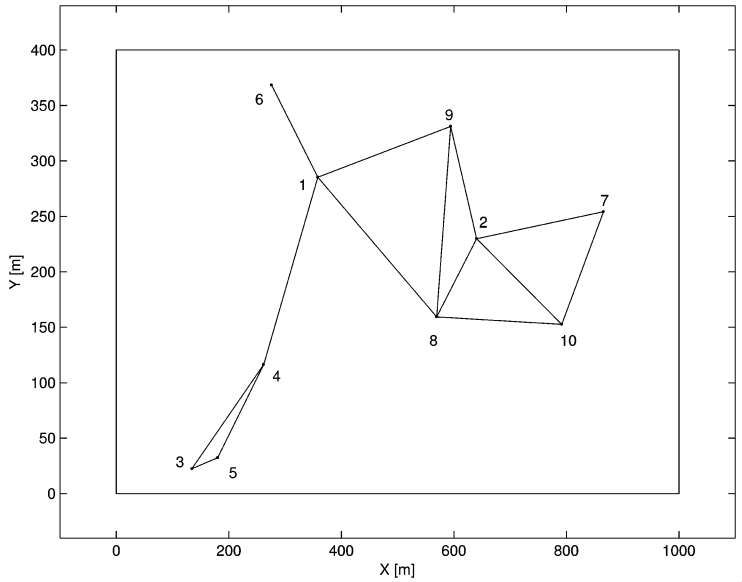

Fig. 10. Network topology used in simulation experiments.

9.1. 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 W. In terms of the probabilistic attacker model discussed in Section 7.3, the network scenario can be specified as follows:

$$B_f^i = \begin{cases} 0.1 & \text{for } i = 1, 4, 8 \\ 0.8 & \text{for } i = 2 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and } B_t^i = 0 \text{ for all } i \in \mathcal{N}, \tag{30}$$

where $\mathcal{N} = \{1, \ldots, 10\}$ is the set of all nodes in the network. We remark that the simulation results are the same for any set of $\{B_t^i\}$ satisfying assumption (22), i.e., $B_t^i \leqslant B_f^i$ for all $i \in \mathcal{N}$.

Figure 11 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[2], 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 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.

[2]We shall also refer to an observation window as a "round".

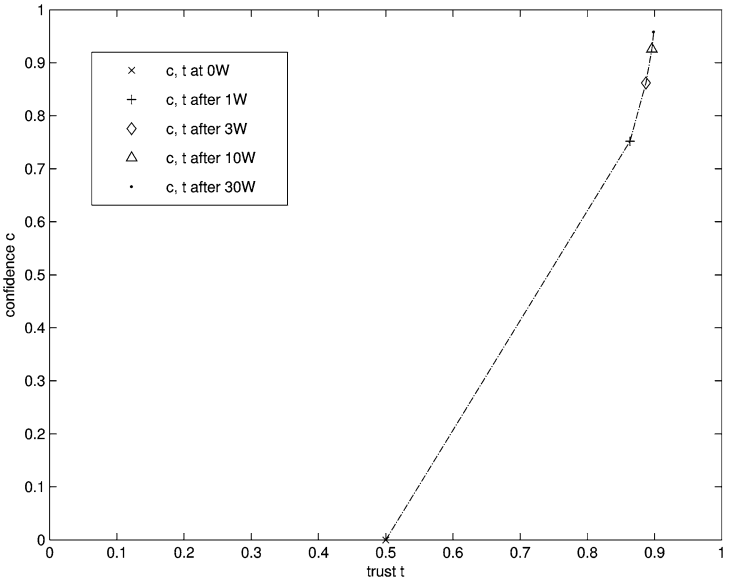

Fig. 11. $(t, c)_{54}$: (trust, confidence) node 5 places on node 4 after 0, 1, 3, 10 and 30 windows (W).

Figures 12, 13, and 14 show the opinion values over time (i.e., windows) 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 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 show that the most appropriate values for the trustworthiness parameters are $x = \sqrt{2}$ and $y = \sqrt{9}$. These parameter values will be used to map trust and confidence to trustworthiness values in our later simulations.

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 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.

9.2. 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. 10 with the same flow as before, i.e., node 5

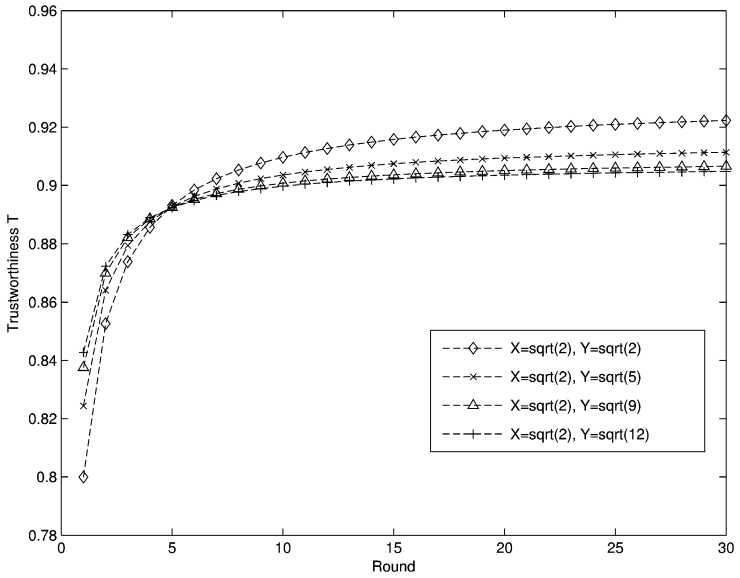

Fig. 12. Opinion/trustworthiness value $P_{5,4} = T_{5,4}$ for different trustworthiness parameter values.

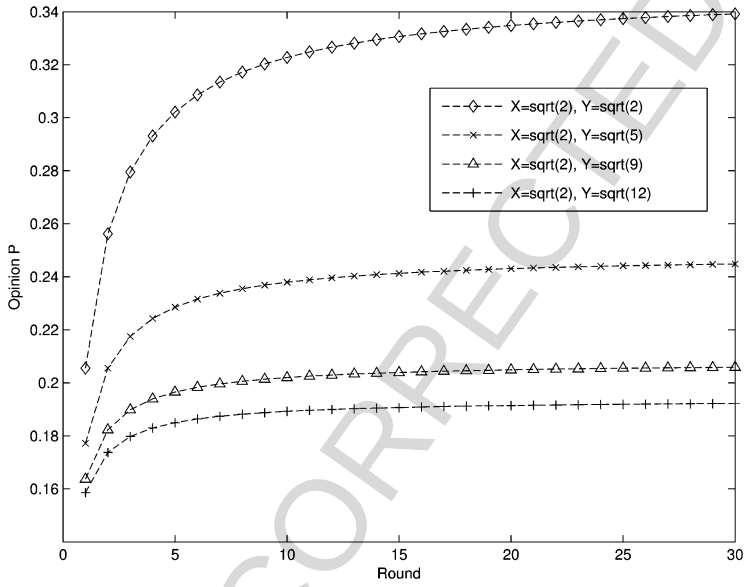

Fig. 13. Opinion value $P_{5,2}$ for different trustworthiness parameter values.

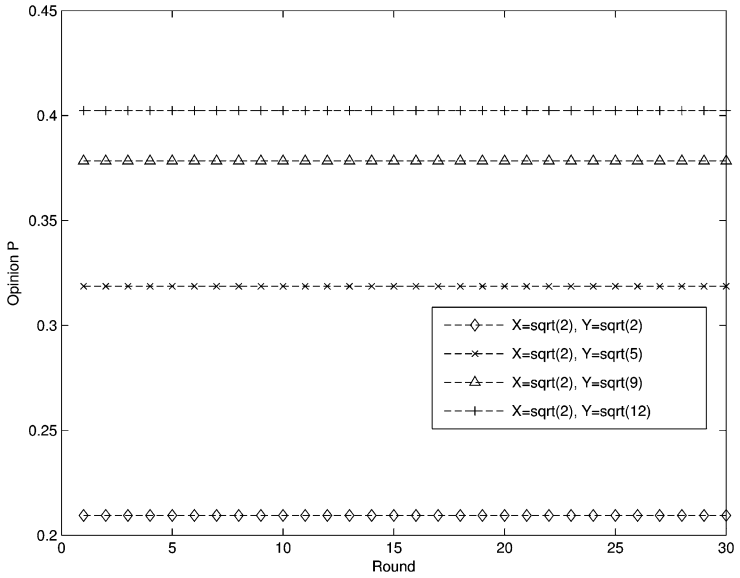

Fig. 14. Opinion value $P_{5,3}$ for trustworthiness parameter values.

sends 20 packets per window for 30 windows. However, in the present scenario, the intermediate nodes 4 and 1 forward 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 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, 50, and 100, respectively, are shown in Fig. 15, Fig. 16, and Fig. 17 respectively. From Fig. 15, 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.
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

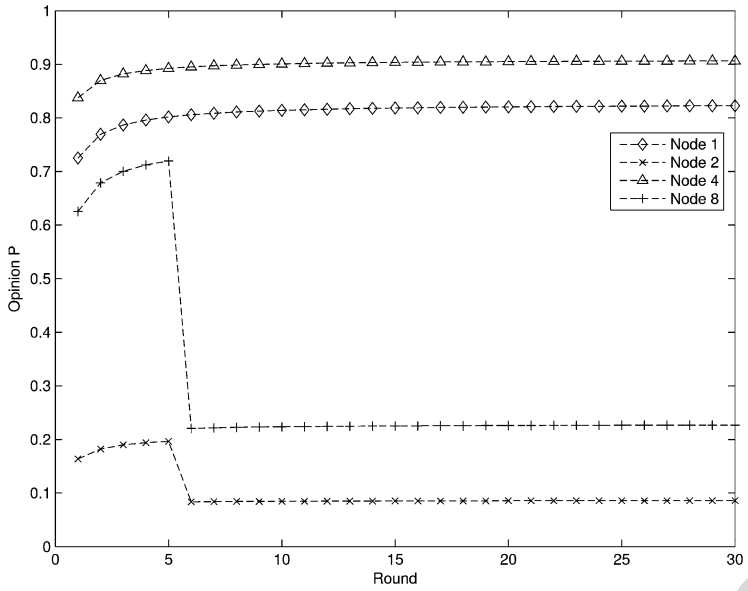

Fig. 15. Opinion values P that node 5 places on the intermediate nodes when $W = 20$.

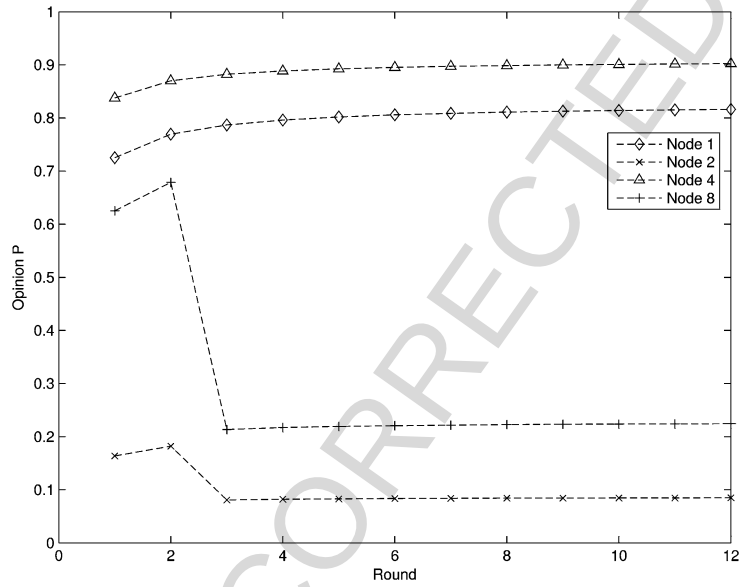

Fig. 16. Opinion values P that node 5 places on the intermediate nodes when $W = 50$.

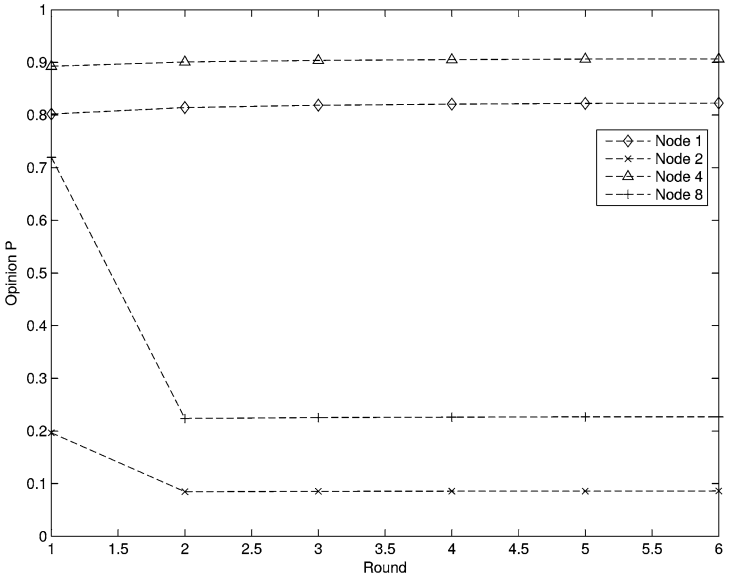

Fig. 17. Opinion values P that node 5 places on the intermediate nodes when $W = 100$.

to node 8, and does not accept its recommendations. The opinion value node 5 has for node 2 drops to 0.09, as expected.

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

Figures 16 and 17 are similar to Fig. 15 with the difference that the size of the window W is increased to 50 and 100, respectively. Hence, for the same simulation time, the trust, confidence and trustworthiness are computed fewer times, and the number of windows is smaller than in Fig. 15 even though the number of packets sent is the same, i.e., 600 packets.

In Fig. 15, 20 packets per window are sent over a span of 30 windows. At window 6, node 8 starts misbehaving. In Fig. 16, 50 packets per window are sent over a span of 12 windows. Node 8 starts misbehaving at window 2. In Fig. 17, 100 packets per window are sent for 6 rounds and node 8 starts misbehaving at window 1. As expected, the smaller the time window W: (1) the sooner a change in a node's behavior is detected and (2) the sooner the source node (in this case, node 5) computes its first opinion about the intermediate nodes. Thus, we see a tradeoff between speed of detection and processing overhead.

9.3. 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.

Figure 18 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. Figure 19 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. 18 are the darker. The cells of value 0.3784 correspond to links between nodes that have never interacted.

We now investigate three different routing scenarios described as follows:

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.

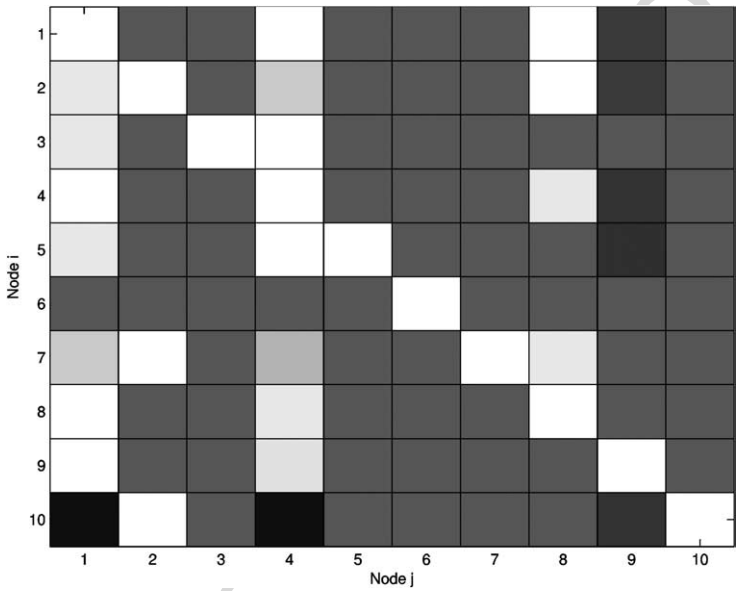

Fig. 18. Opinion values $P_{i,j}$ in gray-scale.

Node j

Node i	1	2	3	4	5	6	7	8	9	10
1	1	0.38	0.38	0.91	0.38	0.38	0.38	0.91	0.28	0.38
2	0.82	1	0.38	0.75	0.38	0.38	0.38	0.91	0.28	0.38
3	0.82	0.38	1	0.91	0.38	0.38	0.38	0.38	0.38	0.38
4	0.91	0.38	0.38	1	0.38	0.38	0.38	0.82	0.25	0.38
5	0.82	0.38	0.38	0.91	1	0.38	0.38	0.38	0.23	0.38
6	0.38	0.38	0.38	0.38	0.38	1	0.38	0.38	0.38	0.38
7	0.75	0.91	0.38	0.68	0.38	0.38	1	0.82	0.38	0.38
8	0.91	0.38	0.38	0.82	0.38	0.38	0.38	1	0.38	0.38
9	0.90	0.38	0.38	0.81	0.38	0.38	0.38	0.38	1	0.38
10	0.09	0.91	0.38	0.09	0.38	0.38	0.38	0.38	0.25	1

Fig. 19. Opinion values $P_{i,j}$ in numerical values.

The implemented protocol finds two possible routes: route $R_1 = \{2, 9, 1\}$ and $R_2 = \{2, 8, 1\}$. From Fig. 19, we can see that node 2 has formed opinions for nodes 9 and 8 already. Node 2 calculates, using Eq. (23), 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. 19, 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 Eq. (23): $V_{R_1} = (P_{4,1} \cdot P_{4,8})^{1/2} = (0.91 \cdot 0.82)^{1/2} = 0.86$, $V_{R_2} = (P_{4,1} \cdot P_{4,9})^{1/2} = (0.91 \cdot 0.25)^{1/2} = 0.47$. 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. 19, node 2 calculates the following routing opinion values, using Eq. (23): $V_{R_1} = (P_{10,8})^{1/1} = 0.38$ and $V_{R_2} = (P_{10,2})^{1/1} = 0.91$. In this case, route R_2 is selected.

In the above scenario, the Hermes scheme is able to determine the opinion metrics with sufficient accuracy to enable a choice of the best routes with respect to reliable packet delivery. We point out that the accuracy of the opinion metrics should improve further in the presence of a larger set of active flows.

10. Conclusion

We propose 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. Using a Bayesian framework, two metrics are defined: trust and confidence, which are computed based on the

empirical first-hand observations of packet forwarding behavior by neighbor nodes. Trust characterizes the belief in the reliability of a neighbor node with respect to packet forwarding. The confidence value associated with a given trust value characterizes the statistical reliability of the trust value. Trust and confidence are mapped into "trustworthiness" metric, which captures the impact of trust and confidence in a single value.

The concept of trustworthiness, defined only between neighbor nodes, is then 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 such as DSR or AODV to improve the reliability of packet delivery in a transparent manner. A windowing scheme is used to expire old observation data. The overhead imposed by the Hermes scheme is mainly computational. Nodes following the Hermes scheme collect statistics based on first-hand observations of packet transmissions on the wireless broadcast channel and compute the trust metrics. The communication overhead due to the propagation of second-hand trust information can be minimized by piggybacking trust information onto the routing control packets.

We remark that the Hermes scheme may stimulate selfish nodes not to forward packets in order to conserve battery power. A scheme for punishing such nodes is beyond the scope of the current Hermes scheme. The objective of the Hermes scheme is to distinguish the subset of nodes that can reliably be characterized as being "good" from among the set of all nodes. A scheme to avoid selfish behavior should incorporate a method of selectively punishing bad behavior. This is an interesting topic for further research.

A simple probabilistic attacker model was proposed to characterize the security properties of Hermes. Our simulation experiments demonstrate the effectiveness of the Hermes framework in distinguishing among bad and good nodes as well as in the selection of more "trustworthy" routes for reliable packet delivery. In [5], we investigate extensions to the Hermes framework to deal with the behavior of nodes that propagate invalid trustworthiness information. In ongoing work, we are studying alternative approaches to evaluating the trustworthiness metric from first-hand network observation data.

Acknowledgement

This work was supported in part by the US National Science Foundation under Grant No. CCR-0209049 and Grant No. ACI-0133390.

References

[1] A.A. Pirzada and C. McDonald, Establishing trust in pure ad-hoc networks, in: *Proc. 27th Australasian Computer Science Conference (ACSC04)*, 2004, pp. 47–54.

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

[3] B. Awerbuch, D. Holmer, R. Kleinberg and H. Rubens, Provably competitive adaptive routing, in: *Proc. IEEE INFOCOM*, Miami, FL, 2005.

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

[5] C. Zouridaki, B.L. Mark, M. Hejmo and R.K. Thomas, Robust cooperative trust establishment for MANETs, in: *Proc. ACM Workshop on Security Ad hoc and Sensor Networks (SASN)*, October 2006.

[6] D. Balfanz, D.K. Smetters, P. Stewart and H.C. Wong, Talking to strangers: Authentication in ad-hoc wireless networks, in: *Proc. Symposium on Network and Distributed Systems Security (NDSS'02)*, 2002.

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

[8] D. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, in: *Mobile Computing*, T. Imielinski and H. Korth, eds, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153–181.

[9] G. Theodorakopoulos and J.S. Baras, Trust evaluation in ad-hoc networks, in: *Proc. 2004 ACM workshop on Wireless Security (WiSe'04)*, 2004, pp. 1–10.

[10] H. Luo, J. Kong, P. Zerfos, S. Lu and L. Zhang, URSA: Ubiquitos and robust access control for mobile ad hoc networks, *IEEE/ACM Transactions on Networking* **12** (December) (2004), 1049–1063.

[11] J.S. Baras and T. Jiang, Cooperative games, phase transition on graphs and distributed trust in MANET, in: *Proc. 43rd IEEE Conference on Decision and Control*, June 2004.

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

[13] L. Capra, Engineering human trust in mobile system collaborations, in: *Proc. 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2004, pp. 107–116.

[14] L. Eschenauer, V.D. Gligor and J.S. Baras, On trust establishment in mobile ad-hoc networks, in: *Proc. Security Protocols Workshop*, Volume 2845, LNCS, 2002, pp. 47–66.

[15] L. Zhou and Z.J. Haas, Securing ad hoc networks, *IEEE Networks Special Issue on Network Security* (November) (1999).

[16] M. Virendra, M. Jadliwala, M. Chandrasekaran and S. Upadhyaya, Quantifying trust in mobile ad-hoc networks, in: *Proc. International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'05: Modeling, Evolution and Engineering)*, 2005.

[17] P. Papadimitratos and Z.J. Haas, Secure routing for mobile ad hoc networks, in: *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)* 2002, 2002.

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

[19] R.K. Nekkanti and C. Wei Lee, Trust based adaptive on demand ad hoc routing protocol, in: *Proc. 42nd ACM Southeast Regional Conference*, 2004, pp. 88–93.

[20] 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*, 2004.

[21] S. Ghazizadeh, O. Ilghami and E. Sirin, Security-aware adaptive dynamic source routing protocol, in: *Proc. 27th Annual IEEE Conference on Local Computer Networks*, 2002.

[22] T. Jiang and J.S. Baras, Ant-based adaptive trust evidence distribution in MANET, in: *Proc. 2nd International Workshop on Mobile Distributed Computing (MDC)*, 2004.

[23] T. Jiang and J.S. Baras, Autonomous trust establishment, in: *Proc. 2nd International Network Optimization Conference*, 2005.

[24] Y.C. Hu, A. Perrig and D. Johnson, Efficient security mechanisms for routing protocols, in: *Proc. Network and Distributed Systems Security*, 2003.

[25] Y.C. Hu, A. Perrig and D. Johnson, Packet leashes: A defense against wormhole attacks in wireless networks, in: *Proc. IEEE Infocom*, San Francisco, CA, 2003.

[26] 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, 2002.

[27] Y.C. Hu, D.B. Johnson and A. Perrig, SEAD: Secure Efficient Distance Vector Routing for mobile wireless ad hoc networks, in: *Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*, 2002, pp. 3–13.