Trust-based On-demand Multipath Routing in Mobile Ad Hoc Networks

Journal:	*IET Information Security*
Manuscript ID:	draft
Manuscript Type:	Research Paper
Date Submitted by the Author:	
Complete List of Authors:	Li, Xin; Shandong University, School of Computer Science and Technology Jia, Zhiping; School of Computer Science and Technology Wang, Luguang Wang, Haiyang; Shandong University, Computer Science and Technology
Keyword:	Ad-Hoc Networks Aspects, Multi-Party Computation, Network Security, Trust Models

Trust-based On-demand Multipath Routing in Mobile Ad Hoc Networks

Xin Li*, Zhiping Jia, Luguang Wang,Haiyang Wang

School of Computer Science and Technology,
Shandong University
Jinan, China, 250101
e-mail: {lx,jzp}@sdu.edu.cn

Abstract— A Mobile Ad hoc NETwork (MANET) is a self-organized system comprised of mobile wireless nodes with peer relationships. Due to multi-hop routing and absence of any trusted third party in open environment, MANETs are vulnerable to attacks by malicious nodes. In order to decrease the hazards from malicious nodes, we introduce the concept of trust to MANETs and build a simple trust model to evaluate neighbours' behaviours — packet forwarding. Extended from AODV route protocol, a trust-based reactive multipath routing protocol for MANETs, termed as Ad hoc On-demand Trusted-path Distance Vector (AOTDV) is proposed. This protocol discovers multiple loop-free paths which are evaluated by hop count and trust. The two-dimensional evaluation provides a flexible and feasible approach to choose a shortest path in all trusted paths to meet the dependable or trust requirements of data packets. Performance comparison of AOTDV and other related routing protocols shows that AOTDV is able to achieve a remarkable improvement in packet delivery ratio and end-to-end delay and to reduce black-hole, gray-hole and modification attacks.

Key words: ad hoc, trust, network, route, protocol

1. Introduction

A mobile ad hoc network (MANET) is a self-organized multi-hop system comprised of mobile wireless nodes. Two nodes out of direct communication range need intermediate nodes to forward their messages. Due to multi-hop routing and open working environment, MANETs are vulnerable to attacks by selfish or malicious nodes, such as packet dropping (black-hole) attacks and selective forwarding (gray-hole) attacks. Therefore dependable packet routing is a significant problem in a MANET [9].

Employing authentication and encryption mechanism, secure routing protocols [19, 20] have been developed to ensure properties such as confidentiality, integrity etc. However, those protocols require a centralized trusted third party, making them impractical for MANETs[5]. In addition, secure routing protocols cannot prevent malicious or compromised nodes that

[*] Contact Author.

are authorized participants to the network from doing any misbehaviour. As in social society, one will trust another person to carry out an action, but the former cannot guarantee the latter's behaviour [2]. Thus the concept of trust is introduced into computing network to measure an expectation or uncertainty that an entity has about another's future behaviour for a certain action. Trust can be derived from direct interactions or from recommendations.

There are two primary motivations associated with trust management in MANETs. At first, trust evaluation helps distinguish between good and malicious entities. Creating trust history, one entity can remember others' behaviours. This memory provides a method for good entities to avoid working with 'ex-convict' or suspect ones. Secondly, trust management offers a prediction of one's future behaviour and improves network performance. The results of evaluation can be directly applied to an incentive for good or honest behaviours while a penalty for selfish or malicious behaviours in the network. The feedback reminds network participants to act more responsibly. These motivations have interested researchers from the areas of information security and computer network in trust management of MANETs.

In this paper, we introduce a simple trust model based on packet forwarding ratio to evaluate neighbours' behaviours, and propose a novel multipath reactive routing protocol for MANETs, termed as Ad hoc On-demand Trusted-path Distance Vector (AOTDV). In a route discovery, this protocol is able to create multiple loop-free paths between a source and a destination through hop-by-hop route. Each route has a cost vector composed of hop count and trust value. Furthermore, this protocol provides a flexible and feasible approach to choose a shortest path in all trusted paths to meet the dependable or trust requirements of data packets. Performance comparison of AOTDV and two related routing protocols shows that AOTDV is able to achieve a remarkable improvement in the packet delivery ratio and alleviate most malicious attacks.

The proposed routing protocol is practical to enhance the dependability of routing and to detect malicious nodes in MANETs. In particular, the main contributions of our work can be summarized as follows:

1. Based on packet forwarding ratio, a simple and practical trust model is created to evaluate the behaviours of route nodes.

2. An on-demand multipath routing protocol (AOTDV) is proposed for MANET, in which top *k* shortest path and trustiest path are formed during one route discovery.

3. QoS-aware packet forwarding is established to satisfy user-specific requirements for dependability. Accordingly an adaptive mechanism is proposed to select a forward path dynamically in terms of the trust requirement of one packet.

4. We evaluate the AOTDV protocol and present experimental results in NS-2 simulator. It is shown that AOTDV is dependable to route packets and alleviate the attacks of malicious nodes in MANETs.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 gives a simple trust model. Section 4 describes a trust-based on-demand routing protocol. Section 5 presents the performance evaluation and experimental results. Section 6 summarizes our conclusions and identifies future work.

2. Related Work

2.1 Trust Model

It is clear that trust relationship involves two entities (subject and object) and a specific action. The uncertainty of trust exists because subject is not sure whether the object will carry out the action or not. One of the earliest literatures about computational trust is Marsh's formalism [3] that uses the outcomes of direct interactions among entities to compute situational and general trust. Situational trust is the level of trust in another for a specific situation, while general trust means overall trustworthiness in spite of the situation.

Several trust models have been developed for trust management. These models can be classified into two groups: centralized models and decentralized models.

In centralized models, trust values are maintained in a common central node or through an authorized third party. The simplest method is to sum the number of positive ratings and negative ones separately and keep a total score which equals to the positive score minus the negative score. This method is used in eBay's reputation forum [1]. The requirement of a trusted third party goes against the nature of MANETs.

In decentralized models, a node assigns a trust/trustworthiness value for every communicated node. Most researchers [21, 22, 23, 24, 31, 34] are advocating the use of ratings and prefer to complex rating aggregation algorithms to evaluate from several aspects and filter out the bad ratings. However, these sophisticated models are not appropriate for MANETs where resources are scarce and network topology is dynamic. Several trust models [25, 26, 30, 27] have been developed for peer-to-peer systems based on sharing recommendation information to establish reputation. Although in principle these models could be applied to routing in MANETs, additional recommendation information exchanging incurs significant network overhead. In particular, Pirzada and McDonald [8] proposed aggregation mechanism, where nodes calculate trust according to multiple observed events including acknowledgements, packet precision, gratuitous route replies, and blacklists. They have obtained promising simulation results, but we argue that similar promising effects can be obtained with a simplified trust model.

2.2 Routing Protocols

Traditional routing protocols in ad hoc network can be categorized into two primary types: proactive and reactive [6]. Proactive routing protocols establish and maintain routes at all instants of time in order to avoid the latency during new route discoveries. Reactive routing protocols do discovery route only when one node tries to transmit packets to another unknown-route node so as to save resources.

The nodes in an ad hoc network generally have limited resources, such as bandwidth and power energy; therefore reactive routing protocols attract more interests. AODV [14] combines the use of destination sequence numbers in DSDV [17] with the on-demand route discovery technique in DSR [18] to formulate a loop-free and single path routing protocol. Unlike DSR which use source routing, AODV is based on a hop-by-hop routing mechanism. Extended from AODV, AOMDV[15] is proposed to discovers multiple loop-free and link-disjoint paths. Experiments show that AOMDV is able to achieve a remarkable improvement in the end-to-end delay.

In information security, cryptographic primitives are often used to ensure properties such as confidentiality, integrity etc. Several secure routing protocols with cryptography have been proposed to protect the ad hoc networks, such as SAODV[19], Ariadne[20], but most of these protocols need centralized units or trusted third-parties to issue digital certificates or monitor network traffics. The requirement for a common trusted authority actually restricts the self-organization nature. Therefore, these protocols are less practical for MANET.

Recently a new class of routing protocol in MANET has been proposed, termed trusted routing protocol, which consists of two parts: a routing part and a trust model [5]. Routing decisions are made according to the trust model. Pirzada et al. [9] evaluated the performance of three trust-based reactive routing protocols (trusted AODV, DSR and TORA) using trust model by varying number of malicious nodes. The results indicate that each trust-based routing protocol has its own peculiar advantage that makes it suitable for application in a particular extemporized environment. Especially AODV routing maintains a stable throughput and surpasses TORA and DSR at higher traffic loads [9]. Therefore, we combined our trust model with AODV and designed a trust-based multipath routing protocol (AOTDV).

Inspired by the literature [8], Griffiths et al. [5] proposed the Simple Trusted AODV (ST-AODV), in which the next-hop node whose trust value is more than a constant threshold is selected as the forwarding node. The main difference between ST-AODV and our AOTDV are as follows: (1) ST-AODV is a single path routing protocol while AOTDV is a multipath protocol, and (2) ST-AODV uses node trust of the next-hop to make routing choices, while AOTDV considers the path trust values of paths to the destination as well as the number of hops, so that the selected next hop gives the shortest trusted path.

3. Trust Model

The trust model essentially performs the function of trust derivation, computation, and application [9]. In our model, each node derives trust factors from packet forwarding ratio. During trust computation, a linear aggregate method is used to estimate the overall trust in a node according to trust factors, and a minimal value method is used to compute a path's trust. Trust application including trust-based route discovery and route selection will be discussed in next section.

3.1 Trust Derivation

No matter what kind of trust models, two types of evolutions, direct trust and indirect trust, are available. Direct trust is first-hand information for neighbours and easy to obtain. In order to simplify trust model, we only use the history of direct interactions among nodes to compute trust.

Trust evaluation in routing procedure is a remark of a sender after it gets the service of a forwarding node. More specifically, a node *j* will give his neighbour *k* a trust score after the node *k* transmits a packet or replies a packet that the node *j* sends. Packet dropping is always due to poor wireless communication quality or heavy traffic or black-hold attack or grey-hold attack. Thus we use packet forwarding ratio to evaluate the quality of forwarding.

Packet Forwarding Ratio (*FR*) is the proportion of packets which have actually been forwarded correctly. Correct forwarding means the forwarding node not only transmits the packet to his next hop node but also forwards devotedly. For instance, a malicious neighbour node forwards the data packet after tampering with data. If the sender monitors this illegal modification, The *FR* of the neighbour will decrease.

Let $N_C(t)$ represent the cumulative count of correct forwarding and $N_A(t)$ signify the total count of all requesting before time t. The count of correct forwarding in a time window (from time t_i-w to t_i) is equal to $N_C(t_i)$- $N_C(t_i$-$w)$, where w represents the length of the time window. Let $FR(t_i)$ be packet forwarding ratio in the i-th window. $FR(t_i)$ is defined as follows::

$$FR(t_i)=\begin{cases} \dfrac{N_C(t_i)-N_C(t_i-w)}{N_A(t_i)-N_A(t_i-w)} , & t_i > w \\ \dfrac{N_C(t_i)}{N_A(t_i)} , & t_i \le w \end{cases} \tag{1}$$

where i=1,2,3,.... Assume that the difference d of t_{i+1}−t_i (i>1) is fixed. $FR(t_i)$ is calculated at a fixed interval of d units of time. The constant d, termed Trust Update Interval [9], is a very critical component of a trust model and determines the time a node should wait before updating a trust value to its neighbour. Forwarding records in recent w units of time are considered and the history records out of recent window fade as time goes by.

In MANETs, packets can be classified into two groups: control packets and data packets. The former include routing packets for request, reply and error. The accuracy of control packets plays a vital role in establishment of accurate routes throughout the network. So packet forwarding ratio is divided into two parts: Control packet Forwarding Ratio (CFR) and Data packet Forwarding Ratio (DFR).

3.2 Node's Trust Computation

The trust of a node *j* to another node *k* is a measure of ensuring that packets which have been sent to node *k* by node *j* for forwarding have actually been forwarded by node *k*. Trust values from the two trust factors (CFR and DFR) are assigned weights in order to determine the overall trust level for a particular node. The direct trust in node *k* by node *j* is represented as T_{jk} and is given by the following equation:

$$T_{jk}(t_i)=w_1\times CFR_{jk}(t_i)+w_2\times DFR_{jk}(t_i) \tag{2}$$

in which $CFR_{jk}(t_i)$ and $DFR_{jk}(t_i)$ respectively represent the control packet forwarding ratio and data packet forwarding ratio observed by node *j* for forwarding node *k* at time t_i. The parameters w_1 and w_2 reflect the weights assigned to *CFR* and *DFR*, respectively.

Table 1. Trust levels of nodes

Level	Trust Value	Meaning
1	[0,0.5]	Malicious
2	(0.5,0.85]	Suspect
3	(0.85,0.95]	Less trustworthy
4	(0.95,1]	Trustworthy

After each interaction, node *j* considers whether the neighbor *k* forwards the packet correctly. If so, the forwarding ratio FR_{jk} increases. Otherwise, FR_{jk} decreases. In our trust model, trust values are limited in a continuous range from 0 to 1. A trust value of 0 signifies complete distrust while a value of 1 implies absolute trust. The middle value of 0.5 means failure probability is equal to that of correct forwarding and values more than 0.5 mean that more correct chances occur than failures. The trust levels of nodes are listed in Table 1.

Each node, based upon its personal experiences, rewards collaborative nodes for their benevolent behaviour and punishes malicious nodes for their malevolent action. If a node is evaluated very low by all its neighbours, it is not allowed to send packets for forwarding. Any reply it gives to route requests is discarded, and any request it initiates is ignored. To minimize the risk of transmission failure, nodes should interact with the trusted ones of the potential partners, whose trust value is above the trust requirement of the packet.

The sender places itself in promiscuous mode after the transmission of any packet so as to overhear the retransmission by the forwarding node. Using this method, a node can know whether the packet which has been sent to a neighbour for forwarding is indeed forwarded or not. The direct trust values can also be shared among neighbors using a higher layer Reputation Exchange Protocol [7]. But we will not explore this aspect in the work here.

3.3 Path's Trust Computation

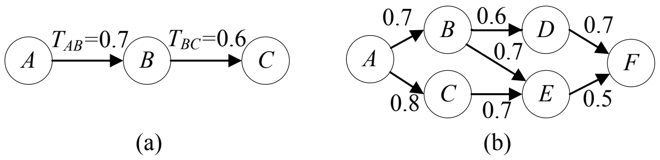
When a source discovers a path to the destination with the help of forwarding nodes, the trust value of the path is able to be computed through the trust values of nodes among the path. According to the axiom [32] that concatenation propagation of trust does not increase trust, the trust value of a path should not be more than the trust values of all nodes between the source and the destination. So, in our model the trust of a path P (denoted by T_P (t_i)) is equal to the minimal one of the nodes' values in the path. i.e.,

$$T_P(t_i)=\min(\{T_{jk}(t_i)|n_j,n_k \in P \ and \ n_j \to n_k\}) \tag{3}$$

in which, n_k and n_k are any two adjacent nodes among the path P and $n_j \to n_k$ means that n_k is the next-hop node of n_j.

The trust computation based on minimal value is similar to opinions in information theory: the information cannot be increased via propagation [32].

As shown in Figure 1(a), the direction edge from A to B denotes the trust T_{AB}. The trust value of path ($A \to B \to C$) is equal to the less one (0.6) of T_{AB} and T_{BC}. Figure 1(b) show a complex graph with branches, in which there are three paths from A to F and the path ($A \to B \to D \to F$) is the most trustworthy path. In next section, we will describe the procedure that AOTDV tries to find the trusted path.



(a) (b)

Figure 1. Path trust computation

4. Trust-based On-demand Routing Protocol

In this section, we describe an on-demand routing mechanism for ad hoc network based on the proposed trust model. At first, the structures of routing table and trust record list are depicted. Then, the procedures of route discovery and routing maintain are discussed. Finally a sequence number method is presented to avoid the routing loop.

4.1 Routing Table

Routing table stores the routes to various nodes in an ad hoc network. Each node maintains a route table composed of multiple routing entries. AOTDV adopts hop-by-hop routing mechanism, in which the source is not expected to have all the information about how to get to the destination; it is sufficient for the source to know only how to get to the next hop. So when a data packet is going to a particular node, it then refers to local routing table to find the address of next hop (named node *j*) to the destination. Once it reaches the node *j*, it again refers to the *j*'s routing table for the address of next hop and so on, until it reaches the final destination.

The routing table in any node *j*, only stores the destinations' routes and reverse routes to the sources interacted with node *j* recently, not all nodes' route in history. This is because the topology of MANET changes dynamically, i.e., the mobile nodes might join or quit the network for some reasons.

Table 2. Structure of routing table entry

Destination
SequenceNumber
ExpirationTime
RouteList {(NextHop$_1$, HopCount$_1$, PathTrust$_1$), (NextHop$_2$, HopCount$_2$, PathTrust$_2$), …}

Table 2 shows the structure of the routing table entry for AOTDV. The routing entry consists of the following fields:

(1) Destination: the identifier of destination node.

(2) Sequence number: the greatest known sequence numbers for destination denotes freshness of routing information. It is used to avoid routing loop (discussed in subsection 4.6).

(3) ExpirationTime: the time after which the route is considered to be invalid. Each time a route entry is used to transmit data from a source toward a destination, the ExpirationTime for the entry is reset to the current time plus a constant (active route timeout).

(4) Next hop: The next hop, or gateway, is the address of the next station to which the packet is to be sent on the way to the final destination.

(5) HopCount and PathTrust: the two metrics compose a cost vector of a path. Different from one dimension cost (eg. number of hops in AODV[14]), here the cost is a vector composed of hop count and trust value of the path from next-hop node to the destination.

Multiple routes leading to the same destination arrange in ascending order of *HopCount*, i.e.,$HopCount_1 \leq HopCount_2 \leq \ldots \leq HopCount_n$. If two paths have same *HopCount*, the one with greater *PathTrust* precedes, i.e., $\forall\ HopCount_i = HopCount_{i+1}$, $PathTrust_i \geq PathTrust_{i+1}$.

Table 3. Example of routing table

Destination	SeqNumber	Next-hop	HopCount	PathTrust	ExpirationTime
1	1	1	1	0.7	30
5	2	1	3	0.6	22
		2	4	0.7	32

Table 3 gives an example of routing table, in which there are two paths to node 5 and their next-hops are 1 and 2 respectively.

4.2 Trust Record List

Table 4. Structure of a trust record

Node ID
N_C and N_A for control packets
N_C and N_A for data packets
Packet Buffer

To remember trust information, we introduce a trust record list. Each node will also maintain a trust record for every neighbour which has been sent packets to for forwarding. A trust record listed in Table 4 comprises a node ID, two integer counters of N_C and N_A for control packets, two integer counters of N_C and N_A for data packets, and a packet buffer. The packet buffer is used to record all packets sent recently. It is a circular buffer, which means that the buffer will cycle and overwrite the oldest packet if it is not removed in time.

Before sending a packet to a neighbour, the sender looks up the trust record corresponding to the neighbour and increases N_A of CFR (the packet is a unicast control packet) or N_A of DFR (the packet is a data packet) by 1. To detect whether a packet is successfully forwarded, the packet will not delete immediately after being sent. Then it will be stored in the packet buffer and wait for acknowledgment. If the packet is forwarded correctly, it will be removed from the packet buffer and the corresponding counter of correct forwarding increases 1.

4.3 Route Strategy

As shown in Figure 2, the overall procedure of AOTDV routing is given as follows: (1) when a source *s* wants to send a data packet to another node *d*, the source first tries to look up destination *d* in its route table. If no such route, it will initiate a route discovery for *d*. if one or more paths to the destination are found after the on-demand route discovery, all paths information will be inserted into the route table. (2) Node *s* selects a trusted route with the least hop count as the next hop (named node *n*). It then compares the path trust values with the requirement of the packet and selects one with less hop count while trust value is more than the requirement of the data packet. (3a) If no qualified route is selected, node s will retry to discover route to *d* until the retry count reaches a threshold 3. (3b) If a qualified route is selected, node *s* inserts the data packet into its packet buffer and sends the data packet to node *n*. After sending, *s* overhears the channel and checks whether the packet will be forwarded correctly. (4a) If the next-hop node drops the packet or is unacceptably slow at forwarding packet, node *s* will not overhear the packet in a certain threshold time. In this case, it observes the forward error and tries to lookup another alternate qualified route to send the packet. (4b) If the packet is forwarded correctly, the source node updates the trust record based on its observation of route quality. The trust record can also be used for malicious node detection.

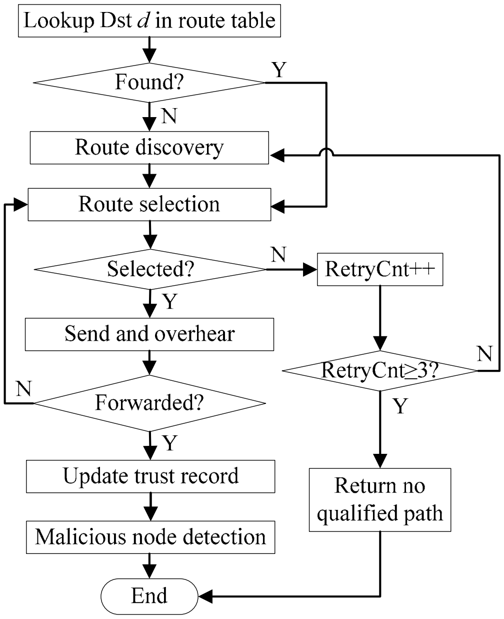


Figure 2. Flow chart of routing procedure

If one neighbor's trust value is lower than a threshold η, it will be regarded as a malicious node, and then deleted from the neighbor set, finally added into the black list. That is, it will be ultimately denied by the whole network.

AOTDV is a multipath on-demand protocol, which tries to alleviate route discovery attempts in dynamic networks by computing multiple paths in a single route discovery round. Multiple paths could be formed at both source node and intermediate nodes. New route discovery is needed only when all paths break or fail to meet the trust requirement of data packets. Multiple paths can also be used to balance load by forwarding data packets on multiple paths at same time.

4.4 Route Discovery and Path Selection

The route discovery process is initiated whenever a source node *s* needs to communicate with another node *d* for which node *s* has no routing information in its routing table. Every node maintains two independent counters: a node sequence number and a broadcast ID. The source node initiates a network-wide flood by broadcasting a route request (RREQ) packet and waits for a route reply (RREP) packet.

4.4.1 Route Request

A RREQ packet contains the following fields: <BroadcastID, SourceAddr, SourceSequenceNo, DestAddr, DestSequenceNo, HopCounter, **RequiredTrust**, **ActualTrust**>.

The first 6 fields including BroadcastID, SourceAddr, SourceSequenceNo, DestAddr, DestSequenceNo, and HopCounter, are similar to the corresponding ones in AODV[14]. The major difference is two additional fields for path trust value, i.e. Required Trust (RT) and Actual Trust (AT). The *RT* represents the path trust value required by the data packet and is set by the source. During the flood, *RT* remains unchanged. The *AT* denotes the minimal one of trust values of nodes that the RREQ has passed by during route discovery. And it is initialized to 1 by the source. During the flood, *AT* varies with the transmission of RREQ packet.

When an intermediate node *j* receives a RREQ from a neighbour *k*,

(1) it checks whether one copy of the same RREQ has been received. If the later copy has less HopCounter (the later path is shorter.) or greater *AT* (the later path is more trustworthy), go to step (2); the RREQ will be discarded;

(2) it creates a reverse route to the source node using the previous hop *k* of the RREQ as the next hop. The path trust of the reverse route is set to min(AT, T_{jk}), i.e. the minimum of *AT* and T_{jk} If T_{jk} is unknown, T_{jk} is initialized to 1 (absolute trust). Here we adopt an optimistic view on the unknown-node trust.

(3a) if a valid route to the destination is available, node *j* generates a RREP;

(3b) or else, node *j* modifies the *AT* of the RREQ using min(AT, T_{jk}), increases HopCounter by one and re-broadcasts the RREQ.

If an intermediate node has a route entry for the desired destination, it determines whether the route is fresh by comparing the destination sequence number in its own route entry with the one in the RREQ. If the RREQ's sequence number for the destination is greater than that route’s in the intermediate node, the intermediate node must not use its recorded route to respond to the RREQ. Instead, the intermediate node rebroadcasts the RREQ.

4.4.2 Route Reply

The intermediate node can reply only when it has a route with a sequence number that is greater than or equal to that contained in the RREQ. If it does have a fresh route to the destination, and if the RREQ has not been processed previously, the node unicast a route reply (RREP) packet back to its neighbour from which it received the RREQ. A RREP packet contains the following information:

<SourceAddr, SourceSequenceNo, DestAddr, DestSequenceNo, HopCounter, LifeTime, **RequiredTrust**, **ActualTrust**>

The first 6 fields including SourceAddr, SourceSequenceNo, DestAddr, DestSequenceNo, HopCounter, and LifeTime are also similar to the corresponding ones in AODV [14]. The Required Trust (RT) and Actual Trust (AT) have same meaning to the ones in RREQ. The *AT* in RREP denotes the minimal one of trust values of nodes that the RREP passed by during route reply. And it is initialized to 1 by the destination.

If an intermediate node receives a RREQ from a neighbour, and if it has multiple paths to the destination, it will reply two copy of RREP, in which one has the smallest hop count and the other has the greatest trust value. If the destination receives multiple copies of RREQ, it will reply the first *k* trusted paths at most, whose path values are greater than the RequiredTrust of the RREQ. After a RREQ packet arrives at a node, a reverse path is established to the source of the RREQ (Section 4.4.1). As the RREP travels back to the source, each node along the path sets up a forwarding route to the destination from which the RREP came, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination.

The parameter *k* is used to control the number of RREPs and to prevent a RREP storm. Nasipuri et al.[33] indicated that additional routes beyond a few provide only marginal benefit. We have used *k*=3 in our experiments.

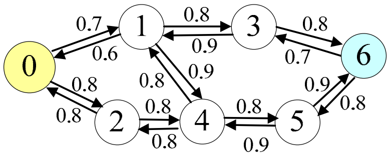
4.4.3 A Route Example

In ad hoc networks on battlefield or business applications, different data have different requirements for importance and trust level. Generally, the more important data are, the more secure and trusted routes they need. Four trust levels are listed in Table 5. When a packet is requested to forward, its trust level will be assigned in the request packet.

Table 5. Trust levels of data packet

Level	Trust Value	Description
1	0.6	Unimportant data
2	0.75	Important data
3	0.85	Very important data
4	0.95	Extremely important data

Assume that, at the beginning every node has no route to node 6 and node 0 tries to send a data packet to node 6. The packet requires that path trust value should be more than 0.7. As shown in Figure 3(a), node 0 initiates a route discovery for node 6 and broadcasts the RREQ with required trust 0.7 to his neighbours. Node 1 receives two copies of RREQ from node 0 and node 4 successively. Node 1 will creates two reverse routes to node 0 (Fig. 3(b)), in which one route entry is <NextHop=0, HopCount=1, PathTrust=0.6>, the other one is <NextHop=4, HopCount=3, PathTrust=0.8>. After receiving multiple RREQ, the destination forms two paths to the source 0 (Fig. 3(c)), in which one route entry is <NextHop=3, HopCount=3, PathTrust=0.6>, the other one is <NextHop=5, HopCount=4, PathTrust=0.8>. Then node 6 unicasts one RREP to node 3 and another RREP to node 5. After receiving the two copies of RREP, node 1 inserts two routes to node 6 into route table. As shown in Fig. 3(d), one route entry is <NextHop=3, HopCount=2, PathTrust=0.8>, the other one is <NextHop=4, HopCount=3, PathTrust=0.8>. At last, node 0 receives two copies of RREP and creates two routes to node 6. One route entry is <NextHop=1, HopCount=3, PathTrust=0.7> (named r_1), and the other one is <NextHop=2, HopCount=4, PathTrust=0.8> (named r_2).



(a) Trust DAG

Dst	RouteList		
	NH	HC	PT
0	0	1	0.6
	4	3	0.8
3	3	1	0.8
4	4	1	0.9

(b) Route table of node 1 before receiving RREP

Dst	RouteList		
	NH	HC	PT
3	3	1	0.7
5	5	1	0.8
0	3	3	0.6
	5	4	0.8

(c) Route table of node 6 after receiving RREQ

Dst	RouteList		
	NH	HC	PT
0	0	1	0.6
	4	3	0.8
3	3	1	0.8
4	4	1	0.9
6	3	2	0.8
	4	3	0.8

(d) Route table of node 1 after receiving RREP

Dst	RouteList		
	NH	HC	PT
1	1	1	0.7
2	2	1	0.8
6	1	3	0.7
	2	4	0.8

(e) Route table of node 0 after receiving RREP

Figure 3. Example of a routing discover procedure

After route discovery, node 0 finds two trusted paths to node 6. According to the required trust value 0.7, node 0 will choose node 1 rather than node 2 as the next hop because the route r_1 has shorter distance than the route r_2. Node1 receives the data packet from node 0 and lookup a shortest path in its route table to node 6. Node 1 will select node 3 as the next hop to forward the packet. Finally, the packet crosses the path ($0 \rightarrow 1 \rightarrow 3 \rightarrow 6$) to its destination. If another data packet requires the path to node 6 with path trust value no less than 0.8, the path ($0 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$) will be selected.

4.5 Route Maintenance and Loop Freedom

The route maintenance in AOTDV is similar to that in AODV, i.e., nodes maintain and update route table when receiving a RREQ, RREP or route error (RERR) packet. When a link failure is detected (by a link layer feedback, for example), a RERR is send back to all sources using that failed link via separately maintained predecessor links. Routes are erased by the REER along its way. When a node receives a RERR, it initiates a new route discovery to fix the link if the route is still needed. Unused routes in the routing table are expired using a timer-based technique [15].

All protocols using broadcast to discovery route will encounter routing loops. For example, an intermediate node *j* broadcasts a RREQ. A neighbor *k* receives the RREQ and also broadcasts it, which in turn is heard by *j.* If *j* accepts the RREQ copy to form a reverse path, this will form a loop route. When more than one path exists from the source *s* to an internal node *i*, multiple copies of the RREQ packet will arrive at *i*. Node *i* forwarding all such copies will lead to routing loops because node *i* could re-broadcasts RREQ that has been forwarded before.

Sequence numbers in AODV play a key role in ensuring loop freedom [15]. Therefore, in order to avoid any possibility of loops, every node maintains a monotonically increasing sequence number for itself. It also maintains the highest known sequence numbers for each destination in the routing table (called "SequenceNumber" in subsection 4.1). Destination sequence numbers are tagged on all routing packets, thus providing a mechanism to compute the relative freshness of two copies of routing packets generated by two different nodes for the same destination.

When receiving a control packet such as a RREQ or RREP packet, a node may create a reverse path to the source or forward path to the destination. However a node should create or update a route on a fresh control packet not on an old control packet. Assume that a node *j* receives a control packet to a destination *d* ($j \neq d$) from a neighbor *k*. Let the variables $SeqNumber_k^d$, $HopCounter_k^d$, $ActualTrust_k^d$, $SeqNumber_j^d$, $RouteList_j^d$ represent the DestSequenceNo, HopCounter and ActualTrust of the control packet, SequenceNumber and RouteList of destination *d* in the route table of node *j* respectively. Let $MaxTrust_j^d$ and $MinHops_j^d$ be the maximum PathTrust and minimum HopCount of multiple paths for destination *d* in the route table respectively. The update rule for route table in AOTDV is given as follows.

```
if ($SeqNumber_j^d < SeqNumber_k^d$) then
    $SeqNumber_j^d = SeqNumber_k^d$;
    $RouteList_j^d$ =NULL;
    insert ($k$, $HopCounter_k$+1, min($ActualTrust_k^d$ , $T_{jk}$)) into $RouteList_j^d$;
elseif ($SeqNumber_j^d = SeqNumber_k^d$) then
    if ($ActualTrust_k^d < MaxTrust_j^d$) or ($HopCounter_k^d < MinHops_j^d$) ) then
        insert ($k$, $HopCounter_k$+1, min($ActualTrust_k$ , $T_{jk}$)) into $RouteList_j^d$;
    endif
endif
```

The route update rule above is invoked on receiving a RREQ or RREP packet. Line(1) and (5)-(6) of the route update rule ensure loop freedom. The protocol only allows accepting alternate routes with lower hop count or ones with higher path trust.

5. Experiment

To examine the performance of AOTDV with respect of AODV [14] and AOMDV [15], we have conducted a comprehensive evaluation using ns-2 network simulator [28]. All experiments are carried out on a PC machine with a Pentium 4 processor (2.4 GHz) and 2GB main memory.

5.1 Experiment Setup

Table 6. Fixed Simulation Parameters

Protocols	AODV, DSR,TORA
Simulation time	500 seconds
Number of nodes	50
Map size	1000m×1000m
Mobility model	Random way point
Traffic type	Constant Bit Rate (CBR)/UDP
Transmission radius	250m
Packet size	512 bytes
Connection rate	4pkts/sec
Weight w_1	0.6
Weight w_2	0.4
Connections	20
Pause time	10 seconds

NS-2 simulator was used to evaluate the performance of three on-demand routing protocols (AODV, AOMDV, and AOTDV) in different conditions. The Distributed Coordination Function (DCF) of IEEE 802.11[29] for wireless LANs is used as the MAC layer protocol. An unslotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) [11] is used to transmit the data packets and route packets. Because the forwarding ratio of control packet is more important than that of data packet, we set the weights w_1 and w_2 using 0.6 and 0.4 respectively. The trust threshold η is set to 0.5, which means the node with trust value less than o.5 will be regarded as a malicious node and added into the black list. The length of the time window w in formula (1) is set to 500 seconds. That is, packet forwarding ratio is computed by the cumulative count of correct forwarding and the total count of all requesting from the beginning (t=0). These fixed simulation parameters are listed in Table 6.

50 nodes are randomly dispersed in a rectangular field with 1000m×1000m. The transmission radius of every node in one hop is fixed at 250m.The node mobility uses the random waypoint model [12] in which each packet starts its journey from a random location to a random destination with a randomly chosen speed. A maximum speed of 0 m/s implies that the MANET

is a static network. Once the destination is reached, another random destination is targeted after a pause time. The varying simulation parameters used in test 1-2 are listed in Table 7.

Malicious nodes launch modification attacks, gray hole attacks or black hole attacks. In our experiment, modification attacks are carried out by altering IP addressed from the control packets or data packets, which pass through the malicious nodes. In gray hole attacks, malicious nods selectively forwards data packets at a ratio. For simplification, we used a constant 30% as the forwarding ratio. In black hole attack, the malicious nodes drop all data packets which are supposed to be forwarded, while delivery route request and reply packets devotedly in order to participate in data transmission. The fractions of modification, gray hole and black hole attacks are 30%, 40%, and 30% respectively.

Table 7. Varying Simulation Parameters

Test No.	Malicious nodes	Max speed
1	10	0-30 m/s
2	0-20	10 m/s

5.2 Performance Metrics

We use four metrics to evaluate the performance of the route protocols, in which the first two metrics are the most important for best effort route and transmit protocols.

(1) **Packet Delivery Ratio**: the fraction of the data packets delivered to the destination nodes to those sent by the source nodes.

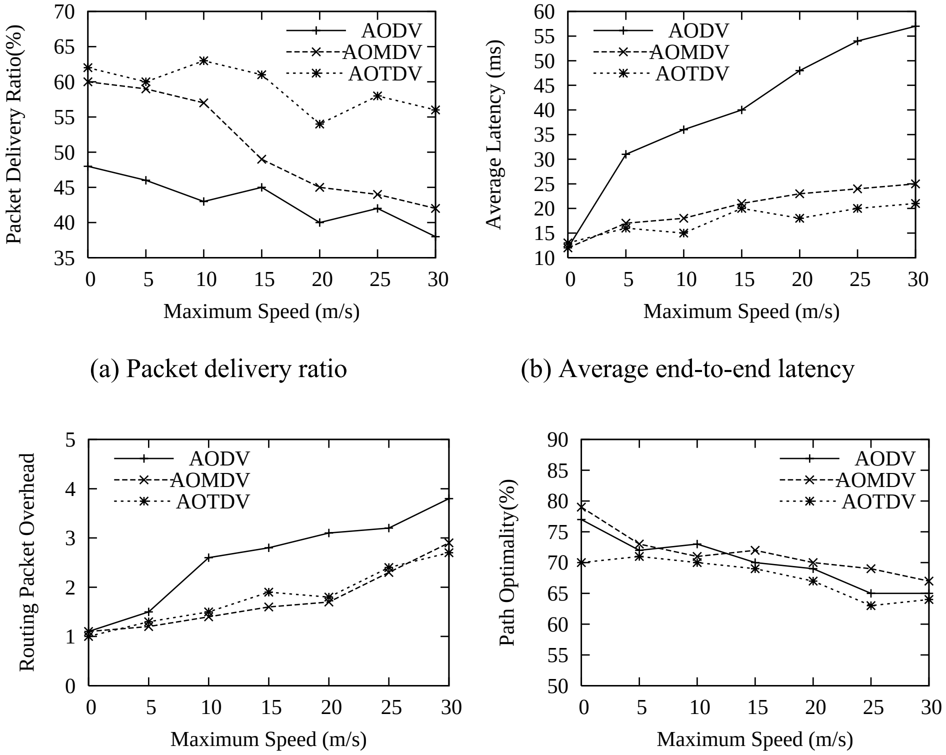
(2) **Average end-to-end Latency**: the average time taken by the data packets from sources to destinations, including buffering delays during route discovery, queuing at the interface queue, retransmission delays at MAC layer and propagation time.

(3) **Routing Packet Overhead**: the ratio of the number of control packets (including route request/reply/error packets) to the number of data packets.

(4) **Path Optimality**: the ratio of the total number of hops in the shortest paths to the total number of hops in the paths taken by data packets.

To decrease the effect of random error, every experiment repeats 50 times and the average of experiment results is used as the performance metrics.

5.3 Test 1: Varying Node Speeds



(a) Packet delivery ratio

(b) Average end-to-end latency

(c) Routing packet overhead

(d) Path Optimality

Figure 4. Test 1: Performance with a varying node maximum speeds

In the first test, we compare the performance of the AOTDV with that of other two protocols as maximum speed of nodes varies from 0 m/s to 30 m/s. As shown in Fig 4(a), the delivery ratios of AODV and AOMDV decline as nodes speed while the delivery ratio of AOTDV fluctuates. The performance differences become more apparent at higher speed. This advancement of AOTDV can be attributed to the improved probability of node behavior detection due to the more interactions. Under lower speed (speed<10m/s), both AOTDV and AOMDV can make use of their multipath feature. Each data packet being forwarded by the intermediary nodes is supported by this multipath feature, which elevates the probability of successful delivery to a trusted node. In contrast, nodes executing AODV only maintain limited routes to a destination and are unable to aid in packet delivery in case of the unavailability of a trusted next hop link leading to a destination.

Figure 4(b) and 4(c) illustrate that the average end-to-end latency and routing packet overhead for these protocols rise with the increase in speed. At higher speeds, the links are frequently disconnected and thus the nodes initiate additional route discoveries to sustain ongoing data connections. At highest speed of 30 m/s, the average latency and routing packet overhead respectively reach their peak value in the test. The routing overhead of AOTDV and AOMTV remains comparatively lower than that of AODV due to its multipath feature. AOTDV has a little lower average latency (2-4 ms) than AOMDV when the speed is greater than 5 m/s.

As shown in Figure 4(d), the path optimality of these protocols degrades as the speed increases. AOTDV has less path optimality than AODV and AOMDV. This is observed due to the fact that in AOTDV intermediate nodes make routing selection considering hop count and trust value, thus the actual paths may sway notably from the best available paths. These longer paths also increase a little latency of the packet transmission.

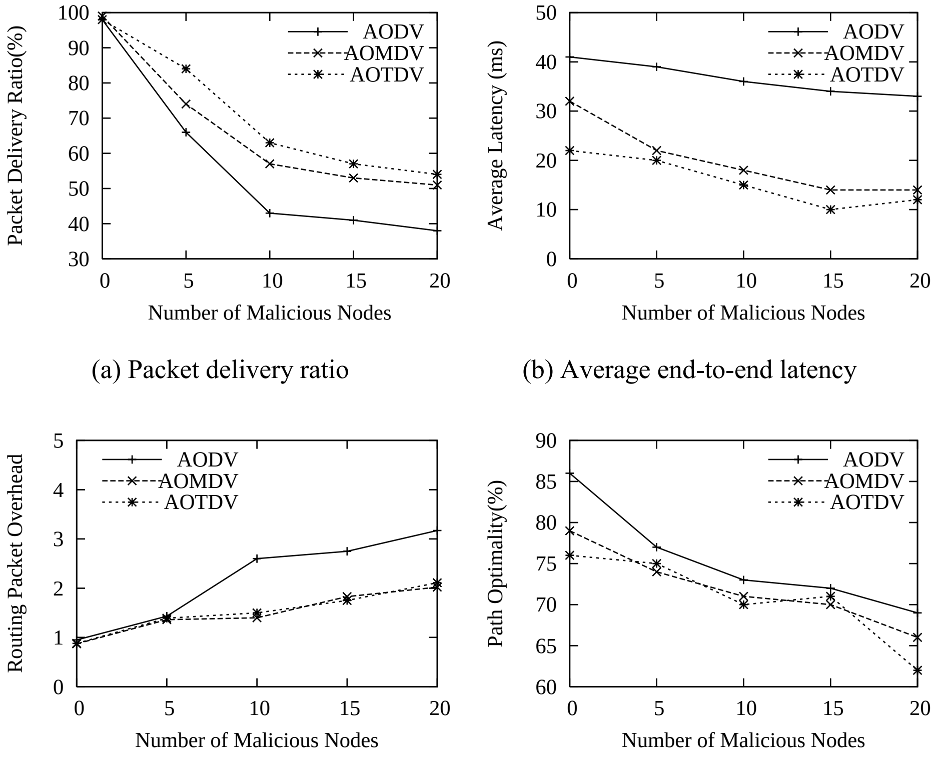
5.4 Test 2: Varying Number of Malicious Nodes

In test 2, we evaluate the effects on these protocols under varying number of malicious nodes. In the absence of malicious nodes, the typical packet loss is about 1 percent for AODV, AOMDV, and AOTDV. As shown in Figure 5(a), the delivery ratio of all protocols degrades sharply as malicious nodes increase. The delivery ratio of AOTDV drops from 99% to 54% as the number of malicious nodes varies from 0 to 20. Lower packet delivery ratio means less network throughput. Malicious nodes essentially limit the interactions of nodes in the network. However, in AOMDV and AOTDV, intermediary nodes have several routes to a destination so that when detecting gray hole or black hole attacks, they can try alternate route to forward packets and thus improve the packet delivery ratio.

As shown in Figure 5(b), the average latency of all three protocols declines slowly with the increase in number of malicious nodes. There is a tremendous reduction in the average latency with AOMDV and AOTDV compared to AODV. This is because availability of alternate routes eliminates route discovery delay that contributes to the end-to-end latency.

When the number of malicious nodes increase to 20 (40 percents of the whole nodes), the routing packet overhead of AOTDV and AOMDV is lower than 2.5 (shown in Figure 5(c)). On the average, AOMDV and AOTDV generates about 1.5 control packets for every data packet while AODV creates about 2.5 control packets. The increased control packet in AODV is primarily due to its route discovery mechanism that broadcasts more RREP packets to look up new routes to destinations.

As shown in Figure 5(d), the AODV in these protocols exhibits the best path optimality (85%) when there is no malicious node. As malicious nodes increase, the path optimality of all three protocols decreases. Overall, the path optimality of AOTDV is less than that of AOMDV, since AOTDV is able to detect and filter out malicious nodes, and find trustworthy paths to destinations even though these paths is longer.



(a) Packet delivery ratio

(b) Average end-to-end latency

(c) Routing packet overhead

(d) Path Optimality

Figure 5. Test 2: Performance with a varying number of malicious nodes

To sum up, the experiment results in test 1 and 2 show that our trust model is effective and the AOTDV protocol performs better than AODV and AOMDV as it gives higher packet throughput (delivery ratio) and lower end-to-end latency.

6. Conclusions

In this paper we have described a simple trust model based on packet forwarding ratio to evaluate neighbours' behaviours. Combined with the model, a novel multipath reactive routing protocol (AOTDV) is proposed to discover trustworthy forward paths and alleviate the attacks of malicious nodes. In this protocol, a source can find multiple trusted paths to a destination in a single route discovery round. New route discovery is needed only when all paths break or fail to meet the trust requirement. This protocol provides a flexible and feasible approach to choose a shortest path in all trusted paths to meet the dependable or trust requirements of data packets. Multiple paths can also be used to balance load by forwarding data packets on multiple paths at same time.

Performance comparison of AOTDV, AODV and AOMDV routing protocols shows that AOTDV is able to achieve a remarkable improvement in the packet delivery ratio and prevent most malicious attacks. For future work, we plan to extend our trust model to other ad hoc network routing protocols like DSR, DSDV and TORA. We will also conduct a comprehensive performance evaluation to compare AOTDV with other trust-based routing protocols.

7. Acknowledgement

This research is sponsored by the Natural Science Foundation of China (NSFC) under Grant No. 90718032 and by the Youth Faculty Award of the School of Computer Science and Technology of Shandong University in China.

8. References

[1] Resnick, P. and Zeckhauser, R.:‘Trust among strangers in Internet transactions: Empirical analysis of eBay’s reputation system’, in Baye, M. (Ed.):‘Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce’ (Elsevier Press,2000, 1st edn.), pp. 127–157

[2] Gambetta, D.:‘Can we trust trust?’, in Gambetta, D. (Ed.):’Trust: Making and Breaking Cooperative Relations’(Oxford Press, 2000, 1st edn.), pp. 213–237

[3] Marsh, S. P.:‘Formalizing Trust as a Computational Concept’, Ph.D. Thesis. Department of Mathematics and Computer Science, University of Stirling ,1994.

[4] Hu,YC., and Perrig, A.:’A Survey of Secure Wireless Ad Hoc Routing’. IEEE Security and Privacy,2004,2,(3), pp.28-39.

[5] Griffiths, N., Jhumka, A., Dawson, A., and Myers, R.:’A Simple Trust Model for On-Demand Routing in Mobile Ad-hoc Networks’, Proc. Int. Symp. on Intelligent Distributed Computing (IDC 2008), 2008, pp. 105-114

[6] Royer, E.M., and Toh, C.K.: ‘A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks’, IEEE Personal Comm.Magazine, 1999, 6, (2), pp. 46-55

[7] Pirzada, A.A., Datta, A., and McDonald, C.:’Propagating Trust in Ad Hoc Networks for Reliable Routing’, Proc. Int. Workshop Wireless Ad Hoc Networks (IWWAN), May 2004.pp.58-62

[8] Pirzada, A.A., and McDonald, C.:’Trust establishment in pure ad-hoc networks’, Wireless Personal Communications, 2006,37,(1),pp39–168

[9] Pirzada, A.A., McDonald, and C., Datta, A.:’Performance comparison of trust-based reactive routing protocols’, IEEE Trans. on Mobile Computing, 2006, 5, (6), 695–710

[10]Marti, S., Giuli, T., Lai, K., and Baker, M.:’Mitigating Routing Misbehavior in Mobile Ad Hoc Networks’, Proc. Int. Conf. Mobile Computing and Networking (MobiCom), ,2000, pp. 255-265

[11]Tanenbaum, A.S.:’The Medium Access Control Sublayer’, in Tanenbaum, A.S.(Ed.):’Computer Networks’(Prentice Hall Press,2002, 4th edn.),pp. 251-270

[12]Bettstetter, C., Resta, G. and Santi, P.:’The node distribution of the random waypoint mobility model for wireless ad hoc networks’, IEEE Transactions on Mobile Computing,2003, 2, (3), pp. 257–269

[13]Cheng, W., Liao, X., Shen, C., Li, S. and Peng, S.:’A Trust-Based Routing Framework in Energy-Constrained Wireless Sensor Networks’, Proc. Int. Conf. on Wireless Algorithms, Systems, and Applications (WASA), 2006, pp. 478-489

[14]Perkins, CE., Royer, EM., and Das, SR.:’Ad-hoc On-demand Distance Vector Routing’, Proc. Int. Workshop on Mobile Computing Systems and Applications (WMCSA),1999, pp.90-100

[15]Marina, MK., and Das, S R.:’On-demand Multipath Distance Vector Routing for Ad Hoc Networks’, Proc. Int. Conf. on Network Protocols. Nov. 2001, pp.11-14

[16]Lee, S J. and Gerla, M.:‘Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks’, Proc. Int. Conf. on Communications, 2001,pp.3201-3205

[17]Perkins, C.E., and Bhagwat, P.:’Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for mobile Computers’, Proc. Int. Conf. ACM SIGCOMM, 1994, pp234-244

[18]Johnson, D., and Maltz, D.:’Dynamic Source Routing in Ad hoc Wireless Networks’, in Tomasz, I., and Hank, K. (Ed.):’ Mobile Computing’ (Kluwer Academic Press, 1996, 1st edn.), pp. 153-181

[19]Zapata, M.G., and Asokan, N.:’Secure Ad hoc On-Demand Distance Vector Routing’, ACM Mobile Computing and Communications Review, July 2002, 3, (6), pp. 106-107

[20]Hu, Y.C., Perrig, A., and Johnson, D.B.:’Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks’, Proc. Int. Conf. Mobile Computing and Networking (Mobicom’02), Atlanta, Georgia, September 2002, pp.12-23.

[21]Buchegger, S., and Boudec J.L.:’A robust reputation system for p2p and mobile ad-hoc networks’, Proc. Int. Workshop on the Economics of Peer-to-Peer Systems, Cambridge MA, U.S.A., June 2004.

[22]Jøsang, A., and Ismail, R.:’The beta reputation system’, Proc.of the 15th Bled Electronic Commerce Conference, Bled, Slovenia, June 2002. pp. 1-14.

[23]Sabater, J., and Sierra, C.:’Regret: Reputation in gregarious societies’, Proc. Int. Conf. Autonomous Agents, Montreal, Canada 2002, pp. 194-195

[24]Srivatsa, M., and Liu, L.:’Securing decentralized reputation management using trustguard’, Journal of Parallel and Distributed Computing, 2006, 66, (9), pp. 1217–1232

[25]Selçuk, A.A., Uzun, E., and Pariente, M.R.:'A reputation-based trust management system for P2P networks', Proc. Int. Symposium on Cluster Computing and the Grid, 2004, pp. 251–258

[26]Xiong, L., and Liu, L.:'PeerTrust: Supporting reputation-based trust in peer-to-peer communities'. IEEE Trans. on Knowledge and Data Engineering, 2004,16,(7), pp. 843–857

[27]Liang,Z.,and Shi, W. :'Analysis of ratings on trust inference in open environments', Performance Evaluation, 2008,65,(2), pp.99-128

[28]http://www.isi.edu/nsnam/ns/, accessed July 2009

[29]802.11:'Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 802.11', 1997

[30]Song, S., Hwang, K., Zhou, R., and Kwok, Y.-K.:'Trusted P2P transactions with fuzzy reputation aggregation', IEEE Internet Computing, 2005,9, (6),pp. 24–34

[31]Zhou, R., and Hwang, K.,'Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing', IEEE Transactions on Parallel and Distributed Systems,2007, 18, (4), pp.460-473

[32]Sun, Y., Yu, W. ,Han, Z.,and Liu, K.J.R.:'Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks', IEEE Journal on Selected Areas in Communications, 2006, 24, (2),pp. 305-317

[33]Nasipuri, A., Castaneda, R., and Das, S.R.:'Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks', Mobile Networks and Applications, 2001,6, (4), pp. 339-349

[34]Jia, Z., Qin, Z., Xu, X., Zhang, R.:'Depth-Expurgation Based Dynamic Trust Evaluation Algorithm for Ad Hoc Neworks', Proc. Int. Conference on Embedded Software and Systems, 2008, pp. 399-404