REVIEW AND ANALYSIS OF TRUST BASED ROUTING IN MANETS

K. Seshadri Ramana
Associate Professor of Dept of MCA,
G.Pulla Reddy Engineering College,
Kurnool-518002, A.P., India.,

Dr. A.A. Chari
Professor Dept of OR&SQC,
Rayalaseema University ,
Kurnool-518002 ,A.P., India.

Dr. N.Kasiviswanth
Head of the Department of CSE,
G.Pulla Reddy Engineering College,
Kurnool-518002, A.P., India.

Abstract:
Ad hoc networks are widely used in military and other scientific areas. With nodes which can move arbitrarily and connect to any nodes at will, it is impossible for Ad hoc network to own a fixed infrastructure. It also has a certain number of characteristics which make the security difficult. Routing is always the most significant part for any networks. One way is to transplant ordinary mechanisms in common networks with some improvement while the other way is to find some other factors such as trust to achieve the objective. This chapter conducts survey about trust in MANETs and current research in trust based routing.

1. Introduction

One of the popular definitions of an ad hoc network is "infrastructure less" network which denotes a network that is free of the conventional routing infrastructure such as fixed routers and routing backbones. One of the characteristics of this network is that the ad hoc nodes are mobile and the primary communication medium is wireless. These ad hoc nodes have the capability of acting as routers and may typically occur in meeting rooms and conferences, personal area networking, battlefield operations, disaster relief and rescue operations, etc.

MANETS that have existing routing protocols make use of the shortest path to the destination when making a selection criterion of their route. One such popularly used lightweight routing protocol is Ad-hoc On-demand Distance Vector Routing (AODV). Selecting such a short path to the destination may not be the best option because these paths may be crammed. They are also likely to have malicious or selfish nodes or are probably badly influenced by other network and physical conditions which the source node may be unaware of. AODV does not have any built-in measures to spot these adverse conditions and its response only consist of information about the number of hops, route freshens, sequence number and the source and destination IDs. SAODV, which is a secure variant of the AODV protocol makes use of one-way hash chains and digital signatures to guarantee message authenticity but it fails to supply any information on route dependability.

One of the ways to improve security and robustness of the protocol is possible when nodes are able to make knowledgeable decisions with regards to route selection that are based on transmitted route requests and further information contained in received route replies. This greatly helps the source node to select a route that optimally serves its purpose. The source node enhances the probability of its packets reaching the destination by utilizing route dependability information.

2. Security Requirements

Two types of authorization decisions are typically required of a router; Firstly, when a routing update is obtained externally, the router has to make a decision whether to alter its local routing information base consequently. This is import authorization. Secondly, a router may perform export authorization each time it gets a demand for routing information. Import authorization is the critical service [1].

Authorization is a matter of policy in traditional routing systems. For instance, gated, a frequently utilized routing program 1 lets the administrator of a router to lay down policies about whether and to what extent to trust routing updates from other routers: e.g., statements such as "trust router X about routes to networks A and B". However, in mobile ad hoc networks, these types of static policies are insufficient (and not likely to be related anyhow). Authorization might necessitate other security services such as authentication and integrity which digital signatures and message authentication codes provide.

Where routing is concerned, confidentiality and non-repudiation neither are nor really considered as critical services though it is often argued that non-repudiation is constructive in ad hoc networks to isolate non-compliant routers: a router A which received an "erroneous message" from router B may use this message to induce other routers that B is misbehaving. This would definitely be helpful if there is a dependable approach of detecting erroneous messages. This does not appear to be a simple assignment.

The problem of compromised nodes is not addressed here as we think that it is not of much importance in non-military scenarios. Availability is also outside of the scope of this paper even though of course it would be advantageous; it does not

appear to be practicable to avert denial-of-service attacks in a network that employs wireless technology (where an attacker can focus on the physical layer without bothering to study the routing protocol).
Consequently, we consider the following requirements in this chapter:

- Import authorization: In this context, authorization does not imply the traditional meaning. It implies that the eventual authority on routing messages with regards to a particular destination node is that node itself. Hence, we will only authorize that route information in our routing table which concerns the node that is sending the information.
- Source authentication: It will help to verify that the node is the same one that it claims to be.
- Integrity: Additionally, we should be able to validate that the routing information that it is sent to us has arrived unaffected.

Data authentication is built when the two last security services are combined together and they are requirements derived from our import authorization prerequisites.

3. Common Attack Scenarios

There are six main properties that any secure networking system should be able to provide: Secrecy, authenticity, integrity, availability, non-repudiation, and access control. It is a breach of security if one or more of these security objectives are contravened by any attack on a computer system. Some of the most common attacks that occur on a distributed computer system [2] are as given below:

- Denial of Service: This takes place when there is non-availability of a network service owing to excess load or breakdown.
- Information theft: This occurs when interpretation of data is through an unauthorized instance.
- Intrusion: This happens when unauthorized person gains admittance to several restricted services
- Tampering: This ensues when data is distorted by an unauthorized person.

Attacks and security goals that an ad hoc network encounters is similar to that of other networks. It becomes easy to gain access to data or to lose the stored (e.g. passwords, cryptographic keys, etc.) data on a node because the substantial network contributors are mobile devices.
Based on these factors, it is all the more important that in an ad hoc network the overall security should not depend on any one factor. One of the most popular means of communication in mobile networks is radio transmission. Eavesdropping on a node is much easier vis-à-vis wired networks. Sometimes intermediate nodes maybe disguised eavesdropper and not linked to some trusted infrastructure. Therefore, end-to-end encryption becomes a vital issue that has to be dealt with in all cases.
This is usually the case because all the nodes of an Ad hoc network work together to make the discovery of network typology and forward packets easy to overcome. These nodes can also produce stale or wrong routes, black holes or routing loops. In addition, there is a strong momentum available for non-participation in the routing system of an Ad hoc network.

Selfish nodes often want to hoard resources for their own use due to consumption of a node's battery power, CPU time, and bandwidth in both the routing system and the forwarding of foreign packets which are limited in mobile devices.
There are three key reasons why a node does not work in accordance with the common routing protocol:

• If malfunctioning nodes merely suffer from a hardware failure or a programming error, it may lead to severe irritation in the routing system of an ad hoc network although this is not deemed an attack.

• Selfish nodes attempt to set aside their own resources, as mentioned above.

• Malicious nodes try to damage other nodes or even the entire network, or concede security aspects in some manner.

Prior to creating a security framework it is worthwhile to an initial overview that is structured and emphasize on what types of attacks are possible in Ad hoc networks so as to prevent malicious nodes from damaging the network. Doing this will ensure that the verification process of picking out attacks that are actually prevented by our security system and where there are still open problems.

4. Classification of Attacks

A structured classification of all probable attacks makes the designing of protection mechanism for existing ad hoc networks more coherent. Though such a type of list might not be exhaustive, it will serve as a checklist for designing a security system. Beginning from an initial attack goal, the attack is developed into sub-goals all along several paths of the tree. Up-to-the-minute attack variations can be effortlessly incorporated by affixing it to the correct tree node [2]. In our notation we note down the trees as simple text where the indentation signifies the tree level. We assign the different trees with capital letters. The root

node of each tree elucidates the overall aim of an attacker, e.g. he wants to save his own resources. If he has dissimilar preferences to accomplish a goal, we register these options labeled with OR as branches of the node. If a goal can be disintegrated into numerous single steps, these steps are also listed as branches labeled with AND. Every one of the children of a node are digitized, so we can noticeably spot each node. If we e.g. write A.2, this means attack tree A (see below), second option (”stop participation in current route”). To reach A.2, the attacker has to take two steps: he must incite a route error AND he must not partake in following route discovery. Again, he has a number of options of provoking route errors, e.g. A.2.1.1, simply create route errors. If a security system averts any of the AND nodes or all of the OR nodes, then this branch of the tree cannot be fulfilled. So if a security system makes a whole attack tree unfulfillable, then the attacker is unable to arrive at his goal - the system is safe. Obviously this is only true if the attack tree is completely comprehensive, which is next to unattainable. But nonetheless attack trees offer a important tool for structuring and analyzing security threats to a system. This type of an analysis should always be the foremost step for executing a security system. Because of restriction of space, we will present only one attack tree at this point. The attack goal, listed in attack tree A on the next page, is to economize the attacker’s own resources, even as having the ability to use the ad hoc network infrastructure for its own reasons. This would be a characteristic attack for selfish nodes which do not want to partake in route discovery or at least do not want to frontward data packets for others. The attack tree is written with an on-demand routing protocol like DSR in mind, but can be effortlessly modified to adapt to other MANET routing protocols.

Attack tree A: Save own resources
OR 1. Do not participate in routing
OR 1. Do not relay routing data
OR 1. Do not relay route requests
2. Do not relay route replies
3. Set hop limit or TTL value in route request/reply to smallest possible value
2. Modify routing data/topology
OR 1. Modify route request
OR 1. Insert additional hops
2. Modify route reply
OR 1. Replace own ID in returned route with detour leading through neighbouring nodes
2. Return completely wrong route, provoking RERR and salvaging
3. Insert additional hops
4. Declare own ID in source route as external
2. Stop participation in current route
AND 1. Provoke route error
OR 1. Create arbitrary RERR messages
2. Do not send ACK messages (causing RERRs in other nodes)
2. Do not participate in following route request (A.1)
3. Do not relay data packets
OR 1. Drop data packets
2. Set hop limit/TTL to 0/1 (causing a RERR)

The attacker’s objectives are attained in three ways:

- By simply disregarding route requests of other nodes
- By masquerading to be part of a route so long that it is certainly not the shortest path
- By just dropping packets

All these three options can be realized in a number of ways, e.g. in A.1.2.2.1 the selfish node does not want to take part in routing so it prefers to change routing/ topology data or more specifically the route replies. If a route reply that contains this node comes back, it deletes itself from the route and places in a list of neighboring nodes (that it presumes are connected to one another) so that when the route response is acknowledged, traffic is detoured around the selfish node.

5. Trust

5.1 What is Trust?

As an important concept in network security, trust is interpreted as a set of relations among agents participating in the network activities. These relations are founded on the proof generated by the prior interactions of entities in a protocol. As a

general rule if these interactions have been true to the protocol, then trust will accumulate between these entities. Trust has also been defined as the degree of belief about the behavior of other entities (or agents) [3]. Establishing trust relationships among participating nodes is vital to facilitate collaborative optimization of system metrics.

Trust and security are two tightly interdependent concepts that cannot be desegregated. For example, cryptography is a means to implement security but it is highly dependent on trusted key exchange. Similarly, trusted key exchange cannot take place without requisite security services in place. It is because of this inter-reliance that both these terms are used interchangeably when defining a secure system.

5.2 Relationship between Trust, Trustworthiness and Risk

The terms trust and trustworthiness appear to be reciprocally used devoid of any apparent distinction. The level of trust is defined as the belief probability varying from 0 (complete distrust) to 1 (complete trust). In this context, trustworthiness is a measure of the actual probability that the trustees will act as expected. Solhaug et al. define trustworthiness as the objective probability that the trustee performs a particular action on which the interests of the trustor depend. Figure 1 give details how trust (i.e., subjective probability of trust level) and trustworthiness (i.e., objective probability of trust level) can be different and how the disparity influences the level of risk the trustor wants to take.

In Figure 1, the diagonal dashed line is assumed to be marks of well-substantiated trust in which the subjective probability of trust (i.e., trust) is equal to the objective probability (i.e., trustworthiness). Depending on the degree to which the trustor is unaware about the difference between the believed (i.e., trust) and the actual (i.e., trustworthiness) probability, there is vagueness about or inaccuracy of the involved risk. To be precise, the subjective aspect of trust brings erroneous risk assessments and incorrect risk management consequently.

Figure 1 exhibits cases in which the probability is miscalculated. In the area under the diagonal line, there is misplaced trust to assorted degrees that the perceived trust is higher than the actual trustworthiness. Even though risk is a fundamental attribute of trust, even well-founded trust, misplaced trust increases risk and as a result the possibility of deceit, as shown in the example marked with a and b in Figure 1. On the other hand, when the perceived trust is lower than the actual trustworthiness as shown in the example marked with a, the trustee is

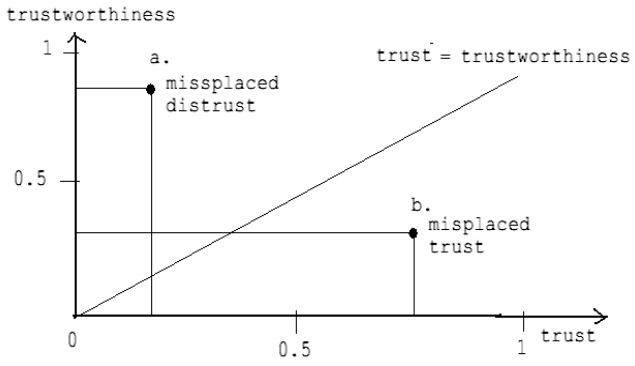


Fig 1. Trust Level

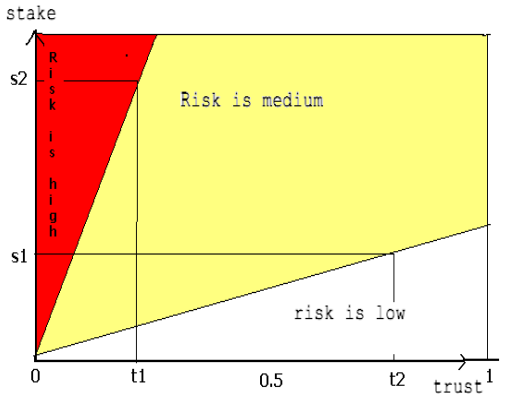


Fig2. Risk and Trust

Distrusted more than warranted. In this case, the trust may mislay potentially good prospects to collaborate with partners of high trustworthiness. The affiliation between trust and risk has been studied.

Figure 2 shows an example of three different risk values: low, medium, and high. When the stake is close to zero, the risk value is low for all trust values. Risk is considered as high, in spite of the projected trust value if the stake is too high. The risk is usually low when the trust value is high. On the other hand, the risk value should be calculated based on the value at stake over and above the risk probability; as shown in Figure 2. High risk exists even for the case of trust value = 1. Also significant are the aspects (or probability) of opportunity and prospect (or the positive consequence of an opportunity). The procurer of rubber

should approximate his or her acceptable risk level in terms of the calculated prospects. In most of the cases, trust is neither proportional nor inversely proportional to risk [3].

5.3 Properties of Trust

In the context of a social network viewpoint, there are three main properties of trust: transitivity, asymmetry, and personalization. Firstly, trust is not perfectly transitive in a mathematical sense i.e., if A trusts B, and B trusts C, it does not guarantee that A trusts C. Secondly, trust is not essentially symmetric which means that it is not identical in both directions. A classic case of asymmetry of trust can be observed in the relations between supervisors and employees. Thirdly, trust is intrinsically an individual outlook. For example, a particular entity will be contrarily evaluated by two people.

5.4 Characteristics of Trust in MANETs

In MANETs, the theory of trust is to be defined with caution because of the distinctive features of MANETs and the intrinsic fickleness of the wireless medium.

The main characteristics of trust in MANETs are given below:

1. The existence of a trusted third party (such as a trusted centralized certification authority) cannot be assumed. Therefore, a decision method to determine trust against an entity should be wholly distributed.

2. Trust should be gauged without too much computation and communication load in a very customizable manner, while also capturing the complexities of the trust relationship.

3. A trust decision framework should not work under the assumption that all nodes are cooperative for MANETs. In an environment that is restricted of resources, selfishness is prone to be rampant over collaboration. For example, to save battery life or computational power.

4. Trust is not static, it is dynamic.

5. Trust is subjective.

6. Trust is not inevitably transitive. The fact that A trusts B and B trusts C does not mean that A trusts C.

7. Trust is asymmetric and not essentially reciprocal.

8. Trust is dependent on context. A could trust B as a wine expert but not as a car fixer. Likewise, in MANETs, if a specific task entails high computational power, a node with high computational power is considered as trusted; while a node that has low computational power but is not malicious (i.e., honest) is distrusted.

6. Routing Misbehavior in MANET

DoS attacks are tough to identify and simple to put into operation by an attacker as there is no hardware required to do so. DoS attacks need more attention as they are considered to be the most susceptible order of attacks for network layers which can lead to the failing of an entire network in the presence of such an attack. The most familiar types of Denial-of-service attacks categorized by researchers are briefly discussed here: [4]

• Rushing Attack – This is a malicious attack that is targeted against on-demand ad hoc network routing protocols that use duplicate suppression at each node. In an on-demand routing protocols, whenever source nodes flood the network with the Route Request packets so as to discover the new routes to the destination, each intermediate forwarding node processes the first Route Request Packet from a particular node to repress the duplicate forwarding. It discards replica packets that appear later. A rushing attacker by passing over some of the routing or MAC layer process can quickly forward these packets. It consequently increases the admission of valid routes for additional data transmission. Almost all the on-demand routing protocols are prone to the rushing attack. For example, DSR, AODV, and secure protocols based on them, such as Ariadne, ARAN, and SAODV, are unable to discover routes longer than two hops when subject to this attack. This attack is also particularly damaging because it can be executed by a comparatively weak attacker.

• Black hole Attack - In black hole routing disruption attack, an attacker first introduces itself in the forwarding group (e.g., by implementing rushing attack), and then it creates a routing black hole, in which all packets are dropped either by sending forged routing packets, the attacker could route all packets for some destination to itself and then discard them, or the attacker could cause the route at all nodes in an area of the network to point "into" that area when in fact the destination is outside the area. As a special case of a black hole, an attacker could generate a gray hole, in which it selectively drops some packets but not others, for example, forwarding routing packets but not data packets. All this results in a poor packet delivery ratio.

• Wormhole Attack – This type of attack is a slighter type of routing disruption attack. A wormhole is created in the network using a pair of attacker nodes A and B and linked via a private network connection. Every packet that A receives from the ad hoc network, A forwards through the wormhole to B, to then be rebroadcast by B; similarly, B may send all ad hoc network packets to A. Such an attack potentially disrupts routing by short circuiting the normal flow of routing packets, and the attackers may also create a virtual vertex cut that they control.

The severity of the wormhole attack comes from the fact that it is difficult to detect, and is effective even in a network where confidentiality, integrity, authentication, and non-repudiation (via encryption, digesting, and digital signature) are preserved. Furthermore, on a distance vector routing protocol, wormholes are very likely to be chosen as routes because they provide a shorter path – albeit compromised – to the destination.

• Jellyfish Attack- Another DoS performed on the transport layer is the subtle jellyfish attack. This DoS attack can be carried out by employing several mechanisms. One of the mechanisms of the jellyfish attack consists in a node delivering all received packets, but in scrambled order instead of the canonical FIFO order. Duplicate ACKs derive from this malicious behavior, which produces zero goodput although all sent packets are received. This attack cannot be opposed successfully by the actual TCP packet reordering techniques, because such techniques are useful on sporadic and non-systematic reordering.

The second mechanism involves performing a selective blackhole attack by dropping all packets for a very short duration at every RTO. The flow enters timeout at the first packet loss caused by the jellyfish attack, then periodically re-enters the timeout state at every elapsed RTO.

The third mechanism consists in holding a received packet for a random time before processing it, increasing delay variance. This causes TCP traffic to be sent in bursts, therefore increasing the odds of collisions and losses; it increases the RTO value excessively; and it causes an incorrect estimation of the available bandwidth in congestion control protocols based on packet delays.

The jellyfish attack is difficult to distinguish from congestion and packet losses that occur naturally in a network, and therefore is hard and resource-consuming to detect.

• Sybil Attack - Sybil attack manifests itself by allowing the malicious parties to compromise the network by generating and controlling large numbers of shadow identities. The fact is that each radio represents a single individual. However the broadcast nature of radio allows a single node to pretend to be many nodes simultaneously by using many different addresses while transmitting. The off-shoot of this Sybil attack is analyzed using Packet Delivery Ratio (PDR) as the performance metric. Theoretical based graphs are simulated to study the influence of Sybil attack in PDR.

Malicious user obtaining multiple fake identifies and pretends to be multiple distinct node in the system malicious node control the decision of the system. The Sybil attack can be categorized into sub categories: presentation of multiple identities simultaneously and presentation of multiple identities exclusively.

The concept of the identifiers exists at different levels and because an identifier only guarantees the uniqueness at the intended level only. Sybil attack can be perpetrated from network layer and application layer where the respective identifiers are IP address and Node ID. Sybil attack can be manifested either by creating new identities or duplicating other identities by disabling them after launching a DoS attack. This mechanism can be either a localized or globalized one depending on the severity of the attack felt by neighboring nodes. Sybil attack can defeat the objectives of distributed environment like fair resource allocation, voting, routing mechanism, distributed storage, misbehavior detection etc

• Neighbor Attack - In this type of attack, an attacker simply forwards the packet without recording its ID in the packet whereas in the normal procedure an intermediate node records its ID in the packet before forwarding it to the next node. In this sort of attack two nodes that are not within the communication range of each other presume that they are neighbors (i.e., one hop away of each other). This misperception results in route disruption.

It is necessary to foresee routing attacks; otherwise when these attacks are carried out (and certainly they will be) we will be unable to identify them as such. If a routing protocol can avert a rushing attack, it will give a certain amount of security against all the other attacks because for all of them, gaining access to the forwarding group is essential prior to introducing a specific type attack.

With reference to the realism of these attacks (real attacks that have been observed against existing networks), there is no or very little existing data. This is possibly due to the fact that ad hoc networks are in practice still used in limited environments such as warfare operations, search and rescue missions, and research centers; while the mainstream architecture for a wireless network is BSS, with "hot spots" offered by various ISPs in airports, train stations, museums, restaurants, and other public places.

6.1 AODV and DSR

In recent years, a variety of new routing protocols targeted specifically at this environment have been developed. Here we consider the differences between the two wirelesses on-demand Ad hoc network routing protocols AODV and DSR that cover a range of design choices.

The major difference is that while the DSR uses source routing, the AODV uses route cache. However, the routing protocol for both of them is on-demand and both of them make use of the best route by choosing minimum hop-count.

Dynamic Source Routing (DSR) is a reactive protocol i.e. it doesn't use periodic advertisements. It computes the routes when necessary and then maintains them. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which the packet has to pass. These routes are stored in a route cache. The data packets carry the source route in the packet header identifying each forwarding hop by the address of the next node to which to transmit the packet on its way to the destination host.

When a node in the ad hoc network attempts to send a data packet to a destination for which it
does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path navigated across the network. The RREP routes itself back to the source by crossing this path rearward. The route carried back by the RREP packet is cached at the source for future use.

Thus in DSR, the destination will reply to all the RREQ packets and send back a RREP packet for each received RREQ. Hence, the source node gets multiple paths to reach each destination and best path will be decided based on minimum hop-count. If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed.

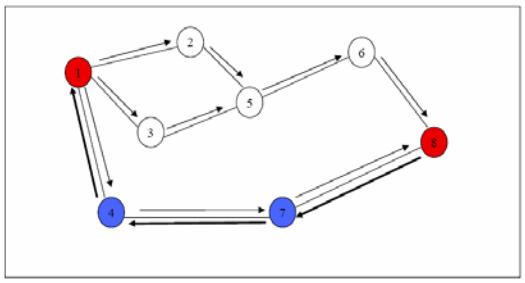
DSR makes very aggressive use of source routing and route caching to find a existing path without any new route discovery or opt for an alternating path to the destination in the presence of route failure or link breakage due to mobility. This will reduce large route discovery overhead and efficiently diminish time delays. No special mechanism to detect routing loops is required. Besides, any forwarding node caches the source route in a packet it forwards for possible future use.

This route cache works well with low traffic load and lesser mobility; conversely it will face some problems when the routes in its cache become expired owing to host mobility. Under these circumstances, the source node will maintain the usage of these expired routes without any notice Furthermore, the expired routes information could also be learned by other nodes and cause pollution to their route caches and that results low throughput

On the other hand, AODV adopts a very different mechanism to maintain routing information. It uses conventional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV depends on routing table entries to transmit an RREP back to the source and, then, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops. All routing packets carry these sequence numbers.

A key attribute of AODV is the maintenance of timer-based states in each node, concerning use of individual routing table entries. A routing table entry is expired if not used recently. A set of predecessor nodes is maintained for each routing table

entry, indicating the set of neighboring nodes which use that entry to route data packets. These nodes are notified with RERR packets when the next-hop link breaks. Each predecessor node, in turn, forwards the RERR to its own set of predecessors, thus successfully removing all routes using the broken link. RERR packets in AODV are intended to inform all sources using a link when a failure occurs which is in contrast to DSR. Route error propagation in AODV can be conceptually envisioned as a tree whose root is the node at the point of failure and all sources using the failed link as the leaves.



AODV Route Discovery

Both the routing protocols use the duplicate-suppression method while forwarding the packet to evade overcrowding and misuse of the important band-width of ad hoc network. The aim of our simulation is to examine how the performance of ad hoc routing protocol changes due to routing misbehaviors. We would also like to evaluate the routing performances of DSR and AODV as both of then share similar on-demand features to check if there is any disparity shown by them.

6.2 Rushing Attack Model

This attack model was explained to you in our earlier work. Our assumption is based on the fact that all the nodes which have equal transmission range and source will choose the optimal path depending on the hop count. For hop count that is identical, the node will opt for the best path based on the utility value which implies a path that has less overcrowding. Each time a team leader accesses a path, the utility value is increased by one which makes the performance of rushing attacks easier for malicious nodes. Three ways that a malicious node can increase the probability that the route that comprises the attacker will be revealed to a certain extent than other valid routes is:

- if an attacker node can expand its transmission range to get access of the far off nodes,
- by obliterating MAC layer delays that are so essential between receiving a request and forwarding it,
- by swiftly forwarding the packets, leaving out some of the routing table operations.

then the malicious node can increase the probability that the route that include the attacker will be discovered instead of other valid routes.

A single RREQ message for each discovery is forwarded at the most for the currently proposed protocols. Both AODV and DSR take into consideration the first packet to arrive to a node by default clearly indicating which packet will be preferred at each hop that will be susceptible to a variation of rushing attack.

6.3 S-HTMRP

One of the key issues concerning MANET is security. In order to realize this objective, a security mechanism has been discussed in this paper for pure ad hoc networks that takes into consideration both routing and packet forwarding. This proposed trust model guarantees security against rushing attack, a variant of DOS attack that is often present in MANET.

What we have tried to propose here is an optional plan of deploying a distributive security mechanism by calculating trust levels from the intrinsic network knowledge, along with randomized packet forwarding mechanism to guarantee security. Hop-count had been substituted by a weight parameter which denotes a value for every path which dynamically alters to mirror path consistency. The resultant routes computed via this mechanism, though somewhat imperfect, have an exact measure of consistency in them.

As a measure of safeguard against malicious attacks in MANET, a number of secure routing protocols have been proposed over the years. A majority of these protocols which endeavor to gain a secure route discovery process are based on PKI

such as digital signature and shared keys. The source node always ascertains the routing reply which is based on digital signatures to ensure that it is arriving from the destination so as to gain route security. ARIADNE offers guarantee that both the source and destination nodes validate the messages. Furthermore, the intermediate nodes also have to include their own digital signature in route request. Conversely, the PKI-based secure routing protocols have a few limitations. It is difficult to implement key generation and distribution - two features that are essential to PKI security – within MANET as it lacks trusted authority. It is clear that the securities of these protocols are founded on similar presumptions: Assurance of security is accessible only if they have an opportunity to configure central or distributed trust authorities in advance. But this pre-configuration requisite challenges the character of MANET which should be unexpectedly and conditionally improvised. Thus, the concept of relative security shield could be more appropriate for MANET. A more effective mechanism we put forth is based on trust evaluation rather than the PKI-based method.

In our proposed mechanism each node will calculate its own trust vector about neighboring nodes by gathering information from normal actions such as packet sending and receiving. This trust vector can be normalized into a single trust value that has been supplied as proof for decision making in routing selection process. Doing this can diminish the ill effects of malicious nodes and also address the problems to some point.

The next section proposed by us is the trust model that is different from most PKI-based schemes discussed earlier. Our primary focus is on post route discovery phase: when packets are being transmitted on discovered routes. The use of cryptography is extremely helpful if we obviously want to defend confidential data that is transmitted over discovered route which must be trusted.

7. EFFECTS OF MALICIOUS NODES IN MANET

Researchers have been able to identify and elaborately explore a range of attacks targeting the network layer in MANET [5]. Malicious nodes have a tendency to disturb the routing protocols and insert themselves between the source and destination path, fabricate packet from source, and assimilate network traffic. Black hole attack is one of the most distinctive offenses in MANET. As discussed earlier, it takes action in two phases: In the routing discovery phase, the malicious node fabricates a route and announces that it has a valid route to the destination node with the sole purpose of dropping packets. In the data forwarding phase, the data packets which are likely to be forwarded to next-hop node are absorbed by the attacker. Hence, a black hole represses the communication of nodes totally. A subtler form of black hole attack takes place when the attacker forward packets selectively; this makes the detection of malicious nodes difficult. Grey hole (which is an extension of the black hole attack) is another kind of attack in which the attacker drops some specific packets by building a grey hole. For example, it absorbs data packets while forwarding route packets. An identical attack that often inflicts MANET is the wormhole attack where a tunnel is created between two conspiring malicious nodes that are connected via a speedy wired network. Another typical fabrication attack is the rushing attack which often targets on-demand routing protocols that restrain duplicate packets at each node. This results in the rejection of legitimate routing messages by nodes as they presume them to be duplicate copies due to the suppression of legitimate routing messages by these malicious routing messages

When we contrast this to a scenario devoid of malicious nodes, we come to the inference that packet delivery ratio is substantially diminished because of the existence of a minute portion of malicious nodes in the network. Additionally, simulation outcomes prove that the performance of an ad hoc network drastically worsens when speed of nodes increase. This can be illustrated by the fact that the more momentum the malicious nodes move with, the larger is the region covered by it. This confirms the bad consequences that even a moderate number of malicious nodes can have on the ad hoc network performance levels. Owing to this, counter measures to suppress misbehaviors become a mandatory introduction for us.

8. TRUST ROUTING

In this part [5], we show you how to incorporate our trust model into MANET routing protocols by presenting their efficacy. We opt for the constantly improvised DSR and AODV models despite the fact that they are still under development.

We first present a brief introduction of these two protocols. We then discuss how to compute each fraction of trust vector for both these protocols. We finally present the entire trust routing process by investigative details of each routing phase.

8.1. DSR and AODV

The Dynamic Source Routing (DSR) protocol is a typical on-demand routing protocol. One of the chief characteristics of this protocol is that all the data packets that are sent in DSR have complete routing information from the source node and thus, they need not be dependent on intermediate nodes to decide which one is next hop. In the routing phase, a node initially broadcasts a *ROUTE REQUEST* message if it requires a route to the destination. If the recipient node is ignorant of the required destination, it affixes its own address to this request packet and rebroadcast this new *ROUTE REQUEST* message. A *ROUTE REPLY* message containing the complete route information from the source to the destination is sent if the recipient node is the destination or an intermediate node which has a route to the destination in its route caches. Finally, the source node receives these route request feedbacks. However, the source node needs to make a decision based on some criteria such as number of hops or latency because there may be many *ROUTE REPLY* messages arrived at source node. In routing maintenance phase, all nodes should append usable routing information from which they have been forwarding or overhearing any packets into their own route caches. If nodes found some route had been broken, a *ROUTE ERROR* message should be sent to each node which had used this specific route.

The Ad-hoc On Demand Distance Vector (AODV), as its name suggests, is a distance vector routing protocol which is exclusively designed for mobile ad-hoc networks. It is also a typical routing of on-demand protocols in MANET which locate the routes only when asked. The route discovery and maintenance phases of the AODV are adapted from the DSR and Destination-Sequenced Distance-Vector Routing (DSDV) to some extent. One distinguishing characteristic of AODV is that it makes broad use of sequence numbers in route control packets so as to avoid the problem of routing loops. In routing discovery phase, a *ROUTE REQUEST* message is also broadcasted by the source node to find an unknown route. An identifier, source IP address, destination IP address, hop counter, sequence number and other control flags are some of the things included in a *ROUTE REQUEST*. The identifier field is used to uniquely symbol the *ROUTE REQUEST* message; the hop counter records the number of intermediary nodes between the sources to the destination; the sequence number is used to decide which packet is newer than the others. If the intermediary node which received the *ROUTE REQUEST* message does not know about the source IP address and message identifier or it does not have a newer route to the destination which is generally indicated by a larger sequence number, it should first add one hop to the hop counter and then rebroadcast this updated route request packet; this intermediary node should keep a reverse route to the source in a definite interval of time. The destination node should send a *ROUTE REPLY* message to the source node if it receives the *ROUTE REQUEST* message or intermediary node has a newer route to the destination. Then again, the sequence number of destination, source IP address, destination IP address, hop counter and other control flags should be included in the *ROUTE REPLY* message.

All intermediary nodes that receive the *ROUTE REPLY* message should augment the hop counter and broadcast this message on the reverse route lines. In routing maintenance phase, the AODV use periodic HELLO messages to discover status of physical link. If a link break discovery for some active route is made, the detector would send a *ROUTE ERROR* message to its neighbors so that nodes which had used that particular route could make their own route information up to date.

Security Flaws of AODV

Malicious nodes can perform many attacks just by not following AODV rules since AODV has no security mechanisms. A malicious node M can carry out the following attacks (among many others) against AODV [1]:

1. Impersonate a node S by forging a RREQ with its address as the originator address.

2. When forwarding a RREQ generated by S to discover a route to D, decrease the hop count field to increase the chances of being in the route path between S and D so it can evaluate the communication between them. A variation of this is to increment the destination sequence number to make the other nodes assume that this is a ‘fresher’ route.

3. Impersonate a node D by forging a RREP with its address as a destination address.

4. Impersonate a node by forging a RREP that claims that the node is the destination and, to add to the impact of the attack, claims to be a network leader of the subnet SN with a big sequence number and send it to its neighbors. Thus it will become (at least locally) a black hole for the whole subnet SN.

5. By not forward certain RREQs and RREPs selectively, not responding to certain RREPs and not forwarding certain data messages. This kind of attack is particularly hard to even detect because transmission errors have similar effect.

6. Forge an RERR message by pretending it as the node S and sending it to its neighbor D. The RERR message has a very high destination sequence number dsn for one of the remote destinations (U). This might cause D to update the destination sequence number corresponding to U with the value dsn and, therefore, future route discoveries performed by D to take a route to U will fail (because U's destination sequence number will be much smaller than the one stored in D's routing table).

7. In accordance with the existing AODV draft, the originator of a RREQ can put a much bigger destination sequence number than the actual one. Additionally, sequence numbers wraparound when arriving at the maximum permissible field size value. This facilitates a very easy attack in where an attacker is able to set the sequence number of a node to any desired value by just sending two RREQ messages to the node.

8.2. Trust Computation in Routing

When we evaluate the experience of a trust vector [5], an important observation that needs to be undertaken is the measurement of the number of out-coming packets the immediate neighboring node has genuinely sent. To understand this, node participation in packet forwarding should be monitored. This can be done by placing all the nodes in the promiscuous mode at all times irrespective of the nodes transmitting control packets or data packets. When it comes to know that its immediate neighbor nodes are forwarding the packet, it checks packet integrity to determine that the packet is unmodified by other malicious nodes. If it observes that the neighbor node passes the integrity test, the out coming packet counter of this neighbor node should be incremented. But if they fail to pass the integrity test or if the neighbor node does not cooperate in forwarding the packets it is supposed to, its corresponding forwarding counter will remain unchanged. After sometime, its experience value would be exceedingly low on account of malicious behavior.

While investigating knowledge of trust vector, use of link-layer acknowledgements is made as the underlying MAC protocol gives feedback of the successful delivery of the transmitted data packets. By doing this, we can easily calculate the MAC layer quality. Some nodes in MANET cannot directly acquire experience and knowledge of trust vector, but only from recommendation of others. This limitation occurs because MANET usually has a transmission range about 100 or 200 meters. We need to consider how trust value can be effortlessly transmitted from one node to the others without much troublesome overhead on network commutation in the evaluation of recommendation of trust vector. Here we proposed a proper scheme based on route discovery process by choosing to expand the trust values of nodes along with the ROUTE REQUEST message. The node should add its trust value about preceding node from which it received the ROUTE REQUEST message before sending a ROUTE REQUEST message to next hop. By doing this, we will enable the trust value to evenly spread along with ROUTE REQUEST message. For example, firstly a ROUTE REQUEST message from node A to node B and node C, then before node B should propagate request message to node D and node E, it should appends its trust value of node A BTA to this request message. Subsequently, node E and node E should go on flood route request, at this time, the node D's and node E's trust value of node B are easily spread along with route request. So does node C.

8.3 Trust Integration In Routing

In this part [5], we will discuss how trust model is integrated into the entire routing process in DSR and AODV protocols. Before embarking on to the discovery of a new route, we firstly need to investigate the routing table or cache to see if there is a workable route already in existence to the destination. In case, there is no proper route information known, a route discovery phase is triggered by flooding the ROUTE REQUEST message. Later on, when we pick a node as the next hop, a number of criteria can be judged such as number of hops or latency. This is done because as a rule, the fastest node that reaches the destination is preferred.

We change old rules in trust routing protocols - we assign trust value as cost of nodes. Hence, whenever a node has some packets that need to be sent or forwarded, it should first scrutinize the routing table or routing cache for all probable paths which can arrive at the same destination. After doing this, it makes a comparative study of trust value of all the next hops in those candidate paths and selects the path that is most proper and has the highest trust value. If the next hop is unknown by the sender or forwarder before, the least number of hops path to the destination is selected. All nodes should select neighbor nodes that have a trust value greater than or equal to the predefined trust value threshold in order to minimize costs. A local link repair process is initiated if there is unavailability of next hop in candidate paths according to trust level which should be greater than the trust threshold. Therefore, if there has been any data packets which should be forwarded with an improper trustworthy next hop, we would stop forwarding action and buffer it for a certain interval of time in which another route discovery would be started to find an alternate route that should be trustworthy. The packet would be sent to an alternate route if such a route is discovered. If this is not the case, then a ROUTE ERROR message would be sent to the source node informing it of the link error. Based on the various

application environments, the trust value can be set up in several ways. Generally, a higher trust value threshold indicates a rigid forwarding policy that needs to be complied by all nodes. This is specially preferred in application where accurate throughput and correct forwarding is a requirement. Despite this preference, it is possible that a misdiagnosis can take place in cases where a heavy traffic node is mistakenly thought to be a malicious node. Conversely, a lower threshold suffices the requirement to detect nodes which exhibit continual malicious behavior. Besides, packets dropping may occur due to high mobility and rapid change in topology irrespective of the nodes been malicious or benevolent. Packet drop can also take place due to MAC layer collisions as a result of overcrowding in communication. Numerous packet droppings are swayed by the inherent nature of MANET such as mobility pattern or traffic load. This statement can be corroborated by the fact that packet delivery ratio is below 100% all the time even when there is no malicious nodes in MANET. Fortunately, this ratio of packet droppings which are due to network characteristic can be ignored even under high mobility situations. So detection of malicious nodes in adherence to suitable trust value threshold under diverse conditions can still be a successful mission to undertake.

9. Existing AODV Trust Framework

First [6], the changes made in our schemes to the existing AODV protocol formats are described. Then the types of trust used in our scheme, their definitions, and their dynamic computation are explained. Afterwards, the use of Choke packets, the local repair mechanism and the handling of malicious nodes are described. The various types of incentives and penalties used in our scheme are explained as well. Lastly, we describe the criteria for the source node to select the better route of the available routes to the destination.

Route List						
Rid_1	$Next\_hop_1$	$Last\_hop_1$	Hop_count_1	$Timeout_1$	ATV_1	RSV_1
Rid_2	$Next\_hop_2$	$Last\_hop_2$	Hop_count_2	$Timeout_2$	ATV_2	RSV_2
Rid_3	$Next\_hop_3$	$Last\_hop_3$	Hop_count_3	$Timeout_3$	ATV_3	RSV_3
:	:	:	:	:	:	:
:	:	:	:	:	:	:

Fig. 1 (a) Route Table Entry

Node ID	*R*	Node Trust
------	--------	------
------	--------	------

Fig. 1(b) Neighbor Table

The main changes our schemes make to the AODV protocol are:

A · Each node maintains an additional data structure called the *Neighbors' Trust Table*. It comprise of neighboring node IDs, their corresponding trust and *r* values. See Fig. 1b.

B · Each route table entry for a given destination stores all the routes from that node to the destination with the highest DSN. The corresponding route trust values as advertised by the nodes (ATV) and the computed RSVs are stored. Each route to a destination can be identified by unique *Rid*. The *Rid* with the highest RSV is stored in the *Advertised Rid* field and advertised to the upstream nodes (Fig. 1a).

C · The RREQ packet has two additional fields: the *Omit Node Flag* and the *Omit Node ID*. The Omit Node Flag, if set, indicates that the node ID mentioned in the Omit Node ID field should be precluded from the route to the destination. The rest of the packet is same as that in the AODV protocol. See Fig. 2a.

D · The RREP packet has added fields to accommodate the route trust and the recommender node's ID. For every RREP, the intermediate node increments the number of hops by one and caches the route trust sent by the downstream node from the route trust field. If the node has individually computed its own trust value on the route then update the route trust and the recommender ID fields with its own route trust value and its node ID (Fig. 2b)..

E · R_ACK is the modified version of the RREP-ACK message of the AODV protocol. The REP-ACK is used to acknowledge the receipt of a RREP (with its *A* bit set) over an unreliable link. Apart from performing the same task as RREP-ACK, an R_ACK functions as a report packet. A report packet would be initiated by the destination to inform the source and the intermediate nodes of the number of packets it received so far since the last transmission of R_ACK. If the value of *Type* field is 3 (the only value used in AODV), then it acts as a RREP ACK. We use 4 as the value of *Type* to send R_ACK as a report packet. See Fig. 2c.

F · the CHOKE packet is a new packet type introduced by our schemes. It is broadcast by a node to its one hop neighbors to indicate congestion in the region. It contains a field for *Node ID*, and a field for *Timestamp*. The *Lifetime* field indicates the lifetime of the packet. The node can broadcast a new CHOKE packet with the *Lifetime* field set to zero if the congestion is cleared earlier than expected. Congestion identification and response are further explained in fig 2.

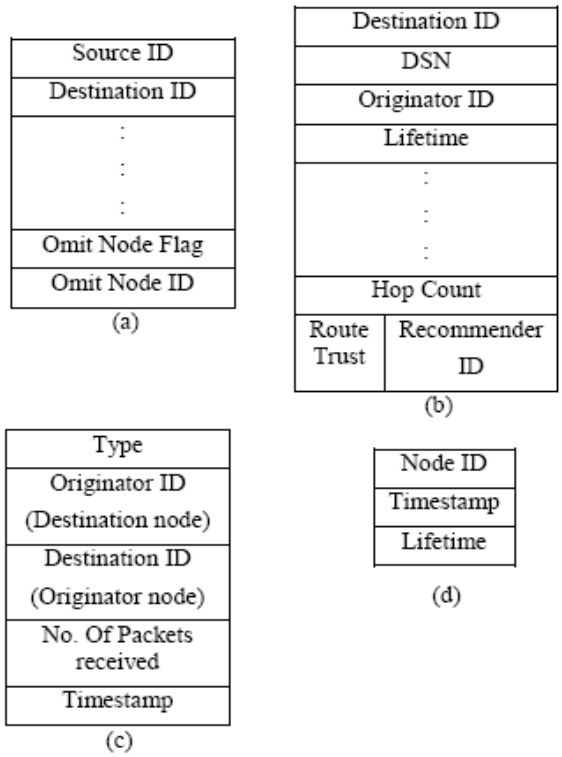


Fig. 2 Packets: (a) RREQ (b) RREP (c) R_ACK (d) CHOKE

9.1. Trust Framework and its Computation

Route trust and node trust are the two trust values associated with the protocol. Route trust is computed by every node for each route in its routing table. It is a reliability measure with which a packet, forwarded by the node on that particular route, can reach the destination. A node trust is also maintained on each of its neighbors by every node. Node trust aids a node *X* in credibly appraising recommendation of a neighbor *N*'s trust on a route passing through *N*. The current number of route requests (*r*) generated by *N* that will be entertained by *X* is directly proportional to *X*'s node trust on *N*. Whenever a network is setup at the initiation stage, the proposed scheme functions almost like AODV.

Initially, a node does not have any information about the credibility of its neighbors, i.e., nodes can neither be trusted fully nor be distrusted fully. So all nodes have 50% initial node trust with $r = R/2$ and this trust remains unaffected until time *tinit.* The route trusts are initially unknown. RREQ's are sent by source node *S* and the routes are established to the destination node *D* as in AODV. All RREQs have the *G* flag set so as to set-up reverse route from *D* to *S*. Each node keeps track of the number of packets it has forwarded through a route. *D* periodically sends R_ACK packets to *S* at a fixed interval between *S* and *D*. All the nodes on the route are able to read the R_ACK packets. Each intermediate node on the reverse route from *D* to *S* checks the R_ACK packets to compute its route trust. Route trust is calculated as a ratio of the number of packets received at *D* to the number of packets forwarded by the node under consideration (from *S* to *D* on that route). Node trust [6] is computed based on the difference between the nodes' ATV to the destination and the OTV computed for the current data transfer. When a node *X* forwards or generates an *RREP*, *X* advertises its trust on the route under consideration to its immediate upstream node *P*. Node *P* caches this route trust value as ATV of node *X* on that route and compares it with the OTV. The node *X* receives an incentive if the OTV is within an admissible range of ATV. If it is not, then it is penalized. The penalties and the incentives are inversely proportional to the node's distance from the destination: the farther a node from the destination, lesser is the information it has on the downstream nodes' behavior. A node is solely responsible for packets reaching the destination if it is only one-hop from the destination. Hence its trust on the route is based on solely its own behavior and link between itself and the destination. A node would have less information about the downstream route conditions and node behavior if it is, say, three hops away from the destination. There is a limit on the rate of RREQs a node can generate in AODV. These generated RREQs help in establishing a route only if the node's one-hop neighbors forward the RREQ or reply to it. The incentives and penalties increase or decrease the *r* value proportionally for each node respectively. The greater the value of *r* the greater is the rate of RREQs (generated by the

node) forwarded or replied to, by its neighbors. Therefore, a node *X*, with all its neighbors having node *X*'s *r* value as zero, cannot establish a route to any destination.

9.2. Choke Packets, Local Repair and Malicious Nodes

In AODV, incoming packets are simply dropped if a node is congested. Our framework tries to curtail packet drops by allowing a node to notify its neighbors of its congestion at the Network layer. A node broadcasts a CHOKE packet (see figure 2(d)) to its immediate upstream nodes if it is unable to forward packets along that route due to traffic congestion. The node broadcasts a new CHOKE packet with a new lifetime if congestion persists after expiry of the lifetime of the previous CHOKE packet. The immediate upstream node *P* promiscuously monitors node *X* to verify its congestion if a node *X*, claims to be congested for more than time *tcongmax.* Node P will penalize node *X* for selfish behavior if node *P* finds that node *X* is purporting congestion. The penalty would be less severe than that imposed for advertising wrong route trust information. *P* tries to find an alternate route to *D* using the *local repair* mechanism inherent in AODV if a node P finds that node *X* is genuinely congested. Node *P* uses the option of precluding the congested node *X* from being a part of the route during the *local repair.* When its ATV on a route is not within a tolerance limit of OTV, a node is said to be advertising inaccurate route trust values. The value of '*r*' for that node maintained by its immediate upstream neighbor, say *P*, decreases if a node *X* repeatedly advertises inaccurate route trust values. After it falls below *rthresh*, *P* promiscuously monitors *X*. *P* isolates *X* by not forwarding any packets through *X* and not entertaining any *RREQ*'s from *X* If *P* observes that node *X* is maliciously dropping packets. Node *P* then summons the *local repair* mechanism to find the alternate route to get to the destination. Also, *P* broadcasts *X*'s malicious behavior using some known security mechanisms. Each node receiving this broadcast can autonomously decide whether to label *X* as malicious. This is viable through certain techniques. The flagged malicious node *X* would remain isolated by node *P* for time *tmal*. After *tmal*, the node *X* is treated as a new node in the network. If node *P* does not succeed in locating an alternate path then it would send the *RERR* message to its immediate upstream node (say *Pp*) on that route. Then node *Pp* tries to find an alternate route to the destination sing the *local repair* mechanism.

9.3. Incentives and Penalties

Tolerance [6]:

$$ATV - r_{thresh} \leq OTV \leq ATV + r_{thresh}$$

Incentives:
- When the OTV is within the range.

Penalties:

- When the OTV is below the range.
- When the OTV exceeds the range.
- When a node purports congestion.

Incentives and penalties are inversely proportional to the nodes' distance from the destination, as discussed above. Successive nodes (in the route from S to D) are given incentives and penalties by their immediate upstream neighbors. If the above condition is satisfied, (the route performs as expected), then incentives are given nodes. The incentive will be:

$r = r + I$ Where I is given by:

$$I = r \times \left(\frac{i}{H} \right)$$

Where *i* is the incentive coefficient and *H* is the distance of the node from the destination node in terms of number of hops.

If the OTV falls below a certain lower tolerance threshold then the successive nodes are penalized as per the following equation:

$r = r - P$, where *P1* is given by:

$$P_1 = r \times \left(\frac{p}{H} \right)$$

Here *p* is the penalty coefficient.

A node is penalized even if the OTV is higher than the range. This is done in order to stop nodes from advertising route trust which is lower than the actual route trust. This penalty would be much less severe than the above penalty, P1.

$r = r - P$, where P2 is given by:

$$P_2 = r \times \left(\frac{q}{H} \right)$$

Here *q* is the penalty coefficient.

A node is lightly penalized for broadcasting superfluous CHOKE packets. The reason for imposing this penalty is to deter nodes from acting selfishly by not forwarding other nodes' packets. The penalty is as shown below:

$r = r - P$ Where P3 is given by

$$P_3 = r \times \left(\frac{z}{H} \right)$$

Where *z* is the penalty coefficient**.**

The values of the constant parameters, *i, p, q, z* lie between 0 and 0.1. All these values can be chosen based on the purpose of the network. All the values should be in unity with the
principle:
$0 \_ z \_ q < p \_ i \_ 1$

This specifies that that the coefficients of penalty for selfish behavior (*P3*) and penalty for advertising lower route trust values (*P2*) are less severe than coefficients of penalty for under-performance (*P1*) and incentive for noble behavior (*I*).

9.4. Route Selection Criteria:

The node *S* may get numerous *RREP* packets in reaction to its *RREQ* packet to *D*. The *route selection criterion* is based on two factors: node trust of the immediate downstream neighbor *N* that suggested the route and on the route trust node *N* has on the required route. The route selection standard is inversely proportional to the number of hops in the route. Several techniques can be created for choosing a route from the existing routes. A few of them are proposed by us.

A source node computes the *Route Selection Value* (RSV) for all its accessible routes to the destination and the route which has the highest RSV is finally chosen. If two routes have similar RSV then the following factors are used to sever the tie:

· The route with highest route trust is chosen.

· If the routes have similar route trust values, then the route with the highest immediate downstream neighbors' node trust (as perceived by the source/immediate upstream node) is selected.

· If the immediate downstream neighbors' node trust is also the same, then the shortest route is preferred.

If all the above are identical then it will randomly pick from amongst those routes which have same RSVs.

Table 1 below shows the symbols used in calculating the RSV and their respective meanings.

Symbol	Meaning
T_{ind}	Trust on the individual neighbor (Node Trust)
T_{avg}	Average of the trusts of all the neighbors that forwarded/generated *RREP*.
RT_{ind}	Trust the individual neighbor has on the Route (Route Trust).
RT_{avg}	Average of all the Route Trusts obtained from individual nodes which forwarded/generated the *RREP*
H_{ind}	Number of Hops in the route proposed by the individual node in its *RREP*
H_{avg}	Average of all H_{ind} s' obtained from individual neighbors which forwarded the *RREP*.

Table 1. Calculating RSV

The following is one method used for computing RSV:

$$RSV = \frac{T_{ind}}{T_{avg}} * RT_{ind} * \frac{H_{avg}}{H_{ind}}$$

Thus, the equation is normalized with respect to node trust and number of hops to destination.
An alternate method for RSV computation is:

$$RSV = \alpha_1\left(\frac{T_{ind}}{T_{avg}}\right) + \alpha_2\left(\frac{RT_{ind}}{RT_{avg}}\right) + \alpha_3\left(\frac{H_{avg}}{H_{ind}}\right)$$

Where a1,a2,a3 are weights assigned for node trust, route trust, and number of hops, respectively. The values of a1, a2,a3 lie between 0 and 1 satisfying the condition a1 + a2 + a3 = 1. This provides the network administrator the liberty to decide the weights for a1,a2,a3 based on the deployment environment. If the network is proposed for highly secure data, such as in a military scenario, then the values of a1 & a2 should be much higher than the value of a3. This implies that more weight is to be given to the route trust and node trust as compared to the route length.

10. Existing DSR Trust Framework

Here we present the improvised form of Trust Enhanced Route selection that needs to be applied to the DSR protocol [7] so as to make the routing protocol security more strong. In our proposed protocol, what we have selected is a more reliable and secure route to the destination based on the trust value of all nodes. This is in contrast to the typical process of route selection in the DSR protocol which tends to select the shortest route destination.

• Every node in the network stores a trust value that is symbolizes the value of the trustiness to each of its neighbor nodes. Adjustment of trust value is done on the basis of experiences that the node has with its adjacent node.

• The trust value for a neighbor node is upgraded when a node receives data packets or acknowledgements from it. Based on trust formation strategy, an initial trust value is assigned for a neighbor node that is initially encountered. If a route consists of familiar nodes, the trust values of these neighbor nodes are used as a basis of assigning initial trust value.

• The trust value for a neighbor node should be decreased, if a requested acknowledgement was not received.

10.1. Components of the Proposed Protocol

The following components comprise the proposed protocol:

1. Trust Unit
   1.1. Initializer
   1.2. Upgrader
   1.3. Administrator
2. Monitor
3. Router

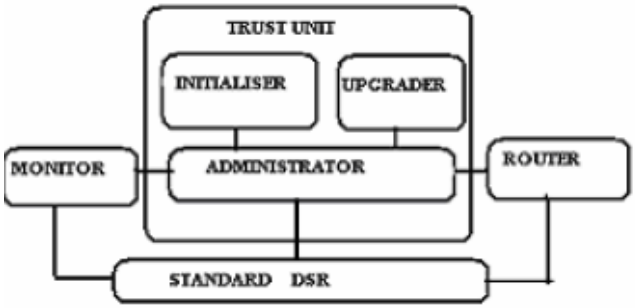


Fig. 1 Components of Trust Enhanced DSR

Trust Unit

*Initialize Module***:** This type of module is used for assigning a trust value for new mobile nodes in a network that are unidentified. In an environment with many malicious nodes, it is best to assign a low trust value. However, if a route has identified nodes, the trust value of these nodes is used as a basis for assigning initial trust value for the new node.

Upgrader Module: This module is utilized for implementing the functions for upgrading trust. Updation of trust values is dependent on the experience a given node has in that specific situation. Two functions that are used for upgrading the trust value for each node encountered in the route - previous trust values and the experience values:

$Tu(Ev,Tv) = (1-C) *E + C*Tv$

Where

Tu: The upgraded trust value

Tv: The existing trust value

Ev: The experience value

C: A constant to express the inflation of trust

The experience value include knowledge of the Acknowledgements received and of the data packets received.

Administrator: The Administrator module offers ways of asking for information about trust values while storing trust information about all known nodes during run time. Therefore, it is utilized as an interface interface between the existing DSR protocol on one hand and the Initialize and Upgrade modules on the other hand.

Router: This kind of module is used to investigate routes based on the relationship status of nodes. This again is dependent on the trust values of nodes in each route and it picks out a route with good Relationship (Friend) based on evaluation. This indicates that the least number of malicious nodes are ensured by taking into consideration the best route with the highest trust rating.

Monitor: The aim of this module is to regulate trust values from acknowledgments that are received. It is essential that a missing acknowledgement is detected quickly as the trust values are used on routing selecting decisions

The trust upgrade module upgrades the trust values for nodes on the stored route whenever an acknowledgement is received. The packet is considered dropped if a requested acknowledgement is not received. So the trust values should be adjusted in a negative way.

10.2. Nature of Relationships between Neighbors in an Ad Hoc Network

In an ad hoc network, the correlation of a node *i* to its neighbor node *j* can be any of the following types:

i. Node i is a stranger to neighbor node j

If node i has never sent or received messages to or from node j, then their trust levels with each other will be very low. Any new node that enters in an ad hoc network will be a *stranger* to all its neighbors. High changes of malicious behavior from stranger nodes is exhibited.

ii. node i is an acquaintance to neighbor node j

If node i has sent or received few messages from node j, their mutual trust levels are neither too low nor too high to be dependable. The nodes need to be observed as there are chances of malicious behavior in them.

iii. node i is a friend to neighbor node j

If node i has sent or received ample messages to or from node j, then the trust levels between them are convincingly high. Moreover, there is a very less probability of nodes misbehaving.

The above relationships are represented as a Relationship table in each node of an ad hoc network.

Consider the node 1 in Fig 2. The Relationship table of node 1 is represented as shown in Table I. A *Relationship estimator is* used in each node to assess the trust level of its neighboring nodes. The relationship estimator verifies the trust level from the administrator module of trust unit and makes a decision regarding the relationship status of each node based on the threshold value. The methods of threshold fixation are discussed below:

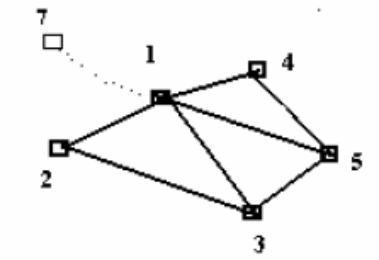


Fig. 2 Nodes in an Ad hoc Network

TABLE I
RELATIONSHIP TABLE FOR NODE 1 IN FIG. 2

Neighbors	Relationship
2	F
3	F
4	A
5	F
7	S

The threshold trust level for a stranger node to turn into an acquaintance of its neighbor is represented by Tacq and the threshold trust level for an acquaintance node to become a friend of its neighbor is denoted by Tfri. The relationships are represented as:

R (node i → node j) = F when $T \geq Tfri$
R (node i → node j) = A when $Tacq \leq T < Tfri$
R (node i → node j) = S when $0<T \geq Tacq$

In addition, the relationship between nodes is asymmetric, (i.e.,) R (node i → node j) is a relationship evaluated by node i based on trust levels calculated for its neighbor node j. R (node j → node i) is the relationship from the friendship table of node j. This is computed by depending on the trust levels allocated for its neighbor. Asymmetric relationships propose that the direction of data Flow may be more in one direction. In other words, node i may not trust node j in the same manner as node j trusts node i or vice versa.

10.3. Routing Mechanism

Whenever a node wants to relay messages to a node located far off, it will send the ROUTE REQUEST to all the neighboring nodes which in turn send the ROUTE REPLY back to it where is sorted by trust ratings. The path that is most trusted is selected by the source. A path with a friendly one hop neighbor is chosen for transferring message. However, if its one-hop neighbor node is an acquaintance, and if the one hop neighbor of the second best path is a friend choose F. Likewise, a most favorable path is preferred based on the degree of friendship that exists between the neighbor nodes.

TABLE II
PATH CHOSEN BASED ON PROPOSED SCHEME

Next hop neighbor in the best path P1	***F***	F	A	***A***	S
Next hop neighbor in the best path P1	A	***F***	***F***	S	***F***

The source opts for the shortest and the next shortest path. Message transfer is done immediately whenever a neighboring node is a friend. This removes the overhead of invoking the trust estimator between friends. But if a neighboring node is an acquaintance or stranger, transfer is done on the basis of ratings. This protocol will join the DSR protocol if all the nodes in the ad

hoc network are friends. The Threshold parameters are design parameters. When simulation is to be carried out, it should be done with appropriate values or all the parameters and the threshold thrust levels in order to achieve optimal performance. There is a tradeoff between offering good security in ad-hoc networks and overall throughput of the network. Therefore, choosing an optimal value is critical for the good network functionality.

References:

[1] Manel Guerrero Zapata and N. Asokan. Securing Ad hoc Routing Protocols. Communication Systems Laboratory Nokia Research Center, 2002.

[2] Frank Kargl, Stefan Schlott, Andreas Klenk, Alfred Geiss, MichaelWeber. Securing Ad hoc Routing Protocols. Email: surname.givenname@informatik.uni-ulm.de.

[3] K.Seshadri Ramana, Dr. A.A. Chari, Prof. N.Kasiviswanth. A Survey on Trust Management for Mobile Ad Hoc Network. International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010

[4] Moitreyee Dasgupta, S. Choudhury, N. Chaki. Routing Misbehavior in Ad Hoc Network. ©2010 International Journal of Computer Applications (0975 - 8887)

[5] Wei Gong1, Zhiyang You, Danning Chen, Xibin Zhao, Ming Gu, Kwok-Yan La. Trust Based Routing for Misbehavior Detection in Ad Hoc Networks. Journal of Networks, Vol. 5, No. 5, MAY 2010. Email: ph.d@live.cn.

[6] Kamal Deep Meka, Mohit Virendra, Shambhu Upadhyaya. Trust Based Routing Decisions in Mobile Ad-hoc Networks. Department of Computer Science and Engineering State University of New York at Buffalo, Buffalo, New York 14260. {kmeka, virendra, shambhu} @cse.buffalo.edu.

[7] N. Bhalaji, A. R. Sivaramkrishnan, Sinchan Banerjee, V. Sundar, and A. Shanmugam. Trust Enhanced Dynamic Source Routing Protocol for Adhoc Networks. World Academy of Science, Engineering and Technology 49- 2009