A Trust Distribution Service for MANETs

Humphrey Rutagemwa and David Kidston
Communications Research Centre
Ottawa, CANADA
{ humphrey.rutagemwa, david.kidston }@crc.gc.ca

***Abstract*—Mobile ad hoc networks (MANETs) are very useful for communications in areas with very little or no infrastructure. However, that very lack of an authoritative management infrastructure makes it imperative that nodes cooperate by passing traffic fairly and securely. The use of a reputation or trust management system can identify failed, misbehaving or malicious nodes. One aspect of trust management that has been relatively overlooked is the trust distribution service. In this paper, we use a trust model that uses local observations from all nodes in the network to calculate per-node reputation using a trust matrix. We have developed a gossip-based distribution service that reduces network-wide reputation convergence time by prioritising critical trust updates. Simulations show that this distribution service substantially outperforms naïve gossip and provides insight into optimal node density for epidemic-based routing schemes in resource constrained networks.**

Keywords-distribution service; trust-based security; epidemic routing; MANETs

I. INTRODUCTION

In order for a mobile ad hoc network (MANET) to function at the network layer, nodes must cooperate in the forwarding of packets from sender to receiver. However, misbehaving nodes may act selfishly to gain an advantage by saving resources or getting an unfair share of the resources. Malicious nodes may be attempting to deny resources to others by interfering with the communication protocol. In MANETs, reputation systems are often used to deal with routing related problems, where nodes are not forwarding packets as they should according to the chosen routing protocol [1].

While there has been significant prior work on defining trust and system models in the literature, little work has been done on the related problems of trust distribution and reputation calculation and services [2-4]. Reputation service is based on the aggregate trust opinion of group members. In order to calculate reputation, local knowledge must be distributed to the rest of the network so that there can be an informed consensus. This requires communication overhead with a trade off in timeliness. Based on the distributed reputation calculation some system level action can be taken to rebalance the use of resources. For example, other nodes could refuse to forward packets from nodes with a very low reputation until it was again in compliance with its forwarding obligations.

In this paper, we describe a novel trust distribution developed to quickly calculate global reputation values in a MANET. Epidemic routing, also known as the gossip protocol, is used for the robust and scalable dissemination of local trust information so that nodes can calculate a global reputation value independently. In order to minimise the calculation time, a prioritisation scheme has been developed based on our previous work [5] that alters the order in which the gossip protocol forwards local trust information. Our simulations show that a prioritisation scheme significantly improves the convergence time required for nodes to have an accurate picture of the reputation of all other nodes in a MANET.

The paper continues as follows. Section II describes the related work in the area of trust-based security and the gossip protocol in MANETs. Section III specifies the trust model. In Section IV, we go into the details of the opportunistic trust distribution system. Section V provides a performance evaluation of this system compared to a naïve approach. The paper concludes in Section VI.

II. RELATED WORK

Many reputation based systems have been proposed to discourage malicious behaviour and identify faulty behaviour in MANETs [2-4]. The majority of papers in this area (e.g. [6]) employ complex algorithms for computing global reputation values based on one or more of identity, past behaviour, time, and their ability to correctly respond to a query or other parameters. Trust in MANETs is usually used either to identify misbehaving nodes, or identify and reduce their impact on the system at large. Systems such as [1] use trust information for secure routing. By identifying trusted nodes, forwarding can avoid potential bad actors creating a more secure communication system. In other work such as [7], trust can be used simply to identify faults of any kind and it is up to a separate system to identify whether the fault is malicious (a security problem) or not (a systems problem). In this paper we also do not consider the trust mechanism, only the trust model and how trust is distributed through the network. In [7], the fault detection system uses gossip as the transport mechanism and was shown to be both adaptable and dynamic to current network conditions. In our work, we use a similar transport method, but by prioritising trust updates we make the best use of the resources available in a low bandwidth environment.

While our work has focused on using gossip for transporting trust values in MANETs, similar work has been done for transporting trust in P2P networks [8]. In that case, trust is very specifically related to the successful rate of P2P interactions, and the gossip protocol for trust distribution is not described in detail. In our work, the naïve gossip protocol has also been simulated in order to compare it with the proposed

This work was supported by Defence R&D Canada (DRDC).

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

prioritised mechanism in MANETs as well as just static ad hoc networks.

In terms of the gossip protocol itself, there has been previous work on altering the relay algorithm to improve system performance. First there has been work in adjusting the probability of retransmitting gossip to improve overhead. By adjusting the gossip probability based on local topology information, the routing overhead could be reduced compared to persistent gossiping or pure flooding while providing increased throughput and lower delay [9]. In our previous work [5] we looked at prioritising gossip in a location service. By enforcing a QoS scheme for relayed location information, the global location convergence time was reduced. A similar approach was found to reduce reputation convergence time.

III. TRUST MODEL

Trust systems work by defining the correct operation of a node. A node may deviate from this definition for one of three reasons. First it may be a *faulty* node, meaning that transient or permanent failures in the medium or the device have caused the detected misbehaviour. A node might equally well be *selfish* in attempting to preserve its own resources by not fully following the current protocol for correct operation. Finally, it could also be a *malicious* node which is acting to deny resources to others. The main goal of a trust system is to identify and then reduce the impact of such nodes.

In order to achieve this, the trust system must collect, analyse, and disseminate trust metrics throughout the network. In a wireless network, each node may passively observe the behaviours of its neighbours and thus assign a local trust level for each without direct interaction. By listening on the shared medium it can increase or decrease its trust based on how well the neighbour follows pre-agreed protocols. In this paper, we define local trust as node A's expectation that it will have a positive outcome when it interacts with a node B that has been directly observed to provide this service in the past. Due to measurement uncertainty in a distributed network environment, the directly measured trust levels are not perfect and may contain a small amount of error.

When first-hand experience or observation does not exist, second-hand information from other nodes must be used. This can take the form of a historical record of third party's observations of the node in question, or simply the current trust of the third parties for a particular node. Reputation is defined here as the globally distributed perception of trustworthiness as calculated from reports from all nodes that have observed the past actions of the node in question.

To achieve this, nodes periodically send out "trust update" reports that are forwarded to all nodes in the network. This update includes the currently measured local trust in all of the node's neighbours. We have used a model similar to [10], where every node aggregates trust level updates directly and constructs a table which represent its level of trust in every node in the network. The aggregated trust value is a node's reputation as observed at the local node and is stored in a trust table. Thus the reputation of a node *i* as viewed from node *j* is a composite of all the updates received at node *j* about node *i* from all of *i*'s neighbours (past and present) in the network. The collection and analysis of trust values is beyond the scope of this paper.

The calculation of reputation is as follows. Let $T_{ij} \in [0,1]$ denote the trust of node *i* in node *j* which is directly measured and assigned by node *i* based on observed behaviour of node *j*, where 0 means the node is completely untrusted and 1 means the node is completely trusted. Also, let $\tilde{T}_{ij}(t) \in [0,1]$ denote an entry in a trust table at node *i*, which represents trust (reputation) of node *i* to node *j* at time $t \in \{0,1,\dots\}$.

Since trust updates may be received from nodes that are not themselves trusted, it is important to give more weight to updates that are created by trusted nodes than the updates that are created by untrusted nodes. Given a network with *N* nodes, for all nodes $i, j, k \in \{1,2,\dots,N\}$, the entry in the trust table at node *i* is aggregated with a trust level update of node *j* in node *k* (i.e., T_{kj}) as follows.

$$P \leftarrow (1-\alpha) * \tilde{T}_{ij}(t) + \alpha * T_{kj} \quad (1)$$

$$\tilde{T}_{ij}(t+1) \leftarrow (1-\beta_{ik}(t)) * \tilde{T}_{ij}(t) + \beta_{ik}(t) * P \quad (2)$$

where $\alpha \in (0,1)$ and

$$\beta_{ik}(t) = \frac{\tilde{T}_{ik}(t)}{1+\tilde{T}_{ik}(t)} \quad (3)$$

Thus reputation is based on the past trust of node *i* in node *j* weighted by the historical metric α while taking into account new information of node *k* towards node *j*.

IV. EPIDEMIC TRUST DISTRIBUTION

In this section, we describe a gossip-based priority-aware distribution scheme for trust update services. Epidemic routing, also known as the gossip protocol, is used in our distribution service to reliably and efficiently transport the trust information. In the gossip protocol, wireless nodes locally broadcast packets so that all nodes within range can interpret the packet. The receivers of the packet then re-forward the packet based on a relay algorithm. Epidemic routing has traditionally been used in a number of areas including in data dissemination (as a type of flooding), database replication (for updating and repairing versioned data), and data aggregation (for compiling and summarising data from an area of the network) [9].

In our distribution service, each node maintains a priority queue of the most recent gossip from each node. Each gossip packet contains the ID of the node, a timestamp, and a list containing the local trust of each neighbour of the sending node. The gossip in the queue is sorted based on a "freshness," which is influenced by multiple factors (described in Case 2 below). Freshness relates to the priority of the packet similar to class of service in network QoS. When an opportunity to send gossip arrives (for example a transmission window in a TDMA MAC), the freshest gossip is sent. In order to avoid starvation a gossip timer can also be maintained so that if no gossip has been sent after a certain period of time, the service will make an explicit request for a MAC transmission window.

For any given network with N nodes, three indexed variables ($Frs.Val_i, Frs.TS_i, Gsp.TS_i$), which correspond to a

node $i \in \{1,2,\dots,N\}$, are maintained to facilitate the prioritisation process. The variables $Frs.Val_i$ and $Frs.TS_i$ are respectively used to track the current value and modified time of the freshness for gossip created by node *i*. The gossip timestamp $Gsp.TS_i$ is used to track the time at which the most recent gossip was created by node *i*. Freshness operates in a range of values from 1 to the maximum freshness window size denoted as *FW*. The operation of the proposed scheme, when a gossip packet is received and transmitted, is described as follows.

A. Receive Gossip

When a gossip packet is received at node *m*, the ID of the node is extracted from the gossip (say node *i*) and its timestamp $Pkt.TS_i$, which shows the time that the gossip was created, is compared to $Gsp.TS_i$.The freshness and timestamps are updated as follows.

CASE 1: New Gossip. If there is no local entry for the sending node, a new entry is created and added to the priority queue. To ensure a new entry is sent out as soon as possible, its freshness is set to maximum value, which is *FW*, and its timestamps are updated as in (4)-(6).

$$Frs.TS_i \leftarrow current\ time \tag{4}$$

$$Gsp.TS_i \leftarrow Pkt.TS_i \tag{5}$$

$$Frs.Val_i \leftarrow FW \tag{6}$$

CASE 2: Newer Gossip. If there is local entry corresponding to the node and the received gossip is newer than the one stored in the queue (i.e. $Gsp.TS_i < Pkt.TS_i$), then the newer gossip replaces the gossip in the queue. In this case, the freshness is calculated based on three factors: first, whether it includes a new neighbour update (neighbour that was not previously a neighbour), second how extreme the trust values are, and third how much the trust values have changed. Let $\boldsymbol{V}_i$ be the set of neighbour nodes of node *i*, which originally created the trust, and define $\delta = 1$ if T_{ij} is new; otherwise, $\delta = 0$. The freshness and timestamps are updated as in (7)-(9). Note that the value of 1 is added in (9) to guarantee a minimal increase in freshness for new gossip. The variable $Frs.Val_i$ cannot exceed *FW*.

$$Frs.TS_i \leftarrow current\ time \tag{7}$$

$$Gsp.TS_i \leftarrow Pkt.TS_i \tag{8}$$

$$Frs.Val_i \leftarrow Frs.Val_i + 1 + \delta + \max_{j \in \boldsymbol{V}_i}\left\{\left(0.5 - T_{ij}\right)^2\right\} + \max_{j \in \boldsymbol{V}_i}\left\{\left(\tilde{T}_{mj}(t) - T_{ij}\right)^2\right\} \tag{9}$$

CASE 3: Duplicate Gossip. When there is a local entry correspond to the node but the received gossip timestamp is the same as the one in queue (i.e., $Gsp.TS_i = Pkt.TS_i$), this implies that at least one of the neighbouring nodes has matching information and so there is no immediate need to rebroadcast the gossip. Consequently, the received gossip is ignored, its freshness value is decremented by one and its timestamp is set to the current time as in (10) and (11).

$$Frs.TS_i \leftarrow current\ time \tag{10}$$

$$Frs.Val_i \leftarrow \max(1, Frs.Val_i - 1) \tag{11}$$

CASE 4: Older Gossip. When there is local entry corresponding to the node but the received gossip is older than the gossip in the queue (i.e., $Gsp.TS_i > Pkt.TS_i$), this means that a neighbour has out of date information and so it should be updated with the newer trust information. In this case, the older gossip is ignored. However, its freshness is incremented by one and its timestamp is set to current time as in (12) and (13).

$$Frs.TS_i \leftarrow current\ time \tag{12}$$

$$Frs.Val_i \leftarrow \min(FW, Frs.Val_i + 1) \tag{13}$$

B. Transmit Gossip

When a free slot to transmit a gossip packet is available, the gossip with the highest freshness value ($Frs.Val_i$) in the queue is selected. In the case where there are two or more gossips with the same $Frs.Val_i$, the gossip with the lowest $Frs.TS_i$ is selected. Note that since only one gossip packet can be received and processed at a time, there will never be two gossips with the same $Frs.TS_i$. A copy of selected gossip from the priority queue is transmitted and its freshness is reset to one and its timestamp is set to the current time as in (14) and (15).

$$Frs.TS_i \leftarrow current\ time \tag{14}$$

$$Frs.Val_i \leftarrow 1 \tag{15}$$

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed distribution service by considering network trust estimation error and normalized network trust estimation error. The network trust estimation error measures the overall amount of errors due to inaccurate or missing trust level information stored across the network. The normalized network trust estimation error is similar to the trust estimation error but puts less weight on errors involving nodes that are far apart and more weight on errors involving nodes that are closer.

Let B_i and $\{x_i(t), y_i(t)\}$ respectively denote the actual behaviour and location of node $i \in \{1,2,\dots N\}$ at time t. The relative distance ($D_{ij}(t)$) between nodes *i* and node *j* and absolute error in trust of node *i* to node *j* aggregated at node *i* ($\xi_{ij}(t)$) can be found as in (16) and (17).

$$D_{ij}(t) = \sqrt{\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2} \tag{16}$$

$$\xi_{ij}(t) = \left|\tilde{T}_{ij}(t) - B_j\right| \tag{17}$$

For a given network with *N* nodes, network trust estimation error ($(E_a(t))$ and normalized network trust estimation error ($E_n(t)$), after exchanging trust information for a period of time *t*, are given as in (18) and (19). Note that the normalizing factor decreases linearly with the distance between nodes.

$$E_a(t) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \xi_{ij}(t) \tag{18}$$

$$E_n(t) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \left(\frac{\xi_{ij}(t)}{1 + D_{ij}(t)}\right) \tag{19}$$

A. Simulation Setup

A customised MATLAB-based simulation was used to evaluate the performance of the proposed distribution algorithm. A group of fixed and mobile nodes with different initial trust levels is considered in a square area organized in 25 blocks. Transmission was simplified so that nodes in one block can transmit to nodes within two blocks. The MAC protocol is assumed to be TDMA-like with each node assigned consecutive transmission rounds. In each round, a node is allowed to send a single trust frame, either one of its own or one that it has heard transmitted from another node (epidemic routing).

The trust distribution system has a freshness window of three to allow for some prioritisation without starvation of low priority flows. Our previous work in [5] found that a larger window provided minimal improvements.

Each pair of nodes is assigned with an initial reputation value towards all other nodes in the network that is initially set to 0.5. In this study, we consider the network scenarios with equal number of trusted ($B_i = 0.8$) and untrusted ($B_i = 0.2$) nodes assigned at random. Trust updates from other nodes are weighted with α = 0.8 from equations (1)-(3). All nodes pass correct trust values. The simulation time (t) and network node density (ρ), i.e. number of nodes per block, are variables. Note that simulated data points are obtained as an average of 200 simulation runs repeated with different random initial node placement within the 25 blocks. Simulation time for each run is up to 1000 slots. We consider a group mobility model where mobile node may move to an adjacent block randomly every 80 time slots. The observed behaviour (T_{ij}) of neighbouring nodes is collected after every 80 slots with a uniformly distributed measurement error, that is $T_{ij} = B_i \pm 0.1$. If not stated otherwise, the parameters for simulated scenarios are given in Table I.

Table I. Simulation Parameters

Parameters	**Range**
Transmission Range, R	2 blocks
Number of allocated slots per node per round	1
Freshness Window, FW	3
Initial trust between each node pair, $\{\tilde{T}_{ij}(0)\}$	0.5
Actual behaviour (Trusted, Untrusted), B_i	(0.8, 0.2)
Parameter α (historical weighting)	0.8

To appreciate the performance gained by the proposed scheme, we also present the performance of a naïve gossip approach which does not include prioritisation. This scheme is similar to the proposed algorithm with FW = 1, thus rebroadcasting received gossips without regard to anything except initial arrival time (FIFO).

B. Static Network Results

Our analysis found that for both sparse and dense static networks, the proposed prioritisation of gossip provided a significant improvement in the trust estimation error over the naïve approach. The results for the static network are shown in Figure 1. The impact of node density on estimation error with our proposed scheme is further explored in Figure 2.

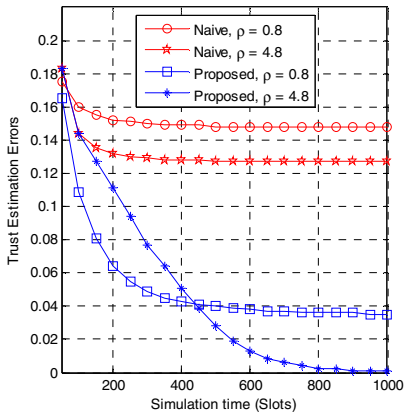

Figure 1. Static Network: Trust Estimation Error vs. Simulation Time

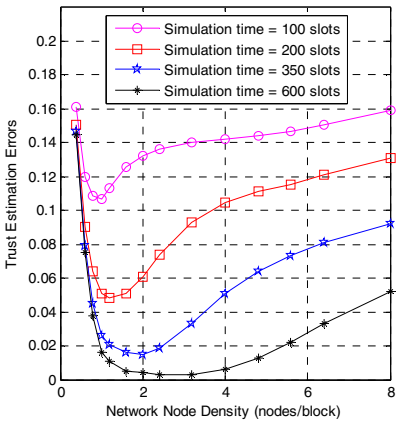

Figure 2. Static Network: Trust Estimation Error vs. Network Node Density

For both the naïve and proposed algorithm, a denser network improved estimation error because more nodes were able to hear each other and thus make use of the broadcast trust values. However, the lack of mobility meant that the trust values did not converge to zero within 1000 time slots except in the dense case for the proposed algorithm. For the sparse network, some cases had an isolated node unable to communicate with others, while in the naïve dense case the convergence towards no error is extremely slow as the gossip must wait a long time to travel across the network FIFO. Note that for the proposed algorithm, the sparse network converges faster in less than 450 time slots. Since there are fewer nodes and less traffic to be sent the error reduces quickly, but does not converge completely due to isolated nodes.

Figure 2 shows the estimation error is very sensitive to the node density. For sparse networks with less than one node per block, network partitioning limits the error estimation. However, once a network is fully connected, the optimal node density for producing low trust estimation error depends on the amount of time that has elapsed before measuring the error. For relatively low error in 100 time slots, a network density of approximately one is optimal. However, if more time is available, a higher node density improves the trust estimation. The subsequent increases in estimation error come due to the increased time for all nodes to relay their trust tables.

It is interesting to note the implication of this finding to sensor networks. Using epidemic routing and a fixed time to distribute information to all nodes, a target node density can be chosen to minimise the information error.

C. *Mobile Network Results*

The impact of mobility on the estimation error in a sparse network with 1.6 nodes per block is shown in Figure 3.

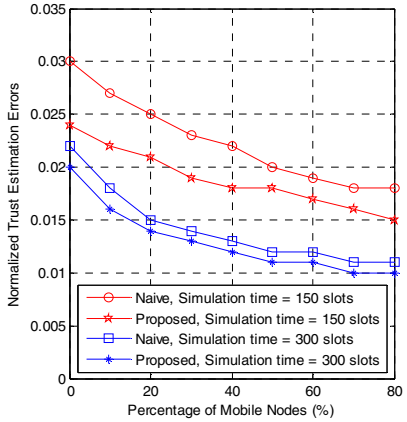

Figure 3. Mobile Network: Normalized Trust Estimation Error vs. % of Mobile Nodes for the Sparse Network

As the number of mobile nodes increases, the normalised estimation error decreases for both the naïve and proposed distribution service. In this sparse network, the proposed scheme works significantly better than the naïve scheme in the short term (150 time slots) as opposed to the long term (300 time slots) when there is more of a chance to fully distribute trust FIFO. Because of measurement error from mobility, the trust error does not reach zero, but an increase in mobility means that information is shared to more nodes decreasing the chance of an isolated node. This decrease in error with increased mobility agrees with current research on information theory which states that mobility increases the transmission capacity of information in networks [11].

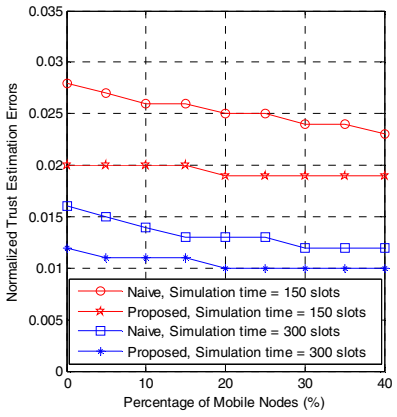

Figure 4. Mobile Network: Normalized Trust Estimation Error vs. % of Mobile Nodes for the Dense Network

The impact of mobility on the estimation error in a denser mobile network with 3.2 nodes per block is shown in Figure 4. As in the previous case, as the number of mobile nodes increases, the error estimation decreases, however this occurs at a much slower rate. Since there is a smaller chance of isolated nodes, the increased transmission capacity is limited. The denser network does however show a greater difference between the naïve and proposed approach. As mentioned previously, this is due to the more efficient use of transmission slots so that new information is spread more quickly in the prioritised scheme.

VI. Conclusions and Future Work

In this work we have described a trust distribution service that uses local observations to quickly and accurately generate global reputation in MANETs. We use epidemic routing to distribute local trust measurements. Simulations show that by prioritising critical trust updates the global convergence time is reduced compared to the traditional gossip protocol. Our simulation also confirmed that node density is a critical component in epidemic based data distribution, with different node densities being optimal depending on the amount of time available.

For future work we are currently planning on investigating the trade-off between resource usage (gossip frequency) and decision making accuracy vs. timeliness. We would also like to investigate the impact of transmission errors on robustness, and how quickly changes in local trust values can be communicated in a network once stable reputations are established.

References

[1] M. Chang, I.R. Chen, F. Bao, and J.H.Cho, "Trust Threshold Based Routing in Delay Tolerant Networks," in *IFIP International Conference on Trust Management*, 2011, pp. 265-276.

[2] M.A. Azer, S.M. El-Kassas, A.W.F. Hassan, and M.S. El Soudani, "A Survey on Trust and Reputation Schemes in Ad-Hoc Networks," in *IEEE Conference on Availability, Reliability and Security (ARES),* 2008, pp. 881-886.

[3] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A Survey of Trust and Reputation Management Systems in Wireless Communications," *Proceedings of the IEEE*, Vol. 98, No. 10, Oct. 2010, pp. 1755-1772.

[4] J.-H. Cho, A. Swami, and I.R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Communications Surveys and Tutorials*, Vol. 13, No. 4, 2011, pp. 562-583.

[5] D. Kidston and H. Rutegemwa, "A Location Service for VHF Tactical Networks", in *IEEE Military Communications Conference*, 2011, pp. 786-791.

[6] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralised Trust Management," in *IEEE Symposium on Security and Privacy*, 1996, pp. 164-173.

[7] M. Elhadef and A. Boukerche, "A Failure Detection Service for Large-Scale Dependable Wirless Ad-Hoc and Sensor Networks," in *IEEE International Conference on Availability, Reliability and Security (ARES)*, 2007, pp. 1-8.

[8] I. Zubair and M.H. Islam, "Adaptive Trust Management in P2P Networks using the Gossip Protocol," in *IEEE-ICET Conference on Emerging Technologies*, 2008, pp. 176-181.

[9] Z-Y Liu, M. Kwiatkowska, and K. Lei, "An Adaptive Epidemic Broadcast Mechanism For Mobile Ad Hoc Networks," in *International Conference on Machine Learning and Cybernetics*, Vol. 7, 2008, pp. 3651-3656.

[10] A. A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *27th Australasian Computer Science Conference (ACSC2004),* Vol. 26, No. 1. Dunedin, New Zealand, 2004, pp. 47–54.

[11] M. Grossglauser and D.N.C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 4, 2002, pp. 477-486.