A Simple Trust model for On-Demand Routing in Mobile Ad-hoc Networks

Nathan Griffiths*, Arshad Jhumka, Anthony Dawson, and Richard Myers

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK
{nathan, arshad}@dcs.warwick.ac.uk

Summary. In a mobile ad-hoc network, nodes cannot rely on any fixed infrastructure for routing purposes. Rather, they have to cooperate to achieve this objective. However, the absence of any trusted third party in such networks may result in nodes deviating from the routing protocol for selfish or malicious reasons. The concept of trusted routing has been promoted to handle the problems selfish and malicious nodes cause to the network. In this paper, we focus on using trust in routing, and show how trust can mitigate against malicious behaviour.

1 Introduction

A mobile ad-hoc network (MANET) is a wireless network with no fixed infrastructure and no central administration. Nodes in the network usually have limited resources for computation, bandwidth, memory, and energy. Because nodes are mobile, the topology of the network varies. Message routing in MANETs is a significant problem. The lack of central administration means that nodes cannot be forced to cooperate for message routing. Nodes may deviate from the protocol for selfish or malicious reasons. For example a selfish user may wish to preserve energy resources, while a malicious user might attempt a denial of service attack. Routing protocols must cope with such selfish and malicious behaviours.

Recently, a new class of routing protocol has been proposed, namely trusted routing. Trusted routing protocols consist of two parts: a routing part and a trust model. Routing decisions are made according to the trust model. Trust and reputation have been used in many settings to cope with uncertainty in interactions. Trust is used to assess the risk associated with cooperating with others; it is an estimate of how likely another is to fulfil its commitments [2, 5]. Trust can be derived from direct interactions and from reputation.

Our work is inspired by Pirzada and McDonald's (hereafter referred to as P&M) trusted routing model [7, 8]. Based on Marsh's [5] work on computational trust, P&M use trust for routing in ad-hoc networks and obtain promising simulation results. Their approach (described below) is sophisticated and combines a range of

* Contact author

situational trust assessments into an overall trust assessment for making decisions. Our view is that although such sophistication offers rich information on which to base decisions, similar levels of resistance to malicious behaviour can be achieved with a simpler approach. Although we accept P&M's results we also find some limitations. For example, they consider a range of mechanisms for malicious behaviour, and their results do not discern the effect of trust against specific types of behaviour. Aspects of P&M's results are counter-intuitive, e.g., network latency decreases as the number of malicious nodes is increased.

2 Background

In this section, we briefly introduce key work that relates to our approach. We begin by introducing the Ad-hoc On-demand Distance Vector (AODV) routing protocol, and then discuss selected trust models and how trust relates to routing.

2.1 Routing protocols

There are two major classes of routing protocols for MANETs: proactive and reactive protocols. In proactive protocols nodes devote resources to tracking routes in a routing table, whereas in reactive protocols, routes are discovered when needed to preserve nodes' resources. In this paper, we focus on the AODV reactive protocol as it is an efficient low-overhead approach. There also exist hybrid protocols, that combine features of proactive and reactive protocols, but these are beyond the scope of this paper.

In AODV [6], when a source node wants to communicate with a destination node, but does not have a route to the destination, it initiates a route discovery. The source node broadcasts a `RREQ` (route request message) to all of its neighbours. Each neighbour that receives the `RREQ` will check in its own routing table to see if it has a route to the specified destination. If not, it will set up a reverse path towards the sender of the `RREQ` and then re-broadcast the `RREQ`. Any node receiving the `RREQ` will generate a `RREP` (route reply message) if it either has a fresh enough route to the destination, or is itself the destination. This `RREP` is then unicast to the next hop towards the originator of the `RREQ`. When a node receives a `RREP`, it updates the appropriate fields in its routing table and in the `RREP`, and then forwards the `RREP` to the next hop until it reaches the original sender. A sender node can have multiple routes to the destination. However, the chosen route is the shortest one between the sender and destination. This relies on the underlying assumption that all nodes are trustworthy and will never deviate from the protocol. In this paper we do not make this assumption, and use trust to mitigate against malicious or faulty behaviour.

2.2 Dependable routing

The majority of routing mechanisms for MANETs rely on the assumption that nodes will never deviate, but in a real-world MANET this assumption is unrealistic. Be-

cause resources in a MANET are scarce, nodes may act selfishly such as not forwarding a message. In the worst case, nodes may act in an arbitrary fashion, i.e., display Byzantine behaviour [1]. Hence, to handle these problems, techniques such as secure routing [11] and trusted routing [7] have been proposed. In secure routing, cryptographic primitives are used to ensure properties such as confidentiality, integrity etc. However, secure routing requires a centralised trusted third party, making it impractical for MANETs. Trusted routing, on the other hand, can be used to handle both selfish and Byzantine nodes. In trusted routing, a trust model is embedded within the routing algorithm, and routing decisions are taken based not on shortest path but on trust values. Thus, in trusted routing the path with the highest trust is chosen.

2.3 Trust models

Numerous models of trust and reputation exist to support cooperation in computational environments [4, 9]. One of the earliest approaches is Marsh's formalism [5]. Marsh uses the outcomes of direct interactions among entities to calculate situational and general trust. Situational trust is the level of trust in another for a specific type of situation, while general trust refers to overall trustworthiness irrespective of the situation. After each interaction an entity considers whether the other entity fulfilled its obligations. If so, then trust increases, but trust decreases if commitments are broken. To minimise the risk of failure entities will interact with the most trusted of the potential interaction partners.

Marsh's formalism is the base of many subsequent models, which supplement trust based on direct interactions with other information sources to inform decision making. For example, sophisticated approaches such ReGreT [10] and FIRE [3] add reputation information provided by third parties and knowledge of social structures to arrive at overall trust assessments. However, whilst powerful, such sophisticated models are not appropriate for routing in MANETs where resources are scarce and knowledge of social relationships between nodes is unlikely to be available.

Several trust models have been developed for peer-to-peer systems [12, 13, 14], based on sharing recommendation information to establish reputation. Although in principle these could be applied to routing in MANETs, there are two important problems. First, there is significant network overhead due to the additional information exchanged. Second, addressing the potential for malicious recommendations requires a trusted third party (or a computationally expensive public-key infrastructure), which goes against the nature of MANETs.

There are few trust mechanisms for ad-hoc networks. Zhou and Haas [15] describe a cryptographic scheme to ensure node integrity. However, their approach requires complex pre-configuration of servers to provide a distributed certification authority and relies on cryptographic operations which are costly in computation and power. P&M propose arguably the most appropriate mechanism, where nodes calculate situational trust according to observed events and then use an aggregated general trust for routing decisions. Nodes record information about others for various event types: acknowledgements, packet precision, gratuitous route replies, blacklists,

HELLO packets, destination unreachable messages and authentication objects. For each type, the proportion of positive events is taken to correspond to the situational trust. Situational trust values are then aggregated using a weighted product to give overall trust. When routing, nodes will forward packets to maximise trust (rather than minimising cost in standard AODV). P&M have obtained promising simulation results, but we argue that similar positive effects can be obtained with a greatly simplified trust model.

3 The Proposed Model: Simple Trusted AODV

3.1 Network model

The setting for our approach is a simple MANET in which we assume that nodes are situated in a bounded 2-dimensional space, within which they are free to move. For simplicity we assume they move randomly around the space. Each node has individual characteristics that define its speed of movement and the range over which it can transmit messages. The positions and transmission ranges define the network neighbourhood, since nodes can only transmit to others within their transmission range, and can only receive messages from others when they are within their range. Thus, if two nodes are within each others' transmission range they are free to communicate, but otherwise intermediate nodes are needed to forward packets. We assume that nodes use AODV as described above, and we describe below our approach for incorporating trust into AODV.

3.2 Attack model

The standard AODV protocol assumes that nodes are fully functional and benevolent, and does not cope well if this is not the case. This has led to the development of trusted routing protocols such as that proposed by P&M. In developing their protocol, P&M describe several possible attacks, and their simulations allow malicious nodes to use any of these. Consequently, it is impossible to evaluate their trust model against *specific* attack types. In this paper, therefore, we concentrate on a small number of specific attacks and test our model against each type individually.

We consider two varieties of blackhole and a greyhole attack. A blackhole is a malicious node that attempts to drop all packets, typically by forging route replies to create fake routes with it as an intermediate node. This allows the blackhole to divert and intercept traffic from across the network, and subsequently drop all packets that it receives. A greyhole can be viewed as a faulty node, rather than explicitly malicious. Greyholes do not falsify route replies, but instead will periodically drop packets. This might be due to a fault or due to malicious intentions. Regardless of the reason, greyholes appear as intermittently faulty nodes to the rest of the network. There are several possible mechanisms to implement these attacks within AODV, and we use the following definitions.

Blackhole on route (`Blackhole-OnRoute`)

This is our simplest blackhole definition, and operates by replying that it has a fresh enough route to the destination whenever it receives a `RREQ`, regardless of whether it actually knows a route. AODV uses sequence numbers to track the freshness of routes. When nodes issue a new `RREQ` or the destination responds the sequence number is increased. A `Blackhole-OnRoute` node claims to have an existing fresh route to the destination and so the generated `RREP` has the same sequence number as the `RREQ`, causing it to be accepted by the original sender, which subsequently creates a route with the blackhole as an intermediate node. This kind of a blackhole is partially guarded against within AODV, since if the original `RREQ` eventually reaches the intended destination a `RREP` will be generated. The reply from the destination itself has an increased sequence number over the `RREQ` and so will overwrite the malicious route setup by the blackhole. Despite this, in our simulations `Blackhole-OnRoute` was able to cause significant packet loss, as the routes it created intercept the first packets sent across any new route until the destination's `RREP` was received.

Blackhole fake destination reply (`Blackhole-FakeDestReply`)

This blackhole is more malicious than `Blackhole-OnRoute`, since in addition to claiming to have a recent enough route to the destination it also increases the sequence number in the `RREP` and so appears to offer a new route. The effect is that `Blackhole-FakeDestReply`'s route is not overwritten by any reply subsequently returning from the destination itself. Thus, a route to the actual destination will only be established when the destination's `RREP` is received before that generated by the `Blackhole-FakeDestReply` node.

Greyhole (`Greyhole`)

The `Greyhole` does not falsify route replies in order to intercept packets, but instead simulates a node having intermittent faults. We characterise a `Greyhole` using two time periods:

- `MAX_TIME_TO_BURST_FAULT`: maximum time to the next burst fault (seconds)
- `MAX_TIME_BURST_FAULT_LASTS`: maximum burst fault duration (seconds)

Using these time periods a node will start a burst fault at a random time between 0 and `MAX_TIME_TO_BURST_FAULT`. The burst fault lasts for a random period between 0 and `MAX_TIME_BURST_FAULT_LASTS`. These parameters can be modified to alter the nature of the faults.

3.3 Trust model — Simple Trusted AODV (ST-AODV)

There are many potential mechanisms for determining whether a node can be trusted, based on observing the nodes' activities and behaviours. The influence of these observations can be combined to determine a trust level. P&M use several aspects

of node behaviour including acknowledgements, packet precision, gratuitous route replies etc., as described in Section 2. Our view is that the effect of malicious nodes can be significantly reduced using a much simpler scheme. We build our trust models using acknowledgements as the single observable factor for assessing trust. We believe that acknowledgements offer an effective indication of a node's trustworthiness.

An acknowledgement is a means of ensuring that packets which have been sent for forwarding have actually been forwarded. There are a number of ways that this is possible, but *passive acknowledgement* is the simplest. Passive acknowledgement uses promiscuous mode to monitor the channel, which allows a node to detect any transmitted packets, irrelevant of the actual destination that they are intended for. Using this method a node can ensure that packets it has sent to a neighbouring node for forwarding are indeed forwarded.

To record trust information about a node, we introduce a `TrustNode` data store, which comprises a `nodeID`, a `packetBuffer`, and an integer `trustValue` for the node. Each node maintains a `TrustNode` for each of the nodes that it has sent packets to for forwarding. To detect whether a packet is successfully forwarded, the packets that have been recently sent for forwarding are stored in the `packetBuffer`. This is a circular buffer, meaning that if packets are not removed frequently enough the buffer will cycle, erasing the oldest elements. Thus, if a node is dropping packets or is being unacceptably slow at forwarding packets then the buffer will cycle. Otherwise, if the node is performing acceptably then when the promiscuous mode detects a forwarded packet, it can be found and removed from the buffer.

In ST-AODV we use a simple trust model, where the `trustValue` for each node is initialised to 0. With each observation, the value is incremented for nodes that are detected to forward packets and decremented for nodes that do not appear to forward packets. To check whether a node is sufficiently trusted we introduce a `minTrust` threshold such that nodes with `trustValue` <= `minTrust` are considered untrusted. If a node is untrusted then it is not sent packets for forwarding, and any replies it gives to route requests are ignored. Once a node becomes untrusted it is barred from consideration for packet forwarding by dropping it from the set of neighbours, removing all routes that use it, and sending out a new `RREQ` to re-establish the removed routes. Similarly, when receiving a `RREP` the first hop node is checked and if it is untrusted then the reply is disregarded. Thus, only routes where the first hop is trusted are established. Nodes make routing choices based on trust as well as the number of hops, such that the selected next hop gives the shortest trusted path.

4 Simulation and Results

To evaluate the effectiveness of ST-AODV we have performed simulations using the ns-2 network simulator[2]. Nodes are situated in a bounded 2-dimensional world about which they wander randomly. We use a network of 50 nodes in the simulations discussed below. The network contains benevolent nodes that use ST-AODV to make

[2] http://www.isi.edu/nsnam/ns/

routing decisions, and malicious nodes that use one of the attacks defined in Section 3. The `minTrust` threshold used for barring nodes is set at -10. We obtain the following metrics from our results (which are averaged over a number of runs):

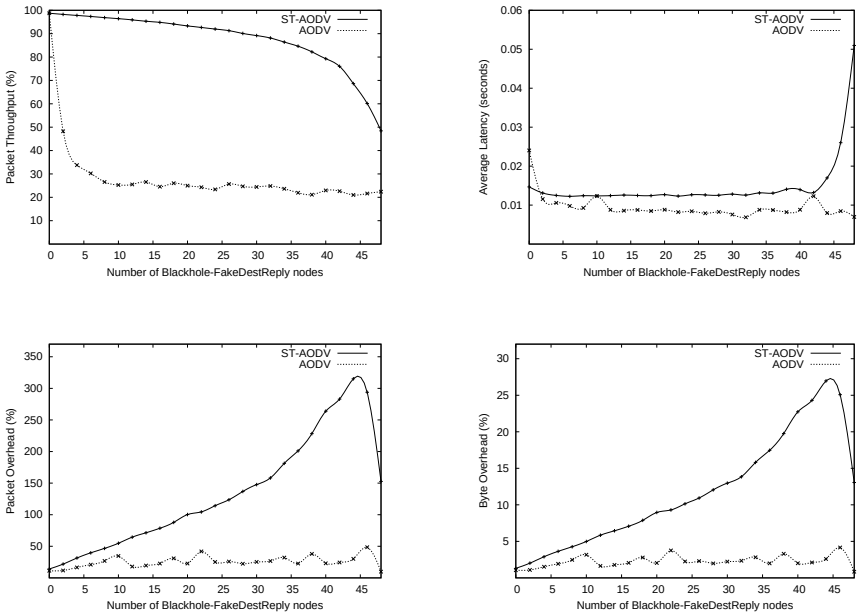

Fig. 1. Results for the `Blackhole-FakeDestReply` attack.

- **Packet throughout**: ratio of packets received by the destination to the number of packets sent (%)
- **Average latency**: average time for packets to reach their destination (seconds)
- **Packet overhead**: ratio of control packets generated to the total number of data packets sent (%)
- **Byte overhead**: ratio of control bytes generated to the total number of data bytes sent (%)

We record these metrics using both standard AODV and ST-AODV for each attack type under various proportions of malicious nodes. Figures 1, 2 and 3 show the results for `Blackhole-FakeDestReply`, `Blackhole-OnRoute` and `Greyhole` attacks respectively. The results show that ST-AODV significantly improves packet throughput under all attack types. As the number of malicious nodes is increased each attack type reduces throughput, but ST-AODV mitigates against this.

In standard AODV a small number of blackhole nodes dramatically reduces throughput, the effect stabilises for moderate numbers, and for `Blackhole-OnRoute` falls off for high numbers (`Blackhole-FakeDestReply` does not fall off further

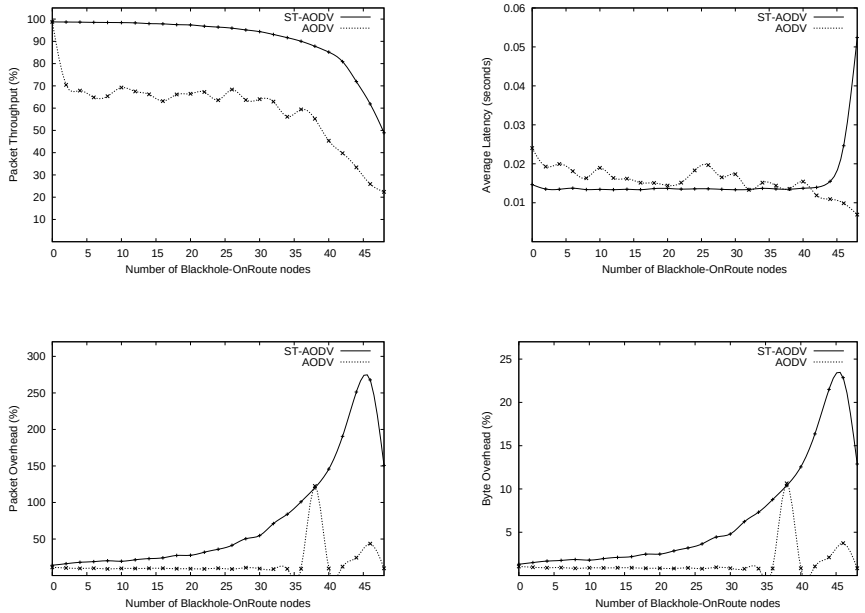

Fig. 2. Results for the `Blackhole-OnRoute` attack.

since throughput has already fallen significantly). The `Greyhole` attack results in a fairly linear throughput reduction as the number of malicious nodes increases. As predicted, `Blackhole-FakeDestReply` has the most effect. For AODV, increasing the number of `Blackhole-FakeDestReply` nodes very soon reduces throughput to around 25% with 10 malicious nodes, while a similar number of `Blackhole-OnRoute` nodes gives around 65% throughput. Regardless of attack type, ST-AODV achieves a good and fairly consistent throughput. For both blackhole attacks a throughput of over 90% is maintained if less than half the nodes are malicious. With standard AODV just 2 malicious nodes reduces throughput to below 70%. Under a `Greyhole` attack the throughput using ST-AODV reduces linearly with the number of malicious nodes (as for AODV), but the rate of reduction is reduced meaning trust is more beneficial with higher numbers of malicious nodes.

For blackhole attacks there is relatively little effect on latency using ST-AODV. Performance is slightly improved for `Blackhole-OnRoute` attacks (by < 0.005 seconds) while it is slightly worse for `Blackhole-FakeDestReply` (again by < 0.005 seconds). Under `Greyhole` attacks latency is reduced by approximately 0.01 seconds using ST-AODV, regardless of the number of malicious nodes. As expected, the packet overhead and the byte overhead are increased by using ST-AODV under all attack types. As the number of malicious nodes is increased the overhead also increases, and more significantly so with higher numbers of malicious nodes. For the `Greyhole` attack the packet overhead is increased by approximately 5% where under

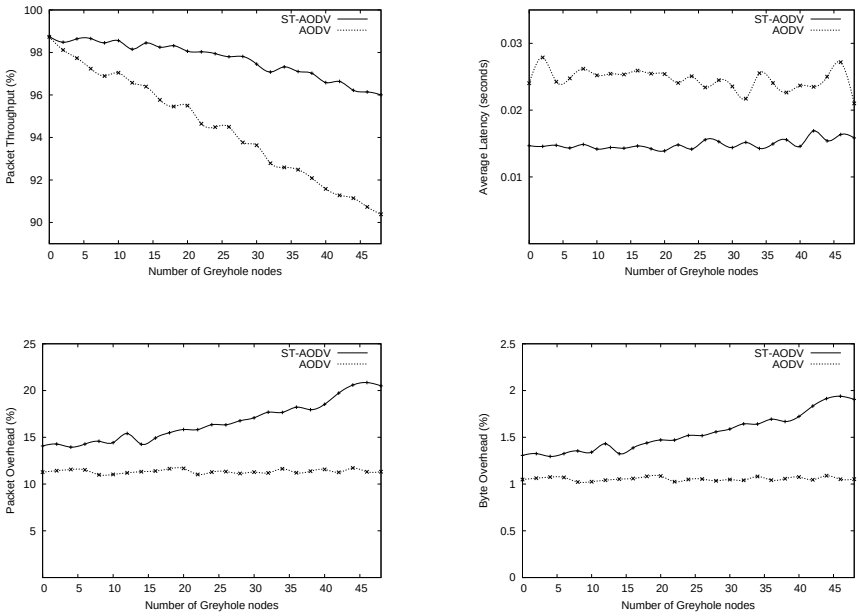

Fig. 3. Results for the `Greyhole` attack.

half the nodes are malicious, rising to around 10% with more malicious nodes. In the `Blackhole-OnRoute` attack the overhead is below 25% with below 25 malicious nodes, but this rises rapidly for higher numbers, peaking at over 200% overhead for 45 malicious nodes. The `Blackhole-FakeDestReply` attack causes the overhead to rise more rapidly, to nearly 100% where half the nodes are malicious. This is as expected, since the `Blackhole-FakeDestReply` attack is more malicious.

5 Conclusions and Summary

We have described a simple trust model that extends AODV to cope with malicious nodes. Our simulations show significant improvements in throughput, at the expense of packet and byte overhead. For low proportions of malicious nodes in the population the increase in overhead is relatively small given the improvement in throughput. Our results also show how different attacks affect a network. In particular, using standard AODV a `Blackhole-FakeDestReply` attack significantly reduces throughput compared to `Blackhole-OnRoute` and `Greyhole` attacks. Using ST-AODV we are able to minimise this difference and to protect the network effectively against all three attacks.

The results presented above are preliminary findings and there are many areas of ongoing investigation. Our results compare favourably to those obtained by P&M in

terms of the improvement in throughput. We find a higher packet and byte overhead than P&M and this requires further investigation. However, P&M's results are unintuitive in that the overhead and latency decrease as more malicious nodes are added. These differences require further investigation.

We are considering several extensions to ST-AODV, including a more flexible (non-linear) trust update function and improved monitoring using promiscuous mode to monitor all traffic, rather than only a node's own packet forwarding requests. We are also investigating more flexible sanctions against untrusted nodes, such as temporary blacklisting. Finally we aim to explore how different trust models perform against different attacks and combinations of attack.

References

1. B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the 1st ACM workshop on Wireless security*, pages 21–30, 2002.
2. D. Gambetta. Can we trust trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.
3. T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
4. A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
5. S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.
6. C. Perkins, E. M. Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. IETF RFC 3561, 2003.
7. A. A. Pirzada and McDonald C. Trust establishment in pure ad-hoc networks. *Wireless Personal Communications*, 37(1–2):139–168, 2006.
8. A. A. Pirzada, McDonald C., and A. Datta. Performance comparison of trust-based reactive routing protocols. *IEEE Trans. on Mobile Computing*, 5(6):695–710, 2006.
9. S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 19(1):1–25, 2004.
10. J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the 1st Int. Conf. on Autonomous Agents in Multi-Agent Systems*, pages 475–482, 2002.
11. K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of the 10th IEEE Int. Conf. on Network Protocols*, pages 78–89, 2002.
12. A. A. Selçuk, E. Uzun, and M. R. Pariente. A reputation-based trust management system for P2P networks. In *IEEE/ACM Int. Symposium on Cluster Computing and the Grid*, pages 251–258, 2004.
13. S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 9(6):24–34, 2005.
14. L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust in peer-to-peer communities. *IEEE Trans. on Knowledge and Data Engineering*, 16(7):843–857, 2004.
15. L. Zhou and Z. J. Haas. Securing ad-hoc networks. *IEEE Network Magazine*, 13(6):24–30, 1999.