

The Trusted Route - Mitigating the Effects of Selfish Nodes in Ad-hoc Networks

David Waiting and Neco Ventura

Department of Electrical Engineering, University of Cape Town

Rondebosch, South Africa

{david, neco}@crg.ee.uct.ac.za

Abstract—Node selfishness refers to nodes that use the resources of an ad-hoc wireless network but do not perform their basic network responsibilities of route management and packet forwarding. A security concern in ad-hoc networks is that nodes may become selfish users of the network in order to conserve their battery or processing power. This results in poor network performance and unreliable network paths. In this paper a distributed trust mechanism is proposed to identify selfish nodes. A novel path metric is also proposed that rates network routes according to the trustworthiness of the intermediate nodes.

Through experimentation on a practical evaluation framework, it is shown that the distributed trust mechanism together with the novel path metric can increase overall traffic throughput in ad-hoc networks where selfish nodes are present.

Index Terms—Trusted Route, Ad-hoc Networks

I. INTRODUCTION

Ad-hoc wireless networks, otherwise known as mesh networks, provide a mechanism for information exchange without the need for pre-existing infrastructure or administrative support. Nodes act as repeaters to transmit data from nearby nodes to peers that are beyond the radio range of the sender. This can result in a network that spans large distances.

Recently mesh networks have become an increasingly popular mechanism to interconnect wireless devices, such as laptop computers and PDAs. Consequently the Intel Cooperation has proposed a new mesh networking standard 802.11s. Unfortunately, a number of important issues surrounding these networks have yet to be addressed. In traditional wireline networks basic support functions such as packet forwarding, routing and network management are carried out by dedicated network elements. In ad-hoc networks these functions are the responsibility of all available nodes.

However, a key problem arising in ad-hoc networks that is not seen in legacy wireless networks, is that the nodes of an ad-hoc network cannot be trusted to perform their functions correctly. Thus the situation arises where one might see selfish nodes using the resources of the network without involving themselves with routing and traffic forwarding. The selfish nodes conserve their own resources, e.g. battery power, to the detriment of the neighbouring devices. Simulation results have shown that if security issues are not taken into account

in routing protocol design then network operation and maintenance will be jeopardized, and network performance will degrade severely [7]. Cooperation enforcement mechanisms must be implemented in ad-hoc networks in order to mitigate the effects of selfish nodes.

II. MOTIVATION

The basic motivation to implement cooperation enforcement mechanisms in ad-hoc networks is to promote fairness and ensure suitable network performance. Several different cooperation enforcement techniques have been proposed in the literature. One approach is to secure the network nodes. Ensuring that all nodes in an ad-hoc network are completely tamper-resistant is near impossible. However, a tamper-proof hardware module, similar to SIM cards in GSM phones, can be used to ensure the correct operation of the network nodes [2]. However, these modules add complexity and cost to the network devices.

Another solution is to employ a dedicated credit server that verifies the cooperativeness of network nodes [11]. The server is used to charge users for sending traffic, and reward users that forward traffic. This requires the network devices to have two network interfaces. A short-range Bluetooth or WiFi connection, for interfacing with the neighbouring nodes, and a long-range interface such as GPRS, for accessing the credit server. Alternatively the devices could wait until they are within range of a proxy static machine that forwards information to the credit server. This mechanism is not ideal in two respects. Firstly, the network devices would possibly need a long-range wireless interface, which is not common with laptop computers and PDAs. Secondly, the credit server implies that there is pre-existing infrastructure. This violates the requirement that only the nodes form the infrastructure of ad-hoc networks.

A better, more-deployable solution to enforce cooperation is a distributed trust and reputation mechanism. Each node determines the trustworthiness of every other node in the network. Thus the nodes can collude to exclude uncooperative nodes from the routing decision and deny them basic network services. A novel distributed trust mechanism is proposed in this paper that enables nodes to assign trust ratings to every other node in the network. Based on these trust ratings uncooperative nodes are excluded from routing decisions, thus improving overall network throughput and reliability.

The authors would like to thank Telkom SA, Siemens, the National Research Foundation (NRF) and the Department of Trade and Industry (DTI) for supporting this research project.

The rest of this paper is organised as follows. Section III details the proposed distributed trust mechanism. Section IV presents a novel high-throughput path metric for routing decisions. Section V describes the architecture and implementation of a suitable evaluation framework. Section VI presents experiment details and results. Section VII compares the proposed mechanisms with those found in the literature, while Section VIII concludes.

III. PROPOSED DISTRIBUTED TRUST MECHANISM

In this section a distributed trust mechanism is proposed. The mechanism provides a simple manner by which nodes can assess the trustworthiness of other nodes in the network, and assign them a *Trust Rating* (TR). In order to establish an accurate TR of a suspect node¹, different sources of trust information must be utilised. In this way if there are uncertainties with regards to a particular source of information other trust metrics can be relied upon to validate the trustworthiness of the suspect node [4]. Four sources of trust information are identified:

- Interactive trust. This information is gathered from direct observations of a suspect node.
- Non-interactive trust. Obtained by querying other devices in the network.
- Rumours. Information spread by concerned nodes about their experiences with a suspect node. Unlike non-interactive trust, rumours are unsolicited.
- Bragging. Information obtained from the suspect node itself.

With all sources of information more recent observations should be given more weight than older observations. Thus the trust ratings respond rapidly to the changing behaviour of a suspect node.

Interactive trust is gathered from observations of neighbouring nodes (one-hop nodes). With promiscuous wireless interfaces neighbouring devices can observe if a suspect node is acting in a cooperative manner. For example, if a packet is sent to a neighbouring node the source node will be able to overhear if this packet is retransmitted or simply dropped. If it is seen that a neighbour has not forwarded a particular packet, then it can be assumed that the node is selfish, and its interactive trust rating will be reduced.

Non-interactive trust is solicited when two nodes are not neighbours, or if they have not spend sufficient time next to each other to deduce a reliable interactive trust measure. This is often the case in ad-hoc networks as nodes may be mobile, or enter and leave the network randomly. In order to solicit non-interactive trust information an INFORMATION REQUEST packet is flooded throughout the network. This packet includes identifiers of the requesting node and the suspect node. Every neighbour of the suspect node responds with an INFORMATION REPLY packet that includes their respective interactive trust information and their own identifier. This is to prevent lying from the suspect node.

¹Suspect nodes are those whose trustworthiness is still undecided

The suspect node itself replies with bragging information, usually based on how much battery power it has left. Bragging information is only used as a last resort. If any node hears bragging that they think is not a true representation of the suspect node's trustworthiness then they can inform the requesting node, and the rest of the network, via a rumour.

Rumours are similar to non-interactive trust except that they can be spread by anyone at any time. Rumours are not solicited and can be used to detect lying suspect nodes. Very little weight should be placed on rumours except if several similar rumours are received confirming that a node is lying.

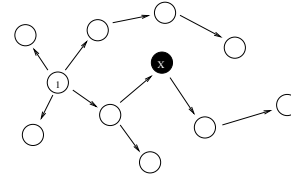


Fig. 1. Node 1 floods an INFORMATION REQUEST packet probing other nodes for information regarding the suspect node X.

Non-interactive trust and rumours are flooded throughout the network, shown in Fig. 1. Therefore, it is unlikely that this information can be intercepted by malicious selfish nodes. In addition, other nodes in the network can view this information and adjust their trust ratings accordingly without soliciting a request themselves, thus saving unnecessary network traffic. Information request packets can also be piggy-backed on ordinary network traffic offering further savings.

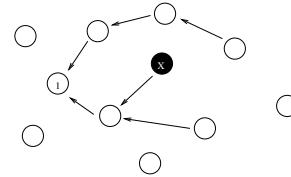


Fig. 2. Neighbours of the suspect node send INFORMATION REPLY packets. The suspect node replies with its bragging information.

Every node maintains a table of trust ratings for all known nodes in the network. The trust ratings are calculated by applying appropriate weights to the different forms of trust and summing them together. The trust rating T for a node n , can be calculated as:

$$T_n = w_1 I_n + w_2 N_n + w_3 B_n + w_4 R_n \quad (1)$$

Where I_n , N_n , B_n and R_n , are the interactive trust, non-interactive trust, bragging and rumour metrics respectively for n . The weights, w_i are adjusted dynamically according to how much relevance is to be placed on the different sources of trust information. T_n is normalised to the range [0..1] so that 0 represents a node that drops all traffic and 1 represents a node that is completely trustworthy. A value of 0.5 indicates that the suspect node's trust rating is still undecided. It has been shown that by employing a trust-based cooperation enforcement mechanism it is possible to reduce selfishness

in ad-hoc networks, such that at equilibrium there will be a coalition of cooperating nodes [9].

IV. THE TRUSTED ROUTE

Several ad-hoc routing protocols have been proposed that allow source nodes to find routes to destination nodes. These protocols can generally be classified as either *on demand* or *table-driven*. On demand protocols only try to establish a route to a destination node when there are packets to be sent. On the other hand, table-driven protocols constantly maintain routes to every other node, thus ensuring that routes are already established when packets need to be sent.

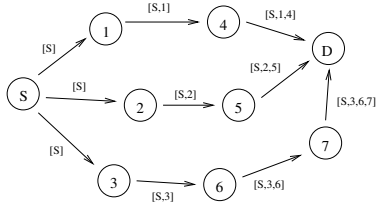


Fig. 3. DSR Route Discovery: Propagating ROUTE REQUEST packets are flooded until they reach the destination node. The destination node chooses randomly between the minimum hop-count routes, and informs the sending node of the chosen route via a ROUTE REPLY packet.

In this paper we focus on a popular on demand routing protocol known as Dynamic Source Routing (DSR) [5]. DSR nodes maintain the entire route to destination nodes in their own route caches. These routes are established by two mechanisms: *Route Discovery* and *Route Maintenance*. During Route Discovery a source node locally broadcasts a ROUTE REQUEST packet with the Time-to-Live field of the IP header set to 1. This allows the source node to query its neighbours for a route to the destination and is called a non-propagating ROUTE REQUEST. If after the nominal one-hop round trip time no reply is received, then the source node sends a propagating ROUTE REQUEST that is flooded throughout the network. This request is answered by either the destination node or another node that has a route to the destination in its route cache. The route caches limit the cost of Route Discovery as nodes can overhear routes and store them for their own use.

When two nodes in an active route move out of range of each other the route fails and a ROUTE ERROR packet is sent to the source node. This mechanism is known as Route Maintenance. The sender then has the choice to use another route to the destination or invoke a new Route Discovery.

DSR considers the route with the least number of hops to be optimal. If two routes have an equal number of hops then the path is chosen randomly between them. A problem with this approach is that the shortest path is not necessarily the best path, particularly when there are selfish nodes present in the network.

In this paper a path metric is proposed that takes into account the trust ratings of intermediate nodes. We call this metric the *Trusted Route*. The Trusted Route metric serves two purposes:

- Routes that only include cooperative nodes will offer higher throughput and reliability. Routes that include

selfish nodes may not function at all and should be avoided if possible.

- By removing selfish nodes from routing decisions they are gradually excluded from the network as if they were not there at all. If selfish nodes can no longer use the network, because they have no routes to destination nodes, their motivation to be cooperative will increase.

In DSR destination nodes that receive route request packets calculate the best path based on minimum hop-count. If they rather based their decision on the trust ratings of the intermediate nodes then better routes can be found. The best route from source to destination is not necessarily the shortest route, but clearly shorter routes must be favoured. The Trusted Route metric is calculated by the following equation:

$$R = \left(\sum_{i=1}^{h-1} T_i \right) (h-1)^{-\frac{3}{2}} \quad (2)$$

Where T is the trust rating of the i th intermediate node in the path, and h is the number of hops between source and destination (there are $h-1$ intermediate nodes). The Trusted Route equation offers a compromise between trust-worthy intermediate nodes and low hop-count routes. Every node periodically issues an INFORMATION REQUEST packet about the nodes in their route caches, unless their is sufficient interactive trust information available, and stores the received information for future reference. If the Trusted Route metric must be calculated, and there is no information available about a node in a particular path, then that node is given a neutral trust rating. This avoids delays in the routing decision.

V. EVALUATION FRAMEWORK

While several cooperation enforcement mechanisms have been proposed in the literature, presently none have been implemented in practical testbeds. Therefore, in this section the architecture and implementation of a practical evaluation framework is presented. The goal being to implement the proposed cooperation enforcement mechanism and integrate the Trusted Route metric into the DSR protocol in a practical setting.

A. Architecture

The architecture of the evaluation framework consists of several machines equipped with wireless networking capabilities. The MAC layer uses a contention based scheme to ensure fair access to the medium without the need for a master / slave relationship. This is to ensure that selfish nodes cannot affect the functioning of neighbouring network nodes at the link-layer. The nodes run routers in order to implement ad-hoc routing protocols, and are able to simulate selfishness by dropping packets selectively.

A static gateway node connects the ad-hoc network to a wireline network. The wireless nodes include the gateway in their routing tables as the default route, therefore all packets not destined for nodes in the ad-hoc network are routed to the gateway node. While the gateway node does not strictly comply with the requirement that ad-hoc networks employ no

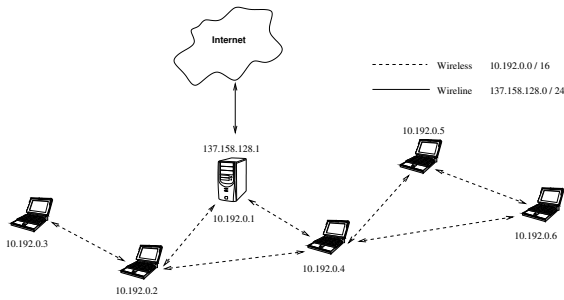


Fig. 4. Logical overview of the evaluation framework architecture.

pre-existing infrastructure, it allows the nodes of the network Internet access, which is a likely use of ad-hoc networks.

B. Implementation

The network nodes are equipped with IEEE 802.11b Cisco Aironet 352 cards running in ad-hoc mode. The 802.11b protocol operates in the 2.4 GHz frequency range, at a maximum data rate of 11 Mbps, and specifies two available medium access functions: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). Since PCF is a polling scheme, often implemented by access points, it is not considered for the evaluation framework. The DCF offers fair access to the wireless medium, and the RTS / CTS mechanism solves the hidden terminal problem [10].

The nodes each run the Click Modular Router [6]. The Click architecture allows for flexible and configurable router configurations that can route packets at high speed. Click runs on the Linux operating system, and when used at kernel level, sits between the protocol stack and the 802.11 interface. Packets are intercepted and modified between the link layer and the network layer. Therefore no alterations are made to the existing OS protocol stack in order to implement the DSR protocol.

UDP is used for all experiments. This is because TCP implements congestion control mechanisms that assume all packet losses occur due to excessive congestion in the network. In ad-hoc networks packet losses may occur due to interference, nodes moving out of range, or node selfishness, and thus TCP does not work well in this environment.

VI. EXPERIMENTS AND RESULTS

Experiments are performed on the evaluation framework in order to emulate practical ad-hoc networks as closely as possible. The details of the experiments performed are presented below:

- The source node S issues a propagating ROUTE REQUEST for the destination node D .
- D calculates the best path through the network using the minimum hop-count metric. This information is then relayed to S using a ROUTE REPLY.
- S then attempts to send ten-thousand 172-byte UDP packets to D . The packets are transmitted every 20 ms. D calculates the number of packets received over the time period, and hence the average data throughput.

- Once the final packet has been sent, all nodes' route caches are scrubbed and the experiment is repeated 200 times.
- The above steps are then repeated, but this time the Trusted Route metric is used instead of the minimum hop-count.

Two different network topologies are used for the above evaluation. The scenarios are designed to illustrate the ability of the proposed distributed trust mechanism to identify selfish nodes, and the effectiveness of the Trusted Route metric in determining efficient network routes.

A. Topology I

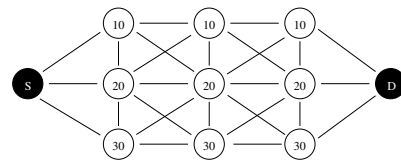
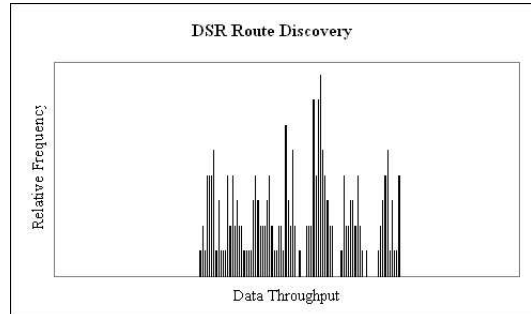
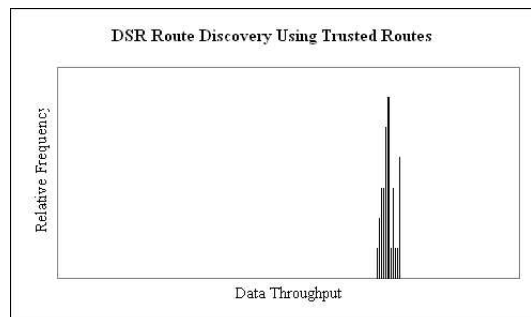


Fig. 5. Testbed topology for first experiment.

The topology of the testbed framework for the first experiment is illustrated in Fig. 5. The values in the figure represent the selfishness of the nodes, e.g. 10 represents a node that drops 10% of all data packets. The nodes forward all routing related and information request packets. In future work we hope to relax this constraint.



(a) Minimum Hop-Count



(b) Trusted Route

Fig. 6. Results from the first experiment. The histograms illustrate the relative frequency that high-throughput routes are chosen using (a) minimum hop-count, and (b) the Trusted Route.

Fig. 6 clearly demonstrates the benefits of using the Trusted Route metric over the minimum hop-count metric with the DSR protocol. In the minimum hop-count histogram the throughput between source and destination nodes varies greatly throughout the experiment. This is attributed to the minimum hop-count metric choosing routes randomly that often include very selfish nodes. The Trusted Route histogram shows that the measured data throughput is not only high but also consistent.

B. Topology II

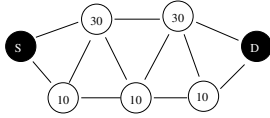
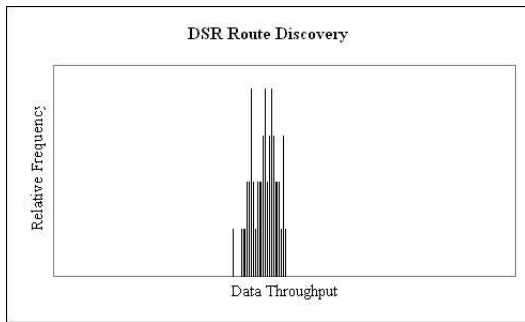
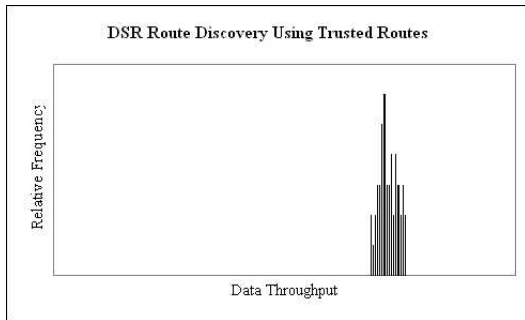


Fig. 7. Testbed topology for the second experiment.

Fig. 7 shows the topology of the testbed for the second experiment. Note that the testbed is intentionally set up so that the selfish nodes form a short route of 3-hops between S and D , and the slightly more cooperative nodes form a longer route of 4-hops.



(a) Minimum Hop-Count



(b) Trusted Route

Fig. 8. Results from the second experiment. The histograms illustrate the relative frequency that high-throughput routes are chosen using (a) minimum hop-count, and (b) the Trusted Route.

The results of the second experiment shown in Fig. 8, show that the minimum hop-count metric consistently selects low-throughput routes, whereas the Trusted Route consistently selects high throughput routes. Under closer inspection it can be seen that minimum hop-count always chooses a route

through very selfish nodes, hence the poor throughput. The Trusted Route chooses longer, higher-throughput links based on the trust-ratings of the intermediate nodes, and consistently out-performs the minimum hop-count metric.

VII. RELATED WORK

Cooperation enforcement schemes, such as CORE [8], CONFIDANT [1], SPRITE [11] and FIRE [4], all aim to reduce selfishness in ad-hoc networks.

CORE is similar to the cooperation enforcement mechanism proposed in this paper, except that CORE cannot accurately verify the reputation of a new node in the network. CORE defines two types of reputation, subjective and indirect, and then splits these categories according to particular functions, for example routing or packet forwarding.

CONFIDANT consists of several components: the monitor, the reputation system, the path manager and the trust manager. The reputation component detects malicious nodes and informs the path manager to scrub routes containing these nodes. The trust manager informs other nodes about neighbour's transgressions. The monitor component receives these messages and passes them to the trust manager. The CONFIDANT protocol can only obtain information about one-hop neighbours, whereas our mechanism actively requests information and assigns trust ratings to all nodes in the route cache.

SPRITE proposes that nodes keep receipts of messages that they forward. These receipts are then sent to a Credit Clearance Service that decides how much to charge and reward individual nodes involved with the a particular transmission. Unlike our system SPRITE relies on pre-existing infrastructure and on people's minor motive of earning money or credits.

The FIRE model is very similar to our mechanism and introduces the concept of certified reputation, which is based on references from other nodes. After every transaction nodes ask for references from their neighbours. In the situation where no other information is available a node offers the references as proof of its trustworthiness. In our system nodes do not provide references when they brag, but are deterred from lying by observant neighbours that will spread negative rumours if they do.

Improving the minimum hop-count path metric has been discussed in the literature. The ETX metric finds high throughput paths in ad-hoc networks by calculating the number of actual radio transmissions required to reach a destination node [3]. The metric uses per-link measurements of packet loss ratios in both directions of a wireless link to predict the number of retransmissions required.

VIII. CONCLUSIONS AND FUTURE WORK

We have proposed a trust-based cooperation enforcement mechanism and a high-throughput path metric for use with the DSR protocol. The architecture and implementation of an evaluation framework was presented that implements the proposed schemes. The Trusted Route metric relies on Trust Ratings from the cooperation enforcement scheme. Experimental results on two different network topologies have shown

that when selfish nodes are present the Trusted Route metric can better identify high-throughput routes in ad-hoc networks than the minimum hop-count metric.

We plan to address the following issues as part of future work. Selfish nodes may act in collusion to increase each other's utility of the network. In this work we have assumed that nodes are purely self interested. However, selfish nodes may cooperate with other selfish nodes in order to increase each other's trust ratings. Moreover, we have not addressed the threat of malicious nodes that aim to purposefully disrupt the network. In future work we hope to better improve the cooperation enforcement model to take into account both colluding and malicious nodes.

REFERENCES

- [1] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks). *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2002.
- [2] Levente Buttyán and Jean-Pierre Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. 2001.
- [3] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. *The Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California*, 2003.
- [4] T. D. Huynh, N. R. Jennings, and N. Shadbolt. FIRE: An Integrated Trust and Reputation Model for Open Multi-Agent Systems. *Proc. 16th European Conference on Artificial Intelligence*, 2004.
- [5] D Johnson and D Maltz. Dynamic Source Routing in Ad-hoc Wireless Networks. *Mobile Computing, T Imielinski and H Korth, Eds.*, 1996.
- [6] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click Modular Router. *ACM Transactions on Computer Science*, 3(18):263–297, August 2000.
- [7] P Michiardi and R Molva. Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks. *European Wireless 2002 Conference*, 2002.
- [8] Pietro Michiardi and Refik Molva. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad-hoc Networks. *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, 2002.
- [9] Pietro Michiardi and Refik Molva. A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad-hoc Networks. *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.
- [10] Fouad Tobagi and Leonard Kleinrock. Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, December 1975.
- [11] Sheng Zhong, Jiang Chen, and Richard Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. 2002.