Incentive Compatible Cost- and Stability-Based Routing in Ad Hoc Networks

Mingming Lu, Feng Li, and Jie Wu *
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

In this paper, we embed an incentive-compatible, efficient, and individual rational payment scheme into our cost- and stability-based routing protocol in ad hoc networks which consist of selfish nodes. Unlike traditional routing protocols in ad hoc networks, which only elicit cost information from selfish nodes, our protocol motivates selfish nodes to report truthfully both their stability and cost information.

Keywords*: Ad hoc networks, routing, incentive-compatibility, payment, stability, VCG mechanism.*

1 Introduction

Two prominent characteristics of ad hoc networks are power scarcity and instability. Existing routing protocols address these two problems either separately or simply combinatorially. Our previous work [5] proposes a new metric to evaluate the efficiency of a routing scheme by integrating link cost and link stability into a single metric and designs an optimal routing algorithm, MaxUtility. In [5], we assume our algorithm has full control of nodes and has priori knowledge of cost and stability information. However, the assumptions are not true in ad hoc networks consisting of selfish nodes. In this paper, we relax the assumptions and design a two-stage routing protocol, which satisfies incentive-compatibility, efficiency, and individual rationality.

In our model, a source intends to buy a path to transmit its packets to a destination and receives a *benefit* for each successfully delivered packet. Although there exists packet loss due to unstable links (nodes), the source still accepts the *partial delivery* of packets if the sum of benefits are larger than the sum of transmission costs.

Assuming that intermediate nodes are selfish and rational but they do not collude and that each node has a priori knowledge about its node stability and the costs of links, where it is an endpoint, our goal is to use incentive methods to motivate intermediate nodes to report true cost and stability information. We use *social welfare*[6], which is defined as the total utility of all the nodes, as the metric to define our goal. Here, the utility of the source is its benefit minus the total payments to the intermediate nodes. A intermediate node's utility is defined as its payment from the source minus its cost.

Most existing works adopt the Vickrey-Clarke-Groves (VCG) mechanism [6] to elicit truthful information from intermediate nodes in order to find the optimal route. The idea behind the VCG mechanism is utilizing the payment to determine intermediate nodes' utility so that each intermediate node's utility is consistent with the social welfare. Therefore, each intermediate node has to tell the truth in order to maximize its utility.

However, the traditional VCG payment scheme [1, 3, 7] adopted in ad hoc networks applies only to the case of successful packet delivery, therefore, the VCG mechanism cannot be directly applied to our problem due to partial packet delivery. If we do not pay intermediate nodes in case of delivery failure, it violates the *individual rationality principle* [6]. This principle points out that each forwarding node should not be worse than a non-forwarding node, i.e. a node should not get negative utility by forwarding a packet. To ensure fairness to intermediate nodes, we design a *partial payment* scheme, in which a node will get paid for each packet that it helps forward, no matter if the packet reaches the destination or not.

Because the incentive method alone cannot stimulate intermediate nodes to report stabilities truthfully, we divide our ad hoc routing protocol into two stages: the route discovery stage and the packet forwarding stage. We postpone the payment calculation after the packet forwarding stage because the true stability can be observed by neighbor nodes during the packet forwarding stage but is unknown in the route discovery stage.

In the route discovery stage, the source collects the costs and stabilities reported by intermediate nodes.

*This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, and CNS 0531410. Email: {mlu2@, fli4@, jie@cse.}fau.edu

A cryptographic technique is integrated to prevent the neighbors from tampering with the reported cost and stability when they forward a route discovery message. In the packet forwarding stage, a neighborhood surveillance mechanism is integrated into our VCG-based payment scheme.

We utilize an incentive method to motivate neighbors to overhear the actual packet forwarding of intermediate nodes, and to report such information to a trusted credit center (TCC). TCC will count the number of packet collected by each node, calculate the stabilities of intermediate nodes, and inform the destination of the stabilities. We also use a cryptography technique to prevent the neighbors from fabricating the information it obtains during the surveillance.

To avoid intractable analysis, we make some assumptions about the behavior of the intermediate nodes, which make the model simple enough but nevertheless lead to useful models. (1) The source and destination are assumed not to be selfish. Although we can remove this assumption by introducing multiple source-destination pairs because competition among source-destination pairs can enforce truthfulness, for the sake of presentation, we consider only one source-destination pair and thus assume both source and destination are obedient. (2) The network is bi-connected. This assumption can suppress the overpayment problem [2], but the overpayment problem cannot be totally removed unless the topology control method [3] is adopted. (3) We assume there is a secure and stable channel, as illustrated in [9], between TCC and each node. The implement of TCC can be either a centralized or distributed implementation.

2 Preliminaries and Related Work

2.1 The related work

Anderegg and Eidenbenz [1] first apply the VCG mechanism to routing protocols in ad hoc networks. They adopted the energy cost of nodes as private information, used the lowest energy cost model to study the ad hoc routing problem. They also presented a reasonable way to estimate a node's cost, whose drawback was pointed out by [10].

Zhong, Li, Liu, and Yang [10] used a two-stage routing protocol to model the ad hoc routing problem. They integrated a novel cryptographic technique into the VCG mechanism to solve the link cost dependence problem. However, none of them took into account the partial delivery and partial payment problem, and thus failed to consider stability as a metric to evaluate the performance of a path in ad hoc networks.

Zhong, Chen, and Yang [9] proposed a system called Sprite, which combines incentive methods and cryptography techniques to implement a group cheat-proof ad hoc routing system. But they only considered the payment problem in the packet forwarding stage and failed to compensate the last node which has forwarded a packet. Compared to [9], we also utilize a centralized authority to collect overheard receipts and distribute credit, but the main differences are that we consider both routing and payment issues, and our payment scheme satisfies the individual rationality principle.

2.2 Preliminaries

In our previous work [5], we consider a unicast routing problem, in which a source s intends to buy a path to send packets to a destination d in an ad hoc network with unstable nodes. The stability $t_{i,j}$ of link (i,j) is modeled as the ratio of received packets by node j to transmitted packets by node i. Inspired by the idea that markets can efficiently allocate limited resources, we model an ad hoc network as a market, in which s is the buyer and intermediate nodes are sellers.

For each successfully delivered packet, s obtains a benefit v. Because of unstable links on the selected route R, packet delivery is not 100% successful. The probability that s obtains v is the product of stabilities of all links on R, i.e., $\prod_{(i,j)\in R} t_{i,j}$. Thus, for each packet, s obtains expected benefit $v \times \prod_{(i,j)\in R} t_{i,j}$ and pays g_i to each intermediate node i on R. The expected utility of s is $u_s = v \times \prod_{(i,j)\in R} t_{i,j} - \sum_{i\in R} g_i$.

For each link $(i,j) \in R$, the transmission cost is $c_{i,j}$. Node i spends $c_{i,j}$ on forwarding a packet to node j. Since a packet is delivered from s to i with probability $\prod_{(x,y)\in R_i} t_{x,y}$, where R_i is the subpath (from s to i) of R, the expected cost of node i is $c_i = c_{i,j} \times \prod_{(x,y)\in R_i} t_{x,y}$ and hence, i's expected utility is $u_i = g_i - c_i$.

The utility of any node not on the selected path is 0. Therefore, the total expected utility of the whole network over a packet delivery on path R is $U = u_s + \sum_{i\in R} u_i = v \times \prod_{(i,j)\in R} t_{i,j} - \sum_{i\in R} c_i$. Our previous work [5] designed an optimal algorithm, called MaxUtility, to compute the route that maximizes the total expected utility.

We observe that for a single-link route (i,j) with link stability $t_{i,j}$ and link cost $c_{i,j}$, the total expected utility is

$$U = v \times t_{i,j} - c_{i,j} \tag{1}$$

The basic idea of the MaxUtility is to calculate the total expected utility backwards, starting from d with initial value being v and repeatedly applying Formula (1) recursively over each link. We illustrate the basic idea by an example shown in Fig. 1. Let $v = 200$. Initially, $U = v = 200$. By applying Formula (1) over link $(1,d)$ ($(2,d)$), at node 1 (2), $U = 200 \times 0.85 - 30 = 140$ (137). Through link $(1,2)$, node 2 can compute a new value of $U = 137 \times 0.7 - 25 = 70.9$ to node 1, which

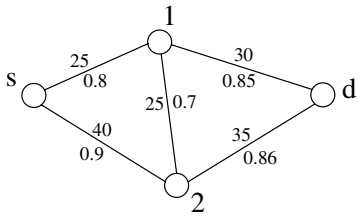

Figure 1. An example illustrating the MaxUtility algorithm.

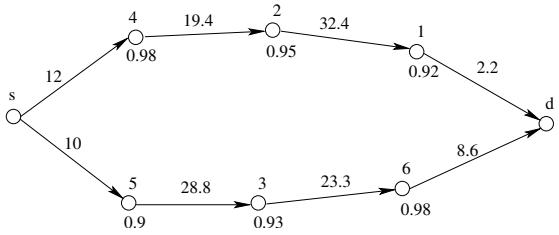

Figure 2. An illustration on how the VCG payment enforce truthfulness on cost but fails to enforce truthfulness on stability.

is less than node 1's current U, 140. In the same way, node 1 cannot update node 2's value. In the end, node 1 (2) updates s' U, which is the final total expected utility. The readers can verify the optimal route is $< s, 1, d >$, whose total expected utility is $140 \times 0.8 - 25 = 87$.

3 Problem Statement

3.1 Model and notations

In our model, the set of mobile nodes $\mathcal{N} = \{1, 2, \cdots, N\}$ forms an ad hoc network, which is modeled as a unit disk graph $(\mathcal{N}, E)$, where E is the set of links. We consider that a source s intends to buy a path to a destination d to deliver packets. For each successfully delivered packet, s will get a benefit v.

For a link (i, j), the link cost $c_{i,j}$ is the minimal power level to connect i and j. Unlike our previous work [5], which considers link stability, in this paper, we consider node stability because node stability characterizes the ability and willingness of a node to help other nodes, and more importantly, the MaxUtility can still apply. For each node i, its stability t_i is the ratio of transmitted packets to received packets. Because s and d intend to communicate and thus have incentive to remain stable during data transmission, we have $t_s = t_d = 1$.

The key point in our ad hoc routing problem is to elicit true cost and stability information from nodes. Only if the routing protocol can elicit true information ω, can the MaxUtility find a real optimal routing path, because ω is the input for the MaxUtility and with different inputs the MaxUtility will return different routing paths. To discriminate the reported information from the real information, $\hat{\omega}$ and ω are used to represent the reported value and the real value, respectively.

To motivate intermediate nodes to tell the truth, i.e. $\hat{\omega} = \omega$, we design a payment scheme, which involves incentive and cryptographic techniques, so that an intermediate node can maximize its utility if and only if it tells the truth on its stability and cost. Our payment scheme is based on but different from the traditional VCG mechanism.

The traditional VCG mechanism can be used to stimulate selfish nodes to cooperate and reveal their private cost information honestly. However, the VCG mechanism cannot be applied to our ad hoc routing problem.

3.2 Failure of the VCG mechanism

We first show how the VCG mechanism can enforce truthfulness in the lowest energy cost routing problem. In the lowest cost routing, node i's VCG payment is the reported cost of the second best path minus the reported cost of the best path plus i's reported cost.

In Fig. 2, there are two paths from s to d. The real link cost is labelled above each link, while the node stability is labelled below each node. If we consider only link cost, the upper path has the real cost 66, while the lower path has the real cost 70.7. Obviously, the upper path should be the optimal (lowest cost) path. However, if node 5 intends to increase its utility by reporting a fabricated cost, say, 20, the optimal path will be the lower path (with cost 61.9). Then node 5's payment will be $66 - 61.9 + 20 = 24.1$, where 66 is the cost of the second best path, 61.9 is the cost of the best path, and 20 is the reported cost of node 5. Note that node 5's utility is $24.1 - 28.8 = -4.7$, where 28.8 is node 5's real cost, but its utility will be 0 if it tells the truth. Because reporting a fabricated cost will not increase node 5's utility, node 5 has no incentive to cheat.

If we take link stability into account, the true expected social welfare of the upper path is 22.6, while the true expected social welfare of the lower path is 19.5. The upper path should be the optimal (maximum expected social welfare) path. However, if node 5 reports a fabricated stability 1 instead of real stability 0.9, the expected social welfare of the lower path will be 22.8. According to the VCG mechanism, in the maximum expected social welfare routing problem, the payment to node i is the expected social welfare of the best path minus the expected social welfare of the second best path plus node i's expected cost. So the payment for node 5 is $22.8 - 22.6 + 28.8 = 29$. Note node 5's reported and real

expected cost are $28.8 \times 1 = 28.8$ and $28.8 \times 0.9 = 25.9$, respectively. Node 5's utility is $29 - 25.9 = 3.1$ in case of cheating, while its utility is 0 in case of telling the truth. Therefore, node 5 can increase its utility by cheating. This example illustrates that the traditional VCG payment scheme cannot guarantee truthfulness in our model.

3.3 The payment scheme

In this subsection, we design a payment scheme so that each node has incentive to report its true cost and stability. To enforce truthfulness of stability, the only tool we can utilize is the payment. At the beginning of a routing session, an intermediate node can utilize all possible strategies to maximize its potential utility. The advantage of intermediate nodes are that their stabilities as well as their costs are private to the routing protocol and they can strategically report fabricated values of their stabilities and costs.

The advantage of the routing protocol is that it can determine the amount of the payment and the time to calculate the payment. It is unnecessary for the protocol to compute the payment and then pay the calculated amount to each selected intermediate node during the routing discovery stage. Instead, it can postpone the payment calculation until the end of the packet forwarding stage.

In our payment scheme, each intermediate node i will receive a payment g_i for each packet after packet forwarding stage. For each packet, i's utility is $u_i = g_i - c_i$, where c_i is the expected cost. Unlike the goal of our routing protocol, which is to maximize the expected social welfare U, the goal of an intermediate node i is to maximize its own utility u_i. In order to prevent intermediate nodes from cheating on their stability and cost, the payment scheme should make the goal of any intermediate node consistent with the goal of the routing protocol. Otherwise, an intermediate node will have incentive to lie.

To achieve that consistency, each intermediate node i's utility must be equal to the *marginal expected social welfare* of i, i.e.

$$u_i = U(R^*) - U(R^*_{-i})$$

where $R^* = \arg\max_{R \in \mathcal{R}} U(R)$ is the optimal route, R^*_{-i} is the optimal path if i were removed from the network, and $U(R^*)$ and $U(R^*_{-i})$ are the expected social welfare associated with paths R^* and R^*_{-i}, respectively. $U(R^*) - U(R^*_{-i})$ is called marginal social welfare because it reflects the part of the expected social welfare contributed by i alone.

Note that if i is in R^*, then $U(R^*) \geq U(R^*_{-i})$, which means i will get a non-negative utility; otherwise, $U(R^*) = U(R^*_{-i})$, and thus, i will have zero utility. Now, i's payment can be defined as $g_i = U(R^*) - U(R^*_{-i}) + c_i$. For convenience, we use notation $U_{-i}(R^*)$ to denote $U(R^*) + c_i$. Therefore, i's payment is

$$g_i = U_{-i}(R^*) - U(R^*_{-i})$$

As we have discussed, the optimal routing path R^* is identified in the routing discovery stage. At that stage, the only information available is the reported information $\hat{\omega}$, which is not necessary equal to real information ω. To discriminate the optimal path found in the route discovery stage and the real optimal routing path, we use notation $R^*(\hat{\omega})$ to denote the former path, and $R^*(\omega)$ to denote the latter path.

Our approach uses a neighbor surveillance method, which is executed during the packet forwarding stage, to collect the true stability information. Thus, at the time to compute the payment, the routing protocol can utilize both real stability information and the reported information $\hat{\omega}$. In our payment scheme, i's true stability is used in the calculation of i's payment.

To represent the properties of our two-stage routing protocol: routing based on reported information and payment based on true stability as well as reported information, we include one more parameter in the calculation of the expected social welfare of paths $R^*(\hat{\omega})$ and $R^*_{-i}(\hat{\omega})$, i.e. $U_{-i}(R^*(\hat{\omega}), \hat{\omega}^i)$ and $U(R^*_{-i}(\hat{\omega}), \hat{\omega})$, respectively, where $\hat{\omega}^i$ denotes the cost and stability information profile, in which i tells the truth. The complete payment for intermediate node i is

$$g_i = U_{-i}(R^*(\hat{\omega}), \hat{\omega}^i) - U(R^*_{-i}(\hat{\omega}), \hat{\omega})$$

g_i and u_i are functions of information $\hat{\omega}$. For brevity, we use g_i and u_i to denote $g_i(\hat{\omega})$ and $u_i(\hat{\omega})$, respectively.

We use Fig. 2 to show how our payment scheme works. In Fig. 2, we still assume node 5 lies on its stability with value 1 instead of the real value 0.9. Again, the optimal path we can find in the route discovery stage is the lower path with maximum expected social welfare 22.8. But when the routing protocol begins to calculate node 5's payment, it already has node 5's real stability 0.9. Thus, the routing protocol can calculate the expected social welfare when node 5 tells the truth, which is 19.5. Therefore, the payment of node 5 is $19.5 + 25.9 - 22.6 = 22.8$. But node 5's cost $25.9 > 22.8$, thus node 5 will get negative utility and will have no motivation to lie.

Our payment scheme is incentive compatible, efficient, and individually rational. Due to the space limitation, we omit the proofs of these properties.

4 A two-stage routing protocol

4.1 The route discovery protocol

In this subsection, we present an on-demand link state based routing protocol in which the source initiates

a route discovery. In traditional link state based protocols, information is spread through flooding techniques. Initially, every node broadcasts its local network view (the cost of each link and the stability of the node itself) to every other node. At the end of this process, every node has a global network view of the network (consistent, up-to-date routing information). Here we adopt a reactive version of the link state approach, assuming (i, j) exists if and only if (j, i) exists.

1. Source sends out a flooding message.
2. Each intermediate node responds to the first request by replying to the message and then forwarding it.
3. The globally directed flooding tree is formed rooted at the source. The first requester becomes the parent of the corresponding node.
4. Each node sends out its encrypted link state (cost of each link and the stability of node itself) to its parent node.
5. The source collects all link state information through the reversed spanning tree, and then, applies the MaxUtility to determine the optimal path.

Our centralized implementation spreads local information in a distributed manner, but computes the optimal routing path at the source in a centralized way. It requires every node to maintain local link state information. The source is in charge of computing the optimal path.

Initially, s broadcasts a request message REQ, and then collects all the $ACKs$ from neighbors before timeout and sets each of those neighbors as its $Child$. After collecting the encrypted link states from all children, s decrypts the link state, constructs a global network view, and calls the MaxUtility.

Any node j (other than s) that receives REQ will check if it is the first time to receive REQ. If so, j will set the sender as the parent, send the sender ACK, forward REQ, and set the timer to receive ACK.

If j receives an ACK from a neighbor before timeout, it will add the neighbor to its $Child$. If j does not receive ACK before timeout, j will identify itself as a leaf node of the flood tree. The leaf node j will encrypt its link state and send the encrypted link state to its parent.

If node j receives a link state message from its neighbor, it will attach its neighbor's encrypted link state to its own link state. After attachment, the node will send the new link state message to its parent.

The source will not calculate the payment for each node until the end of the packet forwarding stage, in order to reduce the effect of fabricated stability. In the packet forwarding stage, we use a neighbor surveillance mechanism to elicit the true value of stability, and compare the true value with the reported value to determine the unit payment received by each node.

4.2 The packet forwarding protocol

In this subsection, we present the protocol for the packet forwarding stage. In the protocol, we utilize the fact that when a selected forwarding node i relays a packet, its non-forwarding neighbor nodes will also receive (overhear) the packet even if they are not the intended receiver.

We can assume the existence of surveillant nodes, such as clusterheads, in the ad hoc network. Also, each clusterhead has a backup clusterhead. In this way, each node, including clusterhead, has a constant number (no more than 12) of clusterhead neighbors, primary or backup. We can use clusterheads only to surveil and report.

To utilize the neighbor surveillance property, we design a mixed approach combining both a cryptographic technique and an incentive method. The cryptographic technique is applied to restrict the actions of the surveillant nodes so that surveillant nodes can only submit or drop the overheard information. We provide a bonus to each valid overheard information, thus, surveillant nodes do have motivation to submit the overheard information and cannot fabricate it.

We adopt a trusted credit center (TCC) to collect the overheard information from non-forwarding nodes. To save bandwidth and storage, instead of a packet itself, the TCC accepts the digest of the packet as a receipt. To motivate nodes to report receipts, the TCC provides a bonus σ for each valid receipt. We assume σ is larger than the cost φ of submitting a receipt. The surveillant cost 12φ can be integrated into the link cost $c_{i,i+1}$. The source can find the optimal routing path based on the modified expected social welfare.

Since submitting a receipt increases the utility of a surveillant node, the surveillant node has motivation to report receipts. But it is still possible for surveillant nodes to fabricate receipts. To prevent fabricated receipts, each forwarding node i will use a hash function to generate a digest for each received packet, encrypt the digest with a shared key between i and the TCC to generate an encrypted receipt to be attached to the packet. Because those surveillant neighbors of i do not know the shared key, they cannot fabricate the encrypted receipt.

To help the TCC verify received receipts, the source s will digest each packet before sending. After sending all the packets, s will use the shared key between s and the TCC to encrypt the set of digests, and submit the encrypted digest to the TCC.

After the packet forwarding stage, each node can submit its collected receipts to the TCC. The TCC will decrypt the receipt set from s and receipts from intermedi-

ate node i, and count the number of packets forwarded by each forwarding node i, which is denoted as f_i. After collecting all the receipts, the TCC will inform the source s of the value of each f_i. s calculates the stability t_i based on $t_i = \frac{f_i}{f_{i-1}}$, where $i-1$ is i's predecessor on the selected routing path $R^*(\hat{\omega})$. By using stability information, s can calculate each node's payment based on the payment calculation proposed in Section 3.

5 Discussion

In this paper, we have assumed that the number of packets is large enough so that the observed stability can be equal to the true stability. Even though the number of packets is not large enough, we can adopt the link quality indicator [8], where the link stability can be measured over the reception of a single packet in a realistic environment. Node stability can be calculated based on the link stability. However, we cannot use the VCG payment because, in practice, the observed stability is not equal to the real stability but within the confidence interval of the real stability. But we can use the first price path auction approach [4] to deal with this problem.

In the first price path auction [4], each intermediate node on the selected path will be paid the amount it bids (reported cost) during the route discovery stage. Compared with the VCG mechanism, the first price path auction does not require complicated computation and is easy to implement, but it is harder to reach an equilibrium than the VCG mechanism. To enforce Nash equilibrium in first price auction, the source can pay a bonus, which increases as the bid decreases, to each intermediate node not on the optimal path so that the forwarding node on the optimal path cannot report a cost value far more than its real cost.

6 Conclusion

In this paper, we propose an incentive compatible cost- and stability-based routing in ad hoc networks, which integrates both partial delivery and partial payment in a unified framework. We embed our payment scheme into our cost- and stability-based routing model in an environment with selfish and unstable nodes. The routing process is divided into two stages: the routing discovery stage and the packet forwarding stage, and define a cost and stability reporting and collection scheme in the routing discovery stage and a neighbor surveillance scheme in packet forwarding stage. We prove that our protocol is incentive compatible, efficient, and individually rational under the assumption that the observed stability is equal to real stability. As part of our future work, we will study a different model, in which both packet transmission and packet receiving consume energy for a mobile node. We will further enhance the proposed model so that the source node's interest could be included.

References

[1] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of MOBICOM '03*, pages 245–259, 2003.

[2] A. Archer and E. Tardos. Frugal path mechanisms. In *SODA '02: Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 991–999, 2002.

[3] S. Eidenbenz, G. Resta, and P. Santi. Commit: A sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes. In *Proceedings of IPDPS'05*, 2005.

[4] N. Immorlica, D. Karger, E. Nikolova, and R. Sami. First-price path auctions. In *EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 203–212, 2005.

[5] M. Lu and J. Wu. Social welfare based routing in ad hoc networks. Accepted to appear in Proceedings of ICPP'06, 2006.

[6] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[7] W. Wang and X. Li. Truthful low-cost unicast in selfish wireless networks. In *Proceeding of Ad Hoc and Sensor Networks (WMAN) in conjunction with IPDPS*, 2004.

[8] Y. Wang, M. Martonosi, and L. Peh. A new scheme on link quality prediction and its applications to metric-based routing. In *Proceedings of ACM SENSYS'05*, pages 288–289, New York, NY, USA, 2005.

[9] S. Zhong, J. Chen, and Y. Yang. Sprite: A simple, cheatproof, credit-based system for mobile ad hoc networks. In *Proceedings of INFOCOM '03*, April 2003.

[10] S. Zhong, L. Li, Y. G. Liu, and Y. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques. In *Proceedings of MOBICOM '05*, pages 117–131, 2005.