

A Price-based Reliable Routing Game in Wireless Networks

Hua Liu and Bhaskar Krishnamachari
{hual,bkrishna}@usc.edu

Departments of Computer Science and Electrical Engineering
Viterbi School of Engineering
University Of Southern California
Los Angeles, CA 90089

We investigate a price-based reliable routing game in a wireless network of selfish users. Each node is characterized by a probability of reliably forwarding a packet, and each link is characterized by a cost of transmission. The objective is to form a stable and reliable routing path between a given source and destination pair. The pricing mechanism involved in this routing game is destination-driven and source-mediated: for each successfully delivered packet, the destination node pays the source, which in turn compensates all nodes that participate in routing the packet. We develop a polynomial-time algorithm for deriving an efficient Nash equilibrium routing path. We also present simulations to evaluate the reliability of the obtained path with respect to prices and source-destination cooperation for different network settings.

I. INTRODUCTION

Recently, there has been increasing interest in applying the tools of game theory to the design of wireless ad hoc networks. This is because a central problem in this domain is providing incentives for selfish users to cooperate with each other in moving information through the network.

We consider a reliable routing game for wireless networks of selfish users that is based on the game-theoretic models proposed and investigated by Kannan, Sarangi, and Iyengar [1], [2], [3], [4], with some modifications. Each node in the network is able to forward a given packet sent to it with some probability (we treat this probability in the abstract in this work, but in practice this unreliability could be caused by processor utilization, sleep cycling, buffer overflow, bandwidth limitation, etc.). The delivery probability of a packet from the source to the destination is then the product of the intermediate node forwarding probabilities. Further, the transmission of a packet at each hop has a cost that depends upon the link quality. The nodes in the network are essentially selfish in that they need compensation if required to relay information for others.

We present and investigate a pull-based routing game that is *destination-driven* and *source-mediated*. This means the destination node will pay some amount of virtual credit/money as payment to the source node for each packet of information that is delivered to it. To motivate nodes on the path, the source

node then offers some kind of payment to every node on its path for every packet it forwards. Given this payment from the source, each node on the path has an incentive to participate in this routing game if it receives more payment in expectation than it pays for each transmission. We consider two kinds of behaviors for the source with respect to the destination: cooperative and selfish. A *cooperative source* will accept any positive payoff, and cooperates with the destination because it is also interested in seeing this information routed end-to-end with optimal reliability. A *selfish source* is interested only in maximizing its own expected profit and is even willing to select a path of potentially suboptimal reliability in order to get that maximum profit.

It is important for the route that is determined to be stable. This means every participating routing node should be faithful and keep forwarding packets along the chosen routing path. From a Game Theory perspective, such a stable configuration corresponds to a Nash equilibrium. The prior literature on this topic has suggested that finding the Nash equilibrium for related reliable routing problems can be NP-hard. We show that for the problem we consider here, a polynomial-time solution exists to find efficient Nash Equilibria; this is based on a suitable modification of Dijkstra's shortest path algorithm.

We also present simulations to evaluate the reliability of the obtained route as a function of the destination and source-offered payments and degree of source-destination cooperation for different network parameter settings.

II. RELATED WORK

The problem of obtaining cooperative routing behavior in wireless ad hoc networks consisting of inherently selfish nodes has received considerable attention in recent years. Two main avenues of research in this regard are (a) reputation and punishment-based techniques and (b) pricing and payment-based techniques.

Reputation-based techniques provide mechanisms to track the behavior of nodes and punish those that behave in a selfish manner. Along these lines, Marti *et al.* [6] present the watchdog and path-rater mechanisms that punish nodes which don't relay packets correctly; the CONFIDANT protocol [7], [8] and the CORE mechanism [22] are also distributed reputation

systems that seek to identify and deal with misbehaving nodes. The OCEAN mechanism [9] seeks to obviate some of the complexity associated with second-hand reputation exchange-based schemes by relying on first-hand observations alone. Srinivasan *et al.* [18], provide a formal game-theoretic framework for reputation/punishment and show that the generous tit-for-tat mechanism can be used to obtain Nash equilibria that converge to Pareto optimal, rational solutions. Equilibrium conditions obtained using similar generous tit for tat strategies taking into account the multihop network topology for static and dynamic scenarios are investigated in [12], [13]. Altman *et al.* advocate a less aggressive punishment policy to improve performance [14]. Urpi *et al.* [10] and Nurmi [17] model the situation as dynamic Bayesian games, which allow effective use of prior history in enforcing cooperation.

The alternative to enforcing cooperation is providing nodes with an incentive to cooperate through payment and pricing mechanisms. Buttyan and Hubaux introduce the notion of NUGLETS, a form of virtual currency that provide an incentive for nodes to cooperate [16]. The use of pricing to obtain incentives for cooperation is also advocated in the works by Crowcroft *et al.* [21] and Ileri *et al.* [15]. In all these schemes, nodes which forward data for others receive credits that can be used to pay others to carry their own data. DaSilva and Srivastava [11] study the tradeoffs between cost and benefit in a game theoretic context to determine how they impact cooperation. Our work can be viewed as closely related to these approaches, as we too provide incentive to the intermediate nodes to cooperate in the routing through the payment offered by the source node, and evaluate the impact of pricing upon cooperation and the utility provided to the source.

With payment-based schemes, however, there is an associated risk of cheating due to false claims by nodes trying to obtain payments they do not deserve. While we do not explicitly tackle this issue in our work, researchers have proposed solutions for handle this potential abuse. The micropayment scheme presented in [20] incorporates an audit mechanism to prevent false claims. SPRITE is another cheat-proof mechanism that uses a credit clearance server to provide payments to nodes for cooperation. Anderegg and Eidenbenz [19] propose the use of the Vickrey-Clark-Groves mechanism to obtain truthful claims for payments.

Our investigations are motivated by the works of Kannan, Sarangi and Iyengar on reliable query routing [1], [2], [3], [4]. They are the first to formulate a game where the node utilities show a tension between path reliability and link costs, and they have considered different interesting variants of this problem. A key difference in this work is that we explicitly allow the null strategy in which nodes may choose not to forward packets to any next-hop neighbor. This allows us to provide a polynomial time algorithm for obtaining an efficient Nash equilibrium path. Another key difference in our work is that we consider the notion of destination and source payments and incorporate them into the utility functions.

III. PROBLEM DEFINITION

In this section, we define the destination driven pricing routing problem formally. A wireless network is modeled as an undirected graph $Graph(V, E)$ where V denotes all the nodes in the network and E represents the link set. Each node v_i in V is associated with a reliability parameter R_i ($0 \leq R_i \leq 1$). R_i indicates the node availability and stability – the probability that it can forward a packet sent to it. Each edge $e = (v_i, v_j) \in E$ has a link cost parameter $C_{i,j}$, which represent the communication set up cost between two end nodes.

There are three kinds of nodes in the network: destination node dst , source node src and other intermediate nodes v_i (where $v_i \in V \setminus \{src, dst\}$) that are candidates for participating in a route between the source and the destination. We assume that both destination node and source node always have node reliability 1. The destination node offers to the source a payment amount G for every packet that is successfully delivered to it. The source in turn offers a payment p (for each successfully delivered packet) that will be given to any intermediate node if it participates in the routing path.

To formulate the core game, we now give the definition of the triplet $(I, (S_i)_{i \in I}, (u_i)_{i \in I})$ where I is the set of players; $(S_i)_{i \in I}$ is the set of available actions with S_i be the non-empty set of actions for player i ; and $(u_i)_{i \in I}$ is the set of payoff functions.

In this game, we define $I = V \setminus \{dst\}$ which means that all nodes except the destination are players¹. In an n nodes network (including source and destination nodes), for each node $v_i \in V \setminus \{dst\}$, its strategy is an n -tuple $S_i = (s_{i,1}, s_{i,2}, \dots, s_{i,n})$ where

$$s_{i,j} = \begin{cases} 1 & , \text{if node } v_j \text{ is } v_i\text{'s next hop in path} \\ 0 & , \text{otherwise} \end{cases}$$

Note that $v_i \in V \setminus \{dst\}$ and $v_j \in V$.

Each strategy tuple has at most one 1. That is,

$$\forall v_i, \sum_{j=1}^n s_{i,j} \leq 1$$

If node v_i 's strategy tuple contains all zeros, node v_i does not participate on packet forwarding in the game. A system strategy profile $(S_i)_{i \in I}$ is a profile which contains all players' strategies in the network. Given this strategy profile, there is either no path from the source to the destination, or else, there is exactly one path \mathcal{P} (since each node can point to only next-hop). Without loss of generality, let's denote $\mathcal{P} = (src, v_1, v_2, \dots, v_h, dst)$. Here h denotes the number of hops between the source node and the destination node (not inclusive). The utility function for each player is defined as follows:

¹While the destination does play a role in offering the payment G , this is a constant that only affects the utility for the source.

For the source node:

$$u_{src} = \begin{cases} 0, & \text{if no path exists} \\ (G - h \cdot p) \prod_{v_i \in \mathcal{P}} R_i - C_{src, v_1}, & \text{otherwise} \end{cases} \quad (1)$$

The utility of the source node equals to the difference between the expected income of the source and the link set up cost from the source node to the first next hop routing node. The expected income of the source is the destination payment minus the source pay to all the intermediate nodes times the probability that the packet is successfully delivered.

For each other node v_i :

$$u_{v_i} = \begin{cases} 0, & \text{if no path exists or if } v_i \notin \mathcal{P} \\ p \prod_{v_k}^{v_k} R_i - C_{v_i, v_{i+1}}, & \text{otherwise} \end{cases} \quad (2)$$

(where we are denoting v_i as the i^{th} node in the path if it participates in it).

The utility of each intermediate routing node equals to the expected payment it obtains from the source node times the ongoing route reliability minus the transmission cost per packet to its next hop neighbor. If the node does not participate in the routing, it gains (and loses) nothing.

We now develop an algorithm to obtain an efficient Nash equilibrium for this game.

IV. THE ALGORITHM

Our goal is to develop an algorithm for computing an efficient Nash equilibrium path that provides maximum reliability while ensuring that all nodes obtain non-negative payoffs². The link between non-negative payoffs and the equilibrium path is given by the following simple lemma.

Lemma 1 If a path exists and it is a Nash Equilibrium, every node on the path must have non-negative payoff.

The proof for this lemma is straightforward. According to the payoff function, a node would rather choose not to participate in routing (with payoff 0) if joining the routing makes its payoff negative. However, note that it is not necessary for all the paths with non-negative payoff to be Nash equilibrium. We will term such a path a PPP (Positive Payoff Path). We will correspondingly term a path with all routing nodes having non-positive payoff an NPP (negative payoff path).

To find a positive payoff path, we first simplify the problem to a more concise representation. According to the definition, we need that for each intermediate routing node v_i , its utility $u_{v_i} \geq 0$. This implies

$$\prod_{k=i}^n R_k \geq \frac{C_{i, i+1}}{p}$$

²We should note that in our model even any shortest-hop path that ensures non-negative payoffs to all nodes is in Nash equilibrium. The algorithm we present could be potentially modified to provide such a shortest-hop Nash equilibrium path; however, our interest is in finding an efficient equilibrium path that also provides maximum reliability. This allows us to characterize the performance of the most efficient equilibrium path that can be obtained under different prices.

To convert the product to summation, we take the logarithm of both sides and get

$$\sum_{k=i}^n \log R_k \geq \log \frac{C_{i, i+1}}{p}$$

Notice that $0 \leq R_k \leq 1$; we take the inverse of each R_k to make each term in the summation positive. The original formula now transforms to

$$\sum_{k=i}^n \log \frac{1}{R_k} \leq \log \frac{p}{C_{i, i+1}}$$

for each v_i . Replacing $\log \frac{1}{R_k}$ by r_k ($r_k \geq 0$) and replacing $\log \frac{p}{C_{i, i+1}}$ by $c_{i, i+1}$, we formulate the problem of finding a PPP in the original graph to an equal problem of finding an NPP in a transformed network graph, where each node has a positive value r_i and each edge is assigned a value $c_{i, j}$, according to the following transformed utility functions \tilde{u} .

For the intermediate node,

$$\tilde{u}_{v_i} = \sum_{k=i}^n r_k - c_{i, i+1}$$

For the source node, we get

$$\sum_{k=1}^n \log \frac{1}{R_k} \leq \log \frac{G - hp}{C_{src, v_1}}$$

Replacing $\log \frac{1}{R_k}$ as r_k as before and also replacing $\log \frac{G - hp}{C_{src, v_1}}$ as $c_{src, nbr}$, we have:

$$\tilde{u}_{src} = \sum_{k=1}^n r_k - c_{src, nbr}$$

With these log-transformed formulae, in the following, we will first find an NPP of smallest $\sum r_k$ from each neighbor of source node. Then, if the source node is selfish, it picks up a feasible path provided by neighbors that gives it smallest $\sum r_k - c_{src, nbr}$ or else if cooperative with the destination, it picks the path with the smallest $\sum r_k$. In either case, the source only participates in routing if its own original expected utility will be positive.

A polynomial time algorithm modified from Dijkstra's algorithm can be applied to find the NPP with the smallest $\sum r_k$ from each neighbor of the source to the destination. The pseudocode for the algorithm is given below.

Note that the original source does not participate in this algorithm, so we denote the neighbor in question as src in the algorithm. In brief, the algorithm starts labeling nodes from the destination, applying Dijkstra's algorithm, with adding negative utility checking step. In the algorithm, each node has a label which is a tuple $(from, l(v_i), \tilde{u}_{v_i})$. The first item in the tuple indicates from which node the label comes, i.e., the next hop of current node starting from source. The second term in the tuple records the summation of r_k , which is analogous to the length in Dijkstra's algorithm. The third term tracks the current \tilde{u} value. This algorithm is applied in turn for each

Finding an NPP with Minimum $\sum r_k$ in Transformed Network Graph

- 1) *Initialize*: Feasible set $FS = \{dst\}$, all other nodes labeled as $(-\infty, -)$, $l(dst) = 0$
 - 2) while $src \notin FS \wedge N(FS) \neq \emptyset$
 - for each $v_i \in N(FS)$
 - while $(\exists v_k \in FS \text{ such that } (v_i, v_k) \in E)$
 - * $l(v_i) = \min(l(v_i), \min_{v_j \in FS \wedge (v_i, v_j) \in E} (l(v_j) + r_i))$ let v_j be the corresponding next hop node
 - * if $\tilde{u}_{v_i} - c_{i,j} \geq 0$: delete edge (v_i, v_j) .
 - * else: update the label triplet to $(v_j, l(v_i), l(v_i) - c_{i,j})$; add v_i to FS ; break
 - end while
 - end for
- end while
-

neighbor of the source before the source picks one of these neighbors to form the path, as described above.

Since the r value is related to nodes instead of the links, we need a definition of neighborhood set for vertices in a given graph $G(V, E)$.

Definition 1 Given a graph $G(V, E)$, $I \subset V, S \subset V$, S is the neighborhood set of I (denote as $N(I)$) if and only if $\forall v_j \in S, v_j \notin I$ and $\exists v_i \in I$ such that $(v_i, v_j) \in E$

Lemma 2 Given graph $G(V, E)$, if $(v_i, v_j) \in E$ is deleted in some step in the Algorithm, (v_i, v_j) does not lie in any NPP from src to dst in the original graph $G(V, E)$.

Proof: (by contradiction) Assume that there is a link (v_i, v_j) between nodes $v_i \notin FS$ and $v_j \in FS$ deleted in some iteration lies in an NPP path $P = (v_1, \dots, v_i, v_j, \dots, v_n)$. First consider that the edge (v_i, v_j) is the first link we delete during the algorithm. Since P is an NPP, we have $\sum_{k=i}^n r_k < c_{i,j}$, i.e. $\sum_{k=j}^n r_k + r_i < c_{i,j}$. Recall that in the algorithm, we check \tilde{u}_{v_i} for trying to label v_i as $\min(l(v_i), \min_{v_j \in FS \wedge (v_i, v_j) \in E} (l(v_j) + r_i))$. And for node v_j , $l(v_j)$ is the minimum summation of r values from node v_j onwards since v_j is in the feasible set. Hence, we have

$$\sum_{k=j}^n r_k + r_i \geq l(v_j) + r_i \geq \min(l(v_i), l(v_j) + r_i)$$

It follows that $\min(l(v_i), l(v_j) + r_i) < c_{i,j}$. Then, according to the algorithm, edge (v_i, v_j) should not be deleted. This contradicts the assumption. Thus edge (v_i, v_j) does not lie in any NPP from src to dst in $G(V, E)$. This argument can now be inductively applied to the second edge that is deleted, the third, and so on, because deleting an edge that does not lie

in any NPP from the graph does not affect the solution to the NPP problem in any way.) \square

Theorem 1 The algorithm to find an NPP path with minimum $\sum r_k$ in the transformed network graph is correct.

Proof: (**Soundness**): the path found by the algorithm in the transformed graph is guaranteed to be an NPP path since it has a check step to make sure each node in the feasible set has a non-positive payoff. The path is guaranteed to have minimum $\sum r_k$ since in the algorithm, we always label smallest feasible $\sum r_k$ first.

(**Completeness**): We need to prove that if there exists an NPP in the graph, the algorithm will return one. According to Lemma 1, since the edge deleted in the algorithm doesn't lie in any NPP, the algorithm doesn't destroy any NPP path in the graph. The algorithm terminates only under two conditions: either it finds the NPP or $N(FS) = \emptyset \wedge src \notin FS$. The latter case indicates that the source and destination are separated into two isolated parts of the graph, which implies that there is no NPP in the original given graph. \square

The computational complexity of the algorithm is polynomial. The Dijkstra's algorithm can be run in time $O(n^2)$. For each edge deletion in our algorithm, we need to retry the labeling, which will cost at most extra $O(n)$ time for each node. So the running time of our algorithm is bounded by $O(n^3)$.

Notice that when mapping the algorithm back to the PPP problem, we always choose the most reliable path among all the feasible paths. In the algorithm we keep adding the nodes with minimum summation of r that still satisfies the negative utility constraints. This observation can be used to prove that path returned by this algorithm is a Nash equilibrium path (if all nodes not on the path choose the null strategy of not picking any next-hop neighbor).

Theorem 2 The path found by the algorithm is a Nash equilibrium path in the PPP finding problem.

Proof (by contradiction): Assume that the algorithm returns a path $P = (v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_j, \dots, v_n)$ which is not a Nash equilibrium. Without loss of generality, suppose only one node v_i wants to switch his next hop from v_{i+1} to v_j , where $j > i + 1$. Path $\hat{P} = (v_0, v_1, \dots, v_i, v_j, \dots, v_n)$ is also a PPP, since the payoff of the nodes before v_j increases by the increase of path reliability (remember $0 \leq R_k \leq 1$) and the payoff after v_j (including v_j) keep unchanged. Thus path \hat{P} is one of the feasible paths. Since the path abandoned some intermediate nodes, the path reliability of \hat{P} is larger than P . This would imply that the algorithm should return path \hat{P} instead of P , which contradicts the assumption. By construction, the node has no incentive to switch its next hop to a node that is not on the returned path since those nodes do not pick any next-hop neighbor. \square

As we mentioned before, the algorithm runs to obtain a positive payoff path to destination from each neighbor of the source node. If the source node is selfish, among all the feasible paths reported from its set of neighbors, it will pick the one that gives its maximum profit according to the source's utility function. If the source node is cooperative, it will pick

the path which gives the highest path reliability.

V. SIMULATION RESULTS

In this section, we present our simulation results. We have two different simulation models that essentially yield different link cost distributions: ARQ-based, and distance-based.

In the ARQ model, we generate the network topology using a realistic link layer model [24]. The link layer model output a directed graph with each edge has its own PRR (packet reception rate). The link cost of edge (v_i, v_j) in our model is calculated as the average of the expected number of transmissions in each direction (assuming ARQ). However, we find that most of the link costs are around 1 in this link layer model.

In the distance-based model, each node has same transmission range, but the link cost is made proportional to the square of the distance between two nodes if they are in each other's transmission range. If the two nodes are out of each other's transmission range, the link cost between these two nodes are set to be infinity. The mathematical representation of the distance-based model is as follows:

$$C_{i,j} = \begin{cases} \alpha \cdot d(i,j)^2 & \text{if } d(i,j) \leq \Gamma \\ \infty & \text{otherwise} \end{cases}$$

where $d(i,j)$ is the distance between node v_i and node v_j ; and Γ is the transmission range of the sensor nodes. In the simulation settings, α is set to 0.1 (we also did extensive simulations for different α values, similar curve trends are observed). The distance-based model shows greater variance in the link costs and thus allows for more tradeoff between link cost and node reliability than in the ARQ model where the link costs are more uniform.

We use a fixed 12×12 square meters area as our simulation area. In the distance-based model, node's transmission range is set to 5 meters. The node reliability is uniformly chosen at random in interval $[0.1, 1]$.

Figure 1 illustrates the path reliability versus source pay for intermediate nodes when fixing G to 300 (a sufficient large amount) for both models. From this figure, we can see that the density of the deployments increases, the maximum reachable path reliability increases. This result is expected. When the source pays more to intermediate nodes, the expected path reliability increase too. We notice in both cases that when p exceeds some threshold the path reliability will remain almost constant. However, the curves for the distance-based model increase a bit more gradually while those for the ARQ-model are sharp — this reflects the greater variance of link costs in the distance-based model.

Figure 2 plots the source gain versus the source pay to the intermediate nodes with fixed number of nodes (30) and area size. Recall that from the source utility function in Section III, source utilities in most cases are dominated by the term of $(G - hp) \prod_{v_i \in Path} R_i$. Increasing p can lead to decreasing of h and increment of $\prod_{v_i \in Path} R_i$. Figure 2 shows that there exists a best strategy point for the source to maximize its payoff, which is at the same routing price no matter how

much destination pay is given in a fixed network topology. The other observation of Figure 2 is that the portion of source gain increases as the destination pay increases. This indicates that even if the destination increase the pay to the source to request a certain reliability path, most of the money goes to the source instead of the routing nodes. It implies that even if the destination increase the pay, it will not get a path with more reliability.

If we examine Figure 1 and Figure 2 together, we will find that at the maximum gain of the source node, the path reliability is close to the maximum path reliability which the network can reach. This gives us an important insight: selfish behavior of source node in such system will not hurt system performance much. Figure 3 shows a side-by-side comparison of source node behaving cooperatively and selfish, for the ARQ model. These figures demonstrate that there is improvement of path reliability when source acts cooperatively, but the improvement is not significant. We also see that the maximum path reliability will not have significant improvement for any fixed network parameter when destination pay exceeds some threshold (around 50 here) that is necessary to obtain a path. On the other hand, the routing path reliability will increase significantly (from 0.39 to 0.74) when changing network parameters (in this particular simulation, we increase the number of nodes in the fixed area).

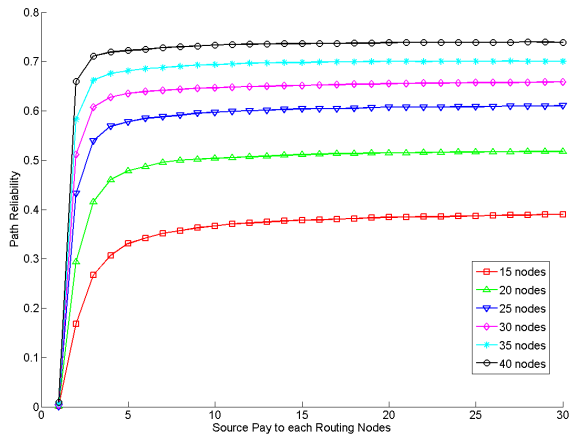
Figure 4 shows the probability that a positive payoff Nash equilibrium path exists as a function of the price offered by the source under both models. For each curve of realistic link layer model, corresponding to a fixed number of nodes (fixed density), we see that the curve increases to a point where it is close to 1. This shows the existence of critical threshold prices (independent of the exact configuration) that ensure the existence of a Nash Equilibrium path with high probability. We also see that this price threshold decreases with the density, a trend that is concrete visualized in the distance-based model which is effected by node distance more seriously. This trend is because with growing density there are more choices to pick the path from, and there are a greater number of high quality links which incur low transmission cost.

VI. CONCLUSION

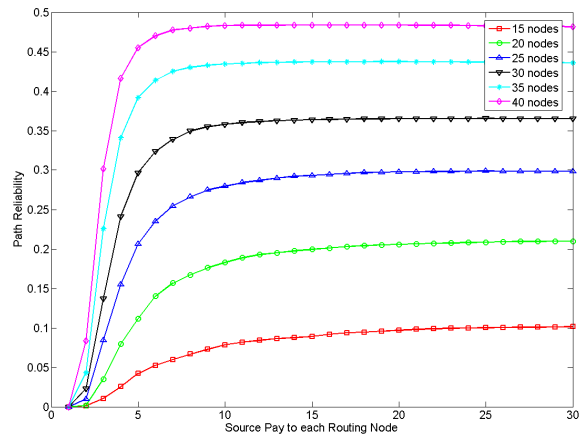
In this paper, we described a destination-driven source-mediated pricing routing scenario involving three different kind of nodes: the destination, the source, and the intermediate nodes. We presented a polynomial time algorithm which can give us a Nash equilibrium path and used it to evaluate the performance of the performance of the game with respect to prices and source behaviors for different network settings.

The simulations results demonstrate several key findings:

- With the increment of the network density, routing paths become cheaper and more reliable and the source payoff will increase.
- Even if the source node acts selfishly, it doesn't necessarily downgrade the reliability of the routing path significantly.

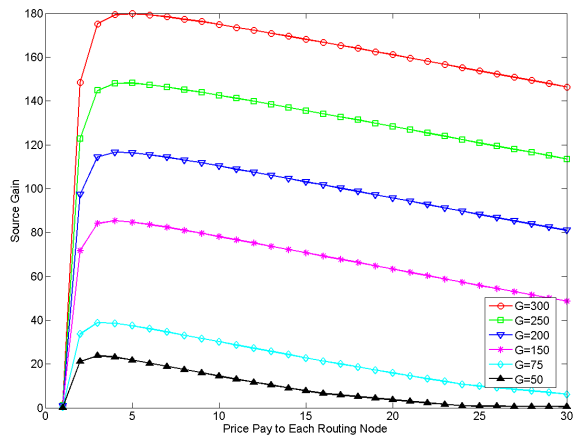


(a)

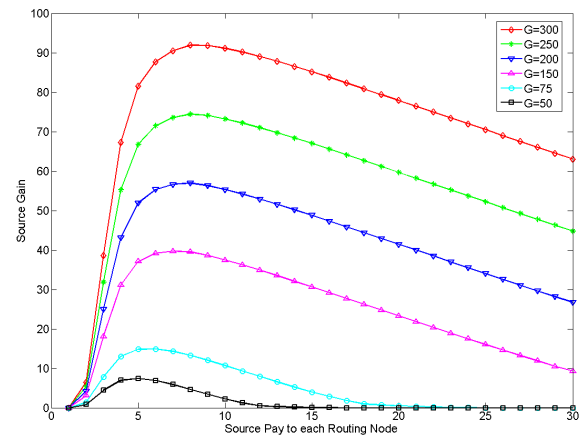


(b)

Fig. 1. Path reliability versus source pay to each routing node when changing number of nodes in a fixed area: (a) ARQ model (b) distance-based model

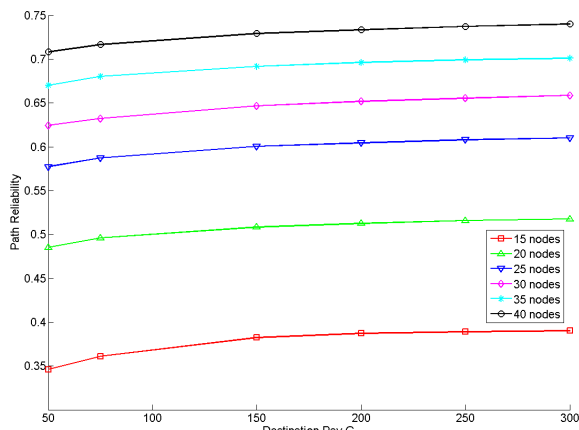


(a)

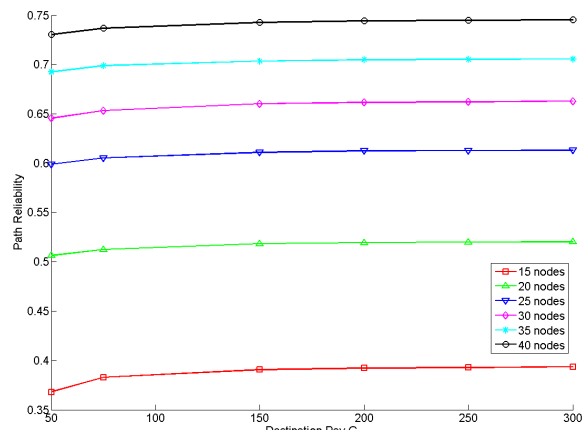


(b)

Fig. 2. Source gain versus source pay to each routing node for different destination pay, when fixing number of nodes and area size: (a) ARQ model (b) distance-based model



(a)



(b)

Fig. 3. Behavior of source node effect on the path reliability: (a) selfish source (b) cooperative source

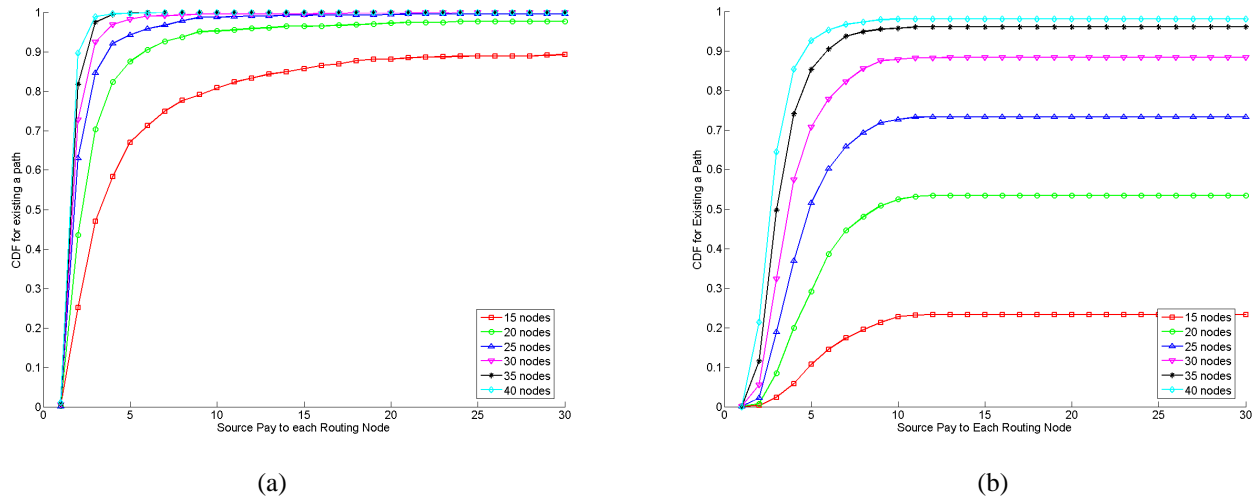


Fig. 4. Cumulative distribution function for the existence of Nash Equilibrium path when increasing source pay to each routing node: (a) ARQ model, (b) distance-based model

- Given a network, the increment of destination pay won't improve path reliability after some threshold. The source will eat up the margins beyond this point. Thus this is the desired point of operation for the destination.

In ongoing work, we have been developing simple mathematical models to analyze the trends demonstrated here via simulations. One direction to extend the game is to add the destination as a player and explicitly incorporate the prices G and p as strategies decided by the source and destination. Another direction that is of interest for future work is to consider scenarios where the destination can choose from several source nodes for a given piece of information. This will allow for an auction to be held among the source nodes to optimize destination's payoff.

VII. ACKNOWLEDGEMENT

We would like to thank members of the USC Autonomous Networks Research Group, anonymous reviewers, and Rajgopal Kannan for useful and constructive feedback. This work has been supported in part by NSF through the CAREER grant CSN-0347621.

REFERENCES

- [1] R. Kannan, S. Sarangi, S.S. Iyengar, "Sensor Centric Quality of Routing in Sensor Networks," IEEE Infocom, 2003.
- [2] R. Kannan, S. Sarangi, S.S. Iyengar, "Game-Theoretic Models for Reliable Path-Length and Energy-Constrained Routing with Data Aggregation in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 6, August 2004.
- [3] R. Kannan, S. Sarangi, S.S. Iyengar, "Sensor-centric energy-constrained reliable query routing for wireless sensor networks", *Journal of Parallel Distributed Computing*, Vol. 64, pp. 839-852, 2004.
- [4] R. Kannan, S. Sarangi, S.S. Iyengar, "A simple model for reliable query reporting in sensor networks Information Fusion," *Proceedings of the Fifth International Conference*, July 2002.
- [5] Zong, Chen, Yang, Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks", IEEE Infocom 2003.
- [6] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, MobiCom 2000*.
- [7] S. Buchegger, J. Y. Le Boudec, Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks, in 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, January 2002.
- [8] S. Buchegger and J. Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002
- [9] S. Bansal, M. Baker, "Observation-based Cooperation Enforcement in Ad Hoc Networks," Technical Report, cs/0307012, arxiv.org.
- [10] A. Urpi, M. Bonuccelli, S. Giordano, "Modelling Cooperation in Mobile Ad Hoc Networks: a Formal Description of Selfishness", *WiOpt Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks INRIA Sophia-Antipolis, France, Mar. 2003*.
- [11] L. DaSilva and V. Srivastava, "Node Participation in Ad Hoc and Peer-to-Peer Networks: A Game-Theoretic Formulation", *Workshop on Games and Emergent Behavior in Distributed Computing*, 2004.
- [12] M. Fleglyzhi, L. Buttyan, and J.-P. Hubaux, Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks – the Static Case, 8th International Conference on Personal Wireless Communications (PWC 2003), Venice, Italy, 23-25 September, 2003.
- [13] M. Fleglyzhi, J. Hubaux, L. Buttyan, "Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol 5, No 5, pp 463-576, May 2006.
- [14] E. Altman, A. A. Kherani, P. Michiardi, and R. Molva. "Non-cooperative Forwarding in Ad-hoc Networks", Technical Report INRIA Report No. RR-5116, 2004.
- [15] O. Ileri, S. Mau, N. Mandayam, "Pricing for Enabling Forwarding in Self-Configuring Ad Hoc Networks", In *IEEE J. Sel. Areas Commun. Special Issue on Wireless Ad Hoc Networks*, Jan. 2005
- [16] L. Buttyan and J. P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *ACM Journal for Mobile Networks (MONET)*, Vol. 8, No. 5, 2003.
- [17] P. Nurmi, "Modelling routing in wireless ad hoc networks with dynamic Bayesian games", 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON) Santa Clara, Calif., Oct 2004
- [18] V. Srinivasan and P. Nuggehalli and C. Chiasserini and R. Rao, "Cooperation in wireless ad hoc networks", In *Proceedings of IEEE Infocom 2003*.
- [19] L. Anderegg, S. Eidenbenz, "Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents", *Pro-*

ceedings of the 9th annual international conference on Mobile computing and networking, San Diego, 2003

- [20] M. Jakobsson, J. Hubaux, and L. Buttyan, "A micropayment scheme encouraging collaboration in multi-hop cellular networks", in Proceedings of Financial Crypto 2003
- [21] J. Crowcroft, R. Gibbens, F. Kelly and S. Ostring, "Modelling incentives for collaboration in mobile ad hoc networks", Performance Evaluation 57 (2004) 427-439.
- [22] P. Michiardi and R. Molva, "CORE: A Collaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks", IFIP - Communication and Multimedia Security Conference 2002
- [23] N. Salem, L. Buttyan, J. Hubaux, M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks", In Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking and Computing ACM MobiHoc, New York, 2003
- [24] M. Zuniga, B. Krishnamachari, Realistic Wireless Link Quality Model and Generator (to generate realistic static wireless ad-hoc/sensor network scenarios as graphs with links labeled with packet reception rates) [Version 1.1], December 2005. Available online at <http://ceng.usc.edu/~anrg/downloads.html>.