Incentive-Compatible Opportunistic Routing for Wireless Networks

Fan Wu[†] Tingting Chen[†] Sheng Zhong[†]
Li Erran Li[⋆] Yang Richard Yang[§]
[†]Department of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY 14260
[⋆]Networking Research Lab, Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974
[§]Department of Computer Science, Yale University, New Haven, CT 06520
{fwu2,tchen9,szhong}@cse.buffalo.edu
erranlli@research.bell-labs.com yry@cs.yale.edu

ABSTRACT

User-contributed wireless mesh networks are a disruptive technology that may fundamentally change the economics of edge network access and bring the benefits of a computer network infrastructure to local communities at low cost, anywhere in the world. To achieve high throughput despite highly unpredictable and lossy wireless channels, it is essential that such networks take advantage of transmission opportunities wherever they emerge. However, as opportunistic routing departs from the traditional but less effective deterministic, shortest-path based routing, user nodes in such networks may have less incentive to follow protocols and contribute. In this paper, we present the first routing protocols in which it is incentive-compatible for each user node to honestly participate in the routing despite opportunistic transmissions. We not only rigorously prove the properties of our protocols but also thoroughly evaluate a complete implementation of our protocols. Experiments show that there is a 5.8%-58.0% gain in throughput when compared with an opportunistic routing protocol that does not provide incentives and users can act selfishly.

Categories and Subject Descriptors: C.2.1 [Computer Communication Networks]: Network Architecture and Design – *Network communications*; C.2.3 [Computer Communication Networks]: Network Operations – *Network Management*

General Terms: Design, Economics, Management

Keywords: Wireless, Incentive

1. INTRODUCTION

User-contributed wireless mesh networks are envisioned as a disruptive technology to deploy broadband network infrastructures to local communities at low cost [2,4,30,32]. Avoiding the problem of rights of installation by installing at private properties and distributing the equipment and management costs among the participating users, such networks present a major cost-effective alternative for overcoming the issues that are stalling the deployment of carrier-deployed wireless mesh networks.

However, the deployment of user-contributed wireless mesh networks has its own challenges. A major challenge, which is not limited only to user-contributed wireless mesh networks but applies to wireless mesh networks in general, is throughput scalability. Due to the highly unpredictable and lossy wireless channels, the throughput achieved by traditional wireless mesh networks can be quite poor. Wireless channel variations and losses are particularly serious in urban environments with many sources of interference [2, 15, 24]. Although users of wireless networks may not demand the same level of available bandwidth as wired networks, it is important that such networks provide sufficient throughput for adaptive, elastic applications to give acceptable user experiences.

To achieve high throughput despite the highly unpredictable and lossy wireless channels, it is essential that such networks take advantage of multi-user diversity and explore transmission opportunities wherever they emerge [37]. Thus, instead of deterministically choosing the next hop before transmitting a data packet, opportunistic routing is emerging as a novel technique to allow any nodes overhearing the packet to participate in forwarding it. In particular, Biswas and Morris [6] introduce the novel ExOR protocol and show that through test-bed experiments, by asking network nodes to opportunistically forward received data packets, they can achieve superior performance than the traditional deterministic forwarding. In [13], Chachulski et al. introduce the MORE opportunistic routing protocol to address issues in ExOR and achieve higher throughput in wireless networks.

The adoption of such opportunistic routing protocols, however, might lead to reduced network throughput when nodes have selfish behavior. In particular, user-contributed wireless mesh networks, like many distributed autonomous systems, suffer common incentive problems such as the free-rider problem, where only a small fraction of user nodes contribute their resources [1], or the adverse selection problem, where user nodes do not reveal truthfully their channel states [7].

Although much progress has been made in designing incentive mechanisms for wireless networks [11], opportunistic routing represents a major departure from the traditional shortest-path based routing paradigm. There can be serious incentive issues in opportunistic routing (see Section 2.2). However, existing incentive mechanisms are mainly based on the simplicity of shortest path routing, and thus no longer fully apply to opportunistic routing.

In this paper, we present the first routing system in which each user is stimulated to honestly participate in network routing despite opportunistic transmissions. We make the following contributions:

- We are the *first* to study the problem of incentives in opportunistic routing and provide solutions.
- We present a simple, novel and practical technique to make *any* member of a class of opportunistic routing protocols

MobiCom'08, September 14–19, 2008, San Francisco, California, USA.
Copyright 2008 ACM 978-1-60558-096-8/08/09 ...$5.00.

incentive compatible. Specifically, this class includes any opportunistic routing protocols that use loss probabilities to calculate the number of forwarding transmissions to impose structure and avoid the scalability issues of opportunistic transmissions. As a comparison, previous incentive-compatible routing mechanisms (e.g., [5, 40]) are typically based on the Vickrey-Clarke-Grove mechanism and thus require that a routing protocol solve the routing problem optimally. They cannot be applied to any theoretically suboptimal, or heuristic-based routing protocols (like MORE). The optimality condition imposes an onerous burden on the designers of opportunistic routing and thus limits their ability to design practical protocols. Unlike the VCG-based techniques, our technique does not impose any such mainly theoretical conditions for optimality.

- We rigorously prove that our technique guarantees that it is a *strict dominant strategy* for each user node to behave honestly. Here strict dominant strategy is a very strong solution concept in game theory. Intuitively, it means that the strategy (of behaving honestly) is *strictly better* than any other strategy for each node regardless of other nodes' behavior.
- We also design an enhanced protocol to prevent cheating not only in reporting loss rates but also in measuring them. Formally, we show that, with this enhanced protocol, it is a *strict Nash equilibrium* for each user node to behave honestly in both measuring and reporting. Intuitively, this means that the strategy of behaving honestly is strictly better than any other strategy for each node when other nodes are honest.
- We completely implement our protocols in Linux and test their performance on the ORBIT lab [33]. The experimental results verify that, with our protocols, a selfish node's cheating behavior decreases its utility. Consequently, there are incentives for nodes to follow our protocols. Our experiments also show that compared with an opportunistic routing protocol that does not provide incentives, our protocols have a throughput gain of 5.8%-58.0%. This is because our protocols can prevent cheating behavior by selfish nodes. Hence they can bring the system throughput back to the high level achieved by opportunistic routing.

The rest of this paper is organized as follows. In Section 2, we present technical preliminaries. In Section 3, we present a simple technique to achieve incentive compatibility in reporting loss probabilities. In Section 4, we develop our technique to prevent cheating in both measuring and reporting loss probabilities. In Section 5, we discuss implementation issues. In Section 6, we report experimental results using the ORBIT testbed. In Section 7, we review related work. In Section 8, we draw conclusions and discuss future work.

2. PRELIMINARIES

Before presenting our system architecture and developing our protocols, we first review the opportunistic routing protocols we consider. We give a simple example to illustrate that nodes have incentives to cheat MORE, an opportunistic routing protocol. We also review relevant game theoretic definitions.

2.1 Basic Opportunistic Routing Protocols

We focus on opportunistic routing, which is an emerging technique to achieve high throughput with lossy wireless links. Instead of choosing the next hop before transmitting a packet, opportunistic routing allows multiple nodes that overhear the transmission to participate in forwarding the packet.

The key issues in the design of opportunistic routing protocols are how to avoid duplicate forwarding, achieve high spatial reuse, and be scalable. Different opportunistic routing protocols (e.g., [6, 13]) solve these issues differently. Since researchers are still trying to improve opportunistic routing and different networks may make different tradeoffs, we set the goal that we develop techniques that can be integrated with a wide class of opportunistic routing protocols. We consider an opportunistic routing protocol as a module which collects link states and computes a forwarding behavior profile for each node i. This modular approach reduces the constraints on designing practical protocols, in particular for opportunistic routing.

Specifically, for ease of presentation, we focus on the class of opportunistic routing protocols whose input is link loss probabilities. Let (i, j) be the link from node i to node j. Then let $\epsilon_{i,j}$ be the link loss probability; that is, if a packet is sent from node i to node j, then with probability $\epsilon_{i,j}$ the packet cannot be decoded. One certainly can extend the input to include link rates. But for simplicity, we focus on the case of fixed link rates. For the forwarding behavior profile, we focus on the case that it specifies the number of times a node forwards a received packet. This can depend on the packet header information such as source address or destination address. We refer to an opportunistic routing protocol satisfying the aforementioned specification a *basic opportunistic routing protocol*. Our objective is to extend an *arbitrary* member of this protocol class to be incentive compatible.

Formally, a basic opportunistic routing protocol (*e.g.*, MORE) works as follows.

Source Node: The source node of a session divides its traffic into a number of batches, where each batch consists of a number of coded packets. These coded packets are computed from the original packets using certain network encoding techniques. Each coded packet has a packet header containing sufficient information for routing. The source node keeps sending coded packets in a batch. It stops the transmission of a batch if the batch is acknowledged by the destination.

Intermediate Node: When an intermediate node hears a packet, the contents of this packet (including the header) decide whether this intermediate node needs to forward the packet. If it needs, then it computes the number of transmissions it needs to make using the loss probabilities. Formally, let z_i be the number of transmissions that node i should make for this packet when the basic routing protocol is used. Let E be the set of edges that are considered by the basic routing protocol for forwarding packets from S to D. Then, the basic routing protocol specifies a function $f()$ which computes

$$z_i = f(S, D, i, \{(i, j, \epsilon_{i,j}) | (i, j) \in E\}),$$

where (S, D) indicates the source and destination.

The preceding computation depends on the loss probabilities of the links in E. Previous protocols assume that this knowledge can be obtained by having each node i report the loss probabilities of its links in E_i, where E_i is a subset of E. We should have that $E = \cup_{i \in V_P} E_i$, where V_P is the set of players, and for all (i, j) such that $i \neq j$, $E_i \cap E_j = \phi$.

Specifically, in MORE, z_i is computed by a distributed algorithm using ETX as the metric [14]. When a transmission is triggered, the node creates a random linear combination of the innovative coded packets that it has heard from the same batch and broadcasts it.

Destination Node: The destination uses the contents of its received packets to decide whether it has sufficient information for decoding. If so, it decodes the packets in this batch and sends an acknowledgment using a traditional routing protocol.

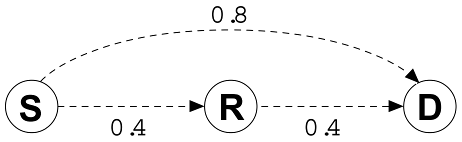

Figure 1: An example scenario to illustrate adverse selection in basic opportunistic routing. There is a session from source S to destination D, with an intermediate node R. True link loss probabilities are shown near the links.

2.2 Example: Adverse Selection in Basic Opportunistic Routing

The basic opportunistic routing protocols assume that user nodes are obedient and follow the protocols. However, a user node may deviate from the specified protocols. In particular, user nodes may not report loss probabilities $\{(i, j, \epsilon_{i,j})|(i, j) \in E\}$ truthfully, because reporting non-truthful loss probabilities may lead to routing decisions $\{z_i\}_{i \in V_P}$ that are more favorable to some nodes.

Consider the MORE protocol, which is one instance of the class of basic opportunistic routing protocols. We use an example scenario shown in Figure 1 to illustrate that a node can benefit from adverse selection in the MORE protocol. The example scenario has a session from source S to destination D. An intermediate node R is between S and D. True link loss probabilities are shown near the links. Using the truthful link loss probabilities, MORE will calculate that R's expected number of transmissions is 1.18. However, by cheating MORE to use 0.1 as the loss probability on link (R, D), node R can reduce its expected number of transmissions to 0.78, a reduction of expected workload by 34%. Consequently, R has incentive to cheat, but this cheating may lead to reduced throughput for the session from S to D.

2.3 Solution Concepts

To study the incentive compatibility of opportunistic routing, we use a *strategic game model*. The players of this game are the intermediate nodes that are required to forward packets. Recall that V_P denotes the set of players. Each player node $i \in V_P$ can choose an action a_i in this game. If every player has chosen an action, then the utility of player i is a function of the profile of all players' actions:

$$u_i = u_i((a_j)_{j \in V_P}).$$

Note that a key component of the definition of a strategic game is the set of potential actions of each player. In this paper, in Sections 3 and 4, we have different sets of available actions and different functions for calculating the utility (because in Section 3 we make a simplifying assumption which we remove in Section 4). We specify the corresponding action sets and functions in detail in these two sections.

Also note that, in reality, a player i can take not only a fixed action, but also a random action following a certain probability distribution. In the latter case, the random action is called a *mixed strategy*. In contrast, a fixed action is also called a *pure strategy*. Suppose that each player i takes a strategy s_i, which can be either pure or mixed. We can always write the utility as a function of the profile of all players' strategies:

$$u_i = u_i((s_j)_{j \in V_P}).$$

As we have mentioned, one solution concept we use in this paper is *strict dominant strategy*. It can be defined as follows.

DEFINITION 1. *A profile s^* of all players' strategies is a strict dominant strategy equilibrium if for all $i \in V_P$, for all strategy $s_i \neq s_i^*$ of player i, for all profile s_{-i} of all other players' strategies,*

$$u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}).$$

Remark Since an action profile is also a strategy profile, it can also be a strict dominant strategy equilibrium.

It is worth noting that strict dominant strategy equilibrium is a very strong solution concept. It requires the equilibrium strategy to be strictly better than any other choice in all situations. Hence it is stronger than a widely used solution concept—dominant strategy equilibrium. Compared with a dominant strategy equilibrium, a strict dominant strategy equilibrium gives an economically rational player an even strong attraction to follow the equilibrium strategy. Specifically, if all nodes following a protocol is a dominant strategy equilibrium, then a node may be able to deviate from the protocol *without being punished*, although the deviation is *not beneficial* to the node. However, if all nodes following a protocol is a strictly dominant strategy equilibrium, then any node deviating from the protocol is punished for its deviation. So, the advantage of using strict dominant strategy equilibrium is that, for each node, deviating from the protocol *always hurts*.

We also use another solution concept called *strict Nash equilibrium*.

DEFINITION 2. *A profile s^* of all players' strategies is a strict Nash equilibrium if for all $i \in V_P$, for all strategy $s_i \neq s_i^*$ of player i,*

$$u_i(s_i^*, s_{-i}^*) > u_i(s_i, s_{-i}^*).$$

The relationship among the involved solution concepts is shown in Figure 2. We can see that a strict Nash equilibrium (which is achieved by our second protocol—the enhanced protocol) is not as strong as a strict dominant strategy equilibrium (which is achieved by our first protocol—the simple extension of basic routing protocol). The reason is that, in the enhanced protocol, we need to deal with more sophisticated cheating behavior.

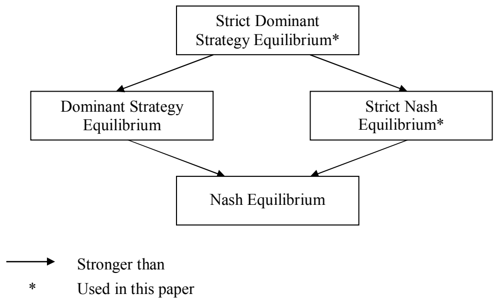

Figure 2: Relationship Among Involved Solution Concepts.

Nevertheless, a strict Nash equilibrium is still stronger than a Nash equilibrium. The difference between these two solution concepts is analogous to the difference between a strict dominant strategy equilibrium and a dominant strategy equilibrium. The relationship between strict Nash equilibrium and dominant strategy equilibrium is more complicated. On one hand, dominant strategy equilibrium requires the consideration of more situations than strict Nash equilibrium, which requires the consideration of only a single situation (where all other players follow the equilibrium strategies). On the other hand, strict Nash equilibrium requires a strict advantage of following the equilibrium strategy, which dominant strategy equilibrium does not. Hence, neither of these two solution concepts is stronger than the other.

3. MOTIVATING HONEST REPORTING OF LOSS PROBABILITIES

Now we present our techniques to integrate incentive compatibility into a given basic opportunistic routing protocol. To make the presentation clearer, we present our core ideas in two steps. In this section, we assume that each node i can measure the loss probabilities of all of its outgoing links, i.e., $\epsilon_{i,j}$ for all $(i,j) \in E$. This may require a slight modification to some basic opportunistic routing protocols (e.g., MORE), if they require that each node report the loss probabilities of its incoming links. We address the measurement issues in the next section. Implementation issues will be discussed in Section 5.

3.1 Overview

If a routing system does not build in proper incentives, a user node may not report its link loss probabilities honestly. Instead, it may compute or conduct probing experiments to determine reporting loss probabilities that can lead to more favorable routing decisions to the node than the true loss probabilities can. Such behaviors can lead to network performance degradation and disruptions.

To prevent dishonest and/or probing behaviors, we need to design the routing protocol so that reporting true loss probabilities is the best strategy of each node. For this objective, we introduce a novel, well-designed payment formula together with an auxiliary transmission. Specifically, in our technique, each intermediate node receives a payment for its service. By *payment* we mean either payment of real money *or transfer of credits*. It depends on the application which of these two methods should be used. In a community network where nodes are voluntarily provided by users, transfer of credits having no monetary values may be a better method. In a more business-oriented environment, some of the users may need to be paid by virtual money that has a cash value. In the sequel, we use the term *payment* only for simplicity of presentation.

Specifically, the payment to an intermediate node consists of two parts: one part is used to cover the data transmissions, while the other part is used to cover auxiliary transmissions that the node is required to make in addition to data transmissions.

What is an auxiliary transmission? It is a transmission that can be used for multiple purposes. For example, it can be used for sending checksums of the data, so that the data transmissions are more reliable. Or it can be used for various control information (like updates of loss probabilities). Note that, in our technique, the volume of auxiliary traffic is very small compared with that of data traffic. Hence, it does not introduce much overhead in communications.

The part of payment for auxiliary transmissions and the volume of auxiliary traffic are carefully designed such that the following requirements are satisfied:

- Both of them are very small (see Sec. 6.4 and Sec. 6.5).
- This part of payment is at least sufficient to cover the cost of auxiliary transmissions.
- A node's gain from the auxiliary transmissions (i.e., the payment it receives for this purpose minus its cost for this purpose) is maximized when it faithfully reports the loss probabilities of its outgoing links.

Therefore, with the payment and auxiliary transmission, a node has incentives to report the loss probabilities faithfully.

3.2 The Protocol

Using the preceding technique, we design an extension of the basic opportunistic routing protocol. In the extended protocol, there are a number of control messages. These control messages should be sent and received reliably using a traditional routing protocol. The cost of control messages should be small such that it can be ignored in the analysis of incentive compatibility —- this is a standard assumption in related literature (e.g., [40]).

Hereafter, we assume that each data packet has a size of L. (It is straightforward to further extend our work to the case in which different packets has different sizes. We ignore this possibility here for simplicity of presentation.) Also, we assume that transmitting a packet of size 1 has one unit of cost.

Now we summarize our protocol. In the protocol description below, we assume there is a Routing Decision Maker (RDM) who collects information of loss probabilities and computes the number of data/auxiliary transmissions and the amount of payment. We have a detailed discussion of RDM in Section 5.

Computing and Sending Routing Decision: In this technique, suppose that the routing protocol receives loss probability $\epsilon'_{i,j}$ for link (i,j) from node i. The RDM computes, for each $(i,j) \in E$,

$$z'_i = f(S, D, i, \{(i, j, \epsilon'_{i,j}) | (i,j) \in E\}),$$

$$z^{\star}_{i,j} = \frac{\alpha(1-\epsilon'_{i,j})^2}{2},$$

and

$$p_i = z'_i L + \sum_{(i,j)\in E} \alpha(1-\epsilon'_{i,j}),$$

where $\alpha > 0$ is a parameter chosen by the system administrator, L is the packet length. The value of α is very small [1]. The output of the routing protocol for each node i, is $(z'_i, \{z^{\star}_{i,j}\}_{(i,j)\in E}, p_i)$.

Making Transmissions and Receiving Payments: Each node i makes z'_i regular data transmissions for each packet it should forward. In addition, for each such packet it is required to send an auxiliary traffic of size $z^{\star}_{i,j}$ to node j, *such that node j receives the auxiliary traffic*. That is, node i has to repeatedly send this auxiliary traffic until it is received by node j. Note that, since $z^{\star}_{i,j}$ is very small, node i does not need to send a packet immediately. In stead, it can accumulate the auxiliary traffic of the entire batch and send them together, when the network is not busy.

As a reward for its service in sending a packet, node i should receive a payment p_i from the source node. Again, this payment is not immediate made to node i. In stead, it is accumulated until the session finishes. At the end of the session, node i receives the total payment in the entire session. This needs only a single control message (which is transmitted reliably using traditional routing).

For convenience of the reader, we have summarized the important symbols we use in Table 1.

3.3 Analysis

We can analyze the above extended protocol using the strategic game model we present in Section 2.

When we analyze this protocol, the action set available to each player node i is $[0,1]^{|\{j|(i,j)\in E\}|}$. Intuitively, this means that the

[1] α is a constant to adjust the payment so that the nodes get the right incentives, but do not need to pay much more than what are needed for their data transmissions. In practice, if nodes can tolerate a small amount of utility loss in certain applications, then we should increase the value of α correspondingly. This also applies to the parameter β we define in Sec 4.2.

Table 1: Important Variables. $\star$ indicates the variable is under control of the (potentially selfish) nodes; and $\triangleright$ indicates the variable is computed by the RDM.

	V_p	set of players
	S	source node
	D	destination node
	E	set of edges in consider
	ϵ	link loss probability matrix
$\star$	ϵ' (ϵ'')	claimed link loss probability matrix
	$f()$	function to compute the number of transmissions
$\triangleright$	z	vector of number of transmissions for ϵ
$\triangleright$	z' (z'')	vector of number of transmissions for $\epsilon'(\epsilon'')$
$\triangleright$	p $(\hat{p})$	payment vector of players
	u	utility vector of players
$\triangleright$	$z^\star_{i,j}$ $(\hat{z}^\star_{i,j})$	auxiliary transmission matrix
	L	packet length

action of player i specifies a loss probability for each outgoing link (i, j). If a player node i deviates from the protocol, it changes one or more of these loss probabilities. Clearly, this reflects the fact that, in this section, we only consider cheating in reporting loss probabilities.

Correspondingly, the utility of player node i is defined as the payment it receives minus the cost of transmissions it makes. Here the cost transmissions includes the cost of data transmissions and the cost of auxiliary transmissions.

THEOREM 3. *Suppose that each player node i makes z'_i data transmissions and receives a payment p_i. Suppose that for each $(i, j) \in E$, node j receives an auxiliary data traffic of size $z^\star_{i,j}$ from node i. Then it is a strict dominant strategy equilibrium for all player nodes to truthfully report loss probabilities.*

PROOF. Consider each $(i, j) \in E$. To make sure a packet is received by node j, the expected number of transmissions node i needs to make is $\frac{1}{1-\epsilon_{i,j}}$. Therefore, the cost of auxiliary traffic $z^\star_{i,j}$ from i to j is $\frac{z^\star_{i,j}}{1-\epsilon_{i,j}}$.

Hence, assuming s^*_i is the pure strategy of following the protocol faithfully, it is easy to see that, when node i uses strategy s^*_i, the expected utility is

$$u_i(s^*_i, s_{-i}) = \frac{\alpha}{2} \sum_{(i,j)\in E} \frac{(1-\epsilon_{i,j})^2}{(1-\epsilon_{i,j})}.$$

In contrast, consider the situation in which node i uses a strategy $s_i \neq s^*_i$, which assigns probability P_1 to action $a^{(1)}_i$, ..., probability P_K to action $a^{(K)}_i$ $(P_1, \dots, P_K > 0;\ \sum_{k=1}^{K} P_k = 1)$. There must be an action $a^{(k_0)}_i = \{\epsilon^{(k_0)}_{i,j}\}_{(i,j)\in E}$ and j_0 such that $\epsilon^{(k_0)}_{i,j_0} \neq \epsilon_{i,j}$.

Then the expected utility of node i when it takes action $a^{(k)}_i = \{\epsilon^{(k)}_{i,j}\}_{(i,j)\in E}$:

$$\begin{aligned}
&u_i(a^{(k)}_i, s_{-i})\\
=&p_i - \left(z'_i L + \sum_{(i,j)\in E} \frac{z^\star_{i,j}}{1-\epsilon_{i,j}}\right)\\
=&\left(z'_i L + \sum_{(i,j)\in E} \alpha(1-\epsilon^{(k)}_{i,j})\right)\\
&- \left(z'_i L + \sum_{(i,j)\in E} \frac{\alpha(1-\epsilon^{(k)}_{i,j})^2}{2(1-\epsilon_{i,j})}\right)\\
=&\frac{-\alpha}{2} \sum_{(i,j)\in E} (((1-\epsilon^{(k)}_{i,j}) - (1-\epsilon_{i,j}))^2\\
&- (1-\epsilon_{i,j})^2)/(1-\epsilon_{i,j})\\
\leq& u_i(s^*_i, s_{-i}).
\end{aligned}$$

For $a^{(k_0)}_i$, since $\epsilon^{(k_0)}_{i,j_0} \neq \epsilon_{i,j}$, we have

$$u_i(a^{(k_0)}_i, s_{-i}) < u_i(s^*_i, s_{-i}).$$

Therefore,

$$\begin{aligned}
u_i(s_i, s_{-i}) &= \sum_{k=1}^{K} P_k u_i(a^{(k)}_i, s_{-i})\\
&< u_i(s^*_i, s_{-i}).
\end{aligned}$$

This means s^* is a strict dominant strategy equilibrium. □

In the above, we have shown that, with our simple extension, it is a strict dominant strategy for each node to behave honestly. As we have emphasized, strict dominant strategy equilibrium is a very strong solution concept. By its definition, in a system there *cannot* be more than one strict dominant strategy equilibria. Consequently, there is a very strong guarantee that the system should converge to the state in which all nodes follow the protocol.

4. PREVENTING CHEATING IN MEASURING AND REPORTING LOSS PROBABILITIES

The preceding section presents a simple and effective technique to motivate each node to honestly report its loss probabilities. The assumption is that a node can determine the true loss probabilities of its links by itself. However, a node needs the cooperation (feedback) of its neighbors to measure link loss probabilities This may lead to cheating behaviors. Note that this problem caused by neighbor feedback may look on the surface similar to the problem of mutual-dependent type in [40]. However, the issues caused by loss probabilities are more challenging than the simpler power control. Thus, additional mechanisms (e.g., a more sophisticated payment formula) are needed to prevent cheating. In this section, we design techniques involving both measurement signals and payment to prevent cheating. The result is an enhanced protocol for incentive-compatible opportunistic routing.

4.1 Overview

Recall that, in Section 3 we use an appropriately designed payment formula, together with an auxiliary transmission to effectively prevent cheating in reporting the loss probabilities of outgoing links. In our enhanced protocol, suppose that we still have each intermediate node report the loss probabilities of its outgoing links. Then, we do not need to worry about each node's cheating in *measuring* the loss probabilities of its outgoing links, because such cheating in measurement is equivalent to cheating in reporting the measured probabilities, which has been prevented. Hence, the major technical challenge to our enhancement is to prevent nodes' cheating in measuring the loss probabilities of their incoming links.

If node i wants to cheat in measuring the loss probability of incoming link (j,i), there are two possibilities for its cheating behavior: Either by cheating in this measurement it makes the measured loss probability larger than the real loss probability, or it makes the measured loss probability smaller. To prevent the first type of cheating, we only need to give node i a small amount of payment, which decreases with the loss probability of link (j,i). To maximize this payment, node i has incentives to keep the loss probability as low as possible. Thus, node i does not have incentives to carry out the first type of cheating.

To prevent the second type of cheating, we introduce a special method to measure the loss probability of link (j,i), which ensures that node i cannot decrease the measured loss probability. This method requires node j to send a number of test signals. To report the loss probability of link (j,i), node i does not directly compute the loss probability and send it to the source node. In stead, node i should just forward the test signals it hears to the source, as its "report" for loss probability of this link. The source node computes the loss probability using a number of these test signals. We use a simple Message Authentication Code function to prevent node i from forging these test signals. Hence, node i either forwards all packets it hears, or forwards part of them. There is no way for node i to forward more packets than it actually hears. That is to say, there is no way for node i to decrease the loss probability of link (j,i).

4.2 The Enhanced Protocol

Below we give a more detailed description of our enhanced protocol. Just as the simple protocol in Sec. 3, it still has a RDM which collects information and performs computation. But unlike the simple protocol, it uses the newly introduced method to measure the loss probabilities.

Sending and Forwarding Test Signals: When there is a request to initialize a session from source node S to destination D, each node $i \in \{S\} \cup V_P$ sends n_t test signals. Here each test signal is of the format $(\mathsf{TEST}, i, j, \mathsf{MAC}_{k_{S,i}}(\mathsf{TEST}, i, j))$, where $k_{S,i}$ is a secret key shared by S and i, and MAC is a cryptographic Message Authentication Code function. Each node $i \in V_P \cup \{D\}$ forwards received test signals to the RDM using traditional routing protocol.

Computing and Sending Routing Decision: Suppose that the RDM collects $n_{i,j}$ test signals for link (i,j), which are forwarded by node j. Then the RDM computes

$$\epsilon''_{i,j} = 1 - \frac{n_{i,j}}{n_t},$$

$$z''_i = f(S, D, i, \{(i, j, \epsilon''_{i,j}) | (i,j) \in E\}),$$

$$\hat{z}^{\star}_{i,j} = \frac{\alpha(1-\epsilon''_{i,j})^2}{2},$$

and

$$\hat{p}_i = z''_i L + \sum_{(i,j)\in E} \alpha(1-\epsilon''_{i,j}) + \sum_{(j,i)\in E} \beta(1-\epsilon''_{j,i}),$$

where $\beta > 0$ is a new parameter chosen by the system administrator. Just as for α, the value of β should also be very small.

Making Transmissions and Receiving Payments: Each node i makes z''_i regular data transmissions for each packet it should forward. In addition, for each such packet it is required to send an auxiliary traffic of size $\hat{z}^{\star}_{i,j}$ to node j, such that node j receives the auxiliary traffic. As a reward for its service in sending a packet, node i should receive a payment $\hat{p}_i$ from the source node.

Each node i accumulates the auxiliary traffic for the packets of a batch, and send them together after successful delivery of the whole batch. The payments due to node i are accumulated and made together at the end of the session.

4.3 Analysis

We can analyze the above enhanced protocol using the strategic game model we present in Section 2.

Denote by $\mathcal{R}$ the set of non-negative real numbers. When we analyze this protocol, the action set available to each player node i is $\mathcal{R} \times [0,1]^{|\{j|(j,i)\in E\}|}$. Intuitively, if node i takes action $(\gamma_i,\ \{\gamma_{j,i}\}_{(j,i)\in E})$, then node sends $\gamma_i n_t$ test signals for measuring loss probabilities, and forwards $\gamma_{j,i} n_{j,i}$ test signals to the source S when it receives $n_{j,i}$ test signals for link (j,i).

Clearly, if node i is honest, it should use a pure strategy s^*_i that chooses $\gamma_i = 1$ and $\gamma_{j,i} = 1$ for all $(j,i) \in E$. If a player node i deviates from the protocol, it either chooses $\gamma_i \neq 1$ (which means it changes the number of test signals to send for measuring loss probabilities), or chooses $\gamma_{j,i} < 1$ for one or more incoming links (j,i) (which means it does not forward all test signals it receives for these incoming link), or does both. Compared with the analysis in Section 3, now a cheating node can have more complicated behavior. This reflects the fact that, in this section, we consider cheating in both measuring and reporting loss probabilities.

The utility of player node i is still defined as the payment it receives minus the cost of transmissions it makes.

THEOREM 4. *Suppose that each node i makes z''_i data transmissions and receives payment $\hat{p}_i$. Suppose that for each $(i,j) \in E$, node j receives an auxiliary data traffic of size $\hat{z}^{\star}_{i,j}$ from node i. Then it is a strict Nash equilibrium for all player nodes to behave honestly in sending test signals and forwarding the received test signals.*

PROOF. Recall that s^* is the pure strategy profile of all player nodes in which each node behave honestly. Hence, s^*_{-i} is a strategy profile of all player nodes other than i such that all other nodes behave honest. In general, if node i takes action a_i and other nodes behave honestly, the expected utility of node i is

$$\begin{aligned}
& u_i(a_i, s^*_{-i}) \\
=& \hat{p}_i - \left(z''_i L + \sum_{(i,j)\in E} \frac{\hat{z}^{\star}_{i,j}}{1-\epsilon_{i,j}} \right) \\
=& \left(z''_i L + \sum_{(i,j)\in E} \alpha(1-\epsilon''_{i,j}) + \sum_{(j,i)\in E} \beta(1-\epsilon''_{j,i}) \right) \\
& - \left(z''_i L + \sum_{(i,j)\in E} \frac{\alpha(1-\epsilon''_{i,j})^2}{2(1-\epsilon_{i,j})} \right).
\end{aligned}$$

We can easily rewrite the above as:

$$\begin{aligned}
& u_i(a_i^{(k)}, s^*_{-i}) \\
=& \frac{-\alpha}{2} \sum_{(i,j)\in E} \frac{(\epsilon_{i,j} - \epsilon''_{i,j})^2}{1-\epsilon_{i,j}} + \sum_{(j,i)\in E} \beta(1-\epsilon''_{j,i}) \\
& + \frac{\alpha}{2} \sum_{(i,j)\in E} (1-\epsilon_{i,j}).
\end{aligned}$$

Plugging $a_i = (\gamma_i, \{\gamma_{j,i}\}_{(j,i)\in E})$ into this equation, we get that

$$\begin{aligned}
&u_i(a_i^{(k)}, s^*_{-i})\\
=&\frac{-\alpha}{2}\sum_{(i,j)\in E}\frac{(\epsilon_{i,j}-(1-(1-\epsilon''_{i,j})))^2}{1-\epsilon_{i,j}}\\
&+\sum_{(j,i)\in E}\beta(1-\epsilon''_{j,i})+\frac{\alpha}{2}\sum_{(i,j)\in E}(1-\epsilon_{i,j})\\
=&\frac{-\alpha}{2}\sum_{(i,j)\in E}\frac{(\epsilon_{i,j}-(1-\gamma_i(1-\epsilon_{i,j})))^2}{1-\epsilon_{i,j}}\\
&+\sum_{(j,i)\in E}\beta\gamma_{j,i}(1-\epsilon_{j,i})+\frac{\alpha}{2}\sum_{(i,j)\in E}(1-\epsilon_{i,j})\\
=&\frac{-\alpha}{2}\sum_{(i,j)\in E}(1-\gamma_i)^2(1-\epsilon_{i,j})\\
&+\sum_{(j,i)\in E}\beta\gamma_{j,i}(1-\epsilon_{j,i})+\frac{\alpha}{2}\sum_{(i,j)\in E}(1-\epsilon_{i,j}).
\end{aligned}$$

Now consider two strategies of node i.

The first strategy of node i is pure strategy s^*_i. If it uses this strategy, its expected utility is

$$\begin{aligned}
&u_i(s^*_i, s^*_{-i})\\
=&\sum_{(j,i)\in E}\beta(1-\epsilon_{j,i})+\frac{\alpha}{2}\sum_{(i,j)\in E}(1-\epsilon_{i,j}).
\end{aligned}$$

The second strategy of node i is an arbitrary strategy $s_i \neq s^*_i$. Suppose that s_i assigns probability P_1 to action $a_i^{(1)}, \ldots$, probability P_K to action $a_i^{(K)}$ ($P_1, \ldots, P_K > 0$; $\sum_{k=1}^K P_k = 1$). Because $s_i \neq s^*_i$, there must be an action $a_i^{(k_0)} = (\gamma_i^{(k_0)}, \{\gamma_{j,i}^{(k_0)}\}_{(j,i)\in E})$ that falls into one of the following two cases:

- **Case 1:** $\gamma_i^{(k_0)} \neq 1$;
- **Case 2:** There exists j_0 such that $\gamma_{j_0,i}^{(k_0)} \neq 1$.

In the first case, because $\alpha > 0$, we can easily obtain that

$$u_i(a_i^{(k_0)}, s^*_{-i}) < u_i(s^*_i, s^*_{-i}).$$

In the second case, because $\beta > 0$, $0 \leq \gamma_{j_0,i}^{(k_0)} < 1$, we can obtain the same inequality.

On the other hand, for all k, clearly we also have that

$$u_i(a_i^{(k)}, s^*_{-i}) \leq u_i(s^*_i, s^*_{-i}).$$

Combining the above two inequalities, we get that

$$\begin{aligned}
u_i(s_i, s^*_{-i}) &= \sum_{k=1}^K P_k u_i(a_i^{(k)}, s^*_{-i})\\
&< u_i(s^*_i, s^*_{-i}).
\end{aligned}$$

This means s^* is a strict Nash equilibrium. □

Unlike a strict dominant strategy equilibrium, a strict Nash equilibrium is not guaranteed to be unique by definition. Below we show that, with our enhanced protocol, the strict Nash equilibrium in Theorem 4 is actually unique.

THEOREM 5. *Suppose that each node i makes z''_i data transmissions and receives payment $\hat{p}_i$. Suppose that for each $(i,j) \in E$, node j receives an auxiliary data traffic of size $\hat{z}^\star_{i,j}$ from node i. Then there is no strict Nash equilibrium other than all player nodes behaving honestly in sending test signals and forwarding the received test signals.*

PROOF. Suppose $s^\triangle$ is a strict Nash equilibrium. For an arbitrary player node i, suppose that $s_i^\triangle$ assigns probability P_1 to action $a_i^{(1)}, \ldots$, probability P_K to action $a_i^{(K)}$ ($P_1, \ldots, P_K > 0$; $\sum_{k=1}^K P_k = 1$). Also suppose that, for each k, $a_i^{(k)} = (\gamma_i^{(k)}, \{\gamma_{j,i}^{(k)}\}_{(j,i)\in E})$.

Because $s^\triangle$ is a strict Nash equilibrium and $P_k > 0$, $a_i^{(k)}$ must be a best response to $s^\triangle_{-i}$. This means:

$$a_i^{(k)} = \arg\max_{a_i} u_i(a_i, s^\triangle_{-i}). \tag{1}$$

On the other hand, using derivations similar to (but slightly different from) those in the proof of Theorem 4, for a general action $a_i = (\gamma_i, \{\gamma_{j,i}\}_{(j,i)\in E})$ of player node i, assuming that $s^\triangle_{-i}$ assigns probability $P'_{k'} > 0$ to actions that forward $\gamma_{i,j}^{(k')} n_{i,j}$ test signals upon receiving $n_{i,j}$ test signals ($k' = 1, \ldots, K'$), we can get that

$$\begin{aligned}
&u_i(a_i, s^\triangle_{-i})\\
=&\frac{-\alpha}{2}\sum_{k'=1}^{K'}\sum_{(i,j)\in E}P'_{k'}(1-\gamma_i\gamma_{i,j}^{(k')})^2(1-\epsilon_{i,j})\\
&+\sum_{k'=1}^{K'}\sum_{(j,i)\in E}P'_{k'}\beta\gamma_{j,i}(1-\epsilon_{j,i})\\
&+\sum_{k'=1}^{K'}\frac{\alpha}{2}P'_{k'}\sum_{(i,j)\in E}(1-\epsilon_{i,j}).
\end{aligned}\tag{2}$$

Combining equations (1) and (2), we can easily have that

$$\begin{aligned}
&\gamma_{j,i}^{(k)}\\
=&\arg\max_{\gamma_{j,i}}\sum_{k'=1}^{K'}\sum_{(j,i)\in E}P'_{k'}\beta\gamma_{j,i}(1-\epsilon_{j,i})\\
=&1.
\end{aligned}$$

Because the above equation holds for all link (j,i) and all action $a_i^{(k)}$ assigned positive probability by $s_i^\triangle$, we know that, for all (i,j) and k', $\gamma_{i,j}^{(k')} = 1$.

Hence, equation (2) becomes

$$\begin{aligned}
&u_i(a_i, s^\triangle_{-i})\\
=&\frac{-\alpha}{2}\sum_{k'=1}^{K'}\sum_{(i,j)\in E}P'_{k'}(1-\gamma_i)^2(1-\epsilon_{i,j})\\
&+\sum_{k'=1}^{K'}\sum_{(j,i)\in E}P'_{k'}\beta\gamma_{j,i}(1-\epsilon_{j,i})\\
&+\sum_{k'=1}^{K'}\frac{\alpha}{2}P'_{k'}\sum_{(i,j)\in E}(1-\epsilon_{i,j}).
\end{aligned}\tag{3}$$

Combining equations (1) and (3), we can easily have that

$$\begin{aligned}
&\gamma_i^{(k)}\\
=&\arg\max_{\gamma_i}\frac{-\alpha}{2}\sum_{k'=1}^{K'}\sum_{(i,j)\in E}P'_{k'}(1-\gamma_i)^2(1-\epsilon_{i,j})\\
=&1.
\end{aligned}$$

So, $s_i^\triangle$ is actually the pure strategy of honestly following the protocol. Equivalently, we have $s^\triangle = s^*$. □

Putting Theorems 4 and 5 together, we have rigorously shown that, with our enhanced protocol, the only strict Nash equilibrium is that all nodes follow the protocol. Hence, it is reasonable for the system to converge to this state.

5. IMPLEMENTATION ISSUES

Using the preceding core techniques, we now present our system implementation architecture to integrate incentive compatibility into a given opportunistic routing protocol. Due to space limitations, we focus on a high-level overview of two key components: routing decision maker and enforcement of routing decision.

Routing Decision Module (RDM): In the preceding sections, we intentionally leave it open on where the link loss probabilities and/or loss probability measurement signals are collected and then the routing decision (payments and forwarding behaviors) are computed. We refer to the module implementing this functionality as the routing decision module (RDM). Note that the RDM's role is to coordinates duplicate forwarding, instead of making an end-to-end (or hop-by-hop) routing decision. The key issues to consider when configuring the RDM include (a) avoiding manipulation (e.g., miscalculation) by involved nodes and (b) being scalable and avoiding single point of failure. In our current design, we support two configurations, indicated by an option in routing messages:

- In the first configuration, the source or destination of each session does the collection and computation. This is particularly suitable for on-demand routing and/or in a hybrid architecture such as [17, 27, 34], where for most traffic either the source or the destination is a base station. If there is a possibility that the RDM node is not trustworthy, a sampling technique is used to validate the computation of the RDM node. That is, for a randomly chosen session, a node may initiate a validation session to test the RDM node. If cheating is detected, a high penalty is assessed (e.g., the node is removed from the system). To prevent potential denial of service attack on such a validation process, we limit the number of sessions that a node initiate sampling.
- In the second configuration, each node computes the part of routing decision that needs to be enforced by itself. The source and destination make random sampling of the RDM at each node.

Routing Decision Enforcement:

After the routing decisions are computed, these decisions need to be *securely enforced*. That is, we must ensure that each node honestly makes all of the transmissions and payments required by the computed routing decision. We refer to the module in charge of this work as the routing decision enforcement module.

Similar to the RDM module, there are also multiple possibilities to implement this module. Part of this module can be implemented by adapting the existing techniques from Sprite [39] and Corsac [40]. First, we calculate the expected number of packets each node should receive in a session, using the routing decision and the loss probabilities. Second, we ensure that the number of packets indeed received is sufficiently close to the expected number. Third, to reduce overheads of secure enforcement, we apply randomly sampling and impose a high punishment on detected cheating behaviors. However, we still need to ensure that the control packets are transmitted securely and that none of them are dropped or ignored. For this issue, more complicated cryptographic techniques are necessary. We leave it to future study.

6. EVALUATIONS

We implement our protocols and conduct extensive experiments on the ORBIT wireless testbed [33]. Our experiments have two objectives. One is to verify that our protocols indeed prevent nodes from deviating from the protocol. The other is to measure the influence of our protocols on the system throughput of opportunistic routing in a wireless network with selfish nodes.

6.1 Methodology

We randomly select 25 nodes from the ORBIT testbed. Figure 3 shows the locations of the nodes. Each node in the testbed is a PC equipped with Atheros AR5002X Mini PCI 802.11a/b/g wireless card attached to an omni-directional antenna. We configure the wireless interface card to operate in 802.11b ad hoc mode, and set the transmission power level at 20 dBm, and the bit-rate at 11Mbps.

Each node in the testbed runs Linux Debian kernel v2.6.22, MadWifi v0.9.3.3 [28], Click v1.6.0 [36], and the MORE package [13]. We set MORE batch size at 32 packets, and packet size at 1500 bytes.

Before running the experiments, we measure pair-wise loss probabilities using a module provided with MORE's package. The loss probabilities between nodes in the testbed are set to values between 24% and 100%.

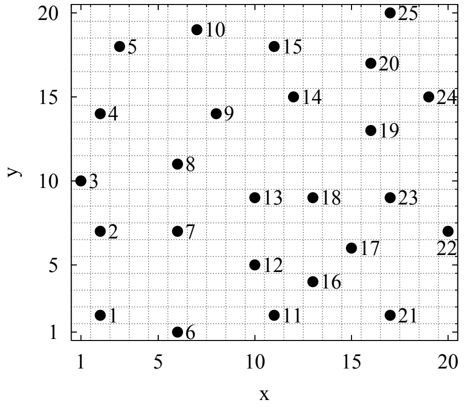

Figure 3: Testbed topology.

Source-Destination Pairs: To evaluate the effects of node locations, we randomly select source-destination pairs in our experiments. After choosing a source-destination pair, we run a session between the pair of nodes for 30 seconds. The source is always backlogged.

Node Behavior: In our experiments, we compare two types of node behavior:

- Honest behavior: Each node follows the protocol faithfully.
- Cheating behavior: As we have mentioned in previous sections, selfish nodes may deviate from the protocol. For each extension of opportunistic routing protocol we have designed, we report results when 20% and 40% of the nodes deviate from the protocol. For experiments on the simple extension, the difference between the loss probability reported by each cheating node and the corresponding real loss probability is $\Delta\epsilon_{i,j} = \epsilon'_{i,j} - \epsilon_{i,j} \in (-0.7, -0.1] \cup [0.1, 0.7)$. For experiments on the enhanced extension, each cheating node sends $\gamma_i n_t$ test signals where $\gamma_i \in (0.0, 0.9] \cup [1.1, 5.0)$, and forwards $\gamma_{j,i} n_{j,i}$ test signals when it receives $n_{j,i}$ test signals for link (j, i), where $\gamma_{j,i} \in (0.0, 0.9]$.

Metrics: We evaluate two metrics:

- Node utility: This metric reflects the impacts of a node's behavior on its own. The target of our evaluation is to verify that, with our protocols, a node's cheating behavior reduces its own utilities. (Thus, our protocols can effectively prevent cheating.) When computing utilities, we set $\alpha = 0.1$ and $\beta = 0.05$.

- Source-destination unicast throughput: This metric reflects the impacts of our designs on the performance of a wireless network with selfish nodes. Our target is to measure these impacts.

6.2 Cheating Behavior and Node Utility

In our first set of experiments we demonstrate that, if a node deviates from our protocols, then its own utility is reduced. For this purpose, we randomly sample several nodes and record the utilities they obtain by following the protocols and by cheating randomly, respectively. The experiment is repeated 100 times with randomly selected source-destination pairs.

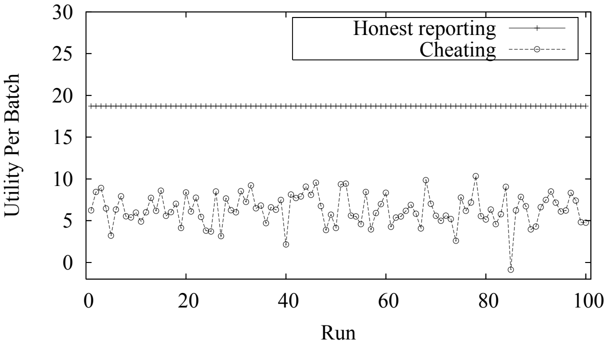

Figure 4: Utilities obtained by node 18 when it is honest vs. cheating, if the simple extension is used. The figure demonstrates that the node can never benefit from cheating.

Simple Extension: Figure 4 illustrates the utilities per batch of a randomly selected node (node 18) if the simple extension is used. In this experiment, the other nodes may either follow the protocol faithfully, or deviate from the protocol by reporting false loss probabilities. We can observe that, the utility obtained by cheating changes from one run to another, sometimes even becoming negative. The reason for this change is that, in each run, the cheating strategy is randomly selected. However, regardless of which cheating strategy is selected, the utility obtained by cheating is always less than the utility obtained by following the protocol.

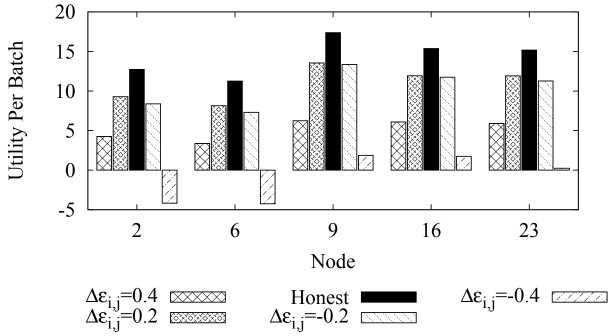

Figure 5: Utilities of five nodes using five strategies: $\Delta\epsilon_{i,j} = 0, \pm 0.2, \pm 0.4$. The transmission is from node 13 to node 3. The honest strategy is always best.

Further results of utility comparison are illustrated in Figure 5. This figure shows five nodes' utilities when each of them uses one of five different strategies: $\Delta\epsilon_{i,j} = 0$ (i.e., being honest), ± 0.2, ± 0.4. We can see that the highest utility is always achieved by the honest strategy only. Furthermore, the more a node deviates from being honest, the less utility this node can obtain.

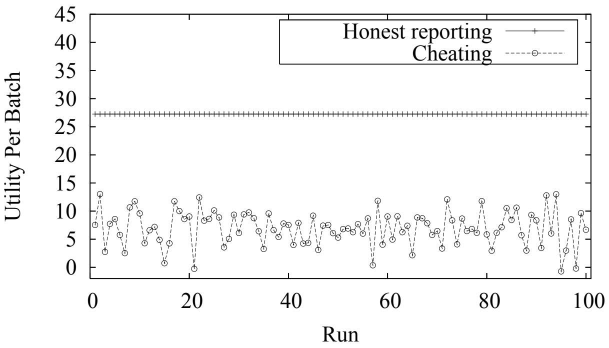

Figure 6: Utilities obtained by node 11 when it is honest vs. cheating, if the enhanced protocol is used. The figure demonstrates that the node can never benefit from cheating.

Enhanced Protocol: Similar experiments are also carried out for the enhanced protocol. Figure 6 shows the utilities obtained by a randomly selected node (node 11) if the enhanced protocol is used. In each run, the cheating strategy is randomly selected. In this experiment, we assume that the other nodes follow the protocol faithfully. Again, regardless of what cheating strategy is used, the utility obtained is always less than the utility obtained by the honest strategy.

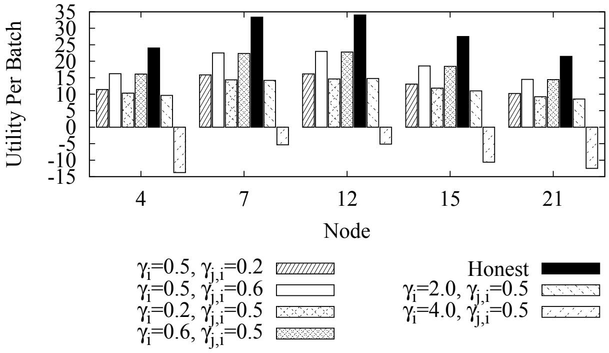

Figure 7: Utilities of five nodes using seven strategies: $(\gamma_i, \gamma_{j,i}) = (4.0, 0.5)$, $(2.0, 0.5)$, $(1, 1)$ (i.e., being honest), $(0.6, 0.5)$, $(0.2, 0.5)$, $(0.5, 0.6)$, $(0.5, 0.2)$. The transmission is from node 10 to node 18. The honest strategy is always best.

Figure 7 shows five nodes' utilities when each of them uses one of seven different strategies: $(\gamma_i, \gamma_{j,i}) = (4.0, 0.5)$, $(2.0, 0.5)$, $(1, 1)$ (i.e., being honest), $(0.6, 0.5)$, $(0.2, 0.5)$, $(0.5, 0.6)$, $(0.5, 0.2)$. Just like what we have seen for the simple extension, the highest utility is achieved by the honest strategy only.

6.3 Impacts on End-to-End Throughput

Our second set of experiments are to demonstrate that our protocols improve the end-to-end performance of opportunistic routing in face of selfish nodes. We use the source-destination unicast throughput as the performance metric. As we have mentioned, selfish nodes may cheat in reporting and/or measuring loss probabilities. Consequently, an opportunistic protocol without incentive compatibility (like the original protocol of MORE) can compute routing decisions that have lower end-to-end performance. In contrast, our protocols can prevent cheating of selfish nodes, and thus can boost the end-to-end performance in face of selfish nodes.

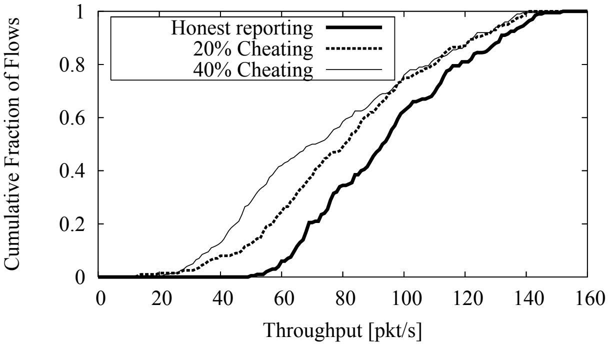

Figure 8: CDF of the unicast throughput achieved with vs. without the simple extension on 200 source-destination pairs. When the original MORE protocol is used, 20% or 40% of the nodes cheat in reporting loss probabilities.

Simple Extension: Figure 8 shows the cumulative distribution function (CDF) of the achieved throughput on 200 randomly selected source-destination pairs in the testbed. The figure shows the results both when nodes honestly report loss probabilities (i.e., when our simple extension is used) and when some of them do not (i.e., when the original MORE protocol is used without our extension). In the latter case, we consider two situations, in which 20% and 40% of the nodes cheat randomly in reporting the loss probabilities, respectively. Cheating nodes randomly select their strategies in the range given before. We observe that the throughput of our simple extension is significantly higher than those of the original MORE protocol. Specifically, for the median case, our simple extension achieves 14.8% (resp., 32.8%) higher throughput than the original MORE protocol when 20% (resp., 40%) of the nodes cheat.

To further understand the benefits of honest reporting and the effects of the length of routing paths, we evaluate the throughput taken over 50 experiments with different source-destination pairs on each number of hops.

Figure 9 shows the average throughput over 50 runs as a function of the number of hops on the path. As expected, the average throughput decreases with the number of hops. However, when the number of hops is larger, the advantage of using our simple extension is more significant. Overall our simple extension achieves 7.8-33.2% and 9.9-58.0% gain in throughput in the case where 20% and 40% of the nodes are cheating, respectively.

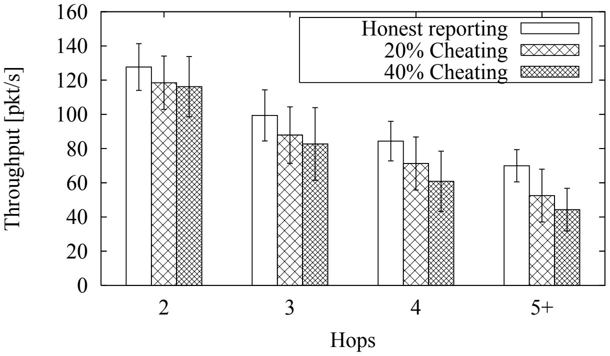

Figure 9: Average throughput as a function of the number of hops on the path, with vs. without the simple extension. Standard deviations are shown using lines.

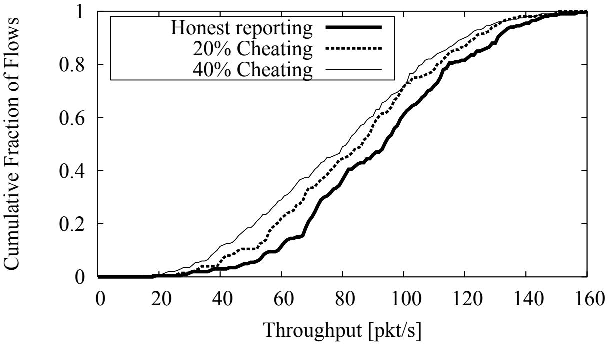

Figure 10: CDF of the unicast throughput achieved by the enhanced protocol vs. the original MORE protocol on 200 source-destination pairs. When the original MORE protocol is used, 20% or 40% of the nodes cheat in measuring and reporting loss probabilities.

Enhanced Protocol: We also carried out similar experiments for the enhanced extension. Figure 10 shows the CDF of the achieved throughput taken over 200 randomly selected source-destination pairs. The figure shows the results both when nodes are honest (i.e., when our enhanced protocol is used) and when some of them cheat randomly (i.e., when the original MORE protocol is used). In the latter case, again we consider two situations in which 20% and 40% of the nodes cheat, respectively. Cheating nodes randomly select their strategies. The throughput of our enhanced protocol is clearly higher than those of the original MORE protocol. For the median case, it is 8.0% (resp., 14.6%) more than the original MORE protocol when 20% (resp., 40%) of the nodes cheat. Compared with Figure 8, the throughput gain of the enhanced protocol looks less significant than that of the simple extension *on average*. This is because the experiments using the enhanced protocol allow the cheating strategy to be more complicated. A random strategy here is harder to prevent, but has less influence on the throughput.

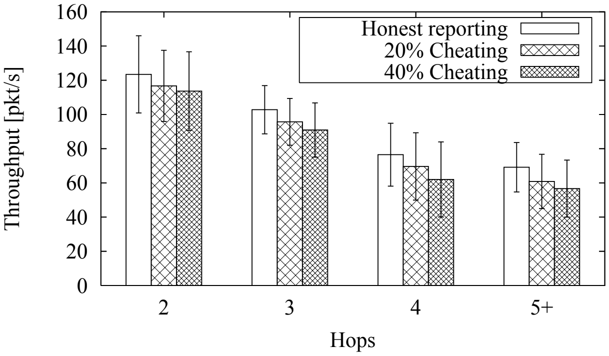

Figure 11: Average throughput as a function of the number of hops on the path, achieved by the enhanced protocol vs. the MORE protocol. Standard deviations are shown using lines.

Figure 11 shows the average throughput over 50 runs as a function of the number of hops on the path. Overall our enhanced protocol achieves 5.8-13.7% and 8.6-23.4% gain in throughput in the cases where 20% and 40% of the nodes are cheating, respectively.

6.4 Overhead

The protocols presented in this paper inherit coding overhead, memory overhead, and packet header overhead from existing opportunistic routing protocols (*e.g.*, MORE). In addition, our protocols require auxiliary transmission to enforce the incentive-capability. On average, the auxiliary traffic for each session is 26.73 KB, which is very small compared with 3.93 MB of data transmitted in 30 seconds. The ratio between auxiliary traffic and data throughput is 0.66%.

Note that the link loss reporting packets and the test signals used to measure link loss probabilities do not constitute our protocols' specific overhead. These packets are used by most of existing wireless routing protocols.

6.5 Auxiliary Payment

The auxiliary payment is also very small compared with the total payment. We randomly sampled 200 source-destination pairs. The results show that the ratio between auxiliary payment and total payment is only 0.23% and 1.20% for the simple extension and the enhanced extension, respectively.

7. RELATED WORK

To the best of our knowledge, so far there has not been any research work on incentive-compatible opportunistic routing. So in this section we focus on the related work on opportunistic routing and cooperation in wireless networks.

7.1 Opportunistic Routing in Wireless Networks

Opportunistic routing belongs to cooperative diversity techniques (e.g. [6, 23, 31]) which take advantage of broadcast transmissions to send information through multiple concurrent relays. Nodes can combine information from multiple signals so that they can make best decisions of routing or forwarding. As an example, protocols in [23] fully exploit spatial diversity in the channel by allowing all nodes that overheard a transmission to simultaneously forward the signal. Another example is the protocol in [6], which optimizes the choice of forwarder from multiple receivers by deferring to choose each hop after transmission.

The concept of opportunistic routing was first developed by Biswas and Morris in the context of wireless mesh networks. They claimed that opportunistic routing can potentially increase the throughput and proposed an integrated routing and MAC protocol, named ExOR, to achieve the throughput gain [6]. To improve the system throughput, Chachulski et al. designed MORE [13], which combines random network coding and opportunistic routing to avoid transmission duplication. Our protocols are incentive-compatible extensions for an opportunistic routing protocol, like MORE, such that the system performance can be maintained in face of selfish nodes.

The basic opportunistic routing protocol uses network coding, which lets the routers encode information in received packets before transmission. Earlier works on network coding [3, 16, 22, 25] mainly focused on mixing information in different packets to achieve multicast capacity. Some recent works studied possible coding opportunities in wireless networks [18–21, 26] to increase system throughput.

7.2 Cooperation in Wireless Networks

The problem of cooperation in wireless networks has received a lot of attention in recent years (e.g., [8, 9, 29, 35]). The solutions proposed so far fall into two categories, credit-based approaches and reputation-based approaches. As our protocols belong to the former category, we mainly discuss works in this category.

Buttyan and Hubaux proposed the first credit-based system [10, 12] in wireless ad-hoc networks in the Terminodes project. In [10], they propose the usage of nuglets, a virtual currency, to pay nodes to forward others' packets. Motivated by the nuglet, several other credit-based systems were proposed to stimulate cooperation in packet forwarding. In [39], Zhong et al. proposed Sprite, a credit-based system which uses a central authority to collect receipts from forwarding nodes. Charges and rewards are based on the receipts. In [34], Ben Salem et al. proposed a charging and rewarding scheme based on symmetric cryptography to make collaboration rational for selfish nodes. In [17], Jakobsson et al. proposed a micro-payment scheme for multi-hop cellular networks to encourage collaboration in packet forwarding.

In [5], Anderegg and Eidenbenz studied cooperation in the (deterministic) routing problem. They applied the VCG mechanism to design a routing protocol for a wireless network with selfish nodes. Then, the authors of [40] proposed Corsac, which integrates VCG and cryptographic technique to solve the combined problem of routing and packet forwarding. Recently, OURS was proposed by Wang et al. [38]. It has much smaller over-payments than VCG-based solutions. Another recent work by Zhong and Wu [41] studied collusion resistance for incentive-compatible routing.

It is easy to see that all the above works to stimulate cooperation are dedicated to traditional, deterministic routing, which chooses the next hop before transmitting a packet. In contrast, our incentive-compatible schemes are designed for opportunistic routing, and can bring the system throughput back to the high level achieved by opportunistic routing protocols despite of potentially selfish nodes.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we present simple, novel techniques to integrate incentive compatibility into a class of opportunistic routing protocols. We integrate our protocols with MORE in a Linux implementation and demonstrate on the ORBIT testbed that (a) cheating decreases a node's utility under our protocols and (b) incentive can substantially improve overall network throughput (5.8%-58.0% in our evaluated settings) in the presence of selfish nodes. Our paper has focused on providing incentives for single-rate opportunistic routing protocols. As for future work, we are interested designing similar simple techniques that can work in multi-rate wireless networks.

9. ACKNOWLEDGMENTS

F. Wu was supported by NSF CNS-0524030; T. Chen and S. Zhong were supported in part by NSF CNS-0524030. Y. Richard Yang was supported in part by grants from the U.S. NSF. We appreciate the anonymous reviewers whose comments led to improvement of the paper. Our shepherd, Tristan Henderson, gave us highly valuable comments to improve the paper.

10. REFERENCES

[1] E. Adar and B. A. Huberman. Free riding on Gnutella. *First Monday*, 5(10), Oct. 2000.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM'04*, Portland, OR, Sept. 2004.

[3] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204 - 1216, 2000.

[4] I. F. Akyildiz and X. Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9), 2005.

[5] 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 the 9th International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, Sept. 2003.

[6] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. In *Proceedings of ACM SIGCOMM'05*, Philadelphia, PA, Aug. 2005.

[7] W. E. Bluhm. *Society of Actuaries 50th Anniversary Monograph*, chapter V: Cumulative Anti-Selection Theory. 1999.

[8] S. Buchegger and J.-Y. Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP '02)*, Canary Islands, Spain, Jan. 2002.

[9] 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 the 3rd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausanne, Switzerland, June 2002.

[10] L. Buttyan and J. P. Hubaux. Enforcing service availability in mobile ad-hoc WANs. In *Proceedings of the First ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc'00)*, Boston, MA, Aug. 2000.

[11] L. Buttyan and J.-P. Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge University Press, 2007.

[12] L. Buttyan and J. P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM Journal for Mobile Networks (MONET), special issue on Mobile Ad Hoc Networks*, summer 2002.

[13] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of ACM SIGCOMM'07*, Kyoto, Japan, Aug. 2007.

[14] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, Sept. 2003.

[15] Ugly truth about mesh networks. http://www.dailywireless.org/2004/06/28/ugly-truth-about-meshnetworks.

[16] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973 - 1982, 2005.

[17] M. Jakobsson, J. P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In *Proceedings of the 7th International Conference on Financial Cryptography (FC'03)*, Guadeloupe, French West Indies, Jan. 2003.

[18] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein. Growth codes: Maximizing sensor network data persistence. In *Proceedings of ACM SIGCOMM'06*, Pisa, Italy, Sept. 2006.

[19] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *Proceedings of ACM SIGCOMM'07*, Kyoto, Japan, Aug. 2007.

[20] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. Médard. The importance of being opportunistic: Practical network coding for wireless environments. In *Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 2005.

[21] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In *Proceedings of ACM SIGCOMM'06*, Pisa, Italy, Sept. 2006.

[22] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782 - 795, 2003.

[23] D. Laneman and G. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12):3062 - 3080, 2004.

[24] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom'01)*, Rome, Italy, July 2001.

[25] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371 - 381, 2003.

[26] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, and a. H. L. E. Ahmed. Achieving minimum-cost multicast: A decentralized approach based on network coding. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, Miami, FL, Mar. 2005.

[27] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A unified cellular and ad-hoc network architecture. In *Proceedings of The 9th International Conference on Mobile Computing and Networking (MobiCom'03)*, San Diego, CA, Sept. 2003.

[28] MadWifi Project Team. http://madwifi.org.

[29] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, MA, Aug. 2000.

[30] Meraki Networks. http://meraki.com.

[31] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of the 11th International Conference on Mobile Computing and Networking (MobiCom'05)*, Cologne, Germany, Sept. 2005.

[32] MuniWireless LLC. http://www.muniwireless.com.

[33] Rutgers ORBIT project team. http://www.orbit-lab.org.

[34] N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *Proceedings of the 4th ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)*, Annapolis, MD, June 2003.

[35] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao. Cooperation in wireless ad hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, San Francisco, CA, Apr. 2003.

[36] The Click Modular Router Project Team. http://www.read.cs.ucla.edu/click/.

[37] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.

[38] W. Wang, S. Eidenbez, Y. Wang, and X.-Y. Li. OURS: Optimal unicast routing systems in non-cooperative wireless networks multihop routing in sensor networks. In *Proceedings of The 12th International Conference on Mobile Computing and Networking (MobiCom'06)*, Los Angeles, CA, Sept. 2006.

[39] S. Zhong, J. Chen, and Y. R. Yang. Sprite, a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, San Francisco, CA, Apr. 2003.

[40] S. Zhong, L. Li, Y. Liu, and Y. R. 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 The 11th International Conference on Mobile Computing and Networking (MobiCom'05)*, Cologne, Germany, Sept. 2005.

[41] S. Zhong and F. Wu. On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks. In *Proceedings of The 13th International Conference on Mobile Computing and Networking (MobiCom'07)*, Montreal, Canada, Sept. 2007.