

An Optimal, Strategy-Proof Scheme for Multi-Path Traffic Assignment in Non-Cooperative Networks

Fan Wu, *Member, IEEE*, Sheng Zhong, *Member, IEEE*, and Jiqiang Liu, *Member, IEEE*

Abstract—Multi-path routing has long been studied as an important routing strategy in networks. Many multi-path routing protocols schedule traffic among multiple paths in order to distribute traffic load. However, existing multi-path routing protocols with traffic assignment require that all nodes in the network follow the protocol, which may not always be a valid assumption when the network consists of selfish nodes. In this paper, we propose an optimal, strategy-proof scheme for multi-path traffic assignment (OSMA) in non-cooperative networks. When OSMA is used, behaving honestly is to the best interest of each selfish node regardless of any other nodes' behavior. Furthermore, our scheme is guaranteed to compute the lowest cost traffic assignment with the existence of these selfish nodes. Our evaluations verify that our scheme is optimal and strategy-proof, and demonstrate that the scheme has very low communication and computation overhead.

Index Terms—Routing, traffic assignment, mechanism design.

I. INTRODUCTION

MULTIPATH Routing has long been studied as an important routing strategy in networks. It provides multiple paths for sending data from a source to a destination to exploit the resources of the underlying physical network. Previous research has demonstrated that multi-path routing can achieve route resilience, higher aggregate bandwidth, smaller end-to-end delays, and better load balancing [9], [32].

Multipath routing has been explored in both wired and wireless networks. In wired network, multi-path routing is implemented as a feature of Asynchronous Transfer Mode (ATM) networks [10], Open Shortest Path First (OSPF) protocol [23], and external Border Gateway Protocol (eBGP) [29]. For wireless networks, multi-path routing is also extensively studied in recent years. A number of multi-path routing protocols for wireless networks have been proposed. Some

of them [20], [21], [27], [36] maintain multiple routes and utilize these routes only when the primary root fails. Others [19], [26], [28] further schedule traffic among multiple paths in order to distribute load. In this paper, we are mainly concerned with the latter, i.e., multi-path routing protocols that assign the traffic among the multiple paths, such that transmissions can be carried out simultaneously over multiple paths. Note that while this work can actually be applied to both wired and wireless networks, we believe it has more potential impacts on wireless networks rather than on wired networks. The reason is that, in many wireless networks, devices are contributed by users, and thus the problem of selfish behavior [3], [31], [33], [39] is probably more important in the context of wireless networks than in wired networks. Therefore, we focus on wireless networks in this paper.

Assigning traffic flow among multiple paths from source to destination such that the resulting forwarding cost is minimized, is subject to the problem of optimal routing [4]–[6], [15], [16]. We note that the existing multi-path routing protocols with traffic assignment require that all nodes in the network follow the prescribed protocol and cooperate with each other. However, this assumption may not be valid when the network consists of selfish nodes [3], [11]–[14], [17], [30], [31], [33]–[35], [37]–[39]. Forwarding traffic flows depletes scarce resources such as power, and reduces available bandwidth to the node itself. When nodes in the network belong to different owners, they may not have incentives to forward others' flows. In this paper, we consider the selfish behavior of nodes in such networks. Specifically, a selfish node is an economically rational node whose objective is to maximize its own utility. So our question is how to design a multi-path routing protocol such that selfish nodes will behave truthfully. In this paper, we propose an optimal, strategy-proof scheme for multi-path traffic assignment (OSMA) in non-cooperative networks. OSMA has great difference from existing works, such as optimal routing protocols. The existing works consider the problem when all the nodes in the network are cooperative; while our work deals with non-cooperative networks. Solving multi-path routing problem in cooperative and non-cooperative networks has different preconditions and assumptions. Consequently, the results in the two scenarios are different.

To the best of our knowledge, there has not been any work addressing selfish behavior for multi-path routing. However, there has been extensive study on traditional unicast and multicast in selfish networks. Considering the complexity and the subtlety of the incentive issues, many researchers apply game-theoretic techniques to analyze and design protocols in wireless and wired networks. In wireless network, vari-

Manuscript received July 15, 2007; revised June 9, 2008 and August 9, 2009; accepted December 10, 2009. The associate editor coordinating the review of this paper and approving it for publication was W. Liao.

Part of the results were presented at IEEE GLOBECOM '07.

F. Wu was with the Computer Science and Engineering Department, University of Buffalo, the State University of New York, Buffalo, NY 14260. He is now with the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (e-mail: fwu@crhc.illinois.edu).

S. Zhong is with the Computer Science and Engineering Department, University of Buffalo, the State University of New York, Buffalo, NY 14260 (e-mail: szhong@cse.buffalo.edu).

J. Liu is with the School of Computer and Information Technology, Beijing Jiao Tong University, Beijing 100044, China (e-mail: jqliu@bjtu.edu.cn).

F. Wu was supported in part by NSF CNS-0524030 and CNS-0845149. S. Zhong was supported in part by NSF CNS-0524030, CNS-0845149, and CCF-0915374. J. Liu was supported in part by Natural Science Foundation of China 60973112 and Major State Basic Research Development Program 2007CB307101.

Digital Object Identifier 10.1109/TWC.2010.03.080760

ous incentive-based approaches have been proposed to solve packet routing or forwarding problems [3], [11], [30], [31], [33], [35], [38], [39]. Wang *et al.* [34] and Yuen *et al.* [37] investigated the problem of bandwidth allocation and multicast tree formation in overlay networks. Feigenbaum *et al.* [12], [13] considered both unicast and multicast in Internet. Felegyhazi *et al.* [14] and Halldorsson *et al.* [17] studied the problem of sharing spectrum using game theory. Finding and punishing deviating nodes is another interesting category of works on incentive-capability. A number of papers are dedicated to mitigate the misbehavior of selfish nodes by this method. For example [7], [8], [22]. Yet another kind of related work is based on VCG auction (e.g., [18]). The similarity is mainly due to the fact that our work is based on VCG, and that VCG in general can be applied to auctions. However, it is worth noting that our use of VCG is nontrivial, and does not directly apply to spectrum auctions.

Although the methods mentioned above can not be directly used in the multi-path routing scenario, we believe that we can develop a game-theoretic solution for multi-path routing that can deal with the selfish behavior of nodes. To design a multi-path routing protocol for selfish networks, instead of starting from scratch, we consider some existing multi-path routing protocol and make it compatible with selfish behavior by redesigning its traffic assignment scheme. That is, we study how to assign the data traffic to the multiple paths established by a given multi-path routing protocol between the source and the destination, such that the participating selfish nodes will behave truthfully. First, we give a game-theoretic model for this problem, which we call traffic assignment game. In this game, nodes are paid for carrying traffic. Then, we propose OSMA for traffic assignment, which is shown to be *strategy-proof* in the above model. Here intuitively, the scheme being *strategy-proof* means that behaving truthfully is to the best interest of each selfish node, *regardless of other nodes' behavior*. This is a very strong solution concept in game theory [24]. It provides a strong guarantee that cooperation is compatible with selfish nodes' incentives. Next, we show that OSMA is guaranteed to compute the lowest cost traffic assignment with the existence of selfish nodes. Note that this is nontrivial because selfish nodes always make decisions to maximize their own utilities, which is usually different from the decisions that can achieve the lowest overall costs in the entire system. We also show that the payment-cost ratio of our scheme is within a bound determined by certain characteristics of the network. Finally, evaluations demonstrate that OSMA has very low communication and computation overhead.

The rest of this paper is organized as follows: In Section II we introduce some preliminaries. In Section III, we present our traffic assignment game model. In Section IV, we go to the details of OSMA and present our analysis. In Section V, we show the results of evaluations. Finally, we conclude the paper and point out potential future works in Section VI.

II. TECHNICAL PRELIMINARIES

Before introducing our model, we need to recall some notations from mechanism design. In the classic model of mechanism design, there is a set of players $N = \{1, 2, \dots, n\}$. Each player $i \in N$ has some private information t_i called

type, which determines its preferences over different outcomes of a game. The players' type vector is denoted by $t = (t_1, t_2, \dots, t_n)$. For each player i , there is a set of available actions A_i . Every player i chooses an action $a_i \in A_i$. As a notational convention, a_{-i} represents the actions of all players except player i . Note that $a = (a_i, a_{-i})$ is an action profile, in which player i takes action a_i and the other players take actions a_{-i} . Base on the selected strategies (and consequently the action profile a) users will obtain their outcome $o(a)$ and payment $p(a)$, where $p(a) = (p_1(a), p_2(a), \dots, p_n(a))$ is the vector of payment to each player. A valuation function $v_i(o(a))$ assigns a monetary value for player i to each possible output $o(a)$. Node i 's utility u_i is a function as follows:

$$u_i(a) = v_i(o(a)) + p_i(a). \quad (1)$$

Given above notations, now we can define a very strong solution concept called *dominant strategy* [25].

Definition 1: A dominant strategy of a player is one that maximizes its utility regardless of what strategies other players choose. Specifically, a_i is player i 's dominant strategy if, for any $a'_i \neq a_i$ and any a_{-i} ,

$$u_i(a_i, a_{-i}) \geq u_i(a'_i, a_{-i}). \quad (2)$$

A *direct-revelation* mechanism is a mechanism in which the only actions available to players are to make claims about their preferences to the mechanism. That is, the strategy of player i is reporting type $\hat{t}_i = s_i(t_i)$, based on its actual preferences t_i . A direct-revelation mechanism is *incentive-compatible* (IC) if reporting truthful information is a dominant strategy for each player. Another important property of a mechanism is *individual-rationality* (IR) — each player can always achieve at least as much expected utility from participation as without participation. Finally, we say a direct-revelation mechanism is *strategy-proof* if it satisfies both IC and IR properties.

Definition 2 (Strategy-proof Mechanism): A direct-revelation mechanism is *strategy-proof* if revealing truthful information is a dominant-strategy equilibrium.

III. A MODEL OF TRAFFIC ASSIGNMENT GAME

We give the detail of our traffic assignment game's model in this section. Consider a network represented by $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of nodes and $E = \{e_1, e_2, \dots, e_m\} \subseteq V \times V$ is the set of communication links, in which $e_k = v_i v_j$ means that node v_i and v_j can communicate with each other directly. Each node $v_i \in V$ has a fixed capacity C_i for data transmission. In wired networks, C_i is the capacity of its network adaptor, while in wireless networks, we assume that C_i is the portion of communication media used by the node (In particular, C_i can be the portion of time slots in TDMA, the bandwidth of a sub-channel in FDMA, the bandwidth of a coding module in CDMA, or the expected bandwidth of a node in CSMA.).

We model multi-path routing with traffic assignment as a mechanism design problem, which we call the *traffic assignment game*. Suppose there are a source node S and a destination node D . Then the player set of the game is $V - \{S, D\}$. Each node $v_i \in V$ knows its cost function $f_i(x)$ and current available bandwidth $0 \leq b_i \leq C_i$, which are

defined as its *type*. Formally, we have type $t_i = \langle f_i(x), b_i \rangle$. The cost function $f_i(x)$ is an increasing function¹ indicating the cost of forwarding one unit of traffic when x units of bandwidth has been used. The cost can be the expected transmission cost when the network is lossy, including expense of consumed power, losses in sacrificing bandwidth to send own traffic flows, and so on. The more bandwidth a node allocates for forwarding traffic flows, the less bandwidth it can use to send its own traffic flows, and the longer latency and higher transmission loss probability its own traffic flows may suffer. Intuitively, a node will get increasingly reluctant to sell its bandwidth, when more and more bandwidth is used. So only an increasing cost function can capture the change of forwarding cost. Suppose a new flow request wants to go through node v_i with bandwidth requirement $q \leq b_i$. Then the cost of node v_i for forwarding this flow is

$$c_i(q) = \int_{C_i - b_i}^{C_i - b_i + q} f_i(x) dx. \quad (3)$$

There is a payment for nodes to provide incentive for them to carry the traffic flows. This payment should cover the cost on forwarding the traffic flows. In this game, each player node v_i chooses an action a_i . The action is declaring a cost function $f_i(x)$ and an available bandwidth b_i . The profile of all players' actions is denoted by $a = (a_i)_{v_i \in V - \{S, D\}}$. This action profile determines both the valuation function $v_i(o(a)) = -c_i(o(a))$ and the payment $p_i(a)$. Here $o(a)$ is the outcome of the game. In our game model, $o(a)$ is equal to the traffic assignment vector R . So the utility of the node is:

$$u_i(a) = p_i(a) - c_i(o(a)). \quad (4)$$

Nodes are rational. The objective of nodes is to maximize their utility.

IV. TRAFFIC ASSIGNMENT SCHEME — OSMA

In this section, we propose our traffic assignment scheme OSMA and present our analysis of optimality, strategy-proofness, and payment-cost ratio. OSMA is designed for assigning traffic among multiple node-disjoint paths, which do not have any nodes in common except the source and destination. So OSMA can be used for any multi-path routing protocol that schedules multiple node-disjoint paths (e.g., [21], [23], [36]). In this paper, we assume that the network topology is biconnected — there exist at least two node-disjoint paths from any source node to any destination node. This is also where our model is different from existing economic incentive routing protocols. The existing economic incentive routing protocols deal with single-path routing, while our work deals with multi-path routing. When this assumption does not hold in some case (e.g., there is only one available path), the schemes in some other literatures can be applied (e.g., [3], [11], [30], [33], [39]). In particular, the authors of [3], [11] applied the VCG mechanism to design a routing protocol for a wireless network with selfish nodes. In [30], Ben Salem et al. addresses the problem of selfishness in multi-hop cellular networks, using a protocol based on symmetric key cryptography.

¹The cost function $f_i(x)$ can be either linear or nonlinear, concave or convex, depending on the characteristics of the network modeled.

In [39], Zhong et al. proposed Corsac, which integrates VCG and cryptographic technique to solve the combined problem of routing and packet forwarding. In [33], Wang et al. proposed their solution OURS, which introduces dummy packets to guarantees low overpayments.

A. Scheme

Given a new traffic request q from a source node S to a destination node D , there is a set of node-disjoint paths $P = \{P_1, P_2, \dots, P_m\}$ found by the multi-path routing protocol (e.g., [21], [23], [36]). The action for each player node v_i on any of these paths is declaring the cost function $f_i(x)$ and available bandwidth b_i .

Algorithm 1 Traffic Assignment Algorithm

Input: Set of paths P , cost functions and available bandwidth $\langle f_i(x), b_i \rangle_{v_i \in P_j \in P}$, and bandwidth requirement q .

Output: Traffic assignment $R = (r_1, r_2, \dots, r_m)$.

- 1: $R = 0^m$.
 - 2: $W = \emptyset$.
 - 3: $\forall P_j \in P$, define $F_j(x) = \sum_{v_i \in P_j} f_i(C_i - b_i + x)$.
 - 4: **while** $q > \sum_{j=0}^m r_j$ **do**
 - 5: **if** $(P \neq \emptyset)$ **then**
 - 6: $k = \underset{P_j \in P}{\operatorname{argmin}} F_j(0)$.
 - 7: Move P_k from P to W .
 - 8: $t = \min\{F_j(0) | P_j \in P\}$.
 - 9: **else**
 - 10: $t = \operatorname{MAX}$.
 - 11: **end if**
 - 12: **if** $\sum_{P_j \in W} F_j^{-1}(t) \leq q$ **then**
 - 13: $r_j = F_j^{-1}(t), \forall P_j \in W$.
 - 14: **else**
 - 15: Use binary search to find $0 \leq d \leq t$, s.t.,
 $\sum_{P_j \in W} F_j^{-1}(d) = q$.
 - 16: $r_j = F_j^{-1}(d), \forall P_j \in W$.
 - 17: **end if**
 - 18: **end while**
-

After collecting all the information, the source node (or the destination node) computes the traffic assignment using Algorithm 1. Intuitively, the algorithm computes the traffic assignment in a water-filling style. It starts by filling the lowest cost path with traffic, until this path becomes less cost-efficient than a different path. Then it switches to the current lowest cost path and fills that path with traffic. It keeps doing this until all the traffic is assigned.

Specifically, in lines 1 – 3, we initialize the traffic assignment vector R , set of selected paths W , and combine the cost functions declared by nodes on each path $P_i \in P$ to get an integrated path cost function $F_i(x)$. Here $F_i(x)$ is the cost on path P_i for forwarding one unit of traffic when x units of traffic has been assigned to path P_i . Then we turn to work on these integrated path cost functions. In lines 4 – 18, we iteratively select the path with the lowest “marginal” cost and move it to W ; and assign as much as possible traffic to all selected paths in W . When the selected paths are no longer cost efficient (i.e., enough traffic has been added) and there

TABLE I
VARIABLES

N	set of players
\bar{t}	vector of types
A_i	action set of player i
a	action profile
$o(a)$	outcome of game
$p(a)$	vector of payments
$v_i(o(a))$	valuation of player i
$u_i(a)$	utility of player i
$G = (V, E)$	network
C_i	capacity of node v_i
$f_i(x)$	cost function of node v_i
b_i	available bandwidth at node v_i
q	bandwidth requirement
$c_i(q)$	forwarding cost on node v_i
P	set of candidate paths
R	vector of traffic assignment
W	set of selected paths
$F_j(x)$	integrated cost function of path P_j

still remains unassigned traffic, the above iteration has to be repeated to find the next lowest-marginal-cost path and add it to the set W . In lines 5 – 11, if P is not empty, we find the path P_k who has the lowest marginal cost and move it from P to W , and set t to the lowest marginal cost in current set P ; If P is empty, the t is set to a large enough value. Here, t will be used to compute the traffic assignment vector. In lines 12 – 17, first we compute the traffic assignment vector by the inverse function $F_i^{-1}(x)$, when the cost level reaches t . Then if the assigned traffic has not met the bandwidth requirement q , the loop continues; otherwise, we use binary search to find the final cost level d which results in a traffic assignment meeting the bandwidth requirement q . Finally, when the iteration stops, the vector R is the final assignment of traffic.

For convenience, we use Table I to summaries the variables used in this paper.

B. Payment to Each Node

There is a payment for each participating node given by the source node S . To calculate the payment to each node v_i in each path in P , we call the Algorithm 1 twice. Suppose v_i is on path $P_j \in P$. The first execution of Algorithm 1 is exactly what we have described in Section IV-A. In the second execution, we remove the path P_j from the network. Let R and R' be the traffic assignment computed by the two executions of Algorithm 1. Then the payment p_i to v_i is defined as follows:

$$p_i = \sum_{P_k \in P - \{P_j\}} \int_{r_k}^{r'_k} F_k(x) dx - \sum_{v_h \in P_j - \{v_i\}} \int_0^{r_j} f_h(C_h - b_h + x) dx. \quad (5)$$

Intuitively, this is the cost difference if the traffic assigned to the path passing node v_i has been assigned to other paths, additional with the forwarding cost on node v_i .

If a node is not in any path in P , it receives no payment.

Just as in [11], [30], [33], [38]–[40], we assume that there is a credit clearance center (CCC) in the system. Each node has an account in the CCC and each transaction has to be processed by the clearance center. The CCC is a server connected to the Internet. So the nodes can access the CCC

whenever they have connections to the Internet to complete the transaction.

By now, we have completed presenting our design of OSMA. We have to note that our design applies to sequential bandwidth requests only. On one hand, if there are multiple simultaneous bandwidth requests for bandwidths, the scheme will have suboptimal performance. Since our scheme does not coordinate sources of simultaneous bandwidth requests, the computed traffic assignments, which share some intermediate nodes, may suffer from higher forwarding costs and payments than expected during the transmission. Furthermore, the total traffic going through a shared node may exceed the node's capacity, which will result in packet loss or high forwarding delay. On the other hand, the existence of multiple simultaneous bandwidth requests may allow a node to cheat some of the sources, in order to get more benefit from the other sources. For example, suppose there are 2 bandwidth requests q_a and q_b on node v_i , and the optimal traffic assignment requires node v_i to contribute all its bandwidth to q_a . But node v_i can get more utility from bandwidth request q_b than q_a . In this case, node v_i can get more utility by simply cheating q_a 's source that it has very high forwarding cost and limited available bandwidth, such that node v_i can allocate more bandwidth to q_b . Clearly, the selfish behavior of node v_i inevitably increases the overall forwarding cost of q_a and q_b . To solve these issues, it is needed to model the problem as a different game, which contains more strategies of the players. We leave these to our future work.

Next, we will prove the following important properties of our scheme:

- 1) If OSMA is used, the traffic assignment is the most cost efficient given nodes' true type.
- 2) And the strategy-proofness of OSMA. In other words, if OSMA is used, every node maximize its utility if and only if it reveals true type.

C. Optimality

Theorem 1: Our traffic assignment scheme OSMA computes the most cost efficient traffic assignment if truthfulness is guaranteed.

Proof: We assume every node declares its real type. Suppose that R is the traffic assignment computed by our algorithm and R' is any traffic assignment for traffic request q . Then we have:

$$\sum_{P_j \in P} r'_j = \sum_{P_j \in P} r_j = q \quad (6)$$

We divide P into two subsets $P^{(1)}$ and $P^{(2)}$, such that $P = P^{(1)} \cup P^{(2)}$, $P^{(1)} \cap P^{(2)} = \emptyset$, and

$$\begin{cases} \forall P_j \in P^{(1)}, r_j > r'_j, \\ \forall P_j \in P^{(2)}, r_j \leq r'_j. \end{cases} \quad (7)$$

Since $P^{(1)} \subseteq W$ according to Algorithm 1, we have,

$$\forall P_j \in P^{(1)}, F_j(r_j) = d. \quad (8)$$

But $P^{(2)}$ may contain the paths whose marginal cost is higher than or equal to d . So we have

$$\forall P_j \in P^{(2)}, F_j(r_j) \geq d. \quad (9)$$

Now we consider the cost difference between the two traffic assignment.

$$\begin{aligned}
& \sum_{P_j \in P} \int_0^{r_j} F_j(x) dx - \sum_{P_j \in P} \int_0^{r'_j} F_j(x) dx \\
&= \sum_{P_j \in P^{(1)}} \int_{r'_j}^{r_j} F_j(x) dx - \sum_{P_j \in P^{(2)}} \int_{r_j}^{r'_j} F_j(x) dx \\
&\leq \sum_{P_j \in P^{(1)}} F_j(r_j)(r_j - r'_j) - \sum_{P_j \in P^{(2)}} F_j(r_j)(r'_j - r_j) \\
&\leq \sum_{P_j \in P^{(1)}} d(r_j - r'_j) - \sum_{P_j \in P^{(2)}} d(r'_j - r_j) \\
&= d \cdot \sum_{P_j \in P} (r_j - r'_j) \\
&= d \cdot \left(\sum_{P_j \in P} r_j - \sum_{P_j \in P} r'_j \right) \\
&= 0 \\
&\Rightarrow \sum_{P_j \in P} \int_0^{r_j} F_j(x) dx \leq \sum_{P_j \in P} \int_0^{r'_j} F_j(x) dx \quad (10)
\end{aligned}$$

So the cost of traffic assignment R is minimal. \blacksquare

D. Strategy-Proofness

In our scheme, the actions available to each node in the network are to declare its private type. Obviously it is a direct-revelation mechanism. To show it has the incentive compatible (IC) property, we will prove that if our scheme is used, telling the truth is a dominant strategy.

Theorem 2: If OSMA is used, declaring the true type (cost function and available bandwidth) is a dominant strategy for each node.

Proof: We will show that a node v_i can not increase its utility by cheating. That is to say, truth telling is a dominant strategy. If the node v_i is not on any path in P , it will definitely get zero utility. If the node v_i is on one of the path P_j in P , we distinguish three cases:

- 1) The node v_i cheats to increase the amount of traffic passing through itself by $\Delta r_j > 0$. One can achieve this by declaring more available bandwidth, or a cost function that has smaller integral value than the real one on the same interval, or both. The traffic on path $P_k \in P - \{P_j\}$ is decreased by Δr_k (where some Δr_k may be less than or equal to 0). But the node's new

utility $u'_i = p'_i - c'_i$ can not be more than u_i because:

$$\begin{aligned}
u'_i - u_i &= (p'_i - c'_i) - (p_i - c_i) \\
&= (p'_i - p_i) - (c'_i - c_i) \\
&= \left(\sum_{P_k \in P - \{P_j\}} \int_{r_k - \Delta r_k}^{r_k} F_k(x) dx \right. \\
&\quad \left. - \sum_{v_h \in P_j - \{v_i\}} \int_{r_j}^{r_j + \Delta r_j} f_h(C_h - b_h + x) dx \right) \\
&\quad - \int_{r_j}^{r_j + \Delta r_j} f_i(C_i - b_i + x) dx \\
&= \sum_{P_k \in P - \{P_j\}} \int_{r_k - \Delta r_k}^{r_k} F_k(x) dx \\
&\quad - \int_{r_j}^{r_j + \Delta r_j} F_j(x) dx \quad (11)
\end{aligned}$$

Since $\forall P_k \in P - \{P_j\}$, $F_k(r_k) = F_j(r_j)$ when $r_k > 0$ and $\sum_{P_k \in P - \{P_j\}} \Delta r_k = \Delta r_j$, we have $u'_i - u_i \leq 0$.

- 2) The node v_i cheats to decrease the amount of traffic passing through itself by $\Delta r_j > 0$. This can be achieved by declaring less available bandwidth, or a cost function that has larger integral value than the real one on the same interval, or both. The traffic on path $P_k \in P - \{P_j\}$ is increased by Δr_k . The node's new utility $u'_i = p'_i - c'_i$ can not be increased because:

$$\begin{aligned}
u'_i - u_i &= (p'_i - c'_i) - (p_i - c_i) \\
&= (p'_i - p_i) - (c'_i - c_i) \\
&= \left(- \sum_{P_k \in P - \{P_j\}} \int_{r_k}^{r_k + \Delta r_k} F_k(x) dx \right. \\
&\quad \left. + \sum_{v_h \in P_j - \{v_i\}} \int_{r_j - \Delta r_j}^{r_j} f_h(C_h - b_h + x) dx \right) \\
&\quad + \int_{r_j - \Delta r_j}^{r_j} f_i(C_i - b_i + x) dx \\
&= \int_{r_j - \Delta r_j}^{r_j} F_j(x) dx \\
&\quad - \sum_{P_k \in P - \{P_j\}} \int_{r_k}^{r_k + \Delta r_k} F_k(x) dx \quad (12)
\end{aligned}$$

Since $\forall P_k \in P - \{P_j\}$, $F_k(r_k) = F_j(r_j)$ when $r_k > 0$ and $\sum_{P_k \in P - \{P_j\}} \Delta r_k = \Delta r_j$, we have $u'_i - u_i \leq 0$.

- 3) The node v_i cheats, but does not change the amount of traffic passing through itself. This may happen either when the node does not get any share of traffic flow even by cheating, or when some other nodes also cheat, which results in having the traffic assigned to v_i unchanged. Since both the payment to v_i and the cost for forwarding the traffic does not change, the node v_i still gets the same utility as that of truth telling. \blacksquare

Now we consider the utility u_i of each node v_i in each path in P :

$$\begin{aligned} u_i &= p_i - c_i \\ &= \sum_{P_k \in P - \{P_j\}} \int_0^{r'_k} F_k(x) dx - \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx. \end{aligned} \quad (13)$$

By Theorem 1, we have that

$$u_i = \sum_{P_k \in P - \{P_j\}} \int_0^{r'_k} F_k(x) dx - \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx \geq 0. \quad (14)$$

Hence, $u_i \geq 0$. So we can see that participating in the game, a node will get non-negative utility under our scheme. If a node stays out of the game, its utility will remain to be 0. So participating is not worse than staying out, which satisfies the individual rationality (IR).

Since our scheme satisfies both IC and IR, we have the following theorem:

Theorem 3: OSMA is a strategy-proof mechanism.

E. Payment-cost Ratio

To ensure the strategy-proofness, OSMA requires the source node S to pay more than the actual cost. In this subsection, we study the ratio of the payment to the cost. We define this ratio as follows:

$$\mathcal{R} = \frac{\sum_{P_j \in P} \sum_{v_i \in P_j} p_i}{\sum_{P_j \in P} \int_0^{r_j} F_j(x) dx}. \quad (15)$$

Theorem 4: Suppose f_{max} and f_{min} are the maximal and minimal value of the cost function $f_i(x)$, $x \in [0, C_i]$ for all nodes, respectively.

$$f_{max} = \max\{f_i(x) | i \in N, x \in [0, C_i]\}, \quad (16)$$

$$f_{min} = \min\{f_i(x) | i \in N, x \in [0, C_i]\}. \quad (17)$$

Then the payment-cost ratio has an upper bound:

$$\mathcal{R} \leq \frac{f_{max}}{f_{min}} \cdot n^2. \quad (18)$$

Proof: The sum of payments to the nodes in path $P_j \in P$ is:

$$\begin{aligned} \bar{p}_j &= \sum_{v_i \in P_j} p_i \\ &= \sum_{v_i \in P_j} \left(\sum_{P_k \in P - \{P_j\}} \int_0^{r'_k} F_k(x) dx \right. \\ &\quad \left. - \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx + \int_0^{r_j} f_i(C_i - b_i + x) dx \right) \\ &= |P_j| \cdot \sum_{P_k \in P - \{P_j\}} \int_0^{r'_k} F_k(x) dx \\ &\quad - |P_j| \cdot \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx + \int_0^{r_j} F_j(x) dx \\ &\leq q \cdot f_{max} \cdot |P_j| \cdot \left(\sum_{P_k \in P - \{P_j\}} |P_k| + 1 \right) \\ &\quad - |P_j| \cdot \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx. \end{aligned} \quad (19)$$

Then the overall payment is:

$$\begin{aligned} &\sum_{P_j \in P} \bar{p}_j \\ &= q \cdot f_{max} \cdot \sum_{P_j \in P} \left(|P_j| \cdot \left(\sum_{P_k \in P - \{P_j\}} |P_k| + 1 \right) \right) \\ &\quad - \sum_{P_j \in P} \left(|P_j| \cdot \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx \right) \\ &\leq q \cdot f_{max} \cdot n \cdot \sum_{P_j \in P} |P_j| \\ &\quad - \sum_{P_j \in P} \left(|P_j| \cdot \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx \right) \\ &\leq q \cdot f_{max} \cdot n^2 \\ &\quad - \sum_{P_j \in P} \left(|P_j| \cdot \sum_{P_k \in P} \int_0^{r_k} F_k(x) dx \right). \end{aligned} \quad (20)$$

So the upper bound of the payment-cost ratio can be calculated as:

$$\begin{aligned} \mathcal{R} &= \frac{q \cdot f_{max} \cdot n^2}{\sum_{P_j \in P} \int_0^{r_j} F_j(x) dx} - \sum_{P_j \in P} |P_j| \\ &\leq \frac{q \cdot f_{max} \cdot n^2}{q \cdot f_{min}} \\ &= \frac{f_{max}}{f_{min}} \cdot n^2. \end{aligned} \quad (21)$$

■

V. EVALUATIONS

In this section, we first evaluate the total cost and payment-cost ratio for sending traffic flows using OSMA. Then we verify strategy-proofness of OSMA. Furthermore, communication overhead and computation overhead are also evaluated.

We evaluate OSMA using GloMoSim [2], and implement our traffic assignment scheme based on DSR [1] protocol. Note that, since we study multi-path routing in this paper, in our experiments we do not directly use the shortest path found by DSR. Instead, we select node disjoint paths from DSR's route cache and our algorithm assigns the traffic flow on these selected paths. We consider a random wireless network with 100 nodes randomly distributed in a terrain area of 500 by 500 meters. Nodes use IEEE 802.11 (at 11Mbps) as the MAC layer protocol. The radio range is set to 140.5 meters. On average, there are 5.1 paths found for each source-destination pair in the evaluation. We assume that each node has capacity 1Mbps. Each node's cost function can be one of following four different increasing cost functions: linear function $f(x) = 4x + 5$, quadratic function $f(x) = x^2 + 1$, reciprocal function $f(x) = \frac{100}{C-x}$, and exponential function $f(x) = e^{x/2}$. The coefficients are randomly selected in this simulation. In practice the type of cost function and the coefficients should be determined by the characteristics of the network. The bandwidth of generated traffic requests follows the normal distribution with mean 0.5Mbps and $\sigma^2 = 2$.

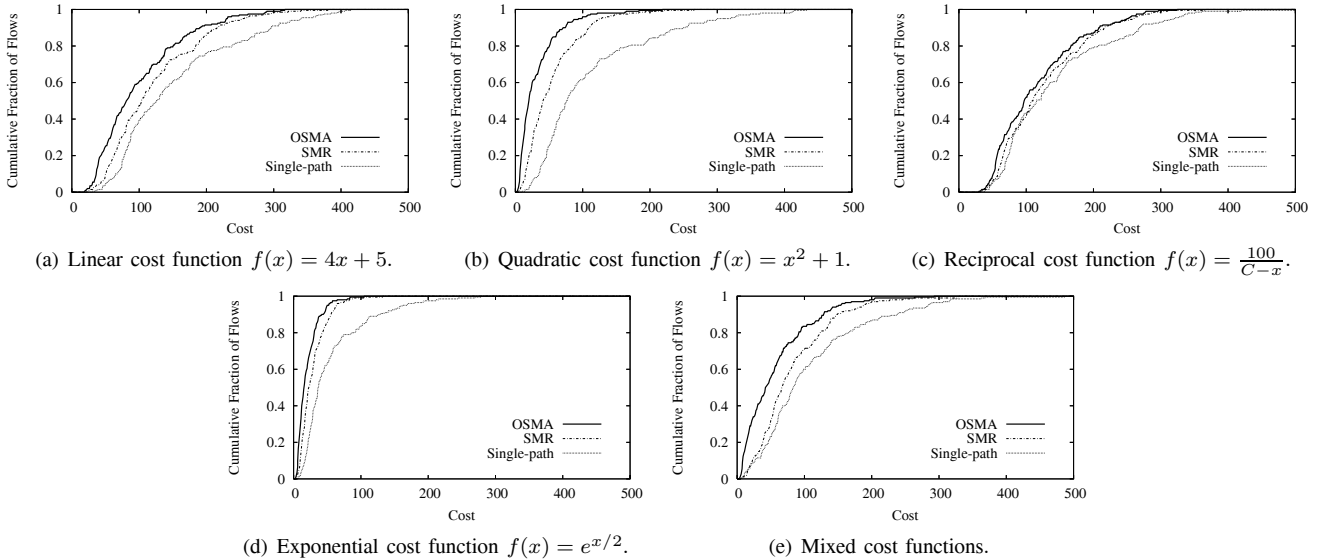


Fig. 1. Cumulative distribution function (CDF) of the forwarding cost for multi-path routing with our traffic assignment scheme vs. SMR and simple-path routing on 200 transmissions with different source-destination pairs. In the first four cases, all the nodes use the same cost function; while in the last case (mixed case), each node's cost function can be anyone of the four. The x-axis is the forwarding cost of the flow. The y-axis label is cumulative fraction of flows, which means cumulative distribution function (CDF) of flows.

A. Forwarding Cost

In section IV-C, we proved that OSMA always computes the most cost efficient traffic assignment given nodes' true types. Here we experimentally compare the forwarding cost of multi-path routing with OSMA vs. that of Split Multipath Routing (SMR) [19] and simple-path routing, which assigns all the traffic flow to the path with the lowest cost. We do the comparison in five cases. In the first four cases, all the nodes use the same cost function (one of the four cost functions listed above for a case); while in the last case (mixed case), each node's cost function can be anyone of the four.

Figure 1 shows cumulative distribution function (CDF) of the forwarding cost for multi-path routing with OSMA vs. SMR and simple-path routing on 200 transmissions with different source-destination pairs. The x-axis is the forwarding cost of the flow. The y-axis label is cumulative fraction of flows, which means cumulative distribution function (CDF) of flows. It is shown that multi-path routing with OSMA achieves significantly lower forwarding cost than that of simple-path routing, no matter in which case. Specifically, at the median points, multi-path routing with OSMA saves 22.1% (40.0%), 51.2% (75.3%), 11.8% (23.6%), 30.4% (60.0%), and 37.3% (53.2%) cost compared with SMR (single-path routing) in the five cases, respectively. In the case of mixed cost functions, 91.5% transmissions' forwarding costs are smaller than 130 using multi-path routing with our scheme, while only 81.5% (70.5%) transmissions' forwarding costs are smaller than 130 using SMR (single-path routing), which is 16.5% (21.0%) lower.

B. Payment-cost Ratio

In section IV-E we gave the upper bound of payment-cost ratio. Here we present the numerical results from experiments on payment-cost ratio. Figure 2 shows the CDF of the payment-cost ratio on 200 transmissions with different source-destination pairs in five different cases. Same as before, in the

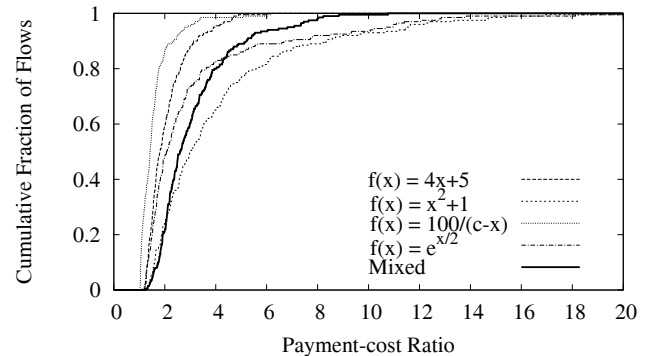


Fig. 2. CDF of the payment-cost ratio on 200 transmissions with different source-destination pairs in five different cases. In the first four cases, all the nodes use the same cost function; while in the last case (mixed case), each node's cost function can be anyone of the four.

first four cases, all the nodes use the same cost function; while in the last case (mixed case), each node's cost function can be anyone of the four. We can see that most of the payment-cost ratios remain small. Specifically, 90% transmissions' payment-cost ratios are no larger than 3.13, 8.34, 2.30, 6.68, and 5.18 in the five cases, respectively.

C. Effect of Deviating

In this section, we show our evaluation results on the effect of deviating from OSMA. We keep track of a player and record her utility in the two cases: following our scheme or deviating from it.

Figure 3 shows a tracked player's utility got by following the traffic assignment scheme and deviating from it on 100 transmissions with different source-destination pairs in the case of mixed cost function. It is shown that, when following our scheme, the player can always obtain non-negative utility. Furthermore, the utility obtained by following the scheme is

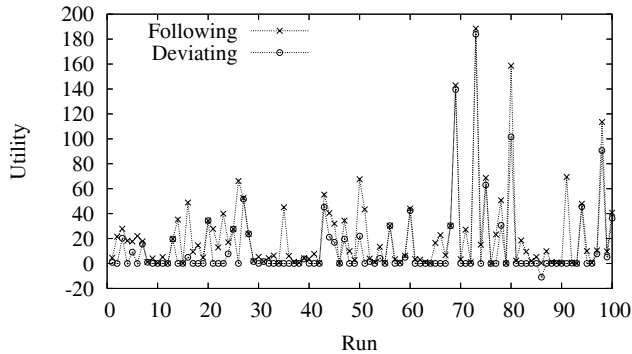


Fig. 3. Utilities got by following the traffic assignment scheme vs. deviating from it on a tracked player in 100 transmissions with different source-destination pairs. Each node's cost function can be any one of the four defined above.

always higher than by deviating from it. This will motivate each player to follow our scheme.

D. Evaluation on Efficiency

We evaluate the efficiency of OSMA in terms of communication overhead and computation overhead. For communication, it is the overhead induced by OSMA for nodes to declare their type (cost function and current available bandwidth). Note that OSMA is based on existing multi-path routing protocol. In other words, it efficiently assigns traffic to the paths obtained by underlying multi-path routing protocol. So the computation overhead we measured does not include the overhead of the underlying protocol. Therefore, if we compare the complexity of our algorithm with another algorithm, we need to compare it with another incentive compatible traffic assignment scheme, not a multipath routing algorithm. However, to the best of our knowledge, there is no such scheme proposed in the literature so far. Our cost functions are very general functions. It can be the expected transmission cost when the network is lossy. For computation, it is the overhead for computing traffic assignment using Algorithm 1.

Theoretically, the overall communication overhead is $N_p N_h (L_f + L_b)$ bytes, where N_p is the number of node-disjoint paths from S to D , N_h is the average number of hops of these paths, and L_f and L_b are the numbers of bytes needed to encode the cost function and available bandwidth, respectively. The most time-consuming part of our algorithm is the binary search. Binary searching 1Gbps bandwidth needs no more than 30 iterations, while binary searching the entire range of real number that represented by type double in C++ language requires no more than 2047 iterations. Thus, the complexity of our algorithm is $\mathcal{O}(n^2)$.

In the evaluation, we assume encoding length for cost function and available bandwidth are 32 bytes and 4 bytes, respectively. We vary the number of paths from 2 to 8, and vary the average hops from 1 to 16. For each case, we repeat the evaluation 1000 time and calculate average communication overhead.

Figure 4 demonstrates the communication overhead using our scheme. Even in the extreme case of 8 paths and 16 average hops, the communication overhead is still less than 5KB. So the communication overhead is very light.

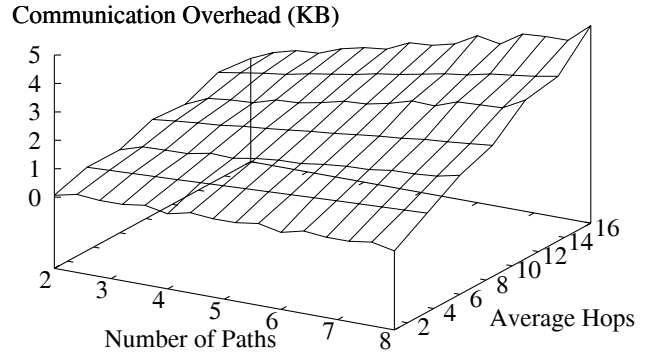


Fig. 4. Communication overhead, using 32 bytes to encode cost function and 4 bytes to encode available bandwidth.

To evaluate the computation overhead, we run our traffic assignment scheme on a laptop with 1.4GHz Centrino CPU and 768MB memory. We also repeat each evaluation 1000 times, and calculate average computation overhead.

Figure 5 shows the computation overhead of two typical cases. From Figure 5(a) and Figure 5(b), we can see that the computation overhead increases along with the number of paths and the average number of hops. However, regardless of which cost function is used, the computation overhead remains very low. For 10 average hops, all the calculation are guaranteed to be finished in less than 30 milliseconds.

We also compare data split and reassembly cost of OSMA and SMR at source and destination for transmitting 15MB data with 1500 bytes per packet. Figure 6 shows our results. Figure 6(a) shows that data split cost of OSMA increases with the number of paths, while that of SMR remains almost the same. This is because OSMA uses every path available, while SMR uses at most two paths. Although data split cost of OSMA is slightly higher than that of SMR, data reassembly costs of OSMA and SMR are almost the same. Figure 6(b) shows that the length of path does not have much impact on data split and reassembly cost.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a game-theoretic solution, namely OSMA, for multi-path routing to deal with the selfish behavior of nodes. OSMA can be used to update any existing multi-path routing protocol that schedules traffic among node-disjoint paths such that the protocol becomes incentive compatible. Our evaluations verify that OSMA is optimal and strategy-proof, and demonstrate that the scheme has very low communication and computation overhead.

There are several potential ways to extend our work. This paper study the traffic assignment problem in case of multi-path unicast routing. So one possibility is to consider how to extend it to deal with multicast. To guarantee the strategy-proofness, the payment may much higher than the real forwarding cost. So another possibility is to study how to lower the overpayment. However, we leave these topics to future study.

REFERENCES

- [1] The Dynamic Source Routing Protocol (DSR), <http://tools.ietf.org/html/rfc4728>.

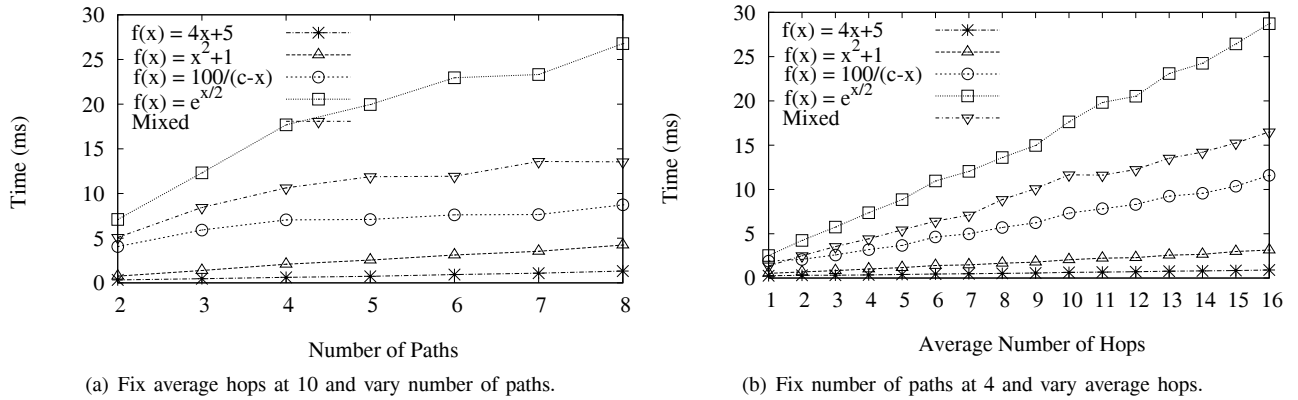


Fig. 5. Computation overhead, using 4 kinds of cost functions, varying number of available paths and average number of hops.

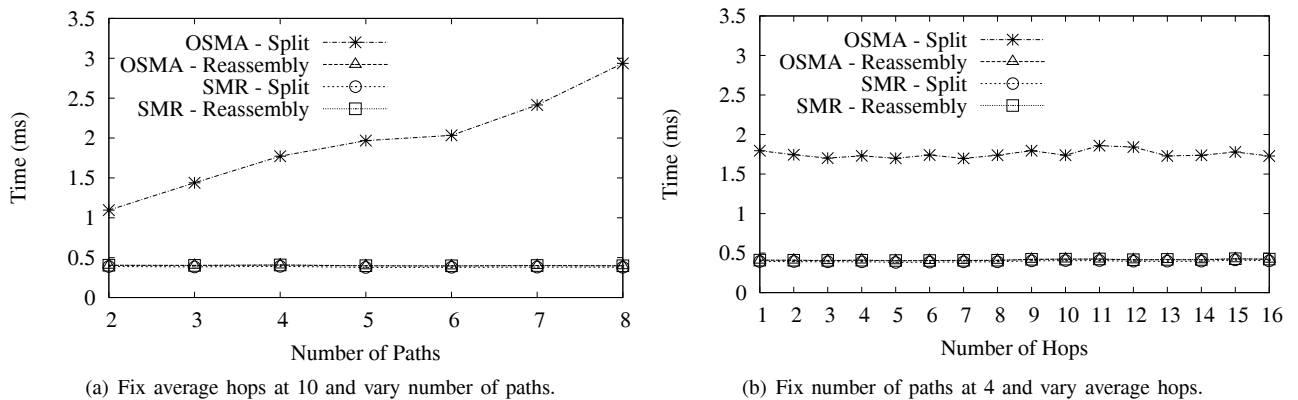


Fig. 6. Data split and reassembly cost with varying number of available paths and average number of hops.

- [2] GloMoSim. <http://pcl.cs.ucla.edu/projects/gloimosim/>.
- [3] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *Proc. Ninth International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, CA, Sep. 2003.
- [4] D. Bertsekas and R. Gallager, *Data Networks, 2nd Edition*. Prentice Hall, 1992.
- [5] D. P. Bertsekas, "Algorithms for nonlinear multicommodity network flow problems," in *International Symposium on Systems Optimization and Analysis*, Rocquencourt, France, Dec. 1978.
- [6] D. P. Bertsekas, E. M. Gafni, and R. G. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 911–919, Aug. 1984.
- [7] S. Buchegger and J.-Y. L. Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks," in *Proc. Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP)*, Canary Islands, Spain, Jan. 2002.
- [8] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: fairness in dynamic ad-hoc networks)," in *Proc. Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002.
- [9] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 885–896, 1999.
- [10] CORPORATE The ATM Forum, ATM user-network interface specification (version 3.0).
- [11] S. Eidenbenz, G. Resta, and P. Santi, "Commit: a sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes," in *Proc. 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc & Sensor Networks (IPDPS '05)*, Apr. 2005.
- [12] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based mechanism for lowest-cost routing," in *Proc. 21st Symposium on Principles of Distributed Computing (PODC '02)*, July 2002.
- [13] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker, "Sharing the cost of multicast transmissions," *J. Computer and System Sciences*, vol. 63, no. 1, pp. 21–41, 2001.
- [14] M. Félegyházi and J.-P. Hubaux, "Wireless operators in a shared spectrum," in *Proc. IEEE INFOCOM '06*, Apr. 2006.
- [15] E. M. Gafni, "Convergence of a routing algorithm," master's thesis, University of Illinois, Urbana, IL, 1979.
- [16] R. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. 25, no. 1, pp. 73–85, Jan. 1977.
- [17] M. M. Halldósson, J. Y. Halpern, L. E. Li, and V. S. Mirrokni, "On spectrum sharing games," in *Proc. 23rd Annual ACM SIGACT-SIGOPS Symposium on Principles Of Distributed Computing (PODC '04)*, July 2004.
- [18] J. Huang, R. Berry, and M. Honig, "Auction-based spectrum sharing," *ACM/Springer J. Mobile Networks and Applications (MONET)*, special issue on WiOpt'04, vol. 11, pp. 405–418, June 2006.
- [19] S. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. IEEE International Conference on Communications (ICC '01)*, 2001.
- [20] S.-J. Lee and M. Gerla, "AODV-BR: backup routing in ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '00)*, Sep. 2000.
- [21] M. Marina and S. Das, "On-demand multi path distance vector routing in ad hoc networks," in *Proc. 9th International Conference on Network Protocols (ICNP '01)*, Nov. 2001.
- [22] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. ACM MobiCom'00*, pp. 255–265, Aug. 2000.
- [23] J. Moy, OSPF (version 2), RFC 2328, 1998.
- [24] N. Nisan and A. Ronen, "Algorithmic mechanism design," in *Proc. STOC'99*, pp. 129–140, May 1999.
- [25] M. J. Osborne and A. Rubenstein, *A Course in Game Theory*. The MIT Press, 1994.
- [26] P. Papadimitratos, Z. Haas, and E. Sirer, "Path-set selection in mobile ad hoc networks," in *Proc. Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, June 2002.
- [27] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM '97*, Apr. 1997.

- [28] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," in *Proc. 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '00)*, Aug. 2000.
- [29] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," RFC 4271, 2006.
- [30] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," in *Proc. Fourth ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, June 2003.
- [31] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in wireless ad hoc networks," in *Proc. IEEE INFOCOM '03*, Mar. 2003.
- [32] H. Suzuki and F. A. Tobagi, "Fat bandwidth reservation scheme with multi-link and multi-path routing in atm networks," in *Proc. IEEE INFOCOM '92*, May 1992.
- [33] W. Wang, S. Eidenbenz, Y. Wang, and X. Y. Li, "OURS: optimal unicast routing systems in non-cooperative wireless networks," in *Proc. Twelfth International Conference on Mobile Computing and Networking (Mobicom '06)*, Sep. 2006.
- [34] W. Wang and B. Li, "Market-driven bandwidth allocation in selfish overlay networks," in *Proc. IEEE INFOCOM '05*, Mar. 2005.
- [35] W. Wang, X. Y. Li, and Y. Wang, "Truthful multicast in selfish wireless networks," in *Proc. Tenth International Conference on Mobile Computing and Networking (Mobicom '04)*, Sep. 2004.
- [36] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *Proc. IEEE INFOCOM '03*, Mar. 2003.
- [37] S. Yuen and B. Li, "Strategyproof mechanisms for dynamic multicast tree formation in overlay networks," in *Proc. IEEE INFOCOM '05*, Mar. 2005.
- [38] S. Zhong, J. Chen, and Y. R. Yang, "Sprite, a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM '03*, Mar. 2003.
- [39] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks," in *Proc. Eleventh International Conference on Mobile Computing and Networking (Mobicom '05)*, Aug. 2005.
- [40] S. Zhong and F. Wu, "On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks," in *ACM MobiCom '07*, Sep. 2007.



Fan Wu is a post doctoral research associate at Department of Electrical and Computer Engineering of University of Illinois at Urbana-Champaign. He received BS degree in Computer Science from Nanjing University, in 2004, and Ph.D. degree in Computer Science and Engineering from University at Buffalo, the State University of New York, in 2009. His research interests include algorithmic game theory, economic incentives for wireless networks and peer-to-peer computing.



Sheng Zhong is an assistant professor at the Department of Computer Science and Engineering of University at Buffalo, the State University of New York. He received his BS (1996), ME (1999) from Nanjing University, and PhD (2004) from Yale University, all in Computer Science. His research interests include economic incentives and data privacy.



Jiqiang Liu is an associate professor at the Department of Computer and Information Technology of Beijing Jiaotong University. He received his BS (1994) and Ph.D. (1999) from Beijing Normal University. His research interests include trusted computing and privacy preserving.