Cost-effective Traffic Assignment for Multipath Routing in Selfish Networks

Fan Wu and Sheng Zhong
Department of Computer Science and Engineering
SUNY at Buffalo, USA
Email: {fwu2, szhong}@cse.buffalo.edu

Jiqiang Liu
School of Computer and Information Technology
Beijing Jiao Tong University, China
Email: jqliu@bjtu.edu.cn

***Abstract*—Multipath routing has long been studied as an important routing strategy in networks. Many multipath routing protocols schedule traffic among multiple paths in order to distribute traffic load. However, existing multipath 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 a traffic assignment scheme to deal with the selfish behavior, which is proved to be strategy-proof. Under our scheme, behaving cooperatively is to the best interest of every node. Extensive evaluations are carried out to show that our scheme has good performance.**

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 multipath routing can achieve route resilience, higher aggregate bandwidth, smaller end-to-end delays, and better load balancing [2], [19].

Multipath routing has been explored in both wired and wireless networks. In wired network, multipath routing is implemented as a feature of Asynchronous Transfer Mode (ATM) networks [3] and Open Shortest Path First (OSPF) protocol [12]. For Mobile Ad-Hoc Network (MANET), multipath routing is also extensively studied in recent years. A number of multipath routing protocols for MANETs have been proposed. Some of them [9], [11], [15], [23] maintain multiple routes and utilize these routes only when the primary root fails. Others [10], [14], [16] further schedule traffic among multiple paths in order to distribute load. In this paper, we are mainly concerned with the latter, i.e., multipath routing protocols that assign the traffic among the multiple paths.

We note that the existing multipath routing protocols with traffic assignment require that all nodes in the network follow the prescribed protocol and cooperate with each other. However, this assumption is not valid when the network consists of selfish nodes [1], [4]–[8], [17], [18], [20]–[22], [24]–[26]. 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 incentive 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 multipath routing protocol such that selfish nodes will behave cooperatively.

To the best of our knowledge, there has not been any work addressing selfish behavior for multipath 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, various incentive-based approaches have been proposed to solve packet routing or forwarding problems [1], [4], [17], [18], [20], [21], [25], [26]. Wang et al. [22] and Yuen et al. [24] investigated the problem of bandwidth allocation and multicast tree formation in overlay networks. Feigenbaum et al. [5], [6] considered both unicast and multicast in Internet. Felegyhazi et al. [7] and Halldorsson et al. [8] studied the problem of sharing spectrum using game theory.

Although the methods mentioned above can not be directly used in the multipath routing scenario, we believe that we can develop a game-theoretic solution for multipath routing that can deal with the selfish behavior of nodes. To design a multipath routing protocol for selfish networks, instead of starting from scratch, we consider some existing multipath 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 multipath routing protocol between the source and the destination, such that the participating selfish nodes will behave cooperatively. First, we give a game-theoretic model for this problem, which we call traffic assignment game. Then, we propose an efficient scheme for traffic assignment, which is shown to be *strategy-proof* in the above model. Here intuitively, the scheme being strategy-proof means that behaving cooperatively is to the best interest of every node, regardless of other nodes' behavior. Furthermore, our scheme is guaranteed to compute the lowest cost traffic assignment. Evaluations demonstrate that our scheme 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 our traffic assignment scheme and prove its optimality and strategy-proofness. In Section V, we show the results of evaluations. Finally, we conclude the paper 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 $N = \{1, 2, \ldots, n\}$. 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 $t = (t_1, t_2, \ldots, t_n)$. For each player i, there is a set of

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

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}. The action profile a decides the outcome $o(a)$ and payment $p(a)$ of the game, where $p(a) = (p_1(a), p_2(a), \ldots, p_n(a))$ 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:

$$u_i(a) = v_i(o(a)) + p_i(a). \tag{1}$$

Given above notations, now we can define a very strong solution concept called *dominant strategy* [13].

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},

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

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.

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 $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E = \{e_1, e_2, \ldots, e_m\} \subseteq V \times V$ 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.

We model multipath 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 $t_i =< f_i(x), b_i >$. The cost function $f_i(x)$ is a increasing function indicating the cost of forwarding one unit of traffic when x units of bandwidth has been used. The cost includes expense of power consumed, losing 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 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

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

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 $p_i(a)$. So the utility of the node is:

$$u_i(a) = p_i(a) - c_i(o(a)). \tag{3}$$

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

IV. Traffic Assignment Scheme

In this section, we propose our traffic assignment scheme and prove its optimality and strategy-proofness. Our scheme is designed for assigning traffic among multiple node-disjoint paths, which do not have any nodes in common except the source and destination. So our scheme can be used for any multipath routing protocol that schedules multiple node-disjoint paths (*e.g.*, [11], [12], [23]). 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. 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., [1], [4], [17], [20], [26]).

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 $P = \{P_1, P_2, \ldots, P_m\}$ found by the multipath routing protocol (*e.g.*, [11], [12], [23]). 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 $< f_i(x), b_i >_{v_i \in P_j \in P}$, and bandwidth requirement q.
Output: Traffic assignment $R = (r_1, r_2, \ldots, r_m)$.

$R = 0^m$.
$W = \varnothing$.
$\forall P_j \in P$, define $F_j(x) = \sum_{v_i \in P_j} f_i(C_i - b_i + x)$.
while $q > \sum_{j=0}^{m} r_j$ **do**
  if $(P \neq \varnothing)$ **then**
    $k = \underset{P_j \in P}{argmin} F_j(0)$.
    Move P_k from P to W.
    $t = min\{F_j(0) | P_j \in P\}$.
  else
    $t = MAX$.
  end if
  if $\sum_{P_j \in W} F_j^{-1}(t) \leq q$ **then**
    $r_j = F_j^{-1}(t), \forall P_j \in W$.
  else
    Use binary search to find $0 \leq c \leq t$, s.t., $\sum_{P_j \in W} F_j^{-1}(c) = q$.
    $r_j = F_j^{-1}(c), \forall P_j \in W$.
  end if
end while

After collecting all the information, the source node (or the destination node) computes the traffic assignment using Algorithm 1. Algorithm 1 first combines the cost functions declared by nodes on each path $P_i \in P$ to get a integrated path cost function $F_i(x)$. Then it turns to work on these integrated path cost functions. In each iteration, a path with the lowest "marginal" cost is selected and added to W; and as much as possible traffic is assigned to all selected paths in W. When the selected paths are no longer cost efficient (i.e., enough traffic has been added) and there 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. When the iteration stops, the vector R is the final assignment of traffic.

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:

$$\begin{aligned} p_i = & \sum_{P_k \in P - \{P_j\}} \int_{r_k}^{r'_k} F_k(x)dx \\ & - \sum_{v_h \in P_j - \{v_i\}} \int_0^{r_j} f_h(C_h - b_h + x)dx. \end{aligned} \tag{4}$$

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.

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

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

C. Optimality

Theorem 1: Our traffic assignment scheme 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:

$$\sum_{P_j \in P} r'_j = \sum_{P_j \in P} r_j = q$$

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

$$\begin{cases} \forall P_j \in P^{(1)}, r_j \geq r'_j, \\ \forall P_j \in P^{(2)}, r_j < r'_j. \end{cases}$$

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

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

So the cost of traffic assignment R is minimal. ■

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 our scheme 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:

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

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

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. Similarly, we can prove that the node's utility can not be increased.
3) The node v_i cheats, but does not change the amount of traffic passing through itself. 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.

■

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

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

Since the traffic assignment computed by our algorithm is optimized, we have $u_i \geq 0$. 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: The traffic assignment scheme presented in this paper is a strategy-proof mechanism.

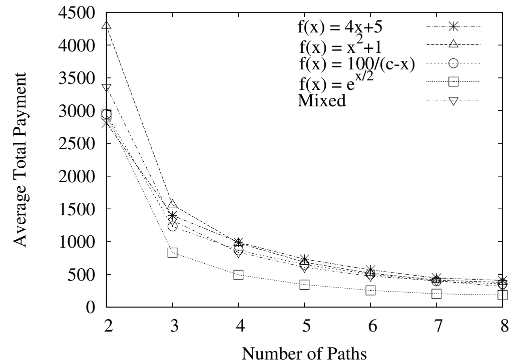
V. Evaluations

In this section, we evaluate the total payment for sending a traffic flow. Furthermore, communication overhead and computation overhead are also evaluated. (Due to limitation of space, we recommend readers to refer to the journal version of this paper for evaluations on the optimality and strategy-proofness, which were proved theoretically.)

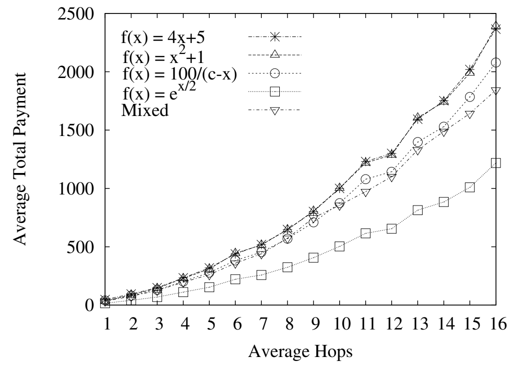
In the evaluation, we assume each node has capacity 10Mbps and available bandwidth uniformly distributed between 5.0Mbps and 10.0Mbps. Each node's cost function can be one of following four different increasing cost functions: linear function $f(x) = 4x + 5$, quadratic function $f(x) = x^2 + 1$, reciprocal function $f(x) = \frac{100}{C-x}$, and exponential function $f(x) = e^{x/2}$. We generate traffic requests randomly between 0 and 5.0Mbps.

A. Evaluation on Total Payment

In section IV-C, we proved that our scheme always computes the most cost efficient traffic assignment given nodes' true types. Here we evaluate how the total payment for sending a traffic flow is affected by the number of paths, the average number of hops, and the cost functions. We do two evaluations on total payment. In both evaluations, we estimate the effect of cost function in 5 cases. In the first 4 cases, all the nodes use the same cost function (one cost function listed above for a case); while in the fifth case (mixed case), each node is allowed to choose a cost function from the the four randomly. In the first evaluation, we fix the average number of hops at 10 and vary the number of paths from 2 to 8. In the second evaluation, we fix the number of paths at 4 and vary the average number of hops from 1 to 16. We repeat each evaluation 1000 times and compute the average total payment.



(a) Fix average hops at 10 and vary number of paths.



(b) Fix number of paths at 4 and vary average hops.

Fig. 1. Average total payment using 4 kinds of cost functions.

Figure 1 shows the results of the evaluations. From Figure 1(a), we can see that no matter which cost function is used, the average total payment falls sharply when the number of paths goes from 2 to 4; and it decreases more and more slowly while the number of paths increases. Intuitively, more paths compete to share the traffic flow will lower the total payment; but too many paths can not further reduce the total payment, since many paths will get no share of traffic flow in this case. Figure 1(b) shows that the total payment grows along with the average hops. Intuitively, the longer the paths are, the more payments are needed to send the traffic flow.

B. Evaluation on Efficiency

We evaluate the efficiency of our scheme in terms of communication overhead and computation overhead.

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.

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.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

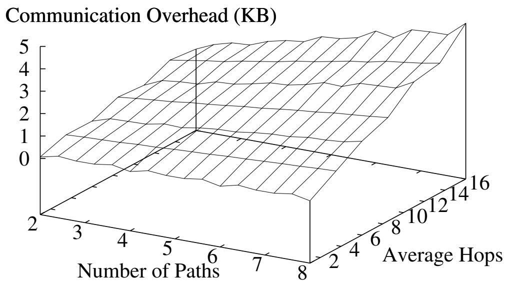
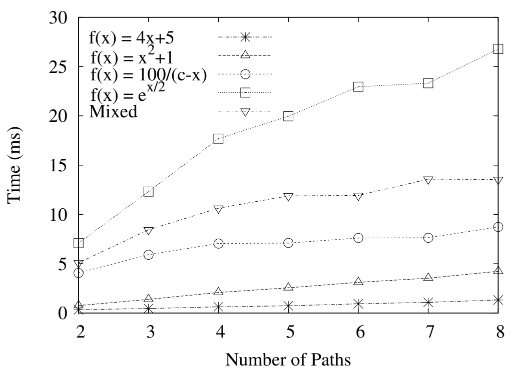


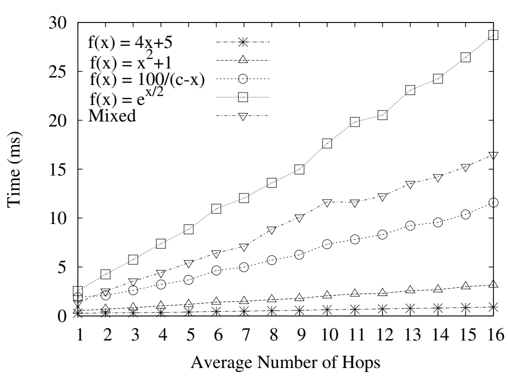
Fig. 2. Communication overhead, using 32 bytes to encode cost function and 4 bytes to encode available bandwidth.

Figure 2 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.

To evaluate the computation overhead, we run our traffic assignment scheme on a laptop with 1.4GHz Centrino CPU and 768MB memory. The setup here is the same as Section V-A. We also repeat each evaluation 1000 times, and calculate average computation overhead.



(a) Fix average hops at 10 and vary number of paths.



(b) Fix number of paths at 4 and vary average hops.

Fig. 3. Computation overhead, using 4 kinds of cost functions, varying number of available paths and average number of hops.

Figure 3 shows the computation overhead of two typical cases. From Figure 3(a) and Figure 3(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.

VI. Conclusion

In this paper, we propose a game-theoretic solution for multipath routing to deal with the selfish behavior of nodes. Our scheme can be used to update any existing multipath routing protocol that schedules traffic among node-disjoint paths such that the protocol becomes incentive compatible. Our evaluations show that our scheme has good performance.

Acknowledgment

This work was supported by NSF CNS-0524030.

References

[1] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: a truthful and cost-eftcient routing protocol for mobile ad hoc networks with seltsh agents," in *ACM MobiCom*, Jun. 2003.
[2] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Tran. on Networking*, vol. 7, no. 6, pp. 885–896, 1999.
[3] CORPORATE The ATM Forum, *ATM user-network interface specification (version 3.0)*.
[4] 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 *IEEE IPDPS*, Apr. 2005.
[5] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based mechanism for lowest-cost routing," in *ACM PODC*, Jul. 2002.
[6] 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.
[7] M. Felegyhazi and J.-P. Hubaux, "Wireless Operators in a Shared Spectrum," in *IEEE INFOCOM*, Apr. 2006.
[8] M. M. Halldósson, J. Y. Halpern, L. E. Li, and V. S. Mirrokni, "On spectrum sharing games," in *ACM PODC*, Jul. 2004.
[9] S. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in *IEEE WCNC*, Sep. 2000.
[10] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *IEEE ICC*, 2001.
[11] M. Marina and S. Das, "On-demand multi path distance vector routing in ad hoc networks," in *IEEE ICNP*, Nov. 2001.
[12] J. Moy, "OSPF (version 2), RFC 2328," 1998.
[13] M. J. Osborne and A. Rubenstein, *A Course in Game Theory*. The MIT Press, 1994.
[14] P. Papadimitratos, Z. Haas, and E. Sirer, "Path-set selection in mobile ad hoc networks," in *ACM MobiHoc*, Jun. 2002.
[15] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *IEEE INFOCOM*, Apr. 1997.
[16] 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 *ACM MobiHoc*, Aug. 2000.
[17] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," in *ACM MobiHoc*, Jun. 2003.
[18] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in wireless ad hoc wireless networks," in *IEEE INFOCOM*, Mar. 2003.
[19] H. Suzuki and F. A. Tobagi, "Fat bandwidth reservation scheme with multi-link and multi-path routing in atm networks," in *IEEE INFOCOM*, May 1992.
[20] W. Wang, S. Eidenbenz, Y. Wang, and X. Y. Li, "OURS: Optimal unicast routing systems in non-cooperative wireless networks," in *ACM MobiCom*, Sep. 2006.
[21] W. Wang, X. Y. Li, and Y. Wang, "Truthful multicast in selfish wireless networks," in *ACM MobiCom*, Sep. 2004.
[22] W. Wang and B. Li, "Market-driven bandwidth allocation in selfish overlay networks," in *IEEE INFOCOM*, Mar. 2005.
[23] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *IEEE INFOCOM*, Mar. 2003.
[24] S. Yuen and B. Li, "Strategyproof mechanisms for dynamic multicast tree formation in overlay networks," in *IEEE INFOCOM*, March 2005.
[25] S. Zhong, J. Chen, and Y. R. Yang, "Sprite, a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *IEEE INFOCOM*, Mar. 2003.
[26] 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 *ACM MobiCom*, Aug. 2005.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.