Truthful Multicast in Selfish Wireless Networks

Weizhao Wang
Dept. of Computer Science
Illinois Institute of Technology, Chicago
wangwei4@iit.edu

Xiang-Yang Li *
Dept. of Computer Science
Illinois Institute of Technology, Chicago
xli@cs.iit.edu

Yu Wang
Dept. of Computer Science
University of North Carolina at Charlotte
wangyu@ieee.org

ABSTRACT

In wireless networks, it is often assumed that each individual wireless terminal will faithfully follow the prescribed protocols without any deviation– except, perhaps, for a few faulty or malicious ones. Wireless terminals, when owned by individual users, will likely do what is the most beneficial to their owners, i.e., act "selfishly". Therefore, an algorithm or protocol intended for selfish wireless networks must be designed.

In this paper, we specifically study how to conduct efficient multicast in *selfish* wireless networks. We assume that each wireless terminal or communication link will incur a cost when it transits some data, and the cost is known to the wireless terminal or communication link itself only. Traditionally, the VCG mechanism has been the *only* method to design protocols so that each selfish agent will follow the protocols for its own interest to maximize its benefit. The main contributions of this paper are two-folds. First, for each of the widely used multicast structures, we show that the VCG based mechanism does not guarantee that the selfish terminals will follow the protocol. Second, we design *first* multicast protocols without using VCG mechanism such that each agent maximizes its profit when it truthfully reports its cost.

Extensive simulations are conducted to study the practical performances of the proposed protocols regarding the actual network cost and total payment.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing Protocols; G.2.2 [**Graph Theory**]: Network problems, Graph algorithms.

General Terms

Algorithms, Design, Economics, Theory.

Keywords

Wireless ad hoc networks, selfish, mechanism design, pricing.

*The work of the author is partially supported by NSF CCR-0311174.

1. INTRODUCTION

In wireless ad hoc networks, each terminal contributes its local resources to forward the data for other terminals to serve the common good, and benefit from resources contributed by other terminals to rout its packets in return. Based on such a fundamental design philosophy, wireless ad hoc networks provide appealing features such as enhanced system robustness, high service availability and scalability. However, the critical observation that individual users who own these wireless devices are generally selfish and non-cooperative may severely undermine the expected performances of the wireless networks. For example, it is traditionally and conveniently assumed that each wireless device will participate in the routing when it is required by the prescribed routing protocols. However, the limitation of energy supply, memory and computing resources of these wireless devices raise concerns about this traditional belief. A wireless device may prefer not participating in the routing to save its energy and resources. Therefore, if we assume that all users are selfish, providing incentives to wireless terminals is a must to encourage contribution and thus maintain the robustness and availability of wireless networking systems. The question turns to how the incentives are designed. Consider a unicast routing protocol based on the least cost path (LCP): each terminal is asked to declare its cost of forwarding a unit data for other terminals, and the least cost path connecting the source and the target terminal is then selected. A very naive incentive is to pay each wireless terminal its declared cost. However, the individual wireless terminal may declare an arbitrarily high cost for forwarding a data packet to other terminals hoping to increase its payment. Here, we would like to design a payment scheme such that every wireless terminal will report its cost truthfully out of its own interest to maximize its profit. This payment scheme is called *strategyproof* in the literature since it removes speculation and counter-speculation among wireless terminals. Then a natural question is how we design such a payment scheme.

The most well-known and widely used strategyproof payment method is so called VCG mechanism family by Vickrey [21], Clarke [6], and Groves [10]. A VCG mechanism uses an output that maximizes the *social efficiency*, i.e., the total valuations of participating agents. Several mechanisms [15, 7, 1], which essentially belong to VCG mechanism family, have been proposed in the literature to ensure that each network agent will report its cost truthfully for unicast. In these mechanisms, the least cost path, which maximizes the social efficiency, is used for routing. To support a communication among a group of users, multicast is more efficient than unicast or broadcast, as it can transmit packets to destinations using fewer network resource, thus increases the social efficiency. A *truthful* multicast protocol, which selfish wireless terminals will follow, is composed of two parts (1) the tree structure that connects the sources and receivers, and (2) the payment to the relay nodes in this tree. Multicast poses a unique challenge in designing strategyproof mechanisms: it is NP-hard to to find the tree structure with the minimum cost, which in turn maximizes the social efficiency. A range of multicast structures, such as the least cost path tree (LCPT), the pruning minimum spanning tree (PMST), virtual minimum spanning tree (VMST) and Steiner tree, were proposed to replace the optimum multicast tree. In this paper, we will not redesign the wheel, instead, we show how payment schemes can be designed for existing multicast tree structures so rational selfish wireless terminals will follow the protocols for their own interests.

The main contribution of this paper is as follows. Firstly, for each of these widely used multicast structures, we show that a simple application of VCG payment method is not strategyproof: a wireless terminal may have incentives to lie about its cost to increase its profit. This is due to the fundamental difference between unicast and multicast: it is NP-hard to find the minimum cost multicast tree that span the sources and receivers, while the least cost unicast path can be found in polynomial time. Secondly, we design a strategyproof payment scheme for each of these multicast structures and prove that each of our payment schemes is the minimum among any truthful payment schemes for a given specific multicast tree structure. To the best of our knowledge, our protocols are the *first* truthful mechanisms that do not reply on VCG mechanisms for routing in selfish networks.

Notice that ensuring that each wireless terminal reports its cost truthfully is only one part of the story of truthful routing, which includes the routing subgame and the forwarding subgame. We do have to guarantee that they will indeed forward the packets. Unfortunately, it has been shown in [24] that there does not exist a dominant strategy solution in which every node always forwards packets in ad-hoc routing and forward games. In this paper, we focus on dominant strategy solutions in routing subgame in multicast. We study both link cost and node cost. For link cost, [24] shows that special care must be taken when designing a mechanism so that the links will report their non-private link types truthfully. In this paper, we assume that such a cryptographic mechanism is in place (e.g, [24]). Given that, we focus on designing truthful multicast routing scheme for link cost model and node cost model.

The rest of the paper is organized as follows. First, we introduce some preliminaries and related works in Section 2. We also present our communication model and the problems to be solved in this paper. We study the strategy-proof mechanism for link weighted network in Section 3 and node weighted network in Section 4. Simulation results are presented in Section 5. We conclude our paper in Section 6 by pointing out some possible future work.

2. PRELIMINARIES AND PRIOR ART

2.1 Preliminaries

In designing efficient, centralized or distributed algorithms and network protocols, the computational agents are typically assumed to be either *correct/obedient* or *faulty* (also called adversarial). Here agents are said to be *correct/obedient* if they follow the protocol correctly. In contrast, economists design market mechanisms in which it is assumed that agents are *rational*. The rational agents respond to well-defined incentives and will deviate from the protocol only if it improves their gain.

A standard economic model for the design and analysis of scenarios in which the participants act according to their own self-interests is as follows. Assume that there are n agents, which could be the wireless devices in a wireless ad hoc networks, the computers in a peer-to-peer networks, or even network links in a network. Each agent i, for $i \in \{1, \cdots, n\}$, has some *private* information t_i, called its *type*, e.g., the cost to forward a packet in a network environment. All agents' types define a type vector $t = (t_1, t_2, \cdots, t_n)$.

A mechanism defines, for each agent i, a set of strategies A_i. For each strategy vector $a = (a_1, \cdots, a_n)$, i.e., agent i plays a strategy $a_i \in A_i$, the mechanism computes an *output* $o = o(a_1, \cdots, a_n)$ and a *payment* vector $p = (p_1, \cdots, p_n)$, where $p_i = p_i(a_1, \cdots, a_n)$ is the money given to the participating agent i. For each possible output o, agent i's preferences are given by a valuation function v_i that assigns a real monetary number $v_i(t_i, o)$ to output o. Let $u_i(t_i, o(a), p_i(a))$ denote the *utility* of agent i at the outcome of the game, given its preferences t_i and strategies profile $a = (a_1, \cdots, a_n)$ selected by agents. A common assumption in mechanism design literature, and one which we will follow in this paper, is that agents are *rational* and have quasi-linear utility functions. The utility function is *quasi-linear* if $u_i(t_i, o) = v_i(t_i, o) + p_i$. An agent is called *rational*, if it always maximizes its utility by finding its best strategy. For a multicast routing protocol, the set of strategies A_k for a terminal k in a direct revelation mechanism is the set of possible costs that terminal k could declare. The utility of the terminal k on a tree connecting the source and the receivers is the payment p_k for terminal k minus its cost c_k. A strategy a_i is called *dominant strategy* if it maximizes the utility regardless of what other agents do, i.e.,

$$u_i(t_i, o(a_i, b_{-i}), p_i(a_i, b_{-i})) \geq u_i(t_i, o(a'_i, b_{-i}), p_i(a'_i, b_{-i}))$$

for all $a'_i \neq a_i$ and all strategies b_{-i} of agents other than i. Here $a_{-i} = (a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n)$ denotes the vector of strategies of all other agents except i.

Hereafter, we only consider direct-revelation mechanism in which the only actions available to agents are to make direct claims about their preferences v_i to the mechanism. A mechanism is *incentive compatible* (IC) if reporting valuation truthfully is a dominant strategy. Another very common requirement in the literature for mechanism design is so called *individual rationality* or *voluntary participation*: the agent's utility of participating in the output of the mechanism is not less than the utility of the agent if it did not participate at all. For convenience, let $t|^i b = (t_1, \cdots, t_{i-1}, b, t_{i+1}, \cdots, t_n)$, i.e., each agent $j \neq i$ reports its type t_j except that the agent i reports type b. Then, IC implies that, for each agent i, $v_i(t_i, o(t)) + p_i(t) \geq v_i(t_i, o(t|^i b)) + p_i(t|^i b)$; and IR implies that, for each agent i, $v_i(t_i, o(t)) + p_i(t) \geq 0$.

Arguably the most positive result in mechanism design is what is usually called the generalized Vickrey-Clarke-Groves (VCG) mechanism by Vickrey [21], Clarke [6], and Groves [10]. The VCG mechanism applies to maximization problems where the objective function is simply the sum of all agents' valuations. A direct revelation mechanism $M = (o(t), p(t))$ belongs to the VCG family if (1) the output $o(t)$ computed based on the type vector t maximizes the objective function $g(o, t) = \sum_i v_i(t_i, o)$, and (2) the payment to agent i is $p_i(t) = \sum_{j \neq i} v_j(t_j, o(t)) + h_i(t_{-i})$. Here $h_i()$ is an arbitrary function of t_{-i}. A VCG mechanism is always truthful [10]. Under mild assumptions, VCG mechanisms are the *only* truthful implementations to maximize the total valuations [9].

Although the family of VCG mechanisms is powerful, but it has its limitations. To use VCG mechanism, we have to compute the exact solution that maximizes the total valuation of all agents. This makes the mechanism computationally intractable in many cases. Notice that replacing the optimal algorithm with non-optimal approximation usually leads to untruthful mechanisms if VCG payment method is used [15]. To make the mechanism tractable, the output method $o()$, and the payment method $p()$ should be computable in polynomial time. Notice that it is NP-hard to find the tree with the minimum cost for multicast. Thus, the VCG mechanism using optimum minimum cost tree as output is not polynomially computable if $P \neq NP$.

In summary, we want to design strategy-proof multicast protocols for a selfish wireless network with the following properties. 1) *Incentive Compatibility (IC)*: an agent will reveal its true cost to maximize its utility no matter what the other agents do; 2) *Individual Rationality (IR)*: an agent is guaranteed to have non-negative utility if it reports its cost truthfully; and 3) *Polynomial Time Computability (PC)*: all computations (the computation of the output and the payment) are done in polynomial time.

2.2 Prior Art on Selfish Routing

How to achieve cooperation among selfish terminals in network was previously addressed in [4, 12, 14, 3, 5, 18, 19]. In [14], nodes, which agree to relay traffic but do not, are termed as misbehaving. Their protocol avoids routing through these misbehaving nodes. In [4, 12, 5, 3], a secure mechanism to stimulate nodes to cooperate is presented. The key idea behind these approaches is that terminals providing a service should be remunerated, while terminals receiving a service should be charged. Each terminal maintains a counter, called *nuglet counter*, in a tamper resistant hardware module, which is decreased when the terminal originates a packet and increased when the terminal forwards a packet.

Routing has been an important part of the algorithmic mechanism-design from the very beginning. Nisan and Ronen [15] provided a polynomial-time strategyproof mechanism for optimal unicast route selection in a centralized computational model. In their formulation, the network is modelled as an abstract graph $G = (V, E)$. Each edge e of the graph is an agent and has a private type t_e, which represents the cost of sending a message along this edge. Their mechanism is a VCG mechanism by using the Least Cost Path (LCP) as its output. Feigenbaum *et. al* [7] then addressed the truthful low cost routing in a different network model. They assume that each node k incurs a transit cost c_k for each transit packet it carries. Their mechanism again is the VCG mechanism. They gave a distributed method such that each node i can compute a payment $p_{ij}^k > 0$ to node k for carrying the transit traffic from node i to node j if node k is on the LCP $\mathsf{LCP}(i, j)$. Anderegg and Eidenbenz [1] recently proposed a similar routing protocol for wireless ad hoc networks based on VCG mechanism again. They assumed that each link has a cost and each node is a selfish agent. Feigenbaum *et. al* [8], by assuming a *fixed* multicast structure, designed a strategyproof mechanism that selects a subset of receivers (each with a privately known willing payment) and then shares the cost of the multicast tree providing the service among the selected receivers so budget balance is achieved.

When applying VCG mechanisms to complex problems such as multicast, a problem emerges: even finding the optimal outcomes is computationally intractable. A critical observation made by Nisan *et al.* [16] and other researchers is that if the optimal outcome is replaced by a polynomial-time computable structure then the mechanism using payment computed based on VCG method is no longer necessarily truthful! This phenomena is almost universal. To address this, Nisan and Ronen [16] introduced a notion of feasible truthfulness that captures the limitation on agents imposed by their own computational limits. They showed that under reasonable assumptions on the agents, it is possible to turn any VCG-based mechanism into a feasibly truthful one, using an additional appeal mechanism. In this paper, we use a totally different approach by using a payment scheme other than the VCG scheme, and we do *not* assume any computational limits on the agents.

2.3 Communication Model

In this paper, as did in the literature, we study two different models of wireless networking: link weighted and node weighted networking. For both models, usually the communication links are needed to be symmetric due to the following requirement: each receiver has to send an acknowledgment packet directly to the sender after it received the data. Thus, in this paper, we consider all communication links as undirected. Actually, our results can apply to case when the link is directed with some minor modification.

In a link weighted network, each communication link incurs a cost when a message is sent over it and the communication link is an agent, e.g., the marginal cost of this link transmitting the data. For example, in a cellular networks, it could be the cost of using the channel. For node weighted network each communication terminal will incur a cost when it has to relay a message for other node. Typical example of a node weighted network is the wireless ad hoc network with fixed transmission range. Throughout this paper, we always assume that the network is **bi-connected**, which implies that if we remove the agent the network is still connected. This assumption is necessary to prevent some nodes from being monopoly and charging arbitrary cost, in addition to increase network robustness.

It is well known that finding the minimum cost multicast tree (MCMT) is NP-hard for both link weighted networks and node weighted networks. So several multicast structures were proposed in the literature to approximate MCMT. In practice, two types of multicast structures are used to meet the requirements of different applications: *source based multicast tree* and *share based multicast tree*. For those applications like online movie, they usually have one or only a few senders and lots of receivers. Therefore, we often use a source based multicast tree in which receivers only receive messages but do not send them. On the other hand, many applications have lots of active senders, such as distributed interactive simulation applications, and distributed video-gaming (where most receivers are also senders). In this case, the share based tree is used to increase the scalability.

In this paper, we study how to design truthful payment schemes for the most widely used multicast trees, including source based trees and shared trees for both edge weighted and node weighted networks. The following assumptions are adopted in this paper: (1) all receivers will relay the data packets for peer receivers for free if it is asked to do so; (2) each relay agent (terminal or link) has a privately known cost to relay a transit traffic for other terminals and the cost is *independent* of the number of its children in the multicast tree; (3) the candidate relay agents (the agents besides the source and the receivers) will not *collude* with each other to improve their gains; (4) all agents are rational; (5) an agent receives zero payment if it is not in the multicast structure; and (6) the source of the multicast will pay the selected relay terminals. If we relax any of first five assumptions, we would have to design different mechanisms. If the sixth assumption is not met, we need design a payment sharing [23] scheme to share the payments fairly among all receivers. Regarding the collusion, notice multicast is a special case of unicast. If we consider the unicast, in reference [22], the authors proved a negative results about the non-existence of truthful payment if general collusion happens, i.e., there is no truthful payment scheme that can prevent any two agents from improving their gains by collusion with each other.

2.4 Problem Statement

Consider any communication network $G = (V, E, c)$, where $V = \{v_1, \cdots, v_n\}$ is the set of communication terminals, $E = \{e_1, e_2, \cdots, e_m\}$ is the set of links, and c is the cost vector of all agents. Here agents are terminals in a node weighted network and are links in a link weighted network. Given a set of sources and receivers $Q = \{q_0, q_1, q_2, \cdots, q_{r-1}\} \subset V$, the multicast problem is to find a tree $T \subset G$ spanning all terminals Q. For simplicity, we assume that $s = q_0$ is the sender of a multicast session if it exists. All terminals or links are required to declare a cost of relaying the message. Let d be the declared costs of all nodes, i.e., agent i declared a cost d_i. Based on the declared cost profile d, we should construct the multicast tree and decide the payment for the agents. The utility of an agent is its payment received, minus its cost if it is selected in the multicast tree. Instead of reinventing the wheels, we will still use the previously proposed structures for multicast as the output of our mechanism. Given a multicast tree, we will study the designing of strategyproof payment schemes based on this tree.

Given a network H, we use $\omega(H)$ to denote the total cost of all agents in this network. If we change the cost of any agent i (link e_i or node v_i) to c'_i, we denote the new network as $G' = (V, E, c|^i c'_i)$, or simply $c|^i c'_i$. If we remove one agent i from the network, we denote it as $c|^i \infty$. Denote $G \backslash e_i$ as the network without link e_i, and denote $G \backslash v_i$ as the network without node v_i and all its incident links. For the simplicity of notation, we will use the cost vector c to denote the network $G = (V, E, c)$ if no confusion is caused.

3. MULTICAST IN LINK WEIGHTED COMMUNICATION NETWORKS

In this section, we discuss how to conduct truthful multicast when the network is modelled by a link weighed communication graph. We assume the communication network is modelled by an undirected graph $G = (V, E, c)$. Here, the value of c_i is only known to each individual link e_i.

We specifically study the following three structures: least cost path tree (LCPT), pruning minimum spanning tree (PMST), and link weighted Steiner tree (LST). Notice that the first structure belongs to the family of the source based multicast tree, while the second and the third structure belong to the share based multicast tree.

3.1 Least Cost Path Tree

In practice, this is the most widely used multicast distribution tree. Notice that, although we only discuss the using of least cost path tree for the link weighted network (i.e., the link will incur a cost when transmitting data), all results we presented in this subsection can be extended to the node weighted scenario without any difficulty.

3.1.1 Constructing LCPT

First, each link e_i will report a cost d_i of forwarding the unit data, which is collected to the source node using the link-state algorithm. For each receiver $q_i \neq s$, we compute the shortest path (least cost path), denoted by $\mathsf{LCP}(s, q_i, d)$, from the source s to q_i under the reported cost profile d. The union of all least cost paths from the source to receivers is called *least cost path tree*, denoted by $LCPT(d)$. Clearly, we can construct LCPT in time $O(n \log n + m)$. Next we discuss how to design a truthful payment scheme while using LCPT as the output.

3.1.2 VCG mechanism on LCPT is not strategyproof

Intuitively, we would use the VCG payment scheme in conjunction with the LCPT tree structure as follows. The payment $p_k(d)$ to each link e_k in LCPT is

$$p_k(d) = \omega(LCPT(d|^k \infty)) - \omega(LCPT(d)) + d_k.$$

We show by an example that the above payment scheme is not strategyproof. In other words, if we simply apply VCG scheme on LCPT, a link may have incentives to lie about its cost. Figure 1 illustrates such an example where link sv_3 can lie its cost to improve its utility.

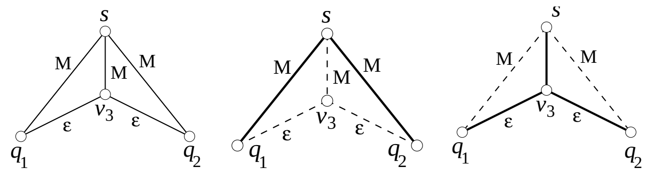


Figure 1: The cost of links are $c(sq_1) = c(sq_2) = c(sv_3) = M$, and $c(q_1v_3) = c(q_2v_3) = \epsilon$. Here, q_1 and q_2 are the receivers.

The payment to link sv_3 is 0 and its utility is also 0 if it reports its cost truthfully. The total payment to link sv_3 when sv_3 reported a cost $d_3 = M - 2\epsilon$ is $\omega(LCPT(c|^3\infty)) - \omega(LCPT(c|^3 d_3)) + d_3 = 2M - (M - 2\epsilon + 2\epsilon) + M - 2\epsilon = 2M - 2\epsilon$ and the utility of link sv_3 becomes $u_3(c|^3 d_3) = 2M - 2\epsilon - (M + \epsilon) = M - 3\epsilon$, which is larger than $u_3(c) = 0$, when $0 < \epsilon < M/3$.

3.1.3 Strategyproof mechanism on LCPT

Now, we describe our strategyproof mechanism that does not rely on VCG payment. For each receiver $q_i \neq s$, we compute the least cost path from the source s to q_i, and compute a payment $p_k^i(d)$ to every link e_k on the $\mathsf{LCP}(s, q_i, d)$ using the scheme for unicast

$$p_k^i(c) = d_k + |\mathsf{LCP}(s, q_i, d|^k \infty)| - |\mathsf{LCP}(s, q_i, d)|.$$

Here $|\mathsf{LCP}(s, q_i, d)|$ denotes the total cost of the least cost path $\mathsf{LCP}(s, q_i, d)$. The final payment to link $e_k \in LCPT$ is then

$$p_k(d) = \max_{q_i \in Q} p_k^i(d) \tag{1}$$

The payment to each link not on LCPT is simply 0.

Before we show that the above payment scheme (1) is truthful, let us illustrate it by a running example of how we pay link sv_3 in Figure 1. If link sv_3 reports a cost M truthfully, then it gets payment 0 since it $\notin$ the LCPT. If link sv_3 reports a cost $M - 2\epsilon$, it is now in the LCPT (composed of links sv_3, v_3q_1, and v_3q_2). Its payment then becomes $\max(p_{sv_3}^1, p_{sv_3}^2)$, where $p_{sv_3}^1 = M - 2\epsilon + |\mathsf{LCP}(s, q_1, d|^{sv_3}\infty)| - |\mathsf{LCP}(s, q_1, d)| = M - 2\epsilon + M - (M - 2\epsilon + \epsilon) = M - \epsilon$, and $p_{sv_3}^2 = M - \epsilon$ similarly. Then the profit of link sv_3 becomes $\max(p_{sv_3}^1, p_{sv_3}^2) - M = -\epsilon$, which is less than what it gets by reporting its truth cost.

THEOREM 1. *Payment (1) based on LCPT is truthful and it is minimum among all truthful payments based on LCPT.*

PROOF. Clearly, when link e_k reports its cost truthfully, it has non-negative utility, i.e., the payment scheme satisfies the IR property. In addition, since payment scheme for unicast is truthful, so e_k cannot lie its cost to increase its payment $p_k^i(c)$ based on $\mathsf{LCP}(s, q_i, d)$. Thus, it cannot increase $\max_{q_i \in Q} p_k^i(c)$ by lying its cost. In other words, our payment scheme is truthful.

We then show that the above payment scheme pays the minimum among all strategyproof mechanism using LCPT as output. Before showing the optimality of our payment scheme, we give some definitions first. Consider all paths from sender s to receiver q_i, they can be divided into two categories: with edge e_k or not. The path having the minimum length among these paths with edge e_k is denoted as $\mathsf{LCP}_{e_k}(s, q_i, d)$; and the path having the minimum length among these paths without edge e_k is denoted as $\mathsf{LCP}_{-e_k}(s, q_i, d)$.

Assume there is another payment scheme $\tilde{p}$ that pays less for a link e_k in a network G under cost profile d. Let $\delta = p_k(d) - \tilde{p}_k(d)$, then $\delta > 0$. Without loss of generality, assume that $p_k(d) = p_k^i(d)$. Thus, link e_k is on $\mathsf{LCP}(s, q_i, d)$ and the definition of $p_k^i(d)$ implies that

$$|\mathsf{LCP}_{-e_k}(s, q_i, d)| - |\mathsf{LCP}(s, q_i, d)| = p_k(d) - d_k.$$

Then consider another cost profile $d' = d|^k(p_k(d) - \frac{\delta}{2})$ where the true cost of link e_k is $p_k(d) - \frac{\delta}{2}$. Under profile d', since $|\mathsf{LCP}_{-e_k}(s, q_i, d')| = |\mathsf{LCP}_{-e_k}(s, q_i, d)|$, we have

$$\begin{aligned}
|\mathsf{LCP}_{e_k}(s, q_i, d')| &= |\mathsf{LCP}_{e_k}(s, q_i, d|^k 0)| + p_k(d) - \frac{\delta}{2} \\
&= |\mathsf{LCP}_{e_k}(s, q_i, d)| + p_k(d) - \frac{\delta}{2} - d_k \\
&= |\mathsf{LCP}(s, q_i, d)| + p_k(d) - \frac{\delta}{2} - d_k \\
&= |\mathsf{LCP}_{-e_k}(s, q_i, d)| - \frac{\delta}{2} \\
&< |\mathsf{LCP}_{-e_k}(s, q_i, d)| = |\mathsf{LCP}_{-e_k}(s, q_i, d')|
\end{aligned}$$

Thus, $e_k \in LCPT(d')$. From the following Lemma 2, we know that the payment to link e_k is the same for cost profile d and d'. Thus, the utility of link e_k under profile d' by payment scheme $\tilde{p}$ becomes $\tilde{p}_k(d') - c_k = \tilde{p}_k(d) - c_k = \tilde{p}_k(d) - (p_k(d) - \frac{\delta}{2}) = -\frac{\delta}{2} < 0$. In other words, under profile d', when link e_k reports its true cost, it gets a negative utility under payment scheme $\tilde{p}$. Thus, $\tilde{p}$ is not strategyproof. This finishes our proof. □

LEMMA 2. *If a mechanism based on a tree T with payment function $\tilde{p}$ is truthful, then for every agent a_k in network, if $a_k \in T$ then payment function $\tilde{p}_k(d)$ is independent of its declared cost d_k.*

PROOF. We prove it by contradiction. Suppose that there exists a truthful payment scheme such that $\tilde{p}_k(d)$ depends on d_k. There must exist two valid declared costs x_1 and x_2 such that $x_1 \neq x_2$ and $\tilde{p}_k(d|^k x_1) \neq \tilde{p}_k(d|^k x_2)$. Without loss of generality we assume that $\tilde{p}_k(d|^k x_1) > \tilde{p}_k(d|^k x_2)$. Now consider agent a_k with actual cost $c_k = x_2$. Obviously, it can lie its cost as x_2 to increase his utility, which violates the incentive compatibility (IC) property. □

Notice that the payment based on $p_k(c) = \min_{q_i \in Q} p_k^i(c)$ is not truthful since a link may lie its cost upward so it can discard some low payment from some receivers. In addition, the payment $p_k(c) = \sum_{q_i \in Q} p_k^i(c)$ is *not* truthful either.

3.1.4 Computational complexity

Assume there are r receivers, for every terminal q_i, we calculate the payment for all nodes $v_k \in \mathsf{LCP}(s, q_i, c)$ based on $\mathsf{LCP}(s, q_i, c)$ using the fast payment scheme for unicast problem [22]. This will take $O(n \log n + m)$ time. So for all terminals, it will take $O(rn \log n + rm)$. Note that we can construct the least cost path tree in time $O(n \log n + m)$. A very natural question is whether we can reduce the time complexity from $O(rn \log n + rm)$ to $O(n \log n + m)$. We leave it as an open question.

3.2 Pruning Minimum Spanning Tree

For LCPT tree, each sender of the multicast group has to build the tree rooted at itself. Although it can be constructed efficiently using the information collected from unicast, still one tree has to be constructed for each possible sender. One way to alleviate this is to construct a common tree that can be used by all possible senders. Minimum cost spanning tree is a reasonable choice. Since we only need the tree to span all the nodes in the multicast group, we could further trim some branches of the MST that does not contain any receivers.

3.2.1 Constructing PMST

First we construct the minimum spanning tree $MST(G)$ on the graph G. We then root the tree $MST(G)$ at sender s, prune all subtrees that do not contain a receiver. The final structure is called Pruning Minimum Spanning Tree (PMST).

3.2.2 VCG mechanism on PMST is not strategyproof

Intuitively, we would use the VCG payment scheme in conjunction with the PMST structure. The payment to an edge $e_k \in PMST(G)$ based on VCG would be as follows

$$p_k(d) = \omega(PMST(d|^k \infty)) - \omega(PMST(d)) + d_k.$$

We show by an example that the above payment scheme is not strategyproof. Figure 2 illustrates such an example where link $q_1 v_1$ has a negative utility when it reveals its true cost.

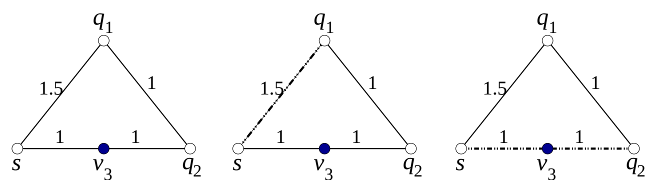


Figure 2: Terminals S is the sender and q_1,q_2 are receivers; $c(sq_1) = 1.5$ **and** $c(q_1q_2) = c(sv_3) = c(v_3q_2) = 1$**.**

If sv_3 reveals its true cost, its payment is $\omega(PMST(G\backslash sv_3)) - \omega(PMST(G) + c(sv_3) = 2.5 - 3 + 1 = 0.5$ and the utility of link sv_3 becomes -0.5, which violates IR.

3.2.3 Strategyproof mechanism on PMST

We now discuss our strategyproof payment scheme using PMST as the output. Instead of applying the VCG mechanism on PMST, we apply VCG mechanism on the MST. The payment for edge $e_k \in PMST(d)$ is

$$p_k(d) = \omega(MST(d|^k \infty)) - \omega(MST(d)) + d_k. \tag{2}$$

For every edge $e_k \notin PMST(d)$, its payment is 0.

Before prove the truthfulness and the optimality of our payment scheme, we first illustrate it by an example of how the payment to link sv_3 is computed. Clearly, the MST without using link sv_3 has total cost 3.5 and the MST when link sv_3 is considered has total cost 3. Thus, the payment to link sv_3 by payment (2) is $3.5 - 3 + 1 = 1.5$ and the utility of link sv_3 is 0.5.

THEOREM 3. *Our payment scheme (2) is truthful and minimal among all truthful payment schemes based on PMST.*

PROOF. For link $e_k \in PMST(d)$ or $e_k \notin MST(d)$, the payment is exactly the payment based on MST structure. Notice the payment based on MST belongs to VCG mechanism, so it is truthful. Thus, if $e_k \in PMST(d)$ or $e_k \notin MST(d)$, it does not have the incentive to lie. Now considering when $e_k \in MST(d) - PMST(d)$. If e_k lies its cost such that $e_k \notin MST(d)$, then it still gets utility 0; else the MST will keep unchanged which implies that e_k is still not in $PMST$. Thus, e_k also don't have the incentive to lie in this case. So our payment scheme (2) is truthful.

For $e_k \in PMST(d)$ our payment is same as the payment for MST, which is a VCG mechanism. Thus, our payment is minimal among all truthful payment scheme if the output is PMST. Detailed proof is omitted here due to space limit. □

3.2.4 Computational complexity

Obviously, we can construct the PMST in time $O(n \log n + m)$. We then analyze the time complexity of computing all links' payment in PMST. Let $G \backslash MST(G)$ be the graph after removing the edges of $MST(G)$ from G. Call the minimum spanning tree of $G \backslash MST(G)$ the second minimum spanning tree, denoted by $MST_2(G)$. It was shown that the total payment to all links in the MST equals to the actual cost of the $MST_2(G)$ in [2]. Also, it is not difficult to calculate payment for every link in PMST in time $O(n \log n + m)$, which is optimal.

3.3 Link Weighted Steiner Tree (LST)

It is well-known [17, 20] that it is NP-hard to find the minimum cost multicast tree when given an arbitrary link weighted graph G. For LCPT and PMST structure, while they usually work well in practice, in some extreme situations, the cost of these structures could be arbitrary larger than the optimal cost. Then it is desirable that we can find a structure such that even in worst case, the cost of structure is at most α times of the optimal. In literature, this structure is said to be a α-approximation of the optimal and α is called the approximation ratio.

Takahashi and Matsuyama [20] first gave a polynomial time algorithm that can output 2-approximation of the minimum cost Steiner tree (MCST). Then a series of results have been developed to improve the approximation ratio. The current best result is due to Robins and Zelikovsky [17], in which the authors presented a polynomial time method with approximation ratio $1 + \frac{\ln 3}{2}$. Takahashi and Matsuyama's algorithm is simpler and can be implemented in a distributed way, which fits the need of wireless networks. Thus, we use this algorithm instead of the algorithm with the best approximation ratio to construct multicast tree.

3.3.1 Constructing the LST

We first review the algorithm by Takahashi and Matsuyama:

ALGORITHM 1. *(Takahashi and Matsuyama [20])*

Repeat the following steps until no receiver remains:

1. Find one of the remaining receiver, say q_i, that is closest to the source s, i.e., the $\mathsf{LCP}(s, q_i, d)$ has the least cost among the shortest paths from s to all receivers.

2. Connect q_i to s using the least cost path between them and contract this least cost path to one virtual vertex. Remove some edges during contracting if necessary. This is virtual source terminal for next round.

For each iteration in Algorithm 1, we call it a round. Let P_i be the path found in round i, and t_i be the receiver it connects with the virtual source terminal. Given r receivers, the method terminates in r rounds. Hereafter, let $LST(d)$ be the final tree constructed by Algorithm 1. The authors of [20] proved that $\omega(LST(d)) \leq 2\omega(MCST(d))$.

3.3.2 VCG mechanism on LST is not strategy-proof

Given a tree $LST(d)$ approximating the minimum cost Steiner tree, a natural payment scheme would be to pay each edge based on VCG scheme, i.e., the payment to an edge $e_k \in LST(G)$ is

$$p_k(d) = \omega(LST(d|^k\infty)) - \omega(LST(d)) + d_k.$$

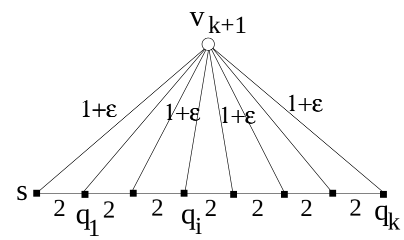


Figure 3: Terminals q_i, $1 \leq i \leq k$ are receivers; the cost of each link $v_{k+1}q_i$ and $v_{k+1}s$ is $1 + \epsilon$, where ϵ is a sufficiently small positive real number. The cost of each link q_iq_{i+1} and sq_1 is 2.

We give an example to show that this payment scheme does *not* satisfy IR property, i.e., it is possible that some edges have negative utility. Figure 3 illustrates the example with terminal s being the source terminal. It is not difficulty to show that, in the first round, link sq_1 is selected to connect terminals s and q_1 with cost 2; in round r, we will select link $q_{r-1}q_r$ to connect to q_r with cost 2. Thus, the tree $LST(G)$ will be just the path $sq_1q_2 \cdots q_k$, whose cost is $\sum_{i=1}^{k-1} c(q_iq_{i+1}) + c(sq_1) = 2k$.

When link $e_1 = sq_1$ is not used, it is easy to see that the final tree $LST(G \backslash e_1)$ will only use terminal v_{k+1} to connect all receivers with total cost $(k+1)(1+\epsilon)$. Thus, the utility of link $e_1 = sq_1$ is $\omega(LST(G \backslash e_1)) - \omega(LST(G)) = (k+1)(1+\epsilon) - 2k = k\epsilon - k + 2$, which is negative when $\epsilon < \frac{k-2}{k}$. Thus, the payment to link sq_1 does not satisfy the incentive rationality property.

3.3.3 Strategy-proof mechanism based on LST

In this subsection we describe our strategyproof mechanism (without using VCG) based on LST. Instead of paying the wireless link based on the final structure LST, we will calculate a payment for each round and choose the maximum as the final payment. Let $w_i(d)$ be the cost of the path P_i selected in the ith round if the cost profile is d.

ALGORITHM 2. ***Truthful payment to e_k based on LST***

1. Use Algorithm 1 to find $LST(d|^k\infty)$. When link e_k is not present, the graph used in the beginning of round i is denoted as $G_i^{-e_k}$.

2. For every round i, considering graph $G_i^{-e_k} \bigcup e_k$, find LCP from s to every remaining receivers and choose the LCP with the minimum weight. For simplicity, we denote this LCP as $\mathsf{P}'_i(d)$.

3. Define the payment for edge e_k in round i as
$$p_k^i(d) = w_i(d|^k\infty) - |\mathsf{P}'_i(d)| + d_k$$

4. The final payment to link e_k on $LST(d)$ is
$$p_k(d) = \max_{i=1}^{r} p_k^i(d) \quad (3)$$

THEOREM 4. *Our payment scheme based on LST is strategy-proof and minimum among truthful payment schemes based on LST.*

PROOF. First, for every round i, the payment scheme $p_k^i(d)$ belongs to VCG mechanism, so e_k gets maximum and non-negative utility from round i if it reveals its true cost c_k. Notice the final payment scheme is the maximal of $p_k^i(d)$ over all round i, so e_k gets maximum and non-negative under payment scheme (3) when it reveals its true cost c_k. Thus, our payment scheme is strategy-proof.

Now we prove the optimality of our payment scheme. We prove by contradiction. Suppose there exists a payment scheme $\tilde{p}$ such that for profile d, $\tilde{p}_k(d) < p_k(d)$, which equals $\tilde{p}_k(d) = p_k(d) - \delta$ ($\delta > 0$). From the IR property, we can assure that e_k is selected under profile d. Here we argue that if $d_k < p_k(d)$, then $e_k \in LST(d)$. Without loss of generality, we can assume $p_k(d) = p_k^i(d)$ for some round i. If e_k is selected before round i, then done. Else, in round i, we have $d_k < p_k(d) = p_k^i(d) = w_i(d|^k\infty) - |\mathsf{P}'_i(d|^k 0)|$. This implies that $w_i(d|^k\infty) > |\mathsf{P}'_i(d|^k 0)| + d_k$, which guarantees that e_k is selected in round i. Considering profile $d|^k p_k(d) - \frac{\delta}{2}$ with e_k's true cost $c_k = p_k(d) - \frac{\delta}{2}$. From lemma 2, e_k's payment under $\tilde{p}$ equals to $\tilde{p}_k(d|^i p_k(d) - \frac{\delta}{2}) = p_k(d) - \delta$, which is smaller than the true cost $c_k = p_k(d) - \frac{\delta}{2}$ of link e_k. This violates the assumption that payment scheme $\tilde{p}$ is truthful, which finishes our proof. ☐

3.3.4 Computational complexity

For every round, the payment $p_k^i(d)$ could be calculated in time $O(n \log n + m)$. There are r rounds, where r is the number of receivers, so overall complexity is $O(rn \log n + rm)$. The question left unsolved is: can we reduce the time complexity to $O(n \log n + m)$, which should be optimal if we can achieve that.

4. MULTICAST IN NODE WEIGHTED COMMUNICATION NETWORKS

In this section, we discuss in detail how to conduct truthful multicast when the network is modelled by a node weighed communication graph. We specifically study the following two structures: virtual minimum spanning tree (VMST) and node weighted Steiner tree (NST). Although LCPT is a very commonly used structure in node weighted wireless networks, but its construction and strategyproof payment scheme are nearly the same as in the link weighted networks, so we omit the discussion of this structure here. Notice both VMST and NST are share-based multicast trees, which implies that the receivers could also be the sender. In practice, for those share-based trees, receivers/senders in the same multicast group usually belong to the same organization or company, so their behavior can be expected to be cooperative instead of uncooperative. Thus, we assume every receiver will relay the packet for peer receivers for free.

4.1 Virtual Minimum Spanning Tree

4.1.1 Constructing the VMST

Our virtual minimum spanning tree structure mimics the overlay network for the multicast. For each pair of nodes in the multicast group, we build a tunnel using the shortest cost path connecting them. Among all the tunnels, we select the minimum cost tree to connect all nodes in the multicast group. We first describe our method to construct the virtual minimum spanning tree.

ALGORITHM 3. ***Virtual MST Algorithm***

1. *First, calculate the pairwise least cost path $\mathsf{LCP}(q_i, q_j, d)$ between any two terminals $q_i, q_j \in Q$ when the declared cost vector is d.*
2. *Construct a virtual complete link weighted network $K(d)$ using Q as its terminals, where the link $q_i q_j$ corresponds to the least cost path $\mathsf{LCP}(q_i, q_j, d)$, and its weight $w(q_i q_j)$ is the cost of the path $\mathsf{LCP}(q_i, q_j, d)$, i.e., $w(q_i q_j) = |\mathsf{LCP}(q_i, q_j, d)|$.*
3. *Build the minimum spanning tree (MST) on $K(d)$. The resulting MST is denoted as $VMST(d)$.*
4. *For each virtual link $q_i q_j$ in $VMST(d)$, we mark every node on $\mathsf{LCP}(q_i, q_j, d)$ as relay node. Thus, a terminal v_k is a relay node iff v_k is on some virtual links in the $VMST(d)$.*

4.1.2 VCG mechanism on VMST is not strategy-proof

In this subsection, we show that a simple application of VCG mechanism on VMST is not strategy-proof. Figure 4 illustrates such an example where terminal v_3 can lie its cost to improve its utility when output is VMST. The payment to terminal v_3 is 0 and its utility is also 0 if it reports its cost truthfully. The total payment to terminal v_3 when v_3 reported a cost $d_3 = M - \epsilon$ is $\omega(VMST(c|^3\infty)) - \omega(VMST(c|^3 d_3)) + d_3 = 2M - (M - \epsilon) + M - \epsilon = 2M$ and the utility of terminal v_3 becomes $u_3(c|^3 d_3) = 2M - (M + \epsilon) = M - \epsilon$, which is larger than $u_3(c) = 0$. Thus, VCG mechanism based on VMST is not strategy-proof.

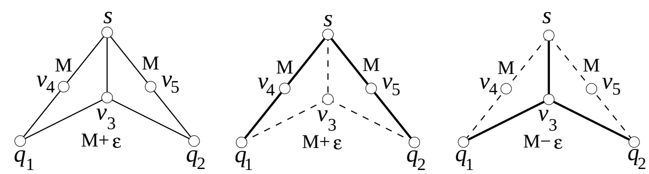


Figure 4: The cost of terminals are $c(v_4) = c(v_5) = M$ and $c(v_3) = M + \epsilon$.

4.1.3 Strategyproof mechanism on VMST

Before discussing the strategyproof mechanism based on VMST, we give some related definitions first. Given a spanning tree T and a pair of terminals p and q on T, clearly there is a unique path connecting them on T. We denote such path as $\Pi_T(p, q)$, and the edge with the maximum length on this path as $LE(p, q, T)$. For simplicity, we use $LE(p, q, d)$ to denote $LE(p, q, VMST(d))$ and use $LE(p, q, d|^k d'_k)$ to denote $LE(p, q, VMST(d|^k d'_k))$.

Following is our truthful payment scheme when the output is the multicast tree $VMST(d)$.

ALGORITHM 4. *Truthful payment scheme based on VMST*

1. For every terminal $v_k \in V \backslash Q$ in G, first calculate $VMST(d)$ and $VMST(d|^k\infty)$ according to the terminals' declared costs vector d.
2. For any edge $e = q_i q_j \in VMST(d)$ and any terminal $v_k \in \mathsf{LCP}(q_i, q_j, d)$, we define the payment to terminal v_k based on the virtual link $q_i q_j$ as follows:

$$p_k^{ij}(d) = |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d)| + d_k.$$

Otherwise, $p_{ij}^k(d)$ is 0. The final payment to terminal v_k based on $VMST(d)$ is

$$p_k(d) = \max_{q_i q_j \in VMST(d)} p_k^{ij}(d). \quad (4)$$

Again we first illustrate our payment scheme by a running example. Node v_3 gets payment 0 when it reports its true cost $M + \epsilon$. When it lies its cost to $M - \epsilon$, let us see how much we will pay. Now the VMST will have two links sq_1 (corresponding to $\mathsf{LCP}(s, q_1, d') = sv_3 q_1$) and sq_2 (corresponding to $\mathsf{LCP}(s, q_2, d') = sv_3 q_2$). In other words, v_3 appears in two virtual links sq_1 and sq_2

of $VMST(d')$. If v_3 is not present, then the VMST still has two links sq_1 (corresponding to $\mathsf{LCP}(s, q_1, d') = sq_1$) and sq_2 (corresponding to $\mathsf{LCP}(s, q_2, d') = sq_2$). Then the payment to v_3 based on link sq_1 is $p_{v_3}^{sq_1} = |LE(s, q_1, d|^3\infty)| - |\mathsf{LCP}(s, q_1, d)| + d_3 = M - (M - \epsilon) + (M - \epsilon) = M$. Similarly, the payment to v_3 based on link sq_1 is $p_{v_3}^{sq_1} = M$. Thus, the final payment to node v_3 is M, which is less than its true cost $M + \epsilon$.

THEOREM 5. *Our payment scheme (4) is strategyproof and minimum among all truthful payment schemes based on VMST.*

Instead of proving Theorem 5, we prove Theorem 6, Theorem 9 and Theorem 11 in the remaining of this subsection.

Before the proof of Theorem 5, we give some related notations and observation. Considering the graph $K(d)$ and a node partition $\{Q_i, Q_j\}$ of Q, if an edge's two end nodes belong to different node set of the partition, we call it a *bridge*. All bridge edges are denoted as $B(Q_i, Q_j, d)$. The bridge edge with the minimum cost is denoted as $MB(Q_i, Q_j, d)$. All bridges q_sq_t over node partition Q_i, Q_j in the graph $K(d)$ satisfying $v_k \notin \mathsf{LCP}(q_s, q_t, d)$ form a bridge set $B^{-v_k}(Q_i, Q_j, d)$. Among them, the bridge with the minimum length is denoted as $MB^{-v_k}(Q_i, Q_j, d)$ when the nodes' declared cost vector is d. Similarly, all bridges q_sq_t over node partition Q_i, Q_j in $K(d)$ satisfying $v_k \in \mathsf{LCP}(q_s, q_t, d)$ form a bridge set $B^{v_k}(Q_i, Q_j, d)$. The bridge in $B^{v_k}(Q_i, Q_j, d)$ with the minimum length is denoted as $BM^{v_k}(Q_i, Q_j, d)$. Obviously, we have

$$BM(Q_i, Q_j, d) = \min\{BM^{v_k}(Q_i, Q_j, d), BM^{-v_k}(Q_i, Q_j, d)\}.$$

We then state our main theorems for the payment scheme discussed above.

THEOREM 6. *Our payment scheme satisfies IR.*

PROOF. First of all, if terminal v_k is not chosen as relay terminal, then its payment $p_k(d|^kc_k)$ is clearly 0 and its valuation is also 0. Thus, its utility $u_k(d|^kc_k)$ is 0.

When terminal v_k is chosen as a relay terminal when reveals its true cost c_k, from the following observation 1 about MST we have $|LE(q_i, q_j, d|^k\infty)| \geq |\mathsf{LCP}(q_i, q_j, d|^kc_k)|$. The lemma immediately follows from

$$p_{ij}^k(d|^kc_k) = |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^kc_k)| + c_k \geq c_k.$$

This finishes the proof. $\square$

OBSERVATION 1. *For any cycle C in graph G, assume e_c is the longest edge in the cycle, then $e_c \notin MST(G)$.*

From the definition of the incentive compatibility (IC), we assume the d_{-k} is fixed throughout this proof. For our convenience, we will use $G(d_k)$ to represent the graph $G(d|^kd_k)$. We first prove a series of lemmas that will be used to prove that our payment scheme satisfies IC.

LEMMA 7. *If $v_k \in q_iq_j \in VMST(d)$, then $p_k^{ij}(d)$ does not depend on d_k.*

PROOF. Remember that the payment based on link q_iq_j is $p_k^{ij}(d) = |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d)| + d_k$, where $LE(q_i, q_j, d|^k\infty)$ is the longest edge of the unique path from q_i to q_j on the overlay tree $VMST(d|^k\infty)$. Clearly, it is independent of d_k. Now considering the second part $\mathsf{LCP}(q_i, q_jd) - d_k$. From the assumption we know that $v_k \in \mathsf{LCP}(q_i, q_j, d)$, so the path $\mathsf{LCP}(q_i, q_j, d)$ remains the same regardless of v_k's declared cost d_k. Thus, the summation of all terminals' cost on $\mathsf{LCP}(q_i, q_j, d)$ except terminal v_k equals to

$$|\mathsf{LCP}(q_i, q_j, d|^k0)| = |\mathsf{LCP}(q_i, q_j, d)| - d_k.$$

In other word, the second part is also independent of d_k. Now we can write the payment to a terminal v_k based on edge q_iq_j as following:

$$p_k^{ij}(d) = |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^k0)|,$$

Here terminal $v_k \in \mathsf{LCP}(q_i, q_j, d)$ and $q_iq_j \in VMST(d)$. $\square$

If a terminal v_k lies its cost c_k upward, we denote the lied cost as $\overline{c_k}$. Similarly, if terminal v_k lies its cost c_k downward, we denote the lied cost as $\underline{c_k}$. Let $E_k(d_k)$ be the set of edges q_iq_j such that $v_k \in \mathsf{LCP}(q_i, q_j, d)$ and $q_iq_j \in VMST(d)$ when terminal v_k declares a cost d_k. From Lemma 7 the non-zero payment to v_k is defined based on $E_k(d_k)$. Following lemma reveals the relationship between d_k and $E_k(d_k)$:

LEMMA 8. $E_k(d_k) \subseteq E_k(d'_k)$ *when* $d'_k \leq d_k$.

We now state the proof that payment scheme (4) satisfies IC.

THEOREM 9. *Our payment scheme satisfies the incentive compatibility (IC).*

PROOF. For terminal v_k, if it lies its cost from c_k to $\overline{c_k}$, then $E_k(\overline{c_k}) \subseteq E_k(c_k)$, which implies that payment

$$\begin{aligned} p_k(d|^k\overline{c_k}) &= \max_{q_iq_j \in E_k(\overline{c_k})} p_k^{ij}(d|^k\overline{c_k}) \\ &\leq \max_{q_iq_j \in E_k(c_k)} p_k^{ij}(d|^kc_k) = p_k(d|^kc_k). \end{aligned}$$

Thus, terminal v_k won't lies it cost upward, so we focus our attention on the case when terminal v_k lies its cost downward.

From Lemma 8, we know that $E_k(c_k) \subseteq E_k(\underline{c_k})$. Thus, we only need to consider the payment based on edges in $E_k(\underline{c_k}) - E_k(c_k)$. For edge $e = q_iq_j \in E_k(\underline{c_k}) - E_k(c_k)$, let $q_I^kq_J^k = LE(q_i, q_j, d|^k\infty)$ in the spanning tree $VMST(d|^k\infty)$. If we remove the edge $q_I^kq_J^k$, we have a vertex partition $\{Q_I^k, Q_J^k\}$, where $q_i \in Q_I^k$ and $q_j \in Q_J^k$. In the graph $K(d)$, we consider the bridge $BM(Q_I^k, Q_J^k, d)$ whose weight is minimum when the terminals cost vector is d. There are two cases needed to be considered about $BM(Q_I^k, Q_J^k, d)$: 1) $v_k \notin BM(Q_I^k, Q_J^k, d|^kc_k)$ or 2) $v_k \in BM(Q_I^k, Q_J^k, d|^kc_k)$. We discuss them individually.

Case 1**:** $v_k \notin BM(Q_I^k, Q_J^k, d|^kc_k)$. In this case, edge $q_I^kq_J^k$ is the minimum bridge over Q_I^k and Q_J^k. In other words, we have $|LE(q_i, q_j, |^k\infty)| \leq |\mathsf{LCP}(q_i, q_j, d|^kc_k)|$. Consequently

$$\begin{aligned} p_k^{ij}(d|^k\underline{c_k}) &= |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^k\underline{c_k})| + \underline{c_k} \\ &= |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^kc_k)| + c_k \\ &\leq c_k, \end{aligned}$$

which implies v_k will not benefit from lying its cost downward.

Case 2**:** $v_k \in BM(Q_I^k, Q_J^k, d|^kc_k)$. From the assumption that $q_iq_j \notin VMST(G(d|^kc_k))$, edge $BM(Q_I^k, Q_J^k, d|^kc_k)$ cannot be q_iq_j. Thus, there exists an edge $q_sq_t \neq q_iq_j$ such that $v_k \in \mathsf{LCP}(q_s, q_t, d|^kc_k)$ and $q_sq_t = BM(Q_I^k, Q_J^k, d|^kc_k)$. This guarantees that $q_sq_t \in VMST(d|^kc_k)$.

Obviously, q_s, q_t can not appear in the same set of Q_I^k or Q_J^k. Thus, $q_I^kq_J^k$ is on the path from q_s to q_t in graph $VMST(d|^k\infty)$, which implies that $|\mathsf{LCP}(q_I^k, q_J^k, d|^k\infty)| = |LE(q_i, q_j, d|^k\infty)| \leq$

$|LE(q_s, q_t, d|^k\infty)|$. Using Lemma 8, we have $\mathsf{LCP}(q_s, q_t, d|^k\underline{c_k}) \in VMST(d|^k\underline{c_k}))$. Thus,

$$\begin{aligned}
p_k^{ij}(d|^k\underline{c_k}) &= |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^k\underline{c_k})| + \underline{c_k} \\
&= |LE(q_i, q_j, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^k c_k)| + c_k \\
&\leq |LE(q_s, q_t, d|^k\infty)| - |\mathsf{LCP}(q_i, q_j, d|^k c_k)| + c_k \\
&\leq |LE(q_s, q_t, d|^k\infty)| - |\mathsf{LCP}(q_s, q_t, d|^k c_k)| + c_k \\
&= p_k^{st}(d|^k c_k)
\end{aligned}$$

This inequality concludes that even if v_k lies its cost downward to introduce some new edges in $E_k(\underline{c_k})$, the payment based on these newly introduced edges is no larger than the payment on some edges already contained in $E_k(c_k)$. In summary, node v_i don't have the incentive to lie its cost upward or downward, which proves the IC. □

Before proving Theorem 11, we prove the following lemma regarding all truthful payment schemes based on VMST.

LEMMA 10. *If $v_k \in VMST(d|^k c_k)$, then as long as $d_k < p_k(d|^k c_k)$ and d^{-k} fixed, $v_k \in VMST(d)$.*

PROOF. Again, we prove it by contradiction. Assume that $v_k \notin VMST(d)$. Obviously, $VMST(d) = VMST(d|^k\infty)$. Assume that $p_k(d|^k c_k) = p_k^{ij}(d|^k c_k)$, i.e., its payment is computed based on edge $q_i q_j$ in $VMST(d|^k c_k)$. Let $q_I q_J$ be the $LE(q_i, q_j, d|^k\infty)$ and $\{Q_i, Q_j\}$ be the vertex partition introduced by removing edge $q_I q_J$ from the tree $VMST(d|^k\infty)$, where $q_i \in Q_i$ and $q_j \in Q_j$. The payment to terminal v_k in $VMST(d|^k c_k)$ is $p_k(d|^k c_k) = |\mathsf{LCP}(q_I, q_J, d|^k\infty)| - c_{ij}^{v_k}$, where $c_{ij}^{v_k} = |\mathsf{LCP}(q_i, q_j, d|^k 0|$. When v_k's declare its cost as d_k, the length of the path $\mathsf{LCP}(q_i, q_j, d)$ becomes $c_{ij}^{v_k} + d_k = |\mathsf{LCP}(q_I, q_J, d|^k\infty)| - p_k(d|^k c_k) + d_k < |\mathsf{LCP}(q_I, q_J, d|^k\infty)|$.

Now consider the spanning tree $VMST(d)$. We have assumed that $v_k \notin VMST(d)$, i.e., $VMST(d) = VMST(d|^k\infty)$. Thus, among the bridge edges over Q_i, Q_j, edge $q_I q_J$ has the least cost when graph is $G\backslash v_k$ or $G(d|^k d_k)$. However, this is a contradiction to we just proved: $|\mathsf{LCP}(q_i, q_j, d|^k d_k)| < |\mathsf{LCP}(q_I, q_J, d|^k\infty)|$. This finishes the proof. □

We now ready to show that our payment scheme is optimal among all truthful mechanisms using VMST.

THEOREM 11. *Our payment scheme is the minimum among all truthful payment schemes based on VMST structure.*

PROOF. We prove it by contradiction. Assume that there is another truthful payment scheme, say $\mathcal{A}$, based on VMST, whose payment is smaller than our payment for a terminal v_k under cost profile d. Assume that the payment calculated by $\mathcal{A}$ for terminal v_k is $\tilde{p}_k(d) = p_k(d) - \delta$, where $p_k(d)$ is the payment calculated by our algorithm and $\delta > 0$.

Now consider another profile $d|^k d'_k$, where terminal has the true cost $c_k = d'_k = p^k(d) - \frac{\delta}{2}$. From Lemma 10, we know that v_k is still in $VMST(d|^k d'_k)$. Using Lemma 2, we know that the payment for terminal v_k using algorithm $\mathcal{A}$ is $p_k(c) - \delta$, which is independent of terminal v_k's declared cost. Notice that $d_k = p_k(d) - \frac{\delta}{2} > p_k(d) - \delta$. Thus, terminal v_k has a negative utility under payment scheme $\mathcal{A}$ when it reveals it true cost under cost profile $d|^k d'_k$, which violates the incentive compatibility (IC). This finishes the proof. □

By summarizing Theorem 6, Theorem 9 and Theorem 11, we get Theorem 5.

4.1.4 Computational complexity

We now discuss how to compute the payment to every relay terminal efficiently. Assume that the original communication graph G has n vertices and m edges.

One naive method of computing the payment works as follows. We first construct the complete graph $K(d)$ and then construct the spanning tree $VMST(d)$ on $K(d)$. It is easy to show the overall time complexity to construct $VMST(d)$ is $O(r^2 + rn\log n + rm) = O(rn\log n + rm)$, where r is the number of receivers. In order to calculate the payment for terminal $v_k \in \mathsf{LCP}(q_i, q_j, d) \in VMST(d)$, we should construct the tree $VMST(d|^k\infty)$, which will take time $O(rn\log n + rm)$. Finding the edge $LE(q_i, q_j, d|^k\infty)$ takes only $O(r)$ time. In the worst case, terminal v_k may appear on $O(r)$ edges of $VMST(d)$. Thus, we can calculate the payment for the single terminal v_k in time $O(r^2) + O(rn\log n + km) = O(rn\log n + rm)$. In the worst case, there could be $O(n)$ terminals on $VMST(d)$, so we can calculate the payment for all relay terminals in tree $VMST(G)$ in time $O(rn^2\log n + rmn)$.

Our improvement uses the fast payment for unicast as a subroutine. For a pair of nodes q_i, q_j, we find the path $\mathsf{LCP}(q_i, q_j, d|^k\infty)$ for every terminal $v_k \in \mathsf{LCP}(q_i, q_j, d)$, which can be done in time $O(n\log n + m)$. It takes $O(r^2 n\log n + r^2 m)$ to find the complete graph $K(d|^k\infty)$ for every terminal v_k. Finding the MST on each such complete graph takes time $O(r^2)$. Thus, we can construct VMSTs for all these n complete graphs in time $O(r^2 n)$. Based on these n VMSTs, it takes $O(r^2)$ to calculate the payment for one terminal. Then, in the worst case, it takes $O(r^2 n)$ to calculate the payment to every relay terminal. Overall, the time complexity of this approach is $O(r^2 n\log n + r^2 m) + O(r^2 n) + O(r^2 n) = O(r^2 n\log n + r^2 m)$. When $r = o(\sqrt{n})$, this approach outperforms the naive approach with time complexity $O(n^2\log n + mn)$. When r is a constant, the time complexity of the above approach becomes $O(n\log n + m)$, which is optimum.

4.2 Node Weighted Steiner Tree (NST)

Compared with LST in link weighted network, the structure of node-weighted Steiner tree (NST) in a node weighted network is even tough. It is well-known [11, 13] that it is NP-hard to find the minimum cost multicast tree when given an arbitrary node weighted graph G, and it is at least as hard to approximate as the set cover problem. Klein and Ravi [13] showed that it can be approximated within $O(\ln r)$, where r is the number of receivers.

4.2.1 Constructing NST

We review the method used in [13] to find a NST. We first introduce some definitions that are essential to construct the NST. A *spider* is defined as a tree having at most one node of degree more than two. Such a node (if exists) is called the center of the spider. Each path from the center to a leaf is called a *leg*. The *cost* of a spider S is defined as the sum of the cost of all nodes in spider S, denotes as $\omega(S)$. The number of terminals or *legs* of the spider is denoted by $t(S)$, and the ratio of a spider is defined as $\rho(S) = \frac{\omega(S)}{t(S)}$.

ALGORITHM 5. ***Construct NST***

Repeat the following steps until no receivers left and there is only one virtual terminal left.

1. Find the spider S with the minimum $\rho(S)$ that connect some receivers and virtual terminals.[1]

[1]For simplicity of the proof, we assume there doesn't have two spiders with the same ratio. Dropping the assumption won't change our results.

2. Contract the spider S by treating all nodes in it as one virtual terminal. The contracted virtual terminal has a weight zero. We call this as one *round*.

All nodes belong to the final unique virtual terminal form the NST.

THEOREM 12. *[13] The tree constructed above has cost at most $2\ln k$ times of the optimal.*

4.2.2 VCG mechanism on NST in not strategy-proof

Again, we may want to pay terminals based on VCG scheme, i.e., the payment to a terminal $v_k \in NST(d)$ is

$$p_k(d) = \omega(NST(d|^k\infty)) - \omega(NST(d)) + d_k.$$

We show by an example that the payment scheme does *not* satisfy IR property: it is possible that some terminal has negative utility. Figure 5 illustrates such an example. It is not difficulty to show that,

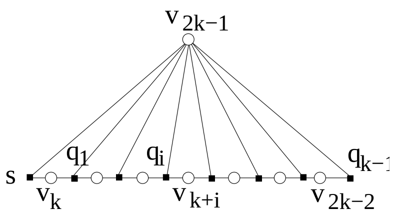


Figure 5: Terminals q_i, $1 \le i < k$ are receivers; the cost of terminal v_{2k-1} is 1. The cost of each terminal v_i, $k \le i \le 2k-2$, is $\frac{2}{2k-i} - \epsilon$, where ϵ is a sufficiently small positive real number.

in the first round, terminal v_k is selected to connect terminals s and q_1 with cost ratio $\frac{1}{k} - \frac{\epsilon}{2}$ (while all other spiders have cost ratio at least $\frac{1}{k}$). Then terminals s, v_k and q_1 form a virtual terminal. At the beginning of round r, we have a virtual terminal, denoted by V_r formed by terminals v_{k+i-1}, $1 \le i \le r-1$, and receivers q_i, $1 \le i \le r$; all other receivers q_i, $r < i < k$ are the remaining terminals. It is easy to show that we will select terminal q_{k+r-1} at round r to connect V_r and q_{r+1} with cost ratio $\frac{1}{k+1-r} - \frac{\epsilon}{2}$. Thus, the total cost of the tree $NST(G)$ is $\sum_{i=1}^{k-1}(\frac{2}{k+1-i} - \epsilon) = 2H(k) - 2 - (k-1)\epsilon$.

When terminal v_k is not used, it is easy to see that the final tree $NST(G \backslash v_k)$ will only use terminal v_{2k-1} to connect all receivers with cost ratio $\frac{1}{k}$ when $\frac{1}{k-1} - \frac{\epsilon}{2} > \frac{1}{k}$. Notice that this condition can be trivially satisfied by letting $\epsilon = \frac{1}{k^2}$. Thus, the utility of terminal v_k is $p_1(d) - c(v_k) = \omega(NST(G \backslash v_k)) - \omega(NST(G)) = -2H(k) + 3 + (k-1)\epsilon$, which is negative when $k \ge 8$, and $\epsilon = 1/k^2$.

4.2.3 Strategy-proof mechanism based on NST

Notice, the construction of NST tree is by rounds. Following, we show that if terminal v_k is selected as part of the spider with minimum ratio under cost profile d in a round i, then v_k is selected before or in round i under cost profile $d' = d|^k d'_k$ for $d'_k < d_k$. We prove this by contradiction, which assumes terminal v_k won't appear before round $i+1$. Notice that the graph remains the same for round i after the profile changes, so spider $S_i(d)$ under cost profile d is still a valid spider under cost profile d'. Its ratio becomes $\omega_i^k(d) - d_k + d'_k < \omega_i^k(d)$ while all other spiders' ratio keeps the same if they don't contain v_k. Thus, spider $S_i^k(d)$ has the minimum ratio among all spiders under cost profile d', which is a contradiction. So for terminal v_k, there exists a real value $B_k^i(d_{-k})$ such that terminal v_k selected before or in round i iff $d_k < B_k^i(d_{-k})$. If they are r rounds, we have an increasing sequence

$$B_k^1(d_{-k}) \le B_k^2(d_{-k}) \le \cdots \le B_k^r(d_{-k}) = B_k(d_{-k})$$

Obviously, terminal v_k is selected in the final multicast tree iff $d_k < B_k(d_{-k})$. Following is our payment scheme based on NST. For a node v_k, if v_k is selected then it gets payment

$$p_k(d) = B_k(d_{-k}). \tag{5}$$

Otherwise, it gets payment 0.

Regarding this payment we have the following theorem:

THEOREM 13. *Our payment scheme (5) is truthful, and among all truthful payment schemes for multicast tree based on NST, our payment is minimal.*

PROOF. From our conclusion that v_k is selected iff $d_k < B_k^i(d_{-k})$, we have $u_k(d) = B_k(d_{-k}) - d_k > 0$, which implies IR. Now we prove our payment scheme (5) satisfies IC by cases. Notice when v_k is selected, its payment doesn't depend on d_k, so we only need to discuss the following two cases:

Case 1: When v_k declares c_k, it is selected. What happens if it lies its cost upward as d_k to make it not selected? From the IR property, v_k gets positive utility when it reveals its true cost while it gets utility 0 when it lies it cost as d_k. So v_k has better not to lie.

Case 2: When v_k declares c_k, it is not selected. What happens if it lies its cost downward as d_k to make it selected? When v_k reveals c_k, it has utility 0, after lying it has utility $B_k(d_{-k}) - c_k$. From the assumption that v_k is not selected under cost profile $d|^k c_k$, we have $B_k(d_{-k}) \le c_k$. Thus, v_k will get non-positive utility if it lies, which ensures v_k revealing its true cost c_k.

So overall, v_k will always choose to reveal its actual cost to maximize its utility (IC property).

Next we prove that our payment is minimal. We prove it by contradiction, suppose there exists such payment scheme $\tilde{P}$ such that for terminal v_k under cost profile d, the payment to $\tilde{P}_i(d)$ is smaller than our payment. Notice in order to satisfies the IR, the terminal must be selected, so we assume $\tilde{P}_i(d) = B_k(d_{-k}) - \delta$, while δ is a positive real number. Now considering the profile $d' = d|^k(B_k(d_{-k}) - \frac{\delta}{2})$ with v_k's actual cost $c_k = B_k(d_{-k}) - \frac{\delta}{2}$. Obviously, v_k is selected, from lemma 2 the payment to v_k is $B_k(d_{-k}) - \delta$. Thus, the utility of v_k becomes $u_k(d') = B_k(d_{-k}) - B_k(d_{-k}) - \delta + \frac{\delta}{2} = -\frac{\delta}{2} < 0$, which violates the IR. This finishes our proof. $\square$

With Theorem 13, we only need focus our attention on how to get the value $B_k^i(d_{-k})$. Before we present our algorithm to find $B_k^i(d_{-k})$, we first review in details how to find the minimum ratio spider. In order to find the spider with the minimum ratio, we find the spider centered at terminal v_j with the minimum ratio over all terminals $v_j \in V$ and choose the minimum among them. The algorithm is as follows.

ALGORITHM 6. ***Find the minimum ratio spider***
Do the following process for all $v_j \in V$:

1. *Calculate the shortest path tree rooted at v_j and spanning all terminals. We call each shortest path a* branch*. The weight of the branch is defined as the length of the shortest path. Notice that the weight of the shortest path doesn't include the weight of the center node v_j of the spider and all the receivers.*

2. *Sort the branches according to their weights.*

3. *For every pair of branches, if they have relay terminals in common then remove the branch with larger weight. Assume the remaining branches are*

$$L(v_j) = \{L_1(v_j), L_2(v_j), \cdots, L_r(v_j)\}$$

sorted in ascending order according to their weights.

4. *Find the minimum ratio spider with center v_j by linear scanning: the spider is formed by the first $t \geq 2$ branches such that $\frac{c_j+\sum_{k=1}^{t} L_k}{t} \leq \frac{c_j+\sum_{k=1}^{h} L_k}{h}$ for any $h \neq t$.*

Assume that the spider with minimum ratio centered at terminal v_j is $S(v_j)$ and its ratio is $\rho(v_j)$. Then the spider with minimum ratio is $S = \{S(v_j) | v_j \in V$ and $\rho(v_j) = \min_{v_i \in V} \rho(v_i)\}$.

In Algorithm 6, $\omega(L_i(v_j))$ is defined as the sum of the terminals' cost on this branch excluding v_j, and $\Omega_i(L(v_j)) = \sum_{s=1}^{i} \omega(L_s(v_j)) + c_j$. If we remove node v_k, the minimum ratio spider centered at v_j is denoted as $S^{-v_k}(v_j)$ and its ratio is denoted as $\rho^{-v_k}(v_j)$. Let $L_1^{-v_k}(v_j), L_2^{-v_k}(v_j), \cdots, L_r^{-v_k}(v_j)$ be those branches in ascending order before linear scan.

From now on, we fix d_{-k} and graph G to study the relationship between the minimum ratio of spider centered at v_j $\rho(v_j)$ and d_k. If the minimum ratio spider with terminal v_k has t legs, then its ratio will be a line with slope of $\frac{1}{t}$. So the ratio-cost function is several line segments. Observe that the number of the legs of minimum ratio spider decreases over d_k. Thus, these line segments have decreasing slopes and there are at most r segments, where r is the number of receivers. So given a real value y, we can find corresponding cost of v_k in time $O(\log r)$. The algorithm to find these line segments is as follows.

ALGORITHM 7. ***Find the ratio-cost function*** $y = \mathcal{R}_{v_j}(x)$
If $j = k$ then apply the following procedures:

1. *Apply step $1, 2, 3$ of algorithm 6 to get $L(v_k)$.*

2. *Set number of legs to $t = 1$, lower bound $lb = 0$ and upper bound $ub = 0$.*

3. *While $t < r$*

   (a) $ub = (t+1) \times \omega(L_{t+1}(v_k)) - \Omega_{t+1}(L(v_k))$.

   (b) $y = \frac{\Omega_t(L(v_k))+x}{t}$ *for* $x \in [lb, ub)$.

   (c) *Set* $lb = ub$ *and* $t = t + 1$.

   (d) $y = \frac{\Omega_r(L(v_k))+x}{r}$ *for* $x \in [lb, \infty)$.

Otherwise, we do as follows:

1. *Remove terminal v_k, apply algorithm 6 to find $S^{-v_k}(v_j)$.*

2. *Find the shortest path with terminal v_k from v_j to every receiver, sort these paths according to their length in a descending order, say sequence*

$$L^{v_k}(v_j) = \{L_1^{v_k}(v_j), L_2^{v_k}(v_j), \cdots, L_r^{v_k}(v_j)\}.$$

Here, r is the number of terminals, and $\omega(L_i^{v_k}(v_j))$ is the sum of terminals on path $L_i^{v_k}(v_j)$ excluding *terminal v_k.*

3. *t is the index for branches in $L^{v_k}(v_j)$ and l is the index for paths in $L^{-v_k}(v_j)$.*

4. *For $L_t^{v_k}(v_j)$ ($1 \leq t \leq r$), there may exists on or more branches in $L^{-v_k}(v_j)$ such that they have common terminals with $L_t^{v_k}(v_j)$. If there are more than one such branches, choose the branch with the minimum cost, say $L_l^{-v_k}(v_j)$. We defined upper bound $upper_t$ for $L_t^{v_k}(v_j)$ equals $\omega(L_l^{-v_k}(v_j)) - \omega(L_t^{v_k}(v_j))$. If there does not exist such branch we set $upper_t = \infty$.*

5. *Initialize $t = 1$, $l = 1$, lower bound $lb = 0$ and upper bound $ub = 0$. Then apply the following algorithm:*
For $t = 1$ to r do {

   (a) *While $lb < upper_t$ do*

      i. *Set $l = 1$*

      ii. *Obtain a new sequence $LT^{-v_k}(v_j)$ from $L^{-v_k}(v_j)$ by removing all branches that has common nodes with $L_t^{v_k}(v_j)$. Let rt be the number of branches in sequence $LT^{-v_k}(v_j)$. For simplicity of our notation, we let $\Delta_l^{-v_k}(v_j) = l \star \omega(LT_l^{-v_k}(v_j)) - \Omega_{l-1}(LT^{-v_k}(v_j)) - c_j$.*

      iii. *While $l \leq rt$ do*
         While $\omega(L_t^{v_k}(v_j)) + lb > \Delta_l^{-v_k}(v_j)$ and $l \leq rt$
            $l = l + 1$
         If $l \leq rt$ then
            Set $ub = \Delta_l^{-v_k}(v_j) - \omega(L_t^{v_k}(v_j)$
         If $ub \geq upper_t$ break;
         Set $y = \frac{\Omega_{l-1}(LT^{-v_k}(v_j))+\omega(LT_t^{v_k}(v_j))+x}{l}$ for $x \in [lb, ub)$

      iv. *Set $lb = ub$.*

      v. *Set $l = l + 1$.*

   (b) *Set $y = \frac{\Omega_{l-1}(LT^{-v_k}(v_j))+\omega(L_t^{v_k}(v_j))+x}{l}$ for $x \in [lb, upper_t)$.*

   (c) *Set $lb = upper_t$.*

}

Given a real value x, the corresponding cost for terminal v_k is denoted by $\mathcal{R}_{v_j}^{-1}(x)$. Finally, we give the algorithm to find value $B_k(d_{-k})$.

ALGORITHM 8. ***Find*** $B_k(d_{-k})$

1. *Remove terminal v_k and find the multicast tree by using spider structure.*

2. *For every round i in the first step, we have a graph called G_i and a selected spider with ratio $\rho_i^{-v_k}$. Adding node v_k and all its incident edges to G_i get graph G_i'.*

3. *Find the function $y = \mathcal{R}_{v_j}^{-1}(x)$ for every terminal v_j in graph G_i' using algorithm 7.*

4. *Calculate $B_k^r(d_{-k}) = \max_{v_j \in V(G_i')} \{\mathcal{R}_{v_j}^{-1}(\rho_i^{-v_k})\}$.*

5. $B_k(d_{-k}) = \max_{1 \leq i \leq r} B_k^i(d_{-k})$

The correctness of the algorithm is omitted due to space limit, please refer to the full version of this paper for details.

4.2.4 Computational complexity

If we use Algorithm 5 to find $NST(d)$, every round we need time $O(rn \log n + rm)$, where r is the number of receivers. Notice there are at most r rounds, so the overall time complexity is $O(r^2 \log n + r^2 m)$. For every node $v_k \in NST(d)$, if we apply Algorithm 6 to calculate the payment, it is not difficult to get time complexity $O(rn \log n + rm)$ for each round. Thus, it takes time $O(r^2 n \log n + r^2 m)$ to find the payment for a single node $v_k \in$

$NST(d)$. In the worst case, there could be up to $O(n)$ terminals in $NST(d)$, so overall time complexity is $O(r^2n^2 \log n + r^2nm)$, which is quite expensive. Finding a more efficient way to reduce the time complexity will be one of our future works.

5. SIMULATION STUDIES

Remember that the payment of our structure is always larger than or equals the structure's actually cost. For a structure H, let $c(H)$ be its cost and $p_s(H)$ be the payment of scheme s based on this structure. We define the overpayment ratio of the payment scheme s based on structure H as

$$OR_s(H) = \frac{p_s(H)}{c(H)}. \quad (6)$$

When it is clear from the context, we often simplify the notation as $OR(H)$.

Actually, there are some other definitions about overpayment ratio in the literature. In [2], the authors propose to compare the payment $p(H)$ with the cost of the new structure obtained from the graph $G - H$, i.e., removing H from the original graph G. Here, we only focus our attention on the overpayment ratio defined in (6).

We conducted extensive simulations to study the overpayment ratio of various schemes proposed in this paper. In our experiments, we will compare the performance of different structures proposed according to three different metrics: actual cost, total payment and overpayment ratio. Notice that, it is meaningless to compare the performance of structures for link weighted network with these structures for node weighted networks. Therefore, we consider LCPT(link weighted version), PMST and LST as one group for link weighted networks and LCPT(node weighted version), VMST and NST as another group for node weighted networks. Figure 6 and Figure 7 show the different multicast structures when the original graph is a unit disk graph (UDG). Here, the grey nodes are receivers.

5.1 Fixed Transmission Range and Fixed Number of Receivers

In the first experiment, we randomly generate n terminals uniformly in a $2000ft \times 2000ft$ region. The transmission range $range$ of each terminal is set to $300ft$. For a link weighted graph, we assume the power needed to deliver a packet on e_i is $c_i(\frac{|e_i|}{100})^\kappa$, where κ is a value between 2 and 5. In our experiment $\kappa = 2.5$ and c_i is randomly drawn from the uniform distribution between 1 and 10. For a node weighted network, the weight of a node i is $c_i * 3^\kappa$ where c_i is randomly selected from a power level between 1 and 10. We vary the number of terminals in this region from 100 to 320, and fix the number of sender to 1 and receivers to 15. For a specific number of terminals, we generate 100 different networks, and compare the performance of different structures according to six different metrics: average cost(AC), maximum cost(MC), average payment(AP) and maximum payment(MP), average overpayment ratio(AOR) and maximum overpayment ratio(MOR).

For link weighted network, as shown in the upper figures of Figure 8, all structures' cost and payment decrease dramatically as the number of terminals increase. The PMST structure is the maximum for both cost, payment and overpayment ratio. But one advantage is that PMST is a shared based tree, which means it can be shared by all receivers/senders on the tree. LCPT is the most commonly used structure for source based tree, and it does win over the other two structures regarding AOR and MOR in our experiment. But in practice, people tend to care more about the actual cost (the so called "social efficiency") and the total payment. From this aspect, LST is the best candidate. Similar to LCPT, LST only needs information of LCP between terminals which can be obtained from the routing table for unicast. Thus, LST can also be implemented in a distrusted way but with more computational cost compared to LCPT.

For node weighted network, as shown in the lower figures of Figure 8, all structures' cost and payment also decrease as the number of terminals increase. Notice for VMST structure, we assume all receivers(senders) will relay message for free, so in order to compare the performance of these structure in a fair way, we set all receivers' private cost to 0 for both LCPT and NST structure. Unlike in link weighted network, the cost and payment of VMST and NST are much lower than the cost and payment of LCPT although the previous two are shared based tree. Like we expected, due to the lowe cost of the VMST and NST structures, the max overpayment ratio of these two structures are very unsteady and much high than the max overpayment ratio of LCPT.

5.2 Random Transmission Range and Fixed Number of Receivers

In our second experiment, we vary the transmission range of each wireless node from $100ft$ to $500ft$.

For link weighted network, the cost c_i of a link e_i is $(c_1 + c_2(\frac{|e_i|}{100})^\kappa)/10$, where c_1 takes value from 300 to 500 and c_2 takes value from 10 to 50. For node weighted network, the cost c_i of a terminal v_i is $(c_1 + c_2(\frac{r_i}{100})^\kappa)/10$, where c_1 takes value from 300 to 500, c_2 takes value from 10 to 50 and r_i is v_i's transmission range. The ranges of c_1 and c_2 we used here reflects the actual power cost in one second of a node to send data at $2Mbps$ rate.

Similar to the fixed transmission experiment, we vary the number of terminals in the region from 100 to 320, and fixed number of sender to 1 and receivers to 15. For a specific number of terminals, we generate 100 different networks, and compare the average cost, maximum cost, average payment and maximum payment, average overpayment ratio and maximum payment ratio.

Figure 9 shows the similar result for both link weighted network and node weighted network as the fixed transmission range experiments.

5.3 Random Transmission Range and Variable Number of Receivers

For structure H, we define cost density $CD(H) = \frac{c(H)}{r}$ and payment density $PD(H) = \frac{p(H)}{r}$, where r is the number of terminals in structure H.

In this experiment, we study the relationship between average cost(AC), average payment(AP), average overpayment ratio(AOR), average cost density(ACD), average payment density(APD) and the number of the terminals. We use the same power cost model in the previous experiment and the number of nodes in the region is set to 200. We vary the number of receivers from 5, 10, 20, $\cdots$ to 50.

Figure 10 shows that when the number of the receivers increases, under most circumstance, the overall payment and cost increase while the cost and payment for every terminal decrease. One exception is for node weighted network. Notice in node weighted network, we set all terminals' cost to 0, so it is naturally to expected that when the number of terminals larger than some threshold, even the total cost and payment will decrease. This experiment shows that more terminals in a multicast group can incur a lower cost and payment per terminal, which is one of the attractive properties of multicast.

6. CONCLUSION

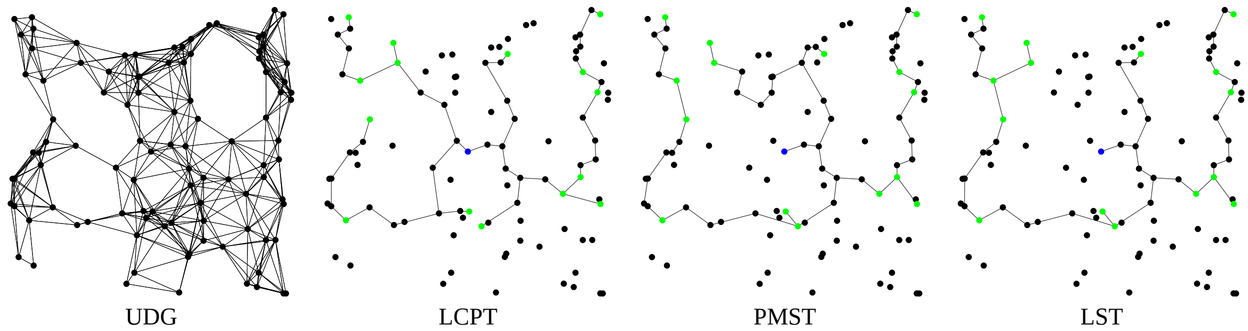


Figure 6: Multicast Structures for Link Weighted Network

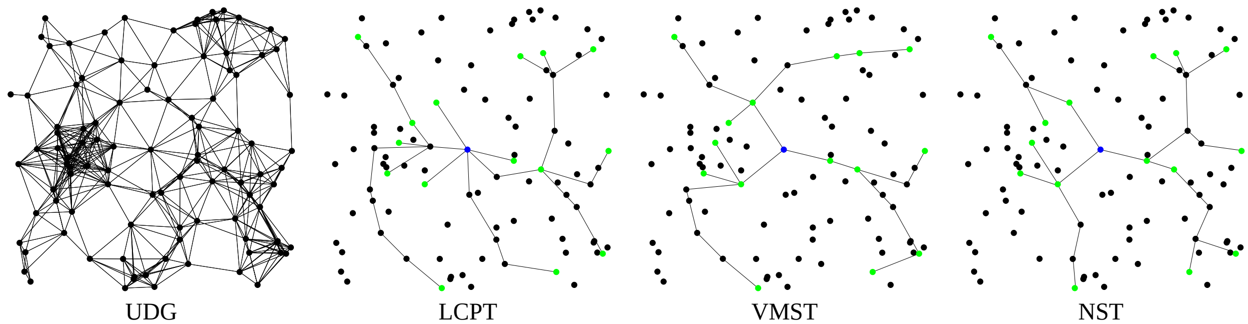


Figure 7: Multicast Structures for Node Weighted Network

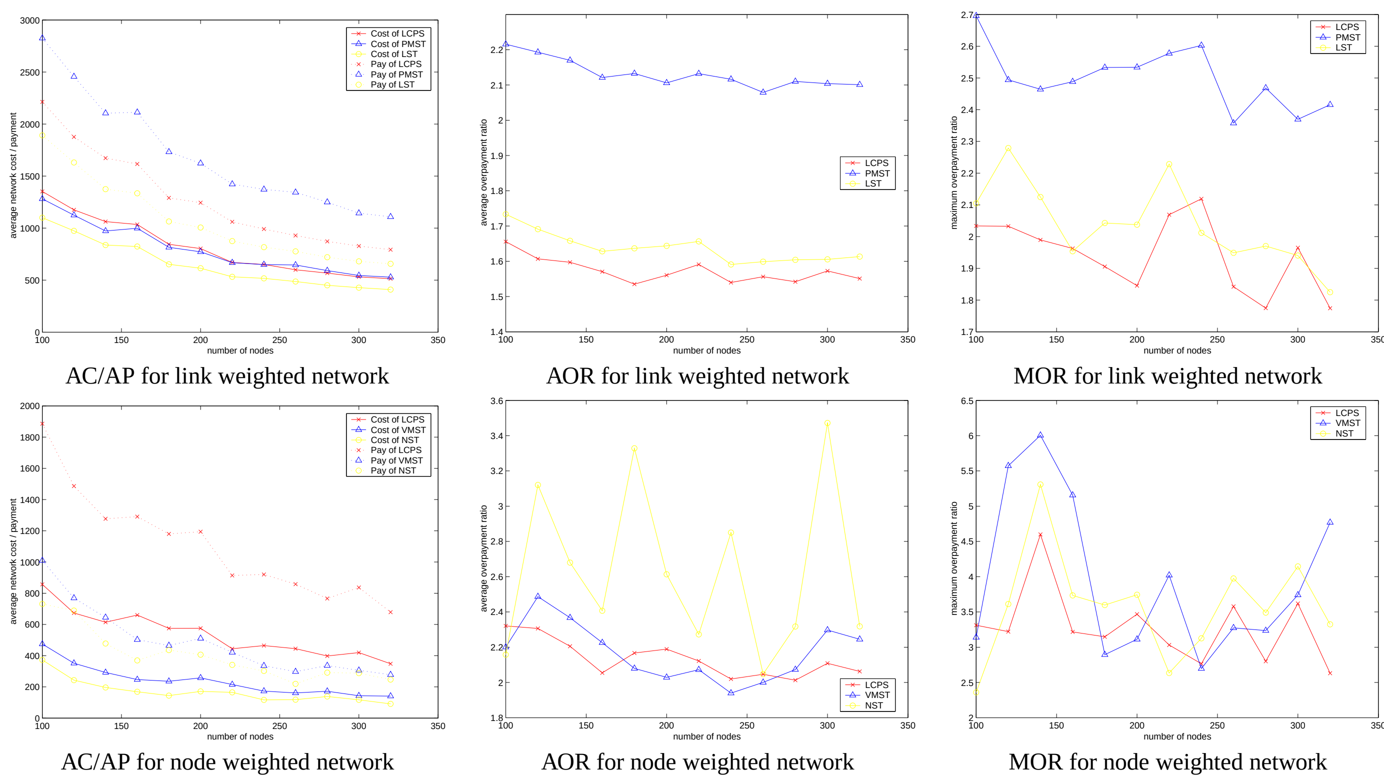
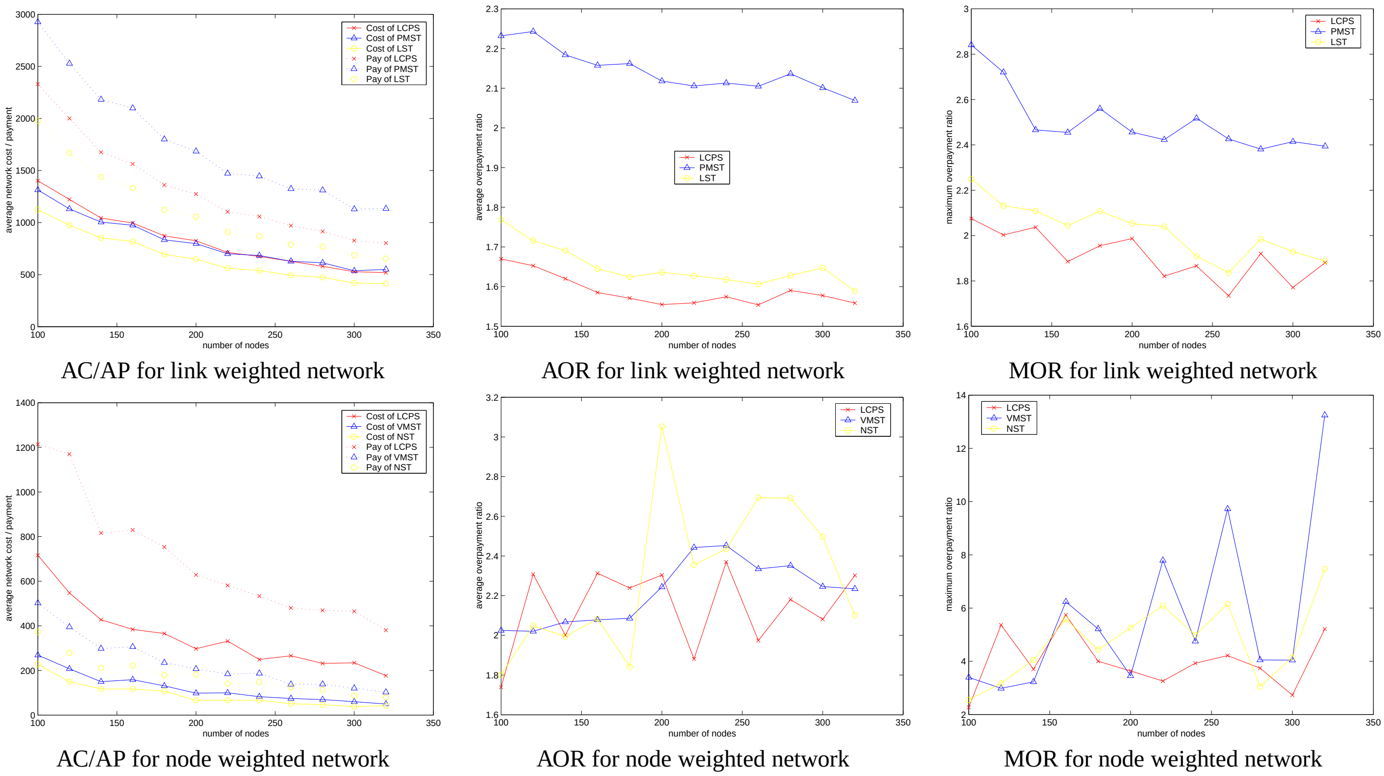


Figure 8: Results when the number of nodes in the networks are different (from 100 to 320) for link weighted structures and node weighted structures. Here, we fix the transmission range to $300ft$.



AC/AP for link weighted network

AOR for link weighted network

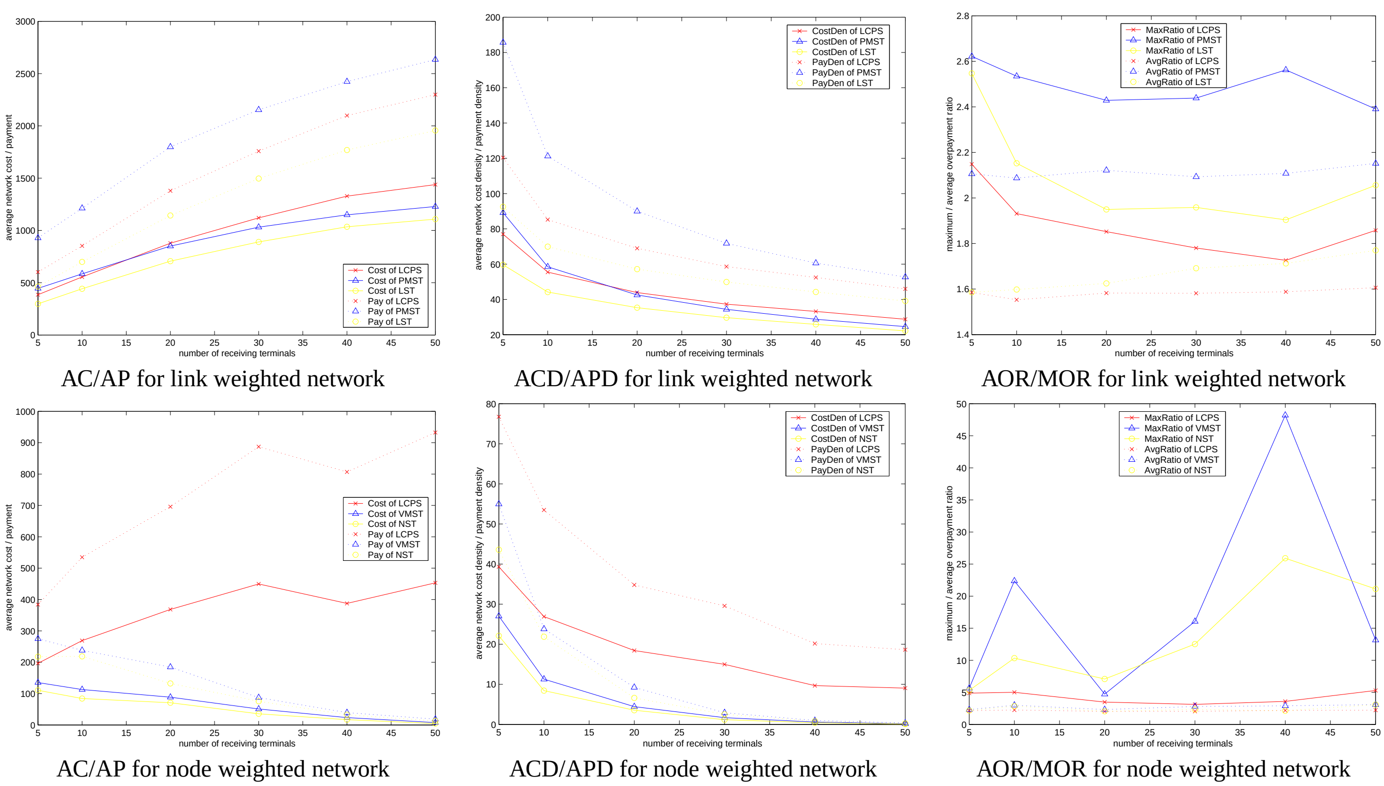
MOR for link weighted network

AC/AP for node weighted network

AOR for node weighted network

MOR for node weighted network

Figure 9: Results when the number of nodes in the networks are different (from 100 to 320) for link weighted structures and node weighted structures. Here, we randomly set the transmission range from $100ft$ to $500ft$.



AC/AP for link weighted network

ACD/APD for link weighted network

AOR/MOR for link weighted network

AC/AP for node weighted network

ACD/APD for node weighted network

AOR/MOR for node weighted network

Figure 10: Results when the number of receivers in the networks are different (from 10 to 50) for both link weighted and node weighted structures. Again, we randomly set the transmission range from $100ft$ to $500ft$.

In this paper, we studied how to conduct efficient multicast in *selfish* wireless networks by assuming that each wireless terminal or communication link will incur a privately known cost when it has to transit some data. For each of the widely used structures for multicast, we designed a strategyproof multicast mechanism such that each agent maximizes its profit when it truthfully reports its cost. The structures studied in this paper are least cost path tree, pruning minimum spanning tree, virtual minimum spanning tree and the edge(node) weighted Steiner tree. Extensive simulations were conducted to study the practical performances of the proposed protocols.

Notice that the payment to each selfish agent is at least its declared cost. This is necessary to ensure that the selfish agent is truthful. Clearly, agents will not participate if we pay less what their true cost are. If we pay the amount the agent asked for, an agent will have incentives to lie by asking more than its actual cost. In all our payment schemes, each agent already maximizes its profit when it reports its true cost even it knows the costs of all other agents! Notice that in the paper only the payment to one session is discussed. When the session is to be repeated, a natural question is how much we should pay for later sessions? One may argue that we only have to pay each agent its true cost for later sessions. Unfortunately, this will not work for selfish agents. When an agent knows that its payment will be its actual cost for later sessions, it could lie its cost upward. By doing this, it may lose for the first session, but the gains in the later sessions will compensate the initial loss.

There are several unsolved challenges we left as future works. First, we would like to design algorithms that can compute these payments in asymptotically optimum time complexities. Second, in this paper, we only studied the tree-based structures for multicast. Practically, mesh-based structures maybe more needed for wireless networks to improve the fault tolerance of the multicast. We would like to know whether we can design a strategyproof multicast mechanism for some mesh-based structures used for multicast. Third, all of our tree construction and payment calculation are performed in a centralized way, we would like to study how to design some distributed algorithm for it.

This paper will lay down a building block for further researches in designing truthful routing protocols for selfish wireless networks. In all our protocols, we assumed that the receivers will always relay the data packets for other receivers for free, and the source node of the multicast will pay the relay nodes to compensate their cost. The source node will not charge the receivers for getting the data. As future works, we have to consider the budget balance of the source node if the receivers have to pay the source node for getting the data; we also have to consider fairness of the payment sharing when the receivers will share the total payments to all relay nodes on the multicast structure. Notice that this is different from the cost-sharing studied in [8], in which they assumed a fixed multicast tree, and the link cost is publicly known, then they showed how to share the total link cost among receivers. Another important task is to study how to implement the protocols proposed in this paper in a distributed manner. Notice that, in [22, 7], distributed methods have been developed for truthful unicast using some cryptography primitives.

7. ACKNOWLEDGMENT

The authors are greatful for a variety of valuable comments from the reviwers and the shepherd Li Li at Bell lab, that improves the presentation of the paper.

8. REFERENCES

[1] ANDEREGG, L., AND EIDENBENZ, S. Ad hoc-vcg: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (2003), ACM Press, pp. 245–259.

[2] ARCHER, A., AND L. TARDOS. Frugal path mechanisms. In *SODA* (2002), pp. 991–998.

[3] BLAZEVIC, L., BUTTYAN, L., CAPKUN, S., GIORDANO, S., HUBAUX, J. P., AND BOUDEC, J. Y. L. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications Magazine 39*, 6 (June 2001).

[4] BUTTYAN, L., AND HUBAUX, J. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications 5*, 8 (October 2003).

[5] BUTTYAN, L., AND HUBAUX, J. P. Enforcing service availability in mobile ad-hoc wans. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing* (2000), pp. 87–96.

[6] CLARKE, E. H. Multipart pricing of public goods. *Public Choice* (1971), 17–33.

[7] FEIGENBAUM, J., PAPADIMITRIOU, C., SAMI, R., AND SHENKER, S. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing.* (2002), pp. 173–182.

[8] FEIGENBAUM, J., PAPADIMITRIOU, C. H., AND SHENKER, S. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences 63*, 1 (2001), 21–41.

[9] GREEN, J., AND LAFFONT, J. J. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica* (1977), 427–438.

[10] GROVES, T. Incentives in teams. *Econometrica* (1973), 617–631.

[11] GUHA, S., AND KHULLER, S. Improved methods for approximating node weighted steiner trees and connected dominating sets. In *Foundations of Software Technology and Theoretical Computer Science* (1998), pp. 54–65.

[12] JAKOBSSON, M., HUBAUX, J.-P., AND BUTTYAN, L. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In *Proceedings of Financial Cryptography* (2003).

[13] KLEIN, P., AND RAVI, R. A nearly best-possible approximation algorithm for node-weighted steiner trees. Tech. Rep. CS-92-54, 1992.

[14] MARTI, S., GIULI, T. J., LAI, K., AND BAKER, M. Mitigating routing misbe-havior in mobile ad hoc networks. In *Proc. of MobiCom* (2000).

[15] NISAN, N., AND RONEN, A. Algorithmic mechanism design. In *Proc. 31st Annual Symposium on Theory of Computing (STOC99)* (1999), pp. 129–140.

[16] NISAN, N., AND RONEN, A. Computationally feasible VCG mechanisms. In *ACM Conference on Electronic Commerce* (2000), pp. 242–252.

[17] ROBINS, G., AND ZELIKOVSKY, A. Improved steiner tree approximation in graphs. In *Symposium on Discrete Algorithms* (2000), pp. 770–779.

[18] SRINIVASAN, V., NUGGEHALLI, P., CHIASSERINI, C. F., AND RAO, R. R. Energy efficiency of ad hoc wireless networks with selfish users. In *European Wireless Conference 2002 (EW2002)* (2002).

[19] SRINIVASAN, V., NUGGEHALLI, P., CHIASSERINI, C. F., AND RAO, R. R. Cooperation in wireless ad hoc wireless networks. In *IEEE Infocom* (2003).

[20] TAKAHASHI, H., AND MATSUYAMA, A. An approximate solution for the steiner problem in graphs. *Math .Jap. 24* (1980), 573–577.

[21] VICKREY, W. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance* (1961), 8–37.

[22] WANG, W., AND LI, X.-Y. Truthful low-cost unicast in selfish wireless networks. In *4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN) of IPDPS 2004* (2004).

[23] WANG, W., LI, X.-Y., AND SUN, Z. Design multicast protocols for non-cooperative networks, 2004. Submitted for publication.

[24] ZHONG, S., LI, L., LIU, Y., AND YANG, Y. R. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks — an integrated approach using game theoretical and cryptographic techniques. Tech. Rep. YALEU/DCS/TR-1286, Yale University, Computer Science Department, 2004.