k-Anycast Game in Selfish Networks

Weizhao Wang⋆ Xiang-Yang Li⋆ Ophir Frieder⋆

Abstract— **Conventionally, many network routing protocols assumed that every network node will forward data packets for other nodes without any deviation. However, this may not be true when nodes are owned by individual users. In this paper, we propose a new routing protocol, called k-anycast routing, that works well even network nodes are assumed to be selfish. In our protocol, the source node will first find a tree that spans k receivers out of a set of possible receivers and pay relay nodes to compensate their costs. We prove that every relay node will maximize its profit when it follows the routing protocol and truthfully declares its actual cost.**

I. Introduction

Conventionally, network routing protocols assumed that each network link/node will forward data packets without any deviation. However, this may not be true when they are owned by individual users. For example, consider a library wireless ad hoc network where each wireless devise is owned by individual student. The wireless devise is often powered by batteries only, thus, it is not in the best interest of a node to forward data packets for other users. When a node refuses to relay data for other nodes when it is supposed to do so by a prescribed routing protocol, the network performance will degrade, and the network connectivity may be broken *de facto*. Thus, we need design a routing protocol that works even when all nodes are assumed to be selfish: it will maximize its own benefit only. In this paper, we assume that each network link/node has a *privately known* cost of providing service for others. It will provide the service only when it gets a payment enough to compensate its cost.

How to achieve cooperation among terminals in selfish networks was previously addressed in [1], [2], [3], [4], [5]. A 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 sends a packet as originator and increased by one when the terminal forwards a packet. Srinivasan *et al.* [6], [7] proposed several acceptance algorithms for each wireless node to decide whether to relay data for other nodes.

Recently, incentive based methods [8], [9], [10], [11], [17] have been proposed for routing in a non-cooperative setting. Majority of such schemes are based on a well-known family of VCG mechanisms (named after Vickrey [12], Clarke [13], and Groves [14]). Each selfish node is paid a monetary value to compensate its cost incurred by providing service to other nodes. VCG mechanisms do have limit: they are applicable only if we can find the optimal solution that maximizes (or minimizes) an utilitarian objective function. For example, VCG mechanisms cannot be used to solve the multicast problem [15] since it is NP-hard to find the minimum cost multicast tree.

In this paper, we first propose a new routing method called k-anycast, which is an extension of anycast routing. Unlike multicast, which is the communication between a single sender and a given multiple receivers, and unicast, which is communication between a single sender and a single receiver, anycast is a communication between a single sender and the nearest receiver among a group of receivers. Anycast could happens both in network and application layer. One common application of anycast is router table updating: one router initiates an update of a router table for a group of routers by sending the data to the nearest router. That router received the data sends the data to its nearest router that has not received the data yet. Repeat this process until all the routers in that group have received the data. With the support by IPv6, anycast is expected to be deployed more widely in the near future. Unfortunately, like unicast and multicast, anycast has its own problem. Let us reconsider the router table updating scenario. Remember that when a router receives the data, it should anycast to its nearest router that has not received the data yet. What if the router goes down or reboots before it sends/receives the date? Obviously, this process will stuck which results in that part of the routers will not be able to receive the data. Another concern about anycast is that the updating process is serialized, which may take a long time.

Now we consider another scenario in application layer: a group of users wants to download a movie via some Peer-to-Peer file-sharing systems, i.e, BitTorrent. Due to the large population of group members, every member usually retrieves the movie from some of the members. In order to speed up the download, the source will choose these members that are not far away. Notice in both applications mentioned above, the source need deliver the data to more than one but not all receivers. Thus, we design a new routing method called k-*anycast* to solve these problems. We formally define the k-anycast problems as following: assume that there is a source node s and a group Q of potential receivers, we need build a tree rooted at s that spans at least k nodes in Q. Here k could be any value between 1 and $|Q|$. If $k = 1$ then it is the traditional anycast problem. When $k = |Q|$, it becomes the multicast problem.

Truthful incentive-based routing protocols have been proposed for unicast [16], [17], [10], and for multicast [18], [15], [19] in selfish networks. For k-anycast problem, if $k > 1$, as we will show later, a VCG mechanism doesn't work. We first propose a new routing method called k-anycast and then design a non-VCG truthful payment scheme based on this new routing method. Notice that, in order to achieve the truthfulness, we does have to pay a compensation to a relay node at least its actual cost. To study how much we "overpay" relay nodes, we conduct extensive simulations on the ratio of the total payment node over the total costs of all relay nodes.

The rest of the paper is organized as follows. First, we introduce some preliminaries and related works in Section II. We propose a strategy-proof routing protocol for k-anycast in Section III. Simulation results are presented in Section IV. We conclude our paper in Section V with possible future work.

⋆Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Chicago. Email: wangwei4@iit.edu, xli@cs.iit.edu, ophir@cs.iit.edu.

II. Preliminaries and Priori Art

A. Preliminaries

In this paper, we assume the network nodes or links are selfish and rational. Here an agent is called *selfish* if it will always try to maximize its gain; an agent is said to be *rational* if it responds to well-defined incentives and will deviate from the protocol only if it improves its gain. A standard model in the literature for the design and analysis of scenarios in which the participants are selfish and rational is as follows.

Assume that there are n agents, which could be wireless devices in a wireless ad hoc networks, computers in peer-to-peer networks, or network links in networks. Each agent i, for $i \in \{1, \cdots, n\}$, has some private information t_i, called its *type*. Here, the type t_i could be its minimum cost to forward a unit data in a network environment. Then the set of n agents 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* $o = \mathcal{O}(a)$ and a *payment* vector $p = (p_1, \cdots, p_n)$, where $p_i = p_i(a)$ is the money given to 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. Then the utility of agent i at the outcome of the game, given its preferences t_i and strategies a selected by all agents, is $u_i(t_i, o) = v_i(t_i, o) + p_i$.

Let $a_{-i} = (a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n)$ denote the vector of strategies of all other agents except i. A strategy a_i is called *dominant strategy* if it maximizes the utility for all possible strategies of all other agents, 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. Thus, an rational agent always tries to maximize its utility u_i by finding its dominant strategy.

In this paper, the strategy of an agent is to report its type. A mechanism is *incentive compatible* (IC) if reporting its *true* type t_i is one of the dominant strategies. A mechanism satisfies *individual rationality* or *voluntary participation* if the agent's utility of participating is not less than the utility of the agent if it did not participate.

Arguably the most important positive result in mechanism design is what is usually called the generalized Vickrey-Clarke-Groves (VCG) mechanism by Vickrey [12], Clarke [13], and Groves [14]. The VCG mechanism applies to maximization problems with a *utilitarian* objective function $g(o, t)$, i.e., $g(o, t) = \sum_i v_i(t_i, o)$. A direct revelation mechanism $M = (\mathcal{O}(t), p(t))$ belongs to the VCG family if (1) the output $\mathcal{O}(t)$ maximizes the objective function $g(o, t) = \sum_i v_i(t_i, o)$, and (2) the payment to an 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}.

It is proved by Groves [14] that a VCG mechanism satisfies IC property. Green and Laffont [20] proved that, under mild assumptions, VCG mechanisms are the *only* mechanism satisfying IC for utilitarian problems. An output function of a VCG mechanism is required to maximize the objective function. 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.

B. Priori Arts on Unicast Routing

Consider any communication network $G = (V, E, c)$, where $V = \{v_1, \cdots, v_n\}$ is the set of communication terminals, $E = \{e_1, e_2, \cdots, e_m\}$ are the set of links, and c is the cost vector of links. Remember that c_i is private to link i in selfish networks. Given a source node s and a destination node v_i, we want to find the path with the minimum total cost. This path is known as the shortest path, denoted as $\mathsf{LCP}(s, v_i, d)$, which can be found by Dijkstra's Algorithm. Consider all paths from source s to destination v_i, they can be divided into two categories: with edge e_j or not. The path having the minimum length among paths with edge e_k is denoted as $\mathsf{LCP}_{e_k}(s, v_i, d)$; and the path having the minimum length among these paths without edge e_k is denoted as $\mathsf{LCP}_{-e_k}(s, v_i, d)$. Fixed the source,for simplicity we denote the length of $\mathsf{LCP}(s, v_i, d)$ as $\mathcal{L}(i, d)$, the length of $\mathsf{LCP}_{e_k}(s, t, d)$ as $\mathcal{L}_{e_k}(i, d)$, and the length of $\mathsf{LCP}_{-e_k}(s, i, d)$ as $\mathcal{L}_{-e_k}(i, d)$ if no confusion is caused. In [16], Nisan and Ronen [16] provided a polynomial-time strategyproof mechanism for optimal unicast route selection in a centralized computational model. The payment to link $e_j \in \mathsf{LCP}(s, v_i, d)$ is

$$p_j(d) = \mathcal{L}_{-e_k}(i, d) - \mathcal{L}_{-e_k}(i, d|^j 0)$$

And the payment to link $e_j \in \mathsf{LCP}(s, v_i, d)$ is 0. Since this payment scheme is a VCG mechanism, so it is truthful.

Feigenbaum ***et. al*** [10] 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. For any two nodes i and j of the network, $T_{i,j}$ is the intensity of the traffic (number of packets) originating from i and destined for node j. Their strategyproof mechanism again is essentially a VCG mechanism. They gave a distributed method such that each node i can compute a payment $p^k_{ij} > 0$ to a node k for carrying the transit traffic from node i to node j if node k is on the least cost path $\mathsf{LCP}(i, j)$.

III. k-Anycast Game

A. Problem Statement

Consider any communication network $G = (V, E, c)$, where $V = \{v_1, \cdots, v_n\}$ is the set of communication terminals, $E = \{e_1, e_2, \cdots, e_m\}$ are the set of links, and c is the cost vector of links. Given a source node s and a set of possible receivers $Q = \{q_1, q_2, \cdots, q_r\} \subset V$, the *k-anycast problem* $1 \leq k \leq q$ is to select k terminals R from Q and build a *tree* that spans these k receivers R. In different applications, we may want to construct a k-anycast tree that optimizes different objectives. For example, we may want to minimize the total cost or minimize the maximum latency of the k-anycast tree. Here, we will consider the k-anycast tree whose maximum length (or called cost in this paper) is minimized.

Given a graph G, we use $\omega(G)$ to denote the total cost of all links in this graph. If we change the cost of a link e_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 link e_i from the network, we denote it as $c|^i \infty$, i.e., the cost of link e_i is assumed to be infinity. Sometimes we use $G\backslash e_i$ to denote the network without link e_i. 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.

In our protocol, a link e_i is required to declare a cost d_i of relaying the message. Based on the declared cost profile d, we should first select the k terminals among Q, and construct the k-anycast tree, then decide the payment for all agents. The utility of an agent is its payment received, minus its cost if it is selected in the k-anycast tree.

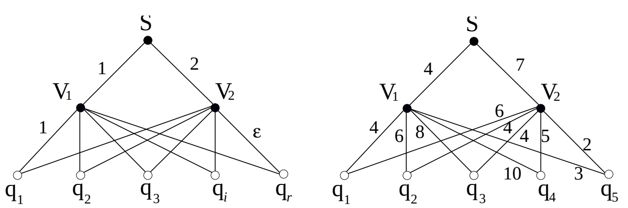
In this paper, we construct the k-anycast tree as follows. First, sort the distances from the source node s to all receivers. For the simplicity of notations, we assume that $\mathcal{L}(i, d) < \mathcal{L}(j, d)$ for any two nodes q_i and q_j with $i < j$. The final tree is then the union of k paths $\mathsf{LCP}(s, q_j, d)$ for $1 \leq j \leq k$, i.e., the first k-shortest paths. We call the final tree as k least cost paths star, denoted as LCPS_k. For simplicity of our notations, let $Q_k(d)$ be the k receivers selected by the method LCPS_k. Following, we will discuss how to compensate the relay links such that they will relay the data out of their own interests and they will declare their costs truthfully.

B. VCG Mechanism is not strategyproof

Intuitively, we would use the VCG payment scheme in conjunction with the k-anycast tree structure LCPS_k as follows. The payment to a link that is not in LCPS_k is 0. And the payment $p_i(d)$ to a link e_i in LCPS_k is

$$p_i(d) = \omega(LCPS_k(d|^i\infty)) - \omega(LCPS_k(d)) + d_i.$$

However, this simple application of VCG mechanisms is not truthful. We show this by an example that the above payment scheme is not strategyproof for any k. Our example will show that the payment of some selected link e_i is negative even it reveal its true cost.



VCG mechanism for LCPS_k is not truthful.

An example of our truthful payment scheme based on LCPS_K.

Fig. 1. Payment Scheme for LCPS_k

The first part of Figure 1 illustrates the example with terminal s being the source node and q_i ($1 \leq i \leq r$) are possible receivers. The cost of link sv_1 and links v_1q_i ($1 \leq i \leq r$) are 1. The cost of link sv_2 is 2 and the cost of links v_2q_{i+1} are ϵ, where ϵ is a sufficiently small positive real number. For any $1 < k \leq r$, it is not difficult to show that, tree LCPS_k is just formed by the link sv_1 plus any k links in the set of links $\{v_1q_1, v_1q_2, \cdots, v_1q_r\}$, whose weight is $1 + k * 1 = k + 1$. Now remove link $e_1 = sv_1$, tree LCPS_k becomes link sv_2 plus any k links in $\{v_2q_1, v_2q_2, \cdots, v_2q_r\}$, whose weight is $2 + k\epsilon$. Thus, the payment to edge sv_1 according to VCG mechanism is $(2 + k\epsilon) - k - 1 + 1 = k\epsilon - k + 2$, and edge sv_1's utility is $k\epsilon - k + 1 < 0$ when $\epsilon < \frac{k-1}{k}$. This violates the individual rationality, which means that the payment based on VCG is not truthful.

C. Strategyproof payment scheme

In subsection III-B, we shown that if we apply VCG mechanism on LCPS_k, it is not strategyproof. In this subsection, we will present a non-VCG strategyproof mechanism using tree LCPS_k. Intuitively, we will pay link e_j the mount that equals to the maximum cost it could declare while it is still selected in LCPS_k. To find this maximum cost for e_j, we will construct two sets of paths: one is the set of shortest paths to all receivers containing link e_j, while the other one is the set of shortest paths to all receivers without using link e_j.

Algorithm 1: **Strategyproof payment scheme for link** e_j

1. For each receiver $q_i \in Q$, find the shortest path $\mathsf{LCP}_{e_j}(s, q_i, d)$ using link e_j. Sort all these shortest paths according to their costs in an ascending order. For simplicity, we assume that the sorting is denoted by an ordering σ_1, i.e., $\mathcal{L}_{e_j}(\sigma_1(t_1), d) \leq \mathcal{L}_{e_j}(\sigma_1(t_2), d)$ for any $1 \leq t_1 \leq t_2 \leq r$. Notice that here $\sigma_1(t)$ denotes that $\mathsf{LCP}(s, q_{\sigma_1(t)}, d)$ is the t-th longest path among all such shortest paths.
2. Similarly, for each receiver $q_i \in Q$, find the shortest path $\mathsf{LCP}_{-e_j}(s, q_i, d)$ *without* using link e_j. Sort all these shortest paths according to their costs in an ascending order. We assume that the sorting is denoted by another ordering σ_0, i.e., $\mathcal{L}_{e_j}(\sigma_0(t_1), d) \leq \mathcal{L}_{e_j}(\sigma_0(t_2), d)$ for any $1 \leq t_1 \leq t_2 \leq r$. Let $\Phi = \{\sigma_0(1), \sigma_0(2), \cdots, \sigma_0(k)\}$.
3. Find the smallest value α such that $\sigma_1(\alpha) \notin \Phi$.
4. Define two variables

$$\kappa_j = \max_{i=1}^{\alpha-1}\{\mathcal{L}_{-e_j}(\sigma_1(i), d) - \mathcal{L}_{e_j}(\sigma_1(i), d|^j 0)\} \quad (1)$$

$$\gamma_j = \mathcal{L}_{-e_j}(\sigma_0(k), d) - \mathcal{L}_{e_j}(\sigma_1(\alpha), d|^j 0) \quad (2)$$

5. Define η_j as

$$\eta_j = \max\{\gamma_j, \kappa_j, 0\}. \quad (3)$$

6. If $e_j \in LCPS_k(d)$ then it gets payment η_j; else it gets payment 0.

We first show how our payment scheme works by the following example illustrated in the second part of Figure 1. There are 5 receivers $q_1, q_2, \cdots, q_5$. Assume that $k = 3$. It is easy to see that LCPS_k is formed by links: sv_1, v_1q_1, v_1q_2, sv_2 and v_2q_5. The selected three receivers will be q_1, q_2, and q_5. Let us see what is the payment for link sv_1. The receivers sorted in increasing order of their shortest paths to the source node using link sv_1 are q_5, q_1, q_2, q_3, and q_4. The receivers sorted in increasing order of their shortest paths to the source node *without* using link $e = sv_1$ are q_5, q_2, q_3, q_4, and q_1. Then $\Phi = \{q_5, q_2, q_3\}$. Clearly, $\alpha = 2$ since q_1 is the first receiver not in Φ. Then $\kappa = \mathcal{L}_{-e}(\sigma_1(1), d) - \mathcal{L}_e(\sigma_1(1), d|^j 0) = \mathcal{L}_{-e}(q_5, d) - \mathcal{L}_e(q_5, d|^j 0) = 6$, and $\gamma = \mathcal{L}_{-e}(\sigma_0(3), d) - \mathcal{L}_e(\sigma_1(2), d|^j 0) = \mathcal{L}_{-e}(q_3, d) - \mathcal{L}_e(q_1, d|^j 0) = 11 - 4 = 7$. Thus, the payment to link sv_1 should be $7 = \max(6, 7, 0)$.

In order to prove payment calculated by Algorithm 1 is truthful we first prove the following two lemmas.

Lemma 1: If a link $e_j \in LCPS_k(d)$ then $d_j \leq \eta_j$.

Proof: If $e_j \in LCPS_k(d)$, there exists at least one i that satisfies $e_j \in \mathsf{LCP}(s, q_i, d)$ and $q_i \in Q_k(d)$. If there are more than one such indices, we choose the one that ranks first in the permutation σ_1. Without loss of generality, we assume such index is $\sigma_1(\beta)$, i.e., its rank is β in sorted shortest paths using link e_j. From the assumption that e_j is on $\mathsf{LCP}(s, q_{\sigma_1(\beta)}, d)$, we have $\mathcal{L}_{e_j}(\sigma_1(\beta), d) \leq \mathcal{L}_{-e_j}(\sigma_1(\beta), d)$, which implies

$$d_j \leq \mathcal{L}_{-e_j}(\sigma_1(\beta), d) - \mathcal{L}_{e_j}(\sigma_1(\beta), d|^j 0) \tag{4}$$

If $\beta < \alpha$, from inequality (4) and equation (1), we have $d_j \leq \kappa_j \leq \eta_j$.

So we only need consider the case when $\beta \geq \alpha$. We prove that $d_j \leq \eta_j$ by contradiction. For the sake of contradiction, assume that $d_j > \eta_j$. Then $d_j > \eta_j \geq \gamma_j = \mathcal{L}_{-e_k}(\sigma_0(k), d) - \mathcal{L}_{e_k}(\sigma_1(\alpha), d|^j 0)$. This implies

$$\mathcal{L}_{e_k}(\sigma_1(\alpha), d|^j 0) + d_j = \mathcal{L}_{e_k}(\sigma_1(\alpha), d) > \mathcal{L}_{-e_k}(\sigma_0(k), d)$$

Combining the above inequality and assumption $\beta \geq \alpha$, we have $\mathcal{L}_{-e_k}(\sigma_0(i), d) \leq \mathcal{L}_{-e_k}(\sigma_0(k), d) < \mathcal{L}_{e_k}(\sigma_1(\alpha), d) \leq \mathcal{L}_{e_k}(\sigma_1(\beta), d)$ for any $1 \leq i \leq k$. Remember that $e_j \in \mathsf{LCP}(s, q_{\sigma_1(\beta)}, d)$, thus $\sigma_1(\beta) \neq \sigma_0(i)$ for any $i \in [l, k]$. Therefore, $\sigma_1(\beta) \notin Q_k(d)$ since there are at least k paths to k different receivers, with length less than $\mathcal{L}_{e_k}(\sigma_1(\beta), d)$. It is a contradiction to that the path $\mathsf{LCP}(s, q_{\sigma_1(\beta)}, d)$ is used. This finishes our proof. ∎

A simple but useful observation about the tree LCPS$_k$ constructed by our method is

Observation 1: If $e_j \notin LCPS_k(d)$, then for any $q_i \in Q_k(d)$, $\mathsf{LCP}(s, q_i, d) = \mathsf{LCP}_{-e_j}(s, q_i, d)$.

Lemma 2: If $e_j \notin LCPS_k(d)$ then $d_j \geq \eta_j$

Proof: We prove by contradiction by assuming that $d_j < \eta_j$. Remember that $\eta_j = \max\{\gamma_j, \kappa_j, 0\}$. We disprove the assumption that $d_j < \eta_j$ by three cases.

Case 1: $\eta_j = 0$. This implies that $d_j < 0$, which is impossible from our protocol.

Case 2: $\eta_j = \kappa_j$. Remember that $\kappa_j = \max_{i=1}^{\alpha-1}\{\mathcal{L}_{-e_j}(\sigma_1(i), d) - \mathcal{L}_{e_j}(\sigma_1(i), d|^j 0)\}$. Without loss of generality we can assume $\kappa_j = \mathcal{L}_{-e_j}(\sigma_1(t), d) - \mathcal{L}_{e_j}(\sigma_1(t), d|^j 0)$, for some index $t \in [1, \alpha - 1]$. From the assumption we have $d_j < \eta_j = \kappa_j = \mathcal{L}_{-e_j}(\sigma_1(t), d) - \mathcal{L}_{e_j}(\sigma_1(t), d|^j 0)$. This implies that $\mathcal{L}_{e_j}(\sigma_1(t), d|^j 0) + d_j < \mathcal{L}_{-e_j}(\sigma_1(t), d)$. Consequently,

$$\mathcal{L}_{e_j}(\sigma_1(t), d) < \mathcal{L}_{-e_j}(\sigma_1(t), d).$$

Thus, $e_j \in \mathsf{LCP}(s, q_{\sigma_1(t)}, d)$, which implies that $q_{\sigma_1}(t) \notin Q_k(d)$.

Observe that $e_j \notin LCPS_k(d)$ implies that we will select Φ as the receivers to be spanned. Thus, $\sigma_1(t) \in \Phi$ implies that we have to select $q_{\sigma_1}(t)$, which is a contradiction to what we proved in the last paragraph.

Case 3: $\eta_j = \gamma_j$. Combining the above equation and the assumption that $d_j < \eta_j$, we get $\mathcal{L}_{e_j}(\sigma_1(\alpha), d) < \mathcal{L}_{-e_j}(\sigma_0(k), d)$. Remember that $e_j \notin LCPS_k(d)$ implies that $Q_k(d) = \Phi$. Thus, $\mathcal{L}_{e_j}(\sigma_1(\alpha), d) \geq \mathcal{L}_{-e_j}(\sigma_0(k), d)$, which is a contradiction. This finishes our proof. ∎

Now we ready to prove our payment scheme satisfies IC and IR.

Lemma 3: Payment scheme (1) satisfies IR.

Proof: If $e_j \notin LCPS_k(d)$ then e_j's valuation and payment are both 0, thus its utility is also 0.

If $e_j \in LCPS_k(d)$, then its payment is η_j. From lemma 1, we know $c_j \leq \eta_j$. Thus, its utility is $\eta_j - c_j \geq 0$. Thus, our payment scheme (1) satisfies IR. ∎

Lemma 4: Payment scheme (1) satisfies IC.

Proof: We show that link e_j won't increase its utility by lying it cost. Notice if the output whether e_j is selected doesn't change, then its utility doesn't change. Thus, we only need to distinguish the following two cases:

Case 1: Edge $e_j \in LCPS_k(d|^j c_j)$. when it declares its true cost c_j, and when it declares a cost as $\overline{c_j} > c_j$, $e_j \notin LCPS_k(d|^j \overline{c_j})$. From lemma 1 we have $c_j \leq \eta_j$. If e_j declares its true cost c_j, it will get utility $\eta_j(d_{-j}) - c_j \geq 0$. If e_j declares its cost as $\overline{c_j}$, then it will have utility 0. Thus, edge e_j will choose to reveal its true cost.

Notice that if it declares a cost as $\underline{c_j} < c_j$, e_j is still in $LCPS_k(d|^j \overline{c_j})$. Thus its utility does not change.

Case 2: Edge $e_j \notin LCPS_k(d|^j c_j)$ when it declares its true cost c_j, and when it declares its cost as $\underline{c_j} < c_j$, $e_j \in LCPS_k(d|^j \underline{c_j})$. From lemma 2 we have $c_j \geq \eta_j$. If e_j declares its true cost $\underline{c_j}$, it will get utility 0. If e_j declares its cost as $\underline{c_j}$, it will have utility $\eta_j - c_j \leq 0$. Thus, edge e_j will also choose to reveal its true cost in this case.

Notice that if it declares a cost as $\overline{c_j} > c_j$, e_j is still not in $LCPS_k(d|^j \overline{c_j})$. Thus its utility does not change.

Overall, edge e_j maximizes its utility when it reveals its true cost c_j, which means payment scheme (1) satisfies IC. ∎

From Lemma 3 and 4, we have the following theorem.

Theorem 5: Payment scheme 1 is strategyproof.

D. Optimality of our payment scheme

We proved that our payment scheme is truthful in subsection III-C. In this subsection, we will prove that it is optimal, i.e., for any strategyproof mechanism $\mathcal{P}$ based on output LCPS$_k$, the payment to any link calculated by $\mathcal{P}$ is greater than or equal to the payment calculated by Algorithm 1. In other words, we cannot find a strategyproof payment scheme that pays less than our payment scheme. Before we prove this, we prove the following lemma regarding all truthful payment schemes based on LCPS$_k$.

Lemma 6: For any strategyproof mechanism $\tilde{p}$ whose output is LCPS$_k$, for every link e_j, if $e_j \in LCPS_k(d)$ then the payment to edge e_j $\tilde{p}_j(d)$ should be independent of d_k.

Proof: We prove it by contradiction by assuming that there exists a strategyproof payment scheme $\tilde{p}$ such that $\tilde{p}_j(d)$ depends on d_j when $e_j \in LCPS_k(d)$. There must exist two different valid declared costs $a_1 \neq a_2$ such that $\tilde{p}_j(d|^j a_1) \neq \tilde{p}_j(d|^j a_2)$, $e_j \in LCPS_k(d|^j a_1)$ and $e_j \in LCPS_k(d|^j a_2)$. Without loss of generality we assume that $\tilde{p}_k(d|^k a_1) > \tilde{p}_j(c|^j a_2)$. Now consider edge e_j with actual cost $c_j = a_2$. Obviously, it can lie its cost as a_2 to increase his utility, which violates the incentive compatibility (IC) property. This finishes the proof. ∎

Now we show that our mechanism is optimal among all strategyproof mechanism using LCPS_k as its output.

Theorem 7: Among any strategyproof mechanism using LCPS_k as the output, our mechanism is optimal.

Proof: We prove it by contradiction. Assume that there is another truthful mechanism $\mathcal{M} = (LCPS_k, \mathcal{P})$, whose payment is smaller than our payment for a link e_j on a graph $G = (V, E)$ with cost vector c. Assume that the payment calculated by $\mathcal{P}$ for link e_j is $\mathcal{P}_j(c) = p_j(c) - \delta$, where $p_j(c)$ is the payment calculated by Algorithm 1 and $\delta > 0$.

Now consider the same graph with a different cost $c' = c|^j d_j$, where $d_j = p_j(c') - \frac{\delta}{2}$. Since $p_j(c') = p_j(c)$, from Lemma 2 we have $e_j \in LCPS_k(c')$. Applying Lemma 6, we know that the payment for link e_j using payment scheme $\mathcal{P}$ is $p_j(c) - \delta$, which is independent of the edge e_j's declared cost. Notice that $d_j = p_j(c) - \frac{\delta}{2} > p_j(c) - \delta$. Thus, edge e_j has a negative utility under payment scheme $\mathcal{P}$ for graph $G = (V, E)$ under cost profile c', which violates the Individual Rationality (IR) property. This finishes the proof. ■

IV. Experimental Study

From Lemma 1, we know the payment to any link is greater than or equal to its actually cost. Thus, the total payment is often larger than the actual cost of the k-anycast tree LCPS_k. Let $c(LCPS_k)$ be its cost and $p(LCPS_k)$ be the total payment by Algorithm 1. We define the overpayment ratio as $OR(LCPS_k) = \frac{p(LCPS_k)}{c(LCPS_k)}$.

No doubt, we don't want to overpay too much to guarantee the truthfulness. But unfortunately, Archer and Tardos have shown a simple example in [21] such that the overpayment ratio could be as large as $\Theta(n)$ for unicast problem. By a simple modification of their example, the overpayment ratio for k-anycast could also be as large as $\Theta(n)$.

We conducted extensive simulations to study the overpayment ratio of LCPS_k structure proposed in this paper. Notice that, we need guarantee that the network is bi-connected to prevent the possible monopoly of some links. Given a random graph of n vertices, it is known that the graph is bi-connected only when its number of neighbors is in the order of $O(\log n)$. In our experiment, we randomly generate n terminals, every terminals' number of neighbors are drawn from a uniform distribution from $[\log(n), 5\log(n)]$. The weight of edge is uniformly and randomly selected from $[20, 100]$.

In our first experiment, we vary the number of terminals in this region from 100 to 490, and fix the number of sender to 1 and receivers to 30. For a specific number of k, we generate 500 different networks, and study the performance of structure LCPS_k according to two metrics: average overpayment ratio (AOR) and maximum payment ratio (MOR). Left and Middle figures of Figure 2 illustrate the maximum overpayment ratio and the average overpayment ratio for three different values: $k = 1$, 10 and 30. When $k = 1$, it is just anycast, and for $k = 30$ it becomes multicast. We also vary the number k from 1 to 30, and fix the number of sender to 1 and receivers to 30. Right figure of Figure 2, we show MOR and AOR when fix the number of terminals as 200 an 400 respectively.

In our simulations, we found that the overpayment ratio has a trend of decreasing when the number of network nodes increase, and it becomes almost steady when the number of network nodes reach some threshold.

When we vary both the k from 1 to 30 and number of nodes from 100 to 400, we summarize our results in Table I and II. It is easy to notice that when fixes n, both MOR and AOR decrease when K increases; when fixes K and decreases n, both MOR and AOR first decrease then become steady. Another important observation is that when n is greater than some value, say 100, both MOR and AOR won't be too large.

V. Conclusion

In this paper, we defined a new routing called k-anycast, which has potential applications in several areas such as peer-to-peer computing. We then studied how to perform k-anycast in selfish and rational networks, in which every node or link will provide services to others only when it receives a payment to compensate its cost, and it will try to maximize its own profit. In this paper, by assuming that each link in the network has a private cost of providing services to other nodes, we design a k-anycast routing protocol such that every node will follow this protocol and will maximize its profit when it reports its cost truthfully. Notice that, without modification, our protocol also works in the scenario when each network node has a private cost of providing services to other nodes.

A possible future work is to design a routing structure that approximates the minimum cost k-anycast tree, and then design a truthful payment scheme based on that structure.

References

[1] L. Buttyan and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 5, no. 8, October 2003.

[2] Markus Jakobsson, Jean-Pierre Hubaux, and Levente Buttyan, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks," in *Proceedings of Financial Cryptography*, 2003.

[3] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of MobiCom*, 2000.

[4] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, and J. Y. Le Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Communications Magazine*, vol. 39, no. 6, June 2001.

[5] L. Buttyan and J. P. Hubaux, "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] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Energy efficiency of ad hoc wireless networks with selfish users," in *European Wireless Conference 2002 (EW2002)*, 2002.

[7] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc wireless networks," in *IEEE Infocom*, 2003.

[8] Noam Nisan, "Algorithms for selfish agents," *Lecture Notes in Computer Science*, vol. 1563, pp. 1–15, 1999.

[9] Yannis A. Korilis, Theodora A. Varvarigou, and Sudhir R. Ahuja, "Incentive-compatible pricing strategies in noncooperative networks," in *INFOCOM (2)*, 1998, pp. 439–446.

[10] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based mechanism for lowest-cost routing," in *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing.*, 2002, pp. 173–182.

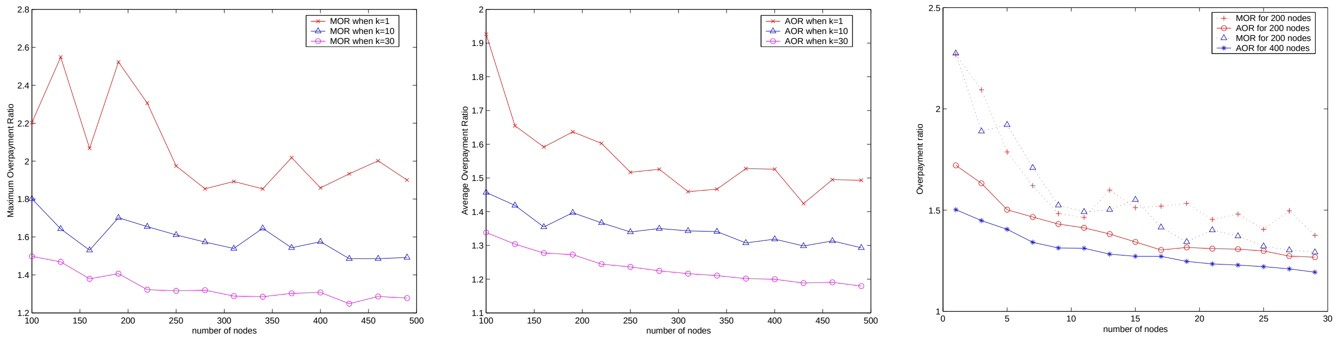
[11] Luzi Anderegg and Stephan Eidenbenz, "Ad hoc-vcg: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. 2003, pp. 245–259, ACM Press.

[12] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders," *Journal of Finance*, pp. 8–37, 1961.

[13] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, pp. 17–33, 1971.

[14] T. Groves, "Incentives in teams," *Econometrica*, pp. 617–631, 1973.

[15] Weizhao Wang and Xiang-Yang Li, "Truthful multicast in selfish and rational wireless ad hoc networks," 2004, Submitted for publication.



MOR for $k = 1{,}10$ and 30 | AOR for $k = 1{,}10$ and 30 | MOR and AOR for 200 and 400 nodes

Fig. 2. Maximum overpayment ratio and Average Overpayment Ratio.

TABLE I
MAX OVERPAYMENT RATIO.

Nodes / Receivers	100	130	160	190	220	250	280	310	340	370	400	430	460	490
3	1.962	1.963	1.630	1.782	1.742	1.821	1.824	1.944	1.738	1.811	1.828	1.817	1.825	1.729
6	1.799	1.659	1.899	1.721	1.738	1.673	1.703	1.834	1.695	1.905	1.626	1.617	1.721	1.771
9	1.718	1.747	1.821	1.756	1.647	1.629	1.678	1.632	1.725	1.630	1.625	1.725	1.528	1.654
12	1.657	1.542	1.568	1.789	1.552	1.537	1.536	1.490	1.496	1.489	1.552	1.491	1.576	1.443
15	1.598	1.584	1.580	1.532	1.505	1.513	1.444	1.476	1.463	1.447	1.439	1.453	1.424	1.428
18	1.637	1.581	1.565	1.475	1.467	1.459	1.456	1.453	1.407	1.414	1.416	1.476	1.399	1.389
21	1.616	1.472	1.593	1.431	1.444	1.508	1.413	1.399	1.375	1.415	1.387	1.396	1.355	1.374
24	1.655	1.504	1.460	1.400	1.442	1.424	1.383	1.372	1.378	1.333	1.357	1.323	1.357	1.318
27	1.496	1.455	1.419	1.394	1.404	1.354	1.383	1.388	1.314	1.331	1.320	1.302	1.300	1.313
30	1.497	1.491	1.437	1.409	1.378	1.318	1.332	1.307	1.306	1.272	1.325	1.277	1.315	1.263

TABLE II
AVERAGE OVERPAYMENT RATIO.

Nodes / Receivers	100	130	160	190	220	250	280	310	340	370	400	430	460	490
3	1.652	1.601	1.433	1.467	1.422	1.479	1.457	1.475	1.458	1.437	1.423	1.363	1.411	1.402
6	1.516	1.448	1.560	1.437	1.446	1.405	1.398	1.407	1.368	1.366	1.342	1.323	1.347	1.340
9	1.480	1.422	1.443	1.401	1.378	1.349	1.379	1.344	1.336	1.331	1.322	1.283	1.310	1.313
12	1.454	1.399	1.379	1.379	1.326	1.339	1.315	1.313	1.299	1.293	1.293	1.278	1.274	1.263
15	1.441	1.371	1.338	1.335	1.326	1.311	1.290	1.289	1.275	1.271	1.258	1.273	1.262	1.246
18	1.436	1.387	1.331	1.331	1.303	1.293	1.279	1.260	1.256	1.245	1.246	1.265	1.244	1.232
21	1.424	1.362	1.349	1.303	1.282	1.274	1.267	1.257	1.245	1.240	1.243	1.262	1.219	1.218
24	1.385	1.345	1.318	1.294	1.284	1.268	1.253	1.241	1.235	1.220	1.215	1.193	1.212	1.205
27	1.356	1.322	1.298	1.272	1.265	1.243	1.233	1.231	1.221	1.221	1.210	1.201	1.191	1.194
30	1.358	1.312	1.283	1.266	1.250	1.235	1.229	1.214	1.211	1.203	1.199	1.194	1.188	1.186

[16] Noam Nisan and Amir Ronen, "Algorithmic mechanism design," in *Proc. 31st Annual Symposium on Theory of Computing (STOC99)*, 1999, pp. 129–140.

[17] Weizhao Wang and Xiang-Yang Li, "Truthful low-cost unicast in selfish wireless networks," in *4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks of IPDPS*, 2004.

[18] Shuchi Chawla, David Kitchin, Uday Rajan, R. Ravi, and Amitabh Sinha, "Profit maximizing mechanisms for the extended multicasting games," Tech. Rep. CMU-CS-02-164, Carnegie Mellon University, July 2002.

[19] Xiang-Yang Li and WeiZhao Wang, "Efficient strategyproof multicast in selfish networks," in *Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, Florida.*, 2004.

[20] J. Green and J. J. Laffont, "Characterization of satisfactory mechanisms for the revelation of preferences for public goods," *Econometrica*, pp. 427–438, 1977.

[21] A. Archer and l. Tardos, "Frugal path mechanisms," in *SODA*, 2002, pp. 991–998.