RPP: A Distributed Routing Mechanism for Strategic Wireless Ad hoc Networks

Min-You Wu
Department of CSE
Shanghai Jiao Tong University
Shanghai, China

Wei Shu
Department of ECE
The University of New Mexico
Albuquerque, NM 87131, USA

***Abstract*—[1] RPP is a distributed routing mechanism for wireless ad hoc networks with *strategic* agents. A strategic agent is rational but selfish and has its own incentive to route traffic for other agents. Forwarding data is definitely not the self-interest of an agent due to consumption of battery. However, if no agent is willing to forward data for others, the ad hoc network itself will break. In this paper, a routing mechanism is designed such that maximizing the benefit of each strategic agent leads to a global optimal system. In this mechanism, an agent accepts payments for forwarding data for other agents if the payment covers their costs. The payment is computed recursively to obtain a cost-efficient and truthful mechanism. The *overpayment* in current mechanisms is completely removed. Analysis and simulation results are also provided.**

I. Introduction

A mobile ad hoc network consists of many nodes, each of which can be viewed as an agent. In this paper, node and agent will be used alternatively. A data message is routed from the source to the destination via intermediate nodes. Most current routing protocols assume that all the agents that make up the ad hoc network are *obedient*. That is, these agents follow the same routing algorithm, such as ODSA [1]. Though this assumption is valid in some environments, it is certainly not in a general ad hoc network. In an ad hoc network, an agent is itself an entity and forwarding data for other agents will drain its battery [2]. Without forwarding data for others, an agent could operate a longer time. Thus, forwarding data is definitely not the self-interest of the agent. However, if no agent is willing to forward data for others, the ad hoc network itself will break. How to create incentives of forwarding data for other agents is an open problem. Therefore, a mechanism needs to be designed to provide agents incentives for forwarding data so that maximizing the benefit of every selfish agent will lead to a global optimal system.

[1]Min-You Wu is currently on leave from The University of New Mexico, USA

In this paper, we introduce a new mechanism, the Recursive Payment Protocol (RPP) mechanism. This truthful mechanism is able to compute a proper payment to each intermediate node. The mechanism is conducted by a distributed algorithm, therefore, is scalable to a large ad hoc network. With this mechanism, we solved the overpayment problem. This mechanism is incentive-compatible and cost efficient.

II. Mechanism Design and Its Application to Ad hoc Networks

In this section, the mechanism design approach is introduced and applied to the ad hoc network routing problem. The mechanism design approach can be briefly described as follows [3], [4]. Each agent has a type t_i known by agent i only. In fact, type t_i is the marginal cost c_i. What agent i reveals is its strategy a_i. In the system, there is a set of possible outcomes $O(a_1,...,a_N)$, indicating which agents participate in the service activity. Each agent has a utility function $u : O \rightarrow R$, where $u_i \in U$ that expresses its preferences over these outcomes. Agent i chooses a strategy a_i to maximize its utility u_i. The utility function of an agent is known by itself. The desired systemwide goal is specified by a *social choice function* $F : U^N \rightarrow O$ that maps each particular instantiation of agents, completely described by their utility functions, into a particular outcome. A social choice function is *strategyproof* if $u_i(F(u)) \geq u_i(F(u|_iz))$, for all i and all $z \in U$. If the agent reveals its real cost t_i in a_i, the utilization will be maximized. In other words, with a strategyproof mechanism, each agent has a *dominate strategy*, which is to reveal the real cost. If F is strategyproof, then no agent has incentive to lie about its real cost, and the desired social goal can be achieved. A goal of the mechanism design is to find a social function that is strategyproof. A mechanism is to generate an outcome set O including a set of *payments* $P = (p_1.,...,p_N)$. Valuation v_i of agent i can be computed as $v_i = v_i(t_i, O)$. According to its valuation and payment function, each agent computes its utility function. An agent does not have an incentive to provide service if its utility is less than zero. There is a class of strategyproof mechanisms, called the VCG mechanism [5], [6], [7]. It utilizes the *second best sealed bid (SBSD)* auctions to compute the payment: $\rho_s\{a_i = \infty\} - \rho_s\{a_i = 0\}$. With the SBSD auctions, the mechanism is truth-telling so each agent will reveal its real type based on its own profit. Thus, the aggregation of decisions of every selfish agent will result in a global optimal solution.

An ad hoc network can be represented by a directed graph G, consisting of a set of nodes $\{n_0, n_1, n_2, ..., n_{N-1}\}$. Each node have two types of energy dissipation, E_i^R/bit and $E_{i,j}^T$/bit. E_i^R/bit is the energy dissipation of receiving and processing a bit at node n_i. Here, the energy required to receive a bit does not depend on where the bit comes from. $E_{i,j}^T$/bit is the energy dissipation of transmitting a bit from n_i to n_j, which depends on how far away the destination n_j is. Assume $d(i,j)$ is the distance between nodes n_i and n_j, and α is the path loss index:

$$E_{i,j}^T/\text{bit} = \upsilon_i + \omega_i \cdot d(i,j)^\alpha,$$

where υ_i and ω_i are parameters. Typical values of these parameters are: E_i^R is around 100 to 200nJ/bit; υ_i is about 45 nJ/bit; and $\omega_i = 10pJ/bit/m^2(\alpha = 2)$ or $\omega_i = 0.001pJ/bit/m^4(\alpha = 4)$ [8], [9]. Note that $E_{i,j}^T$ and $E_{j,i}^T$ can be asymmetric. We assume that a node n_i is able to choose its transmission power P_i^{trans}. The minimum transmission power from node n_i to node n_j can be computed if the following parameters are known: 1) the power at which a node n_j receives the signal $P_{i,j}^{rec}$; 2) the minimum power at which a node is able to receive the signal $P_j^{rec\_min}$; and 3) the signal strength with which the message was sent off P_i^{trans}:

$$P_{i,j}^{trans\_min} = \frac{P_i^{trans} \times P_{i,j}^{rec}}{P_j^{rec\_min}}.$$

The minimum energy dispation of transmitting a bit can be expressed as

$$E_{i,j}^T = \frac{P_{i,j}^{trans_{min}}}{B_i},$$

where B_i is the bandwidth of node n_i. The cost of transmitting a bit is

$$c_i = C_i * (E_i^R + E_{i,j}^T),$$

where C_i is the cost of energy in node n_i. Thus, as long as node n_i reveals its true cost of energy, transmission power, and bandwidth, node n_j is able to determine the unit cost.

Different cost functions could be defined. Here, we define the cost of energy as $C_i = EF_i/ER_i$, where EF and ER are the full energy and the residual energy of node i, respectively. That is, the lower the residual energy, the higher the cost.

Before proposing our mechanism, we now discuss a recent work on mechanism design of ad hoc networks [10], where a payment is calculated for each agent that forwards data for others. There are two models, the central-bank model and the source model. In the central-bank model, the source pays only the true cost for forwarding its data and the central bank pays the bonus. The central bank collects money from all nodes just like collecting tax. Therefore, the voluntary participation rule [3] is violated since the nodes that do not participate are also charged. In the source model, the source pays all the payment including the true cost and a bonus. A bonus will be paid to each intermediate node. Thus, the payment from the source might be higher than the *cost of the second lowest cost path.* This overpayment problem causes the mechanism to be incentive-incompatible since the source will pay more than a reasonable amount defined by the SBSD auction. The approach is not distributed nor scalable since the entire topology information must be sent to the destination node. Also in [10], a *packet purse* model with auctions [11] has been used to create a distributed mechanism that conducts SBSD auctions among their neighbors to find a cheap path to the destination. However, it has been found that it does not compute the most cost-efficient path and is not truthful.

III. The Recursive Payment Protocol

The *Recursive Payment Protocol (RPP)* provides a distributed mechanism to compute the payment to each agent. It is of a source model and the source will pay all of the payment. The overpayment is eliminated completely in this protocol. That is, only the cost of the second lowest cost path is paid by the source. RPP consists of three phases:

- Lowest Cost Path (LCP) discovery;
- payment computing; and
- data transmission and payment delivery.

In the first phase, the LCP will be discovered. Each agent will reveal its real cost and forward the costs of other agents due to a variation of the VCG mechanism. Thus, the destination node is able to obtain the LCP. The most crucial part of the protocol is in the second phase where the payment is computed recursively. The data is actually transmitted in the third phase through the LCP. If the data is successfully transmitted, the payment will be delivered to the agents.

1) LCP discovery

In this phase, the LCP is computed. This procedure mainly follows [12], [10]. The minimum transmission power can be determined at node j if node i correctly sends the value of the current transmission power. With this mechanism, node n_i

will also reveal its real C_i and B_i values so node n_j is able to compute c_i. Then, node n_j will forward the cost to the next node. Thus, the destination node will be able to compute the LCP.

2) Payment computing

In this phase, the payment to each agent on the LCP will be computed. A recursive algorithm is used to compute the payment one by one, starting from the destination node. Assume the LCP is $(n_s, n_{g_1}, n_{g_2}, ..., n_{g_l})$, where n_s is the source node and n_{g_l} is the destination node. First, the destination node computes the payment to its previous node $n_{g_{l-1}}$. Different from a traditional VCG mechanism, the payment (per bit) to node $n_{g_{l-1}}$ is:

$$Q_{g_{l-1}} = c(SLCP(n_s, n_{g_l})) - c(n_s, n_{g_1}),$$

where $c(SLCP(n_s, n_{g_l}))$ is cost of the *Second Lowest Cost Path (SLCP)* excluding all intermediate nodes in the LCP, $n_{g_1}, n_{g_2}, ..., n_{g_{l-1}}$; and $c(n_s, n_{g_1})$ is the cost of n_s to n_{g_1}. The payment is only made to the intermediate nodes and does not include that for the source node so the cost of the source is deducted. The value of $Q_{g_{l-1}}$ is sent to node $n_{g_{l-1}}$, which then computes a payment to node $n_{g_{l-2}}$, and so on. Node n_{g_k} computes the payment to its previous node $n_{g_{k-1}}$ as follows:

$$\begin{aligned} Q_{g_{k-1}} = MIN(&Q_{g_k} - c(n_{g_k}, n_{g_{k+1}}), \\ &c(SLCP(n_s, n_{g_k})) - c(n_s, n_{g_1})), \end{aligned}$$

where Q_{g_k} is the payment n_{g_k} received from node $n_{g_{k+1}}$; $c(n_{g_k}, n_{g_{k+1}})$ is the cost of n_{g_k} to $n_{g_{k+1}}$; and $c(SLCP(n_s, n_{g_k}))$ is the cost of SLCP excluding all intermediate nodes in the LCP, $n_{g_1}, n_{g_2}, ..., n_{g_{k-1}}$. The difference between Q_{g_k} and $Q_{g_{k-1}}$ is the net payment to node n_{g_k}. In this way, the total cost paid by the source is $c(SLCP(n_s, n_{g_l})) - c(n_s, n_{g_1})$, whereas in the current mechanism design multiple SLCP costs might be required.

Each time when a node computes the payment, it needs the cost of SLCP. A distributed algorithm computes all SLCPs at a time. This algorithm is similar to the LCP finding algorithm but any node on the LCP is not sending out messages. In this way, each node on the LCP obtains the SLCP excluding all intermediate nodes on the LCP.

The nodes that participate in the SLCP computing know that they will not be paid as nodes on the LCP. These nodes may not be cooperative in the SLCP computing. In order to ensure participation of these nodes when computing the SLCPs, a sufficient payment is made to each node that participates in the SLCP computing. We assume that the payment to the SLCP computing is relatively smaller than that pays to nodes in the LCP to ensure the mechanism is incentive-compatible to the payer.

3) Data transmission and payment delivery

In this phase, the data is transmitted along the LCP. All nodes use the minimum power required to forward the data. Once the data has been transmitted, the payment will be actually delivered to the agents. The source node may deliver the payment to each intermediate node directly, or it may deliver the total payment to the destination node and the payment is then distributed to each intermediate node. □

When node n_j receives the value of C, B, and P^{trans} from node n_i, it is able to compute the cost c_i. Here we assume that E_i^R is a constant; otherwise node n_i needs to reveal this value too.

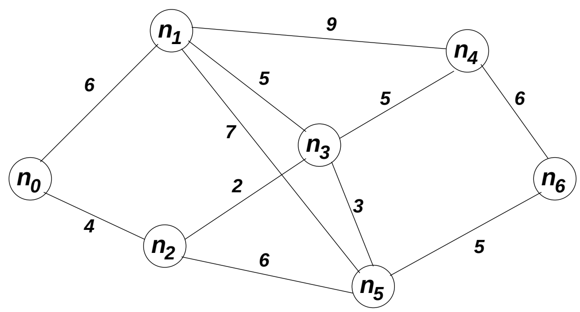
Theorem 1: In RPP, each agent reveals its true C, B, and P^{trans}, and forwards these values from other agents to its neighbors.

Proof: If a higher value of C, $1/B$, or P^{trans} is claimed or forwarded, the cost of forwarding data appears more expensive than it is in reality. If the agent is not on the LCP when claiming and forwarding the true costs, this action will not move it there. If the agent is on the LCP when claiming and forwarding the true costs, it may either no longer be on it by claiming or forwarding a higher value, so the revenue is lost; or it may still be on the LCP but receive the same payment. The SLCP does not change by claiming a higher value either so the payment to the next agent does not change. Thus, the net payment stays the same.

Alternatively, if a lower value of C, $1/B$, or P^{trans} is claimed or forwarded, the cost of forwarding data appears less expensive. If the agent is on the LCP, then it is still on it and obtains the same payment, and the payment to the next agent does not change. If the agent is not on the LCP when claiming and forwarding the true costs, this action would move it there, but the payment received may not be high enough to cover its true cost, or may not have a sufficient fund to pay its neighbor on the LCP. If no sufficient payment is provided for its neighbor, the neighbor will not forward the data and no payment will be made. Thus, truth-telling is a dominant strategy for this mechanism. □

An example of payment computation is shown in Figure 1. In this network, node n_0 is the source and node n_6 is the destination. The cost from n_i to n_j is shown on the edge between n_i and n_j. The LCP from n_0 to n_6 is LCP = $\{n_0, n_2, n_3, n_5, n_6\}$ with $c(LCP)$ = 4+2+3+5 = 14. The SLCP without nodes n_2, n_3, n_5 is $\{n_0, n_1, n_4, n_6\}$ with $c(SLCP)$ =

6+9+6 = 21. For node n_5, the SLCP without nodes n_2, n_3 is $\{n_0, n_1, n_5\}$ with $c(SLCP)$ = 6+7 = 13. For node n_3, the SLCP without nodes n_2 is $\{n_0, n_1, n_3\}$ with $c(SLCP)$ = 6+5 = 11. From the payment computation listed in the figure, the net payment to nodes n_5, n_3, and n_2 is 8, 3, and 6, respectively.



n_{g_k}	c(SLCP)	$c(n_s, n_{g_1})$	c_{g_k}	recv'd	paid	net
n_6	21	4	–	–	17	–
n_5	13	4	5	17	9	8
n_3	11	4	3	9	6	3
n_2	–	4	2	6	–	6

Fig. 1. An example network for payment computation.

IV. Simulation Results

Agents at different locations have various degrees of importance. Some agents must forward more data than others and will receive more payment. The relation of energy consumed and the payment received was shown in [13].

The cost of energy depends on the residual energy of a node. At real time, the residual energy becomes lower when the node transmits data to other nodes so the cost becomes higher. The cost is updated periodically. This update frequency has an impact on the performance. If the update frequency is too high, the overhead will be large. However, a low update frequency will cause stale information. Here, we conduct a simulation to show the realtime performance and its relationship to the cost update frequency. In this simulation, a hundred agents are randomly spread in an area of 1,000 $\times$ 1,000 meters. The communication range is 250 meters. Here, $\upsilon_i + E^R = 150nJ/bit, \omega_i = 10pJ/bit/m^2$, and $\alpha = 2$. The full energy is set to $100J$ and all nodes have the full energy initially. At each time interval, ten pairs of sources and destinations are randomly selected, each transmits 125MB data. When a node has a higher residual energy, a low price can be offered. Our mechanism provides incentives to these nodes so they are willing to forward data for others. On the other hand, a node with low batteries increases its price so it will forward less data. When a node runs out of its energy, it becomes *inactive*. An inactive node cannot be included in any path. This adaptive scheme automatically balances the load and energy consumption, increasing the overall network life.

Shown in Figure 2 is the *life* of the ad hoc network, defined as the number of time intervals without any communication failure due to inactive nodes. It can be seen that when the update frequency is 1, that is, updating the cost every time interval, the life is the longest. The life is about the same when the update frequency is 0.2, updating the cost every five time intervals. When the update frequency is lower than 0.02, the life decreases drastically, due to stale information.

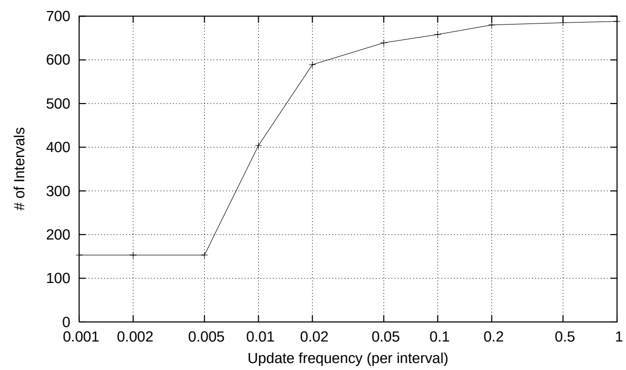


Fig. 2. The life of an ad hoc network.

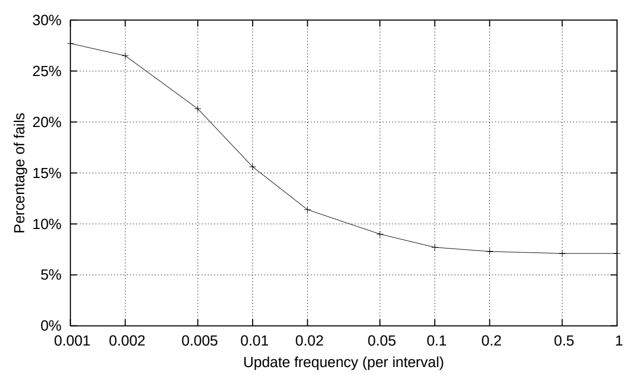
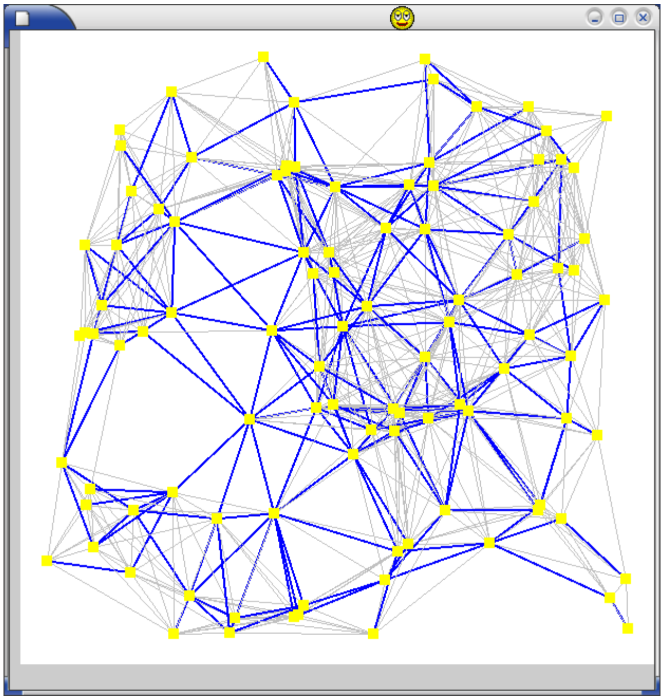


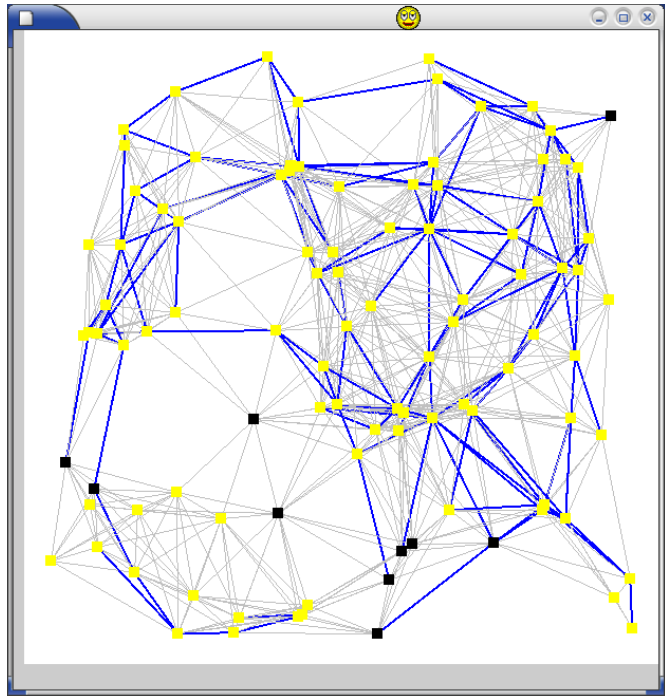
Fig. 3. The percentage of failed communications.

Figure 3 shows the total number of communication failures in 800 time intervals. When the update frequency is higher than 0.05, the percentage of failed communication is less than 10%. The conclusion from these results could be that the update frequency should be between 0.05 and 0.2, that is, updating the cost every ten time intervals.

Figure 4(a) shows the node distribution as well as the initial communication links. The links that have been used to transmit



(a) Snapshot for time interval 1-10



(b) Snapshot for time interval 701-710

Fig. 4. An ad hoc network with 100 nodes.

data in the first ten time intervals are indicated by the bold lines and the others are of gray lines. At this time all nodes are active. Figure 4(b) shows the inactive nodes after 700 time intervals when the update frequency is 0.1. The vertices with black color are the inactive nodes. Here, ten inactive nodes cause about 30% communication failure. The links that have been used to transmit data in the time intervals 701–710 are indicated by the bold lines.

V. Conclusion

The RPP routing protocol presented in this paper is a distributed mechanism, scalable to a large ad hoc network. It provides a cost-efficient routing mechanism for strategic agents and eliminates the overpayment completely. A node with a low battery increases its price so its communication load is reduced, thus, the overall communication load is automatically balanced. This incentive-compatible mechanism ensures proper operations of an ad hoc network.

References

[1] Z. Haas, J. Deng, B. Liang, P. Papadimitrators, and S. Sajama, "Wireless ad hoc networks," 2002. Encyclopedia of Telecommunications, John Proakis, editor.

[2] S. Lindsey, K. Silvalingam, and C. S. Raghavendra, "Power optimization in routing protocols for wireless and mobile networks," in *Handbook of wireless networks and mobile computing, Wiley Series On Parallel And Distributed Computing*, 2002.

[3] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," in *Proceedings of the 6th Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2002.

[4] N. Nisan and A. Rosen, "Algorithmic mechanism design," *Games and Economic Behavior*, vol. 35, pp. 166–196, 2001.

[5] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *Journal of Finance*, vol. 16, pp. 8–37, 1961.

[6] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, pp. 617–663, 1971.

[7] E. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, pp. 17–33, 1971.

[8] T. Rappaport, *Wireless Communications, 2nd edition*. Prentice Hall, 2002.

[9] C. E. Perkins, *Ad Hoc Networking*. Addison Wesley, 2001.

[10] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom '03)*, pp. 245–259, 2003.

[11] L. Buttyan and J. Hubaux, "Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks," Tech. Rep. DSC/2001, 2001.

[12] S. Doshi, S. Bhandare, and T. Brown, "An on-demand minimum energy routing protocol for a wireless ad hoc network," in *ACM Mobile Computing and Communications Review, vol. 6, no. 3, July*, 2002.

[13] M. Wu and W. Shu, "A distributed strategic routing mechanism for wireless ad hoc networks," in *IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication (MWC)*, May 2004.