PARS: Stimulating Cooperation for Power-Aware Routing in Ad-Hoc Networks

Dong Lu[1]
Dept. of Computer Science
Northwestern University
Evanston, IL 60201, USA
donglu@cs.northwestern.edu

Haitao Wu, Qian Zhang,
Wireless & Networking Group
Microsoft Research Asia (MSRA)
49 ZhiChun RD, Beijing 100080, China
{t-hwu,qianz}@microsoft.com

Wenwu Zhu[2]
Intel Comm. Technology China Lab
Intel China Research Center
2 KeXueYuan South RD, Beijing, China
wenwu.zhu@intel.com

Abstract— **Power-Aware routing has proven to be effective in maximizing the life time of ad-hoc networks. However, all the previous works focused on the power-aware routing algorithms in a cooperative network, where all the nodes honestly report battery usage. However, *selfish* nodes in ad-hoc networks tend to misreport lower residual energy to maximize their own usage in practice. In this paper, we first show that it is a dominant strategy for a selfish node to report lower residual energy in a power-aware ad-hoc network, which diminishes the benefits of power-aware routing algorithms. We then propose a novel *P*ower-*A*ware *R*eputation *S*ystem (PARS), to stimulate cooperation on power-aware routing in ad-hoc networks. Analysis and simulation results show that PARS is effective in supporting power-aware routing in selfish ad-hoc networks.**

Keywords- Power-Aware routing, Reputation system, ad-hoc network

I. INTRODUCTION

Power-Aware routing has been extensively studied since Singh, *et al.* [1] first proposed power-aware routing metrics in mobile ad-hoc networks. In an ad-hoc network, all the nodes relay data for each other and the early death of nodes can result in network partition, causing temporary or even permanent interruptions in data communications. Hence, it is very important to maximize the life time of an ad-hoc network.

Power-Aware routing algorithms aim at maximizing the life time of the ad-hoc network by making the power consumption rate on each node more evenly distributed or minimizing the per packet power consumed. Numerous research works have been done on the power-aware routing algorithms [1~4]. However, all the previous works assume that the accurate information of residual energy of the nodes on the candidate paths is known. This assumption holds for the trusted, cooperative ad-hoc networks, but won't be true in the selfish ad-hoc networks where each selfish node has incentives to report much lower residual energy to avoid forwarding data for other nodes and thus save its own energy.

Throughout the paper, when we talk about *power-aware* ad-hoc network, we refer to an ad-hoc network where power-aware routing algorithms are used; we define *selfish* ad-hoc network as an ad-hoc network that consists of selfish yet rational nodes, who participate into the network to maximize their own utility while trying to save their own resources. In Section III, we show that it is a dominant strategy for a selfish node to report a much lower residual energy in order to save its own energy by avoiding packet forwarding for other nodes, and the benefits of power-aware routing algorithms could diminish significantly when there are selfish nodes in the ad hoc network.

Recent studies show that selfish nodes need incentives for cooperation in ad-hoc networks. Most of the research focused on preventing selfish packet forwarding misbehavior, and many schemes have been proposed to address the issue. Among them, reputation systems for ad-hoc networks have obtained much attention recently. Existing reputation systems such as Watchdog and pathrate [6], CONFIDANT [7] and CORE [8] focus on the prevention of data forwarding misbehaviors. While in a power-aware ad-hoc network, a selfish node can bypass the current reputation systems and save energy simply by misreporting lower residual energy. The selfish misreporting can bring negative impacts to the whole network performance, which is shown is Section III.

To address this problem, we propose and evaluate PARS, a reputation system that is designed to stimulate cooperation for power-aware routing in selfish ad-hoc networks. The idea of PARS is to punish the nodes that misreport lower residual energy by isolating them from the network. To achieve this goal, we design two modules in PARS, a *Detection module* and a *Jury module*. The *Detection module* is used to detect energy misreporting, and the *Jury module* forms a jury with neighboring nodes to judge whether a node is misbehaving and isolate a convicted node collectively.

The rest of the paper is organized as follows. Section II talks about related work and Section III illustrates through simulation that it is a dominant strategy for a selfish node to misreport a low energy. In Section IV, we describe the PARS scheme and evaluate its performance. We conclude our work in Section V.

II. RELATED WORKS AND MOTIVATION

Singh, *et al.* first proposed five power-aware routing metrics for ad-hoc networks [1]. They pointed out that it is best to route packets through nodes that have sufficient remaining energy; also it is energy-conserving to route packets through lightly-loaded nodes because the energy expanded in contention is minimized. They also concluded that power-aware routing (built on top of a power-aware MAC protocol)

[1] This work is finished when he was an intern at MSRA.

[2] This work is finished when he was the manager of WN group at MSRA.

can save overall energy consumption and simultaneously increase battery life at all nodes.

Toh [2] emphasized that the power consumption rate on each node must be evenly distributed and the overall transmission power for each connection must be minimized. Toh then proposed four algorithms and compared against each other. Following works [3-4] on power-aware routing further refined the algorithms. However, all the previous works focused on the power-aware routing in the cooperative ad-hoc networks where each node honestly reports its residual energy to the source. There are many ad-hoc networks that may not be cooperative, for example, laptop users in a big academic conference can form an ad-hoc network temporarily to communicate with the Internet. Selfish nodes in ad-hoc network may try to save power by using various cheating strategies. Incentive mechanisms are needed to support the power-aware routing in typical selfish ad-hoc networks.

There are a lot of researches on the Incentives in ad-hoc networks. The research works can be roughly categorized into Micro-payment (Virtual money) mechanism and Reputation system. Nuglet [13] and Sprite [5] are examples of Micro-payment system. Watchdog and Pathrate [6], CONFIDANT [7] and CORE [8] are examples of reputation system. Micro-payment mechanism has a few intrinsic problems. For example, it requires centralized clearance services or trusted hardware. Also, there could be cases where a node can't earn enough virtual money because it is located at the edge of the network. Therefore, we focus on the reputation systems.

Reputation systems work by detecting and punishing the misbehaviors, so that cooperation is more attractive than cheating. Among them, Watchdog and pathrate is the first reputation system for ad-hoc networks. Watchdog is used to detect the misbehaving nodes; and pathrate is to avoid involving those nodes in routing, which actually encourages selfishness since there is no punishment for misbehavior. CONFIDANT addresses this issue by isolating the misbehaviors. However, CONFIDANT assumes pre-assigned trust levels to each node, which is difficult to obtain in practice. CORE doesn't assume any trust levels; instead, it only allows the propagation of good reputation, thus avoids the malicious "Denial of Service" attack. However, bad behavior can only be discovered by individual experience, thus the system can be slow in isolating misbehaving nodes.

Therefore, we believe that all the related works on incentives in ad-hoc networks concentrate on packet forwarding behavior, not on routing. For energy efficient routing, how to enforce the selfish nodes to announce true residual energy is critical. To the best of our knowledge, we are the first to enforce cooperation on power-aware routing in the selfish ad-hoc networks.

III. Selfish Impact on Power-Aware routing

Energy misreporting can significantly mislead the power-aware routing algorithms and negatively impact the network performance. In this section, we illustrate via simulations that in the *power-aware* ad-hoc networks, it is a dominant strategy for the selfish nodes to misreport lower energy to extend its own battery lifetime. In game theory terms, dominant strategy [11] means: Regardless of what he expects his opponents to do, this strategy always yields a better payoff than any others.

A. *Energy misreporting: A dominant strategy for selfish nodes to save energy*

Intuitively, it always helps to save energy by misreporting lower energy in a power-aware ad-hoc network. Since energy reporting is a dynamic sequential game, we can't theoretically prove that is a dominant strategy to report lower energy. We therefore conduct extensive simulations using NS2 [9] to verify our intuition. We implemented the Minimum Battery Cost Routing (MBCR) [2] and Min-Max Battery Cost Routing (MMBCR) [2] in the simulator. MBCR and MMBCR are DSR [10] like source routing protocols, but both of them select the path with the minimum cost. They differ in the definition of cost functions. For each of the candidate path, MBCR calculates the summation of the reciprocal of the residual battery energy on each node as the cost for the path. While MMBCR defines its cost function as the maximum value of the reciprocal of the residual battery energy of the nodes on a given path.

Our simulation scenario is similar to that in [3]. A selfish node will randomly report a very small fraction of its residual energy (0.1% ~ 1% in figure 1 and 2) while an unselfish node will honestly report its residual energy. The percentage of the selfish nodes in different simulations varies from 5% to 100%. Each simulation is randomized by different seeds for 20 runs and we present the average value.

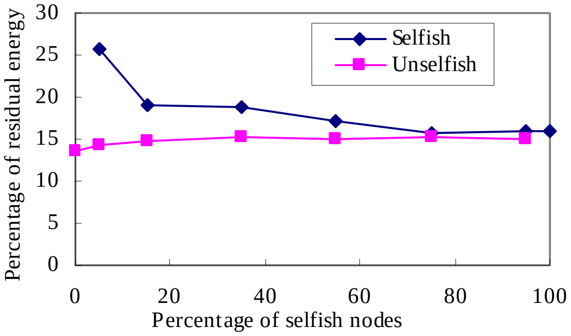

Figure 1. Residual energy in selfish ad-hoc network using MBCR

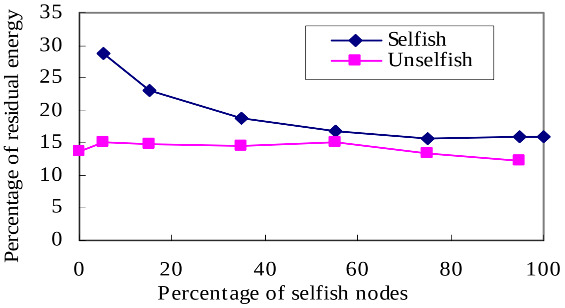

Figure 2. Residual energy in selfish ad-hoc network using MMBCR

Figure 1 and figure 2 show the simulation results of the residual energy of selfish nodes and unselfish nodes as a function of the percentage of the selfish nodes using MBCR and MMBCR routing protocols. It can be seen clearly that:

- Selfish nodes that misreport lower energy levels always have higher residual energy.
- The benefits of energy misreporting decreases with the increase of the percentage of selfish nodes.

We believe the reason is that as more and more nodes start to misreport, the competition among the selfish nodes becomes more severe. As the majority of the nodes start to misreport, both MBCR and MMBCR degrade into a random routing protocol. Since it always helps to save energy by misreporting lower energy no matter how many other nodes have started to misreport, from a game theoretic point of view, it is a ***dominant strategy*** to misreport lower residual energy in the *power-aware* ad-hoc networks.

Through extensive simulations we also find that the benefit of misreporting is closely related to how much lower it misreports. Our simulations show that only reporting significantly lower energy (less than 10% of true energy) can bring significant benefit. This is intuitive because misreporting a much lower residual energy will help to convince MBCR and MMBCR to avoid routes involving the misreporting node(s). Simulation data is not shown here due to space constraints.

B. Negative impacts on the network

Conceivably, misreporting can confuse power-aware routing algorithms and therefore have negative impacts on the network. For example, misreporting can cause the data flow through lower energy nodes and exhaust their energy quickly. The early death of nodes in the ad-hoc network will potentially cause the partition of the network. Also, misreporting can cause the data flow over longer routes and unnecessarily increase the latency and energy consumption.

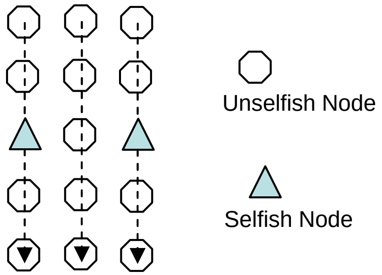

Figure 3. Simulation topology of the static ad-hoc network

In this section, we show the negative impacts on the network through ns2 based simulations. The negative impact is determined by network topology, number of selfish nodes, position of the selfish nodes in the network, the source and destination, nodes movement pattern, etc. To clearly illustrate the effects, we set up a simple static ad-hoc scenario to demonstrate the negative impacts caused by misreporting. We choose a simple scenario due to its simplicity in understanding and explaining. Figure 3 shows the simple topology where there are 15 nodes with vertical and horizontal distance 100m. There are 2 selfish nodes among the 15 nodes. Three source nodes are on the top and three destination nodes are on the bottom. There is no selfish node in DSR and MBCR and MMBCR simulations. We use S-MBCR to denote the selfish ad-hoc network using MBCR protocol. Similarly, S-MMBCR is MMBCR in the selfish network. The selfish nodes in S-MBCR and S-MMBCR simulations misreport a very small fraction of their real residual energy.

Table 1. Simulation results showing negative impacts on the network.

	Mean Delivery Rate (%)	Mean delay (sec)	Mean residual energy of selfish nodes(%)	Mean residual energy of unselfish nodes(%)	Network lifetime (sec)
DSR	86.18	0.0091	N/P	N/P	40.58
MBCR	92.23	0.0088	N/P	N/P	57.74
MMBCR	88.41	0.0092	N/P	N/P	57.74
S-MBCR	79.08	0.011	70.89	27.09	32.39
S-MMBCR	80.38	0.011	67.42	31.19	42.52

Table 1 clearly shows that misreporting can have significant impacts on the network performance. In the simulations, the selfish nodes save power by misleading the power-aware routing algorithms to overload the victim node (the node between the two selfish nodes).

IV. PARS SCHEME

As we've illustrated in the second section, this paper focus on a reputation system for ad-hoc network to stimulate cooperation among the nodes. More specifically, we focus on how to make nodes announce the true residual energy, which is the key for energy efficient routing. As we've shown in the third section, it is the dominant strategy for the node to announce a much lower energy level to avoid being chosen for packet forwarding. So the question is how to stimulate the nodes to announce the true value. Intuitively, there are two approaches to do that: the first is to award those who announce the true value, e.g, better QoS; the second is to punish those who misreport.

For the first approach, a straight forward thought is to provide service differentiation based on reputation and the reported residual energy. The higher energy a node reports and higher reputation, the better service it will receive. This simple scheme will add incentives to report higher energy, but can't detect any misreporting, thus it is a prevention only mechanism. According to Schneier [12], a prevention only strategy only works when the prevention mechanism is perfect. But the simple scheme has the potential that a node may choose to misreport higher or lower energy depending on its needs. In addition, the incentive for nodes to provide QoS for others needs further study.

Therefore, we choose the second approach and propose a Power-Aware Reputation System (PARS). PARS can be built on top of the existing reputation systems such as CONFIDANT [7] or CORE [8], which concentrate on solving the "No forwarding" problem. Actually, PARS is loosely coupled with any of the existing reputation systems, which means that PARS is compatible with any of them. PARS consists of two modules, a *Detection module* and a *Jury module*. Both modules are run on each node. Each node maintains an energy related reputation (E-reputation) and a reputation for the data forwarding behavior. A node will be isolated by all its neighbors if one of the reputations falls below the specified threshold.

A. The Detection module

We assume a DSR-like source routing protocol like MBCR [2] and MMBCR [2]. Each node appends its residual energy in the routing packet in addition to its identity. The *Detection module* is similar to the Watchdog [6] and the Monitor in CONFIDANT [7]. The difference is that the *Detection module* monitors only the energy reported in the routing packets, which implies that the overhead of the monitoring is much lower than that of Watchdog and CONFIDANT, because the number of routing packets was much less than that of the data packets. We designed a set of monitoring rules that all the nodes have to follow.

a) It is not allowed to report abrupt energy change. This is nature because the battery energy is always changing slowly in either charging or discharging mode.

b) At sensing mode, residual energy is always draining at a minimum rate *R*

c) A low energy node should die out in a short time *T* unless it starts to recharge. *T* can be estimated using *T* = Energy/*R*. This rule is based on rule b).

d) Battery recharging is allowed, but not repeatedly within a short time and once starting should reach a relatively high value that can be specified for different systems.

Rules a), b) and c) are combined to detect the nodes that start to misreport a very low energy at any time in operation. If a node starts to misreport a very low energy, he can save energy at the beginning, but can’t send out data soon after that, otherwise he will be detected and isolated. Rule d) is added to detect the nodes that pretend to be recharging and always misreports energy in a very low and small range. For example, it is not allowed to report energy between 0.1% and %1 repeatedly.

One problem with the *Detection module* is that it is not guaranteed that it can detect a misreporting due to the hidden terminal problem. This is similar to that of Watchdog as was pointed out by Marti, *et al.* [6]. We therefore propose the *Jury module* to solve this problem, ensuring that all the neighboring nodes of a misreporting node start to isolate it simultaneously.

B. The Jury module and the jury

All the neighbors of a node consist of a jury for the node. The *Jury module* on each jury member collaborates to judge if a node is misreporting, and if so, reach an agreement and start to isolate the convicted node quickly and simultaneously for a period of time. We put a timeout for the isolation to give the convicted node a second chance to participate into the network. Isolation means to deny any data packets originated from the convicted node. There are two challenges in effectively isolating a misreporting node.

First, once a misreporting is detected, we must be able to isolate the misreporting node quickly and cooperatively. To keep connected to the network, a selfish node has to maintain at least one connection to its neighbors. If all its neighbors start the isolating action simultaneously, then the node is isolated form the network. Second, it is possible that a selfish node may revenge on its neighbors who have detected his misbehavior by lying to other nodes that his neighbors are misreporting energy. To address the two challenges, we introduce the concept of jury under three assumptions:

- A selfish node has incentives to report against any other misreporting behavior, because otherwise it may hurt its own battery life time.
- Selfish nodes will not collude to revenge together on the same node simultaneously.
- Each node has a unique ID that is not forgeable. Our scheme requires it to prevent a selfish node from pretending to be others and revenging on other nodes.

The *Jury module* on each node works this way: Once it detects an energy misreporting from node *M*, it will start to isolate the node and send a Sue Message (SM) to all the neighbors of node *M*. The SM is sent using our proposed Trial Protocol to minimize communication overhead and ensures all neighbors of node *M* are notified in a timely manner. The Trial Protocol is described in the next paragraph. This mutual notification is necessary because not all neighbors can detect energy misreporting due to hidden terminal problem as we discussed in the *Detection module* subsection. Any neighboring nodes of *M* that failed to detect this misreporting start to isolate *M* when it receives no less than 2 SM from different nodes. The magical number of 2 SM comes from the consideration that a selfish node may revenge on other nodes by sending SM to other nodes. Each received SM has a timeout associated with it and a SM will be invalided after the timeout. Note that our assumption of unique un-forgeable ID ensures that a single node can’t isolate a good node without collusion.

The purpose of the Trial Protocol is to minimize communication overhead and ensures all neighbors of node *M* are notified in a timely manner. It works as follows: a node first does a limited flooding broadcast of SM against *M*, and then keeps monitoring if there is still data traffic originated from *M*. If there are any such traffic then send SM to the neighboring node that is relaying data for *M*. As DSR like routing protocols can typically discover several routes simultaneously, a jury member can avoid the routes involving the convicted nods. In this way, node *M* will be isolated by all its neighbors effectively.

C. Performance Evaluation

In this section, we briefly evaluate the performance of PARS, including the effectiveness and overhead.

We evaluate the effectiveness of PARS through simulations. We assume that once a selfish node was detected and isolated for a period of time, it will learn the lesson and start to report residual energy honestly. We use the same simple scenario as shown in figure 3.

Table 2. Simulation results showing effectiveness of PARS

	Mean Delivery Rate (%)	Mean delay (sec)	Mean residual energy of selfish nodes(%)	Mean residual energy of unselfish nodes(%)	Network lifetime (sec)
S-MBCR	89.74	0.0092	40.67	47.64	43.23
S-MMBCR	88.70	0.0099	40.88	47.83	43.23

In comparison with the results in Table1, Table 2 shows clearly that PARS effectively enhances the system performance. For MBCR with selfish nodes, the mean delivery rate has been improved by roughly 10 percent, and the network life time also been longer. Note here the residual energy of selfish nodes are lower than that of unselfish node, which is due to the location where the selfish node lies and packet forwarding for others actually introduce energy cost. We believe the system can perform even better in the long term because of the deterrent effects of PARS.

We analyze the overhead of PARS for the *Detection module* and the *Jury module*. The *Detection module* only monitors the routing packets whose number is much smaller than that of the data packet in a typical ad-hoc network; therefore we believe the overhead introduced in the *Detection module* is much smaller than that of Watchdog and CONFIDANT for packet forwarding behavior detection. Since PARS is supposed to work on top of reputation systems such as CONFIDANT or CORE, the extra overhead introduced by *Detection module* in PARS is very small. The overhead of *Jury module* comes only when a misreporting is detected. Since the reputation systems are mainly deterrent against the misreporting, we believe that once a misreporting node is detected and isolated, it will learn that cheating won’t bring any benefits and therefore stop cheating. Also, our Trial Protocol helps to lower the communication overhead to be smaller than a typical DSR routing request. Therefore, we believe the overhead of PARS is small.

Another interesting issue related to the overhead is the tradeoff in the timeout value for the punishment to those nodes that has been isolated by the Jury. The value should be longer enough to punish the misbehavior of nodes and small enough for fault tolerance and make the convicted node to serve others again.

V. Conclusions and Future Work

Power-Aware routing has been extensively studied in the cooperative ad-hoc networks. However, how to stimulate the selfish nodes to announce true residual battery energy is not addressed. Previous work on stimulating cooperation in selfish ad-hoc networks, such as reputations systems CONFIDANT and CORE, only focus on the “no forwarding” problem, while our work focus on the energy announcement of Power-Aware routing. We propose PARS, a Power-Aware Reputation System to stimulate such cooperation, which can be built on top of current reputation systems such as CONFIDANT or CORE. Through analysis and simulation, we show the effectiveness and overhead of PARS. To our best knowledge, we are the first to address this important and interesting problem.

In addition to the effectiveness of PARS, there are some interesting and challenging issues that need further investigation. For example, we regard how to detect collusion among the selfish nodes as our future work.

References

[1] S. Singh, M. Woo, and C. S. Raghavendra, “Power-Aware routing in Mobile Ad Hoc Networks”, in proceedings of MOBICOM 98.

[2] C.-K Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks”, IEEE Communications Magazine, June 2001.

[3] K. Wang, Yinlong Xu, et al, “Power-Aware On-Demand Routing Protocol for MANET”, in Proceedings of the ICDCS Workshop, 2004.

[4] I. Stojmenovic, X. Lin, “Power-Aware Localized Routing in Wireless Networks”, IEEE Transactions on Parallel and Distributed Systems, Vol. 12, no. 11, Nov. 2001.

[5] Sheng Zhong, Jiang Chen, and Yang Richard Yang, “Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks”, in Proceedings of IEEE INFOCOM '03, San Francisco, CA, April 2003.

[6] S. Marti, T. Giuli, K. Lai, and M. Baker. “Mitigating routing misbehavior in mobile ad hoc networks”, in Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 255--265, 2000.

[7] Sonja Buchegger, Jean-Yves Le Boudec, “Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes--Fairness In Dynamic ad-hoc Networks)”, In Proceedings of MOBIHOC’02, June 2002.

[8] Pietro Michiardi, Refik Molva, “Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks", in proceedings of Communication and Multimedia Security 2002 Conference, September 2002.

[9] The network simulator --- ns2, http://www.isi.edu/ nsnam/ns/.

[10] D. Johnson, D. Maltz, Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", IETF Internet draft, 16-Apr-03.

[11] Roger B. Myerson, “Game Theory: Analysis of Conflict”, Harvard University Press, 01 September, 1997. ISBN: 0674341163.

[12] Bruce Schneier, “Secrets and Lies. Digital Security in a Networked World”, John Wiley & Sons, Inc., 1st edition, 2000.

[13] Levente Buttyan, Jean-Pierre Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks”, Mobile Networks and Applications, Volume 8, Issue 5, page 579–592, 2003.