Effects of Cooperation on Wireless Ad Hoc Network Performance

Maurizio D'Arienzo[1(✉)] and Simon Pietro Romano[2]

[1] Dipartimento di Scienze Politiche, Seconda Università di Napoli,
Caserta, Italy
`maudarie@unina.it`
[2] DIETI, Università di Napoli Federico II, Naples, Italy

Abstract. Unfairness in wireless ad hoc networks is often due to the different ways the single nodes decide to participate in relay operations. We propose to introduce a dedicated ad hoc routing protocol based on results of game theory to force the cooperation of less operative nodes and mitigate the effort of overwhelmed ones. Simulations on an experimental testbed show a better distribution of the energy consumption and an increased fairness in the average delivery ratio achieved by nodes, as well as a prolonged network lifetime.

Keywords: Game theory in wireless networks · Cooperation · Energy efficiency

1 Introduction

Ad hoc networks are composed of several wireless nodes with limited power resources usually provided by accumulators. In these networks each node is an end system and a router at the same time. The limited energy is then not only used to deliver one's own packets to the destinations but also to serve other nodes as message relayer [1,2]. Current routing protocols do not implement any mechanism to verify if other nodes are participating in relay operations, so certain nodes have the chance not to cooperate.

This situation can affect the final performance achieved by single nodes as well as that of the entire network. For instance, the presence of malicious nodes in the network can have a negative impact on the final delivery ratio achieved by well behaving nodes compared to that achieved by uncooperative ones. At the opposite, too much cooperation can lead to an unfair energy consumption because certain nodes, usually the inner nodes of a topology, are more involved than others in relay operations and this causes a greater drain of their energy compared to that spent by border nodes.

In the presence of a behavior detection system, well behaving nodes can deal with uncooperative ones, for instance by refusing to relay packets coming from them. In this way, the malicious nodes experience a delivery ratio reduction and are in fact rather pushed towards a more cooperative attitude. We also aim

at minimizing the residual energy variance, that means to improve the fairness among nodes and avoid an irregular shut down of single nodes. To this purpose, overloaded nodes can periodically switch to a defection state to preserve their own energy, hence inducing a better energy balance among the nodes.

Rather than introducing new energy aware features into the routing protocols, we rely on a decentralized algorithm to track down the behavior of network nodes in order to quickly find alternative paths. We demonstrate that the proposed approach allows for a better balance of energy among the nodes. The algorithm takes inspiration from the results of game theory and enhances an existing ad hoc routing protocol.

2   Related Work

Most of the proposals to mitigate the unfair consumption of energy rely on energy aware routing mechanisms [3]. Similarly to our proposal, [4] proposes to control the energy consumption in each discovered route, and then monitor their state in the following manteinance process. Rather than considering an adaptation at MAC or network layer, the work in [5] formulates the fair energy distribution problem with the same objective functions as our proposal: minimizing the residual energy variance at the same time maximizing the minimum residual node energy.

To address the fair energy consumption issue, in [6] a fair cooperative protocol (FAP) is proposed to improve the overall performance of the whole network. Each node calculates a *power reward* to evaluate the power contributed to and by the others. The work done in [7] still relies on cooperation among nodes. However, it makes the assumption that only a subset of nodes belonging to the same group can be interested in a mutual cooperation instead of a full cooperation involving all nodes.

3   Game Theory Applied to Track Nodes Behaviour

Although game theory is a branch of applied mathematics, it witnessed a great success thanks to the application of its results to a wide selection of fields, including social sciences, biology, engineering and economics, as welll as the study of ad hoc networks [8]. Games can be classified according to various properties. Here we are mainly interested in the difference between *cooperative* and *non-cooperative* games as well as the difference between *strategic games* (played once) and *extensive games* (played many times). A well known non-cooperative game is the *prisoner's dilemma*. In its basic form the prisoner's dilemma is played only once and has been applied to many real life situations of conflict, even comprising thorny issues of state diplomacy. A different version of the prisoner's dilemma is played repeatedly rather than just once and is known as iterated prisoner's dilemma (IPD), which turned out to be a cooperative game under certain circumstances [9]. One of the main result of IPD game is that it stimulates

cooperation. We base our algorithm to mitigate the node selfishness on the results of this version of the game.

In an ad hoc network, the number of nodes and links can change during time, so we consider the number of nodes $N(t)$ as a function of time t. We also define a dynamic array $C(t)$ of $N(t)$ elements for each node of the network. The generic element $c_i(t)$ of $C(t)$ assumes the values (UNKNOWN, COOPERATE, DEFECT) meaning that the behavior of node i at time t is respectively unknown, cooperative or non cooperative. At time $t = 0$ all the values are set to UNKNOWN, since at the beginning each node is not aware of the behavior of the other nodes.

Suppose the generic node s of the network needs to send some traffic to the destination d. The first task is to discover an available path, if it exists, to reach the destination. To this purpose, we consider a source based routing protocol capable of discovering a list $A(t)_{(s,d)i} \; \forall i : 0 < i < P$ of P multiple paths. All the nodes in the list $A(t)_{(s,d)i}$ are considered under observation and marked as probably defecting in the array $C(t)$ unless a positive feedback is received before a timeout expires. The sender s starts sending his traffic along all the discovered paths. If the destination node generates D acknowledgement messages containing the list of all the nodes $L_{(s,d)i} \; 0 < i < D$ traversed, as it happens in some source based routing protocols, the sender s is informed about the behavior of intermediate nodes. For each acknowledgement message received, the sender s can make a final update of the array $C(t)$ by setting the matching elements between the list $L_{(s,d)i}$ and list $A(t)_{(s,d)i}$ as *cooperative*. The mismatches are instead set to *defective* Notice that the last update overwrites the previous stored values and represents the most recent information concerning the behavior of a node.

At the same time, intermediate nodes (those not generating or receiving traffic but still involved in a path) can keep trace of other nodes' behaviors. As soon as a packet to be forwarded (and containing the complete routing list) is received, all the nodes on the path *preceding* the current node are marked as cooperative. Similarly, when an acknowledgement packet is received, all the nodes on the path *following* the current node are marked as cooperative. Missing acknowledgements cause instead a defective mark in the list. Given this algorithm, each node is aware of the behavior of other nodes and can react in the most appropriate way.

The algorithm has been implemented in an existing source based routing protocol for ad hoc networks, the AH-CPN (Ad Hoc Cognitive Packet Network) [10]. AH-CPN is designed to support QoS (Quality of Service) and make an intense use of acknowledgement messages independently from the transport protocol in use. We first modified this protocol to support the search of multiple paths, and then included the new algorithm for the identification of non cooperative nodes [11]. Implementation details are omitted due to limited space.

4   Testbed and Experiments

We tested the proposed routing protocol on a simulated testbed in the ns-2 simulator under different working conditions. The testbed is composed of 25 nodes

arranged in a 5×5 grid topology labeled from 0 to 24. At the end of each of the experiments run on the testbed, we measure the delivery ratio dr_i, the average residual energy of all the nodes μ, their variance ν, as well as the energy e_i spent by node i to successfully deliver one single byte to the destination, which is computed as: $e_i = \frac{Ec_i}{(s_i + rl_i)} * \frac{s_i}{r_i}$, being Ec_i the energy consumed by node i, s_i the total number of bytes sent to the destination, rl_i the number of bytes relayed from node i, and r_i the bytes correctly received at destination. e_i has a dimension of $[Joule/bytes]$.

In the first series of experiments we consider the presence of malicious nodes, which clearly have a negative impact on the final performance of cooperative nodes. In the 25 nodes testbed, nodes $1, 3, 6, 8, 12, 16, 18, 21, 23$ decide to either cooperate or defect for the entire duration of the experiment. The remaining nodes are instead always cooperative. We consider six cases: (i) all nodes cooperate; (ii) node 12 does not cooperate; (iii) nodes $6, 12, 18$ do not cooperate; (iv) nodes $6, 8, 12, 16, 18$ defect; (v) nodes $1, 3, 6, 8, 12, 16, 18$ defect; (vi) all nodes $1, 3, 6, 8, 12, 16, 18, 21, 23$ do not cooperate. A total number of 10 CBR (Constant Bit Rate) UDP traffic sessions are generated between each pair of nodes $(0, 24)$, $(24, 0)$, $(4, 20)$, $(20, 4)$, $(6, 9)$, $(8, 5)$, $(13, 10)$, $(11, 14)$, $(16, 19)$, and $(18, 15)$. Each experiment lasts $720\,s$.

By looking at the first histogram in Fig. 1(a), we can observe how the presence of malicious nodes affects the final performance. Also, in lack of a tracing algorithm, the delivery ratio of defective nodes (marked as def) outperforms that achieved by the cooperative ones ($coop$), whatever the number of defective nodes is. At the opposite, the introduction of the tracing algorithm always favors cooperative nodes ($coop_T$), whose delivery ratio is constantly kept higher than that of defective ones (def_T).

We then evaluate the energy e_i spent to successfully deliver a single byte to the destination in the second row of histograms in Fig. 1(b). Again, the behavior tracking algorithm $mPath\text{-}T$ is able to reverse the values achieved with the plain version of the $AH\text{-}CPN$ protocol. While in the basic protocol version the value e_i of cooperative nodes increases at a pace which closely mirrors the increase in the number of defecting nodes, when the tracing algorithm is enabled such a value is kept low as the number of defecting nodes increases. Also, the energy e_i of defecting nodes shows a raising exponential slope. To balance the unfair energy consumption among the nodes we propose to deliberately push overwhelmed nodes in defection mode for short time intervals. During this second series of experiments the percentage of cooperation of the inner nodes $6, 7, 8 - 11, 12, 13 - 16, 17, 18$ changes over time. In the first experiment, all nodes cooperate. In the second experiment nodes, inner nodes cooperate for a total time of $75\,\%$ out of the total experiment duration. In the third and last experiment, the defection intervals are extended, so the percentage time interval of cooperation is of about $60\,\%$ with respect to the overall duration of the experiment. This time, defective nodes do not send traffic during the above mentioned intervals since their action aims at preserving the energy rather than cheating the other nodes.

(a) Avg delivery ratio

(b) Energy per byte

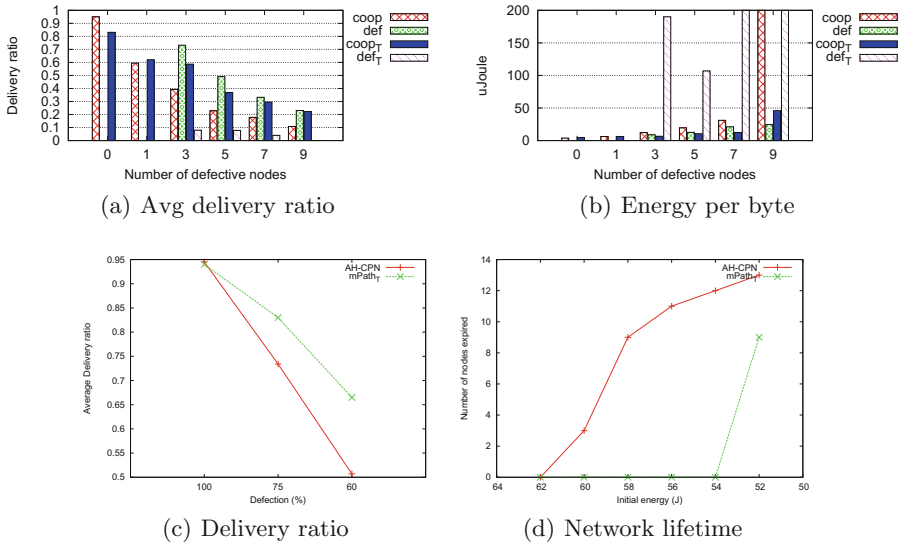(c) Delivery ratio

(d) Network lifetime

Fig. 1. Experimental results

The average residual energy and their variance is significantly reduced when the defection interval of inner nodes increases. The introduction of the proposed tracking algorithm do not show significant improvement with respect to the basic protocol. Notice however that if we look at the comparison of delivery ratio achieved by the two versions of the protocol in Fig. 1(c) (labeled as *basic* (AH-CPN), and *mPath-T* the version with the proposed tracking algorithm), the behavior tracking algorithm clearly outperforms the counterpart when the percentage of cooperation of inner nodes goes down.

We arranged a further experiment to analyze network nodes lifetime. To this purpose we make the testbed start from a situation of low energy and we count the number of nodes that shut down before the natural end of the experiment. As witnessed by the comparative results presented in Fig. 1(d), the introduction of the behavior tracking system again helps reduce the number of node shutdowns as long as the initial energy level is higher than 54 *Joule*. Notice also how the slope of basic protocol is linear, while that of the tracing algorithm looks like an impulse, which again indicates a fairer distribution of residual energy among nodes and then a longer lifetime for the whole network.

5 Conclusions

Current ad hoc routing protocols do not take into account the amount of work done to relay other nodes' traffic, which impacts the residual energy available for a node's own transmissions. This inevitably brings to a situation of unfair energy consumption for certain nodes in favor of other, less involved, ones. In this

paper we propose to introduce some modifications in ad hoc routing protocols to support the identification of nodes' behaviors. Behavioral information gives the busiest nodes a chance to temporarily stop serving the others, while the routing protocol helps discover alternative paths allowing to keep both the residual energy and the overall network performance at fair levels. The improved distribution of energy consumption also prevents the irregular shut down of overloaded nodes, thus increasing the overall network lifetime. Among some possible enhancements we envision the introduction of a self-regulated system to switch the status of each node according to the received behavioral information.

References

1. Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: Energy conservation in wireless sensor networks: a survey. Ad Hoc Netw. **7**(3), 537–568 (2009). ISSN 1570–8705
2. Feng, D., Jiang, C., Lim, G., Cimini Jr., L.J., Feng, G., Li, G.Y.: A survey of energy-efficient wireless communications. IEEE Commun. Surv. Tutorials **15**(1), 167–178 (2013)
3. Mehta, S., Tamanna, Kumar, M.: A comparative study of power aware routing protocols of Ad Hoc network. In: IJCA Proceedings on National Workshop-Cum-Conference on 2011 RTMC (15), pp. 3–7. Foundation of CS, New York, May 2012
4. Asha, T.S., Muniraj, N.J.R.: Energy efficient topology control approach for mobile ad hoc network. Int. J. Comput. Sci. Issues (IJCSI) **10**(4), 289 (2013)
5. Singh, N.K., Simha, R., Narahari, B.: Energy balance in wireless networks using connection segmentation and range control. IEEE Wirel. Commun. Network. **3**, 1871–1876 (2003)
6. Dai, L., Chen, W., Cimini, L.J., Letaief, K.B.: Fairness improves throughput in energy-constrained cooperative ad-hoc networks. IEEE Trans. Wirel. Commun. **8**(7), 3679–3691 (2009)
7. Guha, R.K., Gunter, C.A., Sarkar, S.: Fair coalitions for power-aware routing in wireless networks. IEEE Trans. Mob. Comput. **6**(2), 206–220 (2007)
8. Srinivasan, V., Nuggehalli, P., Chiasserini, C.F., Rao, R.R.: Cooperation in wireless ad hoc networks. In: INFOCOM 2003. vol. 2, pp. 808–817, April 2003
9. Axelrod, R., Dion, D.: The further evolution of cooperation. Science **242**(4884), 1385–1390 (1988)
10. Gelenbe, E., Lent, R.: Power-aware ad hoc cognitive packet networks. Ad Hoc Netw. **2**(3), 205–216 (2004). ISSN: 1570–8705
11. D'Arienzo, M., Oliviero, F., Romano, S.P.: Can cooperation improve energy efficiency in ad hoc wireless networks? Comput. Commun. **35**(14), 1707–1714 (2012). ISSN: 0140–3664