Energy & Throughput Tradeoff in WSN with Network Coding

Nastooh Taheri Javan, Masoud Sabaei and Mehdi Dehghan
Advanced Computer Networks Laboratory, Computer Engineering and Information Technology Department
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
{nastooh, Sabaei, Dehghan}@aut.ac.ir

***Abstract*—Recently, network coding emerged as a promising technology that can provide significant improvements in throughput and energy efficiency of wireless networks. Many implementations of network coding in wireless networks, such as COPE, encourage nodes to overhear to improve the coding opportunities so that they can create better opportunities for coding at the transmitter node through overhearing more packages. In this paper, we have shown that all overheard packets are not necessarily useful for coding; thus, a node can go to sleep mode and shutdown the transceiver to have low power consumption. In this approach, nodes use RL mechanism for duty cycling that is aware of network coding; thereby high throughput and low power consumption can be achieved. In this paper, the duty cycling decision problem is formulated as a Markov decision process. The objective is to maximize the expected total reward of overhearing according to the value of overheard packets.**

Keywords— Wireless Sensor Networks; Network Coding; Coding Opportunity; Coding Gain, Duty Cycling, Reinforcement Learning, MDP.

I. INTRODUCTION

Ahlswede introduced network coding in a pioneering work in 2000[1]. We can broadly define network coding as allowing intermediate nodes in a network to not only forward but also combine their incoming independent information flows.

The first paradigm that illustrated the usefulness of network coding established throughput benefits when multicasting over error-free links. Since then, we have realized that we can get benefits not only in terms of throughput, but also in terms of complexity, scalability, and security. These benefits are possible not only in the case of multicasting, but also for other network traffic configurations, such as multiple unicast sessions. Moreover, they are not restricted to error-free communication networks, but can also be applied to sensor networks [2], peer-to-peer systems [3], and optical networks[4].

The benefit of network coding is illustrated by the following classical example as shown in figure 1. All links are of unit capacity and bits b1 and b2 are to be sent from s to t1 and t2. Without network coding it can be achieved in four time slots while with coding at node 3, i.e. taking XOR of b1 and b2, it is achievable in three time slots.

Wired links are unicast links, but the majority of wireless links (with omni-directional antennas) are broadcast links. Transmissions in a wired network do not interfere with each other, whereas interference is a common case for the wireless medium. Wired nodes are usually static, while wireless was built to support mobility and portability. The wired network design conflicts with the characteristics of the wireless medium. As a result, current wireless networks suffer low throughput, dead spots, and inadequate mobility support. The characteristics of wireless networks might all seem disadvantageous at first sight, but a newer perspective reveals that some of them can be used to our advantage, albeit with a fresh design. The broadcast nature of wireless provides an opportunity to deal with unreliability; when a node broadcasts a packet, it is likely that at least one nearby node receives it, which can then function as the next-hop and forward the packet. This is in stark contrast to the present wireless design, where there is a single designated next-hop, and when it does not receive the packet, the previous hop has to retransmit it[5].

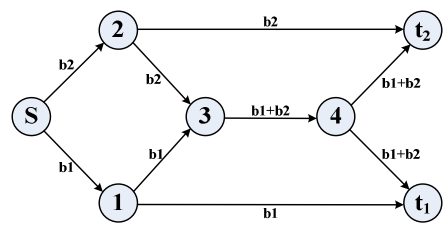

Fig. 1. Example of network coding.

Consider the example in figure 2, where node1 and node2 want to exchange a couple of packets. The radio range does not permit them to communicate directly and thus they need to go through a router. In the current design, node1 sends her packet to the router, which forwards it to node2, and node2 sends his packet to the router, which forwards it to node1. Yet, node1 and node2 could transmit their respective packets to the router, which XORs the two and broadcasts the resulting coded packet. Node1 recovers node2's packet by XOR-ing again with her own, and node2 recovers node1's packet in the same way. The process exploits the existing redundancy in the network to compress the information, delivering two packets in a single transmission, and improving the throughput.

Lot of work has been performed in the field of network coding in wireless network, such as DCAR[6], COPE[7] and so on. In most existing work like COPE, DCAR and etc; network nodes increase coding opportunities with help of overhearing.

For example, in [8], network performance in COPE has been analyzed with and without overhearing, and the authors have shown that coding opportunities have been increased with help of overhearing; consequently, the number of dispatches in the network reduces.

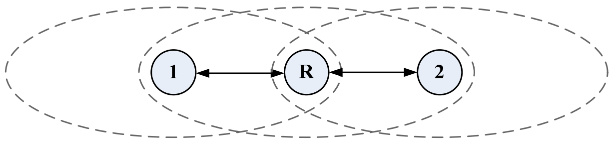

Fig. 2. Coding in wireless networks.

In figure 3, the importance of overhearing in reducing the number of sends in the network which leads to the throughput increase has been shown with a very simple example. In this scenario, there are two streams, one flowing from N1 to N2, and a stream of N2 to N3. In this example, when the router node holds packets a and b. If N3 hasn't overheard packet a from node a packet from node N1, the router has to have two more sending both to N2 and N3 to deliver a and b. But if N3 has overheard the transmitted packet (packet a) which has been sent by N1, the router can send packet a + b to the receiver successfully.

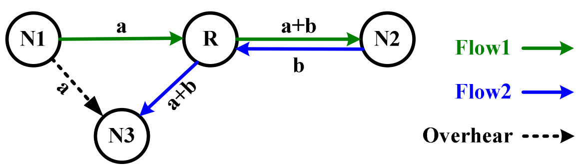

Fig. 3. Effect of overhearing in NC.

In recent years, the theory of network coding in wireless sensor networks has been implemented. In [9, 10, 11] the authors have tried to increase the performance of wireless sensor networks using network coding. This leads to fewer transmissions and thus helps to save transmission and reception energy.

In WSN, wireless channel impairments added with limited available energy demands designing efficient transmission schemes and protocols. On the other hand energy efficiency is a critical issue for the proper functioning of wireless sensor networks and has been the focus of many recent studies.

One of the major limitations on performance and lifetime of such networks is the limited capacity of these finite power sources, which must be manually replaced when they are depleted. Previous works (for example [12] and [13]) show that, in WSNs, energy efficiency can be achieved by periodic duty cycling of sensor nodes, that is, scheduling sensor nodes between active and sleep mode. This could mean switching off the radio transceivers or the sensory devices, or both.

An issue which is addressed in this paper is: *To Overhear or Not to Overhear.*

In fact, on the one hand, to improve the coding efficiency and enhance network coding opportunities, nodes are encouraged to overhear which definitely boosts their energy, and, on the other hand, nodes prefer to turn off transceiver when they have no data to send or receive in order to reduce the energy consumption. In this article, we have tried to explain the problem completely first, then we have formulated using the MDP.

For the main goal of finding the optimal duty cycle, we have decided to employ a reinforcement learning (RL) [14]. RL is a machine learning (ML) approach that finds the optimum value through trial-error iterations.

In this approach, each node independently and according to its received traffic pattern tries to start learning in environment to make the best decision in each slot to sleep or stay awake.

The rest of this paper is organized as follows. The following section deals with the related works. Section 3 describes the preliminary concepts and section 4 describes proposed mechanism, system model and analytical framework in detail. Performance evaluation by simulation is presented in section 5 and concluding remarks are made in section 6.

II. Related Works

In [7], Kitta et.al. Propose COPE, the first practical wireless network coding system. However COPE uses expected transmission count (ETX) as its routing metric and passively waits for coding opportunities, which leads COPE to lose many potential coding opportunities. COPE is a coding architecture which exploits coding opportunity in single path routing in wireless networks. The design and implementation of COPE have demonstrated that network coding techniques can be applied to real wireless networks. However, COPE is mainly designed for single path routing and it adopts a passive approach, essentially just waiting for coding opportunity.

COPE adds a coding layer between IP and MAC layers to detect coding opportunities and exploit them to forward multiple packets in a single transmission. COPE fundamentally incorporates three techniques; opportunistic listening, opportunistic coding, and learning neighbor state. During several intersession (i.e., multiple unicast sessions) flows using opportunistic listening, each node in promiscuous mode over- hears the transmission of its multiple neighboring nodes. The maximum number of original packets transfer within a single transmission, which is the main objective of opportunistic coding. Related to learning a neighbor state, and through the broadcasting of reception reports, a node lets other neighbors become aware of the packets it currently contains. COPE uses the ETX (Expected Transmission Count) metric as its routing function but overall routing is independent of coding opportunistic characteristics of wireless networks.

Authors of [6] propose DCAR, a practical network coding aware routing system. However, DCAR may find sub-optimal path under some scenarios since its metric lacks an import mathematical property, called isotonicity, that requires the weight order of two paths does not change if they are appended or prefixed by a common third path.

Multi-sleep states presented in [15] aimed at introducing the concept of partitioning the sensor node into three main components (Microcontroller, sensor and transceiver units), and presenting the feasible combinations of activity levels for each component in the node. Thus duty cycling shifted from simply switching the node on and off, to controlling the activity level of each of its components. Yet the authors presented the scheme as a power model, without addressing how these power models could be utilized in WSN protocols.

Later, Wang and Liu [16] consider the scenario of providing a network-wide broadcast service and propose to make effective use of local broadcast of wireless medium, achieving a balance between efficiency and latency with coverage guarantees.

An interesting problem for data aggregation in WSNs is the trade-off between lifetime and latency, i.e., when a node waits a longer time for aggregating more data before one transmission, it may prolong its lifetime but at the cost of higher data-delivery latency. The work in [17] has studied this problem, and proposed heuristics for prolonging network lifetime under certain data-delivery latency bounds. A highlight of the heuristics provided in [17] is that each node adjusts its aggregation holding time in an adaptive manner by communicating with its neighboring nodes, hence the lifetime of the nodes in the whole network can be gradually adjusted to a balanced status and the nodal lifetime bottlenecks are reduced.

In [18] authors thoroughly investigated if network coding and duty cycling can be used together for more aggressive energy savings in flood-based wireless sensor networks their main idea is to exploit the redundancy sometimes present in flooding applications that use network coding, and put a node to sleep (i.e., duty cycle) when a redundant transmission takes place (i.e., the node has already received and successfully decoded a sequence of network-coded packets). They proposed a scheme, called DutyCode, in which a multiple access control (MAC) protocol implements packet streaming and allows the network coding-aware application to decide when a node can sleep.

There are a few approaches that have tried to use the RL for duty cycling. DCLA [19] employs a reinforcement learning technique known (Q-learning) to find the optimum duty cycle value through trial-error iterations. RL-MAC (Reinforcement Learning MAC) [20] tackles the problem of optimizing active and sleep periods with the double aim of increasing throughput and saving energy. RL-MAC uses MDP (Markov Decision Process) to model the process of active time reservation. The main drawback of this protocol is that it relies on a constant traffic load over a long period.

III. Preliminary Exploration

As mentioned earlier, most implementations of network coding in wireless networks (e.g., COPE) encourage nodes to overhear in order to increase the chances of coding. In [8] it has been illustrated that the performance of network coding using overhearing is much better than not using it as the number of sending in network decreases to over 40%.

In addition, to reduce energy consumption, duty-cycling is widely used in MAC protocols for wireless sensor networks. With duty cycling, each node alternates between active and sleeping states and leaving its radio powered off most of the time. Thus, each node in network must have a Trade off to overhear or not to overhear.

To clarify the issue, we have conducted a simulation using NS-2. For this purpose, we have implemented the COPE with all the details. The simulated network is deployed in a flat square with 200 meters on each side and there are 100 nodes in this area. The network was modeled with nodes placed randomly and all nodes have the same transmission range of 40 meters. The radio model to transmit and receive packets is RADIO-ACCNOISE which is the standard radio model used. In our simulation, all data packets are 512 Bytes and each simulation time is 300 seconds.

In this scenario, we have randomly established CBR flows between nodes. The duration of each flow is 15 seconds. Therefore, that at any time in the simulations, there is an average of 13 flows in network. For routing the base DSR algorithm is used. In the simulation, nodes keep their transceiver on during the whole simulation time for overhearing (as pointed out in COPE).

Simulation results show that in our scenario, an average of 39% of the packages that have overheard nodes, have done nothing to help network coding opportunities to increase. For example, in figure 4, 28 slots of the life cycle of a node in the network are shown.

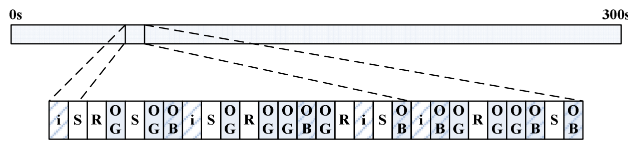

Fig. 4. Result of overhearing

As it can be seen in figure 4, each time slot can have five modes for each node.

1. Node in the mode of sending packet, which is shown as white boxes with the letter S.
2. Node in the mode of receiving packet, which is shown as white boxes with the letter R.
3. Node in mode of overhearing an effective packet in coding, which is shown in gray boxes with OG letters.
4. Node in mode of overhearing an ineffective packet in coding, which is shown in gray hatched boxes with OB letters.
5. Node keeps the transceiver on, even though no packet is received. This is shown in white hatched boxes with letter i.

In this example, the specified node must remain awake in 9 time slot out of 28, because it has data to send or receive in these slots. (Slots S and R) But, according to the idea of COPE, 19 remaining slots are active for overhearing. Moreover, among these 19 slots, only sniffed packets in 9 slots (slots OG) are effective for coding and increase transmitters' coding opportunities. However, in the remaining 10 slots (slots i and OB) the node has not practically sniffed an appropriate package. In this example, 10 slots have not created an opportunity for coding out of 19 overheard slots. In other words, more than 52% of the slots in which nodes have deliberately kept their transceiver on to increase the network coding opportunity, have not been helpful. However, this approximation according to the results of the simulation for total number of nodes is equal to 39%.

The results of this simulation show that nodes can reduce their energy consumption if they can conduct a tradeoff between overhearing and duty Cycling.

In our proposed solution, as it is being described in the next section, nodes start learning in the environment with help of RL in order to make a proper decision to whether keep the transceiver on or off. For this purpose, we have formulated the problem as an MDP.

IV. SYSTEM MODEL AND ANALYTICAL FRAMEWORK

We assume that the nodes are operating based on the idea of COPE. In fact, as it has been implemented in [7], each node makes its neighbors aware of the received packets. The major difference between our work and COPE is in each node's decision making for overhearing its neighbors' sent packets. In COPE and the phase of Opportunistic Listening, each node constantly keeps on its transceiver to overhear the neighbors' packets. But in our idea, each node uses RL to gradually learn that in which time slot it is either more appropriate to keep on its transceiver for overhearing, or it is more convenient to power off its transceiver to reduce the energy consumption.

A. MDP Based Formulation

Markov decision process (MDP) is a widely used mathematical framework for modeling decision-making in situations where the outcomes are partly random and partly under control. A MDP is a discrete time stochastic control process, formally presented by a tuple of four objects (S; A; P_a; R_a).

S is the state space, s ϵ S is the current state, known to users A is the action space, where a ϵ A is the action taken based on the current state. Pa(s) is the probability that action a in state s at time t will lead to state s_0 at time t+1. (i.e. P is a probability transition function S × A × S → [0, 1].) Note that this transition is partly random and partly under control.

Ra is the reward of action a. (i.e. R is the reward function S × A → R.)

Also, we define π as the decision policy that maps the state set to the action set: π: S → A.

Suppose at episode k, the RL agent detects s_k=s ϵ S, the agent chooses an action a_k = a ϵ A(s_k) according policy π in order to interact with its environment. Next, the environment transitions into a new state s_{k+1}=s ϵ S with the probability $P_{ss'}(a)$ and provides the agent with a feedback reward denoted by r_k(s, a).

The goal for the RL agent is to maximize the expected discounted reward or state-value[21], which is represented as:

$$V^{\pi}(s) = E_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_k\big(s_k, \pi(s_k)\big) | s_0 = s\right) \quad (1)$$

This function is evaluation function for policy π. where γ (0 ≤ γ < 1) is the discount factor and $E_\pi\{\}$ denotes the expected return when starting in s and following policy π thereafter. The equation above can be rewritten as

$$V^{\pi}(s) = R\big(s, \pi(s)\big) + \gamma \sum_{s' \epsilon S} P_{ss'}(\pi(s)) V^{\pi}(s') \quad (2)$$

Where R(s, π(s)) = E {r(s,π(s))} is the mean value of the reward r(s,π(s)).

However, in many practical scenarios, as in our case, the transition probability is unknown, which makes it hard to evaluate the policy π. Q-learning [14] is one of the most effective algorithms for learning from delayed reinforcement to determine an optimal policy, in absence of the transition probability.

The main challenge of MDP modeling is to manage its complexity in terms of the number of states, the number of actions, and the time horizon. In our case, the number of actions is limited, and the time horizon is finite. Time is slotted. Each time slot is a time unit. The decision is made at the beginning of the time slot given the information on the current state.

The state of a sensor node is represented by the state vector (n_1, n_2, ... , n_N). N≥1 is the number of node's neighbors. n_i is the number of overheard packets which have been sent from the neighbor i. For example, if in one node the value of n_2 would be equal to 3, it means that there are three overheard packets from neighbor 2 in the buffer. If we assume that B is the buffer size of a node, n_i has to be less than or equal to B, we have also:

$$\sum_{i=1}^{N} n_i \leq B \quad (3)$$

A: $\{a_i\}$ • represents the possible action set, a_1: turn the transceiver on and overhear packets in promiscuous mode, and a_2: turn the transceiver off and go to sleep mode.

Reward function consists of two parts. One part is based on the internal state of a node and the other is based on the value of overheard packets in the senders. The cost of the selected action is determined immediately in each time slot, but the reward would be achieved later.

$$R_t = revenue\ (A^t) - cost\ (A^t) \quad (4)$$

Where *cost* is:

$$cost(A^t) = \begin{cases} 0, & if\ A = a_2 \\ C_i, & if\ A = a_1\ and\ there\ was\ no\ overheared\ packet \\ C_o, & if\ A = a_1\ and\ a\ packet\ was\ overheared \end{cases} \quad (5)$$

Where C_i indicates consumed energy in one time slot for idle listening. This means that the node has been in active mode, but it hasn't overheard any packets. In addition, C_o is the consumed cost for overhearing a packet.

The part of the positive reward of action is based on the value of the overheard packet:

$$revenue(A^t) = \begin{cases} 0, & if\ A = a_2 \\ 0, & if\ A = a_1\ and\ there\ was\ no\ overheared\ packet \\ G_p, & if\ A = a_1\ and\ a\ packet\ was\ overheared \end{cases} \quad (6)$$

Where G_p is the total value of the overheard packet. In fact, if a node at a time slot selects the action a_2 and turns off its transceiver, it certainly cannot overhear any packets; thus, it won't receive any reward. On the other hand, if the node in time slot t chooses action a1 but does not receive any packet, it still will not receive any reward. Finally, if the action a1 is chosen at time t and a packet is overheard, the node will receive a reward according to the value of overheard packet which can be greater than or equal to zero. Moreover, the amount of the reward depends on G_p.

It waits for T time slots for the evaluation of the overheard packet p in order to set the value of G_p node. In fact, the value of the overheard packet p can increase by the passage of time. If the corresponding node receives one or more coded packet-containing packet p- after overhearing the packet p and declaring the packet to its neighbors, this means that packet p was able to increase the coding opportunity in transmitter node; therefore, it causes the reduction of one transmission in the whole network. The recognized reward for a node in this situation is proportionate to the amount of saved energy for a single transmission reduction in network.

The amount of saved energy for each transmission reduction is shown by P_p which contains P_s as the power consumption to send a packet and equal to the number of neighbors of a node. P_p also contains P_r which is the power consumption to receive a packet.

$$G_p = \sum_{t=0}^{T} B_p(t)\,.\,P_p \qquad (7)$$

Where $B_p(t)$ = {0,1} that shows the value of packet p in t time slot. In a way that in each time slot after overhearing packet p and for the next T time slots, if the node receives a coded packet that contains the overheard packet, B_p value for that time slot is equal to one; otherwise, it is equal to zero.

And the value of P_p (saved energy for each transmission reduction in network) equals:

$$P_p=P_s+d.P_r \qquad (8)$$

Where d is the average degree of nodes in the network. (In fact, the average number of neighbors of a transmitter node that receive a packet).

Where P_s is average power need for sending a packet, P_r is average power need for receiving a packet and d is average degree of networks nodes.

Therefore, if the value of G_p is zero for the overheard packet "p", this means that the packet has not participated in any coding situations. In fact, overhearing such packet has nothing to contribute to increase the network coding advantages. However, if the value of G_p is greater than zero for each overheard packet "p", it means that this packet has participated at least once in the corresponding node's neighbors coding situations. The largeness of G_p value indicates the usefulness of the overheard packet "p".

Computing the optimal policy

Our goal is to find the policy (mapping from states to actions) that maximizes the average sum of rewards. We use backward approach to find optimal policy. The proposed algorithm has been depicted in figure 5.

1) Set t = 0 and $V_0(0;P^*) = 0$
2) Substitute t+1 for t, and compute $V_t(K_t^*;P^*)$ by searching $K_t \in H_t$, where $H_t = \{K_{t-1}^*,\ K_{t-1}^*+1, \dots, \widehat{K}_t\}$,

$$i.e., V_t(K_t^*;P^*) = \max_{K_t \in H_t}\left\{R_t(K_t) + \sum_{j=0}^{t-K_t} q(j)V_j(K_j^*;P^*)\right\},$$

$$and\quad K_t^* = \arg\max_{K_t \in H_t}\left\{R_t(K_t) + \sum_{j=0}^{t-K_t} q(j)V_j(K_j^*;P^*)\right\}.$$

3) If $t = T$ then STOP; else go to step 2.

Fig. 5. Optimal policy learning algorithm

V. SIMULATION

To evaluate the performance of proposed algorithm, several simulations were carried out in comparison to the COPE protocol using the NS-2 simulation platform. In this scenario 100 nodes are randomly placed inside an area of 1000 x 1000 m^2. In all scenarios, we have used CBR and Poisson traffic sources with different time intervals. UDP was employed as the transport layer protocol. In this scenario, each node generates Poisson traffic based on statistically distributed inter-arrival times. The average inter-arrival time varies from 0.5 to 10 sec. We assumed the initial energy of all nodes in the network to be 5 joules.

A. Energy Consumption

In figures 6 and 7 the energy comparison between proposed algorithm and COPE has been depicted. As it is obvious, by using our scheme, network energy resources can be used more efficiently. in both algorithms, power efficiency increases (or, alternatively, the per-byte cost of energy decreases) as the traffic load increases, which is reasonable given that more energy is used in transmission and reception rather than idle listening, when traffic load is heavier.

As the figure 6 shows, the average power consumption of the proposed algorithm is less than COPE. Owing to this fact that nodes spend a period of their life in sleeping mode in the proposed method. Therefore, they consume far less energy than active and overhearing mode. On the contrary, all nodes are active in COPE in the whole period of their life to overhear the traffic in their environment.

In figure 7, the number of dead nodes during the simulation time is shown. This form can be a benchmark for comparing the network lifetime of two algorithms. As it is illustrated in the figure, nodes in network are living longer according to the less energy consumption of the proposed algorithm.

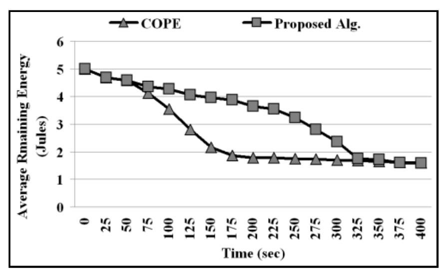

Fig. 6. Remaining energy vs. Time.

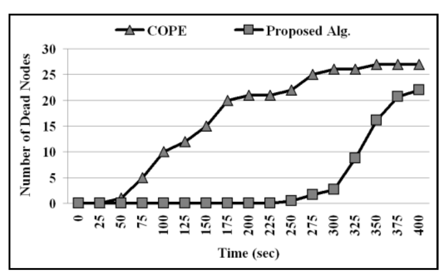

Fig. 7. Number of dead nodes vs. Time.

B. Throughput

Figure 8 shows the network throughput with proposed algorithm and COPE. As it can be seen in the figure, COPE improves the throughput relatively more than the proposed algorithm. This is due to the loss of some coding opportunities in the proposed algorithm. In the proposed algorithm and through the learning range, some useful packets for coding may not be overheard by nodes. Therefore, some coding opportunities that can reduce the number of posts in the network will be lost.

C. *Coding Gain*

In figure 9 coding benefits are shown for both algorithms under different traffic loads. We defined the coding gain as the ratio of the number of transmissions required by the current non-coding approach, to the minimum number of transmissions used by COPE and proposed algorithm to deliver the same set of packets. By definition, this number is greater than or equal to 1. In this scenario We randomly generated UDP flows and varied the offered load by adjusting the number of flows. Each flow has a packet arrival rate ranging from 200 kb/s to 2 Mb/s with a random duration. The source and destination of each flow were randomly chosen.

With focusing on figure 9 it becomes clear that in different conditions the coding gain value in the COPE is better than the proposed method. This is obviously due to the loss of some coding opportunities which is caused by overhearing mechanism in the proposed approach.

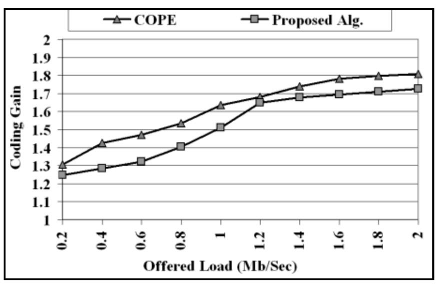

Fig. 8. Coding gain vs. Offered load.

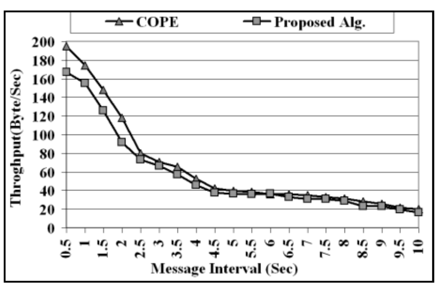

Fig. 9. Throughput vs. Message Interval

I. CONCLUSION

Broadcast nature of wireless networks causes an effective use of network coding theory in these networks. In this paper, we have shown that the non-stop neighbor's traffic overhearing causes unnecessary energy consumption in the network nodes. Since some overheard packets do not effectively contribute to improve coding benefits, while network nodes can switch to sleep mode and turn off their transceiver to save energy instead of overhearing these packets. To solve this problem, using RL for nodes has been proposed and the problem has been formulated as an MDP. In this way, nodes try to make the best decision to maximize the amount of rewards in long-term for staying in active mode and overhearing neighbors' packets or go into sleep mode for energy saving. Simulation results show that the proposed method not only reduces energy consumption of the network, but also increases the network life time where no significant changes (tangible) in the coding opportunities and coding benefit in the network have been occurred.

REFERENCES

[1] R. Ahlswede, N. Cai, S. Y. L. Li, and R. W. Yeung, "Network information flow," IEEE Transactions on Information Theory, vol. 46, no. 4., pp. 1204-1216, 2000.

[2] J. B. Sanson, N. R. Gomes, and R. Machado, "Optimization of Wireless Sensor Network Using Network Coding Algorithm," The Twelfth International Conference on Networks (ICN), pp. 21-24, 2013.

[3] M. Yang and Y. Yang, "Applying Network Coding to Peer-to-Peer File Sharing," IEEE Transactions on Computers, vol. 99, no. 1, pp.1-6, 2013.

[4] Zh. Liu, M. Li, L. Lu, Ch. K. Chan, S. Ch. Liew and L. K. Chen, "Optical Physical-Layer Network Coding," IEEE Photonics Letters, vol. 24, no. 16, pp. 1424-1427, 2012.

[5] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard and H. Rahul, "Wireless Network Coding: Opportunities & Challenges," MILCOM, pp. 1-8, 2007.

[6] J. Le, J. C. S. Lui and D. M. Chiu, "DCAR: Distributed Coding-Aware Routing in Wireless Networks," IEEE Transaction on Mobile Computing, vol. 9, no. 4, pp. 596-607, 2010.

[7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft, "XORs in the air: practical wireless network coding," IEEE/ACM Transactions on Networking, vol. 16, no. 3, pp. 497–510, 2008.

[8] S. Sengupta, Sh. Rayanchu and S. Banerjee, "An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing," IEEE International Conference on Computer Communications (INFOCOM), pp. 1028-1036, 2007.

[9] M. Nagajothy and S. Radha, "Network Lifetime Enhancement in Wireless Sensor Network using Network Coding," International Conference on Control, Automation, Communication and Energy Conservation, pp. 1-4, 2009.

[10] P. M. Glatz, J. Loinig, Ch. Steger and R. Weiss, "A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks," 9th IEEE Malaysia International Conference on Communications, pp. 905-910, 2009.

[11] Y. Wang and Q. Zhang, "Research on Energy Efficiency of Wireless Sensor Networks Based on Network Coding," Information Technology Journal, pp. 1-8, 2012.

[12] D. Feng, Ch. Jiang, G. Lim, L. J. Cimini, G. Feng and G. Y. Li, "A Survey of Energy-Efficient Wireless Communications," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp.167-178, 2013.

[13] R. Maheswar, P. Jayarajan and F. N. Sheriff, "A Survey on Duty Cycling Schemes for Wireless Sensor Networks," International Journal of Computer Networks and Wireless Communications, vol. 3, no. 1, pp. 37-40, 2013.

[14] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," Cambridge, MIT Press, 1998.

[15] A. Sinha and A. Chandrakasan "Dynamic power management inwireless sensor networks," IEEE Design & Test of Computers, vol. 18, no. 2, pp. 62-74, 2001.

[16] F. Wang and J. Liu "On Reliable Broadcast in Low Duty-Cycle Wireless Sensor Networks," IEEE Transaction on Mobile Computing, vol 11, no. 5, pp. 767-779, 2012.

[17] Z. Li, Y. Peng, D. Qiao and W. Zhang, "LBA: Lifetime balanced data aggregation in low duty cycle sensor networks," IEEE INFOCOM, pp. 1844-1859, 2012.

[18] R. Chanadanala, W. Zhang, R. Stoleru and M. Won, "On combining network coding with duty-cycling in flood-based wireless sensor networks," Ad Hoc Networks, vol. 11, no. 2, pp. 490-507, 2013.

[19] R. D. Paz and D. Pesch, "DCLA: A Duty-Cycle Learning Algorithm for IEEE 802.15.4 Beacon-Enabled WSNs," ADHOCNETS, pp. 217–232, 2010.