2016 IEEE First International Conference on Internet-of-Things Design and Implementation

Predictive Power Control in Wireless Sensor Networks

Michele Chincoli*, Aly Aamer Syed†, Decebal Constantin Mocanu* and Antonio Liotta*
*Electrical Engineering Department, Eindhoven University of Technology, The Netherlands
Email: {m.chincoli; d.c.mocanu; a.liotta}@tue.nl
†NXP Semiconductors, Eindhoven, The Netherlands
Email: alyaamersyed@gmail.com

***Abstract*—Communications in Wireless Sensor Networks (WSNs) are affected by dynamic environments, variable signal fluctuations and interference. Thus, prompt actions are necessary to achieve dependable communications and meet quality of service requirements. To this end, the reactive algorithms used in literature and standards, both centralized and distributed ones, are too slow and prone to cascading failures, instability and sub-optimality. We explore the predictive power of machine learning to better exploit the local information available in the WSN nodes and make sense of global trends. We aim at predicting the configuration values that lead to network stability. In this work, we adopt the Q-learning algorithm to train WSNs to proactively start adapting in face of changing network conditions, acting on the available transmission power levels. Our aim is to prove that smart nodes lead to better network performance with the aid of simple machine learning.**

***Keywords*-wireless sensor network (WSN); transmission power control (TPC); Q-learning; software architecture**

I. Introduction

The Internet of Things (IoT) has triggered a new form of industrial revolution, promising to address the most compelling social and economic challenges. Networks of thousand objects will be integrated to the Internet, expanding the global network to several billion nodes. However, communications among so many objects is a true hurdle, as there are many different standards and each node competes the other ones for spectrum [1]. The common approach is to achieve connectivity by increasing the transmission power, particularly in high-density, high interference conditions. Yet this has overall detrimental consequences when it comes to global performance. We want to analyze the benefits in using a collaborative transmission power control algorithm based on predictions using machine learning rather than simplistic reactions.

Moreover because of the variability of the wireless channel and interference, the transmission power might be variable as well. For this reason, we propose the usage of a Reinforcement Learning (RL) algorithm for transmission power prediction based on the actual network conditions. This leads to reducing the overall transmission power in the network and, in turn, a more efficient spectrum and energy utilization.

II. Related Work

Many works in the literature have studied proactive or reactive solutions for Transmission Power Control (TPC). The main difference comes from the choice of link quality estimators, such as Received Signal Strength Indicator (RSSI), Packet Reception Ratio (PRR) and node degree; objective function (i.e., energy efficiency, contention or interference reduction); and on the methods and tools used for validation. In Adaptive Transmission Power Control (ATPC), the authors have discovered a correlation between RSSI and transmission power [2]. The transmission power is controlled by measuring instantaneous RSSI levels. Differently, in [3]-[4], Adaptive and Robust Topology control (ART) is studied where PRR is periodically calculated and compared based on two thresholds. Practical-Transmission Power Control (P-TPC) is a technique that, through the calculation of PRR and a feedback control loop, adapts the transmission power for the next sampling period [5].

All the above techniques are based on various observations and thresholds which, by default, are not capable of taking in consideration the dynamic nature of the WSNs environment. What is worse, the competitive and egoistic nature of those protocols is not compatible with the requirements of high-density wireless communications, whereby the increased power leads to a cascading effect across the network, leading to energy and spectrum wastage.

On the other hand, another interesting research direction in WSNs is given by the use of machine learning methods (e.g., reinforcement learning or RL) to optimize various aspects of WSNs. For instance, in [6] a RL based control mechanism is applied to achieve a high throughput and low power consumption. In [7], it is shown how simple RL (including just three actions - i.e. wait one time step, transmit with low power, and transmit with high power -) can achieve better performance in a stochastic environment (such as WSNs) than static power control algorithms. Also, in [8] it is shown that by using RL to control the sleep period of the sensors the system learns to increase its energy efficiency and network performance.

Despite the initial studies on the application of RL to WSN scenarios, to the best of our knowledge, predictive power control at a fine level of granularity has not yet been

978-1-4673-9948-7/16 $31.00 © 2016 IEEE
DOI 10.1109/IoTDI.2015.11

IEEE computer society

solved. Our aim is to come to understand how predictive algorithms can help improving massive-scale communications through collaborative reduction of transmission power.

III. Method

In this section we describe our approach, Predictive Transmission Power Control ($PreTPC$) in three different aspects: explanation of the Q-learning algorithm; software architecture; and basic operations of the method in a flow chart diagram.

A. Algorithm

We describe here the algorithm used to perform power control in a predictive manner. One of the most suitable self-learning paradigms for this task is reinforcement learning [9]. RL is inspired by psychology, and studies how artificial agents can learn to perform actions in a sequential order to achieve a specific goal. At any specific moment in time, an agent is located in a state. By picking an action to perform, the agent moves to a new state, in which it receives a reward (scalar value). The reward informs it how far it is from its goal (the final state). There are many variants of RL algorithms in the literature, but in this paper we focus on Q- learning [10], which is a widely used one. Furthermore, it has the advantage of being one of the lightest RL methods in terms of required computational resources. This makes it suitable to WSNs.

More formally, Q-learning is a model-free RL technique which learns to compute the quality Q of any state-action combination. The Q function is defined as $Q : S \times A \rightarrow R$, where S is the set of all possible states, A is the set of all possible actions and R is the set of all possible rewards. Initially, before learning, the Q function returns arbitrary fixed values, defined by the designer, and denoted by policy π. During the learning process, at each time t the agent selects an action a_t in a given state s_t. Then, it observes the new state s_{t+1} and a reward r_{t+1} given by the new state, and by using these observations it updates the Q function. Finally, after more iterations, the agent will learn an optimal policy $\pi*$. This policy will offer to the agent the knowledge to choose the best action in a given state to fulfill its goal. The update rule for the Q-learning algorithm is given by:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \\ \alpha_t(s_t, a_t) \cdot [r_{t+1} + \gamma \cdot \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)] \quad (1)$$

where $\alpha_t(s_t, a_t)$ is the learning rate, with all $\alpha \in [0, 1]$, and γ represents the discount factor. Furthermore, the balance between the exploration (learning) of the environment and the exploitation of the learned knowledge can be done using various strategies, such as the ϵ-greedy, ϵ-soft, or softmax [11].

It is clear that in the specific case of WSNs we have a multi agent environment, as each wireless node represents an agent. Herein, it is worth highlighting that a multi agent reinforcement learning framework may offer better performance. However, to avoid the overhead induced in WSNs by the extra messages required in such framework, in this paper we focus just on single agent Q-learning. Thus, each node is an Independent Learner (IL) and runs its own Q-learning algorithm without sharing information with its neighbors. In [12] it was shown that such a naive approach achieves good performance in practice.

For a successful Q-learning implementation the states, the actions, and the rewards have to be designed carefully. In our specific case, we define the set of the states of the environment as a triplet given by the number of retransmissions, the Clear Channel Assessment (CCA) attempts, and the latency; the actions as transmission power levels; and the reward as Packet Error Ratio (PER) level.

B. Software Architecture

To verify our method for WSN, we have used the concepts as defined in the IEEE 802.15.4 standard. The system comprises three main blocks (Fig. 1): 802.15.4 standard component with integrated features, DataBase Manager (DBM) and $PreTPC$.

802.15.4 is the IEEE standard for Low-Rate Wireless Personal Area Network (LR-WPAN) which is meant for WSNs. It specifies only the PHY and MAC layer according to the ISO model integrated in a sensor node. 802.15.4 MAC and PHY run as independent processes with respect to $PreTPC$. They adopt the DBM for collecting, storing and sharing data. The MAC layer is enabled to calculate PER and latency when a packet is transmitted and received. The transmission power, P_{tx}, is chosen polling the DBM by the Transmission Power Management Entity $TPME\_READ.request$ message. The chosen power can be either random (e.g. during the learning transient) or the best one predicted for a specific state. Therefore, P_{tx} is chosen and communicated in the reply by $TPME\_READ.confirm$. The value then is forwarded to the PHY layer that is in charge of actuation. The status of the local network condition, given as the combination of the number of the retransmissions, CCA attempts and latency of a single packet, is transmitted to the DBM by the $TPME\_WRITE.indication$ message. DBM in such case associates the combination of values to one value and stores it in one entry of a table. DBM controls

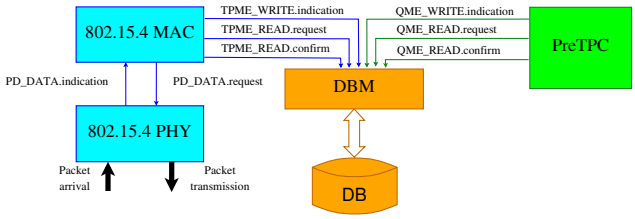

Fig. 1. Software Architecture schema

the memory of the device where the information for the method is stored. This information is provided to the other entities every time when DBM is polled. In $PreTPC$ the Q-learning algorithm is implemented and computed. Through Q-value Management Entity $QME\_WRITE.indication$ and $QME\_READ.request$ messages, $PreTPC$ communicates Q-values and requires the data (state, action and reward) respectively. In conclusion $QME\_READ.confirm$ is the reply to $QME\_READ.request$.

C. Flow Chart

As mentioned in the previous subsection, there are two processes that run in parallel, so similarly the flow chart (Fig. 2) can be read independently on the left and right sides. In the middle we can notice the description of DBM that interacts with the other two entities. On the left side, the additional operations in the MAC layer are shown. First of all, it is checked whether a packet is generated and ready to be sent or not. While in the latter case the process waits for the event to happen, in the former the random flag is read in DBM and the transmission power is computed accordingly. Then, either the random or best action is set. At this point the packet is transmitted and the ACK is expected. The reward based on PER is calculated and sent to the DBM. Other information of the transmission is aggregated, associated to a single value in DBM and stored in the $PacketReceptionHistory$ (PRH) table. On the right side, the tasks of the $PreTPC$ are detailed. The first step is to initialize all the values from the Q matrix to zero. Then the technique stops if no indication of packet history is received. Otherwise, the first entry of PRH is extracted and processed. Before updating the Q-value, the Q matrix convergence is examined. If the matrix has reached convergence then the training of the Q-learning algorithm is terminated. Otherwise, the Q-value related to a state and action is updated. Eventually the action is chosen with the mechanism of ϵ-greedy: a random number with standard uniform distribution is taken and compared with ϵ. If the value is lower than ϵ then the random action is selected, otherwise the best action is preferred. The decision is memorized in the random flag.

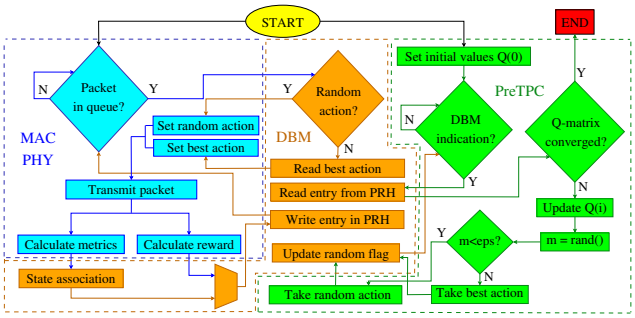

Fig. 2. Flow Chart diagram

IV. Preliminary Results

$PreTPC$ is tested in the network simulator NS3 release ns3.23. In this section, we show the preliminary results regarding the behavior of the method and the Q-learning algorithm. We focus mainly on the effect of TPC on the 802.15.4 MAC and PHY layers. We use a simple scenario, whereby two nodes are installed in a building, adopting [13], the site-general model in ITU-R P.1238-7, for the propagation model and Nakagami for the fading effects. The packet generation follows a Poisson distribution with the average rate of 4 kbit/s.

Considering the Q-learning policy, the system resides in one state, takes some action and moves to another state gaining a reward. Each state expresses the status of communication between the two nodes. Through the number of retransmissions, the system learns whether the receiver has a low Signal Interference Noise Ratio (SINR) due to low received power or high interference, thus losing one or more copies of a packet. Through the number of CCA attempts, the system instead learns whether or not the transmitter detects high energy in its range. That would be a symptom of high interference from the transmitter side. The attributes for the mechanism of retransmissions and CCA attempts in CSMA/CA follow the default values in the standard 802.15.4 specification. Finally, the latency provides more information on the conditions of the communication. It gives direct consequences of the previous factors and more causes (e.g., congestion) measured in terms of time. In our case, the actions represent the discretized transmission power levels. Only random actions are taken ($\epsilon = 1$) in order to study $PreTPC$ in its initial stage, while α is equal to 0,9 and γ to 0.8. The system has 20 available transmission power levels and the sensitivity is equal to -95 dBm. Our goal is that the nodes learn by themselves what the consequences of transmitting packets with a certain transmission power are. This happens through the reward for any specific state, which is given by a non-linear mapping function applied to the PER level, measured in percentages and calculated over the last 10 packets, as our goal is to optimize the communication performance in the WSN. This mapping makes the high PER values to correspond to low reward values and the low PER values to correspond to high rewards. In our specific case PER$\in[0, 100]$, while $r\in[-10, 10]$. As a result of all these parameters computed in Eq. (1), the Q-value for a specific state and action becomes a new Link Quality Estimator (LQE).

In this paper we intend to show a proof of concept that Q-value is an accurate LQE. Therefore we analyze Q-values by monitoring the state 0 and two actions: $MinPow$, where P_{tx} is equal to the minimum -35 dBm, and $MaxPow$, where P_{tx} is equal to the maximum 0 dBm. In the state 0 the number of retransmissions and CCA attempts is equal to zero, and the latency is smaller than 10 ms. During a

simulation, the Q-value of each combination state-action is updated at every packet reception and averaged at the end. Physically, in our scenario, the two nodes are placed at different distances starting from 5 up to 75 meters apart, and in each case the average Q-value is calculated. In this way, Q-values are tested when the signal at the receiver is very good and when weakens causing performance degradation (as the distance between the two nodes is increasing). The simulations results, repeated for 10 times with an average confidence interval of 1.22 and 1.51 for $MinPow$ and $MaxPow$ respectively, are depicted in Fig. 3, showing the trend of the average Q-values (y-axis), while the two nodes are moving apart (x-axis). While the distance between the two couples increases, Q-value diminishes from a high positive value of around 50 units down to a negative value of around -7, due to worsening of the channel quality. Since PER is calculated over a window of packets that have been randomly transmitted at different transmission powers as the nodes go more apart, the probability that more packets get lost within the window increases until a saturation stage is reached. This is expressed by the decay of the curves that becomes slower when the distance between the nodes is higher than 40 meters. The slight difference between the two curves lies on the fact that the reward is the PER of 10-packet long window with a shift of 1 packet. This means that in the case when the lowest transmission power is selected, the next transmitted packet will be lost differently than by using the maximum transmission power. This influences the Q-value update with given limits.

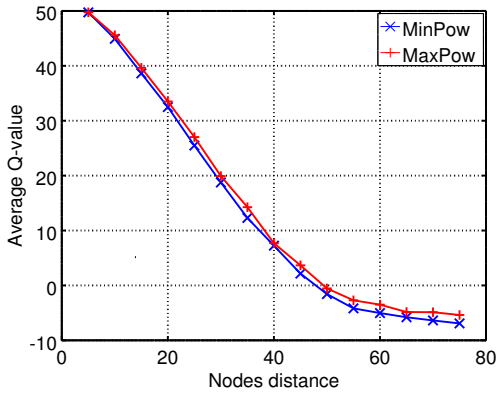

Fig. 3. Average Q-values for $MinPow$ and $MaxPow$ in state 0

V. Conclusion And Future Work

In this work we have introduced a predictive transmission power control technique. The system learns the effects of choosing a transmission power per network status through the help of a Q-learning algorithm. We presented some preliminary results and have shown as a proof of concept that the Q-value is an accurate LQE. As expected the Q-value reveals high positive values when the nodes are very close to each other because of the good link quality and becomes negative as the received power get closer to the sensitivity. We have observed that the values for both minimum and maximum transmission power are very similar and we have learnt that the direct effect of one transmission power on a singular packet loss is dependent to the effect of other transmission powers on other packets.

In the future, we are going to reinforce the independence of the reward caused by one action and to analyze the convergence of the Q matrix when the ϵ-greedy strategy is enabled.

Acknowledgment

This work was jointly supported by NXP Semiconductors (IMPULS program) and Eindhoven University of Technology (IMPULS program and INTER- IoT project grant 687283).

References

[1] A. Liotta, "The cognitive NET is coming," *IEEE Spectrum*, vol. 50, no. 8, pp. 26–31, 2013.

[2] S. Lin, et al., "ATPC: adaptive transmission power control for wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 223–236.

[3] G. Hackmann, O. Chipara, and C. Lu, "Robust topology control for indoor wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 57–70.

[4] M. Chincoli, et al., "Interference Mitigation through Adaptive Power Control in Wireless Sensor Networks." IEEE, Oct. 2015, pp. 1303–1308.

[5] Y. Fu, M. Sha, G. Hackmann, and C. Lu, "Practical control of transmission power for wireless sensor networks." IEEE, 2012, pp. 1–10.

[6] Z. Liu and I. Elhanany, "RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 117–124, 2006.

[7] A. Udenze and K. McDonald-Maier, "Direct Reinforcement Learning for Autonomous Power Configuration and Control in Wireless Networks." IEEE, Jul. 2009, pp. 289–296.

[8] S. Galzarano, et al., "QL-MAC: a Q-learning based MAC for wireless sensor networks," in *Algorithms and Architectures for Parallel Processing*. Springer, 2013, pp. 267–275.

[9] M. Wiering and M. van Otterlo, Eds., *Reinforcement Learning*, ser. Adaptation, Learning, and Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 12.

[10] C. J. C. H. Watkins and P. Dayan, "Technical note: q-learning." Mach. Learn., May 1992, pp. 279–292.

[11] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction." MIT Press, 1998.

[12] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *AAAI/IAAI*, 1998, pp. 746–752.

[13] P. Series, "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz," 2009.