

Experimental validation of a reinforcement learning based approach for a service-wise optimisation of heterogeneous wireless sensor networks

Milos Rovcanin^a, Eli De Poorter^a, Daniel van den Akker^b, Ingrid Moerman^a, Piet Demeester^a, Chris Blondia^b

^a*Ghent University - iMinds, Department of Information Technology (INTEC), Ghent, Belgium*

^b*PATS Research group, Department of Mathematics and Computer Sciences, University of Antwerp - iMinds, Antwerp, Belgium*

Abstract

Due to their constrained nature, wireless sensor networks are often optimised for a specific application domain, for example by designing a custom MAC protocol. However, when several wireless sensor networks are located in close proximity to one another, the performance of the individual networks can be negatively affected as a result of unexpected protocol interactions. The performance impact of this ‘protocol interference’ depends on the exact set of protocols and (network) services used. This paper therefore proposes an optimisation approach that uses self-learning techniques to automatically learn the optimal combination of services and/or protocols in each individual network. We introduce tools capable of discovering this optimal set of services and protocols for any given set of co-located heterogeneous sensor networks. These tools eliminate the need for manual reconfiguration while only requiring minimal a priori knowledge about the network. A continuous re-evaluation of the decision process provides resilience to volatile networking conditions in case of highly dynamic environments. The methodology is experimentally evaluated in a large scale testbed using both single- and multihop scenarios, showing a clear decrease in end-to-end delay and an increase in reliability of almost 25 percent.

Keywords:

Network cooperation, self-awareness, reinforcement learning, linear approximation, network service negotiation, ϵ greedy, logarithmic state

access distribution;

1. Introduction

We are witnessing a continuous increase in the number of wireless communicating devices surrounding us. For constrained wireless sensor networks (WSNs) specifically, many researchers focus on developing optimal network solutions for very specific application domains. This is exemplified by the large number of MAC protocols that exists for WSNs. However, since wireless sensor networks mostly use proprietary or highly customised protocols in unlicensed frequency bands, inter-protocol interference is a typically occurring problem. The fact that these protocols often operate under the implicit assumption that they are ‘alone’ in the wireless environment can cause harmful interference between these protocols [31]. As a result, choosing network protocols and services that perform excellent in one specific environment (e.g. TDMA MAC protocols for optimised single-hop networks) can result in degraded performance in other environments where other sensor networks are also present. Manual configuration and selection of the optimal network protocols proves to be complex and inefficient [1], mainly due to the sheer amount of devices (time consuming) and inability to take dynamically changing network environments into account. Choosing the optimal set of protocols and services is a multi-objective optimisation problem: individual networks generally have different application level requirements, which may even change over time.

To cope with this complexity, intelligent solutions that allow networks to efficiently reconfigure themselves at run time are needed. Such solutions are expected to increase the network performance and simplify the setup of networks for the end users [2]. This paper proposes a cooperation paradigm in which co-located networks are automatically reconfigured at run-time to activate the set of protocols and services that allow the involved networks to operate optimally. The proposed solution relies on a reinforcement learning algorithm that efficiently combines multi-objective optimisation (MOO) with the reinforcement learning (RL) paradigm. It uses linear fitness approximations, extensively used for solving MOO problems, and applies these to the general RL methodology. The cooperation process is initiated without any priori knowledge about the environment.

The main goal of the cooperation process is to verify if the network services and/or network protocols are correctly chosen so that they (i) fulfil the

requirements imposed by a higher level application and (ii) do not negatively influence each other. Since individual networks may benefit from different types of medium access control (MAC) protocols (TDMA, CSMA-CA, ...), the approach supports replacing the full *MAC layer* to support the respective application requirements. In addition, different routing protocols or transport layers may be available in each network and thus also be considered during the process. Finally, *packet sharing* (allowing networks to route packets for each other), *aggregation* (combining multiple data-items in a single packet) and similar network services can be enabled at higher layers depending on the application requirements. Each of these services is an additional variable in the proposed multi-variable optimisation problem. The algorithm allows the optimal operational point (the optimal set of services and protocols) to be selected, while still being able to adapt to changes in the network (eg.: altered interference patterns) or altered application requirements. It is of course possible that the most optimal set of protocols and services for cooperation between the networks yields poorer results than not cooperating at all. The algorithm must be able to detect this in order to allow cooperation to only be enabled if it is beneficial for all participating networks.

The main contributions of this paper include the following:

1. An overview of optimisation and self-learning algorithms for wireless sensor networks.
2. Experimental demonstration that the selection of the optimal network protocols (e.g. MAC protocol) depends on more than the application requirements and should also take into account other networks in the wireless environment.
3. Introduction of a methodology that:
 - is capable of solving a multi-objective optimisation problems;
 - takes into account heterogeneous requirement from multiple co-located networks;
 - can detect degraded performance due to unpredictable interaction between protocols and/or faulty (e.g. buggy) protocol implementations;
 - can adapt the network configuration to take into account changing network requirements and dynamic environments.
4. Experimental evaluation and analysis of the obtained performance gains using a large scale wireless sensor testbed.

The remaining part of the paper is organised as follows. Section 2 gives a brief overview of other machine learning techniques used in the context of sensor networks. A detailed problem statement is given in Section 3. Section 4 introduces the main mathematical concepts of reinforcement learning and the LSPI algorithm. Section 5 explains how the RL methodology is used as a solution to this particular problem. An experimental setup, used for validation and evaluation of the algorithm, is introduced in Section 6. Results and corresponding discussions can be found in Section 7. Future work is described in Section 8, while Section 9 concludes the paper.

2. Related work

The first part of this section gives a brief overview of the optimisation techniques being used for multi-objective optimisation (MOO) problems in heterogeneous wireless sensor networks. The second part presents the most relevant work regarding application of different RL techniques to the most common sensor network problems such as routing, energy efficiency, medium slot allocation etc.

2.1. MOO tools used in heterogeneous WSNs

MOO solutions are typically used to quickly converge to an optimal operating point of a problem in a stable environment with multiple input parameters. Several MOO techniques have been used previously to optimised WSNs.

The authors of [3] propose two evolutionary algorithms (EAs): NSGA-II (Non-dominated Sorting Genetic Algorithm II) [4] and SPEA-II (Strength Pareto Evolutionary Algorithm II) [5] as tools for solving an NP-hard problem of a heterogeneous WSN deployment, while maximising its reliability and minimising packet delay.

Another example is a simultaneous optimisation of the high network lifetime and coverage objectives, tackled in [6]. As opposed to previous methods that tried combining the two objectives into a single objective or constraining one while optimising the other, the newly proposed approach employs a recently developed MO evolutionary algorithm based on decomposition - MOEA/D [7] as a feasible solution.

In [8], a multi-objective hybrid optimisation algorithm is combined with a local on line algorithm (LoA), to solve the Dynamic Coverage and Connectivity problem in WSNs, subjected to node failures. The proposed approach

is compared with an Integer Linear Programming (ILP)-based approach and similar mono-objective approaches, regard coverage, network lifetime. Results show that the presented hybrid approach can improve the performance of the WSN, with a considerably shorter computational time than ILP.

ILP was also used in the service-wise network optimisation problem, published in [2]. The authors used a solution based on a the linear programming methodology in combination with the IBM CPLEX ILPSolver [9] to determine the optimal operational point. In order to produce useful results however, it relies on the expected performance gains for reconfiguration, which is rather difficult to obtain.

Since many of these solutions require stable a priori information, they are mainly useful for well controlled and non-volatile environments, which is quite the opposite of the type of environment for which our algorithm has been developed.

2.2. RL in WSNs

Reinforcement learning predicts future behaviour based on information from the past. As such, Reinforcement Learning is well suited for dynamic environments that show limited change over time.

LSPI (Least Squares Policy Iteration), as a form of reinforcement learning, has previously been used to optimise network layers above the physical layer. Routing and link scheduling problems have been tackled in [10] and [11].

An autonomic reconfiguration scheme that enables intelligent services to meet QoS requirements is presented in [12]. The authors apply, the Q learning technique [13] is to the *route request/route reply* mechanism of the AODV routing protocol [14] in order to influence the failure or the success of the process and thereby decreasing the protocol overhead and increasing the protocols' efficiency.

In [15] a number of approximate policy iteration issues, related to our research such as convergence and rate of convergence of approximate policy evaluation methods, exploration issues, constrained and enhanced policy iteration are discussed. The main focus is on the above mentioned LSTD and its scaled variant algorithm.

Research published in [16] tests and proves the convergence of a model free, approximate policy iteration method that uses linear approximation of the action-value function, using on-line SARSA updating rules. The update rule is how the algorithm uses experience to change its estimate of the optimal value function. SARSA updating is exclusively used in on-policy algorithms,

where the successor's Q value, used to update the current one, is chosen based on the current policy and not in a greedy fashion, as with Q-learning.

Our algorithm uses mechanisms similar to the ones discussed above, but applies these to a new problem domain. As a result, while searching for the optimal set of services and protocols, we expect our methodology to provide a precious insight into dependencies between various network protocols and services. This will be beneficial for future research, as well as for the rest of the research community.

3. Use Case

In this paper we consider a scenario in which two WSNs are deployed in the same wireless environment. Both use the well known but resource constrained Tmote Sky sensor nodes [17]. One network runs an intrusion detection application, while devices belonging to the other network collect temperature measurements.

3.1. Properties of the security network

The following requirements are set up for the security network:

- LONG NETWORK LIFETIME
- LOW END-TO-END DELAY
- HIGH RELIABILITY

Having a long network lifetime is a common requirement to avoid frequent replacement of the batteries in energy constrained wireless sensor networks. The requirements for a low delay and high reliability are motivated by the fact that intrusion events should be reported fast and reliable.

The security network can choose between three different MAC protocols - Time Division Multiple Access (TDMA) [18], Low Power Listening (LPL) protocol [19] and Carrier Sense Medium Control with Collision Avoidance (CSMA/CA) [20]. In addition, two higher layer network services, AGGREGATION and PACKET SHARING, are available. The first service, aggregation [21], is capable of combining multiple data packets in a single packet to reduce the packet overhead. When the PACKET SHARING service is enabled, packets from other networks can be routed over the network. Prior to cooperation, each network selects a preferred MAC protocol based on

its application requirements. The influence of a higher layer network service, AGGREGATION and PACKET SHARING, is taken into consideration once the cooperation is initiated. Within the scope of this work we require the PACKET SHARING service to be enabled either in both networks simultaneously or not at all. The reason for doing so is that enabling this service in one network only is expected to mainly provide performance benefits for the other network (shorter routing paths) while having an (energy wise) impact on the network itself. As a result, requiring a network to enable PACKET SHARING is only ‘fair’ if it is done in both networks at the same time.

3.2. Properties of the temperature monitoring network

The following requirements are set up for the temperature monitoring network:

- LONG NETWORK LIFETIME
- LOW DELAY

Since the monitoring network is not used for critical services, the high reliability requirement is omitted in favour of obtaining a high network lifetime. The set of available services, MAC protocols and a higher level services, completely matches the case of the security network - TDMA, CSMA-CA and LPL, accompanied with AGGREGATION and PACKET SHARING.

3.3. The cooperation process

The process of cooperation starts by exchanging the relevant information between the different networks: available services and predefined user requirements. A dedicated reasoning engine, connected to a sink of one of the two networks, collects all data and iteratively applies multiple service combinations to the networks to determine the most optimal configuration. Once discovered, the most optimal configuration is applied and maintained in both networks. It should be noted that network requirements may change over time. Changes in the network topology or available resources (battery power) might require a re-evaluation of the previously obtained results. Adding a significant number of nodes to a network may, for example, degrade the performance of the MAC protocol to a point that it is better to switch to a different MAC protocol entirely. The reasoning engine must be able to notice such changes in a reasonable time span and reconfigure the networks accordingly. A well know and widely used SOFT MAX [22] state exploration methodology is used to balance between these two confronting objectives.

4. Reinforcement learning

Reinforcement learning [23] is a formal mathematical framework in which an agent manipulates its environment through a series of actions, and in response to each action, receives a reward value. Reinforcement learning (RL) emphasises the individual learning through interactions with his environment, as opposed to classical machine learning approaches that privilege learning from a knowledgeable teacher, or on reasoning from a complete model of the environment [24]. The learner is not told which action to take. Instead, it must find which actions yield a better reward after trying them. The most distinguishing features of reinforcement learning are trial-and-error search and delayed reward.

4.1. RL mathematical fundamentals

RL models a problem as a Markov Decision Process (MDP). Relying on it, the agent can perceive a set $S = (s_1, s_2, \dots, S_n)$ of distinct states and has a set $A = a_1, a_2, \dots, a_n$ of actions it can perform at each state. The agent senses the current state S_t , chooses a current action a_t and performs it. The environment responds by returning a reward $r_t = r(S_t, a_t)$ and by producing the successor state $s' = P(S, a)$. Functions r and $P(s, a)$ are not necessarily known to the agent.

A numerical value, $Q(s, a)$, is assigned to every state/action pair (s, a) , describing the payoff of a given action. The general outlook of the Q function is known as the Bellman equation:

$$Q(s, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q(s', a') \quad (1)$$

where $r(s, a)$ represents the immediate reward for executing action a at state s , while the other argument represents the maximum expected future reward. Factor γ is known as the *discount* factor and its purpose is to make sure that a reward given for the same state/action pair is decreasing over time.

The goal of RL is to learn an optimal behavioural policy function, $\pi(s, a)$, which specifies the probability of selecting action a in a state s , for all states and actions. An optimal policy is one that maximises the expected total return. In “one-step” decision tasks, the return is simply the immediate reward signal. In more complex tasks, the return is defined as the sum of individual reward signals obtained over the course of behaviour.

4.2. Least Squares Policy Iteration - LSPI

LSPI was first introduced by M.G.Lagoudakis and R.Parr [25] as a reinforcement learning solution that efficiently copes with large state spaces. LSPI is a model-free, off-policy method which efficiently uses “sample experiences”, collected from the environment in any manner. The basic idea is reflected through an approximation of a Q function with a linear combination of *basis functions* and their respective weights:

$$Q(s, a; w) = \sum_k \phi_j(s, a)\omega_j \quad (2)$$

Basis functions represent the relevant problem features (e.g. network’s duty cycle, link quality, residual energy of nodes etc). Generally, their number is much smaller than the number of state/action pairs, $k \ll |S||A|$. The ultimate outcome of the algorithm, for the given decision making policy, is the set of weights: $W = (\omega_1, \omega_2, \dots, \omega_k)$.

The mathematical background of the LSPI algorithm, along with a couple of simple application use cases (bicycle ride, inverted pendulum), can be found in [26]. Its application to a multi-objective optimisation problem in heterogeneous wireless networks is given in [27].

5. Framework Construction

This section explains how the aforementioned RL algorithm can be used as a solution to the problem presented in Section 3. As discussed in 4.1, reinforcement learning transforms a given problem into a Markov decision process. Constructing the framework therefore starts with defining the main properties of the underlying MDP.

5.1. States

Let S_i and M_i be the number of services and MAC protocols available for network i . Since each service can either be active or inactive and every service combination can be used with every available MAC protocol, there are a total of $M_i 2^{S_i}$ possible configurations for each network i . Assume for instance that there are two cooperating networks, each providing a set of two network services (Network 1 - serviceA, serviceB; Network 2 - serviceC, serviceD). In that case there are $4^2 = 16$ different service combinations $\{A\}$, $\{B\}$, $\{C\}$, \dots , $\{ABCD\}$, where each combination represents a single set of activated network services. It should be noted that the different networks

are not required to use the same MAC protocol. As further discussed in Section 6, Virtual Gateways [29] are used to enable communication between networks using different MAC protocols. This allows each network to choose its MAC protocol independently from the MAC protocols used by the other participating networks. The total number of states to consider can therefore be defined as follows:

$$N_{states} = \prod_{i \in networks} M_i 2^{S_i} \quad (3)$$

Within the scope of this work we assume that each network has already determined the two most optimal MAC protocols to use prior to engaging in cooperation. This can, for instance, be achieved by applying the methodology presented in this paper to the single network case. Moreover, as discussed in Section 3, there are two services that can be enabled in each network: *Aggregation* and *Packet sharing*. This yields a total of 64 separate states. The *Packet Sharing* service can only be activated in both networks at the same time, which ultimately reduces the number of states down to 32.

5.2. Actions

The underlying MDP allows a decision maker to switch between any two states, meaning that N_{states} actions are available at each state. Taking an action can produce two distinguishable outcomes:

- The engine stays in the current state
- The engine switches to another state

Preserving a current state (taking an action that will keep the engine in the same state in two consecutive episodes) is what the algorithm aims for: discover the optimal state and force an action that will keep it in that particular state from then on. However, due to the nature of the state exploring mechanism, this can also happen if the selected state is not the most optimal one. This will be discussed further in the following sections:

One important property of the designed MDP is that state transition probability $P(s'|s, a) = 1$. In other words, Taking an action a in the state s will always result in state s' .

5.3. Basis functions

Basis functions are indicators of the network performance, regarding given goals. Each network requirement (HIGH RELIABILITY, HIGH NETWORK LIFETIME, LOW DELAY etc.) can be described with one or several basis functions (relevant features). To prevent redundancy, basis functions should be designed to be independent of one another. In combination with the respective weights, they are crucial in the process of calculating the state/action Q values. Generally, the number of basis functions is much smaller than the number of state/action values, $k \ll |S||A|$.

In our use case we rely on a single basis function per network requirement:

- HIGH RELIABILITY - average packet loss (ϕ_1)
- LONG NETWORK LIFETIME - duty cycle (ϕ_2)
- LOW END-TO-END DELAY - average hop count (ϕ_3)

Information about the average packet loss is obtained by comparing the number of packets generated in each network to the total number packets received for each network at the sink node. Hop count and duty cycle information is piggy-backed to every data packet and used at the sink to calculate the average value for each network.

Based on the above requirements, the Q function can be calculated as follows:

$$Q(s, a) = \phi_1\omega_1 + \phi_2\omega_2 + \phi_3\omega_3 \quad (4)$$

5.4. Rewards

A straightforward way of defining the reward function is the relation between the predefined (required) and the measured network performance. In this multi-objective framework, the total reward is calculated as a combination of individual rewards, given for each network requirement (LONG NETWORK LIFETIME, LOW END-TO-END DELAY, HIGH RELIABILITY etc.)

It is useful to enforce an upper limit to the contributions of each network metric. Otherwise, a service combination that significantly ‘overshoots’ one requirement can receive a higher reward than the ones performing somewhat worse, but equally accomplishing all the given requirements. The rewarding function is designed to prevent such behaviour:

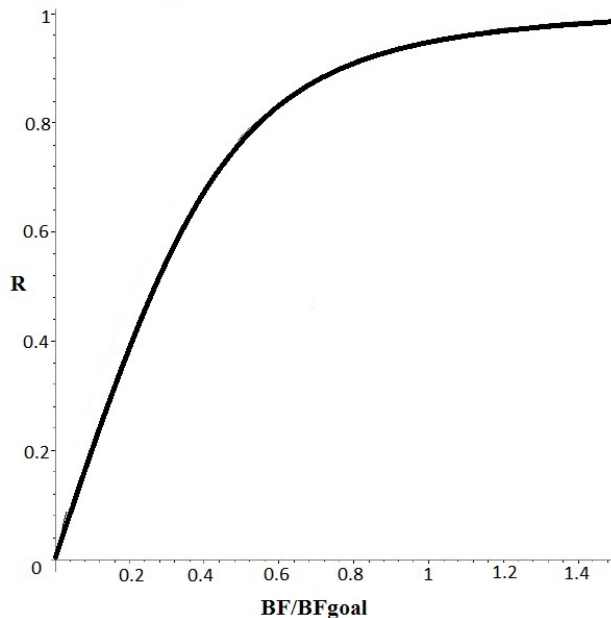


Figure 1: Rewards are calculated using the relative difference between the basis function values collected at the end of an episode and the desired values. The function also sets up a horizontal asymptote to an associated reward, thus making sure the reward increases slowly once the requirements are met

$$R_i = 1 - e^{-3\frac{\phi_i}{\phi_{goal}}} \quad (5)$$

The function increases slowly once the requirements are met (see Figure 1). If the requirements do not describe an upper performance limit, rewards can be unlimited.

5.5. Collecting environmental information

Section 4.2 introduced the idea using information samples, $D = (s_{d_i}, a_{d_i}, s'_{d_i}, r_{d_i} | i = 1, 2, \dots, L)$, in order to ultimately form the approximated version of matrices A and b , crucial in decision policy evaluation. There is a general rule:

$$E(\hat{A}) = \frac{L}{|S||A|} AE(\hat{b}) = \frac{L}{|S||A|} b \quad (6)$$

which describes consistency between the approximated and real values

of matrices A and b , depending on the number of collected samples. The precision of the algorithm increases with the growing number of samples.

By relying on the memorylessness property of the designed MDP, our algorithm is capable of collecting the information regarding every state/action pair from the problem space in N_{states} exploration episodes. The memoryless property is satisfied by the fact that the performance of the network in any given state only depends on the service combination related to that particular state. If, for example, the network is in state S_x during a specific episode, the values of the relevant basis functions $\phi_1, \phi_2, \dots, \phi_k$, (used for calculating the reward) collected at the end of that episode, depend solely on the specific service combination related to state S_x and not on the previous state or the action taken to get to state S_x . This means that transitions $(s_i \rightarrow S_x | i = 0, 1, 2, \dots, n)$, cause by actions a_0, a_1, \dots, a_n , all result in the same values of the relevant basis functions $\phi_1, \phi_1, \dots, \phi_k$. Consequently N_{states} separate Q values, $Q(s_0, a_x), Q(s_1, a_x), \dots, Q(s_n, a_x)$, can be updated after a single episode. (Where a_x denominates the actions that leads the system from whatever state into state s_x).

Relying on this property, the algorithm is divided into two phases:

- Exploration phase - A constrained random walk is used to collect all the samples in as many as N_{states} episodes. Matrices A and b are populated and the initial set of weight factors $W = (\omega_1, \omega_1, \dots, \omega_k)$ is calculated. In combination with the respective basis functions, this set of weights is used to calculate the initial Q values for every state/action pair.
- Exploitation phase - This phase relies on the adopted SOFTMAX state exploring technique. It utilises the initial Q values and tries to enforce the optimal service combination (the optimal state) as much as possible, while investigating the sub-optimal states in order to detect possible performance changes.

Information collected during the exploitation phase is used to update an already existing sample set. The new set of weights is calculated after each episode and the corresponding Q values are updated.

5.6. Required changes for other use cases

Building up a framework is typically use case specific. States, actions and rewards are differently interpreted depending on the scenario. However,

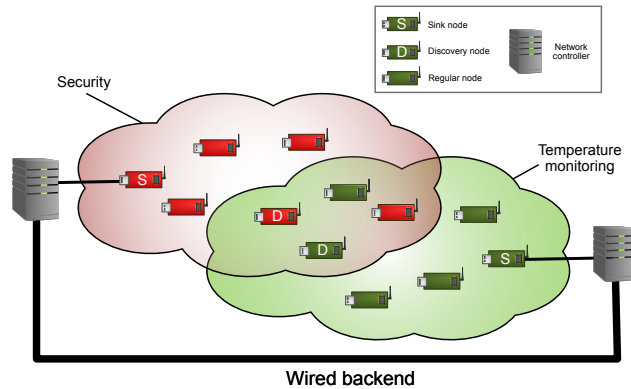


Figure 2: The network architecture used during the experimental tests. Two networks are co-located, a security network and a temperature monitoring network.

the structure of the underlying MDP and the mathematical apparatus that governs it remain the same. Within our problem scope, LSPI's usage can be expanded to additional fields of research.

We provide two examples:

- A straightforward modification is to apply the same concepts to a use case in which the operator has control over the settings of a single network protocol. In this use case, the number of states and actions would directly depend on the number of configurable properties of the protocol and the rewards would be calculated depending on the relevant performance metrics for the given protocol.
- A similar modification can be applied when a single network is under full control of the operator, but uncontrollable outside influences are present. A service-wise optimisation of a single network can then be performed by again selecting the optimal set of services and protocols, this time by taking into account only the performance of the single network. Consequently, the number of state/action pairs would depend on the number of configurable services, plus the number of variable settings for each service. Rewards would be calculated in accordance to a network's application level objectives.

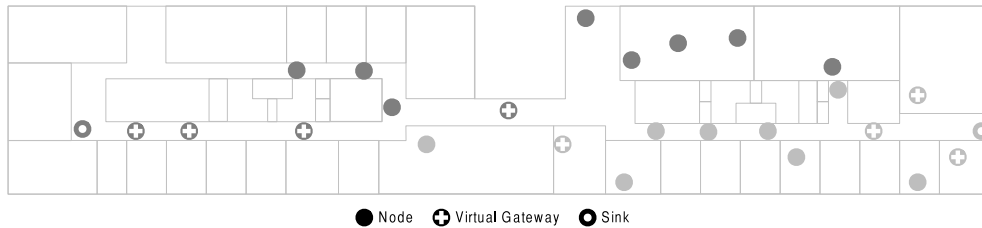


Figure 3: The node deployment used for the real-life testing setup. We use 26 nodes deployed on a single floor of the office building in which the testbed is deployed. These nodes are separated into two networks of 13 nodes each so each network covers the entire floor. The sink nodes are placed at opposite ends of the building and 6 fixed nodes are used as virtual gateways.

6. Experimental setup

In this section the experimental setup used to evaluate the reinforcement learning algorithm is discussed. Figure 2 shows the different nodes which are deployed in the ‘security’ and the ‘temperature monitoring’ network. All sensor nodes in both networks periodically generate data packets which are subsequently forwarded over multiple hops to the nearest available sink. In addition to the ‘regular’ measurements collected by the nodes (temperature, movement detection, ...) these packets also contain duty-cycle and hop-count statistics. The discovery nodes in the networks also periodically broadcast ‘discovery messages’ to allow cooperation between different networks to be initiated. These ‘discovery messages’ contain, among others, the available services and the requirements of the network. The sink node of the network regularly broadcast ‘sink announcement’ messages that are used by the other nodes to discover the route to the nearest (available) sink.

The sink node is connected to the ‘network controller’ over a serial line. The network manager processes the packets received from the sink to calculate network statistics. Once two networks have decided to cooperate, a connection between the network controllers is created over a wired back-end to allow ‘foreign’ data packets to be forwarded to the correct network manager and to allow statistics to be exchanged. The ‘RL engine’ runs on one of the network manager nodes and calculates the configurations of the networks based on the gathered performance statistics and the services and requirements announced in the ‘discovery messages’. Once a new configuration has been calculated, the network manager running the RL engine sends the configuration to the attached sink node which subsequently distributes the new

configuration in the local network. Upon reception of a new configuration, the discovery of the local network forwards the configuration to the discovery node of the ‘foreign’ network which subsequently distributes the configuration in its own network. Afterwards, an activation message is distributed in the same manner to both networks to instruct the nodes to apply the new configuration.

The sensor node software was developed for the T-mote SKY [17] platform using the IDRA framework [28] and the MultiMAC [29] network stack. The MultiMAC network stack is a replacement network stack for TinyOS 2.1.0 that allows multiple MAC protocols to be used simultaneously on a single node. This allows normal sensor nodes to be configured as so-called ‘Virtual Gateways’ which enable communication between nodes using different MAC protocols. Since the requirements of the ‘security’ and ‘temperature monitoring’ network may cause these networks to use different MAC protocols, the presence of Virtual Gateway nodes is essential to allow for cooperation between these networks. The IDRA-framework allows for the easy development of sensor network applications and protocols and was therefore used on top of the MultiMAC network stack to develop the applications and reconfiguration mechanisms needed for our tests.

All tests were performed on the w-iLab.t [30] wireless testbed, which contains several Tmote Sky sensor nodes deployed in an office building. The deployment used is shown on Figure 3.

It should be noted that the performance of the networks depends on which and how many sensor nodes are used as Virtual Gateways. Determining the ideal location of these Virtual gateway nodes however is out of the scope of this work and as a result a ‘fixed’ set of gateway nodes was used for each configuration.

7. Results and discussions

The performance of the reasoning engine and the cooperation methodology is tested using two scenarios: a single-hop scenario (whereby nodes use full transmit power) and a multihop scenario (obtained by reducing the transmit power). Results are further divided into two subsections:

- Results regarding the exploration phase of the algorithm
- Results regarding the exploitation phase of the algorithm

7.1. Single-hop networks scenario

When using full transmit power, the nodes in both networks can reach their sinks in one hop. Due to optimisations in the used RPL-based routing protocol, packets are sometimes routed using an intermediate node to avoid unreliable links. To obtain baseline performance indicators, the performance of the individual networks was first evaluated without cooperation under the following conditions.

- “Stand alone” case. The performance is measured for each network individually, without the other network active. As a result there is no interference between the two networks.
- “Interfered” or “conflicted” case. Networks are co-located but ignore each other entirely, resulting in negative protocol interactions.

Figure 4 shows the average duty cycle and reliability measured in both networks for both the “Stand alone” and “Interfered” case. Results are classified depending on the MAC protocol used during the tests.

A clear performance deterioration for both networks caused by interference, can be observed. The duty cycle increases by at most 20% while the reliability decreases by up to 18%. Only the average number of hops (1.12 for the temperature network and 1.45 for the security network) remains unchanged. Applying the proposed cooperation methodology is expected to shift performances back towards the results obtained during the “stand alone” case.

7.1.1. Discussion on the exploration phase of the algorithm

Both exploration and exploitation phases are performed using 5 minute long episodes. Each node generates a packet once every ten seconds, resulting in six packets per minute. Information regarding the average packet loss, duty cycle and number of hops is retrieved during each episode. The performance of the different states is calculated according to Section 5 and shown in Figure 5 and Figure 6.

The best performing service set is the one that enables the AGGREGATION in both networks, in conjunction with the TDMA and LPL MAC protocols in the temperature monitoring and security network, respectively. While the choice of MAC protocols was expected, due to the network lifetime requirement, the influence of other network services was more difficult to predict. This is true for both the single-hop and the multihop use cases. The obtained results illustrate the following:

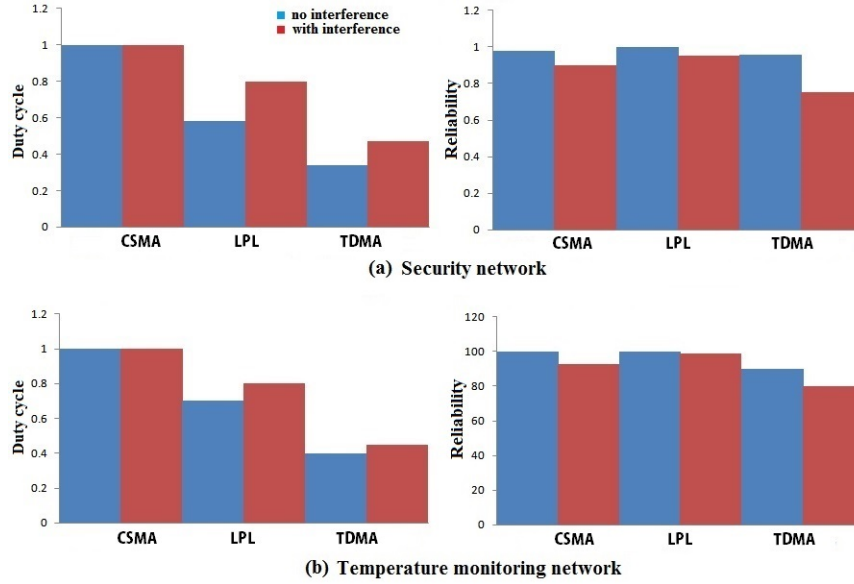


Figure 4: Network performance of the (a) Security network and (b) temperature monitoring network, in terms of a duty cycle and reliability metrics, in situations with and without influences of co-located devices. Tests are performed using different MAC protocols

- Although TDMA MAC protocol is the optimal MAC protocol in the “stand alone” case, this is no longer true when a second co-located network, also using the TDMA protocol, is present. In that case, the optimal performance is achieved by enabling TDMA in the temperature monitoring network and the LPL MAC protocol in the security network.
- In general, AGGREGATION and PACKET SHARING services, do not significantly impact the overall network performance in a single-hop network scenario. This is understandable, since the great majority of the nodes are one hop away from the sink, therefore neither AGGREGATION nor PACKET SHARING is frequently used.
- The obtained reliability for the optimal set is around 99 percent, compared to 93 percent recorded before the cooperation was activated.
- The duty cycle of the security network, when using the optimal service combination, decreased for 12.5 percent (80% down to 70%, while using

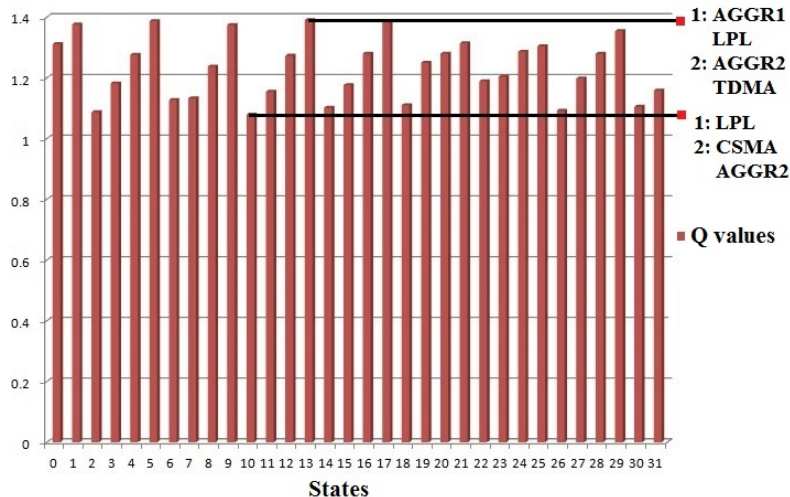


Figure 5: Graphical illustration of the results, obtained during an exploration phase. Service combinations are evaluated using basis functions and rewards explained in Section 5

LPL MAC protocol). The duty cycle of the temperature monitoring network increases from 48%, as noted prior to negotiation up to 54% (12 percent), while using TDMA. This results in a fairer distribution of the network lifetime. In addition, the same configuration results in a 6 percent (from 93 to 99 percent) higher reliability for both networks.

Figure 7 illustrates the performance improvement when applying our cooperation methodology. Even for single-hop networks, taking into account the presence of co-located networks has a significant impact on the performance. Trying to predict which combination of settings and protocols will perform best can be difficult at best, especially when trade-offs have to be made between multiple performance criteria. Our methodology is capable of objectively making this trade-off even in complex situations.

7.1.2. Discussion on the exploitation phase of the algorithm

This section evaluates how efficient our methodology copes with networks changes. The efficiency of the exploitation phase is evaluated based on two criteria:

- The algorithm’s ability to enforce the optimal service set.

State (dec)	State (bin)	PS	Aggr2	Aggr1	MAC2	MAC1	Qvalue	Duty Cycle2	Duty Cycle1	Hops2	Hops1	Reliability
0	00000	-	-	-	TDMA	CSMA_CA	1.311	0.58	1	1.22	1.45	92%
1	00001	-	-	-	TDMA	LPL	1.376	0.56	0.77	1.2	1.45	99%
2	00010	-	-	-	LPL	CSMA_CA	1.087	0.90	0.99	1.2	1.54	87%
3	00011	-	-	-	LPL	LPL	1.182	0.72	0.92	1.2	1.45	92.3%
4	00100	-	-	AGGR	TDMA	CSMA_CA	1.276	0.62	1	1.2	1.45	98%
5	00101	-	-	AGGR	TDMA	LPL	1.387	0.54	0.86	1.2	1.45	93%
6	00110	-	-	AGGR	LPL	CSMA_CA	1.127	0.86	1	1.2	1.45	89%
7	00111	-	-	AGGR	LPL	LPL	1.133	0.87	0.88	1.2	1.45	96%
8	01000	-	AGGR	-	TDMA	CSMA_CA	1.237	0.62	1	1.22	1.45	95%
9	01001	-	AGGR	-	TDMA	LPL	1.374	0.56	0.77	1.2	1.45	90%
10	01010	-	AGGR	-	LPL	CSMA_CA	1.079	0.91	1	1.2	1.45	91%
11	01011	-	AGGR	-	LPL	LPL	1.154	0.86	86	1.2	1.45	97%
12	01100	-	AGGR	AGGR	TDMA	CSMA_CA	1.273	0.62	1	1.2	1.45	92%
13	01101	-	AGGR	AGGR	TDMA	LPL	1.39	0.54	0.73	1.2	1.45	97%
14	01110	-	AGGR	AGGR	LPL	CSMA_CA	1.102	0.86	1	1.2	1.45	89%
15	01111	-	AGGR	AGGR	LPL	LPL	1.176	0.84	0.80	1.2	1.45	96%
16	10000	PS	-	-	TDMA	CSMA_CA	1.28	0.62	1	1.1	1.45	92%
17	10001	PS	-	-	TDMA	LPL	1.385	0.57	0.72	1.22	1.45	99.1
18	10010	PS	-	-	LPL	CSMA_CA	1.11	0.88	0.99	1.1	1.45	86%
19	10011	PS	-	-	LPL	LPL	1.25	0.84	0.85	1.1	1.1	98%
20	10100	PS	-	AGGR	TDMA	CSMA_CA	1.28	0.65	1	1.1	1.45	94%
21	10101	PS	-	AGGR	TDMA	LPL	1.314	0.61	0.81	1.2	1.45	93%
22	10110	PS	-	AGGR	LPL	CSMA_CA	1.189	0.78	1	1.1	1.45	92%
23	10111	PS	-	AGGR	LPL	LPL	1.204	0.94	0.89	1.1	1.09	96%
24	11000	PS	AGGR	-	TDMA	CSMA_CA	1.286	0.65	1	1.1	1.45	97%
25	11001	PS	AGGR	-	TDMA	LPL	1.304	0.62	0.91	1.2	1.45	99%
26	11010	PS	AGGR	-	LPL	CSMA_CA	1.092	0.95	0.99	1.1	1.45	93%
27	11011	PS	AGGR	-	LPL	LPL	1.198	0.93	84	1.1	1.1	95%
28	11100	PS	AGGR	AGGR	TDMA	CSMA_CA	1.28	0.65	0.99	1.1	1.45	94.7%
29	11101	PS	AGGR	AGGR	TDMA	LPL	1.354	0.57	0.88	1.2	1.45	97%
30	11110	PS	AGGR	AGGR	LPL	CSMA_CA	1.105	0.90	1	1.1	1.54	92%
31	11111	PS	AGGR	AGGR	LPL	LPL	1.158	0.86	0.94	1.1	1.1	100%

Figure 6: Tabular presentation of the exploration phase results

- The algorithms ability to readjust its decisions when drastic performance changes occur.

Figure 8 illustrates the algorithm's behaviour with the temperature τ factor set to two distinguish intervals. As expected, for the higher values of the temperature factor, SOFTMAX acts in a uniform way, treating all decisions as equiprobable. Because of this, the reasoning engine often chooses decisions that result in a transfer to sub-optimal states. Lower values of the τ factor clearly results in enforcement of the highest regarded service combinations, keeping the relevant networking parameters (duty cycle, reliability ...) on a highest level and stable.

The algorithm's ability to adapt to sudden changes is tested by suddenly increasing the duty cycle of the TDMA protocol in the temperature monitor-

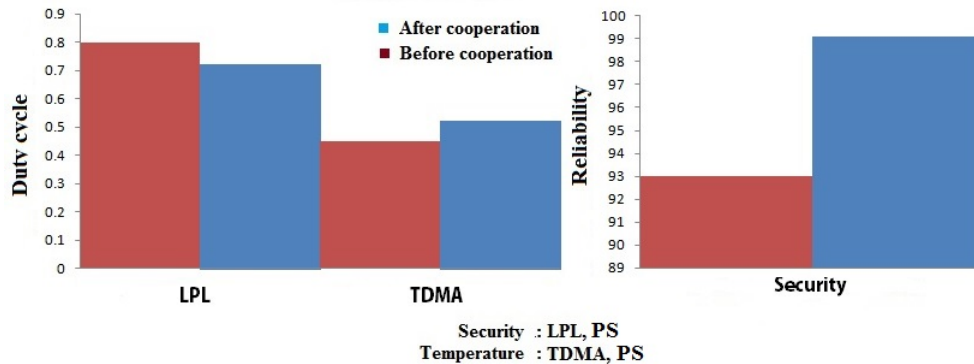


Figure 7: Comparison of the network performances before and after the cooperation is applied.

ing network. This change is considered drastic, since the corresponding value never surpassed 60% during the experiments. Figure 9 shows the obtained results.

When $\tau < 1$, the reasoning engine manages to maintain the optimal service set for more than 95 percent of time. For the same values of τ , it takes approximately 16 exploitation episodes to completely re-adjust its decision making policy and start enforcing a newly determined highest performing service set. Results are significantly worse, in both categories, when τ is set to values higher than 2. As such, we recommend to use low values for τ as long as the current set of services fulfils the application requirements, but to increase this value whenever the application requirements are no longer met.

7.2. Multihop network scenario

To create a multihop network, the nodes' transmission power was reduced, resulting in a change of the average number of hops to around 1.45 in the temperature monitoring network and 2.1 in the security network. As with the previous tests, the duration of the learning episodes was set to 5 minutes and the packet generation rate was set to 6 packets per minute. Figure 10 shows the values recorded for the networks' duty cycle and reliability while these networks cause interference to one another.

7.2.1. Discussion on the exploration phase of the algorithm

Figure 11 illustrates the outcome of the algorithm's exploration phase. Numerical values of all the relevant network metrics, over the entire explo-

ration phase, are given on Figure 12.

In the case of multihop networks, the exploration phase revealed the following:

- PACKET SHARING has a more significant influence on the overall network performance than in the single-hop network scenario. This is clearly visible on figure 11, for the service combinations marked from 16 to 31. This behaviour is expected in a multihop network scenario, since the PACKET SHARING service allows the path to the sink to be considerable shortened. (see Figure 12)
- Enabling the PACKET SHARING in combination with the LPL-MAC protocol, results in a significant decrease of the end-to-end delay. This is clear when the performance for service combinations 19, 23 and 27 are regarded. In network setups where a long network lifetime is not a priority, these states would have a higher priority.
- Having an additional performance factor - PACKET SHARING, results in a more obvious difference between the highest regarded service combinations (compare graphs 5 and 11). Similar outcome can be expected after adding additional basis functions

Figure 13 shows the performance improvement of using our cooperation methodology in a multihop network. Except for a slight increase of the duty cycle in both networks, a drastic improvement is recorded for the other relevant metrics compared to results obtained prior to cooperation. The average number of hops is reduced in both networks (20 percent in the security network, 5 percent in the temperature monitoring network), which should result in a lower end-to-end delay. The reduced hop count also increased the reliability of the security network by almost 25 percent. It should be noted that the duty cycle of the network depends on the specific set of virtual gateway nodes used. The duty cycle of the networks may therefore be further reduced by using an appropriate gateway selection mechanism. This is however out of the scope of this work.

7.2.2. Discussion on the exploitation phase of the algorithm

The capabilities of the algorithm to react to network changes was evaluated, similar to Section 7.1.2. Similar conclusion can be made and as such these results are omitted.

8. Future work

Future work will mainly focus on finding ways to improve the algorithm's efficiency during both phases.

Searching through a problem space can be faster with a help of prediction techniques. This would enable a reasoning engine to predict the performance of several service combinations without actually investigating them. In the use case presented in this paper for example, this would allow the reasoning engine to discard states that involve a combination of LPL and CSMA MAC protocols, after observing just a couple of them. This would reduce the number of learning episodes to almost 1/4 of the entire space.

Similar techniques can be included in the SOFTMAX approach. Once a network disturbance is detected, the engine should be able to detect the cause and at least try to approximate its influence on other service combinations, without actually investigating them. The reasoning engine should, for instance, be able to detect drastic changes in the duty cycle for the TDMA MAC protocol, approximate the effect of this change on the reward for other configurations using the TDMA protocol and adjust the decision making policy accordingly.

Research will also expand into other directions such as automatically optimising the performance of a single network based on a set of configurable parameters. Similar applications can be found in the literature. Our future work will therefore serve as an extension to ongoing research, which is expected to yield some new ways of application.

9. Conclusions

Due to the increasing number of network protocols and services for wireless sensor networks, developers have to make an optimal selection in terms of preferred configuration of the network. However, as this paper has shown, choosing the optimal set of protocols and services is not straightforward. Our research proposes a service-wise protocol optimisation technique for multi-objective, co-located and complex heterogeneous networks. Network services, provided by each sub-net, are used as arguments in the cooperation process. Results show that our reasoning engine is capable of discovering service combinations that improve overall performance for all the networks participating in the cooperation. Diverging high-level objectives and network capabilities are taken into account during the process. The efficiency of the algorithm is

shown in both single-hop and multihop network scenarios. The results are encouraging, especially for a multihop network scenario. Our algorithm was able to discover a service configuration that improved an overall reliability up to 25 percent, by accepting a small increase in both networks duty cycles.

To cope with dynamic environments, a heterogeneous network requires continuous monitoring. Our reasoning engine utilises the SOFTMAX algorithm in order to notice performance fluctuations and adapt to it. During the exploitation phase of the optimisation process, our implementation balances between maintaining the optimal service set and probing sub-optimal states in order to notice possible performance changes. Newly gathered information is used to update decision making rules. In the case of a network disturbance, the algorithm does not require a re-initiation. With the proper choice of a single SOFTMAX argument, (temperature factor - τ), the algorithm will efficiently reshape a decision making policy. By manipulating the same argument, the algorithm's ability to maintain an optimal service set can be increased to an arbitrarily high level.

The authors strongly believe that the problem of interfering, co-located networks will only increase. As such, innovative cross-layer and cross-network solutions that take these interactions into account will be of a great importance to a successful development of efficient, next-generation networks in heterogeneous environments.

10. Acknowledgements

This research is funded by the FWO-Flanders through a FWO post-doctoral research grant for Eli De Poorter and through an Aspirant grant for Daniel van den Akker

11. References

- [1] Wakamiya, N.; Arakawa, S.; Murata, M., "Self-Organization Based Network Architecture for New Generation Networks", 2009 First International Conference on Emerging Network Intelligence, pp.61-68, 11-16 Oct. 2009
- [2] E. De Poorter, B. Latre, I. Moerman and P. Demeester, "Symbiotic networks: Towards a new level of cooperation between wireless networks", Published in Special Issue of the Wireless Personal Communications Journal, Springer Netherlands, 45(4):479-495, June 2008

- [3] J.M. Lanza-Gutierrez, J.A. Gomez-Pulido, M.A. Vega-Rodriguez, J.M. Sanchez-Perez, "Multi-objective evolutionary algorithms for energy-efficiency in heterogeneous wireless sensor networks", SAS 2012 : IEEE Sensors Applications Symposium, Feb 7, 2012 - Feb 9, 2012, Brescia, Italy
- [4] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap and T Meyarivan, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, Parallel Problem Solving from Nature PPSN VI, 2000.
- [5] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, EUROGEN 2001.
- [6] Suat zdemir Baraa A. Attea nder A. Khalil, "Multi-Objective Evolutionary Algorithm Based on Decomposition for Energy Efcient Coverage in Wireless Sensor Networks",
- [7] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, "Evolutionary algorithms for solving multi-objective problems", (2nd ed.). Berlin: Springer, (2007).
- [8] V. C. Flvio Martins, G. Eduardo Carrano, Elizabeth F. Wanner, H. Ricardo, C. Takahashi, Geraldo R. Mateus, "A Hybrid Multi-objective Evolutionary Approach for Improving the Performance of Wireless Sensor Networks", IEEE SENSORS JOURNAL, VOL. 11, NO. 3, MARCH 2011
- [9] http://www.me.utexas.edu/~bard/LP/LP_20Handouts/CPLEX_20Tutorial_20Handout.pdf
- [10] P. Wang, T. Wang, "Adaptive Routing for Sensor Networks using Reinforcement Learning", CIT '06 Proceedings of the Sixth IEEE International Conference on Computer and Information Technology, October 22-25, 2006 Charlotte Convention Center Charlotte, NC
- [11] Z. Ye and A. A. Abouzeid, "Layered Sequential Decision Policies for Cross-layer Design of Multihop Wireless Networks", Information Theory and Applications Workshop (ITA'10), San Diego, CA, Feb 2010.
- [12] Minsoo Lee, Dan Marconett, Xiaohui Ye, S. Yoo , "Cognitive Network Management with Reinforcement Learning for Wireless Mesh Networks",

- IP Operations and Management (2007), pp. 168-179, doi:10.1007/978-3-540-75853-2-15
- [13] Christopher J.C.H. Watkins, Peter Dayan, Technical Note "Q-Learning", Machine Learning, 8, 279-292 (1992)
 - [14] Ad hoc on-demand distance vector (AODV) routing. Networking group request for comments (rfc): 3561, <http://tools.ietf.org/html/rfc3561> (July2003)
 - [15] Dimitri P. Bertsekas, "Approximate Policy Iteration: A Survey and Some New Methods", Journal of Control Theory and Applications, MIT April 2010, Vol.9, pp.310-335, Report LIDS - 2833
 - [16] Theodore J. Perkins, Doina Precup, "A Convergent Form of Approximate Policy Iteration", Advance in neural Information Processing Systems 15, NIPS 2002, Decembre 9-14, Vancouver, British Columbia, Canada
 - [17] url: www.eecs.harvard.edu/tmote-sky-datasheet.pdf
 - [18] D.D. Falconer, F. Adachi, B. Gudmundson, "Time division multiple access methods for wireless personal communications", IEEE Communications Magazine. 02/1995; DOI:10.1109/35.339881
 - [19] R. Jurdak, P. Baldi, C.V. Lopes, "Adaptive Low Power Listening for Wireless Sensor Networks", Mobile Computing, IEEE Transactions on (Volume:6 , Issue: 8), Aug. 2007, 10.1109/TMC.2007.1037
 - [20] L.Kleinrock and F. A. Tobagi, "Packet switching in radio channels: carrier sense multiple-access modes and their throughput-delay characteristics", IEEE Transactions on Communications, vol.23, pp.1400-1416, 1975.
 - [21] Eli De Poorter, Stefan Bouckaert, Ingrid Moerman, Piet Demeester, "Non-intrusive aggregation in wireless sensor networks", Ad Hoc Networks, Volume 9, Issue 3, May 2011, Pages 324-340
 - [22] Richard S. Sutton, Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT Press, Cambridge, MA, 1998, A Bradford Book
 - [23] L. P. Kaelblign, M. L. Littman, A. W. Moore, "Reinforcement learning: A Survey", Journal of Artificial Intelligence Research 4 (1996) 237-285

- [24] T. G. Dietterich, and O. Langley, (2007) “Machine Learning for Cognitive Networks: Technology Assessment and Research Challenges in Cognitive Networks: Towards Self Aware Networks”, John Wiley and Sons, Ltd, Chichester, UK. doi: 10.1002/9780470515143.ch5
- [25] M. Lagoudakis and R. Parr. “Model-free least-squares policy iteration”. In Proc. of NIPS, 2001.
- [26] Michail G. Lagoudakis and Ronald Parr, ”Least-Squares Policy Iteration”, Journal of Machine Learning Research, 4, 2003, pp. 1107-1149.
- [27] Milos Rovcanin, Eli De Poorter, Ingrid Moerman, Piet Demeester, ”A reinforcement learning based solution for cognitive network cooperation between co-located, heterogeneous wireless sensor networks”, ADHoc journal
- [28] Eli De Poorter, Evy Troubleyn, Ingrid Moerman, Piet Demeester, “IDRA: A flexible system architecture for next generation wireless sensor networks”, Wireless Networks, 2011, Vol. 17 (6), pp. 1423-1440
- [29] Daniel van den Akker, Chris Blondia, ”Virtual gateways: enabling connectivity between MAC heterogeneous sensor networks”, International Journal of Sensor Networks, Vol. 14, No. 3, pp. 133-143, Inderscience, 2013.
- [30] Lieven Tytgat, Bart Jooris, Pieter De Mil, Benot Latr, Ingrid Moerman and Piet Demeester, ”UGentWiLab, a real-life wireless sensor testbed with environment emulation”, 6th European conference on Wireless Sensor Networks (EWSN 2009), url: <https://biblio.ugent.be/publication/676545>
- [31] Daniel van den Akker, Chris Blondia, ”On the Effects of Interference between Heterogeneous Sensor Network MAC Protocols”, IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (IEEE MASS), Vol. 0, pp. 560-569, IEEE Computer Society, 2011

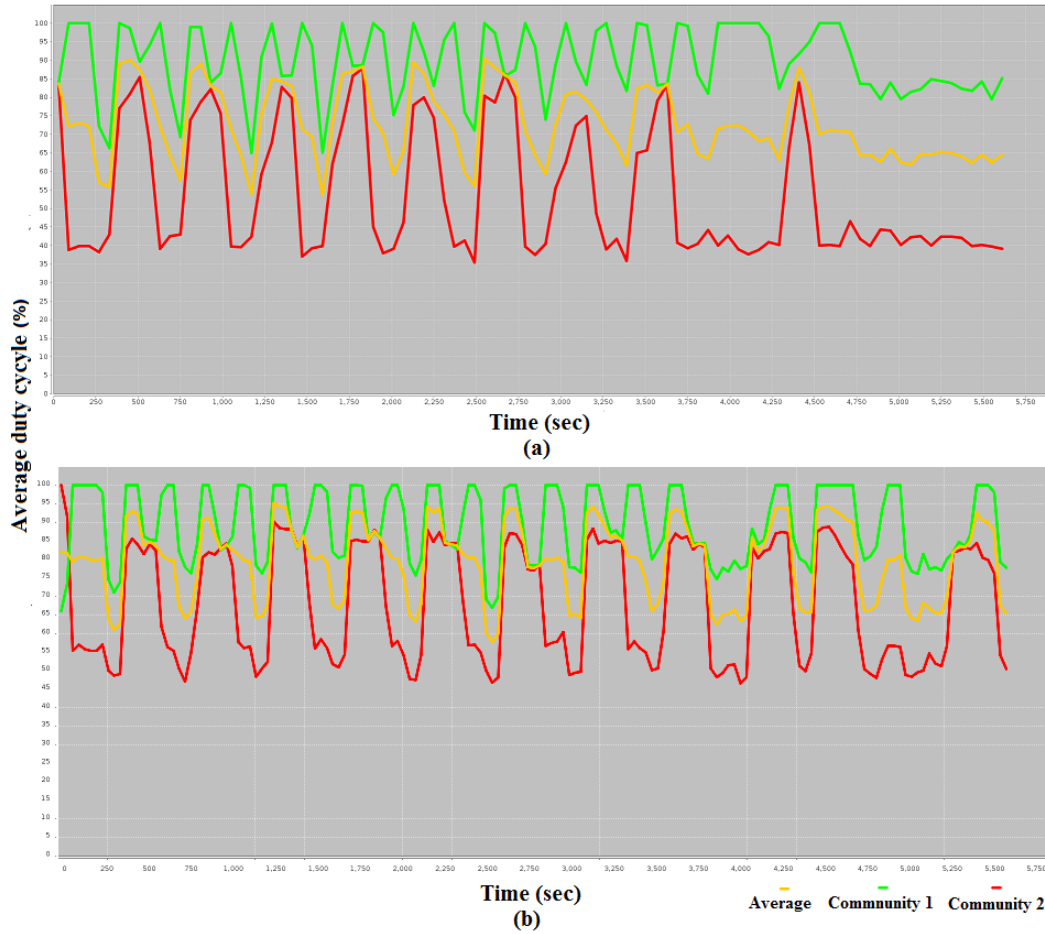


Figure 8: Behaviour of the algorithm, in terms of the networks' duty cycles, during the exploitation phase for: (a) $\tau < 1$ and (b) $\tau > 2$

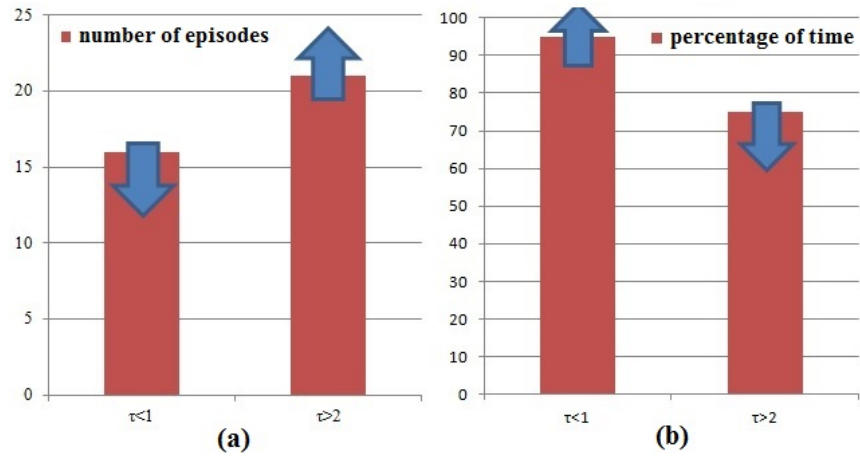
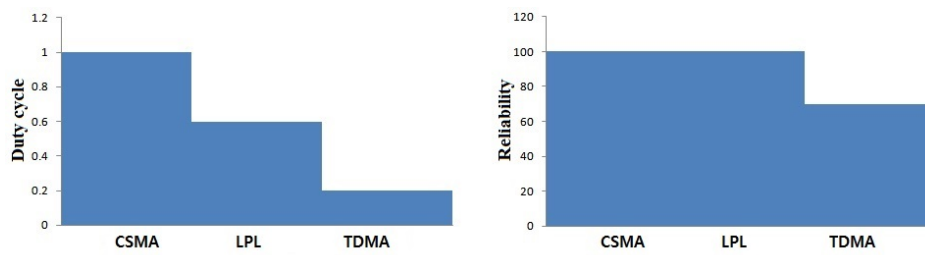
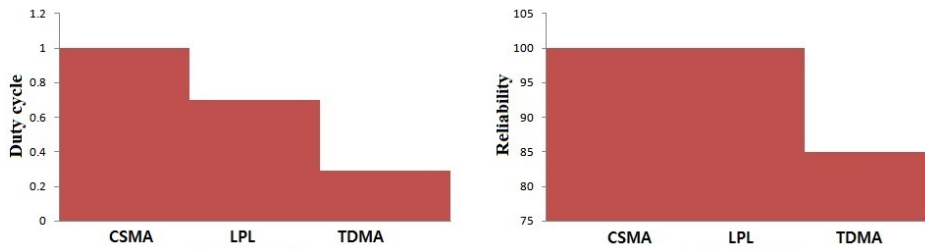


Figure 9: Illustration of the algorithm’s ability to: (a) retain the optimal service combination, in terms of the percentage of time the network was optimally configured and (b) adjust to a network conditions change, in terms of number of episodes needed to fully reshape the decision making policy. Arrows show a tendency with which statistics are changed when (a) τ decreases below 1 or (b) τ increases above 2



(a) Security network



(b) Temperature monitoring network

Figure 10: Values for the performance metrics: duty cycle and reliability, for (a) Security network and (b) Temperature monitoring network, tested while using different MAC protocols

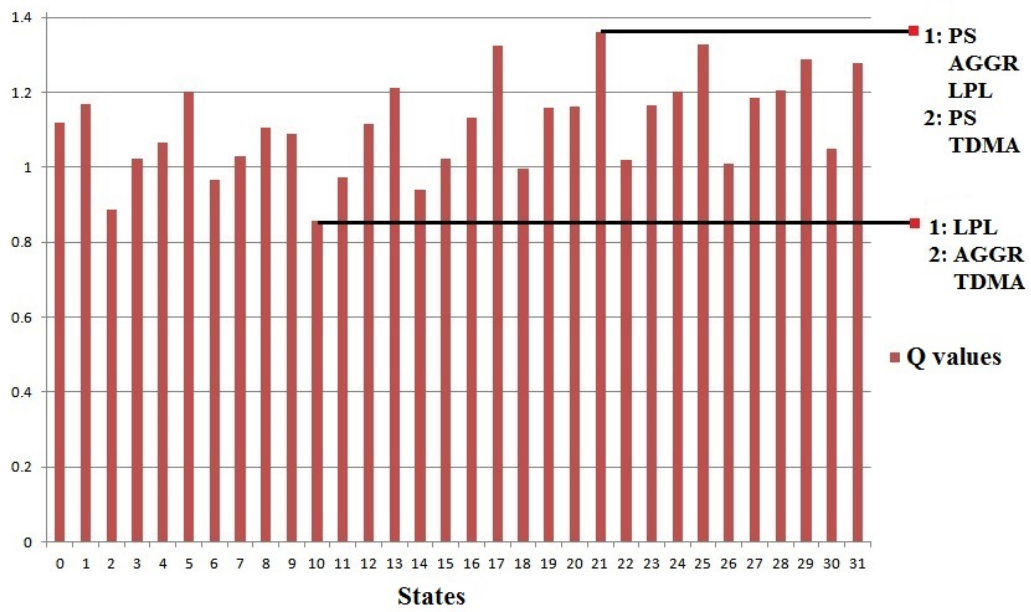


Figure 11: Values for the performance metrics: duty cycle and reliability for: (a) Security network and (b) Temperature monitoring network, tested while using different MAC protocols

State (dec)	State (bin)	PS	Aggr2	Aggr1	MAC2	MAC1	Qvalue	Duty Cycle2	Duty Cycle1	Hops2	Hops1	Reliability
0	00000	-	-	-	TDMA	CSMA_CA	1.119	0.6	1	1.55	2	88
1	00001	-	-	-	TDMA	LPL	1.168	0.61	0.78	1.45	2.1	87
2	00010	-	-	-	LPL	CSMA_CA	0.888	0.93	1	1.33	2	63.5
3	00011	-	-	-	LPL	LPL	1.023	0.89	0.89	1.33	1.9	91.2%
4	00100	-	-	AGGR	TDMA	CSMA_CA	1.065	0.68	1	1.55	2	90.4
5	00101	-	-	AGGR	TDMA	LPL	1.203	0.54	0.79	1.54	1.9	87.5%
6	00110	-	-	AGGR	LPL	CSMA_CA	0.965	0.85	1	1.44	2	84
7	00111	-	-	AGGR	LPL	LPL	1.029	0.87	0.81	1.44	1.9	94
8	01000	-	AGGR	-	TDMA	CSMA_CA	1.107	0.61	1	1.625	2	96.6
9	01001	-	AGGR	-	TDMA	LPL	1.090	0.62	0.85	1.55	2	70.6
10	01010	-	AGGR	-	LPL	CSMA_CA	0.858	0.86	1	1.44	2	55%
11	01011	-	AGGR	-	LPL	LPL	0.972	0.89	0.9	1.44	2	90.08
12	01100	-	AGGR	AGGR	TDMA	CSMA_CA	1.115	0.65	0.9	1.44	2	91.34
13	01101	-	AGGR	AGGR	TDMA	LPL	1.212	0.57	0.84	1.44	2	91.8
14	01110	-	AGGR	AGGR	LPL	CSMA_CA	0.939	0.90	0.99	1.44	2	83
15	01111	-	AGGR	AGGR	LPL	LPL	1.024	0.89	0.87	1.44	1.9	99
16	10000	PS	-	-	TDMA	CSMA_CA	1.131	0.69	1	1.44	1.8	99
17	10001	PS	-	-	TDMA	LPL	1.323	0.58	0.67	1.33	1.7	100
18	10010	PS	-	-	LPL	CSMA_CA	0.995	0.85	1	1.22	2.3	91.8
19	10011	PS	-	-	LPL	LPL	1.158	0.90	0.83	1.33	1.3	100
20	10100	PS	-	AGGR	TDMA	CSMA_CA	1.163	0.70	0.99	1.33	1.7	99
21	10101	PS	-	AGGR	TDMA	LPL	1.362	0.52	0.72	1.375	1.7	99
22	10110	PS	-	AGGR	LPL	CSMA_CA	1.018	0.90	1	1.33	1.9	99
23	10111	PS	-	AGGR	LPL	LPL	1.166	0.91	0.87	1.33	1.2	100
24	11000	PS	AGGR	-	TDMA	CSMA_CA	1.202	0.67	1	1.33	1.7	100
25	11001	PS	AGGR	-	TDMA	LPL	1.328	0.55	0.75	1.375	1.7	99
26	11010	PS	AGGR	-	LPL	CSMA_CA	1.110	0.91	1	1.33	1.9	100
27	11011	PS	AGGR	-	LPL	LPL	1.184	0.89	0.82	1.33	1.2	100
28	11100	PS	AGGR	AGGR	TDMA	CSMA_CA	1.204	0.67	0.99	1.33	1.7	99
29	11101	PS	AGGR	AGGR	TDMA	LPL	1.288	0.57	0.81	1.43	1.8	100
30	11110	PS	AGGR	AGGR	LPL	CSMA_CA	1.050	0.93	1	1.33	1.7	100
31	11111	PS	AGGR	AGGR	LPL	LPL	1.277	0.82	0.74	1.22	1.2	100

Figure 12: Average duty cycles, number of hops and reliability for both networks, recorded during the exploration phase

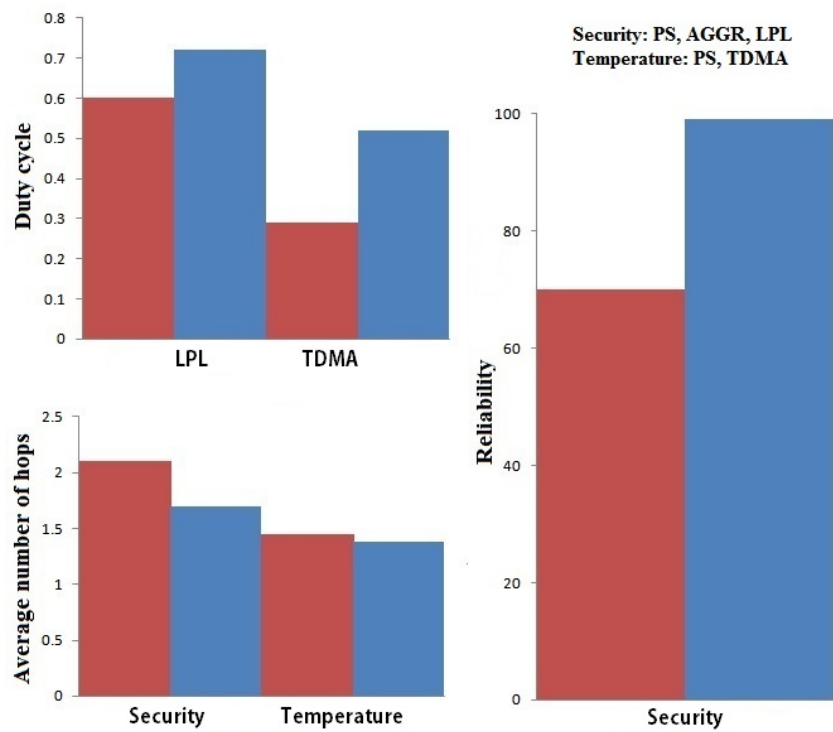


Figure 13: Comparison of the matching network parameters in a multihop network use cases, before and after the cooperation is applied