A reinforcement learning based, service wise optimization of cognitive, heterogeneous wireless sensor networks

Milos Rovcanin
IBCN
University of Ghent, Belgium
milos.rovcanin@intec.ugent.be

Abstract

We are witnessing the creation of the Internet of Things, dictated by the continuous increase in density of the wireless communicating devices. Different sub-nets typically ignore and very often harmfully interfere with one another. Having each individual sub-net configured to perform optimally in accordance to its predefined requirements and capabilities, a question is posed: could their performance be further increased through a process of inter-network cooperation? Network services, provided on different layers, can be used to bind the co-existing sub-nets into a symbiotic community. Symbiosis, in this context, will be manifested through a positive cross-network service influence. Therefore, the main objective of a self-managing optimizing technique would be to determine the optimal set of services for every participating network.

1 Introduction

The concept of inter-network cooperation aims at the interconnection of various every-day wireless objects, making them capable of sharing available resources and capabilities. Manual network configuration, for this purpose, is complex and inefficient [1], mainly due to an amount of co-located devices (time consuming) and the inability to follow the dynamics of the network. There is a need for solutions that efficiently support at run-time cooperation, that would both increase the network performance and simplify the setup for the end users.

Our research is motivated by the fact that a network's performance can be improved through a usage of certain combinations of services, provided by co-located networks (packet sharing, data aggregation, interference avoidance, MAC and routing protocols etc). To optimize co-located wireless networks, a wide range of such optimization techniques and services can be found in literature, ranging from interference avoidance, to dynamic power adaptations and shared routing and MAC solutions. Such a form of cooperation could improve individual network performances, using different combinations of configuration options than the one being pointed out as the optimal ones in a stand-alone case. Obviously, the major issue is to efficiently determine the optimal configuration parameters for all the participating networks, ensuring that the cooperation is mutually beneficial.

In [11], a described framework is capable of dynamically activating or deactivating a number of optimization techniques (referred to as network services). This work did not yet include any directions about when to activate the different network services. As such, a reinforcement learning paradigm was utilized in [**?**] to develop an engine capable of determining the best performing set of the utility services for each cooperating network, in regards to their specific performance requirements. In other words, the engine is able to determine the optimal, joint set of services for the entire network.

Services (negotiation arguments) can be spread over a several functional layers. For example, different types of medium access control (MAC) protocols (TDMA, LPL or CSMA-CA). Various routing protocols (transport layer) may be available in each network and thus also be used during the process. *Packet sharing*, *aggregation* and similar services can be introduced to the process at higher layers. Cooperation is manifested through the positive cross-network influences of these services. Each available service is a variable in this multi-variable optimization problem. The problem is also multi-objective: sub-nets generally have different purposes, thus different high level goals.

Our approach focuses around a centralized reasoning engine, used to calculate the optimal set of network services, activated in all the participating sub-nets. Instead of using mathematical methods for multi-variable, multi-objective function optimization, we are relying on techniques used inside the Least Squares Policy Iteration (LSPI), a reinforcement learning algorithm. Alterations it introduces to the common way of applying the reinforcement learning, along with some specific properties of our use case, make its implementation fairly straight forward. The approach provides tools to precisely assess the performance of the network and clearly identify the optimal state, with a relatively low processing demands. Above all, the reasoning engine stays versatile to adapt to any possible network condition changes.

SenSys'13, November 11–15, 2013, Rome, Italy.

We realize that the initial step of cooperation is the enabling of a two way communication between all the cooperating sides. This is a complex task that demands a number of prerequisites to be met: similar hardware configurations, compatible communication interfaces and transmission techniques. These issues were identified in the early stages of our research. A possible solution is discussed in [2]. The idea is further extended and elaborated in [3].

2 Related work

To our knowledge, there are no self-managing methods, developed to solve the process of identification the optimal service configuration in heterogeneous networks. Research published in [2], proposes a solution based on a linear programming methodology. IBM's CPLEX ILPSolver [4] is used to determine the optimal operational point. However, in order to produce useful results, it demands an input data which is rather difficult to obtain. For an illustration, the influences of each available service on each defined high level network goal, across the symbiotic network, has to be known apriori, which makes this solution usable only in a well controlled and non-volatile environments.

3 Reinforcement learning in the context of heterogeneous network optimization

Reinforcement learning approach [5] [6], models any given problem as a Markov Decision Process (MDP). In our use case, each system state is one distinct service combination. Obviously, the number of states directly depends on the overall number of services available in the symbiotic network. The reasoning engine passes through a discrete state space $S = s_1, s_2, s_3, ..., s_n$, by taking actions, $A = a_1, a_2, a_3, ..., a_n$, at each step.

Performing a certain action can have two possible outcomes:

- Transferring to another state
- Staying at the current state

Preserving a current state should happen only if the given state is determined to be the optimal one, at that point.

The following Bellman equation (1), provides a general method for evaluating every state-action pair. $Q(s,a)$ represents the state-action function, that numerically describes how "good" or "bad" a given decision was. To calculate it, one needs to know the immediate reward $r(s,a)$, given for taking an action a at the state s, as well as the future expected reward:

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

3.1 Least Squares Policy Iteration - LSPI

Idea behind the LSPI [7] algorithm is to calculate the Q function as a linear combination of *basis functions* (relevant environment features) and their respective weights:

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

Combining equations (1) and (2) yields the system of equations that can be presented in the matrix form in the following way:

$$\omega = A^{-1}b \quad (3)$$

$$A = \Phi^T(\Phi - \gamma P^{\pi}\Phi) \quad (4)$$

$$b = \Phi^T R \quad (5)$$

Matrix Φ contains basis functions related to the examined state-action pairs. P^{π} represents a transition probability matrix, describing the probability of a certain state/action pair sequence ((s,a), $(s',\pi(s'))$, while matrix R contains respective rewards. The ultimate result is the set of weights: ω.

In the vast state problem spaces, LSPI allows us to approximate the Q function by using only a portion of the information needed to populate the above given matrices. The necessary information is collected through the process of collecting samples from the environment: $D = (s_{d_i}, a_{d_i}, s'_{d_i}, r_{d_i} | i = 1,2,...,L)$.

In the offline approach, samples are obtained before the initiation of the learning process. The same sample set can be used to evaluate different decision making policies. With the online approach, that we adopted, they are continuously collected and the weight factors are recalculated after a certain number is collected (after each sample, in the extreme case).

4 Past Research

The RL solution has been tested on a rather simple use case, consisting of only 4 different service combinations - system states. The process was divided into two separate phases:

- *Exploration phase* - The reasoning engine uses the memoryless property of the underlying MDP to relatively quickly collect information regarding all the existing state/action pairs. Thanks to this property, Q values of N_{states} state/action pairs are updated at the end of a single exploration episode.
- *Exploitation phase* - The engine tries to balance, using the "ε greedy" algorithm, between keeping the optimal configuration setup active as much as possible and checking if the performance of the sub-optimal states is changed, which may happen as a result of a certain network disturbances that might happen over time.

The calculations were based on an already existing data, gathered during an earlier testing with the linear programming engine. This is justified by the fact that the case scenarios were identical [8] [2].

A limited space of this article prevents any detailed explanation. However, thorough discussions about results achieved during each phase of the algorithm are given in [8]. The important is the notice that the focus of this initial testing was mainly on the behavioral aspects of the algorithm, rather than the final result. Therefore, it should be considered as the proof of concept.

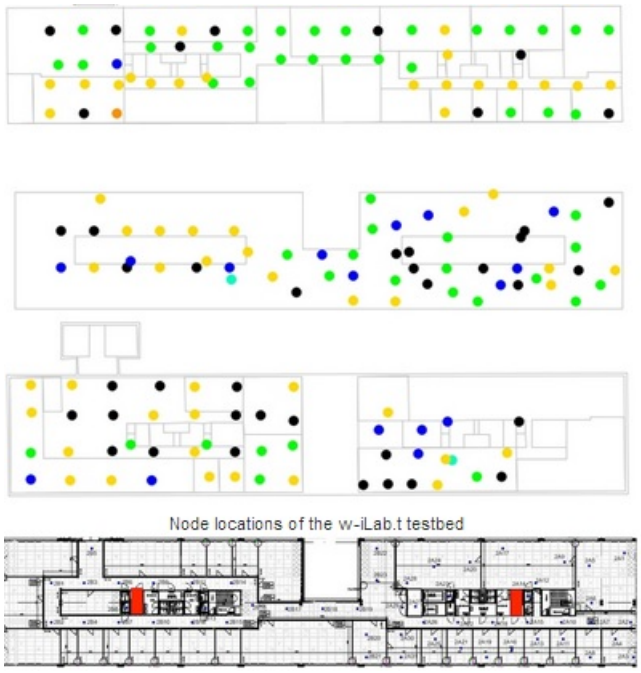

Figure 1. A three layered WiLab.t testbed, located at the iMinds building in Gent, Belgium

5 An ongoing research

The initial approach leaves a huge space for improvement, regarding several aspect of the algorithm. In much larger problem spaces, a scalability of the algorithm will pose a huge issue. The efficiency during the exploration phase will have to be increased. One initiative is to develop techniques that will "intelligently" reduce the number of system states during the exploration phase. Certain performance prediction procedures might be implemented in order to interpolate performances of some service combinations, without actually utilizing them. Some level of an a priori knowledge could help defining a number of possibly forbidden states (e.g. certain services cannot be active at the same time etc.).

The "ε greedy" algorithm results in a not so impressive performance during the exploitation phase. The major drawback is that it chooses equally (uniformly) among all actions, regardless of how "good" a certain action is. Currently, a SOFTMAX based solutions [10] are being investigated. It is expected that a new methodology will enable the reasoning engine to much faster detect any possible disturbances in the network performance.

6 Testing environment

Future solution will be tested in a real-life scenarios, using real-life measurements. WiLab.t [9] is a generic testbed for wireless (cognitive) networks. It consist of more than 200 TMoteSky and IBBT-Rmoni nodes, equipped with light, temperature, and humidity sensors, using a 802.15.4 radio communication. The nodes are deployed (see Figure 1) in three layers (office building floors):

The sensor nodes use a Tiny OS development environment, each running IDRA [11] - a protocol architecture that is designed specifically to cope with the resource-constrained devices. IDRA allows at run-time service selection, meaning that it can dynamically activate any given network network protocols from those available on each device. New network protocols can be added whenever required.

7 Conclusions

There is a strong believe that the problem of interfering co-located networks will only increase, as the number of co-located devices is increasing. As such, innovative cross-layer and cross-network solutions that take these interactions into account will be of a great importance to the successful development of efficient next-generation networks in heterogeneous environments. This paper refers to a research that adapts a reinforcement learning LSPI algorithm to optimize, service wise, multiple co-located networks, taking into account their divergent high level goals. The proposed algorithm aims at discovering the optimal set among the network services provided by every sub-net, participating in a cooperation. The main focus of the future research will be on efficiency of the techniques used for exploiting the obtained environmental data.

8 Acknowledgments

This research is funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders(IWTVlaanderen)through the IWT SymbioNets project, by iMinds through the QoCON project and by the FWO-Flanders through a FWO post-doctoral research grant for Eli De Poorter

9 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] D. Van Akker, "MultiMAC: A Multiple MAC network stack architecture for TinyOS",21st International Conference on Computer Communications and Networks, (ICCCN), JUL 30-AUG 02, 2012, Munich, GERMANY

[4] http://www.me.utexas.edu/ bard/LP/LP

[5] 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

[6] L. P. Kaelblign, M. L. Littman, A. W.Moore,"Reinforcement learning: A Survey", Journal of Artificial Intelligence Research 4 (1996) 237-285

[7] Michail G. Lagoudakis and Ronald Parr, "Least-Squares Policy Iteration, Journal of Machine Learning Research, 4, 2003, pp. 1107-1149.

[8] Milos Rovcanin, Eli De Poorter, Ingrin Moerman, Piet Demeester,"A reinforcement learning based solution for cognitive network cooperation between co-located, heterogeneous wireless sensor networks", ADHoc journal

[9] 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

[10] Richard S. Sutton, Andrew G. Barto, "Reinforcement Learning:An Introduction", MIT Press, Cambridge, MA, 1998, A Bradford Book

[11] E. De Poorter, I. Moerman and P. Demeester, "IDRA: a Flexible System Architecture for Next Generation Wireless Sensor Networks", Wireless Networks Journal, 17(6): 1423-1440, August 2011.