Energy Consumption Optimization in Radio Access Networks (ECO-RAN)*

Technical Report

Anders Mariegaard, Kim G. Larsen, Marco Muñiz, and Thomas Dyhre Nielsen

Department of Computer Science, Aalborg University, Denmark
{am,kgl,muniz,tdn}@cs.aau.dk

Abstract. In recent years, mobile network operators are showing interest in reducing energy consumption. Toward this goal, in cooperation with the Danish company 2Operate we have developed a stochastic simulation environment for mobile networks. Our simulator interacts with historical data from 2Operate and allow us to turn on and off network cells, replay traffic loads, etc. We have developed an optimization tool which is based on stochastic and distributed controllers computed by Uppaal. We have conducted experiments in our simulation tool. Experiments show that there is a potential to save up to 10% of energy. We observe that for larger networks, there exists a larger potential for saving energy. Our simulator and Uppaal controllers, have been constructed in accordance to the 2Operate data and infrastructure. However, a main difference is that current equipment do not support updating schedulers on hourly bases. Nevertheless, new equipment e.g. new Huawei equipment do support changing schedulers on hourly basis. Therefore, integrating our solution in the production server of 2Operate is possible. However, rigorous testing in the production system is required.

1 Introduction

In accordance with the enormous expansion of mobile networks in Denmark and the rest of the world, the number of mobile masts providing coverage has exploded, and with the upcoming expansion of 5G, there will be even more mobile radio devices that require power.

In recent years, it has been in the interest of the mobile operators to bring the power consumption, and the first steps have already been taken. These measures are based on semi-automatic procedures and with strong assumptions e.g. everyone follows the same patterns. A more fully automated approach to the problem, based on artificial intelligence, is desirable and expected to be able to further reduce power consumption. Furthermore, in connection with the sales activities, both in and outside Europe, it has been made clear that the mobile operators are increasingly concerned about mobile network power consumption, now and especially in the future. The background for this is that electricity consumption in the mobile network will increase significantly with the introduction

* In 2022 ECO-RAN wins Project of the Year at Energy Cluster Denmark.

of 5G and several 4G frequency layers. There is therefore both a considerable financial gain by minimizing power consumption, and also a growing interest in contributing to the Danish climate action, where the goal is for Denmark to be energy-renewable in 2050.

There are thus already good market leadership advantages for companies that can demonstrate that they are actively making an effort to achieve this goal. Together with one of our partners 2Operate we carried out feasibility studies that show that some Nokia-specific functions in Nokia's operating system can force the radio units to switch off at certain times – e.g. at 01:00-06:00. That is it is possible to synthesize and implement schedulers which turn on and off power cells.

Conservative calculations show that the savings potential will be €100 – €300 annually per mobile mast per company. In Denmark, they have fire mobile operators together approx. 10,000 locations with their equipment. This gives a total savings potential of DKK 1 - 3 million. euros annually or up to 50 million KWh. Worldwide, this potential will be many times higher.

2 Preliminaries

2.1 Mobile Networks

In this work we consider some geographical area where there is a number of *base stations*. Base stations have number of *cells* and every *cell* operate in some *frequencies*. The geographical location is discredited using *pixels*. Cells provide coverage to pixels and each pixel has a traffic demand. Figure 1 shows a map with the building of the Department of Computer Science at Aalborg University. Base stations are in pink, every base station contain some cells, and pixels correspond to the grid elements.

Frequency Layers Each base station usually consists of a number of cells broadcasting on different frequencies. Lower frequencies are for coverage while higher frequencies are for capacity. For 4G, the 800 MHz frequency layer is considered the coverage layer and must not be turned off in the current setup. There is room for optimization at the higher frequency layers as the needed capacity fluctuates a lot during a typical day. The 4G (LTE) frequencies are: E: 800 MHz, V: 900 MHz, T: 1800 MHz, A: 2100 MHz, L: 2600 MHz.

Power Saving The goal is to shut down capacity layers then they are not needed. e.g. during the night. A current contraint of the system is to maintain coverage to all pixels. In the current system, to ensure this constraint the 800 MHz layer can not be shut down.

Historical Data The company 2Solve has relevant historical data, e.g. the traffic demands for every base station. There is also information about the signal strengh for every pixel and for every frequency layer. Our simulations will be based on existing historical data.

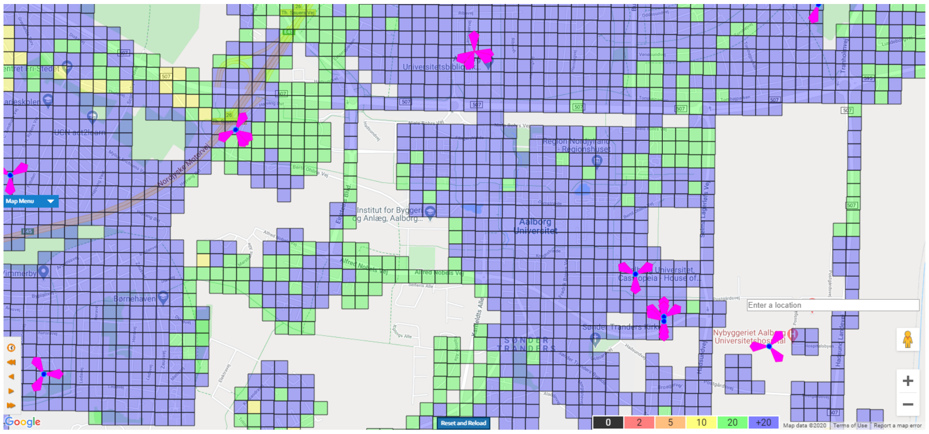

Fig. 1: Base station with three sectors. Sector 1 consists of one 800 cell and one 1800 cell.

3 Computing Near Optimal Strategies

3.1 Optimal Controlling

The main mathematical formalism for our modeling and posterior optimization is a stochastic hybrid game. For details we refer the reader to [4]. At a high level the game is between a controller and the environment. In our concrete scenario the stochastic game $\mathcal{G}$ corresponds to a communication network in which the environment consists of a number of cells $numCells$, pixels $numPixels$ and traffic demand per pixel. The controller consists of modes ON or OFF for every cell. Given a time horizon H e.g. of one day, a *control strategy* σ^H for horizon H, determines if a given cell is ON or OFF. The stochastic of the system come from the traffic demands, which can be represented as probability distributions over the pixels. Note that given a stochastic hybrid game $\mathcal{G}$ and control strategy σ^H, the game under the strategy $\mathcal{G}_{numCells} \upharpoonright \sigma^H$ is a stochastic process which induces a measure on the possible executions of the system.

Definition 1 (Optimal Controlling). *Given a stochastic hybrid game $\mathcal{G}_{numCells}$, synthesize a strategy σ^H which minimizes the expected reward*

$$\sigma^H = \mathsf{argmin}_\sigma \mathbb{E}^{\mathcal{G}_{numCells}}_{\sigma,H}(\mathit{reward})$$

where reward *accumulates the energy usage and a penalty for the lack of coverage*

$$\mathit{reward} = \int_0^H \mathit{penalty}(t) + \mathit{energy}(t)\ dt$$

with

$$\mathit{penalty}(t) = \sum_{i}^{numPixels} \mathit{penalty}(t,i)$$

$$\mathit{penalty}(t,i) = \begin{cases} 0 & \textit{if } \mathit{contribution}(t,i) - \mathit{demand}(t,i) > 0 \\ 1000 & \textit{otherwise} \end{cases}$$

and

$$\mathit{energy}(t) = \sum_{i}^{numCells} \mathit{energy}(t,i)$$

$$\mathit{energy}(t,i) = \begin{cases} 0 & \textit{if cell is off} \\ \textit{cell power + cost per mb} & \end{cases}$$

$\mathit{contribution}(t,i)$ *indicates the cells contribution to pixel* i *at time* t *similarly* $\mathit{demand}(t,i)$ *indicates the demand for pixel* i *at time* t.

In this project we aim to control real world communication networks with hundreds of cells and millions of pixels. Therefore computing the strategy σ^H is intractable. Instead we will compute near optimal-strategies using diverse techniques.

3.2 Online Strategy Synthesis

For this case study our goal is to compute a strategy (controller) σ^H to minimize energy consumption for a long horizon H. As the number of choices for the controller grows exponentially in the horizon, computing the strategy for a long horizon H is intractable. To overcome this problem we resort to the *Online Strategy Synthesis* [4] methodology, where we periodically compute a online strategy σ^h for a short horizon $h < H$. By composing the online strategies σ^h we can control the system for the horizon H. The composed strategy is less optimal than the optimal strategy σ^H but it can be computed effectively.

Figure 2 shows the online strategy synthesis approach for n *cells*, a horizon H of 1 day and controlling every 60 min. Short horizon h of 180 minutes. For n *cells* for the offline controllers σ^H and online controller σ^h there are 2^{720n} vs. 2^{3n} decisions. Thus computing a near-optimal online controller σ^h is clearly more applicable.

The methodology has successfully been applied to multiple case studies involving cyber-physical systems such as, intelligent traffic lights [3], floor heating systems [4], rerouting [2] etc.

3.3 Distributed Online Synthesis

In this project we aim to control large scenarios with hundreds of cells and millions of pixels. Therefore, directly applying online strategy synthesis is not

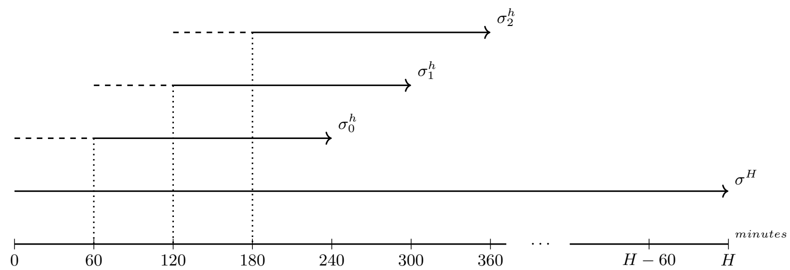

Fig. 2: Online Strategy Synthesis for n *cells*, a horizon H of 1 day and controlling every 60 min. Short horizon h of 180 minutes. For n *cells* for the offline controllers σ^H and online controller σ^h there are 2^{3n} vs. 2^{720n} decisions.

scalable. To overcome this difficulty, we apply Distributed Online Synthesis as in [4]. Given a geographical area with hundreds of cells, we partition it to sub areas which contain at most 8 cells. Then we can compute a online strategy for every partition and then merge the strategies to control the full network.

3.4 Methodology

The real world consists of a number of base stations, cells, pixels, frequency layers, etc. Where the goal is to provide a *controller* that powers ON or OFF cells to save energy. There exist a number of tools which can be used to simulate the behavior of mobile networks.

Figure 3 illustrates our methodology. The real world (or a simulation) require a control or a *strategy* σ^H for minimizing energy consumption for a long horizon H e.g. 3 months. Since the number of choices for the controller grow exponentially on the horizon H, computing a "global" strategy σ^H is not possible. Instead we periodically monitor the system and compute a near-optimal strategy

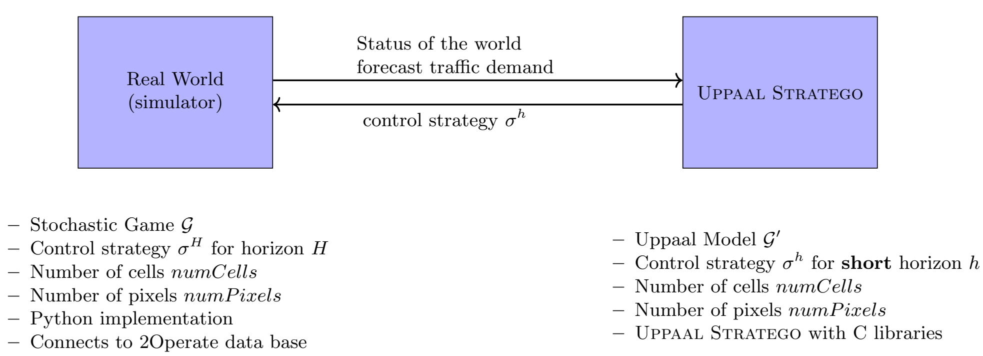

Fig. 3: System Architecture

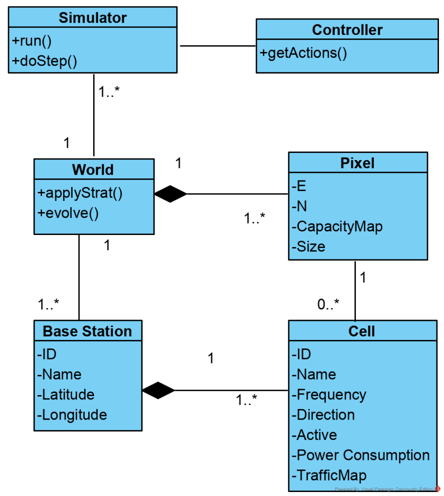

Fig. 4: Simulator architecture

σ^h for a short horizon e.g. 3 hours. In this work we will use UPPAAL STRATEGO to compute online strategies σ^h.

4 Experimental Evaluation

4.1 Simulation Tool

We use a simulation tool written in Python to replay and simulate historical data. From historical data we can observe the coverage contributions of every cell to every pixel. Then we can use this information to reproduce the effects of turning ON or OFF a given cell. In this way we can compute the values required by Definition 1. The traffic demand is based on historical data with the additional assumption that the demand is uniformly distributed over all pixels affected by cell. Figure 4 shows the overall architecture of the network simulator.

4.2 UPPAAL STRATEGO Controller

The main contribution of our work is to synthesize a near-optimal strategy in accordance to Definition 1. Toward this goal we use the tool UPPAAL STRATEGO [1]. The tool developed at Aalborg University and used to facilitate generation and optimization of strategies for abstract games with stochastic and real-time aspects. The tool uses simulation-based statistical machine learning methods.

Figure 5 shows the UPPAAL STRATEGO model for a stochastic hybrid game $\mathcal{G}_{numCells}$. Solid arrows correspond to controllable actions where as dashed arrows correspond to environment actions. At every simulation step and for every

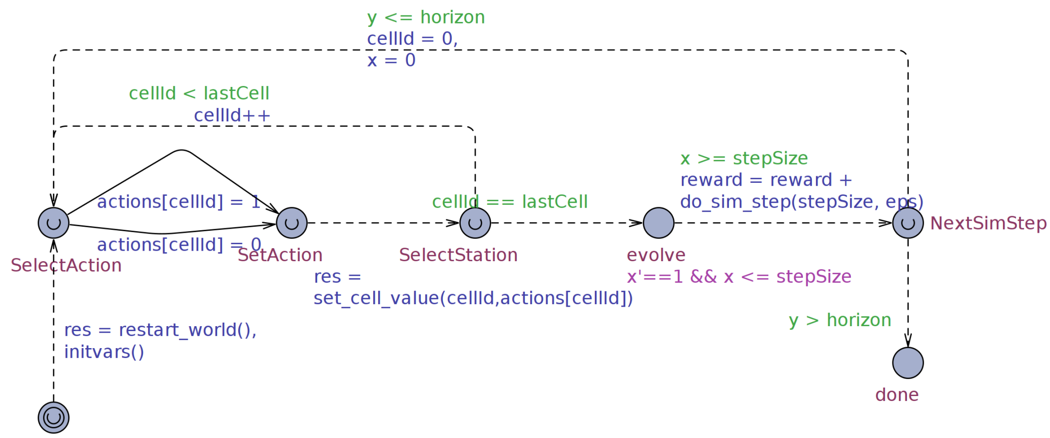

Fig. 5: Uppaal Stratego Controller

cell the controller has the choice to set ON or OFF the given cell, indicated by the command `actions[cellId]=1` or `actions[cellId]=0`. Once actions on cells have been chosen the environment executes its actions, this is done by calling a external C library with the command `do_sim_step(stepSize,eps)`. This function return a real value which is then accumulated in the variable `reward`. These steps are then executed until the short time horizon h has been reached. Uppaal Stratego will perform a number of simulations and used machine learning techniques to find the controllable actions which optimize the variable `reward` . Once the learning is complete Uppaal Stratego returns the near-optimal strategy which is then implemented in the simulator (or the real world).

4.3 Simulation Scenarios and Controllers

As a proof of concept we have chosen to perform a simulation of 1 day in the following two geographical locations in Aalborg, Denmark:

- City Syd with 39 cells and 2687 pixels
- Frydendal - Nørre Tranders with 107 cells and 6138 pixels

In our experiments we have used the following controllers:

- *ALLON* all cells always ON
- Uppaal Stratego as described in Definition 4.2

Table 1 shows the results of the different controllers in the different scenarios. The columns energy, penalty and reward correspond to Definition 1. The values on column energy are computing using a linear function on historical data and a constant for a cost per megabit.

Scenario	*ALLON*			UPPAAL STRATEGO[1]		
	Energy	Penalty	Reward M.	Energy	Penalty	Reward M.
City Syd	3473	0	150	3191	0	141
Frydendal Nørre Tranders	10115	0	436	9347	0	394

Table 1: Experimental results

We observe that while having no penalty, the UPPAAL STRATEGO controller is able to save about 10% energy on a single day. Concerning controller UPPAAL STRATEGO, the computation time for strategies about 22 and 31 hours for City Syd and Frydendal - Nørre Tranders on 16 cores. This means that given sufficient hardware resources, the scenarios could be controlled in real time. This is because using Online Strategy Synthesis (c.f. Section 3.2) will give a time window of up to 60 minutes to compute the next near optimal strategy.

5 Conclusion

Large mobile networks can profit from energy savings. This can be achieved by computing schedulers which turn off or on cells while maintaining some optimality criteria. In this work we have model a given portion of the mobile network as a stochastic game, applied different methodologies and finally used the tool UPPAAL STRATEGO to synthesize near optimal strategies which minimize energy consumption while maintaining coverage.

We have implemented a simulator which replays historical data. We have performed simulations for two large geographical areas in Aalborg, Denmark. Our initial results are encouraging, showing energy savings from up to 10% and showing the scalability of our approach.

Future Work Currently we have distributed controllers which do not communicate with each other. It would be interesting to study cooperative distributed controllers in this contexts. Our traffic demand model is quite abstract and could be refined if more is available. In particular a forecasting model for the traffic development could be of interest. In the same manner our optimization function is quite simple, one could consider to optimize different KPI's.

References

1. G. Behrmann, A. David, and K. G. Larsen. A tutorial on UPPAAL. In M. Bernardo and F. Corradini, editors, *SFM-RT 2004*, number 3185 in LNCS, pages 200–236. Springer, 2004.
2. A. Bilgram, E. Ernstsen, P. Greve, H. Lahrmann, K. G. Larsen, M. Muñiz, P. Taankvist, and T. Pedersen. Online and proactive vehicle rerouting with uppaal stratego. *Transportation Research Record*, 0(0):03611981211000348, 2021.

3. A. B. Eriksen, C. Huang, J. Kildebogaard, H. Lahrmann, K. G. Larsen, M. Muniz, and J. H. Taankvist. Uppaal stratego for intelligent traffic lights. In *Proceedings of the 12th ITS European Congress, Strasbourg, France, 19-22 June*, 2017.
4. K. G. Larsen, M. Mikucionis, M. Muñiz, J. Srba, and J. H. Taankvist. Online and compositional learning of controllers with application to floor heating. In M. Chechik and J. Raskin, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 244–259. Springer, 2016.