Coordinated group adaptation in sensor networks

Daniel Minder Pedro José Marrón Andreas Lachenmann Kurt Rothermel
Universität Stuttgart, IPVS, Germany
{minder|marron|lachenmann|rothermel}@ipvs.uni-stuttgart.de

ABSTRACT
In Wireless Sensor Networks, several algorithms are used to perform different functionality, e.g. routing or clock synchronization. Each algorithm is intended for specific network characteristics and user requirements. But the acutal characteristics and requirements may change during system runtime. TinyCubus and particularly its Tiny Data Managment Framework use adaptation to solve this problem.

In this paper, we first explain the centralized adaptation process. Then, we examine how this can be done localized in the network. Since coordination between local adaptation decisions is found to be necessary, metrics for this coordination and their dependencies are shown.

1. INTRODUCTION
Sensor Networks are used today in several domains to monitor real-world phenomena, e.g. in logistics, health care, biological studies or smart offices. Due to their small dimensions and their autonomous operation, sensor nodes can be installed in places where traditional monitoring is complicated or even impossible. Since the research is still young we expect further application areas in the future.

Despite the variety in domains, most applications share a common set of basic algorithms, e.g. routing, clustering or time synchronisation. However, each particular algorithm is suitable for a set of environments, e.g. mobile or static, where it exhibits best performance. When developing an application, it is a cumbersome task for the programmer to select the algorithms for a specific environment.

Moreover, the conditions of many systems change over time. Consider a logistics application where goods may be stored for several days but are moved to different places using various means of transportation. A single algorithm is unlikely to work in all settings. Therefore, the performance of the sensor network changes significantly over time. The user of the sensor network might change her requirements during runtime as well. For example, he could decide that a high delivery ratio is now more important than low latency. This has to change the behavior of the algorithms as well.

To cope with the changing environment and changing user requirements, we presented the TinyCubus system [1]. Its Tiny Data Management Framework (TDMF) is able to select appropriate algorithms or parameterizations for an algorithm based on network conditions, user requirements and algorithm characteristics.

In larger networks, a centralized adaptation has a high overhead since the network conditions have to be collected at a single node. Moreover, larger networks are likely to be diverse so that different parameterizations for different groups of nodes result in a better behavior than a global setting. Since adjacent groups influence each other, the separate adaptation results of the groups have to be coordinated.

In this paper, we start with a centralized adaptation approach and show how the adaptation can be localized in the network. Then, we examine how the local adaptation decisions can be coordinated to achieve better network performance.

The rest of the paper is organised as follows: In Section 2, the adaptation process of TDMF is shown. The locality of the adaptation is examined in Section 3. Section 4 then deals with the coordination of group adaptation results. The paper concludes with Section 5.

2. ADAPTATION PROCESS

2.1 Simulation
The basic assumption of TDMF is that the accuracy of simulation is sufficient to derive properties for real-world sensor networks. Each algorithm can thus be evaluated for different settings with respect to several performance parameters P. Some parameters, like power consumption, are general for all algorithms, others, like the delivery ratio for routing algorithms, are specific for one particular class. These performance parameters are the output of the simulation and are, therefore, measured. The simulation process can be controled by several input parameters that are divided into groups as well: Some parameters N influence the whole network, e.g. node density or mobility; some parameters A only the algorithms, e.g. the maximal number of retransmissions of a routing algorithm.

The user of TDMF has to decide which input parameter space is relevant for him. For example, in a network that remains static it is of no use to simulate mobile nodes. Depending on the available computing power for simulation, the granularity of the input parameters N and A is chosen. Especially for continuous parameters like network density, reasonable increments have to be set. In a second step, the parts of the first simulation step where the output parameters exhibit a high rate of change can be simulated in more detail.

Input parameters and measured output parameters describe the behavior of an algorithm and are, therefore, called its

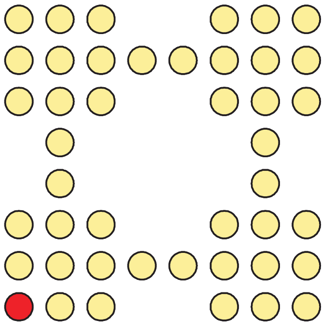

Figure 1: Scenario 1

meta-data. In TDMF, each algorithm module is annotated with this meta-data.

2.2 User preferences

The user of a sensor network has requirements that the application has to fulfil. They can be expressed as an inequation using the elements of P. For example, the user could specify that the estimated lifetime of the sensor network should be at least 1 year and the delivery ratio should be at least 80%.

Two interesting cases can happen: In the first case, more than one algorithm or more than one parameterization of an algorithm fits these requirements. In the second case, none is found. TDMF provides a way for resolving both. If more than one fits, the use can specify an optimization parameter and a direction. For example, the lifetime should be optimized in such a way that the network lives the longest. On the other hand, if no algorithm is found, the user lists the requirements which can be relaxed, e.g. the delivery ratio from 80% to 50%.

2.3 Adaptation process

Input and output parameters change in adaptation compared to the simulation. The network parameters N cannot be influenced by the user or the system but are given by the environment and must be measured. The performance parameters P are given by the user as described above. Thus, the adaptation systems selects the algorithms whose meta-data match N and the conditions over P and performs necessary optimization or relaxation to chose exactly one algorithm. Then, this algorithm is installed and parameterized with parameters A if needed.

2.4 Example

To test and evaluate the adaptation system, we used the Avrora simulator [3]. 44 nodes are arranged in a grid as shown in Figure 1 with 10m distance in horizontal and vertical direction. The base station is located in the lower left corner.

Each node sends a data packet every 10 seconds to this base station. A routing algorithm implementing the GEM metric [2] is used. Two parameters A can be adapted in this algorithm: the transmission power and the maximal number of retransmissions per packet. In simulation, 9 different transmission power levels and 5 different retransmission limits, thus 45 different combinations in total, are evaluated.

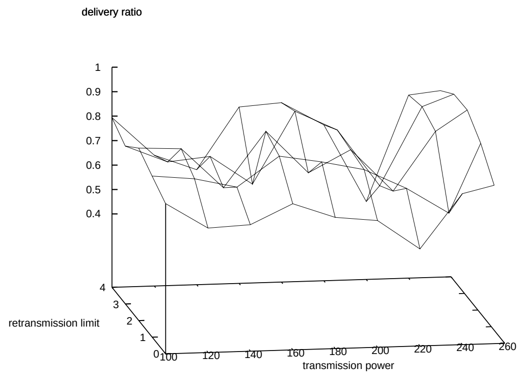

Figure 2: Parameter space for scenario 1

The performance parameters P the delivery ratio and the energy needed are measured. Figure 2 shows the delivery ratio for all 45 simulations.

A very simple adaptation goal could be to maximize the delivery ratio. The adaptation process would select the parameterization "transmission power = 255" and "retransmission limit = 3" which leads to a delivery ratio of 88% on average.

3. DECENTRALIZED ADAPTATION

3.1 Purely Local Adaptation

In the previous global and centralized adaptation approach, messages have to be sent through the network to collect the network parameters and to propagate the adaptation decision. Since the adaptation is often used to maximize the network lifetime, this adaptation message overhead has to be minimized. If the adaptation decision can be made purely local, there would be no message overhead.

We used the simulation results of the previous example and evaluated each of the 45 simulations per node, resulting in a single parameter curve for each node. Then, the adaptation process is executed as explained before on each node locally, resulting in different and independent parameter values for each node.

According to the simulation basis, the worst delivery ratio of a node should be 80%. But in fact, the average ratio for the whole adapted network was 60%, with 29% for the worst single node. The reason for this is that the simulation basis was created using the same parameterization for the complete network, but TDMF uses these values to adapt each node separately. Interferences between nodes were not taken into account.

To cover the interferences, all possible combinations would have to be simulated. This is not possible since it would lead to 45^{44} (or approximately $5.51 \cdot 10^{72}$) combinations in the example. Additionally, the adaptation process need to be done centralized again to assign a suitable parameter combination to all the nodes which contradicts the local approach.

3.2 Group Adaptation

A balance between global and local adaptation is the introduction of groups. Inside a group, the same parameterization is used. The behavior of a group is, therefore, expected to be more stable and to resemble the basic simulations that used the same parameters for the whole network. The disadvantage of this approach is that network parameters have to be collected and the adaptation decision has to be announced, but both happens in a smaller area than in the global case.

The example network is divided horizontally and vertically in the middle, thus forming 4 groups. The 45 basic simulations are evaluated per group, thus forming a new adaptation basis for the group adaptation. The adaptation process uses these per-group results to find optimal settings for each group.

An overall delivery ratio of 89% should be achieved according to the simulation basis. The resulting ratio of 71% is lower than expected, but the gap is lower than in the purely local adaptation.

3.3 Coordinated Groups

Most of the packet losses in the group adaptation example occur at group transitions. In the example, the transmission power of the group with the base station was set to 180, while all other groups use a transmission power of 255. The simulation shows that the packet loss is much higher at the two nodes connecting two groups if the transmission power differs between these groups.

Therefore, to achieve better delivery ratios the parameters of the groups have to be coordinated. Each adaptation process selects not only the best parameter setting for each group but the best k settings. We have chosen $k = 3$. A coordination metric calculates the distance of the transmission power settings for all possible combinations of the settings. Finally, the combination of parameters with the least distance is selected.

The coordination introduced in this approach revives a centralized step that was eliminated when moving from the global to the local view. The difference is that only a few parameters have to be joined at a single place and not all the network conditions from all the nodes.

Simulation shows that using a combination with the highest distance value has a low delivery ratio of 69% while the lowest distance values lead to high delivery ratios of 87%, which is very close to the predicted 89%. This dependency is also expressed by a Pearson's correlation coefficient of -0.78. This coefficient measures the mutual dependency of two variables and ranges from -1 to 1. A value of -1 or 1 show that there is a linear relationship, a value of 0 indicates that a linear model is inappropriate.

4. GENERAL METRICS

Not all scenarios allow to find a simple coordination metric. Moreover, a coordination metric suitable for one scenario might be useless in another. We, therefore, introduce a second scenario with a larger network and change the grouping and parts of the algorithm.

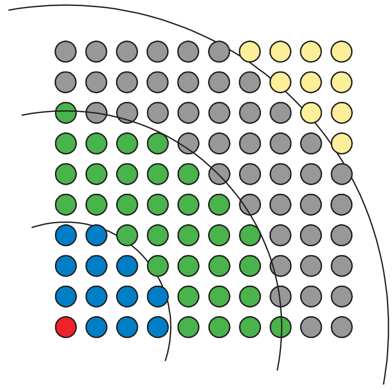

Figure 3: Scenario 2 with circular groups

4.1 Topology

In Scenario 2, 100 nodes are placed on a regular grid with 10m distance in either direction forming a square of 10x10 nodes. 4 groups of 25 nodes each are formed by dividing the network horizontally and vertically in the middle. The base station is still in the lower left corner. Again, the adaptation basis is determined by simulating all 45 combinations of transmission power and retransmission limit for the whole network and by evaluating these simulations per group. Then, the adaptation process selects the best 3 parameter settings for each group.

When applying the distance function of the first scenario as coordination metric to this scenario, "wrong" parameter combinations for the groups are selected. A correlation coefficient between distance metric and delivery ratio for all 81 combinations of 0.21 shows that the metric is less significant in this scenario, but indeed more distant transmission power levels are better than close ones. This contradicts the finding from the first scenario.

4.2 Different Grouping

The analysis of the previous experiment showed a problem in the center of the network where four different groups are adjacent. Such a constellation should be avoided by a clever grouping. Therefore, we examined another method with groups based on the distance from the base station. Figure 3 shows such a grouping with equidistant radii.

The 45 basic simulations of Scenario 2 were reevaluated using this new grouping and, again, the best 3 parameter settings for each group were determined. The simulation of all 81 combinations and the calculation of the correlation coefficient between delivery ratio and the distance metric show that, with a coefficient of -0.50, this grouping reacts similarly to differences in the transmission power as the first scenario.

4.3 Different Routing Metric

The groupwise adaptation of an algorithm is based on the assumption that the algorithm works inside a group independently from the surrounding groups. The previous examples show that this is not the case. Especially intelligent routing metrics react very sensitive to changing traffic due to changed neighboring groups. Thus, these metrics may counteract the adaptation goal.

Therefore, a scenario using static routing trees was set up. The trees were built before simulation by taking into account the radio model characteristics and by trying to balance the number of child nodes in each tree node. That way, a group shows similar behavior also with different adjacent groups.

After simulating the 81 combinations of the best 3 settings for each group based on 45 new basic simulations, the correlation coefficients were calculated. With quadratic groups, the correlation is -0.33, but 0.44 when using circular groups. Compared to the simulations that used the GEM routing metric, the static trees react diametrically opposed to differences in the transmission power.

4.4 Coordination Metrics

As the previous sections have shown, the simple distance metric developed for the example scenario of Section 2 does not work in all cases of the enlarged second scenario. Therefore, a different metric for this scenario has to be found. Since constructive approaches were not successful, possible metrics were tested systematically.

A metric is an addition of single components that are based on the algorithm parameters A. In the given scenario, the tranmission power or retransmission limit of a single group or the "distance" of the transmission power or the retransmission limit between two groups are used. A component is built using a parameter mapped to the interval $[0, 1]$. The *value* resulting from the mapping is fed into the metric. Each component can also be used as $1 - value$ in the metric.

Having 4 groups, this leads to 40 different single components that could be part of a metric. Of course, some combinations make no sense to be used at the same time, e.g. *value* and $1 - value$. Moreover, a metric consisting of too many single components is likely to be useful for a specific setting only. Therefore, we limited the search to 6 single components.

To evaluate each metric, 9 different settings were used: 5 using the GEM metric and 4 using static trees. The settings with GEM metric are further divided into 3 using quadratic groups and 2 using circular groups, the settings with static trees are divided into 2 settings of each grouping. The single settings differ again in further adaptation restrictions like energy.

For each metric and each setting, Spearman's rank correlation coefficient between the metric and the average delivery ratio of the network was calculated. The rank correlation coefficient does not assume a linear relationship between the variables and is, therefore, more suitable, for the abstract metric. The metrics were finally sorted by the average correlation coefficient, thus selecting metrics that pedict well the quality of a parameter combination.

As Figure 4 shows, a good metric covering all 9 settings could be found, exhibiting an average correlation coefficient of 0.30, but ranging up to 0.65 for a single setting. When calculating the best metrics for a subset of the settings that only used (1) static trees, (2) GEM routing metric, (3) quadratic groups or (4) circular groups, even better metrics could be found. As expected, the static trees behave better under adaptation than the algorithm using the GEM routing metric, even using the best coordination metric found. Also, circular groups can be better predicted than quadratic groups due to the "center problem" of the latter.

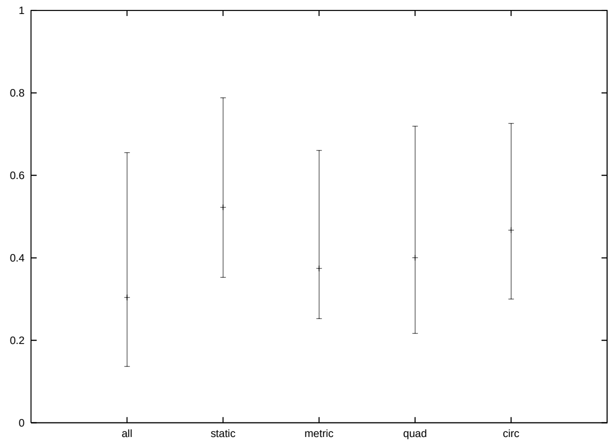

Figure 4: Correlation coefficients of best metric

5. CONCLUSION AND FUTURE WORK

Pure local adaptation of neighoring groups of nodes or even adaptation of single nodes in a sensor network can deteriorate network performance. Coordination of local adaptation decisions can improve the overall adaptation result although this implies additional overhead. The quality of the coordination is heavily dependent on the coordination metric. We have shown that such a metric can be found and that it can be improved further when restricting the domain.

As next steps, we will examine the metrics found in more detail to explain why each single component is part of a metric. Using this knowledge, it might be possible to develop a constructive method to build a metric.

When simulating the algorithms, no network or application characteristics have been changed. In the future, different traffic loads and node densities have to be included in the simulation space to cover the characteristics of the algorithm more completely.

6. REFERENCES

[1] P. J. Marrón, A. Lachenmann, D. Minder, J. Hähner, R. Sauter, and K. Rothermel. TinyCubus: A flexible and adaptive framework for sensor networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, pages 278–289, January 2005.

[2] O. Saukh, P. J. Marrón, A. Lachenmann, M. Gauger, D. Minder, and K. Rothermel. Generic routing metric and policies for WSNs. In *Proc. of the Third European Workshop on Wireless Sensor Networks*, pages 99–114, 2006.

[3] B. Titzer, D. Lee, and J. Palsberg. Avrora: Scalable sensor network simulation with precise timing. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks*, pages 477–482, 2005.