Self-Organized Event Detection in Sensor Networks using Bio-inspired Promoters and Inhibitors

Falko Dressler
Autonomic Networking Group,
Dept. of Computer Science 7, University of Erlangen, Germany
dressler@informatik.uni-erlangen.de

ABSTRACT

Sensor and Actor Networks (SANETs) represent a specific category of massively distributed systems in which coordination and control of the participating networked nodes is especially challenging. Recently, a number of self-organization methods have been published that focus on network-centric operation in such networks. Rule-based Sensor Network (RSN) is a programming approach that supports this kind of operation. It mainly features inherent support for heterogeneous nodes. Until now, the rule execution in RSN is too static for application in highly dynamic environments such as event detection of mobile targets. We present a bio-inspired approach for adaptation of the local rule execution, which is based on an promoter / inhibitor scheme. The application of this biological technique leads to improved reactivity and resource utilization. The advantages are demonstrated based on a comprehensive simulation study.

Keywords

Sensor and actor networks, promoter, inhibitors, network-centric operation, rule-based sensor network, bio-inspired networking

1. INTRODUCTION

In the last couple of years, Sensor and Actor Networks (SANETs) have become one of the main research domains in the networking community. Basically, SANETs represent a specific class of Wireless Sensor Networks (WSNs). Besides the requirements and challenges in terms of energy efficiency and the capability to work on low-resource embedded systems, additional coordination is required. The system-inherent actuation facilities need to be controlled, i.e. activated and driven, by network-inherent sensor measures. This leads to new challenges such as critical real-time operation requirements [2].

In general, the coordination and control of SANETs is still an emerging research area. Most recent approaches focus on network-centric operation as the key paradigm to handle the mentioned challenges. Self-organization is considered the final solution to build energy efficient SANETs [6]. Compared to the classical centralized operation scheme in which sensor nodes are continuously analyzing the environment (measurement) and transmit the measurement data to one or more fixed base stations for further processing, the self-organized principle supports real-time operation without complex global state maintenance [10]. Figure 1 depicts a typical SANET. Several sensor nodes are shown that directly interact with associated, i.e. co-located actuators.

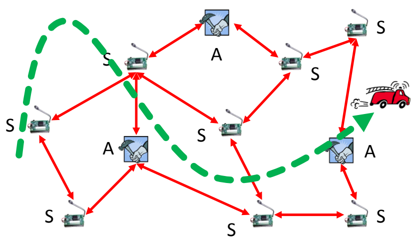

Figure 1: Self-organized network-centric operation in SANETs w/o the need of a centralized controller. The SANET is employed to observe a mobile entity and to (quickly) react on the collected information

Possible solutions can be found in approaches related to the main ideas of *autonomic networking*, i.e. the development of self-managing networks. One idea is to cluster the available sensor and actor systems into groups that enable simple coordination and control strategies. An example is the distributed coordination framework developed by Melodia et al. [17].

In the context of this paper, we concentrate on a specific application scenario that – while simplistic from a high-level point of view – shows a number of characteristic challenges for developing self-organizing SANETs. The scenario is the detection of mobile entities (events) in a given region of interest and the appropriate reaction of the network. The scenario is depicted in Figure 1. A number of sensors try to detect one or more mobile entities and to (quickly) react on the collected information by sending the observed events to one of the actors.

From this example, some specific challenges for developing SANETs can be derived. On the one hand, there are strict real-time requirements, i.e. the need for responsive (or reactive) SANET based event detection. On the other hand, other requirements such as energy efficient operation

Bionetics '08, November 25-28, 2008, Hyogo, Japan

still apply. Obviously, both requirements are not achievable at the same time, i.e. either very fast event detection can be performed to the price of high energy usage or vice versa. Especially in the context of SANETs, the energy performance of the individual nodes also characterizes the energy efficiency of the entire system, and thus, the possible *network lifetime* [4].

In this paper, we study some principles as known from biology to approach exactly these challenges, i.e. to enable the SANET to perform real-time measurements and event detection while reducing the necessary amount of energy. In particular, we employ a bio-inspired approach based on dynamic promoter and inhibitors for self-organized event detection. This approach has already successfully been applied to other problem domains [5, 18].

In order to achieve this goal, we created two separate feedback loops as inspired by similar solutions found in nature [9]. These feedback loops represent promoter / inhibitor functions, i.e. they either stimulate sensor nodes to monitor events, or they suppress this amplification effect if the event detection cannot be performed efficiently. The implementation of the system was done using the Rule-based Sensor Network (RSN) system [8], which allows a simple and heterogeneous programming of WSNs.

We developed an appropriate simulation model to analyze the behavior and performance of the studied approach. It turned out that the dynamic reconfiguration depending on the current network behavior is possible without any global control.

The rest of the paper is organized as follows. Section 2 introduces related approaches for sensor-actor coordination and outlines the basic concepts of RSN. Section 3 introduces the concepts of bio-inspired promoter / inhibitor based feedback loops. Section 4 outlines the ideas and improvements to RSN. The simulation model as well as the obtained results from the performance evaluation are depicted in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORK

In this section, we briefly outline related approaches for sensor-actor coordination that can be applied for completely self-organized operation of SANETs. We primarily focus on event detection schemes that are relevant for the envisioned application scenario. Additionally, we introduce the RSN programming system that builds the basis for the optimized event detection scheme.

2.1 Sensor-Actor Coordination

Most coordination techniques for WSNs and SANETs are based on clustering techniques. Besides simple probabilistic approaches, more sophisticated solutions also incorporate measures for the achieved connectivity in the network, e.g. as used in Span by Chen et al. [3], or even quality of service characteristics such as the reliability of a communication path as demonstrated by Melodia et al. [17].

Another approach is to group nodes according to the main objectives of the sensor network such as a given degree of coverage. Gupta et al. [11] have shown that queries into a sensor network can be optimized based on this measure. Higher level task allocation strategies are also related in the discussed context because actuation represents a specific class of remotely executed tasks. For example, Low et al. [15] employed autonomic networking techniques for task allocation in mobile sensor networks.

Akan et al. published an event detection mechanism for application in SANETs [1, 19], which provides efficient path selection between the monitoring nodes and the event sink. Similarly, the sensor-actor coordination approach by Melodia et al. [17] includes means for associating sensors to adjacent actor nodes.

More recently, new approaches for real-time communication [10] and real-time monitoring [16] have been studied that exploit the possible approximation of the communication reliability for more efficient data communication in SANETs.

2.2 Rule-based Sensor Network

Recently, we developed a rule-based programming system for supporting network-centric operation in heterogeneous SANETs that we named Rule-based Sensor Network (RSN) [8]. Basically, RSN is an architecture for data-centric message forwarding, aggregation, and processing. In earlier work, we proved that RSN explicitly outperforms other WSN protocols for distributed sensing and network-centric data pre-processing in two dimensions: (a) reactivity of the network, i.e. the response times for network-controlled actuation can be reduced, and (b) communication overhead, i.e. the bandwidth utilization on the wireless transmission channels was improved.

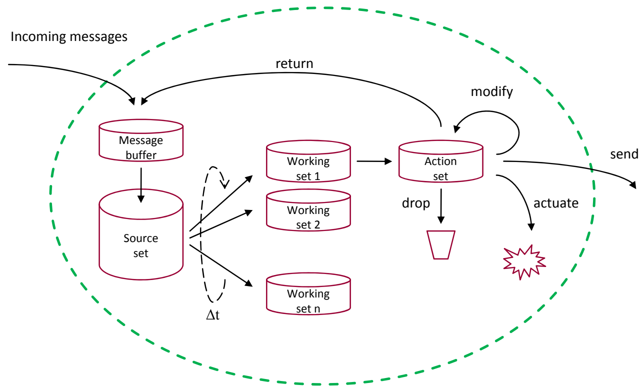

Figure 2: The working behavior of a single RSN node. Received messages are stored in a buffer, selected to a working set according to specific criteria, and finally processed, i.e. forwarded, dropped, etc.

The key objectives motivating the development of RSN were improved scalability and real-time support for operation in SANETs. RSN is based on the following three design objectives that enable the mentioned objectives:

- *Data-centric communication* – Each message carries all necessary information to allow data specific handling and processing without further knowledge, e.g. about the network topology.
- *Specific reaction on received data* – A rule-based programming scheme is used to describe specific actions to be taken after the reception of particular information fragments.
- *Simple local behavior control* – We do not intend to control the overall system but focus on the operation

of the individual node instead. Simple state machines have been designed, which control each node (being either sensor or actor).

Figure 2 depicts the working behavior of a single RSN node. After receiving a message, it is stored in a message buffer. The rule interpreter is started periodically (after a fixed Δt) or after the reception of a new message. An extensible and flexible rule system is used to evaluate received messages and to provide the basis for the node programming scheme. Thus, the local behavior is controlled by a rule interpreter in form of simple state machines. The interpreter is applying the installed rules to previously received messages.

Each rule that is used to process the received messages consisting of two parts, a condition and an action, as shown in Figure 3. Starting with this overview, we will continue to use the specific RSN syntax [8] to outline rules in the examples. The condition is intended to associate messages to a given rule, i.e. an action. In RSN, the specific reaction on received data is achieved by means of predicates. RSN is able to select all messages of a given type or messages with specific content attributes. All selected messages are stored in so called working sets.

```
if PREDICATE then {
  ACTION
}
```

Figure 3: Basic rule composition depicted in RSN syntax. Messages are selected by a predicate and processed by an action

The RSN architecture has been developed for data-centric communication. Thus, instead of carrying address information, each message is encoded using a (type, content) pair. The type describes the message and the attached content. The data itself will usually include a value and application specific meta information such as a geographical position or priority information. The message encoding and processing in RSN are similar to the ones suggested by directed diffusion [13]. Even though the communication scheme is completely different, directed diffusion and RSN both rely on the identification of messages according to representative type information. Each message could be encoded as follow:

```
M := { type, region, confidence, content }
```

In several experiments, the period of RSN execution Δt has been identified as a key parameter for controlling the reactivity vs. energy performance of the entire RSN based network. Basically, the duration of messages stored in the local node introduces an artificial per-hop delay. The optimal value for Δt affects the aggregation quality vs. real-time message processing.

Obviously, the period Δt is critical for particular applications such as data aggregation: the longer messages are stored before being processed, the better the possible aggregation ratio (more messages can be aggregated into a single one); and the longer the period, the longer the artificially introduced per hop delay.

In the selected scenario, i.e. monitoring of dynamic, mobile entities in a SANET, an optimized value of Δt can be exploited for optimized event detection. In the following, we describe a bio-inspired approach based on dynamic promoter and inhibitors for self-organized event detection. This mechanism has already been successfully applied to other problem domains [5, 18].

3. BIOLOGICAL INSPIRATION

In the last years, we studied some aspects of conceptual similar techniques that have been studied in the domain of cellular biology. These investigations lead to completely different communication and control paradigms in an area that is widely known as *bio-inspired networking*. A great number of solutions are thinkable based on bio-inspired approaches [7].

One of such biological mechanisms is the regulation of the arterial blood pressure in mammals based on the renin-angiotensin control. Figure 4 briefly introduces the mechanism. If the pressure falls below a specific threshold, kidney cells produce a protein (renin), which has the function to initiate a cascade of conversions and activations, respectively, of another constitutive but quiescent protein (angiotensinogen) produced by the liver and distributed in several organs. The conversion of this protein to a shorter one (now called angiotensin I) is the first step to form the right answer for solving the initial problem. Further proteins are necessary for the formation of this final answer, e.g. the protein ACE (angiotensin converting enzyme) further modulates this protein, angiotensin I by cleaving it into the short and potent protein angiotensin II. This protein represents the final answer which now has many effects on different cells in different organs in order to increase the blood pressure to normal level. This includes the production of further protein signals such as a hormone called aldosterone, the stimulation of the contraction of smooth muscle cells surrounding blood vessels within the kidney, and the production of the hormone vasopressin in the adenohypophysis in the brain, which finally plays a role in the blood volume regulation. All these effects enhance the blood pressure in the whole body.

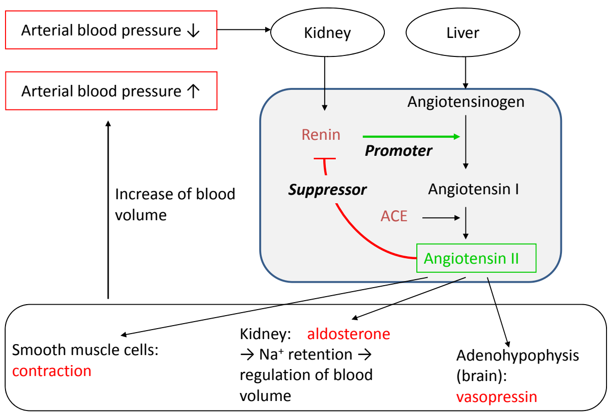

Figure 4: Signaling cascades including a molecular negative feedback mechanism provide means for the regulation of blood pressure (further organs and cascades are needed in the organism, e.g. for generation of vasopressin)

Looking at one of the target cells of angiotensin II in the kidney or smooth muscle cells, the protein binds to certain receptors on the cell surface. This binding induces an intracellular signal transduction cascade that finally results in the aforementioned actions to increase the blood pressure. A molecular negative feedback mechanism finishes the whole cellular reaction. If all receptor are bound by angiotensin II, the reaction is blocked which in turn also blocks the primary conversion of angiotensinogen to angiotensin II in the way that the initial renin secretion is blocked. Therefore, this mechanism describes a very effective remote and local control of the blood pressure, which plays a central role in the body.

In summary, renin is a promoter for the development of angiotensin II, which in turn works as an inhibitor for the production of renin. A smooth self-regulation is the result of this feedback loop [14]. We will describe the application of this methodology for optimized event detection in SANETs in the following section.

4. IMPROVED RSN

As previously mentioned, bio-inspired methodologies can be used to create appropriate feedback loops for adapting the system parameters. In our system, we want to adapt the parameters of the event detection system according to the current environmental conditions. Usually, two kinds of feedback loops are used in combination: positive feedback for short-term amplification and negative feedback for long-term regulation.

The adaptation of biological promoters and inhibitors is depicted in Figure 5. The self-regulating process that amplifies the production of angiotensin II by the production of renin is reflected by the observation of successful local event detection as well as by the overhearing of event notifications from neighboring nodes. Similarly, the suppressing reaction, i.e. angiotensin II inhibits the production of renin, is modeled by unsuccessful trials. Thus, a methodological approach can be developed that is based on the parameterization being adapted to the current situation in the network.

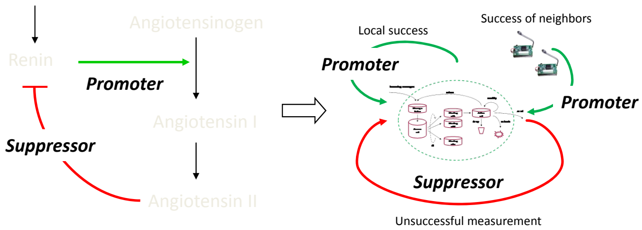

Figure 5: Adapting the biological model to the operation of RSN. The event detection system is controlled by promoter and suppressor mechanisms

Thus, we exploited the characteristics of the continuous mobility pattern of the monitored targets: in each time step, the target under observation can be expected to be present in a close proximity of its last position. Furthermore, we assume to have no knowledge about the specific mobility model, i.e. the exact direction of the target – in the simulation experiments, we employed different mobility pattern of the target to analyze the performance of the approach. It should be noted that both the regulation of blood pressure using renin as a promoter and the adapted scheme for adaptive sensor data collection represent open system, i.e. they can (and need to be) influenced by external events.

In particular, we used the following RSN rules to determine the current network situation and, therefore, to adapt the local rule execution frequency. If the local measurement was unsuccessful, the rule execution period Δt is set to its maximum value to achieve optimized energy performance (in this example, the maximum period is set to 10 s):

```
if :count = 0 then {
  !controlManagement($control :=
    rsnManagementSetEvaluationInterval,
    $text := ”10s”);
  !stop;
}
```

In contrast, if the local measurement was successful, Δt is reduced to the minimum (in the example, it is set to 1 s):

```
if :count > 0 then {
  !controlManagement($control :=
    rsnManagementSetEvaluationInterval,
    $text := ”1s”);
  !send($type := rsnSensorLuminance,
    $value := @maximum of $value);
  !drop;
}
```

Finally, the most interesting case is the exploitation of overheard messages from neighboring nodes. If such a node successfully detects a target, the radio message will usually travel faster compared to the target itself. Thus, depending on the distance to the node that detected the event, the rule execution period Δt can be updated. We used the average of the hop count as a basis measure as there is no localization scheme in place in our example. Alternatively, the real distance to the neighboring node could be evaluated.

```
!controlManagement($control :=
  rsnManagementSetEvaluationInterval,
  $text := @average of $hopCount);
```

Basically, the shown rules only represent the basic idea of the feedback loops to be used to adaptively set the rule execution period Δt. We experimented with a number of settings as well as algorithms. Selected results of these experiments are presented in the following section.

5. SIMULATION EXPERIMENTS

In order to evaluate the efficiency of RSN, we compared it to the typical setup used in other sensor network scenarios for event detection. Multiple sensor nodes are continuously measuring environmental conditions, i.e. detect mobile targets, and transmit this information to actors in their neighborhood. For the communication, we used a gossiping approach [12]. Its key objective is essentially reduced communication overhead compared to other approaches – whereas the probability that messages reach the destination might be very low in specific scenarios such as linear setups. Optimized gossiping approaches are available but out of scope of this article.

5.1 Setup and Scenario

For the simulations, we developed a simulation model using OMNeT++ 3.4b2 [20], a simulation environment free for non-commercial use, and the INET Framework 20060330, a set of simulation modules released under the GPL. OMNeT++ runs discrete, event-based simulations of communicating nodes on a wide variety of platforms and is getting increasingly popular in the communications community. Scenarios in OMNeT++ are represented by a hierarchy of reusable modules written in C++. Their relationships and communication links are stored as Network Description (NED) files. Simulations are either run interactively in a graphical environment or executed as command-line applications.

We implemented RSN in form of a C++ library. This library contains all functionality that is necessary to process RSN statements. RSN statements are formulated in a flexible script language. We integrated the RSN library into the OMNeT++ simulation framework in order to execute intensive tests and experiments with different algorithms for data aggregation, probabilistic data communication, and distributed actuation control.

We investigated the following scenario. A large number of sensor nodes are considered to periodically detect targets in their local vicinity. The results are transmitted to one of the actor nodes.

In order to evaluate the communication behavior in this scenario, we created a simulation model in which 100 sensor nodes are placed on a rectangular playground. The nodes are either distributed in form of a regular grid or on a random pattern. In addition to these sensor nodes, four actor nodes are included in the middle of each quadrant. This setup is depicted in Figure 6. Furthermore, we added a mobile target that moves either on a rectangular trajectory or based on a random waypoint model. We analyzed the event detection performance for different speeds.

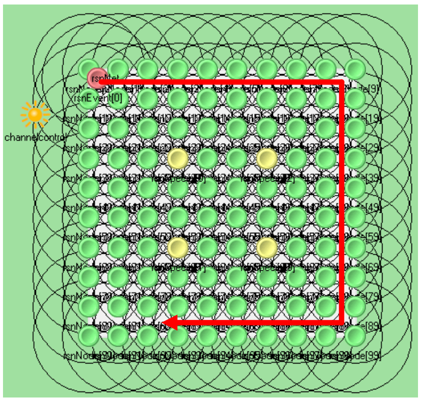

Figure 6: Simulated scenario. We evaluated grid and random deployments as well as different mobility patterns of the mobile target

The following scenarios have been analyzed: (a) static configuration of the RSN rule execution period as a baseline measurement to evaluate the adaptive behavior, and (b) versions of the dynamic feedback based approach. Additionally, we modified the deployment pattern and the mobility model of the mobile target. All the variable parameters using in the simulation are summarized in Table 1.

For all communications, wireless modules working according to the IEEE 802.11b standard have been used. All simulation parameters used to parameterize the modules of the INET Framework are summarized in Table 2.

In the simulated scenario, all the sensors have a RSN program that periodically checks for detected targets. In the adaptive case, the program adapts the rule processing period Δt accordingly. For each detected event, a message is

Table 1: Variable simulation parameters

Parameter	Values
RSN period Δt	static: 1 s, 10 s, 20 s
	adaptive: initially 10 s, 20 s
Node deployment	grid, random
Target mobility	rectangular, random waypoint
Target speed	10 m/s, 20 m/s

Table 2: INET framework module parameters

Parameter	Value
`mac.address`	auto
`mac.bitrate`	2 Mbit/s
`mac.broadcastBackoff`	31 slots
`mac.maxQueueSize`	14 Pckts
`mac.rtsCts`	true
`decider.bitrate`	2 Mbit/s
`decider.snirThreshold`	4 dB
`snrEval.bitrate`	2 Mbit/s
`snrEval.headerLength`	192 bit
`snrEval.snrThresholdLevel`	3 dBm
`snrEval.thermalNoise`	−110 dBm
`snrEval.sensitivity`	−85 dB
`snrEval.pathLossAlpha`	2.5
`snrEval.carrierFrequency`	2.4 GHz
`snrEval.transmitterPower`	1 mW
`channelcontrol.carrierFrequency`	2.4 GHz
`channelcontrol.pMax`	2 mW
`channelcontrol.sat`	−85 dBm
`channelcontrol.alpha`	2.5

gossiped with a probability 0.5 towards the actors. The actors have a much simpler programming. They just record statistics and then discard the received messages.

5.2 Measurement Results

A number of simulations have been executed with the primary objective to analyze the following characteristics:

- Degree of successfully detected events, i.e. the overall ratio of monitored event. In this context, also the message loss probability and, therefore, the degree of events processed by an actor are of interest.

- Energy performance, i.e. the number of measurements compared to the detected events. The energy performance also includes the number of messages that need to be processed by all the nodes to transmit the necessary data messages.

In order to increase the statistical significance of the simulation experiments, all simulations have been executed five times (runs). Each run lasted 30 min. After starting the simulation, the time for each sensor to start its local activities is uniformly distributed over the first 10 s. This behavior first models the initialization of real sensor nodes at arbitrary times and, secondly, it prevents collisions on the MAC layer due to synchronization effects.

All results are shown as boxplots. For each data set, a box is drawn from the first quartile to the third quartile, and the median is marked with a thick line. Additional whiskers extend from the edges of the box towards the minimum and maximum of the data set, but no further than 1.5 times the

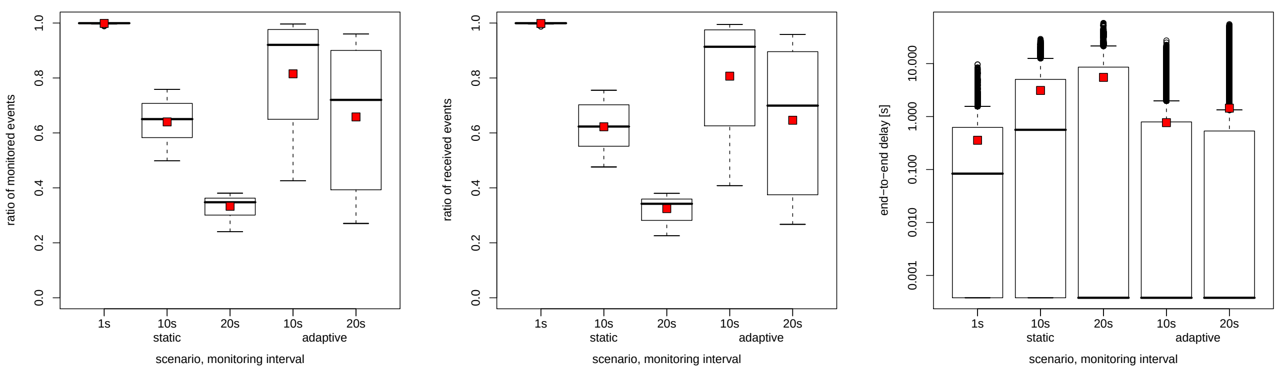

Figure 7: Ratio of detected (monitored) events compared to all possible detections (left) and the ratio of events that have been received and processed by an actor (middle); and observed communication delay (right)

interquartile range. Data points outside the range of box and whiskers are considered outliers and drawn separately. Additionally, the mean value is depicted in form of a small filled square. In most graphs, the overall mean and median are shown in the middle bar.

5.2.1 Degree of successfully detected events

First, we analyzed the number of detected events. This measure provides information about the efficiency of the particular event detection algorithm. This measure also gives a rough idea about the response time of the network. The earlier an event can be detected, the higher the probability that the event can be detected twice or more while it is in the detection range of a sensor. Figure 7 (left) shows the performance of the static vs. the adaptive approach. As can be seen the event detection ratio is optimal for the static scenario with a sampling rate of 1 s – we compare this in the next subsection with the energy performance. Looking at the other measurements, the adaptive approach always outperforms the static one: the median of the detection ratio is about 0.9 in the adaptive case for a sampling rate of 10 s compared to about 0.65 in the static case; and about 0.7 for a sampling rate of 20 s compared to about 0.35. Obviously, the feedback loop works in the positive direction, i.e. the amplification through neighbors and successfully local measurement.

Similar results can be obtained if the ratio of events as received by the actor nodes is analyzed. Figure 7 (middle) shows the simulation results. Actually, only very few message get lost in the network due to the high density and the appropriately selected gossiping probability. Further studies on the communication can be found in [8].

The density of the network obviously also contributes to the event detection quality as some inherent degree of redundancy is achieved. We analyzed this measure to better understand the network behavior (this is also an interesting measure for evaluating the energy performance as discussed in the next subsection). Figure 8 shows the results. As can be seen, the degree of redundancy of the adaptive approaches is not negligible but at least quite low – especially if compared to the clearly improved detection ratio (Figure 7).

Finally, we also analyzed the communication latency as a measure from observing an event until it has been success-

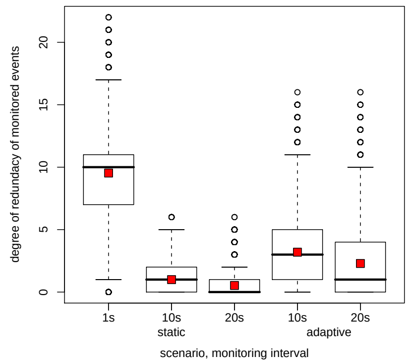

Figure 8: The degree of redundancy of monitored events shows how many sensors measured the same event

fully received and processed by an actor node. The results are depicted in Figure 7 (right) – please note the logarithmic scaling of the Y-axis (end-to-end delay). Obviously the RSN rule processing interval Δt has a strong influence on the observable reactivity of the network. On average, the adaptive scenarios clearly outperform the static ones except the static 1 s solution: for a sampling rate of 10 s, the adaptive version needs less than 1 s for the detection compared to about6 s in the static approach; and for a sampling rate of 20 s, the discrepancy is similar (about 2 s vs. 8 s).

5.2.2 Energy performance

Besides the event detection rate, i.e. the effectiveness of the sensor network, the energy performance is of interest. Especially in the context of SANETs, the overhead also characterizes the energy efficiency of the entire system, and thus, the possible *network lifetime* [4].

Figure 9 shows the simulation results for the energy performance of the event detection scenarios. All the graphs have been normalized to an energy per operation ratio of one, i.e. one event detection operation costs one energy unit. Thus, the graph depicted in Figure 9 (left) is easily explained

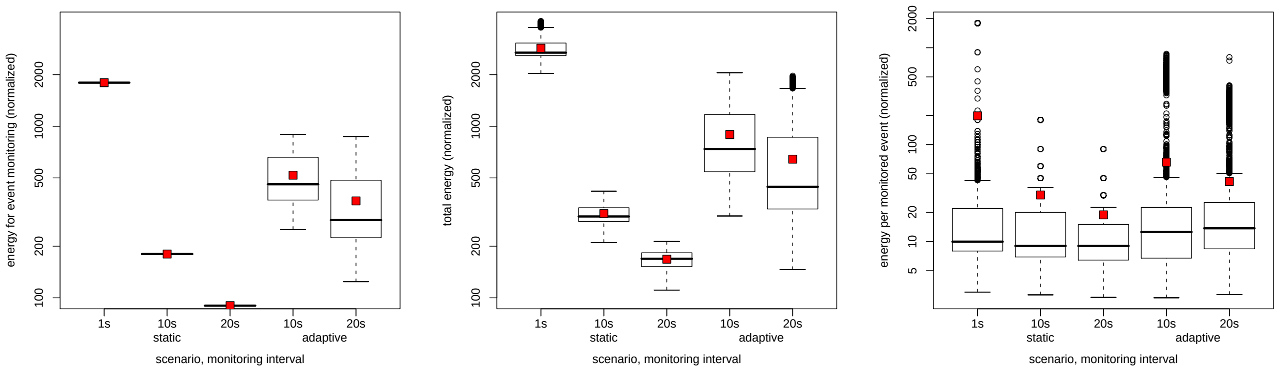

Figure 9: Energy performance of the event monitoring scenarios. Shown is the energy for event monitoring (left), the total energy including all communication activities (middle), and the energy per monitored event (right). All the measures have been normalized an energy per operation ratio of one

and represents a base measure for analyzing the network behavior. For the static scenarios, the event detection rate is exactly defined by the rule execution interval Δt, i.e. the 1 s case needs 20 times more energy compared to the 20 s case. In the adaptive scenarios, the energy values also outline the behavior of the feedback loops. Compared to the static scenario, the algorithm needs roughly 2 times more energy in the 10 s case and about 4 times more energy in the 20 s case.

A similar behavior can be observed for the case of the total energy as depicted in Figure 9 (middle). Here, the communication energy is also considered. As the network is quite dense and the gossiping is configured statically, the communication results in an almost constant increase of the energy load.

In order to analyze the overall behavior of the network, i.e. to compare the energy performance with the quality of the event detection, the energy per monitored event is analyzed in Figure 9 (right). As can be seen, the mean of all energy per monitored event measurements is almost identical (however, the average for the static 1 s scenario and the adaptive scenarios is a bit larger). Therefore, we can conclude that even though more energy is needed in the adaptive solutions, the overall performance is much better as almost no energy is wasted for monitoring activities while there is no target in the monitoring area.

As a final measurement, we analyzed the number of collisions at the MAC layer. This measure allows to determine the load distribution over the time and the ability of the network to afford the necessary number data and protocol message transmissions. The results are shown in Figure 10. It can be seen that this number is quite low (less 1 % of all transmissions) for most scenarios. Thus, no further energy load is added by an overload situation in the wireless medium.

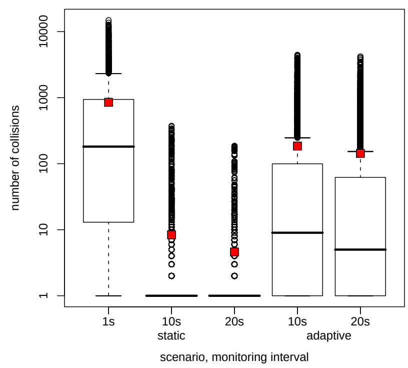

Figure 10: Number of collisions at the MAC layer

6. CONCLUSION

In this paper, we discussed the need for developing solutions for SANETs that provide real-time capabilities, i.e. a given degree of responsiveness, but at the same time a minimized energy load. In other words, the *network lifetime* [4] in improved. There are a number of approaches to enable a self-organized coordination in such SANETs, which also provide real-time operation as a key feature. However, we discovered the need for improved adaptiveness for specific application scenarios such as monitoring of mobile targets.

Inspired by the biological promoter / inhibitor system, we analyzed the applicability of such feedback systems for optimized event detection. This analysis is based on earlier work on RSN, a rule-based programming system for SANETs. Rule-based Sensor Network is able to process sensor data and to perform network-centric actuation according to a given set of rules. In particular, this system is able to perform collaborative sensing and processing in SANETs with purely local rule-based programs.

The bio-inspired approach exploits positive feedback from neighboring nodes that already detected the target, i.e. the probability of target detection increases, and from successful local monitoring. Inhibitory effects are introduced by negative feedback from unsuccessful operations. In summary, it can be said that RSN allows and inherently supports the inclusion of such application scenario specific feedback. The simulation results clearly demonstrate that the achieved network behavior in the adaptive scenarios shows optimized energy performance with improved event detection rates.

7. REFERENCES

[1] Ö. B. Akan and I. F. Akyildiz. Event-to-Sink Reliable Transport in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking (TON)*, 13(5):1003–1016, October 2005.

[2] I. F. Akyildiz and I. H. Kasimoglu. Wireless Sensor and Actor Networks: Research Challenges. *Elsevier Ad Hoc Networks*, 2:351–367, October 2004.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *ACM/Kluver Wireless Networks (WINET)*, 8(5):481–494, September 2002.

[4] I. Dietrich and F. Dressler. On the Lifetime of Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 2008. to appear.

[5] F. Dressler. Bio-inspired Promoters and Inhibitors for Self-Organized Network Security Facilities. In *1st IEEE/ACM International Conference on Bio-Inspired Models of Network, Information and Computing Systems (IEEE/ACM BIONETICS 2006)*, Cavalese, Italy, December 2006. IEEE.

[6] F. Dressler. *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons, December 2007.

[7] F. Dressler and I. Carreras, editors. *Advances in Biologically Inspired Information Systems - Models, Methods, and Tools*, volume 69 of *Studies in Computational Intelligence (SCI)*. Springer, July 2007.

[8] F. Dressler, I. Dietrich, R. German, and B. Krüger. Efficient Operation in Sensor and Actor Networks Inspired by Cellular Signaling Cascades. In *1st ACM/ICST International Conference on Autonomic Computing and Communication Systems (Autonomics 2007)*, Rome, Italy, October 2007. ACM.

[9] F. Dressler, B. Krüger, G. Fuchs, and R. German. Self-Organization in Sensor Networks using Bio-Inspired Mechanisms. In *18th ACM/GI/ITG International Conference on Architecture of Computing Systems - System Aspects in Organic and Pervasive Computing (ARCS 2005): Workshop Self-Organization and Emergence*, pages 139–144, Innsbruck, Austria, March 2005. Springer.

[10] V. C. Gungor, Ö. B. Akan, and I. F. Akyildiz. A Real-Time and Reliable Transport (RT)2 Protocol for Wireless Sensor and Actor Networks. *IEEE/ACM Transactions on Networking (ToN)*, 16(2):359–370, April 2008.

[11] H. Gupta, S. R. Das, and Q. Gu. Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2003)*, Annapolis, Maryland, USA, June 2003.

[12] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. In *21st IEEE Conference on Computer Communications (IEEE INFOCOM 2002)*, pages 1707–1716, June 2002.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *6th ACM International Conference on Mobile Computing and Networking (ACM MobiCom 2000)*, pages 56–67, Boston, MA, USA, August 2000.

[14] C. A. Janeway, M. Walport, and P. Travers. *Immunobiology: The Immune System in Health and Disease*. Garland Publishing, 5th edition, 2001.

[15] K. H. Low, W. K. Leow, and M. H. Ang. Autonomic Mobile Sensor Network with Self-Coordinated Task Allocation and Execution. *IEEE Transactions on Systems, Man, and Cypernetics–Part C: Applications and Reviews*, 36(3):315–327, March 2005.

[16] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks. In *8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002)*, 2002.

[17] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz. A Distributed Coordination Framework for Wireless Sensor and Actor Networks. In *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2005)*, pages 99–110, Urbana-Champaign, Il, USA, May 2005.

[18] G. Neglia and G. Reina. Evaluating Activator-Inhibitor Mechanisms for Sensors Coordination. In *2nd IEEE/ACM International Conference on Bio-Inspired Models of Network, Information and Computing Systems (IEEE/ACM BIONETICS 2007)*, Budapest, Hungary, December 2007.

[19] Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2003)*, pages 177–188, Annapolis, Maryland, USA, June 2003.

[20] A. Varga. The OMNeT++ Discrete Event Simulation System. In *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, June 2001.