Bio-inspired Feedback Loops for Self-Organized Event Detection in SANETs

Falko Dressler

Autonomic Networking Group
Computer Networks and Communication Systems, University of Erlangen, Germany
dressler@informatik.uni-erlangen.de

Abstract. We focus on the question of self-organized scheduling of event detection in SANETs. This question is especially challenging in very dynamic environments. Recently, a number of self-organization methods have been published that focus on network-centric operation in such networks. Based on our previously developed RSN system, which is a light-weight programming scheme for SANETs, we study the feasibility of promoter / inhibitor based feedback for self-organized schedule management of rule executions. Early simulation results outline the feasibility of the approach.

Keywords. Sensor and actor networks, feedback loop, promoter, inhibitor, network-centric operation, bio-inspired networking

1 Introduction

Programming of heterogeneous Sensor and Actor Networks (SANETs) is being investigated since several years. Especially, light-weight solutions are demanded for simplified updates of programs during run-time. Besides the requirements and challenges in terms of energy efficiency and the capability to work on low-resource embedded systems, additional coordination among the nodes need to be supported in SANETs.

Figure 1 depicts a typical SANET. Several sensor nodes are shown that directly interact with associated, i.e. co-located actuators. 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 [1]. Possible solutions can be found in approaches related to the main ideas of *autonomic networking*, i.e. the development of self-managing networks. Accordingly, most recent approaches focus on network-centric operation, i.e. the data management without central base stations, as the key paradigm to handle the mentioned challenges.

In previous work, we developed Rule-based Sensor Network (RSN), a light-weight programming scheme for SANETs [2]. RSN provides all the means for programming heterogeneous SANETs including techniques for updating programmed rule-sets during run-time. An open issue is the scheduling of rule executions, which is especially challenging in dynamic environments, i.e. in case

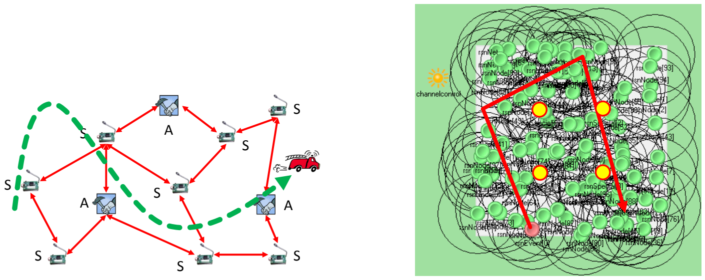

Fig. 1. Network-centric operation in SANETs for dynamic event detection: principle (left) and simulated scenario (right)

of mobility, unforeseeable events, and similar dynamics. The need for optimized scheduling of program execution becomes clear in the following application scenario that – while simplistic from a high lever point of view – shows a number of characteristic challenges for developing self-organizing SANETs. Figure 1 depicts a number of sensors trying to detect one or more mobile targets. Obviously, the detection quality increases if the sensors perform the measurement (and transmit the results) frequently. In contrast, this will lead to many unsuccessful measurements wasting energy and reducing the *network lifetime* [3].

From an algorithmic point of view, first solutions for optimized event detection are available. Akan et al. developed an event detection mechanism for SANETs [4], which provides efficient path selection between the monitoring nodes and the event sink. Similarly, the sensor-actor coordination approach by Melodia et al. [5] includes means for associating sensors to adjacent actor nodes.

This paper presents a bio-inspired promoter / inhibitor based feedback system that allows self-organized schedule management for rule execution. This approach has already successfully been applied to other problem domains [6, 7]. We implemented this scheme for use with RSN. For evaluation purposes, we developed an appropriate simulation model to analyze the behavior and performance. Early results clearly outline the feasibility of the approach.

2 RSN – Rule-based Sensor Network

2.1 RSN operation

Recently, we developed RSN, a rule-based programming system for supporting network-centric operation in heterogeneous SANETs [2]. Basically, RSN is an architecture for data-centric message forwarding, aggregation, and processing, i.e. using self-describing messages instead of network-wide unique address identifiers. In this earlier work, we proved that RSN explicitly outperforms other SANET 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 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.

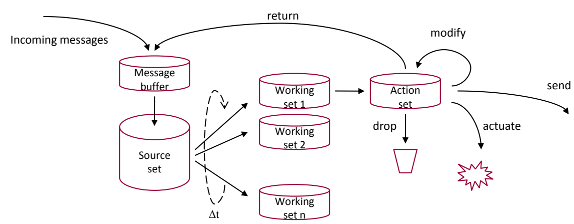

Fig. 2. The working behavior of a single RSN node. Received messages are stored in a buffer. After Δt, they are selected to a working set according to specific criteria, and finally being processed, i.e. forwarded, dropped, etc.

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.

2.2 Feedback loop controlled RSN

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. The concept is inspired by the self-regulating process of blood pressure control by the Angiotensin-Renin regulatory process [8]. The adaptation of biological promoters and inhibitors is depicted in Figure 3. Local success is defined as a successful target detection. We exploit 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. This is depicted as success of neighbors, i.e. identification of a target. 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 patterns of the target to analyze the performance of the approach.

In particular, we used the following RSN rules, which are installed on each sensor node, to determine the current network situation and, therefore, to adapt the local rule execution frequency. If the local measurement was unsuccessful,

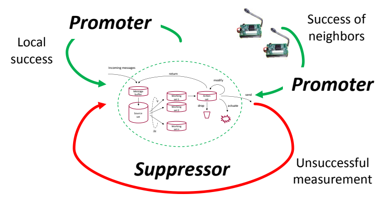

Fig. 3. Feedback loop controlled operation using promoter / suppressor mechanisms

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 {
  !controlMgmt($control:=rsnMgmtSetEvalInt, $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 {
  !controlMgmt($control:=rsnMgmtSetEvalInt, $text:="1s");
  !send($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.

```
!controlMgmt($control:=rsnMgmtSetEvalInt, $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.

2.3 Simulation experiments

In the following, we present some early simulation results. 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. 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 using a random pattern. In addition to

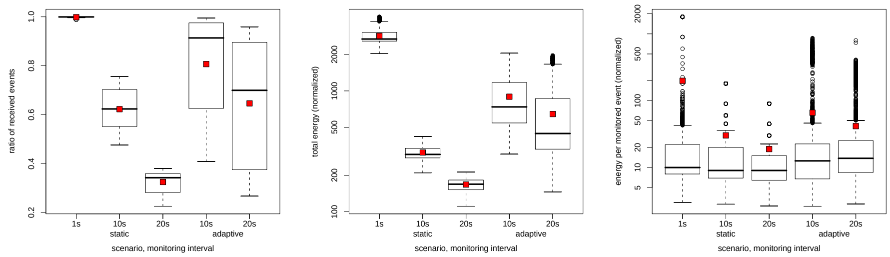

Fig. 4. Ratio of detected (monitored) events compared to all possible detections (left) and energy performance: total energy including all communication activities (middle), and the energy per monitored event (right)

these sensor nodes, four actors are included in the middle of each quadrant as depicted in Figure 1 (right). Furthermore, we added a mobile target that moves either on a rectangular trajectory or based on a random waypoint model.

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) two versions of the dynamic feedback based approach (different initial periods). Additionally, we modified the deployment pattern and the mobility model of the mobile target. 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.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.

The efficiency of the event detection algorithm can be analyzed using the number of detected events. The earlier an event can be detected, the higher the probability another sensor in the local vicinity can detect the same event, i.e. the moving target, again. Figure 4 (left) shows the performance of the static vs. the adaptive approach depicting the ratio of events as received by the actor nodes. As can be seen, the event detection ratio is optimal for the static scenario with 1 s sampling rate. The adaptive approach always outperforms the respective static configurations. Obviously, the feedback loop works in the positive direction, i.e. the amplification via neighbors and successful local measurement.

As a second measure, we analyzed the energy performance, which also describes the energy efficiency of the entire system, and thus, the possible *network lifetime* [3]. Figure 4 (middle) depicts the total energy consumed by the SANET normalized to an energy per operation ratio of one, i.e. one event detection operation costs one energy unit. Both the measurement and the communication energy are considered. For the static scenarios, the event detection rate is exactly defined by the rule execution interval Δt. In the adaptive cases, the energy values also outline the behavior of the feedback loops. In order to compare the energy performance with the quality of the event detection, the energy per monitored event is analyzed in Figure 4 (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 higher). Therefore, we can conclude that even though more energy is needed in the adaptive solutions compared to the static ones, the overall performance is much better as almost no energy is wasted for monitoring activities while there is no target around.

3 Conclusion

We presented early results of an adaptive solution for rule execution on distributed SANET nodes. The algorithm is based on a bio-inspired feedback loop that uses promoters, i.e. positive feedback, and suppressors, i.e. negative feedback. We implemented this scheme based on our previously developed RSN system, which provides collaborative sensing and processing in SANETs with purely local rule-based programs. Our 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. From the simulation results, we can see that the adaptive scenarios provide a good trade-off between energy performance and clearly improved event detection rates.

References

1. Akyildiz, I.F., Kasimoglu, I.H.: Wireless Sensor and Actor Networks: Research Challenges. Elsevier Ad Hoc Networks **2** (October 2004) 351–367
2. Dressler, F., Dietrich, I., German, R., Krüger, B.: 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, ACM (October 2007)
3. Dietrich, I., Dressler, F.: On the Lifetime of Wireless Sensor Networks. ACM Transactions on Sensor Networks (TOSN) (2008) to appear.
4. Akan, Ö.B., Akyildiz, I.F.: Event-to-Sink Reliable Transport in Wireless Sensor Networks. IEEE/ACM Transactions on Networking (TON) **13**(5) (October 2005) 1003–1016
5. Melodia, T., Pompili, D., Gungor, V.C., Akyildiz, I.F.: 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), Urbana-Champaign, Il, USA (May 2005) 99–110
6. Dressler, F.: 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, IEEE (December 2006)
7. Neglia, G., Reina, G.: 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)
8. Janeway, C.A., Walport, M., Travers, P.: Immunobiology: The Immune System in Health and Disease. 5th edn. Garland Publishing (2001)