An Event-based Local Action Paradigm to Improve Energy Efficiency in Queriable Wireless Sensor Actuator Networks

René Bergelt and Wolfram Hardt

***Abstract*—Wireless sensor networks (WSN) are deployed in a multitude of applications both in industrial and academic fields. In recent years, due to the emerge of Internet of Things (IoT) technologies and Vehicle2X communication scenarios, novel challenges for wireless sensor network platforms - regarding hardware and software - arose. Thus, challenges known from big data processing have reached the WSN scope and consequently approaches and methods have been devised to handle these. One such approach is queriable wireless sensor networks which enable their users the specification of sensing tasks in a declarative way without the need to re-program nodes in case the application requirements change. As many current WSN applications feature active parts with which nodes can directly influence their environment, the term wireless sensor actuator networks (WSAN) has been coined, setting such networks apart from solely passively measuring networks. In this article, we present a short introduction to big data processing in wireless sensor networks which motivates the usage of queriable networks. We show that in order to enable a WSAN to carry out actions energy-efficiently and in a timely manner, the novel event-based action model as originally proposed in [1] is favorable. By using a formal quantification model, we demonstrate the positive impact the new system has on the energy efficiency of a network for a given scenario. Additionally, we explore the energy saving potential when using a novel combination of the proposed approach and wake-up receiver technology.**

***Index Terms*—wireless sensor actuator network, event system, queriable networks, wake-up receiver, big data processing.**

I. Introduction

Wireless sensor networks have been originally developed for military purposes but have been quickly embraced by industrial and academic users [2], [3]. As they are able to operate in a self-sufficient way and independently from external infrastructure they enable many different usage scenarios. At the same time, system engineers face challenges regarding the resource and computing restrictions when building networks of very large dimensions as efficient use of node energy and processing power is vital in order to achieve satisfactory network lifespans. Therefore, wireless sensor networks are often implemented in an application-specific way locking the individual nodes to the network's original intended use case [4]. Flexible changes in network behavior or data aggregation are then only possible with effort regarding re-programming and re-deployment of existing nodes. The emerge of Internet of Things (IoT) and Vehicle to Vehicle as well as Vehicle to Environment (V2V, V2X) scenarios which are closely related to wireless sensor networks as they are built on top of similar or even the same technologies, such as communication protocols based on the IEEE 802.15.4 standard [5], calls for more flexible and adaptive WSN platforms. Even more, if the measurement data is the source for big data processing algorithms.

Manuscript received March 2, 2020; revised March 27, 2020. Date of current version March 31, 2020. The associate editor prof. Dinko Begušić has been coordinating the review of this manuscript and approved it for publication.

The paper was presented in part at the Conference on Software, Telecommunications and Computer Networks (SoftCOM) 2019.

R. Bergelt and W. Hardt are with the Professorship for Computer Engineering at Chemnitz University of Technology (e-mails: rene.bergelt@informatik.tu-chemnitz.de, hardt@informatik.tu-chemnitz.de).

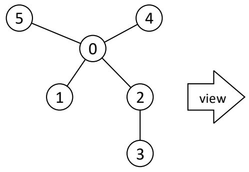
Digital Object Identifier (DOI): 10.24138/jcomss.v16i1.1021

TABLE I
Applying the 3Vs of Big Data as defined in [7] to WSN

Property	Big Data	WSN
Volume	the amount of data to be processed exceeds the capabilities of traditional tools or databases	high node density and many available sensors result in data exceeding local node resources
Velocity	new data/information is fed with high speeds/throughput into the system	sensor readings are only valuable for a certain amount of time before they are outdated and superseded by more current readings
Variety	the data to be processed is unstructured and available in many different formats (text, imagery, video, audio, differing file formats)	designedly heterogeneous WSN are a main focus of many data aggregation strategies (i.e supporting nodes which differ in their hardware and software capabilities)

A. Big Data Processing in Wireless Sensor Networks

Incidently, with an ever increasing amount of nodes in such networks and differing hardware platforms as well as data heterogeneity many challenges related to big data processing directly occur in this context [6]. The common properties of big data applications are circumscribed using the 3V model established by Beyer and Laney in [7]. According to this model, big data applications can be classified by their usage of volume, velocity and variety. As shown in [6], this model can also be applied to certain types of WSN classifying them as big data networks as presented in Table I.



Node ID	Temperature	Humidity
0	22 °C	45%
1	20 °C	40%
2	25 °C	58%
3	25 °C	...
4	...	...
5	...	...

Fig. 1. View of a wireless sensor network as virtual table

Naturally, users expect networks to be able to adapt to changing requirements with ease, for instance when final requirements are not completely clear or available during node deployment. Furthermore, these networks should be open to later enhancements as it is the case in home automation scenarios [8] and for autonomous driving applications [9], [10]. Additionally, automation and Internet of Things applications brought the need for nodes to also directly influence their environment using actuators, e.g. heating or climate control, and to control these active components based on events which have been detected in the area of the network. Consequently, such wireless sensor networks are often called wireless sensor actuator networks (WSAN)[1]. The common factor in all stated use cases is that the reduction of data to transmit necessary information or requested actions in a condensed way is critical in order to operate the network in an energy-efficient way [6].

B. *Queriable Wireless Sensor Networks*

One approach to handle large data volumes in networks as described is to model such networks as virtual database systems and has been first proposed in [11]. Figure 1 illustrates how a sample network with six nodes each having two sensors - temperature and humidity - can be viewed as a virtual table. There, each table column represents a sensor and each row a node and consequently, each cell corresponds to the value of the sensor measurement of the node at aggregation time. From the perspective of big data, such a network presents itself as massively distributed computing and storage system - albeit with very limited resources at the individual units - and is thus predestined to be used with a form of the map reduce algorithm [6]. Basically, a computational problem is split to execution units which are selected based on their data storage and computing capabilities (map step). The partial solutions are then combined to form a single or multiple final results (reduce step) [12]. In database-oriented WS(A)Ns these steps can be defined by users in a declarative way - like with traditional database systems - often by using some kind of query language [1].

So called queriable sensor networks have been implemented by researchers using several systems and concepts which provide varying levels of database abstraction with differing focuses, targeting different applications. One use case which is often encountered in WSANs is that nodes detect events in an area and then invoke a response at the same location. We define such a group of nodes as local action area. This scenario is illustrated in Figure 2. One important parameter

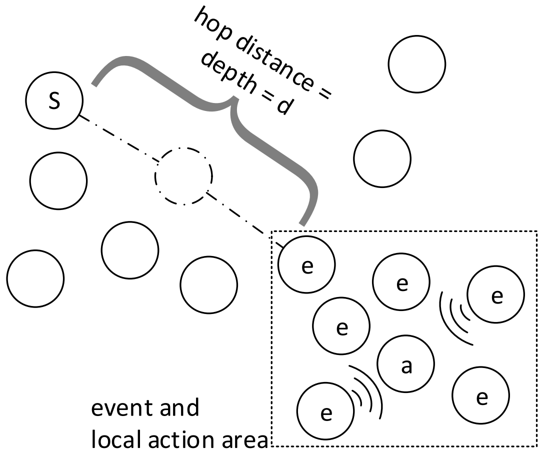


Fig. 2. Local action area in a wireless sensor network at depth d with event nodes e and actuator nodes a

of this scenario is the distance of the event area to the sink node of the network. This distance d is defined as minimum number of hops a packet needs to travel between sink and event area, i.e. the number of nodes which need to forward it. Usually, nodes send their measured sensor data to the sink and the sink may decide that local actions should be invoked by sending a corresponding packet to an actuator node, i.e. events are resolved at a global scope. Unfortunately, this behavior results in a large communication overhead as detected events have to be sent to the sink node where desired actions are generated and re-propagated through the network to affected nodes. However, in scenarios where the event detection area is the same area where actions should be taken, it is desirable that nodes can directly react to local events with local actions.

C. *Overview and Contribution*

In Section II, we briefly provide an overview of the support of existing query systems for local event responses or lack thereof, respectively. Consequently, we propose a dynamic event-based local action paradigm for queriable wireless sensor networks in Section III and how to integrate it with existing query systems for WSN. Subsequently, in Section IV we show that the proposed concept helps in improving event response times and energy-efficiency in wireless sensor actuator networks. The article closes with ideas for improvements and future work of the proposed concept and implementation in Section V.

The concept presented in this article has been first proposed in the paper *An Event-based Local Action Model for Queriable Wireless Sensor Actuator Networks* [1] on the 27th International Conference on Software, telecommunications and Computer Networks (SoftCOM 2019). This article extends on the classification of the concept in the scope of big data WSN and enhances the evaluation by formally quantifying the impact on energy efficiency using an energy efficiency model. Additionally, the energy conservation potential when employing a new approach, which combines the proposed event system with wake-up receiver technology is explored (as suggested in the original paper's *Future Work* section).

[1]or wireless sensor actor networks

II. Related Work

Event detection and tracking are widely researched topics in the wireless sensor network field. As such, a multitude of algorithms exists to efficiently gather event data from sensor nodes [13], [14]. These algorithms mainly differ in the way events can be defined, if the definition can be changed after node deployment and how event information is transmitted to a sink or monitoring node. As with traditional sensor data aggregation, usually in-network processing methods are used to reduce the amount of information to be transferred to the sink or between nodes [13]. In this article, we focus on event detection in queriable wireless sensor networks where users have the possibility to easily define arbitrary events using a declarative language and can change event definitions at runtime. Consequently, algorithms and approaches which do not support either defining events or responses depending on events after node deployment by using a kind of query language are not in the scope of this work.

Most relevant examples for implementations of queriable WSN are TinyDB [15], Cougar [16], SINA [17], SenQ [18], MadWise [19], TikiriDB [20] and Planetary [6]. In [1] we have given a survey on the state of event support of the aforementioned query systems. In summary, TinyDB, SenQ, SINA, TikiriDB as well as Planetary all provide support for node events to some extent. The only exception is MadWise which does come with neither actuator nor event support as it solely focuses on sensor data acquisition. However, there is no system which allows to signalize events to other nodes and only few allow to raise events based on data from surrounding nodes. Mostly, events are seen as pre-defined hardware states which can be added as conditions to queries. As such events are constrained to the node they occurred on. Consequently, event information has to be sent to a central supervisor node (sink) which then decides which actions should be taken, eventually informing affected nodes. That means, with a rising network size reaction times and energy consumption increase as nodes have to forward event and control data between the sink node and the network part which should carry out event reactions.

In this article, we use Planetary and its declarative language PlanetaryQL as the base for our proposed local action model concept. However, all other systems could be adapted to support this model, as it conceptually builds on top of an abstract query processor and declarative query language.

III. Conceptual Design

In order to create a local action model for a queriable WSN the following three main points have to be considered [1]:

event definition How can the user abstractly define an event in the provided query language

event propagation How can events be propagated to nodes and which information can they convey

event response How can the user define actions to be taken based on events

For defining events we proposed to extend the query language of the employed system with possibilities to name and raise events, i.e. indicating that an event occurred [1]. As

```
RAISE evt'high_temperature
FOR 2 HOPS
WHERE temperature > 45
EVERY 5s
```

Lst. 1. Definition of a primitive event in PlanetaryQL

```
RAISE evt'temperature_alert
FOR 10 HOPS
WHERE evt'high_temperature >= 10
EVERY 30s
```

Lst. 2. Definition of a complex event in PlanetaryQL

introducing events should not lead to confusion, their names should be easily distinguishable from other identifiers. For instance, this can be achieved by prepending event names with *evt'*. Since local area events have to be propagated to neighboring nodes, but usually should not traverse the whole network, the user has to be able to specify a maximum propagation distance, for example in hops or meters, when using location-aware nodes. An example for an event definition in an extended version of PlanetaryQL is given in Listing 1. Here, the event *high_temperature* is raised when the temperature sensor reading is above $45\,°C$. The current sensor value is evaluated every 5 seconds.

Basically, two types of events can be distinguished: *primitive* and *complex* events [1]. The first ones only depend on local node information, i.e. hardware events or sensor readings, whereas the latter depend on local information and information received from other nodes (e.g. in the form of events). To facilitate voting algorithms event information is not binary but the event is a counter representing the number of nodes which have raised this event during a definable event timespan. The definition of a complex event which is raised when at least 10 neighboring nodes have raised a high temperature warning event is shown in Listing 2.

When an event is raised by a node, it is transmitted as special purpose message which has been optimized for small size. It only contains the event name and lifespan and/or maximum propagation distance. Upon receiving an event a node re-evaluates all queries from its query store which contain the event in their condition set. As usual, when all conditions are met the corresponding queries are executed. This enables users to formulate queries which do not have an automatic re-evaluation period but instead depend on one or more events and thus can be executed asynchronously to the epoch of the query system. We see events as a natural extension of the query concept instead of a separate concept. Therefore, events can be used at all places where sensor readings can be used when declaring a query, be it aggregation queries or actuator control. Listing 3 shows how a user could have a node light an led in

```
ACT led0(255, 0, 0), buzzer(100)
WHERE evt'temperature_alert
```

Lst. 3. Definition of an actuator controlled by an event in PlanetaryQL red and sound its buzzer upon receiving an event signaling a high temperature. Furthermore, in [1] we also depicted and evaluated how the event system can be used to supersede certain types of sub queries[2]. In the next section we discuss how the proposed concept improves the latency and energy efficiency of event responses in queriable WSAN.

IV. Evaluation and Discussion

The following evaluation results have been obtained by simulating different network and application scenarios in the OMNeT++ simulator[3] based on the extended Planetary database abstraction layer for WSAN. The obtained results are compared to the performance of two other methods. First, a non-queriable system which collects sensor readings without in-network processing and where nodes forward individual packets until they reach the target. And secondly, a queriable WSN platform without event propagation where sensor readings are forwarded to the sink which then sends out control commands to actuators as response. The scenario consists of an event location in a wireless network where an actuator node waits for five neighbor nodes to signal an event to start a response, e.g. activating its actuator. The location of the event area has been modeled as distance of the actuator node to the sink (depth d in hops, see Figure 2). In the following sections we examine the performance of the proposed event system with regard to network traffic reduction, improvement of energy efficiency and energy conservation potential when employing wake-up receiver technology.

A. Network Traffic

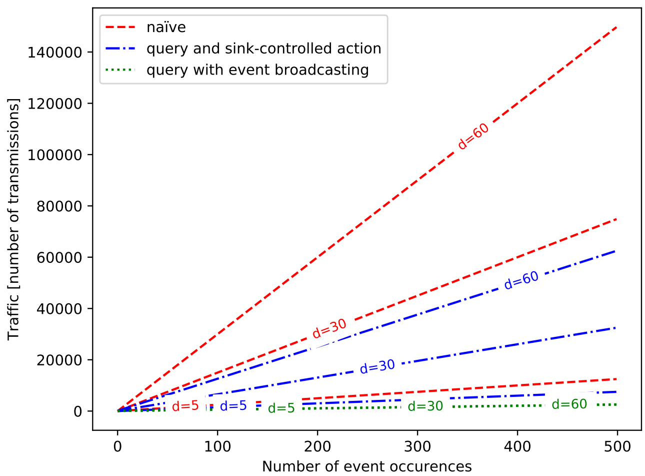


Fig. 3. Network traffic volume in number of transmissions depending on the number of event detections and depth d

Traffic volume is an important factor in wireless sensor networks due to the energy-consumption of wireless transmission and interference becoming more likely when sending more often. Consequently, a major goal in WSN routing are a low transmission count and extended node as well as radio module standby times [22], [23]. Figure 3 shows the traffic volume measured in number of required transmissions to respond to a number of consecutively detected events for the different aforementioned approaches. It has to be considered that the query-based approaches have a preparation cost, i.e. the queries have to be propagated to the network. In the case of periodic queries this cost can be considered a setup cost as the propagation is only required once at the beginning of measurement [15]. Consequently, it becomes negligible with a rising number of event occurrences. However, without event broadcasting in order to invoke an event response an action query has to be sent to the action node every time an event should be carried out. As can be seen in the graph, the naïve approach quickly becomes unfeasible for larger networks as the number of required packet transmissions is too high to be efficiently routed. Since for the event broadcasting approach no communication with the sink is required, the number of transmissions is independent from the event area's distance to the sink. In [1] we have shown a comparison for event response latencies for all three approaches. In summary, the response times for the naïve and non-event query platform rise with increasing network depths and event occurrences, as network round-trips are required. In contrast to this, the event broadcasting system's response time does not depend on the depth parameter as event detection and response are handled directly at the local event area. Accordingly, it provides a much lower latency for event responses.

B. Energy Efficiency

In addition to evaluating the mere reduction in network traffic, in this article, we also want to assess how the event system generally impacts the actual energy consumption and energy efficiency (and thus the expected lifespan) of the WSAN it is used with. In order to compare the energy efficiency of different algorithmic implementations a type of metric is needed. The main problem is that energy efficiency is not the same as energy conservation, since efficiency is defined as the ratio

$$\phi_P = \frac{\text{gain}_P}{\text{cost}_P} = \frac{G_P}{C_P} \tag{1}$$

for a given task or process P, where gain is a measure for the task a WSAN fulfills and cost its energy consumption. In contrast to this, energy conservation just means using less energy without considering possible changes of usefulness or task completion. This means that in order to compare the energy efficiency of different approaches the gain of an operation has to be evaluated. To solve this, a model for quantifying energy efficiency in WSN has been proposed in [23] which allows to compare different approaches with regard to the same application or individual tasks. Basically in this model, a local (for an individual node) and a global energy efficiency are defined. Trivially, the local gain (G_n) is either 0 or 1 depending on whether the node fulfills its functional requirements and contributes to the task at hand (i.e. if it is required to fulfill the task). The local cost is represented by a

[2] A sub query (or sub select, nested query, JOIN in traditional relational database systems) in a WSN is a query which is part of another query, usually on a different, neighboring node [21].

[3] https://omnetpp.org

cost function, which for simplicity we define as $C_n = 1 + E_n$ for a given node n with E_n being the total energy consumption of the given node during the runtime of the observed task. Based on this, the global energy efficiency in a WSN with the node set N for a given task P can be calculated as a sum of the local gains divided by the sum of the total local costs. As data transmission processes play a vital role in WSN we have to take them into consideration, differentiating processes necessary for the task (and thus increasing the gain) and those unnecessary for the task. By assuming, that there always exists a communication path between two given nodes in the network we can adapt the original formula from [23] to

$$\phi_P = \frac{\sum_{\forall n \in N}(G_n * (1 + \sum_{\forall m \in N} \omega_{n,m} * E_{n,m}))}{\sum_{\forall n \in N} C_n} \quad (2)$$

where

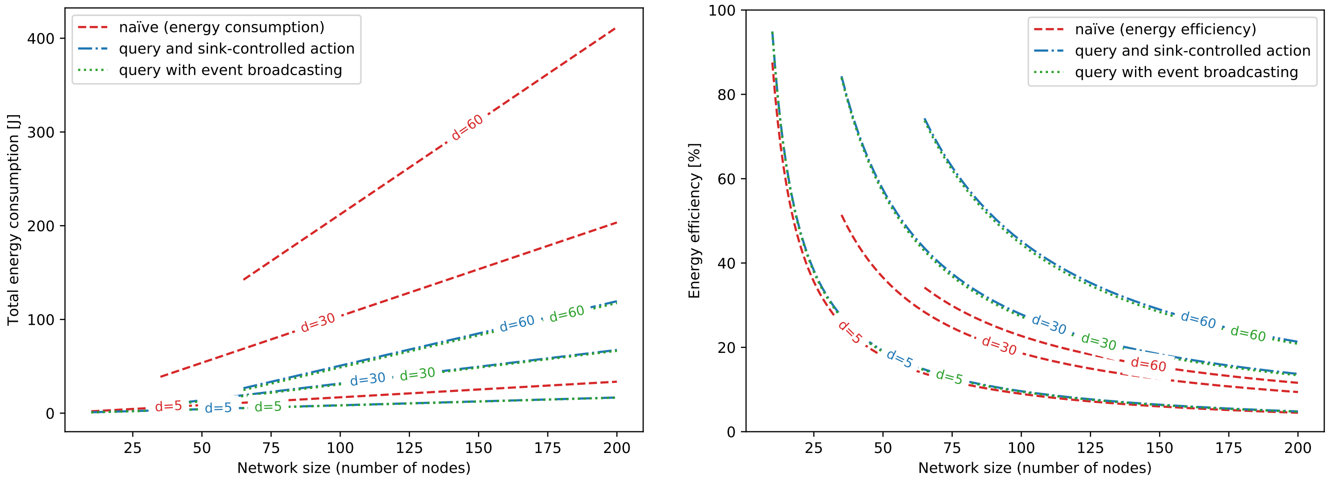
$$\omega_{n,m} = \begin{cases} 1, & \text{communication between node } n \text{ and } m \\ 0, & \text{else} \end{cases} \quad (3)$$

and $E_{n,m}$ equals the energy needed for this communication (which is also included in E_n and therefore in C_n). This means that we reduce the energy efficiency value for all nodes and communication processes which consume energy but are not required to fulfill the observed task. Thus, one could interpret ϕ as ratio between *useful energy* and *total energy* spent. As energy consumption itself is a continuous value we have to compute the energy efficiency for a given timespan, i.e. the time it takes to carry out the observed process or the lifespan of the network. Likewise, this metric does not judge the total amount of energy consumed but only how much of the energy was used to actually fulfill the task. Additionally, it has to be considered that only energy efficiency values which have been calculated using the same cost function can be compared. We can now use this definition to compute energy efficiency values for the aforementioned scenarios. For this, we assume that the nodes of the simulated network are built on top of the PLANet platform, which is a sensor node developed by the University of Chemnitz which is equipped with a wake-up receiver [24]. A wake-up receiver is a specialized, self-contained radio module which listens for a single, specific signal (i.e. communication request) upon which it triggers an event which can "wake-up" the main radio module to carry out the actual communication. This allows a sensor platform to switch to low power modes more often while still staying reachable for communication attempts. For the PLANet boards the energy consumption values for different operation modes have been measured as shown in Table II. As many WSN applications use a protocol based on the wireless communication standard IEEE 802.15.4 (e.g. ZigBee) [8], [26], [27], we use it for our evaluation (data rate is 250 kbit/s at 2.4 Ghz). In the following, we assume that the sensor data which needs to be evaluated for the event detection is 8 byte wide (payload). We assume the packet header size to always be 8 bytes (address + checksum) and an event broadcasting packet to be 12 bytes, containing event name and value. Furthermore, each node needs a time of 100ms to process an incoming packet and forward it, if needed. A packet which instructs a node to invoke an action (as event response) shall be 12 bytes

TABLE II
ENERGY CONSUMPTION OF THE PLANET PLATFORM [23]

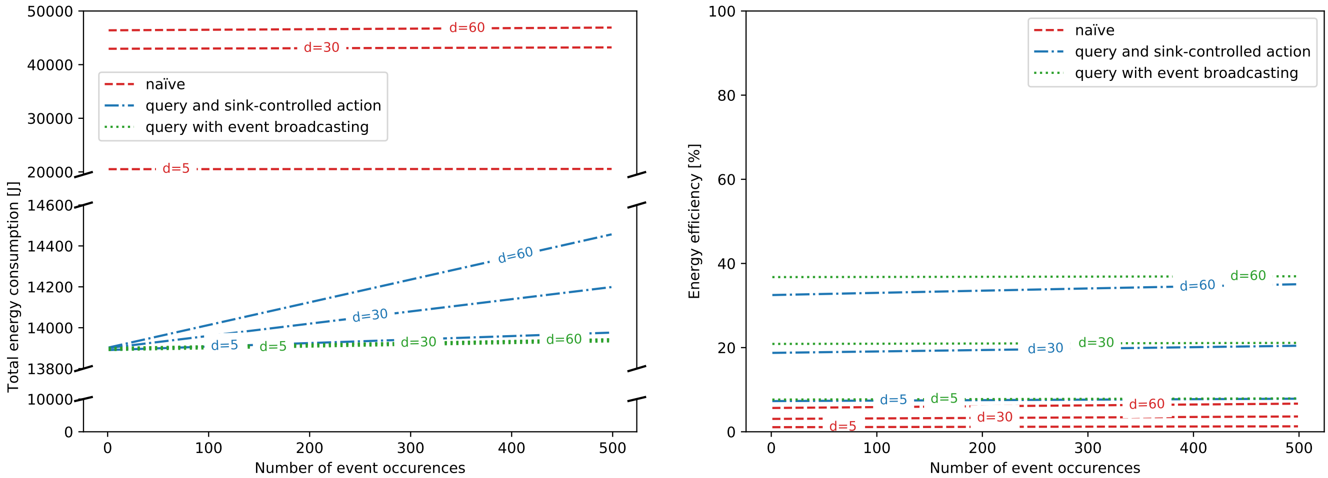
Power consumption in mW	PLANet Board (ARM7)	Radio module (IEEE 802.15.4)	Wake-up receiver µRX1080 [25]
Idle	≤ 162	≤ 115	-
Sleep	≤ 0.16	≤ 0.03	-
Receive	-	≤ 165	≤ 0.03
Transmit	-	≤ 115	≤ 10.24

as well (actuator id and parameters). By using these parameters we can now calculate energy efficiencies for the same scenario as assessed in Figure 3. For this evaluation we assume that there are no environmental influences and no transmission collisions, so the computed energy efficiency values are upper-bounds. As we want to assess the total energy consumption of the observed network, we set the node duty cycle to 80%. This means that nodes sleep 80% of their lifetime and are only reachable for 20% of the time (radio online). First, we compute the energy efficiency of a single event occurrence, i.e. how efficient does the network use its resources for an individual task. The results for energy consumption and accordingly calculated energy efficiency are given in Figure 4 for all three approaches for different network sizes and event area depths. It can be seen that the energy efficiency does not only depend on the actual total energy consumption but also on the number of nodes which are needed to carry out the observed task (detecting an event and triggering a response). For example in a large network where the event occurs at depth 60, more nodes are actually needed to carry out the task as when compared to an event in depth 5, and so the actual efficiency is better. The reason the total energy consumption is lower for smaller depths is that the response from the network takes less time and thus the observed task duration is smaller. Likewise, for the query-based approaches we assume that the event gets triggered as soon as the query has been propagated to the target area. So for smaller depth values a shorter amount of time is needed for propagation to reach the event nodes and thus a shorter duration of network lifetime is measured. For larger networks the energy efficiency values stabilize since the minor increase in traffic to execute event responses diminishes when compared to the idle consumption of the whole network. The curves for the two query-based approaches almost match as only a single additional round-trip for the sink-controlled action is required. The larger the network is the lower the efficiency becomes since the amount of nodes which do not contribute to the task increases. Therefore too much energy is spent on idling nodes which are never needed, negatively affecting the energy efficiency. Unfortunately, the differences in event response times and thus observed network lifetime complicates the comparison the values between different scenarios. So in order to make a general statement of the energy efficiency of the network during its lifetime we have to compare the approaches for longer run times. To reduce the number of influencing parameters we execute the following assessment for a network with



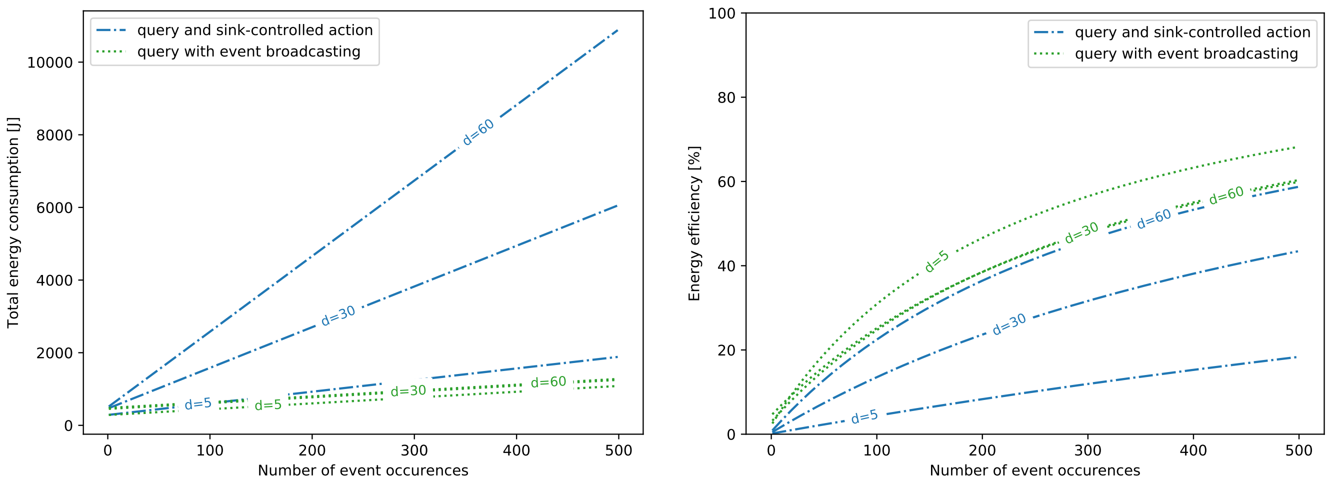
(a) Energy consumption (b) Energy efficiency

Fig. 4. Energy consumption and efficiency for a single event detection and response for different network sizes



(a) Energy consumption (b) Energy efficiency

Fig. 5. Energy consumption and efficiency for multiple event detections and responses for a network of 100 nodes



(a) Energy consumption (b) Energy efficiency

Fig. 6. Energy consumption and efficiency for multiple event detections and responses for a network of 100 nodes by combining approaches with wake-up receiver technology (WuRx)

a fixed number of nodes (n = 100) where an event can occur every 5 seconds. We observe the network until a maximum of 500 events can have occurred which takes around 40 minutes. Additionally, we assume that nodes, which are neither part of the path to the event area (with depth d) nor event nodes, have an average hop count to the sink of $\frac{d}{3}$ to get a more realistically distributed network. Figure 5 shows the energy consumption and efficiency depending on the actual number of events which occurred during the observed time. As expected the total energy consumption of the naïve approach is much higher than that of the other two approaches (consider the split y-axis). Furthermore, the energy consumption of the normal query-based approach visibly increases with the number of events which occur during the observed timespan whereas the event-based approach has no significant additional consumption. For the naïve approach it can be seen that the individual energy efficiency is stable during the whole operation time of the network. This is due to the fact that the actual communication processes are mostly independent of the nodes which detect events - which can also be seen by the fact that the total energy consumption for all cases are stable and only slightly increase with event occurrences for which responses have to be sent (not visible in the used scale). The energy needed to send an event response action is very small and fades when compared to the basic energy consumption of the network. Still, the base consumption for smaller depth values is less since on average less hops are needed to reach the sink node and therefore less packet forwarding is needed. As much energy is spent to keep nodes in standby and for unnecessary communication, the energy efficiency tops at around 5%, being even lower for smaller depth values. In contrast to this, the results when using a query-based system yield a much smaller total energy consumption. Again, the consumption for the normal query-based approach visibly increases with the number of occurred events as additional network round trips become necessary. For the proposed event system all three curves for total energy consumption are basically the same, this is due to the fact that the initial query needs to be propagated to the complete network (in a cold start at least) and the subsequent event detection and action messages are independent of the depth of the event area in the network (as already discussed regarding Figure 3). The energy efficiency curves for both are very similar, but the one for the sink-controlled action approach even increases with the number of observed events as more *meaningful* communication is carried out relative to the idle consumption and initial propagation cost. Both max out at approximately 8%, 20% and 38% for depth values 5, 30 and 60, respectively. The energy efficiency for smaller depth values is lower since the amount of nodes which actually contribute to the task is smaller when compared to the total node count. As such, much energy is *wasted* by non-contributing nodes.

C. Energy Efficiency using Wake-up Receiver Technology

To conclude the evaluation we now assess the potential of wake-up receiver technology to further improve the energy efficiency. In our scenario where event intervals are large when compared to the times needed to actually transmit information, we assume that after each transmission a node can switch itself and its main radio module to low-power mode as it can always be woken asynchronously (i.e. no duty cycling is needed). Subsequently, every transmission has to be preceded by a wake-up signal to the target node(s). Fortunately, the power consumption of these wake-up requests and the listening wake-up receivers is very small (see Table II) when compared to all other processes on the sensor node. Due to the synchronous, i.e. periodic, data aggregation nature of the naïve approach an extension with wake-up receiver technology is not meaningful. Therefore, this approach has been omitted in the following evaluation and only the query-based approaches are shown. Combining these systems with nodes which are equipped with wake-up technologies yields the results shown in Figure 6. The main observation is that the impact of the idle energy consumption of the nodes on the energy efficiency has been removed and it now mainly depends on the network traffic. It is visible that the energy consumption for the sink-controlled action approach rises much more with the number of occurring events since for each event a part of the network (depending on the depth value) has to be woken by wake-up call which leads to increased energy consumption at all of these nodes. As the event-based approach only needs to wake nodes in the actual event area the total energy consumption is much less dependent on both the number of events and the depth of the event area. Consequently, combining wake-up technology with the event-proposed system can be considered a natural fit due to the asynchronous nature of the events. This is also reflected when considering the energy efficiency curves which increase with the number of event occurrences, as discussed before, and are generally higher for the event-based approach as less network traffic for an event response is required. It is noteworthy that for the event-based approach now the energy efficiency for the scenario $d = 5$ is the highest. This is due to the fact that the remaining part of the network, which does not contribute to the task, consumes much less energy in relation to a network without wake-up receiver technology and the initial preparation cost is low due to the average hop count being lower than with the other scenarios. Nonetheless, even for the normal query-based approach energy efficiency increases with the number of events since all communication is still considered necessary (and thus *useful*) for task fulfillment.

V. Conclusion and Future Work

In this article, we have shown that challenges in current and future usages of WSANs can be seen similar to big data processing systems. Consequently, we motivated the use of queriable WSANs to realize applications with such WSANs. We demonstrated how existing platforms for query processing in wireless sensor networks can be extended with an event broadcasting concept to improve response times and reduce network traffic of event area applications. Additionally, we have shown how employing wake-up receiver technology helps in improving the energy efficiency. One of the main benefits of the proposed approach is that the user does not have to state or even implement this explicitly but can demand it by using the system's query language. Furthermore, we have shown how an energy efficiency model for WSN can be applied to a specific scenario and how it can be used to compare the performance of different approaches regarding their energy consumption objectively. However, in order to make more general statements further evaluation is necessary since we only computed the energy efficiency for WSN which carry out a single task. In reality this is rarely the case and many operations are running in parallel affecting the energy consumption and efficiency of the network. In such cases the presented energy efficiency model can still be used by calculating the ratio of total gain over all tasks to the total energy consumption. However, more refined models of network tasks have to be used. For the used energy efficiency quantification the total energy consumption plays only a minor role since it mainly indicates if the energy spent contributed to the network task. This may lead to counter-intuitive values when approaches with a much higher total consumption get good efficiency values and is much more likely when more than one network task is observed. Thus, the total energy consumption always has to be taken into account when comparing the efficiency values. In the future, a refined model which penalizes higher consumptions more could be used.

Future research could focus on using unidirectional transmitters for the signaling of events which possibly results in lower hardware costs and a reduced energy consumption of such nodes. Algorithms to ensure the transmissions in unidirectional scenarios have been proposed in [28] and could be very well applied here. Based on the improvements when using wake-up technologies we have presented in this article, we see a high potential in an extension of the event system to explicitly support use cases, where one node checks for an event and wakes up its neighboring nodes once it occurs. The role of such a monitor node could then be automatically rotated, allowing to evenly distribute energy consumption between nodes, (i.e. automatic node spanning duty cycling). Furthermore, the approach could also be used in conjunction with other platforms than sensor nodes where for instance complex systems raise events which lead to a response of the surrounding wireless sensor actuator network, e.g. for diagnostic applications. In addition, exploring potential usages for query based wireless communication between components of a single vehicle scenarios might be promising since it has been shown that these suffer from a massive packet loss when using higher data rates and uncoordinated transmissions due to packet collisions [29].

References

[1] R. Bergelt and W. Hardt, "An event-based local action model for queriable wireless sensor actuator networks," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2019, pp. 1–6, doi: 10.23919/SOFTCOM.2019.8903666.

[2] Silicon Labs, "The evolution of wireless sensor networks (white paper)," Tech. Rep., 2013.

[3] V. Jindal, "History and architecture of wireless sensor networks for ubiquitous computing," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 7, no. 2, Feb. 2018.

[4] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks - Technology*. John Wiley & Sons, Inc., 2007.

[5] N. Dinh and S. Lim, "Performance evaluations for ieee 802.15.4-based iot smart home solution," *International Journal of Engineering and Technology Innovation*, vol. 6, no. 4, 2016.

[6] R. Bergelt, M. Vodel, and W. Hardt, "Energy efficient handling of big data in embedded, wireless sensor networks," in *2014 IEEE Sensors Applications Symposium (SAS)*, Feb 2014, pp. 53–58, doi: 10.1109/SAS.2014.6798916.

[7] M. A. Beyer and D. Laney, "The importance of 'big data': a definition," *Stamford, CT: Gartner*, pp. 2014–2018, 2012.

[8] D. man Han and J. hyun Lim, "Smart home energy management system using ieee 802.15.4 and zigbee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403–1410, Aug 2010, doi: 10.1109/TCE.2010.5606276.

[9] C. Dannheim, "The vehicle as mobile sensor in a collaborative network," Ph.D. dissertation, Open University Hagen, Germany, 2015, urn: urn:nbn:de:hbz:708-30260.

[10] J. Pan and C. Borcea, *Handbook of Sensor Networking: Advanced Technologies and Applications*. CRC Press, Taylor & Francis Group, 2015, ch. Vehicular Sensor Networks, pp. 19–1 – 19–27.

[11] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks," in *Proceedings Fourth IEEE Workshop on Mobile Computing Systems and Applications*, 2002, pp. 49–58, doi: 10.1109/MCSA.2002.1017485.

[12] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, 2004, pp. 137–150, doi: 10.1145/1327452.1327492.

[13] O. Saleh and K.-U. Sattler, "Distributed complex event processing in sensor networks," *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 2, pp. 23–26, 2013, doi: 10.1109/mdm.2013.60.

[14] A. Nasridinov, S.-Y. Ihm, Y.-S. Jeong, and Y.-H. Park, "Event detection in wireless sensor networks: Survey and challenges," in *Mobile, Ubiquitous, and Intelligent Computing*, J. J. J. H. Park, H. Adeli, N. Park, and I. Woungang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 585–590, doi: 10.1007/978-3-642-40675-1_87.

[15] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, Mar. 2005, doi: 10.1145/1061318.1061322.

[16] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Record*, vol. 31, pp. 9–18, 2002, doi: 10.1145/601858.601861.

[17] H. Ehsan and F. Khan, "Query processing systems for wireless sensor networks," in *Communications in Computer and Information Science*, vol. 150, 04 2011, pp. 273–282, doi: 10.1007/978-3-642-20975-8_30.

[18] A. D. Wood, L. Selavo, and J. A. Stankovic, "Senq: An embedded query system for streaming data in heterogeneous interactive wireless sensor networks," in *Distributed Computing in Sensor Systems*, S. E. Nikoletseas, B. S. Chlebus, D. B. Johnson, and B. Krishnamachari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 531–543, doi: 10.1007/978-3-540-69170-9_40.

[19] G. Amato, S. Chessa, and C. Vairo, "Mad-wise: a distributed stream management system for wireless sensor networks," *Software - Practice and Experience*, vol. 40, no. 5, Apr. 2010, doi: 10.1002/spe.965.

[20] A. P. Sayakkara, P. W. Senanayake, K. Hewage, N. M. Laxaman, and K. D. Zoysa, "The deployment of tikiridb for monitoring palm sap production," in *Proceedings of the 4th international conference on Real-world wireless sensor networks*, Jan. 2010, doi: 10.1007/978-3-642-17520-6_21.

[21] H. Kang, "In-network processing of joins in wireless sensor networks," *Sensors*, vol. 13, no. 3, 2013, doi: 10.3390/s130303358.

[22] S. Li and X. Wang, *Handbook of Sensor Networking: Advanced Technologies and Applications*. CRC Press, Taylor & Francis Group, 2015, ch. Wireless Sensor Hierarchical Networks, pp. 5–1 – 5–17.

[23] M. Vodel, R. Bergelt, and W. Hardt, "Modelling, quantification, optimisation - energy efficiency in distributed, embedded systems," in *2014 IEEE Sensors Applications Symposium (SAS)*, Feb 2014, pp. 47–52, doi: 10.1109/SAS.2014.6798915.

[24] M. Vodel, M. Caspar, and W. Hardt, "A capable, lightweight communication concept by combining ambient network approaches with cognitive radio aspects," in *2010 17th International Conference on Telecommunications*, April 2010, pp. 430–434, doi: 10.1109/ICTEL.2010.5478784.

[25] "Ultra low-current wakeup receiver x1080," Fraunhofer Institut für integrierte Schaltungen IIS, Erlangen, Germany, Tech. Rep., 2010.

[26] S. Lin, J. Liu, and Y. Fang, "Zigbee based wireless sensor networks and its applications in industrial," in *2007 IEEE International Conference on Automation and Logistics*, Aug 2007, pp. 1979–1983, doi: 10.1109/ICAL.2007.4338898.

[27] H. Ayadi, A. Zouinkhi, B. Boussaid, M. N. Abdelkrim, and T. Val, "Energy efficiency in wsn: Ieee 802.15.4," in *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Dec 2016, pp. 766–771, doi: 10.1109/STA.2016.7952060.
[28] P. Parsch and A. Masrur, "On reliable communication in transmit-only networks for home automation," *Journal of Network and Computer Application*, 2017, doi: 10.1016/j.jnca.2017.10.023.
[29] S. Krug, H. Toepfer, T. Hutschenreuther, and J. Seitz, "Towards robust communications of wireless sensor networks in vehicular environments: A case study," *Journal of Communications Software and Systems*, vol. 13, no. 4, 2017, doi: 10.24138/jcomss.v13i4.364.

René Bergelt was born in Germany in 1988. He acquired the German Diploma degree in Computer Science at University of Technology, Chemnitz in 2012. Currently, he works as research associate at the Department of Computer Science, Chair of Computer Engineering at Chemnitz University of Technology, Germany where he also pursues his PhD studies. His current research focuses on energy efficiency optimization strategies in distributed wireless networks and their connection to big data processing. Additional fields of interest are protocol engineering and software engineering and test for embedded systems.

Prof. Dr. Dr. h.c. Wolfram Hardt is professor for Computer Science and head of the Computer Engineering Group at the Chemnitz University of Technology. He was born in Germany in 1965 and received the German Diploma degree in Computer Science in 1991 from the University of Paderborn. Prof. Hardt received the Ph.D. degree in Computer Science from the University of Paderborn in 1996. From 2000 to 2002 he was chair of the Computer Science and Process Laboratory at the University of Paderborn / Germany. In 2003 Prof. Hardt became chair of the computer engineering Dept. at the Chemnitz University of Technology, Germany. He is a member of the Association for Electrical, Electronic and Information Technologies (VDI/VDE), the Association for Computer Science (GI) and the Association for Computing Machinery. Since 2006 he is committee member of the DATE conference - "Design Automation and Test in Europe". His research interests include hardware/software co-design and reconfigurable hardware.