Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks

WOLF-BASTIAN PÖTTNER and HANS SEIDEL, Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig
JAMES BROWN and UTZ ROEDIG, School of Computing and Communication Systems, Lancaster University
LARS WOLF, Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig

Wireless sensor networks for industrial process monitoring and control require highly reliable and timely data delivery. To match performance requirements specialised schedule based medium access control (MAC) protocols are employed. In order to construct an efficient system it is necessary to find a schedule that can support the given application requirements in terms of data delivery latency and reliability. Furthermore, additional requirements such as transmission power may have to be taken into account when constructing the schedule. In this paper we show how such schedule can be constructed. We describe methods and tools to collect the data necessary as input for schedule calculation. Moreover, due to the high complexity of schedule calculation, we also introduce a heuristic. We evaluate the proposed methods in a real-world process automation and control application deployed in an oil refinery and further present a long-term experiment in an office environment. Additionally, we discuss a framework for schedule life-cycle management.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*; *Network topology*; *Wireless communication*; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-based Systems—*Process control systems*; C.4 [**Computer Systems Organization**]: Performance of Systems—*Fault tolerance*; *Reliability, availability, and serviceability*

General Terms: Algorithms, Design, Experimentation, Management, Reliability

Additional Key Words and Phrases: Reliability, Scheduling, Schedule, Timeliness, Wireless Sensor Network, Wireless Sensor and Actor Network, WSAN, WSN

1. INTRODUCTION

Wireless sensor networks for industrial process monitoring and control require highly reliable and timely data delivery. Such high network performance is required as control loops are mapped on wireless networks. Data has to arrive with a set reliability within a strict time bound. To match performance requirements of industrial process monitoring and control applications, specialised medium access control (MAC) protocols are employed. These protocols are generally TDMA based, and the schedule must provide time for transmissions and potential re-transmissions to compensate for lossy links. Examples of such protocols are GinMAC [Suriyachai et al. 2010] in GINSENG [Sreenan et al. 2009] and TSMP [Pister and Doherty 2008] in WirelessHART [HART Communication Foundation 2008].

DOI: http://dx.doi.org/10.1145/0000000.0000000

A static schedule is feasible for the outlined application domain as nodes are fixed and environmental conditions are reasonably stable. It is possible to determine a schedule at network deployment time which remains valid for a relatively long time period. Only occasionally the schedule may have to be adapted to reflect long-term changes in the deployment area. Using a static schedule is beneficial as the system provides deterministic performance with little management overhead at system runtime.

In previous work [Suriyachai et al. 2010] we have shown that it is generally possible to construct a system using a static schedule to support process automation and control applications. However, it is not a trivial task to find the best schedule that fulfils the application requirements during network deployment. The schedule is dependant on the chosen network topology and the quality of the used wireless links. A number of potential topology and schedule configurations exist that can fulfil application performance requirements. Furthermore, additional constrains may have to be applied when constructing the schedule which increases the problem complexity significantly. For example, it may be desirable in an industrial setting to reduce interference with neighbouring networks and, thus, transmission power settings may have to be considered.

In this paper we address the problem of finding the *found best* schedule for industrial process automation and control applications. In this work the *found best* schedule is the schedule which can support the required application performance in terms of delay and reliability constraints while minimising transmission power in order to reduce interference with neighbouring networks.

We split the deployment of the network into two phases: To obtain information about the quality of links, we start with a data gathering phase after nodes have been placed in the field. The information collected during this initial measurement is used to calculate a schedule which is then distributed onto the nodes before the network is put into regular operation.

The contributions of this paper are:

— *Data Gathering:* To find the best schedule it is necessary to provide a comprehensive set of measurement data from the deployment area. We describe tools and methods to collect this data.
— *Schedule Determination:* We describe the algorithms required to calculate the best schedule. As an exhaustive search is not feasible in practical deployments we describe a heuristic that reduces the necessary computational effort to practical levels.
— *Deployment Evaluation:* We evaluate the effectiveness of the proposed schedule determination methods in a practical deployment scenario; an oil refinery in Sines, Portugal. We furthermore present long-term experiments from an office environment.

Our experiments show that the proposed tools and methods for data gathering and schedule determination reduce transmission power (and thus interference with neighbouring networks) by up to 99.95 % (with up to 57.65 % reduced power consumption for transmitting packets) while at the same time data delivery reliability is improved by up to 1.5 %.

The focus of this paper lies on schedule determination during network deployment. However, even in relatively static environments conditions change and schedules do not remain valid for ever. Hence, we also discuss how the proposed methods can be integrated in a schedule management framework. Initial data collected at deployment time can be augmented during network operation and periodically this data set may be used to determine a new schedule that is then pushed into the network.

The paper is organised as follows. In the following subsection, we detail the practical use case and its features. In the next section we discuss related work before presenting

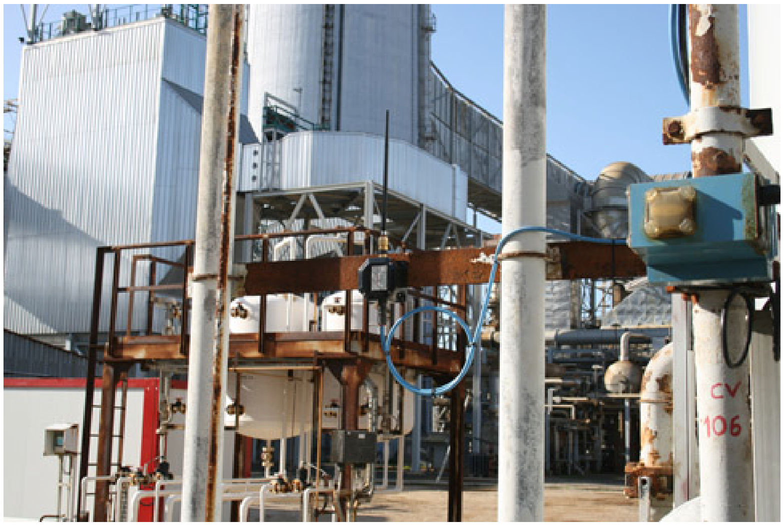


Fig. 1: Sensor Node in the Petrogal oil refinery in Sines, Portugal.

general aspects of schedule construction for industrial process monitoring and control applications in Section 3. Section 4 describes our tools and methods for link information gathering. Section 5 shows how this data is used to calculate a schedule. Here we describe as well a heuristic to improve calculation performance. Section 6 describes our evaluation in three different testbeds including our practical use case in the refinery. Section 7 discusses the schedule management framework and the paper concludes with Section 8.

1.1. Application Scenario

Typical industrial plants employ a significant number of sensors and actuators in the field that are conventionally connected to a control room using wires. In the Petrogal oil refinery [O'Donovan et al. 2014; Pöttner et al. 2011] in Sines, Portugal, 35000 sensors and actuators are deployed within 320 hectares (see Figure 1). In this specific plant, the sensors and actuators belong to one of the following three systems: indicatory, semi-automatic control and automatic control. Usually, most industrial plants have these kinds of systems to monitor and control their processes. Here, they are used for leakage detection, to measure the product flow and/or pressure in pipes, to measure temperature of burners or the fluid level of tanks.

The Petrogal oil refinery is exploring the possibility of replacing the existing wired sensor/actuator infrastructure by wireless networks. As it is not feasible to convert all 35000 sensors to wireless at once, it is the aim to convert sections of the fixed infrastructure associated with individual production processes to wireless. As a result, a very large number of small wireless networks (typically 6 to 24 nodes) connected to a fixed backbone infrastructure will emerge. Within each wireless network strict scheduling requirements apply as critical process control tasks are implemented.

The Petrogal oil refinery in Portugal employs an indicatory system in which the sensor readings are presented to operators in the control room. This allows for the monitoring of ongoing processes and to visualize the current state of the plant. For most sensors, set-points can be defined to notify the operator when certain sensor readings leave pre-defined bounds. For the indicatory system all samples should be delivered to the control room in time, but the system can tolerate a couple of lost values. More specifically, the delay may be up to 3 s with a delivery rate of 99 % for the samples. Each sensor is sampled every 1-2 s.

The semi-automatic control system is similar to the indicatory system while allowing the operators to issue actuation commands to actuators in the field. Based on sensor readings the operators may decide to modify certain aspects of the process and do so by changing actuator settings in the field. From a network perspective the information now flows in both directions whereas the acceptable delay, the reliability and the sampling rate is similar to the indicatory system.

The automatic control system is used to realize automated control loops within the refinery. Operators set parameters for the control loops but the actuation decision is taken automatically. The acceptable maximum delay is significantly below the indicatory system at 2 s to enable the system to fulfil its control tasks in time. Generally, the acceptable reliability and the sampling rate is comparable to the indicatory system.

In the refinery, nodes are attached to sensors in the field at predetermined locations. Since the sensors are part of the physical processes, the locations are determined by the process and cannot be altered (e.g., to benefit the network). However, additional relay nodes may be deployed if this is required for unobstructed operation of the network.

TDMA networks with exclusive slot usage are limited in size by the logical multi-hop topology that is employed. GinMAC, as used in our work at the refinery in Sines, has a TDMA slot length of 10 ms, thereby allowing 200 slots within a period of 2 s. Since multi-hop transmissions and retransmissions have to be accommodated within the maximum permissible delay, networks cannot grow to arbitrary size. Thus, due to this size limitation and the argument regarding conversion from wired to wireless networks discussed above, we assume that wireless networks for industrial monitoring and control are limited to 24 nodes each. To support the high number of sensors and actuators in typical industrial facilities, multiple networks of this size will be installed in parallel. Networks are organized by geographical proximity and in relation to the physical processes to which the sensors and actuators belong. This allows for sensors and actuators belonging to the same physical process to be within the same network.

The use of exclusive frequencies might help to enable the operation of networks which are in close proximity or even overlap. However, due to the large number of relatively small networks, there are too few frequency channels to ensure exclusivity across the entire deployment site. Instead the finite available channels will have to be reused spatially. Therefore, it is important to reduce the interference of each network by selecting proper transmission power levels to enable greater reuse of the available channels.

Another possibility to enable multiple collocated networks might be to distribute these networks to different time spans. However, this would have significant drawbacks resp. can be used as additional dimension only: Especially for larger industrial installations such as a refinery, there would not be enough time to do this for the complete area. Thus, interference reduction by transmission power control would be needed anyway. Further, there is a significant increase in complexity to coordinate between these networks and also to have time synchronization between them; this should be avoided from an engineering point of view.

2. RELATED WORK

Several projects in the field of wireless sensor networks provided solutions for improving transmission reliability and for reducing transmission power. However, most existing work has not investigated reliability and transmission power control in the context of time-critical data delivery. Next we discuss related work on link quality estimation and metrics available to describe link quality and existing work on transmission power control. We then discuss related work on reliability improvement in the context of time-critical data delivery and in particular schedule determination for time-critical data delivery.

Many link quality estimation techniques have been described in the literature (see [Baccour et al. 2012] for a survey). Depending on the used technique the suitability of links may be judged very differently. Thus, link quality estimation techniques are generally selected to support application specific requirements. The typical procedure is that links are monitored, measurement data is extracted and, finally, this measurement data is evaluated to produce a link quality estimation. Link monitoring can be performed in an active or a passive manner [Kim and Shin 2006]. Passive monitoring re-uses data transmissions to extract measurement data. In the case of active monitoring dedicated probe transmissions are used. In this work we use active monitoring mainly before network deployment resp. commissioning to produce a rich data base necessary for precise link quality estimation. After deployment active monitoring is only possible to some degree in a time-critical network as interference between probing and data delivery must be avoided.

Simple link quality estimators use directly (or the average of) a measured value such as received signal strength (RSSI), Packet Reception Rate (PRR) or Required Number of Packet Transmissions (RNP) (see [Srinivasan et al. 2006] and [Cerpa et al. 2005]). To improve performance of these simple estimators more complex processing algorithms such as Exponentially Weighted Moving Average [Woo and Culler 2003] or a Kalman filter [Senel et al. 2007] can be used. It is also possible to use abstract score-based link quality estimators as, for example, used in MetricMapi [Wang et al. 2007]. It has been shown that the aforementioned link quality estimators are not very useful in the context of time-critical applications as they do not capture the position of losses within the sequence of transmissions. Quality metrics measuring burst lengths of losses in a sequence of transmissions are here more useful. Examples of such metrics are the β-factor [Srinivasan et al. 2008] and the B_{max} notation introduced by Munir et al. [Munir et al. 2010] which we use within this work (Details are given in Section 3).

Link quality is strongly influenced by the selected transmission power [Correia et al. 2007]. Existing work has investigated fixed transmission power control (FTPC) where transmission power is fixed for individual nodes and dynamic transmission power control (DTPC) where nodes adapt transmission power dynamically. For example, Lin et al. [Lin et al. 2006] have shown that transmission power on individual links can be adjusted dynamically to keep link quality within desired bounds. Jeong et al. [Jeong et al. 2007] have shown that dynamic transmission power control does not achieve significantly better results than a fixed transmission power control. In the wireless sensor network domain transmission power control is generally used to reduce energy consumption. However, in other domains such as cellular radio systems transmission power control is used successfully to reduce interference with neighbouring networks [Rao 2007]. In our work we use FTPC with the aim to reduce interference with neighbouring networks.

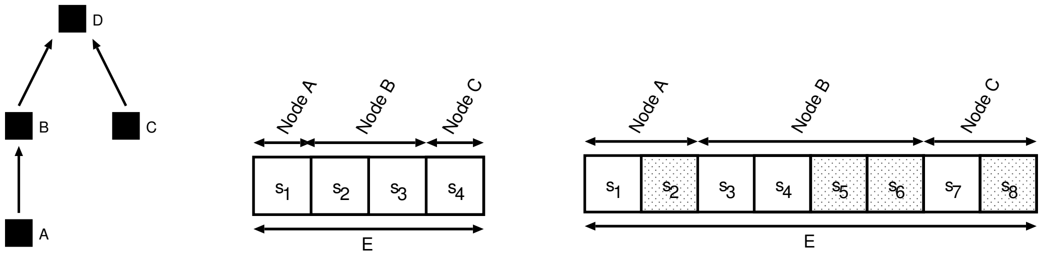
Existing solutions for reliable and timely data delivery in wireless sensor networks are generally constructed around a purpose-built medium access control protocol (MAC) (see [Suriyachai et al. 2012] for a survey). Some of the proposed solutions assume, unrealistically, that only links are included in a topology which do not experience any losses (for example, FTDMA [Chintalapudi and Venkatraman 2008], PEDAMACS [Ergen and Varaiya 2006]). Then, transmissions can be organised such that required delay bounds are respected. Other solutions such as MMSPEED [Felemban et al. 2006], Dwarf [Strasser et al. 2007], WirelessHART [HART Communication Foundation 2008], ISA 100.11a [International Society of Automation 2009] and IEEE 802.15.4e [IEEE 2008] send packets on multiple paths to compensate for losses. It is assumed that the provisioning of multiple transmission paths can compensate for all losses and under this assumption given delay bounds are met. Especially for WirelessHART, which is generally used for process automation and control applications as also considered in this paper, a number of methods have been proposed to determine transmission schedules under these assumptions. In [Saifullah et al. 2010], schedules are computed based on the earliest deadline of all packets. Reliability of the network is seen as a duty of the underlying layers via channel hopping and multi-path routing, whereas reliability guarantees cannot be given. The mechanism is extended in [Saifullah et al. 2011], but still packets are assumed to be transmitted in one slot, without the chance for retransmissions. The authors of [Han et al. 2011] construct a so-called reliable graph that represents the topology of the network. All nodes in the graph must have at least a certain number of outgoing links, to make multi-path routing possible. This approach also assumes, that multi-path routing and channel hopping are enough for a reliable network operation, but cannot give guarantees regarding reliability. Finally, [Zand et al. 2011] considers multiple traffic streams and also bursty traffic when calculating schedules for WirelessHART and ISA 100.11a networks but reliability is not considered during the computation. However, reliability can be ensured only if the properties of the underlying parts are known. Thus, to construct a system providing proper reliability assurances, the reliability characteristics of all links must be taken into account. GinMAC [Suriyachai et al. 2010] and Munir et al. [Munir et al. 2010] follow this approach and we build on this prior work.

A GINSENG [O'Donovan et al. 2014] system using GinMAC [Suriyachai et al. 2010] uses a worst-case link description using the B_{max} notation. The transmission schedule is determined before network deployment using this worst-case description. As a result, a valid schedule is determined but this schedule is not optimal as not all links may observe this worst-case behaviour. GinMAC contains a runtime mechanism called BurstProbe [Brown et al. 2011] that can be used to verify if the initially assumed worst-case link characteristic still holds. Our work extends this approach by characterising individual links in the deployment and by using this fine-grained information for schedule calculation.

Munir et al. [Munir et al. 2010] produce a schedule of packet transmissions for routing multiple data streams through a network while guaranteeing maximum latency for the individual streams. The approach is based on an initial measurement phase, in which a large number of packets are transmitted on each available link to characterize the burstiness (using the B_{max} notation) of the link. In the dimensioning phase, a centralized algorithm then selects the shortest route for each of the streams. This dimensioning already includes the anticipated retransmissions based on the burstiness of the included links. After calculating the route, the algorithm specifies the latency bound per stream that can be guaranteed in the network. After deploying the schedule onto the nodes, the network goes into the operational phase. Different to our approach, Munir et al. do not consider variable transmission power in order to reduce interference to other networks which is an important aspect in industrial deployments. Also, Munir use a 21 day measurement phase which is impractical in realistic industrial deployments. In our work, we investigate how the measurement phase can be reduced while still obtaining a network with sufficient reliability. Finally, we investigate a deployment in a real-world industrial setting and stability of a long-term deployment.

3. SCHEDULES FOR TIME-CRITICAL DATA DELIVERY

In order to detail our solution for determining the *found best* schedule we have to first summarise briefly how schedules may generally be constructed for industrial process monitoring and control applications. We have detailed some aspects of this basic schedule design in [Brown et al. 2011].



(a) Topology

(b) Schedule without re-transmission

(c) Schedule with re-transmission

Fig. 2: Simple topology and possible schedules for error free and lossy channels.

3.1. Scheduling for Timely Data Delivery

Consider the simple network topology given in Figure 2a. Assume nodes A, B and C have to deliver data with period T to the sink node D. In order to guarantee timeliness a TDMA schedule is applied. Each slot in the schedule accommodates the actual data transmission and a short acknowledgment from the receiver. In this paper, transmissions within a slot refer to both the original data packet and corresponding acknowledgment. If we assume that all nodes are in interference range of each other, the simple schedule such as the one shown in Figure 2b could be used. Node A transmits in slot s_1 to node B which uses slot s_2 to forward data from A to D; B uses slot s_3 to transmit its own data to D. Node C uses slot s_4 to transmit data to D. The resulting schedule $S = \{s_1, s_2, s_3, s_4\}$ has a duration (which we refer to as an *epoch*) of $E = |S| \cdot t = 4 \cdot D$ (with D being the slot length). Providing forwarding slots enables data from all nodes to be delivered within the epoch to the sink. We refer to the schedule as *valid schedule* if it allows us to deliver data within the required period T. The schedule is valid if the epoch is shorter than the period ($E \leq T$).

3.2. Scheduling for Reliable Data Delivery

The schedule is only valid in situations where all transmissions are successful. In a wireless environment error free channels are rare and capacity for potential retransmissions must be incorporated within the schedule. Figure 2c shows a schedule for the aforementioned simple topology which allows for one retransmission on each link for each transmission. The epoch length has now doubled to allow for reliable and timely data delivery on potentially limited lossy links. The schedule is valid if $E \leq T$ and if no more than every second transmission is erroneous. Given the harsh radio environment where some sensor networks operate it is a challenge to provision the correct number of retransmission slots in advance.

There are different methods available to describe link reliability. Common methods are Expected Transmission Count (ETX) or Packet Reception Rate (PRR). Using PRR gives a worst-case link reliability by a value P_{max} indicating that at least P_{max} transmissions out of x transmissions are successful. The problem with such a metric is that it does not capture the position of losses within the sequence of x transmissions. For example, the schedule allowing for retransmissions shown in Figure 2 is not valid if transmissions in two or more successive TDMA slots fail. P_{max} might be large compared to x indicating a good quality link. However, this might not be entirely true if losses appear in bursts.

It has been shown that burst lengths [Munir et al. 2010] are a much better metric to capture worst-case link reliability for networks that have to support time-critical data delivery. We define worst-case link reliability using the two values B_{max} and B_{min}: a

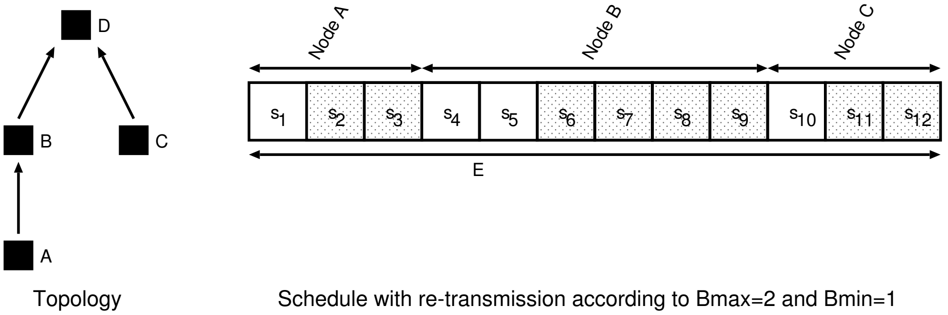

D
B
C
A
Topology
Node A
Node B
Node C
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
E
Schedule with re-transmission according to Bmax=2 and Bmin=1

Fig. 3: Topology and possible schedules for lossy channels with $B_{max} = 2$, $B_{min} = 1$.

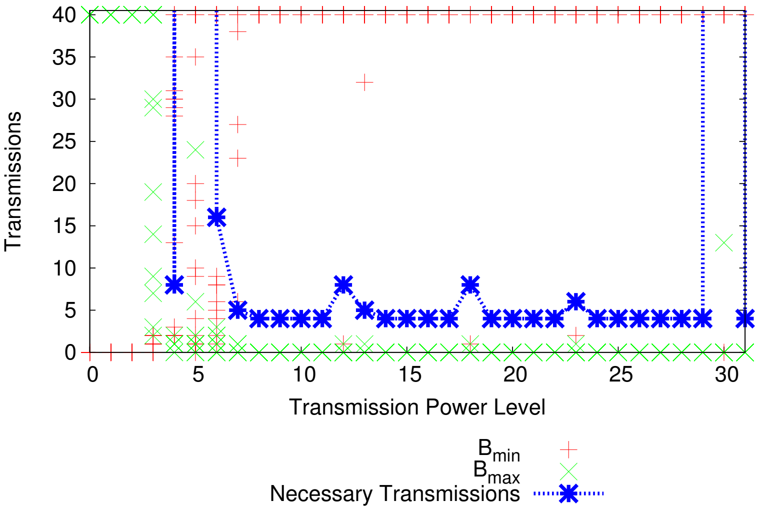


Fig. 4: Measured B_{max} and B_{min} values and necessary transmissions to deliver 4 data packets on a link used in our evaluation deployment in the refinery.

link has no more than B_{max} consecutive transmission errors and provides at least B_{min} consecutive successful transmissions between two error bursts. As we will show, it is possible to determine B_{max} and B_{min} during network deployment and to determine a schedule that can handle the *observed* worst-case. Figure 3 shows the schedule for the example topology for $B_{max} = 2$ and $B_{min} = 1$. Again, this schedule can only be used if $E \leq T$.

3.3. Burst Behaviour

When evaluating burst behaviour of links it is possible to consider the distributions of observed burst errors. The absolute values for B_{max} and B_{min} might be observed rarely. If a system can tolerate some packet loss it may be possible to dimension for a B_{max} and B_{min} that do not represent the rare worst-case observed on a link.

Figure 4 shows the recorded B_{max} and B_{min} values on a link in our evaluation deployment in the refinery at each available transmission power (we explain the deployment scenario in detail later in Section 6). Furthermore, the figure states how many transmissions would be necessary to deliver 4 data packets given the worst-case B_{min}

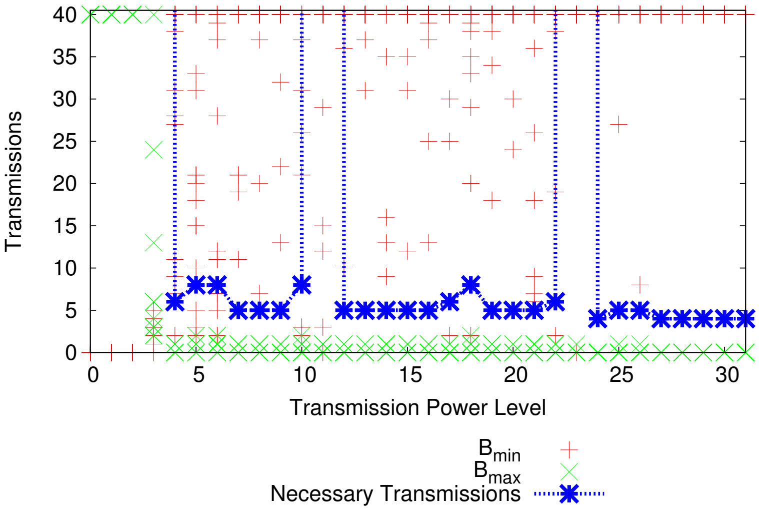


Fig. 5: Measured B_{max} and B_{min} values and necessary transmissions to deliver 4 data packets on a link used in our evaluation deployment.

and B_{max} values that have been measured for each transmission power. We use the number of transmissions for 4 packets as a way to assign a practical meaning to the abstract values for B_{min} and B_{max}. A sequence of 40 probe transmission is used 10 times for each transmission power level. For each probe sequence of 40 transmissions we determine B_{max} and B_{min} values. At the lowest transmission power level 0 $B_{min} = 0$ as no transmission succeeds; $B_{max} = 40$ since all transmissions fail. As transmission power increases burst sizes decrease leading to improved B_{max} values. At power level 31 no packets are lost and $B_{min} = 40$ $B_{max} = 0$. However, in the transitional area it can be seen that the burst size measurements in each probe sequence vary. For example, at transmission power level 5 the worst observed B_{max} is 24. However, this burst size is only observed once and the next measured burst has a length of only 7. In literature, this transitional zone is referred to as the “grey zone” [Zhao and Govindan 2003] in which it is not quite clear if a packet will be successfully received or not. In our work we avoid links which have a large B_{max}. We have found that links with a large B_{max} are not stable enough to be useful for time-critical data delivery as also observed in [Munir et al. 2010].

Figure 5 shows measured B_{max} and B_{min} values on another link. Here we can see that for nearly all power levels B_{max} is not larger than 1. In this case B_{min} varies due to the variable frequency of burst errors.

Link quality changes over time and so do burst sizes. However, the worst-case description of a link in terms of B_{max} and B_{min} is generally accurate over long time periods. Thus, it is possible to design a topology and schedule based on B_{max} and B_{min} which stays valid for a reasonable time period. Figure 6 shows how many transmission attempts would be necessary to transmit 4 data packets on the given links over a 24 hour period. We have selected 4 exemplary links at the same transmission power level to make the links comparable. For this figure we exemplarily use the highest transmission power level 31 although we do not expect it to be used in a calculated schedule. Later in the paper (see Section 5) we describe our approach to determine the appropriate transmission power per link.

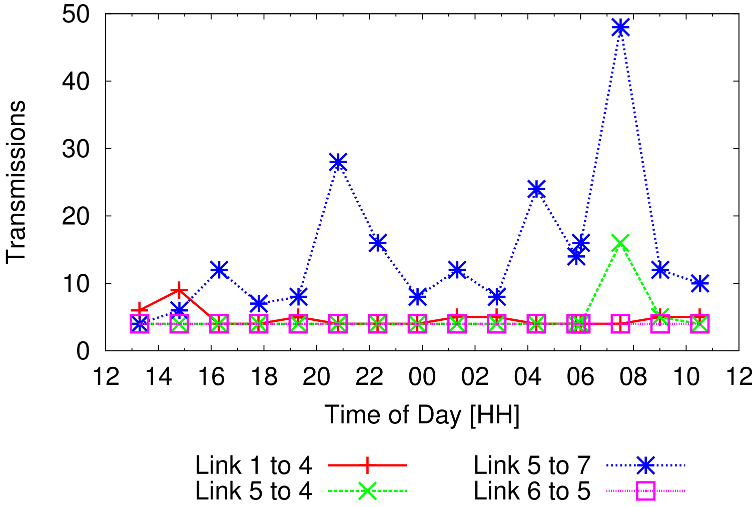


Fig. 6: Necessary transmissions to deliver 4 data packets for 4 different links at the highest transmission power level 31 in a 24 hour period.

From this figure, we learn that most links have a stable burstiness that can easily be expressed using worst-case numbers for B_{max} and B_{min}. The link from node 5 to 7 shows the largest variations in burstiness, but also this link can be characterised using the worst-case $B_{max} = 11$ and the worst-case $B_{min} = 1$. While this allows to accurately characterize the worst-case behaviour of the link in the observed period, the behaviour may change over a very long time period. Therefore, periodic topology and schedule re-construction is necessary after a certain period. We show in Section 6 how long schedules can be used in our deployment and analyse the time that is necessary to correctly capture the burst behaviour of a link.

3.4. Scheduling and Spatial Re-use

The epoch length can be reduced if not all nodes are within communication range of each other. In this case spatial re-use of TDMA slots is possible and the epoch length can be reduced.

However, it has to be noted that in industrial process automation and control scenarios it is common that not all nodes in a factory are grouped within one network. Generally a large number of small networks, whose sink nodes are connected via a cabled backbone infrastructure are deployed with each involved in controlling one particular production process aspect. Within each wireless network nodes are generally in interfering range of each other and spatial re-use is generally not possible. However, it is important to reduce interference range of a network in order to allow the operation of other nearby networks (of the same type).

The transmissions of one network can interfere with those of another that operates on the same and adjacent frequencies. Transceivers can tolerate interference as long as the received power of the interference is lower to a certain degree than that of signals of communicating peers. For the Chipcon CC2420 IEEE802.15.4 compliant transceiver [Texas Instruments Incorporated 2013] the co-channel rejection is specified as -3 dB (at -82 dBm with a PER of 1%), which means that a signal must be 3 db higher than any concurrent interference for it to be received with a PER of 1 %. Rejection values for adjacent and alternate channels are also specified in the

CC2420 datasheet [Texas Instruments Incorporated 2013]. Selecting a lower transmission power on each link of a network will minimise its interference on the area surrounding the network. This will enable nearby networks to reuse frequencies and adjacent frequencies closer together allowing for more networks.

For practical deployments it is important to reduce transmission power whenever possible to reduce the interference range of a wireless network and to allow frequency re-use to maximise the number of possible networks as outlined in Section 1.1. A schedule must be found which supports $E \leq T$; minimizes transmission power of each node while supporting B_{max} and B_{min} for the set transmission power level.

Again, as we will show, it is possible to determine B_{max} and B_{min} during network deployment for all transmission power levels and to take these measurements into account when determining the schedule used in the network.

3.5. Valid, Optimal and Best Schedules

3.5.1. Valid Topology. A schedule is constructed for a specific topology. A number of *valid* topology configurations may be available that can be considered for network operation. We define a topology configuration as valid if the following conditions are met:

- The topology forms a tree with the root node as sink.
- All nodes are included within the topology.
- All nodes in the topology are reachable within H hops from the sink.
- All nodes in the topology have no more than C children.

We constrain the topology in terms of tree depth H and node outdegree C as this reduces the search space dramatically. Furthermore, practical considerations often impose limitations on these values. For example, a node with a large number of children has to dedicate most of its processing time on forwarding tasks which leaves little room for sensing. Thus, a large out-degree C is not desirable. TDMA networks require tight time synchronisation with the sink being usually the time source. A large hop distance H is not desirable as the accurate propagation of synchronisation signals is difficult over many hops.

3.5.2. Valid Schedule. A schedule is considered *valid* if it satisfies the following requirements:

- The schedule is based on a *valid* topology.
- The schedule allows each node to transmit the necessary number of messages to the sink within one epoch to support the application.
- The schedule provisions sufficient retransmission slots according to B_{max} and B_{min} for the used links.
- The schedule's epoch length E is smaller than the delay requirement T.

We assume here that all nodes only require the transmission of a single message every epoch and that all messages have the same delay requirement T. However, we have found that this requirement is not too restrictive in practice as industrial settings have generally very uniform performance requirements[1].

3.5.3. Optimal Schedule. Multiple valid topologies with multiple valid schedules may exist for a given set of nodes. However, to reduce potential interference with neighbouring networks it is essential to select the *optimal* schedule. We define the optimal schedule in this context as the one that fulfils the following criteria:

[1] We deployed our system in an oil refinery where requirements are uniform to all deployed nodes associated with one production process. Typically only nodes associated with one production process are placed in one network.

— The optimal schedule is the schedule among the valid schedules which has the smallest energy signature ϵ.

For each schedule it is known how many transmission (including retransmission) slots each node uses and what transmission power is used in these slots. We define the energy signature ϵ of a schedule S as the sum of the transmission powers M_i set in each of the k transmission slots multiplied by the slot length D.

$$\epsilon = \sum_{i=0}^{k} M_i \cdot D \tag{1}$$

The rationale behind using the energy signature is to capture potential interference with a simple metric that is easy to use in practical deployments. We are aware that minimizing the defined energy signature is a very rough approximation. For example, individual transmission powers of nodes are not taken into account and the location of nodes in relation to other nodes they may interfere with is also not regarded. However, we believe that it is reasonable to reduce the overall output power of a network in order to reduce its interference capabilities.

A side effect of minimizing transmission powers is the reduction of energy consumption of nodes during operation. It has to be noted that the energy signature is not equivalent to the energy consumption. Energy consumption of nodes depends on slots used for transmission, slots used for reception and number of listening operations.

3.5.4. Best Schedule. Finding the “optimal” schedule requires complete knowledge of the burstiness of all links between the nodes. Since it is unlikely in practice that such knowledge will be achieved, the optimal schedule may not be contained in the set of valid schedules with respect to the data that has been gathered for a specific deployment.

In the context of this paper we refer to the best (i.e., ϵ-minimal) schedule that is contained in the set of valid schedules as the *best* schedule.

3.5.5. found best Schedule. When employing the heuristic, even the *best* schedule may not be found because certain links are excluded from the calculation altogether. We therefore refer to the best (i.e., ϵ-minimal) schedule that is contained in the set of valid schedules that are considered by the heuristic as *found best* schedule.

3.6. Summary

A schedule constructed as described is able to provide absolute guarantees regarding message delay. All messages that arrive at the sink arrive within the required time frame. However, messages might be lost and do not arrive if B_{max} and B_{min} on links is different than assumed at the time the schedule is constructed. Thus, it is important to provide methods able to determine these link characteristics correctly.

4. DATA GATHERING

In order to determine a topology and a corresponding schedule as described in the previous section it is necessary to collect link quality information in terms of B_{min} and B_{max} for all available links in the deployment. In a network with N nodes a total number of $N \cdot (N-1)$ node combinations are available. Combined with M transceiver power levels, data for a total number of L links has to be gathered:

$$L = N \cdot (N-1) \cdot M \tag{2}$$

Different power levels must be considered during data gathering as burstiness on links is heavily dependent on transmission power.

Each of the L links is tested by using a sequence of probe transmissions. The sender expects an acknowledgement from the receiver for each probe transmission. For each probe transmission the sender stores if the transmission was acknowledged or not. The result of the sequence of probe transmissions is stored as a *probe pattern* in the form of a sequence of “1” and “0” (“1” represents an acknowledged probe transmission and “0” stands for an unacknowledged probe transmission). The time delay between probe transmissions is determined by the slot length used in the schedule that is computed later. A sufficient number of P probe transmissions must be used to characterize each link. Generally, P must be large enough to capture B_{min} and B_{max} fully and to cover a sufficient period of time such that worst case link behaviour is actually observed. A formal method to determine a minimal P is not yet available. However, we found that relatively small values of P are sufficient to determine link characteristics in realistic deployments. For example, in [Brown et al. 2011] we have shown that $P = 8$ is sufficient to evaluate links that are typically considered to form topologies[2]. In our evaluation we use a probe pattern of length $P = 40$ (covering a probe duration of $400ms$ for a slot length of $D = 10ms$) which was found to be sufficient to capture link characteristics with enough detail [3].

Probe patterns on all links are collected during a data gathering phase after initial network deployment. Once the probe patterns are collected the schedule is determined and the network becomes operational. Since links tend to show a time-varying characteristic due to environmental influences such as interfering devices, etc. each link should be probed multiple times. This allows to capture the worst-case burstiness of a link. The amount of time that is necessary to capture the worst-case burstiness is scenario-dependant and cannot be generalized. We detail this aspect in Section 6.5.

Data gathering can be carried out by deploying a purpose build firmware on nodes which is then changed to the required application firmware prior to the network becoming operational. Alternatively, node firmware may be able to switch into a data gathering mode during application operation. Probe patterns collected by nodes can either be stored in flash for collection after data gathering is complete, or, as an alternative, nodes may form a bootstrapping network to communicate the collected probe patterns to a central location. To further improve resource usage for storage and communication of probe results it is also possible to transform probe patterns into a B_{min} and B_{max} value for each link on nodes directly after the probe pattern is recorded.

Link probing must be carried out in a collision free manner. Thus, to successfully measure link quality, methods for time synchronisation and a probe schedule must be specified. Time synchronisation can be achieved by periodically propagating a numbered beacon through the network which marks the start of a probe frame. Probe frames can be assigned statically to nodes based on the beacon number. A node starts (after a guard time has passed) a probe sequence if the probe frame is assigned to it.

To test all available links once, L probe frames are necessary. Taking slot duration D, number of probe transmissions P, number of nodes N and number of transmission power levels M into account the time necessary to gather all data for schedule calculation t_{probe} can be calculated as:

$$t_{probe} = D \cdot P \cdot L = D \cdot P \cdot N \cdot (N-1) \cdot M \tag{3}$$

[2] Longer probe sequences are necessary to evaluate links of low quality. However, such links would generally not be included in a topology as it would lead to very large schedules

[3] Using $P = 40$ allows to have a basic upstream schedule in parallel to be able to transmit the obtained patterns to the sink in the same epoch

Each node collects probe pattern as sequence of bits. For P probes, N nodes in the network and M power levels, each node collects b_{probe} bits of link quality information:

$$b_{probe} = P \cdot (N - 1) \cdot M \tag{4}$$

If probe patterns are transformed on nodes into B_{min} and B_{max} values the following data amount is required to represent probe results:

$$b_{probe\_converted} = 2 \cdot B \cdot (N - 1) \cdot M \tag{5}$$

Here B represents the size in bits required to store a B_{min} or B_{max} value. Obviously, on-node conversion of probe patterns into B_{min} and B_{max} values is only useful if $2 \cdot B < P$.

In our evaluation setting (see Section 6) we use a network with $N = 13$ nodes, $M = 32$ transmission power levels of a CC2420 transceiver and a slot duration of $D = 10ms$ and $P = 40$ probe transmissions. In this particular network data gathering can theoretically completed after $t_{probe} = 33.28min$. However, in practice probe sequences cannot be scheduled back to back as beacons to mark the start of probe sequences must be distributed as well. Each node has to store $b_{probe} = 1.920kB$ of data which can easily be stored on a node's flash memory or be transmitted wirelessly to a central point in the network. Thus, we omit in our evaluation on-node conversion of patterns into B_{min} and B_{max} values even though this would reduce storage requirements by a factor of 8 if we assume $B = 4$. A value of $B = 4$ would be sufficient if we assume that links characterised by $B_{max} > 8$ would not be used in the constructed topology.

5. SCHEDULE DETERMINATION

A schedule for the network can be calculated based on the information collected in the data gathering phase (see previous section). The aim here is to construct a schedule which is the *found best* schedule according to the definition given in Section 3.5.

We describe first how the *found best* topology and schedule can be determined. As this procedure is computationally infeasible for even small networks we provide as well a heuristic which reduces computational efforts to reasonable times while making concessions in terms of schedule quality.

5.1. Schedule Determination

To determine the schedule we apply the algorithm shown in Algorithm 1. Input to the algorithm are B_{min} and B_{max} values of all potential links in the network (stored in the variable *links*). The output of the algorithm is the *found best* schedule (and associated topology) as defined in Section 3.5 (stored in variable *best*).

As first step, a set containing all possible topologies using *links.calc()* is created. For each of these topologies we can now check if it is a valid topology using *t.check()* and, if this is the case, we can calculate a schedule for this topology using *s.calc()* and verify if this schedule is valid using *s.check()*. If the schedule is valid and better than the stored schedule we update the best solution. The algorithm terminates when all potential topologies are tested. Thus, the *found best* schedule and associated topology is found.

In the next paragraphs we describe some of the functions used in Algorithm 1 in more detail.

5.1.1. Creating the Topology Set. A set containing all possible topologies using *links.calc()* is created. A recursive algorithm is used that iterates through all outgoing links on all nodes to create all possible topologies of the network. The result is stored in *topologies*. For a network with N nodes and M power levels, this creates a total number of $((N - 1) \cdot M)^N$ combinations.

Algorithm 1 Find best topology and schedule for a given set of links

```
function CALC_BEST($links$)
    $best \leftarrow empty$
    $topologies \leftarrow links.calc()$
    for all $t$ in $topologies$ do
        if NOT $t.check()$ then
            continue
        end if
        $s = t.calc()$
        if NOT $s.check()$ then
            continue
        end if
        if $s.compare(best.s)$ then
            $best.s \leftarrow s$
            $best.t \leftarrow t$
        end if
    end for
    RETURN($best$)
end function
```

5.1.2. Topology Validity Check. *t.check()* in Algorithm 1 tests if the topology t is valid as defined in Section 3.5. It is tested if the topology provides all nodes a path to communicate with the sink and if the topology is free of cycles as only tree topologies rooted at the sink node are considered. It is also tested if the topology is not deeper than H hops and that no node has more than C child nodes. If all these tests have a positive outcome the topology is considered valid.

5.1.3. Schedule Calculation. *t.calc()* in Algorithm 1 is used to compute the schedule associated with a topology. The implementation of *t.calc()* is shown in Algorithm 2.

The schedule must allow all nodes to send one packet to the sink within one epoch. To create a schedule like this, the tree topology has to be traversed from bottom to top (see Figure 3). In Algorithm 2 variable *nodes* contains all nodes of the topology sorted according to their position in the tree topology. The recursive function *count_children()* determines how many child nodes are located in the tree underneath the specified node. For all these child nodes c and the node itself transmission slots according to B_{min} and B_{max} values must be provisioned. B_{min} and B_{max} values for the upstream link of node N are determined using the functions *get_bmin()* and *get_bmax()*. The number of necessary slots for the node to transmit a total of o packets is calculated as:

$$n_{slots} = \lceil \frac{o}{B_{min}} \rceil \cdot B_{max} + o \tag{6}$$

For sending packets, at least B_{max} slots have to be scheduled to overcome a potential sequence of burst losses. Afterwards, up to B_{min} slots can be schedule to transmit data packets. If only one packet should be transmitted, $B_{max} + 1$ slots have to be scheduled whereas multiple packets may benefit from a $B_{min} > 1$ so that they may be scheduled back to back without provisioning another B_{max} in between.

Using the function *add()* the number of required slots for this node is added to the end of the schedule s.

If the node has child nodes an additional downstream slot is added to the schedule. This slot is necessary for a downstream message that is used to enable time synchronisation and command messages to nodes. More slots could be added using B_{min} and

B_{max} values if reliable transmission to nodes is required (e.g. to implement control loops using actuators).

Algorithm 2 Calculate a schedule for a given topology

```
function CALC()
    $s \leftarrow empty$
    for all $n$ in $nodes$ do
        $c \leftarrow n.count\_children()$
        $B_{min} \leftarrow n.get\_bmin()$
        $B_{max} \leftarrow n.get\_bmax()$
        $o \leftarrow c + 1$
        $s.add(\lceil \frac{o}{B_{min}} \rceil \cdot B_{max} + o)$
        if $c > 0$ then
            $s.add(1)$
        end if
    end for
    RETURN(s)
end function
```

5.1.4. Schedule Validity Check. s.check() in Algorithm 1 tests if the computed schedule is a valid schedule and can be used. It is checked if the computed schedule has an epoch length smaller than the delay requirement T.

5.1.5. Determine the Best Schedule. s.compare(best.s) in Algorithm 1 compares the new calculated schedule with the already know best schedule. The energy signature ϵ is used to determine schedule quality as outlined in Section 3.5. If the new computed schedule outperforms the already stored solution the schedule and associated topology are stored as new best value.

5.2. A Heuristic for Schedule Determination

The complexity of the previously outlined algorithm for schedule determination is determined by the number of link combinations that have to be tested. In a network with N nodes and M power levels $((N-1) \cdot M)^N$ combinations are evaluated. Hence, the algorithm has a complexity of $O((M \cdot N)^N)$. Thus, even for small networks the determination of a schedule is computationally infeasible. For example, in our refinery testbed with $N = 6$ nodes and $M = 32$ power levels, $5.67h$ were required to calculate the schedule (see Section 5.1).

To reduce computational effort a heuristic is necessary. To reduce complexity we reduce the number of links that are included in the search space. Obviously, the heuristic may not be able to find the best schedule in all cases as not all link combinations are tested and the burstiness of some links may not be captured completely. To ensure that the heuristic obtains a good result it is essential to have a good strategy on which links to include in the search space.

We can reduce the number of links at each node by pruning links that are unlikely to be used by the schedule calculation because they have a bad reliability or require high transmission power.

In a first step, we look at the list of outgoing links of each node and remove bad links. Those are links on which no packets could be transmitted at all and links with a B_{max} value above a threshold $T_{B_{max}}$. This step ensures, that only links with reasonable reliability are used.

In a second step, we now sort the outgoing links ascending by transmission power, so that links with the lowest transmission power are the first in the list. Multiple links with the same transmission power are now sorted by the B_{max} value in ascending order. Now we have low-power links with high reliability in the front-most spots of the list. Finally, we sort the links by descending B_{min} to ensure, that we prefer links that can transfer many packets in sequence.

To reduce the number of link combinations, only the first T_L links of the list for each node are used. This reduces the number of outgoing links from $(N-1) \cdot M$ to T_L. The complexity of the algorithm is reduced from $O((M \cdot N)^N)$ to $O((T_L)^N)$. Essentially the input *links* for Algorithm 1 is reduced.

While the complexity of the algorithm using the heuristic still has an exponential dependency on the number of nodes N, this is only a theoretical problem. In practice, networks will not exceed 24 nodes.

In our evaluation we show, that with a value of $T_L = 5$ reasonably good schedules can be produced.

6. EVALUATION

Our main evaluation setup is at a real-world refinery production environment in Sines, Portugal. We only had time-limited access to this testbed as our experiments had to fit with refinery operations. To provide a longer-term evaluation we use a second existing testbed in an office environment at the University of Luebeck, Germany. Finally, we use a third existing testbed in an office environment at the University of Braunschweig in Germany to evaluate the impact of transmission power reduction on interference.

6.1. Heuristic Quality

To study the effects of increasing network size on the schedule calculation we analyse schedule construction assuming an increasing subset of nodes in the industrial deployment as active. Schedule calculation is carried out on a relatively powerful machine featuring 2 Intel Xeon E5520 CPU's with a total of 8 cores running at $2.27\ GHz$. Our schedule calculation is multi-threaded by dividing the amount of schedules and topologies that have to be calculated onto multiple threads. In this evaluation, we have used 6 threads in parallel running on the dataset acquired in the oil refinery in Portugal.

As established in Section 5.2, calculating the schedule without using a heuristic is computationally infeasible. In a relatively small network with $N = 6$ nodes and $M = 32$ power levels, it takes $5.67\ h$. When using the heuristic, the time can be brought down to $0.14\ s$ for the same network. The calculation time increases exponentially and for the target scenario with 13 nodes a calculation without heuristic is not feasible as network operation should commence shortly after the initial data gathering phase. The computation completes in $196.6\ s$ when using the heuristic algorithm.

The drawbacks of using the heuristic in terms of the quality of the schedule that is constructed must be evaluated. To evaluate its use we compare the schedule calculation with and without heuristic up to a network size of $N = 6$ nodes (6 was shown to be a feasible limit for acceptable calculation time).

As outlined in Section 3.5.3 we aim to construct schedules which minimize the energy signature ϵ. Thus, we compare the success of our heuristic in achieving this optimisation goal using the calculation without heuristic as comparison baseline. The results are shown in Table I.

As the heuristic limits the search space it is not possible to find the optimal solution in most cases and a schedule with a larger energy signature than necessary is selected. The schedules found using the heuristic have in average a $19.8\ \%$ larger energy signature. However, this drawback is compensated by the gain in computation time. Generally, but not necessarily in all cases, the best schedules also require less

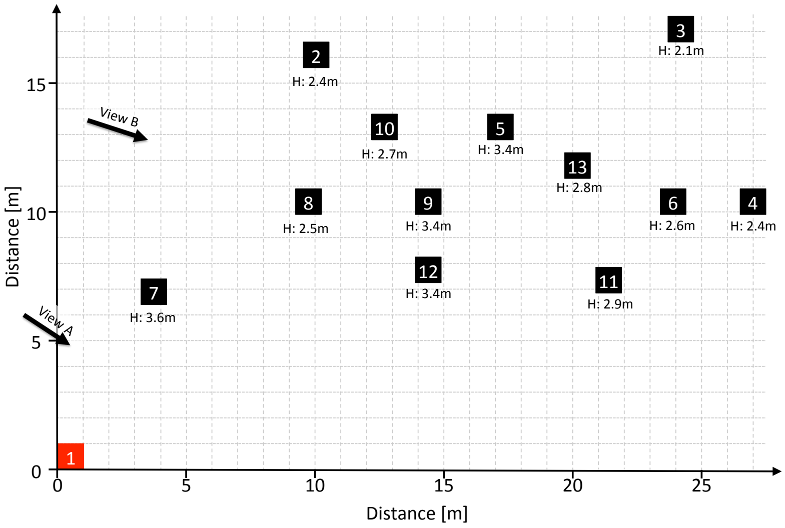


Fig. 7: Network layout with node height over ground (Node 1 is sink).

transmission slots. The results shown in Table I display this pattern; schedules found using the heuristic have more transmission slots and the energy signature is higher. It also has to be noted that schedules that are determined by any other means (i.e. manually) have poor performance in terms of data transport reliability. We will show this aspect in detail in the next section.

	without Heuristic		with Heuristic		
Nodes	Length [slot]	ϵ [μWs]	Length [slot]	ϵ [μWs]	Difference in %
2	3	0.55	3	0.55	0 %
3	5	1.21	6	1.39	15.3 %
4	7	1.85	12	2.57	39.0 %
5	11	2.36	14	2.83	20.3 %
6	14	3.06	18	3.81	24.5 %

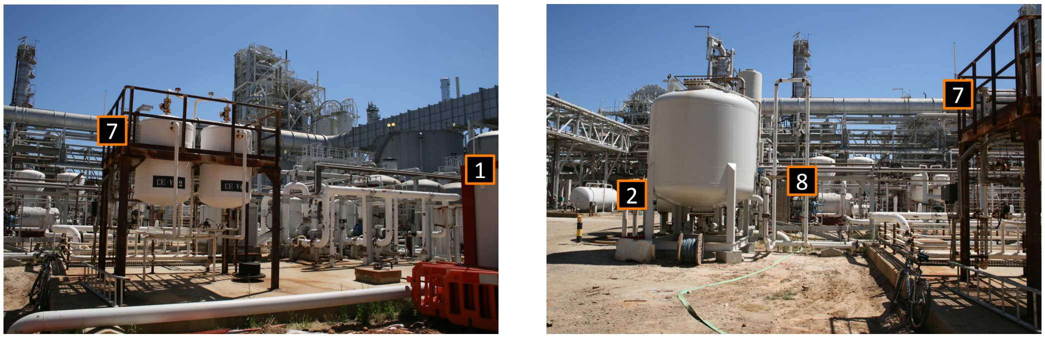
Table I: Schedule Length and Energy Signature ϵ

6.2. Industrial Deployment

For the experiments we use the wireless network shown in Figures 7 and 8. A number of sensors are connected to pipes and tanks which are used to measure flow, pressure and filling levels. Gathered data is collected and transferred to a sink which then feeds the data into the refinery back-end systems.

Twelve sensors plus one sink node are used to implement the desired production process in the deployment area. This specific process requires collecting data from each node once every second. Furthermore, it is necessary to ensure that data arrives reliably within a second at the sink node.

The challenge is to find a topology and associated schedule that fulfils the application requirements and minimizes interference. Transmission power of nodes should be reduced as much as possible in order to allow reuse of frequencies in other wireless networks used within the refinery.



(a) View A

(b) View B

Fig. 8: Deployment area

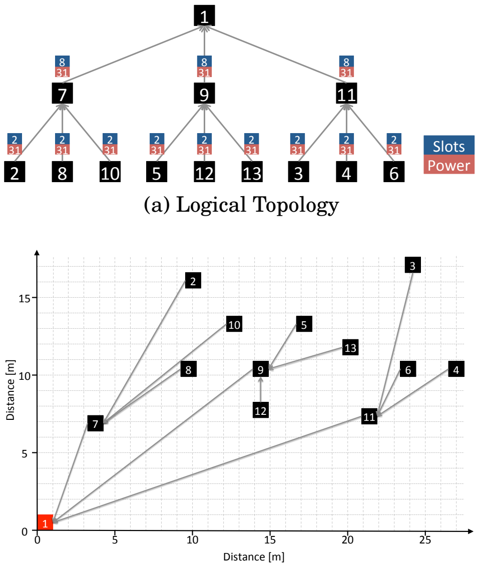
6.2.1. Data Gathering. To collect the necessary data for schedule determination we deploy dedicated measurement software on all nodes in the testbed. This Contiki-based measurement software establishes time synchronisation among nodes in order to ensure collision free probe scheduling. Burstiness on all available links in terms of B_{min} and B_{max} is measured using a sequence of 40 probe transmissions. With $N = 13$ nodes and $M = 32$ available power levels $N \cdot (N - 1) \cdot M = 4992$ links are tested. One measurement of all links is completed in $4992s$ (one probing cycle per epoch of $1\ s$).

The refinery is operational 24/7 and environmental differences between workdays are not experienced. We have therefore measured links over a duration of $23.4\ h$ (roughly one day). The measurement system allows us in this time span to probe all links 17 times, with probe times distributed homogeneously over the day. The collected transmission patterns on the available links are used as input for schedule calculation as described next. A discussion and analysis on the amount of probe data collected can be found in Section 6.5.

6.2.2. Schedule Quality. Without the automated schedule determination method described in this paper it is necessary to determine the schedule manually. A schedule is selected based on physical distance of nodes, visible obstacles in the communication path and knowledge from a limited number of link tests (testing PRR on some links intended to be used). A schedule may then be deployed and network performance is monitored for a while which then may lead to small schedule modifications until a state is reached in which network performance is acceptable. Obviously, such process is cumbersome, error prone and is not likely to lead to an optimized schedule. However, this process was used to setup networks in the evaluation scenario before the method described in this paper was available. Hence we use this *handpicked* schedule as baseline for comparison with the *computed* schedule using the described heuristic based on automated data gathering.

Fig 9 shows the handpicked topology and associated schedule information. A balanced tree is used and all nodes transmit with maximum power level of 31. Links are operated based on the observation that $B_{min} = 1$ and $B_{max} = 1$ is sufficient leading to, for example, 8 transmission slots on links in the first level of the tree (links from node $7, 9, 11$ to the sink) and to 2 transmission slots for all remaining nodes.

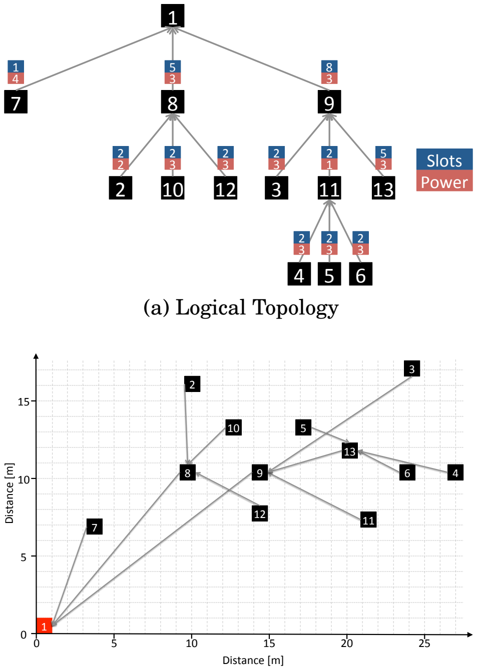
Fig 10 shows the schedule computed using the heuristic and the data collected during the automated gathering process. As it can be seen, a more unbalanced topology is selected in which transmission power levels are far lower than in the handpicked topology (for example, power level 3 for transmissions from node 9 to the sink). As a



(a) Logical Topology

(b) Physical Topology

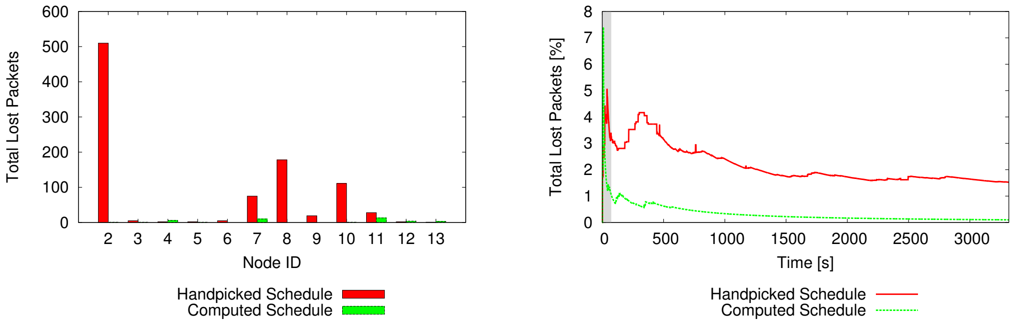
Fig. 9: Handpicked Topology: The topology used in the deployment when determining a schedule by hand (Node 1 is sink).



(a) Logical Topology

(b) Physical Topology

Fig. 10: Computed Topology: The topology determined by the heuristic using data collected in the automated data gathering process (Node 1 is sink).

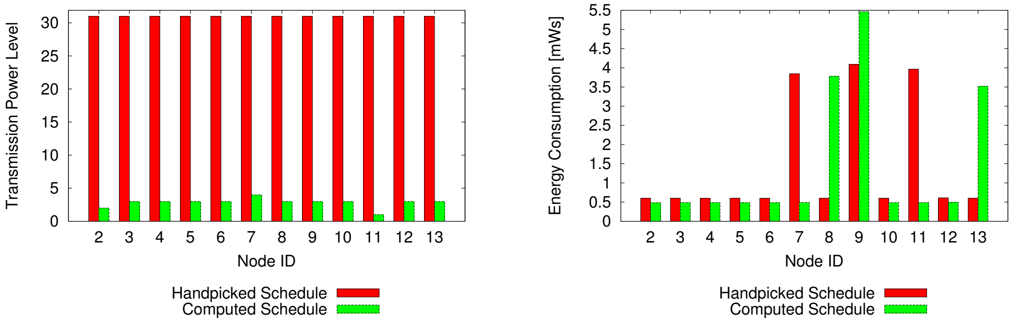


(a) Packet Loss per Node

(b) Packet Loss over Time

Fig. 11: End-to-end transmission reliability using handpicked and computed schedule for the oil refinery. Grey area marks initial starting phase, in which some nodes have been out of sync.

result different B_{min} and B_{max} are used on each link leading to a diverse allocation of transmission slots on each used link. For example, 5 slots are used on the link from node 8 to the sink while 8 slots are necessary on the link from node 9 to the sink.



(a) Transmission Power Setting per Node.

(b) Energy Consumption per Node

Fig. 12: Transmission Power and Energy Consumption per Node in the oil refinery.

6.2.3. Transmission Reliability. We tested both schedules in the deployment for a duration of 55 minutes. In that time each node reports a message once a second which must be delivered to the sink node within one second. Both schedules, handpicked and computed by the heuristic, ensure that if data is delivered, it is delivered within the requested time frame. However, some packets are lost as links in the deployment do not adhere to the assumed B_{min} and B_{max} characteristic that is the base for the schedule calculation. Figure 11a shows the end-to-end loss for each node.

As it can be seen from Figure 11a it is clear that the computed schedule is far superior to the handpicked schedule as less data losses (and therefore deadline misses) occur. Figure 11b shows the cumulative total from all nodes of the same packet losses as a distribution over time. It can be seen that losses tend not to be distributed equally over time. This result has to be expected as the calculated schedule is based on a detailed link quality measurements of all potential links. However, even the calculated schedule cannot prevent occasional data loss as link performance may drop temporarily below the worst-case performance observed in the data gathering phase. At the end of the measurement, the network using the handpicked schedules has lost $1.53\ \%$ of the data packets, while the same network with the computed schedule lost only $0.01\ \%$ of the packets. This is an improvement by $1.5\ \%$ in packet delivery ratio.

The schedule was found to be stable and supports the application requirements for the entire duration of the evaluation (55 minutes). Unfortunately we could not carry out a longer experiment to see when this schedule becomes unusable as access to the refinery testbed was time-limited. To analyse this aspect we carried out a long term experiment in another deployment which is detailed in the next section.

When a schedule becomes unusable a new schedule might have to be calculated and deployed. We discuss this issue in more detail in Section 7.

6.2.4. Interference and Energy Consumption. The schedule calculation aims to reduce the overall energy signature of the schedule while ensuring that data is transmitted reliably and timely. As a side effect, energy consumption of nodes as well as transmission power is reduced.

Figure 12a shows the transmission power settings for all nodes in the network when using the handpicked and computed schedule. As shown, transmission power of nodes and, thus, potential interference with other networks is drastically reduced when using the computed schedule. This is remarkable as network performance in terms of end-to-end transmission reliability is improved as well (see previous paragraph).

We have measured as well during the experiment the overall energy consumption in the network to be $17.328\ mWs$ with the handpicked and $17.156\ mWs$ with the computed schedule. The energy consumption difference per node for the two schedules is shown in Figure 12b. The savings in terms of energy consumption are not dramatically ($1\ \%$ is saved) as both schedules have a comparable length (34 transmission slots for the handpicked schedule while the computed schedule uses 36 transmission slots) and savings are achieved mostly due to reduced transmission powers.

Reducing transmission power helps in preserving energy on battery powered nodes. However, in the deployment context of refinery automation the concern is not necessarily energy consumption. Nodes have mostly a wired power supply as sensors and actuators are often unable to run on battery power. For example, a valve for closing an oil pipe cannot operate on a few AA batteries. Wireless communication is used to reduce complexity of the cabled infrastructure and to use wires purely for power distribution.

Interference is drastically decreased due to reduced transmission powers. At the maximum power level of 31, the CC2420 transceiver emits $1\ mW$ of output power at a current draw of $17.4\ mA$ [Texas Instruments Incorporated 2013]. The computed schedule uses power levels between 2 and 4 which translate into $0.001 - 0.007\ mW$ of output power at a current draw of $7.5-8.1\ mA$. Since range is dependent on the emitted output power, this shows that potential interference with neighbouring networks is dramatically reduced when using the computed schedule.

6.3. Long-Term Schedule Stability

To investigate a longer schedule deployment we use 9 nodes in the Wisebed [Coulson et al. 2012] installation at the University of Luebeck, Germany. Schedules should be able to stay valid for extended time periods. In this experiment the deployed schedule stays active for 308 hours (compared to 55 minutes as evaluated in the refinery deployment). We have configured 8 nodes as sensors and 1 node as sink node. The nodes are located in a building covering an area of roughly $10x21m^2$.

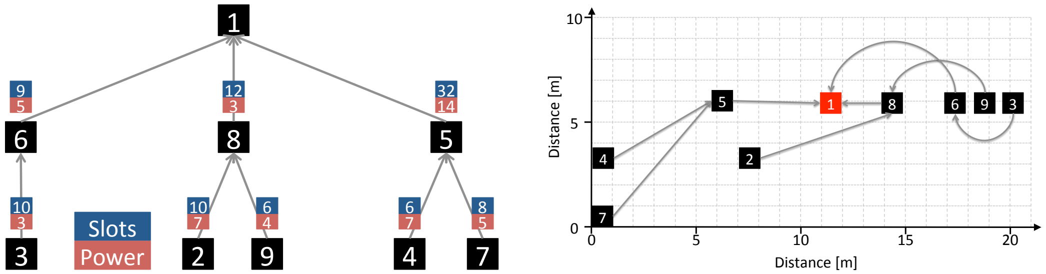
6.3.1. Data Gathering. We use the same approach for data gathering as the previous experiment. 40 probes per link were sent 110 times over a total duration of 152.5 hours or 6.35 days. We collect data starting at the start of a long weekend and continued over almost 3 full workdays to capture differences in the environment between these two periods. We analyse the temporal dependency of the environment in Section 6.5.

6.3.2. Schedule Quality. For the office environment we have used our algorithm to calculate the topology and schedule. The environment does not provide enough links with a B_{max} below our default threshold of 4. We therefore increased the B_{max} limit to $T_{B_{max}} = 10$ in order to find a suitable topology and schedule which includes all nodes. We show the outcome of the calculation in Figure 13. The most problematic link is from node 5 to the sink according to the initial measurements which needs 32 slots to ensure its packets can be forwarded correctly. It is also the link operating at a comparably high power level of 14 whereas all other links stay well below 10.

For this network we did not produce a handpicked schedule, as the goal is to show that a static schedule can be produced which is valid for a long time.

6.3.3. Schedule Reliability. We have tested the computed schedule for a total of 308 hours. The schedule has an epoch of $E = 1.5s$ and each node sends one packet per epoch to the sink.

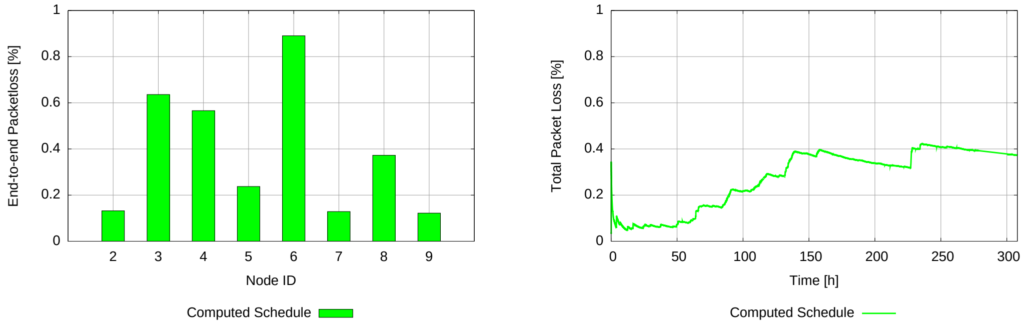
Figure 14b shows the total packet losses of the network as a distribution over time. As we see, the network never violates the application requirement of 99 % packet de-



(a) Logical Topology

(b) Physical Topology

Fig. 13: Computed Topology: The topology determined by the heuristic using data collected in the automated data gathering process (Node 1 is sink).



(a) Packet Loss per Node

(b) Packet Loss over Time

Fig. 14: End-to-end transmission reliability using computed schedule for the office environment in Luebeck.

livery rate. In Figure 14a we have broken down the end-to-end losses to the individual nodes. We see, that the losses are not distributed evenly between the nodes.

Similarly to the industrial deployment the schedule allowed for reliable data delivery for the entire duration of the evaluation (308 hours). This shows, that a static schedule can be used over extended periods if correctly provisioned even in dynamic environments such as office buildings with numerous sources of interference.

6.4. Interference Analysis

To investigate the impact of transmission power reduction on interferance an existing fixed Wisebed [Coulson et al. 2012] testbed in a university office building in Braunschweig, Germany was used. A network of 13 nodes of which one node is the sink was used with a 14th node as observer of interference levels at the fringe of the deployment area. All nodes are at fixed locations in an area of roughly $46x36m^2$. The node positions are shown in Figure 15.

For the deployment we construct a handpicked schedule using full transmission power on all links and a computed schedule based on gathered link measurement data. The schedules have both an epoch of $E = 1.8s$ and both report one packet per node per epoch. The handpicked schedule contained 78 slots whilst the computed schedule contained 170 slots. The transmision power of the handpicked schedule was set to maxi-

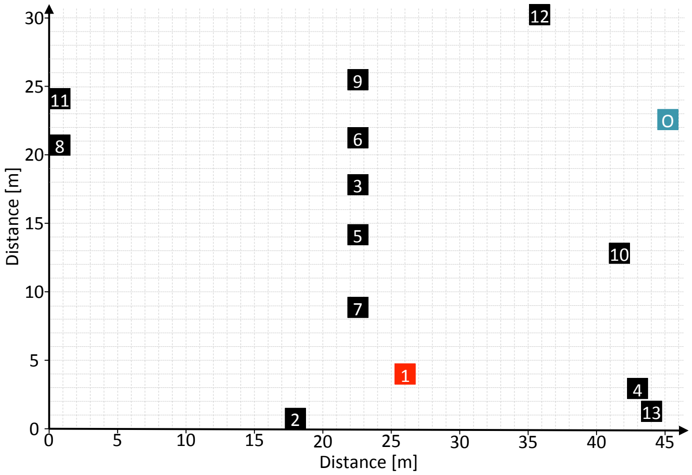


Fig. 15: Network layout in Braunschweig; Node 1 is sink, Node "O" measures interference.

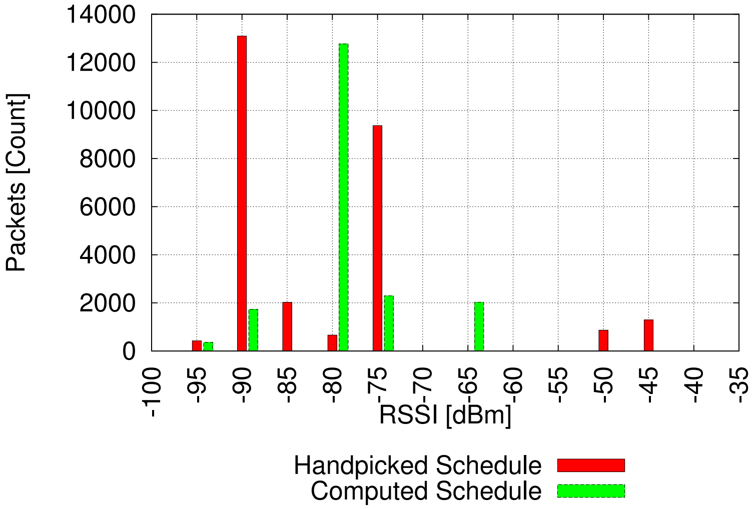


Fig. 16: Distribution of RSSI for the two schedules.

mum (level 31) whilst the computed schedule varied from level 3 to level 20 with an average of 7.9.

The interference measurement node was deployed on the edge of the network for a 2 hour time period to evaluate the observed interference of the two schedules. The node was deployed with a custom firmware that recorded transmission frequency in addition to RSSI. Overall the node received 27722 transmissions during the handpicked schedule test and 19183 during the computed schedule test. This was in contrast to the greater number of slots used for the computed schedule compared to the handpicked schedule. The lower number of transmissions received for the computed schedule can be attributed to the reduced transmission power used on each individual link. Figure 16 illustrates the distribution of RSSI of the received transmissions. For

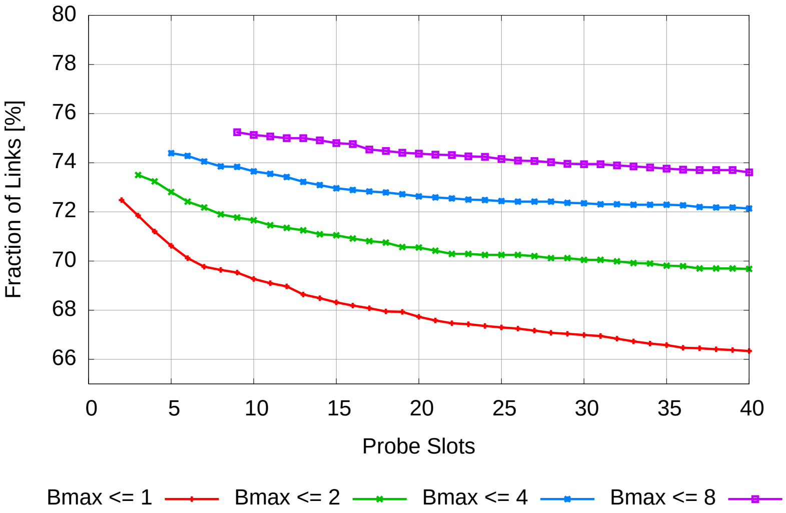


Fig. 17: Links below a certain Threshold $T_{B_{max}}$ over the number of probe slots.

the handpicked schedule there are three groupings of transmissions at -45 dBm, -75 dBm and -90 dBm. As the links within the handpicked schedule all use the maximum transmission power these groups must be from nodes at different distances from the sampling node. The figure shows that the computed schedule is less spread than the handpicked schedule with the majority of packets being at -80 dBm. Overall these results show that the interference of the computed schedule is lower than that of the handpicked schedule.

6.5. Analysis of Required Probe Effort

The worst-case B_{max} and B_{min} characteristic for each link is necessary to calculate a valid schedule. We refer to the worst-case determined from measurement data as the *observed worst-case* which usually differs from the *absolute worst-case* that we may never observe in measurements. The question is how many probe messages are required, the frequency and timing a link should be probed. In order to reduce measurement time before a network can become operational it is necessary to explore the volume of link probing that is required.

6.5.1. Determining the required number of probe slots. To evaluate how many probe slots are necessary to capture the burstiness of a link, we look at the dataset acquired in the oil refinery in Portugal during the 23.4 h data gathering period. Since we cannot know the real worst-case burstiness of a link, we assume that the observed worst-case is a good approximation of this value. This is a reasonable assumption as the experiment in Section 6.2 revealed only minimal packet loss. Our heuristic only considers links with a $B_{max} \leq T_{B_{max}}$ to avoid inclusion of bad quality links. Discarding more links due to more accurate probing will produce a schedule of higher quality.

We use the dataset gathered at the refinery which uses $P = 40$ probe transmissions and artificially limit this dataset to a subset where only the first P' probe transmissions are used. We then calculate the burstiness for the reduced probe pattern set and evaluate how many links are below the threshold $T_{B_{max}}$. With increasing P' fewer links will be available for topology construction as they are identified correctly as having a too high B_{max}.

The results in Figure 17 show, that with 2 probe slots more than 72 % of the links have a B_{max} less or equal to 1. When increasing the number of probe slots, the number of links considered by the heuristic first decreases drastically to 68 % of the links when using 19 probe slots. From then on, the slope is lower while reaching just above 66 % for 40 probe slots. This graph shows that already a very small number of probe slots identifies links correctly. Increasing the number of probe slots beyond 20 for a threshold of $T_{B_{max}} = 4$ is not justified.

We used a threshold of $T_{B_{max}} = 4$ in the refinery deployment. For this setting, using 5 probe slots leaves 74.1 % of the links available to the heuristic. Increasing the probe slots to 20 brings this number down by 1.76 % whereas the increase to 40 probe slots eliminates only another 0.49 % of links. If we aim to have 1 % of inaccurately classified links within the set used for schedule calculation, 14 probe slots (instead of the 40 we used) would have been sufficient.

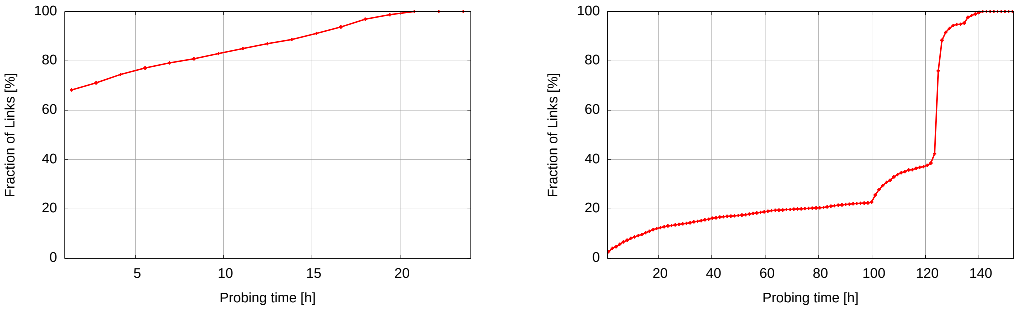
Accepting a certain inaccuracy of link assessment in terms of B_{min} and B_{max} may lead to problems when running a calculated schedule in the network when experiencing a worst-case burst loss. However, only if the burstiness of a link that is later chosen by the schedule calculation algorithm has been incompletely assessed (meaning that either B_{min} has been found to be higher than it really is or that B_{max} is found to be lower than it really is) this may have consequences during network operation. However, showing the effects of error margins when measuring B_{min} and B_{max} is out of scope for this work. We expect, that small errors will not have a significant effect on network operations since the network will most likely stay within the bounds the schedule was designed for even when seeing seldom worst-case loss situations.

6.5.2. Determining the required number of probe sequences. Probing a link is done by sending P probe packets within a so-called probing epoch. To ensure that the burstiness of each link is captured in the measurement, each link should be measured at different points in time to capture temporal variations. As shown in the previous paragraph, it is not helpful to simply use more probe slots. Since for our schedule determination algorithm (see Section 5) a link is defined by a sending node, the receiving node as well as the transmission power, the following results represent all transmission power levels available on the CC2420 radio chip.

The reasonable amount of probing epochs per link depends on the characteristics of the environment. Especially time-varying interference or background traffic has to be captured to ensure a reliable schedule for the dependable network. We call the time between the first probing epoch and the last probing epoch *probing time*. We expect, that in most environments time-varying patterns repeat; however this may not necessarily be the case for all environments. To produce a “complete” picture of the burstiness of a channel, a link has to be probed multiple times during one of those “repeat” cycles of the environment. Essentially we have to find out what the minimum time for a given environment is in which the links have to be probed to observe the worst-case burstiness.

For the datasets acquired in our two testbeds, we evaluate the minimum probing time that is necessary to find the worst-case burstiness that we observed during our measurements. However, we cannot ensure that our measurement has captured the absolute worst-case.

We have to determine the minimum probing time per link in which the worst-case burstiness is represented. This means, that for b probing epochs we look at all windows of length a with $a \leq b$ and decide whether the worst-case burstiness that we have observed for this link in the full dataset is present in the window a. We shift the window through all probing epochs and only if the worst-case burstiness is contained in all windows, we know the minimum number of probing epochs for this specific link. We



(a) Refinery Deployment Sines

(b) Office Deployment Luebeck

Fig. 18: Fraction of links for which the worst-case burstiness is contained in all windows of length probing time in the data sets from the two deployments.

then calculate the probing time by multiplying the number of probing epochs with the duration of a probing epoch.

Figure 18a shows the fraction of links in the oil refinery that show their worst-case burstiness in all windows with the probing time duration. The figure shows, that 68.21 % of the links in the refinery only need to be probed once to make sure to capture their worst-case burstiness. Since probing all links in the network once takes 4992 s, this value is the minimum probing time for those links. This result implies, that the links in the refinery are relatively stable. Increasing the probing time to allow each link to be probed multiple times also increases the amount of links that show their worst-case burstiness in the respective time frame. After 20.8 h of probing, all links have shown their worst-case burstiness at least once.

Figure 18b shows the same analysis for the office environment in Luebeck. Here a change in the environment is observed after 100 h, so that 37.67 % of the links show their final burstiness within a window of 120.64 h. Then the activity in the environment increases even more, so that finally 100 % of the links show their worst-case burstiness within 141.44 h.

The probing time must cover the time of network operation in which the worst-case is observed. As it is hard to predict when this event will be it seems to be a reasonable strategy to cover the duration in which the deployment area experiences the full spectrum of environmental changes. In the refinery this would be one work day. In an office environment several days may be needed as some days are busier than others.

6.6. Findings

The proposed schedule computation method using detailed gathering of link quality information and subsequent schedule calculation has two main benefits compared with the default method of using a handpicked schedule. First, it is possible to find schedules which provide much better end-to-end transmission reliability. Second, transmission power of nodes is lowered which helps to reduce interference with neighbouring networks.

Given the large number of sensors in this particular application case (the refinery uses 35000 sensors) it must be assumed that other networks will need to re-use the same transmission frequency and the proposed method allows us to run these networks in closer proximity to each other.

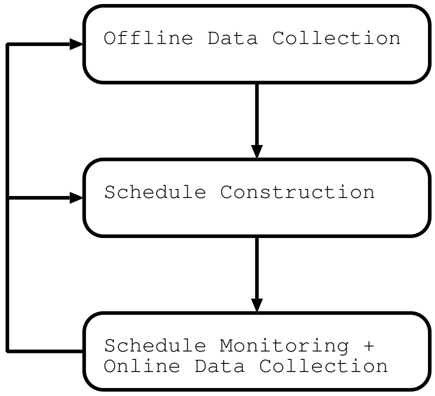


Fig. 19: Steps in schedule life-cycle management.

Given the number of networks that will be in operation to accommodate 35000 nodes it is obvious that automated methods for configuring schedules are necessary. Creating a handpicked schedule for all these networks is not feasible. The proposed method provides an automated mechanism which results in a network with good performance.

7. OVERALL FRAMEWORK

So far we described tools and methods for off-line schedule construction before a process automation and control network is made operational. As we have shown in our experiments the computed *found best* schedule stays valid for relatively long periods. Generally, environmental conditions in a factory environment do not change much on short time scales [Brown et al. 2011].

However, even though it is likely, it cannot be guaranteed that the environment is not subject to changes. Therefore, it is necessary to monitor the functionality of a deployed schedule. If it is detected that a schedule becomes infeasible (for example, the provisioned re-transmission slots are not sufficient anymore as B_{max} and B_{min} values for a link have changed) a new schedule must be computed and installed. It may also be the case that environmental conditions improve and a better schedule than the current one could be used. Thus, a new *found best* schedule should be computed and deployed.

In our previous work [Brown et al. 2011] we have presented a method named BurstProbe which allows us to monitor accurately B_{max} and B_{min} on links and, thus, the functionality of the employed schedule. Furthermore, BurstProbe allows us to collect measurements on B_{max} and B_{min} on links while the network is running. However, the number of samples collected by BurstProbe are far less than the samples collected by the off-line dimensioning method outlined in the previous sections. Therefore, data collected by BurstProbe can be used to correct a failing schedule but it is unlikely to obtain enough sample points to obtain an optimal (resp. a much better *found best*) schedule.

To manage a network used for time-critical data delivery a framework as illustrated in Figure 19 can be used.

Before deployment off-line data collection as described in this paper can be used to collect information necessary for schedule calculation. Then a valid and *found best* schedule based on this data can be constructed using the aforementioned heuristic. Thereafter the schedule is deployed and while the network is running the validity of the schedule is constantly monitored. For example, BurstProbe [Brown et al. 2011] can be used to achieve the goal of collecting link quality information while the network is in operation.

When it is determined that the schedule is invalid two correction methods can be applied.

The first method is to use the fresh but relatively sparse link quality data collected during network operation and to combine it with the rich but potentially outdated link quality data from the initial off-line measurements to calculate a new schedule. This method would ensure minimal downtime of the network as a service interruption would only occur for a brief moment when the new schedule is deployed.

The second method is to fall back to off-line measurements to ensure a complete and fresh base of link quality data is available to calculate a valid schedule which is as good as possible (i.e., the *found best* schedule). Yet, during measurement and schedule calculation the network cannot be used for its intended operation. Thus, it is important to reduce measurement time as discussed in this paper. However, a temporary stop of operations is not necessarily a problem as factories tend to schedule maintenance operations for equipment in the production process anyway. In such a case, schedule reconstruction can be aligned with general factory maintenance.

8. CONCLUSION

TDMA-based MAC-protocols can provide the time guarantees needed to support industrial process monitoring and control scenarios. However, such TDMA systems need schedules which allow for timely data delivery while taking channel characteristics into account.

In this paper we have presented a method for the construction of such schedules and evaluated it in a real industrial setting. We have detailed an algorithm to calculate the *found best* schedule and presented suitable heuristics to reduce the computational complexity. Our approach considers not only requirements in terms of data delivery latency and reliability, but also reduces transmission power which is important for industrial scenarios. We have also discussed a framework for schedule life-cycle management to constantly adapt to changing environmental conditions.

Our evaluation was carried out on a real-world wireless sensor network deployment in an oil refinery in Portugal. The results show that while the computation time can be reduced significantly by using the heuristic, the energy signature and the length of the epoch is increased. Still, the savings in computation time outweigh the increases.

Furthermore, we have compared the computed schedule (with heuristics) to the previously used handpicked version that was crafted using expert knowledge. The results show, that the computed schedule shows significantly increased reliability, reduced energy consumption and drastically reduced interference with neighbouring networks.

We have confirmed the findings for the refinery with a long-term experiment in a university office building in which the computed schedule operated for 308 hours within the application requirements. Additional tests in another testbed further showed that the computed schedule reduces interference significantly.

8.1. Future Work

In future work, the actual goal of the optimization can be further tweaked. In this paper, we minimize the total energy consumption of the whole network and the evaluation shows that this is a viable path to reduce potential interference with neighbouring networks. However, optimizing for a minimal energy consumption of each node can further reduce the potential interference.

Additional work is needed on the schedule life-cycle management, e.g. to determine strategies for schedule maintenance.

ACKNOWLEDGMENTS

This work has been partially supported by the European Commission under the contract FP7-ICT-224282 (GINSENG) and INFSO-ICT-317826 (RELYonIT).

REFERENCES

BACCOUR, N., KOUBÂA, A., MOTTOLA, L., ZÚÑIGA, M. A., YOUSSEF, H., BOANO, C. A., AND ALVES, M. 2012. Radio link quality estimation in wireless sensor networks: A survey. *ACM Trans. Sen. Netw. 8,* 4, 34:1–34:33.

BROWN, J., MCCARTHY, B., ROEDIG, U., VOIGT, T., AND SREENAN, C. J. 2011. BurstProbe: debugging time-critical data delivery in wireless sensor networks. In *Proceedings of the 8th European conference on Wireless sensor networks*. EWSN'11. Springer-Verlag, Berlin, Heidelberg, 195–210.

CERPA, A., WONG, J. L., POTKONJAK, M., AND ESTRIN, D. 2005. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. MobiHoc '05. ACM, New York, NY, USA, 414–425.

CHINTALAPUDI, K. K. AND VENKATRAMAN, L. 2008. On the design of mac protocols for low-latency hard real-time discrete control applications over 802.15.4 hardware. In *Proceedings of the 7th international conference on Information processing in sensor networks*. IPSN '08. IEEE Computer Society, Washington, DC, USA, 356–367.

CORREIA, L. H., MACEDO, D. F., DOS SANTOS, A. L., LOUREIRO, A. A., AND NOGUEIRA, J. M. S. 2007. Transmission power control techniques for wireless sensor networks. *Computer Networks 51,* 17, 4765 – 4779.

COULSON, G., PORTER, B., CHATZIGIANNAKIS, I., KONINIS, C., FISCHER, S., PFISTERER, D., BIMSCHAS, D., BRAUN, T., HURNI, P., ANWANDER, M., WAGENKNECHT, G., FEKETE, S. P., KRÖLLER, A., AND BAUMGARTNER, T. 2012. Flexible experimentation in wireless sensor networks. *Commun. ACM 55*, 82–90.

ERGEN, S. AND VARAIYA, P. 2006. Pedamacs: power efficient and delay aware medium access protocol for sensor networks. *Mobile Computing, IEEE Transactions on 5,* 7, 920–930.

FELEMBAN, E., LEE, C.-G., AND EKICI, E. 2006. Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks. *Mobile Computing, IEEE Transactions on 5,* 6, 738–754.

HAN, S., ZHU, X., MOK, A., CHEN, D., AND NIXON, M. 2011. Reliable and Real-Time Communication in Industrial Wireless Mesh Networks. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011 17th IEEE*. 3 –12.

HART COMMUNICATION FOUNDATION. 2008. HART Communication Protocol Specication, HCF SPEC 13 Revision 7.1. Available from the HART Communication Foundation.

IEEE. 2008. Wireless Personal Area Networks: Proposal for Factory Automation. IEEE Proposed Standard 802.15.4-15/08/0571r0.

INTERNATIONAL SOCIETY OF AUTOMATION. 2009. ISA-100.11a-2009 Wireless systems for industrial automation: Process control and related applications. Available from the International Society of Automation.

JEONG, J., CULLER, D., AND OH, J.-H. 2007. Empirical analysis of transmission power control algorithms for wireless sensor networks. In *Networked Sensing Systems, 2007. INSS '07. Fourth International Conference on*. 27–34.

KIM, K.-H. AND SHIN, K. G. 2006. On accurate measurement of link quality in multi-hop wireless mesh networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. MobiCom '06. ACM, New York, NY, USA, 38–49.

LIN, S., ZHANG, J., ZHOU, G., GU, L., STANKOVIC, J. A., AND HE, T. 2006. Atpc: adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*. SenSys '06. ACM, New York, NY, USA, 223–236.

MUNIR, S., LIN, S., HOQUE, E., NIRJON, S. M. S., STANKOVIC, J. A., AND WHITEHOUSE, K. 2010. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IPSN '10. ACM, New York, NY, USA, 303–314.

O'DONOVAN, T., BROWN, J., BÜSCHING, F., CARDOSO, A., CECELIO, J., DO O, J., FURTADO, P., GIL, P., JUGEL, A., PÖTTNER, W.-B., ROEDIG, U., SA SILVA, J., SILVA, R., SREENAN, C., VASSILIOU, V., T. VOIG AND, L. W., AND ZINONOS, Z. 2014. The GINSENG System for Wireless Monitoring and

Control: Design and Deployment Experiences. *ACM Transactions on Sensor Networks (TOSN) 10,* 1. accepted for publication.

PISTER, K. S. J. AND DOHERTY, L. 2008. TSMP: Time Synchronized Mesh Protocol. In *International Symposium on Distributed Sensor Networks (DSN)*. IASTED, Orlando, Florida, USA.

PÖTTNER, W.-B., WOLF, L., CECÍLIO, J., FURTADO, P., SILVA, R., SILVA, J. S., SANTOS, A., GIL, P., CARDOSO, A., ZINONOS, Z., DO Ó, J. M., MCCARTHY, B., BROWN, J., ROEDIG, U., O'DONOVAN, T., SREENAN, C. J., HE, Z., VOIGT, T., AND JUGEL, A. 2011. Wsn evaluation in industrial environments first results and lessons learned. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*. DCOSS '11.

RAO, A. 2007. Reverse link power control for managing inter-cell interference in orthogonal multiple access systems. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. 1837–1841.

SAIFULLAH, A., XU, Y., LU, C., AND CHEN, Y. 2010. Real-Time Scheduling for WirelessHART Networks. In *Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*. RTSS '10. IEEE Computer Society, Washington, DC, USA, 150–159.

SAIFULLAH, A., XU, Y., LU, C., AND CHEN, Y. 2011. End-to-End Delay Analysis for Fixed Priority Scheduling in WirelessHART Networks. In *Proceedings of the 2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*. RTAS '11. IEEE Computer Society, Washington, DC, USA, 13–22.

SENEL, M., CHINTALAPUDI, K., LAL, D., KESHAVARZIAN, A., AND COYLE, E. 2007. A kalman filter based link quality estimation scheme for wireless sensor networks. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*. 875–880.

SREENAN, C., SA SILVA, J., WOLF, L., EIRAS, R., VOIGT, T., ROEDIG, U., VASSILIOU, V., AND HACKENBROICH, G. 2009. Performance control in wireless sensor networks: the GINSENG project - [Global communications news letter]. *Communications Magazine, IEEE 47,* 8, 1 – 4.

SRINIVASAN, K., DUTTA, P., TAVAKOLI, A., AND LEVIS, P. 2006. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*. SenSys '06. ACM, New York, NY, USA, 419–420.

SRINIVASAN, K., KAZANDJIEVA, M. A., AGARWAL, S., AND LEVIS, P. 2008. The β-factor: measuring wireless link burstiness. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. SenSys '08. ACM, New York, NY, USA, 29–42.

STRASSER, M., MEIER, A., LANGENDOEN, K., AND BLUM, P. 2007. Dwarf: delay-aware robust forwarding for energy-constrained wireless sensor networks. In *Proceedings of the 3rd IEEE international conference on Distributed computing in sensor systems*. DCOSS'07. Springer-Verlag, Berlin, Heidelberg, 64–81.

SURIYACHAI, P., BROWN, J., AND ROEDIG, U. 2010. Time-Critical Data Delivery in Wireless Sensor Networks. In *Distributed Computing in Sensor Systems*, R. Rajaraman, T. Moscibroda, A. Dunkels, and A. Scaglione, Eds. Lecture Notes in Computer Science Series, vol. 6131. Springer Berlin / Heidelberg, 216–229.

SURIYACHAI, P., ROEDIG, U., AND SCOTT, A. 2012. A survey of mac protocols for mission-critical applications in wireless sensor networks. *Communications Surveys Tutorials, IEEE 14,* 2, 240–264.

TEXAS INSTRUMENTS INCORPORATED. 2013. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. http://www.ti.com/lit/gpn/cc2420.

WANG, Y., MARTONOSI, M., AND PEH, L.-S. 2007. Predicting link quality using supervised learning in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev. 11,* 3, 71–83.

WOO, A. AND CULLER, D. 2003. Evaluation of efficient link reliability estimators for low-power wireless networks. Tech. Rep. UCB/CSD-03-1270, EECS Department, University of California, Berkeley.

ZAND, P., CHATTERJEA, S., KETEMA, J., AND HAVINGA, P. 2011. D-SAR: A Distributed Scheduling Algorithm for Real-time, Closed-Loop Control in Industrial Wireless Sensor and Actuator Networks.

ZHAO, J. AND GOVINDAN, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*. SenSys '03. ACM, New York, NY, USA, 1–13.