Estimating Packet Reception Rate in Noisy Environments

James Brown and Utz Roedig
School of Computing and Communications
Lancaster University, Lancaster, United Kingdom
{j.brown,u.roedig}@lancaster.ac.uk

Carlo Alberto Boano and Kay Römer
Institute for Technical Informatics
Graz University of Technology, Graz, Austria
{cboano, roemer}@tugraz.at

***Abstract*—For the design of dependable and efficient wireless sensor networks it is essential to estimate the achievable packet reception rate (*PRR*) in the deployment environment. Making such estimation is not trivial as packet delivery success depends on the level of interference present in the deployment area. In this work we show that it is possible to obtain a meaningful representation of the expected interference levels at the target location by measuring the probability distribution function of idle period lengths, and use this to estimate *PRR* before network deployment. We show how a probability distribution function of idle period lengths can be measured using off-the-shelf sensor nodes. We illustrate how to exploit this methodology to estimate *PRR* in dependence of the used packet length, and show that relatively short measurement periods provide enough data to obtain accurate predictions. We carry out an extensive experimental evaluation showing that Wi-Fi interference can be captured using this method which allows *PRR* predictions in such Wi-Fi interference setting with an average error of only 3.2%.**

Keywords*—Dependability, Packet Reception Rate (**PRR**), Packet Size, Probability Density Function, Radio Interference, Reliability, Wireless Sensor Networks.

I. Introduction

For the design of dependable and efficient Wireless Sensor Networks (WSNs) it is essential to estimate achievable Packet Reception Rates (*PRR*) in the deployment environment. Their knowledge can be used both to fine-tune communication protocols and to evaluate if network performance is sufficient to support a given application. For example, one can employ *PRR* estimations to customise a TDMA protocol such that sufficient spare capacity is provided for potentially needed retransmissions. Similarly, these estimations can be used to calculate end-to-end data delivery rates, an essential step to judge if an application can meet its performance requirements.

Besides the used packet size, achievable *PRR* depends to a large extent on the presence of radio interference in the deployment area as WSNs operate in license-free ISM bands and share the radio spectrum with other wireless technologies. This problem is especially relevant in the 2.4 GHz frequency space as wireless sensor nodes need to coexist with IEEE 802.11 (Wi-Fi) devices which transmit at higher power levels [1]. Estimating the achievable *PRR* is not trivial as it depends on the specific interference patterns at the network deployment site. Some previous work estimated the *PRR* by making general assumptions regarding the nature of interference expected at the target area [2]. Such methods, however, are often inaccurate as the actual encountered interference deviates from the chosen general interference model. Other previous work made use of test measurements on links to estimate achievable *PRR* in a later deployment [3]. However, such techniques are tailored to specific protocols and results cannot be generalised.

In this paper we present a novel measurement-based method to estimate achievable *PRR* for specific deployment areas based on the characteristics of interference. We use wireless sensor nodes to capture the interference patterns at a given location by measuring how long the channel remains idle and by computing the corresponding idle distribution. We define an *idle period* as a time interval in which the received signal strength (RSS) of a radio transceiver remains below a given threshold R_{Thr}, indicating that no harmful interfering source is active. We then measure the Probability Distribution Function (PDF) of idle period lengths, referred to as *IDLE-PDF*. As we will show, the *IDLE-PDF* can be captured efficiently using resource constrained motes, and can be used to estimate *PRR* with very high accuracy. We further present an extensive evaluation of the proposed method focusing on two different aspects.

First, we analyse the cost of obtaining the *IDLE-PDF* using off-the-shelf sensor motes and show that, with a surprisingly short measurement duration, we can obtain a sufficiently detailed interference model. Second, we analyse the *PRR* prediction accuracy of the proposed prediction method using different interference scenarios. We show that the proposed *PRR* prediction method has a high accuracy: estimated *PRR* and measured *PRR* indeed differ on average by only 3.2%. The contributions of the paper are specifically:

- ***PRR* estimation using the *IDLE-PDF***: We introduce the *IDLE-PDF* as efficient metric for capturing detailed interference patterns. We also describe methods for *PRR* estimation based on the *IDLE-PDF*.
- **Measurement tool**: We present a tool to measure the *IDLE-PDF* based on off-the-shelf sensor node hardware. We show that the tool is able to capture detailed interference data with little storage requirement.
- **Evaluation of *PRR* estimation**: We provide an evaluation of the proposed approach in environments with different interference patterns.

The next section describes related work. In Section III we give the theoretical background of our work: we first give a formal definition of the *IDLE-PDF* and we then describe how this distribution can be used to estimate *PRR*. In Section IV we discuss how the *IDLE-PDF* is captured in a deployment area and we describe the dependency between measurement effort and measurement accuracy. In Section V we analyse the efficiency and accuracy of the proposed method using several interference scenarios. Section VI concludes the paper.

II. Related Work

The aim of our work is to measure interference *before* deployment and to use the measurement for *PRR* prediction.

There has been a vast body of work on interference measurement. However, the existing work generally does not aim to measure pre-deployment interference to predict achievable *PRR*. For example, existing work has addressed interference measurement for the purpose of interference classification [4], [5], to generate realistic interference for testbeds [6], and to select transmission channels in interference scenarios [7], [8].

A small body of work has similar aims to our work which we discuss in detail in the next paragraphs.

Huang et al. [2] have carried out a statistical analysis of interference traces and presented a generic model that characterizes the white spaces in Wi-Fi traffic. They use this model to estimate *PRR* in dependence of packet size and use this information to schedule transmissions such that delivery ratio is maximised, following the idea by Chowdhury and Akyildiz [9]. Differently from our approach, these two works rely on a generic interference model, whereas we use interference information collected at the deployment area to construct a specific interference model that is not bound to a specific technology (e.g., Wi-Fi).

Shariatmadari et al. [10] have proposed a method for *PRR* estimation based on interference measurements. It is assumed that data on a link will be transmitted with a fixed rate. A receiver measures periodically (using the expected data transmission frequency) the observed interference. The measurement is used to estimate the achievable *PRR*. The method is used to rank channels using *PRR* as link quality metric. This work differs from ours in many aspects. First, the interference measurement is not based on probability distributions. Second, the estimation technique requires assumptions regarding traffic patterns used in the network. Third, there are no guidelines about how long before deployment the interference should be characterised.

Pöttner et al. [3] characterise an interference environment by values B_{min} and B_{max}, and send a trail of test packet transmissions on a link. B_{max} describes the largest number of subsequent transmission failures while B_{min} describes the maximum number of subsequent successful transmissions. The two values can be used to configure a communication protocol such that one can give transmission guarantees for the measured interference. Similar to our work, interference patterns are determined by measurement carried out before deployment. However, the recorded interference patterns are only useful to configure protocols using transmission spacing and packet lengths as used in the measurements (i.e., specific TDMA protocols). The method presented in this paper, instead, follows a more general approach.

In this work we also describe a method for updating at runtime the interference measurement collected before the deployment. This is necessary as the interference environment may change over time. The related work outlined previously does not provide this feature and relies on the assumption that the interference does not vary significantly over time. A notable exception is the work by Brown et al. [11] which describes a method to update initially measured B_{min}/B_{max} measurements while the sensor network application is running.

III. Packet Reception Rate and Interference

In this section we give a definition of the *IDLE-PDF* and we describe how this distribution can be used to estimate *PRR*. First we describe a closed form solution to compute the *PRR* from the *IDLE-PDF*. As the closed form solution has its limitation with arbitrary shaped *IDLE-PDF* distributions we then describe a solver based on Monte Carlo simulation.

A. *The* IDLE-PDF

Interference levels can be measured by sampling the energy level in a transmission channel over time. If the sampled energy level is above a given threshold R_{Thr}, a packet transmission would be destroyed by concurrent activities in the frequency channel. This is an approximation of the interference process but is a reasonable assumption for the work presented in this paper (as shown by our evaluation). Thus, interference can be represented as a sequence of idle (free channel) and busy (ongoing activity in the medium) periods of different length. Using such a recorded interference trace it is then possible to analyse success rates of packet transmissions. For example, to estimate *PRR*, a number of transmissions can be randomly placed in the recorded interference trace and the success or failure of these transmissions can be evaluated. Unfortunately, due to the volume of data it is not possible to store reasonably long interference traces of this type on practical measurement systems.

To reduce the required storage space for interference traces we decide not to store the actual sequence of idle and busy periods and their respective length. Instead, we record the distributions of observed busy and idle period lengths. This measurement does not allow us to reproduce the exact measured interference trace but it allows us to produce an interference trace which exhibits the same statistical distributions of idle and busy periods and period lengths. Thus, the recorded distribution of idle and busy period lengths (*IDLE-PDF* and *BUSY-PDF*) can be used for analysis of *PRR* instead of using an actual recorded interference trace. The *IDLE-PDF* and *BUSY-PDF* can be measured over arbitrary time period with a fixed storage volume requirement. Thus, measuring *IDLE-PDF* and *BUSY-PDF* is a practically feasible method for collection of detailed interference patterns.

A transmitter usually performs a Clear Channel Assessment (CCA) before attempting a transmission. If a transmitter aims to send a packet during a busy period (i.e., when there are other ongoing activities in the channel stronger than R_{Thr}), the CCA would return false, and the transmission would be deferred. If the transmitter aims to send a packet during an idle period (i.e., while the channel is free), the CCA will return true and the transmission is started. The latter can only complete successfully if the remaining idle period is longer than the duration necessary for packet transmission.

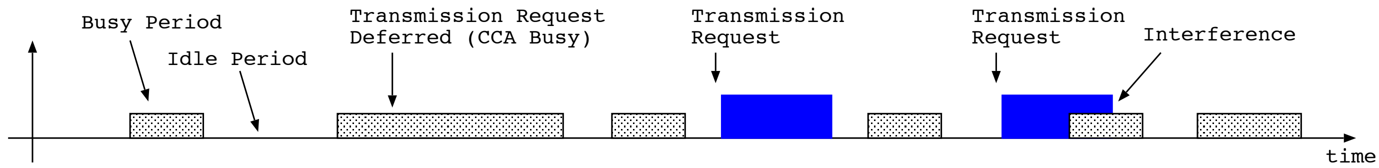


Figure 1: Example sequence of idle and busy periods. The first transmission is successful while the second one is subject to interference. A fixed packet length L is used.

The CCA test assumes that communicating nodes are within the same collision domain, such that the channel state for the transmitter is the same as the receiever. This outlined transmission behaviour is illustrated in Figure 1. As transmissions are not attempted in a busy environment only the *IDLE-PDF* is necessary for analysis of the *PRR*. Hence, in the remaining paper we focus on measurement and analysis of the *IDLE-PDF*. We use the mathematical notation $p_i(x)$ to refer to the probability distribution function *IDLE-PDF*.

In this work we only consider the raw packet transmission process and do not assume a particular Medium Access Control protocol (MAC) behaviour which may have an impact on the aforementioned assumptions. We do not assume particular transmission scheduling policies and assume random placement of transmissions within an idle period. For example, if a 1-persistent Carrier Sense Multiple Access (CSMA) strategy would be used idle and busy periods would need to be considered. However, we believe the presented assumptions are compliant within many used WSN MAC protocols.

In Section IV we will describe in detail how a suitable *IDLE-PDF* can be obtained from measurements taken using off-the-shelf sensor nodes, and discuss the effects of different durations and resolutions of the measurement. For the remainder of this section, we assume that we obtained an *IDLE-PDF* that gives an accurate representation of interference in the target area.

B. Closed Form Solution

The start of a packet transmission falls within an idle period as we assume CCA before a transmission attempt. The packet transmission has a probability P of completing successfully. This probability depends on the length of the idle period and as well on the packet length L. Figure 1 illustrates two transmissions, one successful while the other transmission is unsuccessful due to interference. A transmission is equally likely to start at any point within an encountered idle period of length y and a transmission will complete successfully if it starts at any point before $y - L$ in the idle period. The probability $P_a(y)$ of a successful transmission in an idle period of length y can therefore be computed as:

$$P_a(y) = \frac{y - L}{y} \qquad \forall y > L \tag{1}$$

The probability $P_b(y)$ of attempting a transmission in an idle period of length y is dependent on the measured *IDLE-PDF* $p_i(y)$ and is given as:

$$P_b(y) = \frac{y}{E[y]} \cdot p_i(y) \tag{2}$$

$E[y]$ is the expected value of random variable y. The overall probability P of successfully transmitting a packet in an interference environment characterised via the *IDLE-PDF* can be calculated by summing the products of the probability of a given idle size with the probability of the transmission being successful for this size:

$$P = \int_L^\infty P_a(y) \cdot P_b(y)\, \mathrm{d}y \tag{3}$$

In many observed interference scenarios, the *IDLE-PDF* follows an exponential distribution. In this case the *IDLE-PDF* $p_i(x)$ is given as:

$$p_i(x) = \begin{cases} \lambda \cdot e^{-\lambda \cdot x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{4}$$

With Equation 4, Equation 3 becomes:

$$P = \int_L^\infty \lambda^2 \cdot e^{-\lambda \cdot y} \cdot (y - L)\, \mathrm{d}y = e^{-\lambda \cdot L} \tag{5}$$

This closed form solution using an exponential distribution is useful for quick *PRR* estimation. An exponential function can be fitted to the measured *IDLE-PDF* to determine λ. Using Equation 5 and a packet length L provides an estimation of the achievable *PRR*. The computational effort of this method is dominated by the used curve fitting algorithm. This method is computationally cheap compared to the Monte Carlo Method we describe next.

C. Monte Carlo Solver

In some environments, a measured *IDLE-PDF* $p_i(x)$ may not be approximated by a well known distribution function. In this case the aforementioned closed form solution as described by Equation 3 is not straightforward to solve for P. To solve the equation in these cases we use a Monte Carlo approach.

The *IDLE-PDF* distribution is used to create a trace of duration T seconds consisting of idle periods only. As transmissions do not occur in busy periods these do not have to be included in this trace. Then, for N packets with length L a random (with uniform distribution) transmission start point $t < T$ is selected within the created trace. Transmission success or failure of each packet is recorded. This process is repeated for R runs and then the average transmission success rate across all runs is calculated and, thus, the packet delivery probability for packets of length L in presence of interference with *IDLE-PDF* $p_i(x)$ is determined.

The accuracy of this approach depends on the computational effort invested which is given by the number of tries specified via N and R. To estimate the effort necessary for obtaining sufficiently accurate results we compare the closed form solution with results obtained using the Monte Carlo method. A distribution is created using Equation 4 with a $\lambda = 100$. This distribution is used in the Monte Carlo Solver and its output is compared with the results given by Equation 3 (Using packet sizes of $L = 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$ bytes). The solver is configured with $T = 100$, $N = 1000$ and $R = 100$. The results produced by the simulator are within 0.44% on average to those of the closed form solution and have a maximum *PRR* difference of 1.42%. When $R = 50$ and $R = 10$ is used the average difference between predicted *PRR* by the model and the solver increases by 0.04% and 0.05% respectively. This suggests that reasonable accuracy can be obtained with limited numbers of simulation runs.

IV. Capturing the *IDLE-PDF*

In this section we discuss *IDLE-PDF* measurement considerations and describe a measurement tool based on standard sensor node hardware.

A. *The* IDLE-PDF *Capture Tool*

The *IDLE-PDF* is captured in the deployment area. Obviously it would be possible to deploy dedicated equipment to carry out this measurement which would result in undesirable additional deployment costs. Thus, we aim to carry out interference measurements with the same sensor node hardware used for the final application. As a result, accuracy and detail of interference measurements are limited by the available hardware. However, a benefit is that interference can be measured at the exact position where it occurs and with the same hardware impacted by the interference signal.

Modern IEEE 802.15.4-compliant radio transceivers provide the capability of reading the received signal strength (RSSI) in absence of packet transmissions. When sampled at a high rate, these measurements can be used to quantify the level of interference at a given node. Following the approach used in [6], [12] (i.e., by boosting the CPU speed, optimizing the SPI operations that are used to interface the radio), one can indeed perform a high-speed sampling of the RSSI register on Maxfor MTM-CM5000MSP nodes up to 50 kHz.

To capture the distribution of idle and busy periods, we build a Contiki application that carries out RSSI sampling as described previously and computes statistics on the idle and busy periods on a specified channel until a set amount of RSSI samples R is collected (in our application we use $R = 12.5$ million samples for a 5 minutes sampling window). We make sure every interrupt is disabled and that no other process can interfere with our operations since we need to sample at the highest possible rate. This way, we achieve a sampling rate of one RSSI value approximately every 24 μs.

We introduce an RSSI threshold R_{Thr} defining whether a channel is idle or busy (RSSI values above R_{Thr} identify a busy channel, RSSI values below R_{Thr} identify an idle channel). We count the number of consecutive RSSI readings in which a channel remained idle or busy and as soon as the current channel state (idle, busy) differs from the previous one, we increment a field in one of two arrays $A_{idle}[i]$ and $A_{busy}[i]$. Each array holds 16 fields which correspond to different ranges in length of an idle or busy periods. On recording a period, the length of the period is compared to 16 variable length running ranges and the corresponding field within either the idle or busy array is incremented. Because of the limited memory of the nodes, we truncate the maximum duration of an idle or busy period to 100 ms.

Storage requirements. In our implementation we chose to quantise an idle or busy sample (1 bit), counting the length of a period in sample units and store this recorded value using 16 variable length ranges. An alternative to this would be to store a trace of recorded samples as either RSSI values or a count of consecutive samples of a given state (idle, busy). Whilst some compression of a trace may be possible (e.g. run-length encoding) storing a distribution, even with no compression, requires significantly less storage. For comparison, during a typical 5 minutes sample with no artificial interference 25722 idle and busy periods are present. Storing each period as a 16-bit integer would require approximately 101 Kbytes of memory for both distributions. This compares to just 128 bytes for both distributions storing 16 ranges of 32-bit values.

Limitations. The achieved sampling rate is sufficiently high to identify the short instants in which the radio medium is idle due to the Inter-Frame Spaces (IFS) between 802.11 b/g packets as shown by Hauer et al. [7], [13]. Although the achievable 50 kHz sampling rate is sufficient to detect IEEE 802.11b frames, it may not be enough to capture all 802.11g/n frames (the minimum size of a Wi-Fi packet is 38 bytes, and the maximum speed of Wi-Fi transmissions is 11, 54, and 150 Mbit/s for 802.11b/g/n standards, respectively).

B. *Updating the* IDLE-PDF *at Runtime*

The environment in which a sensor network is deployed is typically dynamic and may change over time. In this case the *IDLE-PDF* captured before deployment may become invalid once the network becomes operational. It is therefore useful to measure the *IDLE-PDF* periodically at runtime to either verify that the *IDLE-PDF* used for network and application configuration is still valid or to produce an entirely new *IDLE-PDF* in case the surrounding environment has radically changed. We refer to this process as *runtime assurance*.

Runtime assurance can run alongside normal WSN software on nodes within a deployment. Periodically, a sensor node may hand control to runtime assurance which then carries out an interference measurement. The same limitations regarding measurement time, duration and location applies as discussed in the previous paragraphs. Runtime assurance may execute during times a node would normally enter a sleep state in order to ensure interference measurements do not impact on normal node operation. The software used for measurement is the same as the one used for capturing the *IDLE-PDF* before deployment (see previous paragraph).

C. Measurement Considerations

It is our aim to capture interference patterns in a deployment area such that the measurement allows us to estimate packet delivery probabilities during later network operation. Independent of the applied measurement technique this approach can only be successful if the interference impacting on the deployed network is present during measurement.

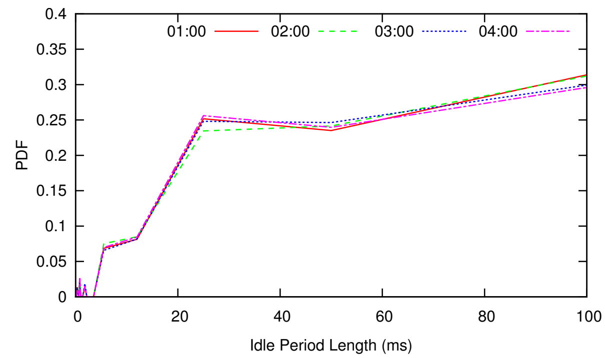
When to measure. Obviously, measurements must be carried out when a representative interference is present in the deployment. In most deployment areas interference can vary much over time. For example, in an office building Wi-Fi usage during the day will create much interference while during night little activity will be observed. Similarly, interference measured during weekends is typically lower than during work days.

Figure 2 illustrates the impact of measurement time on the shape of the recorded *IDLE-PDF*. Figure 2a shows 4 *IDLE-PDFs* captured at night in our university building. Each *IDLE-PDF* is captured over a period of 5 minutes and all sampling phases are one hour apart. All 4 *IDLE-PDFs* have a very similar shape and any of the 4 captured *IDLE-PDF* would be a good representation of the interference environment present at night. The 4 *IDLE-PDFs* shown in Figure 2b are captured by the same device during daytime. For most times the interference is similar to the night time interference; however, the *IDLE-PDF* recorded at 12:00 clearly differs. During this *IDLE-PDF* sampling, an increase in interference (most likely a very active Wi-Fi client) is present, and the *IDLE-PDF* distribution is shifted towards short idle periods. This increases the chances that a transmission is not completed before interference occurs, likely leading to a loss of the transmitted packet. The 4 *IDLE-PDFs* shown in Figure 2c are captured in the evening. The *IDLE-PDF* at 18:00 is comparable to the PDF at night while the other 3 *IDLE-PDF* show higher levels of interference.

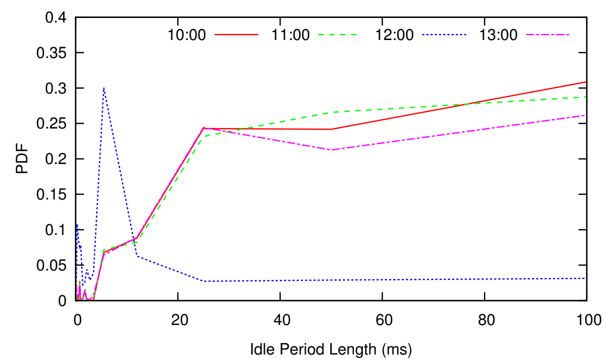
Therefore, the point in time in which the measurement is carried out has a strong impact on the obtained *IDLE-PDF*. However, it depends on the users intention on how to use the *IDLE-PDF* which informs the approach to take to deal with the observed temporal changes. If the user is interested in long-term average packet delivery reliability figures all measured *IDLE-PDF* at different points of time can be aggregated into one single average-case *IDLE-PDF*. The average-case *IDLE-PDF* for the 12 example PDFs shown in Figure 2 is given in Figure 2d. If the user is interested in worst-case packet delivery reliability that might be encountered in the deployment it would be better to select the worst-case *IDLE-PDF* (the *IDLE-PDF* leaving to the worst *PRR* prediction) as representation of the interference situation. For the example given in Figure 2 one would select the *IDLE-PDF* from 12:00 in period 2.

How long to measure. As previously shown it is important to consider the point in time when to measure the *IDLE-PDF*. The next important aspect to consider is the necessary sampling duration at each sampling time. Shorter sampling periods would be beneficial as shorter sampling durations would require less energy. We intend to use sensor nodes as sampling devices which often rely on battery power.

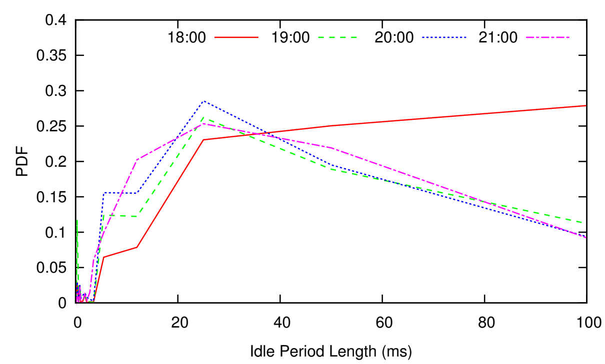
Figure 3a shows an *IDLE-PDF* captured at one point in time where different sampling durations are used (300, 60, 31, and 9.6 seconds). As it can be seen, the shape of the *IDLE-PDF* is not very dependent on the sampling duration.



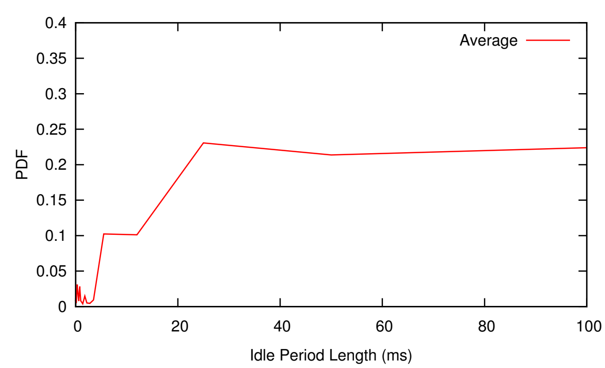
(a) Period 1



(b) Period 2

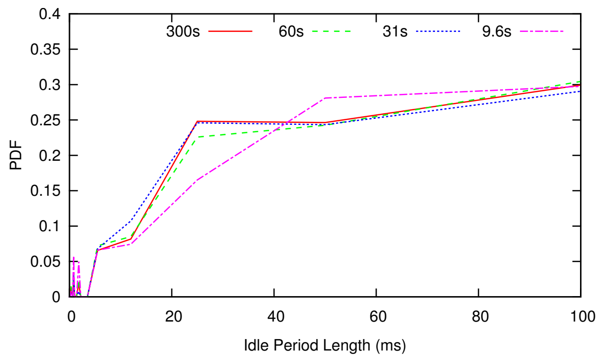


(c) Period 3

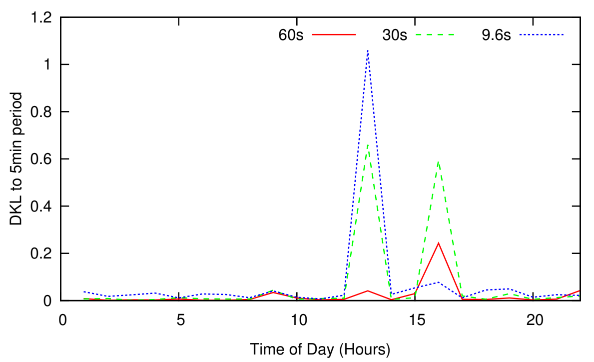


(d) Aggregated *IDLE-PDF*

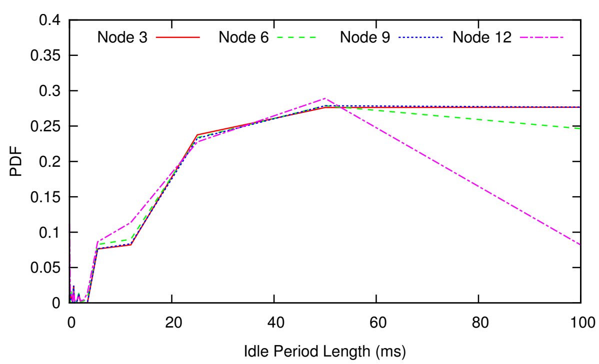
Figure 2: *IDLE-PDF* captured by four nodes on Channel 19 with $5min$ sampling periods at different times of the day. The first shows those captured at night; the second during working hours; the third during the evening. The last shows the aggregation of all three periods.



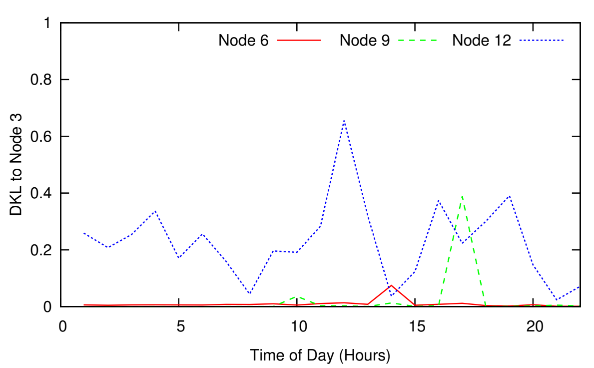
(a) Different Sampling Periods



(b) D_{KL}, Sampling Periods



(c) Different Sampling Locations



(d) D_{KL}, Sampling Locations

Figure 3: *IDLE-PDF* on Channel 19. The first two figures describe the impact of using different sample durations. The second two figures describe the impact of using different sample locations.

Formally, the Kullback-Leibler (KL) divergence can be used as a measure of the information lost when approximating the probability distribution with high sampling duration by a probability distribution with lower sampling duration. Small values of the KL measure are good as little information is lost due to lower sampling duration. Figure 3b shows the KL divergence with 300 seconds sampling duration as base over an entire day. As it can be seen the KL divergence is very low except at two measurement points. At these points, as expected, the KL divergence is higher for low sampling durations. This analysis shows that high sampling durations provide little benefit in terms of probability distribution accuracy of the *IDLE-PDF*. Combining this insight with the previous aspect on "when to measure" it seems more beneficial to distribute a large number of small sampling periods over a long time period instead of using one long sampling period at one point in time.

Where to measure. An idle period is defined as a time duration in which the energy detected in a transmission channel stays below a given threshold R_{Thr}. If interference is measured in one place it is obviously not guaranteed that interference measured at a different place in the same area is similar. Thus, most accurate results can be achieved when measuring the *IDLE-PDF* with the node that is later used as receiver for the packets for which the delivery success rate is computed. However, it is possible to record interference at one location and then use this measurement to predict the outcome of transmissions at different locations in the vicinity of the measurement spot.

Figure 3c shows the *IDLE-PDF* captured at 4 different locations at the same time (the used capture devices are spaced 3 meters apart). As can be seen, the measured interference patterns are similar. Figure 3d shows the KL divergence with the distribution captured at one node as base over an entire day. As it can be seen, the interference patterns at two nodes are almost identical (very small KL value) to the reference node while one node (node 12) experiences a slight variation in terms of measured *IDLE-PDF*. These experiments show that it is not necessary to capture the *IDLE-PDF* at every single node in the deployment. Instead, it is sufficient to capture the distribution once for an interference area (in practice, however, it might be difficult to identify these areas).

V. Evaluation

For evaluation we use Maxfor MTM-CM5000MSP sensor nodes running the Contiki operating system [14]. The nodes are used for both the transmission of data packets and to capture the *IDLE-PDF* using the software described in Section IV. All experiments are carried out in a university office environment in a room of approximately 5 m^2, vacated for the duration of each experiment. During packet transmission tests, the sending sensor node transmits 16 packets per second with size L randomly chosen from 12 fixed packet sizes to a paired receiver on a specific channel. Inter-packet spacing is approximately $1/16$ of a second with a small amount of jitter introduced to avoid any potential synchronisation effects. Each sender and receiver pair are placed approximately 3 meters apart around the centre of the room. In the experiments we use two types of interference. Firstly, Wi-Fi networks operating on channel 1 in the building carrying university network traffic

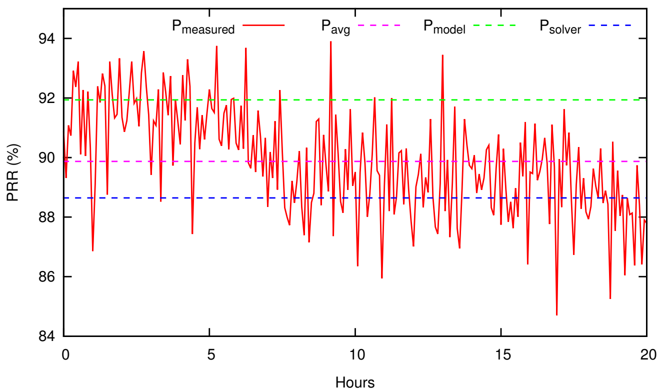


Figure 4: Predicted *PRR* using the model and solver and actual *PRR* for a packet size of 5 under background interference.

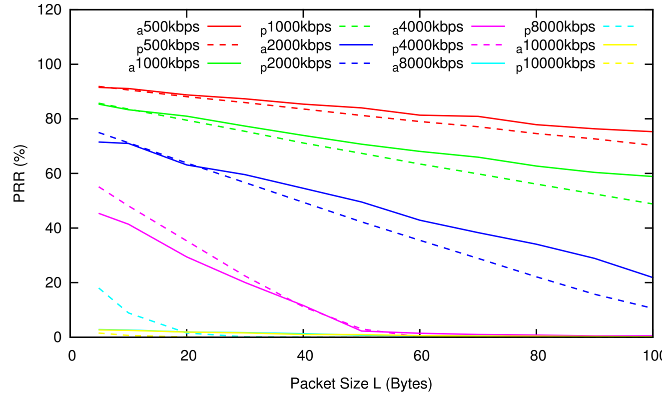


Figure 5: Predicted and actual *PRR* for different packet sizes under increasing interference (predicted prefixed with p and actual prefixed with a).

are used as a form of uncontrolled background interference, the access point is located approximately $4m$ from the room perimeter. Secondly, we use the network traffic generator *Iperf* between two desktop PCs interconnected via Wi-Fi 802.11g on channel 3 to produce controlled Wi-Fi interference. The Wi-Fi network used to generate controlled interference operates on an isolated network using a channel that is not occupied by other access points. The access point is placed in one corner of the room with connection to the first PC whilst the client adapter is placed in the opposite corner connected to the second PC.

A. *Model Based* PRR *Prediction*

In our first experiment we use interference introduced by the university Wi-Fi network. We record the *IDLE-PDF* over a period of 24 hours where in each hour the *IDLE-PDF* is recorded with a sample period of 5 minutes. The average *IDLE-PDF* distribution is then calculated from the 24 individual recorded distributions. Thus, an interference profile is created which describes the average interference level over 24 hours. We use the exponential model introduced in Section III to model this measured average *IDLE-PDF*. We then transmit packets of size L = 5 bytes over a duration of 20 hours and record the achieved *PRR* which is then compared to the model predicted *PRR*. To determine parameter λ which characterises the exponential model, we fit the model to the measured data using a total least squares fit. The model is expected to give a *PRR* prediction close to the observed *PRR* where the observed *PRR* may oscillate around the predicted value.

The results of the experiment are shown in Figure 4. The measured *PRR* (sliding window of 5 minutes) is shown over the experiment duration of 20 hours. The model predicted *PRR* matches very well the observed values. As expected, the *PRR* oscillate around the predicted value. The overall measured *PRR* over the entire duration of the experiment is $P_{measured} = 89.9\%$, while the model prediction is $P_{model} = 91.9\%$.

In this case, the exponential model produced from the captured distributions can be used to estimate the achievable *PRR* reasonably well. However, in many cases the observed distributions do not follow exactly an exponential shape and therefore the model will produce less accurate approximations. More accurate results can even be obtained in the case presented here by using the Monte Carlo Solver described in Section III. The solver will always produce more accurate results as it uses the actual measured *IDLE-PDF* instead of an approximation as used by the exponential model. Figure 4 includes the *PRR* prediction obtained using the solver which provides a *PRR* prediction of $P_{solver} = 88.6\%$. The prediction of the solver is 1.3 pp (percentage points) below the actual achieved *PRR*; in case of the model the prediction is 2 pp above the achieved *PRR*.

It has to be noted that the interference distribution used for *PRR* prediction was recorded during 24 hours before the experiment was run for 20 hours. The interference is introduced by packet transmissions on the university campus network. This shows that interference characteristics can be very stable over long periods of time. In situations where this would not be the case, the *IDLE-PDF* might need to be updated during network operation as we have described in Section IV.

B. *Monte Carlo Simulation Based* PRR *Prediction*

We now use our network traffic generator to produce six different levels of Wi-Fi interference. We vary the rate of the generated interference between 500 $kbit/s$ and 10 $Mbit/s$, and record the *IDLE-PDF* over a 1-hour period measuring two distributions with a sample duration of 5 minutes and using the average of these two distributions. We then calculate the expected *PRR* for different packet lengths L in presence of the different interference intensity levels using the Monte Carlo solver. Thereafter we transmit packets of the different sizes L in the environment exposed to the different interference intensity levels. For each intensity we record the achieved *PRR* for each packet size L over a 2 hours window. The aim of this experiment is to evaluate achievable *PRR* prediction accuracy.

The results of the experiment are shown in Figure 5. The figure shows that generally for lower interference intensities the predicted *PRR* closely matches that of the recorded values across all packet sizes. *PRR* predictions under 500 $kbit/s$ of interference have an average error (average distance between predicted *PRR* and actual *PRR* over all packet sizes) of 2.32% and a worst-case error (maximum distance between predicted *PRR* and actual *PRR* for all packet sizes) of 4.97% as described in Table I.

Table I: Average and worst-case *PRR* prediction error. The overall average of all average errors is 3.18%.

Interference Level	Worst-Case Error	Average Error
$500kbit/s$	4.97%	2.32%
$1000kbit/s$	10.07%	4.13%
$2000kbit/s$	13.10%	6.62%
$4000kbit/s$	9.66%	2.7%
$8000kbit/s$	15.07%	2.41%
$10000kbit/s$	1.85%	0.91%
Overall Average	9.12%	3.18%

As the interference intensity increases, both the average error and worst-case error increases. At an interference level of 2000 $kbit/s$ the worst-case is 13.10% and the average is 6.62%. The error eventually falls as the *PRR* approaches 0%, this can either be due to the increased level of interference as seen with 10000 $kbit/s$ of interference intensity or with increased packet transmission size at lower intensity as with 4000 $kbit/s$.

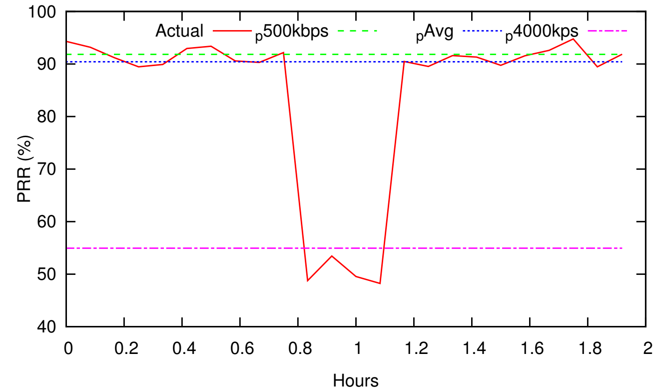
C. IDLE-PDF *Variations*

In Section IV-C we have shown that the *IDLE-PDF* measurement must be carried out at a point in time at which the interference of interest is present. In this experiment we show the impact of changes in the *IDLE-PDF* on predicted packed delivery rates. In this experiment we create controlled Wi-Fi interference in the testbed which varies over time.

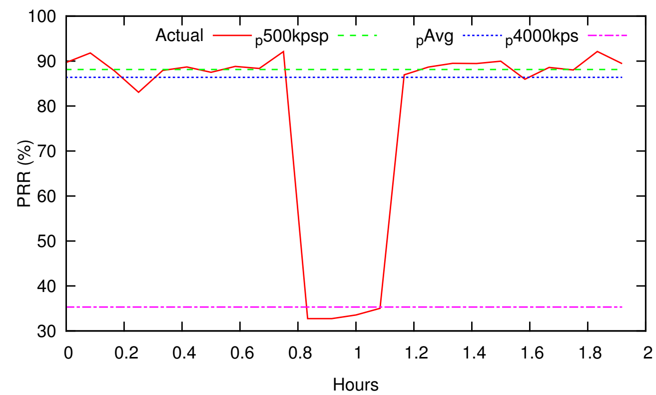
The experiment consists of transmissions of 0.5 $Mbit/s$ for a duration of 50 minutes. Thereafter, we increase the level of interference by increasing the transmission speed on the Wi-Fi network to 4 $Mbit/s$ for a duration of 20 minutes. Finally, we drop the level of interference back to 0.5 Mb/s for a duration of another 50 minutes. Before starting the experiment, we create the interference levels of 0.5 $Mbit/s$ and 4 $Mbit/s$ and record both *IDLE-PDFs* by using a sampling duration of 5 minutes over 2 hours. We select the average *IDLE-PDFs* for both cases, and use them with our Monte Carlo Simulator to determine expected *PRR* for both interference scenarios.

Figure 6 shows the results of the experiment for two packet sizes (L_1 = 5 bytes, L_2 = 20 bytes). The achieved *PRR* over time (using a 5 minutes sliding window) is shown. In addition the *PRR* using the two different *IDLE-PDFs* is shown as well. It can be seen that the achieved *PRR* is dropping in the period of increased interference. For both interference levels very accurate *PRR* are predicted when using the respective *IDLE-PDF* (a maximum difference in *PRR* of 5.1% for 500Kbps and 6.7% is observed).

The experiment also shows that it is essential to capture a suitable *IDLE-PDF* for prediction of achievable network performance (see discussion in Section IV-C). If it is for example important to predict worst-case *PRR* the *IDLE-PDF* representing high interference levels would be suitable. However, this would require that the worst-case *IDLE-PDF* was observed before network deployment.



(a) Packet Size L = 5 bytes



(b) Packet Size L = 20 bytes

Figure 6: Predicted and measured *PRR* in an environment with variable interference. Wi-Fi traffic is used as source of interference (0.5 $Mbit/s$ for a duration of 50 minutes, 4 $Mbit/s$ for a duration of 20 minutes, 0.5 $Mbit/s$ for a duration of 50 minutes).

VI. Conclusions

In this paper we have described a method for capturing interference in a deployment area by recording the distribution of idle period lengths (the *IDLE-PDF*). We have shown how a recorded *IDLE-PDF* can be used to estimate the expected *PRR* in a deployment area. To estimate the *PRR* using the *IDLE-PDF* we provide two methods. The first method assumes that the *IDLE-PDF* can be approximated using an exponential distribution. By fitting the exponential distribution to the measured distribution it is possible to obtain the *PRR*. The second method uses a Monte Carlo method to provide the *PRR* estimation from the measured *IDLE-PDF*. The first method requires less computational effort while the second method provides slightly better prediction accuracy and works as well in situations where the *IDLE-PDF* does not follow an exponential distribution. As our evaluation shows the Monte Carlo based prediction has an accuracy with an average difference of 3.2% between predicted *PRR* and measured *PRR* (see Table I).

We have found that the *IDLE-PDF* is a good measure for capturing interference in a resource efficient way. Capturing the *IDLE-PDF* requires a constant amount of storage space independent of the sampling duration while the alternative method of recording interference traces requires an amount of storage space proportional to the interference measurement duration. Thus, the *IDLE-PDF* can be captured efficiently with off-the-shelf nodes during network operation; this feature is important for updating *IDLE-PDF* measurements once the network is deployed. We have also discussed how an *IDLE-PDF* should be measured and have shown that frequent but short sampling is beneficial compared to long but infrequent sampling. A representative *IDLE-PDF*, i.e., an *IDLE-PDF* able to produce accurate *PRR* predictions, can be captured in many situations with a single sampling period of 30 seconds.

In this paper we have focused on prediction of *PRR*. However, we believe that analysis of the captured *IDLE-PDF* can be also be used for other purposes. For example, analysis of the *IDLE-PDF* on all channels can provide information for channel ranking. Channels providing the best *PRR* may be used by a communication protocol. This application has the same aim as the work presented by Shariatmadari et al. [10] but using the *IDLE-PDF* would allow more efficient interference analysis on nodes and would not require assumptions on traffic patterns used in the network.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union $7th$ Framework Programme (FP7-ICT-2011.1.6) under grant agreements n° 317826 (RELYonIT).

REFERENCES

[1] C. A. Boano and K. Römer, "External radio interference," in *Radio Link Quality Estimation in Low-Power Wireless Networks*, ser. SpringerBriefs in Electrical and Computer Engineering - Cooperating Objects, N. Baccour, A. Koubâa, C. A. Boano, L. Mottola, H. Fotouhi, M. Alves, H. Youssef, M. A. Zuniga, D. Puccinelli, T. Voigt, K. Römer, and C. Noda, Eds. Springer International Publishing, Jul. 2013, pp. 21–63.

[2] G. J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting wifi white space for zigbee performance assurance," in *Proceedings of the* 18^{th} *IEEE International Conference on Network Protocols (ICNP)*, oct 2010, pp. 305–314.

[3] W.-B. Poettner, H. Seidel, J. Brown, U. Roedig, and L. Wolf, "Constructing schedules for time-critical data delivery in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 3, 2014.

[4] S. Zacharias, T. Newe, S. O'Keeffe, and E. Lewis, "Identifying sources of interference in rssi traces of a single IEEE 802.15.4 channel," in *Proceedings of the 8th International Conference on Wireless and Mobile Communications (ICWMC)*, jun 2012, pp. 408–414.

[5] N. M. Boers, I. Nikolaidis, and P. Gburzynski, "Sampling and classifying interference patterns in a wireless sensor network," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 1, pp. 2:1–19, nov 2012.

[6] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga, "JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *Proceedings of the* 10^{th} *International Conference on Information Processing in Sensor Networks (IPSN)*, apr 2011, pp. 175–186.

[7] J.-H. Hauer, V. Handziski, and A. Wolisz, "Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks," in *Proceedings of the* 6^{th} *European Conference on Wireless Sensor Networks (EWSN)*, feb 2009, pp. 17–32.

[8] C. Noda, S. Prabh, M. Alves, C. A. Boano, and T. Voigt, "Quantifying the channel quality for interference-aware wireless sensor networks," *ACM SIGBED Review*, vol. 8, no. 4, pp. 43–48, nov 2011.

[9] K. R. Chowdhury and I. F. Akyildiz, "Interferer classification, channel selection and transmission adaptation for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, jun 2009, pp. 1–5.

[10] H. Shariatmadari, A. Mahmood, and R. Jäntti, "Channel ranking based on packet delivery ratio estimation in wireless sensor networks," in *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, apr 2013, pp. 59–64.

[11] J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. J. Sreenan, "Burstprobe: Debugging time-critical data delivery in wireless sensor networks," in *Proceedings of the* 8^{th} *European Conference on Wireless Sensor Networks (EWSN)*, feb 2011, pp. 195–210.

[12] C. A. Boano, M. A. Zúñiga, K. Römer, and T. Voigt, "JAG: Reliable and predictable wireless agreement under external radio interference," in *Proceedings of the* 33^{rd} *IEEE International Real-Time Systems Symposium (RTSS)*. IEEE, Dec. 2012, pp. 315–326.

[13] J.-H. Hauer, A. Willig, and A. Wolisz, "Mitigating the effects of RF interference through RSSI-based error recovery," in *Proceedings of the* 7^{th} *European Conference on Wireless Sensor Networks (EWSN)*, feb 2010, pp. 224–239.

[14] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the* 1^{st} *International Workshop on Embedded Networked Sensors (EmNetS)*, Tampa, FL, USA, nov 2004.