Retransmission strategies for cyclic polling over wireless channels in the presence of interference

Giovanni Gamba, Federico Tramarin
Department of Information Engineering, University of Padova,
{giogamba, tramarin}@dei.unipd.it

Andreas Willig
Telecommunication Networks Group
Technical University of Berlin
awillig@tkn.tu-berlin.de

Abstract

In this paper we consider retransmission strategies for centralized cyclic polling-based systems over wireless channels subject to external interference. The considered strategies differ in the time when retransmissions for one particular node are carried out, and in the number of retransmissions that can be carried out for one node. We show experimentally and by simulation that two related strategies introduced in this paper, called the queuing-bounded strategies, significantly outperform the traditional strategy in which all admissible trials towards one node are carried out subsequently in terms of the average number of nodes that cannot be successfully served in a cycle. These performance gains are achieved without increasing the average total transmission effort.

1. Introduction

In many wired industrial networks like for example WorldFIP or PROFIBUS-DP [15] a large share of the overall traffic comes from periodic exchanges of data between a central controller and its attached sensor / actuator stations (which we henceforth call nodes). This traffic typically takes place in a cyclically repeated periodic window, which is reserved exclusively for the treatment of periodic traffic, whereas sporadic traffic is handled in a separate time window. These two windows alternate in time.

In wireless industrial networks [17, 18] it can be anticipated that the same will be true, but as opposed to wired transmission media, the wireless exchange of data is subject to a number of disturbances, including channel fading and external interference. These disturbances can directly affect the delivery rate of packets and subsequently the controller might have to operate on an inconsistent view of the physical process. Taking the underlying wireless physical layer as given, the major control knob to improve the delivery rate are link-layer retransmissions. To accommodate these retransmissions, extra time has to be allocated in the periodic window. This extra time can be used in different ways. In one common approach, the bounded immediate retransmission (BIR) approach, for each node there is a limited number of transmission trials available, which are handled successively, i.e. without serving other nodes between two trials for one node. When this maximum number of trials has been exhausted and the controller has not received the requested data or acknowledgement frames, the node is considered as having failed during this cycle. Due to the transient nature of errors on the wireless channel, it might well happen that a node that fails within this cycle, can be successfully served during the next cycle.

In this paper we consider different retransmission strategies for centralized wireless industrial systems which are subjected to external interference. We demonstrate experimentally and by simulation that there are strategies which reduce the fraction of failed nodes during the periodic window over the BIR strategy for the same size of the periodic window and the same number of nodes to be served. These alternative strategies do not insist on making all trials for a node subsequently, and they also allow in a very simple fashion to use the time for trials that “good” nodes have not used in order to serve “bad” nodes with an increased number of trials.

The experiments are carried out with Telos motes platforms carrying a ChipCon CC2420 transceivers that are IEEE 802.15.4-compliant [4], [9] and which operate in the 2.4 GHz ISM band. The simulation model considers a system with a similar physical layer. For both the experimental and the simulation-based assessment it is assumed that the interference is generated by a static interferer (like e.g. a Wifi device) which is not influenced by the controllers or the nodes periodic transmissions (i.e. even if the interferer runs a CSMA-type MAC, it does not recognize the ongoing periodic transmissions).

The paper is structured as follows: in the next Section 2 we explain our assumptions on the network, the organization of a cycle, the operation of the interferer and the major performance measures considered in this paper. Following this, in Section 3 we precisely describe the retransmission strategies considered in this paper. In Section 4 we explain the simulation setup and present the results of a simulation-based performance study. In Section 5 we do the same for the experimental study. In Section 6 we discuss related work and in Section 7 we offer our conclusions.

2. System model and assumptions

2.1. Network model

We consider a system of one central station (the controller) and a number N of sensor / actuator stations (the nodes). When we do not want to discriminate amongst node and controller, we simply speak of a station.
All stations are stationary. The set of nodes does not change over time and all node addresses are known to the controller. It is assumed that the network has already been set up, i.e. all nodes have successfully associated to the controller.

All stations share a common wireless medium (i.e. work in the same frequency band) such that each node can communicate with the controller. It is, however, in general not assumed that each node hears all the other nodes. On the physical layer we assume that all stations are compliant to the IEEE 802.15.4 standard. They operate in the 2.4 GHz ISM band and use all the same modulation scheme and transmit power. There is no channel coding used. The controller regulates access to the channel, i.e. the controller can be considered as a master and the nodes can be considered as slaves. A polling-based scheme is adopted, i.e. a scheme in which the controller sends a request-frame to an individual node (possibly carrying some output data for this node) and the node immediately answers with a response-frame (again, possibly carrying some data). The controller is able to determine whether it successfully received a response-frame or not, i.e. it can obtain binary feedback. It is assumed that a client-server interaction pattern is adopted, i.e. the data generated by one node is not immediately relevant to any other node, but only to the controller.

An important assumption is that the controller does not perform any carrier-sensing operation before transmitting a request-frame, and vice versa a node does not perform carrier-sensing before transmitting a response packet. This is a common assumption in industrial master-slave systems, but has the disadvantage that the system has no chance to respond directly to external interference.

2.2. Organization of the cycle

The cycle organization is fairly standard and shown in Figure 1. A cycle has a fixed duration (cycle period) of $\Delta_s$ seconds. A cycle starts with a synchronization or beacon packet broadcast by the controller to all nodes. The main purpose of this packet is for the nodes to maintain time synchronization with the controller. It is not strictly necessary for a node to receive each beacon and in this paper we do not care about the beacons anymore.

The remaining cycle is sub-divided into two windows. In the first window, the periodic window, the controller handles all periodic traffic. The size of the periodic window, $\Delta_p$ is chosen such that each node can at least be polled once (involving transmission of a request packet and a response packet) and furthermore some additional time budget is available to carry out retransmissions (which again consist of request and response). All these retransmissions, however, have to happen within the periodic window. It may happen that for one or more nodes the controller has not obtained any response at the end of the periodic window. In this case, the nodes are said to have failed during the window. In practical implementations, nodes that fail successively for a number of periodic windows would be removed from the polling sequence. We do not adopt this policy in this paper, i.e. nodes are always polled, no matter how often they have failed in the past.

The following aperiodic/idle window is of no further concern to us – it can (like e.g. in WorldFIP) be used to carry out aperiodic message exchanges, or it can be a pre-defined idle phase which the controller can use to process the responses. The only relevant assumption about this window is that it is in no way available for handling periodic traffic anymore.

2.3. Traffic model

For the purposes of this paper it suffices to assume a very simple traffic model: in each cycle the controller produces for node $n$ output data of length $l_{o,n}$ and vice versa node $n$ produces input data for the controller of length $l_{i,n}$. In addition, a fixed-length packet header and trailer consisting of $l_h$ bytes is added to the user data. For the purposes of this study we make the simplifying assumption that all output data have the same length, i.e. $l_{o,n} = l_o$ for some constant $l_o > 0$ and all $n$. Similarly, for the input data we assume $l_{i,n} = l_i > 0$.

With the assumption of equal data sizes we assume that for all schemes there are during the periodic window a total number of $N \times K$ trials available for some integer $K > 1$.

2.4. Interference model

In both our experiments and the simulations the interference is generated artificially from a relatively simple stochastic process. The interferer does not perform any carrier-sensing on the common channel, so that the interferer’s behaviour is not influenced by the controller and the nodes (please note that by the absence of any carrier-sensing operation before the request and response frames in the periodic window the controller and the nodes in turn also do not change their transmission behaviour in response to the interference).

The interference process is depicted in Figure 2. It alternates between bursts, during which the external interferer transmits, and gaps, during which the external interferer is silent. The interferer is static and uses the same transmit power in all the bursts. It is assumed that the interferer uses a completely different rate and modulation scheme than the controller and that the signals generated by the interferer can be regarded as being white noise for the controller and nodes, so that the activities of the interferer look like a time-varying noise level. This is common assumption for modeling external interference.
To keep the generation of the interference simple, all gaps are iid exponential random variables of a given average gap length. The burst lengths are iid and have a uniform distribution drawn from a given interval $[b_l, b_u]$ where the bounds $b_l$ and $b_u$ are counted in channel bit times.

There is, however, a crucial difference between our experiments and the simulations regarding the directivity of the interferer. In the experiments we have used a directional antenna with a relatively narrow beam. If we point this antenna to one node, then the reception of this node is distorted, but the transmissions of this node can still be heard by the other nodes, when they are outside the antenna beam. In contrast, in the simulations the interferer is assumed to have a perfectly omni-directional antenna. For this reason the experimental and simulation results are not directly comparable.

### 2.5. Major performance measures

The main performance criterion is the delivery rate. More specifically:

- The downlink delivery rate gives the average fraction of nodes which successfully receive their output data within the periodic window. The downlink complete cycles measure gives the fraction of cycles in which all $N$ nodes receive their output data.

- Vice versa, the uplink delivery rate gives the average fraction of nodes which successfully deliver their data to the controller within the periodic window. An alternative formulation of this criterion is to obtain the long-term percentage of uplink complete cycles, i.e. of cycles in which the controller manages to obtain answers from all $N$ nodes.

Since controller and nodes use a request-response communication pattern, the uplink-delivery rate is always at least as large as the downlink delivery rate. We therefore concentrate on the uplink delivery rate, or equivalently, the (average) number of nodes for which no uplink packet is received (which denote as average number of unserved nodes).

A second criterion is the efficiency of each scheme, which we define here as the fraction of the periodic window in which the controller is busy with transmitting and receiving data. The earlier the controller finishes the data transfers, the better.

### 3. Considered retransmission strategies

We describe the four retransmission strategies considered in this paper.

#### 3.1. Bounded Immediate retransmissions (BIR)

This is a baseline scheme. It works as follows: for each node at most $K$ trials can be made. If the first trial fails, the controller immediately starts the next trial, until all $K$ trials have been exhausted, the controller receives a response successfully or the periodic window ends. Then the next node is served, if the periodic window has not yet ended.

Technically, this scheme is a simple ARQ scheme with a bounded number of immediate retransmissions. The efficiency of this scheme will depend on the channel coherence time (i.e. the time for which the channel does not change its characteristics [12]) of the particular channel between node and controller.

Besides its dependence on the channel coherence time this scheme has the disadvantage that the trials not used by one node cannot be used by another node.

#### 3.2. Unbounded Immediate retransmissions (UIR)

This scheme is similar to the BIR scheme, but there is no limit of $K$ for the trials towards one receiver – the controller can perform as many retransmissions for a node as would fit into the periodic window. This strategy has been included to assess the effect of the bounded number of retransmissions.

#### 3.3. Queued retransmissions (QR)

The controller maintains a FIFO queue of node addresses. When this queue is empty or the periodic window is exhausted, the controller stops working on periodic data exchanges. Otherwise, the controller takes the first entry, say address $i$, from the queue (and removes it from the queue) and, provided that enough remaining time is available in the periodic window, performs one single trial towards node $i$. If this trial fails, a new entry for $i$ is appended to the queue. At the beginning of the periodic window the queue is initialized with all $N$ addresses in the sequence from 1 to $N$.

In effect, by this approach first all $N$ nodes are tried once. The successful nodes are not considered any further during the current cycle, and for the failing nodes a new trial is appended to the queue. This has two effects:

- The spacing between the first trial and the first retransmission is larger on average, which allows to deal with larger channel coherence times. The spacing between the second and third trial is in general random.

- The number of retransmissions that can be performed for one node might be larger than $K$, provided other nodes have required fewer trials.

Please note that the queueing mechanism (where the controller always takes the HOL of the queue for the next trial until complete success or end of periodic window) can in principle also be applied in the BIR or UIR scheme: whenever a trial fails, the address of the failing node could be made the new HOL, unless he exhausted already all his $K$ trials.

#### 3.4. Adaptive Queued retransmissions (AQR)

This scheme is similar to the QR scheme but with one important difference. The controller maintains for each node long-term statistics about the average probability with which one trial is successful. In this paper we have adopted for the simulations a simple exponentially weighted moving average:

$$ s_{n+1} = \alpha s_n + (1 - \alpha) s_{n+1} $$

where $s_n$ is the average after $n$ trials, $s_{n+1}$ is the average after $n+1$ trials, $s_{n+1}$ is the outcome (success or failure).
of the \( n \)-th trial and \( \alpha \) is a parameter. We have chosen
\( \alpha = 0.9 \), i.e. most weight is put on the history.

At the start of a cycle, instead of initializing the queue with 1, 2, \ldots, \( N \) it is initialized according to the average success probability for a trial: nodes having the largest success probability are tried first.

### 4. Simulation-based evaluation

In this section we describe the results of a simulation-based performance study. As already mentioned in Section 2.4, the interference model supported by the simulator (with an omnidirectional interferer) differs from the interference created in the measurement setup, which is directional.

We first describe the simulation model, then we present and discuss the results.

#### 4.1. Simulation model

The simulation model is using a proprietary discrete-event simulator written in Common Lisp [6, 5]. This simulator includes the protocols used in this paper, a separate interferer node and a wireless channel model linking all stations (including the interferer) together.

The fixed simulation parameters are given in Table 1. The cycle time has been set to 400 ms. For one single trial (consisting of request and response) the parameters for the packet lengths and the transceiver turnaround times have been chosen so that one trial takes 200 ms. This choice has been made to be consistent with the measurement setup, the turnaround times in the simulation model accommodate the processing delays observed in the setup. The periodic window has a length of 330 ms, which is sufficient to accommodate 16 trials (i.e. \( K = 2 \)). The overall setup consists of the controller, eight nodes and an interfering node. The controller has been placed in the center (position \((0, 0)\)), the nodes have been placed equidistantly in a circle of 7 m radius around the center. The rightmost node is placed at position \((7, 0)\).

For the position of the interferer we have considered two settings:

- In the first setting the interferer is placed at position \((14, 0)\) and therefore disturbs the rightmost nodes most. We call this, somewhat imprecisely, the one-disturbed-node scenario.
- In the second setting the interferer is placed very close to the controller, so that all nodes are disturbed in the same way by the interferer and the controller is disturbed most. We call this the all-disturbed-nodes scenario.

For the wireless channel we assume a narrowband channel following the log-distance path loss model [12], using a reference distance of 1 m and a path loss exponent of \( \gamma = 3 \). It is furthermore assumed that all channels are independent of each other.

The interferer follows the pattern described in Section 2.4. For the simulations, we have varied the interferer parameters as summarized in Table 2.

#### 4.2. Results

We show in Figures 3 and 5 for an average interference gap length of 10 ms and the one-disturbed-node scenario the average number of unserved nodes for the short and long average interference burst lengths, respectively. In Figure 4 we show similar results for the case of 30 ms average gap lengths and short bursts. The results for the 50 ms case look very similar to the curves for the 30 ms case and are not shown here. The following points are noteworthy:

- For the 10 ms average gap duration case, it can be seen in Figure 3 that for the shorter burst lengths the BIR scheme shows consistently the worst performance, and the queued retransmission schemes (QR and AQR) consistently show the best performance (with the AQR scheme having slight advantages). For interferer transmit powers between -90 and -80 dB the UIR scheme shows a performance that is in between the performance of the BIR scheme and the queued schemes. For the longer burst lengths (compare Figure 5) the UIR scheme shows the worst performance over a range of interferer transmit powers, while again the queued schemes are consistently the best ones.

- For 30 ms and 50 ms average gap times (see Figure 4) and short interference burst lengths the BIR scheme shows the worst performance, whereas the three other schemes do not differ appreciably. The same is true for the longer bursts at 30 ms average gap duration, although the three schemes UIR, QR and AQR show slightly different performance, with the queued schemes being the best ones.

<table>
<thead>
<tr>
<th>PHY parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit power ( P_t )</td>
<td>1 mW (-30 dB)</td>
</tr>
<tr>
<td>Noise power ( N_0 )</td>
<td>-173.0 dBW/Hz</td>
</tr>
<tr>
<td>Reference distance ( d_0 )</td>
<td>1 m</td>
</tr>
<tr>
<td>Reference path loss ( PL_0 )</td>
<td>-50 dB (see [14],[10])</td>
</tr>
<tr>
<td>Path loss exponent ( \gamma )</td>
<td>3</td>
</tr>
<tr>
<td>Modulation and data rate</td>
<td>BPSK@250 kbps</td>
</tr>
<tr>
<td>Transceiver turnover times</td>
<td>2232 bit times</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAC-Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed overhead size</td>
<td>152 bits</td>
</tr>
<tr>
<td>Beacon payload size</td>
<td>48 bits</td>
</tr>
<tr>
<td>Downlink payload size</td>
<td>48 bits</td>
</tr>
<tr>
<td>Uplink payload size</td>
<td>168 bits</td>
</tr>
<tr>
<td>Max. trials for BIR scheme</td>
<td>2</td>
</tr>
<tr>
<td>Cycle time</td>
<td>400 ms</td>
</tr>
<tr>
<td>Periodic window size</td>
<td>330 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation precision parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation duration</td>
<td>10800 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IF parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit power</td>
<td>((-50, -60, -80, -82, \ldots, -90)) dB</td>
</tr>
<tr>
<td>Avg. gap length</td>
<td>((10, 30, 50)) ms</td>
</tr>
<tr>
<td>Burst length distr.</td>
<td>([U[1, 10], U[2, 20]) ms</td>
</tr>
</tbody>
</table>

**Table 1. Fixed simulation parameters**

| Burst length distr. | \([-50, -70, -90]\) dB |

**Table 2. Varied interferer parameters**
• While not shown in the figures, all our results for the all-nodes-disturbed scenario are very similar in the following sense: the BIR is always significantly worse than the other three schemes. On the other hand, the differences between the three schemes UIR, QR and AQR is always small and often negligible.

As a conclusion, when all nodes are distorted in the same way (as in the all-nodes-disturbed scenario), there are practically no differences between UIR, QR and AQR, but all three are significantly better than BIR. When the interference situation among the nodes becomes distinct, the performance of the schemes starts to differentiate as well and the AQR shows the best performance. The BIR scheme, on the other hand, shows the worst performance most of the time. One possible explanation for this is depicted in Figure 6, which shows that the BIR scheme has smaller average utilization of the periodic window. This hints to the interpretation that the other schemes can benefit from their ability to allocate more trials than \( K \) to poor nodes when the good nodes have not used their budget.

5. Measurement results

5.1. Measurement setup

Figure 7 shows the measurement setup that has been used in our experiments. The sensor network comprises of one controller and eight nodes. All the experimental part of this work has been performed in a non-anechoic room, so non-ideal effects of a real-life environment could not be excluded. However a preliminary scan of the chosen band showed no spurious emissions.

As has been described in Section 4.1, we have used two different scenarios for our experiments. In the first scenario nodes have been evenly spaced on a semi-circumference of 60 cm radius, with the controller in the center. A directional antenna has been placed behind the controller in order to limit irradiated interference to it. In the second scenario, the interference has to be directed only toward one node.
For the sake of clarity, we used the same sketch to represent the situation, with a switch that plugs the RF interference signal into a second antenna (the leftmost in the figure), with the beam directed only toward this node. The interference power has been regulated to disturb only communications regarding this node.

We used the TinyOS 2.1 operating system to develop the specific high level application of each node. This application implements the different queueing systems developed in this work into the sensor network. The underlying protocol stack is basically the default protocol stack delivered with TinyOS 2.1, but we modified the CSMA/CA and CCA mechanisms to effectively get rid of the carrier sense functionalities.

The application provides the correct management for the queue storing the node addresses, the beacon transmission, the poll request generation and, on the node-side, the generation of correct response to incoming requests.

The interference signal behaviour has been described in Section 2.4. The real signal used in our experiments was an AWGN signal, with a bandwidth of 5 MHz, centered over the same IEEE 802.15.4 channel used for the WSN (in this case the 26-th, i.e. 2.48 GHz), produced by a RF Agilent E4433B signal generator. To generate the described pattern we have used a pulsed mode, using a baseband signal generator as trigger. The baseband signal generator is able to reproduce an arbitrary waveform from a succession of values from a file. Thus we have first generated this succession of points sampling the stochastic process described in Section 2.4, then we have used the baseband signal generator to reproduce a signal according to the succession. Finally, we have used this signal as a trigger for the RF signal generator, to switch on and off the radio.

Note that in this approach the interferer does not react (by not using any carrier sense mechanism) to the traffic generated by the sensor network.

The RF signal produced has been irradiated with a directional antenna placed behind the controller, with the main lobe covering the WSN area.

The software produced and installed on sensors provides also a "sense" functionality, in which each mote senses the channel for 30 seconds, collecting RSSI measurements and returning a packet with the collected statistics. This mechanism provides the noise/interference floor of each node, if the estimation is done with the interference switched always-on. Using this information it is possible to state that, in the experiments, regarding the SNR values measured, only responses to the poll request are affected by the interference, whereas poll requests are received correctly. The signal and noise levels estimation are shown, qualitatively, in Figure 8. Each node senses an interference level of roughly -45 dBm, and the controller estimates roughly -30 dBm. From data collected during normal transmissions, the power received in the up- and down-link is around -30 dBm.

Thus, at the node side, a poll request is received with a SNR of 15 dBm, whereas at the controller side, the response has an SNR of 0 dBm. Please note that, for the nodes type in the experiments, whereas an SNR level of 15 dBm is sufficient for a correct reception, an SNR of 0 dBm leads to a 100% probability of error.

5.2. Results

In our experiments we have fixed some parameters with respect to the simulation, in order to shorten the time requirements for the tests. First, we have fixed the IF power, which was set, experimentally, to 8 dBm at the instrument side (RF signal generator), since we want a level of at least -30 dBm at the controller. Then, we have fixed the transmission power for nodes at the maximum allowed by IEEE 802.15.4, namely 1 dBm. Also the channel has been chosen as the 26-th of the standard, centered at 2.48 GHz.

The MAC parameters has been chosen accordingly to the Table 1. The IF signal has been reproduced with the same characteristics presented in the simulation section, therefore we have considered three different mean values for the exponential random variable (representing gap spaces between bursts), but only one value for the uniform one, precisely we have used $U[1, 10]$ ms.
In order to obtain comprehensive statistics on data collected from measurements, we have chosen to run a single experiment for approximately 15 minutes, allowing the transmission of about 2300 cycles. Finally, each experiment has been carried out with three repetitions, spaced in time, to avoid correlations with environmental parameter variations.

Results from measurements are collected in Figures from 9 to 12. There is a separate graph for the number of unserved nodes and for the utilization of cycles, either for the all-disturbed scenario and the one-disturbed one. The results shown in each graph are the averages taken over all three repetitions of an experiment, in all cases the results of the repetitions were very close to each other. The following points are noteworthy:

- In terms of the average number of unserved nodes the QR scheme is consistently better than the BIR scheme (see Figures 9, 11). In can also be observed that the number of unserved nodes decreases with increasing interference gap time.

- Both schemes have approximately the same average cycle utilization (compare Figures 10, 12). It can also be noted that for increasing interference gap time the utilization approaches 50%. In the one-disturbed-node scenario the difference in utilization between 10 ms gap time on one hand and 30, 50 ms gap time on the other hand is less pronounced than for the all-disturbed-node scenario.
6. Related work

In the literature different works face the problem of interference among different wireless systems. In [7] a theoretical and simulative analysis of interference among WiFi and Bluetooth networks is offered, while in [8] a similar approach is used to evaluate interference among WiFi and IEEE 802.15.4 nodes.

A more complex model and an exhaustive set of simulations are presented in [13]. In [2] a fully theoretical and experimental approach is provided in order to evaluate the performance of a industrial polling system based on IEEE 802.15.4-compliant nodes. The cyclic polling is quite similar to the BIR scheme considered in this paper, apart from the CSMA/CA setup. The analyzed interference scenario used a pulsed interference with fixed burst and gap times. The same polling scheme is used in [1]. In this paper the presence of real-life interference (Bluetooth, WiFi, ZigBee interferers) on a IEEE 802.15.4 network is experimentally investigated varying several CSMA/CA parameters. In all these testbeds the standard TinyOS BIR scheme is used, and no investigation on retransmission strategies is offered.

To the best of our knowledge, the queueing-based retransmission schemes proposed in this paper have not been described so far in the context of industrial communications, nor have they been investigated under external interference. However, wireless fair scheduling schemes [11], [16] have some similarities to the queueing-based schemes. In these schemes the fading nature of the wireless channel is exploited to serve different nodes when the channel towards the current node becomes bad (see also [3] for an early facet of this idea).

7. Conclusions

In this paper we have presented a simple, queueing-based retransmission scheme and investigated its performance (measured in the average fraction of unserved nodes) under different interference scenarios, showing that it is consistently better than the standard BIR scheme. This queueing mechanism deviates from BIR in two important ways: firstly it can increase the time duration between two trials towards a node, which can be beneficial to overcome channel coherence times on the order of a packet length or more, and secondly it allows to allocate more trials to poor nodes when the good nodes have used less than their nominally allowable trials.

There is a lot of potential for future work. Firstly, it is worthwhile to consider more general interference patterns, for example incorporating real WiFi traffic. Secondly, it would be very interesting to incorporate transmit power control into the cyclic polling scheme. Thirdly, it is also interesting to evaluate the jitter observed in the nodes and in the controller, especially in the queueing strategies.

References