Source-Node Selection to Increase the Reliability of Sensor Networks for Building Automation

Atis Elsts
SICS Swedish ICT
atis.elsts@sics.se

Abstract

We experimentally investigate the performance of IEEE 802.15.4 radio links and their failure modes in an office building, and, based on this study, propose an adaptation mechanism to conciliate application-level reliability requirements with the underlying network-level properties. The mechanism has two aspects: the spatially-adaptive aspect, implemented through adaptively selecting one or more source nodes for each application-level data connection, and the frequency-adaptive aspect, implemented through IEEE 802.15.4 channel hopping and blacklisting.

Through extensive trace-based simulations and experiments in a test network, we show that the mechanism satisfies application requirements on maximal information age as long as at least one of the potential source nodes is connected to the rest of the network, at the same time showing lower energy consumption than non-adaptively using multiple source nodes.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer Communication Networks

General Terms

Performance, Reliability

Keywords

Sensor networks, building automation

1 Introduction

The *smart building* is a lucrative application area for the Internet of Things. Automated collection of environmental variables of interest (temperature, air quality, light intensity) is required to enable remote monitoring and closed control loops for HVAC (heating, ventilating, and air conditioning) systems. However, battery-powered wireless systems are more vulnerable to link and node-level failures than their wired analogues. While there are many approaches (Section 5) that can successfully increase reliability of communication through retransmissions, frequency diversity, redundant encoding, multipath & opportunistic routing, and flooding, they typically cannot cope with nodes becoming completely unreachable. That is what delay-tolerant networking techniques do [24]; however, they cannot be applied to systems that require constant flow of recent information to make operational decisions.

In reliability-critical systems such as industry automation [2] the failure problem is solved by extensive redundancy [4] or by run-time assurance [29]. Both solutions lead to increased maintenance: in the former case, because node and especially battery replacements are needed more frequently, as energy usage is increased due to additional traffic; in the latter, as every failure must be operatively fixed.

However, for the typical HVAC application a temporary failure leads to an inconvenience rather than a life-threatening situation. Therefore the number of maintenance operations should be minimized; it is a goal that cannot be compromised by the need for reliability. Furthermore, while in industry automation extremely stringent delay requirements and high datarates are common, HVAC-related environmental variables are changing slowly and sampling periods of many seconds are often used. This leads to the need to stress a different aspect of reliability. The main property required for such a system to work reliably is to have bounded *maximal information age* on the decision-making (receiver) nodes; the data they have should never be more than a few minutes old in order to enable low *miss time* [19] (the time with suboptimal temperature or air quality).

Contribution. We present a system that uses a combined approach of spatial diversity and frequency diversity to increase the reliability (as measured by the maximal information age) and lifetime of building automation applications in conditions of link and node failures.

Design overview. The system is designed to exploit data harvesting node redundancy in sensor networks deployed for building automation. It functions through adaptively activating one or more data source nodes for each receiver node. The decision which source nodes to activate is made by each receiver autonomously and is based on end-to-end ETX metric with hysteresis (to avoid churn in the network) and decay (to enable periodical re-exploration of links to inactive nodes if performance is suboptimal). We implement our sys-

International Conference on Embedded Wireless Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

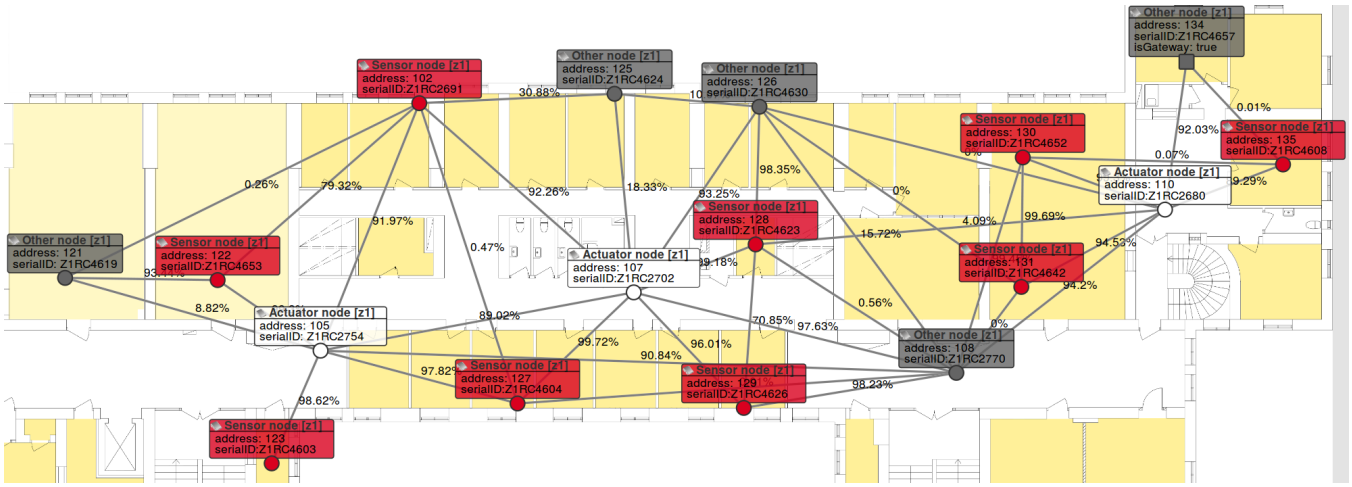


Figure 1: **The test network.** On the links, average link-level PRR in IEEE 802.15.4 channel 26 are shown

tem on top of MiCMAC [21], a channel-hopping MAC protocol, which we extend with per-link channel blacklisting. The receiver node dynamically learns how many channels are blacklisted on each of the source nodes, and uses this information along the ETX to decide how many source nodes to activate. We show that combining the spatially-adaptive and frequency-adaptive aspects in this cross-layer approach gives better results compared to what using them both separately would give. The selection mechanism is lightweight: it does not require introduction of any link measurement traffic, but relies solely on application-level data and control packet transmissions.

Assumptions. The system is designed for building automation applications with two kinds of nodes in the network: redundant and opportunistically deployed sensor nodes, each wirelessly connected (through single or multiple hops) to a receiver node with control and data processing capabilities. The receiver nodes in the network are assumed to be joined in a wireless or wired backbone network.

Our approach is based on the well-known observation that the HVAC-relevant environmental variables are highly correlated in open areas [5] [16]; we assume that sampling just a single node in a given area is sufficient for up-to-date data.

We limit the scope of our contribution to harsh network conditions where: (1) neither simple retransmissions nor multichannel techniques are sufficient; (2) the outer layer of the network is sparse and therefore not well suitable for routing or flooding based approaches; (3) either node faults are present, or there are link faults that cannot be easily fixed by interference-mitigation mechanisms (*e.g.*, caused by shadowing or prolonged, intensive interference).

Results. We compare the system with two state-of-art CSMA-based MAC protocols — the single-channel ContikiMAC and multichannel MiCMAC, and show that under good conditions the performance is not compromised by the adaptive mechanism, while under faults the system avoids using bad nodes and bad channels and successfully bounds the maximal information age on the receivers while keeping the average duty cycle $< 1\,\%$. The evaluation is based on extensive (5760 h long) simulations run on top of 48 h long testbed packet traces, and a 48 h experiment in the testbed.

Structure of the paper. We start with presenting preliminary experiments aimed to characterize network-level properties in a real-world office building (Section 2). We base the design of our system on these properties and present it in Section 3. The system is evaluated in Section 4. The next, Section 5, reviews the state-of-art work on building reliable sensor networks and positions our work among them; finally, Section 6 concludes this paper.

2 Wireless link properties in an office building

2.1 Setup

We deploy 17-node sensor network in a Uppsala University campus building (Fig. 1) to investigate link quality changes, obtain packet traces for running simulations, and, finally, to use it for testing our system on real hardware. The network coexists with the university's IEEE 802.11g network; there are 35 WiFi access points installed in the four-floor building where our test network is set up. The WiFi network is heavily used during the regular working hours, in part by students who have lectures in the same building.

We use Zolertia Z1 sensor nodes equipped with CC2420 radio. To measure the link quality we use a custom Contiki-based test application that broadcasts out bursts of packets (each 66 bytes) from each node in the network in a sequential order; all non-sending nodes listen and log the PRR, RSSI, and LQI. We do not use any MAC protocol, turn off hardware ACK, and set the CCA threshold to the Contiki-default value, *i.e.*, -90 dBm. Consequently, the PRR data incorporates both packet reception failures and packet transmission failures, which happen whenever energy level higher than -90 dBm is detected before starting a transmission. On the other hand, some packets are still received even on weak links with energy levels below this bound: the CC2420 chip has -94 dBm Rx sensitivity [28].

During the data collection phase we ran two types of tests: first, multiple week-long measurements using long packet bursts (10 000 packets within 1 minute per burst), each exploring a single channel (Fig. 2 shows an example link on

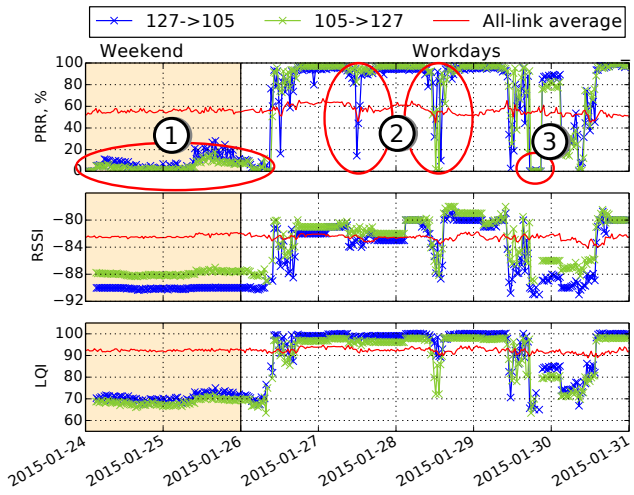


Figure 2: **An representative example of week-long dynamics in link performance,** channel 20. ① Constantly bad performance during a weekend. ② Performance is erratic during working hours because of increased interference. ③ A prolonged period with almost-zero PRR.

channel 20); second, day-long four-channel measurements, using shorter bursts of 100 packets (Fig. 4, Fig. 3). The second type of test is repeated twice, with 5 month interval (April and September, 2015, Fig. 5).

In the second type of test, after all 17 nodes in the network have completed the sending of their packets, the network-wide active IEEE 802.15.4 2.4 GHz channel is switched according to a pre-computed pseudorandom schedule, which contains channels 12, 15, 18, and 21 with equal frequency. Going through these four channels takes approximately one minute, leading to efficient packet rate 100-per-minute for each of channels on each of the 17 potential receivers.

2.2 Observations

2.2.1 Interference

The traces show significant presence of external interference on all selected channels, especially active during daytime and working days, as opposed to nights and weekends. The interference pattern is recognizable by unstable PRR, RSSI, and LQI metrics together with increased RSSI and decreased PRR. However, the performance of the test network is *not*, on the average, substantially higher during weekends. Although the number of channels capable of bidirectional communication (*i.e.*, non-zero PRR in both directions) then is significantly higher, all-link average PRR remains stable as shown in Fig. 6. For instance, the link in Fig. 2 shows erratic performance during the week, but decreased, although stable, PRR and RSSI levels during weekend. Additional RSSI sampling (not shown here) confirms that WiFi interference in form of periodic beacon is still present; it is just the user-driven interactive traffic that decreases.

2.2.2 Time diversity

The results show that link performance varies on at least three different timescales:

1. **Second and minute timescale.** Bullet 2 in Fig. 2 displays an example of rapidly changing link performance. Our measurements show that occasionally a link's performance drops as fast as 50 percentage points of PRR within 3 seconds (drop on an average PRR of 32 packets). This pattern is characteristic for periods when active interference is present.
2. **Hour and day timescale.** Bullets 1 and 3 in Fig. 2 show examples of this: a link dropping close to zero and remaining there for extensive periods; in particular, bullet 3 shows a period when tens of thousands of successive packets are not communicated (each point in the figure is a measurement of 10 000 packets).
3. **Month timescale.** As an example, the channel 15 shows the best PRR in April measurements from all channels on link from node 107 to node 129 (Fig. 5a), but drops to being unusable in September.

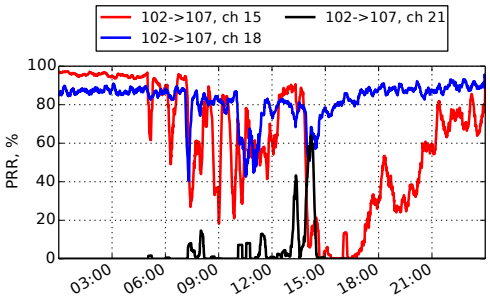
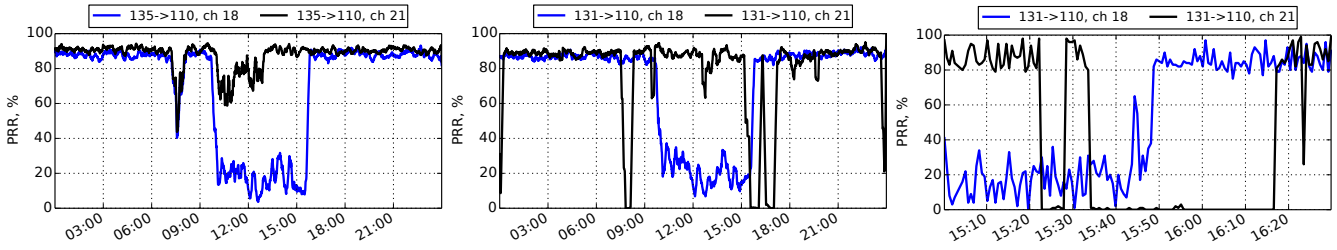


Figure 3: **A representative example of link performance on a single sender→receiver link on multiple channels.** Channel 15, initially the best one, at approximately 15:00 rapidly becomes unusable. Subsequently, channels 18 and 21 have similar quality for some time; however, a system that uses history from a longer-period would be able to predict that channel 15 should be preferred over 21.

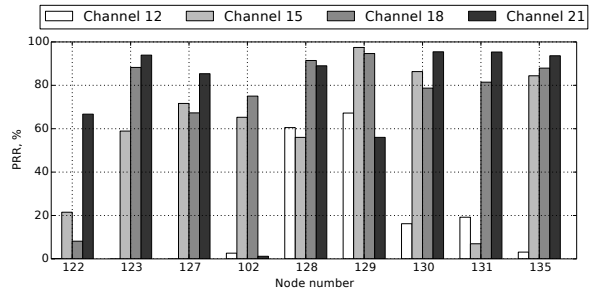
While the first type of time dynamics could be fixed by increasing the number of retransmissions or by other interference mitigation techniques [18] [12] [13], the second would be nontrivial to work around. Furthermore, the situation is complicated by the fact that the links fail due to several different causes. For example, the extensive drops on channel 21 in Fig. 4b are caused by interference: the RSSI of the few received packets is high. In contrast, the bad weekend performance in Fig. 2 is caused by a weak link, as the RSSI is close to the receiver sensitivity threshold. While the first type of drop would require an interference co-existence mechanism [13], for the second, packet redundancy or datarate reduction would be better solutions. If state-of-art CSMA MAC protocols such as ContikiMAC were used, then both also would gain from CCA threshold adaptation [13]: however, for the first one the transmission detection threshold should be increased, while for the second decreased — the default ContikiMAC threshold is -90 dBm, so ContikiMAC on nodes 105 and 127 would *not* be able to detect and receive even those few packets that are getting through in the period 1 (Fig. 2).

At least some of these periods are caused by a node repeatedly failing to start transmission because of CCA check failures. We performed an additional quick experiment to confirm this: we disabled the CCA check before transmission on node 135, and observed that the performance of the node 135→node 110 link was improved from 0 % to almost 100 % PRR. However, this configuration is at odds with the expected fairness properties of the network.

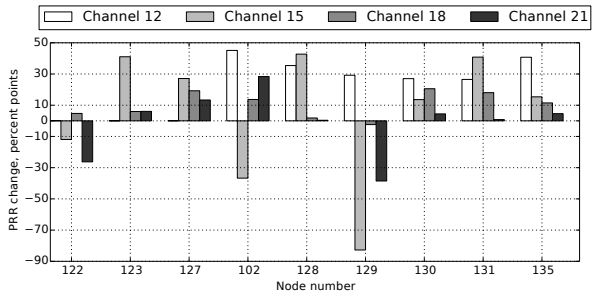


(a) From node 135 (b) From node 131 (c) From node 131, between 15:00-16:30

Figure 4: **A representative example of link performance between a single receiver and multiple senders.** Links from both nodes are equally affected by interference on channel 18, suggesting a receiver-nearby interferer. However, on channel 21, interference affects the link from node 131 more than the link from node 135. Therefore node 135 is the preferable source in this pair; the other node, 131, has two prolonged periods with no usable channels. (Channels 12 and 15 have permanently bad quality on both of these links.)

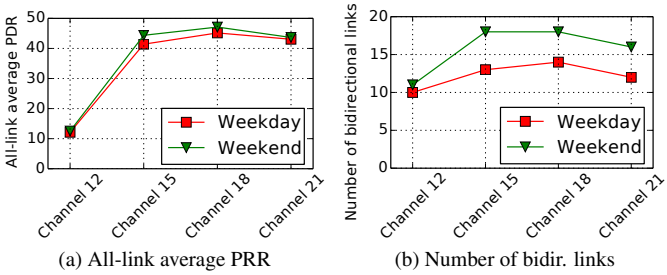


(a) Measurements in April



(b) Measurements in September, shown as differences from the April measurements

Figure 5: **(a) Average performance of different channels on different receivers. (b) Long-term per-channel changes in link performance.** The graphs show 24-hour average PRR from traces recorded on two different days with 5 month offset.



(a) All-link average PRR (b) Number of bidir. links

Figure 6: **Differences in average performance in a working day and in a free day.**

Finally, the differences between months are large enough to suggest that relying on single-time measurements is not sufficient to protect the network from link-level faults in the future. Furthermore, in the whole network there are 25 different channel-links that have good ($\geq$ 95 % PRR) performance in either April or September alone, but just 11 channel-links that have good performance in both months.

2.2.3 *Space and frequency diversity*

Most of the channels perform well in some parts of the network, and worse in other parts (Fig. 5). This suggests that despite the modest spatial dimensions of the network region-specific channel allocation would be beneficial.

Additionally, some of the links are highly asymmetrical: for example, the link between nodes 110 and 135 has stable $\geq$ 80 % PRR in one direction, but close to zero PRR in the other direction. As discussed in the previous subsection, the problem in this area is caused by an active WiFi access point near node 135; the node does not transmit any packets for many minutes because of failing CCA checks.

Some of the nodes have periods when none of the four investigated channels provide acceptable performance (for example, node 131 in Fig. 4), so that they are effectively disconnected from the rest of the network.

One limitation of our study is that just four channels out of 16 were measured. In particular, channel 26 is interference-free in this testbed, therefore it would be a better choice for a real deployment. However, it is not safe to assume that in real world buildings there always are some channels that are both interference-free and free-to-use.

The probability of finding at least one usable channel is increased as the number of potentially active channels is increased. However, even if a larger number of active channels are available, using them comes at a cost. In particular, MiCMAC is not optimized for that; with the default protocol parameters, using n channels makes all broadcast communication in MiCMAC, as well as unicast communication between phase-unsynchronized neighbors, n times more expensive.

2.2.4 *Summary*

Minute, hour, and even day-long link failures is a fact in this network; they are caused by external interference and by signal fading. Some multichannel links occasionally become completely unusable. Mitigating the risks is not trivial: as the performance is significantly different between different months, *a priori* selection of good nodes would not be a reliable solution; as the performance is region-specific, same applies to the selection of good channels.

3 Design of the adaptive system

3.1 Application scenario and requirerements

We consider an HVAC application with multiple control loops in the network: temperature data is measured by sensor nodes and must be delivered to receiver nodes, which are connected either to a backbone network or directly to heater controllers. This application is part of a building automation system; it must be built on top of a network that is unreliable as described in Section 2.

The receiver nodes require timely information reflecting the current state of the environment in order to make accurate decisions. For example, to ensure inhabitant comfort, room temperature data should be received by a heater controller with delay no larger than a few minutes.

Consequently, the most important performance metric for such an application is the *maximal age of information* on receiver nodes. We define the *age of information* at time *t* on node *n* as *the difference between t and the origination time of the message with most recent origination time among the messages received on the node n*. This parameter is dependent on three factors: data packet origination frequency, transmission delay, and end-to-end PDR. The age of information is clearly distinct from the *packet delay* metric: while the delay measures the time it takes for a single packet to arrive from the source to the destination and is typically less than a few seconds, the age of information can easily be as high as multiple hours if no packets are getting through.

For this kind of application, the required bound on the age of information is on the order of minutes [19]. Therefore, only a prolonged period with low PDR can realistically lead to a violation of this bound. As a result, the application requires that long periods without packets are avoided.

We assume that redundant sensor nodes are deployed for this application. This assumption is justified as the operators of the network have an incentive to have more than one device for each sensing area in order to avoid time when the network is at risk (i.e. has a single point of failure). Furthermore, they also have incentive to deploy more than *two* devices in order to make maintenance operations schedule-based, rather than need-based, and therefore less expensive [9] (maintenance operations should be performed immediately if the network is at risk). As the price of sensor network hardware is expected to decrease in the future, hardware costs are expected to make up progressively smaller part of the total cost of ownership.

Finally, we assume that temperature measurements are highly correlated within each area [5]. It means that, for many applications, data from any single one of sensor nodes in the area is directly sufficient for the receiver. More demanding applications can do data post-processing to better reconstruct target location data from nearby measurements: modern techniques such as compressive sensing show accurate results even in absence of complete information [16].

3.2 Design overview

We use a combined approach the exploits both spatial and frequency diversity: the former through multiple source nodes, the latter through multiple channels.

The two goals of our design are: (1) to achieve bounded maximal information age on the receiver nodes even in presence of link and sensor node failures; (2) to optimize the lifetime of the network by remaining energy efficient during periods with good communication quality.

To deal with these conflicting goals, our approach is to try to activate those and only those source nodes that are required to satisfy (1), while deactivating the rest of the nodes to contribute towards (2). The challenge is to decide which nodes to activate and keep activated. We use end-to-end expected transmission count (ETX) with hysteresis and a slow temporal decay as the primary metric for this decision.

Section 2 observations are instrumental to our design:

- **Link quality changes are regional.** We allow to switch the nodes used as the active sensor data sources, so that a single bad node does not cause the system to fail.
- **Link quality changes are hard to predict.** We use a reactively adaptive approach, and rely on application data and control messages to track the current state of the links in the paths from sensor to receiver nodes. The metrics on the links the system does not measure slowly decay back to their initial values to reflect the system's uncertainty about their actual state.
- **Link quality changes may be rapid.** We allow multiple sensor nodes to be simultaneously active per a single receiver to increase the probability that at least one of them succeeds in delivering its measurements. We always preselect and activate a single backup node, unless the network conditions are good ($\geq$80 % end-to-end PDR *and* majority of channels not blacklisted on the selected active node). In periods of unsatisfactory performance, we allow more than two active nodes. We bias the control protocol towards more aggressive node activation and less aggressive deactivation; it avoids extensive control message floods, while still remains capable of rapidly restoring the operation of the system in case the active node fails.
- **Channel quality is regional.** We add per-link channel blacklisting to improve upon the single network-wide channel hopping schedule in MiCMAC.

The decision which sensor nodes to activate is made on the receiver node autonomously; it is not necessary to communicate with a network-wide central agent for this. The receiver node learns the path quality from a sensor node by looking at the number of packet (re)transmissions required in the path. This number is included in data messages; this is the only extra data that our scheme requires to transmit, and is bounded by $\log_2 max\_transmissions$ bits per packet.

The ETX metric in our system avoids links that are completely asymmetric, as it is primarily based on unicast data messages for which link-level ACK are sent. If the sender node does not get an ACK for a packet, it retransmits the packet, allowing the receiver node to update the ETX on each subsequent reception. By default, we let opposite-direction control messages to influence the path ETX metric as well; however, as Section 4.3.2 shows, doing that is not always beneficial and in some cases should be disabled.

To make the decision whether to activate a backup node, we additionally look at the number of blacklisted channels, because when the number of good channels on the primary

active node is small, the system has a lower reliability margin, so the backup node should be activated even if the ETX is good. In this way we combine the frequency-diversity and spatial-diversity aspects of our approach.

Additionally, we propose a novel algorithm that allows the receiver node to operatively learn which channels are blacklisted on the sensor node without exchanging any data with it. The algorithm is sufficient when data producer and consumer nodes are directly connected; for the multihop case, the number of blacklisted channels should be included in data messages (4 additional bits per packet).

3.3 Software preliminaries

We build our software on top of Contiki operating system; it includes ContikiMAC, a highly efficient low-power MAC protocol [6]. ContikiMAC uses periodic radio wakeups and CCA sampling in order to detect ongoing transmissions. A ContikiMAC transmission consists of a number of packet strobes, each followed by a short period of silence in which an IEEE 802.15.4 ACK is expected. A unicast transmission is stopped when it is acknowledged by the receiver. A broadcast transmission goes on for the duration of the whole slot length to ensure that all neighbors have the chance to wake up and hear it. Unicast communication with known receivers in ContikiMAC is optimized by phase locking: first, the sender learns the schedule of the receiver based on the time when an ACK is received from it; afterwards, the sender starts its next transmissions right before the receiver wakeup time.

MiCMAC is a multichannel extension of ContikiMAC; it employs pseudorandom channel hopping. Unicast communication with known receivers in MiCMAC is optimized similarly to ContikiMAC: both phase and channel locking are used. The active channel for a known neighbor is deduced from a pseudorandom hopping sequence that is shared by all nodes. As a consequence, the overhead of MiCMAC for unicast communication is small. However, broadcast messages either have to use a separate channel, or be sent for n times larger duration. Evaluation results of MiCMAC does not show large gains from using a separate channel for broadcast [21], therefore we apply the alternative, shared channel approach in this work. We chose MiCMAC because its implementation is available and has been experimentally evaluated with good results, showing better performance than Chrysso [15], another state-of-art CSMA multichannel protocol.

For the transport and routing layers we rely on Contiki Rime networking stack. On top of Contiki, we use a custom middleware library [8] that implements sensor data collection, and handles data connections between sensor and receiver nodes. It provides a control message API for remote activation and deactivation of these connections.

3.4 Frequency diversity

MiCMAC only supports network-wide channel hopping schedule. To handle the regional differences in channel quality, we extend it with per-link channel blacklisting.

The adapted version of MiCMAC uses transmission success rate as the blacklisting metric. The past performance is incorporated in this metric using an EWMA (exponentially weighted moving average) filter. A channel is blacklisted if and only if its quality metric is less than 40 % of the best channel's metric. In this way, at least one channel always remains active. The parameter α of the EWMA filter is set to 0.2, causing blacklisting to happen when, for example, in the initial good state five transmission in a row fail.

If no packets are transmitted on a channel, its quality metric slowly (EWMA with $\alpha = 0.005$) decays back to the initial good value of 1.0.

A successful transmission is detected when an ACK from the neighbor node is received, therefore this metric describes the combined quality of the link in both directions. A transmission failure indicates that either the transmission was never started because of a failed CCA check in the radio driver, or that the packet or its ACK were transmitted, but not received.

A CCA failure signals either external or internal interference. This blacklisting scheme is designed to detect and avoid the former; however, it is not compromised by the presence of the latter, because the blacklisting threshold is relative to the best channel. Transmission failures due to internal interference (contention) are mathematically expected to be evenly divided on all active channels, therefore they are all expected to be penalized by an equal amount.

A node never transmits packets on channels it has blacklisted; however, the node still listens to all channels, and may receive packets on blacklisted channels, as neighboring nodes do not share their blacklisting information.

An ACK for a packet is always transmitted on the same channel as the packet. Switching the channels for the ACK transmission would require that the transmission is done in software rather than by radio chip, but sending software-ACKs is a time-consuming operation on low-power sensor nodes. In particular, on mote-class constrained devices it would require that the ACK waiting time in MiCMAC is increased, which in turn would reduce the energy efficiency of this protocol.

The receiver node should be capable of learning which channels are blacklisted on each active sensor node, so that it can detect a high number of bad channels and proactively activate a backup node. We show that no additional information has to be included in data packets for this. Instead, we use a specific channel-selection algorithm on the source node of each packet. The channel of a packet is a function of four variables: the packet's sequence number, the retransmission attempt, the node ID, and the blacklisted channels. Essentially, we use the first three variables to compute an ordering of all network-wide active channels, and then take the first non-blacklisted channel from that ordering. For the transmission of the packet on a specific channel, the source node waits until MiCMAC switches the radio on the destination node to that channel.

Upon reception, the destination node extracts the source address, the sequence number and the retransmission number from the packet; all three are always included as required for path ETX calculation, as described in Section 3.2. The node uses them to compute the channel priority ordering for that packet, and then compares it with the channel on which the packet is received. It marks as blacklisted all channels that in the ordering are situated before the reception channel.

On reception of several packets, all blacklisted channels are eventually learned with high probability. For example, numerical simulations show that if 50 % out of 8 total channels are blacklisted then after 20 transmissions all four blacklisted channels are learned with 95.5 % probability.

To efficiently compute the channel ordering on mote-class devices, we use pre-generated lookup tables — one per each possible number of total active channels. Both each row and each column of the lookup table is a pseudorandom channel hopping sequence that contains each active channel at least once. For example, this lookup table is used in our implementation for 8 active channels (with channels offsets numbered from 0 to 7):

$$A_{ij} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 4 & 6 & 2 & 3 & 1 & 7 & 0 \\ 1 & 6 & 5 & 0 & 7 & 4 & 3 & 2 \\ 2 & 3 & 7 & 1 & 5 & 0 & 4 & 6 \\ 4 & 2 & 3 & 7 & 0 & 6 & 1 & 5 \\ 3 & 0 & 1 & 6 & 2 & 7 & 5 & 4 \\ 6 & 7 & 4 & 5 & 1 & 2 & 0 & 3 \\ 7 & 5 & 0 & 4 & 6 & 3 & 2 & 1 \end{pmatrix}$$

We use the randomness in this table to enable pseudorandom channel hopping between sending each two subsequent packets, including two retransmissions of the same packet.

The channel ordering of a packet is defined to be a column in this table:

$$(a_{ij}, a_{(i+1)_{mod N}\, j}, \ldots, a_{(i+N-1)_{mod N}\, j})^T,$$

where the indices of the first element a_{ij} is determined by a linear function:

$$i \times N + j = R \times \text{seqnum} + \text{tx\_attempt} + \text{node\_id},$$

where R is a number larger than the maximal number of retransmissions and relatively prime to the number of active channels; $R = 9$ in our implementation. The `node_id` variable is included because packets with equal sequence numbers are likely to be generated at the same time on several nodes; to reduce collisions, neighboring nodes should *not* try to transmit these packets using the same channel.

Intuitively, new packets and new retransmissions advance the indices first column-wise and then row-wise. Blacklisted channels advance the indices row-wise.

This mechanism trades off reliability for latency. The channel blacklisting is not expected to increase energy consumption compared to baseline MiCMAC and ContikiMAC, but *is* expected to increase delay, as the sender node has to wait while the receiver node starts listening to the specific channel. This has implications on the packet retransmission timeout in higher network layers: for MiCMAC on n channels with channel blacklisting, we increase this timeout by a factor of n.

Finally, we optimized some parameters of Contiki network stack to make MiCMAC less aggressive, as by default MiCMAC is more vulnerable than ContikiMAC to high-energy usage on bad links:

- how tight it holds on a neighbor's phase-lock: up to 16 packets or 2 min with no acknowledgments;
- the number of maximal retransmissions if the neighbor's phase is not known: to count a single failed MiCMAC transmission as three MAC-layer transmissions;
- MAC-layer backoff duration: increase by a factor of 1.5.

3.5 Spatial diversity

The system achieves spatial diversity by sensing the same physical phenomenon in the same area on more than one node. To achieve higher energy efficiency, we require that only the nodes with the best path qualities communicate the measurements to the receiver nodes, unless the system is in an alarm state.

We assume that the time is loosely (on the order of seconds) synchronized between the sensor nodes, so that the sequence number of a data packet has 1:1 mapping with a network-wide time interval, regardless on which node it is generated.

The decisions to activate and deactivate connections are made on the receiver nodes. A receiver node keeps ETX information about the path quality to each of the potential sensor nodes. These ETX values are updated when:

1. A packet is received from a sensor node. If a duplicate packet is received from that node, its contents are ignored, but the ETX is still updated. This is sufficient to recognize single-direction-only links as unusable.
2. A packet is transmitted to a directly connected sensor node and either ACKed by the destination, or the maximal number of retransmissions is reached.
3. Once every sensing period:
   (a) For active nodes from which an message was expected, but not received in the *previous* period: to mark the transmissions as failed.
   (b) For all inactive nodes: to slowly decay ETX to the average expected ETX value (defined as $2.5 \times$ *hopcount* in our setup).

For a packet not successfully received or transmitted, the maximal number of retransmissions + 1 is used as the value for the ETX update.

The ETX is calculated by applying an EWMA filter of past values. The filter is biased to react on failures faster: $\alpha_{good} = 0.05$ is used for successful transmissions (number of retransmissions less than $bad\_num\_tx \times hopcount$), and $\alpha_{bad} = 0.15$ on other transmissions. α_{bad} is in particular selected so that after just 3 missed packets the ETX value of a link crosses 4.0 (a blacklist threshold) if starting from a perfect link with 1.0 ETX value. α_{good} in turn allows the ETX to change from 5.0 to 1.0+ε (with $\varepsilon < 0.01$) if, *e.g.*, 120 packets succeed on the first attempt (corresponding to 30 minutes if 15 second packet interval is used).

The decay α is set to 0.0001, a small value to reflect the absence of any information, requiring many hundreds of updates to significantly change the ETX metric.

The nodes which should be activated are re-elected each sensing period (15 seconds in our setup). Depending on whether the network is detected to be in a safe state, the election algorithms is run either one or two times. A network is defined to be in a safe state if the active node has ETX value less than $1.25 \times hopcount$ (*i.e.*, ≥ 80 % end-to-end PDR) and, additionally, the majority of channels are not blacklisted on it.

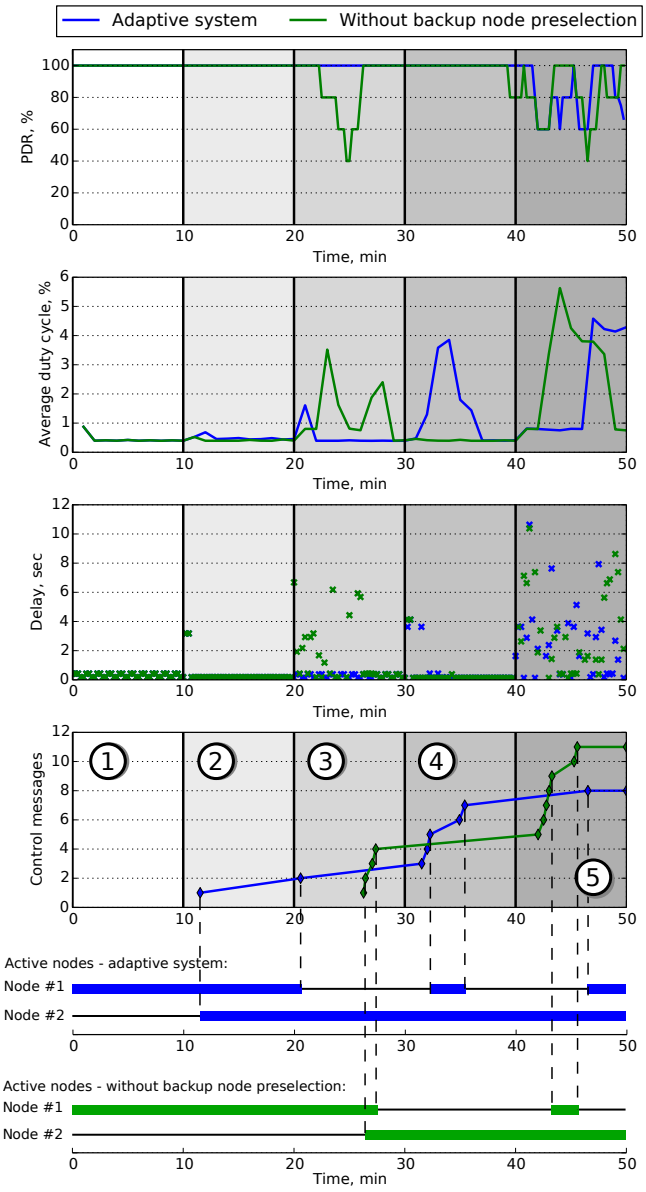


Figure 7: **Dynamic behavior of the adaptation mechanism.** Data from simulations of 2-channel, 2-sensor node scenario, where initially all four channel-links have 90 % link-level PRR, and after each 10 minutes, one the channel-links is degraded to 10 % PRR.

A node is eligible to be elected if it is not blacklisted. A node is blacklisted when its ETX value becomes larger than the value of $blacklist\_threshold \times hopcount$. Nodes are periodically unblacklisted, with period length proportional to their ETX value. Backup nodes additionally must have ETX value lower than $backup\_threshold \times hopcount$ to be eligible. The blacklisting helps to avoid reconsidering nodes that have shown bad link quality. However, in harsher conditions this feature might have to be disabled: see Section 4.3.1 for a discussion.

The elections are won by the candidate node who has the lowest hysteresis-adjusted ETX value: the ETX of already activated nodes is divided by 1.5. The adjustment is done to make the system more stable and reduce the network churn. The adjustment is applied only when comparing between node ETX values, not in the eligibility determination stage.

The EWMA-filtered ETX value is a good way for tracking medium and long-term performance of the system. However, the system must react faster when the application-defined bound B on the maximal information age is under threat. To enable this, we include a faster-reaction option in the protocol: if the receiver node detects that a relaxed bound of $B/2$ has been violated, it switches the system to the alarm state. In this state, the system first unblacklists all nodes, and then forces the election of a backup node. In this case, nodes with up to $forced\_backup\_threshold \times hopcount$ ETX value may be elected.

After both elections are completed, the new status (active/inactive) of the nodes is compared with the old status, and a control message is sent to each of the nodes activated by these elections. However, to avoid the extensive traffic in case a formerly active node becomes unreachable, we do not send deactivation messages immediately after the elections. Instead, we wait until a data message from the deactivated node is received and reply to it with a NACK: a deactivation message. Even more, a node may be re-elected to be active before a NACK has been sent to it: in this case the sending of the activation message is skipped as well.

Additionally, if the system is in the alarm state, deactivation messages are never sent. In this way the system may keep *more than two* nodes active if the performance is unsatisfactory.

3.6 An illustrative example

Fig. 7 shows how the adaptive system reacts when link-level faults are introduced in the network. Unless all channels on all links are bad (as in period ⑤), the system is able to recover to 100 % PDR within minutes and to operatively re-normalize node radio duty cycle.

In the following analysis, the bullets reference the five periods in Fig. 7 — from "both channels on both links have 90 % PRR" in period ① to "both channels on both links have 10 % PRR" in period ⑤.

The blue graph in Fig. 7 shows the behavior of the complete adaptive system:

② A bad channel on node #1 is detected and node #2 proactively activated as a backup.

③ Node #2 with two good channels advances from backup state; node #1 is deactivated.

④ A bad channel on node #2 is detected; the controller briefly activates node #1, detects that is has no good channels and operatively deactivates it.

⑤ No good channels exist, therefore both nodes are kept active to maximize the PDR.

The green graph in Fig. 7 shows the behavior of the system configured to not to activate backup nodes:

③ After several minutes of mediocre performance, the ETX of node #1 finally falls below the default (initial) ETX value of node #2, so that node #2 is activated.

⑤ No good channels exist; the scheme is going to slowly alternate between both nodes, periodically sending control messages.

Through this comparison we show that in this example:

- The version of the system that includes backup node pre-selection is able to achieve higher PDR while maintaining similar energy usage.
- Using data about blacklisted channels helps to make better decisions about which nodes to activate (*e.g.*, to activate node #2 in period ②). In this way, the frequency and spatially adaptive aspects of the system both working together give better results than they would give if both were used separately.

4 Experimental evaluation

4.1 Application scenario

We use the testbed network with the intention to mimic the essential communication-related aspects of a building automation system as described in the introduction of this paper. For single hop experiments we select three nodes (105, 107, and 110 in Fig. 1) for the role of data consumers, and nine nodes for the role of data sources. Each receiver node has three associated data source nodes. We use Contiki Rime stack with maximum of 7 MAC-layer retransmissions. For the adaptive approaches, connections between sensor and receiver nodes are dynamically activated and deactivated; for the other approaches, the state of these connections are set from program code at initialization.

In line with the assumptions described in the introduction about slow-changing environmental variables, we set the sampling period of this application to $S = 15$ seconds and maximal information age to $6S = 90$ seconds, leading to fault detection time of $3S = 45$ seconds: the receiver node enters the alarm state if three consecutive packets from all active sensor nodes are missed. For the multihop case, these values are increased by a factor of *hopcount*.

In the experiments described in this section, each sensor node is potentially sending to only one receiver. However, the API of the runtime system allows multiple receivers for a single source node [8], with independent activation and deactivation of each dataflow.

4.2 Simulations

4.2.1 Simulation setup

We use Cooja [22] coupled with RealSim [26] for trace based-simulations and run it on top of packet traces obtained from our test network (Section 2). RealSim is reported to provide high fidelity between MAC protocol behavior in simulations and testbeds [26]. This plugin takes dynamic radio-link performance statistics from a specifically prepared trace file that consists of a lists of events (*i.e.*, changes in radio link properties). During an execution of the simulation the plugin dynamically creates radio links between nodes and dynamically updates PDR, RSSI and LQI values on each link. Asymmetric links are supported in this setup: each direction of a link gets independent performance metrics.

To enable this simulator-based experimentation we extended the functionality of the Cooja *DirectionalGraph* radio medium (used by RealSim) with the support for links on specific radio channels and fixed a known bug in Cooja's existing support for multiple channels. These improvements have been merged in the mainstream version of Contiki.

The simulations consist of multiple experiments, each run for 24 h first on April traces and then for 24 h on September traces. For each of the eight different solutions we evaluate at least 9 experiments; these experiments have different starting configurations so that the space of possible initial conditions is fully covered. For example, for multichannel solutions we change which sensor node is initially active; for single-channel solutions we vary both the node and the channel. When calculating the results, in each experiment we ignore the first 5 minutes (20 data packets) to allow the adaptive system to learn link qualities.

We use these constant values in our implementation: $etx\_blacklist\_threshold = 4.0$, $etx\_backup\_threshold = 5.0$, $etx\_forced\_backup\_threshold = 7.0$, and $bad\_num\_tx = 5$. The system is quite robust to selection of these parameters: we experimented with re-running all the adaptive system simulations with parameter values set to 60 %, 80 %, 120 %, and 140 % of those above, and got results with no more than 0.1 percent-point difference in average PDR and no more than 0.15 percent-point difference in average duty cycle.

4.2.2 Single hop simulation results

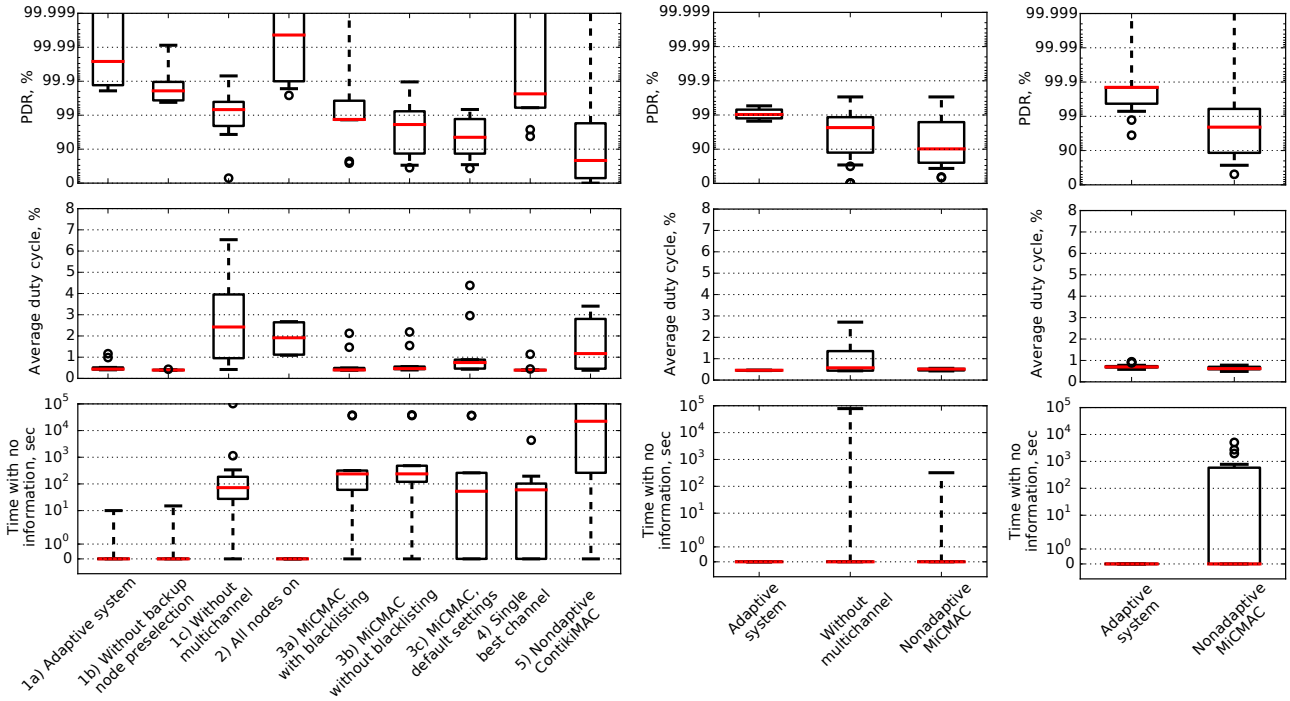
We compare the following dynamic source node selection approaches (Fig. 8a): (1a) the full adaptive system, (1b) the system without backup node pre-selection, (1c) the system on top of single-channel ContikiMAC ("without multichannel") with a number of baseline approaches: (2) keeping all source nodes permanently active, (3a) MiCMAC protocol with adaptive channel blacklisting, (3b) unmodified MiCMAC, (3c) unmodified MiCMAC with its default parameter values, (4) optimal dynamic selection of the single best channel, (5) single-channel ContikiMAC.

The complete adaptive system (1a) shows good results in all three metrics. First, it shows 99.97 % median PDR, 99.94 % average-case PDR, and 99.81 % worst-case PDR (among all different 48-hour experiments, 11458 out of 11480 packets). It was almost always able to keep the information age below the bound of 90 sec: the total duration of violations is 10 seconds among the 432 h total duration (0.0006 %), and shows median average duty cycle of 0.44 % (median maximal duty cycle is 0.58 %).

The version without backup node selection (1b) is not able to give as high PDR, although it shows better energy usage.

Similar and even better PDR is shown by the always-on system (2): 99.996 % median, 99.94 % average, and 99.74 % worst-case PDR. However, its median average duty cycle is 1.92 % and the median maximal duty cycle is 5.27 %. With such heavy radio usage, the sensor nodes with bad links would soon run out of energy. In contrast, the adaptive system disables these nodes; in case link conditions later change, it will re-enable them back again and thus significantly extend the lifetime of the network. The energy consumption of the receiver is increased as well (from 0.44 % to 0.54 % median) because the node has to receive and acknowledge additional packets and their retransmissions. While simple, this approach is suboptimal.

Hopping over multiple channels shows a clear advantage in this setup. Non-adaptive MiCMAC (3a-c) shows results that are even better than those of the adaptive system without



(a) Trace-based simulation, single hop (b) Trace-based simulation, multihop (c) Testbed, single hop

Figure 8: **Application-level performance,** data from multiple trace-based simulations (48 h each experiment) and testbed experiments (3 h each). The red lines show the median performance, the boxes show the first-to-third quartile range, the whiskers stretch $1.5 \times IQR$ (inter-quartile-range) beyond the boxes, and the small circles show outliers. *PDR*: end-to-end application-level PDR, accounting packets received from at least one out of three sensor nodes. *Average duty cycle:* average in the 4-node cluster for single hop results, average in the whole network for multihop results. *Time with no information:* total duration of periods in each experiment when the age of information on a receiver is over $90 \times$ *hopcount* seconds.

multiple channels; in particular, MiCMAC has lower duty cycle. However, neither channel hopping nor source-node switching alone (1c) can provide sufficiently good performance; for them, at least one violation of the bound on the age of the information is the typical behavior, rather than an exception.

The graphs confirm that both the channel blacklisting feature and the adapted set of parameters of MiCMAC (Section 3.4) are advantageous for this application in this testbed compared to its default code.

The optimal dynamic selection of the single best channel (4) performs slightly better than MiCMAC, but still not as good as the adaptive system. This approach is a kind of upper bound on what could be realistically achieved with an approach based on asymmetric channel adaptation (as in ARCH [23] and 802.15.4e AMCA [1]). Results from a real implementation would be worse than this, because of necessarily suboptimal and delayed channel selection decisions, as well as control overhead.

Even baseline ContikiMAC (5) is able to achieve 100.0 % PDR if a good source node and a good channel are preselected at design time. However, for such a selection, expert knowledge in wireless network design and a careful predeployment site survey are required; furthermore, by its nature this static allocation is not able to react on unanticipated changes in link quality.

4.2.3 *Multiple hop simulation results*

For multihop experiments we select the node 130 as the single sink in the network, and have two groups of sensor nodes sending data to it simultaneously (three nodes in each group). There are 3-4 wireless hops between the sensor nodes and the sink node (Fig. 1); we assume 3 hops on average and use it as the *hopcount* parameter of the system.

We use channel blacklisting in the whole network. However, the receiver node bases its decisions about whether to activate a backup node solely on the number of good channels on the active sensor node; it does not take into account the blacklisting state of the intermediate nodes.

The multihop results (Fig. 8b) show few surprises. The PDR is lower because of longer paths and increased network contention. The duty cycle remains similar. The bound on the maximal age of information metric is relaxed in this setup. The complete adaptive system is able to keep within this bound, while the single-channel option and nonadaptive MiCMAC fail in some experiments.

4.3 Testbed

4.3.1 *Bringing the system to the testbed*

Inspired by the success of trace-based simulations reported by the authors of RealSim [26], we assumed that deploying the system in a real network would be a breeze. We were wrong. Even though the code could be run on real nodes without modification, our preliminary experiments

showed severe mismatch between simulation and real-world results.

We identified these two causes for the difference:

- RealSim does not support dynamic modeling of background noise [26], therefore cannot capture effects from external interference. The MAC protocols we use are sensitive to presence of interference; specifically, they are going to back off transmissions if medium contention is detected. As a result, in simulator external interference typically results in transmitted and lost packets; in the real world, it results in packets not being transmitted. The difference is important for the channel blacklisting mechanism and for radio duty cycle.
- The time-resolution of the traces is too small to account for second or sub-second *correlations* in channel performance on spatially different links. Such correlations are present whenever the interferer is localized on the receiver side, rather than on the sender side.

To address these limitations we tweaked our design:

- In order to also blacklist channels in which external interference is continuously present, we changed the MiCMAC code to account for a MAC backoff in the same way as for a failed transmission.
- To cope with the correlations in short-term performance, we made the MAC layer more aggressive by increasing the maximal number of retransmissions to 15.

However, even these changes were not sufficient to keep than maximal information age below 90 sec. It turned out that node blacklisting (Section 3.5) is counterproductive in these even harsher conditions. Only after disabling it we managed to obtain a satisfactory performance even in the worst group of sensor-receiver nodes.

The remaining differences between simulations (Fig. 8a) and the final real world experiments (Fig. 8b) can be explained by the facts that real-world runs were shorter (3 h *vs.* 24 h), the system parameters were more aggressive, and the environment itself had slightly changed compared to the days when the packet traces were gathered.

4.3.2 Testbed results

In the testbed experiment we interleave 3 h periods of running the adaptive system with 3 h of running the non-adaptive version of MiCMAC. Both options use the same set of MAC-layer parameters. We keep the system running for 48 h in total (during two workdays), and show each 3-hour test as a separate datapoint in Fig. 8c.

The results (Fig. 8c) show similar median PDR for both approaches: 99.86 % for the adaptive system, 99.57 % for MiCMAC. However, the worst-case PDR is much better for the former: 96.43 % *vs.* 50.71 %.

More importantly, the adaptive solution is able to always bound the maximal age of the information to 90 seconds, while MiCMAC is not: some experiments show hundred-of-second long periods with no information.

Both solutions show similar energy usage, which is higher than in simulations because of higher packet losses and more aggressive retransmissions: 0.70 % and 0.68 % median average duty cycle for the adaptive system and MiCMAC respectively; 1.12 % and 1.11 % median maximal duty cycle.

We investigated the performance of the only experiment where the adaptive system failed to reach at least 98.7 % PDR. The cause was in the design assumption that a sender node should not be activated if its link has bad quality in the receiver-to-sender direction. In the experiment, the receiver node 110 failed to transmit control packets to the sensor node 130 because of failing CCA checks; however, the node 110 could both successfully receive packets and successfully send IEEE 802.15.4 ACKs to the node 130, as sending an ACK does not require CCA. As a result, the system failed to activate the node 130 (the only one with a usable link) for some time. This problem could be avoided by either disabling CCA checks before sending, or by not taking into account receiver-to-sensor direction traffic in the path ETX calculation. Both changes are trivial to implement; however, as they would lead to large repercussions in the behavior of the system, they should be enabled only on some nodes in some specific environments, rather than by default.

5 Related work

Frequency diversity. Exploiting multiple MAC-layer channels for transmissions is a widely used approach that can successfully deal with temporal dynamics of wireless links [11].

In particular, pseudorandom channel hopping treats the link performance as essentially unpredictable and increases delivery rate through retransmitting packets on different channels. This approach is used by protocols such as ISA100.11a [14], WirelessHART [4], and IEEE 802.15.4e-2012 TSCH [1]; they have roots in industry automation systems and are designed to deal with very stringent reliability and latency requirements. These protocols merge channel hopping with TDMA scheduling to additionally minimize collision probability. However, the need to construct a transmission schedule and to have network-wide time synchronization makes them more complex to use and to reason about; additionally, they do not have the property of failing gracefully. In best-effort networks and sparse traffic scenarios, low-power asynchronous MAC protocols are traditionally preferred. Among these CSMA-based MAC layers, approaches that rely on channel hopping include MuChMAC [3], EM-MAC [27] and MiCMAC [21].

A distinct approach is to try to find and use interference-free channels on per-link basis. This idea is exploited by Chrysso [15] and ARCH [23], and is also now included as part of IEEE 802.15.4e-2012 in the form of AMCA (asynchronous multichannel adaptation).

We selected MiCMAC as the basis for our work because it is a multichannel extension of the well-known ContikiMAC, a protocol widely used in the research community and proven to have state-of-art performance in sparse traffic and best-effort networks [20].

Spatial diversity. Wireless networking can cope with localized interference and fading through using nodes placed in different locations to route around the affected areas. In particular, multipath and opportunistic routing provides per-packet diversity [7], thus enabling high end-to-end PDR.

Network flooding is a simple mechanism that is nevertheless known to give very high reliability [17] [10]. However, even optimized flooding that allows simultaneous transmis-

sions (through radio capture effect and constructive interference) still adds non-negligible overhead [10] to flood the whole network, which is hardly needed for building automation where large proportion of the control loops are relatively localized. Additionally, this approach is not trivial to combine with channel diversity.

The basic idea in the current paper — dynamic task allocation on distinct competing agents — is a long-known technique in distributed systems [25]. Our previous work [8] already applied this technique to sensor networks; however, it used a fully centralized task allocation algorithm and did not exploit frequency diversity.

Distinction. Unlike routing and flooding based approaches, our system does not rely on forwarding node diversity. Therefore it is better suited to sparse networks vulnerable to partitioning, in particular, to leaf nodes becoming disconnected from the rest of the network. Unlike the other network-level approaches, our system can also survive source-node failures.

By using channel blacklisting, we improve the performance by ruling out interfered and fading-affected channels on per-link basis. By using channel hopping rather that just adaptive channel selection we maximize the probability that multiple retransmissions successfully delivers a packet when otherwise the delivery would fail under bursty packet losses.

6 Concluding remarks

We have experimentally determined multichannel link dynamics in an IEEE 802.15.4 network in office environment, and based on this study, developed an application-level adaptation mechanism for more reliable sensor data harvesting on multiple receiver nodes within the network.

The mechanism is designed for situations when simple channel hopping is not able to give satisfactory performance, and the design of the network is sparse, therefore not suitable for routing or flooding-based approaches. The mechanism is simple to implement and use, and it can be applied to networks with link faults caused both by interference and shadowing, as well as with node faults.

We show that the mechanism is able to counteract link-level failures and to achieve 99.86 % median PDR in a testbed, as well as to keep the age of the maximal information on receiver nodes within the bound of 90 seconds and radio duty cycle below 1 %.

Acknowledgments

The author thanks Simon Duquennoy and Charalampos Orfanidis for their help and comments on the paper. This work was partially supported by the ProFuN project (SSF).

7 References

[1] IEEE 802.15.4e Wireless Standard — Amendment 1: MAC sublayer. `http://goo.gl/W4HIiF`.

[2] J. Akerberg, M. Gidlund, and M. Bjorkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *IEEE INDIN*, pages 410–415. IEEE, 2011.

[3] J. Borms, K. Steenhaut, and B. Lemmens. Low-overhead Dynamic Multi-channel MAC for Wireless Sensor Networks. In *EWSN*, pages 81–96, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] D. Chen, M. Nixon, and A. Mok. *WirelessHART(TM): Real-Time Mesh Network for Industrial Automation*. Springer, 2010. ISBN: 1441960465.

[5] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, pages 588–599, 2004.

[6] A. Dunkels. The ContikiMAC Radio Duty Cycling Protocol. SICS Tech. Rep. T2011:13, 2011.

[7] S. Duquennoy, O. Landsiedel, and T. Voigt. Let the Tree Bloom: Scalable Opportunistic Routing with ORPL. In *ACM SenSys*, Rome, Italy, Nov. 2013.

[8] A. Elsts, F. H. Bijarbooneh, M. Jacobsson, and K. Sagonas. ProFuN TG: A Tool for Programming and Managing Performance-Aware Sensor Network Applications. In *IEEE SenseApp*, 2015.

[9] M. L. Fairbairn, I. Bate, and J. A. Stankovic. Improving the dependability of sensornets. In *IEEE DCOSS*, pages 274–282, 2013.

[10] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *ACM/IEEE IPSN*, pages 73–84, 2011.

[11] A. Gonga, O. Landsiedel, P. Soldati, and M. Johansson. Revisiting multi-channel communication to mitigate interference and link dynamics in wireless sensor networks. In *IEEE DCOSS*, pages 186–193, 2012.

[12] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Norden, and P. Gunningberg. SoNIC: classifying interference in 802.15.4 sensor networks. In *ACM/IEEE IPSN*, pages 55–66, 2013.

[13] A. Hithnawi, H. Shafagh, and S. Duquennoy. TIIM: Technology-Independent Interference Mitigation for Low-power Wireless Networks. In *ACM/IEEE IPSN*, pages 1–12, 2015.

[14] International Society of Automation. ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications. `https://goo.gl/sXUMDY`.

[15] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso – a multi-channel approach to mitigate external interference. In *IEEE SECON*, pages 449–457, 2011.

[16] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu. Data loss and reconstruction in sensor networks. In *IEEE INFOCOM*, pages 1654–1662, 2013.

[17] P. A. Levis, N. Patel, D. Culler, and S. Shenker. *Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks*. University of California, Berkeley, 2003.

[18] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power Zigbee Networks. In *ACM SenSys*, pages 309–322, 2010.

[19] J. Lu et al. The smart thermostat: using occupancy sensors to save energy in homes. In *ACM SenSys*, pages 211–224, 2010.

[20] M. Michel and B. Quoitin. Technical Report: ContikiMAC vs X-MAC performance analysis. *arXiv preprint arXiv:1404.3589*, 2014.

[21] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multi-channel. In *IEEE DCOSS*, pages 2–9, 2014.

[22] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with Cooja. In *IEEE LCN*, 2006.

[23] M. Sha, G. Hackmann, and C. Lu. ARCH: Practical channel hopping for reliable home-area sensor networks. In *IEEE RTAS*, pages 305–315, 2011.

[24] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2):215–233, 2003.

[25] S. M. Shatz, J.-P. Wang, and M. Goto. Task allocation for maximizing reliability of distributed computer systems. *IEEE Transactions on Computers*, 41(9):1156–1168, 1992.

[26] M. Strübe, F. Lukas, B. Li, and R. Kapitza. DrySim: simulation-aided deployment-specific tailoring of mote-class WSN software. In *ACM MSWiM*, pages 3–11, 2014.

[27] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. EM-MAC: A Dynamic Multichannel Energy-efficient MAC Protocol for Wireless Sensor Networks. In *ACM MobiHoc*, pages 23:1–23:11, New York, NY, USA, 2011. ACM.

[28] Texas Instruments. CC2420 datasheet. *Reference SWRS041B*, 2007.

[29] Y. Wu, K. Kapitanova, J. Li, J. A. Stankovic, S. H. Son, and K. Whitehouse. Run time assurance of application-level requirements in wireless sensor networks. In *ACM SenSys*, pages 197–208, 2010.