

MOR: Multichannel Opportunistic Routing for Wireless Sensor Networks

Peilin Zhang*

Olaf Landsiedel†

Oliver Theel*

*Carl von Ossietzky University of Oldenburg, Germany

†Chalmers University of Technology, Sweden

{peilin.zhang, theel}@uni-oldenburg.de olafl@chalmers.se

Abstract

Wireless Sensor Networks (WSNs) share the 2.4 GHz ISM band with a number of wireless technologies, such as WiFi and Bluetooth. This and external interference from electrical devices, such as, for example, microwaves, deteriorate the reliability of many routing protocols in WSNs. Multichannel communication strategies allow routing protocols to provide reliability in presence of interference.

In this paper, we propose robust, reliable, and energy-efficient Multichannel Opportunistic Routing (MOR) for WSNs. MOR employs both opportunistic routing and opportunistic multichannel hopping strategies, in order to improve the robustness of the network to interference. Furthermore, it empowers MOR to take advantage of not only the spatial and temporal diversities as traditional opportunistic routing in WSNs does but also the frequency diversity. We implement MOR in Contiki and conduct extensive experiments in the FlockLab testbed.

Under interference MOR provides an end-to-end packet delivery ratio (PDR) of more than 98%, while other protocols such as, for example, ORPL, obtain a PDR of merely 25%. Additionally, MOR's duty cycle stays below 2% for these settings and latency is less than 2 seconds. In interference-free scenarios, MOR achieves a performance similar to our baseline protocol ORPL, with only an approximately 0.3% increment of the duty cycle.

1 Introduction

Over the past decade, Wireless Sensor Networks (WSNs) began to play a significant role as an enabling technology in a large number of applications, including healthcare, industry and agriculture. Due to the limited number of radio channels in WSNs, sensor nodes share the 2.4 GHz ISM band with each other, as well as with other wireless technologies

such as WiFi, Bluetooth, to name but a few. As a result, there exist not only internal interference within the network but also plenty of cross-technology interference (CTI) [1]. For instance, under interference from WiFi devices and Microwaves, the performance of the X-MAC protocol can degrade by over 50% [2, 3], resulting in high network latency and reduced reliability.

Multichannel hopping schemes in WSNs can efficiently mitigate the interference, as shown by a number of existing approaches, e.g., [4–11]. By exploiting the frequency diversity, these approaches are able to improve the reliability and robustness against internal interference within the network as well as external CTI. Meanwhile, a number of challenges arise: As more channels are involved in the communication, the power consumption increases accordingly, e.g., due to channel allocation and switching. Furthermore, the time-to-rendezvous (TTR) between sender and receiver is another crucial factor when utilizing multiple channels in duty-cycled WSNs, that indirectly determines the end-to-end latency of the whole network. We argue in this paper, that most state-of-the-art multichannel protocols for WSNs fail to provide the best-effort balance between latency and power consumption.

Opportunistic routing has drawn much attention from research communities because of its capability to improve the performance of wireless networks, e.g., [12–17]. It exploits the broadcast nature of the wireless channel and selects multiple potential candidates as next hop to forward packets. Instead of relying on one “good” single path, opportunistic routing utilizes multiple paths to route data from source to destination. Consequently, it effectively improves reliability, reduces delay and power consumption, and highly increases resilience to wireless link dynamics. However, most approaches to opportunistic routing in WSNs are limited to a single channel. As a result, their performance strongly deteriorates in presence of interference.

In this paper, we propose MOR, a Multichannel Opportunistic Routing scheme for low-power duty-cycled multihop WSNs. Incorporated with opportunistic routing, MOR is able to effectively increase the end-to-end reliability, and to reduce the end-to-end latency as well as the power consumption. Moreover, MOR empowers opportunistic routing *more* opportunistically on multiple channels. It

fully takes advantage of frequency diversity to maintain a best-effort resilience to dynamic interference. MOR trades a slight portion of energy for low-power listening (LPL) on multiple channels, while improving the reliability, minimizing the latency and the power consumption.

We implement MOR in Contiki [18] and conduct extensive experiments in the 30-node testbed FlockLab [19]. We compare MOR with selected state-of-the-art single-channel and multichannel protocols. Our evaluation shows that MOR effectively limits the impact of interference: Under interference, MOR provides an end-to-end PDR of more than 98%, while other protocols such as ORPL [16] achieve a PDR of merely 25%. Moreover, MOR's duty cycle settles below 2% for these settings and the latency is less than 2 seconds. In interference-free scenarios, MOR achieves a performance similar to our baseline protocol ORPL, with only an approximately 0.3% increment of duty cycle.

We make the following contributions in this work:

- We propose MOR, *Multichannel Opportunistic Routing*, for duty-cycled multihop WSNs. By opportunistically exploiting temporal, spatial, and frequency diversities, MOR achieves good performance in both, interference-free and interfered conditions.
- As a basis for MOR, we introduce a lightweight channel-hopping strategy for *asynchronous LPL*-based MAC protocols. It guarantees a fast rendezvous between sensor nodes, where the sender and the receiver both opportunistically perform fast channel hopping in each active duty cycle.
- We implement MOR in Contiki, and evaluate the performance of the protocol in terms of end-to-end reliability, latency, and radio duty cycle in the FlockLab testbed.

The remainder of this paper is organized as follows. Section 2 explains the basis of our proposed protocol and provides a brief overview of it. Section 3 details the design of MOR, followed by the performance evaluation elaborated in Section 4. Section 5 discusses related work, with a focus on multichannel MAC and routing protocols in WSNs. Section 6 provides concluding remarks.

2 Background and Overview

In this section, we briefly provide the required background on both, channel hopping and opportunistic routing in WSNs. Next, we introduce the basic concepts of MOR.

2.1 Channel Hopping Strategies in WSNs

Regarding the selection of channels, channel hopping strategies fall into two categories: “whitelisting” and “blind hopping” [6]. In whitelisting, neighboring nodes agree on which channels to use at what point in time for their communication. In blind channel hopping, nodes do not know which channels their neighboring nodes use at what point in time. To establish communication, nodes uniformly hop over all utilized channels, i.e., up to 16 radio channels in IEEE 802.15.4.

Practically, there are two types of channel allocations in multichannel communication, i.e., static channel allocation and dynamic channel allocation [20]. Depending on the sce-

narios, dynamic channel allocation can be more effective if the interference condition is changing dynamically over time. It, however, often performs complex rendezvous algorithms, thus resulting in non-trivial communication computing overhead. To balance the performance and the computing overhead of the sensor node, MOR chooses to use static channel allocation.

The main goal of any channel hopping scheme is to increase the robustness to interference. We observe three approaches of channel hopping strategies in wireless communication: fast channel hopping, slow channel hopping, and hybrid channel hopping [20]. Fast channel hopping switches in each time slot to a new channel. Fast channel hopping is used in a number of applications and standards in order to improve the secrecy and to make the system more robust against jamming or interference. For example, Bluetooth and WirelessHART [21] employ fast channel hopping. Meanwhile, this approach increases the overhead for a packet transmission, i.e., frequent channel switching makes a device consume energy faster than others. Slow channel hopping stays for multiple continuous time slots on a single channel before switching. Compared to fast channel hopping, slow channel hopping generates less latency when two devices need to rendezvous on a common channel. Hybrid channel hopping combines both fast and slow channel hopping, where fast channel hopping improves the robustness to the interference and slow hopping accommodates fast rendezvous.

MOR exploits the hybrid channel hopping scheme. That is, duty-cycled sensor nodes perform fast channel hopping to ensure the robustness to the interference and always-on nodes, i.e., the sink, employ the slow channel hopping scheme to guarantee the fast rendezvous of the last hop.

2.2 Opportunistic Routing in WSNs

ORW [15] is an opportunistic routing scheme for duty-cycled WSNs. Using *anycast*, data packets in ORW are forwarded by the neighboring node which wakes up first, successfully receives the packet, and provides routing progress, as shown in Figure 1(b). Approaches to opportunistic routing in duty-cycled WSNs differ from traditional unicast, where packets are addressed to one specific neighbor, as shown in Figure 1(a). ORW is able to sufficiently reduce delay and energy consumption and improve the resilience to wireless link dynamics. Furthermore, ORPL integrates the concepts of opportunistic routing with RPL [22], the standard protocol for low-power and lossy IPv6-based networks. ORPL provides any-to-any and on-demand traffic. Both ORW and ORPL utilize the expected duty-cycled wakeups (EDC) [15] as routing metric, which allows nodes to select the set of neighboring nodes that provide sufficient routing progress. Experimental results from testbeds show that ORW and ORPL outperform the state-of-the-art solutions including RPL and Collection Tree Protocol (CTP) [23] in terms of latency, power consumption, robustness, and scalability.

2.3 MOR in a Nutshell

MOR extends opportunistic routing with multichannel hopping to combine their key advantages: low latency and high energy efficiency of opportunistic routing with strong robustness to interference of multichannel hopping. Thus,

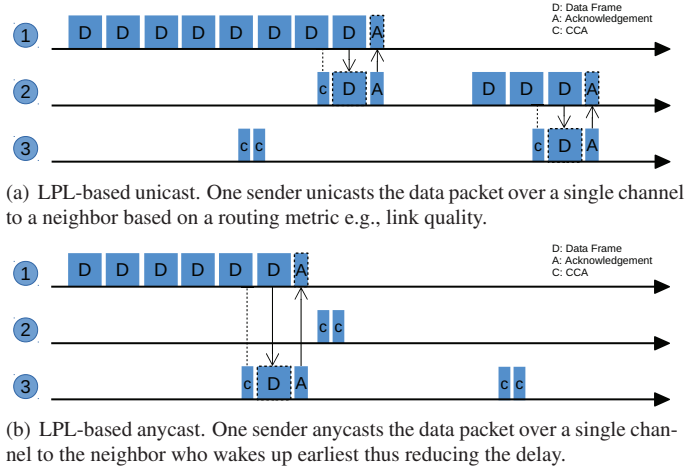


Figure 1: LPL-based unicast and anycast.

MOR inherits the spatial and temporal diversities of opportunistic routing and additionally exploits the frequency diversity of multichannel routing.

MOR builds on ORPL: It employs the EDC routing metric and the integration with RPL. Additionally, unlike a number of synchronous MAC protocols for WSNs, e.g., [5] and [9], MOR is based on asynchronous LPL. It thus does not lead to additional synchronization overhead within the network, and efficiently operates its channel hopping without coordination overhead. Moreover, MOR does not only transmit opportunistically, but also selects channels opportunistically: For each listening and (re)transmission of the underlying MAC layer, MOR utilizes a new channel. For example, while in ORPL it takes multiple transmissions of the MAC on a single channel until one neighboring node wakes up and successfully receives the packet, MOR does each of these (re)transmissions on a different channel.

Overall, MOR extends the concept of opportunistic routing to the frequency domain. That is, in MOR, the first node that wakes up on the rendezvous channel and successfully receives the packet, acts as a forwarder, and thus provides the routing progress. We show in our experimental evaluations that MOR significantly improves robustness in presence of interference when compared to other state-of-the-art protocols. In addition, we show that the duty cycle of MOR is roughly 0.3% more when compared to our baseline ORPL in interference-free scenarios.

3 Design of MOR

In this section, we detail the design of MOR. We discuss the allocation of channels, opportunistic channel rendezvous of senders and receivers, and implementation aspects for integrating multichannel hopping scheme into opportunistic routing.

3.1 Channel Allocation

In MOR, we utilize a subset of the 16 IEEE 802.15.4 channels. To determine this subset of channels, we execute a number of sets of experiments in FlockLab, respectively on 16 individual Zigbee channels. We use the standard protocol

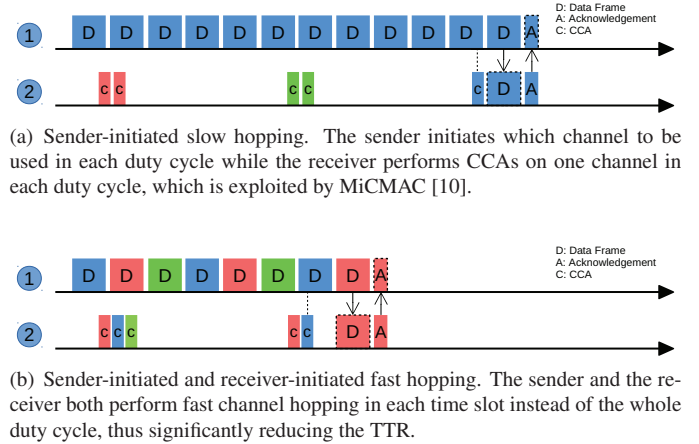


Figure 2: Multichannel hopping schemes: slow hopping and fast hopping.

ContikiMAC/RPL in Contiki. Note that these experiments aim to help evaluating the diversity of each channel, instead of the performance of the protocol. Figure 3 reveals the link qualities of the 16 channels in FlockLab. As shown in the figure, there are only eight “good” channels with higher than 50% end-to-end PDR: channel 26, 25, 20, 15, 21, 22, 19, 14 (sorted in order, with best quality first).

Employing all these eight channel might not be advantageous, while considering the trade-off between the number of channels utilized and the computing overhead: The more channels are utilized, the longer time is required for the nodes to rendezvous, because every receiving node needs to scan the whitelist of the channels. Besides, since low-power listening is required on each channel to exploit frequency diversity, the total energy consumption for channel sensing increases correspondingly as the number of channel increases. Finally, using some “bad” channels does not help to improve the reliability but leads to high latency and energy consumption.

Therefore, MOR chooses to assign three “best” Zigbee channels by default, namely, channel 15, 25, and 26. These channels are orthogonal to WiFi channels in most scenarios: Even with a fully deployed WiFi network (IEEE 802.11 channel 1, 6, and 11), there are still a few channels that are free from the interference (i.e., channel 15, 20, 25, and 26) in [10] and [11]. Furthermore, the effectiveness of the number of channels will be discussed in the following subsection and evaluated by experiments in Section 4.

3.2 Channel Rendezvous

In this section, we discuss the rendezvous scheme of MOR. MOR operates without synchronization: A node does not know when and on which channel a neighboring node wakes up. As a result, a node opportunistically transmits repeatedly and on different channels until its packet has been received. It reduces the time until the sender and the receiver select the same channel at the same point in time. As a result, it minimizes the TTR and provides an upper bound.

Formally, the *rotation closure property* in and asyn-

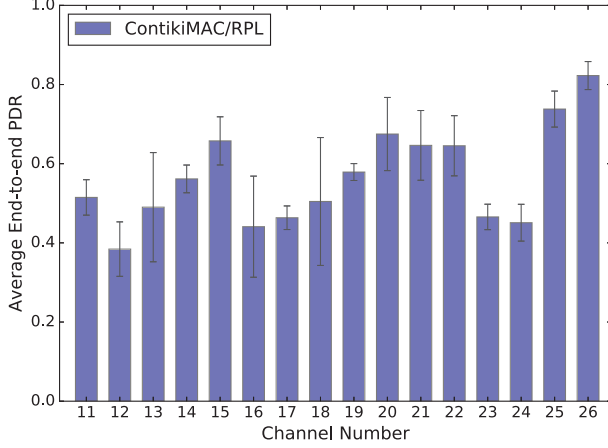


Figure 3: End-to-end reliability of 16 Zigbee channels evaluated with ContikiMAC/RPL in FlockLab.

chronous channel hopping system can be defined as follows by [24]:

$$\forall k, l \in [0, T-1], C(\text{rotate}(\mu, k), \text{rotate}(\nu, l)) \geq m, \quad (1)$$

where positive integer m is the degree of channel overlaps in the system and T represents the number of time periods. $C(\mu, \nu)$ denotes the number of rendezvous channels between two channel hopping sequences μ and ν . Thus, a channel hopping sequence, e.g., μ of T , can be represented as a set of channel indices: $\mu = \{\mu_0, \mu_1, \dots, \mu_{T-1}\}$. Furthermore, $\text{rotate}(\mu, k)$, for example, denotes a cyclic rotation of channel hopping sequence μ by k time slots, i.e.,

$$\text{rotate}(\mu, k) = \{\nu_j \mid \nu_j = \mu_{j+k \bmod T}, j \in [0, T-1]\}, \quad (2)$$

where j, k are non-negative integers.

Generally, if two channel hopping sequences satisfy Equation 1, then these two nodes can rendezvous on at least m distinct channels. For example, given $T = 3$, $\mu = \{1, 2, 3\}$ and $\nu = \{3, 2, 1\}$, there exist $k = 0$ and $l = 0$ satisfying Equation 1, i.e., $C(\text{rotate}(\mu, 0), \text{rotate}(\nu, 0)) \geq 1$. Specifically, it means that by using these two sequences μ and ν , two nodes can rendezvous at least on one channel, i.e., in the second time slot in channel 2 in this example.

If assigning T to 4, $\mu = \{1, 2, 3, 4\}$ and $\nu = \{4, 3, 2, 1\}$, it renders $C(\text{rotate}(\mu, 0), \text{rotate}(\nu, 0)) \geq 1$ being false. Therefore, $T = 3$ guarantees that $C(\mu, \nu) \geq 1$ is always true regardless of the value of non-negative integers k and l . That means, these two sequences rendezvous at least one channel no matter how each individual sequence rotates. In this case, utilizing three hopping channels in MOR is appropriate for maximizing the probability of fast rendezvous. Consequently, an upper bound of the rendezvous time can be provided as five (i.e., $2T - 1$) time slots by these channel hopping sequences. Based on this, we construct our channel rendezvous sequences in *round-robin* fashion, as $\mu = \{15, 25, 26\}$ and $\nu = \{15, 26, 25\}$ for transmitting and receiving, respectively.

3.3 Fast Channel Hopping

MOR employs fast channel hopping for both senders and receivers as shown in Figure 2(b). Thus, for each (re)transmission in the MAC layer, senders in MOR transmit on a different channel, allowing it to quickly iterate over the channels in use. Receivers, upon duty-cycled wakeup, sense the multiple channels. As a result, MOR ensures that senders and receivers rendezvous quickly with this iteration.

Before each transmission, the sender performs two consecutive CCAs to check whether the medium is free. If a node detects a busy channel, it switches to a different channel and repeats the CCA process. Once the channel is free, the transmission sequence consists of three steps: i) sending, ii) waiting for an acknowledgment (ACK), if there is an ACK received, the transmission is completed, otherwise iii) the sender switches to the next channel based on the rendezvous sequence. As soon as it gets an ACK from the receiver, the sender enters to a low-power mode. Alternatively, it keeps the transmission of the packet until a time-out occurs. If not successfully transmitted, this packet will be retransmitted in the next active period after a pre-defined time. In a word, the sender performs fast channel hopping in each individual time slot, and channels are chosen according to the rendezvous sequence. The fast channel hopping strategy of MOR is depicted in Figure 2(b).

Generally, when a receiver wakes up, it first senses the channel activity, and then hops to the next channel if it does not detect anything. If the receiver detects a data packet on a particular channel, it prepares itself with the correct channel ready for the next time slot and then goes into a fast-sleep mode. Furthermore, when a receiving node wakes up in each duty cycle, it performs LPL by a number of consecutive channel sensings, one per channel used by MOR. We set the number of CCAs M to the number of channels N plus 1, i.e., $M = N + 1$. In MOR, the default number of channels is $N = 3$, thus M is set to 4 by default. This increases the probability of early rendezvous and the randomness of the channel selection: Receivers use a different starting index of the hopping sequence to sense the channel every time they wake up. It exploits the frequency diversity more opportunistically.

Overall, MOR trades more energy consumption of channel sensing and switching than slow channel hopping strategies and single-channel approaches. However, this portion of extra energy highly improves the robustness to interference by exploiting the frequency diversity, thus enabling MOR to attain a better network performance. We illustrate this further in Section 4.

3.4 Implementation Aspects

In this section, we discuss the implementation details of MOR.

3.4.1 MAC and Routing Layer

We implement the multichannel extensions of MOR in the MAC layer, i.e., we integrate them with ContikiMAC. While MOR is tailored to opportunistic routing and ORPL in particular, our modifications are transparent to any routing protocol. To ensure a fair comparison, we retain the modifications of the ContikiMAC version of ORPL, e.g., 63 milliseconds (ms) guard time for phase locking. This includes

five retransmission attempts with exponential backoffs of the MAC layer, i.e., on top of the LPL of ContikiMAC.

Moreover, we choose EDC as the routing metric of MOR. Since ORPL uses only one channel for communication, EDC calculation in ORPL is not appropriate for MOR. Thus, we disable the minimal “penalty” function for updating the EDC value in the MAC layer. Because in MOR, if one particular channel is busy, the sending node simply switches to another one, which shall not have a negative effect for the routing metric EDC.

3.4.2 Hit and Hop: Carrier Sense

With the fast multichannel hopping strategies of MOR, a node always switches to another channel after a time slot, i.e., the time period to complete one packet transmission in MOR. It is, as a result, more challenging for a receiver to not only rendezvous on the same channel with the sender, but also detect the data packet. Thus, it requires the protocol to ensure that the receiver can firstly rendezvous on the same channel with the sender, and secondly detect the data packet and successfully receive it.

As shown in Figure 2(b), every time it rendezvous with the sender on a particular channel, the receiver prepares itself for the reception of data on the next-hop channel that is determined by the sender’s round-robin hopping sequence. Basically, the receiver follows the process of “*Hit* → *Hop* → *Sleep* → *Listening*”, whereas i) “*Hit*” stands for the receiver successfully rendezvousing with the sender on a common channel, ii) “*Hop*” means that it then hops to the next channel and enters a fast “*Sleep*” mode, iii) afterwards, before the start of next time slot in the sender, the receiver wakes up again, listens to the channel and receives the packet. Therefore, while waking up after “*Sleep*” in this procedure, the node can always receive the packet on the assigned channel. Throughout this work, we assume that two consecutive channels in the hopping sequence are not always interfered at exactly the same time point. This is assumed, because the channels utilized by MOR are orthogonal to WiFi interference in most cases.

3.4.3 Slow Hopping of the Sink

By default, in both unicast and opportunistic routing protocols including CTP, RPL, ORW, and ORPL, the sink node is not duty-cycled. MOR reflects this always-on nature of the sink in its channel-hopping strategy. Thus, the sink executes the slow hopping strategy. For example, if MOR uses three channels, the sink node hops to another channel every 1/3 of the duration of the duty cycle. As a result, MOR helps the last-hop nodes to attain a rendezvous with the sink within the duration of one duty cycle. In addition, this shortens the strobing time and effectively reduces the power consumption of the last-hop nodes.

3.5 Summary

In summary, MOR effectively extends the concept of opportunistic routing to the frequency domain: In MOR the first node that wakes up on the rendezvous channel and successfully receives the packet, acknowledges and acts as a forwarder. In next section, we show that in our experiments MOR drastically improves robustness in presence of interference when compared to other state-of-the-art protocols. Fur-

ther, while in interference-free scenarios, MOR trades only a small portion of duty cycle to achieve a similar performance as our baseline protocol ORPL.

4 Performance Evaluation

In this section, we perform an extensive experimental evaluation of MOR. We compare MOR to the state-of-the-art, including ContikiMAC/RPL, ORPL, MiCMAC/RPL [10], and Oppcast [11], respectively in scenarios with and without interference.

4.1 Methodology

We use the FlockLab testbed [19] for our experimental evaluation. FlockLab is located at ETH Zurich and consists of 30 TelosB nodes inside and outside of an office building. We assign node 16, a node on the edge of the network, as the network sink to expand the network diameter. In all the experiments, we use the maximum transmission power of the cc2420 radio chip, i.e., 0 dBm. We run a periodic data collection application, where each sensor node transmits a 64-byte payload as UDP datagram over 6LoWPAN to the sink node with an average interval of 2 minutes. The default wakeup frequency of all protocols is 2 Hz. We use JamLab [2] to generate interference in a deterministic and reproducible manner.

For each experimental setting, we perform five independent runs and each run lasts 60 minutes. Experiments with three interfering nodes are executed for 90 minutes. All the results are averaged over these five runs, and the standard deviations are shown by error bars. Following recent trends, such as EWSN Dependability Competition 2016, we include the whole experimental run in our evaluation, including the starting phase of network stabilization. This is also justified by the short stabilization time of ORPL and other opportunistic routing protocols, as we show in our evaluation.

4.1.1 Protocols

We compare MOR to RPL, ORPL, MiCMAC and Oppcast, four state-of-the-art routing protocols, which are all implemented in Contiki.

- ContikiMAC/RPL: RPL is a unicast, tree-based data collection protocol. It uses the expected transmission count (ETX) routing metric (by default) and operates over a single radio channel. We run RPL on ContikiMAC [25], Contiki’s default power-saving MAC. It is duty-cycled and employs LPL with optional phase lock. In our experiments, we choose channel 26 to limit external interference and obtain predictable performance.
- ContikiMAC/ORPL: ORPL is an extension of RPL and employs opportunistic routing over a single channel. As for ORPL, we use channel 26 if not noted else. ORPL utilizes EDC as routing metric.
- MiCMAC/RPL: MiCMAC extends ContikiMAC with a multichannel hopping scheme to exploit the frequency diversity. By default, MiCMAC runs RPL as routing protocol.
- Oppcast: Oppcast is an opportunistic, multichannel data collection protocol based on low-power probing (LPP). It applies a combination of spatial and frequency diversities. It selects three best Zigbee channels out of 16

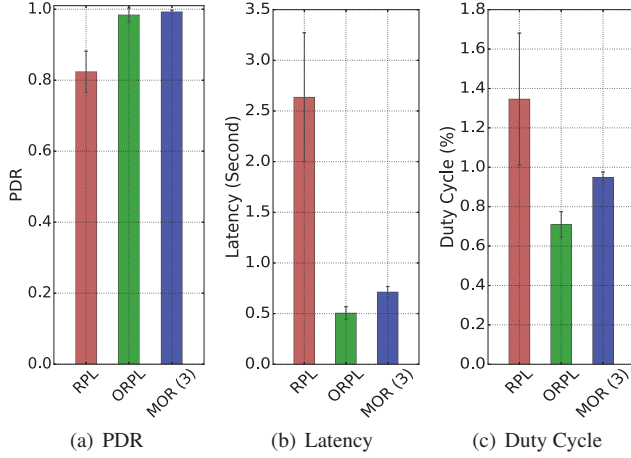


Figure 4: Effectiveness of opportunistic routing. In interference-free scenarios, ORPL performs slightly better than MOR in terms of latency and duty cycle, since it operates over a single channel, leading to less overhead than in MOR. In terms of PDR, MOR achieves similar performance to ORPL. ORPL and MOR outperform RPL on all three metrics.

for channel hopping and it considers the hop count as the routing metric. MOR is different from Oppcast in utilizing different MAC-layer techniques and different routing metrics.

4.1.2 Metrics

We focus on three key metrics to evaluate the performance of the protocols: reliability, latency, and energy efficiency.

- **Reliability (PDR):** The packet delivery ratio (PDR) is the ratio of the number of packets that are successfully delivered to a destination over the number of packets that are sent by the sender in an end-to-end communication. It represents the reliability of the protocol. In most cases, PDR is the basic evaluation metric of a network. To calculate PDR, we log both, the packets sent by each node and the ones received by the sink.
- **Latency:** Latency is the time elapsed from the application on the source node handing the packet to the MAC layer until the packet arrives at the sink's collection application. Thus, latency in this paper represents the end-to-end latency on the application level. Minimizing end-to-end latency in random access networks is one of the key goals of protocol design, especially for mission-critical applications. In this paper, we measure latency based on the timestamps of the serial outputs from the source nodes and the sink node. The same method is used by ORW and ORPL.
- **Energy Efficiency (Radio Duty Cycle):** The duty cycle is the portion of radio-on time over the total time. It is a hardware-independent indicator of power consumption and describes the energy efficiency of the protocol. We measure duty cycle by using the software-based energy profiler [26] of Contiki.

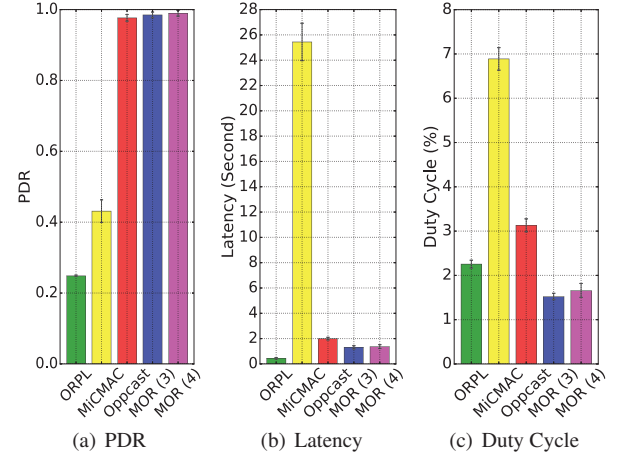


Figure 5: Multichannel routing under interference. MOR is superior to other protocols in the light of PDR, latency, and duty cycle. As a single-channel protocol, interference has the strongest impact on ORPL. MiCMAC and Oppcast improve over ORPL, but MOR outperforms them with its fast hopping strategy.

4.2 Cost of Multichannel Routing

As first step, we compare MOR to single-channel routing protocols, i.e., ContikiMAC/RPL and ContikiMAC/ORPL. Our goal is to measure the overhead of MOR compared to the traditional, single-channel routing protocols in scenarios without interference. In this scenario, RPL and ORPL operate on channel 26 and MOR utilizes three channels: 26, 25, and 15.

We show that the multichannel operation of MOR leads to a reasonable overhead when compared to both RPL and ORPL: Figure 4 presents the results of these three protocols, with respect to PDR, latency, and duty cycle. Taking latency and duty cycle into account, ORPL outperforms both MOR and RPL protocols, with MOR outperforming RPL. MOR inherits key advantages of ORPL such as the high PDR. Namely, MOR achieves an average PDR of 99.26%, slightly better than the one of ORPL (98.41%). On the other hand, utilizing more communication channels, MOR inevitably suffers an approximate 0.21 second longer average end-to-end latency than ORPL. Also, its duty cycle increases from roughly 0.70% to 0.95% when compared to ORPL. These results show that multichannel routing in MOR does not come for free, but with a – as we argue – reasonable overhead. Later, we show that this overhead becomes negligible, once we add interference or switch away from channel 26.

4.3 Benefits of Multichannel Routing

In this section, we evaluate the performance of MOR and related approaches under interference. We compare MOR with our baseline protocol ORPL and two other state-of-the-art multichannel protocols for WSNs, namely MiCMAC/RPL and Oppcast. By default, MiCMAC/RPL utilizes four channels. In contrast, Oppcast utilizes three channels – at least the version provided by the authors to us. Thus,

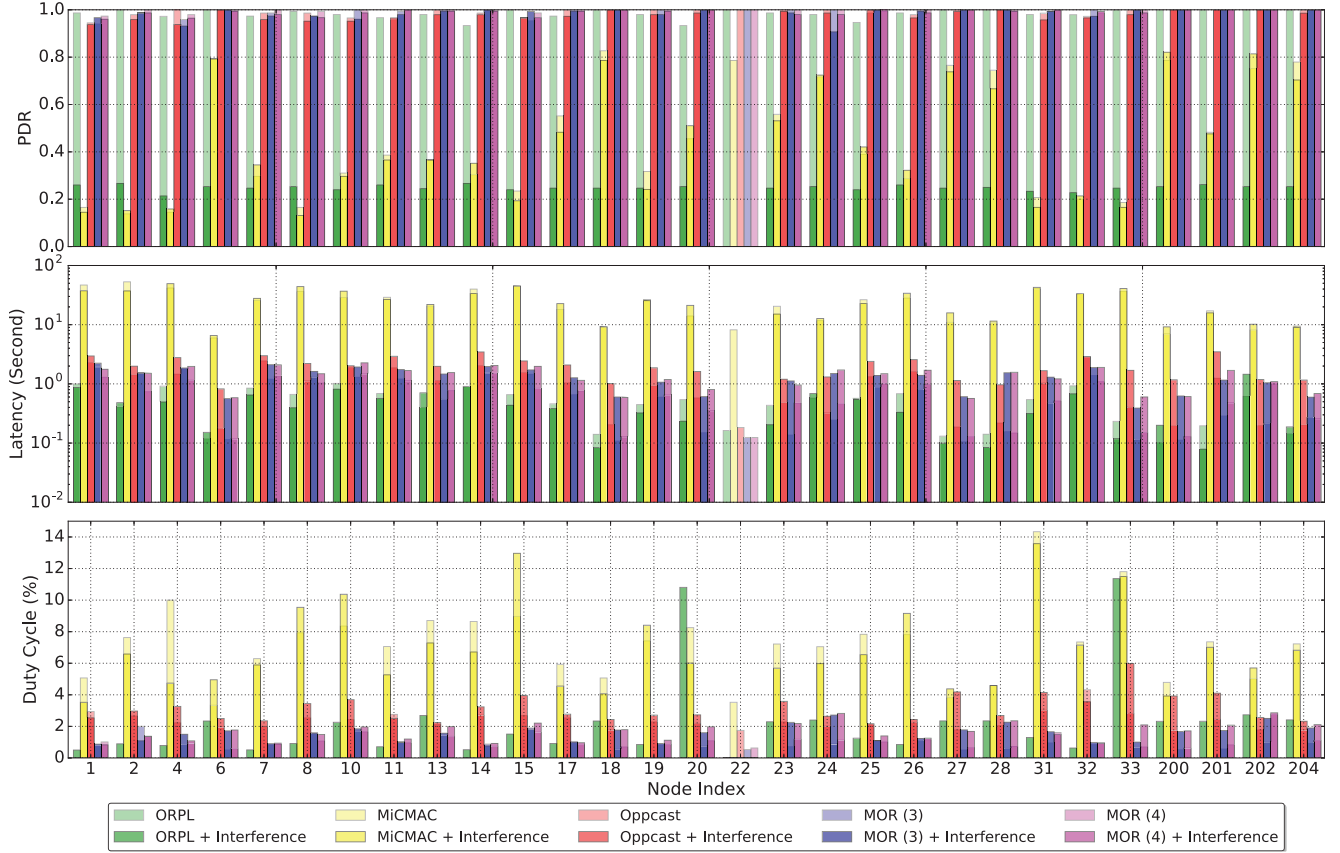


Figure 6: Summary of key metrics with node indices in interference-free and interfered scenarios. The lighter colors represent all the nodes performing in the interference-free scenarios, while the darker colors depict the nodes in the emulated-interfered scenarios. Note that the latency values are shown in log-10 scale.

to ensure a fair comparison, we depict results for two configurations of MOR: with three and four channels, denoted as MOR (3) and MOR (4). We use channels 26, 25, and 15 and channels 26, 25, 20, 15, respectively, in MOR (3) and MOR (4).

We use JamLab [2] to introduce external interference in the testbed. In this setup, a JamLab node acts as a jamming (or an interference) node. For the experiments in this section, we select node 22 in FlockLab as the jamming node, which is close to the sink node 16. We hence expect node 22 to strongly influence the performance of the different protocols. To ensure fairness, we enable the source of interference after the start of each experiment so that each protocol has time to complete the initial setup of its routing tables. We use a power of 0 dBm on channel 26 and keep it on until the test is completed. Our goal, here, is to illustrate the impact that a single interfering node has on the performance of each individual protocol. Later, we extend to dynamic interference scenarios with multiple jamming nodes. Figure 5 shows the key metrics of the above-mentioned protocols under an augmented interference. As a single-channel protocol, ORPL suffers the most: When compared to the scenario without interference, its PDR drops to approximately 25% while its radio duty cycle rises to above 2%.

MiCMAC/RPL shows a PDR of about 43% with very high latency and duty cycle.¹ Oppcast, a recent state-of-the-art multichannel protocol, performs fairly well in terms of PDR and latency. However, in terms of duty cycle, Oppcast performs worse than ORPL and both configurations of MOR. Oppcast has a higher duty cycle: ORPL has roughly 2.2%, Oppcast 3.1%, and MOR 1.5% and 1.7%, for 3 and 4 different channels respectively.

In contrast, MOR outperforms other protocols under interference: Both configurations of MOR are able to obtain a high reliability of over 98.5%. Similarly, MOR achieves a lower latency and the lowest duty cycle: MOR shows a less than 1.4 seconds latency and less than 1.7% duty cycle with both configurations.

Figure 6 summarizes the key metrics of every individual sensor node under two conditions accordingly: with and without emulated interference, respectively in ContikiMAC/ORPL, MiCMAC/RPL, Oppcast, MOR (3) and MOR (4). Please note that we use log-10 scale for the y-axis of latency in Figure 6 as well as in Figure 7. Overall, MOR accomplishes a duty cycle that is roughly half the one

¹These performance results differ from the results reported in the original paper on MiCMAC [10]. We contacted the authors on this and they indicated that the discrepancy is due to a bug in the released code base.

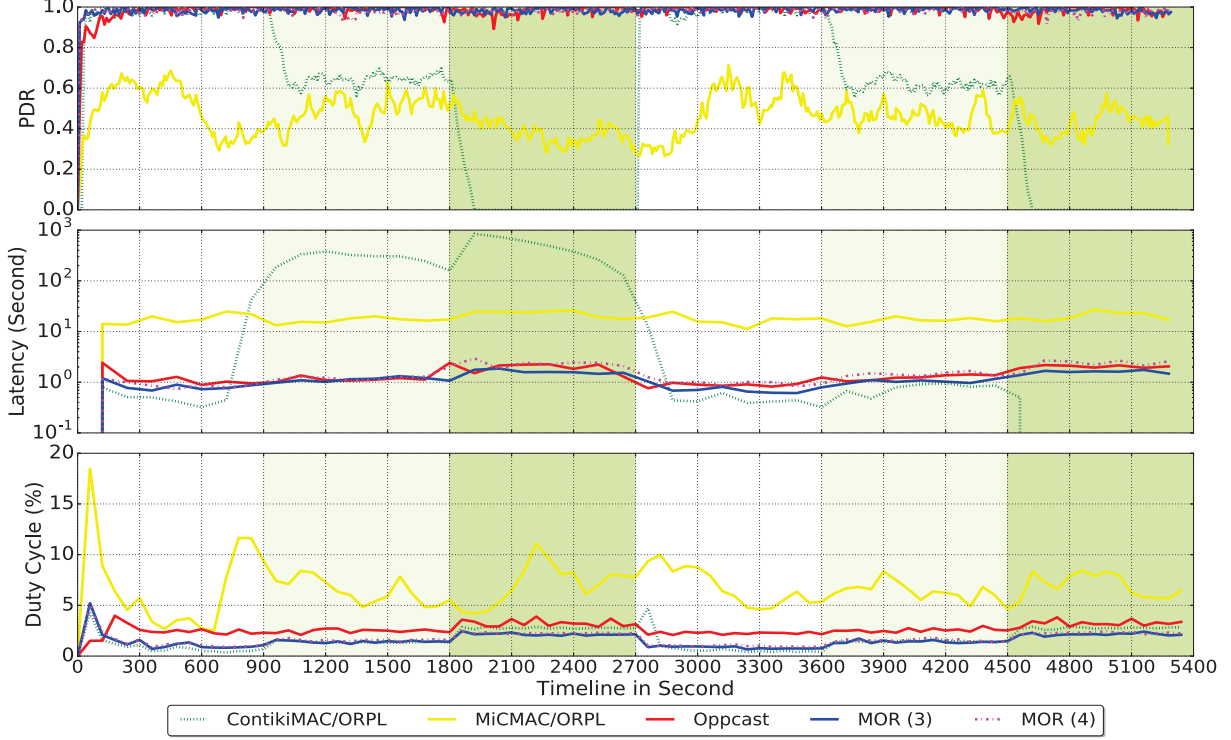


Figure 7: Impact of dynamic interference. The areas filled with white color represent the interference-free conditions, while the areas filled with light green and dark green indicate the interfered conditions, with one jamming node and three jamming nodes corresponding. Note that the latency values are shown in log-10 scale.

of Oppcast, the second best protocol in this setting, while, in addition achieving improved reliability and latency.

4.4 Resilience to Dynamic Interference

In section, we evaluate MOR, ORPL, MiCMAC/RPL, and Oppcast under dynamic interference: Three jamming nodes are dynamically enabled throughout a 90-minute experiment. We select node 15, 19, and 22 of FlockLab as jamming nodes. These nodes are well distributed over the testbed.

To move beyond channel 26, we interfere channel 15 in this experiment and ORPL uses channel 15 as its single channel accordingly. We divide each 90-minute test into two periods of 45 minutes. Each 45-minute period consists of three phases of a 15-minute run: i) no jamming node is enabled, ii) one jamming node (node 15) is enabled, and iii) three jamming nodes are enabled. Our aim is to evaluate the impact of different levels of interference on the performance and the ability of the protocols to recover after interference.

Figure 7 illustrates our metrics over time: Both MOR and Oppcast bear a strong capability to withstand the interference, maintaining higher PDR, lower latency and duty cycle, independent of interference levels. These protocols benefit from frequency diversity, that is, while there is interference on a certain channel, the other channels can be effectively utilized opportunistically. Nonetheless, the higher radio duty cycle of Oppcast becomes apparent: It is constantly roughly twice as high as MOR (3), independent of whether there is interference or not. Furthermore, under severe inter-

fered conditions, MiCMAC/RPL earns a better performance than ContikiMAC/ORPL, with respect to the average end-to-end PDR, and latency, however, with a high duty cycle of approximately 8%.

On the contrary, the performance of the single-channel ContikiMAC/ORPL degrades along with the aggressiveness of the interference, i.e., the more aggressive the interference is, the lower reliability ORPL gains. It is interesting to observe how the performance of ContikiMAC/ORPL recovers once interference ends. Overall, the results underline that MOR obtains a robust performance even under strong adverse conditions, outperforming the state-of-the-art protocols.

4.5 Impact of Low-level Parameters

In this section, we provide a set of low-level benchmarks to further evaluate the key parameters, for example, the number of assigned channels and the wakeup rate in the MAC layer.

4.5.1 Wakeup Interval

At first, we investigate how the wakeup interval of sensor nodes in MOR affects our metrics of reliability, latency, and duty cycle. We preserve the same settings as before, e.g., the number of nodes is 30, and we generate one data packet per node every two minutes. In this experiment, we configure wakeup intervals of 62.5, 125, 250, 500 and 1000 ms, representing channel check rates of 16, 8, 4, 2, and 1 Hz, respectively. Figure 8 depicts the impact of the differ-

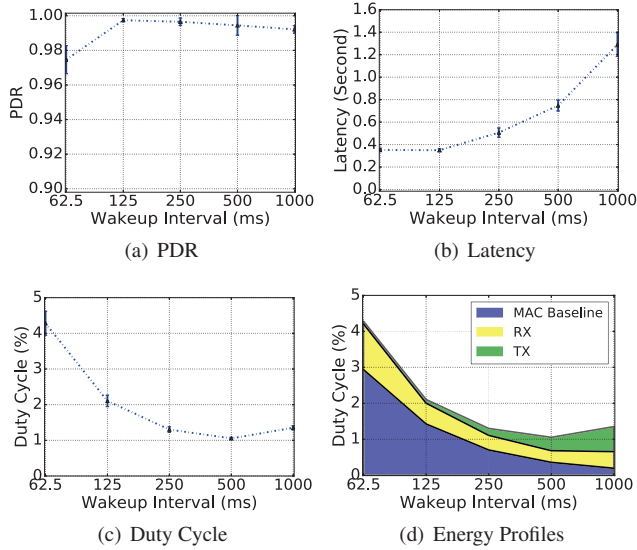


Figure 8: Effectiveness of the channel check rate in MOR. Performance metrics vary along with utilizing different channel check rates.

ent channel check rates in the MAC layer. The results underline that configurations with channel check rates of 2 Hz efficiently balance end-to-end reliability, latency, and power consumption. More specifically, the detailed energy profiles are illustrated in Figure 8(d). Basically, the energy cost of the MAC baseline decreases when increasing the wakeup interval: A larger channel check rate results in more channel listening and thus increases the power consumption. Additionally, when the wakeup interval increases, the energy spent by each transmission is increasing: When the channel check rate decreases, the strobing time of a packet is also increased until a rendezvous with a receiver on the same channel happens.

4.5.2 Number of Channels

Next, we evaluate the impact of the number of channels MOR utilizes. We expect that the power consumption increases when the number of utilized channels increases. Using more channels inherently increases the time until rendezvous and adds channel switching overhead, LPL overhead on each individual channel, and so on. Figure 3 indicates that there are only eight “good” channels in FlockLab, i.e., channels with more than 50% end-to-end PDR: channel 26, 25, 20, 15, 21, 22, 19, and 14 (sorted in order with best quality first). To quantify the impact of the number of channels in detail, we run experiments of MOR in FlockLab using from two to eight channels from these channels. In this experiment, we do not add additional interference next to the interference that is already present in the testbed, e.g., from WiFi, Bluetooth, and so forth.

Figure 9 demonstrates how the performance metrics, i.e., PDR, latency, and duty cycle, change when using more channels in MOR. PDR stays high, while both latency and radio duty cycle increase – as expected – when increasing the number of channels used. The latency here, however, does not

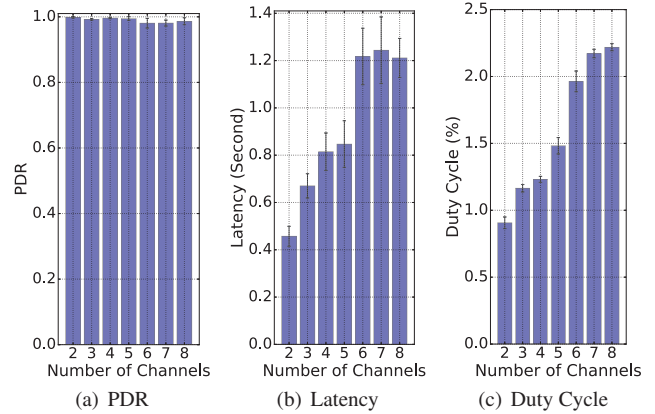


Figure 9: Cost of the number of channels used in MOR. While PDR stays high, latency and duty cycle increase for each channel added.

increases linearly with the number of channels. One possible reason for this can be that, as the number of channels increases, several “not-so-good” channels are also in use, which produces a negative effect on latency. Therefore, we argue that it is sufficient to choose the number of channels to reflect the amount of interference expected. Additionally, we show that even under strong interference, three channels are sufficient to maintain good performance.

4.6 Discussion

Table 1 summarizes our experimental results in three scenarios: i) interference-free, ii) with only one interference source near the sink, and iii) with three dynamic interference sources across the network.

Our experimental results reveal that multichannel routing in MOR comes at a cost: In an interference-free environment MOR sacrifices latency and radio duty cycles when compared to ORPL. However, under interference MOR outperforms other state-of-the-art protocols, including Oppcast and MiCMAC. MOR attains approximately half the duty cycle than Oppcast, the protocol with the second best results and also improves in terms of reliability and latency over the state-of-the-art. Moreover, MOR shows these results independently of the level of interference.

Meanwhile, there are only few limitations in MOR: Practically, the channel rendezvous sequence becomes longer and more complex as the number of channels increases. As a result, the probability of rendezvous cannot always be guaranteed to be 100%. The rendezvous time can also vary strongly. In addition, in scenarios with very aggressive interference, i.e., simultaneously on many channels, MOR can only keep its robust performance when at least one channel is available at each point in time.

To sum up, in this section, we demonstrate the performance of different protocols respectively in three scenarios: interference-free, interfered, and dynamic interfered scenarios. Our experimental results reveal that in interference-free scenario, MOR effectively inherits the benefits from opportunistic routing. It achieves the best performance in inter-

Table 1: Summary of experimental results. MOR maintains a best-effort end-to-end PDR regardless of interference. Under adverse interfered conditions, MOR improves the end-to-end latency while preserving higher power efficiency. Oppcast is also a robust protocol with respect to PDR, latency, and resilience to interference, but it consumes more energy.

Network Settings		Protocol	PDR (%)	Latency (s)	Duty Cycle (%)
Application: Many-to-one data collection Experiment duration: 60 minutes Tx power: 0 dBm Channel check rate: 2 Hz Average packet generation interval: 2 minutes		ContikiMAC/RPL	82.38 (max: 91.56 min: 76.69)	2.64 (max: 3.17 min: 1.94)	1.35 (max: 1.67 min: 0.87)
		ContikiMAC/ORPL	98.41 (max: 99.88 min: 95.05)	0.50 (max: 0.63 min: 0.32)	0.65 (max: 0.77 min: 0.55)
		MOR (3)	99.26 (max: 99.77 min: 98.62)	0.71 (max: 0.79 min: 0.65)	0.95 (max: 0.99 min: 0.92)
Application: Many-to-one data collection Experiment duration: 60 minutes Tx power: 0 dBm Channel check rate: 2 Hz Average packet generation interval: 2 minutes Jamming channel: Channel 26 Jamming node (node 22): 15 minutes OFF + 45 minutes ON		ContikiMAC/ORPL	24.87 (max: 25.12 min: 24.60)	0.45 (max: 0.53 min: 0.38)	2.25 (max: 2.32 min: 2.10)
		MiCMAC/RPL	43.13 (max: 47.45 min: 39.65)	25.45 (max: 26.93 min: 23.88)	6.89 (max: 7.21 min: 6.68)
		Oppcast	97.68 (max: 99.13 min: 96.79)	2.00 (max: 2.10 min: 1.84)	3.13 (max: 3.33 min: 2.93)
		MOR (3)	98.51 (max: 99.52 min: 97.48)	1.31 (max: 1.51 min: 1.18)	1.52 (max: 1.62 min: 1.45)
		MOR (4)	98.97 (max: 99.64 min: 97.59)	1.37 (max: 1.57 min: 1.24)	1.66 (max: 1.87 min: 1.52)
Application: Many-to-one data collection Experiment duration: 90 minutes Tx power: 0 dBm Channel check rate: 2 Hz Average packet generation interval: 2 minutes Jamming channel: Channel 15 Jamming node (node 15): (15 minutes OFF + 30 minutes ON) x 2 Jamming node (node 19): (30 minutes OFF + 15 minutes ON) x 2 Jamming node (node 22): (30 minutes OFF + 15 minutes ON) x 2		ContikiMAC/ORPL	64.5 (max: 66.20 min: 62.34)	140.13 (max: 163.20 min: 104.40)	1.69 (max: 1.77 min: 1.60)
		MiCMAC/RPL	45.72 (max: 50.00 min: 41.52)	25.87 (max: 29.29 min: 21.23)	6.76 (max: 7.10 min: 6.22)
		Oppcast	99.21 (max: 99.56 min: 98.60)	1.43 (max: 1.49 min: 1.39)	2.70 (max: 2.74 min: 2.66)
		MOR (3)	99.35 (max: 99.66 min: 98.98)	1.17 (max: 1.46 min: 1.07)	1.56 (max: 1.77 min: 1.49)
		MOR (4)	99.33 (max: 99.66 min: 98.88)	1.56 (max: 1.73 min: 1.39)	1.68 (max: 1.79 min: 1.63)

ferred scenarios in terms of PDR, latency, and duty cycle, compared to other state-of-the-art protocols. Besides, MOR is able to maintain the robust performance even in adverse conditions of dynamic interfered scenario.

5 Related Work

Multichannel communication is essential to provide reliable communication under interference and is part of many standards such as Bluetooth and WirelessHART [21]. In the domain of WSNs, multichannel communication helps to, for example, improve reliability, resilience to interference, throughput, and reduce latency [4–11]. These approaches take advantage of location-specific knowledge of the wireless channel: its diversities in frequency, time, and space. As a result, these ensure reliable, and robust co-existent wireless communication.

In the following, we group approaches to multichannel routing into two classes, according to the MAC layer they base on: multichannel routing for i) synchronous, and ii) asynchronous protocols. In synchronous MAC protocols, sensor nodes maintain a tight time synchronization and the wakeups of each node are commonly scheduled to when neighboring nodes wake up. Asynchronous MAC protocols, on the other hand, establish communication between two nodes that are on different active/sleep schedules.

5.1 Synchronous MAC Protocols

Y-MAC [5] is an energy-efficient multichannel MAC protocol for WSNs. It is a TDMA-based MAC protocol, thus requiring accurate time synchronization. In Y-MAC, sensor nodes exchange the remaining time in the current time slot to synchronize their starting points for the next slot. A light-weight channel hopping mechanism is implemented in Y-MAC that enables multiple nodes to communicate simultaneously on multiple channels. This mechanism increases network throughput and reduces latency. Experimental re-

sults demonstrate that Y-MAC is able to achieve a low duty cycle under light traffic conditions and ensures an energy-efficient transmission of bursty messages under high traffic conditions.

MC-LMAC [9] is a multichannel MAC protocol, designed to maximize the throughput of WSNs by coordinating transmissions over multiple channels. In MC-LMAC, time is slotted and each node is assigned the control over a time slot to transmit on a particular channel. Hence, MC-LMAC takes advantage of both, scheduled and multichannel communication, which can minimize communication collisions, overcome the increased contention and interference on the limited bandwidth, and improve the throughput and channel utilization. Simulation results show that MC-LMAC obtains significant bandwidth utilization and high throughput while ensuring an energy-efficient operation.

Orchestra [27], autonomously provides Time Slotted Channel Hopping (TSCH) [28] in RPL networks. In Orchestra, nodes autonomously compute their own local schedules and maintain the schedules allocated to a particular traffic plane, i.e., application, routing, and MAC. Orchestra (re)computes local schedules without signaling overhead. Instead, it only relies on the existing network stack information to maintain the schedules. Extensive evaluations in simulation and in two different testbeds demonstrate the practicality of Orchestra and its ability to consistently achieve a very high delivery ratio in the order of 99.99%, while obtaining a balance between latency and energy consumption.

5.2 Asynchronous MAC Protocols

MuChMAC [7] is a low-overhead multichannel MAC protocol, which combines TDMA with asynchronous MAC techniques, and requires no coordination or tight synchronization between nodes. MuChMAC is a receiver-initiated multichannel MAC protocol. In every time slot, each node

switches its radio channel according to a pre-defined channel assignment, which is based on the parallel rendezvous principle [29]. The channel is calculated based on a node's ID and the current slot number following a pseudo-random hopping sequence. Experiments in a testbed demonstrate the applicability of MuChMAC and show that it can efficiently operate multichannel communication without coordination or synchronization overhead.

Chryso [30] is a multichannel protocol for data collection. In Chryso, sensor nodes are organized in parent-children groups, where each parent-children group uses two channels: one for packet transmissions and one receptions. When a node in Chryso detects interference on one channel, both parent nodes and child nodes switch to another channel based on a channel hopping policy. The authors of Chryso show its reliability under severe WiFi interference and jamming.

Efficient Multichannel MAC (EM-MAC) [8] introduces mechanisms for adaptive receiver-initiated multichannel rendezvous and predictive wakeup scheduling. To achieve high energy efficiency, EM-MAC enables a sender to predict both the receiver's transmission channel and wakeup time. In EM-MAC, a node is able to select channels dynamically based on the channel conditions it senses. In this matter, it avoids utilizing channels that are heavily loaded or are undesirable because of interference or jamming. In their evaluation, the authors show that it can achieve a low duty cycle, low latency, and high PDR under interference.

MiCMAC [10] is a multichannel extension of Contiki-MAC based on LPL. MiCMAC performs a sender-initiated channel hopping. Namely, in every wakeup period, the channel is determined by the sender according to a pseudo-random sequence. Similar to the phase-lock mechanism in ContikiMAC, a channel-lock mechanism is integrated in MiCMAC to shorten the rendezvous time between the sender and the receiver on various communication channels. Experiments show that MiCMAC improves the performance of the network in terms of reliability, latency, duty cycle, and resilience to external interference.

Oppcast [11] is a multichannel LPP-based data collection protocol. It opportunistically utilizes both, broadcast and unicast transmissions to maintain good network performance in presence of interference. Oppcast selects and uses three good channels, i.e., channel 15, 25 and 26, out of all the 16 Zigbee channels. In Oppcast, both receivers and senders simultaneously perform channel hopping with a round-robin principle. Based on opportunistic routing, Oppcast takes advantage of the spatial diversity and it utilizes the hop count as a routing metric to optimize performance. Experiments in a large-scale testbed show that Oppcast maintains consistently high reliability, low latency, and low duty cycle in several urban scenarios.

5.3 Summary

Multichannel routing is essential for reliable communication under interference and it has received significant attention in the recent years. Nonetheless, most approaches focus on traditional unicast routing. We argue in this paper that opportunistic routing, such as ORPL and ORW, opens new design options for reliable, multichannel communica-

tion. Thus, in MOR, we extend the concept of opportunistic routing to the frequency domain: The first node that i) wakes up on the rendezvous channel, ii) successfully receives the packet, and iii) provides routing progress, acknowledges and acts as a forwarder. We show in our experimental evaluation that MOR significantly improves robustness in presence of interference when compared to other state-of-the-art protocols.

6 Conclusion

This paper introduces MOR, a multichannel opportunistic routing protocol for low-power duty-cycled WSNs. MOR applies multichannel hopping strategies in opportunistic routing, thus exploiting spatial, temporal, and frequency diversities in WSNs. The opportunistic nature of the packet forwarding in MOR is essential for its performance: In contrast to traditional approaches to unicast routing, e.g., RPL or CTP, MOR does not have to ensure rendezvous with one particular parent. MOR only needs a rendezvous with one of the typically many potential forwarders. Thereby, MOR benefits from both spatial and frequency diversities: If one neighbor is not available on a particular channel, then it either utilizes a different forwarder or a different channel.

We implement our protocol in Contiki and evaluate it with extensive experiments in FlockLab. With trading only a slight portion of power consumption, MOR achieves higher than 98.50% average end-to-end reliability, and less than 1.60 seconds average end-to-end latency, in both, interference-free and severely interfered environments. Furthermore, MOR maintains a more robust resilience to a highly dynamic interference with less duty cycle, while compared to other protocols. To sum up, MOR outperforms the state-of-the-art protocols in the light of end-to-end reliability, latency, and power consumption.

Acknowledgments

We would like to thank Simon Duquennoy, Beshr Al Nahas, and Mobashir Mohammad, for their big help and technical support in carrying on this work. Besides, we thank the Computer Engineering Group at ETH Zurich for providing the FlockLab testbed. We also thank the anonymous reviewers for their valuable comments. This research was supported by the German Research Foundation through the Research Training Group DFG-GRK 1765: System Correctness under Adverse Conditions (SCARE).

7 References

- [1] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the RF smog: Making 802.11n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 170–181, New York, NY, USA, 2011. ACM.
- [2] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Ziga. Jamlab: Augmenting sensor network testbeds with realistic and controlled interference generation. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 175–186, April 2011.
- [3] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.
- [4] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in

- wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 1–14, New York, NY, USA, 2008. ACM.
- [5] Y. Kim, H. Shin, and H. Cha. Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 53–63, Washington, DC, USA, 2008. IEEE Computer Society.
 - [6] T. Watteyne, A. Mehta, and K. Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '09, pages 116–123, New York, NY, USA, 2009. ACM.
 - [7] J. Borms, K. Steenhaut, and B. Lemmens. Low-overhead dynamic multi-channel MAC for wireless sensor networks. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pages 81–96, Berlin, Heidelberg, 2010. Springer-Verlag.
 - [8] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 23:1–23:11, New York, NY, USA, 2011. ACM.
 - [9] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Networks*, 9(1):73–94, 2011.
 - [10] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multi-channel. In *Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '14, pages 2–9, Washington, DC, USA, 2014. IEEE Computer Society.
 - [11] M. Mohammad, X. Guo, and M. C. Chan. Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments. In *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12, April 2016.
 - [12] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(1):69–74, January 2004.
 - [13] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 8(12):1622–1635, Dec 2009.
 - [14] X. Mao, S. Tang, X. Xu, X. Y. Li, and H. Ma. Energy-efficient opportunistic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(11):1934–1942, Nov 2011.
 - [15] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson. Low power, low delay: Opportunistic routing meets duty cycling. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 185–196, New York, NY, USA, 2012. ACM.
 - [16] S. Duquennoy, O. Landsiedel, and T. Voigt. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, New York, NY, USA, 2013. ACM.
 - [17] S. Guo, L. He, Y. Gu, B. Jiang, and T. He. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. *IEEE Transactions on Computers*, 63(11):2787–2802, Nov 2014.
 - [18] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462, Nov 2004.
 - [19] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 153–166, New York, NY, USA, 2013. ACM.
 - [20] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi. The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 15(1):101–120, First 2013.
 - [21] HART Communication Foundation. HART field communication protocol specification. *Revision HFC_SPEC12*, 6, 2006.
 - [22] T. Winter. RPL: IPv6 routing protocol for low-power and lossy networks. 2012.
 - [23] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.
 - [24] K. Bian and J. M. Park. Asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *Proceedings of the 2011 IEEE INFOCOM*, pages 236–240, April 2011.
 - [25] A. Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report 2011:13, Swedish Information and Communications Technology (ICT), 2011.
 - [26] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. Software-based online energy estimation for sensor nodes. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, pages 28–32, New York, NY, USA, 2007. ACM.
 - [27] S. Duquennoy, B. A. Nahas, O. Landsiedel, and T. Watteyne. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 337–350, New York, NY, USA, 2015. ACM.
 - [28] T. Watteyne, M. Palattella, and L. Grieco. Using IEEE 802.15. 4e time-slotted channel hopping (TSCH) in the internet of things (IoT): Problem statement. RFC 7554, IETF, 2015.
 - [29] H. S. W. So, J. Walrand, and J. Mo. McMAC: A parallel rendezvous multi-channel MAC protocol. In *Proceedings of 2007 IEEE Wireless Communications and Networking Conference*, pages 334–339, March 2007.
 - [30] V. Iyer, M. Woehrle, and K. Langendoen. Chryso—a multi-channel approach to mitigate external interference. In *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 449–457. IEEE, 2011.