Interference-Resilient Ultra-Low Power Aperiodic Data Collection

Timofei Istomin  
University of Trento, Italy  
timofei.istomin@unitn.it

Matteo Trobinger  
University of Trento, Italy  
matteo.trobinger@unitn.it

Amy L. Murphy  
Bruno Kessler Foundation, Italy  
murphy@fbk.eu

Gian Pietro Picco  
University of Trento, Italy  
gianpietro.picco@unitn.it

ABSTRACT

Aperiodic data collection received little attention in wireless sensor networks, compared to its periodic counterpart.

The recent CRYSTAL system uses synchronous transmissions to support aperiodic traffic with near-perfect reliability, low latency, and ultra-low power consumption. However, its performance is known under mild interference—a concern, as CRYSTAL relies heavily on the (noise-sensitive) capture effect and targets aperiodic traffic where “every packet counts”.

We exploit a 49-node indoor testbed where, in contrast to existing evaluations using only naturally present interference to evaluate synchronous systems, we rely on JamLab to generate noise patterns that are not only more disruptive and extensive, but also reproducible. We show that a properly configured, unmodified CRYSTAL yields perfect reliability (unlike Glossy) in several noise scenarios, but cannot sustain extreme ones (e.g., an emulated microwave oven near the sink) that instead are handled by routing-based approaches. We extend CRYSTAL with techniques known to mitigate interference—channel hopping and noise detection—and demonstrate that these allow CRYSTAL to achieve performance akin to the original even under multiple sources of strong interference.

CCS CONCEPTS

• Networks → Network protocol design;

KEYWORDS

Synchronous transmissions, wireless sensor networks, energy efficiency, interference resilience, channel hopping.

1 INTRODUCTION

Aperiodic data collection received little attention in wireless sensor networks, compared to its periodic counterpart. A notable exception is our CRYSTAL system, recently proposed in [15]. Originally designed to exploit the synergy with data prediction, CRYSTAL uses synchronous transmissions [8] to support aperiodic and sparse traffic with near-perfect reliability, low latency, and ultra-low power consumption.

Is CRYSTAL resilient to strong interference? However, the remarkable performance of CRYSTAL was reported only under mild interference; we ran experiments on channel 20 and 26 which "showed very similar performance [...] during the night runs; however, the daytime results were inconsistent and difficult to assess" and therefore "the results only from night runs on channel 26" were included [15].

Statements like these are not uncommon in the related literature, as discussed later. However, this is of particular concern here because interference i) potentially undermines CRYSTAL at the core by hampering the capture effect it heavily relies on, and ii) increases the overhead of achieving near-perfect reliability of aperiodic and sparse traffic in which “every packet counts”, possibly precluding ultra-low power consumption.

Hence, whether the remarkable performance in [15] holds under strong interference is an open question, answered in this paper by analyzing the performance of CRYSTAL under several, increasingly disruptive noise patterns and introducing techniques to boost its resilience to strong interference without sacrificing ultra-low power consumption.

Natural vs. Generated Interference. We report experiments in a 49-node indoor testbed and exploit its natural interference, mostly WiFi, in line with the evaluation of well-known synchronous transmissions systems [8, 9, 16, 27].

Actually, reliance on natural interference is the only methodology hitherto adopted for evaluating them. Despite Glossy and derivatives being commonly considered highly resilient to interference, the extent to which this holds has never been ascertained under noise patterns that are i) repeatable, and ii) more extensive and disruptive than natural ones.

We raise the standard of evaluating synchronous transmissions under interference by reporting, for the first time, results based on the reproducible generation of realistic noise patterns. We use JamLab [1], described in §2, to emulate WiFi devices and microwave ovens in our experimental setup (§3).
Performance Metrics and Comparison Baselines. We evaluate CRYSTAL using packet delivery rate (PDR) and duty cycle (DC) as metrics for reliability and energy consumption, respectively. Moreover, as CRYSTAL relies on unmodified Glossy, we indirectly evaluate it with the same experiments under interference; as mentioned above, we argue this is a contribution per se.

We observe that none of the proposals tackling interference found its way into the mainstream. Hence, we choose the readily-available RPL [25] and ORPL [6] as baselines (§4), in line with analogous works [18, 20, 28].

Results and Contributions. We show (§5) that all protocols sustain natural interference, but only Glossy and CRYSTAL achieve near-perfect PDR, with a much lower DC. Under JamLab-emulated WiFi, RPL reliability degrades even with a single jammer; with several covering the entire testbed, ORPL also degrades, while CRYSTAL still achieves near-perfect PDR. Interestingly, roles are reversed when an emulated microwave oven is placed 1m from the sink; ORPL achieves near-perfect PDR, while CRYSTAL falls below 80%.

These results pushed us to explore two techniques to improve the resilience of CRYSTAL (§6). The first allows nodes to escape interference by executing each transmission-acknowledgement pair—a core CRYSTAL constituent—on different channels, based on a network-wide hopping sequence. This approach, which uses Glossy unmodified, is notably different from protocols in the literature that apply channel hopping inside Glossy [7, 17, 21]. Second, noise detection at all nodes enables them to schedule extra transmissions in a decentralized way, increasing packet delivery. This fights interference, effectively providing a “safety net” when channel hopping alone is insufficient, but may keep nodes unnecessarily active, which detrimental in the sparse traffic targeted by CRYSTAL.

Our experimental results (§7) show that the combination of these two techniques, to the best of our knowledge novel in the context of synchronous transmissions, achieves near-perfect reliability in the very challenging scenarios where both microwave ovens and WiFi are simultaneously present. Overall, we confirm that the original CRYSTAL (and the underlying Glossy) can tolerate the moderate levels of interference commonly found in office environments. However, CRYSTAL can also be modified with relative ease to sustain much stronger interference patterns while retaining its ultra-low power consumption.

Finally, we concisely survey related work (§8), before ending the paper with brief concluding remarks (§9).

2 BACKGROUND

We offer the necessary background on synchronous transmissions and CRYSTAL, along with the JamLab infrastructure used to generate reproducible interference patterns.

2.1 Synchronous Transmissions: CRYSTAL

Synchronous transmission protocols, pioneered by Glossy [8], build on two properties of the IEEE 802.15.4 PHY: constructive interference and capture effect. These occur when packet transmissions by neighboring nodes are initiated within a tiny temporal interval (0.5μs and 160μs, respectively) and yield a successful reception instead of a collision. Constructive interference works when the packet is the same, yielding high reliability due to the combination of the identical signals; the capture effect, instead, works with different packets, one of which is received with a probability depending on the density of neighbors and their signal strength.

Glossy exploits these properties to construct network-wide floods that are extremely i) fast, as each node receiving a packet immediately retransmits it, preserving the required tight timing ii) reliable, due to the above PHY-level properties, and the inherent spatial and temporal redundancy of flooding. To increase reliability, packets are retransmitted by each node N times; the value of N is the main knob to control the tradeoff between reliability and energy consumption.

Aperiodic, sparse data collection. CRYSTAL [15] builds a schedule-ule atop Glossy that, unlike works geared towards periodic data collection [9, 24], is designed to efficiently support aperiodic, sparse traffic like the one stemming from applying data prediction [13, 22] to regular, periodic traffic. Prediction quenches the majority of application messages, inducing sporadic traffic interleaved with long, quiescent intervals. However, a sudden change in the monitored phenomena may invalidate the prediction model, which must be regenerated and sent to the sink, possibly by multiple nodes at once. Table 1, adapted from [15], shows an example traffic profile resulting from applying data prediction to the well-known 36-day Intel dataset [14] containing temperature samples gathered with a period of 30s, hereafter called epoch. After data prediction is applied, the majority (82.1%) of the total 102686 epochs is empty, as the sink can predict the next value based on the last model reported by each node. However, in a non-negligible fraction of epochs, U ≥ 1 concurrent senders must send model updates. Further, as packets carry models rather than raw data, the loss of a single one has a much larger impact on the reliability of the overall system.

CRYSTAL in a nutshell. To reconcile these requirements, CRYSTAL builds a network-wide transport protocol, in which i) a transmission (T) slot is used by U concurrent senders to disseminate their packet; these floods “compete” until, thanks to the capture effect and Glossy redundancy, one reaches the sink with high probability ii) a subsequent acknowledgment (A) slot is used by the sink to flood the identifier of the sender whose packet it received, informing the others whether re-transmission is needed because their packet was “overcome” by another or no packet was received at the sink.

Table 1: An aperiodic, sparse traffic profile; number and fraction of epochs with U concurrent senders.

<table>
<thead>
<tr>
<th>U</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>84.3K</td>
<td>15.5K</td>
<td>2.2K</td>
<td>606</td>
<td>46</td>
<td>1</td>
</tr>
<tr>
<td>%</td>
<td>82.1</td>
<td>15.1</td>
<td>2.2</td>
<td>0.14</td>
<td>0.038</td>
<td>0.005</td>
</tr>
</tbody>
</table>
values improve reliability but with higher energy consumption. Other parameters are described in [15], e.g., the duration \( G \) of guards and the number \( Z \) of consecutive missed acknowledgements.

**Baseline configuration.** In essence, Crystal builds a reliability layer atop Glossy, which strikes different tradeoffs w.r.t. energy consumption by exploiting the interplay between the two layers. As in Glossy, the number \( N \) of retransmissions in each flood is key, but in Crystal this can be set independently \((N_S, N_T, N_A)\) for each phase; the same holds for the maximum slot duration \( W \), another key Glossy parameter.

The configuration used in the paper (Table 2) is adapted from the original. First, our testbed has a larger diameter than Indriya, used in [15]. This forced us to use larger values for the intervals \( W_T \) and \( W_A \) to allow Glossy floods to complete; we determined the optimal value using the methodology of [15]. Second, we experiment with combinations of \( N_T \) and \( N_A \) values to explore the impact of the T phase w.r.t. interference. The values of the remaining parameters \( W_S, G, R, Z \) are unchanged.

Finally, we use two power settings, high (0dBm) and low (−7dBm); the former is the default throughout the paper.

### 2.2 Generating Interference: JamLab

As we argued in §1, the ability to reproduce interference patterns is key to our study. Therefore, we rely on JamLab [1], which achieves this goal using the same mote-class nodes available in a testbed, and whose software faithfully emulates various types of interference relevant to IEEE 802.15.4, including Bluetooth, WiFi, and microwave ovens. These have very different characteristics. Bluetooth interferes with all IEEE 802.15.4 channels, as it uses a channel hopping scheme. WiFi spans 4 IEEE 802.15.4 channels with interference that is significantly stronger than Bluetooth, but also based on the type of data traffic. Microwave ovens, depending on model and load, may interfere with several consecutive channels, if not all, and induce very strong, continuous interference for 5-10ms, alternated with inactive periods of 10-15ms [1, 12]. According to [1], channels 20–26 are affected the most.

Hereafter, to put ourselves in the worst-case scenario, we focus only on WiFi and microwave ovens, as they yield the strongest interference. Similarly, we select the most challenging of the WiFi patterns offered by JamLab (JL_WIFI4) and configure the jammers to transmit modulated carrier at the maximum power (0dBm).

One criticism of JamLab is that the interference sources it can mimic are limited, and real environments may contain different ones. While this is true, the aforementioned characteristics of WiFi and microwave are different enough to cover a broad spectrum of noise patterns; further, by combining them, we create an even more challenging interference scenario for our experiments.

Another JamLab limitation is that real interference sources often interfere with many contiguous IEEE 802.15.4 channels at the same time; in contrast, a JamLab node generates noise on a single channel. The majority of the proposed protocols, including the synchronous transmissions ones described in §2.1 and the mainstream ones in §4, operate on a single channel: therefore this limitation does not affect the experiments in §5. However, in §7 we explore channel hopping and address this JamLab limitation with a channel mapping strategy.

Finally, although we use the maximum TX power (0dBm) of motes, this is much smaller than real interference sources (e.g., 25 and 60dBm for WiFi and microwave ovens, respectively). As suggested in [1], we use therefore multiple motes, strategically placed in our testbed (Figure 2).

### 3 TESTBED INTERFERENCE SCENARIOS

The experiments we report were performed in our local testbed, composed of 49 Tmote Sky nodes deployed (Figure 2) in a 60 x 40 m² office area, subject to WiFi interference. Similar to other reports [18], the latter \( i \) is more intense during the day and less at night and during the weekends, and \( ii \) varies depending on the channel considered. In addition to this natural interference, we leverage controlled JamLab generated interference, enabling repeatable experiments. Overall, we define four types of interference (Table 3). The choice of channels for natural interference derives from an extensive, cross-channel measurement campaign, which identified the best (26) and worst (18) channels during night and day, respectively. The generated interference is created at night on channel 26 (i.e., under

![Figure 1: Crystal in a nutshell.](image1)

![Figure 2: Position of the jammers in the testbed.](image2)
Table 3: Types of interference.

<table>
<thead>
<tr>
<th>Type of interference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>T-LOW testbed at night/weekends, channel 26</td>
</tr>
<tr>
<td></td>
<td>T-HIGH testbed during the day, channel 18</td>
</tr>
<tr>
<td>Generated</td>
<td>J-WIFI JamLab WiFi interference (JL_WIFI)</td>
</tr>
<tr>
<td></td>
<td>J-MWO JamLab microwave oven interference</td>
</tr>
</tbody>
</table>

natural T-LOW interference). Our evaluation uses varying numbers of JamLab jammers for each type and combines different types in the same experiments, to obtain challenging, realistic setups. Node 1 is the sink in all experiments.

Figure 3 quantitatively compares the various types of interference while Figure 4 shows its effect on the number of links, their qualities and the network radius (Glossy hopcount). The natural T-LOW (Figure 3a) exhibits an average noise of ~93dBm, rather stable and uniform across the network. The interference in natural T-HIGH is drastically different (Figure 3b). The average noise is ~86dBm, but several nodes are exposed to much higher noise, reaching ~50dBm. This affects the network topology by reducing the number of perfect links by 1/3, yielding a 10% increase in the average hopcount (Figure 4).

The interference generated via JamLab yields stronger noise than the natural one. Figure 3c shows the J-WIFI interference generated by node 7 alone, the closest (1m) to the sink. Figure 3d shows instead the effect of 6 J-WIFI jammers, including node 7, chosen to cover the entire testbed (Figure 2). Compared to Figure 3b, J-WIFI subjects the network to a noise slightly higher in average (~86dBm) and variance (Figure 3d); this affects significantly the network topology, increasing the average hopcount by 20% w.r.t. T-HIGH (Figure 4).

Figure 3e shows the noise generated by a J-MWO jammer on node 7. About 1/4 of the network (obviously including the sink) is severely affected, with an average noise from ~80 to ~65dBm, far higher than the previous scenarios. We experiment with alternate placements of the J-MWO jammer which clearly affects differently the sink, but also has different global effects on the network (Figure 4). Moreover, we also experiment with the combination of 2 J-MWO and 4 J-WIFI, the resulting noise (Figure 3f) is significantly higher than in all previous scenarios, yielding a stronger impact on network topology (Figure 4). This scenario combined with a reduced TX power of the network nodes (LP in Figure 4) is the most challenging we consider in this paper. Further, when studying specific effects of different jammers and their combinations on protocols, to eliminate the topology bias, we stick to a single 43-node network with the remaining 6 nodes being either active as jammers or switched off. Obviously, the 43-node network is more challenging as it is less connected (Figure 4).

4 BASELINE MAINSTREAM PROTOCOLS

We describe the protocols we use as a baseline to compare against CRYSTAL, along with the configuration used in the experiments. All protocols in this paper run atop Contiki.

4.1 Protocol Descriptions

RPL [25], the Routing Protocol for Low-power Lossy Networks, is an IETF standard. RPL can be seen as an evolution of CTP [10] that, instead of a tree, maintains a directed acyclic graph rooted at the sink. Therefore, each node maintains multiple parents towards the

Figure 3: Noise levels for the scenarios in Table 3.

Figure 4: Link quality distribution (PRR) and network radius (mean/max) in various interference scenarios.
We initially chose a value of 8Hz; this is the default, commonly used the default mode 3.

−

with several values ranging from 1, 2, 4 Hz, as they may provide better performance under interference. We observed this to be the case for ORPL, which performs best at 2Hz. Therefore, hereafter we report only about wake-up intervals of 2Hz and 8Hz; in general, these also strike a different balance between PDR and DC, and are therefore interesting to compare. The other configurations always perform worse, and are omitted due to space constraints.

Choosing the right CCA. The Clear Channel Assessment (CCA) mechanism is used by CSMA link layers to deter a packet transmission if the medium is busy. Its configuration significantly affects the interference resilience of the stack.

The CC2420 radio offers three modes where the CCA reports a busy medium upon detecting 1) energy above threshold 2) valid IEEE 802.15.4 data, regardless of energy threshold 3) energy above threshold or valid IEEE 802.15.4 data.

We verified that the default –90dBm energy threshold in Contiki-MAC yields unacceptable performance; baseline protocols achieve PDR<30% even with natural T-HIGH interference. We tested them with several values ranging from –60 to –90 dBm under T-HIGH and generated interference. The value of –77 dBm yielded the best performance and is our choice; in fact, this is the default for CC2420.

As for the CCA mode, the protocols considered use mode 3, the default. With JamLab nodes, the question arises whether the noise patterns they emit can be detected by other nodes as legit IEEE 802.15.4 data, instead of interference. We performed dedicated experiments comparing results obtained with CCA modes 1 and 3, observing essentially the same performance. Therefore, hereafter we used the default mode 3.

Retransmissions. RPL and ORPL employ different strategies w.r.t. layer 2 retransmissions; a maximum of 7 is allowed by RPL when an acknowledgment is not received, and 4 by ORPL. However, a retransmission can be triggered also by a CCA detecting a busy channel, in which case a few subtleties of the Contiki operating system come into play. Contiki v.3.0, used by RPL, considers 5 busy CCAs as equivalent to a failed transmission. The two events are completely unrelated in Contiki v.2.7, used by ORPL, allowing for an unlimited number of CCAs till the channel is free.

We did not modify these settings, as changing these default parameters may have unexpected and undesired effects whose analysis is outside the scope of this paper. We mention them here because they are useful in interpreting the results we present in the next section, e.g., the superior performance of ORPL under strong interference next to the sink.

5 CRYSTAL VS. THE MAINSTREAM

We compare the protocols in §4 against CRYSTAL, and indirectly Glossy, when exposed to the same interference. Aside from the intrinsic value and novelty of this experimental comparison, this serves a stepping stone towards a CRYSTAL design tolerating stronger interference, discussed in §6.

5.1 Experimental Setup

We analyze CRYSTAL and the baseline protocols in the interference scenarios described in §3. We setup a number U of concurrent senders between 0 and 48; U=0 means absence of traffic while U=48 offers a stress case where all nodes but the sink are senders. These parameters match the use cases described in [15] in which a data prediction scheme is applied to periodic data collection applications (e.g., sensing light in a road tunnel or temperature in an office environment). Data prediction reshapes traffic from periodic into sporadic; yet, in a single epoch, U nodes may need to transmit data. The PDR of Glossy is derived from CRYSTAL experiments as the PDR of the T phase when U=1 (Table 7).

In CRYSTAL, all U senders attempt their data packet transmission at exactly the same time, i.e., in the first T phase of the epoch, whose duration we set to E=2s. Baseline protocols have much higher latency, especially under interference; we set a longer E=10s for them, denoting solely the period according to which packets are generated. In reporting DC, we re-scale the values measured for CRYSTAL to 10s, to enable direct comparison between the two protocol classes. Unlike CRYSTAL, the epochs of baseline protocols are not synchronized.

Finally, all results are based on several 1-hour runs. For baseline protocols, these are preceded by a 30-minute period since bootstrap, allowing network topology to stabilize. For CRYSTAL, the total number of packets sent per configuration varied from 5000 to 500k, but typically was around 5k–40k. PDR is computed over the total number of packets sent. Instead, DC is the averaged over values from each 1-hour run, whose variation is anyway negligible.

5.2 Natural Interference: T-LOW

We first consider the T-LOW scenario (§3), which is akin to several evaluations in the literature, including [6, 11]. As expected, the MAC wake-up interval bears a significant effect: RPL performs best at 8Hz, while ORPL achieves near-perfect PDR at 2Hz. Further, its DC is much lower than RPL thanks to opportunistic behavior.

These results were derived with a single sender, U=1. Table 6 shows results for other values of U; we consider only ORPL as the performance of RPL is significantly lower. The PDR of ORPL decreases when traffic increases; ORPL still achieves a good PDR=97.8% with U=20, but drops to PDR=73% when all nodes transmit in each epoch. DC similarly increases sharply with U.
Concerning the mainstream baseline protocols, Table 4 shows a
generalized decrease in PDR accompanied by significant increases in DC. As in t-low, ORPL is the protocol with the best performance. The price to pay, however, is the nearly twofold DC increase for both 2 and 8Hz, as a result of longer idle listening and retransmissions induced by interference. Varying the number U of senders (Table 6) shows a similar trend of decreasing PDR and increasing DC.

Instead, Crystal performs quite well (Table 5). PDR is perfect or near-perfect regardless of the value of U: the occasional (4 out of total 700k) packet loss for some values of U is likely due to the unpredictable nature of t-high. Further, DC is nearly identical to the t-low case. For instance, in the worst-case scenario of N_T=3 and U=48, the increase in t-high w.r.t. t-low is a negligible 0.22%. This is partly ascribed to the inherent reliability of the Glossy protocol Crystal builds upon. However, our experiments also show that Glossy by itself does not achieve perfect PDR. The superior reliability of Crystal is due to its redundancy mechanisms built atop Glossy, overcoming daytime noise with little additional overhead. Another way to look at this is to observe that even in the configuration with N_T=2, i.e., less reliability in the Glossy layer, Crystal still achieves the same PDR as N_T=3, while of course enjoying better DC.

5.4 Generated Interference: J-WIFI
We turn our attention to noise patterns we can control via JamLab (§3). We first analyze a single j-wifi jammer next to the sink, then 6 of them fully covering the network. We focus on U=1 as this is sufficient to draw the observations motivating the further work described in the next sections.

Single jammer next to the sink. We use a single jammer, node 7 in Figure 2; its placement is challenging, at only 1m from the sink. RPL shows a reasonable PDR=84%, while ORPL yields near-perfect PDR with both 2 and 8Hz, and a DC comparable to t-high (Table 8). In the same conditions, Crystal achieves perfect PDR and lower DC than ORPL (Table 9). This remarkable performance is mainly a consequence of the perfect performance of Glossy (Table 7).

Six WiFi jammers covering the entire network. We next consider 6 JamLab nodes generating WiFi interference across the entire network like t-high, but with significantly higher noise (§3). As RPL shows low performance even with a single jammer, we focus on ORPL, which has significant difficulty overcoming this noise...
Table 8: Generated noise: Baseline, $U=1$.

<table>
<thead>
<tr>
<th>node ID</th>
<th>protocol</th>
<th>wake-up (Hz)</th>
<th>PDR (%)</th>
<th>DC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>RPL</td>
<td>8</td>
<td>84</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>89</td>
<td>1.30</td>
</tr>
<tr>
<td>7</td>
<td>ORPL</td>
<td>8</td>
<td>99.7</td>
<td>1.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>99.9</td>
<td>0.59</td>
</tr>
<tr>
<td>6</td>
<td>ORPL</td>
<td>8</td>
<td>60</td>
<td>3.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>64</td>
<td>1.70</td>
</tr>
<tr>
<td>42</td>
<td>ORPL</td>
<td>8</td>
<td>98.3</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>98.6</td>
<td>0.844</td>
</tr>
<tr>
<td>13</td>
<td>ORPL</td>
<td>8</td>
<td>99.7</td>
<td>1.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>98.0</td>
<td>0.67</td>
</tr>
<tr>
<td>7</td>
<td>ORPL</td>
<td>8</td>
<td>99.1</td>
<td>2.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>99.8</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 9: Generated noise: Crystal, $U=1$.

<table>
<thead>
<tr>
<th>node ID</th>
<th>$N_T$, $W_T$</th>
<th>$R$</th>
<th>PDR (%)</th>
<th>DC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>3, 8</td>
<td>2</td>
<td>100</td>
<td>0.457</td>
</tr>
<tr>
<td></td>
<td>2, 6</td>
<td></td>
<td>100</td>
<td>0.403</td>
</tr>
<tr>
<td>6</td>
<td>3, 8</td>
<td>2</td>
<td>100</td>
<td>0.497</td>
</tr>
<tr>
<td></td>
<td>2, 6</td>
<td></td>
<td>100</td>
<td>0.443</td>
</tr>
<tr>
<td>42</td>
<td>3, 8</td>
<td>2</td>
<td>100</td>
<td>0.507</td>
</tr>
<tr>
<td></td>
<td>2, 6</td>
<td></td>
<td>99.52</td>
<td>0.430</td>
</tr>
<tr>
<td>13</td>
<td>3, 8</td>
<td>2</td>
<td>100</td>
<td>0.459</td>
</tr>
<tr>
<td></td>
<td>2, 6</td>
<td></td>
<td>100</td>
<td>0.405</td>
</tr>
<tr>
<td>7</td>
<td>3, 8</td>
<td>2</td>
<td>78.5</td>
<td>0.433</td>
</tr>
<tr>
<td></td>
<td>2, 6</td>
<td></td>
<td>78.6</td>
<td>0.425</td>
</tr>
<tr>
<td>7</td>
<td>3, 8</td>
<td>6</td>
<td>100</td>
<td>1.11</td>
</tr>
<tr>
<td>7</td>
<td>6, 12</td>
<td>2</td>
<td>100</td>
<td>0.839</td>
</tr>
</tbody>
</table>

level, regardless of the wake-up interval; in the best case, 2Hz achieves PDR=64% (Table 8).

Glossy achieves near-perfect PDR (Table 7), becoming perfect once combined with the Crystal mechanisms built atop, yielding a DC only 12% higher than T-LOW (Table 9). However, when $U>1$ (results omitted due to space limitations) Crystal experiences a slight PDR decrease of 1–2%. The reason is that, with very high interference throughout the network, no alternate, good paths exist for packets to reliably reach the sink.

### 5.5 Generated Interference: J-mwo

We study the impact of a JamLab-emulated microwave oven, causing interference much stronger than WiFi and with different temporal patterns (§3). We move the jammer progressively closer to the sink, yielding increasingly challenging scenarios. Given the results in the previous section, our comparison against mainstream protocols considers only ORPL, as RPL yields unacceptable performance.

**Jammer far from the sink, node 42.** We first use a jammer on node 42, far from the sink, in a corner of the network, and amid a dense neighborhood; its noise affects neighboring nodes, but bears limited influence to the rest of the network.

ORPL performs well in this scenario (Table 8) although with a DC increased w.r.t. lower-noise scenarios. This is due to its buffering and continuous attempts to re-transmit packets until it finds the channel free (§4.2). Recall that the J-mwo scenario induces periods of strong interference alternated to periods with no interference (§2.2). Therefore, the buffering and infinite CCA retries in ORPL effectively delay packets when the microwave oven interference is active, enabling their transmission during no-interference periods. Nevertheless, these retransmissions do increase the DC.

Crystal instead achieves perfect PDR (Table 9). Nevertheless, the underlying Glossy layer is affected by interference (Table 7); therefore, reliability in Crystal comes at the cost of a higher DC. This cost is even higher than with 6 WiFi jammers, although in the latter case the PDR of Glossy is worse. The reason is the position of node 42; being in a corner of the network, its strong interference causes the loss of acknowledgments in that neighborhood, triggering retransmissions from the corresponding senders and unnecessarily keeping the entire network awake to help forwarding. Instead, in the scenario with 6 WiFi jammers covering the entire network, packet losses are spatially and temporally distributed, and the redundancy brought by both Glossy and Crystal enables packets to more easily find routes “around” the interference.

**Jammer close to the sink, node 13.** We now move the jammer to node 13 at ~4m from the sink. Intuitively, this is likely to be more disruptive than the far-away node 42, but less than an even closer placement, discussed next.

Yet, our results tell a different story. The PDR of ORPL is nearly perfect (Table 8) and achieved with a DC ~20% lower w.r.t. node 42 above. The same holds for Crystal (Table 9), which achieves perfect PDR with a DC ~9% lower w.r.t. node 42, thanks to the perfect reliability of Glossy (Table 7). This improved performance arises from jammer position. Node 13 is closer to the sink than 42 and induces stronger interference on it, but it is also more “central”, allowing packets to follow routes “around” it. Instead, node 42 is in the network corner, where noise disruption is much harder to compensate via alternative routes.

**Jammer next to the sink, node 7.** When moving the J-mwo jammer on node 7, at 1m from the sink, the PDR of Crystal significantly degrades for the first time, causing a 21.5% packet loss (Table 9), mainly due to the fact that, unlike previous scenarios, the underlying Glossy layer loses 32.1% of the packets. The reason is that the interference on node 7 is so strong and so close that Glossy cannot overcome it. Receiving packets via alternate routes, as with node 13, is no longer an option because all routes are jammed by interference, given that the sink is basically at the center of it.

In contrast, ORPL achieves near-perfect PDR also in this case (Table 8) and with a DC only marginally different w.r.t. the interference source on node 13. From the point of view of ORPL, the two situations are virtually the same: i) both node 7 and 13 are in the center of the network, unlike the more challenging corner placement of node 42, and ii) in both cases, buffering and retransmissions guarantee that a packet not received by the sink due to interference is eventually received in the periods without it.

Instead, Crystal dissemination is designed to be as fast as possible, even with the redundancy it builds atop the even shorter one-shot Glossy floods. Consequently, Crystal and Glossy cannot exploit a “wait-and-see” strategy as in ORPL.

### 5.6 Is There a Better Configuration?

We study a configuration yielding perfect PDR in the worst scenario for synchronous transmissions, i.e., node 7 as J-mwo jammer. We explore two options: in Crystal, and in the underlying Glossy.
Crystal: Keeping the network awake. We observed that an asset of ORPL is that it can retransmit until interference ceases. The Crystal analogous comes from increasing $R$, i.e., the number of consecutive silent TA pairs detected before determining that communication is over and it is safe to enter sleep mode until the next epoch ($\S 2.1$). Increasing $R$ keeps the network awake longer, even when the sink reports via its $A$ slot that no packet arrived in the previous $T$ slot. This gives senders more opportunities to attempt retransmission under interference. Indeed, Table 9 shows that $R=6$ enables perfect PDR. However, keeping the network awake for $3x$ longer than before causes a nearly $3$-fold increase in DC.

Glossy: Increasing redundancy. An alternative is to make the underlying Glossy layer more reliable. The main knob to achieve this is to increase the number $N$ of retransmissions during a flood, and increase the slot duration $W$ to ensure the flood has enough time to complete ($\S 2.1$). We verified that, when pure Glossy is used in isolation, a setting $N=10$, $W=17$ yields $PDR=99.76\%$. However, the reliability provided by Crystal atop Glossy enables the use of a smaller $N$, considerably reducing DC. Table 9 shows that with $N_f=6$, $W_f=12$, Crystal achieves perfect $PDR$ (despite Glossy yielding only $PDR=83.86\%$, see Table 7) but nearly doubles DC, as each packet is transmitted twice as many times w.r.t. $N_f=3$.

In summary, a proper static configuration of Crystal or Glossy parameters enables perfect reliability but with unacceptable power consumption w.r.t. ORPL (which however does not achieve perfect $PDR$). Ideally, perfect $PDR$ should come without increasing significantly the DC observed in the other scenarios in Table 9, i.e., at most $0.50\%$. Further, over-provisioning for the worst case, as these static configurations do, is undesirable. Ideally, Crystal should dynamically adapt to interference, bearing extra energy costs only when needed.

6 TAMING STRONG INTERFERENCE

We illustrate a technique to escape interference and a complementary one to fight it after detecting its presence.

Escaping Interference: Channel Hopping. Exploiting frequency diversity is a well-known technique for interference resilience. Interference usually affects only some of the 16 channels available in IEEE 802.15.4 ($\S 2.2$). Therefore, a channel-hopping sequence can be used network-wide to enable subsequent TA pairs to move to different channels, reducing the probability that two consecutive ones both execute on noisy channels. This simple modification does not affect any Crystal parameters.

Channel hopping is driven by the $S$ phase (Figure 5); the channels of TA pairs in the epoch depend on the $S$ channel, itself based on a predefined sequence. This mechanism realigns all nodes to the same channel at the epoch start, independent of the number of TA phases they executed in the previous one.

A key decision is which channel to use next. WiFi and microwave ovens are common noise sources, jamming 4 and 7 adjacent channels, respectively ($\S 2.2$). Spacing the current and next channel apart by 4 channels is sufficient to escape WiFi, but not microwave ovens. Therefore, our implementation uses a hopping sequence with $7$-channel spacing; alternate hopping sequences can exploit a priori knowledge about interference. Notably, selecting the number of channels to hop over requires little knowledge compared, e.g., to approaches that probe the environment and limit themselves to channels with the least interference [28].

Fighting Interference: Noise Detection. Our next technique relies on the ability to detect abnormally high noise levels. Recall from $\S 2.1$ that, in Crystal, the distributed termination condition relies on counting silent pairs and missed acknowledgements. Under high noise, these missing-packet conditions often occur even when a packet was transmitted, but encountered interference. If noise strikes during the $T$ phase close to the sink, the sender will re-transmit the packet in the next $T$ slot. If the sink still does not receive the packet in $R$ consecutive $T$ slots, it mistakenly detects termination and puts the whole network to sleep. Instead, noise in the network periphery may cause a node to similarly miss $Z$ acknowledgements and go to sleep, likely before the sink. In both cases, data may remain un-delivered because termination was falsely detected.

Adding noise detection and changing termination conditions fights these cases. Noise detection can be easily achieved by periodically checking the CCA pin of CC2420; in our implementation, all nodes perform the CCA every $64\mu$s while listening during $T$ or $A$ phases, and define high noise when $RSSI > -60$dBm is detected at least $80$ times. This threshold is designed to detect only very high noise, e.g., a microwave oven; lower thresholds would unnecessarily trigger the scheduling of extra TA pairs, e.g., in the WiFi scenarios of $\S 5$, where even the unmodified Crystal achieves perfect reliability.

As for distributed termination, intuitively, in the presence of noise missing packets do not count towards termination, keeping the network awake and allowing more opportunities for data and acknowledgments to escape the interference. Recall that receiving any packet keeps a node awake to serve as a forwarder. We make the following modifications to Crystal:

- define $R_{noise}$ as the maximum number of consecutive slots
  - i) without a packet and ii) with high noise.
- change the termination rule at the sink; the network goes to sleep when either i) $R$ non-noisy no-data $T$ slots occur since the last received data, or ii) $\max(R, R_{noise})$ consecutive noisy no-data $T$ slots occur.
- change the termination rule elsewhere; a node goes to sleep when either i) it receives a sleep command from the sink, or ii) it detects $Z$ non-noisy no-data slots since the last packet received in $T$ or $A$, or iii) $\max(Z, R_{noise})$ consecutive noisy, silent $A$ slots occur.

We empirically determined that $R_{noise}=6$ strikes a good balance between reliability and energy consumption.

Fighting and Escaping Interference. Although both these techniques improve performance along some dimension, it is only

![Figure 5: Channel hopping in Crystal. The number on each Crystal phase denotes the channel used.](image-url)
through their combination that very strong interference can be effectively overcome with very low energy consumption, as shown next. Indeed, frequency diversity reduces the probability of the sink to be exposed, in consecutive TA pairs, to high noise levels from the same source, mitigating the above drawback of noise detection. On the other hand, the ability to detect and react to noise is helpful in reducing packet loss when hopping from one bad channel to another one.

7 UNDER STRONG INTERFERENCE
We now evaluate the techniques in §6 and show that they not only overcome the interference scenarios considered in §5, but also sustain much higher noise levels, detailed next.

7.1 Experimental Setup
We extend our experimental setup along two dimensions.

Channel mapping. Testing our channel hopping mechanism in principle requires reproducing interference across multiple channels, something JamLab cannot do (§2.2). We overcome this limitation via a mapping between the 16 channels of IEEE 802.15.4 and those in the testbed. Whenever our channel hopping mechanism decides to switch to a channel $c$, a corresponding channel $c_{\text{real}}$ is instead used for communication, based on a predefined mapping $c \rightarrow c_{\text{real}}$ based on the interference types and channels affected we want to reproduce. For instance, when emulating a microwave oven, we map channels 20–26 to the real one used by j-mwo jammers.

More challenging interference scenarios. As described later, extending CRYSTAL with the techniques in §6 allows it to sustain much stronger interference than the one in §5, which considered the separate effect of generated j-wifi and j-mwo interference. Therefore, we now focus on the combined effect of these two interference types. We combine them in two ways, yielding the scenarios in Table 10. The first, $\text{COMBINED}_{\text{split}}$, combines the two types of interferences by placing each on different real channels. This significantly reduces the chances that channel hopping finds a good channel, and increases the likelihood to hop from one type of interference to the other. The second scenario, $\text{COMBINED}_{\text{nd}}$, is even more challenging, placing j-mwo and j-wifi jammers on the same real channel, generating noise that is the sum of the two. Increasingly challenging scenarios can be generated by determining the number $n$ of channels this strong interference is mapped to. Table 10 shows we experiment with $n$ ranging from 7 (i.e., when j-mwo and j-wifi fully overlap) to 16 (i.e., all channels jammed by the same combined interference).

Besides combining interference types, we also strengthen j-mwo, the most disruptive one, by using 2 jammers simultaneously, the worst in §5: node 7 next to the sink, and node 42 in the corner. As for j-wifi, using the scenario with 6 jammers would force us to remove 8 nodes in total, further reducing the network size. Therefore, we used 4 j-wifi jammers that, we verified, yield a noise pattern close to natural T-HIGH.

As mentioned at the end of §3, we use the resulting 43-node network across all scenarios. To re-establish our ORPL baseline, Table 11 (left) reports experiments with 4 j-wifi, 2 j-mwo, and their combination over a single channel. ORPL performance is good also with 2 j-mwo, but degrades significantly even with only 4 j-wifi jammers, instead of the 6 used in §5.4.

7.2 Channel Hopping
We are now ready to study CRYSTAL extended with channel hopping as discussed in §6. We call this variant CRYSTAL$_{\text{CH}}$, to distinguish it from the original single-channel one, and call CRYSTAL$_{\text{ND}}$ the variant that also adds noise detection.

Table 12 reports experiments under natural T-HIGH interference, without channel mapping and with CRYSTAL$_{\text{CH}}$ hopping across all 16 channels. A comparison with Table 5 shows that CRYSTAL achieves perfect PDR (no packets lost of total 150k sent) regardless of $U$, and does so with $N_T=2$, which generally yields worse PDR w.r.t. $N_T=3$. Further, DC is 1–2% lower than the single-channel version under T-HIGH.

A bigger question lingering from §5 is whether CRYSTAL$_{\text{CH}}$ can overcome j-mwo interference next to the sink. We first analyze the performance in the $\text{COMBINED}_{\text{split}}$ scenario (Table 13). Recall this subsumes the scenario with j-mwo on node 7 (end of §5) by adding a second jammer on node 42, defining a much more challenging setup. Indeed, when hopping out of j-mwo interference, found with a $T=43.75\%$ probability, there is still a 37.5% chance to stumble on j-wifi interference, and only a 18.75% chance to enjoy T-low interference. Nevertheless, CRYSTAL$_{\text{CH}}$ achieves perfect PDR for $U=1$ and three-nines reliability for $U>1$, $N_T=3$. Further, this is achieved with only a slight increase in DC w.r.t. our lowest-interference scenario, T-LOW: 14.3% and 12.6% for $N_T=2$ and $N_T=3$, respectively.

The next step is to identify the limit of CRYSTAL$_{\text{CH}}$, which clearly depends on the type of interference applied and number of channels affected. Table 15 explores this limit by using the $\text{COMBINED}_{\text{nd}}$ scenario of Table 10. The interference is stronger, as it is the sum of 2 j-mwo and 4 j-wifi, which in Table 13 are instead split on

---

**Table 10: Scenarios with combined interference generated by 2 j-mwo and 4 j-wifi.**

<table>
<thead>
<tr>
<th>scenario</th>
<th>#channels jammed</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{COMBINED}_{\text{split}}$</td>
<td>7, 6</td>
<td>jammers on different real channels based on type, mapped on different sets of channels</td>
</tr>
<tr>
<td>$\text{COMBINED}_{\text{nd}}$</td>
<td>$n \in {7, 10, 13, 16}$</td>
<td>all jammers on one real channel, itself mapped on $n$ channels</td>
</tr>
</tbody>
</table>

**Table 11: ORPL (2Hz) in a 43-node network, $U=1$.**

<table>
<thead>
<tr>
<th>scenario</th>
<th>TX Power 0 dBm</th>
<th>TX Power ~7 dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-LOW</td>
<td>99.6</td>
<td>0.997</td>
</tr>
<tr>
<td>T-HIGH</td>
<td>98.5</td>
<td>0.776</td>
</tr>
<tr>
<td>4 j-wifi</td>
<td>61.0</td>
<td>1.35</td>
</tr>
<tr>
<td>2 j-mwo</td>
<td>97.8</td>
<td>1.19</td>
</tr>
<tr>
<td>2 j-mwo 4 j-wifi</td>
<td>65.0</td>
<td>2.14</td>
</tr>
</tbody>
</table>
When interference affects so many channels that it becomes difficult to escape it, the only other choice to improve reliability is to fight it with noise detection (§6).

Indeed, starting from 10 channels jammed, unlike the channel hopping alone, its combination with the noise detection (Table 15, right) achieves two- to three-nines PDR with $U=1$. For $U>1$ the performance gain is even more visible as CrystalCH achieves $PDR>96.8\%$ even with 13 channels jammed.

Noise detection becomes more important as the number $n$ of jammed channels increases. The extreme case is when all channels are jammed by the same strong interference (Table 14); channel hopping becomes pointless and reliability is provided entirely by noise detection, which performs quite well. Indeed, the $PDR$ achieved here is only marginally lower than in combined, with the worst-case $U=42$ achieving $PDR=90\%$. To put this value in context, we observe that it is $i)$ comparable with what RPL achieves in t-low with $U=1$ (Table 4), and $ii)$ more than what ORPL achieves in the natural t-high (no microwave ovens) with $U=20$ (Table 6).

The price to pay for this remarkable reliability is energy consumption. A drawback of noise detection is that high noise keeps the network awake even without packet transmissions (§6). This is reflected in the DC increase as the number $n$ of jammed channels increases, which increases the likelihood of remaining unnecessarily awake. This is clearly undesirable for $U=0$; yet, it is key to reliability as $U$ increases, as seen by comparing the two sides of Table 15. The actual impact of this increased DC on the overall energy consumption depends on the aperiodic traffic at hand, as we analyze in §7.5.

On the other hand, we also verified that under t-high, unlike the extremely challenging scenario above, the DC of CrystalND does not increase w.r.t. CrystalCH (Table 12) since interference is never strong enough to trigger our noise detection mechanism.

### 7.3 Channel Hopping and Noise Detection

These scenarios are very challenging, both in absolute terms and w.r.t. the literature, making the performance of CrystalCH already remarkable. Nevertheless, we can push reliability even further. When interference affects so many channels that it becomes difficult to escape it, the only other choice to improve reliability is to fight it with noise detection (§6).

Indeed, starting from 10 channels jammed, unlike the channel hopping alone, its combination with the noise detection (Table 15, right) achieves two- to three-nines PDR with $U=1$. For $U>1$ the performance gain is even more visible as CrystalCH achieves $PDR>96.8\%$ even with 13 channels jammed.

Noise detection becomes more important as the number $n$ of jammed channels increases. The extreme case is when all channels are jammed by the same strong interference (Table 14); channel hopping becomes pointless and reliability is provided entirely by noise detection, which performs quite well. Indeed, the $PDR$ achieved here is only marginally lower than in combined, with the worst-case $U=42$ achieving $PDR=90\%$. To put this value in context, we observe that it is $i)$ comparable with what RPL achieves in t-low with $U=1$ (Table 4), and $ii)$ more than what ORPL achieves in the natural t-high (no microwave ovens) with $U=20$ (Table 6).

The price to pay for this remarkable reliability is energy consumption. A drawback of noise detection is that high noise keeps the network awake even without packet transmissions (§6). This is reflected in the DC increase as the number $n$ of jammed channels increases, which increases the likelihood of remaining unnecessarily awake. This is clearly undesirable for $U=0$; yet, it is key to reliability as $U$ increases, as seen by comparing the two sides of Table 15. The actual impact of this increased DC on the overall energy consumption depends on the aperiodic traffic at hand, as we analyze in §7.5.

On the other hand, we also verified that under t-high, unlike the extremely challenging scenario above, the DC of CrystalND does not increase w.r.t. CrystalCH (Table 12) since interference is never strong enough to trigger our noise detection mechanism.

### 7.4 A Different Topology: Low Power

We present results with the lower transmission power of $−7$dBm. This reduces the number of neighbors and increases network diameter (Figure 4), yielding a more challenging topology.

To re-establish the ORPL baseline, we repeated experiments in the new topology (Table 11, right). ORPL performs close to the high-power setting with only minimal (t-low) or j-mwo interference, but shows drastic performance degradation in the presence of j-wifi, with an almost halved PDR.

We ran several Crystal experiments, confirming the trends hitherto observed. However, DC increases slightly in all cases, as we must use larger Glossy slots to handle the larger network diameter...
Table 16: Low power: CrystalCHNd.

<table>
<thead>
<tr>
<th>U</th>
<th>NT</th>
<th>PDR (%)</th>
<th>lost 1 pkt in</th>
<th>DC (%)</th>
<th>NT</th>
<th>PDR (%)</th>
<th>lost 1 pkt in</th>
<th>DC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0.541</td>
<td>1.072</td>
<td>100</td>
<td>4</td>
<td>0.252</td>
<td>3.340</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.709</td>
<td>0.969</td>
<td>100</td>
<td>4</td>
<td>0.527</td>
<td>1.942</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.865</td>
<td>1.696</td>
<td>100</td>
<td>4</td>
<td>0.736</td>
<td>2.646</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.335</td>
<td>2.646</td>
<td>100</td>
<td>4</td>
<td>0.942</td>
<td>4.194</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>2.125</td>
<td>8.032</td>
<td>100</td>
<td>4</td>
<td>0.942</td>
<td>4.194</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>7.562</td>
<td>7.784</td>
<td>100</td>
<td>4</td>
<td>0.942</td>
<td>4.194</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>3</td>
<td>Inf</td>
<td>14.982</td>
<td>100</td>
<td>4</td>
<td>0.942</td>
<td>4.194</td>
<td></td>
</tr>
</tbody>
</table>

Table 17: CrystalCHNd: performance with the aperiodic, sparse, real-world traffic profile shown in Table 1.

<table>
<thead>
<tr>
<th>interference scenario</th>
<th>NT = 2</th>
<th>NT = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDR (%)</td>
<td>DC</td>
<td>PDR (%)</td>
</tr>
<tr>
<td>T-LOW</td>
<td>100</td>
<td>0.105</td>
</tr>
<tr>
<td>T-HIGH</td>
<td>100</td>
<td>0.105</td>
</tr>
<tr>
<td>combined</td>
<td>99.487</td>
<td>0.198</td>
</tr>
</tbody>
</table>

Further, this is achieved with DC = 0.2% depending on NT. This is twice the baseline established by natural interference, but in absolute terms it is remarkably small w.r.t. the energy consumption commonly reported in the state of the art.

8 RELATED WORK

We survey approaches that share our goal of making multi-hop protocols for low-power wireless communication resilient to environmental interference. Interference has also been studied from a security perspective by identifying several types of jamming attacks and related countermeasures. Although not directly related to our contribution, these techniques may inspire alternate resilience mechanisms; we refer the interested reader to [19].

CSMA + Channel Hopping. Adding channel hopping to combat interference is well accepted in the literature, with recent works modifying standard, CSMA protocol stacks. MiCMAC [20] extends ContikiMAC with channel hopping, resulting in a synchronization-free MAC with high PDR under WiFi interference. MiCMAC mechanisms require transmitting and receiving nodes to synchronize in time as well as across channels, increasing latency. Oppcast [18] and MOR [28] offer full-stack alternatives to RPL and MicMAC, combining channel hopping and opportunistic routing to combat high latencies while also escaping high interference.

As MOR code is not available, we offer an informal, numerical comparison with the evaluation in [28], performed on FlockLab with WiFi on one channel and an effective U = 2.1. Using this jammed channel plus two free ones, MOR shows the best results: PDR = 99.35% and DC = 1.56%. We compare to a more challenging scenario with constant, generated WiFi traffic on all channels in our testbed where Crystal shows PDR = 100% and DC = 0.559 for U = 2. This DC is nearly three times smaller w.r.t. MOR, and achieved without any interference avoidance mechanisms. Naturally, with more concurrent packets, the DC of Crystal increases, however the same is true for other protocols. Further, in the absence of traffic, a common case in §7.5, Crystal maintains DC < 0.4%, levels that duty cycling protocols cannot achieve due to required periodic channel probing. Finally, to manage latency, these protocols hop among few channels, selected during pre-deployment evaluations. In contrast, Crystal can use all channels without affecting its performance, allowing it to adapt to changing interference.

TDMA + Channel Hopping. TSCH [26] with Orchestra [5] scheduling offers a protocol in which all nodes follow a repeating, slotted schedule, with local and independent slot allocation. The number and type of slots is statically determined, according to expected traffic. Results from Indriya show Orchestra with 47 slots maintains PDR = 99.99% with an average DC = 0.4%, without interference; in an analogous setting, Crystal consumes twice as much, DC = 0.8%. However, in Orchestra the duty cycle of nodes varies significantly.
across the network, with nodes closer to the sink reporting much higher values. Further, Orchestra is designed for periodic data, which is critical to statically configure slot parameters. In the aperiodic, dynamic scenarios considered in §7.5, Orchestra would overdimension for the worst case, unable to reduce DC under low traffic. Finally, Orchestra has not been evaluated under interference.

**Synchronous Transmissions + Channel Hopping.** The combination of channel hopping and synchronous transmissions has also been used to increase parallelism for bulk data dissemination in Splash [2] and Pando [3]. Both protocols also see improvements due to diverse noise levels across channels, but their approaches are not competitive at low data rates.

In the context of the EWSN dependability competition [23], the three winning approaches in 2017 [7,17,21] perform channel hopping inside Glossy, a contrast to the noise resilience mechanisms we designed on top of Glossy. However, these solutions were highly specialized for the (single-sender) competition scenario and are not immediately reusable towards our goals. Instead, we evaluated CrystalND with concurrent senders and in a wide range of intense interference. Analyzing and exploiting the interplay between Glossy-level channel hopping and our Crystal-level techniques is intriguing, but is beyond the scope of this paper, albeit in our short-term research plans.

9 CONCLUSIONS AND FUTURE WORK

This paper set out to evaluate Crystal’s ability to sustain aperiodic, sparse traffic under strong interference. As Crystal relies on Glossy, we also offer a noise resilience evaluation for it, along with the two mainstream protocols, RPL and ORPL, we chose as baselines.

Unlike existing works limited to natural WiFi interference, we subjected these protocols also to the stronger noise generated by JamLab-emulated microwave ovens, which exhibit different interference patterns similar to those found in real environments. In our reproducible and controlled setup we showed, for the first time, that ORPL is very resilient to this type of interference, while Crystal is not. This motivated us to extend it with a combination of channel hopping and noise detection. We showed that our enhanced CrystalND protocol achieves unprecedented, near-perfect reliability even against the combination of emulated WiFi and microwave ovens, along with a per-mille duty cycle in the aperiodic, sparse traffic targeted by Crystal.

Regarding future work, a promising avenue is to distill the knowledge gained from the experimental campaigns presented in this paper into models able to identify the proper Crystal configuration given a known or estimated pattern of interference; this could potentially inform and greatly simplify in-field system configuration.

Furthermore, our work shows that effective strategies to overcome interference can be implemented atop Glossy, in contrast with the current trend of incorporating them inside it. A design similar to ours could be applied, in principle, to other Glossy-based systems (e.g., [9], [16]). However, a more fundamental research question, implicitly opened by this paper, concerns the tradeoffs between the two approaches; further study is required to identify under which conditions the techniques considered are more effective if implemented atop Glossy or, vice versa, inside it. The answer to this question could enable a novel design combining both approaches, yielding unprecedented levels of resilience to interference.

The source code of Crystal is freely available as open source at https://github.com/d3s-trento/crystal.

10 ACKNOWLEDGEMENTS

This work is partially supported by the CrystalClear project funded through the EU H2020 project WiSHFUL (grant agreement n. 645274), and by the Approxim@ction project funded by EIT Digital.

REFERENCES