Oppcast: Exploiting Spatial and Channel Diversity for Robust Data Collection in Urban Environments

Mobashir Mohammad
School of Computing
National University of Singapore
mobashir@comp.nus.edu.sg

XiangFa Guo
School of Computing
National University of Singapore
xiangfa@comp.nus.edu.sg

Mun Choon Chan
School of Computing
National University of Singapore
chanmc@comp.nus.edu.sg

***Abstract*—ZigBee shares the 2.4 GHz ISM band with a number of wireless technologies like WiFi, Bluetooth, and common household appliances like a microwave and a cordless phone to name a few. Due to the large-scale penetration of these technologies in urban environments, ZigBee communication suffers from severe cross-technology interference (CTI). Data collection in the presence of such highly dynamic CTI is quite challenging. Our work first examines the different deployment environments under the influence of planned and unplanned CTI and later proposes Oppcast, a robust and energy-efficient data collection protocol that carefully exploits a combination of spatial and channel diversity to eliminate the need for performing expensive channel estimation in advance.**

Our extensive evaluation in both a large-scale testbed (Academic Institution) and various urban environments (Carpark, Residential Complex, Shopping Mall and Cafeteria) shows that Oppcast is not only robust to CTI with reliability consistently maintained above 98.55%, but is also up to 2.4 times more energy efficient than the state-of-the-art data collection protocols. The rationale behind Oppcast exhibiting high robustness in highly dynamic environments is a significant increase in the number of communication opportunities it gets by exploiting multiple routes over multiple channels towards the destination.

I. Introduction

In the past decade or so, extensive research has been done to improve the performance of low-power wireless communication. These protocols have shown to be highly effective in relatively stable and controlled environments, for example, indoor WSN testbeds such as Indriya [6], FlockLab [31], Twist [20], and Motelab [38]. In these testbeds, WiFi interference tends to be restricted to a relatively fixed set of channels. Typically, the APs are set up to operate on the non-overlapping WiFi channels. In the 2.4GHz spectrum, such a planned WiFi deployment allows ZigBee to coexist and perform well only if certain ZigBee channels (e.g., Channels 15, 20, 25 and 26) are utilized.

Recently, a lot of emphasis has been paid to successfully classify the different sources of cross-technology interference (CTI) (e.g., WiFi, Bluetooth, Cordless Phones and Microwave) based on fast channel sampling and to take necessary actions after determining the cause of the drop in a protocol's performance. The observation that motivated these works is that each of the different CTI sources exhibits unique features in the signal samples that can be collected in an Anechoic chamber. However, from our measurements, in environments whereby the APs are owned by different entities and set-up in an unplanned manner, the CTI problem can be much more dynamic and severe. Such environments are actually fairly common if one considers AP deployments in residential buildings and on streets where shop owners deploy their own WiFi networks. The channel interference patterns can vary significantly over time when the WiFi network usage changes as residents leave/return home and shop owners open/close their shops.

Deployment of WSNs in an urban environment with fast changing CTI behavior and/or unplanned WiFi deployment is challenging in two ways. First, as the CTI behavior changes, it is often not possible to simply pick a "good" channel and stay on it. Second, with sufficient APs deployed in an unplanned manner, there may not be any "good" channel at all that is free from CTI.

In this paper, we propose *Oppcast*, a routing protocol that is robust in urban environments with dense unplanned AP deployment. The key idea behind Oppcast is that it exploits spatial and channel diversity at the same time. By combining opportunistic message forwarding and the use of multiple channels, the set of potential receivers increases significantly with a small increase in overhead but substantial improvement in reliability.

Our evaluation shows that Oppcast significantly outperforms ORPL (an opportunistic routing protocol using a single channel) and MiCMAC (a multi-channel MAC protocol that runs over RPL) in terms of reliability, in particular, in the presence of CTI. When evaluated in urban areas with unplanned WiFi networks, Oppcast achieves reliability over 98.55% in all cases, while the reliability of ORPL varies from 68% to 100% depending on the environment.

In this work, we make the following contributions:

- We observe that interference in an urban environment is highly dynamic because of the presence of unplanned CTI. This causes difficulty in finding a ZigBee channel that remains "good" for an extended duration, thus causing performance degradation of existing protocols due to reduced communication opportunities.
- We propose Oppcast, a protocol that allows robust data collection in an urban environment without channel estimation by exploiting spatial and channel diversity.

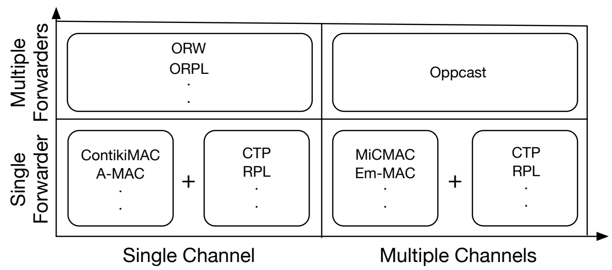


Fig. 1. Positioning Oppcast among other protocols.

- We implement Oppcast on Contiki and show that it is highly robust, achieving a reliability of at least 98.55% during 255 hours of experiments in four different urban deployments suffering a large amount of unplanned CTI.

The rest of this paper is organized as follows. We present related work in Section II and the motivation for our protocol in Section III. The design of Oppcast is presented in Section IV, followed by the evaluation results in Section V. Finally, we discuss some limitations in Section VI, before concluding in Section VII.

II. Related Work

With "IoTification" leaving a highly congested 2.4 GHz ISM band at one's disposal, handling CTI has become increasingly important. Over the last decade or so, enormous research has been performed to design protocols robust to CTI. More specifically, the work in this area can be broadly aligned along one of the following major directions: (1) Identification and classification of the source of interference, (2) Tackling interference by exploiting spatial diversity, (3) Evading interference by using channel diversity, (4) Surviving interference by adding redundancies or error correcting capabilities, and (5) Performing interference cancelation using smart antennas (MIMO).

A considerable amount of effort has been put into detection and classification of interference [21], [22], [25], [33], [34]. Musaloiu-E. et al. [33] showed that co-located ZigBee and WiFi networks could cause up to 58% packet losses in ZigBee and proposed a WiFi interference detection technique using periodic RSSI sampling. SoNIC [21] later introduced a novel approach to identify the source of interference based on the uniqueness of the RSSI fingerprints that each interference source generates when sampled at an extremely high rate for a short duration. The authors showed significant packet error ratio reduction when SoNIC was implemented at the sink node. Recently, TIIM [22] also proposed a lightweight machine learning approach to detect the interference source followed by suggesting suitable interference mitigation strategies. By doing this, TIIM could achieve 30% improvement in packet reception rate. The common drawback of most of these approaches is the measurement overhead, which can be substantial given that the interference can be quite dynamic. This makes such solutions inefficient for energy-constrained devices.

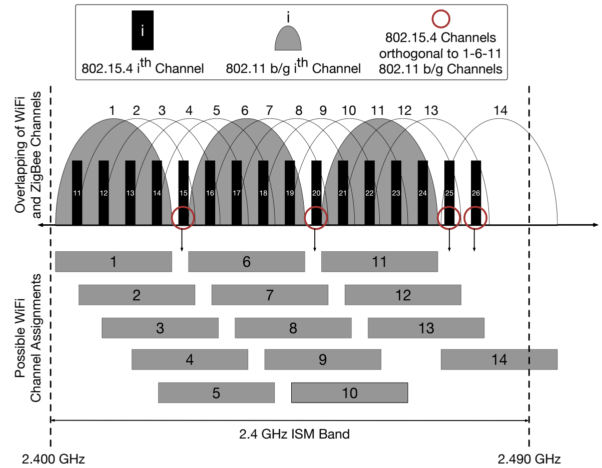
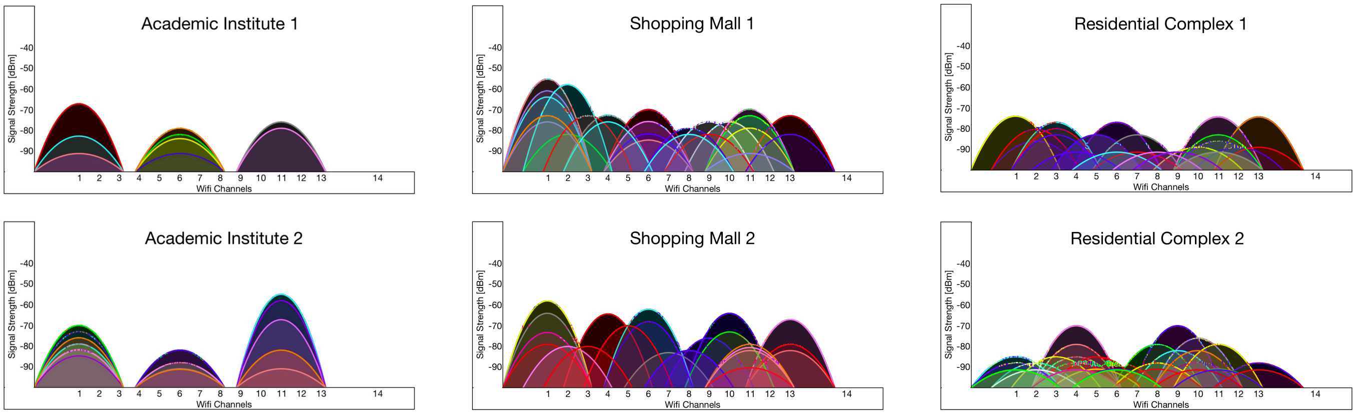


Fig. 2. Coexistence of ZigBee and WiFi in the 2.4GHz ISM band leading to CTI under different WiFi AP channel assignment strategies.

Interference mitigation by exploiting spatial diversity is another interesting technique which allows a node to opportunistically send packets to any next-hop node that provides "sufficient progress" [3], [13], [28]. One such opportunistic routing protocol for data collection is ORW [28], which makes use of EDC as a routing metric to define "sufficient progress". ORPL [13], a more recent protocol also uses ORW-like opportunistic routing and provides support for one-to-many and any-to-any communications besides many-to-one data collection. Because both ORW and ORPL run over a single channel, their performance is highly susceptible to the availability of a "good" channel. In addition, due to the use of repeated anycast transmissions until the reception of an explicit acknowledgement to achieve opportunistic routing, many duplicate packets are generated, especially when the network is dense (causing multiple neighbors to overhear the packet) and/or channel is interfered (causing multiple retransmissions). Although this substantial increase in network traffic enhances packet reception reliability, it increases overall energy consumption.

Exploiting channel diversity to evade interference has also been studied for a long time. Many multi-channel protocols for synchronous and asynchronous communication to improve reliability have been proposed in the past [8], [26], [39]. In [37], it is shown that a simple channel hopping strategy can significantly improve network's performance. A recent research, exploiting multiple channels, called ILTP [7] allows smart channel hopping to exploit "intermediate" quality (IQ) links by transforming them into "good" links. ARCH [32] is another multi-channel protocol that switches to a remote channel in the frequency domain if the current channel is poor.

Innumerable MAC protocols for WSNs have also been proposed [4], [24], [27], [29], [36] with the most recent work being MiCMAC [1], which promises significant improvements in terms of packet reception reliability on a real testbed in the presence of interference. However, protocols exploiting a set of channels involve the additional cost of identifying that set of



(a) Academic Institutes (b) Shopping Malls (c) Residential Complexes

Fig. 3. Depiction of the extent of WiFi interference in different dynamic urban deployment environments.

"good" channels. Identifying such channels can incur substantial overhead if channel behavior does not remain stable for a sufficiently long duration. It still remains an open problem even though many methods have been proposed over the years [2]. TSCH belongs to an alternate category of MAC protocols that allows time-synchronized channel hopping and promises extremely reliable communication [9], [12]. However, it comes at the cost of maintaining transmission schedules, limiting the flexibility of joining/leaving of network nodes. At the other end of the spectrum lies LWB [15], which aims at eliminating the need for a MAC by exploiting synchronous transmissions [16] and achieves extremely low latency.

A slightly different approach to handling interference is taken by a set of works like BuzzBuzz [30] where sufficient redundancy is added in the form of multiple headers or sophisticated error correction codes to the entire payload. This helps in recovering packet corrupted by interference. For 802.11 networks, Maranello et al. [18] presented a novel partial packet recovery approach where CRC is applied on blocks of the payload instead of the entire payload.

Besides the conventional *interference-avoidance* paradigm, the *cooperative cross-technology interference mitigation* (CIM) paradigm offers an interesting alternate approach to perform technology-independent interference cancelation (TIIC) [17], [23]. With the use of signal processing techniques and MIMO, it is possible for multiple heterogeneous networks to cooperatively cancel/mitigate the interference to each other. Unfortunately, it requires specialized antennas and *"there is a lack of study on both the feasibility and theoretical performance limits of CIM"* [23].

Compared to the works mentioned, we position Oppcast as shown in Figure 1. It carefully exploits spatial and channel diversity to completely eliminate expensive channel sampling while still maintaining reliable communication in highly dynamic urban environments with similar or lower energy consumption and shorter latency.

III. MOTIVATION

Cross-technology interference (CTI) is a major deterrent for widespread sensor network deployment in urban environments. ZigBee networks share the 2.4 GHz ISM band with widely used wireless technologies like WiFi, Bluetooth, and common household appliances like cordless phones. This exposes ZigBee communication to constant interference predominantly from WiFi-enabled devices like laptops and smartphones. Researchers often assume that there are at least 4 usable ZigBee channels (15, 20, 25, and 26) that are orthogonal to the most commonly used WiFi channels (1, 6, and 11) as shown in Figure 2 [1], [8], [30]. While this still holds true in laboratory and other controlled settings, our measurements show that this is not true in many urban environments. We find that, in an urban environment, one faces highly dynamic wireless channels. Depending on where nodes are deployed, one can expect different interference patterns as described in the rest of this section.

A. Interference Pattern

Depending on whether the WiFi deployment is planned or unplanned, one can find two representative wireless environments, (a) planned CTI, and (b) unplanned CTI.

1) Planned CTI: They are found in places where there is a single administrative authority handling WiFi Access Point (APs) deployments like academic institutes, libraries, corporate offices, industries, etc. The deployed WiFi APs follow what we call the X-Y-Z rule of channel assignment, which implies the selection of at most three orthogonal WiFi channels among the ones shown in Figure 2. Figure 3(a) illustrates a WiFi Analyzer's[1] output inside two different academic institutions. Clearly ZigBee channels 15, 20, 25, and 26, which lie orthogonal to WiFi channels 1, 6, and 11, are mostly unoccupied. Most of the state-of-the-art protocols are optimized to perform on these available channels and evaluated on testbeds deployed in educational institutions like Indriya [6], which exhibit such a planned CTI. It is worth mentioning

[1]WiFi Analyzer, http://a.farproc.com/wifi-analyzer

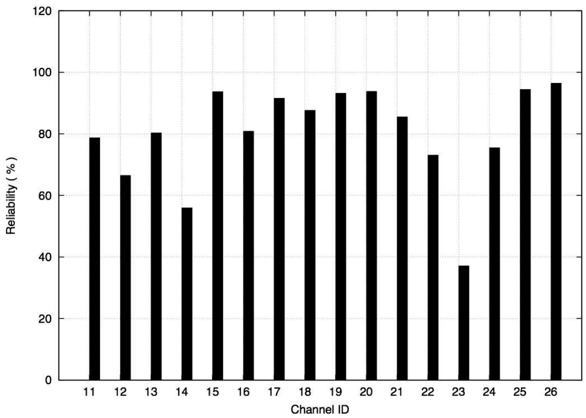


Fig. 4. Performance evaluation of single channel data collection protocols (RPL/ContikiMAC) on different ZigBee channels with varying CTI levels.

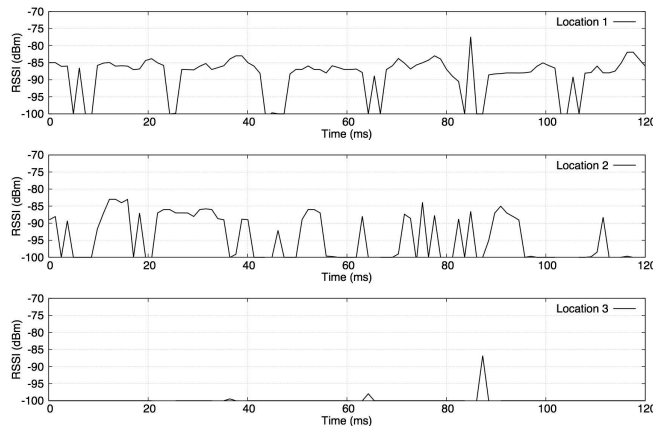


Fig. 5. Interference due to WiFi is confined to the range of the deployed WiFi AP.

that even within these planned deployments, random users sometimes might enable their own WiFi hotspots, and thereby cause transient interference even to those initially available ZigBee channels.

2) Unplanned CTI: In places like residential complexes, shopping malls, cafeterias, etc., where there is a lack of a centralized control, WiFi channel usages are of the kinds illustrated in Figure 3(b) and 3(c). In such environments, finding a CTI-free ZigBee channel is quite unlikely. In this WiFi Jungle, the reliability of single channel sensor network protocols is likely to suffer. Figure 4 provides an example of the impact CTI has on one such single channel data collection protocol (RPL/ContikiMAC) when executed on Indriya for an hour each on all the 16 ZigBee channels. Each node is set to generate data once every 4 minutes and reliability is computed as the number of packets received over the sum of the total number of packets transmitted by each node. One can observe how the reliability fluctuates from as low as 38% on channel 23 to as high as 98% on channel 26. A detailed performance evaluation of RPL under different wireless interference levels is presented in [19], where a reliability as low as 10% is reported under the influence of strong interference. With the ongoing rapid "IoTification" leading to almost every house or shop deploying its own WiFi AP, it is safe to assume that CTI would worsen further in the future, leading to an even more congested 2.4 GHz ISM band. In one of our WiFi signal scans inside a residential complex spanning over 24 hours, we observed WiFi beacons from as many as 39 APs interfering with a ZigBee channel. This highlights the severity of CTI due to WiFi transmissions in today's time. With many countries allowing unrestricted use of all the WiFi channels[2], the problem is quite challenging and important.

Takeaway: Deployments in unplanned urban environments cannot rely on channel estimation since channels are highly dynamic. Finding a channel guaranteed to be CTI-free over a long period is non-trivial and may not even be possible in some cases. Predicting the onset of CTI is quite challenging as well. Protocols, therefore, should provide robustness to CTI without the need of performing channel estimation in advance.

B. Localized Interference

Interference on a channel is not only highly dynamic but also confined to the neighborhood of the source generating it. Each CTI source has a specific range, be it WiFi, Bluetooth, a microwave or a cordless phone. If finding a channel that remains free from interference for one receiver is challenging, finding a channel that is free from interference for all nodes throughout the network in an unplanned WiFi deployment will be even more difficult.

To illustrate this localized behavior, we scan ZigBee channel 26 at a rate of 8KHz using the cc2420 radio on a TelosB device at three different locations outside a shopping mall. Figure 5 shows the interference pattern on these nodes at three different locations, two of which are spatially close to each other (Location 1 and Location 2). It can be seen that even on ZigBee channel 26, which is expected to be orthogonal to WiFi, there is enormous WiFi traffic in Location 1 (top) and 2 (middle). Meanwhile, Location 3 (bottom) being few hops away, observes a significantly cleaner channel.

Takeaway: The assumption that there is a network-wide non-interfered ZigBee channel in an unplanned WiFi deployment does not hold in many cases. Allowing each node to blacklist an interfered channel in its vicinity would introduce channel coordination issues since different nodes might blacklist a different set of channels potentially causing network partitioning. Therefore, one needs a multi-channel protocol with efficient channel selection and coordination.

C. Communication Opportunities

Most often, sensor network deployments are relatively dense with each node having many neighbors. Figure 6 illustrates the network density of three popular wireless sensor network testbeds on one of the least interfered channels. Even on the sparsest network Flocklab [31], we can find 5 neighbors

[2]IEEE Standard 802.11-2007, Table 18-9.

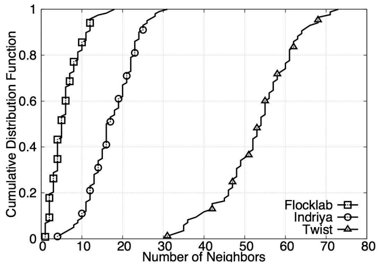


Fig. 6. CDF of the neighbor count on testbeds deployed inside academic institutions.

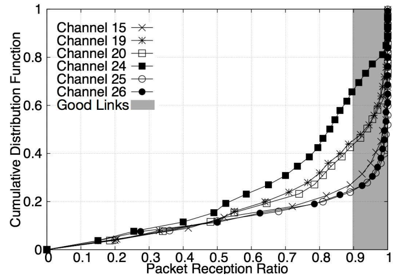


Fig. 7. The number of "good" links reduces significantly due to CTI on certain channels.

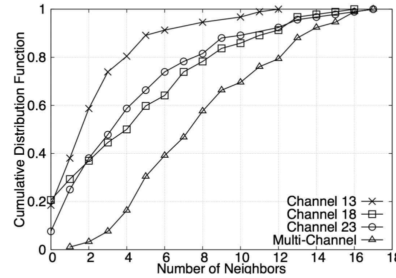


Fig. 8. Exploiting multiple channels increases the spatial opportunities dramatically.

on average. Indriya and Twist have about 15 and more than 50 neighbors on average, respectively. This observation has motivated researchers to exploit opportunistic routing where data is routed over a DODAG (Direction Oriented Directed Acyclic Graph with each node having multiple parents) instead of a TREE topology (each node has a single parent) [13], [28]. However, from our measurements, as shown in Figure 7, we observe that CTI reduces the number of opportunities, or the percentage of neighbors having "good" connectivity ($PRR > 0.9$) significantly from around 75% on channel 26 to around 35% on channel 24. This negatively affects the performance of the protocols exploiting spatial opportunities. We performed PRR-based link quality estimation on Indriya on channels 13, 17 and 23 (non-orthogonal to WiFi) and obtained the number of neighbors each node has with $PRR > 0.9$. Later we computed the number of neighbors each node would have if one was allowed to use all of the three interfered channels together. Figure 8 illustrates that, on average, we can find up to a 4 times increase in the number of opportunities, with the number of "good" quality neighbors getting boosted from around 2 if only channel 13 is used, to around 8 if the union of all neighbors over the three interfered channels can be considered.

Takeaway: The number of "good" quality neighbors reduces dramatically due to CTI. Exploiting spatial diversity over multiple channels could potentially boost a protocol's performance in the presence of dynamic CTI, which is typical in an urban environment, due to increased opportunities.

D. Summary

The above observations highlight three important considerations while designing protocols for a highly dynamic urban environment:

- Due to the prevalence of unplanned CTI, the designed protocols should not rely on static single-channel allocation and/or expensive channel estimation.
- Finding a network-wide "good" channel is improbable as every CTI source has a defined interference range.
- If the deployment is dense, one might want to exploit spatial diversity over multiple channels as it provides significantly higher communication opportunities than over a single channel, which is prone to interference.

To this end we design Oppcast, a data collection protocol that carefully exploits spatial and channel diversity to provide:

- reduced energy consumption because of the elimination of channel estimation
- improved robustness to CTI because of multi-channel communication
- reduced latency because of the usage of opportunistic routing

IV. Design

In this section, we present Oppcast, a multi-channel probe-based receiver-initiated opportunistic routing protocol that uses **opp**ortunistic uni**cast** transmissions to improve reliability with minimal duplicate transmissions in the presence of CTI that causes packet corruptions.

A. Oppcast in a Nutshell

Like any receiver-initiated protocol, each Oppcast node periodically broadcasts a PROBE, if the medium is idle, to announce that it is awake and ready for a packet reception. Each PROBE contains the hop-count (how far it is from the sink, which is at hop-count 0). After probing, the node keeps its radio on for a short while (7.8125ms in our implementation) anticipating a response. Any node with some DATA to transmit waits with its radio ON, listening for a PROBE request. Upon a successful PROBE reception, the receiver opportunistically transmits its DATA to the probing node to get relayed to the SINK as long as the probing node has a lower hop-count. Successful DATA reception gets acknowledged by a subsequent ACK, which concludes the current transaction (PROBE-DATA-ACK), after which both the nodes restart the periodic probing schedule. This opportunistic routing over receiver-initiated MAC assists Oppcast in achieving low end-to-end latency.

As an enhancement to the receiver-initiated MAC, Oppcast incorporates channel diversity to counter CTI by attempting the above transaction over multiple channels. However, unlike typical multi-channel protocols, Oppcast does not perform channel estimation to identify "good" channels. Instead, it only operates on a small set of ZigBee channels that are far apart from each other. The channels are far apart in the sense that a single WiFi channel should only interfere with no more than

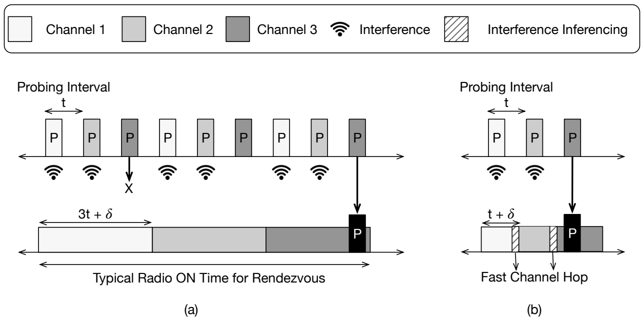


Fig. 9. (a) Naive channel hopping scheme allowing senders to idle listen for thrice the probing interval even if the channel is heavily interfered. (b) Fast channel hopping strategy in Oppcast reduces energy wastage caused by idle listening on an interfered channel.

one of these channels. In the case of the 2.4GHz band, only three channels are therefore utilized.

A key characteristic in Oppcast is that unlike other multi-channel protocol like MiCMAC [1] where only the receivers perform quick channel hopping, both the sender and the receiver hop through all the channels quickly. The advantage of such an approach is that the sender can attempt transmissions on all three channels in a short time. Thus, as long as one of the channel is good, the chance of successful transmission is high.

Oppcast is unique in that the presence of a sufficient number of neighbors in a dense network deployment is exploited both in the use of multiple channels and opportunistic routing so that the likelihood of the sender and a receiver to meet on a good channel is increased.

The key components in Oppcast are

- channel selection and coordination to exploit channel diversity;
- use of opportunistic routing to exploit spatial diversity.

In the rest of this section, we discuss the different components in greater detail.

B. Channel Diversity

A bad channel selection is among the primary reasons for poor performance. We have seen how exploiting channel diversity gives an enormous performance boost [13], [28]. However, the efficient usage of multiple channels requires (1) A smart channel selection strategy and (2) An efficient channel coordination scheme.

1) Channel Selection: Identifying which channels to use to mitigate the impact of CTI is a well-studied problem. The WSN research community has proposed numerous Link Quality Estimators (LQEs), which help to choose the best channel: Hardware-based LQEs (RSSI, LQI or SNR), Software-based LQEs, which further include PRR-based LQEs (PRR, KLE, etc.), RNP-based LQEs (ETX, Four-bit, etc.) and Score-based LQEs (WRE, MetricMap, etc.) [2]. However, each of these techniques has an associated overhead that is proportional to the number of channels to scan and the size of the network. Keeping up with updated channel information or performing accurate channel prediction incurs non-trivial overhead.

Oppcast eliminates the need for channel estimation. Instead, it picks 3 ZigBee channels that are far apart so that all the three channels may interfere only if transmissions are performed over three orthogonal WiFi channels simultaneously.

More formally, assume Z_i is the set of ZigBee channels overlapping with WiFi channel i where $i \in \{1, 2, \cdots, 14\}$. Then the set of Oppcast channels $\theta = (c_1, c_2, c_3) \in Z_{cp}$, where Z_{cp} is the Cartesian product of 3-tuples Z_i, Z_j and Z_k and i, j and k are three orthogonal WiFi channels. For instance, considering that most WiFi (802.11 g/n) channels have a bandwidth of 20MHz and are separated by 5MHz, we can have at most 4 combinations of 3 orthogonal WiFi channels: (1,6,11), (2,7,12), (3,8,13), and (4,9,14) as shown in Figure 2. As an example, for the most popular WiFi channel assignment (1,6,11) as shown in Figure 3(a) we have $i = 1$, $j = 6$ and $k = 11$, which gives $Z_i = 11, 12, 13, 14, 15$, $Z_j = 16, 17, 18, 19, 20$ and $Z_k = 21, 22, 23, 24, 25, 26$. Thus we can choose any $(c_1, c_2, c_3) \in Z_i \times Z_j \times Z_k$, giving Oppcast 150 channel combinations to choose from.

2) Channel Coordination: Channel coordination is an essential part of a multi-channel communication solution. An efficient channel coordination scheme should allow sender and receiver to rendezvous on the same channel quickly. Oppcast runs a simple yet efficient channel coordination scheme where the sender and the receiver uses a round robin schedule to select its channel. To ensure that a sender receives at least one probe request in its wake-up interval, a naive solution is to wait on a channel for a duration of 3 probe intervals and then hop to the next channel and wait again for 3 probe intervals as shown in Figure 9(a). However, this scheme is inefficient since if the selected channel suffers interference, it is likely that further transmissions will be unsuccessful as well. Hence, attempting to wait for more probes on the same channel may not help much and would lead to energy wastage besides incurring additional latency.

Fast Channel Hop (FCH) In Oppcast, a sender, instead of continuously listening for 3 probing intervals before hopping to the next channel, performs a Fast Channel Hop (FCH). FCH enables a node to switch to the next channel if it fails to receive any probe within a probe interval. The idea is based on two assumptions: (1) A node has a fair number of neighbors; and (2) neighbors send probe uniformly on all candidate channels. Based on these assumptions, as long as one of the 3 channels the node listens on is good, it is likely that it will hear from at least one of its neighbors.

To further improve performance, Oppcast infers the channel quality passively, without any overhead, based on probe reception rate and learns which channel to start the round robin schedule from in the subsequent rounds to achieve an earlier rendezvous. This avoids energy wastage and reduces packet delivery latency.

While FCH reduces energy wastage, it is prone to the channel-chasing problem. Here is an illustrative example. A receiver sends probes on channel 12, 18 and 24 in a round-robin channel. A sender tries to capture a probe following the same order of channel 12, 18 and 24. When the two communicating nodes start at different channels and hop at the same rate, the two nodes will not encounter each other. In order to avoid this problem, Oppcast lets the sender and the receiver select channels in a round robin fashion, but in reverse orders. For example, if the receiver hops on channels 12, 18 and 24, the sender hops in reverse order on channels 24, 18 and 12. In this way, the sender and receiver are guaranteed to meet with high likelihood within 3 probe periods.

C. Spatial Diversity

Besides channel diversity, Oppcast exploits two forms of spatial diversity each with a different goal as explained below.

1) Node Diversity: The cost one has to pay for exploiting channel diversity is the increase in latency, which is mainly caused due to channel coordination. For every additional channel, the transmitter and corresponding receiver have to hop onto a common channel to rendezvous. This inevitably increases the encounter time. Oppcast reduces this inefficiency by exploiting node diversity through opportunistic unicast transmissions with the following considerations.

Duplication Control: Given the gains of opportunistic routing, many wireless sensor network data collection protocols have exploited it [13], [28]. Instead of using anycast to transmit data packets repeatedly until getting acknowledged like traditional opportunistic routing protocols, Oppcast waits for probe requests from potential forwarders who are closer to the sink and performs **opp**ortunistic uni**cast** or an "**oppcast**" transmission. Compared with anycast, Oppcast has two benefits: (1) It significantly lowers the duplication of the traffic in a network, which indirectly reduces energy consumption; and (2) Like any receiver-initiated MAC (A-MAC, Ri-MAC, etc.), it minimizes the time the sender and its intended receiver occupy the wireless medium by not using ContikiMAC style strobed data transmission.

Routing Metric: To understand if a probe request is an opportunity and provides sufficient routing progress towards the destination, Oppcast uses a simple hop-based routing metric instead of traditional LQEs like ETX [5] that fail to perform well for highly dynamic CTI of an urban environment. A node computes its routing metric as shown in Algorithm 1. Whenever it receives a probe containing the sender's hop-count information, it adjusts its own hop-count to shift closer to the probing node. Since many neighboring nodes probe periodically, the hop-count eventually converges. Moreover, since a node updates its routing metric on reception of any probe from any of its neighbor, it lets the node rapidly react to routing metric changes in a dynamic urban environment because probes are much more frequent than data packets. Depending on how hostile the urban environment is, one can choose a proper value for α of the exponentially weighted moving average (EWMA) filter used in the algorithm, which defines how dynamically the routing metric should get updated. In all our experiments, we used $\alpha = 0.6$.

Data: My Node ID: M, Neighbor Node ID: N
$M.hop \leftarrow \infty$;
while $true$ **do**
  foreach *probe from neighbor* N **do**
    if $M.hop < N.hop$ **then**
      $M.hop = \alpha * M.hop + (1 - \alpha) * (N.hop - 1)$;
    else
      $M.hop = \alpha * M.hop + (1 - \alpha) * (N.hop + 1)$;
  end
end

Algorithm 1: Hop-based routing metric estimation without link estimates

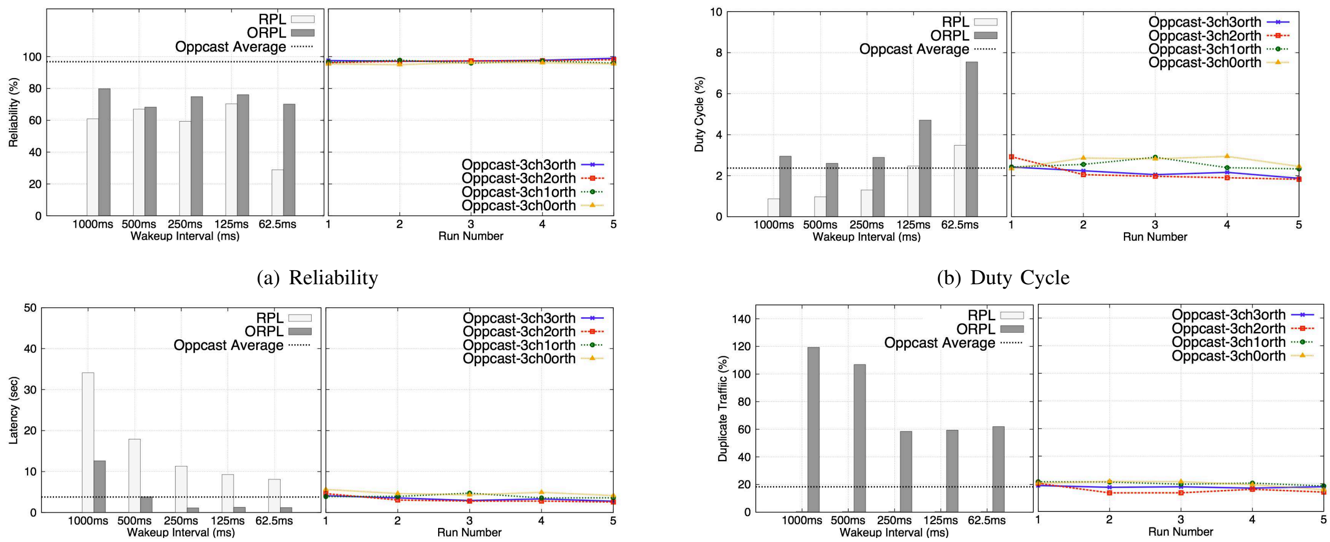
2) Path Diversity: Adding redundancy in any form helps in making the protocol more robust [18], [30]. Since Oppcast routes data over a DODAG instead of a TREE topology, it has multiple alternate paths to exploit on its way towards the destination. Because CTI is mostly confined to a specific zone of the entire network (e.g. around the range of WiFi AP), Oppcast can exploit path diversity to achieve further robustness. Each node, depending on how interfered the channels are, can choose to replicate the data over N different paths by responding to N unique probe requests. From our experiments, transmission on a single path is usually enough. However, allowing data duplication over additional routes allows even higher end-to-end data reliability to be achieved, but with a higher overall energy consumption.

V. Evaluation

In this section, we first evaluate Oppcast against the state-of-the-art data collection protocols, which separately exploit either spatial (ORPL [13]) or channel diversities (MiCMAC [1]). Oppcast consistently outperforms others despite the presence of CTI in a large-scale testbed and different urban environments. Later, we show how adding redundancy to Oppcast through multi-path routing can boost the reliability to near perfection.

A. Experimental Details

We have implemented Oppcast on Contiki [10] using its Rime stack to provide light-weight communication between the nodes. For all the experiments, the maximum transmission power for the cc2420 chip on the TelosB is used. Each node generates traffic at an inter-packet interval of 4 minutes to be consistent with the results reported by other protocols in the respective papers. A probing interval of 1 second and a probe length of 8 bytes are used to request data from neighboring nodes. A probe contains the hop-based routing metric to exploit opportunistic communication. To enable performance analysis at the sink node for non-testbed experiments, each packet contains network statistics like the sequence number of the current packet and the average duty cycle until the time the packet was generated. Each relay node marks the packet with its node ID to identify the path that the packet takes towards the sink node.



(a) Reliability

(b) Duty Cycle

(c) Latency

(d) Traffic Duplication

Fig. 10. Comparison of Oppcast against RPL/ContikiMAC and ORPL/ContikiMAC running on Indriya for different wake-up intervals on channel 24.

B. Baseline Protocols

We consider three different types of protocols having respective Contiki implementations to evaluate against:

RPL/ContikiMAC: This is a tree-based data collection protocol using ETX [5] as the routing metric with traffic carried in UDP datagrams over 6LowPAN towards the sink node.

ORPL/ContiMAC: This is an extension to RPL, which exploits node diversity by transmitting data towards the sink node over a DODAG using EDC [28] as the routing metric. Duty-cycled anycast transmission is used to achieve opportunistic routing.

RPL/MiCMAC: This is RPL again, but running over a multi-channel MAC to exploit channel diversity by supporting pseudo-random channel hopping with phase and channel lock mechanisms. It uses ETX as the routing metric.

C. Evaluation Metrics

We evaluate performance using the following three standard metrics.

Reliability: Reliability is the end-to-end packet reception rate, averaged over all the nodes. It indicates how robust a protocol performs.

Duty Cycle: Duty cycle is the percentage of the total time the radio is kept on. It is estimated using Energest [11], an energy estimation module on Contiki, and averaged over all the nodes.

Latency: Latency is the time taken from the packet injection to its reception by the sink node. We evaluate latency in testbed experiments. Due to the lack of availability of the exact dispatch time of packets in urban deployments, we skip latency measurements in those experiments.

To allow the network to stabilize, the metrics are computed around 15 minutes after deployment and with at least an hour of protocol running period.

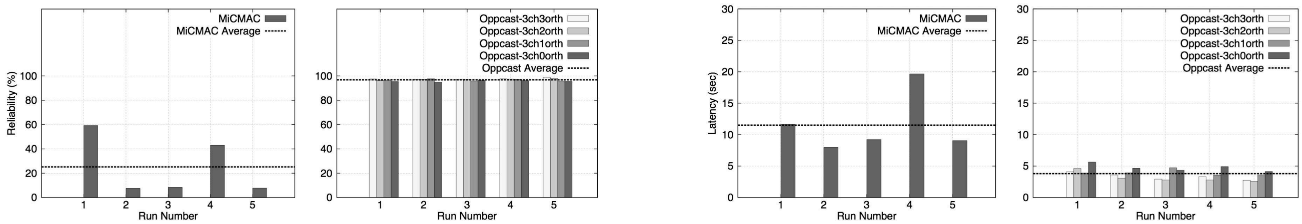
D. Testbed Specific Information

Indriya [6] is used for preliminary evaluation. There are 96 nodes in the network (at the time of experiment) deployed over three floors. The WiFi APs are set to operate on WiFi channels 1, 6, and 11, giving us at least 4 ZigBee channels (15, 20, 25, and 26) free to use. This can be confirmed from Figure 4 where RPL shows the best reliability on ZigBee channels 15, 20, 25 and 26. Due to students setting up their own private WiFi hotspots, sometimes some of those channels may get interfered too. Node 67, which is deployed at level two in a department foyer, is selected as the sink node, since it should likely suffer more serious WiFi interference. Moreover, unlike any typical evaluation strategy where non-interfered channels are considered, we only use those ZigBee channels that suffer interference on Indriya in all our experiments. Besides the testbed, we also extensively evaluate the performance of different protocols on actual sensor network deployments in different urban environments.

E. A Case for Channel Diversity

In this section, to evaluate the effectiveness of channel diversity, we compare existing single channel solutions with Oppcast.

ORW [28] and ORPL [13] are similar single-channel protocols that exploit spatial diversity through opportunistic routing using duty-cycled anycast transmissions. However, like any other single channel protocol, the performance degrades if an interfered channel is chosen. To emulate an urban environment on Indriya, we ran RPL and ORPL over ContikiMAC on ZigBee channel 24. From Figure 10 one can see that while ORPL outperforms RPL in terms of reliability and latency, the absolute ORPL reliability is observed to be as low as 68.25% at ORPL's default wake-up interval of $500ms$. One way of improving the reliability is to wake up more often



(a) Reliability

(b) Latency

Fig. 11. Comparison between Oppcast and RPL/MiCMAC when run on different combination of channels. Default wake-up interval of $125ms$ is set for MiCMAC.

to give more opportunities for receiving the packets. Due to the selection of an interfered channel, no consistent reliability improvement is observed even when the wake-up interval is reduced down to $62.5ms$. Moreover, because of ORPL's use of anycast transmission, an enormous duplicate traffic is observed, which reaches as high as 120% of the original traffic. This traffic surge causes a rise in overall energy consumption, as can be seen from Figure 10(b).

In comparison to RPL and the improved ORPL over ContikiMAC, Oppcast performs consistently better with near perfect reliability by exploiting channel diversity in combination with spatial diversity. In the legends of Figure 10, Oppcast-3ch**X**orth signifies that **X** out of 3 randomly chosen Oppcast channels are orthogonal to the WiFi Channels 1, 6 and 11. This means Oppcast-3ch3orth refers to the least interfered scenario, where all the three channels are chosen randomly from CTI-free ZigBee channel 15, 20, 25 and 26 while Oppcast-3ch0orth corresponds to the most interfered channel combination. Data collection using each of the Oppcast settings is repeated 5 times and the results are shown. It can be seen that interference has minimal impact on the reliability of Oppcast with the mean packet reception reliability over all the 20 runs of an hour each being close to 97%. This spans over 20 hours of experimentation during weekdays where the interference is expected to be higher. Moreover, Oppcast also achieves 2.7 times shorter latency, 6 times lower duplicate traffic and 1.2 times lower energy consumption in comparison to ORPL.

F. A Case for Spatial Diversity

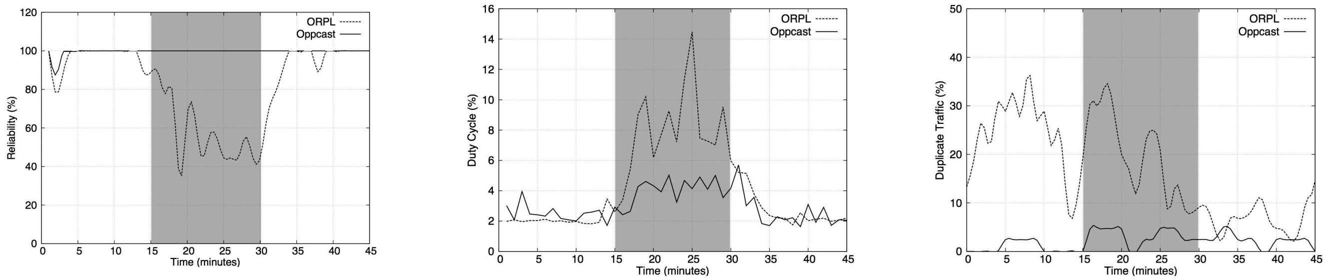
Is channel diversity enough to obtain the benefits highlighted in the previous section? RPL/MiCMAC [1] is a representative example of exploiting channel diversity to survive interference. Recall from Figure 8 that without spatial diversity, a wrongly chosen set of interfered channels will cause a drop in reliability because of a significant reduction in the number of neighbors having "good" links. Moreover, finding the optimal channel set as we know is non-trivial. To test the performance of MiCMAC in the presence of CTI due to WiFi, we ran it 5 times on 4 randomly selected ZigBee channels (Run 1 includes channels supposedly orthogonal to WiFi, while the rest uses 4 non-orthogonal ZigBee channels) and compared against Oppcast running over different combinations of orthogonal and non-orthogonal ZigBee channels as explained in the previous section. The default wake-up interval of MiCMAC ($125ms$) is chosen while Oppcast is made to probe every 1 second. The results are shown in Figure 11(a). One can see that while MiCMAC succumbs to severe interference due to lack of enough opportunities in an interfered channel by not exploiting spatial diversity, Oppcast consistently ensures high reliability even in the Oppcast-3ch0orth setting. We expected Run 1 of MiCMAC, which was on 4 ZigBee channels orthogonal to WiFi to perform as reported in [1]. However, perhaps due to a change in the WiFi deployment around Indriya, we observed a reliability of 59.2%. Moreover, it can be seen from Figure 11(b) that while the latency for MiCMAC reaches as long as $19.66sec$ (average of $11.51sec$ over the 5 runs), Oppcast curbs it to an average value of $3.79sec$ over all the 20 runs. This makes Oppcast close to 3 times faster than non-opportunistic protocols exploiting only channel diversity. We would like to point out that if one is certain of the availability of 4 "good" channels, a multi-channel protocol like MiCMAC should be able to perform on par with Oppcast in terms of reliability but would still suffer longer latency due to the use of non-opportunistic routing.

G. Channel Utilization

Making a protocol multi-channel does not necessarily mean that all the channels are being optimally utilized. We have seen in Figure 7 that in each of the channels we have at least a few "good" links ready to be exploited. Thus, each channel should be used more or less proportionally. To check if Oppcast actually utilizes each of its three channels equally, we compute the percentage of the total packets that were communicated over each of the three channels. From Figure 13 we see that roughly an equal number of communications happened over each of the Oppcast channels based on the results of all the 20 runs of Oppcast on Indriya. This has two advantages:
(1) It minimizes the self-interference (ZigBee packet collisions), by allowing the data to be distributed on all the channels uniformly, and
(2) It minimizes the communication delay by utilizing each of the channels equally to communicate and not having to wait for a specific good channel for nodes to rendezvous.

H. Impact of Exploiting Path Diversity

Because of the localized nature of CTI as illustrated in Figure 5, even though some of the paths towards the des-



(a) Reliability

(b) Duty Cycle

(c) Traffic Duplication

Fig. 12. Stress test: Oppcast is resilient to transient CTI, maintains high reliability without much increase in duty cycle, and has consistently lower redundant duplicate traffic.

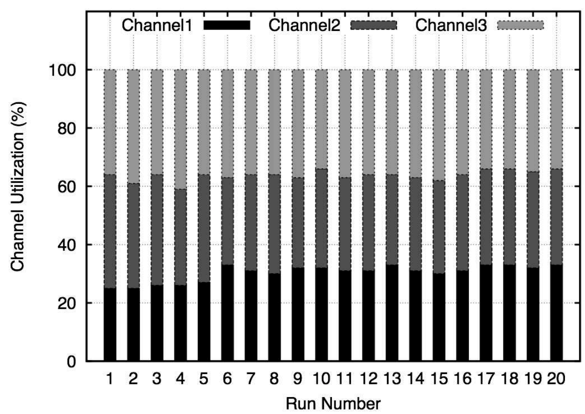


Fig. 13. Oppcast uses each of its channel almost equally for data communication, thus minimizing self-interference and maximizing channel utilization

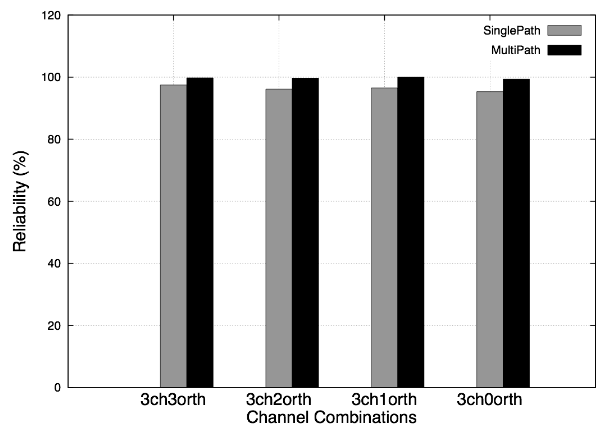


Fig. 14. Adding redundancy through exploiting path diversity boosts the reliability further for Oppcast

tination get disconnected, adding redundancy through path diversity can ensure end-to-end connectivity. Oppcast already provides sufficient robustness by exploiting just channel and node diversity. However, exploiting path diversity by forwarding additional duplicate packets over multiple unique paths improves it further. Figure 14 illustrates this point where we run back-to-back experiments on Indriya with and without path diversity for 4 different sets of Oppcast channels. Each run lasts for an hour as usual. It can be seen that the reliability gets boosted just by exploiting one more additional path towards the Sink. This, of course, comes at the cost of increased energy consumption since now double the original traffic needs to be supported by the network. Use of simple network coding like in [8] can help reduce the energy consumption by requiring much lower redundancy.

I. Resilience to CTI

One of the main Oppcast design goals is to survive severe CTI, which is a common requirement in urban deployments. Since in an urban setting we have no control over the CTI, to stress test the performance of Oppcast and ORPL, we design the following experiment. Oppcast and ORPL are made to run on a 20 node testbed (10 each) in parallel with 2 nodes paired together to ensure the same topology and with ZigBee channels 11, 17 and 23 selected for Oppcast and channel 22 selected for ORPL. In the middle of the experiment, we introduce WiFi interference by continuously flooding UDP packets for 15 minutes on WiFi channel 11 which interferes with the two adjacent ZigBee channels 22 and 23. The results are presented in Figure 12. We can see that in the first 15 and the last 15 minutes of no interference, Oppcast and ORPL have similar performance in terms of both reliability and duty cycle. When interference is introduced at the 15^{th} minute, while Oppcast maintains the reliability at 100%, it drops to around 38% for ORPL. Even in terms of energy, Oppcast consumes up to 2 times lower energy due to consistently lower duplicate traffic and the use of multi-channels. For ORPL, high duplicate traffic is an expected behavior in a dense deployment and on a "good" channel as seen in [13]. However, a drop in duplicate traffic is seen towards the end. This is possibly due to the lowering of the network density due to physical obstruction caused by cars in the carpark.

J. Evaluation in Urban Environments

From all the above experiments on various interfered channels of Indriya, ORPL offered better reliability than RPL/MiCMAC (illustrated in Figure 10(a) and Figure 11(a)). Therefore, we select ORPL as the baseline protocol for evaluating Oppcast on the 20 node testbed deployed in different urban environments with 2 nodes paired together running

TABLE I
COMPARISON BETWEEN ORPL AND OPPCAST IN DIFFERENT URBAN ENVIRONMENTS

Deployment Location	Protocol	Sensor Position	Run	Duration	Reliability	Min Reliability	Duty Cycle	Max Duty Cycle	Duplicate Traffic	Max Duplicate Traffic
Carpark	ORPL/ContikiMAC	Floor	1	45 Hours	86.09	86.09	NA	NA	2.85	10.60
		Floor	2	46 Hours	97.42		NA		10.60	
		Ceiling	3	90 Hours	93.07		NA		7.69	
	Oppcast	Floor	1	45 Hours	98.55	98.55	NA	NA	2.55	2.55 (4.2x)
		Floor	2	46 Hours	99.80		NA		1.51	
		Ceiling	3	90 Hours	99.67		NA		1.28	
Residential Complex	ORPL/ContikiMAC	Random	1	9 Hours	99.21	92.45	4.65	4.23	26.54	26.54
			2	38 Hours	92.45		3.95		17.3	
			3	24 Hours	99.79		4.23		4.23	
	Oppcast	Random	1	9 Hours	98.4	98.40	3.47	3.47 (1.2x)	5.54	5.54 (4.8x)
			2	38 Hours	99.14		3.04		4.73	
			3	24 Hours	99.62		2.3		2.91	
Shopping Mall	ORPL/ContikiMAC	Inside	1	1 Hour	99.87	99.53	9.39	9.39	47.21	47.21
		Outside	2	1 Hour	99.53		7.42		12.79	
	Oppcast	Inside	1	1 Hour	100	100	3.96	3.96 (2.4x)	10.61	10.61 (4.5x)
		Outside	2	1 Hour	100		2.82		8.64	
Cafeteria	ORPL/ContikiMAC	Random	1	1 Hour	99.32	99.32	4	4	17.03	17.03
	Oppcast	Random	1	1 Hour	100	100	2.6	2.6 (1.5x)	4.05	4.05 (4.2x)

ORPL and Oppcast each to ensure the same topology of up to 3 hops. Due to memory constraints on the TelosB devices, we connect the sink node to a Raspberry Pi[3] to store test logs for an extended duration. Since the network is much smaller in comparison to Indriya, we reduce the inter-packet interval to 1 minute. The rest of the protocol parameters are left unchanged. We run the experiments in 4 representative places of urban environments, namely:

- A carpark in the basement of a building.
- A 4 bedroom apartment in a residential complex.
- Inside and outside a five-storey shopping mall.
- A cafeteria with open/closed roof sitting arrangements.

Table I summarizes the results of 255 hours of parallel execution of both ORPL and Oppcast over randomly selected channels. No prior channel estimation was performed and random sensor locations are selected for different runs. We observe reliability as low as 86.09% for ORPL with Oppcast maintaining a minimum of 98.55% in the Carpark. Oppcast outperforms ORPL in the other environments as well with up to 2.4 times lower energy consumption. This is due to the repeated transmission attempts made by ORPL on the same interfered channel. Oppcast on the other hand gracefully hops to the non-interfered channel using FCH strategy. Moreover, Oppcast achieves this high reliability with up to 4.8 times lower duplicate traffic. It shows that Oppcast is not only highly robust in extremely dynamic urban environments, but also energy efficient.

VI. DISCUSSION

With Oppcast, we have shown how robust data collection is possible even in highly dynamic urban environments. Here we discuss some limitations of our approach and possible improvements.

Channel Availability: Although we have shown how difficult it is to find CTI-free channel in an urban environment, if however, one has to deploy sensors where at least one channel is guaranteed to be available free from CTI (e.g. Planned

[3]Raspberry Pi, https://www.raspberrypi.org/

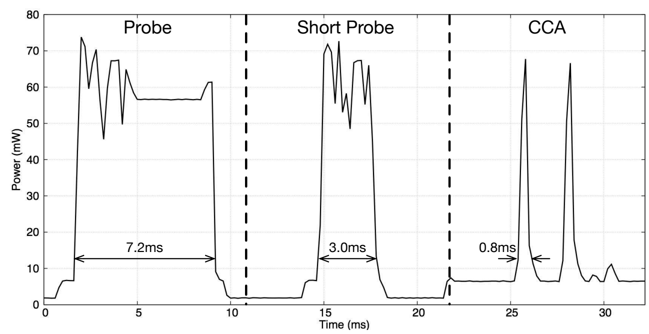


Fig. 15. Energy consumption comparison for probing, short probing and Clear Channel Assessment (CCA).

WiFi, Wilderness, etc.), the baseline protocols should be able provide similar reliability and with lower energy consumption as shown in respective papers. Oppcast is designed primarily for hostile dynamic urban environments.

Power-Efficient Wake-up: Baseline protocols use Low Power Listening (LPL) based MAC that incorporates Clear Channel Assessment (CCA). Oppcast, on the other hand, is inspired from Ri-MAC [35] and A-MAC [14] and uses Low Power Probing (LPP) based MAC. This design choice of LPP over LPL is to enable opportunistic routing with significantly reduced duplicate traffic. Moreover, in an interfered environment, CCA becomes less efficient due to a very high false wake-up rate [40].

However, on the downside, as seen in Figure 15, Probes (7.2ms radio ON) are roughly 4.5 times more expensive than CCA. One way of making Oppcast more energy efficient is to employ HACK (Hardware automatic acknowledgment in 802.15.4 standard) as done in A-MAC. This allows a node to sleep immediately after probing if it doesn't detect a HACK, which implies that none of its neighbors have any data to send. On enabling HACK, we can make Probes shorter (3.0ms radio ON) and 2.4 times more energy efficient than Probes without HACK as seen in Figure 15.

From our experience in urban environments, nodes quite often fail to detect HACK and choose to turn the radio off immediately even if their neighbors have data to send. This leads to packets getting dropped repeatedly. To ensure reliability, we chose not to use HACK in our current implementation. However, we would like to investigate it further given the energy savings it promises.

VII. Conclusion

We present Oppcast, a multi-channel probe-based receiver-initiated opportunistic routing protocol that uses opportunistic unicast transmissions to improve reliability with reduced duplicate transmissions in the presence of highly dynamic CTI, which characterizes an urban environment. Through extensive evaluation in both a large-scale testbed (Academic Institution) and real urban settings (Carpark, Residential Complex, Shopping Mall and Cafeteria) we illustrate the capability of Oppcast to maintain consistently high reliability of more than 98.55% in all the challenging deployments with up to 2.4 times reduced energy consumption and up to 4.8 times reduced duplicate traffic in comparison to the state-of-the-art data collection protocols, ORPL, RPL/ContikiMAC and RPL/MiCMAC.

VIII. Acknowledgement

We would like to thank the anonymous reviewers and our shepherd, Koen Langendoen, for their valuable comments and suggestions.

This work was supported in part by the Agency for Science, Technology and Research (A*STAR), Singapore, under SERC Grant 1224104049.

References

[1] B. Al Nahas, S. Duquennoy, V. Iyer, and T. Voigt. Low-power listening goes multi-channel. In *DCOSS*. IEEE, 2014.

[2] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

[3] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[4] J. Borms, K. Steenhaut, and B. Lemmens. Low-overhead dynamic multi-channel mac for wireless sensor networks. In *Wireless Sensor Networks*. Springer, 2010.

[5] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[6] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda. Indriya: A low-cost, 3d wireless sensor network testbed. In *TRIDENTCOM*. Springer, 2012.

[7] M. Doddavenkatappa, M. C. Chan, and B. Leong. Improving link quality by exploiting channel diversity in wireless sensor networks. In *RTSS*. IEEE, 2011.

[8] M. Doddavenkatappa, M. C. Chan, and B. Leong. Splash: Fast data dissemination with constructive interference in wireless sensor networks. In *NSDI*. USENIX, 2013.

[9] L. Doherty, W. Lindsay, and J. Simon. Channel-specific wireless sensor network path data. In *ICCCN*. IEEE, 2007.

[10] A. Dunkels, B. Gronvall, and T. Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *LCN*. IEEE, 2004.

[11] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *SenSys*. ACM, 2007.

[12] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne. Orchestra: Robust mesh networks through autonomously scheduled tsch. In *SenSys*. ACM, 2015.

[13] S. Duquennoy, O. Landsiedel, and T. Voigt. Let the tree bloom: scalable opportunistic routing with orpl. In *SenSys*. ACM, 2013.

[14] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *SenSys*. ACM, 2010.

[15] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. Low-power wireless bus. In *SenSys*. ACM, 2012.

[16] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *IPSN*. IEEE, 2011.

[17] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. *ACM SIGCOMM Computer Communication Review*, 41(4):170–181, 2011.

[18] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. R. Miller. Maranello: Practical partial packet recovery for 802.11. In *NSDI*. USENIX, 2010.

[19] D. Han and O. Gnawali. Performance of rpl under wireless interference. *Communications Magazine, IEEE*, 51(12):137–143, 2013.

[20] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *REALMAN*. ACM, 2006.

[21] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Norden, and P. Gunningberg. Sonic: classifying interference in 802.15. 4 sensor networks. In *IPSN*. ACM, 2013.

[22] A. Hithnawi, H. Shafagh, and S. Duquennoy. Tiim: technology-independent interference mitigation for low-power wireless networks. In *IPSN*. ACM, 2015.

[23] Y. Hou, M. Li, X. Yuan, Y. T. Hou, and W. Lou. Cooperative cross-technology interference mitigation for heterogeneous multi-hop networks. In *INFOCOM*. IEEE, 2014.

[24] 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.

[25] V. Iyer, F. Hermans, and T. Voigt. Detecting and avoiding multiple sources of interference in the 2.4 ghz spectrum. In *Wireless Sensor Networks*. Springer, 2015.

[26] V. Iyer, M. Woehrle, and K. Langendoen. Chrysso: A multi-channel approach to mitigate external interference. In *SECON*. IEEE, 2011.

[27] Y. Kim, H. Shin, and H. Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *IPSN*. IEEE, 2008.

[28] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *IPSN*. IEEE, 2012.

[29] H. K. Le, D. Henriksson, and T. Abdelzaher. A practical multi-channel media access control protocol for wireless sensor networks. In *IPSN*. IEEE, 2008.

[30] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *SenSys*. ACM, 2010.

[31] 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 *IPSN*. IEEE, 2013.

[32] G. H. M. Sha and C. Lu. Arch: Practical channel hopping for reliable home-area sensor networks. In *RTAS*. IEEE, 2011.

[33] R. Musaloiu-E and A. Terzis. Minimising the effect of wifi interference in 802.15. 4 wireless sensor networks. *International Journal of Sensor Networks*, 3(1):43–54, 2008.

[34] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *SenSys*. ACM, 2008.

[35] 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 *SenSys*. ACM, 2008.

[36] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. Em-mac: a dynamic multichannel energy-efficient mac protocol for wireless sensor networks. In *MobiHoc*. ACM, 2011.

[37] T. Watteyne, A. Mehta, and K. Pister. Reliability through frequency diversity: why channel hopping makes sense. In *PE-WASUN*. ACM, 2009.

[38] G. Werner-Allen, P. Swieskowski, and M. Welsh. Motelab: A wireless sensor network testbed. In *IPSN*. IEEE, 2005.

[39] Y. Wu, J. Stankovic, T. He, S. Lin, et al. Realistic and efficient multi-channel communications in wireless sensor networks. In *INFOCOM*. IEEE, 2008.

[40] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu. Zisense: towards interference resilient duty cycling in wireless sensor networks. In *SenSys*. ACM, 2014.