Escape or Exploit? A Noise-Modulation-Based Communication Under Harsh Interference

Xin Li
SARI, CAS & SIST, ShanghaiTech University
lixin01@sari.ac.cn

Xiaoyuan Ma
SARI, CAS & UCAS
maxy@sari.ac.cn

Peilin Zhang
Carl von Ossietzky University of Oldenburg
peilin.zhang@informatik.uni-oldenburg.de

Pei Tian
SARI, CAS & UCAS
tianpei021@gmail.com

Jianming Wei
SARI, CAS
wjm@sari.ac.cn

ABSTRACT

To survive under interference is a critical requirement of Wireless Sensor Networks (WSNs) in practical applications. At present, existing solutions can be divided into two categories: waiting in time and hopping in channels. However, the interference can be continuously in high-intensity and covers all the channels from 11 to 26 of IEEE 802.15.4. Under such harsh interference, these escaping-based methods could not work any more.

To this end, we propose MoteScatter, a novel noise-modulation-based communication. It is on the basis of backscatter communication but it does not require any dedicated hardware. We implement the prototype on Tmote sky, a commercial WSN device. The transmitter in MoteScatter reflects, i.e., exploits the interference with different amplitudes to deliver information. We change the impedance of the RF antenna on Tmote sky via switching the power amplifier of CC2420 on and off. The receiver extracts the information in the reflected interference with different received signal strength values. We show that MoteScatter can communicate dependably under the harsh interference which other escaping-based protocols can not work. The reliability of MoteScatter is up to 83%. It provides a new paradigm to cope with interference in WSNs.

CCS CONCEPTS

• **Networks → Cyber-physical networks**; • **Computer systems organization → Sensor networks**;

KEYWORDS

Backscatter Communication, Wireless Sensor Network, Dependability Communication

1 INTRODUCTION

Background. A Wireless Sensor Network (WSN) plays an important role of the Cyber-physical System (CPS). It has been applied in various critical scenarios such as industrial manufacturing, smart city and precise agriculture [20]. There exists plenty of interference in the band of 2.4 GHz. WSN has to share its radio medium with other communication technologies such as Wi-Fi and Bluetooth due to the limited ISM band [11]. Moreover+, there are some electrical appliances, such as microwave ovens and induction cookers [3], also radiating electromagnetic waves of 2.4 GHz. To survive under interference has become a critical requirement of WSN in many practical applications. To address this problem, a large number of solutions have been proposed, such as 6TiSCH [9], MiCMAC [1], and eOFPCOIN [18]. Generally, these solutions can be divided into two categories, waiting in time and hopping in channels. A simple example of the waiting in time is a delay-tolerant network. Nodes can restore the messages and communicate with each other until the channel becomes clear. For delay-sensitive applications, hopping in channels is a better choice. Nodes in multi-channel hopping mechanisms avoid interfered channels and select another channel with some strategies to communicate. All of these methods escape interference to communicate at another moment or on another channel.

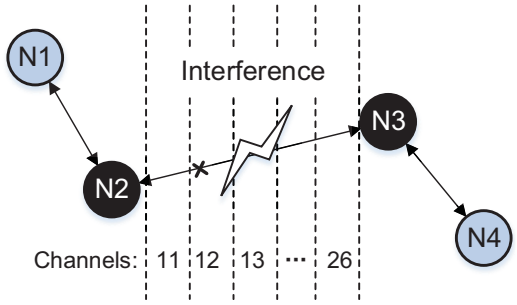

Figure 1: A WSN under interference. We assume that N2 has a packet to send N3, and the link between N2 and N3 is interfered by noise. If the interference is continuously in high-intensity and covers all the channels from 11 to 26 of IEEE 802.15.4, the escaping-based methods (such as the waiting-in-time and hopping-in-channels) could not work any more.

Motivation. As shown in Figure 1, we assume that Node 2 (N2) has a packet to send to N3. The link between N2 and N3 is interfered. As mentioned above, the packet can be successfully received by N3

RealWSN, November 4, 2018, Shenzhen, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

DOI: 10.1145/3277883.3277890

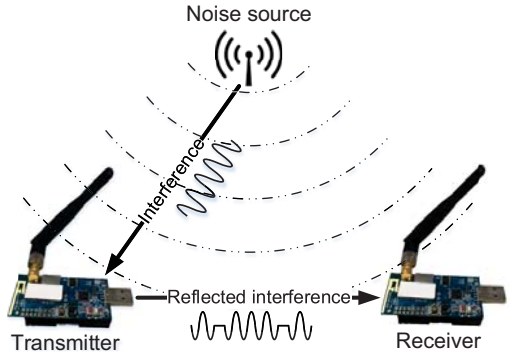

Figure 2: Overview of MoteScatter. There are two roles: the transmitter and the receiver. The transmitter reflects the interference via switching the PA on and off. The receiver then demodulate the reflected interference with different amplitudes.

only when the link is clear or on another channel. However, if the interference is in high-intensity for a long time, the waiting-in-time method is not feasible because the duration of the noise is unknown practically. Even worse, if the interference covers all the channels from 11 to 26 of IEEE 802.15.4 [2], the hopping-in-channels can not work either. In a word, under such harsh interference, these escaping-based mechanisms can not work.

To this end, we propose MoteScatter, a noise-modulation-based communication. Nodes in MoteScatter exploits the interference to communicate with each other. Its principle is backscatter communication. Specifically, as shown in Figure 2, the transmitter in MoteScatter sends information by reflecting the interference with different amplitudes, namely an amplitude modulation (AM). The receiver then can receive the information according to the reflected interference strength, i.e., demodulation of the AM signal.

Contributions. In traditional backscatter communication, transmitter relies on the dedicated hardware to reflect the surrounding radio waveform. In this paper, we implement MoteScatter prototype based on Tmote sky [6], a commercial WSN device. We also record the waveform of reflected interference on the USRP B200 [19], a software defined radio platform.

We male two contributions in this work as follows:

- We propose MoteScatter, a novel noise-modulation-based communication mechanism by exploiting the interference.
- We implement MoteScatter on the commercial WSN device without the dedicated circuit.

Outline. The remainder of this paper is organized as follows. Section 2 discusses the related work, with foci on dependable communication mechanisms and backscatter communication. Section 3 explains the design of MoteScatter. In Section 4, we present the experiments of MoteScatter. We conclude this paper and discuss the future work in Section 5.

2 RELATED WORK

Dependable Communication. Nowadays, to our knowledge, all of the dependable communication protocols apply escaping-based mechanisms, i.e., waiting in time or hopping in channels. 6TiSCH [9] is a classic synchronous mesh network protocol based on Time Slotted Channel Hopping (TSCH) [22]. It autonomously calculates the local schedule and assigns channels to slot frames at each iterative transition. Its end-to-end reliability reaches 99.99% in real deployment. The asynchronous WSN protocol MiCMAC [1] is also a multi-channel hopping protocol. It is extended from ContikiMAC [7]. MiCMAC can maintain above 90% data yield, when the reliability of ContikiMAC drops to 40% in noisy environments. Different from MiCMAC, MOR [24] combines a multi-channel hopping mechanism with the opportunistic routing strategy. Under interference, MOR provides an end-to-end packet delivery ratio (PDR) of more than 98%. As a multi-channel concurrent transmission (CT)-based Glossy [10] protocol, eOFPCOIN [18] achieves a high reliability of 80% under the strongest interference in the evaluation of EWSN 2018 Dependability Competition[1].

In summary, all the protocols achieve high reliabilities. However, when the interference keeps high-intensity continuously over all the channels, all the escaping-based protocols will fail because there is no chance to escape.

Backscatter Communication. Backscatter devices [17] communicate with each other by reflecting the ambient wireless signal. The transmitter in backscatter absorbs or reflects the ambient radio waveform by changing the impedance of the radio frequency (RF) circuit to deliver information. Meanwhile, the receiver can receive the information in the reflected waveform. Compared to traditional RF transmitters with high power consumptions, this mechanism is more energy-efficient because the transmitter does not need to actively generate the RF signal. Therefore, backscatter is commonly used to implement battery-free devices. For instance, ambient backscatter [16] empowers devices with the energy harvested from the ambient RF to communicate. Moreover, the battery-free cellphone [21] is also implemented based on backscatter communication. The backscatter device can also be used as a transport intermediary such as Wi-Fi Backscatter [14] and Interscatter [13]. Wi-Fi Backscatter [14] implements a battery-free Wi-Fi communication to bridge more RF-powered devices with the Internet. Interscatter [13] makes use of backscatter communication to achieve the cross-tech communication from Bluetooth to Wi-Fi. All the above applications require dedicated backscatter devices. However, NICScatter [23] is a backscatter communication on a commercial Wi-Fi network interface card (NIC) without any particular hardware. NICScatter hacks the Wi-Fi NIC to send packets via the incident signal to another computer. The sender in NICScatter changes the RF impedance by powering on/off the Wi-Fi NIC.

In this paper, we propose MoteScatter which is inspired by NICScatter. However, we modulate the interference to achieve a novel paradigm of dependable communication.

3 MOTESCATTER

3.1 MoteScatter in a Nutshell

In this section, an overview of MoteScatter is presented. Different from escaping-based mechanisms, MoteScatter exploits interference. Namely, nodes in MoteScatter communicate with each other

[1] https://iti-testbed.tugraz.at/blog/page/11/ewsn-18-dependability-competition-final-results/

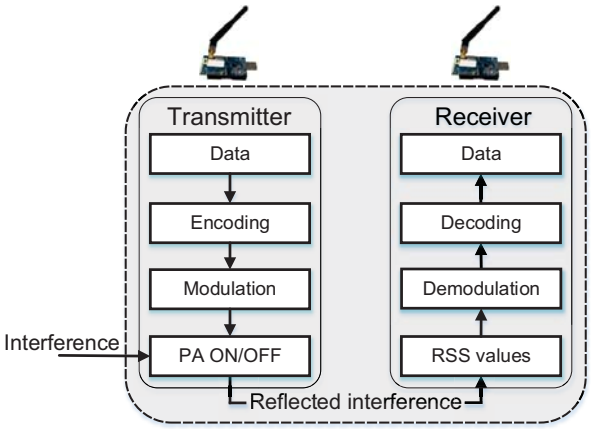

Figure 3: Framework of MoteScatter. The interference is in high-intensity over all the channels for a long time. The transmitter modulates the interference by switching the PA on and off. The receiver demodulates the reflected interference via RSS values.

via reflecting the interference. Therefore, MoteScatter can still work even when the interference is continuously intensive over all the channels.

As shown in Figure 3, the transmitter changes the impedance of the antenna of Tmote sky via configuring the corresponding registers. Then, the interference is modulated, i.e., reflected with different intensities. Thus, the transmitter can deliver binary bits.

The received signal strength indicator (RSSI) is a measurement of the power presented in a received radio signal. The receiver makes use of the RSS values to read the waveform of the modulated interference. The reflected interference can be demodulated as the bit "1" when the RSS value is greater than a certain threshold, and vice versa.

3.2 Transmitter

Noise Source. In order to simulate the hash interference environment, we use JamLab [5] as the noise source. JamLab can generate interference directly by CC2420 [12], the transceiver on Tmote sky. It has two basic modes: the modulated mode and the unmodulated mode. The interference in the modulated mode is over a broader band than that in the unmodulated mode. To interfere in a broader band as possible as we can, we choose the modulated mode. The transmit power is set to 31 (0 dBm) and the channel is 26 (2.48 GHz).

Power Amplifier (PA) Operation. MoteScatter does not require a dedicated hardware to control the impedance of RF antenna. The impedance of the antenna on Tmote sky is different when the node works in different states, for example, power on or off. The

OPERATION	OFF (ohms)	ON (ohms)
Power Supply	$18.3 - j29.6$	$117.5 - j3.2$
Power Amplifier	$18.3 - j29.7$	$87.9 - j17.4$

Table 1: The impedances of Tmote sky under several states at 2.48 GHz.

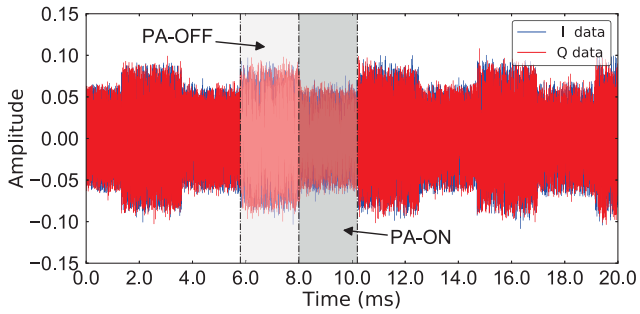

Figure 4: Waveform of the reflected interference. The interference generated by JamLab is reflected by the transmitter via switching the PA on and off each 2 ms.

impedances of the antenna at 2.48 GHz under several states are measured by the vector network analyzer. As shown in Table 1, powering the mote on/off and switching the PA both can change the RF impedance significantly. However, Tmote sky can not power on/off by itself. Relatively speaking, switching the PA is easier to implement in practice. We switch PA on and off via the registers MANAND and MANOR in CC2420 as described in [25]. The specific operation of PA on can be configured as follows:

```
setreg(CC2420_MANAND, 0xFDFF);
setreg(CC2420_MANOR, 0x0400);
```

and for PA off is as follows:

```
setreg(CC2420_MANAND, 0xFFFF);
setreg(CC2420_MANOR, 0x0600).
```

Waveform of the Reflected Interference. As shown in Figure 4, we use the USRP to record the reflected interference. There are two characteristics: 1) When the PA is switched off, the amplitude of the reflected interference is larger than the on state. It shows that switching the PA on/off does reflect/absorb the interference. 2) During the PA is in the on or off state, the amplitude of the reflected interference does not change. It means that the impedance of antenna is stable. As a result, the energy of the absorbed/reflected interference is almost constant. These ensure that the transmitter can modulate the reflected interference with two different amplitudes.

Switching Rate. To explore the fastest transmission rate of MoteScatter, we use the USRP to observe the reflected interference, and switch the PA on and off with a given interval. The waveform is processed by the `mag` block and the `low pass filter` block in GNU radio [4]. As shown in Figure 5, we adjust the switching interval from 50.5 µs (19.8 kHz) to 6.2 ms (161.29 Hz). We find that the amplitude of the reflected interference changes significantly, even when the switching interval is only 50.5 µs, i.e., the fastest PA switching rate.

3.3 Receiver

RSSI Trace. Most of wireless devices are equipped with RSSI. We exploit the RSSI of CC2420 to detect the power of the reflected interference. The recorded RSSI trace is shown in Figure 6. As presented in Section 3.2, the RSS values sensed by the node can

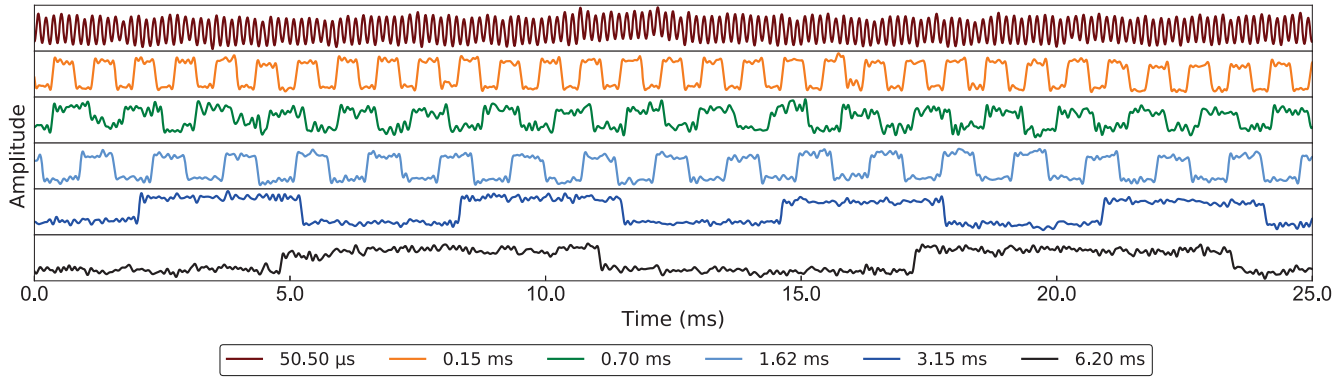

Figure 5: Waveforms of the reflected interference with different switching periods. The reflected interference is backscattered by Tmote sky with given interval.

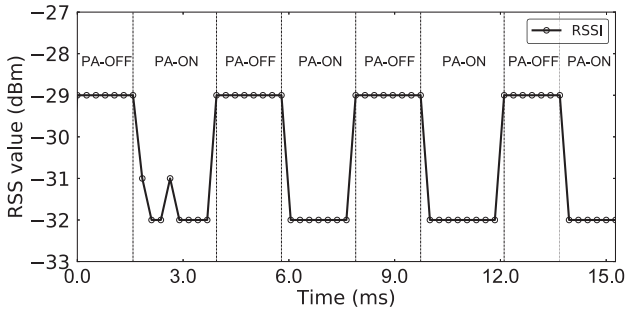

Figure 6: The RSSI trace of the reflected interference. The reflected interference is detected by the receiver while the transmitter is switching the PA on and off every 260 µs.

profile the interference reflected by the transmitter. During the transmitter's PA is in the state of on or off, the RSS readings of the receiver are relatively stable. Consequently, it is feasible to demodulate the reflected interference with RSS values on Tmote sky.

Scanning Rate. We measure the RSSI scanning rate on FlockLab [15], a wireless sensor network test platform located at ETH Zurich, Switzerland. The result is shown in Table 2. According to the datasheet of CC2420 [12], the RSS value is averaged over an 8-symbol period (0.128 ms). Namely, the RSSI sampling rate of CC2420 is 7812 Hz. As shown in Table 2, the routine of `do_rssi_scan()` encapsulated in Contiki [8] Operating System (OS) 3.0 takes an average of 0.26 ms. Therefore, for the receiver, the fastest RSSI scanning rate with the service provided by Contiki OS is 3846 Hz.

Make the Receiver Robust. If the amplitude of the interference is constant, the receiver is able to extract the information successfully so far. However, it is not reasonable to assume the amplitude of the interference is constant in practical applications. As shown in Figure 7, we find that the RSSI trace of the daytime reflected interference has significant fluctuations compared to the one of the nighttime. To make Motescatter work under various interference, we 1) let nodes in MoteScatter communicate in packet; 2) design a

OPERATION	TIME (ms)	RATE (Hz)
RSSI_SAMPLE_CC2420()	0.128	7812
do_RSSI_SCAN()	0.26	3846
do_RSSI_SCAN() +HIGH_PASS_5orders() +LOW_PASS_10orders()	9.43	106

Table 2: RSSI scanning rate.

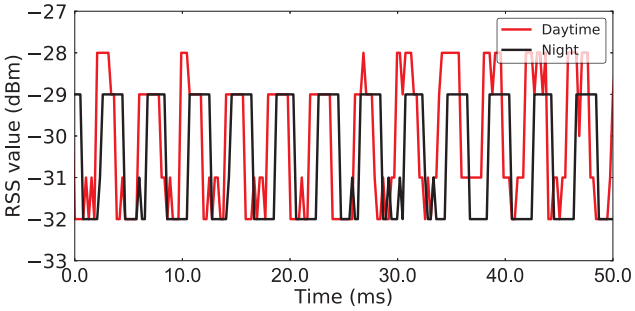

Figure 7: RSSI traces of reflected interference during day and night. The RSSI trace of the daytime reflected interference has significant fluctuations compared to that of the night.

finite impulse response (FIR) filter on the receiver. Specifically, the filter consists of a 5-order high-pass filter and a 10-order low-pass filter.

As mentioned above, the execution of the RSSI scanning function requires 0.26 ms. But the filter, which requires average 9.17 ms, is time-consuming. That means, the filter reduces the RSSI scanning rate severely. Therefore, in MoteScatter, after the preamble of a packet is detected successfully, the rest waveform of the packet, i.e., the reflected interference, is recorded and filtered later.

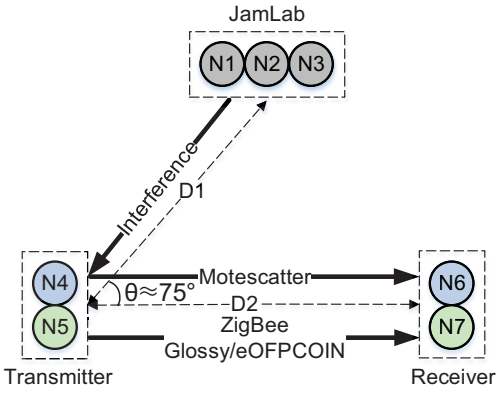

Figure 8: Deployment of the evaluation. D1 is the distance from JamLab (N1, N2 and N3) to transmitters (N4 and N5), and D2 is the distance between transmitters and receivers (N6 and N7). N1, N2 and N3 generate the modulated noise with JamLab. N4 and N6 work as the transmitter and the receiver of MoteScatter respectively. The escaping-based protocols, Glossy or eOFPCOIN, are running on N5 and N7.

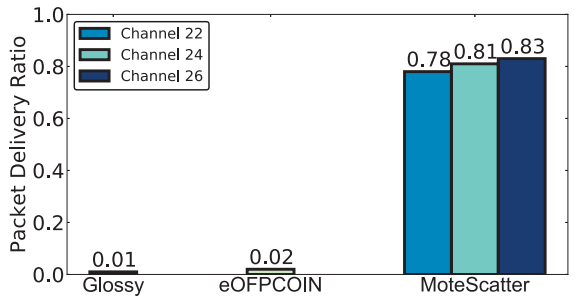

Figure 9: Reliability of Glossy, eOFPCOIN and MoteScatter. The escaping-based protocols Glossy and eOFPCOIN cannot work, While MoteScatter achieves high PDRs on all the three channels.

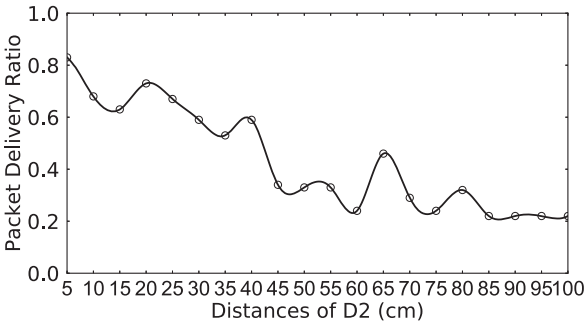

Figure 10: PDR with different distances of D2. We set D1 as 5 cm, and measure the PDRs of MoteScatter at different distances between transmitter and receiver (D2).

4 EVALUATION

4.1 Settings

We implement MoteScatter on Tmote sky, and evaluate the reliability and the communication distance. The experiment scenario is illustrated in Figure 8. D1 is the distance from the noise to the transmitter, and D2 is the distance between the transmitter and the receiver. In this evaluation, we set the transmission rate as 100 bps, and the transmitter sends a packet with 8 bits per 0.5 ms. In our prototype, we use JamLab to generate continuous interference on Tmote sky on three channels. Two nodes are used as the transmitter and the receiver of MoteScatter respectively. To compare with the the traditional escaping-based protocols, the other two nodes run Glossy [10] or eOFPCOIN [18] in our experiments. In this implementation, we use seven Tmote sky nodes, and the specific settings are as follows:

(1) Three nodes (N1, N2 and N3) generate the modulated interference with JamLab. They are with the maximum transmit power (0 dBm) on the channel 26, 24 and 22 respectively.
(2) N4 and N6 work as the transmitter and the receiver of MoteScatter. The escaping-based protocols, Glossy or eOFPCOIN, are running on N5 and N7.
(3) The communication channel of Glossy is set to 26. eOFPCOIN hops channel on 22, 24 and 26. The transmit power of the escaping-based protocols are set to the maximum, i.e., 0 dBm.
(4) We set D1 as 5 cm to simulate a harsh interference.
(5) We change D2 to evaluate the communication range and dependability of MoteScatter.

4.2 Performance

MoteScatter vs. Escaping-based Protocols. We compare MoteScatter with Glossy and eOFPCOIN to evaluate the reliability based on the Packet Delivery Ratio (PDR). The PDR is the ratio of the successfully received packets to the transmitted packets. As shown in Figure 9, the nodes with the escaping-based protocols hardly communicate and the PDRs are nearly equal to 0. The continuous interference in high-intensity is a great challenge for the escaping-based protocols. On the contrary, MoteScatter achieves high PDRs on all the three channels. All the experiments are carried out in an office with Wi-Fi and the channel near 22 of IEEE 802.15.4 is more likely to be affected with the Wi-Fi signal. Thus, the PDR of MoteScatter on the channel 22 is lower than others.

PDR vs. Distance. In this experiment, we measure the PDRs of MoteScatter at different distances between transmitter and receiver (D2 in Figure 8), and the result is shown in Figure 10. A large communication distance in MoteScatter brings about a great attenuation of the reflected interference. Thereby, the weak reflected interference is hard to be demodulated by the receiver correctly. Thus, PDR decreases with the increasing distance.

We also consider the relationship between PDR and the distance with another view. In MoteScatter, the interference is the carrier of the communication. That is to say, the receiver and the transmitter in MoteScatter work only when they share the same carrier. The receiver can hardly receive the carrier if D2 is much longer than

D1 in this experiment. Thereby, the PDR drops naturally.

5 CONCLUSION AND FUTURE WORK

Conclusion. In this paper, we present MoteScatter, a noise-modulation-based communication via reflecting interference. Different from the escaping-based mechanisms such as multi-channel hopping, MoteScatter, exploiting interference, provides a novel approach to communicate under interference. Its design is based on backscatter communication. To reflect the interference, we change the impedance of the RF antenna on Tmote sky via switching the power amplifier of CC2420. The receiver can receive the reflected interference by RSS values. We implement the prototype and evaluate the performance experimentally. The results show that MoteScatter can communicate under the harsh interference which the current escaping-based protocols can not work. The reliability of MoterScatter under interference is up to 83%.

Future work. MoteScatter is a typical amplitude modulation communication system. Our future work mainly has three directions: 1) Optimizing the reliability with channel coding. In this prototype, we use the simplest modulation and encoding mechanism. In the future, we are going to combine the spreading spectrum to enhance the reliability and extend the communication distance. 2) Improving the RSSI scanning rate. In the current prototype of MoteScatter, we profile the reflected interference with RSS values. The RSSI scanning rate of the receiver in current MoteScatter (3846 Hz) is provided by the Contiki OS 3.0. It is actually much lower than the maximum switching rate of the PA (19.8 kHz), namely the maximum transmission rate we can reach (19.8 kbps). Thus we still have space to improve the communication rate. 3) Cross-tech communication. MoteScatter provides a novel way to achieve backscatter communication via the PA operation. In principle, the reflected interference in MoteScatter can be received by other co-existed communication technologies in 2.4 GHz, such as Wi-Fi and Bluetooth.

ACKNOWLEDGMENTS

We would like to thank the Computer Engineering Group at ETH Zurich, Switzerland for providing the FlockLab testbed. We would also like to thank Zhice Yang, for his help and inspiration at the beginning of this work. This research is supported by the National Key R&D Program of China (2016YFC0801500).

REFERENCES

[1] Beshr Al Nahas, Simon Duquennoy, Venkatraman Iyer, and Thiemo Voigt. 2014. Low-Power Listening Goes Multi-channel. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2014)*. IEEE, 2–9.

[2] IEEE Standards Association et al. 2011. IEEE Std 802.15. 4-2011, IEEE standard for local and metropolitan area networks–Part 15.4: Low-rate wireless personal area networks (LR-WPANs).

[3] Babak Azimi-Sadjadi, Daniel Sexton, Ping Liu, and Michael Mahony. 2006. Interference effect on IEEE 802.15.4 performance. In *Proceedings of 3rd international conference on networked sensing systems (INNS), Chicago, IL*.

[4] Eric Blossom. 2004. GNU radio: tools for exploring the radio frequency spectrum. *Linux journal* 2004, 122 (2004), 4.

[5] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer, and Marco Zúñiga. 2011. JamLab: Augmenting Sensornet Testbeds withRealistic and Controlled Interference Generation. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM/IEEE, 175–186.

[6] Moteiv Corporaton. 2006. Tmote sky: Datasheet. http://www.crew-project.eu/sites/default/files/tmote-sky-datasheet.pdf.

[7] Adam Dunkels. 2011. The ContikiMAC radio duty cycling protocol. (2011).

[8] Adam Dunkels, Björn Grönvall, and Thiemo Voigt. 2004. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. IEEE, 455–462.

[9] Simon Duquennoy, Atis Elsts, Al Nahas, and George Oikonomou. 2017. TSCH and 6TiSCH for Contiki: Challenges, design and evaluation. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2015)*. 11–18.

[10] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. 2011. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM/IEEE, 73–84.

[11] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. 2011. Clearing the RF smog: Making 802.11n robust to cross-technology interference. *ACM SIGCOMM Computer Communication Review* 41, 4, 170–181.

[12] Texas Instruments. 2007. CC2420 datasheet. http://focus.ti.com/lit/ds/symlink/cc2420.pdf.

[13] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. 2016. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 356–369.

[14] Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R Smith, and David Wetherall. 2014. Wi-Fi backscatter: Internet connectivity for RF-powered devices. *ACM SIGCOMM Computer Communication Review* 44, 4, 607–618.

[15] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. 2013. FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM/IEEE, 153–166.

[16] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. 2013. Ambient backscatter: Wireless communication out of thin air. *ACM SIGCOMM Computer Communication Review* 43, 4, 39–50.

[17] Wanchun Liu, Kaibin Huang, Xiangyun Zhou, and Salman Durrani. 2017. Next generation backscatter communication: Theory and applications. *arXiv preprint arXiv:1701.07588* (2017).

[18] Xiaoyuan Ma, Peilin Zhang, Weisheng Tang, Xin Li, Wangji He, Fuping Zhang, Jianming Wei, and Theel Oliver. 2018. Using enhanced OFPCOIN to monitor multiple concurrent events under adverse conditions. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks, Dependability Competition*. 211–212.

[19] Ettus Research. online. USRP B200. https://www.ettus.com/product/details/UB200-KIT.

[20] John A Stankovic. 2014. Research directions for the internet of things. *IEEE Internet of Things Journal* 1, 1 (2014), 3–9.

[21] Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua R Smith. 2017. Battery-free cellphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 25.

[22] Thomas Watteyne, Vlado Handziski, Xavier Vilajosana, Simon Duquennoy, Oliver Hahm, Emmanuel Baccelli, and Adam Wolisz. 2016. Industrial wireless IP-Based cyber-physical systems. *Proceedings of the IEEE* 104, 5 (2016), 1025–1038.

[23] Zhice Yang, Qianyi Huang, and Qian Zhang. 2017. NICScatter: Backscatter as a covert channel in mobile devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 356–367.

[24] Peilin Zhang, Olaf Landsiedel, and Oliver Theel. 2017. MOR: Multichannel opportunistic routing for wireless sensor networks. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 36–47.

[25] Peilin Zhang, Xiaoyuan Ma, Oliver Theel, and Jianming Wei. 2018. Concurrent transmission-based packet concatenation in wireless sensor networks. In *Proceedings of the 43rd Annual IEEE Conference on Local Computer Networks (LCN 2018)*. Chicago, USA.