DiVA
http://www.diva-portal.org

Postprint

This is the accepted version of a paper presented at *International Conference on Embedded Wireless Systems and Networks (EWSN '16)*.

Citation for the original published paper:

Pérez-Penichet, C., Varshney, A., Hermans, F., Rohner, C., Voigt, T. (2016)
Do Multiple Bits per Symbol Increase the Throughput of Ambient Backscatter Communications?.
In: *EWSN '16 Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks* (pp. 355-360).

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-306897

Do Multiple Bits per Symbol Increase the Throughput of Ambient Backscatter Communications?

Carlos Pérez-Penichet, Ambuj Varshney, Frederik Hermans, Christian Rohner, Thiemo Voigt

Uppsala University, Sweden

Abstract

Backscatter wireless communications have exceptionally stringent power constraints. This is particularly true for *ambient* backscatter systems, where energy and wireless carrier are both extracted from weak existing radio signals. The tight power constraints make it difficult to implement advanced coding techniques like spread spectrum, even though such techniques are effective in increasing the communication range and robustness in this type of systems. We draw inspiration from *µcode*, a previous backscatter coding approach, where data bits are encoded in single-data-bit chip sequences of considerable length to gain robustness. We introduce a new coding technique that encodes several bits in a single symbol in order to increase the data rate of ambient backscatter, while maintaining an acceptable compromise with robustness. We study the proposed technique by means of simulations and characterize the bit error rate and data rate dependencies. A comparison with *µcode* is drawn and the benefits of each approach are analyzed in search for the best strategy for increasing data rate while maintaining robustness to noise.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Backscatter Communications

Keywords

Backscatter Communications, Coding, Simulation

1 Introduction

Ambient backscatter communications devices are very attractive in the context of the Internet of Things. These devices exploit existing radio signals both to harvest energy for their own operation and to communicate with other sensors [6]. As a consequence they have the potential to operate battery-free for extended periods of time. In order for this vision to be taken to its full potential, the communication range of these devices needs to be increased to and beyond room level. At the same time, the data rate of needs to be kept reasonably high. Previous research efforts by Parks et al. [10] have shown good results in increasing communication range. They propose to use coding techniques that are appropriate for low-power operation as demanded by the constraints of ambient backscatter communications.

The approach proposed by Parks et al. employs a coding scheme inspired by spread spectrum techniques to increase the robustness and thereby also the range of communication[1]. Their scheme represents each data bit by one symbol; each of the two possible symbols is in turn represented by a predefined chip sequence. While encoding only a single bit in each symbol provides maximum resilience to noise, it is potentially wasteful when channel conditions are good. Under high signal-to-noise ratio (SNR), it may be preferable to increase the data rate by coding several data bits in each symbol.

In this paper, we present and evaluate a new coding scheme that extends the previous scheme and has the potential to increase the data rate while preserving energy efficiency and robustness. At its core, our new coding scheme represents multiple bits per symbol by defining new chip sequences which are constructed by combining the original symbols from *µcode*. While coding multiple bits per symbol may make transmissions more vulnerable to noise and interference under unfavorable channel conditions, it increases the data rate in low-noise scenarios. This trade-off between robustness and data rate makes our approach potentially advantageous in favorable situations. Furthermore, changing the number of chips per symbol (and hence the duration of information symbols) can also have an impact both on robustness and data rate. One can envision a future system that dynamically plays with both of these

[1]Note that communication range is tied to robustness since the lowest signal-to-noise ratio (SNR) value where information exchange is still possible essentially defines the communication range for these devices.

trade-offs in order to adapt the data rate to changing link conditions. In this paper, we investigate the effects that these two aspects—the number of bits per symbol and the length of chip sequences—have on the robustness and data rate of backscatter communications.

The contributions of this paper are as follows:

- We extend an existing solution, *µcode*, with a new multiple-bit-per-symbol technique that increases the data rate.
- We develop a simulation framework for studying the performance of our solution, as well as the trade-off between the length of chip sequences and robustness.
- We compare our approach with *µcode* for various channel conditions. Our results suggest that even though multiple-bits-per-symbol encoding increases the data rate, the reduction in robustness makes it more beneficial to reduce the number of chips per data symbol when in search for an increase in data rate, provided that the channel conditions allow it.

The rest of this paper is organized as follows: Section 2 discusses related work, while Section 3 presents the necessary background. Section 4 describes our approach and Section 5 presents the simulation framework we have implemented. We evaluate our results comparing our technique to existing work in Section 6. Before we summarize our results and present some future work, we briefly discuss a way to implement our approach in hardware in Section 7.

2 Related Work

Backscatter communication holds vast potential for battery-less sensing devices owing to a reduction of orders of magnitude in energy cost when compared to conventional radio communication [3]. This communication method has been extensively studied in the past for RFID applications like inventory management and object tracking [12]. A central design element of these applications is the presence of a high-powered reader to generate the carrier wave, as well as to read backscattered reflections. The reader is usually expensive, and the data rate and communication range are extremely limited. This has limited the potential applications of the backscatter-enabled devices. Recent efforts that aim to alleviate these shortcomings can be roughly divided into two categories: (1) improving the traditional backscatter communication and (2) overcoming the need for an expensive reader in traditional backscatter.

In the former category, we find efforts to improve either communication range or data throughput in traditional backscatter systems. Zhang et al. leverage the fact that computation is more energy expensive than communication in backscatter systems, and improve the communication throughput to the order of megabits per second while keeping the energy consumption in the microwatts [14]. Hu et al. exploit the asymmetry in computational power between the nodes and the reader to design a protocol that enables high-throughput concurrent transmissions [4]. Zhang et al. present a network stack called QuarkNet which dynamically scales its size depending on the available energy, thereby improving the range and throughput [13]. The EPCglobal Class-1 Generation-2 (C1G2) standard is used in RFID communication systems [1]. Zheng et al. present a protocol called Harmony which improves the bulk data performance while adhering to the C1G2 standard [15].

The latter category of recent work tries to address the constraints imposed by the need for a reader and the central nature of the communication. Nikitin et al. aim to reduce the cost of the reader and present a design of a simple and a low-cost reader for RFID applications [9]. Further, Nikitin et al. show that it is possible to have communication between co-located tags and demonstrate tag-to-tag communication at a range of a few centimeters [8]. These are still limited owing to the need for a separate reader.

Liu et al. present *ambient backscatter*, a significant shift in the design of backscatter communication systems [6]. Ambient backscatter leverages ambient RF signals, such as TV signals, both as a carrier for communication and source of energy. This strategy avoids the need for a separate reader and enables device-to-device communication and battery-free operation [6]. Recent research efforts on ambient backscatter include the use of antenna cancellation to improve throughput and coding to improve range [10], enabling full-duplex backscatter communication [7] and using ambient WiFi signals as the carrier [5, 2]. Our work investigates whether we can improve the coding scheme described by Parks et al. [10] and thereby complements related work on ambient backscatter [6, 10, 7, 5, 2].

3 Background

Coding techniques such as code division multiple access (CDMA) or Direct Sequence Spread Spectrum (DSSS) are commonly used in conventional radio communications to enable concurrent transmissions, increase communication range, and reduce the impact of interference [11]. In these techniques, sequences of pseudorandom chips are used to represent information symbols. For instance, the chip sequence '1110100010' could represent a zero bit while sequence '0011001001' could represent a one bit. At the receiver side, the signal is correlated to these chip sequences to determine which information symbol has been sent by selecting the highest correlation value among the possible symbols. The length of the chip sequence is directly proportional to the robustness to noise, with a longer sequence resulting in higher robustness but also lower data rate. This trade-off means that data rate can be traded for communication range. A drawback of such coding techniques is that they usually require tight synchronization between sender and receiver.

Parks et al. [10] have successfully devised a prototype that employs the same underlying principle to signifi-

cantly increase the communication range of an ambient backscatter platform. At its core, their technique ($\mu code$) leverages spread spectrum to recover a weak signal in a noisy environment. Using these techniques to receive information in backscatter devices introduces two additional challenges: (1) The correlation operation is computationally and energetically expensive. (2) Performing synchronization at the receiver would require more energy than what is typically available in backscatter nodes.

$\mu code$ overcomes the first challenge by performing the correlation in an energy-efficient fashion in the analog domain and detecting the transmitted symbol with a single comparator. The second issue has been solved by employing periodic chip sequences, thus avoiding the need for tight synchronization between sender and receiver. Their transmitter uses the following chip sequence to represent zero and one data bits:

$$\begin{array}{ll} 1010\cdots10 & \text{as bit 1} \\ \underbrace{0000\cdots00}_{N} & \text{as bit 0} \end{array}$$

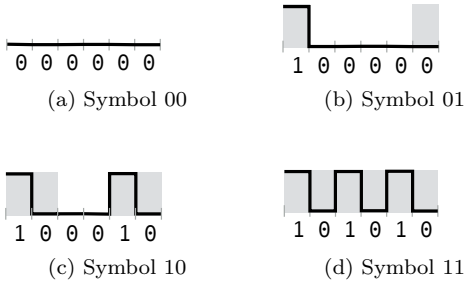
To decode an incoming transmission, the receiver performs the correlation by computing an in-phase component (I) by performing a dot-product operation of the received signal with an alternating sequence of ones and zeroes. The quadrature component (Q) is calculated by a dot-product with a 90 degree offset sequence of ones and zeroes instead. Note that 90 degree offset is equivalent to a time offset of half a chip duration. The receiver then is able to decode data bits by computing $|I|+|Q|$, which is a constant value that depends only on the number of chips per symbol (N) in the chip sequence, but is independent of the phase offset. In fact, the receiver performs this computation three times over a duration of $\frac{N}{3}$ chips each and the final result is obtained by comparing the majority vote of three adjacent computations to a threshold. The threshold is obtained from the average of the received signal during the preamble of each packet.

The result by Parks et al., however, shows that the data rate of the system is compromised by the use of the code. This compromise is worth making in noisy environments where a low data rate is better than no data at all. However, in less challenging conditions, the potential for a significant data rate increase would be wasted since the number of bits per symbol and chips per symbol of the code are fixed.

We present a flexible multi-bit coding scheme that could allow increased flexibility by selecting the number of bits per symbol and the number of chips per symbol.

4 Multi-bit Coding

While the solution introduced by Parks et al. is efficient in computing correlations and avoids phase recovery, its one-bit-per-symbol encoding appears conservative. Inspired by quadrature modulation, where information typically is encoded in both magnitude and phase, we extend their scheme to encode multiple bits per symbol. To extend the scheme beyond two symbols, we encode the data bits in the magnitude (computed as the correlation by equation 1) by duty cycling the alternating chip sequence: for a two-bit symbol we generate four sequences with 0, $N/3$, $2N/3$, and N chips alternating between ones and zeroes, and the rest zeroes. Figure 1 shows a representation of our chip sequences and their corresponding signals for the case of two bits per symbol and six chips per data bit ($N = 6$). Note that the chip sequences for 00 and 11 correspond to bit 0 and 1 in $\mu code$, respectively.



(a) Symbol 00 (b) Symbol 01

(c) Symbol 10 (d) Symbol 11

Figure 1: **Representation of the chip sequences and corresponding signals for the case of two bits per symbol and six chips per data bit.** Symbols 00 and 11 are the ones used in $\mu code$ to represent zero- and one-bits respectively.

Encoding on alternating chip sequences is essential to maintain the beneficial correlation and synchronization properties of $\mu code$. For the same reason, to remain phase-independent, we limit our coding to varying the correlation magnitude.

For decoding, we compute the in-phase (I) and quadrature (Q) components in analogy to Parks et al., with them, the absolute correlation value D is computed:

$$D = |I| + |Q| \tag{1}$$

where D is 0, $N/3$, $2N/3$, N depending on the transmitted symbol.

In order to demonstrate the feasibility of our scheme and to evaluate its performance in comparison to $\mu code$, we simulate both coding approaches.

5 Simulation Framework

We have simulated the encoding and decoding process from a relatively high-level perspective. The simulation is implemented as a Python script. The general framework of the simulation is conceptualized in Figure 2.

At a high level, the simulation works as follows: (1) A large number of random bits is generated at the transmitter. (2) Those bits are encoded using the desired method: $\mu code$ or our approach. (3) The encoded signal is transfered to the receiver through a simulated channel of specified conditions. (4) At the receiver, the signal is decoded with the corresponding decoder. (5)

The received data is compared with the transmitted bits in order to compute the resulting bit error rate (BER).

Once one of the two encode/decode approaches is selected, it is also possible to specify the number of chips per symbol to be used in the simulation. In the case of our multiple-bit-per-symbol approach it is also possible to select the number of bits per symbol to use.

We have chosen to simulate the wireless channel as an additive white Gaussian noise (AWGN) channel as it is a simple and well-known channel model. It would be straightforward to add support for other models in our framework. The AWGN channel section of the simulation works as follows: (1) The channel receives as input the chip sequence of the signal to be transmitted and the desired SNR value. This occurs after encoding so this process and the channel in general do not depend in any way on the coding scheme used. (2) The channel simulator computes P_e, the probability of chip error based on the theoretical equation for P_e of PCM under an AWGN channel [11]. (3) Based on the probability of chip error and the number of chips received as input, the corresponding number of chips are flipped uniformly at random. (4) The output is the subsequent signal with flipped chips.

At this point, it is important to notice an important detail. Even though it is customary to plot BER curves as a function of E_b/N_0 (energy per bit over noise spectral density), we have chosen to make a distinction in this aspect in order to avoid confusion. Because in this particular case, the chips that compose the chip sequence are the ones being modulated as pulse-code modulation (PCM) signals and not the data bits themselves, we have used the notation E_c/N_0 to refer to the energy per chip over noise spectral density. E_c/N_0 refers to the ratio between the energy per transmitted chip and the noise energy per Hertz over the signal bandwidth [11]. Our plots in Section 6 are drawn against this quantity.

The number of data bits in error and the BER values are computed by comparing the decoded signal at the receiver with the original data bit sequence.

Note that for higher values of E_c/N_0, when the BER values tend to be rather small, it becomes necessary to simulate a rather large number of data bits in order to gather statistically significant data.

All results from the simulation are obtained by running it multiple time for each parameter combination and averaging the results in the end. The error bars when presented have been computed from the standard deviation of the corresponding points across multiple curves.

As a sanity check, Figure 3 displays a comparison between the theoretical BER curve for a PCM signal that goes through an AWGN channel and our simulation of *µcode* for the boundary case of $N = 1$, which is identical to PCM. The figure shows that there is a very good agreement between the simulation and the theoretical curve.

Our simulation can be used to evaluate and compare our proposed approach. This can be achieved by changing the encoding and decoding stages to select the desired approach in this simulation schema. The simulation parameters control the number of bits per symbol and the number of chips per symbol among other simulation-specific parameters.

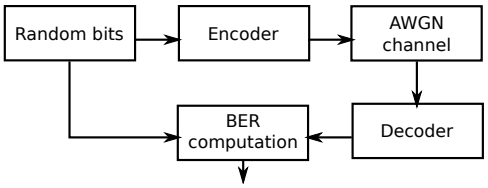


Figure 2: **Block diagram of the simulation framework.** In each run of the simulation, we encode a large number of random bits, transform the resulting chip sequences with an AWGN channel, and then decode them again.

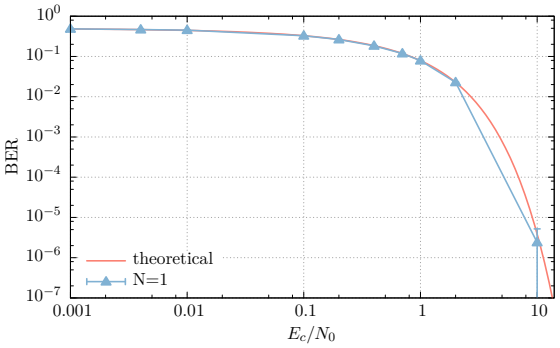


Figure 3: **Comparison of the theoretical BER curve of PCM to our equivalent simulated results.** Our simulation generates an almost perfect match to the theoretical values.

6 Evaluation

In this section, we compare the performance of our technique to that of *µcode*. The results are obtained with the simulation framework described in the previous section.

We first aim to understand how communication robustness changes with the length of chip sequences. Figure 4 presents the dependency of bit error rate (BER) vs. energy per chip over noise spectral density (E_c/N_0) of our technique for different numbers of chips per symbol (N). The figure shows that as the number of chips increases, there is a displacement of the curves towards smaller BER values. This behavior is expected since more chips per symbol make each symbol more robust. In other words, the longer the chip sequences, the more chips need to be flipped for the transmitted symbol to be decoded incorrectly at the receiver.

We next try to answer the following central question: how does our technique compare to *µcode*? To answer this question, we have performed simulation runs with both *µcode* and our scheme changing the number of chips per symbol N.

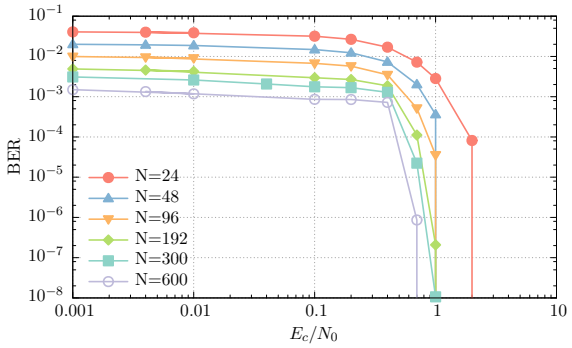


Figure 4: **BER vs E_c/N_0 for different number of chips per symbol (N) in our approach.** As expected, increasing the number of chips per symbol increases robustness at the cost of data rate.

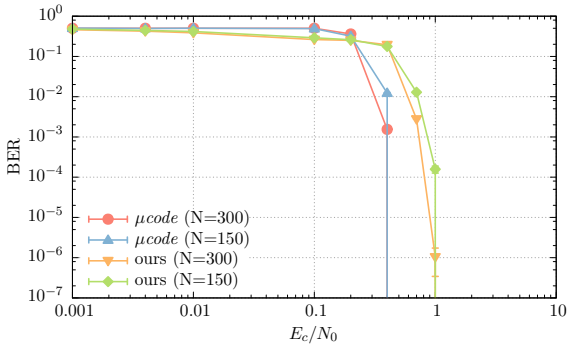


Figure 5: **Comparison of BER vs E_c/N_0 between *µcode* and our new approach for different numbers of chips per symbol.** Our results suggest that even though our approach increases the data rate, a better performance gain is obtained by reducing the number of chips per symbol when conditions allow.

Figure 5 presents a comparison of the BER vs. E_c/N_0 curves of *µcode* against our approach for different number of chips per symbol. The figure shows how the introduction of more than one bit per symbol reduces the robustness of the encoding. This creates a trade-off between data rate and vulnerability to bit errors. The figure also shows that, for both approaches, the effect of increasing the length of the chip sequences is to increase the robustness. Equivalently, increasing the number of chips per data bit has the same effect.

Notice that in Figure 5 the curve corresponding to *µcode* with $N = 300$ has the same data rate as the one corresponding to our approach with $N = 150$. While these curves have the same data rate, the BER of *µcode* is lower for reasonably high values of E_c/N_0. More surprising is that even the curve of *µcode* for $N = 150$ shows better robustness than both curves from our approach. This suggests that, in order to increase the data rate, it is more advantageous to reduce the number of chips per symbol rather than to increase the number of data bits per symbol.

The results of our simulation clearly show that a trade-off exists between robustness and the number of bits per data symbol. The highest robustness is reached for a single bit per data symbol. Yet another trade-off exists between robustness and the length of the chip sequences. The longer the chip sequence, the more resilient the encoding signal will be against noise. These results are expected. Figure 5, however, shows that it should be more beneficial to reduce the value of N, rather than to increase the number of bits per symbol which is more surprising.

Another unexpected insight drawn from our results is related to the shape of curves in Figures 4 and 5. The bend in these curves becomes sharper as N increases and Figure 5 depicts that for lower values of E_c/N_0 our approach becomes better than *µcode*. This can be explained by the fact that, while this type of codes introduce a certain error correcting gain, this effect works at a disadvantage when the chip error rate is too high.

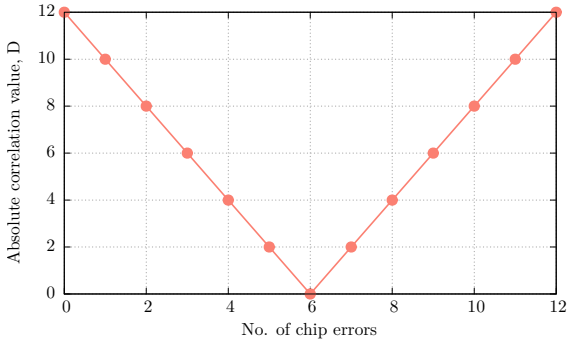


Figure 6: **Absolute correlation values as a function of the number of chip errors for $N = 12$.** The values decrease as the number of chip errors increases.

In those cases the effect is actually the opposite and the bit error rate is increased by the code rather than decreased. This can be explained by analyzing the correlation values as the number of chip errors increases. Figure 6 shows the values of the absolute correlation as a function of the number of chip errors for $N = 12$ as computed with equation (1). As the correlation diminishes, the symbols become progressively more similar to an erroneous symbol and once the absolute correlation falls bellow $N/2$, the symbol will be decoded erroneously. This happens at an error ratio of 0.25 which corresponds to the value of $E_c/N_0 = 0.1$ in Figure 5.

In the next section, we discuss how these ideas could be carried into practice by implementing a hardware prototype. One drawback of our approach of multiple bits per symbol is that it would consume more energy compared to Parks et al.'s approach since more comparators would be needed.

7 Hardware Design

Ambient backscatter platforms usually have a thresholding stage after the analog filtering and computa-

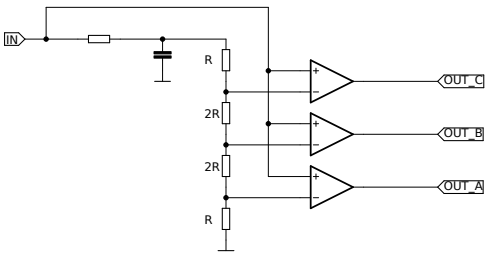


Figure 7: **Thresholding circuit for a 2-bit per symbol code.** The proposed thresholding stage uses three comparators rather than one to decode two-bits-per symbol codes while remaining backwards-compatible to single-bit encoding.

tion phase where the signal is transformed into a single bit [6, 10]. This thresholding is usually implemented with a comparator and low-pass filter. In order to make it possible to implement multiple-bit-per-symbol coding, we propose to change the existing backscatter receiver design with a thresholding stage as depicted in Figure 7. The new thresholding stage has three digital outputs, compared to just one in previous designs. This allows to receive 2-bit per symbol transmissions while remaining backwards compatible for single-bit symbols. The values of the two bits can be computed from equation 2. This computation may be performed by the platform's micro controller or by dedicated digital circuitry. Output B in Figure 7 is equivalent to the output of the single-output threshold used in previous designs.

$$\begin{aligned} O_1 &= B \\ O_0 &= \neg(B \otimes C) \wedge A \end{aligned} \tag{2}$$

8 Conclusion

In this paper, we have explored the idea of coding multiple bits per symbol in ambient backscatter communications. Our simulations exposed that the trade-off between data rate and robustness has two different manifestations: on the one hand, the trade-off is controlled by the length of chip sequences, since longer chip sequences are more robust but result in lower data rates. On the other hand, encoding more bits per symbol increases the data rate, but reduces robustness.

The central question of this paper is whether encoding multiple bits per symbol increases the throughput in ambient backscatter. In light of the results presented in Section 6, we conclude that under good channel conditions, it is favorable to decrease the length of a chip sequences (i.e., using less chips per symbol) rather than to increase the number of bits per symbol. While both approaches increase the throughput, the former provides better robustness.

The existence of the trade-offs we have found allows us to envision a system that performs dynamic rate adaptation for backscatter devices. The design and implementation of such a rate adaption could be worthwhile to explore in future work, and it can be informed by the results of our paper.

9 References

[1] Epc c1g2 standard. http://www.epcglobalinc.org/standards/uhfc1g2.

[2] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti. Backfi: High throughput wifi backscatter. *SIGCOMM Comput. Commun. Rev.*, 45(5):283–296, Aug. 2015.

[3] M. Buettner. *Backscatter Protocols and Energy-Efficient Computing for RF-Powered Devices*. PhD thesis, 2013.

[4] P. Hu, P. Zhang, and D. Ganesan. Laissez-faire: Fully asymmetric backscatter communication. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 255–267, New York, NY, USA, 2015. ACM.

[5] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 607–618, New York, NY, USA, 2014. ACM.

[6] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient Backscatter: Wireless Communication out of Thin Air. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 39–50, New York, NY, USA, 2013. ACM.

[7] V. Liu, V. Talla, and S. Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 67–78, New York, NY, USA, 2014. ACM.

[8] P. Nikitin, S. Ramamurthy, R. Martinez, and K. Rao. Passive tag-to-tag communication. In *RFID (RFID), 2012 IEEE International Conference on*, pages 177–184, April 2012.

[9] P. V. Nikitin, S. Ramamurthy, and R. Martinez. Simple low cost uhf rfid reader. In *Proc. IEEE Int. Conf. RFID*, pages 126–127, 2013.

[10] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith. Turbocharging Ambient Backscatter Communication. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 619–630, New York, NY, USA, 2014. ACM.

[11] Simon Haykin. *Communication Systems*. Wiley, Hoboken, NJ, 5 edition, Mar. 2009.

[12] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 237–248, New York, NY, USA, 2014. ACM.

[13] P. Zhang and D. Ganesan. Enabling bit-by-bit backscatter communication in severe energy harvesting environments. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI'14, pages 345–357, Berkeley, CA, USA, 2014. USENIX Association.

[14] P. Zhang, P. Hu, V. Pasikanti, and D. Ganesan. Ekhonet: High speed ultra low-power backscatter for next generation sensors. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 557–568, New York, NY, USA, 2014. ACM.

[15] Y. Zheng and M. Li. Read bulk data from computational rfids. In *INFOCOM, 2014 Proceedings IEEE*, pages 495–503. IEEE, 2014.