IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 2, 2020, accepted March 11, 2020, date of publication March 19, 2020, date of current version March 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981868

Adaptive Transmission Repetition and Combining in Bistatic WiFi Backscatter Communications

HWANWOONG HWANG AND JI-HOON YUN
Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
Corresponding author: Ji-Hoon Yun (jhyun@seoultech.ac.kr)

This work was supported by the Seoul National University of Science and Technology (SeoulTech) through the Advanced Research Project.

ABSTRACT Extracting data bits from bistatic-backscattered WiFi signals by threshold-based decoding is challenging due to severe fluctuation of OFDM signals. We propose adaptive transmission repetition to combat this problem. The key idea is to let a backscatter transmitter repeat transmissions and a receiver combines those so that signal fluctuation is filtered out due to its time-changing patterns, as observed in our testbed experiments, thus reducing bit error rate. However, excessive transmission repetitions lead to the waste of channel time, thus we propose an algorithm to adapt the number of repetitions so as to maximize the effective throughput in the present communication condition. We demonstrate via testbed experiments that the proposed algorithm adapts well to various communication conditions and achieves near-best throughput performance.

INDEX TERMS Ambient backscatter communication, WiFi backscatter, ultralow-power communication, backscatter tag, IoT.

I. INTRODUCTION

Ambient backscatter communication [1] is widely considered as a means of ultralow-power communication of low-end tag-typed devices (e.g. sensor node) with no infrastructure deployment in Internet of Things (IoT) environments. The ambient backscatter communication is realized by letting a tag device reflect and absorb ambient signals such as TV broadcast [2], FM radio [3], LoRa [4], and WiFi signals [5] in the air according to the data bits to transmit, thus not needing a dedicated signal source. Then, the amplitude, phase, or both of the carrier signal is modulated accordingly so that a receiver can decode the data bits from it.

In particular, considering ambient WiFi signals as a signal source of backscatter communication is promising since WiFi access points have widely been deployed, and thus the presence of carrier signals to backscatter is highly probable everywhere. Furthermore, most handheld devices (e.g. smartphones, tablet PCs) adopts the latest generation of WiFi. These devices can be used as a data-collecting node or an Internet gateway for WiFi backscatter tags.

Bistatic backscatter communication—a transmitter modulates the amplitude of a carrier signal into two levels (as a tag does by switching between reflection and absorption states) and a receiver detects the levels using an amplitude threshold—provides a universal transmitter and receiver framework for all communication types, i.e., downlink (gateway-to-tag), uplink (tag-to-gateway), and tag-to-tag communication, thanks to its simplicity of implementation.

However, decoding data bits from bistatic-backscattered WiFi signals is challenging since a WiFi signal itself has inherent fluctuations due to the high peak-to-average-power-ratio (PAPR) nature of orthogonal frequency-division multiplexing (OFDM). While early research works on WiFi backscatter communication [5], [6] did not tackle this problem and assumed the conventional decoding method as used for other types of backscatter communication, there have been later attempts to enhance the performance of WiFi backscatter communication. Frequency shift (FS) adopted by FS-Backscatter [7] greatly enhances the communication range of WiFi backscatter communication by avoiding the strong interference of a WiFi carrier signal at the receiver side, but at the expense of additional bandwidth usage. HitchHike [8] and FreeRider [9] enable commodity WiFi receivers to decode backscattered signals by letting a tag perform codeword translation of a received WiFi signal to convey data bits on it, but require two receivers tuned to distinct (carrier and backscattered) channels and another

The associate editor coordinating the review of this manuscript and approving it for publication was Rakesh Matam.

entity to combine these signals to finally decode the data bits. The authors of [10] proposed a pattern-matching-based decoding algorithm that identifies unique patterns of signal samples that arise from smoothing of WiFi signals to filter out noisy fluctuation. Multiple antennas were considered to improve detection performance [11]–[14]; In particular, the authors of [11] considered individual detection thresholds for different antennas and those of [14] proposed a design to cancel the direct-link interference. Kim and Lee [15] proposed a receiver design to decode backscatter signals in multiple channels. In [16], three-state switching (reflection, absorption, and negative-reflection states) was proposed. In [17], the authors introduced a maximum-likelihood detector that determines a detection threshold. There have also been many approaches to extend the capability of backscatter communication such as a fully-analog backscatter tag design for capacitive sensors [18], exploitation of multiple ambient carrier sources [19], a low-power WiFi transmitter exploiting the backscattering effect internally [20], and long-range backscatter communication based on chirp spread spectrum [21].

In this paper, we propose a new approach to combat the WiFi signal's fluctuation problem, which can also be combined with the above solutions for further enhancement, named *adaptive transmission repetition*. The key idea is to let a backscatter transmitter repeat transmissions and a receiver combines those so that signal fluctuation is filtered out due to its time-changing patterns. First, we make experimental observations on how transmission repetitions and a simple accumulative sum of received signals change the shape of the resulting signal, i.e., reducing fluctuation, but preserving backscatter patterns, and enhance bit error rate. However, increasing the number of transmission repetitions to strengthen the robustness of transmission consumes longer channel time, thus not always leading to enhanced throughput performance. To solve this problem, we design an algorithm to adapt the number of transmission repetitions so as to optimize the effective throughput in the present communication condition under asynchronous feedback of transmission results (due to the changing availability of carrier signal or channel time). We demonstrate via our prototype implementation and testbed experiments in office and hallway scenarios that the proposed design outperforms no repetition and achieves close to the best fixed-number-of-repetitions case in various conditions.

In summary, the main characteristics of the proposed scheme are listed as follows:

- Enhancement of the bit error rate (BER) by transmission repetition and simple cumulative sum of received signals that enables filtering out the WiFi carrier signal's inherent fluctuation while preserving the low complexity of the backscatter system.
- Adaptive transmission repetition to balance between robust decoding and channel-time consumption, thus achieving close to the best fixed-number-of-repetitions case in various conditions.
- Adaptation under asynchronous feedback of transmission results due to the sporadic availability of carrier signal or channel time.

The rest of the paper is organized as follows. In Section II, we present experimental observations on transmission repetition. The proposed algorithm is described in Section III. The experimental setup and performance results are shown in Section IV. Finally, Section V concludes the paper.

II. EXPERIMENTAL OBSERVATION ON TRANSMISSION REPETITION

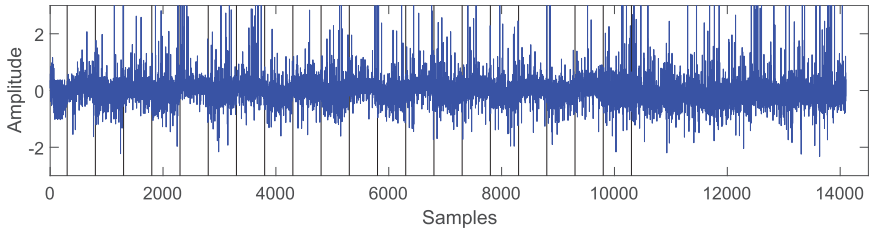
Fig. 1 shows the normalized received signals after combining different numbers of repetitions when a tag sends alternating data zero and one at 20 kbps (20 bits for a WiFi frame). Combining signals is a simple accumulative sum due to feasible implementation at tags. The detailed experimental configuration is described in Section IV.A. The figure shows that the fluctuation of the WiFi frame signal itself is severe and, with no repetition, the backscatter pattern is not easily noticeable. As the received signals are accumulated, the backscatter pattern becomes more apparent, thus leading to the expectation of lower bit error rate (BER).

The BER results of an experiment in the office scenario are given in Fig. 2 for the tag positions of Fig. 4. We consider the tag's bit rate of 200 kbps. In each tag position, we vary the number of repetitions from zero up to seven. The BER values equal to or lower than 10^{-4} are presented as 10^{-4}. Throughout all tag positions, we see that a higher number of repetitions always yields lower BER. Different tag positions result in different BER values; typically, as the tag gets farther from both the transmitter and receiver, i.e., when it moves from A to D, it has higher BER values. Therefore, in order to achieve a certain level of BER, the tag needs different numbers of repetitions in different positions. A similar trend is also observed in the experimental results of the hallway scenario as shown in Fig. 3.

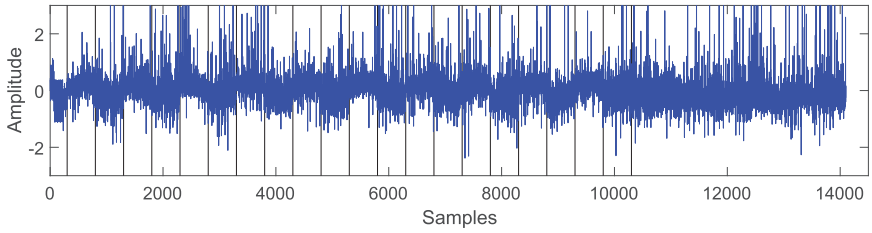
As mentioned earlier, however, transmission repetition consumes additional channel time. Table 1 illustrates that the maximum achievable throughput decreases fast as the number of transmission repetitions increases. Therefore, it is desirable for a backscatter transmitter to spatio-temporally adjust the number of transmission repetitions so as to maximize the effective throughput.

III. REPETITION ADAPTATION ALGORITHM

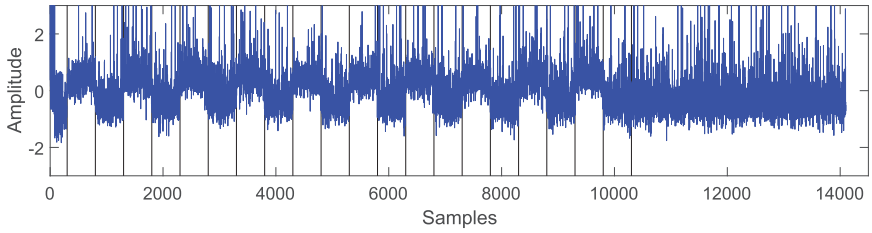
Based on the observations made in the previous section, we design an algorithm to adapt the number of transmission repetitions. The pseudo-code of the algorithm is given in Algorithm 1. We define a set of stages for each of which a specific number of transmission repetitions, denoted by N_i for stage i, are made for each packet, while $N_i > N_{i+1}$ for all i. Let R_i be the maximum achievable throughput of stage i. As the number of repetitions increases, longer channel time is used and thus the maximum achievable throughput always decreases. Therefore, we have $R_i < R_{i+1}$ for all i. Table 1 shows an example of stage definition, which will also be



(a) No repetition



(b) One repetition



(c) Ten repetitions

FIGURE 1. Normalized combined signals with different numbers of transmission repetitions.

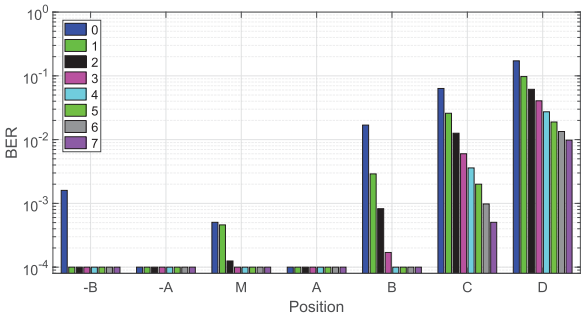


FIGURE 2. Bit error rate in different tag positions with a varying number of transmission repetitions in the office scenario.

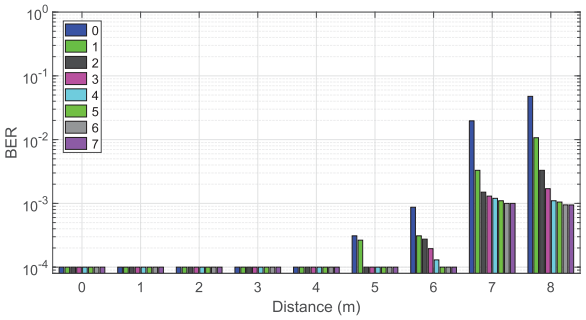


FIGURE 3. Bit error rate in different tag positions with a varying number of transmission repetitions in the hallway scenario.

TABLE 1. Stage definition example of the adaptation algorithm for first scenario.

Stage	1	2	3	4	5	6	7	8
Number of repetitions	7	6	5	4	3	2	1	0
Max throughput (kbps)	1.25	1.428	1.667	2	2.5	3.34	5	10

used in the evaluation section. The algorithm starts from stage one, which is the lowest maximum-throughput, but with the highest robustness. Then, the algorithm has to keep making the following decisions during its operation:

- When to make a stage transition;
- Which stage to transition to.

In what follows, we describe the algorithm design for stage transition.

A transition to a higher stage (i.e., less repetitions) is triggered when S consecutive packets are transmitted successfully, based on the expectation that less repetitions may also achieve successful transmission. To realize this behavior, the algorithm manages the consecutive success count s, increasing it by one when a packet is transmitted successfully, but resetting it to zero if decoding of a packet finally fails over all repetitions. When s gets equal to S, the algorithm changes the current stage to a higher one.

Selection of the higher stage to transition to is designed as follows. Whenever a receiver receives a repeated transmission

Algorithm 1 Repetition Adaptation Algorithm

```
s: consecutive-success count
f: consecutive-failure count
i: current stage (1 ≤ i ≤ I)
Initialize s and f as zero
Initialize i as one
for all packets to transmit do
    for k ∈ {1, ..., N_i} do
        Transmit the packet
        if Decoding succeeds then
            s ← s + 1, f ← 0
            m(s) ← k
            Go to DECODE_DONE
        end if
    end for
    if Decoding fails then
        f ← f + 1, s ← 0
    end if
    DECODE_DONE:
    if s ≥ S then
        m* ← max{m(1), ··· , m(S)}
        i ← max{j|N_j ≥ m*, 1 ≤ j ≤ I}
        s ← 0
    else if f ≥ F then
        i ← max{i − 1, 1}
        f ← 0
    end if
end for
```

signal of a packet, it adds the signal onto and attempts to decode data bits from the accumulated signal-sum of the packet. Assume that the transmitter is in stage i, performing N_i repetitions in total for each packet. If the receiver succeeds in decoding of a packet when $n(\leq N_i)$ repetitions are made, this implies that it may not need more repetitions than n in the next time. Thus, the receiver records the minimum number of repetitions that were needed for the successful transmission of each packet as $m(s)$ for sth packet ($s = 1, \cdots, S$) before increasing the stage. Finally, the algorithm concludes that at least m^* repetitions are needed in the next stage where $m^* = \max_{1\leq s\leq S} m(s)$, which is a conservative choice of repetitions that made the latest S packets all transmitted successfully. When the success count s reaches S, the algorithm determines the next stage as the highest one (least repetitions) having the number of repetitions equal to or higher than m^*. That is, the algorithm accelerates the transition to a properly-high stage rather than simply moving to the next higher stage. For example, suppose that a tag is in stage three (five repetitions) of Table 1, transmits four packets (each for five times) and a receiver decodes each packet successfully after two ($m(1)$), two ($m(2)$), three ($m(3)$) and two ($m(4)$) repetitions, respectively, having m^* as three. If S is four, the consecutive success count reaches S and the algorithm moves to stage five having three repetitions.

A transition to a lower stage (i.e., more repetitions) is designed differently. If the transmitter is in stage i, performing N_i repetitions for each packet and the receiver fails to decode a packet after combining N_i receptions, the receiver increases the consecutive failure count f by one (resetting the consecutive success count s to zero). When f reaches F, the algorithm assumes that more repetitions are required to protect its packet reception, and moves to the next lower stage. Since the algorithm has no knowledge of the number of additional repetitions that are needed to make a packet reception successful, a stage decrease is made in a conservative manner.

The remaining design problem is feedback, which also depends on who runs the algorithm. If a transmitter runs the algorithm, then a receiver needs to feed back the results of previous transmissions to the transmitter so that the transmitter determines the stage accordingly. In WiFi backscatter communication, however, the transmission of feedback may not always be immediate due to the changing availability of carrier signal or channel time.[1] Therefore, when the receiver sends feedback information to the transmitter, it sends all transmission results that have not been fed back yet. Then, the transmitter considers all fed-back transmission results to determine the current stage.[2] But, in this case, the amount of feedback increases as the delay of feedback increases. Alternative is to let the receiver run the algorithm and give the explicit feedback of the stage to use to the transmitter. That is, the receiver tracks stage changes along with the success and failure counts. Since the feedback information is limited to the single value of the current stage, the amount of feedback remains minimal and constant regardless of the delay of feedback.

In practical deployment scenarios, there may exist contention and collision issues in two cases: (1) between tags (inter-tag case); (2) with other non-carrier WiFi signals. There have been a number of proposals to resolve contention between multiple RFID tags [22]–[25]. Some attempts have also been made for ambient backscatter communication [2], [26]. The proposed scheme can be combined with a wide range of such proposals, thus still benefiting from transmission repetition for robust communication in multi-tag environments. However, the collision problem with other non-carrier WiFi signals, which in particular happens with frequency shift [7], has not been explored well in the literature; When the frequency of a backscattered signal is shifted to an adjacent channel, it is vulnerable to collision with the WiFi transmitters of the channel due to the weak energy. We leave this as our future work.

[1] If a feedback transmitter is a backscatter node (i.e., tag), it can send feedback information on carrier signals only. If a feedback transmitter has the capability of signal generation by itself (i.e., gateway), it needs to obtain a channel-access opportunity to send feedback information.

[2] It is desirable to suppress multiple stage decreases at once since the effectiveness of an increased number of repetitions has not be tested yet.

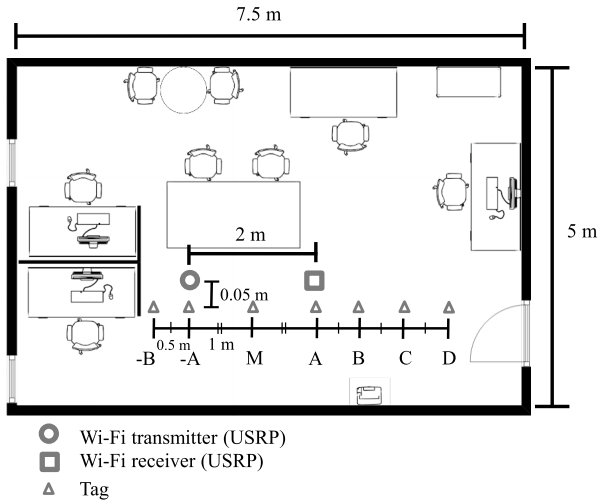


FIGURE 4. Office scenario: experimental setup and tag positions.

IV. PERFORMANCE EVALUATION

In this section, we prototype and evaluate the proposed algorithm in two experimental scenarios: office and hallway.

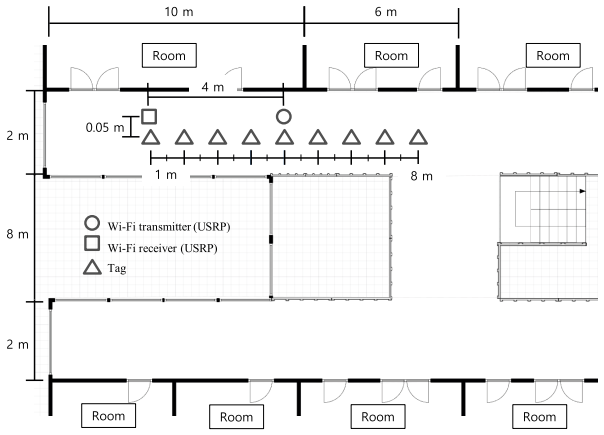


FIGURE 5. Hallway scenario: experimental setup and tag positions.

A. ENVIRONMENTAL SETUP

Fig. 4 shows the experimental setup of the office scenario. We use two Ettus USRPs (N210 and N310) [27] to generate and receive WiFi signals, respectively, which are separated by two meters. Seven different tag positions are considered, each depicted as a triangle in the figure. Position M is taken 5 cm apart from the middle point between the two USRPs. Positions A and -A are one meter apart from Position M. Other positions are separated by 0.5 m from each's adjacent one. Every position is 0.5 m high from the ground. The hallway scenario is illustrated in Fig. 5; two USRPs are deployed four meters apart. A tag is first placed together with the receiver USRP and moved towards the transmitter USRP with one-meter steps until getting apart from the receiver USRP by eight meters.

The gr-ieee802-11 module of GNU Radio [28] is used to generate WiFi carrier signals according to IEEE 802.11g's OFDM frame format. The transmitter USRP generates and transmits a WiFi frame every 10 ms. Each WiFi frame is composed of a 64 μs physical-layer header and a following MAC frame of 1528 bytes. It is transmitted at a bit rate of 9 Mbps (QPSK modulation and 3/4 code rate), which is 1.4 ms long. The USRPs operate at the center frequency of 2.432 GHz. The receiver USRP captures the I/Q signals of received WiFi frames via two antennas, each at the sampling rate of 10 MHz (for moderate computational loads) and pre-processes them according to the μMO procedure [2] so that the communication distance is expanded thanks to diversity. The threshold to differentiate data one and zero is set as the average amplitude of a received WiFi frame. The backscatter tag is implemented using Analog Devices' RF switch ADG902 [29] for reflection and absorption of ambient WiFi signals. The information bits are generated as alternating one and zero and coded using FM0 coding [30]. Then, the backscatter tag transmits coded bits at 200 kbps. Each packet of the tag is composed of 50 information bits and two packets can be transmitted over a WiFi carrier frame. In the hallway scenario, the frequency-shift technique [7] is applied to extend the communication range. The repetition adaptation algorithm uses the stage definition of Table 1. The total number of WiFi frames generated for each experiment is 500 where each of received WiFi frames is represented as 14,000 signal samples. The performance metrics are the bit error rate (BER) of coded bits and the throughput of information bits.

B. PERFORMANCE EVALUATION RESULTS

First, the throughput performance of the fixed-number-of-repetitions (fixed-repetition for short) cases and the proposed adaptation algorithm at various distances is shown in Fig. 6. Throughput is measured while moving the tag from Position A (0 m) to D (1.5 m). Throughput at 0 m can be seen as the maximum achievable throughput since the BER at this position (A) is close to zero as seen in Fig. 2. A higher number of repetitions in this position lowers throughput since BER is already close to zero and more repetitions increase the consumption of channel time only. As the distance gets longer, the throughput of all repetition cases decreases gradually. However, a larger number of repetitions results in slower throughput decrease, thus having a longer communication range. This is because, as the distance gets longer, the gain resulting from repetitions increases more than the penalty of longer channel-time consumption. The throughput of the proposed adaptation algorithm (depicted as a red curve) is shown to closely follow the envelope of the fixed repetition cases. However, for some points, the algorithm achieves lower throughput than the best fixed-repetition case (e.g. for the distance of 0.1 to 0.3 m, the algorithm is worse than the zero-repetition case). This results from occasional stage changes (either increase or decrease) and delayed return to the best stage.

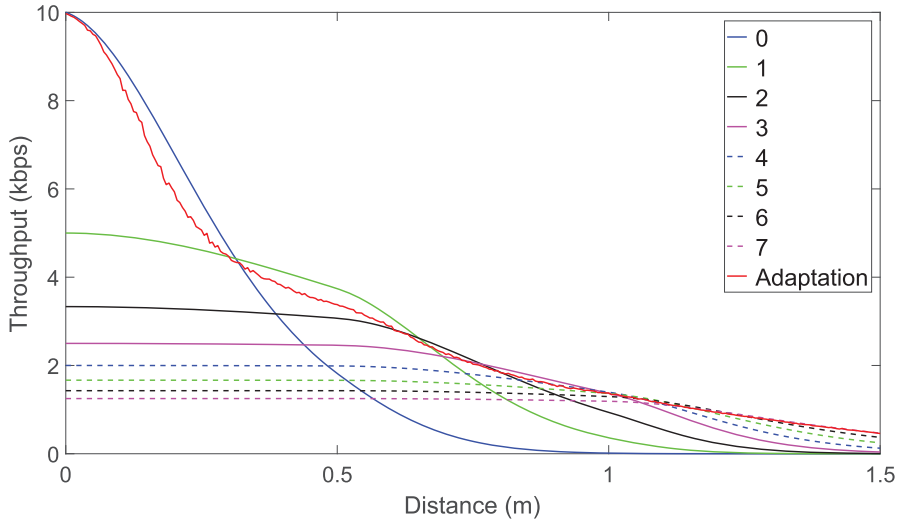
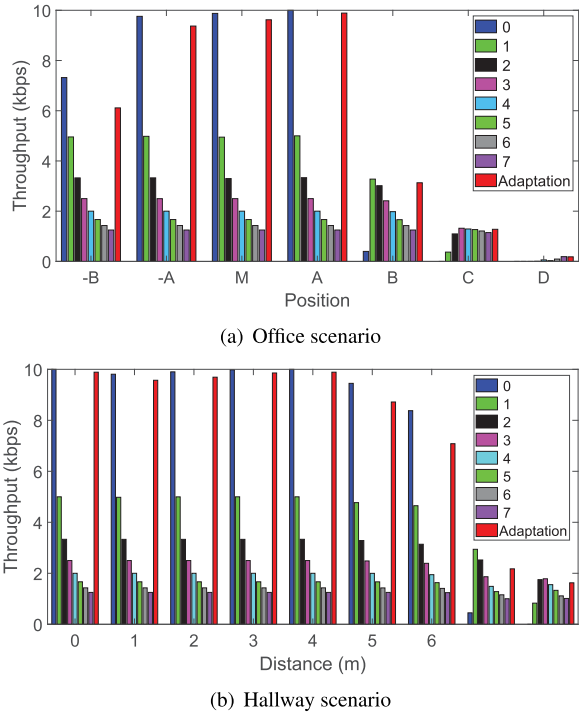


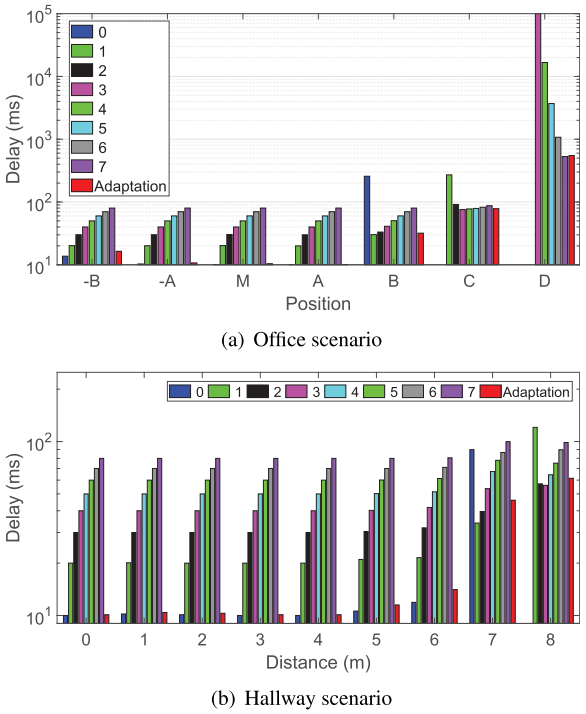
FIGURE 6. Throughput of fixed and adaptive repetition for varying tag distance.



(a) Office scenario

(b) Hallway scenario

FIGURE 7. Throughput of fixed and adaptive repetition for various tag positions in office and hallway scenarios.



(a) Office scenario

(b) Hallway scenario

FIGURE 8. Delay of fixed and adaptive repetition for various tag positions in office and hallway scenarios.

The average throughput results in all positions of the two experimental scenarios are shown in Fig. 7. As shown in the figure, the adaptation algorithm (red bar) achieves close to the best fixed-repetition case in all positions while there exists no single best one of fixed repetition. However, the throughput of the adaptation algorithm is slightly lower than that of the best fixed-repetition case due to the overhead of stage changes (waiting time before making a stage-change decision and sometimes wrong stage decisions). In the positions except -B of the office scenario (Fig. 7(a)), the throughput difference of the adaptation algorithm to the best fixed-repetition case is limited; it is less than 5% in the other positions. In Position -B, the difference is as high as 16%, but the algorithm achieves 23% higher throughput than the second best case. In the hallway scenario (Fig. 7(b)), the throughput difference of the adaptation algorithm to the best fixed-repetition is less than 10% in most positions. At 6 m (from the receiver), the adaptation algorithm achieves 15% lower throughput than

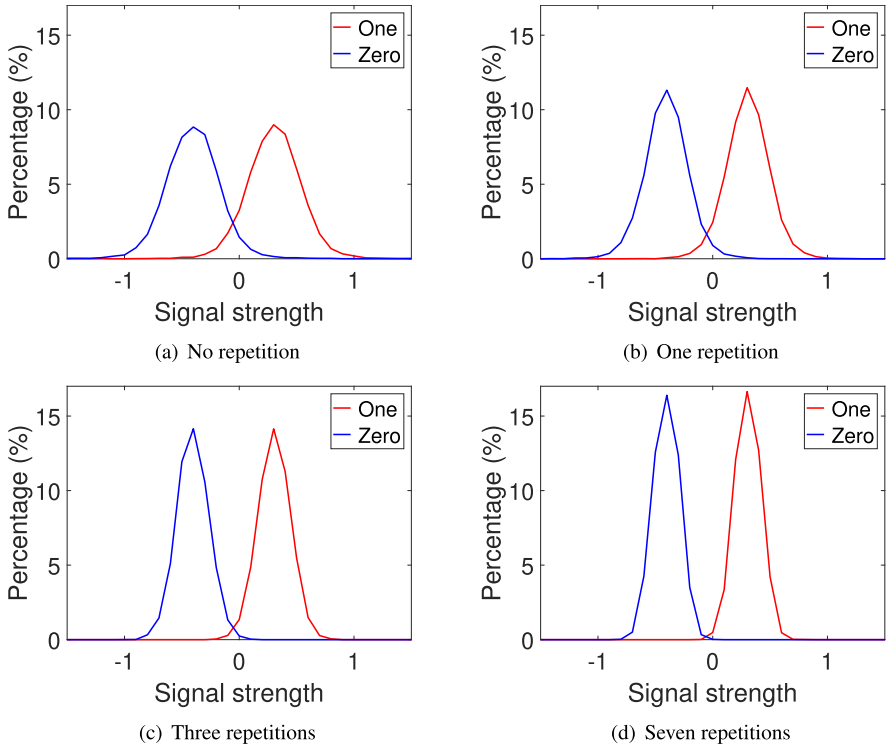


FIGURE 9. Distribution of normalized signal strength for a varying number of repetitions in Position C.

the best fixed-repetition case, but 52% higher throughput than the second best one. At 7 and 8 m, the adaptation algorithm is the third best one since the first and second best cases have similar throughput with each other.

The average delay results of both scenarios are shown in Fig. 8 where the delay is measured as the period from the time when a packet is first considered for transmission to the time when it is successfully decoded by the receiver. If a packet is not successfully transmitted by a configured number of transmission repetitions, the receiver flushes out the buffer of the accumulated signal sum of the packet and the tag retransmits (restarts transmission repetitions of) the packet. When BER is low (-B to A in the office scenario and 0 to 6 m in the hallway scenario), as the number of repetitions increases, the channel time used for a packet gets increased proportionally and so does the delay. However, with high BER (B to D in the office scenario and 7 to 8 m in the hallway scenario), a small number of repetitions does not benefit sufficiently from repetitions and results in long delay. In particular, the zero-repetition case has to retransmit a packet until a single transmission of the packet has no bit error, thus suffers long delay under high BER. In Fig. 8(a), the zero-repetition case for Position C and zero-to-two repetition cases for Position D are not shown since they fail to transmit any packet during the experiment. Likewise, the zero-repetition case for 8 m is not shown in Fig. 8(b).

In Fig. 9, the distributions of normalized signal strength for different repetitions in Position C are shown. Signal-strength values are normalized so as to have zero mean and unit variance. Thus, signal-strength zero can be considered as the decoding threshold. With no repetition, the signal-strength distributions of different bit data have much overlap with each other. Some signal-strength values of data one and zero appear negative and positive, respectively, thus leading to wrong decoding results and high BER. As more repetitions are applied, however, data one and zero have more distinct signal-strength distributions. Finally, with seven repetitions, almost all signal-strength values of data one are positive, and most values of data zero are negative; The signal-strength distributions of data one and zero have little overlap with each other.

TABLE 2. Power consumption and energy efficiency comparison.

	Peak power	Bit rate	Bits/μJ
WiFi (CC3200)	824.4mW	54 Mbps	65.5
BLE (CC2650)	18.3mW	1 Mbps	54.6
ZigBee (CC2620)	18.3mW	250 kbps	13.7
WiFi backscatter	24.2μW	100 kbps	4132.2

C. POWER CONSUMPTION

Table 2 shows the peak power consumption (i.e., power consumption during transmission) and energy efficiency (bits/μJ) of wireless LAN/PAN radio modules and ours.[3]

[3]We use the same hardware architecture and parts as [8].

The values of WiFi, BLE and ZigBee are obtained from the specification documents of Texas Instruments' corresponding radio modules, CC3200 [31], CC2650 [32], and CC2630 [33], respectively. The WiFi backscatter module has the lowest data bit rate among all, but its power consumption is considerably lower than the others. Thus WiFi backscatter significantly improves energy efficiency, e.g. 63× higher than WiFi. If a successful delivery of a packet requires seven transmission repetitions (the highest number of repetitions considered in the experiments) by the proposed scheme due to a poor channel condition, thus consuming 7× power, the energy efficiency is still significantly higher than the other modules (e.g. 9× higher than WiFi).

V. CONCLUSION

In this paper, we proposed a repetition adaptation algorithm for bistatic backscatter communications using ambient WiFi signals. The proposed algorithm adapts the number of transmission repetitions such that BER is enhanced at the lowest consumption of channel time, thus always achieving high throughput in diverse communication conditions. The experimental study with real WiFi traffic demonstrated that the proposed algorithm achieves close to the best throughput performance in all considered conditions. The proposed algorithm can be combined with or extended for other techniques of a longer communication range, higher-order modulation, etc. to achieve higher overall performance.

A number of issues remain for future work. They include more efficient encoding and decoding mechanisms such as OFDM and OFDMA, exploitation of multiple carrier sources, and also resolution of contention and collision between tags as well as with other non-carrier WiFi signals (especially when frequency shift is applied).

REFERENCES

[1] N. Van Huynh, D. T. Hoang, X. Lu, D. Niyato, P. Wang, and D. I. Kim, "Ambient backscatter communications: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2889–2922, 4th Quart., 2018.

[2] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith, "Turbocharging ambient backscatter communication," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 619–630, Aug. 2014.

[3] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, "FM backscatter: Enabling connected cities and smart fabrics," in *Proc. USENIX NSDI*, 2017, pp. 243–258.

[4] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," *ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–24, Sep. 2017.

[5] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi backscatter: Internet connectivity for RF-powered devices," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 607–618.

[6] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, "BackFi: High throughput WiFi backscatter," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 283–296, Aug. 2015.

[7] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proc. Conf. ACM SIGCOMM Conf.*, 2016, pp. 370–383.

[8] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "HitchHike: Practical backscatter using commodity WiFi," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM SenSys*, Nov. 2016, pp. 259–271.

[9] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "FreeRider: Backscatter communication using commodity radios," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, 2017, pp. 389–401.

[10] H. Hwang, J.-H. Lim, J.-H. Yun, and B. Jeong, "Pattern-based decoding for Wi-Fi backscatter communication of passive sensors," *Sensors*, vol. 19, no. 5, p. 1157, 2019.

[11] C.-H. Kang, W.-S. Lee, Y.-H. You, and H.-K. Song, "Signal detection scheme in ambient backscatter system with multiple antennas," *IEEE Access*, vol. 5, pp. 14543–14547, 2017.

[12] W.-S. Lee, C.-H. Kang, Y.-K. Moon, and H.-K. Song, "Determination scheme for detection thresholds using multiple antennas in Wi-Fi backscatter systems," *IEEE Access*, vol. 5, pp. 22159–22165, 2017.

[13] G. Yang, Y.-C. Liang, R. Zhang, and Y. Pei, "Modulation in the air: Backscatter communication over ambient OFDM carrier," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1219–1233, Mar. 2018.

[14] G. Yang and Y.-C. Liang, "Backscatter communications over ambient OFDM signals: Transceiver design and performance analysis," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[15] T. Kim and W. Lee, "Channel independent Wi-Fi backscatter networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 262–270.

[16] Y. Liu, G. Wang, Z. Dou, and Z. Zhong, "Coding and detection schemes for ambient backscatter communication systems," *IEEE Access*, vol. 5, pp. 4947–4953, 2017.

[17] G. Wang, F. Gao, R. Fan, and C. Tellambura, "Ambient backscatter communication systems: Detection and performance analysis," *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4836–4846, Nov. 2016.

[18] S.-N. Daskalakis, S. D. Assimonis, E. Kampianakis, and A. Bletsas, "Soil moisture scatter radio networking with low power," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 7, pp. 2338–2346, Jul. 2016.

[19] C. Yang, J. Gummeson, and A. Sample, "Riding the airways: Ultra-wideband ambient backscatter via commercial broadcast systems," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[20] B. Kellogg, V. Talla, J. R. Smith, and S. Gollakot, "Passive Wi-Fi: Bringing low power to Wi-Fi transmissions," in *Proc. NSDI*, 2016, pp. 151–164.

[21] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt, "LoRea: A backscatter architecture that achieves a long communication range," in *Proc. ACM SenSys*, 2017, pp. 1–14.

[22] A. A. Mbacke, N. Mitton, and H. Rivano, "A survey of RFID readers anticollision protocols," *IEEE J. Radio Freq. Identif.*, vol. 2, no. 1, pp. 38–48, Mar. 2018.

[23] M. Chen, S. Chen, Y. Zhou, and Y. Zhang, "Identifying state-free networked tags," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1607–1620, Jun. 2017.

[24] L. Chen, I. Demirkol, and W. Heinzelman, "Token-MAC: A fair MAC protocol for passive RFID systems," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1352–1365, Jun. 2014.

[25] Z. Huang, R. Xu, C. Chu, Z. Li, Y. Qiu, J. Li, Y. Ma, and G. Wen, "A novel cross layer anti-collision algorithm for slotted ALOHA-based UHF RFID systems," *IEEE Access*, vol. 7, pp. 36207–36217, 2019.

[26] T. Kim and D. Kim, "Multi-dimensional sparse-coded ambient backscatter communication for massive IoT networks," *Energies*, vol. 11, no. 10, p. 2855, 2018.

[27] Ettus Research. *USRP Networked Series*. Accessed: Mar. 19, 2020. [Online]. Available: http://ettus.com/product-categories/usrp-networked-series/

[28] *GNU Radio*. Accessed: Mar. 19, 2020. [Online]. Available: https://www.gnuradio.org

[29] *Analog Devices, ADG902*. Accessed: Mar. 19, 2020. [Online]. Available: http://www.analog.com/en/design-center/evaluation-hardware-and-software/eval%uation-boards-kits/eval-adg902.html.

[30] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: Wireless communication out of thin air," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 39–50, Aug. 2013.

[31] Texas Instruments. *CC3200, SimpleLink Wi-Fi and Internet-of-Things Solution*. Accessed: Mar. 19, 2020. [Online]. Available: https://www.ti.com/product/cc3200.

[32] Texas Instruments. *CC2650, SimpleLink Multi-Standard 2.4 GHz Ultra-Low Power Wireless MCU*. Accessed: Mar. 19, 2020. [Online]. Available: http://www.ti.com/product/cc2650

[33] Texas Instruments. *CC2630, SimpleLink 6LoWPAN, ZigBee Wireless MCU*. Accessed: Mar. 19, 2020. [Online]. Available: https://www.ti.com/product/cc2630

HWANWOONG HWANG received the B.S. and M.S. degrees in electrical and information engineering from the Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electrical and information engineering. His current research focuses on wireless communications for the Internet of Things (IoT) devices.

JI-HOON YUN received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University (SNU), Seoul, South Korea, in 2000, 2002, and 2007, respectively.

From 2007 to 2009, he was a Senior Engineer with the Telecommunication Systems Division, Samsung Electronics, Suwon, South Korea. He was a Postdoctoral Researcher with the Real-Time Computing Laboratory, University of Michigan, Ann Arbor, MI, USA, in 2010. He is currently an Associate Professor with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea. Before joining SeoulTech, in 2012, he was with the Department of Computer Software Engineering, Kumoh National Institute of Technology, as an Assistant Professor. His current research focuses on mobile networking and computing.