WiTAG: Rethinking Backscatter Communication for WiFi Networks

Ali Abedi
University of Waterloo

Mohammad Hossein Mazaheri
University of Waterloo

Omid Abari
University of Waterloo

Tim Brecht
University of Waterloo

ABSTRACT

WiFi-based backscatter systems provide the potential to deliver battery-free sensors (tags) which can transmit data using a WiFi network. Existing backscatter systems have several problems which make them impractical to deploy and operate using existing WiFi networks. First, they require software or hardware modifications to WiFi access points and devices. Second, they do not work with WiFi networks that use a security protocol such as WPA. Third, they interfere with existing WiFi communication because they reflect their signal to another channel without implementing channel sensing. In this paper, we present WiTAG which addresses these problems, making the implementation and deployment of backscatter systems significantly more practical. In contrast with existing systems that build tags to communicate using the physical layer, we take a radically different approach by building tags that leverage features of the MAC layer to communicate. We design tags which can selectively interfere with subframes (MPDUs) in an aggregated frame (A-MPDU). This enables standard compliant communication using modern 802.11n and 802.11ac networks with minimal infrastructure and without requiring hardware or software modifications to any devices. The evaluation of our prototype system shows that with a client and an access point that are 8 meters apart, a tag can achieve data rates of 40 Kbps when located anywhere between the two devices.

1 INTRODUCTION

Backscatter systems are very attractive as a means of communication for wireless sensors in applications ranging from implantable body sensors to farm monitoring [3, 7, 10, 18]. This is because of their low cost, small form factor and ease of maintenance since they do not require batteries.

Traditional backscatter systems such as RFIDs require a specialized reader to read the tag values. The high cost and large form factor of these readers have made them difficult to deploy and have limited the adoption of RFID tags in many applications. To overcome these challenges, researchers have recently designed WiFi-based backscatter systems [16, 19]. Their vision is to design backscatter tags which can be read using WiFi devices to potentially reduce the complexity and cost of deploying these systems by using deployed WiFi infrastructures instead of specialized readers. To realize this vision, WiFi-based backscatter systems need to satisfy the following key requirements:

- WiFi Compatible: They should be compatible with already deployed commodity access points (APs) without requiring any hardware or software changes. Specifically, they should be compatible with the newest 802.11n and 802.11ac WiFi standards.
- Work with Encryption: Due to security and privacy concerns the system must work with WiFi networks that use WPA or WEP encryption.
- Low-Power: Similar to traditional RFID tags, these tags need to be extremely low-power so that they can harvest their energy from the environment and operate without requiring a battery.
- Non-Interfering: They should not create any interference for other WiFi devices existing in the network. Due to the low power requirements of backscatter tags, prior to transmitting (reflecting) their signal, they can not afford to perform carrier sensing. Therefore, tags must use a new communication mechanism which does not interfere with other WiFi devices.

If the above requirements are satisfied, we can envision having low-power, battery-free wireless sensors which do not require specialized readers and can be deployed in environments with already-deployed WiFi access points.

However, to the best of our knowledge, no current backscatter system satisfies all of these requirements. Recent systems require specialized hardware which does not work with commodity WiFi access points (e.g., BackFi [2]). HitchHike [16] works only with 802.11b networks, which are becoming obsolete. The systems presented in FreeRider [17] and MOXcatter [19] work with 802.11g and 802.11n, respectively, but still require modifying existing APs. Furthermore, almost all

HotNets-XVII, November 15–16, 2018, Redmond, WA, USA

ACM ISBN 978-1-4503-6120-0/18/11…$15.00
https://doi.org/10.1145/3286062.3286084

of these systems reflect the signals onto an adjacent channel without performing carrier-sensing. Therefore, their signals interfere with other WiFi devices operating on that adjacent channel. Most importantly, because state-of-the-art approaches modify physical-layer symbols, they do not work if the network uses encryption.

In this paper, we present WiTAG, a novel WiFi-based backscatter system that takes a radically different approach by altering the wireless channel to communicate data by leveraging MAC-layer features to satisfy all of the above requirements. Specifically, we make the following technical contributions:

1) We have designed WiTAG, a novel backscatter communication system that enables communication between battery-free tags and WiFi devices. Specifically, WiTAG is fully compatible with existing 802.11n and ac WiFi networks (no modifications to devices are required). WiTAG achieves this compatibility by utilizing MAC-layer "frame aggregation" available in 802.11n and 802.11ac standards. These standards place multiple MAC-layer data units (subframes) in a large PHY-layer packet (aggregated frame), as shown in Figure 1. A WiFi device sends a special aggregated frame to an AP in the network using a physical rate that is likely to be successfully received. This frame acts as a query for the tag. In WiTAG the tag embeds its data into the frame by selectively corrupting some subframes of the frame. Then, the AP sends a block ACK back to the transmitting WiFi device, indicating which subframes have and have not been successfully received. That device extracts the data being communicated by the tag from the bits in the block ACK. Subframes that are not corrupted by the tag are received successfully (represented by a 1 in the block ACK) and those that are corrupted are not received successfully (represented by a 0 in the block ACK). Note that the AP is completely oblivious to the existence of the tag and does not require any modification. Further, because the tag communicates its data by selectively corrupting subframes in query frames, it does not reflect signals onto a secondary channel and does not interfere with other WiFi devices in the network. Most importantly, because tags communicate by corrupting encrypted or unencrypted MAC-layer subframes WiTAG works with networks that use encryption.

2) We have implemented and tested our novel low-power backscatter technique to show that it can be used to selectively corrupt specific subframes in an aggregated frame. Note that an active radio which has a transmitter can easily corrupt a subframe by transmitting an interfering signal. However, a backscatter tag is a passive device and does not have a transmitter. To solve this problem, we design a backscatter tag which can modify the wireless channel during the transmission of a subframe, hence corrupting that subframe. Because frame aggregation performs channel estimation only once at the beginning of the aggregated frame, modifying the channel for a short period of time during the transmission of a subframe ensures that the subframe can not be decoded. We have implemented and evaluated WiTAG in both line of sight and non line of sight scenarios. Our results show that with a client (querying) device, placed 8 meters from the access point, our tag can achieve a rate of 40 Kbps when located anywhere between the two devices.

2 RELATED WORK

Backscatter communication systems have gained significant attention in the recent years [6, 7]. These systems typically require a specialized reader to generate the trigger signal and to receive the backscattered data. The high cost and large form factor of these readers have made them difficult to deploy and have limited the adoption of RFID tags in many applications. More recent work such as WiFi backscatter [8], BackFi [2] and Passive WiFi [9] have eliminated the need for specialized readers by utilizing WiFi devices instead. However, all of these systems still require either specialized hardware to trigger the tag or modifications to WiFi devices to receive the tag's signal [16, 17]. Therefore, deploying these systems requires modifications to existing WiFi infrastructures which is costly and impractical.

A new category of backscattered systems has been explored recently that utilizes only commodity WiFi devices. In HitchHike [16], a WiFi device transmits an 802.11b packet that is received by an access point (AP 1) and a tag. The tag embeds its data in the packet by changing the transmitted 802.11b symbols to other valid symbols. The tag also shifts the signal to a non-overlapping channel where another access point (AP 2) receives the backscattered signal. Finally, AP 1 and 2 transfer the received packets to a host where the original and backscattered packets are compared in order to extract the data embedded by the tag.

Although HitchHike utilizes commodity WiFi devices, it is not compatible with existing WiFi networks for a number of reasons: 1) HitchHike can not be used with networks that use security protocols such as WPA and WEP because after HitchHike modifies existing symbols in the encrypted packet, it can no longer be decrypted. 2) The CRC of backscattered packets fail due to modifying the packet payload. Access points would normally drop such packets assuming that these packets are corrupted. Therefore, HitchHike requires modifications to the access points to make sure they do not drop these packets. 3) HitchHike only works with 802.11b networks, while most of today's WiFi networks are 802.11n and ac. 802.11b devices use direct-sequence spread spectrum (DSSS) communication scheme which is fundamentally different from the frequency-division multiplexing (OFDM) scheme used in 802.11n and ac networks. 4) In order to decode the backscatter packets, HitchHike requires receiving both the original and backscatter signal. Therefore, it requires an additional access point.

More recent work like FreeRider [17] and MOXcatter [19], propose using similar backscatter communication systems for 802.11g and 802.11n standards. In order for the tag to work with 802.11g networks, FreeRider changes an OFDM symbol to another valid OFDM symbol by changing the phase of the

signal. For example, no phase offset represents zero and a 180 degree phase offset represents one. MOXcatter proposes a backscatter system that works with WiFi networks that utilize MIMO communication such as 802.11n and ac. Due to the complexity of MIMO signals MOXcatter cannot perform the phase offset on individual OFDM symbols. Therefore, to transmit 0s and 1s, it changes the phase of the signal for each packet instead. Since FreeRider and MOXcatter reflect the original signal to a secondary channel, they have the same compatibility limitations and shortcomings as HitchHike.

In addition to these compatibility limitations, FreeRider, MOXcatter, and HitchHike also suffer from carrier sensing and power consumption shortcomings. Due to their power constraints, these tags cannot afford to implement channel sensing operations. Therefore, they may cause interference for other WiFi devices operating on the secondary channel. Moreover, the secondary channel has to be at least 20 MHz from the primary channel to avoid self-interference. As a result, these tags require their hardware (oscillators and RF switches) to operate at 20 MHz or higher frequency in order to shift the backscattered signal to a non-overlapping channel, significantly increasing power consumption [18].

In contrast with previous systems that modify physical-layer symbols, WiTAG alters the wireless channel to communicate data by leveraging MAC-layer features. As a result, WiTAG is compatible with existing WiFi networks, requires no modifications to access points, and works with both open and encrypted networks. In addition, because WiTAG does not backscatter to a secondary channel, it does not require channel sensing or high-frequency components, resulting in significantly lower power consumption than previous systems.

3 BACKGROUND

WiTAG takes advantage of MAC-layer frame aggregation (combining several subframes to create larger frames) to communicate data by briefly altering the wireless channel. Therefore, we now provide background on frame aggregation and PHY layer channel estimation and correction.

3.1 802.11 Frame Aggregation

The IEEE 802.11n and ac standards provide a frame aggregation mechanism to improve the efficiency of the MAC layer [4, 5]. In order to avoid overheads such as performing channel sensing and transmitting an acknowledgment per frame, multiple *MAC Protocol DATA Units* (MPDUs) are combined into a larger aggregated frame (A-MPDU), as illustrated in Figure 1. By aggregating multiple subframes, the overhead is amortized over more useful bits and therefore the efficiency of the MAC layer improves significantly. The receiver of an A-MPDU transmits a *block ACK* back to the sender which reports the fate of the individual subframes inside the A-MPDU. A block ACK is similar to the legacy 802.11 acknowledgment, however rather than acknowledging the successful reception of one frame, it reports the fate of each MPDU using a 64-bit bitmap. WiTAG leverages this

Figure 1: 802.11n/ac A-MPDU structure

frame aggregation scheme and selective block ACKs to allow tags to transmit data to a WiFi device.

3.2 Channel Estimation and Correction

In all 802.11 standards including n and ac, the PHY header starts with multiple known training symbols. The receiver utilizes these symbols to perform channel estimation [4, 5]. Since known symbols are transmitted over all OFDM subcarriers, the receiver can estimate the phase and amplitude per subcarrier. This is known as Channel State Information (CSI). Then, the receiver uses the estimated CSI to correct the received signal during the transmission of subframes. In Section 5, we explain how WiTAG alters the wireless channel for a short period of time during the transmission of a subframe, resulting in a subframe that can not be decoded.

4 DESIGN OVERVIEW

WiTAG is a WiFi-based backscatter system for existing WiFi networks. It enables a battery-free tag to communicate data to existing WiFi devices without requiring any modifications. Figure 2 shows WiTAG's architecture. At a high level, a client WiFi device (laptop or cellphone) sends a packet to an existing access point (AP). This packet acts as a query for the tag. The tag detects the starting point of the packet and embeds its data into the packet. The AP receives the packet and transmits the block ACK for the packet to the client. Note that since the sole purpose of query packets is to enable the tag to transmit its data, the client does not use their content for communication.

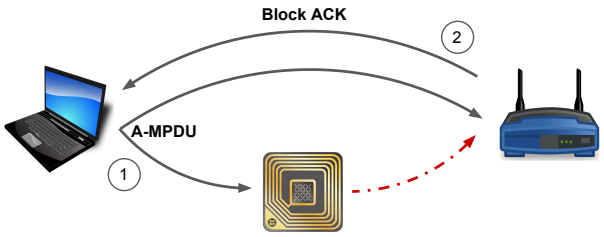

Figure 2: Design overview

Figure 2 shows that WiTAG operates in two steps. In the first step (labelled 1), the client device, transmits an A-MPDU (consisting of n subframes) to an AP. During the transmission of each subframe, the tag either does nothing, or it corrupts the subframe. If the tag does nothing, the subframe will be decodable at the AP. If the tag corrupts the subframe, it will not be decodable. Therefore, the tag can decode its data by selectively corrupting some subframes and not others. To transmit 1, the tag does nothing during the transmission of

a subframe so that it is received and decoded successfully.[1] To transmit 0, the tag corrupts the subframe so that the AP cannot decode that subframe. The result will be a sequence of failed or successfully decoded subframes that represent the bits for the tag's data. In Section 5, we describe how a passive tag can create interference for WiFi subframes in order to corrupt the frame.

In the second step (labelled 2 in Figure 2), the access point transmits a block ACK to the client device to notify the sender about the status of the subframes in the last A-MPDU. The client device obtains the tag's data directly from the block ACK. Note that although we use the example of a client device transmitting a query packet, the AP could also initiate this process. More importantly, both the client and AP obtain the tag's data irrespective of which device initiates the query.

WiTAG does not require any modifications to existing WiFi networks because the access point receives normal A-MPDUs and transmits normal block ACKs. Therefore it is not even aware of backscatter communications. Similarly, the client device transmits and receives normal 802.11 frames and hence requires no modifications to the MAC or PHY layer. It only requires an application that reads the tag's data from block ACKs. Since WiTAG utilizes MAC-layer A-MPDU aggregation, it is oblivious to the complexities of the PHY layer. As a result, it works with any modulation scheme, coding rate, MIMO configuration, guard interval, and channel width. Most importantly, this feature makes WiTAG compatible with new standards and works with WEP and WPA encrypted packets. In addition to currently available 802.11n and ac networks, WiTAG will be compatible with the 802.11ax standard (released in 2019) because it also supports A-MPDU aggregation [1].

4.1 Throughput of WiTAG

WiTAG transmits one bit per A-MPDU subframe, therefore, the throughput of WiTAG depends on the number of subframes that can be transmitted over a period of time. Our goal is to minimize the transmission time of MPDUs (i.e., subframes) to maximize the throughput of WiTAG. The transmission time of an MPDU depends on the PHY-layer transmission rate and the payload of the MPDU. In our design, MPDUs carry no actual data and are only created for the tag to encode its data. Therefore we can transmit MPDUs with no payload (i.e., only a MAC header) to minimize the transmission time. To further increase throughput, we can use the highest PHY-layer transmission rate that achieves a near-zero error rate, so that frame losses due to path loss or interference are not confused with a tag's data. Note that in WiFi networks we can never guarantee an error rate of zero, because environmental factors such as the movement of people and interference from other devices are typically not under our control. As a result, WiTAG requires a mechanism to detect and correct possible errors, which is a topic of future work.

[1] A robust PHY-layer transmission rate is used to ensure that subframes are successfully decoded when no interference is generated by a WiTAG tag.

5 CORRUPTING A SUBFRAME

In the previous section we have explained how WiTAG sends its data to a WiFi device by corrupting some A-MPDU subframes so that they can not be decoded. In this section, we describe how WiTAG corrupts a subframe.

As described in Section 3, an A-MPDU includes a single PHY header followed by n subframes. The header is used to estimate the channel. This estimated channel is then used to correct the channel for subsequent subframes. Note, because an A-MPDU transmission lasts a few milliseconds and wireless channels do not change during this short time [12, 13],[2] a single channel estimation at the beginning of the A-MPDU is sufficient to successfully correct and decode all subframes. However, if a tag modifies the wireless channel during the transmission of a subframe, then the channel estimation done at the beginning of the A-MPDU will no longer be valid for that subframe and as a result the subframe will not be received successfully. Therefore, WiTAG can selectively corrupt some subframes by changing the wireless channel during their transmission.

5.1 How to change the wireless channel

A wireless channel consists of a direct and multiple indirect paths created by reflectors in an environment. Therefore, if the phase or amplitude of a signal reflected from one of these reflectors changes, the wireless channel will change. A tag in WiTAG uses an antenna which can be switched between two modes: reflective and non-reflective. An antenna is reflective when it is short circuited and non-reflective when it is open circuited [11]. Therefore, the tag can quickly change the wireless channel by switching its antenna between these two modes. Because this switching process can be done very quickly, the tag can be non-reflective during the channel estimation (done at the beginning of an A-MPDU), and then become reflective during the transmission of a subframe. This will corrupt the subframe since the channel estimation is no longer valid for that subframe.

5.2 How to maximize the change in channel

In order to minimize the bit error rate of the data backscattered by a tag, we need to maximize WiTAG's change in the wireless channel. Larger channel changes increase WiTAG's ability to corrupt a subframe which results in a lower bit error rate. Further, for the same bit error rate, a larger channel change increases the distance at which WiTAGs can operate.

Wireless channels can be modelled by complex numbers (represented by vectors). Figure 3 (left) shows the impact that WiTAG has when changing the wireless channel. The vector labelled h (in green) shows the channel when the tag is not reflecting (i.e., in open circuit mode) and the vector labelled h' (in blue) shows the channel when the tag is reflecting (short circuit mode). The vector between h and h' (in black) represents the amount of change created by the tag. Ideally,

[2] The coherence time for WiFi channels are typically about 100 ms

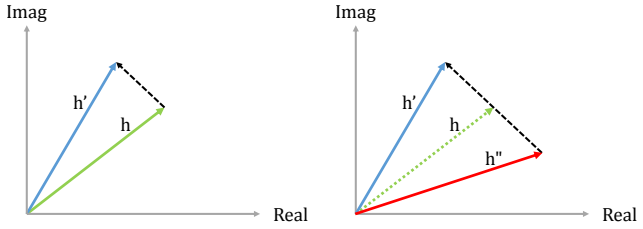

Figure 3: Changing the channel

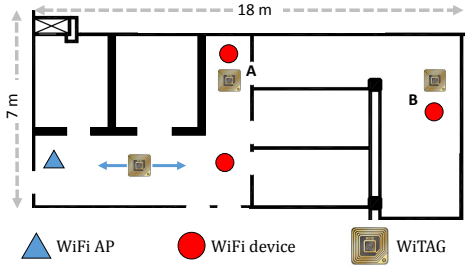

Figure 4: Floor plan of our test bed

we would like to maximize the magnitude of this vector. As explained previously, increasing this difference will result in a lower bit error rate and an increase in the distance at which WiTAGs can operate.

Our idea is that instead of switching the tag's antenna between reflecting (short circuit) and non-reflecting (open circuit) modes, we design and implement a tag which is always reflecting, but which can switch the phase of the reflected signal between 0 and 180 degrees.[3]

Figure 3 (right) shows the impact of this technique for changing the wireless channel. The h' (in blue) and h'' (in red) vectors show wireless channels when the tag is reflecting with 0 and 180 degrees, respectively. For reference, the green vector shows what the state of the channel would have been if there was no tag. This figure shows that our technique significantly increases the impact of WiTAG on the channel, reducing the bit-error rate (BER) and increasing the range of WiTAG. We utilize this technique in our WiTAG prototype, which we empirically evaluate in Section 6.

6 EVALUATION

6.1 Test Bed and Implementation

We conduct our experiments in lab and office spaces in a building on a university campus as illustrated in Figure 4. We conduct experiments in both line of sight and non line of sight scenarios while people are walking around. We built a prototype of WiTAG using a SKY13314-374LF switch [15], an AT91SAM3X8E microcontroller and a WiFi omnidirectional antenna. Our WiFi access point and WiFi client device are desktops with a TP-Link TL-WDN4800 wireless N adapter that supports up to three streams (i.e., a 3x3:3 MIMO configuration), and are equipped with standard antennas. This is an unmodified commodity WiFi adapter.

6.2 Experimental Results

First, we evaluate the performance of WiTAG in terms of bit error rate (BER) and throughput. We place the AP on one side of the room and the WiFi client 8 meters away on the other side, as shown in Figure 4. We place the tag between the AP and the client. The WiFi client device continuously transmits A-MPDUs packets, and WiTAG embeds its data into these packets. The AP receives the packets and transmits the block ACKs for the packets to the client. The client extracts the tag's data by examining the block ACK bitmap. We compare the decoded bits with the expected bits to measure the BER. In each measurement, the tag sends data for one minute. We run each experiment 4 times for each of the 7 different locations to compute the BER.

Figure 5 shows the BER of WiTAG when the tag is placed at different distances from the client. The results show that WiTAG achieves a low BER at all locations between the AP and WiFi client. Specifically, the BER is as low as 0.01 when the tag is close to the AP or the WiFi client. However it slightly increases when the tag is in the middle of the AP and client. This is due to the fact that the strength of a reflected signal (received at the receiver) is proportional to $\frac{1}{D_s^2 \times D_r^2}$, where D_s and D_r are the distance of the reflector object from the sender and receiver, respectively [14]. If the reflector is between the sender and receiver then $D_s + D_r$ is constant and is equal to the distance between the sender and receiver. In this case, $\frac{1}{D_s^2 \times D_r^2}$ is minimized when the reflector is exactly in the middle of the sender and receiver (i.e. $D_s = D_r$). Therefore, because the strength of the reflected signal (received at the receiver) is minimized, the BER is slightly increased. On the other hand, the strength of the reflected signal increases as the reflector is moved closer to the AP or the client, and hence the BER is decreased. Figure 5 also shows the throughout of WiTAG (i.e., the number of bits sent successfully over one second). The figure shows that WiTAG achieves a throughout of 40 Kbps when the tag is close to the client or the AP with a slight drop to 39 Kbps when the tag is about half way between the AP and the device. Therefore, WiTAG achieves stable throughput for different locations of the tag while the AP and the WiFi client are 8 meters apart.

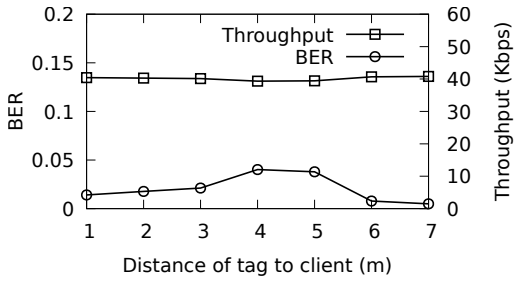

Figure 5: BER and throughput of WiTAG. The client and AP are 8 meters apart.

[3]Note, this can be implemented simply by connecting two short circuited cables with different lengths to the output of the switch. The difference between the length of these cables is a quarter of wavelength and therefore they create a 180 degree phase difference.

Next, We evaluate the robustness and performance of WiTAG in non line of sight scenarios. We place the tag one meter from the client while the AP in another room. We run 60 measurements, each lasts for one minute. We repeat our experiment for two different locations A) and B) as shown in Figure 4. During these experiments, students work in the lab and move around in the vicinity of the AP and the client. The line of sight path between the AP and the client is obstructed by metal cabinets, concrete and wooden walls, and doors.

Figure 6 shows the CDF of the BER for both locations. The figure shows that WiTAG achieves low BERs at all times, even in the presence of obstacles and people that are moving. Specifically, the BER 90'th percentile is 0.007 and 0.018 for locations A and B, respectively. The BER at location B is slightly higher because there are more obstacles blocking the line of sight and therefore the signal is significantly attenuated. However, WiTAG's performance is very stable over an extended period of time even when the AP and client device are 17 meters apart and the line of sight is completely blocked.

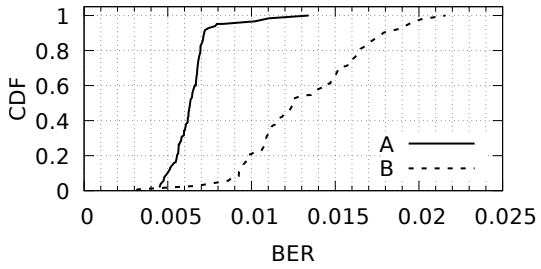

Figure 6: BER of WiTAG: non line of sight scenarios (Figure 4). A and B are about 7 and 17 m from the AP.

These results show that the tag works best when placed near the client or the AP. However, in some situations placing the tag close to the AP provides the advantage of permitting more clients to obtain data from the tag. As our results show WiTAG achieves throughput of up to 40 Kbps. Although existing systems report throughputs of 1 Kbps – 300 Kbps, WiTAG offers the significant advantages of working with encryption and without modifications to existing WiFi networks.

7 DISCUSSION

Query Packet Detection: WiTAG's tag needs to distinguish query packets from other packets sent by other WiFi devices. In addition, it needs to determine the subframe length since it varies from one A-MPDU to another, depending on the physical transmission rate.

Both of these issues can be dealt with by transmitting a specific, known bit pattern in the payload of the first few subframes (trigger subframes) to indicate that the packet is a query packet. This distinguishes query packets from packets being transmitted by other devices and enables the tag to measure the subframe lengths. Since each A-MPDU aggregates up to 64 subframes this does not have a significant impact on the data rate. The specific bit patterns in the trigger subframes can be chosen so they generate different signal amplitudes. The tag then uses an envelope detector and a comparator to detect the trigger subframes (i.e., the beginning of the query packet) and also to determine the timing between two consecutive subframes.

Power Consumption: The major source of power consumption in a backscatter tag is their clock generation block, which is typically an oscillator [18]. An oscillator's power consumption is proportional to the square of the clock frequency (i.e., the higher the clock frequency, the higher the power consumption). Prior work such as HitchHike [16], FreeRider [17], and MOXcattter [19] need to shift the backscatter signal to another channel. As a result, they require an oscillator to operate at least at 20 MHz. The power consumption of precision oscillators in the MHz range is higher than 1 mW, rendering battery-free implementation impractical [18]. Therefore, instead of using high-precision oscillators, these studies use ring oscillators which consume only tens of micro watts. However, ring oscillators suffer from low accuracy and their frequency is significantly impacted by temperature.[4] Therefore, these systems work only in environments where the temperature is very stable. In contrast, because WiTAG does not require shifting the signal to another channel, it does not require a high-frequency oscillator. Specifically, it can use a 50 KHz clock which is highly accurate and very stable with changes in temperature, while consuming only a few microwatts of power. Since WiTAG's hardware is simpler and operates at a much lower frequency than previous work, we anticipate our tag's power consumption will be much lower. Our future work will evaluate the end-to-end power consumption of WiTAG.

8 CONCLUSION

This paper presents WiTAG, a system that enables backscatter WiFi communication using existing WiFi networks. Previous WiFi backscatter systems are impractical to deploy using existing networks because: (1) They require modifications to existing WiFi access points; (2) They do not work with WiFi networks that use security protocols such as WEP and WPA; (3) They require power hungry 20 MHz oscillators; (4) They interfere with existing WiFi communication because transmissions are being reflected to a secondary channel without implementing channel sensing.

In contrast, WiTAG takes a radically different approach from previous work by leveraging the use of A-MPDUs (frame aggregation) to enable backscatter communication. WiTAG works with the latest WiFi standards (802.11n and ac), does not require any modification to access points, and does not interfere with existing WiFi communication (since it does not use a second channel). Most importantly, because WiTAG selectively alters the wireless channel to communicate data by leveraging MAC-layer frame aggregation (rather than modifying PHY-layer symbols) it is compatible with both open and encrypted WiFi networks.

[4]For example, a 5°C change in the temperature can shift the frequency by 600 KHz, which significantly increases the error rate of backscatter systems [18].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback. Tim Brecht and Ali Abedi thank the Natural Sciences and Engineering Council of Canada (NSERC) for partial funding for this project.

REFERENCES

[1] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera. IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi. *IEEE Wireless Communications*, 24, 2017.

[2] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti. BackFi: High Throughput WiFi Backscatter. In *SIGCOMM*, 2015.

[3] S. N. Daskalakis, A. Collado, A. Georgiadis, and M. M. Tentzeris. Backscatter morse leaf sensor for agricultural wireless sensor networks. In *IEEE SENSORS*, 2017.

[4] M. S. Gast. *802.11n: A Survival Guide*. O'Reilly, 2012.

[5] M. S. Gast. *802.11ac: A Survival Guide*. O'Reilly, 2013.

[6] H. Hassanieh, J. Wang, D. Katabi, and T. Kohno. Securing rfids by randomizing the modulation and channel. In *NSDI*, 2015.

[7] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *SIGCOMM*, 2016.

[8] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi Backscatter: Internet Connectivity for RF-powered Devices. In *SIGCOMM*, 2014.

[9] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith. Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions. In *NSDI*, 2016.

[10] Y. Ma, N. Selby, and F. Adib. Drone Relays for Battery-Free Networks. In *SIGCOMM*, 2017.

[11] Y. Ma, N. Selby, and F. Adib. Minding the Billions: Ultra-wideband Localization for Deployed RFID Tags. In *MobiCom*, 2017.

[12] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate Ad Hoc networks. In *MobiCom*, 2002.

[13] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen. Rate adaptation for 802.11 multiuser MIMO networks. In *Mobicom*, 2012.

[14] M. Skolnik. *Radar Handbook, Third Edition*. Electronics electrical engineering. McGraw-Hill Education, 2008.

[15] Skyworks Solutions, Inc. *SKY13314-374LF: 0.1 to 6.0 GHz GaAs SPDT Switch*, 12 2018. Rev. 1.

[16] P. Zhang, D. Bharadia, K. Joshi, and S. Katti. HitchHike: Practical Backscatter Using Commodity WiFi. In *SenSys*, 2016.

[17] P. Zhang, C. Josephson, D. Bharadia, and S. Katti. FreeRider: Backscatter Communication Using Commodity Radios. In *CoNEXT*, 2017.

[18] P. Zhang, M. Rostami, P. Hu, and D. Ganesan. Enabling practical backscatter communication for on-body sensors. In *SIGCOMM*, 2016.

[19] J. Zhao, W. Gong, and J. Liu. Spatial Stream Backscater Using Commodity WiFi. In *MobiSys*, 2018.