Efficient Backscatter with Ambient WiFi for Live Streaming

Qiwei Wang*, Jihong Yu†, Can Xiong*, Jia Zhao§, Si Chen§, Rongrong Zhang‡, Wei Gong*¶

*University of Science and Technology of China, China

†Beijing Institute of Technology, China

‡Capital Normal University, China

§Simon Fraser University, Canada

¶Correspondence: weigong@ustc.edu.cn

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322447

***Abstract*—Backscatter communication with ambient excitations receives great attention recently as it provides a practical battery-free way to convey various IoT data. However, state-of-the-art solutions are of low data rates and thus cannot serve high-bandwidth applications, e.g., live streaming. This paper presents Hermit Crab, the first WiFi-backscatter system that achieves high-throughput communication for video streaming. The key contribution is a differential decoding algorithm using pilot phase. By doing so, it supports single symbol encoding, which is much faster than multi-symbol encoding of previous systems. In addition, Hermit Crab can recover the production data and tag data at the same time. Through extensive experiments, we show that it achieves throughputs of up to 960 Kbps with 802.11g ambient signals, which is 7.6x better than the state-of-the-art system. We also demonstrate that with such good throughputs, it can stably support live streaming of 480p videos at 30 fps.**

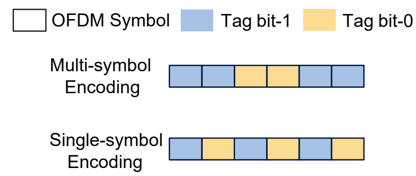


Fig. 1: Single-symbol encoding. When tag transmits 1/0 alternating data. Multi-symbol redundant coding makes system throughput efficiency greatly reduced. In contrast, single-symbol non-redundant method can achieve the theoretical maximum.

I. Introduction

Backscatter communication with ambient carriers has drawn considerable attention in the past few years since it offers a promising way to achieve battery-free communication with existing wireless infrastructure, e.g., ubiquitous WiFi. Several pioneer researchers have put in significant efforts to push front of the line [1]–[5]. Initially, packet-level WiFi backscatter was proposed, which decodes backscatter signals based on the RSSI changes. The data rates of such methods are of only tens to hundreds of bits per second. As a consequence, researchers look into bit-level backscatter with ambient signals and boost data rates to tens of Kbps. While those improvements are sufficient for various IoT applications, such as temperature monitoring, they still cannot meet the demand of high-throughput applications. Meanwhile, wearable cameras have gained recent interest for a wide range of multimedia applications, like life-casting, live streaming malls, and medical surgeries. As such, how to get together backscatter communication and wearable multimedia applications presents a crucial challenge.

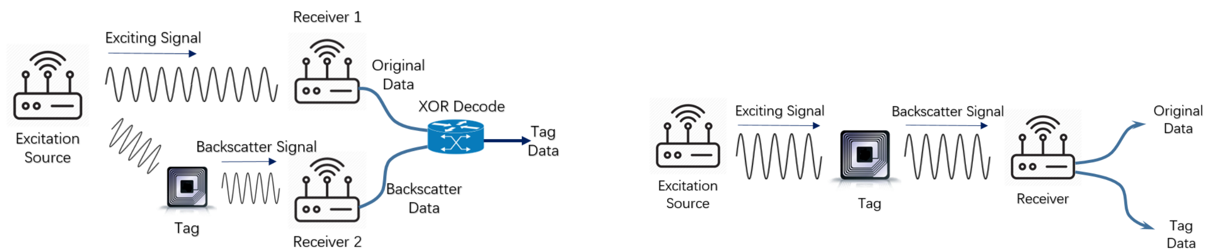
To enable quality backscatter communication for live streaming, we have to make backscatter throughput high enough. Also, when using the ambient signals as carriers, backscatter communication should not affect ongoing data transfers. Unfortunately, none of existing solutions can satisfy these requirements. One of the main reasons is the inefficient demodulation. Most existing systems introduce redundant coding to achieve a lower BER for decoding tag data, which inevitably causes throughput degradation. For example, MOXcatter [5] encodes one tag bit using at least 2 OFDM symbols. In addition, existing systems require an additional receiver to obtain the original carrier signal, causing extra overhead.

To push the limits of ambient backscatter, we identify two critical challenges as follows.

(a) How to efficiently modulate/demodulate without redundant coding? Previous backscatter systems, including FreeRider [4], MOXcatter [5], employ codeword translation [3] to modulate tag data. Such translation will convert one codeword to another while ensuring the backscattered signal legitimate WiFi packets. However, due to the burst error problem, the backscatter system with single-symbol encoding inevitably faces high bit-error-rate (BER). As a result, redundant coding is introduced to make a trade-off between throughput and BER, enabling reliable communication.

(b) How to demodulate the tag data directly from the backscattered signal? Fig. 2(a) shows the architecture of a traditional backscattering system. Existing codeword translation scheme modulates one valid symbol by changing its phase and frequency to another valid symbol. This leads to that previous systems require the knowledge of both the original data and

*Corresponding author: Wei Gong.



(a) Previous OFDM Wi-Fi backscatter systems overview.

(b) Hermit Crab overview.

Fig. 2: Comparison of Hermit Crab's architecture and prior backscatter systems.

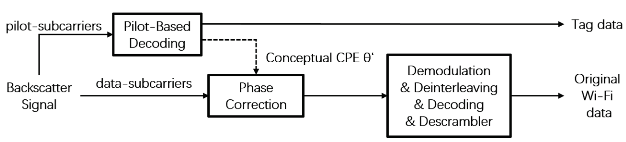


Fig. 3: Hermit Crab overview. It calculates the phase shift of the pilot subcarriers to demodulate tag data, then recovers the productive data of excitation signals.

received data to do backscatter decoding, which is why they need two receivers with one on the original channel and the other on the backscatter channel. How to demodulate the tag data directly from the backscattered signal and thus reduce the hardware overhead becomes the key to improving backscatter throughput.

In this paper, we introduce Hermit Crab, a novel backscatter system that can achieve high-throughput for live streaming. It takes a distinct modulation approach. By using pilot subcarriers, it avoids the negative impact of modulation errors, achieving extremely low BER even with single-symbol encoding. Our key insight is an important characteristic of the pilot subcarrier: the data of the pilot subcarrier is predefined, which means both the original data and modulated data can be decoded from the backscattered signal at the same time. Also, it does not require the deployment of additional receivers, a single receiver is adequate. Note that the recovered carrier signal can also be demodulated by a commercial WiFi receiver, so our backscatter system will not interfere with the communication on the original channel.

To validate our design, through experiments we show that Hermit Crab can achieve $1e^{-3}$ BER using single-symbol encoding and the maximum throughput is 960 Kbps, which is 7.6x better than the state-of-the-art system, MOXcatter. With the help of such excellent physical-layer support, we demonstrate that it can stably support live streaming of 480p videos at 30 fps.

II. Hermit Crab Overview

Hermit Crab is a novel backscatter communication system that uses the single-symbol encoding method to maximize the throughput of OFDM backscatter system. As shown in Fig. 3, the detailed workflow is as follows. Hermit Crab first calculates the phase rotation value introduced by tag modulation for each OFDM symbol through the pilot subcarriers. Then, it leverages this phase value to decode the tag data transmitted on each OFDM symbol. After that, Hermit Crab calibrates each OFDM symbol based on the previously calculated phase rotation value. The calibrated backscatter signal will be demodulated according to the standard WiFi receiver process. Because the phase offset introduced by tag modulation is calibrated and eliminated, the data on the carrier can be successfully recovered. One big advantage of Hermit Crab is that it can decode tag data and carrier data simultaneously using only one receiver.

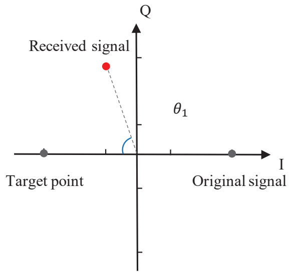
III. Hermit Crab Design

The redundant coding method will greatly reduce the performance of the backscatter system. We can take the ideal throughput of a single-symbol encoding scheme as a baseline. For a redundant coding scheme that encodes one tag bit with n symbols, the throughput is only $1/n$ of the baseline's performance. This has become a bottleneck that severely restricts the performance of the backscatter system.

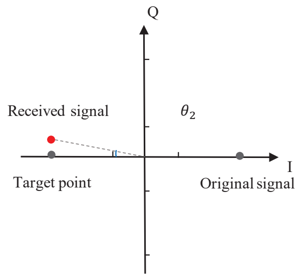
A. Hermit Crab Modulation

We are inspired by the codeword translation and try to directly use the phase information of the backscattered signal for demodulation. We use the phase and amplitude information to convey tag data. Here, we only need to extract the phase and amplitude added by the tag from the backscattered signal to demodulate the tag data. Unlike codeword translation in HitchHike [3], the excitation signal only acts as a carrier in our method. the data content of the carrier itself does not affect the modulation, nor does it need to know the carrier content for demodulation.

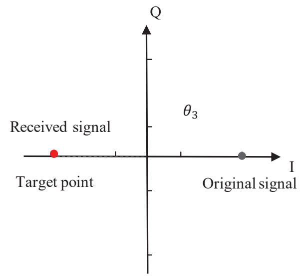
The key insight to realize the Hermit Crab modulation lies in the pilot signal. As Fig. 5 shows, assume that the k^{th} subcarrier of the n^{th} symbol of the excitation signal can be expressed as $X_{n,k}$ in the time domain. Then, we can describe the tag modulation process as multiplying $X_{n,k}$ by $e^{j\phi}$. This is expressed by rotating $X_{n,k}$ in the complex coordinate system by $j\phi$. For instance, if the tag transmits bit 0, it rotates 0°; if the tag transmits bit 1, it rotates 180°.



(a) Unprocessed received signal: Phase offset θ_1 includes channel loss, frequency offset, and tag modulation.



(b) Received signal after compensation: Phase offset θ_2 includes tag modulation and residual phase error that are not fully compensated.



(c) Received signal after differential: Phase offset θ_3 includes tag modulation only.

Fig. 4: For unprocessed received signal, it is the combined result of channel loss, frequency offset, and tag modulation. After channel compensation and frequency correction, there are two kinds of phase offset: tag modulation and residual error. Residual error is usually small, negligible compared to tag modulation. But the residual error will continue to accumulate, Since the residual error of each symbol can basically be regarded as a fixed value, it can be eliminated by the differential.

At this time, the received backscattered signal becomes $X_{n,k} * e^{j\phi}$. This means that as long as we know the data of k^{th} subcarrier and n^{th} symbol of the carrier, we can know the phase offset $e^{j\phi}$. Then, we can decode the tag data. Exactly, the four pilot subcarriers with the subscripts -21, -7, 7, and 21 are pre-defined in the protocol. No additional receivers or equipment modifications required, we can directly use the received backscattered signal for decoding. Since we decode directly at the signal level, we do not rely on the payload like codeword translation. Because the pilots of each OFDM symbol are independent of each other, we can implement single-symbol modulation. According to our simulation experiments, Hermit Crab still achieves extremely low BER when using single-symbol modulation.

TABLE I: Tag modulation / demodulation based on phase offset.

Modulation	Phase changed by tag	Tag data
BPSK	0°	0
	180°	1
QPSK	0°	00
	90°	01
	180°	11
	270°	10

B. Demodulation by Pilot Tone

When demodulating OFDM frames, the equalizer response computed from a burst preamble is used to correct many imperfections in the OFDM signal [6]. For more accurate decoding data, pilot tone (or pilot subcarriers) is introduced in the OFDM system.

Taking 802.11g as an example, in each OFDM symbol, subcarriers -21, -7, 7, and 21 are dedicated to pilot tone. These pilot subcarriers modulate a known data sequence with BPSK. At the receiver, we can calculate the phase or amplitude offsets by comparing the ideal signal and actual received signal on the pilot subcarriers for each OFDM symbol. Since each OFDM symbol has pilot subcarriers, we can use it to decode single-symbol encoding tag data. For a backscatter system, the backscattered signal $R_{n,k}$ at subcarrier k of the n^{th} OFDM symbol can be obtained by Eq. 1 as follows:

$$R_{n,k} = X_{n,k}H_kI_n + W_{n,k} \approx X_{n,k}H_kI_n \tag{1}$$

Where $W_{n,k}$ is linearly transformed AWGN with unchanged variance σ_n^2 at subcarrier k and this kind of additive white noise does not cause any constellation rotation of OFDM symbol and can be omitted.

Then, $X_{n,k}$ and H_k represent the transmitted data on k^{th} subcarrier of n^{th} symbol and the channel frequency response from channel estimation respectively. The term I_n is a phase rotation that has the same effect on all subcarriers and is usually called the common phase error (CPE), which is the residual error that has not been eliminated after frequency correction and channel compensation. Generally, CPE is very small and we can ignore it. When there is tag modulation, the phase offset introduced by the tag modulation is much larger than the CPE. So, we can simply use I_n to indicate the tag modulation phase rotation.

As shown in Fig. 4(a), we can calculate a phase offset θ_1 between the ideal signal and the received signal. This phase offset θ_1 includes three parts: phase offset caused by the channel loss, phase offset caused by the frequency offset, and phase rotation introduced by the tag modulation. The phase offset caused by channel loss and frequency shift also exists in the active radio transmission process. Commercial Wi-Fi receivers can eliminate this type of phase deviation.

For Wi-Fi 802.11g, through the Short Training Field (STF) and Long Training Field (LTF), the receiver can estimate the frequency deviation and channel estimation. Then, the receiver can compensate and correct these errors. Generally, only a

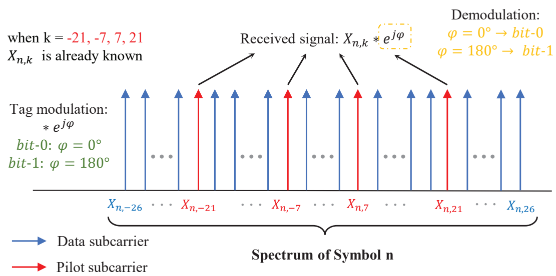


Fig. 5: Hermit-Crab modulation: Tag modulation is equivalent to multiplying $e^{j\phi}$. Then, the received signal is equivalent to $X_{n,k} * e^{j\phi}$. For pilot tone, $X_{n,k}$ is already known, So we can extract the tag data directly from the backscattered signal.

small residual error will be retained after compensation, which is the CPE mentioned above. We can calculate the CPE from the pilot subcarriers and then correct it more accurately. This process is called phase tracking. Generally, after this step, we can get very accurate IQ samples and then demodulate Wi-Fi data.

However, in the backscatter system, the remaining phase offset θ_2 includes not only the CPE but also the phase rotation introduced by the label modulation data, which is shown in Fig. 4(b). We can not separate the CPE from the phase rotation introduced by tag modulation, and demodulate the tag data with the tag modulation phase rotation. But fortunately, after compensation, this residual error is already very small. It is negligible compared to the phase rotation introduced by tag modulation and can be ignored. In other words, we can regard θ_2 as the tag modulation phase rotation, and then use θ_2 directly to demodulate tag data. Based on this idea, we first propose a preliminary single-symbol demodulation scheme.

As shown in the Eq. 1, for pilot subcarrier, the $X_{n,k}$ is a known term. H_k is the channel frequency response, it can be eliminated within the entire frame through channel compensation. We can get θ_2 by calculating the phase differences $\theta_{n,k}$ between received signal $R_{n,k}$ and reference data (ideal signal) $X_{n,k}$:

$$\begin{aligned}\hat{\theta}_{n,k} =&\angle(\sum_{i\in\rho} X_{n,k}H_kR^*_{n,k})\\ =&\angle(\sum_{i\in\rho} X_{n,k}H_k(X_{n,k}H_kI_n)^*)\\ =&\angle(\sum_{i\in\rho} I_n)\end{aligned} \tag{2}$$

Where $R^*_{n,k}$ is conjugate complex of $R_{n,k}$. ρ denotes the set of pilot subcarrier index, and $X_{n,k}H_k$ is the estimate of transmitted signal $X_{n,k}$ under channel H_k. By first multiplying the reference data $X_{n,k}$ by the channel frequency response H_k, the result of the product is then correlated with the backscattered signal $R_{n,k}$ to eliminate the influence of the phase offset caused by the wireless channel. Then, we obtain the modulation phase $\hat{\theta}_{n,k}$. According to this value, we can decode the tag bit. As analyzed earlier, CPE is usually small. Normally, it does not affect the demodulation result. We confirmed this through experiments.

C. *Differential Decoding Algorithm*

However, the phase rotation calculated by the preliminary single-symbol demodulation scheme includes not only tag modulation phase rotation, but also CPE. We can not eliminate CPE from I_n to more accurately estimate the phase shift introduced by tag modulation. When the packet length is short, this small error will not affect the system performance. However, if the data packet is longer, the CPE will gradually increase and the interference from the CPE can no longer be ignored. For Eq. 1, when the influence of tag modulation is not considered and the effect of the frequency offset is introduced, then the received signal is:

$$R_{n,k} = X_{n,k}H_ke^{j2\pi kt_\Delta n\frac{T_s}{T_u}} \tag{3}$$

Where T_s and T_u are the duration of one OFDM symbol and valid data in one OFDM symbol respectively.

It can be seen that the value of CPE becomes larger as the number of the OFDM symbol and subcarrier number increase. When the Wi-Fi packet is short, the value of CPE is negligible compared to the phase shift caused by tag modulation. However, as n and k increase, the value of CPE will gradually increase. When the Wi-Fi data packet is long enough, the value of CPE will affect the demodulation result. The BER of Hermit Crab will rise significantly.

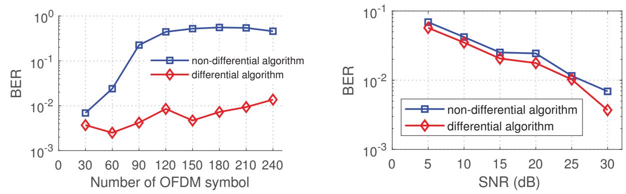
To solve this restriction, we need to improve our design. Our solution is based on the insight: Although CPE is increasing with the number of OFDM symbols, the phase difference between two adjacent symbols is always very small. This phenomenon is not affected by the number of symbols and the index of subcarriers. Therefore, we try to solve this problem with a differential method. The evolved Hermit Crab performs correlation operations on subcarriers at the same position of two adjacent OFDM symbols to avoid the possible negative effects of CPE accumulation:

$$\begin{aligned}Z_{n,k} =&R_{n,k}X^*_{n,k}[R_{n-1,k}X^*_{n-1,k}]^*\\ =&|X_{n,k}|^2|X_{n-1,k}|^2|H_k|^2e^{j2\pi kt_\Delta n\frac{T_s}{T_u}}e^{-j2\pi kt_\Delta(n-1)\frac{T_s}{T_u}}\\ =&|X_{n,k}|^2|X_{n-1,k}|^2|H_k|^2e^{j2\pi kt_\Delta\frac{T_s}{T_u}}\\ =&|H_k|^2e^{j2\pi kt_\Delta\frac{T_s}{T_u}}\end{aligned} \tag{4}$$

and then, the relative phase shift of two adjacent OFDM symbols can be defined as:

$$\Phi_{n-1,n} = \angle(\sum_{i\in\rho} Z_{n,i}) \tag{5}$$

As shown in Fig. 4(c), through the difference between adjacent symbols, we can significantly eliminate the influence of residual errors. The received signal after the differential process basically contains only the phase rotation introduced by the tag modulation.



(a) BER of backscattered signal at different packet lengths.

(b) BER of backscattered signal at different SNRs.

Fig. 6: Performance comparison of evolved Hermit Crab with difference algorithm and original Hermit Crab without difference algorithm.

IV. Evaluation

Since many OFDM systems are similar in design, we choose the most common OFDM system as the representative to evaluate the performance of Hermit Crab in the simulation. The excitation signal is 802.11g with BPSK modulation. In the experiment, we assume that the excitation signal transmitter, the backscatter tag, and the receiver are all working at line-of-sight (LoS). In addition, we also simulate the CFO, SFO, signal-to-noise ratio (SNR), and other factors to evaluate the system performance of Hermit Crab as realistically as possible.

A. Differential algorithm v.s. non-differential algorithm

1) Impact of OFDM symbol number: Due to the gradual accumulation of CPE, the performance of the original Hermit Crab will drop sharply in the case of long data packets (a large number of OFDM symbols). Therefore, we incorporate the differential mechanism and propose evolved Hermit Crab. The experiment is simulated in an environment with an SNR of 30, thereby highlighting the effect of the different algorithms on BER.

As depicted in Fig. 6(a), the evolved Hermit Crab with differential algorithm can maintain an extremely low BER regardless of packet length. Conversely, for the original Hermit Crab that does not use a differential algorithm, when the data packet is short, the BER is low. But, as the packet length increases, the CPE continues to accumulate. When the number of OFDM symbols is above 90, the BER reaches $1e^{-1}$ level.

There are two points to note: First, it can be seen that when the number of OFDM symbols changes from 60 to 90, the BER increases rapidly. This is because when the number of OFDM symbols in a Wi-Fi packet exceeds 70, the value of CPE has exceeded $\pi/2$. At the receiver, all tag data thereafter will be reverse decoded (0 to 1, 1 to 0), resulting in a sharp rise in BER. Second, when the BER increases to about 50%, it tends to stabilize. The BER even slightly decreases when the number of symbols is 240. This is because the phase change is periodic. When the CPE gradually increases to about 2π, the decoding of the tag data is correct again. Therefore, although the CPE continues to increase, it may even happen that the BER decreases.

2) Impact of SNR: We also fix the packet length to 30 symbols and analyse the impact of SNR on these two algorithms. Fig. 6(b) depicts that, as expected, whether it is the evolved Hermit Crab or the original Hermit Crab, as the SNR decreases, the BER gradually increases. This is because the lower the SNR, the less accurate the result of channel estimation, resulting in a larger residual CPE after correction. We also find that the BER of evolved Hermit Crab is lower than the original Hermit Crab. This shows that the differential algorithm can not only solve the BER reduction problem caused by the increase in number of symbols but also increase the performance stability of Hermit Crab.

B. Performance of Hermit Crab

1) Backscatter Throughput: As shown in Fig. 7(a), we try to use BPSK, QPSK, 8PSK, and 16PSK for data modulation on a single symbol. When the data modulation method is 16PSK, the throughput of Hermit Crab can reach a maximum of about 960 Kbps. Even only BPSK is used, Hermit Crab's data rate can still reach 240 Kbps, which is significantly higher than the throughput of state-of-the-art works using redundant coding.

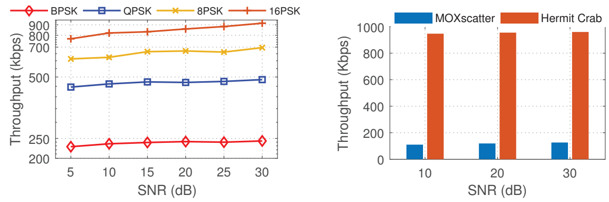
To further illustrate our contribution, we compare Hermit Crab with the-state-of-art research (MOXcatter). In this experiment, 802.11g with BPSK modulation is used as the excitation signal. To highlight the performance gap caused by the difference in methods, we conduct experiments under different SNRs to minimize errors. As we can see in Fig. 7(b), the maximum throughput that MOXcatter can achieve is only 126 Kbps, while Hermit Crab can reach up to 960 Kbps, which is about 7.6 times that of MOXcatter.

Besides, When the SNR drops from 30 to 10, the throughput of Hermit Crab only changes from 960 Kbps to 947 Kbps, while MOXcatter decreases sharply from 126 Kbps to 114 Kbps. The performance degradation of MOXcatter is significantly greater than that of Hermit Crab. This result shows that Hermit Crab not only has a higher throughput than the previous OFDM backscatter system but also is more robust to SNR. This is because the pilot-based backscatter technology gets rid of the limitation of redundant coding, which can greatly improve the throughput of the backscatter system. The introduction of the differential algorithm has played a good anti-jamming role.

According to our analysis, Hermit Crab can recover productive data of the excitation signal from the backscattered signal after decoding tag data. We also evaluate Hermit Crab's original signal recovery capability. According to our experiments, more than 95% of Wi-Fi packets can be recovered correctly when SNR = 30. Even if in the worst scenario (when SNR = 5), more than 80% Wi-Fi packets can still recover the carrier data from backscattered signals.

C. Application Case: Live Streaming

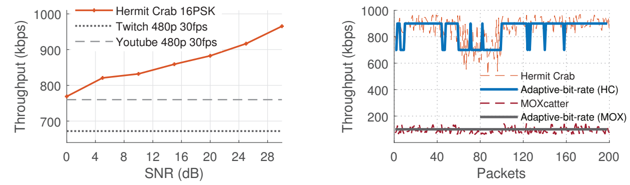
Live video streaming is bandwidth-sensitive. Many factors, such as encoding method, stream resolution and frame rate affect instantaneous bitrates for streaming. For instance, if we



(a) Hermit Crab throughput under various modulation methods.

(b) Performance comparison with previous backscatter system.

Fig. 7: Critical performance evaluation of Hermit Crab.



(a) Required upload speed for video streaming v.s. Hermit Crab.

(b) ABR algorithm for Hermit Crab and MOXcatter.

Fig. 8: Hermit Crab for live streaming.

upload a 480p video at 30 frames per second, according to the official recommendation from Twitch, the minimal upload speed should be more than 672 Kbps. For YouTube, the upload bandwidth should be over 760 Kbps [7]. As shown in Fig .8(a), when Hermit Crab uses 16PSK for modulation, even if the wireless channel is very poor (SNR = 0), the throughput is still above the required speed. This shows that Hermit Crab can meet the bandwidth requirements of live streaming media under varying channel quality.

Dynamic Adaptive Streaming over HTTP (DASH) is one of the mainstream video streaming technologies. Its adaptive bit rate (ABR) logic works as follows: Generally, DASH divides a video into many segments. Each segment can dynamically choose different Representation (code rate, resolution and frame rate) to download according to network conditions or client delay. In our simulation, we set three kinds of bitrates: 900 Kbps (for 480p 30fps), 700 Kbps (for 480p 15fps), and 100 Kbps (only for audio). We choose rate-based algorithm that estimates each bitrate from the average date rate of the past three packets. As Fig. 8(b) depicts, the throughput of Hermit Crab is always higher than 672 Kbps (Twitch) and 720 Kbps (Youtube). According to the ABR algorithm, we can choose a bitrate of 900 Kbps to download a 480p 30fps video. Although the throughput sometimes drops due to the network conditions, Hermit Crab can still support 750 Kbps bitrate to download 15fps video, as compared with MOXcatter with 100 Kbps bitrate, which is difficult to support video streaming.

V. Related Work

Low-cost, low-power, and low-complexity backscatter communication systems have attracted much attention today. [2] first combines this technology with Wi-Fi, opening up a precedent for Wi-Fi backscatter. After that, Wi-Fi Backscatter and FS-Backscatter [8] try to use CSI and RSSI to convey tag data. Since one or more Wi-Fi frames can only transmit one tag bit, the throughput of these works is limited to a few Kbps. To overcome the poor performance of backscatter system, Passive Wi-Fi [9] implements symbol-level Wi-Fi backscatter. Passive Wi-Fi generates 802.11b frames by reflecting a single tone, it also introduces a side-band backscatter design to improve spectrum utilization. Interscatter [1] utilizes a modified BLE device to generate a single tone for backscatter communication. HitchHike [3] proposes the codeword translation, which can directly take ambient 802.1b signal as carrier. Therefore, both transmitters and receivers can directly use commercial equipment, which weakens the limitation on the excitation signal and enhances the versatility of backscatter communication. FreeRider [4] extends codeword translation to OFDM Wi-Fi, Bluetooth, and ZigBee. MOXcatter [5] is a pioneer in exploring the MIMO backscatter system.

VI. Conclusion

To conclude, we believe that Hermit Crab marks an important milestone of high-throughput backscatter communication with ambient WiFi. It is the first WiFi-backscatter system that supports single-symbol encoding. Consequently, it pushes the limits of communication throughput to almost 1 Mbps and requires only a single receiver. By doing so, it opens the door for a variety of high-throughput backscatter applications. After showing the possibility of marriage of ambient backscatter and multimedia streaming, we hope it will fuel more community interests along this line and make the dream of ubiquitous live streaming come true.

VII. Acknowledgment

This work was supported by NSFC Grant No. 61932017 and 61971390, and the Fundamental Research Funds for the Central Universities No. WK2150110013.

References

[1] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proc. of ACM SIGCOMM*, 2016.

[2] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-fi backscatter: Internet connectivity for rf-powered devices," in *Proc. of ACM SIGCOMM*, 2014.

[3] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "Hitchhike: Practical backscatter using commodity wifi," in *Proc. of ACM SenSys*, 2016.

[4] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "Freerider: Backscatter communication using commodity radios," in *Proc. of ACM CONEXT*, 2017.

[5] J. Zhao, W. Gong, and J. Liu, "Spatial stream backscatter using commodity wifi," in *Proc. of ACM MobiSys*, 2018.

[6] Pilot tracking (ofdm demodulation). [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/customOfdm/Content/customOFDM_EqlzrTrckng_Tracking.htm

[7] Speed of streaming. [Online]. Available: https://restream.io/blog/what-is-a-good-upload-speed-for-streaming/

[8] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proc. of ACM SIGCOMM*, 2016.

[9] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive wi-fi: Bringing low power to wi-fi transmissions," in *Proc. of USENIX NSDI*, 2016.