FAQs about OFDMA-Enabled Wi-Fi backscatter

We categorize frequently asked questions (FAQs) about OFDMA Wi-Fi backscatter into the following classes for the convenience of readers:

1) What is the motivation to realize OFDMA in Wi-Fi backscatter systems?
2) Why and how synchronization can be realized in the OFDMA backscatter system?
3) The problem the existing backscatter systems have in common is the short tx-to-tag range (within several meters); what is the real use case that OFDMA backscatter approach can benefit?
4) What is the value of the proposed OFDMA backscatter system if it can not be realized with COTS Wi-Fi devices?
5) How the power consumption of the backscattering tag is computed specifically?
6) How could the OFDMA backscatter system coexist with regular Wi-Fi systems?
7) How the channel coordination among tags, and retransmissions of tags in the OFDMA backscatter system are realized?
8) What if the frequency selective fading happens to some tags? Is it possible to let the tag generate all the 48 data subcarriers in 802.11g?
9) In Figure 18 and Figure 19 of the manuscript, why the BER varies dramatically but the throughputs are almost the same across subcarriers? How the throughput is calculated?

Q1: What is the motivation to realize OFDMA in Wi-Fi backscatter systems?

Wi-Fi backscatter is to realize backscatter communication in Wi-Fi systems. "With the rapid rise of connectivity needs from pervasive IoT applications, the Wi-Fi system itself is moving away from further improving the peak speed with a single device connected, but targets at increasing the rate of data exchange by all users [16–19], where it is more important to support a large number of connected IoT devices featured by short bursts of data [16, 17]. To this end, the next-generation Wi-Fi 802.11ax performs a ground-up reworking of the core multiple access mechanism to realize orthogonal frequency division multiple access (OFDMA). and the Wi-Fi system itself is moving towards OFDMA." ***(2nd paragraph of Introduction Section, left column, page 1 of the submission.)***

Moreover, ``The adoption of OFDMA in next-generation Wi-Fi indicates that fewer Wi-Fi devices with single-carrier modulation in the core will be used in the future, moreover, almost all of the modern Wi-Fi devices use 802.11g/n based on OFDM [5]; therefore, there is a strong necessity to investigate how to enhance the capacity of the Wi-Fi backscatter system accommodating the multi-carrier modulation mechanism." OFDMA is based on OFDM that is the multi-carrier modulation mechanism. ***(2nd paragraph, right column, page 1 of the submission.)***

Q2: Why and how synchronization can be realized in the OFDMA backscatter system?

We smartly design our backscatter communication mechanism in the 802.11g OFDM framework, where the inherent tx-rx synchronization scheme of 11g is utilized. However, OFDMA backscatter requires to synchronize all the components in the system, including tx, tags and rx. Our design of the waveform from the excitation signal transmitter can help realize tx-tag synchronization, where the synchronization preamble is generated by OOKing the continuous wave. The tag-rx synchronization is partially realized by utilizing the inherent CP design of 11g OFDM mechanism, and the residual asynchronization effects incur phase offsets, which are calibrated by our own design ***(Section 5, page 7 of the submission)*****.**

The basic principle of synchronizing all components in the OFDMA backscatter system is summarized in our submission ***(Section 3, page 3 of the submission)*****;** we also verify our synchronization design with experimental results ***(Section 7.1, page 9 of the submission)*****.**

Q3: The problem the existing backscatter systems have in common is the short tx-to-tag range (within several meters); what is the real use case that OFDMA backscatter approach can benefit?

"The limitation that existing Wi-Fi backscatter systems [3–5] including ours have in common is the short tx-tag range, which is affected by the sensitivity of the receiver in the tag. In particular, higher sensitivity of the tag's receiver could allow longer range but incur considerable more power consumption, and lower sensitivity can save much power but results in short tx-tag range [3]. However, the high concurrency of our backscatter system still could be utilized in the scenario where IoT devices are super-densely deployed, such as the wireless avionics intra-communications (WAIC) [35, 36]. The purpose of WAIC is to replace the data-carrying wires in the widebody passenger jet with wireless links, which could save about 1.8 tons of the aircraft's weight and reduce the corresponding wiring complexity and uncertainty [36].

For example, the Airbus A380's every single wing is installed with 10,000 sensors [37], and the aircraft totally has as many as 25,000 sensors [38]. The wingspan and the body length of A380 are 79.75m and 72.75m respectively, which means that there will be more than 68-250 sensors in every single meter of the aircraft on average; therefore, the current tx-tag range of our approach is still usable for the scenario." ***(1st and 2nd paragraph of Section 8, page 11 of the submission)*****.**

Q4: What is the value of the proposed OFDMA backscatter system if it can not be realized with COTS Wi-Fi devices?

Our work is the first effort to realize OFDMA in Wi-Fi backscatter. The value of the work is to reveal the particular challenges for implementing OFDMA in Wi-Fi backscatter, present simple but effective designs to overcome those challenges, and implement a prototype system to validate our design. OFDMA is the fundamental feature of next-generation Wi-Fi 802.11ax, and our work takes one step ahead to realize OFDMA backscatter in 802.11g. We implement our system using WARP, which is a widely-adopted software defined radio platform in the community. It is worth of trying to do some modification if the concurrency can be improved by 48 times, and our work present a technical route to achieve this.

Q5: How the power consumption of the backscattering tag is computed specifically?

We design an integrated circuit for the tag's digital processing module and perform simulations to measure the tag's power consumption. Details of power consumption can be found in Section 7.2 *(Power consumption, page 10 of the submission)*.

Q6. How could the OFDMA backscatter system coexist with regular Wi-Fi systems?

The OFDMA backscatter system uses the excitation signal transmitter to first reserves the Wi-Fi channel, which is realized by using request-to-send (RTS) and clear-to-send (CTS) frames as in Passive Wi-Fi [3].

We conduct experiments to verify that our system has similar impact on existing Wi-Fi systems as regular Wi-Fi systems: “OFDMA backscatter system coordinates channels with other Wi-Fi systems using RTS and CTS messages as in [3]. In the experiment, we use a laptop with Intel(R) Dual Band Wireless-AC 3165 Wi-Fi chipset and a NETGEAR Nighthawk AC1900 wireless router to act as the transmitter and receiver of a Wi-Fi system, and use iPerf3 [39] to record the corresponding TCP throughputs as experimental results. The devices are transmitting at the highest data rate of 802.11g in CHANNEL 1 of 2.4G frequency band. We compare the performance of the Wi-Fi system in the presence of two devices: 1) another Wi-Fi router transmits every 10ms, and 2) the OFDMA backscatter system turned on every 10ms. Figure 23 plots the TCP throughput and shows that our system has a similar impact on the Wi-Fi transmission as the regular Wi-Fi transmitter.” *(Coexistence with other Wi-Fi systems, right column, page 11 of the submission.)*

Q7. How the channel coordination among tags, and retransmissions of tags in the OFDMA backscatter system are realized?

Such functions are realized in the network layer protocol, which are not the focus of the

OFDMA backscatter system; we do not include this part in the submission due to the limitation of the space. The basic design is as following:

Network Stack Design

We first describe the structure of downlink control frame which broadcasted after the preamble by the transmitter. This frame enables transmitter control all the tags (i.e. allocating subcarriers, ACK for retransmission etc.). Then, we describe the structure of uplink data frame which is from the tags. Finally, we will discuss the ACK and retransmission design.

Downlink frame structure

In the transmitting frame illustrated in Fig. 3 of paper, the to-tag frame will contains following frames: i) to-tag preamble which is a fixed sequence (we choose an 11-bit barker code for our system). ii) Several control frames for different tags. iii) The start transmitting frame to trigger all the tags to transmit. Following we will first specifically describe the structure of control frame:

0:9	10:15	16:18	19:20
Tag ID	Subcarrier Index	Rate	Checksum

Structure of control frame

Tag ID bits are the tag addresses, which are the destination of the frame. The tag ID is set when deploying the tags. The number of tag IDs can be greater than the number of subcarriers to enable time domain multi-access. The enhanced tag will have several tag IDs which is corresponding to the SSB modules. Minding that all bit-1s tag ID is reserved for start transmitting frame.

Subcarrier index is used to allocate the subcarrier of the tags. There are 48 subcarriers in our design and each subcarrier is corresponding to one subcarrier index. The rate part is used to control the modulation and coding mechanism of tags which makes rate adaption feasible. The tags can transmit at different rates which are the combination of BPSK, QPSK and 1/2, 2/3, 3/4 coding rate.

Then the start transmitting frame:

0:9	10:57	58:59
Tag ID	ACK	Checksum

Structure of start transmitting frame

Tag ID is all bit-1s to illustrate that this is a broadcasting start transmitting frame. ACK has 48 bits for 48 subcarriers. The ACK mechanism will be discussed specifically in the following section.

Uplink frame structure

Tag ID bits are the tag addresses, which are the source of the frame and the same as the tag ID of downlink control frame. Length is the data numbers transmitted by tags in bytes. The last part will be the data.

0:7	8:17	18:27	......
Preamble	Tag ID	length	Data

The uplink frame structure

ACKs and retransmissions

The receiver listens to the ACKs and conveys this information back to the tags. We do not adopt the design in Passive Wi-Fi, where the ACK is directly sent to tags by receiver, in order to avoid the unreliable AM modulated signal in the long range. Specifically, if the ACK is successfully decoded at the receiver, it delivers the ACK to the transmitter. And then the transmitter sets the ACK bit in the start transmitting frame to 1 and sends it to the tags. If the ACK is not received at the transmitter before ACK timeout, it will set the ACK bit to 0. When a tag receives a 0 of its allocated subcarrier in the next transmission, it retransmits its data.

Q8. What if the frequency selective fading happens to some tags? Is it possible to let the tag generate all the 48 data subcarriers in 802.11g?

If some tag's throughput is constantly low, it is possible the frequency selective fading happens. To deal with this issue, we could reassign subcarriers to tags in the next round of communication; moreover, the enhanced tag technique also can help for resolving the issue.

"It is worth mentioning that the enhanced tag technique also helps resolving the frequency selective fading issue by conveying the information with multiple subcarriers; for the regular tag generating the single subcarrier, the frequency selective fading could be resolved by reassigning subcarriers using the upper layer protocol described in \cite{video}. Theoretically, it is possible to enhance the tag with the capability of reflecting many even all the subcarriers, which however will incur high power consumption and big-size tag thus in conflict with our motivation." ***(3rd paragraph, right column, page 6 of the submission.)***

Q9. In Figure 18 and Figure 19 of the manuscript, why the BER varies dramatically but the throughputs are almost the same across subcarriers? How the throughput is calculated?

"**Capacity and concurrency.** We conduct experiments to show that our OFDMA backscatter system can support 48 tags' concurrent transmissions. Those 48 tags are randomly placed in $4m$ neighborhood of the transmitter and the receiver is 10m away from the transmitter. The 48 data subcarriers in 802.11g are assigned to those tags respectively; tags cyclically transmit $5M$ data upon receiving the excitation signal, which are in the form of frames and stored in the connected FPGA.

Figure 18 illustrates the experimental results with BPSK and 1/2 coding rate. The vertical axis of BER subfigure is in logarithm, and we purposely set those 0-valued BER to be 1×10^{-5} for convenience of demonstration as there is no definition of $\log_{10}0$. The pilot and null subcarriers are not used to transmit data, thus the corresponding throughput and BER readings are set to be 0 and 1×10^{-5} respectively also for convenience of demonstration. We can see that the BERs of those tags vary from the level of 10^{-5} to 10^{-2}, and the corresponding throughputs range from 114.3kbps to 63.6kbps. The throughput is the number of bits in all correctly received frames per second. For the tag transmitting 5M data in a cycle, the lowest BER is in the level of 10^{-5}; when the BER is in the level of 10^{-2}, it is reasonable that the corresponding throughput reduction is about 50kbps as shown Fig. 18. We can see that a number of BERs are in the level of 10^{-3}, thus the corresponding throughput reductions of the tags are around 5-10kbps, because it may happen that a number of bit errors occurring in the same data frame, which also influences the throughput calculation. The network capacity (maximum aggregate throughput) of the system is 5.2Mbps. Figure 19 shows the experimental results with QPSK and 3/4 coding rate, where the capacity of the network can achieve 16Mbps with 48 tags transmit concurrently." *(3rd and 4th paragraphs, right column, page 10 of the manuscript.)*

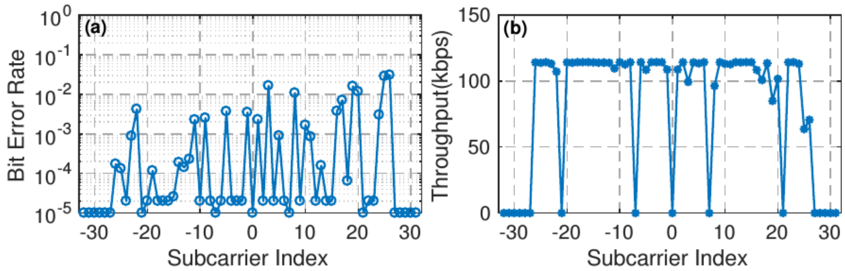

Figure 18: Capacity and concurrency with BPSK.

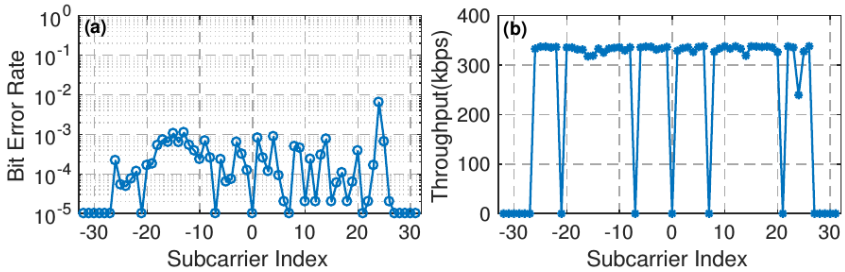

Figure 19: Capacity and concurrency with QPSK.