Mighty: Towards Long-Range and High-Throughput Backscatter for Drones

Xiuzhen Guo, *Member IEEE,* Yuan He*, *Senior Member, IEEE,* Longfei Shangguan, *Member, IEEE,* Yande Chen, *Student Member IEEE,* Chaojie Gu, *Member, IEEE,* Yuanchao Shu, *Member, IEEE,* Kyle Jamieson, *Member, IEEE,* and Jiming Chen, *Fellow, IEEE*

Abstract—While *small* drone video streaming systems create unprecedented video content, they also place a power burden exceeding 20% on the drone's battery, limiting flight endurance. We present **Mighty**, a hardware-software solution to minimize the power consumption of a drone's video streaming system by offloading power overheads associated with both video compression and transmission to a ground controller. Mighty innovates a high performance co-design among: **(1)** a ring oscillator-based, ultra-low power backscatter radio; **(2)** a spectrally-efficient, non-linear, low-power physical layer modulation and multi-chain radio architecture; and **(3)** a lightweight video compression codec-bypassing software design. Our co-design exploits synergies among these components, resulting in joint throughput and range performance that pushes the known envelope. We prototype Mighty on PCB board and conduct extensive field studies both indoors and outdoors. The power efficiency of Mighty is about 16.6 nJ/bit. A head-to-head comparison with a *DJI Mini2* drone's default video streaming system shows that Mighty achieves similar throughput at a drone-to-controller distance of up to 150 meters, with 34–55× improvement of power efficiency than WiFi-based video streaming solutions.

Index Terms—Wireless Communication, Backscatter, Drone

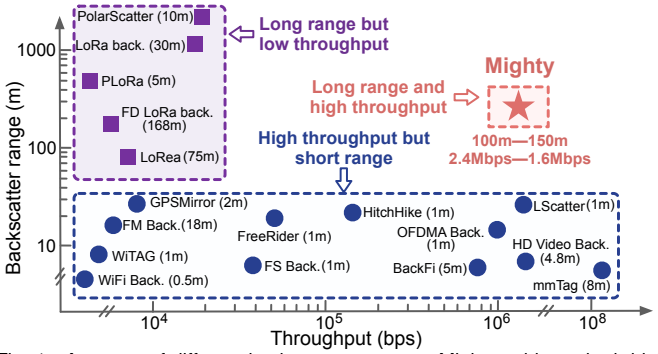


Fig. 1. A survey of different backscatter systems. Mighty achieves both high throughput and long range. The distance inside parentheses denotes the carrier generator-to-tag range required for the quoted backscatter range.

1 Introduction

Unmanned Aerial Vehicles (UAVs), also known as *drones*, are among the most disruptive innovations in the past few decades. With the miniaturization of sensors and ubiquitous wireless connectivity, small consumer-model drones equipped with cameras have become increasingly popular, creating video content of unprecedented quality. While on-board camera systems have many novel applications, such as videography and urban modeling, the use of cameras also adds weight, computation, and more importantly, communication overhead to such *small* drones, with significant associated power consumption.

Taking DJI Mini2 [1] as an example, the on-board video streaming system shoots videos with a 4K camera, processes frames in real-time, and offloads the compressed videos to a remote controller over a wireless link. These hardware components remain active as long as the drone flies, draining up to 20% extra battery power (*cf.* §2.1). Furthermore, this power overhead increases dramatically with increasing video resolution and flying distance from the ground controller, impeding the widespread deployment of such drones. Recently, however, three trends have arisen that may break this stalemate:

- Advances in backscatter technology enable radios to transmit at a few micro-watts in active mode by offloading the power-consuming carrier generator to a dedicated gateway.
- Emerging deep learning-based image recovery techniques, *e.g.*, super-resolution, are able to recover an image from low-resolution to high-resolution.
- Lightweight, energy-dense, and economical lithium ion portable power sources are arriving on the market, bolstering available energy at the drone ground station.

The case for power offloading for small drones. In this paper, we demonstrate a synergy between the above trends, resulting in an opportunity to offload the power overhead associated with the drone's video processing and transmission to the ground controller. The backscatter radio transmits video streams by modulating carrier signals sent from the controller [2]. As the power consumption of communication is dominated by the cost of generating a carrier signal, the backscatter radio successfully moves the power burden on video transmission from small drones *on the fly* to the drone controller *on the ground*. The asymmetric power supply between drone and controller makes backscatter radio a good fit for this scenario. From a video processing perspective, video frames can be sent at a lower resolution and upscaled on the controller using advanced image recovery algorithms. It allows drones to bypass power-intensive frame compression and video codec while alleviating wireless traffic critical to backscatter links.

Although power offloading is an attractive goal, realizing it in practice is challenging due to a tradeoff between our two fundamental design objectives: on one hand, we seek to build a wireless link to connect drones over long ranges, while on the other, we seek high link throughput to accommodate bulky video offloading. The tradeoff emerges as the received signal strength of backscatter signals drops sharply over long ranges (the strength of

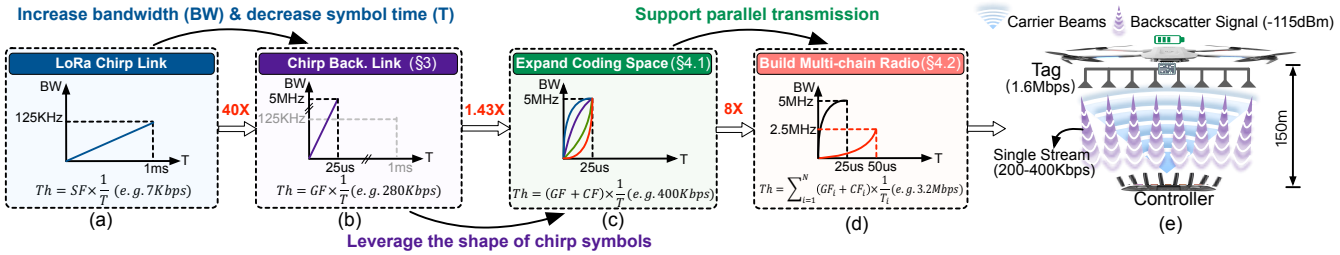


Fig. 2. A step-by-step explanation of Mighty's link throughput. (a) Conventional LoRa link achieves *e.g.*., 7 Kbps throughput. (b) Mighty improves the link throughput to 280 Kbps by reducing the symbol time from 1ms to 25 μs meanwhile increasing the symbol bandwidth from 125 KHz to 5 MHz (§3). (c) Mighty then improves the link throughput to 400 Kbps by exploring a new coding space on chirp symbols (§4.1). (d) Mighty further proposes a multi-chain radio architecture to improve the link throughput from 400 Kbps to 1.6 Mbps at a range of 150 m (§4.2). The strength of received backscatter signals is around -115 dBm.

backscatter signals can easily fall under the ambient noise floor after a round-trip attenuation), leading to a low link throughput. Reserving a large bandwidth to increase link throughput is not always feasible, particularly in the overcrowded unlicensed band.

State-of-the-art backscatter systems either target shorter communication ranges to achieve a higher link throughput [3], [4], [5], [6], or improve communication range at the cost of throughput [7], [8], [9], [10], [11]. Analog High Definition (HD) Video Backscatter [4] enables 30 fps 1080p video streaming at up to 8 feet distance, and 10 fps 720p video streaming at up to 16 feet distance. mmTag [5] takes advantage of the ultra-high bandwidth of millimeter-wave to achieve up to hundreds of Mbps link throughput. However, its effective operating range is *ca.* 8 meters due to signal attenuation over distance. On the other hand, the link throughput of long-range backscatter systems such as LoRea [7], PLoRa [9], and LoRa backscatter [10] is a few Kbps, too low to offload videos. Figure 1 summarizes the field of related work: to our best knowledge, no prior work simultaneously satisfies our dual requirements of high throughput and long range.

In this paper, we present **Mighty**, a hardware/software solution that reduces video streaming power consumption through hierarchical power offloading. As shown in Figure 2, Mighty achieves this by making technical innovations in the hardware layer, physical layer, and application layer, respectively.

• **Hardware-layer**. We propose a chirp-based backscatter radio design that enables the drone to offload videos to the controller 150 m away at a few micro-watts power consumption. Mighty is based on a key observation that the noise resilience of the chirp symbol is proportional to the multiplication of its symbol time and bandwidth [12], [13]. By reducing the chirp symbol time while increasing the chirp bandwidth, we are expected to see remarkable growth in link throughput without hurting the link distance. For instance, compared to a standard LoRa link with 7 Kbps throughput, our design achieves a maximum 280 Kbps throughput at a communication range of 150 m by reducing the symbol time from 1 ms to 25 μs while increasing the chirp bandwidth from 125 KHz to 5 MHz. We take advantage of the ring oscillator's low-power nature to synthesize chirp symbols, meanwhile pushing the limits of ring oscillator's frequency resolution using an ultra-low power voltage converter.

• **Physical-layer**. To accommodate video streaming, we make two innovations to improve the link throughput further. First, we leverage the shape of chirp symbols—a coding space orthogonal to the conventional chirp modulation—in a new way that increases a single backscatter link's spectral efficiency, and hence throughput. For instance, by encoding three more bits per chirp symbol using different shapes, the maximum link throughput grows from 280 Kbps to 400 Kbps. Second, we propose a multi-chain architecture that allows the drone to transmit multiple data streams concurrently (Figure 2).[1] By harvesting the above two innovations, Mighty improves the link throughput further to 1.6 Mbps at a radio range of 150 m.

TABLE 1
Mighty saves up to 85% power of drone video streaming by redesigning the video processing and transmission modules.

	Conventional camera pipeline [1], [16]	Our design
Imaging	CMOS array+ADC / 585mW	CMOS array+ADC / 585mW
Processing	Compression+Codec / 3750mW	Frame selection / 640 mW
Communication	WiFi/LTE radio / 1500-3000mW	Backscatter / 8.7-69.8mW
Total	5835-7335mW [1], [16]	1234-1294mW

• **Application-layer**. Although the throughput of backscatter links get improved remarkably with the above hardware and PHY-layer innovations, it is still too low to accommodate bulky video streaming. To address this issue, we replace the power-intensive video codec and compression on drones with a lightweight key frame selection algorithm. This allows the drone to transmit a few interest frames at low resolution to fit the capacity of the underlying backscatter link. We then customize an advanced super resolution algorithm DRUNet [15] and deploy it on the ground controller to recover high definition (HD) frames from these interest frames. The customized model takes into account the backscatter link dynamics when recovering images from their noisy, low-resolution receptions.

Experiment results. Mighty's physical layer and hardware innovations together enable the throughput comparable to a drone's active radio, 4.2–1.6 Mbps as radio range grows to 150 meters, 30–82× higher than existing long-range backscatter systems [10]. The head-to-head comparison with the drone's default video streaming system shows the marriage of video compression bypassing design and the backscatter radio reduces the power consumption by around 85% (Table 1).

2 Background

In this section, we first analyze the power overhead of a small drone's video streaming system (§2.1). We then survey literature directly related to our design (§2.2).

2.1 Power Overhead Analysis

The drone video streaming system (VSS) consists of three components: imaging, processing, and communication. The imaging part captures images using ultra-low power CMOS arrays. The analog pixels value (*i.e.*, voltage signals) output by each CMOS sensor

[1] In our radio architecture, each individual radio chain is connected to a different antenna. The range of each chain will not suffer with an increasing number of chains because the power of excitation signal arriving at each radio chain is determined by the cross-sectional area of its own antenna [14].

gets digitized through analog to digital converters (ADCs). VSS then compresses video frames in the digital domain before sending them to the controller through wireless links.

To better understand the design space, we program a *small* drone DJI Mini2 [1] to transmit a 4K video at 30 fps and measure the power consumption of the above modules in active mode. The camera pipeline consists of three parts: imaging, processing, and communication. We use a Keysight U8485A thermocouple power sensor to measure the energy consumption of each functional unit. The result is shown in Table 1. The imaging part consumes a few hundred mW due to the use of a high-sampling rate ADC (10 MHz). The optical lens and CMOS arrays are extremely low-power [17]. The video processing module involves computationally intensive frame compression and thus consumes 6.4× more power than the imaging part. Similarly, the communication module relies on power-consuming DAC and power amplifiers, and consumes a maximum power of 3 W power, 5× higher than the imaging part. In summary, the video processing and communication modules account for 90% of the overall power consumption of the drone's video streaming system.

2.2 Related Work

Low-power camera. Prior works can be broadly divided into two groups. The first group focuses on designing self-powered cameras to avoid battery replacement. These systems need to be placed near the carrier source to ensure a higher energy harvesting efficiency. The second group of works is on designing an event-driven camera [18], [16], [19] that looks for specific events and turns on the primary imaging pipeline if necessary. Our design optimizes both the video frame processing pipeline and the underlying wireless radio architecture to minimize power consumption.

Backscatter systems. With the development of Internet of Things [20], [21], we have witnessed remarkable advances in backscatter technology and application [22], [4], [2], [23], [24]. Existing works either leverage a higher bandwidth or improve the spectrum efficiency to increase link throughput. OFDMA-WiFi [3] achieves 5.2 Mbps link throughput by using spectrum-efficient OFDMA modulation. LScatter [25] achieves up to 13.6 Mbps link throughput by modulating LTE traffic at tens of nanoseconds per symbol. mmTag [5] explores the ultra-wide bandwidth at millimeter wave band to achieve hundred of Mbps throughput. PolarScatter [26] enables reliable wide-area backscatter networks by exploiting channel polarization based on polar codes, which achieves up to 11.5× throughput gain and extends the communication range by 1.9× compared with the state-of-the-art long-range backscatter. GPSMirror [23] leverages passive and ultra-low-power backscatter tags to enable meter-level GPS positioning for unmodified mobile devices and the GPSMirror tag can provide coverage up to 27.7 m.

Long-range backscatter designs are mostly based on spectrum spreading modulation: LoRea [7] adopts frequency shifting to expand the range to a few kilometers. LoRa backscatter [10] supports kilometer-scale communication by synthesizing chirp signals from a sinusoidal tone. PLoRa [9] synthesizes standard LoRa packets using ambient LoRa transmissions. However, the backscatter range drops remarkably with increasing distance between signal generator and tag [27]. Besides, the throughput of these systems is limited to tens of Kbps and thus cannot accommodate video streaming.

Our design also shares the similarity with another group of works that replaces active radios with backscatter radios to save power. WISPCam [28] uses an RFID reader to capture and offload low-resolution images to the gateway. HD video backscatter [4] proposes an analog backscatter that allows the camera to bypass the image processing. However, the range of these designs is limited to a few meters. LF-backscatter [29] allows an IoT device to switch between active and passive transmission mode depending on its battery. Likewise, Morpho [30] develops an active-passive radio that allows an IoT device to adapt transmission to link dynamics. However, both systems are designed for short-range wireless communication and thus cannot be applied to our scenario.

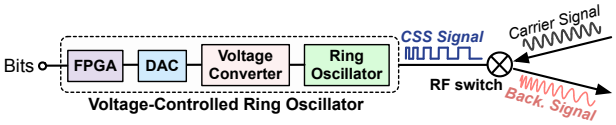


Fig. 3. An overview of Mighty's backscatter radio.

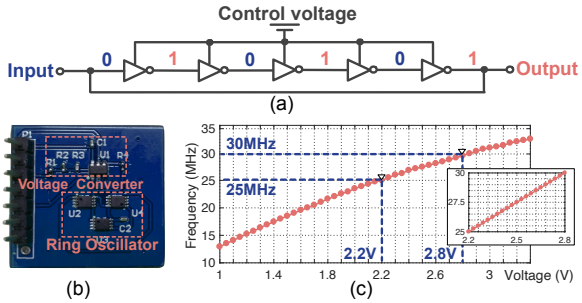


Fig. 4. Mighty's ring oscillator. (a) Schematic. (b) Hardware prototype. (c) Output frequency vs. control voltage.

In addition, CurvingLoRa [31] uses non-linear chirps to multiplex concurrent LoRa transmissions. In contrast, Mighty exploits these chirp shapes when they overlap in both space, frequency, and time, over a single backscatter link, to add a new signal dimension for communication, in order to increase backscatter throughput and power efficiency. FS-backscatter [32] uses a ring oscillator to generate a constant 20 MHz frequency shifting signal. Unlike FS-backscatter, Mighty leverages the ring oscillator to generate chirp symbols whose frequency grows continuously over time. This requires precise voltage control to generate chirps with different initial frequency offset in the granularity of tens of KHz (§3.3).

3 Long-range Backscatter Link

3.1 Why Choose CSS Modulation?

Mighty explores backscatter technology to reduce the drone's communication overhead while retaining high link throughput for video streaming. To satisfy the long-range and high-throughput dual requirements, a nature question to our design is *which modulation is best suited for our scenario*?

Chirp spread spectrum (CSS) uses its entire allocated bandwidth to modulate a signal , making it robust to narrow-band interference and multi-path fading. *The processing gain of a CSS symbol is proportional to the multiplication of its symbol time and bandwidth* [12], [13]. For example, the link throughput can be improved by 10× by reducing the symbol time by 1/10 while increasing the chirp bandwidth by 10×, without sacrificing the processing gain (*i.e.*, communication range). This unique property allows the transmitter to trade-off the symbol time and bandwidth to satisfy different applications' requirement. We thus adopt CSS modulation in our design. Figure 3 shows the workflow of Mighty tag.

3.2 Synthesize Chirp Symbols at a Few μWs

Conventional chirp generation methods are ill-suited to our system because they heavily rely on power-consuming hardware components such as high-precision clocks and phase lock loop.

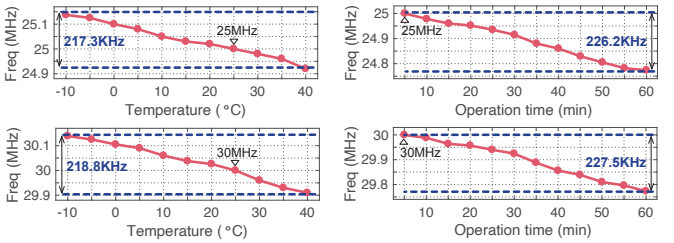
Generating chirps using a ring oscillator (RO). To alleviate power consumption, we adopt *ring oscillator*, an ultra-low power frequency synthesizer to generate chirp symbols. A ring oscillator consists of an odd number of inverters in a loop with the output of the last stage inverter fed back to the input of the first, as shown in Figure 4(a). Since the output signal of the last stage inverter has a reversed logic as the input of the first stage inverter, the circuit oscillates continuously. The frequency of the ring oscillator is determined by the delay of each inverter which can be further controlled by the input voltage to this ring oscillator. Figure 4(b) shows our ring oscillator prototype consisting of three TI SN74AUP3G04 low-power triple inverter gate [33]. We measure its frequency output at different input voltage settings (Figure 4(c)). The frequency output grows linearly with the growing input voltage, forming a coherent chirp symbol. As the input voltage grows gradually from 2.2 V to 2.8 V over 50 μs, the frequency output grows from 25 MHz to 30 MHz, following the same pace.

What if the ring oscillator is not stable? The frequency of ring oscillator is sensitive to temperature variations [32]. As shown in Figure 5(a), we observe 218.8 KHz frequency offset as the board temperature grows from –10° to 40° C. We also examine the impact of operation time and plot the result in Figure 5(b). The result shows that the frequency output varies up to 227.5 KHz over a course of 60 minutes. In our system, the backscatter radio adopts 5 MHz chirp bandwidth to achieve a better trade-off between noise resilience and throughput. Since the receiver's sampling rate is usually at tens of MHz, it can well capture the backscatter signal in the presence of frequency shift. The receiver can then leverage synchronization algorithms such as Schmidl-Cox [34] to compensate the frequency offset and decode the backscatter signal accordingly. In practice, the frequency shift is way lower than 227.5 KHz because the ambient temperature is unlikely to vary abruptly throughout the journey (*e.g.*, less than half an hour for most drones). Hence the ring oscillator's instability will not ruin the backscatter communication.

Contribution of the ring oscillator design. The ring oscillator is an ultra-low power clock synthesizer to generate oscillation signals. Existing works, such as FS-backscatter [32], leverage the ring oscillator to generate a 20 MHz oscillation signal to shift the frequency band of the backscatter signal to a different frequency band from the carrier signal. Different from existing works, Mighty is the first-of-its-kind work to exploit the ring oscillator to generate the chirp signals. The frequency output of the ring oscillator grows linearly with the growing input voltage, forming a coherent chirp symbol. Hence, Mighty *can synthesize chirps with different initial frequency, bandwidth, and symbol time by varying the starting voltage, ending voltage, and the time-span of input voltage.*

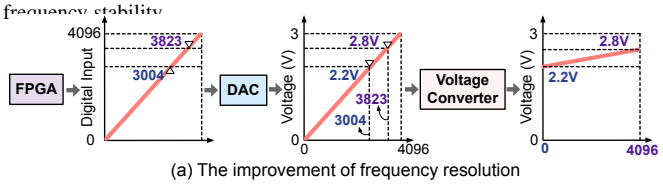
3.3 Boost the RO's Frequency Resolution

Similar to LoRa PHY-layer design [35], our backscatter radio encodes data by varying the initial frequency of a chirp symbol. We define ***Granularity Factor (GF)*** as the number of bits encoded on a chirp by varying the chirp's initial frequency offset. The maximum number of GF is determined by the ring oscillator's *frequency resolution*. A higher frequency resolution yields a higher throughput. In our system, the ring oscillator's frequency

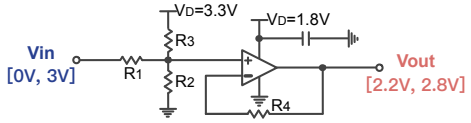


(a) Frequency vs. temperature (b) Frequency vs. operation time

Fig. 5. Impact of temperature and operation time on the ring oscillator's frequency stability



(a) The improvement of frequency resolution



(b) Voltage converter

Fig. 6. Frequency resolution improvement of the ring oscillator. (a) The voltage converter scales down the range of input voltage to improve the frequency resolution of the ring oscillator. (b) The schematic of our voltage converter circuit.

is controlled by its input voltage, which is generated by a digital-to-analog converter (DAC). Blindly using a high-precision DAC in hope of improving the ring oscillator's frequency resolution is not feasible because the power consumption of DAC grows exponentially with its precision. For instance, a 14-bit DAC consumes 20× more power than a 12-bit DAC.

The frequency resolution of a ring oscillator is represented by $BW/(I_{end} - I_{start})$, where BW is the chirp bandwidth. I_{end} and I_{start} represent the maximum and the minimum value of a DAC's input. When the input of a 12-bit DAC grows linearly from 3004 to 3823, the output voltage will grow from 2.2 V to 2.8 V, leading to a frequency growth of 5 MHz (Figure 4(c)). The frequency resolution thus equals $\frac{5\ MHz}{3823-3004} = 6.1\ KHz$. The above equation reveals two possibilities to improve the ring oscillator's frequency resolution: minimize BW or maximize $I_{end} - I_{start}$. However, since the bandwidth BW also scales with the DAC's input, maximizing the denominator may not necessarily improve the ring oscillator's frequency resolution. As shown in Figure 4(c), when the DAC's input grows from 0 to 4096, the output voltage will grow from 0 V to 3 V. Feeding this 3 V voltage span into the ring oscillator will produce a chirp with 30 MHz bandwidth. In such a case, the frequency resolution drops to $\frac{30MHz}{4096-0}$=7.32 KHz.

We design a low-power voltage converter to improve the ring oscillator's frequency resolution. As shown in Figure 6(a), the voltage converter takes the voltage output from the DAC as the input, downscales it to a smaller range before feeding it into the ring oscillator. This allows the backscatter radio to expand the input range of DAC without worrying about the bandwidth growth. Let's take a retrospect of the example shown in the previous paragraph. The growth of the DAC's input (from 0 to 4096) leads to a voltage change of 3 V. The voltage converter then kicks in, downscaling the 3 V voltage change to 0.6 V ([2.2 V,2.8 V]), yielding 5 MHz chirp bandwidth. As a result, the frequency resolution grows to $\frac{5\ MHz}{4096-0} = 1.2\ KHz$, 6× higher than before ($\frac{30MHz}{4096-0}$=7.32 KHz). Accordingly, the maximum number of bits that can be encoded on

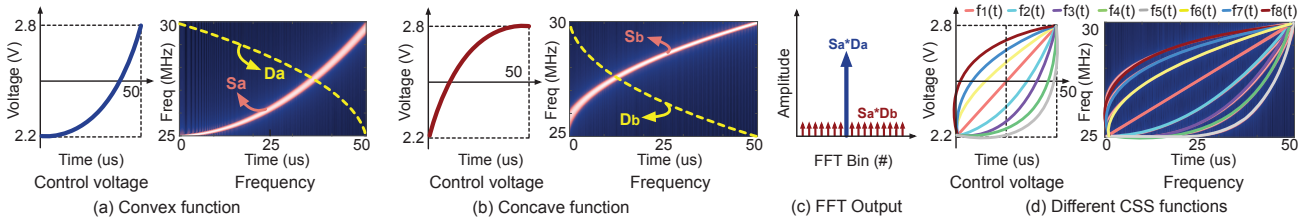


Fig. 7. Encoding extra information on a chirp by modulating the shape of this chirp symbol. (a) A convex chirp symbol. (b) A concave chirp symbol. (c) The energy scattering and converging effect. (d) Eight chirps in different shapes.

this chirp (GF) grows from 9 to $\lfloor \log_2(4096-0) \rfloor = 12$.

Hardware implementation. The voltage converter circuit shown in Figure 6(b) consists of three resistors with $R_1 = 1\ M\Omega$, $R_2 = 1.5\ M\Omega$, $R_3 = 300\ K\Omega$. Its output can be represented as $V_{out} = \frac{1}{5}V_{in} + 2.2$. The power consumption of resistors' dissipation is 19.3 μW.

4 Improve the Link Throughput

In this section, we further propose two innovations on the PHY-layer and hardware-layer respectively to improve the link throughput at the long link distance settings.

4.1 Expand the Coding Space

We seek to encode more information on a chirp symbol without hurting the chirp's noise resilience. In approaching such a design, we are inspired by the orthogonality of non-linear chirps, as Figure 7 shows (*cf.* §2.2 for a comparison with other work that uses non-linear chirps [31]). Let S_a and S_b be a convex and concave chirp, respectively. We further assume D_a and D_b be the complex conjugate of S_a and S_b. When S_a multiplies with D_a, the energy of S_a will converge to a single FFT bin, emerging an FFT peak. In contrast, when S_a multiplies with D_b, its energy will be spread over multiple FFT bins, where the overall energy peaks are inherently weak (Figure 7(c)). This unique energy scattering and converging phenomenon allows the receiver to recognize a chirp in different shapes by checking the energy pattern of its multiplication with different down-chirps in the frequency domain. The frequency output of a ring oscillator changes with its input voltage, allowing us to generate chirp symbols with different shapes by varying the ring oscillator's input voltage.

This observation motivates us to explore the chirp shape as an orthogonal coding space to improve the link throughput. We define ***Curving Factor (CF)*** as the number of bits encoded by varying the shape of a chirp. For instance, the use of eight chirps in different shapes allows us to encode extra CF=3 bits on the chirp symbol (Figure 7(d)). To facilitate the chirp generation, we design different polynomial functions to control the input voltage of the ring oscillator. To demodulate these CF bits information, the receiver multiplies the incoming chirp with the complex conjugate of these eight chirps respectively and detects the presence of a single FFT peak among all eight multiplication results. It then tracks the position of this FFT peak to demodulate the GF bits information encoded by the initial frequency offset of this chirp.

How large is this coding space? Theoretically, we can generate numerous chirp symbols in different shapes by using different polynomial voltage control functions. However, in practice, the coding space is limited because as the number of chirps grows, a new chirp symbol will increasingly resemble one of the old ones. After de-chirping, this pair of symbols would generate similar energy peaks on the frequency domain and thus confuse the demodulator. We run extensive benchmarks to understand this coding space in different channel bandwidth and symbol time settings. The results show that when the symbol time is set to 300, 100, and 50 μs with a chirp bandwidth of 5 MHz, the maximum number of bits that can be encoded is 5, 4, and 3, respectively.

4.2 Build a Multi-chain Backscatter Radio

We take the above backscatter design as the fundamental block and propose a multi-chain backscatter by stitching multiple RF switches together. The overall throughput of this backscatter radio scales with the number of chains. The communication range will not suffer with the number of chains because each radio chain is connected to a different antenna and the power arriving at each radio chain will not be split. One can shift the backscatter signal to non-overlapping channels away from the carrier signals to avoid interference. However, generating backscatter chirps across multiple channels aggravates the power burden at the tag because the power consumption scales with the number of clock frequencies. In addition, this approach also introduces additional wireless spectrum usage and deployment overhead.

In Mighty, multiple bit streams are transmitted on the same frequency band to improve the channel efficiency. The side effect, however, is severe symbol collisions at the receiver end. Motivated by the spreading factor design in LoRa [35], we allow each RF chain to modulate data using different chirp symbol time and bandwidth to ensure the receiver can successfully demodulate these collided backscatter streams. Figure 8 shows the workflow of the multi-chain backscatter radio.

- **Data stream segmentation**. The backscatter tag first segments data into multiple streams and allocates them to different radio chains using a serial-to-parallel converter. Each radio chain adopts a different bandwidth and symbol time to ensure that the receiver can demodulate the collision symbols.
- **Modulation**. On each radio chain, the backscatter tag encodes data by varying both the initial frequency offset and the shape of the chirp symbol. This is achieved by adjusting the input voltage to the ring oscillator. The codeword is divided into two parts, with the former part encoded by the shape function of the chirp and the latter encoded by the chirp's initial frequency offset, as shown in Figure 8(b).
- **Demodulation**. The drone's underground controller stores a group of down-chirp symbols with different shapes (Figure 8(c)). It demodulates the video stream by multiplying each down-chirp with the received backscatter symbols. It then detects the single FFT peak on the multiplication result to decode the CF bits. Subsequently, it tracks the position of this single FFT peak to decode the GF bits. Finally, the ground controller combines the decoded information from eight RF chains using a parallel-to-serial converter (Figure 8(d)).

Packet detection and synchronization. Following the LoRa packet format, we construct a preamble for each backscatter packet

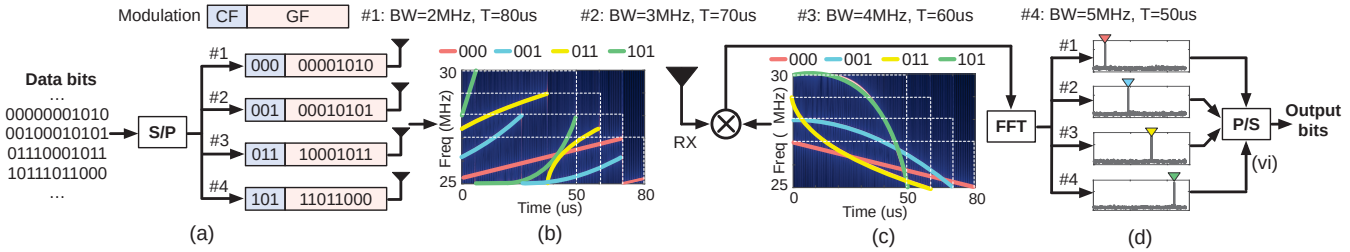


Fig. 8. The modulation and demodulation of Mighty. (a) The image data is segmented into multiple streams and allocated to different radio chains using a serial-to-parallel (S/P) converter. (b) Each radio chain modulates data streams based on the proposed modulation schemes. (c) The receiver demodulates each data stream through de-chirping. (d) The receiver combines the decoded information from each RF chain using a parallel-to-serial (P/S) converter.

using 10 linear up-chirps, followed by 2.25 linear downchirps as syncword (SFD). This preamble design allows the Tx&Rx to achieve symbol-level synchronization using standard cross-correlation [36]. Motivated by CoLoRa [37] and NELoRa [38], we further estimate and compensate for the carrier frequency offset and sampling time offset based on the dechirping results of preamble and SFD, respectively. To avoid self-interference, the backscatter radio shifts the backscatter signal to a non-overlapping channel 25MHz away from the carrier signal.

Self-interference. We adopt the method of frequency shifting, where the RF switch with the switching rate of 25MHz moves the backscatter signals to a non-overlapping frequency band, to avoid the self-interference at the Tx&Rx.

4.3 Link Throughput Analysis

We analyze Mighty's throughput step by step. (1) The throughput (Th) of chirp modulation-based backscatter link depends on the Granularity Factor (GF) and the symbol time (T), *i.e.* $Th = GF \times \frac{1}{T}$. (2) By introducing the shape of chirp symbols as an orthogonal coding space to encode extra Curving Factor (CF) bits, the throughput grows to $Th = (GF + CF) \times \frac{1}{T}$. (3) By incorporating multiple backscatter chains, the overall throughput further grows to $\sum_{i=1}^{N}(GF_i + CF_i) \times \frac{1}{T_i}$, where N is the number of radio chains, GF_i, CF_i, and T_i are granularity factor, curving factor, and symbol time of the i^{th} chain.

5 Power Offloading for Video Processing

Video streaming over backscatter link consumes 480× less power than using the drone's active radio. However, transmitting an HD (1080p) raw video at 30 fps requires 497.7 Mbps throughput if every pixel is encoded by eight bits, which far exceeds the throughput of both backscatter and active radio links. The default video processing module on drones adopts video codec and compression that reduce throughput to a few Mbps at the cost of significant power consumption (§2.1).

We propose a video compression-bypassing mechanism (§5.1) that allows the drone to transmit only a few selected raw frames at a lower resolution. We then propose a link quality-aware super-resolution algorithm to recover HD frames from the noisy receptions and further leverage frame interpolation to synthesize missing frames (§5.2), as shown in Figure 9.

5.1 Simplify the Video Processing Pipeline

We take advantage of frame redundancy to minimize the power consumption of on-board video processing. We bypass the power-consuming video codec and compression by selecting sporadic interest frames from the raw video stream.

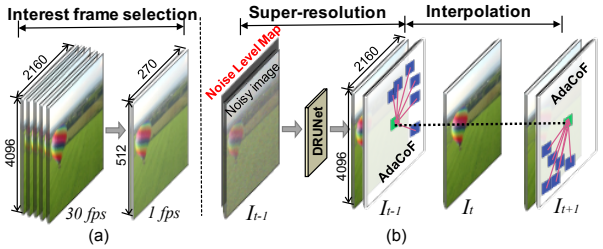


Fig. 9. Power offloading for video processing. (a) The algorithm selects interest frames and transmits them at a lower resolution ([4096×2160 30fps] → [512×270, 1fps]). (b) The controller leverages a super-resolution algorithm to recover the high-resolution frame ([4096×2160, 1fps]) from the raw reception and then interpolates missing frames.

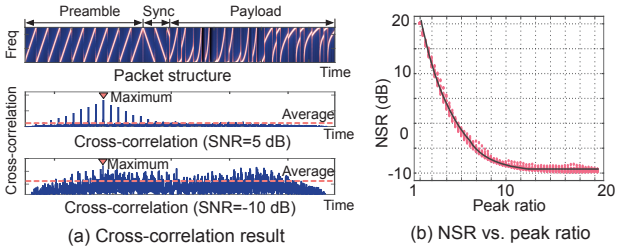


Fig. 10. The relationship between the cross-correlation result and the noise level map.

Interest frame selection. We adopt *optical flow coupled with perspective guidance* to select interest frames due to the following two reasons. First, compared to those bulky deep learning [39] or image hashing-based designs [40], the optical flow achieves decent frame selection performance with less computation overhead. Second, the combination of perspective guidance allows the frame selection algorithm to address the impact of the drone's motion, jitter, and perspective transformation on the optical flow vectors.

The algorithm first extracts 32 feature points from the current frame #$(n+1)$ with Shi-Tomasi corner detection algorithm [41]. For each feature point and its 7×7 neighborhood pixels, the proposed solution then leverages L-K Algorithm [42] to calculate the optical flow vector of each feature point between two consecutive frames, say frame #n and #$(n+1)$. A frame is denoted as an interest frame as long as its optical flow does not conform to the perspective law (checked by Epipolar Geometry [43]).

5.2 Recover HD Video on the Controller

Upon receiving an interest frame from the drone, the controller first leverages a super-resolution algorithm to recover high-quality frames from noisy data. It then adopts frame interpolation to compensate for missing frames.

Single image super-resolution (SISR). The received interest frame is likely to be noisy due to signal attenuation over the backscatter link. We adopt DRUNet [15], a cascade model

consisting of image denoising and super-resolution two parts, and make significant customization to ensure its decent performance in our scenario. The raw image is first fed into the denoiser where the noise level map is leveraged to compensate for the image noises and improve its Peak Signal to Noise Ratio (PSNR).

• **Build the noise level map.** The default noise level map of DRUNet characterizes imaging noises introduced by electronic components of CMOS array. These noises are relatively stable. However, the image noise in our system mostly comes from bit errors caused by the link noise that may vary drastically due to the drone's movement. We build a dynamic noise level map to characterize link noise dynamics.

The impact of link noise is inversely proportional to the signal-to-noise ratio (SNR) of the received signal. The higher the impact of link noise, the lower the SNR, and vice versa. We estimate the impact of link noise (*i.e.*, noise-to-signal ratio, NSR) by measuring the SNR of the received signal. Unfortunately, the measurement of SNR is inaccurate in long-range backscatter systems because the received backscatter signals are easily buried in ambient noises. We instead use the maximum cross-correlation coefficient obtained in packet detection as the proxy of the received signal strength because the former is proportional to the latter [44] (Figure 10(a)).

To minimize the impact of noise, we normalize this value by computing the ratio of the maximum and the average cross-correlation coefficient, termed as peak ratio. The higher this peak ratio, the lower the impact of link noise (*i.e.*, NSR). We measure the peak ratio in different NSR settings offline and interpolate the results, as shown in Figure 10(b). The resulting interpolation function allows us to estimate the NSR using the measured peak ratio. Upon receiving a new packet, the controller calculates the peak ratio using the packet preamble. It then estimates the NSR and uses it to establish the noise level map.

Video frame interpolation. We leverage AdaCoF, a light-weight DNN [45], to interpolate missing frames. AdaCoF takes advantage of the kernel-based approach and flow-based approach, and thereby applies to a broad domain involving complex motions. This is the most important consideration for Mighty to select AdaCoF network to recover the video stream from the small drone.

6 Implementation

We prototype Mighty on a Raspberry Pi 4 equipped with an HD camera AR1335 [46] and an 8-chain backscatter tag, as shown in Figure 11(a). The total weight is around 180 g. The backscatter tag is implemented on PCB hardware using commercial off-the-shelf components and a low-power GW1N-UV9QN48 [47] FPGA. The FPGA controls the voltage input of the ring oscillator through a 12-bit DAC TLV5619 [48]. The weight and form factor of our prototype can be reduced dramatically after ASIC fabrication.

Controller. We implement a controller using three software defined radios (SDR) USRP N310. These SDRs are connected to a batch of omnidirectional antennas with 3 dBi gain. As shown in Figure 11(c), two transmitter SDRs are time synchronized by using an Octoclock-G GPS disciplined oscillator (GPSDO) with a 10 MHz reference signal. We set the sampling rate of the receiver SDR to 25 MHz. The decoding algorithm is implemented in MATLAB.

Beamformer. We build a transmitter beamformer for the underground drone controller using an 8-antenna array shown in Figure 14(a). The transmitter updates the phase of carrier signals based on the path difference between each antenna and drone, thus enhancing the carrier strength at the drone. This is feasible because the drone equipped with a Global Positioning System (GPS) and Inertial Navigation System (INS) can inform the ground controller its real-time location and moving speed. This eight-antenna beamformer brings 7.2-8.5 dB gain to the transmitter signal. The beamformer gain fluctuates slightly with the distance from the drone, as the phase offset compensation accuracy among different antennas varies with the distance.

Transmitter power setups. FCC [49] requires the output power of the transmitter being fed into the antenna should be no more than 30 dBm (1 W). Moreover, for the transmitter with multiple antennas, its maximum allowable Effective Isotropic Radiated Power (EIRP) is 36 dBm. Following the FCC guideline, we set the transmitter power to 15 dBm. The output power fed into the antenna array is $P = P_{single} + 10lgN = 15 + 10lg8 = 24dBm$, where P_{single} and N are the power fed into every single antenna and the number of antennas (N=8), respectively. The directional gain (*a.k.a.*, beamforming gain) brought by the antenna array is $G = G_{single} + 10lgN = 3 + 10lg8 = 12dBi$. The EIRP of our prototype, as shown in Figure 11(d), is $P + G = 24 + 12 = 36dBm$, complying with the FCC requirement.

Video recovery. We implement both super-resolution and frame interpolation modules in PyTorch. We run these machine learning models on a desktop equipped with an Intel Xeon 3.5 GHz 4-core CPU and an Nvidia Titan Xp GPU. These models take 0.73 s to process a 1080p video streaming transmitted at 30 frames/second. We envision the advanced model pruning and compression techniques [50], [51] would facilitate the deployment of these bulky models on the drone's controller. We adopt the pre-trained model from [52], [53] and replace the default noise level map with our link quality-aware customization. The effectiveness of these pre-trained models has already been demonstrated in diverse environments, including cities, countryside, forests, architecture, streets, *etc.*.

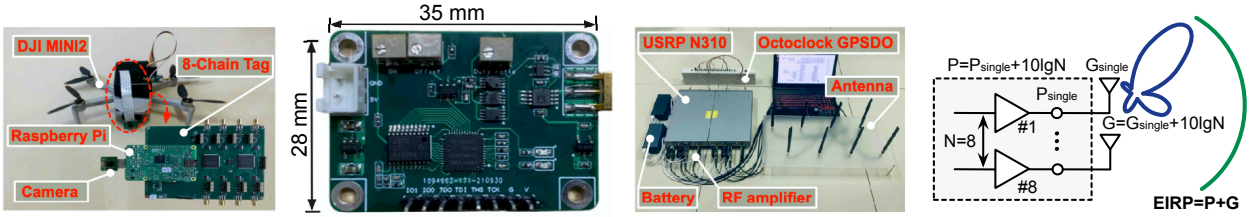
7 Evaluation

We evaluate Mighty on video transmission (§7.1), networking performance (§7.2), and power consumption (§7.3). We also conduct a head-to-head comparison with Bluetooth and LoRa (§7.4) on the communication range, link throughput, and power efficiency. Unless otherwise posted, the transmitter and the receiver are collocated on the ground. The controller transmits sinusoidal carriers on the 470 MHz[2] frequency band and receives backscatter signals on the 495 MHz frequency band with 25 MHz frequency shifting.

Evaluation baselines and metrics. Existing long range backscatter designs [10], [9], [7] achieve merely up to tens of Kbps throughput. They cannot serve as good baselines for video offloading. Accordingly, we conduct head-to-head comparisons with the DJI Mini2 drone's default video streaming system both indoors and outdoors using the following metrics.

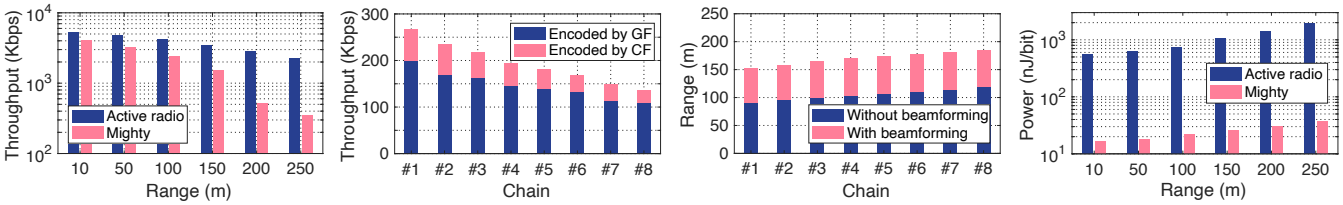
Evaluation setting. We conduct experiments both indoors and outdoors. For outdoor experiments, we assess Mighty's performance in various weather conditions (windy, rainy, and sunny days) in an open field. The drone is hovering over fixed locations to transmit the video streaming. The noise floor in the outdoor experiments is around -90 dBm to -100 dBm due to the relatively clear channel of 470 MHz. For the indoor experiments, we put a Mighty tag in the hallway and move its distance from the carrier source. The noise floor in the outdoor experiments is around -60 dBm to -70 dBm

[2] Working on 2.4 GHz, 5 GHz, and 900 MHz ISM band may introduce severe interference to other ongoing communication traffic of the drone.



(a) Prototype (180g) (b) Single-chain tag (c) Controller-TX (d) Transmission power

Fig. 11. (a) Mighty prototype. (b) The single-chain backscatter radio. (c) USRP-based controller. (d) Transmission power. The development of the miniaturization of antenna technology will further bolster the fabrication factor of the ground controller.



(a) Throughput (b) Throughput of different chains (c) Range of different chains (d) Power consumption

Fig. 12. Head-to-head comparison with the drone's onboard active video transmission systems.

since there may be other interference signals in the same frequency band.

• **BER (Bit Error Rate)** refers to the ratio of error bits to the total number of bits received by Mighty.

• **Throughput** measures the amount of received data correctly decoded by Mighty within one second.

• **Power efficiency** measures the amount of energy required to transmit a single bit of data. It is equal to the ratio of the power consumption to the corresponding throughput when transmitting video streams.

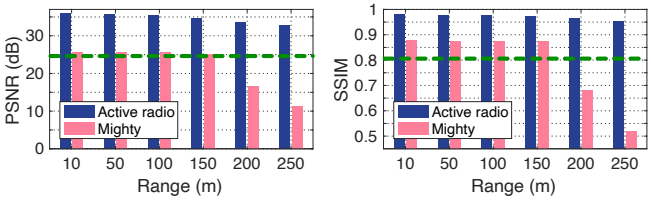
• **PSNR (Peak Signal-to-Noise Ratio)** compares two images by measuring the mean squared error of the corresponding pixel values across the entire image and returns its inverse.

• **SSIM (Structural Similarity Index)** compares two images based on three parameters: structure, contrast, and luminance.

7.1 Field Studies

We put the drone in different locations for precise ranging. The drone uses both its active radio and Mighty to transmit the same 4K video for evaluation. The bandwidth (BW) and curving factor (CF) are set to 5 MHz and 3, respectively. Mighty varies the chirp symbol time from 10 μs to 110 μs and granularity factor (GF) from 5 to 12 to balance the throughput and communication range.

Link throughput. We gradually increase the distance between the controller and the drone to measure the link throughput. As shown in Figure 12(a), Mighty achieves comparable throughput with the drone's active radio in short-range settings. For instance, when the drone is 10 meters away from the controller, Mighty achieves 4.2 Mbps throughput, 1.1 Mbps lower than active radios. As we increase the range, the throughput gap between the active radio and Mighty grows gradually. It ends up with 2.3 Mbps when the drone is placed 250 m away from the controller. This is expected since the strength of backscatter signals suffers a round-trip attenuation and the active system only suffers path loss once. On the other hand, the throughput of both systems declines with the distance between the drone and the controller. For instance, when the distance grows from 10 meters to 250 meters, we have to increase the symbol time from 10 μs to 110 μs. The increased symbol time leads to a lower throughput but extends the communication range. Nevertheless, Mighty still achieves 1.58 Mbps throughput at the range of 150 m.



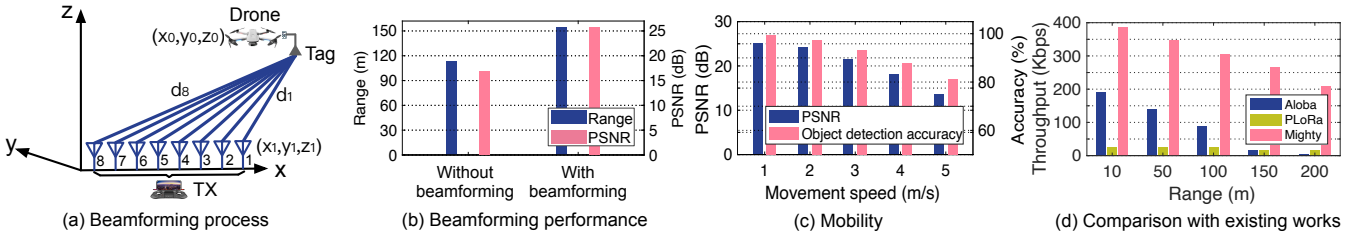
(a) PSNR (b) SSIM

Fig. 13. Performance of video quality. The green line indicates the minimum PSNR and SSIM required for video recovery.

Ablation study of different chains. We take the link throughput result (1.58 Mbps at 150 m) as an example to understand the contribution of each radio chain. As shown in Figure 12(b), we have the following observations. The granularity factor (GF) achieves the throughput of 200–107 Kbps for different chains because each chain adopts a different symbol time to ensure the orthogonality of data streams. By encoding three more bits with curving factors (CF), the throughput of each radio chain grows by 67 Kbps–28 Kbps. The total throughput of different chains varies from 267 Kbps to 135 Kbps and the integrated throughput of eight chains is 1.58 Mbps. In addition, the radio chains (links) achieve consistent long range (at around 100 m) as shown in Figure 12(c). The beamformer further improves the communication range to around 150 m.

Power efficiency. We compare the power efficiency of these two systems as shown in Figure 12(d), Mighty consumes orders of magnitude lower power than the active radio. Specifically, when the radio range is set to 10 m, Mighty achieves 16.6 nJ/bit power consumption, 34× lower than active radio (558.7 nJ/bit). When the range increases to 250 m, Mighty achieves 55× lower power consumption than the active radio (46.5 nJ/bit vs.2560 nJ/bit).

Video quality. By default, the DJI Mini2 drone's controller runs a frame recovery algorithm to improve the quality of frame receptions. We compare the video quality recovered by both the drone's default video recovering algorithm and Mighty's algorithm. Specifically, the DJI Mini2 drone's active radio transmits 4K videos. Its frame recovery algorithm improves the image quality without changing its resolution. In contrast, Mighty transmits videos at a lower resolution to save power. These low-resolution frames are recovered to a higher 4K resolution by Mighty's super-resolution algorithm.



(a) Beamforming process (b) Beamforming performance (c) Mobility (d) Comparison with existing works

Fig. 14. (a) Establishment of beamformer. (b) Mighty performance under beamformer. (c) Mighty performance under mobility. (d) Comparison with other backscatter works.

TABLE 2
Average power consumption of system.

	Imaging	Frame selection	Backscatter communication		Total
Component	AR1335module	Raspberry Pi 4	Tag (100% duty cycle)		
			Peak power	Power efficiency	
Power	585mW	640mW	1-Chain: 8.72mW 8-Chain: 69.8mW	1-Chain: 16.5nJ/bit 8-Chain: 16.6nJ/bit	1294mW

Figure 13(a) and Figure 13(b) show the PSNR and SSIM[3] of the video frames recovered by these two algorithms, respectively. Although Mighty's frame recovery task is more challenging, it still achieves decent video quality (with an average PSNR of 25.43 dB) within the range of 150 m, slightly lower than that achieved by the drone's default algorithm. On the other hand, the SSIM of video frames recovered by Mighty is 0.89×–0.92× of the SSIM of video frames recovered by the drone's default video recovery algorithm.

7.2 Micro-benchmarks

Beamformer gain. The performance gain of Mighty brought by the beamformer is satisfactory as shown in Figure 14(a) and Figure 14(b). The beamformer improves the communication range from 113.4 m to 154.5 m, 0.36× higher than before. The PSNR of the recovered video is further improved from 16.8 dB to 25.7 dB, 0.53× higher than before.

Comparison with other backscatter systems. We compare Mighty with two state-of-the-art backscatter systems, namely, Aloba [55] and PLoRa [9]. Aloba and PLoRa modulate backscatter signals on the top of LoRa carrier signals. PLoRa encodes one bit per LoRa symbol, while the throughput of Aloba is determined by the rate of ON-OFF keying operation. Figure 14(d) shows the experiment result. The throughput of Mighty is 2.1–41.6× and 11.3—17.2× of that of Aloba and PLoRa across the communication range of 10-200 m.

Mobility. We conduct experiments to evaluate Mighty's performance under mobility. In our experiment, the distance between the ground controller (Tx&Rx) and the drone is 100 m. The drone carries Mighty and moves around the controller at speeds of 1 m/s, 2 m/s, 3 m/s, 4 m/s, and 5 m/s. The drone captures the bicycles appearing on the playground and transmits the detected interest frame to the controller through Mighty. We measure the accuracy of object detection and the PSNR of recovered video frames at the controller. The result is shown in Figure 14(c). The accuracy of object detection drops from 99.6% to 81.2% and the PSNR of recovered video frames drops from 25.1 to 13.7 as the flight speed grows to 5 m/s.

7.3 Power Consumption Analysis

Overall system power consumption. Table 2 summarizes the average power consumption of our system. The AR1335 imaging module consumes about 585 mW for imaging; the Raspberry Pi 4 consumes around 640 mW for data reading and frame selection. The peak power consumption of our eight-chain backscatter radio is 69.8 mW. Accordingly, the power efficiency is 16.6 nJ/bit. We evaluate the power efficiency of Mighty in §7.1 and compare its power efficiency with other low-power active radios in §7.4.

Backscatter radio power consumption breakdown. We take a closer look at the power consumption of our backscatter radio design. We examine the peak power consumption, which refers to the power consumption of the backscatter radio for continuous video transmission within 1 second and it means the duty cycle of 100%. The specific outcomes are presented in Table 3.

(1) Peak power consumption estimation results. We first estimate the peak power consumption of individual design components using the rated voltage and current settings from the hardware datasheet. Among these hardware components, the most power-hungry parts are the FPGA, DAC, and voltage converter, which account for 64.7%, 21.8%, and 8.2% of the total power consumption, respectively. Using the aforementioned estimation, the peak power consumption for a single-chain Mighty tag falls within the range of approximately 19.3–19.8 mW. When considering an eight-chain radio configuration, this peak power consumption increases to a range of 154.4–158.4 mW.

Considering the practical voltage and current settings, the actual power consumption of Mighty tag is lower than that estimated based on the datasheet. The actual power consumption of FPGA is reduced to around 2.83 mW, 4.4× lower than the theoretical power consumption estimated based on the hardware datasheet, since it only uses 29% of the total I/O resource and 11% of the core. The actual power consumption of the voltage converter is also 0.6× lower than that presented in the hardware datasheet due to the lower supply voltage and current. Accordingly, the total power consumption of the single-chain and eight-chain radio is around 8.7–9.2 mW and 69.6–73.6 mW, respectively.

(2) Peak power consumption measurement results. We measure the power consumption of the backscatter radio by continuously transmitting video streams. The measured results are consistent with the estimated results. The measured peak power consumption of the single-chain and the eight-chain radio is 8.72 mW and 69.8 mW. Due to the dynamics of video transmission and the operation of electronic components, the measured power consumption fluctuates compared to the estimated power consumption. It is worth noting that the measured average power consumption of the ring oscillator is 306 μW, which is 40%–70% lower than the power estimation based on the datasheet. The cause of this discrepancy remains to be investigated, and our hypothese is that the effective driving voltage of the CMOS transistor in the ring oscillator is lower than the bias voltage of the SN74AUP3G04 chip. However, it's worth emphasizing that our measured power

[3] PSNR larger than 25 dB and SSIM larger than 0.8 indicate a good image quality [54].

TABLE 3
Power consumption analysis of Mighty tag.

	Peak power consumption (100% duty cycle)					
	Estimation				***Measurement***	
	Based on the rated voltage and current settings		Based on practical voltage and current settings		**Practical power consumption measurement**	
FPGA	12.75mW	1-Chian	2.83mW	1-Chain	**2.23mW**	**1-Chain**
DAC	4.3mW	19.3–19.8mW	4.3mW	8.7–9.2mW	**5.14mW**	**8.72mW (16.5nJ/bit)**
Voltage converter	1.63mW		987.3uW		**922.4uW**	
Ring OSC	514uW–1mW	8-Chain	514uW–1mW	8-Chain	**306uW**	**8-Chain**
RF switch	120uW	154.4–158.4mW	120uW	69.6–73.6mW	**125uW**	**69.8mW (16.6nJ/bit)**

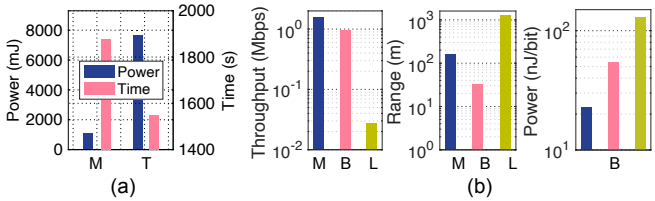


Fig. 15. (a) Mighty (M) vs. conventional HD camera (T). (b) Mighty (M) vs. Bluetooth (B) and LoRa (L).

consumption for the ring oscillator aligns with the value reported in the FS backscatter study [32].

< Discussion on the practical power consumption.> In practice, Mighty only backscatters interest frames (10% of total frames) while sleeping for the remaining frames. Accordingly, the backscatter radio works only 10% of the time for video transmission, *i.e.*, 10% duty cycle. The average power consumption of our single-chain and eight-chain radio thus drops to 870 μW and 6.9 mW, respectively. The throughput drops to 10% in this condition since that the throughput is proportional to the duty cycle. However, the power efficiency is independent of the duty cycle, and Mighty's power efficiency is 16.6 nJ/bit, which is 34–55$\times$ higher than that of the active radio. We can further leverage the ASIC fabrication to improve the power efficiency significantly.

Comparison with conventional HD camera-based video transmission system. We further compare the power consumption of the eight-chain Mighty with the conventional HD camera-based video transmission system on transmitting a 1s video clip. Figure 15(a) shows the result. Mighty reduces the power consumption by 6$\times$ compared to the HD camera (*i.e.*, the power drops from 7335 mJ to 1294 mJ). Such power reduction successfully prolongs the flight endurance of the drone by around 20% (*i.e.*, grows from 1,550s to 1,870s).

7.4 Comparison with Bluetooth and LoRa

We further compare Mighty with Bluetooth and LoRa radios as shown in Figure 15(b). The throughput of Mighty is 1.6$\times$ (1.55 Mbps vs.0.95Mbps) and 57.4$\times$ (1.55 Mbps vs.27 Kbps) higher than Bluetooth and LoRa. On the other hand, the communication range of Mighty is 5.1$\times$ (162 m vs.32 m) longer than Bluetooth due to the anti-noise capability of chirp modulation. Whereas, the communication range of Mighty is 7.7$\times$ (162 m vs.1250 m) shorter than LoRa because Mighty sacrifices the symbol time to improve the throughput. Finally, the power efficiency of Mighty is 7.5$\times$ higher than Bluetooth (16.6 nJ/bit vs. 124.5 nJ/bit) and 16.2$\times$ higher than LoRa (16.6 nJ/bit vs. 268.5 nJ/bit).

8 Discussion

Extend to other applications. In addition to the video transmission on the small drone, more generally, Mighty can be used in many application scenarios, for example, keeping track of the status of machines in the factory (e.g., vibration, noise, rotation), monitoring the home security with a static camera sensor, and uploading the information of all bulk goods. On the one hand, these applications demand high-throughput (about Mbps) communication links for data forwarding. On the other hand, these data forwarding links should be also low-power and long-range, allowing sensors to transmit their data back to the gateway hundreds of meters away without extra human intervention or frequent battery. Therefore, Mighty with the throughput of 1.58 Mbps and transmission range of 162 m is applicable to the above scenarios.

Scalability to large-scale deployment. The application of Mighty can be extended to different types of drones. In addition to DJI Mini2, Mighty can also be applied for syma X35 and Parrot Mambo FPV Mini. The performance (transmission rate and communication range) of Mighty is related to the parameters of the chirp symbols. In order to scale Mighty's performance in large-scale applications, we need to improve the transmission rate or extend communication range. On the one hand, we can increase the bandwidth to increase the granularity factor (GF) and curving factor (CF) of the chirp symbols, which can improve the transmission rate and further support the higher resolution video transmission. On the other hand, we can extend the length of the chirp symbols to improve the anti-noise ability of the chirp signals and thus achieve longer communication distance.

9 Conclusion

We have presented the design, implementation, and evaluation of Mighty, a low-power, long-range, and high-throughput backscatter system for drones. Mighty develops a power-efficient backscatter radio and a lightweight video compression-bypassing design. The head-to-head comparison shows that Mighty can achieve similar throughput at a distance of up to 160 meters while consuming 34–55$\times$ less power.

Acknowledgment

This work is supported by the National Science Fund of China under grant No. 62088101, No. 62202264, No. 62394341, No. 62394344, No. 62425207, No. U21B2007, the Fundamental Research Funds for the Central Universities 226-2024-00004, 226-2023-00111, and Key Research and Development Program of Zhejiang Program No. 2024C01065.

References

[1] "Drone of DJI Mini2," https://www.dji.com/mini-2.
[2] X. Guo, Y. He, Z. Yu, J. Zhang, Y. Liu, and L. Shangguan, "RF-Transformer: A unified backscatter radio hardware abstraction," in *Proceedings of ACM MobiCom, Sydney, Australia, October 17-21*, 2022.
[3] R. Zhao, F. Zhu, Y. Feng, S. Peng, X. Tian, H. Yu, and X. Wang, "OFDMA-enabled WiFi backscatter," in *Proceedings of ACM MobiCom, Los Cabos, Mexico, October 21-25*, 2019.

[4] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith, "Towards battery-free HD video streaming," in *Proceedings of USENIX NSDI, Renton, WA, USA, April 9-11*, 2018.

[5] M. H. Mazaheri, A. Chen, and O. Abari, "mmtag: a millimeter wave backscatter network," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 463–474.

[6] J. Zhang, R. Xi, Y. He, Y. Sun, X. Guo, W. Wang, X. Na, Y. Liu, Z. Shi, and T. Gu, "A survey of mmWave-based human sensing: technology, platforms and applications," *IEEE Communications Surveys & Tutorials,*, vol. 25, no. 4, pp. 2052–2087, 2023.

[7] A. Varshney, O. Harms, C. P. Penichet, C. Rohner, and T. V. Frederik Hermans, "LoRea: A backscatter architecture that achieves a long communication range," in *Proceedings of ACM SenSys, Delft, Netherlands, November 06-08*, 2017.

[8] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, "Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems," in *Proceedings of USENIX NSDI*, 2022.

[9] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "PLoRa: A passive long-range data network from ambient LoRa transmissions," in *Proceedings of ACM SIGCOMM, Budapest, Hungary, August 20-25*, 2018.

[10] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," in *Proceedings of ACM UbiComp, Maui, HI, USA, September 11-15*, 2017.

[11] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Efficient ambient LoRa backscatter with ON-OFF keying modulation," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 641–654, 2022.

[12] "Time-bandwidth product," https://www.radartutorial.eu/09.receivers/rx53.en.html.

[13] T. Virolainen, J. Eskelinen, and E. Haggstrom, "Frequency domain low time-bandwidth product chirp synthesis for pulse compression side lobe reduction," in *Proceedings of IEEE International Ultrasonics Symposium, Rome, Italy, September 20-23*, 2009.

[14] "Free space loss model," http://www.sis.pitt.edu/prashk/inf1072/Fall16/lec5.pdf.

[15] K. Zhang, W. Zuo, and L. Zhang, "Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Early access, pp. 1–16, 2021.

[16] S. Naderiparizi, P. Zhang, M. Philipose, B. Priyantha, J. Liu, and D. Ganesan, "Glimpse: A programmable early-discard camera architecture for continuous mobile vision," in *Proceedings of ACM MobiSys, Niagara Falls, New York, USA, June 19-23*, 2017.

[17] S. Hanson, Z. Foo, D. Blaauw, and D. Sylvester, "A 0.5v sub-microwatt CMOS image sensor with pulse-width modulation read-out," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 45, no. 4, pp. 759–767, 2010.

[18] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision," in *Proceedings of ACM MobiSys, Taipei, Taiwan, June 25-28*, 2013.

[19] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *Proceedings of ACM SenSys, Virtual Event, Japan, November 16-19*, 2020.

[20] W. Tian, C. Gu, M. Guo, S. He, J. Kang, D. Niyato, and J. Chen, "Large-scale deterministic networks: Architecture, enabling technologies, case study, and future directions," *IEEE Network,*, vol. 38, no. 4, pp. 284–291, 2024.

[21] Y. Liu, X. Shi, S. He, and Z. Shi, "Prospective positioning architecture and technologies in 5G networks," *IEEE Network,*, vol. 31, no. 6, pp. 115–121, 2017.

[22] H. Jiang, J. Zhang, X. Guo, and Y. He, "Sense me on the ride: Accurate mobile sensing over a LoRa backscatter channel," in *Proceedings of ACM SenSys, Coimbra, Portugal, November 15-17*, 2021.

[23] D. Huixin, X. Yirong, Z. Xianan, W. Wei, Z. Xinyu, and H. Jianhua, "Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter," in *Proceedings of ACM MobiCom, Madrid, Spain, October 2-6*, 2023, pp. 1–15.

[24] X. Na, , X. Guo, Z. Yu, J. Zhang, Y. He, and Y. Liu, "Leggiero: Analog WiFi backscatter with payload transparency," in *Proceedings of ACM MobiSys, Helsinki, Finland, June 18-22*, 2023.

[25] Z. Chi, X. Liu, W. Wang, Y. Yao, and T. Zhu, "Leveraging ambient LTE traffic for ubiquitous passive communication," in *Proceedings of ACM SIGCOMM, Virtual Event, USA, August 10-14*, 2020.

[26] S. Guochao, W. Wei, Y. Hang, Z. Dongchen, G. Peng, and J. Tao, "Exploiting channel polarization for reliable wide-area backscatter networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 12, pp. 4338–4351, 2021.

[27] X. Guo, Y. He, N. Jing, J. Zhang, Y. Liu, and L. Shangguan, "A low-power demodulator for LoRa backscatter systems with frequency-amplitude transformation," *IEEE/ACM Transactions on Networking*, vol. 32, no. 4, pp. 3515–3527, 2024.

[28] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "WISPCam: A battery-free RFID camera," in *Proceedings of IEEE RFID, San Diego, CA, USA, April 15-17*, 2015.

[29] P. Hu, P. Zhang, and D. Ganesan, "Laissez-faire: Fully asymmetric backscatter communication," *ACM SIGCOMM computer communication review*, vol. 45, no. 4, pp. 255–267, 2015.

[30] M. Rostami, J. Gummeson, A. Kiaghadi, and D. Ganesan, "Polymorphic radios: A new design paradigm for ultra-low power communication," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 446–460.

[31] C. Li, X. Guo, L. Shangguan, Z. Cao, and K. Jamieson, "CurvingLoRa to boost LoRa network capacity via concurrent transmission," in *Proceedings of USENIX NSDI, RENTON, WA, USA, April 4-6*, 2022.

[32] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proceedings of ACM SIGCOMM, Salvador, Brazil, August 22-26*, 2016.

[33] "Inverter SN74AUP3G04," https://pdf1.alldatasheetcn.com/datasheet-pdf/view/317334/TI/SN74AUP3G04.html.

[34] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE transactions on communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[35] "LoRa Alliance," https://www.lora-alliance.org/.

[36] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[37] S. Tong, Z. Xu, and J. Wang, "CoLoRa: Enabling Multi-Packet Reception in LoRa," in *Proceedings of IEEE INFOCOM, Toronto, ON, Canada, July 6-9, 2020*, 2020.

[38] C. Li, H. Guo, S. Tong, X. Zeng, Z. Cao, M. Zhang, Q. Yan, L. Xiao, J. Wang, and Y. Liu, "NELoRa: Towards Ultra-low SNR LoRa Communication with Neural-enhanced Demodulation," in *Proceedings of ACM SenSys, Coimbra, Portugal, November 15-17, 2021*, 2021.

[39] H. Hu, Y. Lin, M. Liu, H. Cheng, Y. Chang, and M. Sun, "Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos," in *Proceedings of IEEE CVPR, Honolulu, HI, USA, July 21-26*, 2017.

[40] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Transactions on Image Processing*, vol. 29, no. 1, pp. 1271–1284, 2020.

[41] Y. He, "Research on micro-expression spotting method based on optical flow Features," in *Proceedings of ACM MM, Virtual Event, October 20-24*, 2021.

[42] J. Villegas and A. G. Forbes, "Analysis/synthesis approaches for creatively processing video signals," in *Proceedings of ACM MM, Orlando, Florida, USA, November 3-7*, 2014.

[43] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, "Jump: Virtual reality video," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–13, 2016.

[44] X. Fan, L. Shangguan, R. Howard, Y. Zhang, Y. Peng, J. Xiong, Y. Ma, and X.-Y. Li, "Towards flexible wireless charging for medical implants using distributed antenna system," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–15.

[45] H. Lee, T. Kim, T. young Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proceedings of IEEE CVPR, Seattle, WA, USA, June 13-19*, 2020.

[46] "CMOS image sensor AR1335," https://www.e-consystems.com/ar1335-camera-module.asp.

[47] "Low-power FPGA GW1N-UV9QN48," https://www.gowinsemi.com/en.

[48] "12-bit Low-power DAC TLV5619," https://www.ti.com/store/ti/zh/p/product/?p=TLV5619CPWR.

[49] "FCC rules for unlicensed wireless equipment operating in the ISM bands," FCCbasicsofunlicensedtransmitters.

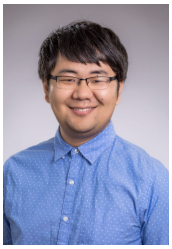
[50] Z. Zhan, Y. Gong, P. Zhao, G. Yuan, W. Niu, Y. Wu, T. Zhang, M. Jayaweera, D. Kaeli, B. Ren *et al.*, "Achieving on-mobile real-time super-resolution with neural architecture and pruning search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4821–4831.

[51] T. Zhang, S. Ye, K. Zhang, J. Tang, W. Wen, M. Fardad, and Y. Wang, "A systematic dnn weight pruning framework using alternating direction method of multipliers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 184–199.

[52] "The source code of super-resolution algorithm DRUNet," https://github.com/cszn/DPIR.
[53] "The source code of frame interpolation algorithm AdaCoF," https://github.com/HyeongminLEE/AdaCoF-pytorch.
[54] "Digital picture formats and representations," https://www.sciencedirect.com/topics/engineering/peak-signal.
[55] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Aloba: Rethinking on-off keying modulation for ambient LoRa backscatter," in *Proceedings of ACM SenSys, Virtual event, November 16-19*, 2020.

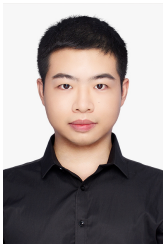
Xiuzhen Guo is an Assistant Professor with the College of Control Science and Engineering, Zhejiang University. She received her B.E. degree from Southwest University, and her PhD degree from Tsinghua University. Her research interests include wireless networks, Internet of Things, and mobile computing. She is a member of IEEE and a member of ACM.

Yuan He is an associate professor in the School of Software and BNRist of Tsinghua University. He received his B.E. degree in the University of Science and Technology of China, his M.E. degree in the Institute of Software, Chinese Academy of Sciences, and his PhD degree in Hong Kong University of Science and Technology. His research interests include wireless networks, Internet of Things, pervasive and mobile computing. He is a senior member of IEEE and a member of ACM.



Longfei Shangguan is an assistant professor in the Department of Computer Science, University of Pittsburgh. He received his B.E. degree in Xidian University, and his PhD degree in Hong Kong University of Science and Technology. His research interests include networking, IoT, and wireless systems.

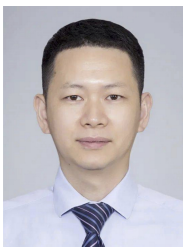
Yande Chen is currently a PhD student in Tsinghua University. He received his B.E. degree in Tsinghua University. His research interests include backscatter communication and wireless sensing.



Chaojie Gu (Member, IEEE) received the B.Eng. degree in information security from the Harbin Institute of Technology, Weihai, China, in 2016, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2020. He was a Research Fellow with Singtel Cognitive and Artificial Intelligence Lab for Enterprise, in 2021. He is currently an Assistant Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include IoT, industrial IoT, edge computing, and low-power wide area networks.

Yuanchao Shu (Senior Member, IEEE) is a Qiushi Professor with the College of Control Science and Engineering at Zhejiang University, China. Prior to joining academia, he was a Principal Researcher with the Office of the CTO, Microsoft Azure for Operators, and the Mobility and Networking Research Group at Microsoft Research Redmond. He joined Microsoft Research after receiving Ph.D. from Zhejiang University in 2015. He was also a joint Ph.D. in the EECS Department at the University of Michigan, Ann Arbor. He currently served on the editorial board of IEEE Transactions on Wireless Communications and ACM Transactions on Sensor Networks, and was a member of the organizing committee and TPC of conferences, including MobiCom, MobiSys, SenSys, SEC, IPSN, Globecom, ICC, etc. He received ACM China Doctoral Dissertation Award (2/yr), IBM PhD Fellowship, and five best paper/demo awards from leading CS/EE conferences.

Kyle Jamieson is a professor in the Department of Computer Science at Princeton University, where he has been on the faculty since 2015, after seven years on the faculty at University College London, U.K. Jamieson leads the Princeton Advanced Wireless Systems (PAWS) Group, whose charter is to build and experimentally evaluate wireless networking and wireless sensing systems for the real world that cut across the boundaries of digital communications, signal processing, and computer networking. He received the B.S. (2001), M.Eng. (2002), and Ph.D. (2008) degrees in Computer Science from the Massachusetts Institute of Technology. He then received a Starting Investigator fellowship from the European Research Council in 2011, Best Paper awards at USENIX 2013 and CoNEXT 2014, a Google Faculty Research Award in 2015, and the ACM SIGMOBILE Early Career Award in 2018.



Jiming Chen (Fellow, IEEE) received B.Sc and PhD degree both in Control Science and Engineering from Zhejiang University, Hangzhou, China in 2000 and 2005, respectively. He is currently a professor with the Department of Control Science and Engineering at Zhejiang University and president of Hangzhou Dianzi University. His research interests include IoT, networked control, wireless networks. He serves on the editorial boards of multiple IEEE Transactions, and the general co-chairs for IEEE RTCSA'19, IEEE Datacom'19 and IEEE PST'20. He was a recipient of the 7th IEEE ComSoc Asia/Pacific Outstanding Paper Award, the JSPS Invitation Fellowship, and the IEEE ComSoc AP Outstanding Young Researcher Award. He is an IEEE VTS distinguished lecturer. He is a fellow of the CAA.