Low-Power Wireless Video Streaming and Applications

Ali Saffari

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Joshua R. Smith, Chair

Payman Arabshahi

Arka Majumdar

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Low-Power Wireless Video Streaming and Applications

Ali Saffari

Chair of the Supervisory Committee:
Prof. Joshua R. Smith
Electrical and Computer Engineering

Wireless video streaming has recently become an exciting topic of research in IoT, computer vision, and security systems. However, for some of these applications, power consumption of wireless video camera systems is a bottleneck. The issue can be addressed in two separate sections: 1- optimizing the power for wireless data transfer; since the data size in video-related applications is typically larger compared to other sensors, and 2- sensing power optimization, wherein we aim to improve video resolution while keeping the power consumption of the system low.

In the first part, we focus on communication and aim to design a reliable and power-efficient wireless network of low-resolution wireless image sensors using backscatter communication technology. Also, we show that our network of low-resolution and solar-powered cameras can be used in security applications and human occupancy detection systems in buildings.

Once we show that our wireless network is power-efficient, we will improve the video resolution. To enable low-power high-resolution video capturing, we use a dual-mode camera system, which switches between low-resolution gray-scale and high-resolution color video recording modes. We stream low-resolution image sensor data (gray-scale) with occasional high-resolution reference frames to the basestation. Our machine learning model implemented at the basestation generates high-resolution color frames from the low-resolution

video data based on the reference frames.

ACKNOWLEDGMENTS

This dissertation is only made possible through the support of many people to whom I owe my deepest gratitude.

First, I would like to thank my advisor, Professor Joshua Smith, for all the support and mentorship and for allowing me to work on a broad scope of problems. I feel very fortunate to work in his lab. I am very grateful for the opportunity that Josh gave me, and I will always remember his kindness and support.

I also want to thank Dr. Saman Naderiparizi, my first mentor, who helped me get started in the Sensor Systems Lab. Also, I want to thank Professor Zerina Kapetanovic; I worked on my first paper with her and learned a lot.

I am very grateful to have had the opportunity to work with Mohamad Katanbaf on multiple projects. During our collaboration, I learned a lot of skills from him that helped me with my Ph.D. work and finding a job. He changed my professional life! I also enjoyed our conversations during our tea breaks.

I would like to thank my other labmates in the Sensor Systems Lab: Rohan Menon, Jared Nakahara, Brody Mahoney, Dr. Gregory Moore, Dr. Patrick Lancaster, Boling Yang, Shanti Garman, Kedi Yan, Paolo Torrado, and Timothy Amish. It was a great pleasure knowing them.

Special thanks go to the director of the Mobile Intelligence Lab, Professor Shyam Gollakota, and past/current lab members, Dr. Ali Najafi, Dr. Mehrdad Hessar, Bandhav Veluri, and Professor Vikram Iyer. I am grateful for knowing them and working with them. Also, I would like to thank Dr. Vaishnavi Ranganathan for her mentorship during my internship at Microsoft Research. She also helped me with the job search process.

I want to thank Professor Payman Arabshahi, Professor Nadya Peek, and Professor Arka Majumdar for their time, interest in my work, and valuable comments. It was an honor to have them on my committee.

Finally, I would like to thank my parents and my brother, Mohamad, for their support throughout the past ten years while I was away from home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Power-efficient video recording and streaming are major requirements in developing battery-free IoT sensors, from wearable cameras to smart home monitoring systems and smart agriculture applications. To enable this, we must find a low-power and reliable commutation technology for an energy-efficient wireless data transfer. As seen in Fig. 1.1, active radios typically are too power-hungry to be supported by harvested ambient light energy (up to a few mW). This becomes a bigger problem in video systems with huge data sizes. To address this challenge, we propose to use backscatter communication for data transfer. In a backscatter deployment, a transmitter generates the carrier, a sensor node synthesizes packets by reflecting the carrier, and a receiver demodulates the backscattered packets. Thus, the sensor does not consume any energy to generate the RF signal for data transmission.

In the second chapter, we use a (112×112) analog image sensor to record gray-scale video. In order to transfer the video data to the basestation, Pulse Width Modulation (PWM) backscatter technology is used. Next, we show that our system can operate as a security camera. Finally, this chapter shows analog sensors and PWM backscatter face multiple issues, such as low-quality images, compared to digital sensors with similar resolution and power consumption. They also suffer from low dynamic range. These issues will eventually force us to switch to digital image sensors and Frequency Shift Keying (FSK) backscatter.

Next, we design and implement a digital camera system using FSK backscatter communication technology and perform several experiments to verify the range of the communication system under different operating conditions. Adding a low-power downlink radio to the new camera design enables a two-way link between the sensors and basestation. The basestation can address each sensor in the network via a unique ID and receive data from them using

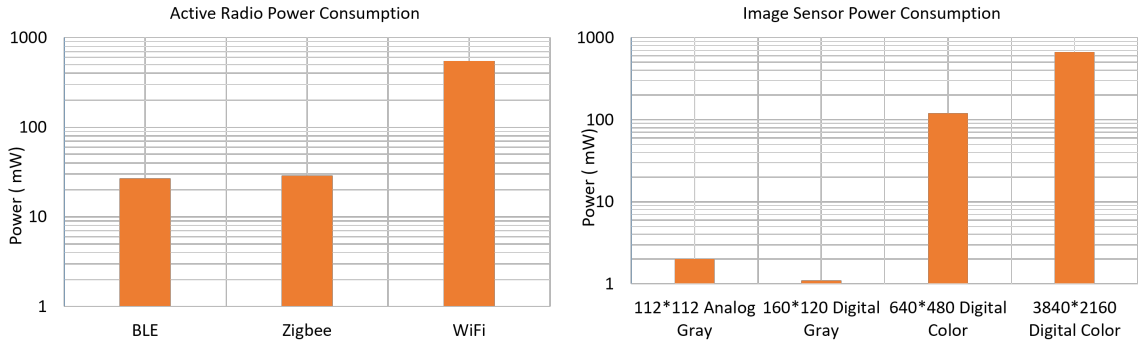


Figure 1.1: **Power consumption comparison for active radios (left), and image sensors (right).**

time division multiplexing. In contrast, the analog design does not have a down-link communication connection; therefore, only one sensor can be used in the network. The fourth chapter uses a digital system, explores the human occupancy detection application in more detail, and verifies that our design can be deployed for multiple weeks in a building to detect human presence.

Finally, in Chapter 5, we address the second challenge, improving the image resolution while keeping power consumption low. As mentioned, throughout the first four chapters, we focus on power optimization and reliability of the communication system and explore applications for gray-scale low-resolution video cameras. However, color and higher image resolutions are sometimes essential to satisfy a minimum set of requirements. Using a high-resolution image sensor with the FSK backscatter design (developed in the previous sections) faces two challenges: 1- as seen in Fig. 1.1, power consumption increases as the image size improves, and 2- higher image resolution means more pixels, which results in larger data sizes and ultimately higher communication energy consumption. This chapter shows that a dual-camera image sensor can provide high-quality VGA (480p) images while consuming 2.1 mA of current.

Chapter 2

BATTERY-FREE WIRELESS ANALOG VIDEO STREAMING CAMERA SYSTEM

2.1 Introduction

Battery-free wireless sensors have been actively researched for many years. Advancements in this area have led to the development of battery-free low throughput/low-power sensors for temperature, light and pressure [1,2] as well as high throughput/more power hungry sensors such as cameras. Over the past few years, researchers have shown that we can harvest sufficient power from ambient energy sources such as Wi-Fi and RFID readers to power off-the-shelf low resolution cameras. While harvested power can capture still images, it is insufficient for video streaming [3].

The critical obstacle preventing systems such as the WISPCam [3] from streaming live video is the camera's high power consumption. Conventional camera architectures consist of an array of photo-diodes that sense the image, a high bandwidth and low-noise amplifier (LNA) that amplifies the signal generated by the photo-diodes, and a high rate analog-to-digital converter (ADC) that digitizes the amplifier output. For many digital video streaming applications, a video compression module is also necessary to reduce required communication bandwidth. Note that most camera power consumption results from the use of power hungry components including the LNA, ADC and compression module [4, 5].

More recently, researchers have redesigned the conventional camera architecture to remove power hungry components from the camera and delegate them to a wireless access point (AP) in an asymmetric wireless communication setting such as backscatter [6]. In [6], the raw analog voltage generated by the pixels is converted into a pulse width modulated (PWM) signal and then fed into the backscatter module, avoiding the power hungry compo-

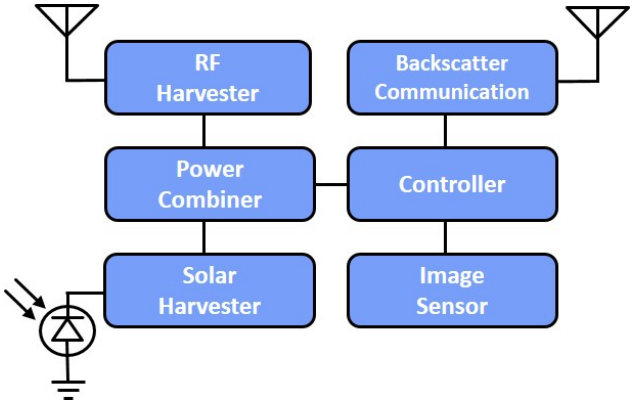


Figure 2.1: **Battery-Free Wireless Video Streaming Architecture.** We design a battery-free video streaming camera that harvests both RF and solar power.

nents of a camera such as the ADC and amplifier. Although this design is very low power, the prototype built in [6] does not operate based on harvested energy. Therefore, building an end-to-end system that can harvest its energy from ambient sources and stream video to a wireless AP remained an unsolved challenge.

Further, although the redesigned camera architecture reduces the power a conventional camera consumes by a few orders of magnitude, existing off-the-shelf cameras [7] that best match the proposed architecture still burn more power than is harvestable from an FCC-compliant RF source at useful ranges. In reality, we can harvest micro-watt level power from an FCC-compliant RF source at medium to far distances (a few feet to a few tens of feet); however, aforementioned cameras burn a few mW. Bridging the gap between available harvested power and required power would enable the design of security and monitoring cameras that do not require wires, thereby significantly reducing infrastructure installation and maintenance costs. A battery-free video streaming camera can also provide a monitoring system for hard-to-reach areas and energy-constrained applications, such as Kilobots and insect-scale robots [8, 9].

This paper presents the first battery-free camera that streams live video to a wireless AP and harvests all of its energy by aggregating ambient light and RF. Fig. 2.1 shows our system

Figure 2.2: **Prototype for Our Battery-Free Video Streaming Camera with Dual Power Harvester.** The FPGA is located behind the solar cell.

architecture. We evaluated the video streaming camera in indoor and outdoor environments under different lighting conditions. In outdoor scenarios on a sunny day, our camera can stream 13 fps live video to the AP using a 2.2 cm×0.7 cm solar cell. In indoor scenarios under normal office lighting conditions (about 500 lux), our camera can stream at > 5 fps live video at a distance of up to 10 ft, leveraging both RF and ambient light energy harvesting, and 3 fps at a distance of up to 150 ft, mainly relying on light energy harvesting.

To show the feasibility of our video streaming camera for real applications (e.g., a surveillance or monitoring camera), we build a battery-free camera that streams video to a nearby AP. We set up our camera outside a building on a sunny day and record video frames backscattered by the camera for eight hours. We show that our camera can backscatter video frames at 1 to 9 fps during a day when light intensity remains between 300 and 3000 lux. We implement a prototype of our battery-free video streaming camera using a 112×112 image sensor controlled by an IGLOO nano FPGA, as shown in Fig. 2.2.

Our Contributions. Here, we list our main contributions:

- We develop the world's first battery-free and wire-free live video streaming system.
- We design and evaluate a dual energy harvester that aggregates energy from both ambient light and RF signals.
- We demonstrate a dual antenna architecture one used for energy harvesting and the

other for communication that increases wireless communication and RF power harvesting range compared to a single antenna counterpart.

2.2 Related Work

Related work falls into two categories, backscatter communication and power harvesting, which we now describe.

Backscatter communication. Previous research in backscatter [10–13] shows high data rate backscatter communication using Wi-Fi, Bluetooth or TV broadcast signals. Some research [14] designs a high data rate QAM backscatter modulator that works in the UHF band. Other work [15, 16] focuses on optimizing data flow operation from a sensor to a backscatter communication module to reduce power consumption. These works use an ADC to convert a sensor's analog output to digital information that is transmitted using digital backscatter. As noted in the last section, for high throughput sensors such as cameras, ADCs are the primary power-consuming component of the system.

More recent research [17, 18] is demonstrating that backscatter is a feasible vehicle for wide-area, low-power communication. This work focuses on low data rate communication optimized for Internet-of-Things applications, which can scale to hundreds of devices in wide-area scenarios. However, it provides a throughput that is far too low for an application like video streaming.

In this work, we leverage a technique called analog backscatter [19, 20]. This technique transmits raw analog signals generated by a sensor, here an image sensor, and hence does not require key power consumers like an ADC. Furthermore, analog backscatter is viable for high-throughput applications such as video streaming. We build on previous works [6, 21] to design a battery-free video streaming camera that removes the power-hungry components from the sensor node and delegates them to a plugged-in access point.

Power harvesting. In the RF power harvesting domain, some research [22, 23] harvests energy from ambient TV signals. Other work [24] presents a power harvesting system that uses RF signals from Wi-Fi transmissions or prototypes a wireless sensing platform that

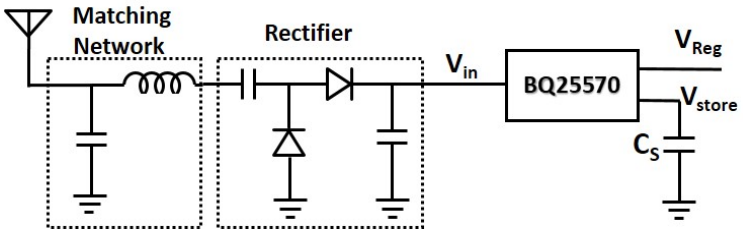


Figure 2.3: **RF Harvesting Circuit.**

harvests energy from TV broadcast signals and cellular base transceiver stations [25]. In [20], researchers design a battery-free cellphone that harvests energy from either RF or ambient light, but it does not combine both. Previous work on low-power camera design [3, 26, 27] builds battery-free wireless cameras that capture still images and backscatter the pixels to a nearby RFID reader. These works, based on the WISP [28] platform, send a frame every ten seconds when the camera is about one foot from the RF source and every tens of minutes at longer distances.

Finally, researchers in [29] demonstrate a large form-factor low-resolution camera that harvests energy from incident light and captures one frame per second. The image is sent via cable to a computer. In contrast, our work shows a fully wireless, battery-free video streaming camera that harvests energy by combining power from both an RF signal and ambient light.

2.3 System Design and Implementation

Our battery-free camera contains five key components, as shown in Fig. 2.1: 1) an RF and a solar power harvester, 2) a power combiner, 3) a controller, 4) an image sensor, and 5) a backscatter communication system. The camera harvests energy from both the RF signal and ambient light and combines the energy. The AP generates a single-tone RF signal, which the camera uses for RF energy harvesting and backscattering the video frames. This section explains each design component.

RF power harvester. As Fig. 2.3 shows, the RF harvester includes three key components:

1) a matching network, 2) a rectifier, and 3) a power harvester. The matching network matches the antenna impedance to the conjugate of the rectifier impedance. We use a single-stage high-frequency LC network with the values of $L = 33\ nH$ and $C = 5.1\ pF$. After this matching, the RF signal goes to a full-wave rectifier, which converts the RF signal to DC voltage [24]. We use 33 pF capacitors and HSMS-285C diodes to implement the rectifier.

Finally, the rectifier's output connects to a TI BQ25570 energy harvester. To provide energy for the camera, this buck/boost charger IC operates in a forced duty-cycling mode that corresponds to the input power. As shown in Fig. 2.4, the harvester accumulates energy on a storage capacitor. When the voltage of the capacitor, V_{store} in Fig. 2.3, reaches a programmable threshold (V_{max}), the buck converter activates and supplies power from the storage capacitor to the camera until the capacitor discharges to another programmable threshold (V_{min}). Thus, the amount of stored energy during one cycle can be calculated from equation (2.1)

$$E = \frac{1}{2} C_S \Big(V_{max}^2 - V_{min}^2 \Big) \tag{2.1}$$

Considering the minimum voltage for camera operation and the voltage drop on the buck converter, we set $V_{min} \geq 3.4V$. Capacitor leakage current increases as its voltage increases; therefore, to preserve harvested power, we set $V_{min} = 3.4V$. We now have two knobs that can change the amount of energy supplied to our battery-free video streaming camera in each duty cycle, V_{max} and C_S.

The harvester begins from a cold start until V_{store} reaches 1.8 V, after which, the maximum power point tracking (MPPT) inside the IC begins to operate. The MPPT enhances RF power harvesting efficiency by optimizing the IC input impedance. During the cold start, the minimum rectified voltage from which the IC can still harvest energy is 600 mV. When V_{store} crosses 1.8 V, the MPPT activates and reduces the minimum required rectified voltage to 100 mV.

Dual RF-Solar power harvester. As shown in Fig. 2.8, relying only on RF energy

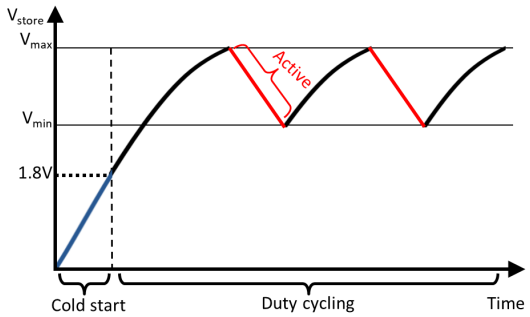


Figure 2.4: **Power Harvesting Duty Cycle.** The power harvester IC stores energy on the capacitor. Once the voltage reaches V_{max}, the harvester powers the camera until the capacitor discharges to V_{min}.

harvesting from an FCC-compliant RF source forces the battery-free video streaming camera to be located within 17 ft of the AP. In addition, the camera cannot produce frames at a rate that exceeds 1 fps for distances beyond 9 ft, which limits the camera's utility and potential applications. As a result, RF power energy harvesting alone cannot unleash the potential that battery-free video streaming can provide.

To solve this problem, we combine RF with solar energy harvesting. For the latter, we use the same boost charger IC but without a rectifier since solar cell output is already a DC voltage. In an outdoor environment, our dual power harvester mainly relying on an unexposed-to-direct-sunlight solar cell with dimensions of 3.5 cm×4.2 cm can provide enough power for continuous video streaming at 13 fps. In an indoor environment, a dual power harvester provides sufficient power to achieve a frame rate that exceeds 5 fps at a distance of up to 10 ft from the access point using both the RF power harvester and a solar cell (9 cm×7.9 cm). This harvester also maintains a steady frame rate (3 fps) at farther distances, leveraging solar power harvesting in an indoor environment.

The key challenge to dual power harvesting is coordinating operation of two boost charger ICs in order to store energy from two different sources. As shown in Fig. 2.5, both ICs connect to the camera using an ADG774 analog multiplexer. The harvesters work independently. Whenever any harvester stores enough energy, multiplexer selects that harvester to power the

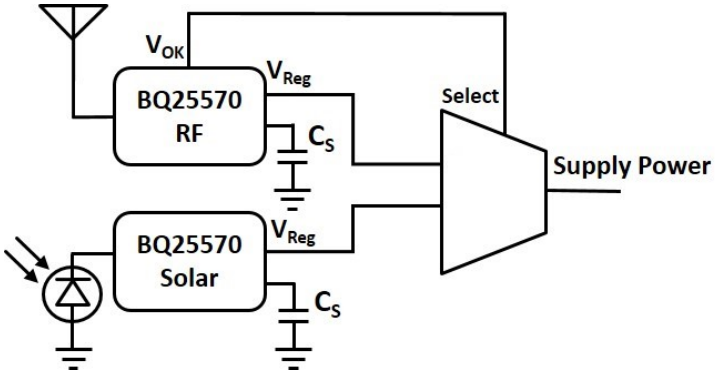


Figure 2.5: **Dual Power Harvesting Architecture.** A dual power harvesting circuit that stores RF and solar power for our battery-free video streaming camera.

camera, giving the higher priority to the RF harvester. Thus, if both harvesters concurrently have enough energy, the camera first gets powered by RF harvested energy; and once this energy is consumed, the power source switches to harvested solar energy.

The BQ25570 IC activates a signal, named V_{OK}, when the harvester accumulates enough energy on the storage capacitor (C_S). We use V_{OK} here for the address input of the multiplexer. We use a fast multiplexer alongside a small capacitor at the input of the camera power supply to ensure negligible drops in camera supply voltage when switching between two harvesters. This guarantees that the camera does not brown out when switching occurs.

Image sensor. We use a 112×112 resolution gray-scale random pixel access image sensor from CentEye, which provides raw analog readout access to all the pixels. The image sensor has two internal row and column registers, which let the user output the raw analog voltage of a specific pixel. The analog voltage of this image sensor is a function of light intensity. We use this image sensor functionality to design our analog backscatter communication.

Controller. We use IGLOO nano, AGLN250V2 FPGA to implement our controller state machine. To capture video frames, the FPGA sweeps through the pixels by setting the row and column registers' value in a raster-scan (row-by-row and left to right) mode. To generate the master clock for our battery-free camera state machine, we use an SiT802 oscillator. This oscillator consumes only 180 μW of power in active mode, however, there is a 70 ms latency

between oscillator power-up and when it outputs a clock signal. During this start-up period, FPGA operation is paused, and the image sensor remains in shut-down mode to save power. To design this part of our control system, we analyze two approaches and pick the one that lessens energy loss during oscillator start-up. We show the architectures for both approaches in Fig. 2.6.

First approach. Our goal here is to decrease the oscillator's start-up delay by never turning it off, instead forcing the oscillator to enter a standby mode when no operation is required. The SiT802 oscillator's standby mode consumes 4 μW of power and becomes active 3 ms after exiting from standby. In this approach, we supply power continuously to the oscillator but keep it in standby mode until the storage capacitor has reached V_{max} voltage. When the harvester accumulates enough energy on the storage capacitor to power up the camera, it outputs an enable signal (V_{OK}), which we use to enable the oscillator. As noted, switching between active and standby mode takes 3 ms, and, once the oscillator starts working, the FPGA initializes the image sensor. To do this, we must power: a voltage regulator to supply a regulated voltage to the oscillator, an inverter gate to invert the polarity of (V_{OK}) used to control the oscillator's operation mode, and the oscillator in standby mode. The regulator, inverter and oscillator in standby mode consume 2.5 μW, 3.5 μW, and 4 μW, respectively. The total energy consumed by these components when our system does not transmit any video frames is shown in equation (2.2)

$$E_1 = (P_{regulator} + P_{inverter} + P_{osc}) \times t_{stb} = 10 \times t_{stb}\ \mu J \tag{2.2}$$

where t_{stb} is the standby time.

Second approach. Here, to conserve energy during oscillator power up, we power the image sensor after the oscillator's 70 ms delay. To do this, we use a switch that is controlled by the FPGA and gates the power to the image sensor. We use an ADG774 as the switch which consumes 3.3 μW. During the 70 ms delay, the FPGA is paused because the clock signal is not available; during this period, the FPGA consumes 240 μW. The oscillator's power is 180 μW in active mode. Equation (2.3) shows the energy consumption of this approach,

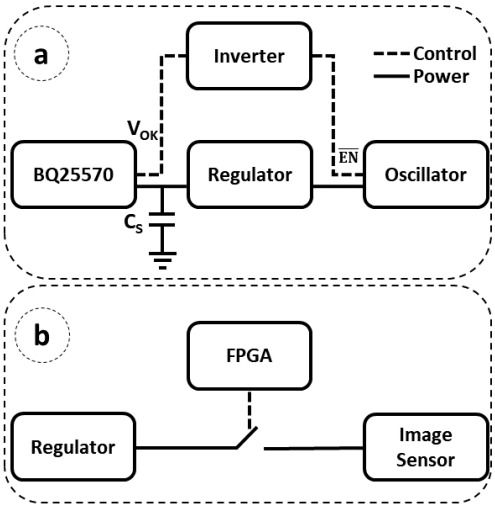


Figure 2.6: **Proposed Architectures.** Alternative approaches for saving energy during oscillator start up.

where P_{FPGA} is the FPGA's power consumption when it is powered up with no clock source, and t_{delay} is the oscillator's delay.

$$E_2 = (P_{switch} + P_{FPGA} + P_{oscillator}) \times t_{delay} \approx 29.6\ \mu J \tag{2.3}$$

Equating E_1 and E_2 provides the boundary condition that suggests which approach is more efficient in terms of energy loss. As a result, $t_{stb} = 2.96s$ (equivalently, frame rates > 0.34 fps) is the threshold below which *first approach* has a lower energy loss.

Backscatter communication. To transmit video frames to the access point, we use backscatter communication. A naive solution would connect image sensor output directly to an analog RF switch and use analog backscatter [19] to send video frames. However, the problem here is that the low dynamic range of the pixel voltages maps to a very small subset of radar cross-sections at the antenna. Assuming that both channel and receiver add noise to the signal, we would receive a low Signal to Noise Ratio (SNR) at the AP, which means that the communication range between the camera and the AP would be limited to short distances. Solving this dynamic range problem could be done using an Automatic Gain

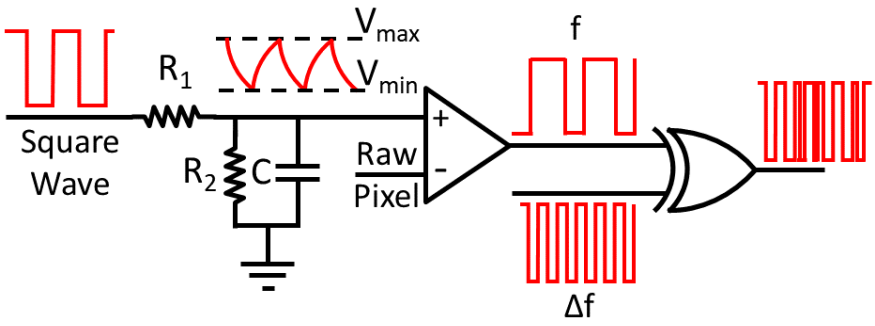


Figure 2.7: **Pulse Width Modulation Backscatter Architecture.**

Control (AGC). However, this typically involves a power-hungry linear amplifier. Another approach could use a high-resolution ADC to convert the image sensor's analog voltage to a digital signal and send the video frames digitally, leveraging digital backscatter [3]. However, this alternative faces the same issues as the AGC approach and cannot meet our power constrains.

We choose to solve this problem using Pulse Width Modulation (PWM) to send video frames to the AP. This is equivalent to a single-bit ADC using PWM, whereby our analog data translates to the timing information of the pulses. In other words, the duty cycle of the pulses is defined by the analog voltage of raw pixel values. We design our PWM module using a passive RC block and a comparator.

Fig. 2.7 shows our PWM design. One way to generate a PWM encoded signal uses a triangular wave with a primary frequency of f as the reference and compares it to the raw information-containing signal [6]. Comparator output is a PWM signal with a primary frequency of f and a time-varying duty cycle proportional to the amplitude of the raw input signal.

Here, we use the FPGA to generate a square wave with a primary frequency of f and amplitude of A, which is then low-pass filtered to generate an approximated triangular wave. We can adjust the triangular wave's V_{max} and V_{min} by tuning R_1, R_2, C and A according to equations (2.4) and (2.5). During charging period, the capacitor's voltage increases from an initial V_{min} with time constant $\tau = (R1||R2)C$ to the maximum voltage of $V_{max} <$

$V_M = AR_2/(R_1 + R_2)$. During discharge, the capacitor's voltage decreases from the initial V_{max} with the same time constant, τ.

$$V_{charging}(t) = (V_M - V_{min})(1 - e^{\frac{-t}{\tau}}) + V_{min} \tag{2.4}$$

$$V_{discharging}(t) = V_{max}e^{\frac{-t}{\tau}} \tag{2.5}$$

$$V_{max} = V_M(1 - e^{\frac{-1}{2f\tau}}) + V_{min}e^{\frac{-1}{2f\tau}} \tag{2.6}$$

$$V_{min} = V_{max}e^{\frac{-1}{2f\tau}} \tag{2.7}$$

By setting $t = 1/2f$, the triangular wave's maximum and minimum voltage can be calculated using equations (2.6) and (2.7). V_{max} and V_{min} should be set to ensure the image sensor's analog output falls within the V_{max} and V_{min} range. Finally, the triangular wave is compared to the image sensor's analog output. The comparator's output is a PWM signal whose duty-cycle is proportional to the analog voltage of the raw pixel. Equation (2.8) shows the duty cycle corresponding to analog pixel value P.

$$PWM(P) = 0.5 + f * \tau * ln(\frac{V_M - P}{V_M - V_{min}} * \frac{V_{max}}{P}) \tag{2.8}$$

A conventional challenge in backscatter-based communication systems is in-band interference caused by the transmitted single-tone RF signal. The receiver picks up this very strong signal and, if it falls in the communication band, its phase noise can completely overwhelm the received signal. To solve this problem, sub-carrier modulation is used to shift the backscattered information frequency spectrum. In our system, we use an XOR gate to shift the backscattered data frequency spectrum by Δf. We input our PWM signal and a square wave with the primary frequency of Δf to an XOR gate and the output is an up-converted version of the PWM signal. This technique addresses the self-interference problem, which increases the signal-to-noise ratio (SNR) and thus the operating range of our backscatter communication.

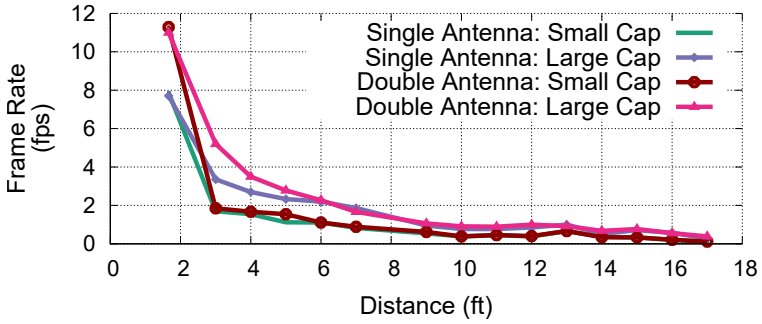


Figure 2.8: **RF Harvesting Evaluation.** Frame rates of our video streaming camera at different distances when it is powered only by the RF harvester. Note that we define the frame rate as average number of frames over a short period of time (a few minutes).

Dual antenna architecture. Our system uses two antennas: one connected to the RF harvester to harvest RF energy, and the second connected to an RF switch to enable backscatter communication. Previous design [3] uses a single antenna for both energy harvesting and backscatter communication, and loading of the antenna switches between short and matched impedance when backscattering. In our dual antenna design, loading of the backscatter antenna switches between open and short impedance, which are farther apart on the Smith Chart [30]. This results in a higher Delta Radar Cross-Section, improving the SNR and thus communication range of our dual antenna approach relative to a single antenna implementation.

2.4 Evaluation and Application

We now evaluate multiple aspects of our battery-free video streaming camera. First, we assess our RF and solar power harvesting circuits. Next, we characterize the performance of our battery-free video streaming camera using our dual harvester design. We then show the video quality of our video streaming camera. Finally, we evaluate our system in a real application deployment.

RF harvesting evaluation. We deploy our video streaming device in a lab environment. We use a USRP X300 software-defined radio connected to a power amplifier to generate a

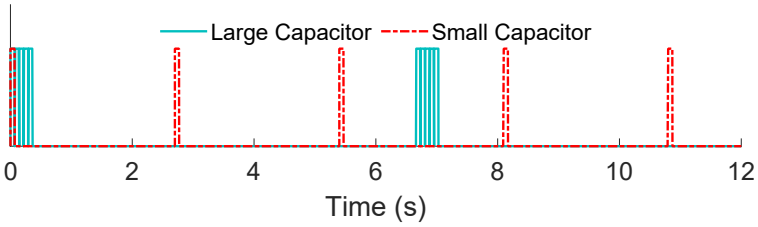


Figure 2.9: **Time Domain Comparison of Video Frame Generation Using Different Capacitor Sizes.** A large capacitor produces bursts of frames, while a small one produces single frames at shorter intervals.

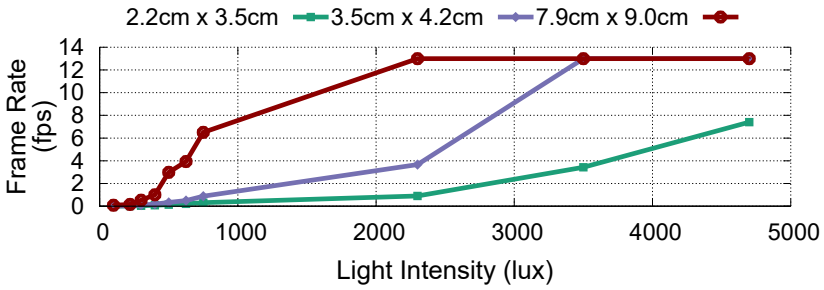


Figure 2.10: **Solar Harvesting Evaluation.** The frame rate of our video streaming camera at different lighting conditions for three different solar cell sizes. We collect data for light intensity of $lux < 750$ in an office building and the rest in outdoor environment.

single-tone RF signal. We set the power amplifier output to 30 dBm and connect it to a 6 dBi patch antenna to comply with FCC regulations for the 900 MHz ISM band. We evaluate our RF harvester for both single and dual antenna approaches.

For the single antenna approach, we use a 2 dBi whip antenna for both energy harvesting and backscatter communication and tune the matching network to ensure the antenna is matched to a 50 Ω load. We also evaluate RF energy harvesting in a dual antenna approach, assigning one antenna for backscatter communication and the other for energy harvesting.

Fig. 2.8 shows the update rate of our battery-free video streaming camera at different distances from the AP. This plot shows that as distance increases, frame rate decreases. We also evaluate the RF harvester using a small or large storage capacitor. As shown in

Fig. 2.8, using a large storage capacitor increases the frame rate in both cases. According to equation (2.1) increasing capacitor size results in storing more energy and therefore sending more video frames once the capacitor discharges. As a result, the controller must initialize the image sensor only once. In contrast, using a small capacitor results in repeated image sensor initialization for lower numbers of video frames. Thus, using a large capacitor increases the frame rate slightly; however, a large capacitor produces bursts of frames, while a small one produces single frames at shorter intervals. Fig. 2.9 shows the distribution of video frames over time.

We can achieve up to 12 fps at close distances when we use two antennas; in contrast, we achieve only 8 fps when we use the single antenna architecture. This is because during backscatter communication in the single antenna approach, the load connected to the antenna is modulated between a matched and short-circuit, causing reflection of some incident RF power rather than full absorption. In other words, during backscatter, RF to DC efficiency drops due to antenna load impedance modulation. However, at longer distances, the frame rates of the dual and single antenna approaches converge because the frame rate is very low and the battery-free camera spends a negligible period of time in backscatter mode.

Solar power harvesting. We next evaluate the performance of our video streaming camera powered by the solar power harvester. In this experiment, we use a small capacitor. We evaluate our system indoors and outdoors under different lighting conditions. Fig. 2.10 shows the results for three different solar cells with dimensions of 2.2 $cm \times 3.5$ cm, 3.5 $cm \times 4.2$ cm, and 7.9 $cm \times 9$ cm. In indoor scenarios, under normal office lightning conditions ($lux \approx 500$), our camera transmits up to 3 fps video frames. However, in an outdoor environment with a light intensity of $lux > 4500$ (not exposed to direct sunlight), our video camera backscatters 8 fps video using the solar cell with the smallest surface area.

Dual power harvesting. To evaluate our dual harvester video streaming camera, we use a USRP X300 software-defined radio connected to a power amplifier to generate a 30 dBm single-tone signal at 900 MHz into a 6 dBi antenna and to receive the backscattered video frames from the camera. We use two 2 dBi whip antennas for RF harvesting and backscatter

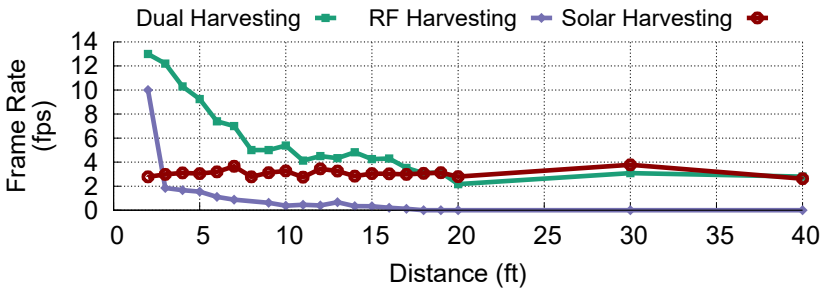


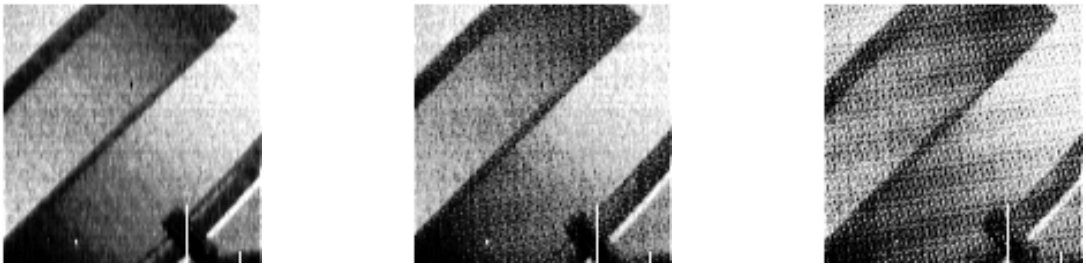
Figure 2.11: **Dual Power Harvesting Evaluation.** Video streaming camera operating with a single or dual power harvesting source. Note that we show this plot up to 40 ft to better represent our results at near distances. This result will persist up to 150 ft with the same trend as we see after 20 ft.

communication. For the solar harvester, we use a halogen lamp to provide a light intensity of 500 lux at the surface of the solar cell. We connect a 9 $cm \times 7.9$ cm solar cell to our solar power harvester and use separate small capacitors for each power harvester to ensure the camera sends only one frame when the storage capacitor is charged to V_{max}.

To perform this evaluation, we vary the distance between the camera and the AP and measure the frame rate of our battery-free video streaming camera which is powered by the dual power harvester. Fig. 2.11 shows the frame rate of the camera at different distances; we also show the frame rate of the camera when it is powered by the solar harvester at fixed 500 lux and by the RF harvester. We learn the following from this plot:

- Up to a distance of 18 ft, where the RF power harvester stops working, we observe a monotonically decreasing frame rate as the distance increases. After the distance of 18 ft, the frame rate becomes almost constant because the camera is being powered only by the solar cell, which is independent of the distance from the AP. At very close distances (less than 8 ft), the camera harvests a considerable amount of energy from the RF source, bringing the frame rate up to about 13 fps at 2 ft.

- The frame rate of the dual harvester is greater than the sum of the frame rate of indi-



(a) Baseline (b) $PSNR = 33\ dB$ (c) $PSNR = 28\ dB$

Figure 2.12: **Sample Video Frames.** Video frames from our video streaming camera at different PSNR values.

vidual harvesters because individual harvesters must initialize the image sensor every time they power up the camera; thus, they use some of the stored energy for initialization. In contrast, for the dual harvester, the time during which the RF harvester provides energy can overlap with the time during which the solar harvester powers up the camera. Therefore, the camera remains active and does not need to initialize the image sensor. Instead, it uses initialization energy to send more video frames to the AP.

Analog video quality. To evaluate the video quality of our backscatter communication system, we power our video streaming camera with a battery. We use a USRP connected to a power amplifier as the AP to transmit 30 dBm single-tone signal at 900 MHz into a 6 dBi antenna and receive the backscattered video frames from the camera. We vary the distance between the camera and AP and, at each distance, we send a 20 second video clip at a rate of 13 fps. To collect the ground-truth video, we record the camera's output with a National Instrument USB-6361 DAQ. We use the PSNR metric to evaluate our received video quality using PWM backscatter communication. We calculate PSNR using received video frames at the AP and recorded data with the DAQ. As a rule of thumb, video frames with a PSNR of 25 dB or higher are considered to be acceptable frame quality compared to the ground-

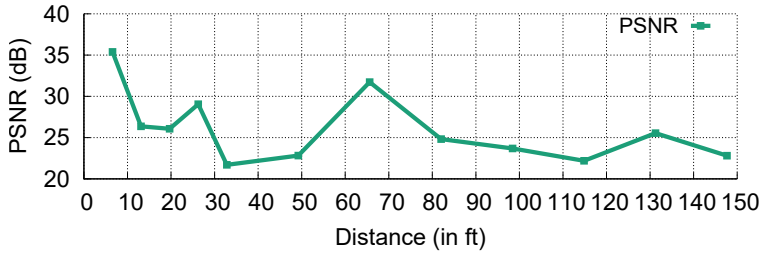


Figure 2.13: **Analog Video Quality.** The quality of backscattered video frames over distance using the PSNR metric.

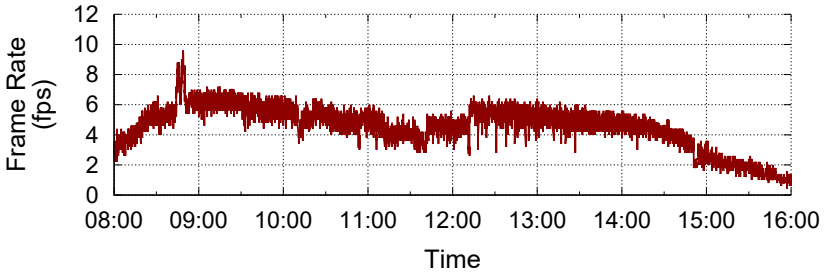


Figure 2.14: **Application Deployment.** Video streaming frame rate achieved by our battery-free video camera over the course of eight hours.

truth. We plot snapshots of video frames for a gray-scale ramp image in Fig. 2.12, which are recorded with our video streaming camera along with corresponding PSNR values to show the quality of our recorded video.

Fig. 2.13 shows PSNR at different distances in a room with low lighting conditions ($lux = 100 - 200$). In general, as distance increases, PSNR decreases as well. However, due to multipath effects, at some locations the PSNR increases as distance increases. The average PSNR of the received signal is greater than 22.5 dB at a distance of up to 150 feet from the AP. Beyond this distance, the USRP cannot decode the frames reliably since the SNR of backscatterd video from the camera falls below the minimum required SNR at the receiver input.

Application deployment. Wireless cameras are very popular for security applications and smart home monitoring systems. However, existing wireless cameras must either be plugged in or require frequent battery replacement/recharging. To demonstrate our battery-free

video camera's applicability for surveillance and home monitoring applications, we deploy our dual-harvester battery-free camera in an outdoor environment, connect a solar cell with dimensions of 4.5 cm×2.2 cm to the solar harvester, and use a USRP X300 as the AP. We set up our camera at the distance of 10 ft from the AP and record video frames from 8 AM to 4 PM during a sunny day where light intensity changes over time. Fig. 2.14 shows the frame rate of our battery-free video streaming camera during this eight hours of operation. The camera provides video at frame rates that vary between 1 fps and 9 fps depending on power availability. We expect that the solar harvester provides power that varies at a lower rate than the power provided by the RF harvester, because the AP is placed in a lab environment with people moving around. Thus, jitters observed in the frame rate are caused by abrupt changes in the RF power absorbed by the RF harvester.

Before 8 AM and after 4 PM, the amount of available solar energy is insufficient and the camera is powered only by the RF harvester. In this case, since the distance between the camera and the AP is 10 ft, our battery-free video streaming camera sends video frames to the AP with the frame rate of 0.4 fps.

2.5 Conclusion

This paper presented the first battery-free video streaming camera with a dual power harvester that combines energy from both light and RF sources. By combining RF and light power, we achieved a higher frame rate at short distances compared to an RF-only power harvester. In addition, the use of solar power harvesting increased the operating range of our battery-free video camera to the point where it was no longer limited by RF power harvesting. We proposed a video streaming architecture with two separate antennas for backscatter communication and RF power harvesting to increase the efficiency of our RF power harvester and improve the range of backscatter communication. Finally, we deployed our video streaming camera for a day-long experiment, showing its potential for surveillance and monitoring applications.

Chapter 3

MULTISCATTER: MULTISTATIC BACKSCATTER NETWORKING FOR BATTERY-FREE SENSORS

3.1 Introduction

Recent advances in energy harvesting, physical sensors, wireless networks, data processing, and machine intelligence picture a future where billions of everyday objects turn into smart, connected devices that can sense their surroundings, communicate their data, and react to human interactions and other environmental stimuli. This vision, however, has run into practical limitations of wireless communication networks.

An ideal wireless technology for a battery-free or self-powered sensor network should possess two essential qualifications: the sensor nodes should consume very little energy since the harvested energy is scarce, and the infrastructure to communicate with the sensor nodes should be low-cost to make widespread adaptation feasible. Current commercial wireless technologies do not satisfy these two objectives simultaneously. While recent works have demonstrated battery-free BLE [31,32], Sub-GHz [33], and LoRa [34] networks, these radios may consume too much energy to support applications demanding update rates of multiple packets per second [35]. On the other hand, passive RFID readers are expensive and suffer from a short communication range, resulting in high infrastructure cost to cover a practical setting with passive RFID readers [36]. In addition, RFID does not typically support the use of arbitrary sensors such as cameras and microphones.

Recent bistatic backscatter systems [11,17,37,38] have shown promise to find a middle-ground to this challenge by employing the backscatter technology to reduce the energy burden of wireless communication, while leveraging the economies of scales and ubiquity of industry-standard protocols such as WiFi, Bluetooth, ZigBee, and LoRa to reduce the cost of reader

infrastructure. Bistatic backscatter systems still suffer from a shorter communication range than conventional active radios, and a single receiver (RX)-transmitter (TX) pair is insufficient to cover a multi-bedroom or multi-floor house at practical data rates. The limited coverage of backscatter systems complicates the user experience and limits their use cases.

To extend the coverage of a wireless network, the idea of using multiple base units is suggested and implemented in cellular [39, 40] and WiFi [41, 42] networks for a long time. However, high propagation loss, strong interference, frequency-dependant operation and non-homogeneous base units complicates the deployment of a backscatter network with multiple base units (see §3.2 for a primer on backscatter systems).

Furthermore, the battery-free sensor nodes have strict energy limitations, while the base units have higher computational and energy resources. This imbalance requires the system to push the network management tasks to the base units while keeping the sensor nodes as simple as possible.

We notice that unlike a cellular base station that covers a circle around itself, the coverage area of the bistatic backscatter system is almost entirely limited to the area between the TX and RX units (see Fig. 3.2(a)). This difference means that we do not need another TX-RX pair to double the coverage area of a bistatic backscatter system. We only need to add a second TX unit and share one RX unit between the two TX units (or vice versa). We extend this idea of sharing TX units and RX units among one another, to form a multistatic network of many TX and RX units that covers more extensive areas, as shown in Fig. 3.1. This solution still uses backscatter to communicate with the sensor nodes, which reduces the energy harvesting barrier for a practical solution. Although this solution needs multiple base units similar to an RFID solution, each TX or RX base unit is significantly cheaper than an RFID reader.

3.1.1 MultiScatter

In this work, we present MultiScatter, the first multistatic backscatter network for battery-free sensing, where a network of multiple RX and TX base units is employed to deliver

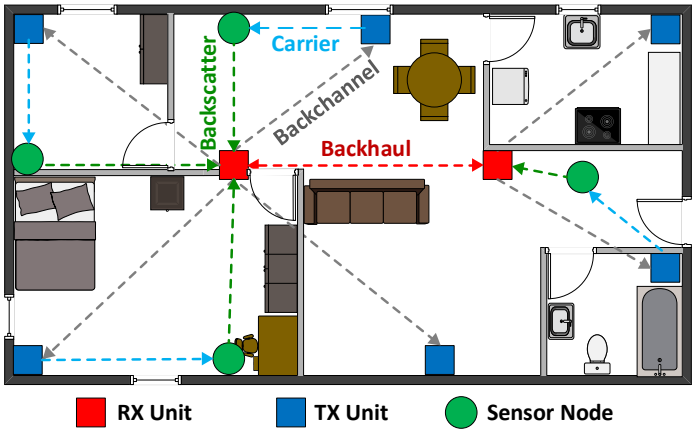


Figure 3.1: **We deploy multiple RX (red squares) and TX (blue squares) base units and introduce several coordination techniques to extend the coverage of low-cost backscatter networks and deliver seamless connectivity to battery-free sensor nodes (green circles).**

backscatter connectivity to many battery-free sensor nodes over a wide area. We extend the idea of bistatic backscatter with a single RX-TX pair to multistatic backscatter with multiple RX and TX base units, where any RX-TX pair can communicate with any sensor node in its vicinity. We introduce a MAC layer for the multiscatter backscatter network that performs two critical tasks: first, it selects the best RX-TX pair, carrier power, and frequency in communicating with each sensor node to maximize the throughput, and second, it handles the handover between base units as the sensor node moves inside the coverage area. This seamless connectivity addresses one of the remaining challenges in using backscatter systems and creates many new opportunities in smart homes and smart buildings by enabling peripheral, wearable, and medical devices such as pill bottles [43] and insulin pens [44] to use backscatter to transfer data to the cloud.

Summary of architecture

Fig. 3.1 shows the architecture of the system. The RX units manage the operation of the TX units and sensor nodes. We make this design choice since, unlike the TX units, the RX units have direct access to the backscatter channel state information.

The RX units support three different wireless communication interfaces. They can receive backscattered packets from the sensor nodes (dashed green line in Fig. 3.1). The RX units can also actively send packets to the TX units; we call this the backchannel link (dashed gray line). The RX units use the backchannel link to activate the best TX unit for each sensor node and set TX parameters like carrier frequency and power. Finally, a backhaul link (dashed red line) allows RX units to communicate with other RX units to share network state information, including the sensor nodes' connection states.

The TX units support two wireless interfaces: they receive active packets from the RX units, and transmit the commands and carrier signal to the sensor nodes (dashed blue line).

Deployment design support tool

To handle the complex coverage scenarios of a multistatic backscatter system in indoor environments, we develop a deployment planning tool to assist us in finding the optimum placement of RX and TX base units. While using two different base unit types is a major factor in lowering the cost of backscatter infrastructure, it complicates the deployment. The backscatter coverage depends on several factors, including the carrier source power, receiver sensitivity, and receiver blocker tolerance [45], and the objects and obstacles in indoor environments further complicate the coverage of a multistatic backscatter system. Our tool takes into account the backscatter range equations, attenuation through obstacles in indoor environments, and possible base unit coordinates and suggests the best spots for RX and TX units.

Improved Power-Coverage-Data rate trade off

Our system design provides high data rate and wide coverage area for sensor nodes that are low-power enough to operate from harvested energy. The power reduction is achieved by using backscatter communication. Previous work has demonstrated battery-free microphones [20] and cameras [21, 46, 47]. However, these early results are not general: they relied on special-purpose analog modulation schemes that were custom designed for the specific

sensor data. This work presents a general-purpose sensor node platform that can backscatter arbitrary digital sensor data as 802.15.4g-compliant packets. The demonstrated sensors include cameras and microphones, as well as sensors for temperature, humidity, and illuminance. Our sensor nodes are equipped with an ultra low power radio to wake them from sleep mode and receive commands. The sensor nodes can operate entirely on the energy harvested from a $2in^2$ solar cell under indoor light intensity levels.

Previous long-range backscatter deployments such as [17, 18, 38, 48] used a single TX-RX pair and extended the communication range by using very sensitive, low data rate protocols. However, this technique is not scalable. The coverage area of one base unit has an upper limit, and one would need more than one base unit to extend the coverage beyond that. In this work, we provide wide area backscatter deployment coverage using a larger number of multi-statically defined small cells, which allows us to maintain higher data rates for the battery-free sensor nodes.

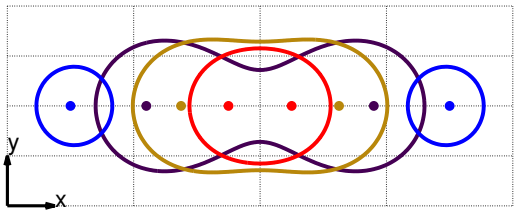
3.1.2 Contributions

Our main contributions are summarized below:

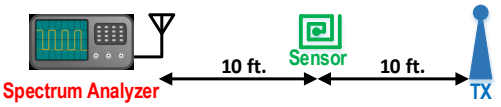
• We present the first multistatic backscatter network for battery-free sensing. Our system relies on three different wireless links and uses several coordination techniques between the base units to deliver a high throughput seamless connectivity over a wide area to battery-free backscatter sensor nodes. Our MAC layer implementation selects the best RX-TX pair in communicating with each sensor node to maximize the throughput and handles the handover between base units as the sensor node moves inside the coverage area.

• We develop a deployment planning tool to assist us in finding the optimum placement for RX and TX base units to maximize the coverage in indoor environments. Our tool considers the backscatter range equations, attenuation through obstacles in indoor environments, and potential base unit coordinates and suggests the locations for RX and TX units.

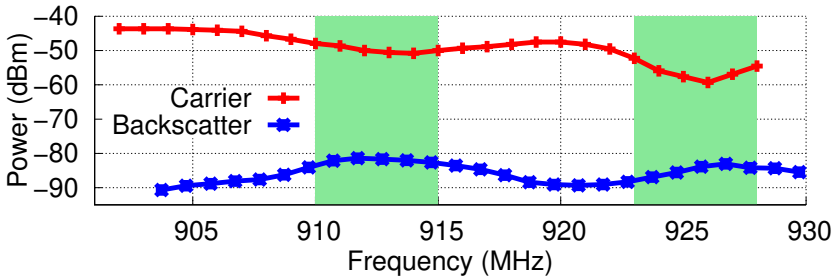
• We build the MultiScatter system, including the RX and TX base units, and sensor nodes



(a) Backscatter coverage dependence on the RX-TX distance.



(b) Multi-path fading experiment setup.



(c) Carrier and backscatter signal power at the RX.

Figure 3.2: **Backscatter coverage map (a). Effect of multi-path fading on backscatter link (b,c).**

and evaluate it in different scenarios. Our evaluations show that the MultiScatter can cover a 23400ft^2 educational complex using 5 RX units and 20 TX units for a total cost of $569. Furthermore, the aggregate throughput of the backscatter network increases by using more RX units. Using 5 RX units, our backscatter network collects data from multiple sensors at an aggregate rate of 375 Kbps, 4.17 times faster than a single sensor node throughput.

Platform availability. MultiScatter's design files are available at:

https://github.com/MultiScatter/Hardware

3.2 Backscatter Primer

Unlike conventional wireless networks where the communication happens between two devices- an access point and a mobile device- three devices are involved in a bistatic backscatter communication; a TX unit that generates the carrier signal, a backscatter sensor node that

modulates and reflects the carrier, and an RX unit that listens to, and decodes the sensor node data.

One of the main challenges in backscatter systems is the strong interference at the receiver due to the carrier signal. Bistatic, or half-duplex, backscatter systems rely on physical separation of the TX and RX units to reduce the carrier interference [11, 18, 37], while monostatic, or full-duplex, backscatter systems rely on self-interference cancellation circuits for this purpose [49–51]. Another popular technique to mitigate the carrier interference is using sub-carrier modulation at the sensor node [11, 12, 18, 37, 48, 49, 51, 52] to generate the backscatter packet at a frequency offset from the carrier signal. This technique pushes the carrier signal out of the desired band at the receiver and reduces the receiver's sensitivity loss, since receivers can tolerate out-of-band interference.

The coverage of a bistatic backscatter system depends on the relative distance of the RX and TX units and characteristics such as receiver sensitivity and blocker tolerance, transmitter output power, backscatter node conversion gain and antenna gain as described in [11, 45]. We use the signal propagation model to plot the backscatter coverage in four different cases in Fig. 3.2(a). The RX and TX locations are shown with circles of the same color in each case. When the TX and RX are too close to each other, they completely cover the area between themselves, but the carrier interference limits the coverage area (red plot). As the RX-TX distance increases, the coverage area expands to reach its maximum (brown plot). If we further increase the RX-TX distance, first, the coverage width at the center starts to shrink (purple plot), and finally, we lose the coverage at the center and the coverage map becomes disjoint (blue plot). Although we use a hypothetical backscatter system and the free-space propagation model in this analysis, the coverage variation trend is similar in all bistatic backscatter systems, and it highlights the importance of RX and TX units placement in covering large spaces with backscatter connectivity.

The wireless propagation in indoor environments is subject to multipath fading. To show the effect of the multipath fading, we perform a simple experiment. We set a transmitter to generate a continuous wave carrier signal, and a backscatter node, placed 10 ft away from the

TX, to reflect the TX signal at 1.7 MHz offset frequency, as shown in Fig. 3.2(b). We use a spectrum analyzer, placed 10 ft away from the backscatter node, to measure the carrier and corresponding backscatter signal powers. Fig. 3.2(c) shows the carrier and backscatter signal power as we sweep the carrier frequency from 902 MHz to 928 MHz. The backscatter (i.e., desired) signal power varies 10dB in this experiment, while the carrier (i.e., interference) signal power varies 15dB. The two signal powers are independent of each other, and their peaks and valleys happen at different frequencies. The best frequency bands for backscatter communication are the ones with maximum backscatter signal power and minimum carrier power as shown in Fig. 3.2(c) with green strips.

3.3 System Design

This section describes the backscatter network architecture used to cover a wide area and communicate with many sensor nodes. We start with one RX-TX pair and extend the coverage first by adding more TX units and finally by including more RX units. Throughout this section, we assume that each sensor node has a unique identification number (ID), and the backscatter network has prior knowledge of the sensor nodes' IDs.

3.3.1 Single RX - Single TX

In the simplest form, a bistatic backscatter system consists of one RX unit, one TX unit, and one or several sensor nodes. The RX and TX should coordinate with one another to achieve three main functionalities: improving the link reliability, time-frequency alignment, and handling retransmissions. We use a wireless backchannel between the RX and TX units to communicate and coordinate between the two devices.

The communication cycle of the system is shown in Fig. 3.3. It starts with selecting the optimum parameters for the frequency channel and carrier power to maximize link throughput and minimize the error rate in communicating with the specified sensor node. The RX uses scores calculated based on the results of previous communication cycles for this purpose. The RX selects the frequency channel with the highest score that satisfies the frequency

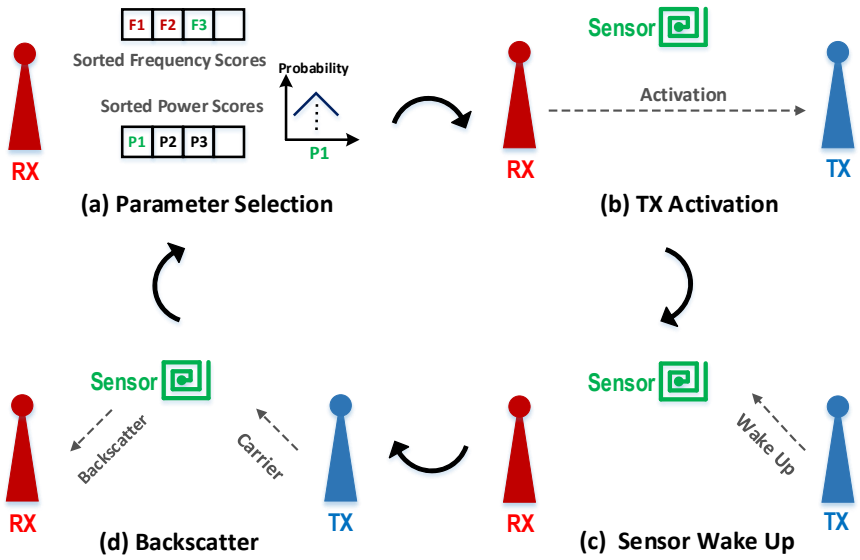


Figure 3.3: **Proposed Communication cycle.**

hopping requirements of the regulatory mandates. It also finds the carrier power with the highest success rate and uses it as the center of probability distribution to select the carrier power (Fig. 3.3(a)). After selecting the communication parameters, the RX shares them with the TX unit. The TX unit has been in listening mode until it receives an activation packet from the RX unit on the backchannel. The activation packet also specifies the duration of the carrier signal that the TX unit has to generate, as well as the sensor node identification number and the command for the sensor node (Fig. 3.3(b)). Once the TX unit receives the activation packet, it sends a packet to the sensor node, which contains the sensor node ID and the requested command, to wake the sensor node (Fig. 3.3(c)). Finally, the sensor node wakes up, receives the command, and responds appropriately by transmitting one or several backscatter packets. During this period, the TX generates the carrier signal, and the RX listens for the backscatter packets (Fig. 3.3(d)). When the backscatter duration is over, the RX unit analyzes the received packets based on the command issued to the sensor node and decides whether retransmission is required or not. It also updates the frequency and power score vectors based on the communication cycle results. The devices follow the same cycle

for the subsequent communications between them.

The first step of the communication cycle is to select backscatter frequency and power level. This step is vital to improve the overall throughput and reliability of the system since backscatter and carrier signal strengths at the receiver vary significantly at different frequency channels due to the multipath in indoor environments, as mentioned in Sec. 3.2. Similarly, the carrier power could affect the performance of the backscatter communication too. Increasing carrier power increases both the backscatter signal power and TX carrier interference at the RX unit at the same time, and might improve or deteriorate the performance of the backscatter link [45]. We assign a success score to each communication cycle based on the ratio of the successfully received packets to the total expected packets, and update the score for the frequency channel and power level used in the cycle with a moving average formula. To explore the entire space of possible frequency and power values, we set an exploration probability based on the error rate over a predefined number of previous communication cycles. In exploration mode, we select the two parameters randomly [37].

The communication cycle allows the RX unit to control the TX carrier frequency, start time and duration through the activation packet. This control is essential for two reasons. First, the TX unit can not generate the carrier signal indefinitely since it would waste energy and generate too much interference for other wireless users in the vicinity. The TX unit should only generate the carrier when required by the sensor node, which means the devices should be time-aligned to utilize the spectrum and energy resources efficiently. Furthermore, the TX has to hop between different frequencies in the ISM band to satisfy regulatory mandates [53]. Since the desired signal frequency at the RX is equal to the TX frequency plus the subcarrier modulation frequency of the sensor node [11, 18, 37], the RX needs to know these two values in order to frequency align itself with other devices. Using the activation packet to initiate and control the TX operation guarantees time and frequency alignment between the devices.

In the proposed communication cycle, the RX unit sends the sensor node command to the TX unit, and the TX unit passes the command to the sensor. Thus, the data communication

effectively happens between the RX unit and the sensor, with the TX unit facilitating the data transfer between them. The two base units have different capabilities in a bistatic backscatter network. The RX unit demodulates the backscatter signal and detects whether the desired packet is successfully received or failed, and the TX unit can communicate with the sensor node. By using the TX unit as a relay, the RX unit has complete control to ask for retransmissions.

3.3.2 Single RX - Multiple TX

The system described in Sec. 3.3.1 highlights the fact that the data exchange happens between the RX unit and the sensor nodes. Thus, adding more TX units to the system improves the backscatter system coverage without significantly affecting the system architecture. One approach to developing such a network with multiple TX units is to command all units simultaneously to generate a carrier signal at a specific frequency. However, this method faces two major challenges: first, using multiple transmitters would not guarantee better performance because signals from different transmitters might combine destructively and decrease the backscatter signal power at the receiver. Second, it would increase the system energy consumption and interference to other wireless systems in the vicinity. Thus, the communication system must select one TX unit for each communication cycle to maximize the throughput. We use the RX unit to control the system's operation and set the TX unit at the start of each cycle.

To compare the performance of different TX nodes in communicating with each sensor node, we assign a confidence score to each TX unit. The confidence scores are initialized to 1 and updated with an exponential moving average formula using the cycle success rate each time the TX unit-sensor node combination is used. Fig. 3.4 shows sample confidence tables in the single RX - multiple TX arrangement for two sensor nodes. The frequency and power score vectors depend on the geometrical placement of the devices with respect to each other and change based on the selection of the TX unit. Thus, the RX unit needs to record one set of these score vectors for each TX unit-sensor node combination.

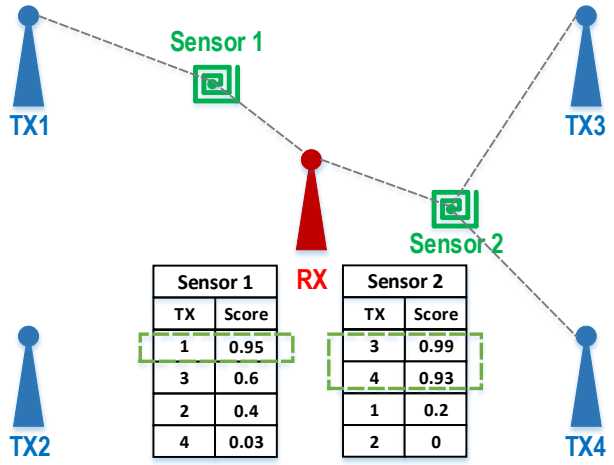
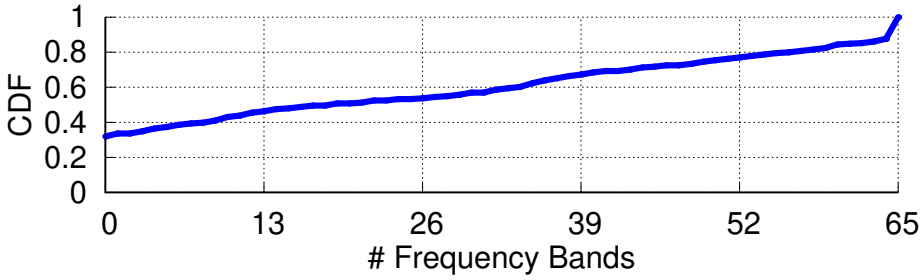


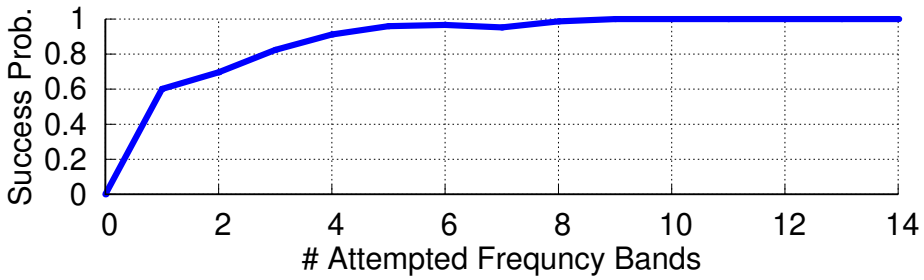
Figure 3.4: **TX unit selection in a single RX - multiple TX scenario based on assigned confidence scores.**

We follow the following procedure to select a TX unit based on the confidence table at the start of the communication cycle. First, we select the TX units with confidence scores higher than 0.9. If the set has more than one member, we use a weighted random selection to select one of them. If the set has no member, we filter out the TX units with confidence scores less than 0.2 times of the maximum confidence score, and use a weighted random selection to select one of the remaining TX units. Once the TX unit is selected, we follow the communication cycle, as explained in Sec. 3.3.1, with the selected TX unit. We assign a unique ID to each TX unit and include the transmitter ID in the activation packet.

Similar to the single RX-single TX architecture, the TX units are in listening mode until they receive the activation packet. The TX units only move forward with transmitting the carrier if their ID is matched to the one in the activation packet. Otherwise, they discard the activation packet and return to the listening mode.



(a) Distribution of number of operative bands at 240 locations.



(b) Probability of communication with a sensor node based on number of equal-distant frequency bands attempted.

Figure 3.5: **Selecting the number of frequency bands for the search procedure.**

3.3.3 Multiple RX - Multiple TX

Increasing the number of TX units around one RX unit allows us to expand the backscatter coverage around the RX unit in different directions. However, we can only achieve a truly scalable system and cover areas beyond the reach of a single RX-TX pair by employing more than one RX unit. In the Multiple RX - Multiple TX arrangement, we break the area that we need to cover into several sections and use one RX unit with one or multiple TX units to cover each section. Similar to cellular networks, a sensor node might move between these sections. Since the backscatter sensor nodes do not have the hardware or energy resources to determine the link quality metrics, the base units have to handle the handover. We add a server role to the network that assigns the sensor nodes to the RX units and updates these assignments as the sensor nodes move. The server could be one of the RX units, or a separate

device. We equip the RX units with WiFi and use it as the backhaul link to communicate with the server.

Once a sensor node moves from section A to section B, we need to change its assigned RX unit. For this purpose, first, the RX_A must alarm that the sensor node is missing, and then, the RX_B must search for the sensor node and successfully communicate with it. A sensor node could move to any other section between two communication cycles in a backscatter network, since the sections are small and the time interval between communication attempts could be long.

We use the number of consecutively dropped packets as the metric to detect a sensor node is disconnected from its assigned RX unit since once a sensor node leaves a given section, all backscatter communication attempts with that sensor would fail.

We define the re-assignment cost ($Cost_{RA}$) as the average number of communication cycles that the system needs to correctly assign a new RX unit to a relocated sensor node. We can calculate the re-assignment cost in number of communication cycles using equation 3.1:

$$Cost_{RA} \ = \ (Thr_{drops} \ + \ 0.5 \ \times \ N_{TX} \ \times \ N_{ch}) \ \times \ (\frac{1}{1 \ - \ PFA}) \tag{3.1}$$

where Thr_{drops} is the threshold for consecutive dropped packets that is used to flag a missing node, N_{TX} is the number of TX units in the network, N_{ch} is the number of communication cycles that an RX-TX pair needs to search for the sensor node, and PFA is the probability of false alarm. There is a trade-off in selecting the threshold to announce the disconnection to the server. Higher thresholds waste network resources by trying to communicate with a sensor node that has left the section. On the other hand, lower thresholds increase the probability of false alarm. To select the optimum threshold that minimizes the cost, we need to know the N_{ch} for a reliable search procedure and model the probability of false alarm at different Thr_{drops} values.

Search Procedure (N_{ch} selection). An ideal search procedure should be fast and accurate. Due to the lack of existing knowledge about channel properties, the RX units have to try different frequency channels to increase the likelihood of successful communication with the

sensor node at the new location. However, increasing the number of frequency channels in the search procedure would increase the search duration (N_{ch}). Previous studies [54, 55] show that the multipath fading loss in indoor environments becomes uncorrelated as the frequency separation increases. To find the optimum number of frequency bands for the search procedure, we set up TX-RX base units at 12 different locations with 20 sensor node testing points (240 total) around them. For each setup, we set the carrier power at the maximum and try all the frequency bands between 902 MHz and 928 MHz with 400 KHz channel spacing to communicate with the sensor node at each point.

Fig 3.5(a) shows the distribution of number of operative bands over the 240 points, with 35% of the points not responding at any frequency and 13% of the points responding at all of them. We consider the points which respond at least at 10% of the frequency bands (150 points) as the target group and plot the probability of communicating with the sensor at these target points based on the number of attempted frequency channels in Fig 3.5(b). The figure shows that trying 9 channels results in 99% search accuracy. We consider some margin for errors in these experimental results and use 12 equally spaced frequencies in our search routine, which means the search procedure takes 12 commutation cycles.

Threshold Selection. The probability of false alarm (PFA) is the ratio of false alarms to total alarms. The main cause of successive dropped packets that leads to a missing node alarm is the movement of the sensor nodes. Ideally, we would like the RX unit to generate an alarm only when the sensor node is moved to a different section. However, moving the sensor within the boundaries of a section could result in multiple consecutive dropped packets, too, since the RX units have to re-tune the score vectors for the new sensor location as described in Sec. 3.3.1. This further complicates the threshold selection. We can calculate the false alarm rate using equation 3.2:

$$PFA = P_{inside} \times PFA_{inside} + P_{outside} \times PFA_{outside} \tag{3.2}$$

where P_{inside} and $P_{outside}$ are the probability of a sensor moving inside and outside of a given section respectively, and the PFA_{inside} and $PFA_{outside}$ are the probability of generating an

alarm as a result of those moves. Based on our definition, any alarm generated when the sensor moves outside the section is true, and hence, $PFA_{outside}$ is equal to zero. Also, we assume an 80%-20% split between the probabilities of moving inside and outside a section. Thus, to estimate the PFA, we must model the probability of false alarm caused by moving a sensor inside a section for different thresholds.

We perform an experiment by moving the sensor node within the boundaries of one section 100 times in 5 different environments. Fig. 3.6 (blue line) shows the probability of generating an alarm for different thresholds in these experiments. If we apply these probabilities to equation 3.2 and assume N_{TX} and N_{ch} are equal to 10 and 12 in equation 3.1, we can calculate the re-assignment cost for different threshold levels as shown in Fig. 3.6 (red line). Based on this plot, the optimum threshold of consecutive dropped packets for a system with 10 TX units is 22 packets.

Server. The server tracks the assigned RX unit for each sensor node. Once one of the RX units detects a missing sensor, it notifies the server. The server then randomly notifies one of the other RX units to search for the missing sensor node and waits for the search result. If the search fails, the server notifies another RX unit, and this procedure continues until one of the RX units successfully communicates with the sensor node. The server holds a predefined period before notifying the same RX unit again.

3.3.4 Base Units Placement

Selecting the number of base units to cover an area and finding their optimum placement could be challenging, especially in residential and commercial buildings, which are the primary target of the MultiScatter system. Thus, we need a tool to help us plan the deployment and estimate the coverage area for a given placement of base units without the time-consuming process of deploying the units in the field. Although several RF propagation simulators are available [56, 57], to the best of our knowledge, none of them supports backscatter systems.

Several factors complicate the placement of base units in indoor environments. First,

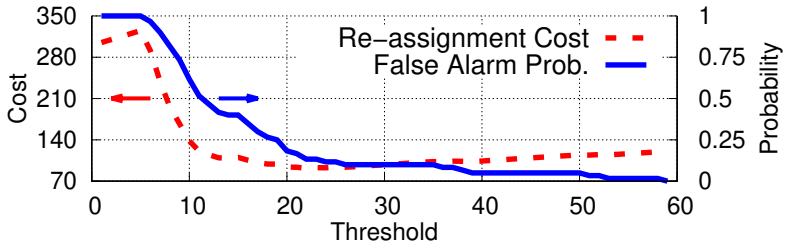


Figure 3.6: **Probability of false alarm (blue) and the overall re-assignment cost in communication cycles (dashed red) for different consecutive dropped packets thresholds.**

the coverage of a single RX-TX pair of our multistatic network is a complex function of the RX-TX relative distance, as shown in Fig. 3.2(a), as well as other factors such as TX carrier power, RX sensitivity, and blocker tolerance [45]. Furthermore, indoor areas are most often divided into several sections by walls, partitions, or other obstacles. The RF signals attenuate as they pass through different materials. It is essential to consider the effect of these obstacles in the placement of the RX and TX units to achieve optimal performance. Finally, the base units have high energy consumption and need a permanent energy supply such as a wall power outlet. This adds another limit to the placement of the base units since we prefer to keep the base units closer to the wall power outlets and avoid long wires.

Our tool uses the Friis formula [58] to estimate the backscatter signal and carrier interference power levels at the receiver. We use the backscatter coverage equations in [45] to determine the backscatter coverage at each point in the environment. We estimate the attenuation through walls and other obstacles using the proposed models in [59–61]. Although our simulator is limited to path loss and absorption loss models in two dimensions and does not consider RF propagation effects such as reflections, diffraction, and scattering, it still provides valuable insight for placing the target area's base units.

Our deployment tool takes the geometry of the area and coordinates of the base units as the inputs and calculates the maximum backscatter signal power amongst all the RX-TX pairs at each point to generate a coverage map. We follow a manual trial-and-error process

to find the number and position of the base units required to achieve the desired coverage. In each iteration, we move the base units and add more units, if needed, to get the desired coverage. More enhanced features, such as automatic placement of the base units to achieve defined coverage goals, are left for future work.

We can use a hard threshold coverage function that outputs one when backscatter signal power is above RX sensitivity and the carrier interference is below the RX blocker tolerance, and zero otherwise, or a soft threshold coverage function that transitions smoothly between zero and one.

We use our simulator to plot the backscatter signal power in a 900 sq^2 two-bedroom apartment unit in two different cases, a single RX - two TX (Fig. 3.7(a)), and a single RX - three TX (Fig. 3.7(b)) configurations, where a brighter color shows a stronger backscatter signal power. In this simulation, we assume all devices use antennas with 0 dBi gain, and frequency band, carrier power, backscatter conversion loss, path loss exponent, and attenuation through internal walls are equal to 915 $MHz, 25\ dBm, 4\ dB, 2$ and 15 dB, respectively. The results clearly show that adding one more TX unit and placing the base units appropriately have a significant effect on the performance of the MultiScatter system.

During our evaluations (Sec. 3.6), we notice that placing the base units closer to walls or other large objects could improve backscatter performance. The strong multipath resulted from such large objects could help the backscatter systems by creating a frequency-dependent non-flat attenuation profile, resulting in enough operative frequency bands for backscatter communication. Improving the simulator to model these higher-order effects and creating a more accurate coverage map is left for future work.

3.4 Sensor Node Design

Our goal is to design a battery-free sensor node platform that can accommodate different sensors and communicate with the base units. We reduce the wireless power consumption of the sensor node by using low-power backscatter for the sensor node uplink and an ultra-low-power wake-up radio for the sensor node downlink. The wake-up radio allows the sensor node

(a) One RX and two TX units

(b) One RX and three TX units

Figure 3.7: **Simulating the backscatter signal power for a two-bedroom apartment using the deployment planning tool.**

to stay in idle mode until the base units activate it. We also implement a gating technique to reduce the energy burden of power-hungry sensors by adding low-power auxiliary sensors. For example, a motion sensor is used to gate the operation of the camera. Finally, we use a low-power micro-controller with several communication protocols in our sensor node platform, which can interface with various commercial sensors.

FSK Backscatter. The sensor node transmits the data to the RX unit using Frequency-Shift Keying (FSK) backscatter communication. In FSK backscatter, bits '0' and '1' are transmitted by changing the state of the RF switch connected to the antenna at frequencies f_0 and f_1. This results in a backscatter packet with $(f_0 + f_1)/2$ subcarrier modulation and $|f_0 - f_1|/2$ frequency deviation.

We use a temperature-compensated VCO to generate the variable frequency signal that controls the RF switch. We use two NMOS switches to control the resistor that sets the oscillator frequency, as shown in Fig. 3.8. Switch Q1 controls the backscatter subcarrier modulation frequency, and switch Q2 sets the transmitted bit to '1' or '0'. We use the micro-controller SPI interface derived by the MCU oscillator to generate accurate bit periods. Compare to using two individual oscillators to generate f_0 and f_1 frequencies [48], using a single controlled oscillator to generate both frequencies reduces the power consumption and ensures phase continuity in switching between the two frequencies.

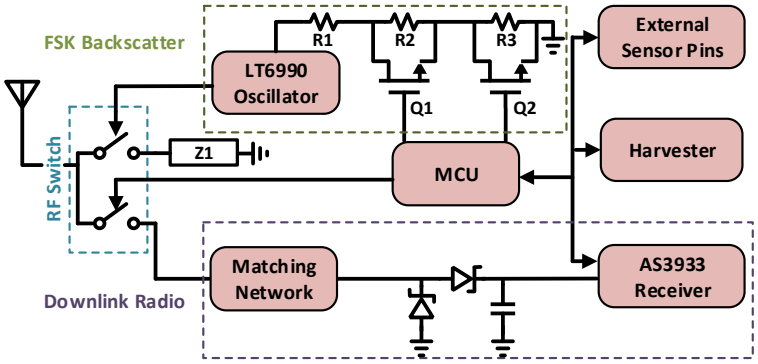


Figure 3.8: **Sensor node block diagram.**

Low Idle Power Consumption. The sensor node operates in the idle mode until it is activated by the base units to enable one of its sensors or report its latest recorded data. Therefore, our sensor nodes must be power efficient in the idle mode. Although the FPGAs allow us to implement custom communication protocols, they consume more power in the idle mode than the MCUs. We use a low-power MCU with several low-power idle modes in this work.

To wake up the MCU from the idle mode, we use a low-power and low-frequency wake-up radio [37]. While the MCU is in the idle mode, the wake-up radio listens to the packets transmitted by the TX units. If the received packet matches the sensor node's 16-bit ID, the radio generates an interrupt to the sensor node MCU and passes the received command to it.

Sensor Gating. We use low-power auxiliary sensors to gate the operation of the power-hungry sensors. For instance, a motion sensor can enable the operation of an image sensor when there is movement in the room. This helps us reduce the power consumption of the sensor node by limiting the operation of the power-hungry sensor (image sensor) to informative events. Similarly, the microphone activates only when the sound intensity passes a defined threshold.

We also implement second gating criteria by monitoring the energy storage element of the sensor node. The MCU measures the voltage of the energy storage element and sends it

back to the base units in response to each sensor enable command. The base units monitor this voltage and use it to gate the commands issued to the sensor node. If the voltage falls below a defined threshold, the base units bypass the sensor reading requests from that sensor node and allow the energy harvesting unit to charge the storage element. The base units continue to monitor the energy storage voltage with a more extended reading period and resume regular operation once the voltage reaches a certain level.

Energy Harvesting and Storage. The energy harvesting unit of the sensor node collects energy from ambient light and stores it on an energy storage element. The energy storage element connected to each sensor node should store enough energy to support reading the physical sensors connected to the sensor node and transmitting their data to the base units. While supercapacitors have longer life cycles and can be employed to build truly battery-free sensor nodes, rechargeable batteries have higher energy capacity and can tolerate longer energy drought periods [62, 63]. Our energy harvesting unit can work with both types of energy storage elements.

Modular Design. We follow a modular design approach to building our sensor nodes. The MCU, wireless communication units, temperature, humidity, and illuminance sensors are placed together as the basic sensor node. The energy harvesting unit, camera, and microphone sensors are designed as add-on boards that mount on top of the basic sensor node through 5 power and 12 Input/Output (IO) pins (see Fig. 3.9(c)-(e)). The modular design allows us to support more sensors without redesigning the communication section.

3.5 Implementation

We implement MultiScatter, consisting of the RX base units, TX base units, and sensor nodes, for operation in 902-928 MHz ISM band on 4-layer FR4 PCB. The system uses FSK modulation and 802.15.4g packet structure with seven preamble bytes, 16 CRC bits, and no whitening for communication between devices.

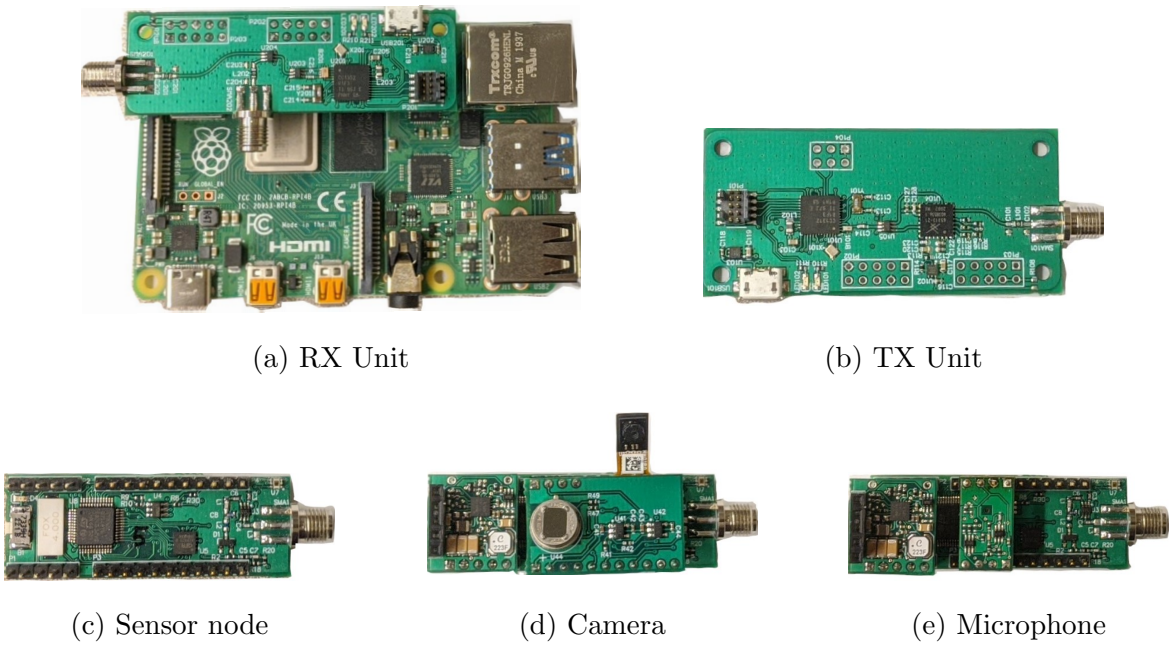
3.5.1 Base Units

Receiver Units. The RX unit has two parts, a communication board with the CC1352R wireless MCU, and a Raspberry Pi 4B single-board computer (RPi) with 2GB RAM for processing and backhaul. The two boards are connected to each other, as shown in Fig. 3.9(a), and communicate through a 1Mbps serial link. The CC1352R supports FSK modulation with data rates from 0.3 to 1985 Kbps and RX filter bandwidths from 4.3 to 3767 KHz. The RX filter bandwidth determines the carrier attenuation and should be set accurately considering the data rate, frequency deviation, and subcarrier modulation frequency. Our CC1352R board consumes 30 mW for its operation on average. The RPi board runs the Raspbian operating system. It processes the received data from the sensor nodes and determines the activation packet parameters. The RPi sends out this information to the CC1352R MCU to start the communication cycle, and the CC1352R MCU passes the received backscatter packets to the RPi as they are received. We implement the data handling and communication protocols explained in Sec. 3.3 using Python 3.7, which runs on the RPi.

Transmitter Units. Each TX unit consists of a CC1312R sub-GHz wireless MCU that receives the activation packet and generates the carrier signal, and a SKY65313-21 power amplifier (PA) that amplifies the carrier signal up to 28 dBm. Our TX unit consumes up to 3.3 W while transmitting carrier at 28 dBm and its ground planes distribute the heat generated by the PA to limit the PA surface temperature to $65°C$. Fig. 3.9(b) shows the fabricated TX unit.

Server. One of the RX units also handles the server tasks. We use the python socket.io protocol [64] to handle the communication between the RX units and server.

Data Collection Scheduling. The data collection process from each sensor initiates when the sensor node marks the availability of new data or when a pre-defined timeout is reached. The data collection might take one or several iterations. At each iteration, the RX unit first checks all the ongoing data collection processes for completion. If a portion of the data is missing, the RX unit generates the commands to read the missing parts and stores



(a) RX Unit (b) TX Unit

(c) Sensor node (d) Camera (e) Microphone

Figure 3.9: **Prototype hardware.**

them in a queue. Next, all the commands in the queue are executed, and the responses to each command are processed. Once the queue is empty, the iteration is completed. We use time-division multiple access to accommodate multiple sensor nodes. Each sensor node has a unique ID, and only one sensor node responds to the command from the RX unit at each cycle.

Cost Analysis. We design MultiScatter intending to reduce the infrastructure cost of a wide area backscatter network. Our analysis using pricing data from Octopart shows that at low volumes of 1,000 units, the TX and RX base units cost $16.1 and $49.6 (including $35 for the RPi), respectively. The number of base units required to cover a specific building depends on its floor plan. We provide two sample cases in Sec. 3.6.

3.5.2 Sensor Node

We select STM32L071C8, an ultra-low-power ARM Cortex-M0+ MCU with 20 KBytes RAM, as the sensor node MCU. The memory capacity is enough to store one image frame or one second of audio recording. A 4 MHz external crystal generates the MCU clock. We use AS3933 wake-up radio for downlink, and LT6990 VCO for generating the backscatter uplink signal. The sensor node is also equipped with HDC2080 humidity and temperature sensor, and opt3002 ambient light sensor. The sensor node is powered up using a 2.5 V regulator. We set the MCU core voltage to 1.2 V to reduce its power consumption. The antenna is shared between the wake-up-receiver downlink and backscatter uplink via the AS213-92LF RF switch, as shown in Fig. 3.8. A 47pF capacitor is chosen for $Z1$ (see Fig. 3.8), which has a small impedance at the carrier frequency.

The sensor node supports four uplink data rates, 31.25 Kbps, 62.5 Kbps, 125 Kbps, and 250 Kbps. We use 1 MHz subcarrier modulation with 25 KHz frequency deviation for the two lower data rates, and 1.79 MHz subcarrier modulation with 80 KHz frequency deviation for the two higher data rates, by setting the values of R1, R2 and R3 to 26.7 $K\Omega$, 22.1 $K\Omega$ and 2.5 $K\Omega$, respectively.

In each communication cycle, the sensor node receives a one-byte command from the base units. Some commands invoke an action, such as taking a picture, while others request specific information, such as a portion of the recorded image. The sensor includes the received command in its response to the base units for verification.

Energy Harvester. We design the energy harvesting unit with the BQ25570 IC. We use AM-1801 and AM-1816CA solar panels, designed for indoor operation with areas of 2 in^2 and 17 in^2 respectively, to supply the energy harvester board. The energy harvested from the solar cells is stored on a 100 mF super-capacitor to support sensors such as a camera and a microphone.

Camera. We use a Himax HM01B0 image sensor running in 120×120 *pixel* QQVGA mode as the main sensor, and a Panasonic EKMB1104111 motion sensor as the auxiliary

gating sensor. Once the motion sensor is activated, or a take-picture command from the base unit is received, the MCU turns on the image sensor and enables its internal Phase-Locked Loop (PLL) to generate a 3 MHz clock for the image sensor. Once an image is captured, we turn off the motion sensor to avoid image overwrite and inform the base unit that an image is available. For data transmission, we divide one image into 12 large sections or 120 small sections, and the base unit can request a large or small section with a dedicated command to complete reading the image. Once the base unit successfully receives the image, it re-activates the motion sensor.

Microphone. Our microphone board is designed with a VM1010 microphone sensor. The microphone has two operational modes, wake-up sound mode, and normal mode. In wake-up sound mode, which we use to gate the normal mode, it waits for an acoustic event to trigger the normal mode. The microphone records data for 1 second at 8 KS/s in normal mode. Transition between gating and normal modes and data transmission are similar to the camera.

Size and Cost Analysis. While we use commercial-off-the-shelf components to build our sensor node, recent works on integrated energy harvesters [65, 66], wake up radios [67, 68], and backscatter radios [69] clearly show the feasibility of integrating all electronics sections of our backscatter sensor node in an ASIC to significantly reduce its size and cost. In this case, the sensor node cost and size would be limited by the energy harvesting (i.e., solar cell) and the energy storage (i.e., supercapacitor) elements. In this work, we use a 100 mF supercapacitor to support sensors such as cameras and microphones. However, a 3.2 $mm \times 2.5$ mm 47μF capacitor stores enough energy to read the low-power sensors such as temperature and humidity sensors and transmit the data via a backscatter packet 3 times. Furthermore, recent advances in solar cell design [70] have shown 4.5 $\times$ energy harvesting efficiency under indoor lighting conditions compared to the solar cells used in this work, which means a 0.45 in^2 solar panel would generate enough energy to power up our sensor node.

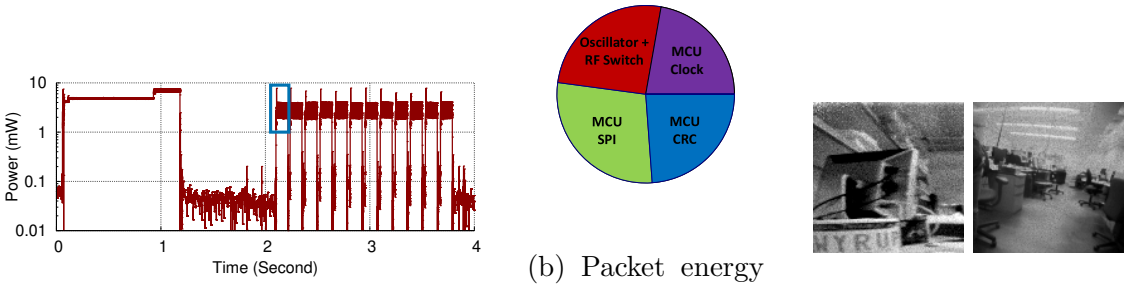
Energy Efficiency. The sensor node consumes 2.26 mW while transmitting backscatter packets, which is comparable to the commercial Sub-GHz active radios. However, we should note that this prototype is built with COTS components, while the commercial active radios are specifically optimized for low-power operation. Our evaluation in section 3.6 shows that the MCU consumes 74% of the energy in backscatter mode, which highlights the vast potential for improving energy efficiency by designing backscatter ASICs.

3.6 Evaluation

We perform several experiments to demonstrate the performance of the MultiScatter system. First, we measure the power consumption of the sensor node with the camera module and evaluate the maximum sensor update rates using two solar cells with different sizes. Next, we evaluate the communication range of our system in line-of-sight and demonstrate the scalability of our solution in covering a two-floor educational complex. We verify the accuracy of our planning tool by comparing its outputs with measurements taken in an apartment unit. Also, we examine how deploying multiple RX and TX units affects the achievable aggregate throughput of the system. Finally, we show the handover performance when sensor nodes move in the network. We use omnidirectional whip antennas for all devices in our evaluations.

3.6.1 Power Consumption

We evaluate the sensor node power consumption using the camera module since it is the most power-hungry sensor. Our setup consists of one RX-TX pair and a sensor node operating at 250 Kbps. Fig. 3.10(a) shows the measured power that the camera consumes while capturing and transmitting one image. The operation of the camera starts in the idle mode, where it consumes 39 μW on average. At time = 0.06 second, the sensor node is activated to take a picture and transfer it to the MCU. This process takes 1.2 second, and the average power is 5.04 mW. MCU PLL operation at the start and end of the process causes peaks in the power plot. Next, the RX unit requests transmission of the first section of the image, marked with the blue rectangle in Fig. 3.10(a), at time = 2.1 second. This is followed by requests for



(a) Camera power for a frame

(b) Packet energy breakdown

(c) Sample images

Figure 3.10: Battery-free camera evaluation results.

transmission of the other 11 sections to complete the picture. The image transmission takes 1.69 second, and the average power during this time is 2.26 mW.

Fig. 3.10(b) breaks down the energy that the sensor node consumes in transmitting a 38 byte backscatter packet. The MCU core consumes 23.8%, and 28.3% of the total energy (4.21 μJ) during CRC calculation and data transfer, while the MCU clock accounts for another 22.2%. The LTC6990 oscillator plus the RF switch consume the remaining 25.7%. In our prototype, the MCU consumes three-quarters of the total energy in backscatter mode, highlighting the vast efficiency potentials in designing custom ASICs for backscatter sensor nodes.

Fig. 3.10(c) shows sample images captured by our battery-free camera.

3.6.2 Energy Harvesting

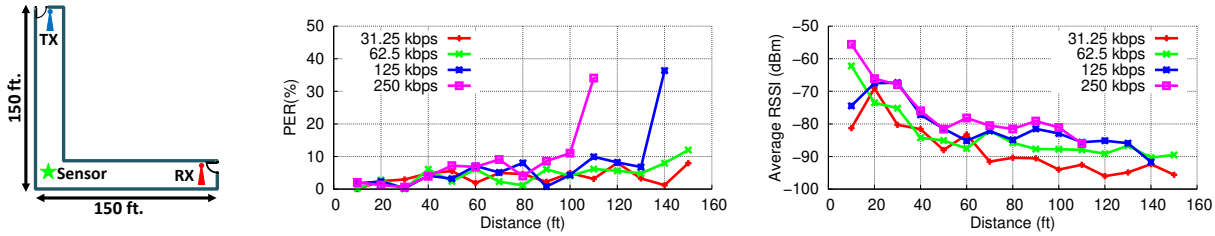
We use AM-1801 and AM-1816CA solar panels to supply the energy harvester board and evaluate the battery-free sensor nodes update rates. We set up the solar-powered sensors in an office building with fluorescent ceiling lights. The illuminance at the surface of the solar cell is 300 lux. We record data for 6 hours using each solar panel and sensor. Table 3.1 lists the achieved average update rates using each solar panel, as well as the idle power consumption, sensing energy, and communication energy for each sensor. Our evaluation

Table 3.1: **Power consumption measurements for camera, microphone, and environmental sensors.**

Measurement		Cam.	Mic.	Env.
Idle Power (μ W)		39	51	25
Sensing Energy (mJ)		6.05	3.51	0.025
Communication Energy (mJ)		3.82	2.84	0.014
Solar-Powered	2 $in^2 panel$	280	268	1
Update Rate (Sec.)	17 $in^2 panel$	17	11	0.2

shows that the sensing energy is 62%, 56%, and 65% of the total consumed energy for the camera, microphone, and environmental sensors, respectively, which shows that the sensing energy is the most power-consuming part of the sensor and limits the sensor nodes update rate.

The 2 in^2 solar cell can provide enough power for the operation of the camera and microphone sensors every 280 and 268 seconds. This is made possible by reducing the sensor node power consumption in idle mode using the gating sensors. We can increase the camera and microphone update rate to 17 and 11 seconds by using the larger solar panel. The sensor node 100 mF super-capacitor stores enough energy to transmit 21 images or 36 seconds of audio with a single charge (3.6 v-4.5 v). The small solar cell supplies enough energy to read the environmental sensors, which include the temperature, humidity, and illuminance sensors, every second. The system in [71] uses the CC2650 wireless MCU to read a temperature and humidity sensor and transmit the result via a BLE packet. The evaluation in [71] shows that the setup consumes an average of 4.04 mA from a 3 v battery over 30 ms which results in a total energy consummation of 363.3 μJ per measurement transfer. This energy includes the energy required to wake the MCU from the low-power mode, communicate with the sensor, activate the radio, and transmit the data. In comparison, our sensor node consumes 39 μJ per measurement transfer, which is 9 $\times$ lower than the BLE packet transfer implemented in



(a) Experiment setup. (b) PER vs Distance. (c) RSSI vs Distance.

Figure 3.11: **Line-of-sight communication range.**

[71].

3.6.3 Range

We set up the system in a 300×5 ft hallway with floor plan shown in Fig. 3.11(a) to evaluate the communication range of the MultiScatter system in line-of-sight. The sensor node is in the middle of the hallway (indicated with the green star). In each measurement, we place the RX and TX units at the same distance from the sensor node and measure the Packet Error Rate (PER) of backscatter communication over 1000 packets. We place the sensor node at the same distance from TX and RX units since it is the most challenging setup in backscatter deployments [45, 72]. If the sensor is placed closer to one base unit, the other base unit could be placed further away while maintaining the same PER. Fig. 3.11(b) shows the PER for four data rates as we increase the RX and TX distances from the sensor node, from 10 ft to 150 ft. Our results show that the MultiScatter system can operate up to 150 ft (length of the hallway) at the data rate of 31.25 Kbps (PER $< 10\%$) with an average RSSI of -95 dBm. The maximum achievable range reduces to 100 ft when the sensor node transmits data at 250 Kbps with the RSSI of -80 dBm. Fig. 3.11(c) shows that the RSSI values at 10 ft for 31.25 Kbps and 62.5 Kbps data rates are less than the next several measuring points, although the distance is increasing. This happens due to the carrier power tuning in the MultiScatter system, as explained in Sec. 3.3.1. At this short distance between the devices, the backscatter link achieves low PER with lower carrier power.

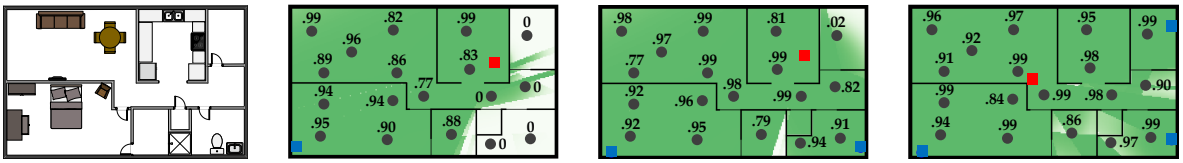
(a) Ground Floor. (b) First Floor. (c) RSSI.

Figure 3.12: **Backscatter coverage in a two-floor educational complex. RX units, TX units and tested sensor node spots are shown with red squares, blue squares and green circles, respectively.**

3.6.4 Coverage

To evaluate the performance of MultiScatter in a non-line-of-sight scenario, we set up the system in a two-floor educational complex, covering multiple rooms separated by glass and wood doors, as well as concrete walls, with a total area of 23400 ft^2. Fig. 3.12(a) and (b) show the floor plan of the ground and first floor of the building. In our setup, we have 5 RX units (red squares) placed at the height of 3 ft from the floor, and 20 TX units (blue squares) attached to the concrete walls at 6 ft height. This combination of base units costs $569 at low volumes based on our cost analysis (see Sec. 3.5). We place the sensor node at 5 ft height and move it between 130 test points (green circles), while it communicates with the base units at 125 Kbps. The sensor node transmits 1000 backscatter packets at each test point, and we measure the PER and RSSI of the backscatter packets. Our results show that all test points have PER less than 15 %, and the median of RSSI is -77 dBm (Fig. 3.12(c)), which demonstrates that our system is fully capable of covering the entire building.

Also, we evaluate the performance of MultiScatter in a 34.5 $ft \times 21$ ft single bedroom apartment (Fig. 3.13(a)). We use our deployment planning tool to determine the location of base units. To verify our tool model, we repeat the experiment using one RX unit and one to three TX units. In each test, we move the sensor node to 18 points around the apartment



(a) Floor plan (b) 1 RX - 1 TX unit (c) 1 RX - 2 TX units (d) 1 RX - 3 TX units

Figure 3.13: **Verifying the planning tool in a single bedroom apartment coverage. The simulated coverage map, shown in green in three different cases, is a close estimate of the measured PER results at 18 different points around the apartment.**

and calculate the PER over 500 packets. Fig. 3.13(b)-(d) show the simulated backscatter coverage using a soft threshold coverage function, where the darker points have a higher chance for successful backscatter coverage. The measured PER at each of the selected points is also specified in the figures. Our experiment results verify that our planning tool is a close estimate of the measured coverage.

3.6.5 Throughput

We deploy 5 RX, 20 TX, and 10 sensor node units operating at 125 Kbps in the building shown in Fig. 3.12 to evaluate the performance of MultiScatter in a multi-RX multi-TX scenario. The RX and TX units are placed as shown in Fig. 3.12(a) and (b), while the sensor node location is shown in the same figures with numbers 1 to 10. In this experiment, we assign two sensor nodes to each RX unit, select a subset of the sensor nodes with 1 to 5 members, command them to transmit data to their assigned RX units, and measure the achieved data rate at each RX unit. To calculate the aggregate throughput, we add up the individual throughput of all RX units involved. Each point in Fig. 3.14 shows the results of one measurement, where the numbers in the parenthesis indicate the location of the sensor nodes involved in the measurement.

Our results in a single RX setup show that the maximum achievable throughput using

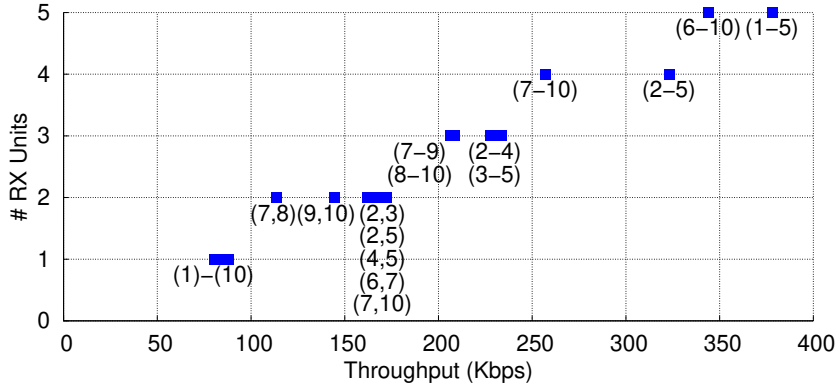


Figure 3.14: **Aggregate throughput with multiple RX units.**

one sensor node is 90 Kbps. All sensor nodes achieve a throughput close to this nominal data rate when they operate alone. Comparing the nominal data rate to the physical data rate of 125 Kbps shows a 28 % overhead. Our analysis shows that 10 % of this overhead is caused by the TX activation and sensor wake up in the communication cycle. The other 18 % is due to wasted time interval between backscatter packets.

The aggregate throughput of the network also increases linearly as the number of RX units and sensor nodes in the measurement increases. The measurement with sensor nodes placed at locations (1-5) achieves an aggregate throughput of 375 Kbps using 5 RX, which is 4.17 $\times$ faster than a single sensor node. This result shows that multiple backscatter sensor nodes can operate simultaneously and share the spectrum resources among themselves. In cases where two close sensor nodes communicate with two different RX units, such as (7,8) and (9,10), we observe a lower aggregate throughput compared to the cases where sensor nodes are placed further away from each other, such as (2,3) or (7,10). Our analysis shows that this is mainly due to interference in waking up the sensor nodes since the sensor node wake-up radio uses amplitude modulation, which is more susceptible to interference. The second factor affecting the throughput is the interference caused by the carrier signals from other backscatter links. As the number of RX units increases, the number of simultaneous carrier signals increases too which contributes to data loss, especially where the sensors from

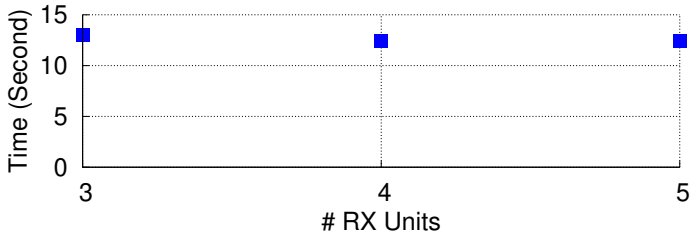


Figure 3.15: **Handover duration.**

multiple RX units are deployed near each other.

Although we test with a maximum of ten sensor nodes due to the limited hardware we have at our disposal, there is no strict limitation on the number of sensor nodes in the proposed network stack. We use 16-bit IDs for the sensor nodes in this work which limits the number of sensor nodes to 65535, but that limitation can be relaxed by using 32-bit IDs. In the current implementation, each backscatter cycle takes a minimum of 10ms. Therefore, each RX unit can communicate with up to 100 sensor nodes each second.

3.6.6 Handover

To demonstrate that MultiScatter can manage the movement of sensor nodes between RX units, we set up the system in an office building with three isolated rooms, walk within these rooms at an average pace while carrying the sensor node. We repeat the experiment 10 times for each scenario where the system uses 3, 4, and 5 RX units and 2 TX units for each RX unit. We use a threshold of 20 consecutive dropped packets in all three measurements.

Our results in Fig. 3.15 show that the average time difference between the last successfully received packet at the previous location and the first one at the new location is 13, 12.4, and 12.4 seconds when our setup has 3, 4, and 5 RX units, respectively. This time includes both the relocating time, 8.2 seconds on average, and assigning a new RX unit to the sensor node placed in the new location. This result suggests that the RX assignment process is not significant compared to the movement time in our experiments, and once the sensor node is settled in its new location, the network successfully reassigns it to a new RX unit.

3.7 Related Work

Our work is related to prior efforts in backscatter communication, and battery-free sensing.

Backscatter Communication. Our work builds on recent efforts in advancing bistatic backscatter solutions to deliver ultra-low-power wireless connectivity. These efforts were focused on designing backscatter solutions compatible with existing wireless standards [11, 12, 18, 37, 38, 52, 73–78], and improving backscatter throughput [17, 79–81], range [18, 48], reliability [37, 82] and deployability [49, 83]. While these works use a single RX-TX pair in their deployments, we use multiple RX and TX base units to realize wide area and scalable bistatic backscatter.

A wireless protocol to handle the concurrent transmission of backscatter devices was presented in [17], which uses chirp spread spectrum modulation to share the channel bandwidth between many devices communicating with a single monostatic backscatter access point. The access point can cover a wide area since each backscatter node has a low data rate that allows the backscattered signals to be recovered at very low signal powers. Using a single access point is not a scalable solution. In this work, we use a network of backscatter base units to extend the coverage at data rates up to 250 Kbps. Although we use 802.15.4g standard with FSK modulation and time-division multiplexing in this work, the techniques presented can be combined with other modulation and multiple access methods.

A scalable backscatter sensor mesh was introduced in [84] that uses distributed excitation to enable multi-hop backscatter between sensor nodes. Although this solution enables the sensor nodes to communicate with each other at longer distances, it requires an excitation source close to each sensor node resulting in significant deployment limitations. In our architecture, the sensor nodes only communicate with the base units. The network of base units handles the tasks that allow the sensor nodes to move freely inside the coverage area without adding any complexity to the sensor nodes.

Battery-free sensing. Battery-free solutions rely on the harvested energy from RF [3, 20, 28, 46, 85], solar [86–88], vibration [89, 90], motion [91], thermal [87] and magnetic [92]

energy sources for their operation. Researchers in [93] list mean energy generated by 50 activities in residential buildings. Applications such as video [21, 46, 94] and audio [20] streaming, gaming [88], full-body temperature mapping [92], food quality monitoring [95], pulse oximetry [96,97], structural health monitoring [34], flying insects EMG telemetry [98], indoor acoustic localization [99], eye-tracking [100], visible light sensing [86] and many others [91, 101–104] have been realized with battery-free platforms. In this work, we develop a modular battery-free sensing platform based on bistatic backscatter wireless communication, and use camera, microphone and environmental sensors as examples to show the platform capabilities.

3.8 Conclusion

We present the first wide-area multistatic backscatter network for battery-free wireless sensors. We discuss the limitations of backscatter systems, such as the limited coverage area, frequency-dependent operability, and sensor node limitations in handling network tasks. Our MAC layer implementation is designed to overcome these limitations by selecting the best RX-TX pair, carrier power, and frequency in communicating with each sensor to maximize the throughput and handling the handover between base units as the sensor moves inside the coverage area. We build low-cost hardware using COTS components and prototype proof-of-concept battery-free sensors for capturing images, audio, and environmental data.

This work is a first step toward, and a sample implementation of wide-area low-cost backscatter networks. There are many opportunities to improve the system in future work. Techniques such as joint sensing and communication, coherent transmission of carrier signals and detection of backscatter packets, localizing and tracking the movements of sensor nodes for faster hand over, and many more can be employed to further extend the throughput, coverage, and reliability, and reduce the energy consumption and latency of wide area multistatic backscatter networks.

Chapter 4

BATTERY-FREE CAMERA OCCUPANCY DETECTION SYSTEM

4.1 Introduction

Recent advances in sensors, wireless networks, and computer vision have enabled us to design smart devices that sense their surroundings and interact with humans and other intelligent objects. For instance, recent efforts in the development of smart surveillance systems have improved the accuracy of automated security systems wherein smart sensors are used to detect human presence. However, deployment of these automated systems is limited to scenarios where plug-in sensors communicate with a powerful processor for occupancy detection. An ideal human occupancy detection system must satisfy two major requirements: first, the wireless sensors should be maintenance-free and do not require battery charging or replacement. Second, a low-cost processor should be able to detect human presence by using a fast and accurate algorithm.

Outdoor occupancy detection systems typically use high-resolution solar-powered cameras for human presence detection. However, these systems might require battery charging if the harvested energy could not support the operation of the camera [105]. Implementation of solar-powered cameras is even more challenging in indoor cases as the solar panels harvest less energy indoor. To address this challenge, researchers have studied other sensors such as carbon dioxide, temperature and humidity [106, 107], and passive infrared (PIR) sensors [108, 109] to replace the camera with a low-power sensor. However, these systems still need batteries for operation. Furthermore, environmental data is susceptible to abrupt pattern changes due to external factors such as an opened window or door and is generally restricted to an enclosed environment.

Image data, on the other hand, can provide a clear visual indication of a human presence. However, due to the limited computational resources on an embedded device, human detection commonly relies on shallow models such as Support Vector Machine [110–112] to perform real-time detection on images. Object detection and image classification models such as You Only Look Once (YOLO) networks [113–116], Single Shot MultiBox Detector (SSD) [117], Few-shot models [118, 119] and RetinaNet [120] are generally too computation-intensive to perform real-time detection on an embedded system without GPU. In many cases, to perform inference or detection using a deep learning model, we require a computer with considerable computation capacity or an expensive device such as Jetson TX2 [116] to be able to handle the streaming images captured by the cameras. These examples show a contradicting trade-off between computation resources and detection performance. On the one hand, we want to reap the benefit of high detection accuracy from deep learning models. Still, ideally, on the other hand, we want to achieve that without expending excessive computation resources.

In this work, we design an occupancy detection system that uses a **lightweight yet accurate algorithm** implemented on a **low-cost processor** to analyze data from **battery-free cameras** and detect human presence. In this system, cameras capture low-resolution images and transmit them to a hub equipped with a low-cost processor for occupancy detection. We use a low-power communication system called backscatter to send data to the hub in the implementation of battery-free cameras. Since the required energy for data transmission is low, a solar panel in an indoor setup can harvest enough energy from ambient light to support the camera's operation.

The **contributions** of this paper are listed below:

1. Implementation of wireless battery-free cameras that harvest energy from ambient light.

2. Deployment of occupancy detection YOLOv5 model on an embedded system.

3. Demonstration of inferencing speed, detection accuracy, and reliability of the proposed system.

4.2 Related work

We review the prior efforts in battery-free sensing and detection algorithm in this section.

Battery-free sensing. To enable battery-free sensing, two conditions must be satisfied; first, operation of the sensor node, sensing and communicating, should be power efficient, and second, a harvester should provide enough energy to support this power-efficient operation. Backscatter communication technology has addressed the first challenge and has been used to design and implement battery-free sensors. In [28], a wireless sensing platform is developed using RFID readers. The reader in RFID technology uses backscatter communication to receive data from the battery-free sensors in the network. Researchers in [3,26,27], introduce a battery-free camera that captures still images and transmits them via RFID backscatter communication. Since the required energy for the operation of this camera is provided by harvesting energy from the RFID reader, the operating range is only limited to a few meters.

In [21, 46], a wireless battery-free camera is designed. This system uses an analog image sensor to stream video. The authors use Pulse Width Modulation (PWM) to convert the analog pixels value to digital for backscatter data transmission.

As mentioned, energy harvesting is a major obstacle in implementing battery-free sensors. To address this challenge, researchers have been trying to improve the efficiency of exiting harvesters and design novel boards capable of harvesting energy from multiple sources. In [46], a dual harvester board provides power for the camera's operation by collecting energy from ambient light and RF signals simultaneously, allowing the camera to operate at longer distances. Researchers in [88] design the first battery-free personal gaming device powered by collected energy from sunlight and the user actions. Other sources of energy are used to power battery-free sensors such as thermal [87], vibration [90], and motion [91].

Detection algorithm. In the realm of object detection, or specifically, in human detection,

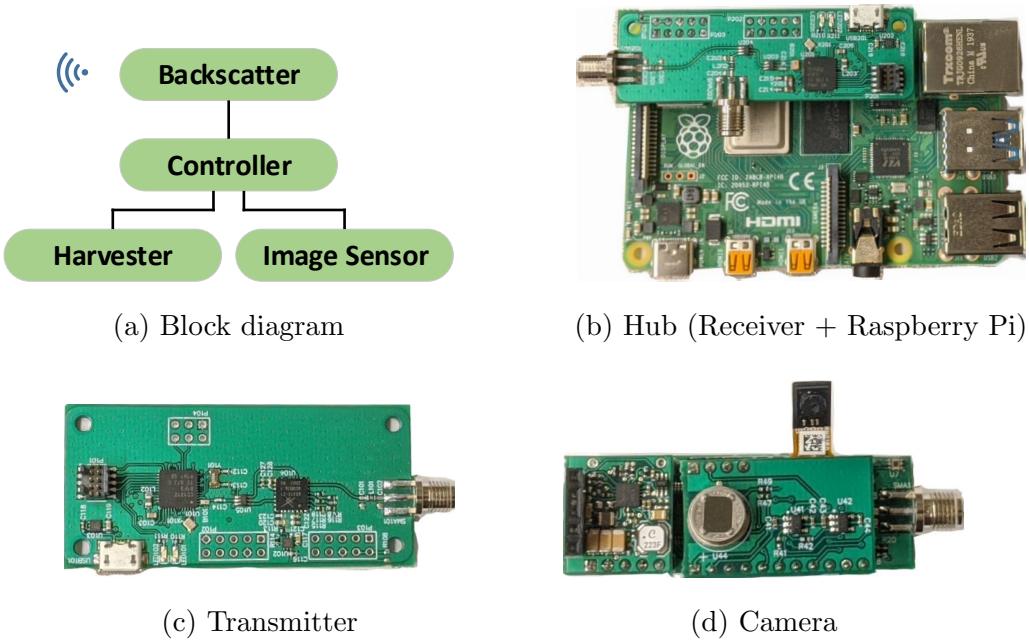
many techniques and algorithms are proposed in previous literature. For example, the combination of feature extraction using Histogram of Gradient (HOG) and the Support Vector Machine (SVM) as the classifier [110–112] are frequently used. In these papers, HOG is used to effectively extract the human feature and contour from an input image, and these extracted features are then fed into an SVM for classification. On top of that, with the small model size and inferencing speed, it becomes a suitable candidate to be deployed on an embedded system to perform real-time detection.

In [110,111], the detection is performed on relatively more straightforward human posture like standing or walking, and the whole human body is perfectly visible. In [112], the test cases presented in the paper are slightly more challenging. The images are captured in a certain angle of depression, and the humans' figures appear smaller in the images. However, the performance of the detection algorithm under occlusion cases is still not discussed in the papers.

On the other hand, with the recent advances in deep learning, there are also papers that use deep learning models such as YOLO [114–116] to detect person for occupant density detection, human behavior, and motion analysis. In [115], the authors claim that the YOLOv3 model is performing better in cases of partial occlusion. In terms of inferencing speed, however, in [114], the authors mention that the algorithm would require a GPU to perform inferencing in a real-time fashion. An example of this is shown in [115], whereby using a Jetson TX2 and TensorRT optimization, the system can achieve 10 FPS of processing speed. While deep learning models offer a more robust detection performance, deploying a deep learning model on an embedded system for real-time detection remains a challenge.

4.3 Sensor design

Our goal is to design a battery-free camera that captures low-resolution images and transmits them to a hub via wireless communication. To enable this, we use a low-power microcontroller to read images from the image sensor and send them to the receiver using backscatter communication technology. Our solar harvester stores enough energy to support the oper-



(a) Block diagram

(b) Hub (Receiver + Raspberry Pi)

(c) Transmitter

(d) Camera

Figure 4.1: **Prototype hardware.**

ation of the camera. Figure 4.1 shows the block diagram of our system and our camera prototype.

4.3.1 Image Sensor

We use a Himax HM01B0 low-power image sensor that outputs gray-scale video with a bit depth of 8 per pixel to capture 120 × 120 pictures. This image sensor has the lowest power consumption among the low-resolution sensors since it consumes only 1.1 mW to stream video at 30 frames per second. The process of capturing an image starts with enabling the image sensor. Next, the microcontroller unit (MCU) reads the image row by row and saves it in memory. Once the picture is recorded, the MCU disables the image sensor to keep the camera's power consumption in sleep mode low.

To improve the image quality, we remove the static noise in the images. A white paper is placed in front of the camera's lens, and hundreds of images are recorded to estimate this noise. Taking an average across these images provides an estimate of the static noise. We

Figure 4.2: **Image sample after (left) and before (right) enhancement.**

remove this offset by subtracting the per-pixel average from each pixel of the image that we record. Figure 4.2 shows a sample image before and after the noise removal process.

4.3.2 Microcontroller and Energy Harvester

We use STM32L071C8, an ultra-low-power MCU with a clock frequency of 4 MHz, to initialize the image sensor, receive images from it, and transmit them in multiple packets to the hub using backscatter communication. In the following paragraphs, we explain the process of capturing an image and transmitting it to the receiver in more details.

To save energy, we keep our system in sleep mode and wait for the hub to request an image. Our camera is equipped with the low-power Amplitude Shift Keying (ASK) receiver used in [37] to enable this down-link communication. The ASK receiver listens to the commands from the hub and wakes the microcontroller from sleep mode upon detection of the image request command.

Once the command is received, the microcontroller's Phase-Locked Loop (PLL) generates a 3 MHz clock signal for the operation of the image sensor and initializes it via the I2C bus. Next, the image sensor captures an image and uses the SPI bus to transfer each row's pixels value to the MCU. This process is finished when the MCU reads all of the 120 rows. Finally, to transmit the image to the hub, the MCU breaks down the image into 12 portions or 120 rows. The hub can request transmission of a portion or row by sending the dedicated command to the camera.

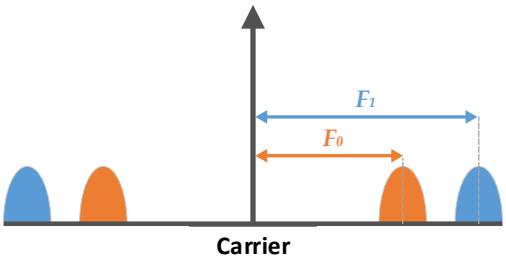


Figure 4.3: **FSK backscatter in frequency domain.**

The MCU uses the backscatter block for image data transmission. We stream the raw data to the hub and do not perform any compression on the camera board because the power and time required for compression are higher than the needed time and energy to stream raw data [121]. Moreover, compression becomes more critical where in contrast to our system, the communication cost is high, so the data size must be as small as possible. To make our communication system more robust, the microcontroller computes the Cyclic Redundancy Check (CRC) of each communication packet and transmits it at the end of the packet. If this CRC does not match the CRC that the hub calculates, the hub requests re-transmission of the packet.

We use a TI BQ25570 solar energy harvesting Integrated Circuit (IC) to power the camera's operation. The amount of energy that this IC stores in the supercapacitor depends on the size of the solar panel and light intensity in the experiment room; using a larger solar panel or deployment in a room with higher light intensity results in a higher update rate.

4.3.3 Backscatter Communication

Backscatter communication is a low-power wireless technology wherein a transmitter unit generates a carrier signal, a sensor node reflects the carrier, and a receiver unit decodes the sensor node data [21, 23, 45, 49]. Since the sensor does not need to generate any signal to communicate with the receiver and only reflects the carrier, the power consumption for data transmission is low.

We implement a Frequency Shift Keying (FSK) backscatter unit to transmit the images to the hub at the data rate of 125 Kbps. In our design, the microcontroller changes the state of the RF switch connected to the antenna at two frequency rates to reflect the carrier. Figure 4.3 shows the carrier signal and bit 1, and bit 0 with frequency offset of f_1 and f_0 from the carrier, respectively. Our communication system allows the camera to communicate with the receiver while there is up to 130 ft distance between them [122]. Our custom-designed receiver board equipped with a Raspberry Pi 4 for data processing and the transmitter prototype are shown in Figure 4.1.

4.4 Algorithm

This section discusses the algorithm that we use to perform occupancy detection on the embedded system.

4.4.1 YOLOv5 Network

To detect human presence, we implement a recent state-of-the-art detection algorithm, YOLOv5 [113]. Similar to most single-stage object detection models [117, 120], YOLOv5 consists of three components, each with an essential role in the model.

The first component of the model, known as the model backbone, is a feature extractor network that extracts rich and useful features from the input images. Many model backbones are developed over the years, such as Cross Stage Partial Network (CSPNet) [123] and Residual Network [124]. In YOLOv5, the former is used as the model backbone. The next component is the model neck, whose primary function is to generate feature pyramids. Feature pyramids are an essential block that allows the model to detect objects at different sizes and scales by constructing multi-scale feature maps. Similar to YOLOv4, YOLOv5 uses the Path Aggregation Network (PANet) [125] which combines the top-down and bottom structure as the model neck. The last component of the model is the model head, which generates the predictions for bounding boxes and object classes.

The output tensor dimensions for YOLOv5 can be expressed as $d \times (b + n)$, where d is

the number of detection layers, $b = (x, y, w, h, c)$ representing the predicted bounding box's coordinates, width, height and the prediction confidence, and n is the number of classes. The model has three detection layers in our application, and since we are only interested in a single class, human presence detection, the output tensor dimensions are updated to $3 \times (5+1) = 18$. The models are introduced with four different sizes; small (s), medium (m), large (l), and extra-large (x), with the model sizes sorted in ascending order. The model sizes are increased by increasing the model height and width multiplier, which will increase the number of layers and layer channels. Considering the small input image size and the limited computation resources on Raspberry Pi, we choose the s model with a 0.33 height multiplier and 0.5 width multiplier for our application. The resulting model weight is merely 14 MB with 7.25 M parameters and 16.8 GFLOPs in the model.

4.4.2 Training

Collected images by the cameras are labeled and annotated with bounding boxes around the occupants. We employ several data augmentation techniques such as brightness varying, image flipping, and mosaic data augmentations to generate a richer training image variation. These data augmentations procedures are essential to enhance the robustness of the trained model to various scenarios such as different illuminance levels and human position in the images. For this single-class detection implementation, we generate up to 4,308 images after augmentations(excluding mosaic augmentation), and we use 3,016 images for training, 430 images for validation, and 862 images for testing. In the next step, these images are fed into the YOLOv5 network for model training. The model is trained on Google Colab using a Tesla T4 GPU for 300 epochs. After training the model, the model weight and architecture are saved for inferencing purposes.

4.4.3 Deployment

The model weights are first converted into Tensorflow Lite (TF-Lite) format before deploying the saved model on Raspberry Pi. As a result, we can efficiently perform model inferencing

on the Raspberry Pi using the TF-Lite Interpreter. TF-Lite Interpreter is a lightweight model interpreter library dedicated to model inferencing on various embedded systems. This essentially reduces the required memory size on the embedded system as it exempts the need to install a full-blown deep learning library to run the model.

4.5 Evaluation

We perform several experiments to demonstrate the performance of our system. First, we measure the power consumption of the camera. Second, we deploy our camera in different locations to evaluate the accuracy of the model.

4.5.1 Camera Power Consumption

As mentioned in section 4.3.2, the MCU breaks down the image data into 12 portions or 120 rows to transmit data to the hub. Figure 4.4 shows the power that the camera consumes to send one portion of an image to the hub. Each portion is composed of 60 communication packets. The peaks in the power consumption plot are caused by the MCU computing the CRC of each packet. To measure the camera's power consumption, we use a 2.2 Ω resistor in series with the camera and measure the voltage of the resistor, which is 2.2x the current the camera and the resistor consume. We analyze the voltage data that the oscilloscope records to report the power numbers. Since the resistor value is small, we can assume that the camera consumes the total current. We list the energy that the camera needs to capture and transmit an image in Table 4.1.

To evaluate the harvester's performance, we use AM-1801 and AM-1816CA solar panels with areas of 2 in^2 and 17 in^2, respectively, in an office with the light intensity of 300 lux for collecting energy. Table 4.1 lists the average update rates that we can achieve using these solar panels.

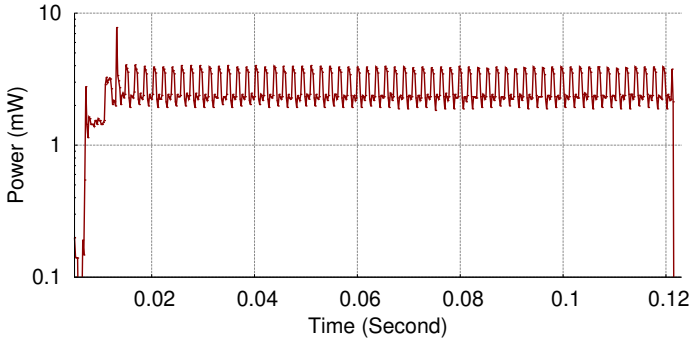
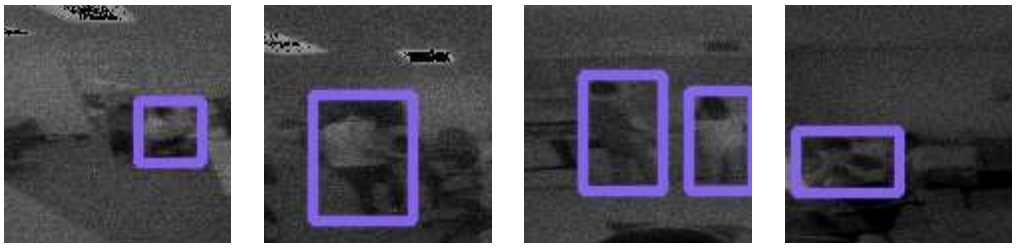


Figure 4.4: **Camera's power consumption during transmission of a portion of the image.**

Sleep Power (μ W)		30
Sensing Energy (mJ)		6.05
Communication Energy (mJ)		3.82
Solar-Powered	2 $in^2 panel$	280
Update Rate (sec.)	17 $in^2 panel$	17

Table 4.1: **Power consumption measurements for camera.**



(a) Lab (Cam. 1) (b) Lab (Cam. 2) (c) Kitchen (d) Living Room

Figure 4.5: **Sample detection results for each experiment zone.**

4.5.2 Experiment Setup

The system setup consists of two camera nodes, a transmitter, and a receiver attached to a Raspberry Pi. During the system testing, the cameras are installed in each zone to capture

images from different directions and angles. The system is tested at two indoor locations, a residential unit, and a computer lab. For privacy concerns, the system is only set up in the common areas such as the kitchen and the living room in the residential unit. These selected testing locations are chosen to provide diverse indoor testing scenarios. For example, in the kitchen, occupants are often standing and walking as they are cooking or cleaning the dishes, while in the living room, occupants are usually sitting on the couch watching television or using their laptops. These scenarios produce images with different human postures, lighting conditions, and occupants' distance to the camera (human figure size in pictures), enabling a more comprehensive system performance evaluation.

In each location, we run the system for $\sim$10 hours from sunrise to sunset, with $0 \sim 2$ occupants in the zone throughout the experiment period. Moreover, to evaluate the reliability of the system, a final experiment is performed with an extended period, where we run the system uninterrupted for up to *5 days*, with the camera nodes installed in two zones (kitchen and living room) simultaneously. This experimental setup allows the system to cover a large area in the residential unit, with up to a $\sim$30 *ft* euclidean distance between the receiver and camera sensor nodes.

4.5.3 Experiment Results

In this section, we discuss the performance of our system at locations mentioned in Section 4.5.2. Figure 4.5 shows four sample images wherein the YOLOv5 model detects occupants and annotates them with bounding boxes.

Each example image shown in the figure depicts a different zone/background or posture. For example, in Figure 4.5 (a), the image shows the person in a sitting position and far away from the camera. In contrast, another camera at the same location captures a picture of the person walking toward the door from a different angle. The model can detect the person in both scenarios despite the different posture and distance from the camera. Moreover, in Figure 4.5 (c), the image shows that our model can detect multiple people in one image. Finally, in Figure 4.5 (d), the image shows a scenario where the occupant is looking at the

laptop in the living room with only the top half of the body visible. This example shows that the model can detect the person despite the occlusion of the lower half body and it is robust to a small angle of depression.

On top of the output illustrations, Table 4.2 also provides a summary of the outcome of the performed experiments. In each zone, the camera captures and inferences up to ∼3,000 images, and in the last experiment, the system records more than 30,000 images, with an inference time of only ∼100ms per image. To simplify results presentation, we use a 0.5 confidence threshold in the inferencing, and each frame with a detected person(s), will be labeled as *occupied*, and otherwise, as *unoccupied*. After evaluating these occupied and unoccupied frames, each experiment achieves an accuracy of >90%, including the last experiment with two zones and an extended experiment period. This shows that the system can accurately detect human presence and operate reliably in a large area for a long duration.

Zones	**Total Num. Images**	**Accuracy(%)**
Lab	2,951	99.56
Kitchen	3,508	99.72
Living Room	3,082	92.53
Kitchen and Living Room (5 days)	32,914	92.72

Table 4.2: Table shows the detection accuracy rates for different testing locations and the total number of camera images collected using the developed system.

Every system has its advantages and shortcomings. In our case, since our proposed detection system is vision-based, the images captured by the camera are inherently reliant on the surrounding illuminance. Under the low light condition, images captured appear much darker with small pixel intensity values, as shown in Figure 4.6. The red circle in Figure 4.6 indicates an occupant in the frame, but the image captured does not contain enough contrast

Figure 4.6: **A sample image captured in a dark environment with the occupant circled in red.**

that shows the human figure in the image. The combination of low-resolution and dark images causes human detection to become a challenging task. This situation can potentially be remedied by equipping the detection system with simple motion detection and image enhancement algorithms. In our future work, the adaptation of this system in low light conditions will be further addressed in more detail.

4.6 Conclusion

This paper proposed a power-efficient and high-performing occupancy detection system deployed on an embedded system with a low-cost processor. Using a solar energy harvester and backscatter communication, we successfully powered the low-power camera sensor and transmitted the captured images to the receiver. The power consumption of the camera for sensing and transmitting an image was discussed in detail. For the detection algorithm, we deployed a state-of-the-art detection model, the YOLOv5 network, on the Raspberry Pi to detect occupants accurately. Multiple indoor experiments were performed to demonstrate the performance of the proposed system. The experiments showed promising results with $> 90\%$ detection accuracy and a fast inferencing time of $\sim$100ms.

Chapter 5

LOW-POWER COLOR VGA CAMERA SYSTEM

5.1 Introduction

Streaming high-quality video from low-power wireless devices across a range of applications from battery-powered and battery-free security cameras [6,46], wearable cameras (e.g., Google glasses), as well as insect-scale robots [121], is a challenging problem. State-of-the-art low-power systems can only stream 112×112 monochrome video at 13 frames per second (fps) while consuming 3.3 mW [46]. Higher-quality streaming is difficult due to the significantly larger data rates involved that consume much higher power. A popular solution is to employ dedicated codec hardware and compress the video stream to reduce the transmission data rates. While this approach reduces the energy required for communication to manageable levels, it comes at the cost of increased power consumption for data processing. For instance, a codec-based design results in the system power consumption of around 1 W for streaming 720p video at 30 fps [6]. Thus, so far, there has been a strict tradeoff between wireless video quality and system power consumption.

To enable high-quality video streaming, we need to use high-resolution image sensors, but they typically use hundreds of mW, which is beyond the power budget of IoT sensors (see Fig. 5.1). To address this issue, we propose a dual-camera system, wherein a low-power gray-scale image sensor is always active and streams video to a nearby basestation; a duty-cycled high-resolution color image sensor sparsely transmits one frame to the basestation as the reference frame. Our machine learning model implemented at the basestation will use the reference frame to upsize and colorize the low-resolution images. Fig. 5.2 shows the proposed dual camera system.

In our dual-camera system, we can duty-cycle the operation of the power-hungry sensor

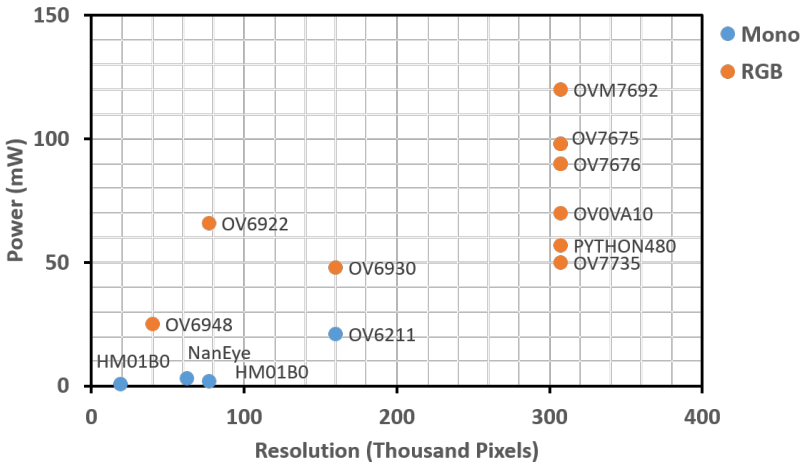


Figure 5.1: **Power consumption of RGB and monochrome image sensors vs. image size.**

and mostly rely on the data from the low-resolution sensor, which eventually are up-sized and colorized by the machine learning model using the reference frames. This design allows us to implement a low-power camera system that captures high-quality color video while consuming a few mWs. This design is low-power enough to enable many applications in computer vision, security systems, IoT, and wearable devices.

In the following sections, we focus on the hardware design of this project and show how we design and implement our dual camera system. Next, we deploy our system, capture low-resolution (160×120) video and VGA key-frames (640×480), and show that our model can generate high-quality color VGA video frames. Finally, we evaluate the system's power consumption and compare the results with the state-of-the-art camera systems.

5.2 System Design and Implementation

We aim to design a wireless low-power camera system for streaming VGA color video at 15fps. In our system, a low-power QQVGA (160×120 pixels) image sensor streams gray-scale video to the basestation with a frame rate of 15. To upsize and colorize the QQVGA frames, we use a powerful color VGA sensor that transmits one color VGA frame to the basestation per

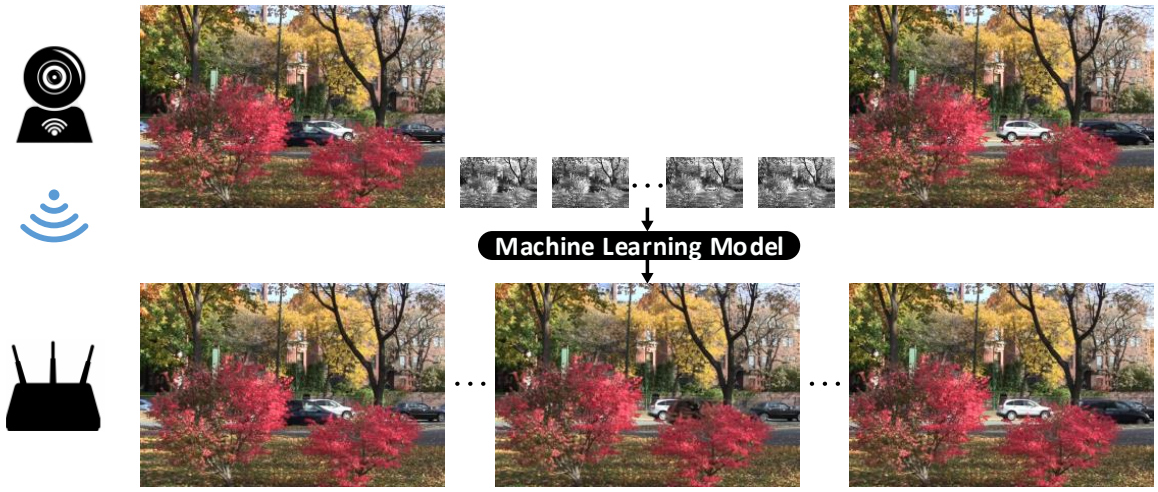


Figure 5.2: **Overview of our dual camera system. We use low-resolution gray-scale frames and reconstruct high-resolution color frames using color inferred from the periodic high-resolution key-frames.**

second. Our machine learning model implemented at the basestation receives 1 VGA and 15 QQVGA frames as inputs in one second and generates 15 frames of VGA color video. This approach allows us to keep the power-hungry sensor active for a short time (duration of a frame) and duty-cycle its operation, ultimately enabling us to build a more power-efficient VGA camera system than a single VGA image sensor camera system wherein the power-hungry image sensor is always active. Fig. 5.3 shows the block diagram of the system; two sensor nodes capture videos and transmit them to the basestation. The following subsections explain how the system is designed and implemented.

5.2.1 Low-resolution gray-scale image sensor

We use a low-power STM32L496 microcontroller to initialize a Himax HM01B0 image sensor in QQVGA mode and record 15 frames of gray-scale video in one second. Our MCU

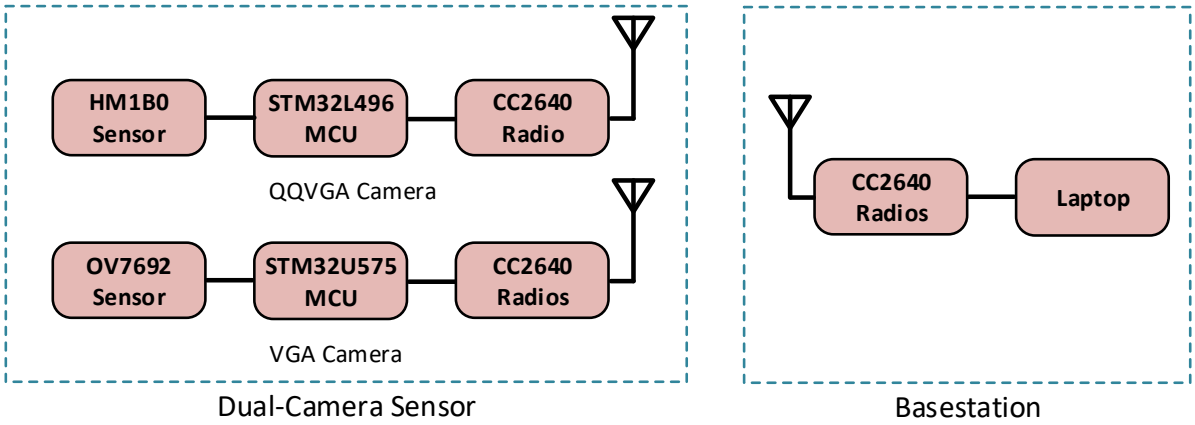


Figure 5.3: **Dual camera system block diagram.** The system comprises two key parts: 1- a basestation and 2- a dual camera sensor node; the sensor node is built with low-resolution and high-resolution image sensors.

(Microcontroller Unit), operating at 80 MHz, uses its PLL (Phase Locked Loop) module to provide a 10 MHz clock for the image sensor operation. Once the image sensor clock is ready, the MCU uses the I2C (Inter-Integrated Circuit) module to initialize the image sensor in streaming mode and send commands to the Himax camera. Next, the MCU uses the SPI module (Serial Peripheral Interface) in slave mode to receive the 8-bit pixels value from the image sensor (SPI master). A DMA (Direct Memory Access) channel is enabled to transfer the data from the sensor (peripheral) to the MCU (memory). This allows us to leave the MCU central core out of the reading process and keep it in sleep mode, which results in lower power consumption. Finally, to transfer the collected data to the basestation, we can use a TI CC2640R2F radio in 8-FSK mode with a maximum throughput of 2.5 Mbps. The radio operates in the 2.4 MHz band and generates RF signal at 0 dBm power level. Our STM32L496 MCU uses a DMA channel to transfer the video data to the TI radio's memory via a UART (Universal Asynchronous Receiver Transmitter) connection.

5.2.2 High-resolution color image sensor

We use a STM32U575 MCU plus an OmniVision OV7692 color VGA image sensor to record one key-frame per second. The MCU, operating at 160 MHz, generates a 12 MHz clock

signal for the image sensor's operation and initializes it in YUV 4:2:2 mode through an I2C connection. In this mode, the image sensor outputs 16 bits of data per pixel, resulting in a frame data size of 5 Mb, which is smaller than our MCU's RAM capacity. After the initialization process, the sensor captures a picture; the MCU uses the digital video port (DVP) interface with 8 data lines and three synchronization signals (horizontal reference, vertical reference, and pixel clock) to transfer the image. A DMA channel of the microcontroller is activated to transmit 8 bits of data on the positive edge of the pixel clock signal until a frame is complete. The central MCU core is mainly kept in sleep mode during image reading to preserve energy. To reduce the energy consumption even more, we keep the VGA image sensor active only for 40ms which is the duration of the initialization and reading processes. As mentioned above, by the time t = 40 ms, the MCU has stored one frame in memory and deactivated the image sensor. Now, the MCU has to transfer the recorded image to the radio for wireless data transmission to the basestation. As mentioned above, the data size of the high-resolution camera system is 5 Mb, while the radio's maximum throughput is 2.5 Mbps. Therefore, we can divide the data into two sections and use two separate TI radios operating at two different center frequencies to transfer the data. Our STM32U575 MCU can enable two DMA channels and two UART modules (one for each radio) to send the video data to the radios' memory. The UART transfer happens from time t = 40 ms to t= 990 ms. After data transmission, the MCU prepares the system for capturing the next frame.

5.2.3 Synchronization and Lens Alignment

In this subsection, we explain: first, how the laptop program finds the end and start of a frame, second, how we locally assign a synchronized timestamp to each frame captured by the low-resolution and high-resolution systems, and finally, how we deploy the dual-sensor system to minimize the misalignment between the lenses.

In order to recover the transferred images, the basestation must be able to find the first pixel of each frame. To enable this, we add a specific three-byte footer at the end of every frame (13, 0, and 10) and transfer it alongside the image data (see Fig. 5.4). Our python

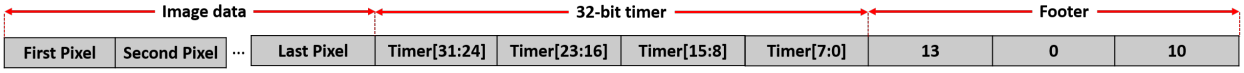


Figure 5.4: **Packet structure of the dual camera sensor node.** Each cell has a byte of data. Our sensors record the image data and add the timer value and footer to the end of the packet. Timer data and footer are used to synchronize the sensors with the laptop.

program implemented at the basestation looks for the first footer in the coming data to find the start pixel and reconstruct the images.

Our machine learning model must know when each frame is captured to sort the input frames correctly. We implement a 32-bit timer on the low and high-resolution MCUs with a precision of 1 ms. To synchronize the low-resolution and high-resolution timers, the laptop program sends a command to the MCUs to start the timer. After capturing a frame, the MCU reads the 32-bit timer value and appends it to the image data (Fig. 5.4). As a result, the basestation receives a frame with time information embedded in the data array. As mentioned, these two timers (low-resolution and high-resolution MCU timers) must be triggered simultaneously to benefit from the time information and prevent any mismatch between the values. This is enabled by the basestation sending a start signal to the MCUs.

Finally, we build a prototype for evaluation. We use this prototype to capture low-resolution, high-resolution, and ground-truth image frames and transfer them to the basestation (laptop) for data processing. The ground-truth camera helps us compare our reconstructed video with a regular VGA sensor video. To reduce the misalignment between the three lenses, we place the image sensors close to each other as much as possible. This approach helps us capture frames with a similar field of view. Fig. 5.5 shows our prototype hardware.

5.2.4 Machine learning model

Note: since I was responsible for the hardware development of this project, I focused on the implementation and testing of the prototype throughout Chapter 5. This subsection

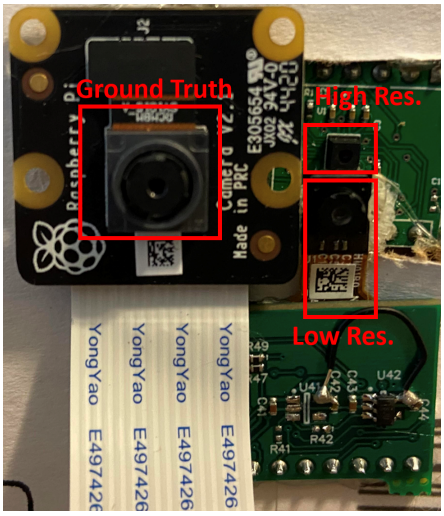


Figure 5.5: **Prototype of our dual camera system.** We use Raspberry Pi cameras for ground truth. The flex PCB cable (FPC) allows us to place the two cameras close with super-glue.

briefly reviews the machine learning model. For more information, please refer to the archive paper [126].

We propose Recurrent Key-Frame Network (RKFN) for keyframe-based super-resolution and colorization. Similar to many RNN architectures, our method is based on a recurrent element called Frame Memory. We train two versions of our model, one for the keyframe-based video super-resolution (KeyVSR) and the other for the key-frame-based video super-resolution and colorization (KeyVSR-C) task.

5.3 Evaluation

We perform multiple experiments in this section to measure the power consumption of our dual-camera system. Also, we qualitatively compare our generated frames with state-of-the-art machine learning models. For quantitative results, please refer to the archive paper [126].

Table 5.1: **Power consumption of the low-resolution camera.**

Components	Current Consumption (mA)	Supply Voltage (V)
STM32L496 MCU	4.57	1.8
Himax HM01B0 Sensor	0.97	2.8
TI CC2640R2F Radio	7.14	2.8

5.3.1 Power Analysis

In this section, we measure the average power that our system needs for capturing video and transferring the data to the basestation via a wireless connection.

Low-resolution Sensor. We use Himax HM1B0 in QQVGA mode to record video at 15fps. The image sensor consumes 2.72 mW. We mostly keep our STM32L496 MCU in sleep mode, and the average power consumption is 8.23 mW. TI CC2640R2F radio in 8-FSK mode with a maximum throughput of 2.5 Mbps and operating in the 2.4 MHz band can be used to transfer the data to the basestation via a wireless connection. The radio consumes 20 mW when it outputs a signal at 0 dBm. Therefore, the total power consumption for the wireless low-resolution camera system is 30.95 mW.

High-resolution Sensor. We use the OmniVision OV7692 color VGA sensor to capture a frame per second. Enabling the image sensor and recording data takes 40 ms, and the image sensor consumes 77.06 mJ during this period. Once the image is entirely received by the STM32U575 MCU, we turn off the image sensor to save power. Thus the average power consumption of the image sensor is 3.08 mW. The MCU's average power consumption is 32.4 mW. To transfer the key-frame to the basestation, we can use two TI CC2640R2F radios for wireless data transmission; therefore, the total required power for the transmission process is 40 mW (2×20 mW). The wireless high-resolution camera system requires 75.48 mW for its operation.

Table 5.2: **Power consumption of the high-resolution camera.**

Components	Current Consumption (mA)	Supply Voltage (V)
STM32U575 MCU	18	1.8
OV7692 Sensor	1.10	2.8
2×(TI CC2640R2F Radio)	2×(7.14)	2.8

Table 5.3: **Comparison of existing wireless video streaming systems.** Note that we report active power when the system is ON and streaming.

	Active Power	Resolution	Frame rate	Color	Comms
BackCam [127]	9.7 mW	160 × 120	1 fps	greyscale	backscatter
BeetleCam [121]	4.4-18 mW	160 × 120	1-5 fps	greyscale	radio
Analog CentEye [46]	3.3 mW	112 × 112	13 fps	greyscale	backscatter
Blink [128]	700-1800 mW	640 × 480/740p	7.5-30 fps	color	radio
Our prototype	106 mW	640 × 480	15 fps	color	radio

Power Comparison. Our system can achieve color 640×480 wireless video streaming at 15 fps while consuming only 106 mW. The active power is much lower than existing wireless video streaming systems at comparable frame rates and resolution (Table 5.3). Please note our dual-mode design, where the high-resolution camera is heavily duty-cycled, only 5.8 mW of this power is from our cameras (13x reduction), realizing our goal of making the camera sensor power negligible. The rest is consumed by the microcontrollers and radios, which we do not currently optimize in our system.

5.3.2 Video Quality Analysis

In this subsection, we qualitatively compare our results with state-of-art super-resolution methods. In the next paragraph, we use existing datasets (Vid4) to generate output frames for different models. Fig. 5.6 shows the generated frames. Next, we place our prototype (Fig. 5.5) in the lab and record video with the sensors. Our machine learning model uses the collected data to generate high-quality videos (Fig. 5.7).

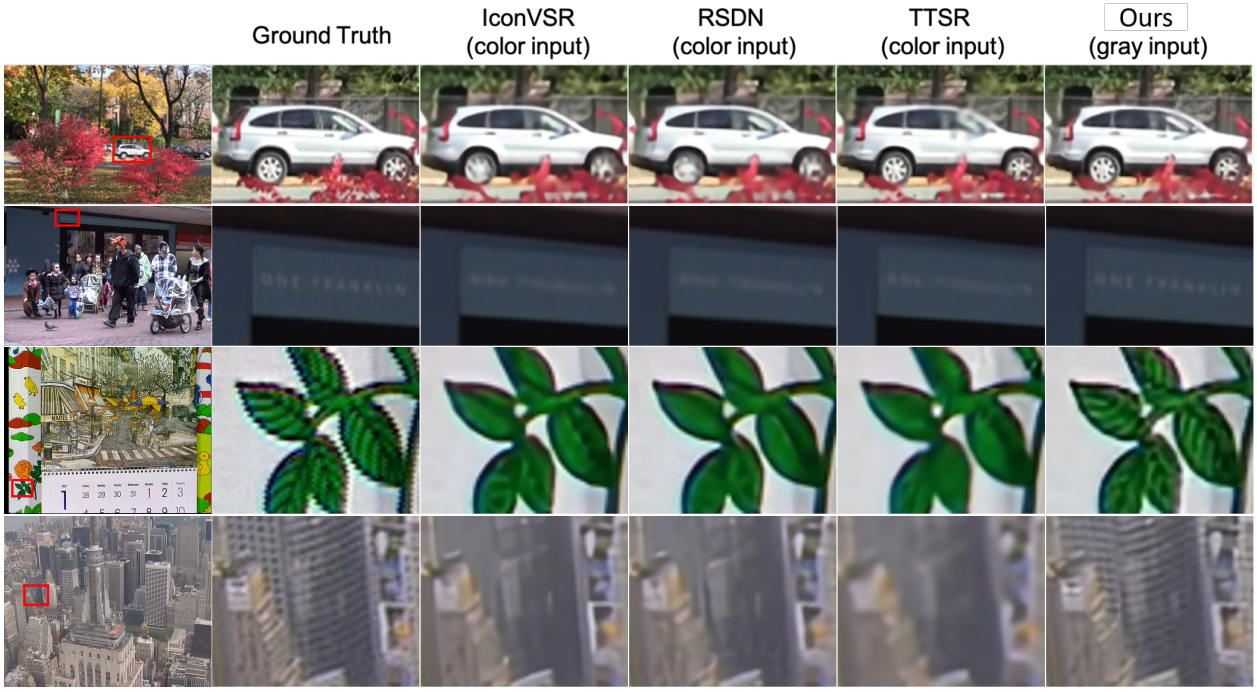


Figure 5.6: **Qualitative comparison of results with state-of-art SR methods on Vid4 dataset.**

(a) Y-channel comparison with video super-resolution method.

(b) RGB-channel comparison with Reference-based method.

Figure 5.7: **Qualitative comparison on real-world data.**

Chapter 6

CONCLUSION

In conclusion, this dissertation presents low-power wireless video streaming techniques and applications. We designed an analog video streaming camera system that captures low-resolution and gray-scale images and transfers them to the basestation via analog backscatter technology. This camera harvests energy from ambient light and RF signals for its operation. Next, we mentioned that analog sensors and analog backscatter cause multiple problems that affect the data quality and limit the applications of analog backscatter. Instead, we designed a modular digital backscatter communication system that can cover a wide area and accommodate many sensing modalities. We used a digital low-resolution sensor alongside other sensors (audio, temperature, humidity, and illuminance) to detect human presence in buildings. This was the primary motivation behind developing a low-power wireless video streaming system for low-resolution cameras. Finally, we designed a dual-camera system that consumes 106 mW to stream VGA video at 15 fps. In this design, we heavily duty-cycled the VGA sensor (power-hungry) operation and relied on the frames captured by the QQVGA image sensor (low-power). Our model can generate high-quality frames using low-quality images and occasional VGA reference frames.

The work presented in this dissertation can be followed up along these directions.

Dual-Camera sensor node MCU power optimization As mentioned in the last chapter, we focused on optimizing the system's power consumption by duty-cycling the power-hungry sensor. However, there is a significant opportunity to optimize the power that MCUs consume. It may make sense to use a single MCU in the system that can control two sensors and multiple radios.

Dual-Camera system and backscatter communication. We presented an FSK

backscatter communication system with a maximum data rate of 250 Kbps in Chapter 3. Still, we did not use this communication system in developing the dual camera design since we needed more bandwidth (higher data rate). One could improve the communication system to increase the data rate or use off-the-shelf backscatter radio ICs from Jeeva Wireless Inc. to optimize the radio's power consumption. This could help us significantly decrease the total power of communication(10-1000x reduction).

Dual-Camera system ASIC design. To build our system, we used off-the-shelf components. One could design a single chip that performs the same tasks as our system. Since this chip is designed for this specific application, it will be exceedingly power-efficient. This design can enable many applications where the form factor, battery life, and heating are crucial. For example, we can develop smart glasses that capture high-quality images while powered by a small battery.

BIBLIOGRAPHY

[1] A. Sample and J. R. Smith, "Experimental results with two wireless power transfer systems," in *Radio and Wireless Symposium, 2009. RWS'09. IEEE*, pp. 16–18, IEEE, 2009.

[2] "Tire pressure and brake temperature systems – smartstem," Accessed November 2018. http://www.craneae.com/Products/Sensing/SmartStem.aspx.

[3] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "WISPCam: A battery-free RFID camera," in *2015 IEEE International Conference on RFID (RFID)*, IEEE, April 2015.

[4] 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 the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 292–305, ACM, 2017.

[5] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl, "Energy characterization and optimization of image sensing toward continuous mobile vision," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 69–82, ACM, 2013.

[6] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith, "Towards battery-free hd video streaming," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018.

[7] "Centeye stonyman image sensor datasheet," 2017. http://www.centeye.com/products/current-cen teye-vision-chips/.

[8] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[9] J. James, V. Iyer, Y. Chukewad, S. Gollakota, and S. B. Fuller, "Liftoff of a 190 mg laser-powered aerial vehicle: The lightest wireless robot to fly,"

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

[11] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive wi-fi: Bringing low power to wi-fi transmissions," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, NSDI'16, (Berkeley, CA, USA), pp. 151–164, USENIX Association, 2016.

[12] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, (New York, NY, USA), pp. 356–369, ACM, 2016.

[13] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith, "Turbocharging ambient backscatter communication," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 619–630, 2015.

[14] S. J. Thomas and M. S. Reynolds, "A 96 mbit/sec, 15.5 pj/bit 16-qam modulator for uhf backscatter communication," in *2012 IEEE International Conference on RFID (RFID)*, pp. 185–190, IEEE, 2012.

[15] P. Zhang, P. Hu, V. Pasikanti, and D. Ganesan, "Ekhonet: High speed ultra low-power backscatter for next generation sensors," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, pp. 557–568, ACM, 2014.

[16] P. Zhang and D. Ganesan, "Enabling bit-by-bit backscatter communication in severe energy harvesting environments.," in *NSDI*, pp. 345–357, 2014.

[17] M. Hessar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*, NSDI'19, (Berkeley, CA, USA), pp. 271–283, USENIX Association, 2019.

[18] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "Lora backscatter: Enabling the vision of ubiquitous connectivity," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, pp. 1–24, Sep. 2017.

[19] V. Talla and J. R. Smith, "Hybrid analog-digital backscatter: A new approach for battery-free sensing," in *RFID (RFID), 2013 IEEE International Conference on*, pp. 74–81, IEEE, 2013.

[20] V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith, "Battery-free cellphone," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, pp. 1–20, June 2017.

[21] M. Hessar, S. Naderiparizi, Y. Wang, A. Saffari, S. Gollakota, and J. R. Smith, "Wireless video streaming for ultra-low-power cameras," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, June 2018.

[22] A. N. Parks and J. R. Smith, "Sifting through the airwaves: Efficient and scalable multiband rf harvesting," in *RFID (IEEE RFID), 2014 IEEE International Conference on*, pp. 74–81, IEEE, 2014.

[23] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: Wireless communication out of thin air," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, (New York, NY, USA), pp. 39–50, ACM, 2013.

[24] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith, "Powering the next billion devices with wi-fi," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '15, (New York, NY, USA), pp. 4:1–4:13, ACM, 2015.

[25] A. N. Parks, A. P. Sample, Y. Zhao, and J. R. Smith, "A wireless sensing platform utilizing ambient rf energy," in *Power Amplifiers for Wireless and Radio Applications (PAWR), 2013 IEEE Topical Conference on*, pp. 160–162, IEEE, 2013.

[26] S. Naderiparizi, Z. Kapetanovic, and J. R. Smith, "Wispcam: An rf-powered smart camera for machine vision applications," in *Proceedings of the 4th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, pp. 19–22, ACM, 2016.

[27] S. Naderiparizi, Y. Zhao, J. Youngquist, A. P. Sample, and J. R. Smith, "Self-localizing battery-free cameras," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 445–449, ACM, 2015.

[28] A. Sample, D. Yeager, P. Powledge, A. Mamishev, and J. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 2608–2615, Nov. 2008.

[29] S. K. Nayar, D. C. Sims, and M. Fridberg, "Towards self-powered cameras," in *Computational Photography (ICCP), 2015 IEEE International Conference on*, pp. 1–10, IEEE, 2015.

[30] H. Singh, E. D. D. J, H. S. Rawat, and R. George, *Fundamentals of EM Design of Radar Absorbing Structures (RAS)*. Springer, 2018.

[31] J. de Winkel, C. D. Donne, K. S. Yildirim, P. Pawełczak, and J. Hester, "Reliable timekeeping for intermittent computing," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM.

[32] L. Sigrist, R. Ahmed, A. Gomez, and L. Thiele, "Harvesting-aware optimal communication scheme for infrastructure-less sensing," *ACM Trans. Internet Things*, vol. 1, no. 4, 2020.

[33] K. Geissdoerfer and M. Zimmerling, "Bootstrapping battery-free wireless networks: Efficient neighbor discovery and synchronization in the face of intermittency," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 439–455, USENIX Association, 2021.

[34] G. Loubet, A. Takacs, E. Gardner, A. D. Luca, F. Udrea, and D. Dragomirescu, "LoRaWAN battery-free wireless sensors network designed for structural health monitoring in the construction domain," *Sensors*, vol. 19, p. 1510, March 2019.

[35] V. Talla, J. Smith, and S. Gollakota, "Advances and open problems in backscatter networking," vol. 24, no. 4, pp. 32–38.

[36] A. Coustasse, M. Tomblin, and C. Slack, "Impact of radio-frequency identification (rfid) technologies on the hospital supply chain: A literature review," *Perspectives in health information management / AHIMA, American Health Information Management Association*, vol. 10, p. 1d, 10 2013.

[37] M. Katanbaf, V. Jain, and J. R. Smith, "Relacks: Reliable backscatter communication in indoor environments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, pp. 1–24, June 2020.

[38] 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 the 2018 Conference of the ACM Special Interest Group on Data Communication - SIGCOMM18*, ACM Press, 2018.

[39] G. Pollini, "Trends in handover design," *IEEE Communications Magazine*, vol. 34, pp. 82–90, March 1996.

[40] R. Arshad, H. ElSawy, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Handover management in dense cellular networks: A stochastic geometry approach," in *2016 IEEE International Conference on Communications (ICC)*, IEEE, May 2016.

[41] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/WiFi handover with multipath TCP," in *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design - CellNet '12*, ACM Press, 2012.

[42] A. Croitoru, D. Niculescu, and C. Raiciu, "Towards wifi mobility without fast handover," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, (Oakland, CA), pp. 219–234, USENIX Association, May 2015.

[43] J. Mauro, K. B. Mathews, and E. S. Sredzinski, "Effect of a smart pill bottle and pharmacist intervention on medication adherence in patients with multiple myeloma new to lenalidomide therapy," *Journal of Managed Care & Specialty Pharmacy*, vol. 25, pp. 1244–1254, Nov. 2019.

[44] D. C. Klonoff and D. Kerr, "Smart pens will improve insulin therapy," *Journal of Diabetes Science and Technology*, vol. 12, pp. 551–553, Feb. 2018.

[45] M. Katanbaf, A. Saffari, and J. R. Smith, "Receiver selectivity limits on bistatic backscatter range," in *2020 IEEE International Conference on RFID (RFID)*, IEEE, Sep. 2020.

[46] A. Saffari, M. Hessar, S. Naderiparizi, and J. R. Smith, "Battery-free wireless video streaming camera system," in *2019 IEEE International Conference on RFID (RFID)*, IEEE, April 2019.

[47] A. Saffari, S. Y. Tan, M. Katanbaf, H. Saha, J. R. Smith, and S. Sarkar, "Battery-free camera occupancy detection system," in *Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning*, EMDL'21, (New York, NY, USA), p. 13–18, Association for Computing Machinery, 2021.

[48] A. Varshney, C. Pérez-Penichet, C. Rohner, and T. Voigt, "Lorea: A backscatter architecture that achieves a long communication range," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems - SenSys 17*, ACM Press, 2017.

[49] M. Katanbaf, A. Weinand, and V. Talla, "Simplifying backscatter deployment: Full-duplex lora backscatter," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 955–972, USENIX Association, 2021.

[50] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, "Backfi: High throughput wifi backscatter," *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 283–296, Aug. 2015.

[51] J. F. Ensworth, A. T. Hoang, T. Q. Phu, and M. S. Reynolds, "Full-duplex bluetooth low energy (BLE) compatible backscatter communication system for mobile devices," in *2017 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, IEEE, Jan. 2017.

[52] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*, ACM Press, 2016.

[53] "Electronic Code of Federal Regulations, Title 47, Chapter i, Subchapter A, Part 15, Subpart C, section 15.247," 2019. www.ecfr.gov.

[54] A. Lazaro, D. Girbau, and D. Salinas, "Radio link budgets for UHF RFID on multipath environments," *IEEE Transactions on Antennas and Propagation*, vol. 57, pp. 1241–1251, April 2009.

[55] D. Kim, M. Ingram, and W. Smith, "Measurements of small-scale fading and path loss for long range RF tags," *IEEE Transactions on Antennas and Propagation*, vol. 51, pp. 1740–1749, Aug. 2003.

[56] "Wireless insite 3d wireless prediction software," 2021.

[57] "Tap software," 2021.

[58] H. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, pp. 254–256, May 1946.

[59] I. Rodriguez, H. C. Nguyen, N. T. K. Jorgensen, T. B. Sorensen, and P. Mogensen, "Radio propagation into modern buildings: Attenuation measurements in the range from 800 MHz to 18 GHz," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, IEEE.

[60] A. Asp, Y. Sydorov, M. Valkama, and J. Niemela, "Radio signal propagation and attenuation measurements for modern residential buildings," in *2012 IEEE Globecom Workshops*, IEEE.

[61] A. Asp, Y. Sydorov, M. Keskikastari, M. Valkama, and J. Niemela, "Impact of modern construction materials on radio signal propagation: Practical measurements and network planning aspects," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, IEEE.

[62] S. Zhang and N. Pan, "Supercapacitors performance evaluation," *Advanced Energy Materials*, vol. 5, p. 1401401, Dec. 2014.

[63] Poonam, K. Sharma, A. Arora, and S. Tripathi, "Review of supercapacitors: Materials and devices," *Journal of Energy Storage*, vol. 21, pp. 801–825, Feb. 2019.

[64] "Python socket.io project," 2021.

[65] A. Devaraj, M. Megahed, Y. Liu, A. Ramachandran, and T. Anand, "A switched capacitor multiple input single output energy harvester (solar + piezo) achieving 74.6% efficiency with simultaneous MPPT," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, pp. 4876–4887, Dec. 2019.

[66] A. Quelen, G. Pillonnet, P. Gasnier, F. Rummens, and S. Boisseau, "32.3 electromagnetic mechanical energy-harvester IC with no off-chip component and one switching period MPPT achieving up to 95.9% end-to-end efficiency and 460% energy-extraction gain," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, IEEE, Feb. 2020.

[67] K. R. Sadagopan, J. Kang, S. Jain, Y. Ramadass, and A. Natarajan, "A 365nw -61.5 dBm sensitivity, 1.875 cm2 2.4 GHz wake-up receiver with rectifier-antenna co-design for passive gain," in *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, IEEE, June 2017.

[68] V. Mangal and P. R. Kinget, "28.1 a 0.42nw 434mhz -79.1dbm wake-up receiver with a time-domain integrator," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, IEEE, Feb. 2019.

[69] P.-H. P. Wang, C. Zhang, H. Yang, D. Bharadia, and P. P. Mercier, "20.1 a 28μw IoT tag that can communicate with commodity WiFi transceivers via a single-side-band QPSK backscatter communication technique," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, IEEE, Feb. 2020.

[70] H. Michaels, M. Rinderle, R. Freitag, I. Benesperi, T. Edvinsson, R. Socher, A. Gagliardi, and M. Freitag, "Dye-sensitized solar cells under ambient light powering machine learning: towards autonomous smart sensors for the internet of things," *Chemical Science*, vol. 11, no. 11, pp. 2895–2906, 2020.

[71] "Humidity & temp sensor node for star networks enabling 10+ year coin cell battery life ref design," 2016.

[72] R. Kashyap, "Applications of wireless sensor networks in healthcare," in *Advances in Wireless Technologies and Telecommunication*, pp. 8–40, IGI Global, 2020.

[73] J. F. Ensworth and M. S. Reynolds, "BLE-backscatter: Ultralow-power IoT nodes compatible with bluetooth 4.0 low energy (BLE) smartphones and tablets," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, pp. 3360–3368, Sep. 2017.

[74] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "Freerider: Backscatter communication using commodity radios," in *Proceedings of the 13th International Conference on emerging Networking and Technologies - CoNEXT17*, ACM Press, 2017.

[75] A. Abedi, M. H. Mazaheri, O. Abari, and T. Brecht, "Witag: Rethinking backscatter communication for wifi networks," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks - HotNets '18*, ACM Press, 2018.

[76] J. Zhao, W. Gong, and J. Liu, "Spatial stream backscatter using commodity WiFi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys 18*, ACM Press, 2018.

[77] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Passive-zigbee: Enabling zigbee communication in IoT networks with 1000x+ less power consumption," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems - SenSys '18*, ACM Press, 2018.

[78] W. Gong, L. Yuan, Q. Wang, and J. Zhao, "Multiprotocol backscatter for personal IoT sensors," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, ACM, Nov. 2020.

[79] X. Liu, Z. Chi, W. Wang, Y. Yao, and T. Zhu, "Vmscatter: A versatile {MIMO} backscatter," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pp. 895–909, 2020.

[80] R. Zhao, F. Zhu, Y. Feng, S. Peng, X. Tian, H. Yu, and X. Wang, "OFDMA-enabled wi-fi backscatter," in *The 25th Annual International Conference on Mobile Computing and Networking - MobiCom19*, ACM Press, 2019.

[81] F. Zhu, Y. Feng, Q. Li, X. Tian, and X. Wang, "DigiScatter: efficiently prototyping large-scale ofdma backscatter networks," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ACM, June 2020.

[82] 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*, SIGCOMM '18, (New York, NY, USA), pp. 446–460, ACM, 2018.

[83] A. Galisteo, A. Varshney, and D. Giustiniano, "Two to tango: hybrid light and backscatter networks for next billion devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ACM, June 2020.

[84] J. Zhao, W. Gong, and J. Liu, "Towards scalable backscatter sensor mesh with decodable relay and distributed excitation," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ACM, June 2020.

[85] T. Miller, S. S. Oyewobi, A. M. Abu-Mahfouz, and G. P. Hancke, "Enabling a battery-less sensor node using dedicated radio frequency energy harvesting for complete off-grid applications," *Energies*, vol. 13, no. 20, p. 5402, 2020.

[86] A. Varshney, A. Soleiman, L. Mottola, and T. Voigt, "Battery-free visible light sensing," in *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems*, ACM, Oct. 2017.

[87] M. Magno, X. Wang, M. Eggimann, L. Cavigelli, and L. Benini, "InfiniWolf: Energy efficient smart bracelet for edge computing with dual source energy harvesting," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, March 2020.

[88] J. de Winkel, V. Kortbeek, J. Hester, and P. Pawełczak, "Battery-free game boy," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, pp. 1–34, Sep. 2020.

[89] F. Orfei, C. B. Mezzetti, and F. Cottone, "Vibrations powered LoRa sensor: An electromechanical energy harvester working on a real bridge," in *2016 IEEE SENSORS*, IEEE, Oct. 2016.

[90] X. Li, L. Teng, H. Tang, J. Chen, H. Wang, Y. Liu, M. Fu, and J. Liang, "Vipsn: A vibration-powered iot platform," *IEEE Internet of Things Journal*, 2020.

[91] Y. Song, J. Min, Y. Yu, H. Wang, Y. Yang, H. Zhang, and W. Gao, "Wireless battery-free wearable sweat sensor powered by human motion," *Science Advances*, vol. 6, p. eaay9842, Sep. 2020.

[92] S. Han, J. Kim, S. M. Won, Y. Ma, D. Kang, Z. Xie, K.-T. Lee, H. U. Chung, A. Banks, S. Min, S. Y. Heo, C. R. Davies, J. W. Lee, C.-H. Lee, B. H. Kim, K. Li, Y. Zhou, C. Wei, X. Feng, Y. Huang, and J. A. Rogers, "Battery-free, wireless sensors for full-body pressure and temperature mapping," *Science Translational Medicine*, vol. 10, p. eaan4950, April 2018.

[93] Y. Zhang, Y. Iravantchi, H. Jin, S. Kumar, and C. Harrison, "Sozu: Self-powered radio tags for building-scale activity sensing," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ACM, Oct. 2019.

[94] C. Josephson, L. Yang, P. Zhang, and S. Katti, "Wireless computer vision using commodity radios," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks - IPSN '19*, ACM Press, 2019.

[95] R. A. Potyrailo, N. Nagraj, Z. Tang, F. J. Mondello, C. Surman, and W. Morris, "Battery-free radio frequency identification (RFID) sensors for food quality and safety," *Journal of Agricultural and Food Chemistry*, vol. 60, pp. 8535–8543, Aug. 2012.

[96] J. Kim, P. Gutruf, A. M. Chiarelli, S. Y. Heo, K. Cho, Z. Xie, A. Banks, S. Han, K.-I. Jang, J. W. Lee, K.-T. Lee, X. Feng, Y. Huang, M. Fabiani, G. Gratton, U. Paik, and J. A. Rogers, "Miniaturized battery-free wireless systems for wearable pulse oximetry," *Advanced Functional Materials*, vol. 27, p. 1604373, Nov. 2016.

[97] H. Zhang, P. Gutruf, K. Meacham, M. C. Montana, X. Zhao, A. M. Chiarelli, A. Vázquez-Guardado, A. Norris, L. Lu, Q. Guo, C. Xu, Y. Wu, H. Zhao, X. Ning, W. Bai, I. Kandela, C. R. Haney, D. Chanda, R. W. Gereau, and J. A. Rogers, "Wireless, battery-free optoelectronic systems as subdermal implants for local tissue oximetry," *Science Advances*, vol. 5, p. eaaw0873, March 2019.

[98] S. J. Thomas, R. R. Harrison, A. Leonardo, and M. S. Reynolds, "A battery-free multichannel digital neural/EMG telemetry system for flying insects," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, pp. 424–436, Oct. 2012.

[99] Y. Zhao and J. R. Smith, "A battery-free RFID-based indoor acoustic localization platform," in *2013 IEEE International Conference on RFID (RFID)*, IEEE, April 2013.

[100] T. Li and X. Zhou, "Battery-free eye tracker on glasses," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking - MobiCom '18*, ACM Press, 2018.

[101] X. Lin and B.-C. Seet, "Battery-free smart sock for abnormal relative plantar pressure monitoring," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, pp. 464–473, April 2017.

[102] S. R. Krishnan, C.-J. Su, Z. Xie, M. Patel, S. R. Madhvapathy, Y. Xu, J. Freudman, B. Ng, S. Y. Heo, H. Wang, T. R. Ray, J. Leshock, I. Stankiewicz, X. Feng, Y. Huang,

P. Gutruf, and J. A. Rogers, "Wireless, battery-free epidermal electronics for continuous, quantitative, multimodal thermal characterization of skin," *Small*, vol. 14, p. 1803192, Oct. 2018.

[103] P. Gutruf, R. T. Yin, K. B. Lee, J. Ausra, J. A. Brennan, Y. Qiao, Z. Xie, R. Peralta, O. Talarico, A. Murillo, S. W. Chen, J. P. Leshock, C. R. Haney, E. A. Waters, C. Zhang, H. Luan, Y. Huang, G. Trachiotis, I. R. Efimov, and J. A. Rogers, "Wireless, battery-free, fully implantable multimodal and multisite pacemakers for applications in small animal models," *Nature Communications*, vol. 10, Dec. 2019.

[104] S. Lee, H. Wang, J. Wang, Q. Shi, S.-C. Yen, N. V. Thakor, and C. Lee, "Battery-free neuromodulator for peripheral nerve direct stimulation," *Nano Energy*, vol. 50, pp. 148–158, Aug. 2018.

[105] "Solar-powered hd camera, https://www.amazon.com/powered-wireless-surveillance-outdoor-business/dp/b07wv49x9y," 2021.

[106] S. Y. Tan, H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, "A flexible framework for building occupancy detection using spatiotemporal pattern networks," in *2019 American Control Conference (ACC)*, pp. 5884–5889, IEEE, 2019.

[107] S. Y. Tan, H. Saha, M. Jacoby, A. R. Florita, G. P. Henze, and S. Sarkar, "Granger causality-based hierarchical time series clustering for state estimation," *IFAC-PapersOnLine*, vol. 51, no. 1, 2020.

[108] L. Wu and Y. Wang, "A low-power electric-mechanical driving approach for true occupancy detection using a shuttered passive infrared sensor," *IEEE Sensors Journal*, vol. 19, no. 1, pp. 47–57, 2018.

[109] J. Andrews, M. Kowsika, A. Vakil, and J. Li, "A motion induced passive infrared (pir) sensor for stationary human occupancy detection," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1295–1304, 2020.

[110] O. A. G. González, A. M. P. Soberanes, V. H. G. Ortega, and J. C. S. Savedra, "Embedded system for human detection applied to domotics," *Research in Computing Science*, vol. 148, pp. 103–115, 2019.

[111] L. He, Y. Wang, S. Velipasalar, and M. C. Gursoy, "Human detection using mobile embedded smart cameras," in *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–6, IEEE, 2011.

[112] S. Saypadith, W. Ruangsang, and S. Aramvith, "Optimized human detection on the embedded computer vision system," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1707–1711, IEEE, 2017.

[113] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, Marc, Y. Kwon, oleg, wanghaoyang0106, Y. Defretin, A. Lohia, ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, Doug, Durgesh, and F. Ingham, "ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," 2021.

[114] J. Wang, J. Huang, Z. Feng, S.-J. Cao, and F. Haghighat, "Occupant-density-detection based energy efficient ventilation system: Prevention of infection transmission," *Energy and Buildings*, vol. 240, p. 110883, 2021.

[115] I. Mutis, A. Ambekar, and V. Joshi, "Real-time space occupancy sensing and human motion analysis using deep learning for indoor air quality control," *Automation in Construction*, vol. 116, p. 103237, 2020.

[116] R. Schrijvers, S. Puttemans, T. Callemein, and T. Goedemé, "Real-time embedded person detection and tracking for shopping behaviour analysis," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 541–553, Springer, 2020.

[117] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[118] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2020.

[119] H. Saha, S. Y. Tan, A. Saffari, M. Katanbaf, J. R. Smith, and S. Sarkar, "Few shot clustering for indoor occupancy detection with extremely low-quality images from battery free cameras," 2020.

[120] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, p. 21–37, 2016.

[121] V. Iyer, A. Najafi, J. James, S. Fuller, and S. Gollakota, "Wireless steerable vision for live insects and insect-scale robots," *Science Robotics*, vol. 5, no. 44, 2020.

[122] M. Katanbaf, A. Saffari, and J. R. Smith, "Multiscatter: Multistatic backscatter networking for battery-free sensors," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, (New York, NY, USA), p. 69–83, Association for Computing Machinery, 2021.

[123] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019.

[124] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[125] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018.

[126] B. Veluri, A. Saffari, C. Pernu, J. Smith, M. Taylor, and S. Gollakota, "Neuricam: Video super-resolution and colorization using key frames," 2022.

[127] C. Josephson, L. Yang, P. Zhang, and S. Katti, "Wireless computer vision using commodity radios," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, IPSN '19, (New York, NY, USA), p. 229–240, Association for Computing Machinery, 2019.

[128] "Investigating power consumption of blink." https://cam-do.com/blogs/camdo-blog/investigating-power-consumption-of-blink.